



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

Control y supervisión para un prototipo
de proceso de embotellado bajo el
estándar IEC 61131-3

TESIS

Que para obtener el título de
Ingeniero Eléctrico Electrónico

PRESENTA

Braulio Yair Raygoza Pérez

DIRECTOR DE TESIS

Dr. Hoover Mujica Ortega



Ciudad Universitaria, Cd. Mx., 2024

Jurado asignado

Presidente: Mtra. Gloria Correa Palacios
Secretario: Dr. José Daniel Castro Díaz
Vocal: Dr. Hoover Mujica Ortega
1^{er} suplente: Ing. Jeshua Perea Constantino
2^{do} suplente: Ing. David Quintanar Villalba

Ciudad Universitaria, Departamento de Control y Robótica, Laboratorio de
Automatización.

Ciudad de México.

Director de tesis

Dr. Hoover Mujica Ortega

Dedicatoria

A mis padres Jacinto y Adela, por el apoyo y amor incondicional que siempre me han dado.

Agradecimientos

A mi familia; la cual siempre me ha dado el apoyo para ser la mejor versión de mi.

A mis amigos y compañeros; que sin ellos la vida como estudiante hubiera sido mucho más complicada.

A los buenos profesores; por sembrar la semilla del conocimiento y mejorar mi formación académica y profesional.

A la vida misma; por darme y permitirme compartir con aquellos que tanto aprecio.

Resumen

En la actualidad, el rápido desarrollo e implementación de nuevas tecnologías están llevando a las industrias a buscar constantemente formas de mejorar su eficiencia y competitividad en un entorno empresarial cada vez más dinámico y desafiante. En este contexto, la adopción de prácticas de manufactura flexible, reconfigurable y sostenible se vuelven fundamentales. Estas prácticas no solo dependen de una infraestructura sólida, sino también de sistemas de control robustos que garanticen el óptimo funcionamiento de los procesos.

El presente trabajo describe la implementación de un sistema de control y supervisión para el módulo de reciclaje de una maqueta prototipo que representa un proceso de embotellado industrial. El módulo en cuestión está conformado por un brazo robótico de 4 grados de libertad con una pinza como efector final, 3 bandas transportadoras, un riel con carro transportador, sensores fotoeléctricos y de límite, pistones de doble y simple efecto con retorno por muelle. El proceso de reciclaje consiste en usar el robot para mover una caja con 4 botellas a la banda transportadora, la cual las llevará hacia la zona de vaciado simulada con pistones, para finalmente llegar a la zona de reciclado donde el robot se encargará de separar las botellas de la caja para reintegrarlas al proceso.

Por otra parte, se diseñó un sistema de control de movimiento para las bandas transportadoras, con el fin de evitar movimientos bruscos que puedan comprometer la calidad de los productos, reducir el desgaste mecánico en las bandas y optimizar el consumo energético. Además, la implementación de un sistema de monitoreo del consumo energético proporciona datos cruciales con el potencial de convertirse en información accionable que permita tomar decisiones más inteligentes y eficientes.

Para lograr estos objetivos, se utilizó el estándar industrial IEC 61131 para el diseño de la arquitectura de control, la metodología GRAFCET para la implementación de las rutinas de control y View of Things, una de las tecnologías de visualización más recientes, para el desarrollo de la interfaz de supervisión. Todo lo anterior respaldado por un previo y exhaustivo estudio del prototipo.

Tras la implementación del sistema de control y supervisión, se logró no solo la automatización y optimización del módulo en cuestión, sino también la capacidad de monitorear y gestionar el proceso desde cualquier dispositivo móvil gracias a la interfaz de supervisión web implementada. Además, con estas acciones, se consiguió validar con éxito la arquitectura de control propuesta, demostrando su eficacia y fiabilidad para entornos industriales a mayor escala.

Índice general

Índice de figuras	viii
Índice de tablas	xii
Acrónimos	xiii
1. Introducción	1
1.1. Motivación	2
1.2. Antecedentes	3
1.3. Formulación del problema	7
1.4. Objetivo	9
1.4.1. Objetivos particulares	9
1.5. Contribuciones	9
1.6. Organización de la tesis	10
2. Marco Teórico	11
2.1. Celda de manufactura robotizada	11
2.2. Arquitectura de control	11
2.3. Topología de red	12
2.3.1. Topología tipo bus	12
2.3.2. Topología tipo estrella	12
2.3.3. Topología tipo árbol	13
2.3.4. Topología tipo anillo	13
2.4. Protocolo PROFINET	14
2.5. Sistemas de control secuencial	14
2.6. Metodología GRAFCET	15
2.7. Estándar IEC 61131-3	15
2.8. Interfaz Humano Máquina	16
2.9. Controlador SIMATIC S7-1500	16
2.10. Periferia descentralizada SIMATIC ET 200SP	17
2.11. Switch Ethernet SCALANCE X308	18
2.12. Brazo robótico Dobot Magician	19
2.13. Módulo de expansión 24V Industrial I/O Shield	20
2.14. Controlador de motor DC EM-170	21
2.15. Transformador de corriente MES-62/30	22
2.16. Transductor de corriente LTS 15-NP	23

2.17. Entorno de desarrollo TIA Portal	24
2.17.1. WinCC Unified View of Things	24
3. Desarrollo	27
3.1. Levantamiento técnico	27
3.1.1. Dispositivos de automatización	27
3.1.2. Actuadores	28
3.1.3. Sensores	30
3.1.4. Otros dispositivos eléctricos y electrónicos	32
3.1.5. Tabla de ordenamiento de variables	32
3.1.6. Diagramas eléctricos	34
3.2. Diseño e implementación de la arquitectura de control	35
3.3. Pasarela de conexión	36
3.4. Descripción funcional del prototipo	37
3.5. Diseño e implementación del sistema de control y supervisión	40
3.5.1. Rutina de control LD	40
3.5.2. Rutina de control SFC	47
3.5.3. Rutina de control para movimiento de motores DC	55
3.5.4. HMI web	60
4. Resultados	62
4.1. Nivel de campo	62
4.2. Nivel de control	64
4.3. Nivel de supervisión	86
5. Conclusiones y trabajo futuro	88
5.1. Trabajo a futuro	89
Apéndice A. Diagramas eléctricos	90
Apéndice B. Rutinas de control	95
B.1. Diagrama escalera	95
B.1.1. Arranque suave	103
B.2. Gráfico de funciones secuenciales	106
Referencias	109

Índice de figuras

1.1. Modelo de simulación de producción industrial 4.0	4
1.2. Arquitectura de control para un proceso de fermentación de cerveza	5
1.3. Arquitectura de control para un sistema de bombeo	6
1.4. Sistema automático de clasificación y transporte de frutas y verduras basado en el brazo robótico Dobot Magician	7
1.5. Vista superior del módulo 4	8
1.6. Vista inferior del módulo 4	8
2.1. Topología tipo bus	12
2.2. Topología tipo estrella	12
2.3. Topología tipo árbol	13
2.4. Topología tipo anillo	14
2.5. PLC SIMATIC S7-1500	17
2.6. Periferia descentralizada SIMATIC ET 200SP	18
2.7. Switch Ethernet SCALANCE X308	19
2.8. Brazo robótico Dobot Magician	20
2.9. Módulo de expansión de Arduino 24V Industrial I/O Shield	21
2.10. Controlador de motor DC EM-170	22
2.11. Transformador de corriente MES-62/30	23
2.12. Transductor de corriente LTS 15-NP	24
2.13. Comparación de objetos en VoT y WinCC Unified PC	26
3.1. Periferia descentralizada ET 200SP del módulo 4	28
3.2. Motor 24VDC	29
3.3. Electroválvulas de 3 vías y 2 posiciones	29
3.4. Sensor fotoeléctrico SICK	30
3.5. Sensor de límite	31
3.6. Encoder incremental para motor	31
3.7. Diagrama de conexiones generales del módulo 4	34
3.8. Arquitectura de control del módulo 4	35
3.9. Pasarela de conexión	36
3.10. Empaque de botellas terminado en la base de recepción	38
3.11. Empaque en la banda transportadora superior	38
3.12. Vaciado de botellas	38
3.13. Caja y botellas separadas	39

3.14. Distribución de cajas y botellas vacías	39
3.15. Cinta transportadora en la posición de la base de recepción	39
3.16. Peldaño 1 de la rutina LD	40
3.17. Peldaño 2 de la rutina LD	40
3.18. Peldaño 3 de la rutina LD	41
3.19. Peldaño 4 de la rutina LD	41
3.20. Peldaño 5 de la rutina LD	41
3.21. Peldaño 6 de la rutina LD	42
3.22. Peldaño 7 de la rutina LD	42
3.23. Peldaño 8 de la rutina LD	43
3.24. Peldaño 9 de la rutina LD	43
3.25. Peldaño 10 de la rutina LD	43
3.26. Peldaño 11 de la rutina LD	44
3.27. Peldaño 12 de la rutina LD	45
3.28. Peldaño 13 de la rutina LD	46
3.29. Peldaño 14 de la rutina LD	46
3.30. Peldaño 15 de la rutina LD	46
3.31. Peldaño 16 de la rutina LD	46
3.32. Peldaño 17 de la rutina LD	47
3.33. Estado y transición 1 de la rutina SFC	48
3.34. Estado y transición 2 de la rutina SFC	49
3.35. Estado y transición 3 de la rutina SFC	49
3.36. Estado y transición 4 de la rutina SFC	50
3.37. Estado y transición 5 de la rutina SFC	50
3.38. Paso y transición 6 de la rutina SFC	51
3.39. Estado y transición 7 de la rutina SFC	51
3.40. Estado y transición 8 de la rutina SFC	52
3.41. Estado y transición 9 de la rutina SFC	52
3.42. Estado y transición 10 de la rutina SFC	53
3.43. Estado y transición 11 de la rutina SFC	53
3.44. Paso y transición 12 de la rutina SFC	54
3.45. Estado y transición 13 de la rutina SFC	54
3.46. Estado y transición 14 de la rutina SFC	55
3.47. Comportamiento en la velocidad del motor en función del número de muestras que se envían al controlador y el voltaje que recibe el motor	56
3.48. Peldaño 1 de la rutina de control de movimiento	58
3.49. Peldaño 2 de la rutina de control de movimiento	58
3.50. Peldaño 3 de la rutina de control de movimiento	58
3.51. Peldaño 4 de la rutina de control de movimiento	59
3.52. Peldaño 5 de la rutina de control de movimiento	59
3.53. Peldaño 6 de la rutina de control de movimiento	59
3.54. Peldaño 7 de la rutina de control de movimiento	60
3.55. Peldaño 8 de la rutina de control de movimiento	60
3.56. Interfaz de control y supervisión web	61

4.1. Interfaz principal del software DobotStudio	63
4.2. Botón Home en DobotStudio	63
4.3. Panel de operación en DobotStudio	72
4.4. Configuración manual de IP en una computadora	72
4.5. Selección, identificación y conexión con el PLC	73
4.6. Configuración de protección	74
4.7. Configuración de acceso	74
4.8. Configuración de la periferia	75
4.9. Configuración de IP de la periferia	76
4.10. Configuración de los módulos de entradas digitales	76
4.11. Configuración del módulo de salidas digitales	77
4.12. Configuración del módulo de salidas analógicas	77
4.13. Configuración del módulo medidor de energía	78
4.14. Configuración del módulo de entradas analógicas	78
4.15. Comunicación del PLC y la periferia	79
4.16. Compilación del hardware y software de la periferia	79
4.17. Compilación del hardware y software del PLC	80
4.18. Descarga de la configuración de hardware de la periferia	80
4.19. Descarga al PLC	81
4.20. Descarga de la configuración de hardware del PLC	81
4.21. PLC en línea	82
4.22. Upload en el PLC	82
4.23. Configuración del proyecto terminada	83
4.24. Activación del servidor web en el PLC	83
4.25. Creación de usuarios	84
4.26. Exportación del certificado	84
4.27. Instalación del certificado web	84
4.28. Comparación de corriente con arranque suave y directo	85
4.29. HMI web resultante	86
4.30. HMI web en dispositivos móviles	87
A.1. Vista inferior de los dispositivos del módulo 4	90
A.2. Diagrama de conexiones generales del módulo 4	91
A.3. Diagrama de conexiones de dispositivos del módulo 4	92
A.4. Diagrama de conexiones de alimentación del módulo 4	93
A.5. Diagrama de conexiones de neutros y tierras del módulo 4	94
B.1. Rutina LD parte 1	95
B.2. Rutina LD parte 2	96
B.3. Rutina LD parte 3	97
B.4. Rutina LD parte 4	98
B.5. Rutina LD parte 5	99
B.6. Rutina LD parte 6	100
B.7. Rutina LD parte 7	101
B.8. Rutina LD parte 8	102

B.9. Arranque suave parte 1	103
B.10. Arranque suave parte 2	104
B.11. Arranque suave parte 3	105
B.12. Rutina SFC parte 1	106
B.13. Rutina SFC parte 2	107
B.14. Rutina SFC parte 3	108

Índice de tablas

2.1. Diferencias entre las soluciones HMI del WinCC Unified System	26
3.1. Dispositivos de automatización del módulo 4	28
3.2. Actuadores del módulo 4	29
3.3. Sensores del módulo 4	30
3.4. Otros dispositivos eléctricos y electrónicos del módulo 4	32
3.5. Entradas digitales	33
3.6. Salidas digitales	33
3.7. Entradas analógicas	34
3.8. Salidas analógicas	34
3.9. Configuraciones de la bobina primaria del transductor LEM	45
3.10. Puntos a interpolar con el polinomio de Lagrange	57
3.11. Calculo de muestras para los diferentes tiempos de rampa	58

Acrónimos

- PLC** Controlador Lógico Programable. 1, 2, 3, 4, 5, 9, 10, 15, 16, 20, 27, 36, 37, 64, 72, 73, 80, 81, 82, 83, 88, 89
- IEC** Comisión Electrotécnica Internacional. 1, 2, 9, 10, 14, 15, 40, 88
- LD** Diagrama de Escalera. 1, 9, 10, 15, 27, 40, 45, 47, 55, 58, 88
- FBD** Diagrama de Bloques de Funciones. 1, 15
- IL** Lista de Instrucciones. 1, 15
- ST** Texto Estructurado. 1, 15
- SFC** Gráfico de Funciones Secuenciales. 1, 2, 9, 10, 15, 27, 40, 42, 55, 60, 88
- GRAFCET** Gráfico Funcional de Control de Etapas y Transiciones. 2, 10, 15, 88
- RTU** Unidad Terminal Remota. 5
- UPS** Sistema de Alimentación Ininterrumpida. 5
- HMI** Interfaz Humano Máquina. 5, 16, 24, 25, 41, 47, 60, 88, 89
- E/S** Entradas y Salidas. 14, 17
- DCS** Sistema de Control Distribuido. 16
- TIA Portal** Portal de Automatización Totalmente Integrado. 24, 60, 72
- VoT** Visión de las Cosas. 24, 25, 27, 36, 43, 60, 86
- HTTPS** Protocolo Seguro de Transferencia de Hipertexto. 86
- SSL** Capa de Puertos Seguros. 86
- TLS** Seguridad de la Capa de Transporte. 86

Capítulo 1

Introducción

La automatización industrial, tal como se conoce hoy en día ha sido producto de la necesidad de mejorar la eficiencia en los procesos, misma que comprende mejorar la calidad de los productos, abaratar costos de fabricación, así como evitar exponer al ser humano a actividades peligrosas, entre otros. La automatización de procesos industriales, tal como lo menciona [García Moreno, 2014], es uno de los principales objetivos de las empresas que buscan ser cada vez más competitivas en un entorno cambiante y agresivo. Es aquí donde surge la necesidad de tener un sistema de control y supervisión que permita garantizar el óptimo control y desempeño de un proceso.

Un sistema de control robusto busca añadir elementos, tanto físicos como digitales, para lograr su objetivo. Que dicho sea de paso, es tener la capacidad de reaccionar ante situaciones previstas como a las imprevistas, con el fin de situar al proceso y al personal que lo asiste en una condición más favorable [García Moreno, 2014].

Por otra parte, la aparición del Controlador Lógico Programable (PLC, por sus siglas en inglés) representó un antes y después para la automatización industrial, pues debido a sus atributos han sido ampliamente usados en la industria para mejorar la productividad. Actualmente es difícil encontrar una industria que no utilice un PLC de un modo u otro. El éxito de estos dispositivos se debe a que son confiables, flexibles, adaptables y sobretodo programables, ya que anteriormente un cambio de lógica requería de un recableado, lo que era muy tardado y costoso [Ramanathan, 2014].

En un principio no existía conceptos o metodología para programar PLC, lo que provocó múltiples contratiempos como la incompatibilidad entre los sistemas. Para superar estos contratiempos, en 1993 se creó la norma Comisión Electrotécnica Internacional (IEC, por sus siglas en inglés) 61131 para estandarizar la programación de PLC, proporcionando un lenguaje común e independiente del fabricante. Dentro de la norma destaca su tercera parte, la IEC 61131-3, donde se especifica la sintaxis y semántica de dos lenguajes gráficos, el Diagrama de escalera (LD, por sus siglas en inglés) y el Diagrama de bloques de funciones (FBD, por sus siglas en inglés), también dos lenguajes textuales, la Lista de instrucciones (IL, por sus siglas en inglés) y el Texto estructurado (ST, por sus siglas en inglés) y uno secuencial, el Gráfico de funciones secuenciales (SFC, por sus siglas en inglés). Cabe destacar que este estándar no solo define a los lenguajes de programación, sino que también proporciona tipos de datos comunes que pueden ser compartidos entre estos lenguajes.

Ahora, hay que destacar que el hablar de control secuencial es sinónimo de automatización secuencial. Este tipo de automatismo, como lo explica [García Moreno, 2014], consiste en un sistema cuyo funcionamiento emplea una secuencia de fases claramente diferenciadas con condiciones de transición lógicas que permiten efectuar cambios de estado en el sistema controlado. Específicamente,

como describe [Ramanathan, 2014], SFC es un lenguaje gráfico de alto nivel basado en la red de Petri, con mucha similitud al estándar francés Gráfico Funcional de Control de Etapas y Transiciones (GRAF CET, por sus siglas en francés), el cual representa a los principales estados de un proceso, el cambio de estados y la razón detrás del mismo. La mayor ventaja de este lenguaje gráfico es que posee la capacidad de ser integrado a otros lenguajes de programación del estándar IEC 61131-3.

Finalmente, es muy importante mencionar las implicaciones de automatizar un proceso, pues van más allá de solo mejorar su eficiencia. Al encomendar a un sistema de control una parte o la totalidad de un proceso, la automatización se posiciona en un nivel superior al del proceso mismo. Actualmente la automatización se encuentra en un marco técnico-económico donde intervienen especialistas de diferentes rubros, dando así equipos de trabajo multidisciplinarios que intervienen no solo en el proceso, sino también en la distribución y gestión de los productos o servicios ofrecidos tanto a nivel industrial, comercial y administrativo [García Moreno, 2014]. Esto sin mencionar el rápido desarrollo e implementación de nuevas tecnologías que permiten una manufactura avanzada promovidas por la industria 4.0.

1.1. Motivación

La seguridad en un proceso es uno de los principales objetivos que buscan las empresas. Está seguridad es la expectativa de que el sistema no representa un riesgo para la vida en general. Por otra parte, la seguridad funcional es la capacidad de un sistema para realizar acciones que garanticen un estado seguro en el proceso controlado. La programación bajo el estándar IEC 61131-3 es altamente segura, pues este estándar ha sido certificado con éxito para su uso en aplicaciones de seguridad en diferentes tipos de sistemas, ya que ha aprobado los requisitos del estándar IEC 61508, el cual cubre la seguridad funcional de los sistemas eléctricos, electrónicos y electrónicos programables [Toon, 2002].

En la actualidad, el estándar IEC 61131-3 se ha convertido en la piedra angular para el desarrollo tanto de hardware como de software en numerosas empresas líderes, tales como Siemens, Rockwell Automation y Schneider Electric. Este estándar no solo ha sido adoptado por gigantes de la industria, sino que también en una variedad de sectores, desde la automotriz, química, petroquímica, metalúrgica, producción de energía, alimentaria y farmacéutica, entre otros. Lo más destacable de este estándar es su capacidad para asegurar la compatibilidad de las técnicas de programación, independientemente del hardware o software utilizado, o de la industria en la que se aplique. Esto significa que los profesionales pueden confiar en que sus habilidades y conocimientos en programación serán transferibles y aplicables en una amplia gama de contextos industriales.

Del mismo modo, los PLC son ampliamente utilizados en diversas industrias en México, en la industria automotriz en empresas como General Motors, Ford, Volkswagen [INEGI, 2016], alimentaria como Bimbo, Nestlé, PepsiCo y de producción de energía como CFE, son un ejemplo. Estos dispositivos son fundamentales para la automatización de procesos, el control de líneas de producción, el monitoreo de sistemas y la mejora de la eficiencia operativa. Algunas de las aplicaciones específicas de los PLC en la industria incluyen el control de procesos de producción, llenado y envasado en la industria alimentaria, así como el control de múltiples procesos simultáneos en la industria automotriz.

Actualmente las empresas se encuentran con grandes retos debido al sistema de producción complejo y cambiante, en respuesta a la creciente necesidad de productos innovadores y personalizados. De acuerdo con [Hassan Reza, *et al.*, 2021], las empresas requieren una excelente conectividad y congruencia con sus procesos para resolver estos problemas de manera efectiva. Esto se logra con la

implementación de la digitalización y la industria 4.0. A su vez esto implica la implementación de tecnologías emergentes que permitan la reestructuración de la industria convencional para tener acceso a información en tiempo real acerca de los procesos de producción y logística, y así proporcionar una excelente exposición de las empresas.

Por otra parte, con el crecimiento económico global y la búsqueda constante de eficiencia han impulsado a los países más industrializados a promover la producción de bienes cerca de los mercados de destino. Este enfoque ha generado la adopción de estrategias como la sustitución de importaciones y la deslocalización como medios efectivos para reducir costos [Quiroz Cuenca, 2021]. Estas estrategias no sólo responden a la necesidad de minimizar los gastos de transporte, sino que también reflejan un cambio hacia una producción más ágil y adaptable a las demandas del mercado local, además de fomentar la creación de empleo y la transferencia de conocimientos tecnológicos, contribuyendo así al fortalecimiento de las economías locales y a una mayor estabilidad en el ámbito global.

Actualmente, México vive un momento de oportunidad para su desarrollo industrial, pues la integración comercial con America del Norte a través del *nearshoring* de las cadenas de producción hoy es una realidad que permitirá diversificar nuestras industrias así como incursionar en otras, con el objetivo de fortalecer nuestra habilidad para innovar, desarrollar tecnología, formar y capacitar a nuestro personal, establecer nuevos parques industriales y brindar un apoyo crucial a las pequeñas y medianas empresas, dada su relevancia en nuestra economía. En general, la relocalización en nuestro país implica la oportunidad de impulsar la exportación, la inversión productiva, la generación de empleos y la mejora de los niveles de vida en la población [Hacienda, 2023].

1.2. Antecedentes

En 2019, un equipo de ingenieros de la Universidad Tecnológica de Rajamangala, en Tailandia, desarrolló un avanzado modelo de simulación para la producción industrial 4.0, utilizando un PLC como elemento clave [Luejai, *et al.*, 2019]. Este proyecto consta de un prototipo equipado con motores, transportadores, sensores fotoeléctricos e inductivos, así como pistones de simple y doble efecto, entre otros elementos. Estos componentes permitieron simular con precisión un proceso de suministro de productos a través de un transportador, con la capacidad de interactuar tanto a través de una pantalla táctil como de un sistema en línea mediante un teléfono inteligente, como se muestra en la Figura 1.1. Los resultados obtenidos revelaron que este modelo de simulación puede operar de manera altamente eficiente y gestionar efectivamente los desafíos propios de la automatización de procesos en la industria real. Esto sugiere la viabilidad de su implementación como parte integral de sistemas de control en entornos industriales reales.



Figura 1.1 Modelo de simulación de producción industrial 4.0

Por otra parte, con el rápido avance de la tecnología automatizada en la industria, la demanda de sistemas de control automático ha experimentado un crecimiento acelerado en múltiples sectores, entre ellos la industria alimentaria. Un ejemplo notable se evidencia en la labor de ingenieros chinos de la Universidad Tecnológica de Henan, en 2015, quienes en respuesta a la creciente necesidad de asegurar la calidad en la producción de cerveza, desarrollaron un sistema de control y supervisión automático para el proceso de fermentación, basado en un controlador SIMATIC [Xiaodong, *et al.*, 2015]. El sistema de control propuesto consta de tres estaciones; la estación de operador, la estación de ingeniería y la estación de control. De donde el núcleo del sistema es el PLC SIMATIC S7-414. La arquitectura de control propuesta se muestra en la Figura 1.2 y el algoritmo de control consistió en un control PID de temperatura. Como resultados finales, los ingenieros obtuvieron que el sistema de control basado en el PLC SIMATIC es más flexible y confiable en comparación a los sistemas ya existentes en las cervecerías, aportando así un gran valor práctico a la mejora de este proceso industrial.

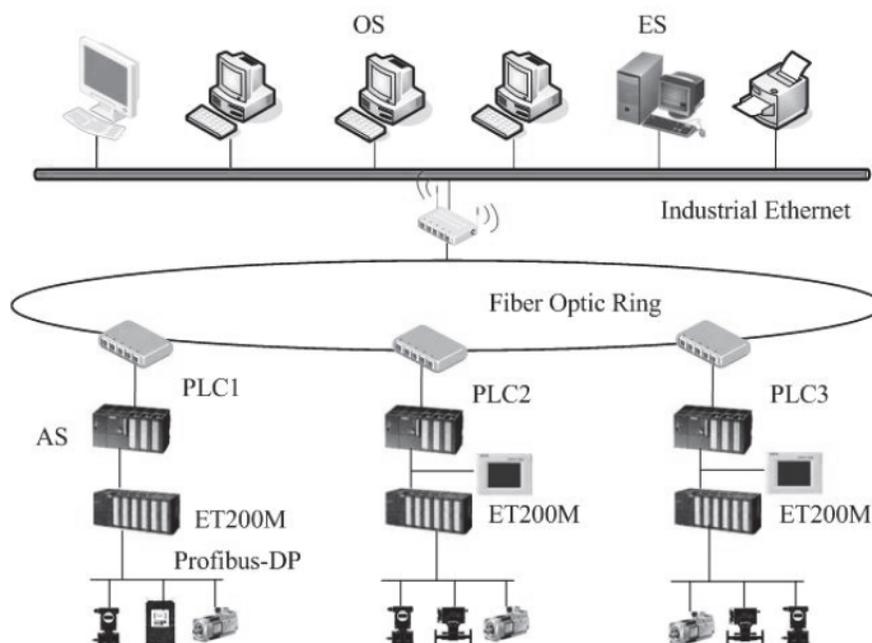


Figura 1.2 Arquitectura de control para un proceso de fermentación de cerveza

En el año 2021, ingenieros de la Universidad Técnica de Gheorghe Asachi en Rumania desarrollaron un sistema de control y supervisión remoto para un sistema de bombeo de agua situado a una distancia considerable de 50 metros de la sala de control. Este innovador sistema se basa en un PLC 1200 y una periferia descentralizada ET 200SP, utilizando el protocolo Modbus Unidad Terminal Remota (RTU, por sus siglas en inglés) como medio de comunicación [Gabor y Livint, 2021]. El sistema de bombeo consta de un motor de 7.5 [kW] accionado por un arrancador suave alimentado de un contactor trifásico con un interruptor de protección tripolar, hay que destacar que el arrancador cuenta con un puerto Modbus RTU para su conexión directa con la periferia. Adicionalmente, se dispone de un Sistema de Alimentación Ininterrumpida (UPS, por sus siglas en inglés) de respaldo eléctrico con salida a 220 [V] para el arrancador y la periferia. Por otra parte, en el cuarto de control se tiene al controlador S7-1200, un panel para una Interfaz Humano Máquina (HMI, por sus siglas en inglés) y un switch Ethernet, como se muestra en la Figura 1.3. Finalmente, los resultados obtenidos revelaron la eficacia sobresaliente de este sistema de control y supervisión. La capacidad de enviar y recibir información de manera remota a través de un único par de cables, gracias al protocolo Modbus RTU y la periferia descentralizada, no solo simplifica la instalación eléctrica de cada equipo de planta, sino que también reduce significativamente el tiempo y los costos asociados con la mano de obra. Este enfoque innovador demuestra cómo la tecnología puede optimizar los procesos industriales, ofreciendo soluciones eficientes y rentables para el monitoreo y control de sistemas incluso en entornos distantes y desafiantes.

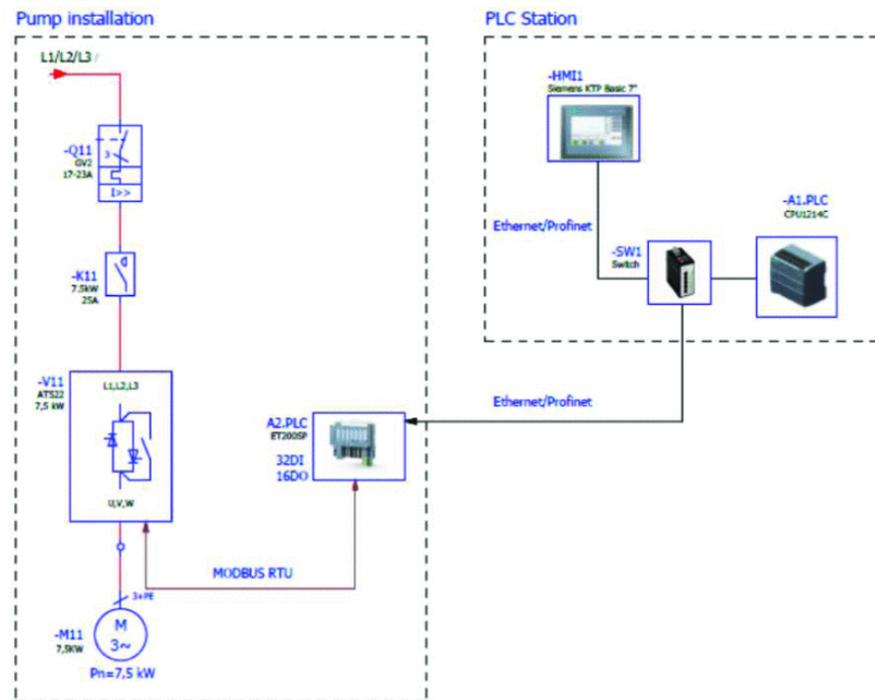


Figura 1.3 Arquitectura de control para un sistema de bombeo

Por otra parte, en los últimos años el desarrollo de la robótica en la automatización industrial ha sido impulsada por avances tecnológicos, desde los primeros manipuladores teleoperados hasta los modernos cobots y la integración en entornos de la Industria 4.0. Este proceso ha transformado la forma en que se llevan a cabo las tareas industriales, mejorando la eficiencia y la seguridad en diversos sectores. Un ejemplo de esto es el sistema automático de clasificación y transporte de frutas y verduras basado en el brazo robótico Dobot Magician desarrollado por ingenieros de la Universidad Tecnológica de Tianjin en China [Li, *et al.*, 2023]. Este sistema de control busca mejorar el bajo nivel de eficiencia en la clasificación y transporte en la producción agrícola. El sistema consta de un brazo robótico, una computadora, una cámara industrial y una banda transportadora como se muestra en la Figura 1.4. Estos componentes trabajan en conjunto para permitir que el robot identifique, clasifique y disponga adecuadamente cada tipo de fruta o verdura. Los resultados obtenidos tras las pruebas demuestran que la precisión en la detección y organización de objetos supera el 90 %, lo que representa un alto nivel de eficiencia. Estos resultados hacen que la implementación de este sistema en la producción agrícola sea viable y prometedora, ofreciendo la posibilidad de aumentar la productividad y mejorar los procesos en este sector clave de la economía.

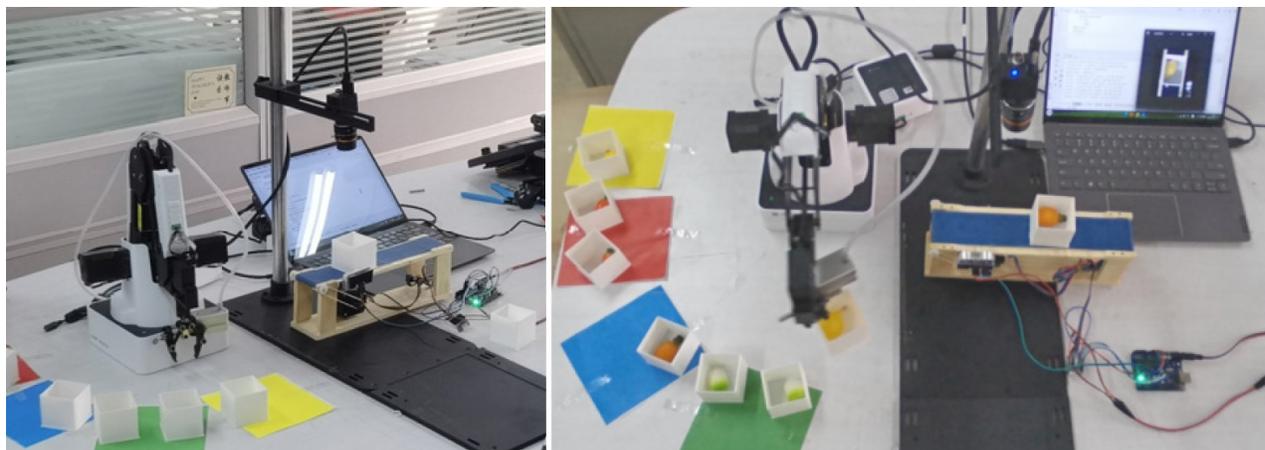


Figura 1.4 Sistema automático de clasificación y transporte de frutas y verduras basado en el brazo robótico Dobot Magician

1.3. Formulación del problema

Se dispone de un prototipo de una celda de manufactura robotizada, fabricada por la empresa mexicana Hitech Ingenium en el año 2017. Está conformada por 4 módulos que en conjunto simulan el funcionamiento de una planta de embotellamiento industrial. La problemática se centra en analizar y comprender el funcionamiento del cuarto módulo que realiza el reciclaje dentro del proceso.

Como se puede observar en la Figura 1.5 y la Figura 1.6, este módulo cuenta con un brazo robótico de 4 grados de libertad con una pinza como efector final, 3 bandas y un riel con carro transportador, 5 motores de corriente directa con su respectivo controlador, un encoder incremental, 5 sensores fotoeléctricos, 2 sensores de límite, 4 pistones de doble efecto y uno de simple efecto con retorno por muelle, 5 electroválvulas y un mini compresor. De los cuales se desconocen datos técnicos, así como sus conexiones eléctricas y neumáticas.

Además, la falta de sensores o transductores limita la capacidad de medir variables eléctricas clave, como el voltaje y la corriente consumida por los dispositivos conectados al módulo de reciclaje.

Asimismo, la ausencia de una interfaz de supervisión en tiempo real dificulta la monitorización de variables de interés, como los estados de los actuadores y los consumos eléctricos.

Por último, la carencia de una estrategia y arquitectura de control impide la medición y la actuación sobre los dispositivos mencionados anteriormente, obstaculizando así el desarrollo de las acciones de reciclaje correspondientes a este módulo.

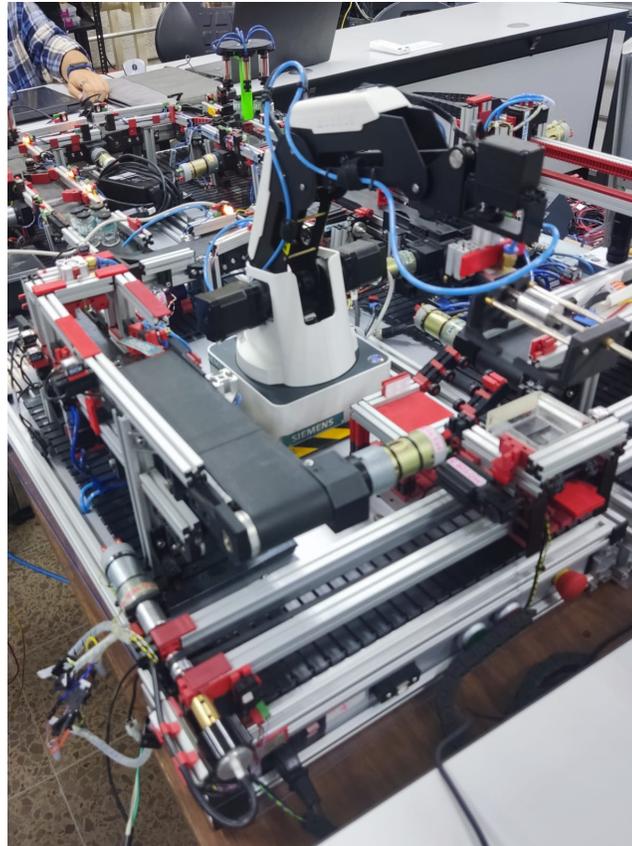


Figura 1.5 Vista superior del módulo 4

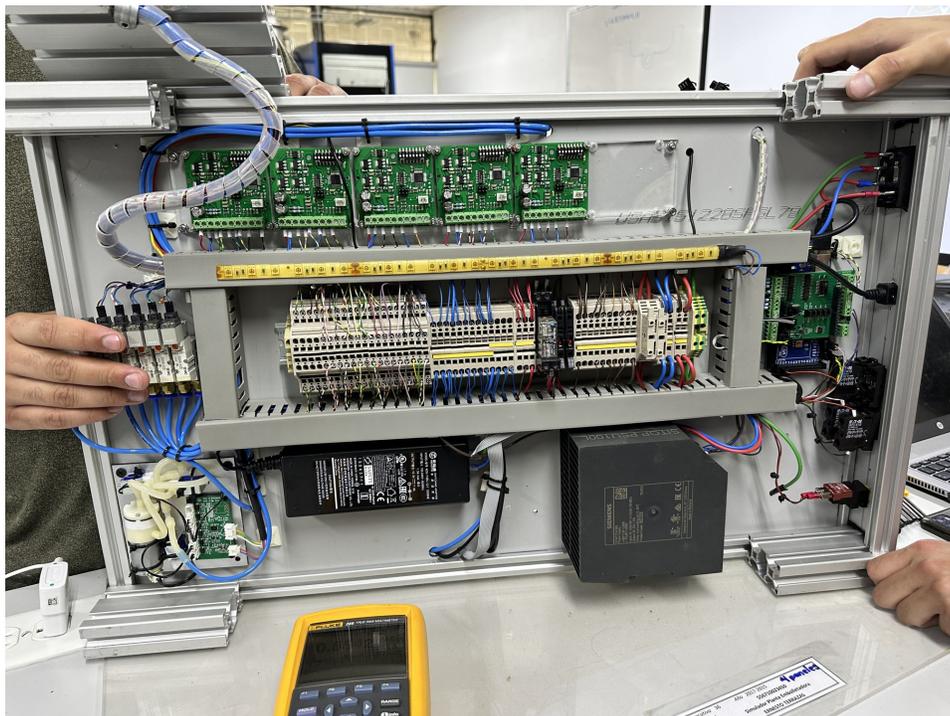


Figura 1.6 Vista inferior del módulo 4

1.4. Objetivo

Diseñar y desarrollar un sistema de control secuencial y supervisión para el cuarto módulo de la maqueta prototipo. Este sistema permitirá automatizar y optimizar la etapa de reciclado del proceso y así comenzar la gradual puesta en marcha de todo el prototipo.

1.4.1. Objetivos particulares

- Desarrollar el levantamiento técnico y diagramas de la plataforma con base en dibujos paramétricos utilizando herramientas de diseño asistido por computadora (CAD).
- Diseñar una arquitectura de control flexible y confiable.
- Desarrollar una rutina de control bajo los lenguajes del estándar industrial IEC 61131-3.
- Diseñar una interfaz de control y supervisión en tiempo real.
- Implementar indicadores clave de desempeño para medir la eficiencia energética de la plataforma.

1.5. Contribuciones

- Se realizó un levantamiento técnico, el cual incluye la detallada documentación de los componentes que conforman la plataforma para comprender el equipo y funcionamiento del módulo.
- Se generó diagramas de conexiones eléctricas que describen la distribución e interconexiones de todos los dispositivos de este módulo.
- Se realizó una descripción funcional de la plataforma donde se explica el funcionamiento del módulo en cada una de sus etapas y transiciones.
- Se diseñó e implementó una arquitectura de control confiable para interconectar y programar al PLC y la periferia descentralizada, así como también dar cabida a otros dispositivos de automatización que se agreguen en un futuro.
- Se propuso y desarrolló la rutina de control en el lenguaje secuencial SFC en conjunto con el diagrama de escalera LD, con la finalidad de expresar la rutina de control en etapas, acciones y transiciones bien definidas que garanticen un óptimo y correcto funcionamiento del sistema, además de ser más sencillo de comprender para futuros operadores.
- Se desarrolló un *script* de Arduino para el control de los movimientos del brazo robótico. Este script sirve como una pasarela de conexión entre el controlador industrial y el manipulador para enviar y recibir información que permita desarrollar la rutina de control.
- Se diseñó e implementó una interfaz de control y supervisión en un servidor web, en la cual es posible iniciar, detener o reiniciar la rutina de control, así como poder conocer la etapa actual del proceso y algunos consumos eléctricos.
- Se realizaron arreglos, modificaciones y mantenimiento pertinentes al diseño original para mejorar el funcionamiento del módulo.

1.6. Organización de la tesis

El presente trabajo de tesis se encuentra dividido en cinco capítulos, siendo el presente el que concierne a la introducción. Los siguientes cuatro se describen a continuación:

En el **Capítulo 2** se presenta el marco teórico pertinente al desarrollo de este prototipo, donde se presentan conceptos generales referentes a la automatización tales como la arquitectura de control, algunas topologías de red, el protocolo de comunicación PROFINET, la definición del control secuencial, la metodología GRAFCET para la solución de sistemas secuenciales, así como también la norma estándar IEC 61131-3. También se presentaran conceptos relacionados al hardware del módulo tales como el PLC, la periferia descentralizada, el brazo robótico, entre algunos otros. Finalmente, se abordaran conceptos relacionados al software utilizado.

En el **Capítulo 3** se presenta el proceso de desarrollo y puesta en marcha de este módulo. En un principio se muestra el desarrollo del levantamiento técnico haciendo especial énfasis en su hardware, para posteriormente mostrar el proceso de implementación de la arquitectura de control y finalmente el diseño e implementación del sistema de control.

Para validar el funcionamiento del módulo, en el **Capítulo 4** se muestran los resultados obtenidos del sistema de control y supervisión a nivel de campo, de control y de supervisión.

Finalmente, en el **Capítulo 5** se establecen las conclusiones y el potencial trabajo a futuro sobre este proyecto. Los diagramas obtenidos en el desarrollo del módulo se incluyen en el **Apéndice A**. Finalmente, en el **Apéndice B** se presentan las rutinas de control implementadas en diagrama escalera LD y SFC.

Capítulo 2

Marco Teórico

2.1. Celda de manufactura robotizada

Una celda de manufactura robotizada es un sistema de producción automatizado que utiliza robots industriales para realizar tareas específicas dentro de un área de trabajo definida, denominada celda. Estas celdas están diseñadas para mejorar la eficiencia, precisión y consistencia en los procesos de manufactura.

El concepto de automatización se puede aplicar en diferentes niveles. En el primer nivel, conocido como nivel de dispositivos, se manejan sensores y actuadores dentro de lazos de control. El segundo nivel, o nivel de máquina, coordina el funcionamiento de varios dispositivos. En el tercer nivel, llamado nivel de sistema o celda, se organiza y opera una celda de manufactura, la cual consiste en un conjunto de máquinas o estaciones de trabajo conectadas por un sistema de manejo de materiales; este nivel incluye líneas de producción y ensamblaje. El cuarto nivel, denominado nivel de planta, genera planes operacionales como los de procesamiento, control de inventarios, compras, control de calidad, entre otros. El quinto nivel es el nivel de empresa, que gestiona la información corporativa relacionada con mercadeo, ventas, contabilidad, investigación y diseño, programación de producción, etcétera.

La automatización de una celda de manufactura robotizada se clasifica en el tercer nivel, ya que implica la coordinación de máquinas o unidades entre las cuales se trasladan piezas con el fin de producir piezas terminadas. El control de cada unidad corresponde al nivel de máquina, y la operación de cada actuador se sitúa en el nivel de dispositivo [Sánchez-Chávez y Martell-Chávez, 2019].

2.2. Arquitectura de control

Como menciona [García Barroso, 2015], la arquitectura de control se presenta visualmente de manera detallada, ofreciendo una representación gráfica de los elementos asociados con el control, comunicaciones y supervisión en las instalaciones, subprocesos o máquinas. Esto abarca no sólo los equipos, sino también las redes y buses de comunicación implicados en el sistema. Una buena arquitectura de control posee las siguientes características:

- Visualiza de manera sencilla a todo el sistema de control.
- Identifica equipos y sus principales características.
- Identifica las redes y nodos de comunicación.

2.3. Topología de red

En términos simples, una topología de red se trata de la estructura física que puede tomar una red de comunicación, algunas de estas son la tipo bus, estrella, anillo y árbol [Pardo, 2015].

2.3.1. Topología tipo bus

La topología de bus es la configuración más básica para enlazar una red, pues como se observa en la Figura 2.1, todos los dispositivos de comunicación están conectados en un bus lineal, lo que tiene como ventaja de requerir una menor cantidad de cableado. Sin embargo, si falla un elemento de enlace, la comunicación a través del elemento fallido ya no es posible. En consecuencia, la red se divide en 2 subsegmentos [Siemens, 2023a].

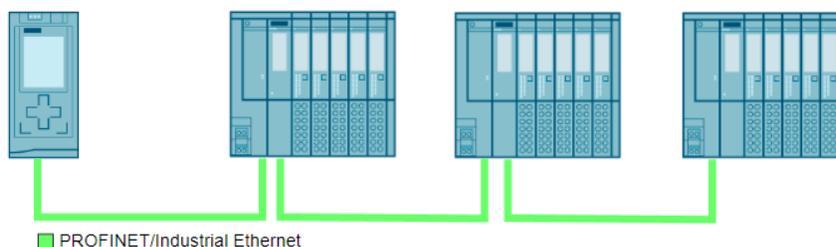


Figura 2.1 Topología tipo bus

2.3.2. Topología tipo estrella

Como se aprecia en la Figura 2.2, la topología de estrella implica conectar los dispositivos a un switch general que se encarga de retransmitir la información recibida de todos los dispositivos. Si falla un solo dispositivo, a diferencia de otras estructuras, esto no provoca el fallo total de la red. Sólo si falla el switch parte de la red también lo hará [Siemens, 2023a].

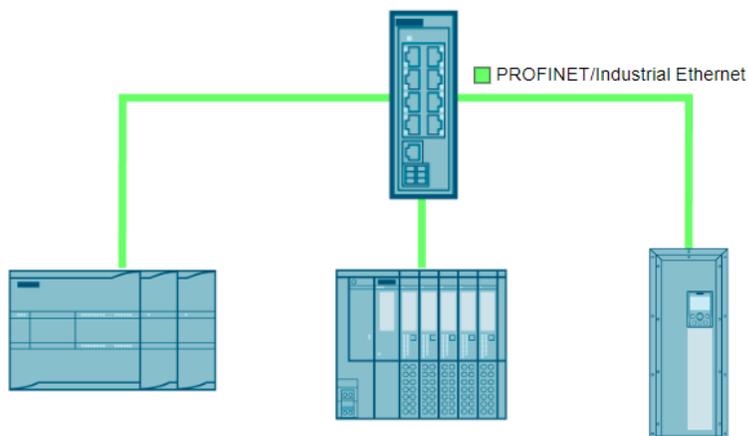


Figura 2.2 Topología tipo estrella

2.3.3. Topología tipo árbol

La topología de tipo árbol es esencialmente una configuración donde múltiples topologías en estrella están interconectadas, pero en lugar de conectarse a un solo equipo central, se utiliza un nodo central a partir del cual se ramifican las demás conexiones, tal y como se aprecia en la Figura 2.3.

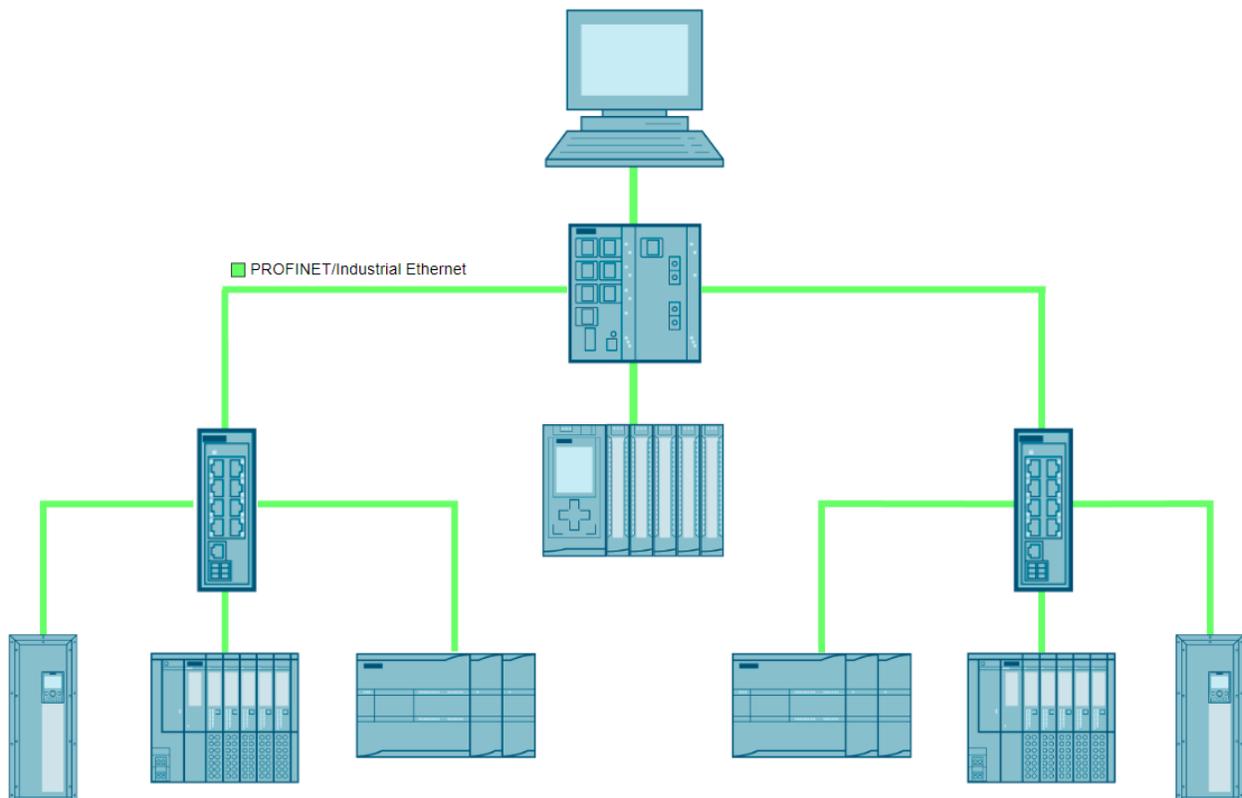


Figura 2.3 Topología tipo árbol

2.3.4. Topología tipo anillo

La topología de anillo implica la conexión de varios dispositivos en una disposición circular, utilizando un único cable por el cual la información se transmite en una dirección específica, ya sea en sentido horario o antihorario tal como se observa en la Figura 2.4. Para aumentar la disponibilidad de la red una topología de bus lineal se conecta al anillo a través del llamado administrador de redundancia. En caso de haber una interrupción en la red, el administrador de redundancia se encarga de que los datos se redirijan a través de una conexión de red alternativa en el anillo [Siemens, 2023a].

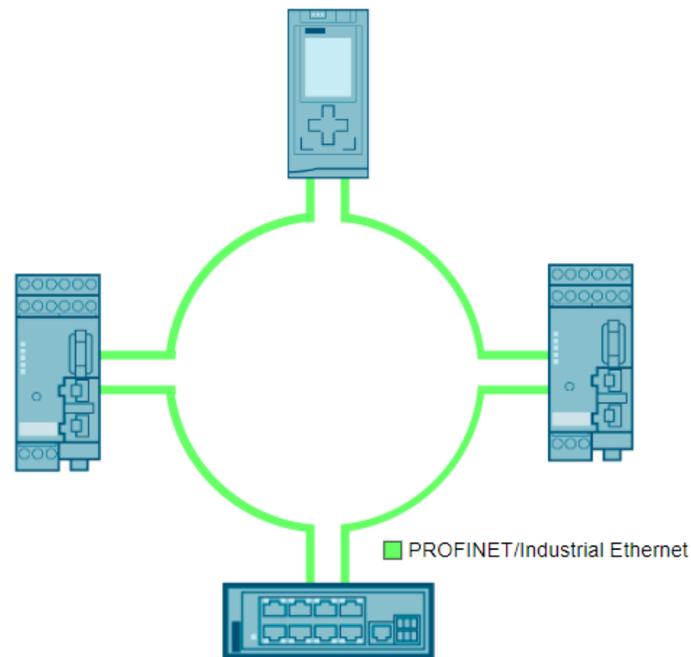


Figura 2.4 Topología tipo anillo

2.4. Protocolo PROFINET

El protocolo PROFINET es un estándar internacional incorporado a los estándares IEC 61158 y IEC 61784 para el Ethernet industrial [Dorf, 2015]. PROFINET es más que un óptimo sistema de comunicación para la automatización, pues como se explica en [Pigan y Metter, 2008] es un estándar integral que cumple con todos los requisitos para la implementación de Ethernet en entornos de automatización industrial, esto abarca desde la comunicación a nivel de controlador hasta la automatización con sistemas de Entradas y Salidas (I/O, por sus siglas en inglés), y se extiende incluso a aplicaciones avanzadas de control de movimiento. Por lo tanto, PROFINET es adecuado para muchas aplicaciones de automatización.

2.5. Sistemas de control secuencial

Actualmente en el marco básico de la automatización existen diferentes sistemas de control lógico, cada uno de los cuales ejecuta una función limitada y claramente definida para procesar señales de entrada y obtener señales de salida, donde la activación y configuración de estas funciones se logra mediante distintas señales de control. Para llevar a cabo secuencias de procesos complejas, como las especificaciones de fabricación de productos, es necesario coordinar y activar las diversas funciones en el momento adecuado y con los parámetros correctos [Siemens, 2020]. Esta tarea puede llevarse a cabo mediante el uso de sistemas de control secuencial.

Los sistemas de control secuencial posibilitan el procesamiento escalonado y discreto de eventos en procesos secuenciales y paralelos mediante la utilización de gráficos de funciones secuenciales, que también reciben el nombre de secuenciadores. Estos sistemas inducen cambios en el estado operativo y

a los ajustes de estado que dependen de estados o eventos previamente definidos en los sistemas de control lógico preexistentes, con el fin de llevar a cabo el flujo secuencial deseado [Siemens, 2020]. Los gráficos de funciones de secuenciadores también son conocidos como gráficos de funciones secuenciales SFC.

2.6. Metodología GRAFCET

GRAFCET es un acrónimo de *Graphe Fonctionnel de Commande Étape Transition*, que significa Gráfico Funcional de Control de Etapas y Transiciones. Como lo menciona [Castellanos y Solaque, 2010], este es un método gráfico de modelado y diseño de sistemas automáticos que se fundamenta en la teoría de Redes de Petri. Dentro de las ventajas que introduce este método, destaca por su enfoque gráfico, ofreciendo una representación visual clara de la lógica de control en sistemas automatizados. Su estructura jerárquica facilita la división de sistemas complejos en módulos manejables, permitiendo una fácil comprensión y gestión del diseño. Al centrarse en el manejo de estados y transiciones, resulta especialmente efectivo para sistemas secuenciales, y su diseño modular simplifica la identificación y solución de problemas.

Por lo tanto, la adaptabilidad a cambios, la compatibilidad con estándares industriales, y la facilidad de traducción a código de los lenguajes de PLC hacen que GRAFCET sea una herramienta versátil y ampliamente utilizada en entornos industriales, proporcionando un método efectivo para el desarrollo de sistemas automatizados.

2.7. Estándar IEC 61131-3

Con la aparición de los PLC en la industria y los fabricantes diseñando sus propios dispositivos y métodos de programación, surgió la necesidad por parte de los usuarios de tener lenguajes uniformes e independientes del fabricante, lenguajes de programación de alto nivel y herramientas de desarrollo similares a las que existían desde hace muchos años en el mundo de las computadoras [Karl-Heinz y Tiegelkamp, 2001]. A raíz de esta problemática, en 1993 surge el estándar IEC 61131, en donde sus cinco partes resumen los requisitos de los PLC modernos. Estos requisitos se refieren al hardware del PLC y al sistema de programación. El estándar incluye tanto los antiguos conceptos que ya se utilizaban en la programación de estos controladores, como varios de los actuales métodos de programación. Como resultado de la creación de este estándar se creó una base para una programación uniforme aprovechando los conceptos modernos de la tecnología de software.

La IEC 61131-3 es la tercer parte de esta norma. Aquí se especifica la sintaxis y la semántica de un conjunto unificado de lenguajes de programación para controladores programables. Estos son dos lenguajes gráficos, el Diagrama de Escalera (LD) y el Diagrama de Bloques de Funciones (FBD), dos lenguajes textuales, la Lista de Instrucciones (IL) y el Texto Estructurado (ST) y uno secuencial, el Gráfico de Funciones Secuenciales (SFC), [IEC, 2003]. Cabe destacar que este estándar no es un conjunto de reglas estrictas que se deban cumplir en su totalidad, sino más bien es una guía para que los fabricantes implementen parte del estándar en sus dispositivos y sistemas de automatización. Así mismo, los fabricantes deben evidenciar en qué áreas cumplen o no con los requisitos establecidos por la normativa [Karl-Heinz y Tiegelkamp, 2001].

2.8. Interfaz Humano Máquina

Las Interfaces Humano Máquina (HMI) constituyen la piedra angular de la interacción entre humanos y tecnología, facilitando así la comunicación y el control de dispositivos, sistemas y software. Su diseño, guiado por principios de usabilidad y accesibilidad, busca garantizar una interacción intuitiva y satisfactoria. Ya sea en entornos industriales, donde se controlan complejas operaciones, o en dispositivos cotidianos como teléfonos inteligentes. Hablando específicamente del ámbito industrial, como lo menciona [Hollifield, *et al.*, 2008] actualmente cada proceso controlado por un Sistema de Control Distribuido (DCS, por sus siglas en inglés) moderno utiliza una HMI compuesta por dispositivos físicos junto con software que en conjunto presentan en tiempo real los procesos de la planta. Por lo tanto, estas interfaces permiten al usuario supervisar y administrar eficientemente a los procesos industriales.

Ya que la HMI juega un papel muy importante en el control operativo de una planta, es fundamental una interfaz bien diseñada para garantizar operaciones fluidas, un conocimiento óptimo de la situación del proceso y una respuesta eficiente ante situaciones anormales. En contraste, una interfaz mal diseñada tiene el potencial de comprometer la seguridad, la producción y la calidad. Es por eso que existen estándares como la ANSI/ISA 101.1-2015 que se enfocan en proporcionar pautas y buenas prácticas para el diseño e implementación de interfaces de usuario en sistemas automatizados.

2.9. Controlador SIMATIC S7-1500

El sistema de automatización S7-1500 que se muestra en la Figura 2.5, es un PLC desarrollado por Siemens, una empresa líder en tecnología industrial. Como se describe en [Siemens, 2013] este sistema proporciona la adaptabilidad y el rendimiento esenciales para gestionar las exigencias de la mayoría de industrias, además su estructura escalable permite ajustar el controlador según las necesidades específicas del proceso [Siemens, 2023c]. Por otra parte, algunas características destacadas del sistema de automatización S7-1500 son:

- Rendimiento mejorado en comparación con modelos anteriores.
- Comunicación avanzada para conectarse con otros dispositivos y sistemas en la planta, facilitando la integración en entornos industriales complejos.
- Características de seguridad integradas para garantizar la protección de las máquinas y de las personas. Esto puede incluir funciones como la monitorización segura, la detección de fallos y otras medidas de seguridad.
- Diseño modular que permite adaptar y expandir el sistema según las necesidades específicas de la aplicación.
- Proporciona un entorno de programación avanzado que facilita la creación y el mantenimiento de programas de control.
- En algunos modelos, se pueden integrar capacidades de conectividad a la nube para facilitar la supervisión remota, el mantenimiento predictivo y otras aplicaciones basadas en la nube.



Figura 2.5 PLC SIMATIC S7-1500

2.10. Periferia descentralizada SIMATIC ET 200SP

El sistema ET 200SP que se muestra en la Figura 2.6, es una periferia descentralizada de entradas y salidas remotas desarrollado por Siemens. Este sistema es usualmente utilizado en entornos industriales para la conexión de sensores y actuadores a un controlador central. Facilita la conexión a los dispositivos en el campo ya que proporciona módulos compactos y modulares que pueden distribuirse a lo largo de una máquina o el proceso [Siemens, 2024b]. Algunas características destacadas del sistema descentralizado ET 200SP son:

- Modularidad, ya que pueden configurarse según las necesidades específicas del sistema lo que facilita la expansión y adaptación. Cuenta con módulos de entradas y salidas digitales y analógicas, módulos tecnológicos, de comunicación y especiales, así como arrancadores de motor y sistemas neumáticos.
- Múltiples opciones de comunicación para integrarse con controladores y sistemas de control superiores. Dependiendo del modelo, estos sistemas pueden disponer de las interfaces MultiFieldbus, PROFINET o PROFIBUS.
- Diseño compacto ya que sus módulos suelen ser pequeños para ahorrar espacio y permitir su instalación en lugares cercanos a los puntos de E/S.

- Fácil instalación y una puesta en marcha rápida. Además de contar con intercambio en caliente, así como arranque y funcionamiento con slots vacíos.

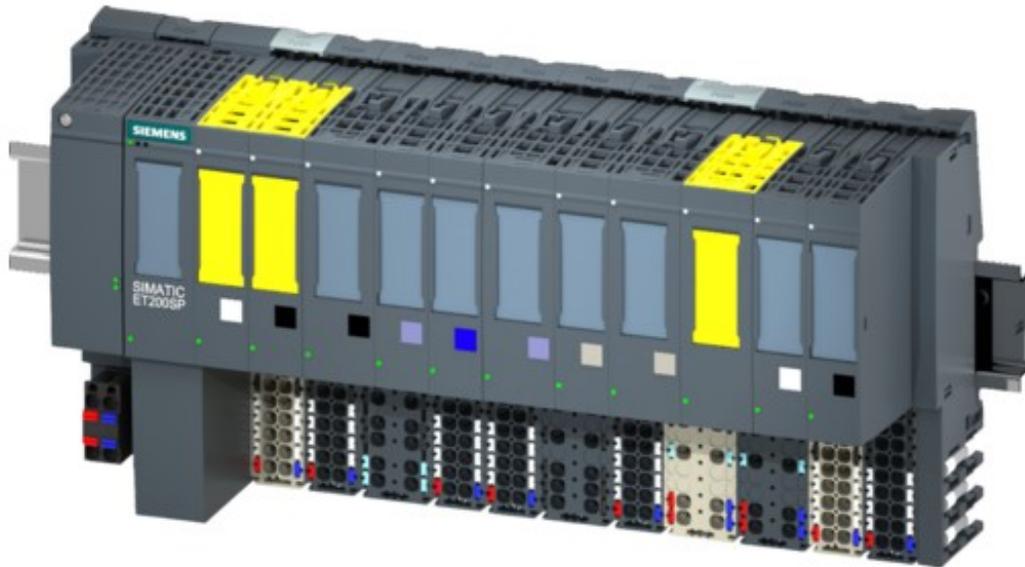


Figura 2.6 Periferia descentralizada SIMATIC ET 200SP

2.11. Switch Ethernet SCALANCE X308

El switch SCALANCE X308 que se muestra en la Figura 2.7, es un conmutador Ethernet a nivel industrial desarrollado por Siemens. Este dispositivo cumple con todos los requisitos de las redes industriales y sistemas de bus, ya sea conmutación, enrutamiento, aplicaciones de seguridad, acceso remoto o LAN inalámbrica industrial [Siemens, 2023b]. Algunas características destacadas de este switch son:

- 10 puertos de conexión RJ45 con velocidades de 10 a 1000 Mbit/s
- LED de diagnóstico
- Contacto de señalización de error con botón de selección/configuración
- Administración de redes
- Gestor de redundancia integrado

Como lo menciona [Siemens, 2024a] la tecnología más aplicada en áreas LAN es el Ethernet, y un dispositivo con las cualidades del SCALANCE X308 hace posible la creación de redes extensas con varios dispositivos, incrementando el caudal de datos y simplificando la ampliación de la red.



Figura 2.7 Switch Ethernet SCALANCE X308

2.12. Brazo robótico Dobot Magician

Como se aprecia en la Figura 2.8, Dobot Magician es un brazo robótico multifuncional de escritorio con 4 grados de libertad desarrollado por la empresa china DOBOT. Este robot es versátil y está diseñado para ser utilizado en entornos educativos, de investigación y en algunas industrias. Equipado con diferentes efectores finales que amplían su funcionalidad, el Dobot Magician puede realizar diferentes actividades como impresión 3D, grabado láser, manipulación de objetos, escritura y dibujo. Además de estar equipado con 13 puertos de interfaz disponibles para el desarrollo de proyectos [Dobot, 2019].

Este brazo robótico suele utilizarse para enseñar conceptos de programación y robótica a estudiantes, ya que cuenta con un entorno de programación visual que facilita la creación de secuencias de movimientos y acciones para el robot. Además de poder ser programado por más de 20 lenguajes de programación, de los que destacan Python, C++, Arduino, Java, entre otros.



Figura 2.8 Brazo robótico Dobot Magician

2.13. Módulo de expansión 24V Industrial I/O Shield

Tal como se muestra en la Figura 2.9, este es módulo de expansión para Arduino desarrollado por la empresa Rugged Circuits. Su finalidad es ampliar las capacidades de un Arduino MEGA convencional para poder desempeñar actividades industriales tal como si se tratase de un PLC. Este módulo permite tener a los usuarios de Arduino funciones muy similares a las de un controlador, ofreciendo 8 entradas digitales aisladas de 3.3 [V] - 24 [V], 8 salidas digitales aisladas de 5 [V] - 28 [V], 4 entradas analógicas configurables, leds indicadores de estado y comunicación RS-232. Dando así flexibilidad y muchas prestaciones por solo una fracción del precio de un costoso PLC [Rugged Circuits, 2024].

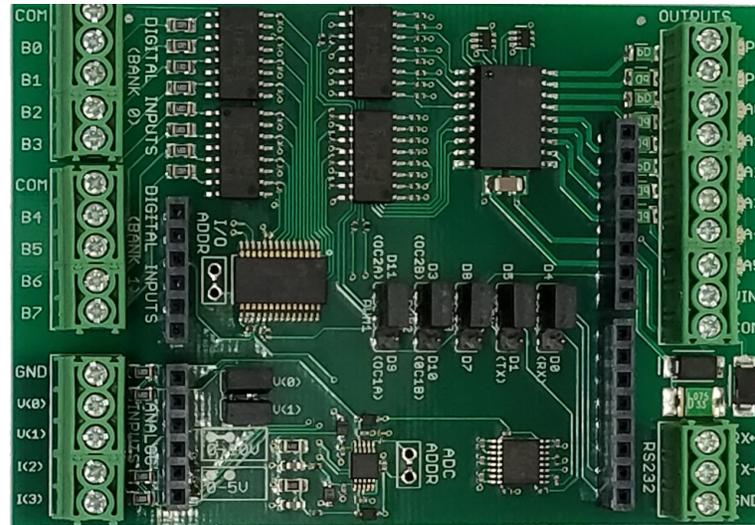


Figura 2.9 Módulo de expansión de Arduino 24V Industrial I/O Shield

2.14. Controlador de motor DC EM-170

El controlador de motor EM-170 desarrollado por la empresa Electromen Oy que se muestra en la Figura 2.10. Está pensado para pequeños motores DC entre 12 y 24 [V] con una corriente máxima de 1.5 [A], y es capaz de manejar el motor en distintas velocidades, cambiar su dirección de giro y frenarlo. Utiliza el principio de modulación por ancho de pulso (PWM), lo que le otorga una alta eficiencia en su etapa de potencia. Como lo menciona [Electromen, 2024] algunas de las características destacadas son:

- Ajuste de velocidad mediante señal de voltaje analógica.
- Retroalimentación del voltaje del motor para mantener la velocidad constante ante cambios en la tensión de operación.
- Límite de corriente ajustable para limitar el torque del motor, mediante interruptores DIP.
- Probado bajo estándares industriales.



Figura 2.10 Controlador de motor DC EM-170

2.15. Transformador de corriente MES-62/30

Como se aprecia en la Figura 2.11, este es un transformador de corriente con relación de transformación de $30/5$ [A], desarrollado por la empresa Dixsen. Su finalidad es, bajo el principio de la inducción electromagnética, convertir la corriente de un circuito de alta corriente en una corriente proporcional y más manejable que puede ser fácilmente medida por instrumentos de medición estándar. Esto es útil para monitorear y proteger equipos eléctricos, así como para medir y controlar el consumo de energía eléctrica en aplicaciones industriales y residenciales [Dixsen, 2024].



Figura 2.11 Transformador de corriente MES-62/30

2.16. Transductor de corriente LTS 15-NP

El Transductor de corriente LTS 15-NP de LEM, mostrado en la Figura 2.12, es un dispositivo utilizado para medir corriente eléctrica en sistemas industriales y de potencia tanto en corriente alterna como directa. Este transductor es una opción confiable y precisa para medir corriente eléctrica, ofreciendo una combinación de precisión, compatibilidad y durabilidad [LEM, 2017]. Algunas de las características destacadas son:

- Soporta hasta 15 [A] en sistemas de corriente alterna.
- Alta precisión en la medición de corriente, lo que lo hace adecuado para aplicaciones donde la precisión es crítica.
- Diseño robusto y compacto, ideal para soportar condiciones ambientales adversas y entornos industriales exigentes, lo que garantiza su fiabilidad y durabilidad a largo plazo.
- Rápido tiempo de respuesta, lo que lo hace adecuado para aplicaciones donde se requiere un control preciso y rápido de los sistemas eléctricos.

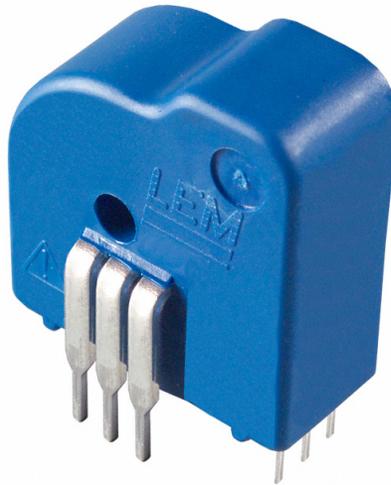


Figura 2.12 Transductor de corriente LTS 15-NP

2.17. Entorno de desarrollo TIA Portal

El Portal de Automatización Totalmente Integrado (TIA Portal, por sus siglas en inglés), es un entorno de ingeniería de software desarrollado por Siemens. Este software se utiliza para la programación y configuración de sistemas de automatización industrial, como controladores lógicos programables, interfaces humano-máquina, sistemas de control de movimiento y otros dispositivos relacionados. Además, TIA Portal es conocido por su enfoque de integración, lo que facilita la conexión y la comunicación entre diferentes dispositivos y componentes del sistema.

Este software permite a los ingenieros de control y automatización trabajar de manera eficiente ya que integra software básico como STEP7, WinCC Unified, SINAMICS, SCOUT TIA, funcionalidades como multiusuario y administración de energía, así como todo el hardware del proyecto de automatización: control, HMI, motores, periféricos descentralizados, administración de motores, control de movimiento y distribución de energía, además de herramientas de simulación [Siemens, 2024c]. Todo lo anterior y más en un único entorno de desarrollo.

2.17.1. WinCC Unified View of Things

Partiendo de la filosofía SIMATIC WinCC Unified System, la cual ofrece un sistema de visualización abierto y flexible que combina tecnologías web y de borde, modernas y seguras con la finalidad de preparar a una infraestructura para el futuro y así superar los retos de la digitalización [Siemens, 2024b]. Siemens ofrece la herramienta WinCC Unified View of Things.

Como se describe en [Siemens, 2021] SIMATIC WinCC Unified View of Things, también conocido como la Visión de las Cosas (VoT, por sus siglas en inglés), es una aplicación web que está integrada en todos los controladores SIMATIC S7-1500 con el firmware TIA Portal V17 o superior, y no necesita de ningún hardware o software adicional. Esta herramienta permite operar y monitorear parámetros de control relevantes directamente a través del servidor web alojado en la memoria del controlador, lo que permite integrar una HMI al proyecto sin la necesidad de tener un panel de visualización.

De acuerdo a [Siemens, 2021], esta herramienta está basada en la tecnología WinCC Unified Technology. Esto significa que garantiza la coherencia y compatibilidad de las interfaces desarrolladas

en View of Things, con WinCC Unified PC y Unified Comfort Panels, ya que la mayoría de objetos son compatibles con estas otras herramientas [Siemens, 2024b].

Como se menciona en [Siemens, 2024b], VoT implementa un editor intuitivo que permite arrastrar y soltar para crear la interfaz de visualización web, y dado a que no requiere escribir ningún código HTML, tampoco requiere de conocimiento alguno en programación. Únicamente se requiere de una licencia de ingeniería para WinCC (TIA Portal) [Siemens, 2021].

De acuerdo a [Siemens, 2024b], lo que distingue a VoT es que está diseñada para tareas de visualización sencillas basadas en web. Esto hace que sea una sencilla solución de visualización HTML con la que siempre se tendrá a la vista las cosas más importantes del proceso. Sin embargo, según [Siemens, 2021], el mayor inconveniente de VoT es que utiliza un número limitado de objetos en pantalla en comparación con WinCC Unified PC y Unified Comfort Panels como se muestra en la Figura 2.13, lo cual puede ser una limitante en algunos proyectos. Entonces, si la visualización requiere requisitos más sofisticados, se recomienda utilizar una solución HMI totalmente equipada como lo es WinCC Unified PC o Unified Comfort Panels.

De acuerdo con [Siemens, 2024b], se ofrece una descripción de los casos y posibilidades de uso de una visualización web con VoT:

- Uso según la demanda: Si no se necesita un control y monitoreo continuo, y la visualización es usada ocasionalmente, por ejemplo, para mantenimiento regular.
- Condiciones de instalación estrecha: Cuando la maquina está en una infraestructura complicada que dificulta la instalación de un panel HMI en sitio.
- Uso de herramientas similares: Cuando los usuarios no están familiarizados con el uso de HMI industriales. Como en el sector agrícola.
- Mas que solo la industria: Cuando no se requiere robustez en los dispositivos de operación industrial, y los dispositivos móviles tradicionales son suficientes para el control y monitoreo.

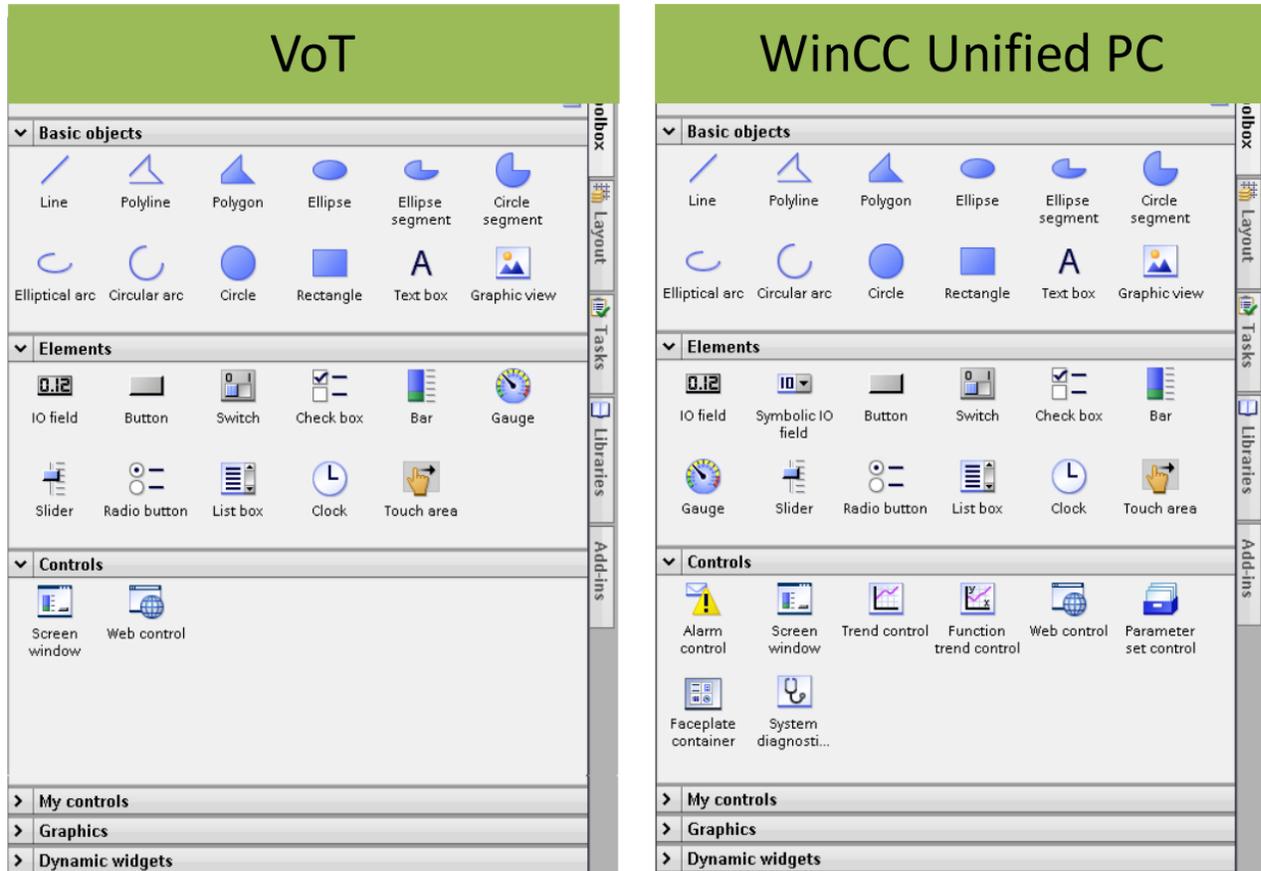


Figura 2.13 Comparación de objetos en VoT y WinCC Unified PC

Finalmente, en la Tabla 2.1, se presentan las principales diferencias entre WinCC Unified View of Things, Unified Comfort Panel y WinCC Unified PC [Siemens, 2024b]:

	WinCC Unified Unified Comfort	Unified Comfort Panel	WinCC Unified PC
Objetos y elementos básicos (e.g. Campo de E/S, botón, indicador)	✓	✓	✓
Gráficos (incl. SVG dinámico)	✓	✓	✓
Pantallas	✓	✓	✓
Faceplates		✓	✓
Alarmas		✓	✓
Archivos		✓	✓
Auditoría e informes		✓	✓
Control de parámetros		✓	✓
WinCC Unified Collaboration		✓	✓
Opciones de planta inteligente			✓

Tabla 2.1 Diferencias entre las soluciones HMI del WinCC Unified System

Capítulo 3

Desarrollo

En este capítulo se describe el proceso de desarrollo y puesta en marcha del cuarto módulo del prototipo, el capítulo se divide en cinco secciones principales. En la Sección 3.1 se hace una identificación de todos los dispositivos que integran a la plataforma para posteriormente documentar información relevante a dichos dispositivos. Después se procede a realizar la identificación de todas las conexiones eléctricas, para posteriormente realizar los diagramas pertinentes con herramientas de diseño CAD.

En la Sección 3.2 se explica la arquitectura de control propuesta para los dispositivos de automatización utilizados en este proyecto. En la Sección 3.3 se expone el desarrollo de la pasarela de conexión entre el PLC y el brazo robótico. En la Sección 3.4 se realiza la descripción funcional de este módulo con la finalidad de comprender cuales son las funciones a desempeñar. Finalmente, en la Sección 3.5 se describe el diseño e implementación de las rutinas de control implementadas en LD y SFC, así como la interfaz de supervisión web desarrollada con la herramienta VoT.

3.1. Levantamiento técnico

Se realizó la identificación de todos los elementos que conforman este módulo. Estos van desde los dispositivos de automatización como la periferia descentralizada junto a sus módulos de expansión, pasando por sensores fotoeléctricos y de contacto, actuadores tales como motores de corriente directa, electroválvulas y cilindros neumáticos, hasta otro tipo de dispositivos eléctricos y electrónicos tales como controladores de motor y botones.

3.1.1. Dispositivos de automatización

El único dispositivo de automatización propio de este módulo es la periferia descentralizada ET 200SP modelo IM155-6 PN HF, mostrada en la Figura 3.1. Este sistema cuenta con 6 módulos de expansión los cuales constan de 2 módulos de entradas digitales a 24 [V], uno de salidas digitales a 24 [V] y 0.5 [A], un módulo de salidas analógicas, un módulo medidor de energía de hasta 480V de corriente alterna y un módulo de entradas analógicas. En la Tabla 3.1 se especifican estos dispositivos.

MODELO	NO. SERIE	CANTIDAD
ET 200SP Bus Adapter BA 2xRJ45	6ES7193-6AR00-0AA0	1
ET 200SP IM155-6 PN HF incl. Server Module	6ES7155-6AU00-0CN0	1
SIMATIC ET 200SP Digital Input Module	6ES7131-6BF00-0CA0	2
SIMATIC ET 200SP Digital Output Module	6ES7132-6BH00-0BA0	1
SIMATIC ET 200SP Analog Output Module	6ES7135-6HD00-0BA1	1
SIMATIC ET 200SP Analog Input Module Energy Meter	6ES7134-6PA20-0BD0	1
SIMATIC ET 200SP Analog Input Module	6ES7134-6HB00-0CA1	1

Tabla 3.1 Dispositivos de automatización del módulo 4

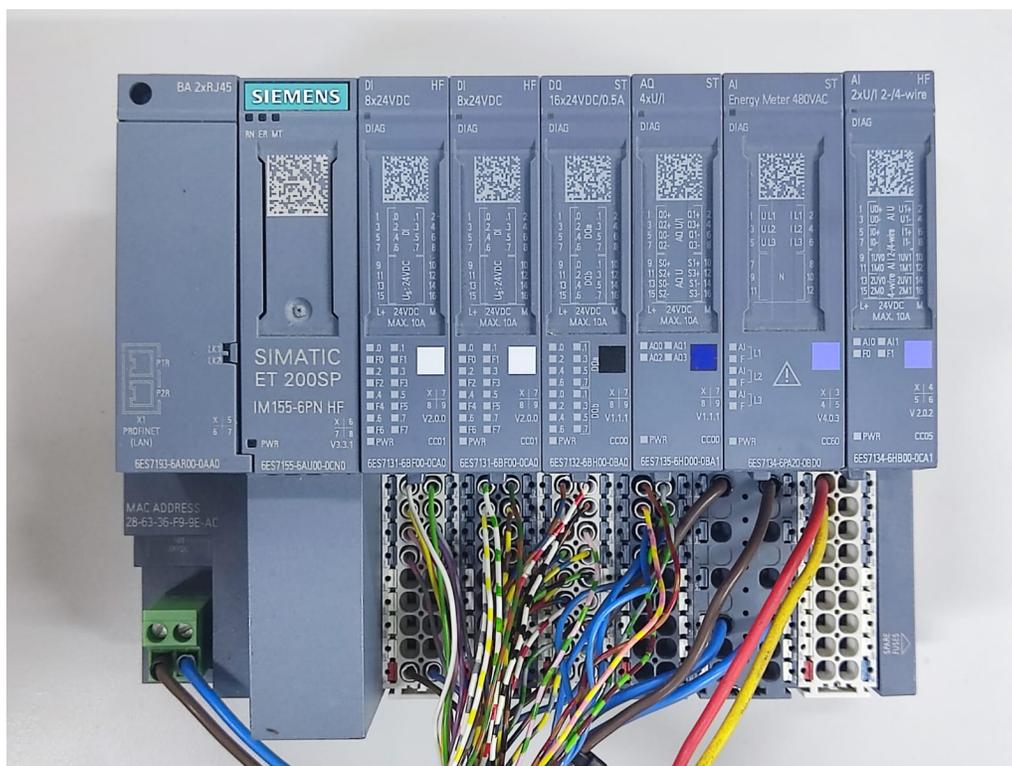


Figura 3.1 Periferia descentralizada ET 200SP del módulo 4

3.1.2. Actuadores

Un actuador es un dispositivo con la capacidad de efectuar una acción o movimiento en respuesta a una señal de control. En otras palabras, recibe una señal de entrada y la convierte en una acción física. Los principales actuadores de este módulo son los motores eléctricos de corriente directa a 24 [V] como el mostrado en la Figura 3.2, estos son la pieza clave de las bandas y la cinta transportadoras. Por otra parte, están las electroválvulas de 3 vías y 2 posiciones mostradas en la Figura 3.3, las cuales funcionan a 24 [V] y activan a los cilindros neumáticos de doble efecto y al de simple efecto con retorno por muelle. En la Tabla 3.2 se especifican estos dispositivos.

MODELO	NO. SERIE	CANTIDAD
MOTOR DC 24V	S/N	5
ELECTROVÁLVULA SMC 3/2	SYJ314M-5L0ZD	5

Tabla 3.2 Actuadores del módulo 4

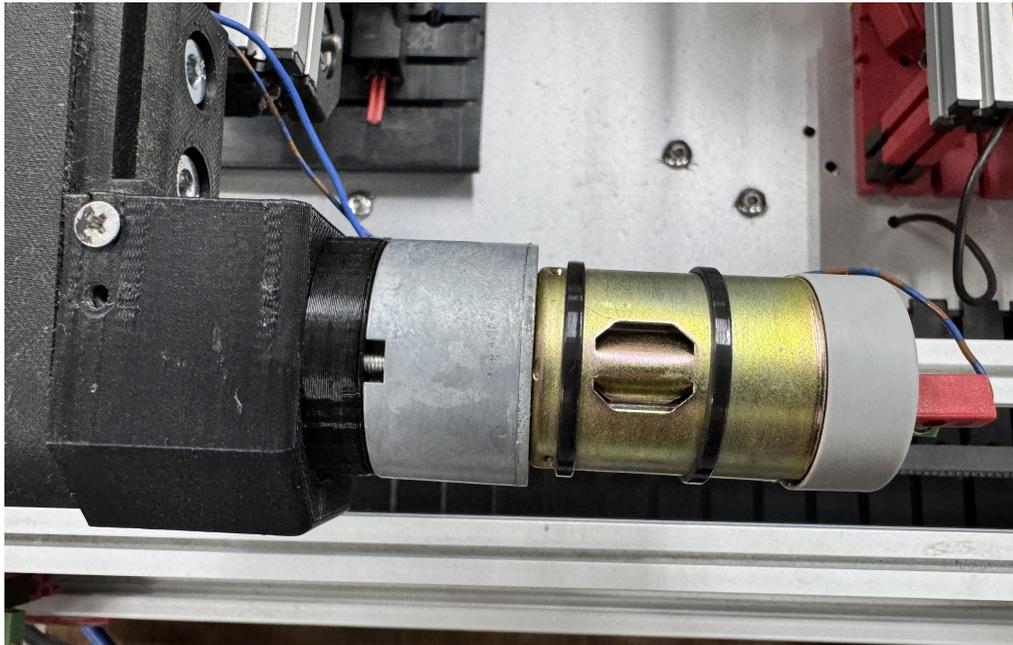


Figura 3.2 Motor 24VDC

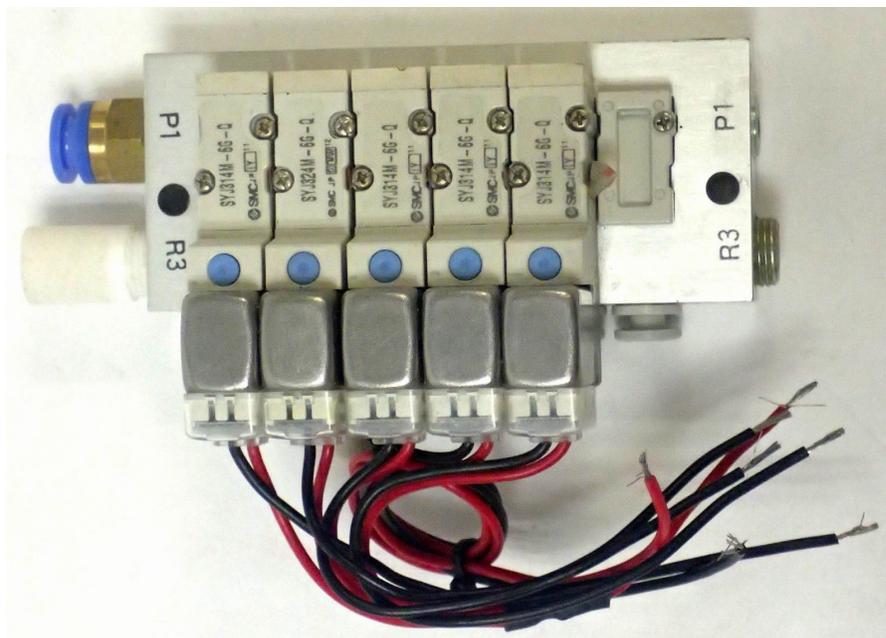


Figura 3.3 Electroválvulas de 3 vías y 2 posiciones

3.1.3. Sensores

Los sensores son dispositivos que miden alguna propiedad física y convierten esa información en señales eléctricas que pueden ser interpretadas, almacenadas o utilizadas de alguna manera. Recordando que existen diferentes tipos de sensor, cada uno diseñado para medir una propiedad específica: temperatura, luz, proximidad, movimiento, entre otros. Los sensores utilizados en este módulo son de 3 tipos, por una parte, hay sensores fotoeléctricos con lógica positiva y negativa, que detectan proximidad y operan a 24 [V] como el mostrado en la Figura 3.4. Por otra parte, hay sensores de límite que detectan posición y también funcionan a 24 [V], tal como el que se muestra en la Figura 3.5. La diferencia entre estos 2 sensores es que los de proximidad miden sin contacto con el objeto a sensar gracias al principio de reflexión de luz, mientras que los de límite si tienen contacto ya que funcionan como un interruptor mecánico. Por último, hay un encoder incremental para motor mostrado en la Figura 3.6, que proporciona información sobre el movimiento relativo del eje desde la última posición de referencia conocida. En la Tabla 3.3 se especifican estos dispositivos.

MODELO	NO. SERIE	CANTIDAD
Sensor fotoeléctrico de proximidad PNP	MHT1-P122	4
Sensor fotoeléctrico de proximidad NPN	MHT1-N020S23	1
Sensor de límite	S/N	2
Encoder incremental	S/N	1

Tabla 3.3 Sensores del módulo 4



Figura 3.4 Sensor fotoeléctrico SICK

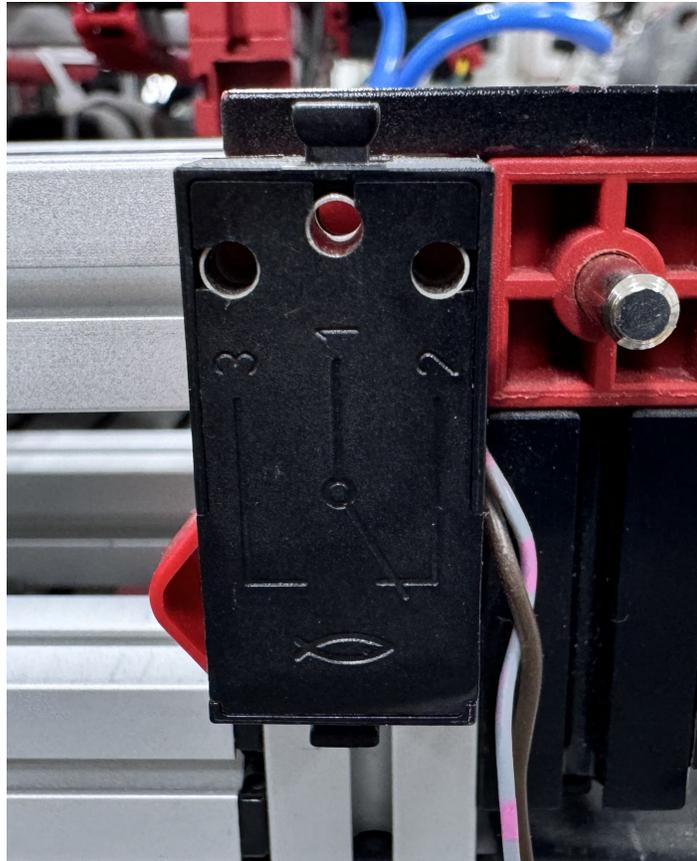


Figura 3.5 Sensor de límite

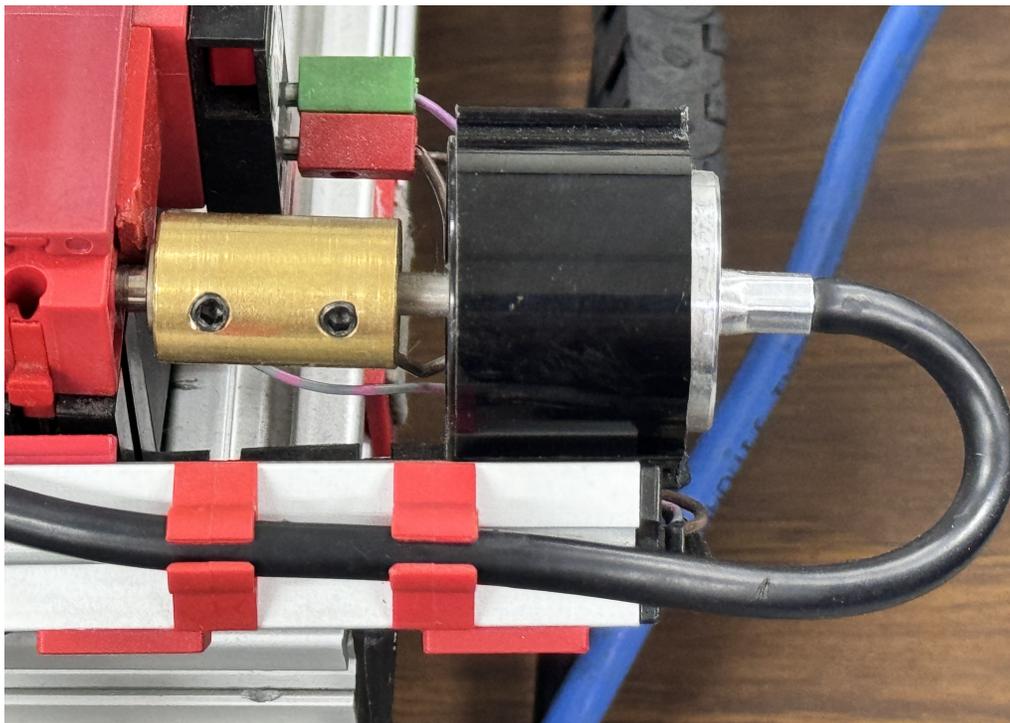


Figura 3.6 Encoder incremental para motor

3.1.4. Otros dispositivos eléctricos y electrónicos

Además de lo antes mencionado, también se utilizaron otros dispositivos eléctricos como las fuentes de alimentación para el robot y la de todo el módulo en general, relevadores de protección, mini válvulas solenoide a 12 [V] y 2 botones normalmente abiertos, uno para el inicio y otro para el paro del proceso, y uno normalmente cerrado que funge como un paro eléctrico.

Por otra parte, también hay dispositivos electrónicos, como los controladores para los motores de DC a 24 [V], un controlador ON/OFF con 6 salidas y 2 configuraciones: una que enciende las primeras y últimas dos salidas, y otra que enciende las primeras 4 salidas, un Arduino Mega y un módulo de expansión industrial para Arduino. En la Tabla 3.4 se especifican estos dispositivos.

MODELO	NO. SERIE	CANTIDAD
Controlador de motor DC	EM-170	5
Controlador ON/OFF	S/N	1
Arduino MEGA	A000067	1
Módulo de expansión industrial Arduino	24V Industrial I/O shield	1
Fuente SITOP PSU 100L	6EP1334-1LB00	1
Fuente del robot	ADS-110DL-12-1	1
Relé electromagnético Weidmuller	RCI314AC4	1
Relé electromagnético	S/N	2
Mini válvula solenoide 12v	S/N	3
Botón NA ATON MOELLER	M22-D-R/K10	1
Botón NA ATON MOELLER	M22-D-G/K10	1
Botón NC Siemens	3SB2908-0AA	1

Tabla 3.4 Otros dispositivos eléctricos y electrónicos del módulo 4

3.1.5. Tabla de ordenamiento de variables

Una vez conocidos los dispositivos de automatización del módulo 4, es necesario conocer que dispositivos se conectan a la periferia descentralizada ET 200SP. Para ello se realizó una minuciosa identificación de todas las conexiones físicas de la ET 200SP. Como resultado se obtuvo el siguiente ordenamiento de variables: entradas digitales en la Tabla 3.5, salidas digitales en la Tabla 3.6, entradas analógicas en la Tabla 3.7 y salidas analógicas en la Tabla 3.8.

Entradas digitales

DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN
I0.0	SENSOR_0	Sensor fotoeléctrico 0
I0.1	SENSOR_1	Sensor fotoeléctrico 1
I0.2	SENSOR_2	Sensor fotoeléctrico 2
I0.3	SENSOR_3	Sensor fotoeléctrico 3
I0.4	SENSOR_4	Sensor fotoeléctrico 4
I0.5	SENSOR_5	Sensor de límite 1
I0.6	SENSOR_6	Sensor de límite 2
I0.7	SENSOR_7	Sensor fotoeléctrico 5
I1.0	ENCODER_1	Señal 1 del encoder
I1.1	ENCODER_2	Señal 2 del encoder
I1.2	ENCODER_3	Señal 3 del encoder
I1.3	A0_SHIELD	Salida A0 del I/O shield
I1.4	A1_SHIELD	Salida A1 del I/O shield
I1.5	A2_SHIELD	Salida A2 del I/O shield
I1.6	RED_BUTTON	Botón NA rojo
I1.7	GREEN_BUTTON	Botón NA verde

Tabla 3.5 Entradas digitales

Salidas digitales

DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN
Q0.0	DIRECTION_M1	Sentido de giro del motor 1
Q0.1	DIRECTION_M2	Sentido de giro del motor 2
Q0.2	MOTOR2	Encendido/apagado del motor 2
Q0.3	DIRECTION_M3	Sentido de giro del motor 3
Q0.4	MOTOR4	Encendido/apagado del motor 4
Q0.5	DIRECTION_M4	Sentido de giro del motor 4
Q0.6	DIRECTION_M5	Sentido de giro del motor 5
Q0.7	MOTOR5	Encendido/apagado del motor 5
Q1.0	EV1	Electroválvula 1
Q1.1	EV2	Electroválvula 2
Q1.2	EV3	Electroválvula 3
Q1.3	EV4	Electroválvula 4
Q1.4	EV5	Electroválvula 5
Q1.5	B0_SHIELD	Entrada B0 del I/O shield
Q1.6	B1_SHIELD	Entrada B1 del I/O shield
Q1.7	B2_SHIELD	Entrada B2 del I/O shield

Tabla 3.6 Salidas digitales

Entradas Analógicas

DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN
IW4	METER_CAC	Voltaje AC
IW6	METER_VAC	Corriente AC

Tabla 3.7 Entradas analógicas

Salidas Analógicas

DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN
QW2	SPEED_M1	Velocidad del motor 1
QW4	SPEED_M2	Velocidad del motor 2
QW6	SPEED_M3	Velocidad del motor 3
QW8	SPEED_M4	Velocidad del motor 4

Tabla 3.8 Salidas analógicas

3.1.6. Diagramas eléctricos

Una vez conocidos todos los dispositivos que conforman la plataforma, así como su direccionamiento, es necesario identificar de manera sencilla las interconexiones entre cada uno de sus elementos. Para ello se diseñó con ayuda de herramientas CAD, un diagrama general de conexiones eléctricas para representar al módulo 4, como se observa en la Figura 3.7. La particularidad de este diagrama es que puede visualizarse de diferentes maneras, ya sea con solo las conexiones entre dispositivos como se muestra en la Figura A.3, o las conexiones de alimentación eléctrica como se ve en la Figura A.4, o las conexiones de neutros y tierras visualizadas en la Figura A.5, o con cualquier combinación de las anteriormente mencionadas. Todos los diagramas anteriores se presentan en el Apéndice A para su mejor visualización.

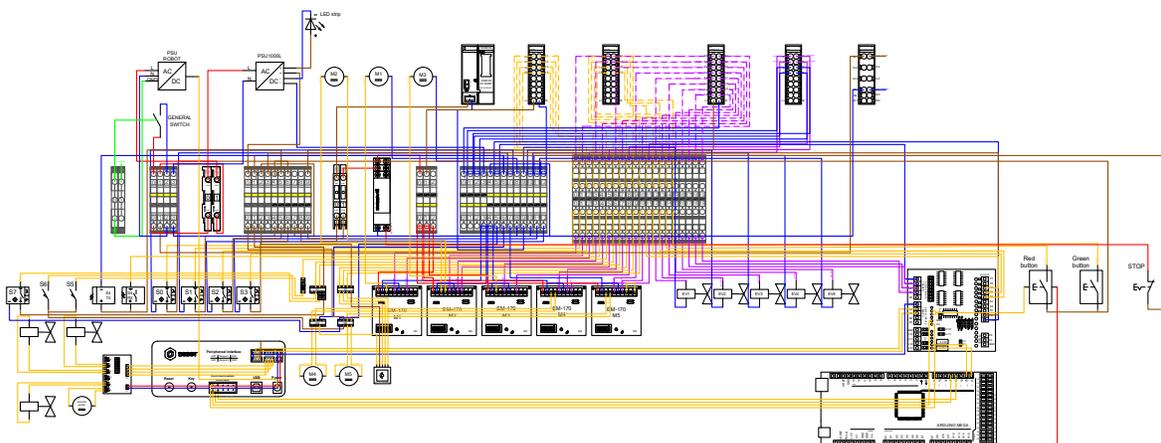


Figura 3.7 Diagrama de conexiones generales del módulo 4

Lo primordial fue implementar el PLC el cual es un SIMATIC S7-1500 con CPU 1516F-3 PN/DP, este controlador cuenta con la capacidad integrar interfaces de supervisión web gracias a la herramienta VoT. Por otra parte, para poder interconectar la periferia descentralizada de la etapa de reciclado, así como cualquier otro dispositivo de automatización a futuro, se decidió incorporar un switch Ethernet SIMATIC SCALANCE X308. Esta elección proporciona una sólida base para la expansión y la integración de nuevos componentes de manera eficiente. Finalmente se aprovecho la infraestructura existente en el laboratorio, con la cual fue posible conectar la arquitectura propuesta a otro switch Ethernet y, a su vez, a un punto de acceso inalámbrico. La arquitectura resultante se muestra en la Figura 3.8.

3.3. Pasarela de conexión

Hay que destacar que el robot no puede recibir señales digitales mayores a 5 [V], ni enviar señales digitales mayores a 12 [V]. Al no tener la capacidad de conectarse directamente a un controlador que, en este caso, recibe y envía señales digitales a 24 [V], fue necesario implementar una pasarela de conexión entre el PLC y el brazo robótico. Al igual que en las redes de comunicación, una puerta de enlace facilita la interconexión y la comunicación entre redes distintas, esta pasarela permite comunicar al PLC y al brazo robótico sin alterar sus respectivos voltajes de operación.

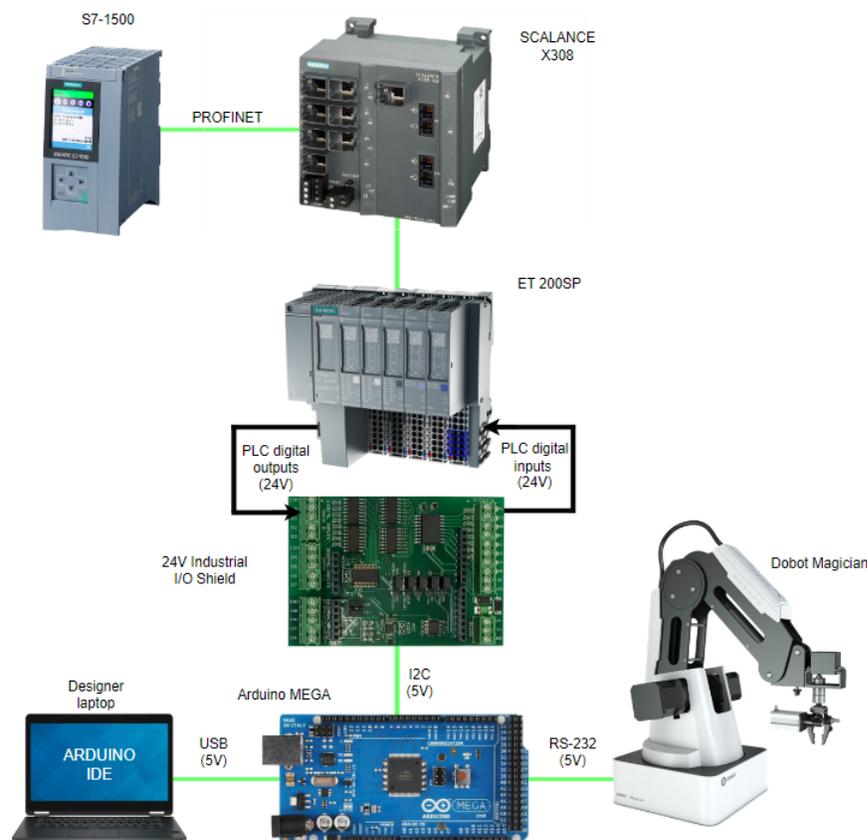


Figura 3.9 Pasarela de conexión

Hay que destacar que para este caso en particular, existen diferentes maneras de aplicar esta pasarela de conexión, pero la implementada por los diseñadores originales del módulo fue haciendo uso de un Arduino MEGA y un módulo de expansión industrial, el 24 [V] Industrial I/O Shield. De esta manera el PLC puede enviar señales digitales a 24 [V] que recibe el módulo de expansión y convierte a señales de control que procesa e interpreta el Arduino MEGA, para posteriormente, mediante el puerto serie mandar comandos de control a 5 [V] al robot. De manera análoga, el robot manda señales de control a 5 [V], que recibe el Arduino MEGA y hace que el módulo de expansión envíe señales digitales a 24 [V] hacia el controlador. El esquema de conexiones de esta pasarela de conexión se representa en la Figura 3.9.

3.4. Descripción funcional del prototipo

El siguiente paso es describir el funcionamiento de este módulo, así como también la manera en que se relaciona con los módulos aledaños. Primeramente hay que recordar que el prototipo se conforma por 4 módulos que en conjunto simulan un proceso de embotellado industrial, y el módulo desarrollado en este texto es el cuarto, encargado del reciclaje de las botellas.

El comienzo de la rutina sucede cuando el módulo recibe un empaque de botellas terminado (caja con cuatro botellas llenas) en la base de recepción, como se muestra en la Figura 3.10. Una vez que se cumple este proceso el robot procede a levantar el empaque y ponerlo en la banda transportadora superior, muestrado en la Figura 3.11. Una vez que el robot dejó el empaque, este se traslada por la banda hasta bajar por la rampa, como se aprecia en la Figura 3.12a, y llegar a la zona de vaciado, simulada con un pistón que presiona a las 4 botellas como se muestra en la 3.12b. Posteriormente llega a la zona de reciclaje, donde un pistón detiene la caja mientras que el robot saca las botellas para colocarlas en la banda transportadora de un costado, la cual se encarga de regresarlas al módulo 1 para volver a iniciar el proceso, como se aprecia en la Figura 3.13. Una vez la caja se encuentre vacía, la misma banda que la trajo la lleva en reversa hacia el riel transportador, como se muestra en la Figura 3.14a. Una vez que la caja se encuentre en el transportador, se lleva hasta el módulo 3 donde se deja la caja vacía para continuar con el proceso de dicho módulo, como se aprecia en la Figura 3.14b.

Sin embargo, lo anteriormente mencionado es el funcionamiento para cuando todos los módulos del simulador se encuentren operativos, dado que no es el caso se tuvieron que hacer dos modificaciones al funcionamiento del cuarto módulo para que este pudiera operar sin la necesidad del resto de módulos. La primera modificación es que la rutina comenzará cuando se presione el interruptor verde del módulo o de la interfaz, y la caja con botellas se encuentre en la base de recepción. La segunda modificación consta en llevar a la cinta transportadora con la caja vacía, hasta la posición de la base de recepción como se muestra en la Figura 3.15, y no hasta el módulo 3 como originalmente se diseñó.

Adicionalmente se agregó la función de un reinicio físico. Hay que destacar que a falta de un tercer botón normalmente abierto para esta función, el reinicio del proceso se realizará mediante el accionamiento simultáneo de los botones de inicio y paro.

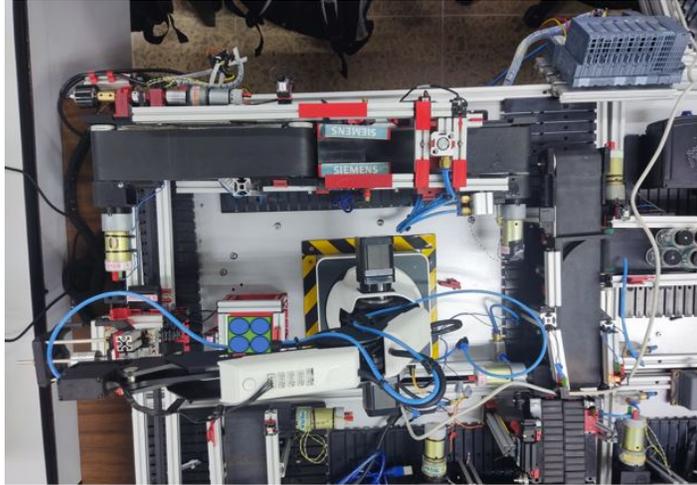


Figura 3.10 Empaque de botellas terminado en la base de recepción

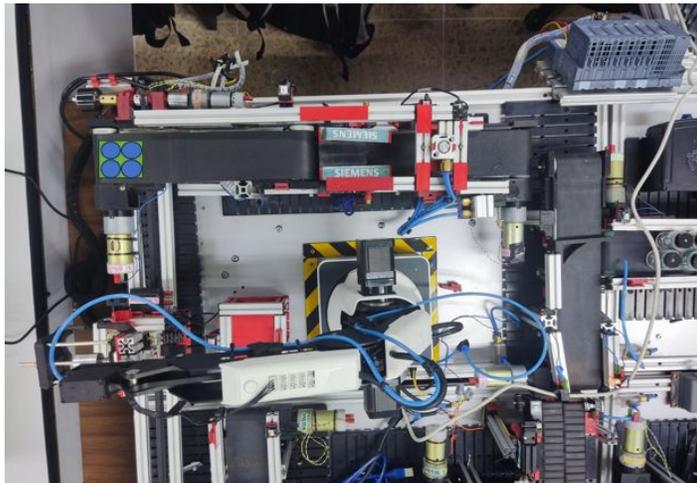
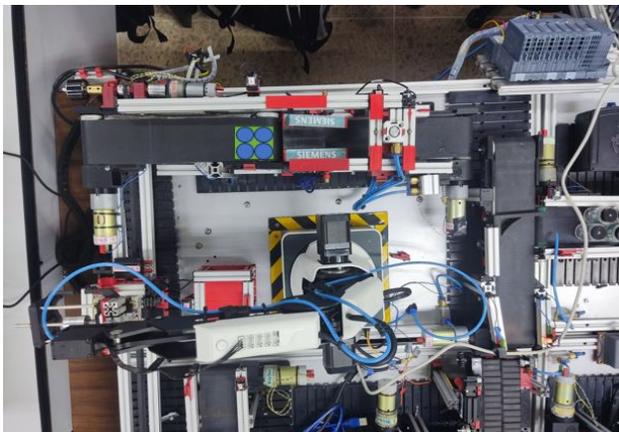
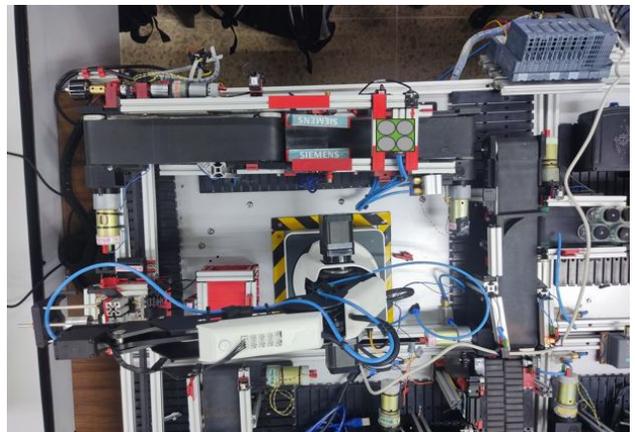


Figura 3.11 Empaque en la banda transportadora superior



(a) Botellas llenas antes de bajar



(b) Botellas vacías después de bajar

Figura 3.12 Vaciado de botellas

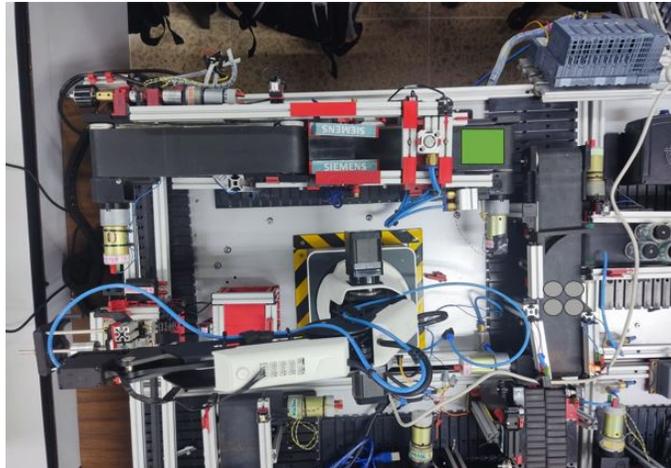
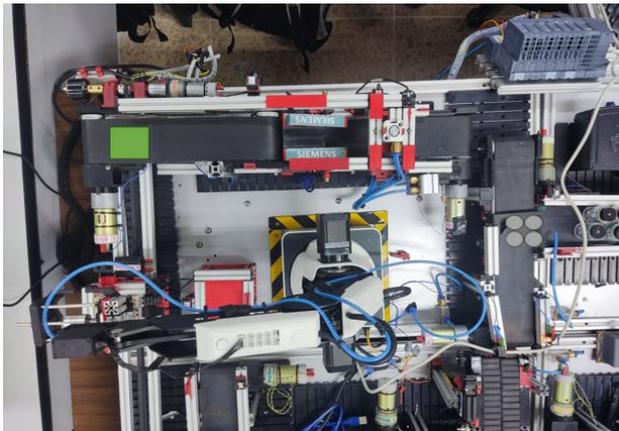
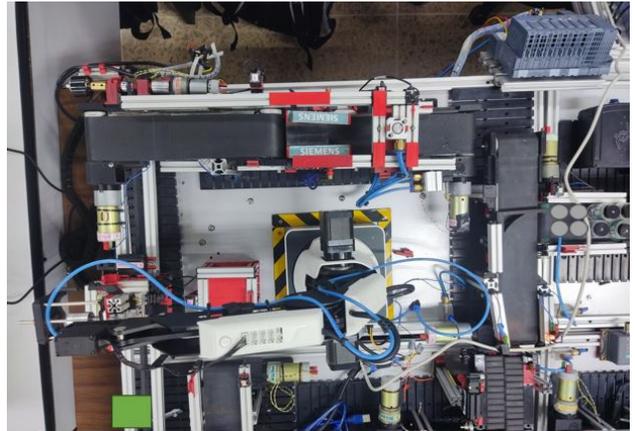


Figura 3.13 Caja y botellas separadas



(a) Caja vacía hacia el transportador y botellas vacías hacia el módulo 1



(b) Caja vacía en el transportador y botellas vacías en el módulo 1

Figura 3.14 Distribución de cajas y botellas vacías

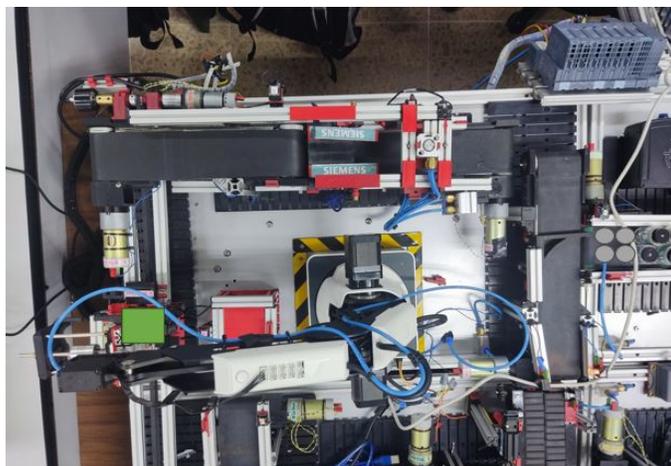


Figura 3.15 Cinta transportadora en la posición de la base de recepción

3.5. Diseño e implementación del sistema de control y supervisión

El objetivo de las rutinas de control es la puesta en operación del módulo conforme a lo establecido en la descripción funcional. Adicionalmente se implementó un control de arranque suave para los motores, esto con la finalidad optimizar consumos energéticos, reducir el desgaste mecánico de los motores y demostrar la capacidad para evaluar diferentes estrategias de control en este prototipo. La estrategia de control principal está conformada por dos lenguajes del estándar IEC 61131-3, en este caso en particular son complementarios ya que uno no funciona sin el otro. Por una parte, se tiene a la rutina en el gráfico de funciones secuenciales SFC, donde la rutina se define en etapas y transiciones que garanticen un óptimo funcionamiento. Por otra parte, está el diagrama en escalera LD en el cual invoca a la rutina SFC y complementa a la misma con algunas funciones tal como las condiciones de inicio, paro y reinicio de la rutina, la configuración de velocidad de los motores, la interpretación de las variables eléctricas medidas, el indicador de la etapa actual de la rutina SFC y la interpretación del encoder.

3.5.1. Rutina de control LD

En la Figura 3.16 se muestra el peldaño 1, donde se invoca a la rutina SFC, pues si este paso no se realiza, el control secuencial diseñado en dicha rutina nunca se aplicaría al proceso.

Network 1: INVOCA A LA RUTINA SFC

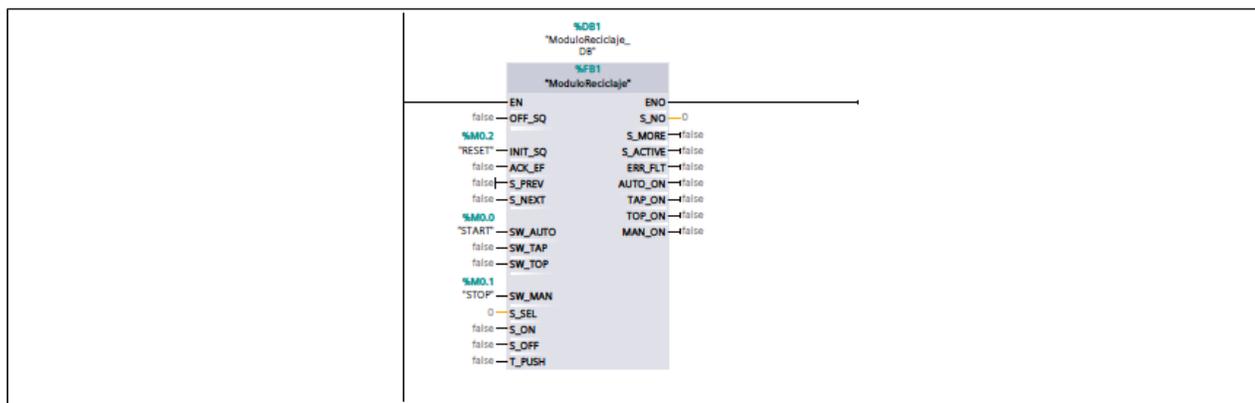


Figura 3.16 Peldaño 1 de la rutina LD

En la Figura 3.17 se muestra el peldaño 2, donde se realiza una comparación lógica para iniciar la primera trayectoria del brazo robótico con la variable *B1_SHIELD*. Esta comparación implica detectar el empaque con botellas llenas en la base de recepción simultáneamente con la activación de la variable *ON*, que señala que la rutina SFC ha comenzado.

Network 2: DETECTA QUE LA RUTINA HAYA INICIADO Y LA CAJA ESTE EN POSICION

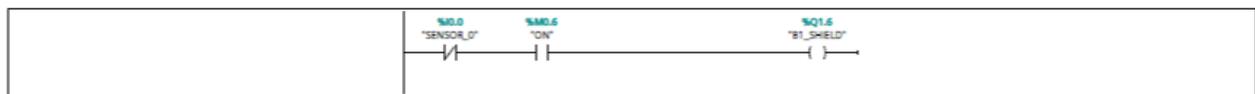


Figura 3.17 Peldaño 2 de la rutina LD

En la Figura 3.18 se muestra el peldaño 3, donde se configura la disposición lógica para el inicio de la rutina. Esta disposición consiste en dos variables en paralelo, *GREEN_BUTTON* y *START_REM*, que tienen la capacidad de iniciar la rutina, junto con una variable en serie, *RESET*, que evita que la rutina se inicie nuevamente durante un reinicio físico, el cual se lleva a cabo al presionar simultáneamente los botones de inicio (verde) y paro (rojo).

Network 3: INICIO DE LA RUTINA



Figura 3.18 Peldaño 3 de la rutina LD

En la Figura 3.19 se muestra el peldaño 4, donde se configura la disposición lógica para detener la rutina. Esta disposición incluye un enclavamiento con prioridad a la desconexión, con las variables en paralelo *RED_BUTTON* y *STOP_REM*, que pueden detener la rutina. Además, hay variables en serie, *START* y *RESET*, que permiten levantar el paro cuando se reanuda o reinicia la rutina.

Network 4: PARO DE LA RUTINA

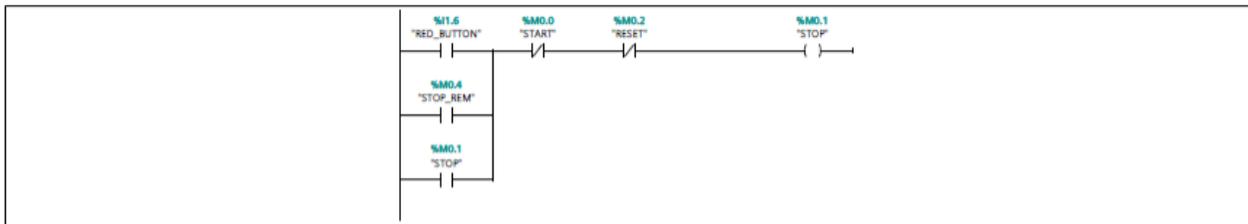


Figura 3.19 Peldaño 4 de la rutina LD

En la Figura 3.20 se muestra el peldaño 5, donde se establece la disposición lógica para reiniciar la rutina. Esta configuración consta de dos condiciones en paralelo. La primera implica dos variables en paralelo, *GREEN_BUTTON* y *RED_BUTTON*, debido a la ausencia de un tercer botón normalmente abierto para permitir el reinicio. La segunda condición en paralelo es cuando se activa la variable *RESET_REM* desde la HMI web.

Network 5: REINICIO DE LA RUTINA

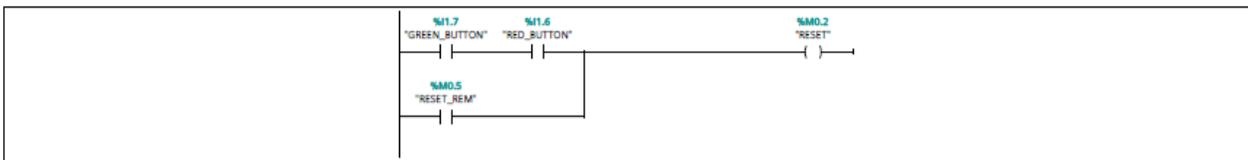


Figura 3.20 Peldaño 5 de la rutina LD

En la Figura 3.21 se muestra el peldaño 6, donde se lleva a cabo la configuración de la velocidad de los motores. Esto es necesario ya que inicialmente todos los motores deben ajustarse a una misma velocidad. Sin embargo, durante la ejecución del proceso, es necesario aumentar la velocidad del motor 2 para facilitar la subida de la caja vacía al carro transportador. La velocidad de los motores se ajusta mediante señales PWM, las cuales varían según el voltaje enviado desde el convertidor digital-analógico del módulo de salidas analógicas de la periferia hacia el controlador del motor. Este voltaje, a su vez, depende del valor decimal enviado.

Network 6: CONFIGURACION DE VELOCIDAD EN MOTORES

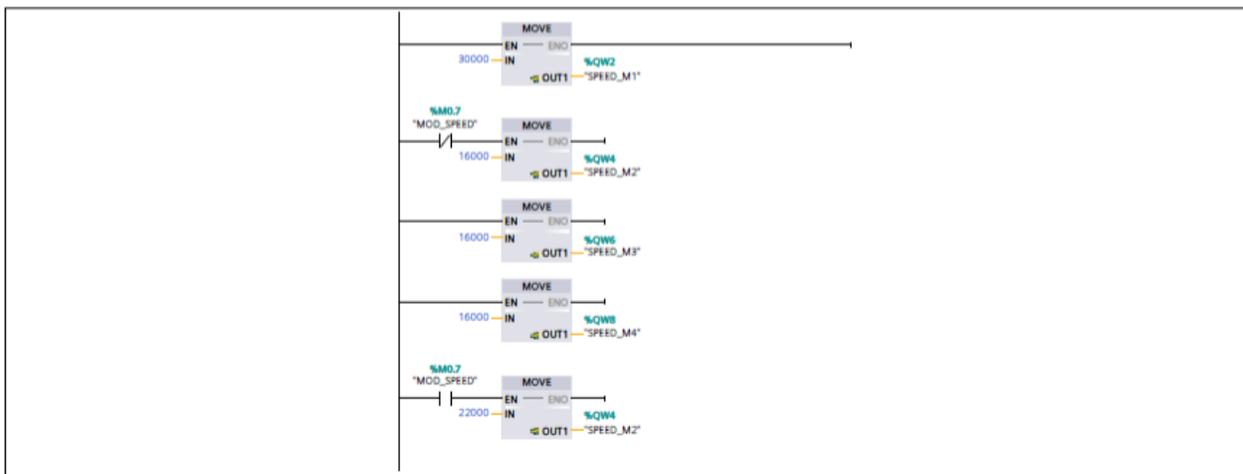


Figura 3.21 Peldaño 6 de la rutina LD

En la Figura 3.22 se muestra el peldaño 7, donde se realiza la configuración del encoder para posicionar al carro de la cinta transportadora. Esta configuración se lleva a cabo cuando la variable *ENCODER_1* se activa, lo que permite que un contador ascendente registre los pulsos provenientes de una señal del encoder para posicionar el carro en la ubicación deseada. Además, se cuenta con la variable *RESET_ENCOD* que permite reiniciar el contador.

Network 7: CONTADOR DE ENCODER PARA POSICIONAR LA CINTA TRANSPORTADORA

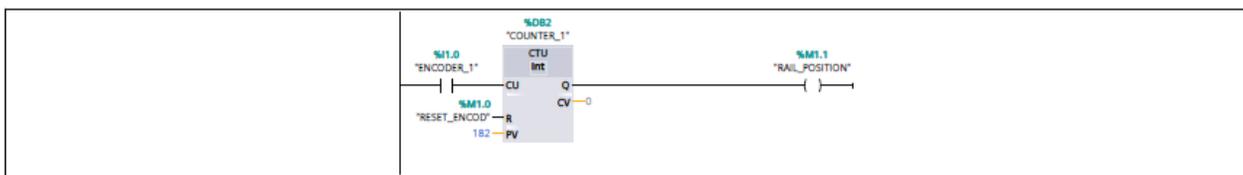


Figura 3.22 Peldaño 7 de la rutina LD

En la Figura 3.23 se muestra el peldaño 8, donde la variable *"ModuloReciclaje_DB".S_NO*, proveniente de la rutina SFC y que indica el paso actual, se transfiere a la variable *SFC_STEP*. Esta acción se realiza porque la interfaz no puede acceder directamente a la dirección de la variable *"ModuloReciclaje_DB".S_NO*. Por lo tanto, para poder leer su información, es necesario transferirla a

una variable a la que la interfaz sí pueda acceder. Es importante recordar que la interfaz está diseñada con la herramienta VoT, la cual presenta ciertas limitaciones que pueden causar este tipo de problemas.

Network 8: INDICADOR DE ESTADO ACTUAL DE LA RUTINA SFC



Figura 3.23 Peldaño 8 de la rutina LD

En la Figura 3.24 se muestra el peldaño 9, donde se lleva a cabo la cuantificación del primer indicador de rendimiento, que es el voltaje en corriente alterna consumido por el cuarto módulo en general. Este proceso se logra escalando la medición proveniente del módulo medidor de energía, dado que esta se encuentra en [mV].

Network 9: CALCULO DEL VOLTAJE AC CONSUMIDO

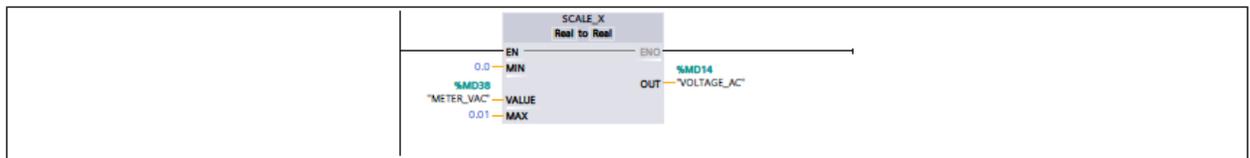


Figura 3.24 Peldaño 9 de la rutina LD

En la Figura 3.25 se muestra el peldaño 10, donde se lleva a cabo la cuantificación del segundo indicador de rendimiento, que es la corriente alterna consumida por el cuarto módulo en general. Este proceso se logra mediante la linealización de las mediciones provenientes del convertidor analógico-digital del módulo medidor de energía de la periferia. Posteriormente, se limitan las mediciones de corriente en un rango mayor a 0 [A]. Esto se debe a que el transformador de corriente utilizado para medir la corriente tiene un rango de medición mínimo de 0.8 [A]. Por debajo de este límite, las mediciones entregadas son erróneas.

Network 10: CALCULO DE LA CORRIENTE AC CONSUMIDA

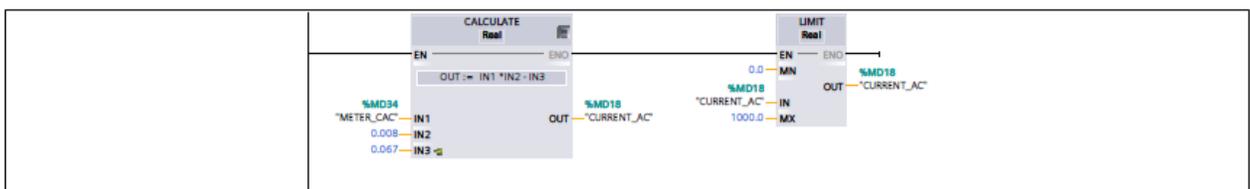


Figura 3.25 Peldaño 10 de la rutina LD

En la Figura 3.26 se muestra el peldaño 11, donde se lleva a cabo la cuantificación del tercer indicador de rendimiento, que es la potencia instantánea en corriente alterna consumida por el cuarto

módulo en total. Este proceso se logra multiplicando el voltaje y la corriente en alterna previamente calculados.

Network 11: CALCULO DE LA POTENCIA INSTANTANEA AC CONSUMIDA

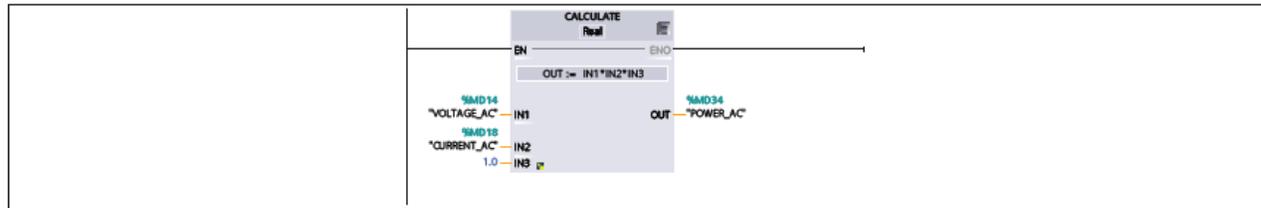


Figura 3.26 Peldaño 11 de la rutina LD

En el peldaño 12 se lleva a cabo la cuantificación del cuarto indicador de rendimiento, que es la corriente continua consumida por uno de los motores DC. Es importante tener en cuenta que el convertidor analógico-digital del módulo de entradas analógicas utiliza 27,648 muestras. Además, el rango de voltaje que este módulo puede medir es de 1 a 5 [V].

Dado esto, se requiere realizar una interpolación lineal para estimar los valores fuera del rango de medición. Aunque el consumo de corriente del motor no excederá este rango, esta práctica se realiza con el fin de aumentar la robustez del procesamiento de la señal de corriente medida.

$$m = \frac{muestra_{max} - muestra_{min}}{V_{max} - V_{min}} \quad (3.1)$$

$$m = \frac{27648 - 0}{5 - 1} = 6912 \quad (3.2)$$

Para calcular el voltaje real medido por el transductor LEM se determinó la siguiente relación:

$$V_o = \frac{muestras_{medidas}}{m} + 1 \text{ [V]} \quad (3.3)$$

$$V_o = \frac{muestras_{medidas}}{6912} + 1 \text{ [V]} \quad (3.4)$$

Posteriormente, es necesario convertir el voltaje medido a la corriente real. Según [LEM, 2017], el voltaje de salida en el transductor es:

$$V_o = 2.5 + \left(\frac{0.625 * I_P}{I_{PN}} \right) \text{ [V]} \quad (3.5)$$

Donde I_P representa la corriente real medida, y I_{PN} es la corriente primaria nominal RMS del transductor. Por lo tanto, se procede a despejar I_P de la expresión 3.5:

$$I_P = \frac{(V_o - 2.5) * I_{PN}}{0.625} \text{ [A]} \quad (3.6)$$

Es importante destacar que I_{PN} depende de la configuración empleada en la bobina primaria. Como se observa en la Tabla 3.9, la configuración utilizada es la tercera, la cual proporciona una mayor sensibilidad en las mediciones al aumentar la inductancia del transductor. Por lo tanto, $I_{PN} = 5 \text{ [A]}$:

$$I_P = \frac{(V_o - 2.5) * 5}{0.625} [A] \tag{3.7}$$

Sustituyendo el resultado de la expresión 3.4 en la expresión 3.7, se obtiene la corriente real medida del motor.

Con las expresiones anteriores desarrolladas, se procede a implementarlas en la rutina LD utilizando la instrucción *CALCULATE*. En la primera instrucción se calcula la expresión 3.4, y en la segunda instrucción se calcula la expresión 3.7. Posteriormente, se limitan las mediciones de corriente en un rango mayor a 0 [A], tal como se muestra en la Figura 3.27.

Network 12: CALCULO DE LA CORRIENTE DC CONSUMIDA

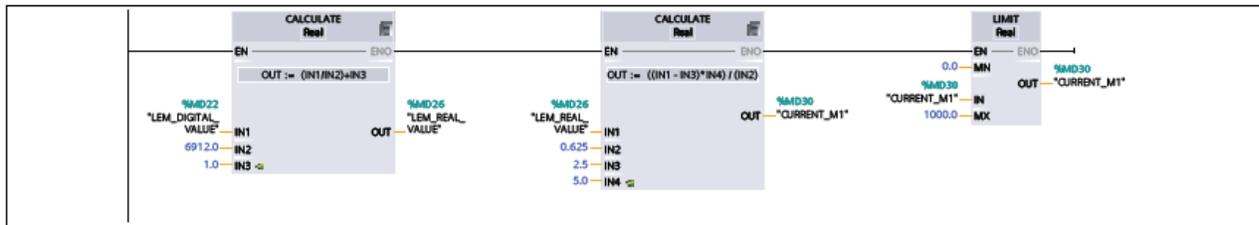


Figura 3.27 Peldaño 12 de la rutina LD

Number of primary turns	Primary nominal RMS current	Nominal output voltage V_{out}	Primary resistance R_P [mΩ]	Primary insertion inductance L_P [μH]	Recommended connections
1	±15	2.5 ±0.625	0.18	0.013	<pre> 6 5 4 OUT ○ ○ ○ IN 1 2 3 </pre>
2	±7.5	2.5 ±0.625	0.81	0.05	<pre> 6 5 4 OUT ○ ○ ○ IN 1 2 3 </pre>
3	±5	2.5 ±0.625	1.62	0.12	<pre> 6 5 4 OUT ○ ○ ○ IN 1 2 3 </pre>

Tabla 3.9 Configuraciones de la bobina primaria del transductor LEM

En la Figura 3.28 se muestra el peldaño 13, donde se lleva a cabo un arreglo lógico para que una variable indique cuando el motor 2 está en movimiento. Esto es necesario porque el motor tiene dos variables que lo hacen moverse, y cada una corresponde a un sentido de giro.

Network 13: INDICADOR DE MOVIMIENTO DEL MOTOR 2

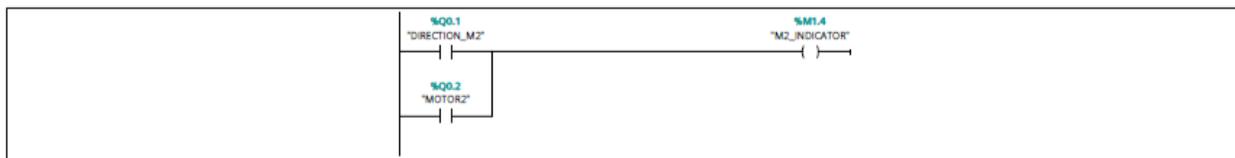


Figura 3.28 Peldaño 13 de la rutina LD

En la Figura 3.29 se muestra el peldaño 14, donde se lleva a cabo el mismo arreglo lógico que se realizó en el peldaño 13, pero para el motor 4.

Network 14: INDICADOR DE MOVIMIENTO DEL MOTOR 4

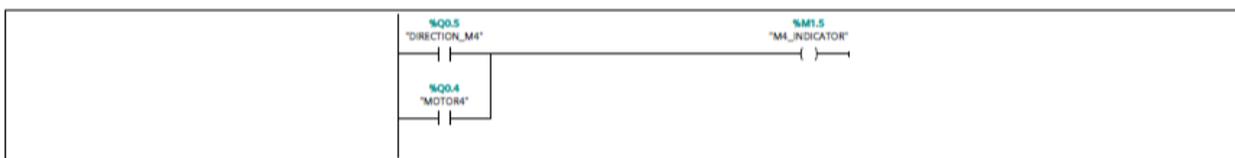


Figura 3.29 Peldaño 14 de la rutina LD

En la Figura 3.30 se muestra el peldaño 15, donde se lleva a cabo el mismo arreglo lógico que se realizó en el peldaño 13, pero para el motor 5.

Network 15: INDICADOR DE MOVIMIENTO DEL MOTOR 5

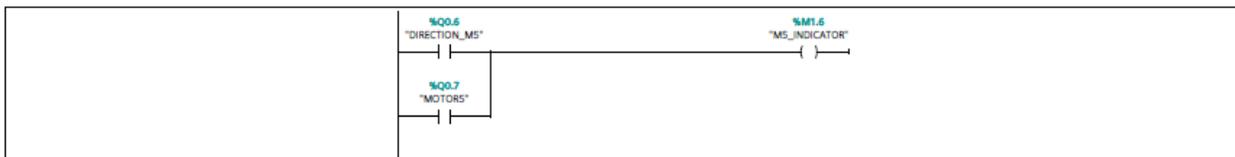


Figura 3.30 Peldaño 15 de la rutina LD

En la Figura 3.31 se muestra el peldaño 16, donde se lleva a cabo el mismo arreglo lógico que se realizó en el peldaño 13, pero ahora para indicar cuando el robot se encuentra en movimiento. Esto se debe a que hay dos variables que indican cuando el robot está desempeñando una trayectoria.

Network 16: INDICADOR DE MOVIMIENTO DEL ROBOT

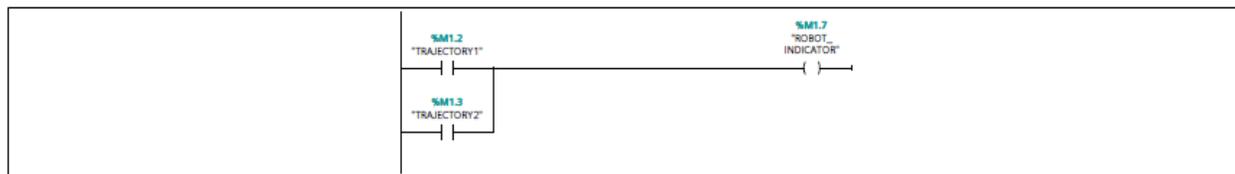


Figura 3.31 Peldaño 16 de la rutina LD

Finalmente, en la Figura 3.32 se muestra el peldaño 17, donde se diseñó un oscilador de 1 segundo de encendido y 1 segundo de apagado, que se activa cuando la variable *ROBOT_INDICATOR* está activa. Esto se realiza con la única finalidad de animar la imagen del robot en la HMI web.

Network 17: ANIMACIÓN DEL ROBOT EN HMI

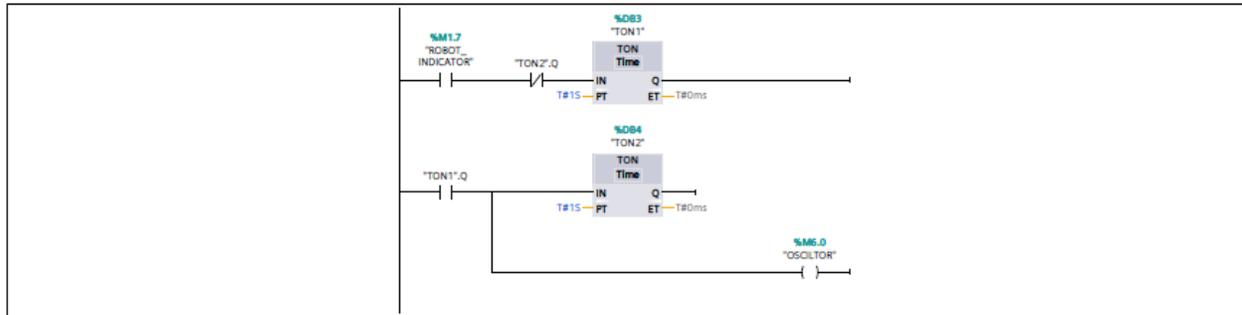


Figura 3.32 Peldaño 17 de la rutina LD

3.5.2. Rutina de control SFC

En la Figura 3.33 se muestra el estado 1, donde se realiza la inicialización de todas las variables a utilizar. Por otra parte, en la transición 1 se espera el cambio de estado en la variable *START*, la cual se encuentra en la rutina LD e indica que el botón verde físico o digital fue accionado.

S1 - [Initial step];Step1

INICIALIZACION DE TODAS LAS VARIABLES

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"ON"
		R	"DIRECTION_M1"
		R	"MOTOR2"
		R	"DIRECTION_M2"
		R	"DIRECTION_M3"
		R	"MOTOR4"
		R	"DIRECTION_M4"
		R	"MOTOR5"
		R	"DIRECTION_M5"
		S	"EV1"
		R	"EV2"
		S	"EV3"
		R	"EV4"
		R	"EV5"
		R	"B0_SHIELD"
		R	"B1_SHIELD"
		R	"B2_SHIELD"
		R	"MOD_SPEED"
		R	"RESET_ENCOD"
		R	"TRAJECTORY1"
		R	"TRAJECTORY2"

T1:Trans1

INICIO DE LA RUTINA CON EL BOTÓN VERDE FISICO O EL DIGITAL

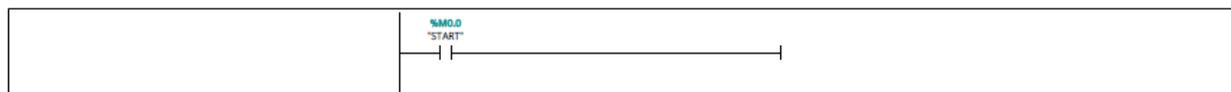


Figura 3.33 Estado y transición 1 de la rutina SFC

En la Figura 3.34 se muestra el estado 2, donde se activa la variable *ON* que indica que el proceso ha iniciado y la variable *TRAJECTORY1* indica que el robot comenzará a moverse, a la par que se activa el motor 5 ubicado dentro del carro de la cinta transportadora, con la finalidad de empujar la caja hacia la base de recepción. Por otra parte, en la transición 2 se da un retardo de 1 segundo con la finalidad de dejar caer la caja de manera correcta.

S2:Step2

LA RUTINA INICIA, SE ACTIVA EL MOTOR 5 Y LA VARIABLE DE TRAYECTORIA 1

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		S	"ON"
		S	"DIRECTION_M5"
		S	"TRAJECTORY1"

T2:Trans2

SE DA UN RETARDO DE 1 SEGUNDO

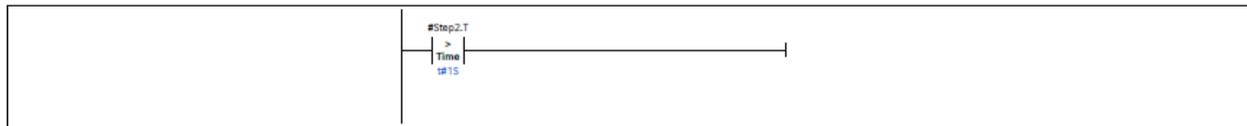


Figura 3.34 Estado y transición 2 de la rutina SFC

En la Figura 3.35 se muestra el estado 3, donde se detiene al motor 5 y se activa el motor 4 que es el encargado de mover al carro hacia la banda transportadora. Por otra parte, en la transición 3 se espera a que el sensor de límite 1 detecte que el carro llegó a la posición deseada.

S3:Step3

SE DETIENE EL MOTOR 5 Y SE ACTIVA EL MOTOR 4

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"DIRECTION_M5"
		S	"MOTOR4"

T3:Trans3

SE ACTIVA EL SENSOR 5 CUANDO EL CARRO DEL RIEL LLEGA A LA BANDA



Figura 3.35 Estado y transición 3 de la rutina SFC

En la Figura 3.36 se muestra el estado 4, donde se desactiva la variable *ON* con la finalidad de evitar que la rutina vuelva a iniciar antes de que termine la actual, al igual que se apaga el motor 4. Por otra parte, en la transición 4 se espera a que el robot mande la señal de que ya terminó de ejecutar su primera trayectoria.

S4:Step4

APAGA LA VARIABLE DE ENCENDIDO Y EL MOTOR 4

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"ON"
		R	"MOTOR4"

T4:Trans4

ESPERA A QUE EL ROBOT TERMINE LA PRIMER TRAYECTORIA



Figura 3.36 Estado y transición 4 de la rutina SFC

En la Figura 3.37 se muestra el estado 5, donde se activa el motor 1 correspondiente a la banda transportadora superior, con la finalidad de llevar el empaque con botellas llenas hacia la rampa de descenso. Por otra parte, en la transición 5 se espera a que el empaque comience a pasar frente a sensor 1.

S5:Step5

ACTIVA EL MOTOR 1

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		S	"DIRECTION_M1"

T5:Trans5

DETECTA CUANDO LA CAJA COMIENZA A PASAR POR EL SENSOR

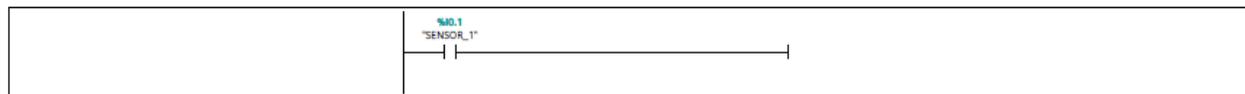


Figura 3.37 Estado y transición 5 de la rutina SFC

En la Figura 3.38 se muestra el estado 6, donde no se ejecuta ninguna acción. Por el contrario, en la transición 6 se espera que el sensor 1 deje de detectar al empaque.

S6:Step6

NO SUCEDE NADA

Actions:

Actions:			
Interlock	Event	Qualifier	Action

T6:Trans6

ESPERA A QUE LA CAJA TERMINE DE PASAR POR EL SENSOR



Figura 3.38 Paso y transición 6 de la rutina SFC

En la Figura 3.39 se muestra el estado 7, donde se detiene al motor 1, a la vez que se baja la rampa, esto con la finalidad de bajar al empaque a la banda inferior y siga avanzando. Por otra parte, en la transición 7 se verifica a manera de redundancia, que el sensor no este detectando nada, y que el motor de la banda superior se encuentre apagado.

S7:Step7

DETIENE EL MOTOR 1, ACTIVA EL MOTOR 2, Y BAJA LA RAMPA

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"DIRECTION_M1"
		S	"MOTOR2"
		R	"EV1"
		S	"EV2"

T7:Trans7

COMO REDUNDANCIA SE VERIFICA QUE EL SENSOR 1 NO DETECTE NADA Y EL MOTOR 1 ESTE APAGADO

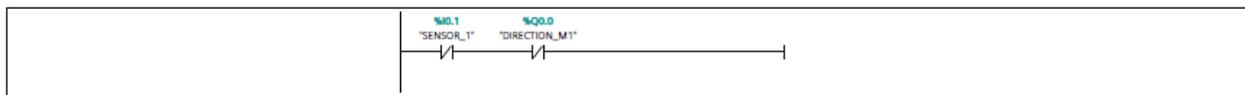


Figura 3.39 Estado y transición 7 de la rutina SFC

En la Figura 3.40 se muestra el estado 8, donde se activa el motor 2 correspondiente a la banda transportadora inferior. Por otra parte, en la transición 8 se espera a que el sensor 2 deje de detectar el empaque y lo comience a detectar el sensor 3, esto con la finalidad de posicionar al empaque justo debajo del pistón que presionará a las botellas.

S8:Step8

ACTIVA EL MOTOR 2

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		S	"MOTOR2"

T8:Trans8

LA CONDICION ES VALIDA CUANDO EL SENSOR 2 DEJA DE DETECTAR A LA CAJA Y LA EMPIECE A DETECTAR EL SENSOR 3

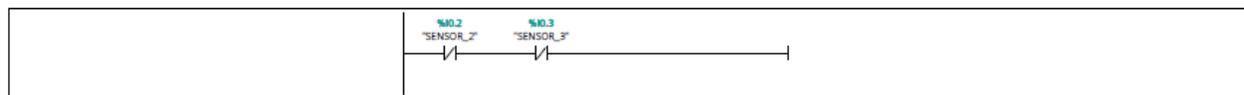


Figura 3.40 Estado y transición 8 de la rutina SFC

En la Figura 3.41 se muestra el estado 9, donde se detiene al motor 2 y se activa el pistón que presionará a las botellas, a la vez que desactiva la variable *TRAJECTORY1* que indica que el robot termino de ejecutar la primer trayectoria. Por otra parte, en la transición 9 se da un retardo de 2 segundos para simular el proceso de vaciado de las botellas.

S9:Step9

DETIENE AL MOTOR 2, BAJA EL PISTON QUE PRESIONA LAS BOTELLAS, TERMINA LA TRAYECTORIA 1

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"MOTOR2"
		R	"EV3"
		S	"EV4"
		R	"TRAJECTORY1"

T9:Trans9

DA UN RETARDO DE 2 SEGUNDOS PARA SIMULAR EL PROCESO DE VACIADO DE BOTELLAS

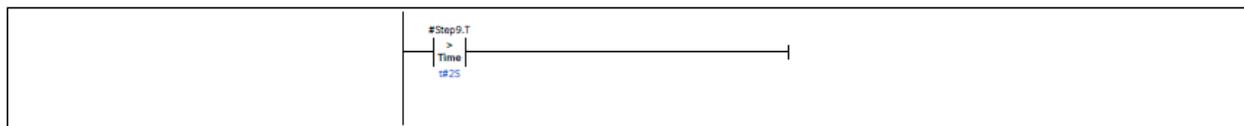


Figura 3.41 Estado y transición 9 de la rutina SFC

En la Figura 3.42 se muestra el estado 10, donde se vuelve a activar el motor 2 para que el empaque ahora con botellas vacías llegue a la zona de recolección de botellas, a la par que desactiva el pistón que simula el vaciado para que permita avanzar al empaque, y también activa la variable *TRAJECTORY2* que indica que el robot volverá a moverse para ejecutar la segunda trayectoria. Por otra parte, la transición 10 espera a que los sensores 2 y 3 dejen de detectar al empaque con botellas vacías.

S10:Step10

VUELVE A ACTIVAR EL MOTOR 2, SUBE EL PISTON QUE PRESIONA LAS BOTELLAS Y ACTIVA LA TRAYECTORIA 2

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		S	"MOTOR2"
		S	"EV3"
		R	"EV4"
		S	"TRAJECTORY2"

T10:Trans10

EL SENSOR 2 Y 3 DEJAN DE DETECTAR LA CAJA



Figura 3.42 Estado y transición 10 de la rutina SFC

En la Figura 3.43 se muestra el estado 11, donde se detiene al motor 2, se activa el pistón de sujeción y la variable *B2_SHIELD* que ejecuta los comandos de movimiento correspondientes a la segunda trayectoria. Esto se hace con el propósito de posicionar la caja exactamente frente al último pistón. De esta manera, dicho pistón detendrá la caja, asegurando que no se vaya junto con las botellas cuando el brazo robótico baje para recogerlas. Por otra parte, la transición 11 espera a que el robot mande la señal de que ya terminó de ejecutar su segunda trayectoria.

S11:Step11

DETIENE AL MOTOR 2, ACTIVA AL PISTON QUE SUJETA LA CAJA, Y ACTIVA LA 2DA TRAYECTORIA DEL ROBOT

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"MOTOR2"
		S	"EV5"
		S	"B2_SHIELD"

T11:Trans11

ESPERA A QUE EL ROBOT TERMINE SU 2DA TRAYECTORIA



Figura 3.43 Estado y transición 11 de la rutina SFC

En la Figura 3.44 se muestra el estado 12, donde inicialmente se desactiva la variable *B2_SHIELD*, luego se activa la variable *MOD_SPEED* para aumentar la velocidad del motor 2. Además, se activa el motor 2 de la banda transportadora inferior en dirección al carro transportador. Simultáneamente, se eleva la rampa y se desactiva el pistón que sujeta la caja. Por otra parte, la transición 12 espera a que el sensor 2 detecte el paso de la caja vacía en dirección al carro de transporte.

S12:Step12

DESACTIVA EL BIT DE LA 2DA TRAYECTORIA DE ROBOT, ACTIVA EL BIT QUE CAMBIA LA VELOCIDAD DEL MOTOR 2, SUBE LA RAMPA Y DESACTIVA EL PISTON DE SUJECIÓN

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"B2_SHIELD"
		S	"MOD_SPEED"
		S	"DIRECTION_M2"
		S	"EV1"
		R	"EV2"
		R	"EV5"

T12:Trans12

ESPERA A QUE EL SENSOR 2 DETECTE LA CAJA VACIA



Figura 3.44 Paso y transición 12 de la rutina SFC

En la Figura 3.45 se muestra el estado 13, donde se vuelve a activar el motor 5 situado dentro del carro de transporte, pero esta vez en dirección opuesta al paso 2. Esto se hace para subir la caja hacia el carro en lugar de empujarla hacia fuera. Además, se activa el reinicio del encoder como medida de seguridad. Esto se realiza en caso de que el contador haya registrado algún movimiento del motor durante los pasos anteriores, asegurando así llegar a la posición deseada. Por otra parte, la transición 13 espera a que el sensor 7 detecte que la caja vacía ya subió al carro transportador.

S13:Step13

ACTIVA EL MOTOR 5 PARA QUE LA CAJA SUBA AL CARRO DE TRANSPORTE Y REINICIA EL CONTADOR DEL ENCODER

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		S	"MOTORS"
		S	"RESET_ENCOD"

T13:Trans13

ESPERA A QUE LA CAJA SUBA AL CARRO DE TRANSPORTE

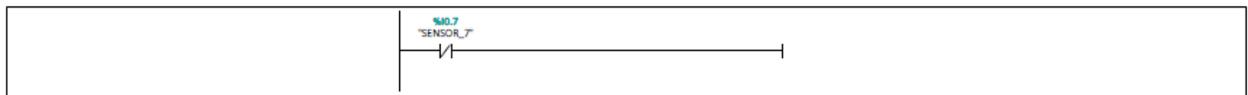


Figura 3.45 Estado y transición 13 de la rutina SFC

Finalmente, en la Figura 3.46 se muestra el estado 14, donde se desactiva el motor 2 de la banda inferior, también se desactiva el motor 5 del carro transportador, también se desactiva la variable de reinicio del encoder, y nuevamente se activa el motor 4 pero en dirección del extremo contrario de la cinta transportadora donde se encuentra el carro, todo lo anterior al mismo tiempo. Por otra parte, la transición 14 espera a que el encoder active la variable *RAIL_POSITION*, la cual indica que el carro transportador junto a la caja vacía se encuentran frente a la base de recepción.

S14:Step14

DESACTIVA EL MOTOR 2 Y 5, APAGA EL REINICIO DEL ENCODER Y ACTIVA EL MOTOR 4

Actions:

Actions:			
Interlock	Event	Qualifier	Action
		R	"DIRECTION_M2"
		R	"MOTORS"
		R	"RESET_ENCOD"
		S	"DIRECTION_M4"

T14:Trans14

ESPERA A QUE EL CARRO SE COLOQUE EN LA POSICIÓN DESEADA DE ACUERDO AL CONTADOR PARA TERMINAR EL PROCESO

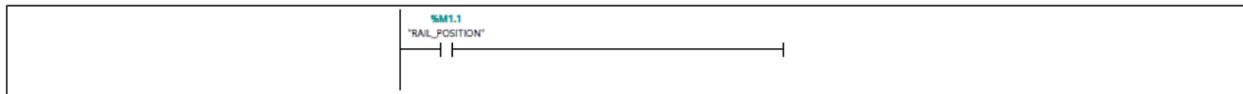


Figura 3.46 Estado y transición 14 de la rutina SFC

De esta manera termina la rutina de control SFC, regresando al paso 1 a esperar que nuevamente se cumpla la condición de transición 1 para volver a empezar la rutina.

3.5.3. Rutina de control para movimiento de motores DC

Esta rutina, desarrollada en lenguaje LD, está diseñada para controlar el movimiento de cualquier motor del cuarto módulo. Específicamente, este control de movimiento incluye un arranque suave para los motores de corriente directa, cuyo objetivo es prevenir un mayor desgaste en las bandas transportadoras debido al estrés mecánico generado por un arranque directo con el máximo torque del motor. Además, busca mejorar el consumo de corriente, lo que resulta en un consumo energético más eficiente y, por ende, una optimización del módulo. En un caso más realista, se pretende evitar que el producto sufra perturbaciones que puedan ocasionar cambios en su calidad o simplemente evitar que el producto se derrame.

Primeramente, hay que mencionar los requerimientos necesarios para el arranque suave. El usuario podrá seleccionar el tiempo en el que el motor llegue a su velocidad máxima, este tiempo puede ser de 1, 2, 3, 4 o 5 segundos. También se debe tener una variable que permita activar el motor, y otra que active o desactive el arranque suave.

Una vez conocidos los requerimientos necesarios, hay que destacar que el convertidor digital-analógico del controlador del motor es de 16 bits. Además, el motor cuenta con una banda muerta, donde con un valor menor a 6000 muestras el motor no se mueve, lo que es equivalente a un voltaje de 4.39 [V]. Por otra parte, a partir de las 30000 muestras, o 21.97 [V], el incremento de velocidad es mínimo, por lo que aumentar el voltaje se vuelve ineficiente. En la Figura 3.47 se muestra el comportamiento de la velocidad del motor en función del número de muestras que se envían al controlador y el voltaje que recibe el motor.

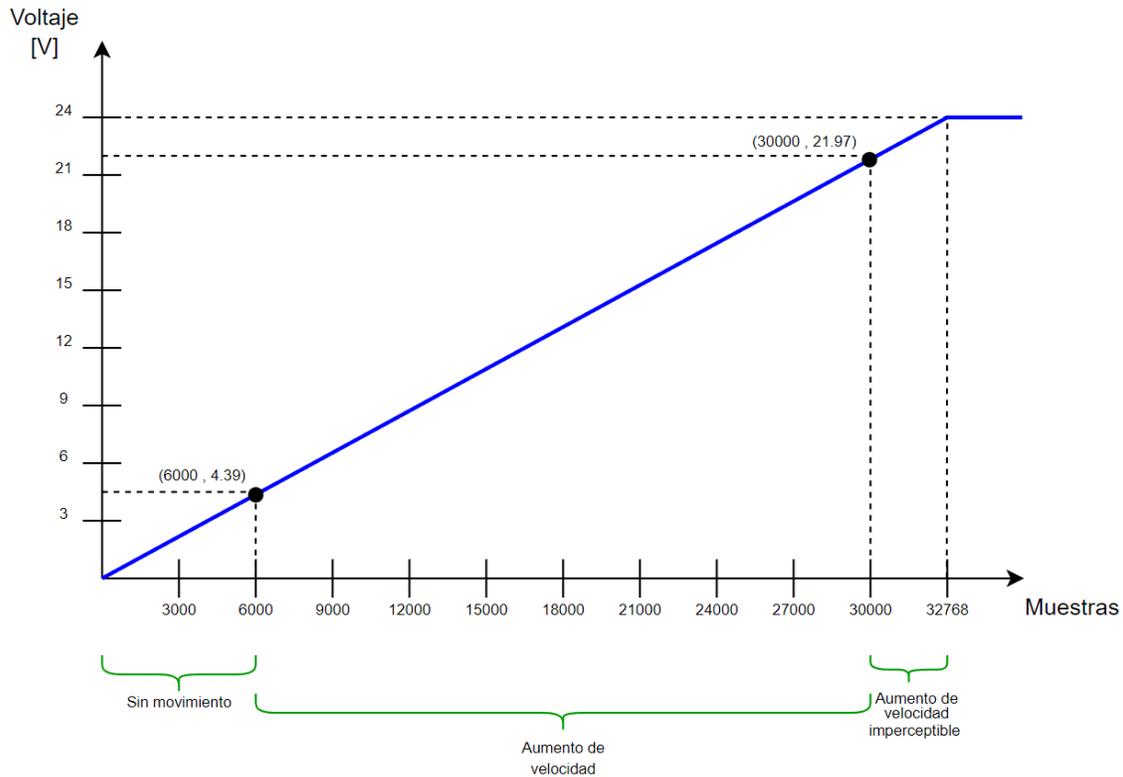


Figura 3.47 Comportamiento en la velocidad del motor en función del número de muestras que se envían al controlador y el voltaje que recibe el motor

Como se puede observar, el cambio considerable en la velocidad sucede entre las 6000 y 30000 muestras, por lo que es necesario variar la rampa de voltaje entre este rango. De esta manera, se obtiene la siguiente expresión que proporciona el valor de muestras para realizar el arranque suave.

$$muestras = offset + rampa \quad (3.8)$$

Donde el *offset* es de 6000 muestras y el valor de *rampa* siempre debe ser de 24000. El problema se presenta ya que el valor de la rampa depende del acumulador del temporizador con el tiempo seleccionado por el usuario, así que este acumulador debe estar multiplicado por un factor que le permita llegar a las 24000 muestras sin importar el tiempo seleccionado. Para lo anteriormente mencionado, se diseñó una manera de obtener una expresión que siempre entregue un factor correspondiente al tiempo de rampa deseado. Acotados a los requerimientos, se optó por desarrollar un polinomio de Lagrange. En la Tabla ?? se muestran los puntos deseados a interpolar.

Tiempo de rampa [s]	Factor
1	24
2	12
3	8
4	6
5	4.8

Tabla 3.10 Puntos a interpolar con el polinomio de Lagrange

Partiendo de la expresión del polinomio de Lagrange:

$$P(x) := \sum_{i=0}^n y_i l_i(x) \quad (3.9)$$

Donde y_i son las coordenadas en y de los puntos dados, y $l_i(x)$ son los polinomios base de Lagrange, que se definen como:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{y_i - y_j} \quad (3.10)$$

Aplicando esta expresión a los puntos de interpolación:

$$P(x) = 24 \cdot \frac{(x-2)(x-3)(x-4)(x-5)}{24} + 12 \cdot \frac{(x-1)(x-3)(x-4)(x-5)}{-6} \\ + 8 \cdot \frac{(x-1)(x-2)(x-4)(x-5)}{6} + 6 \cdot \frac{(x-1)(x-2)(x-3)(x-5)}{-24} \\ + 4.8 \cdot \frac{(x-1)(x-2)(x-3)(x-4)}{24} \quad (3.11)$$

Finalmente expandiendo y simplificando la expresión anterior se obtiene el polinomio de Lagrange:

$$P(x) = factor = 0.2x^4 - 3x^3 + 17x^2 - 45x + 54.8 \quad (3.12)$$

Con el polinomio anterior se obtendrá el factor para cada tiempo de rampa, garantizando siempre obtener las 24000 muestras. Este factor se multiplicará por el tiempo de rampa deseado en milisegundos, y al sumarles el desfase de 6000 muestras, siempre dará como resultado las 30000 muestras, donde el motor muestra su mejor desempeño de velocidad.

$$muestras = offset + (t_{rampa})(Factor) \quad (3.13)$$

En otras palabras, se acota el arranque suave en un intervalo de voltaje variable entre 4.39 a 21.97 [V], sin importar el tiempo de rampa seleccionado. Esto se muestra en la Tabla ??.

t_{rampa}	Factor	offset + (t_{rampa})(Factor)	Muestras
1000	$0.2(1)^4 - 3(1)^3 + 17(1)^2 - 45(1) + 54.8$	$6000 + (1000)(24)$	30000
2000	$0.2(2)^4 - 3(2)^3 + 17(2)^2 - 45(2) + 54.8$	$6000 + (2000)(12)$	30000
3000	$0.2(3)^4 - 3(3)^3 + 17(3)^2 - 45(3) + 54.8$	$6000 + (3000)(8)$	30000
4000	$0.2(4)^4 - 3(4)^3 + 17(4)^2 - 45(4) + 54.8$	$6000 + (4000)(6)$	30000
5000	$0.2(5)^4 - 3(5)^3 + 17(5)^2 - 45(5) + 54.8$	$6000 + (5000)(4.8)$	30000

Tabla 3.11 Cálculo de muestras para los diferentes tiempos de rampa

Una vez obtenidas las expresiones necesarias para calcular la rampa para el arranque suave, se procedió a implementarlo en una rutina LD de acuerdo a los requerimientos solicitados.

En la Figura 3.48 se muestra el peldaño 1, donde se declara la variable que encenderá y apagará el motor.

Network 1: VARIABLE PARA ENCENDER EL MOTOR

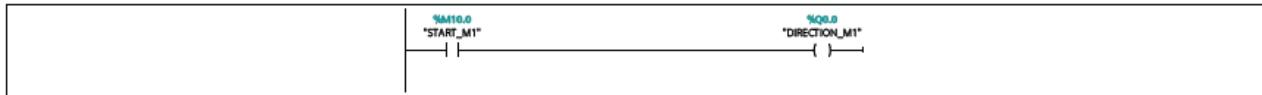


Figura 3.48 Peldaño 1 de la rutina de control de movimiento

En la Figura 3.49 se muestra el peldaño 2, donde se coloca la variable que activa el arranque suave. Cuando la variable está activa, permite a la rutina de control realizar el gradual aumento de velocidad de acuerdo con el tiempo de rampa seleccionado; por el contrario, si está desactivada, envía las 30000 muestras directamente al controlador del motor.

Network 2: VARIABLE PARA ACTIVAR EL ARRANQUE SUAVE



Figura 3.49 Peldaño 2 de la rutina de control de movimiento

En la Figura 3.50 se muestra el peldaño 3, donde se coloca la variable para que el usuario pueda ingresar el tiempo de rampa seleccionado.

Network 3: VARIABLE PARA ASIGNAR EL TIEMPO DE RAMPA



Figura 3.50 Peldaño 3 de la rutina de control de movimiento

En la Figura 3.51 se muestra el peldaño 4, donde se procede a realizar el cálculo del factor de acuerdo con la expresión 3.12.

Network 4: CALCULO DEL FACTOR CON EL POLINOMIO DE LAGRANGE

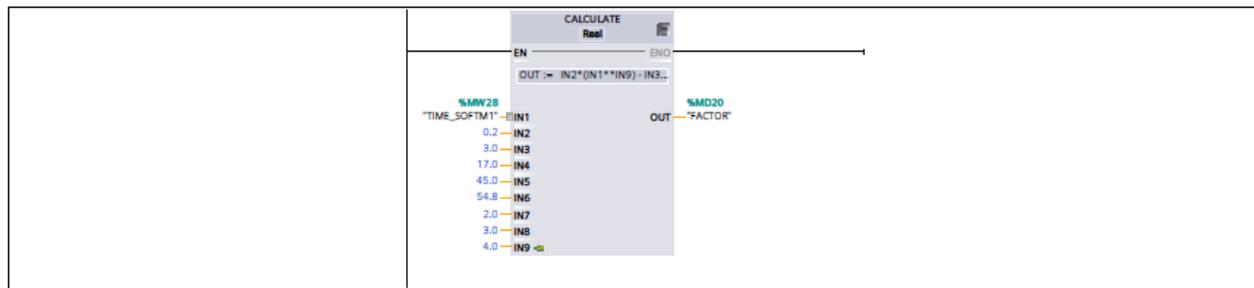


Figura 3.51 Peldaño 4 de la rutina de control de movimiento

En la Figura 3.52 se muestra el peldaño 5, donde se escala el tiempo de rampa para pasarlo de segundos a milisegundos, ya que es el tipo de dato que acepta el parámetro preestablecido (PT) del temporizador.

Network 5: ESCALADO DEL TIEMPO DE RAMPA PARA EL TEMPORIZADOR

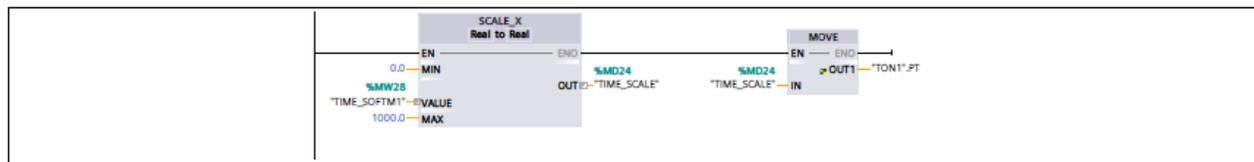


Figura 3.52 Peldaño 5 de la rutina de control de movimiento

En la Figura 3.53 se muestra el peldaño 6, donde se realiza un arreglo lógico que permite activar al temporizador cuando la variable de arranque suave y la de arranque del motor están encendidas. Recordando que este temporizador permite realizar la rampa del arranque suave en el tiempo seleccionado.

Network 6: TEMPORIZADOR PARA LA RAMPA SELECCIONADA

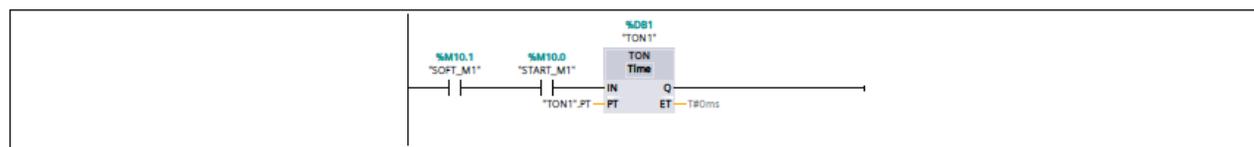


Figura 3.53 Peldaño 6 de la rutina de control de movimiento

En la Figura 3.54 se muestra el peldaño 7, donde se convierte el parámetro de tiempo transcurrido (ET) del temporizador a un tipo de dato real, con la finalidad de poder operar con dicho valor.

Network 7: CONVERSION DEL TIPO DE VARIABLE DEL TEMPORIZADOR A REAL

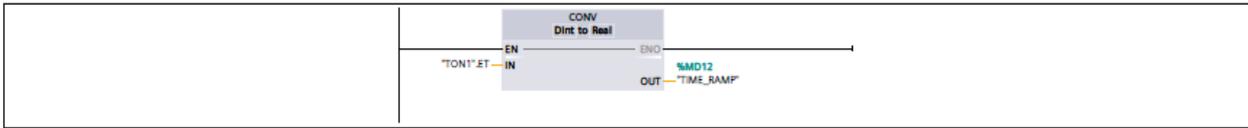


Figura 3.54 Peldaño 7 de la rutina de control de movimiento

Finalmente, en la Figura 3.55 se muestra el peldaño 8, donde se realiza el cálculo de las muestras que se envían al controlador de acuerdo con la expresión 3.13. Adicionalmente, se tiene un arreglo lógico que permite calcular las muestras solo cuando el temporizador del arranque suave está activo.

Network 8: CALCULO DE MUESTRAS PARA EL CONTROLADOR DEL MOTOR

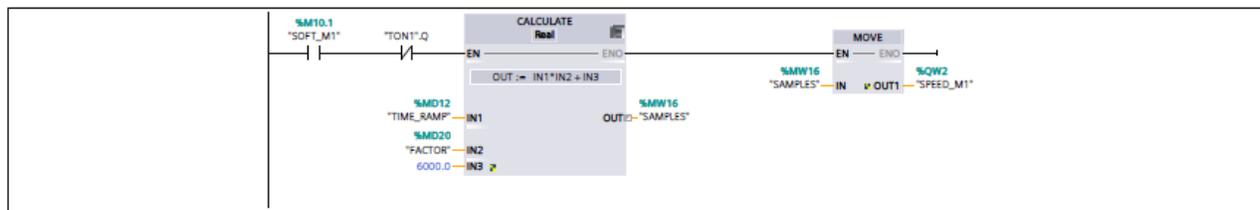


Figura 3.55 Peldaño 8 de la rutina de control de movimiento

Adicionalmente, en la Figura 3.27 se muestra la parte de la rutina de control principal que se reutilizó, correspondiente a la cuantificación de la corriente de un motor. Esto se hace con la finalidad de poder realizar mediciones de consumo y poder comparar el arranque suave con el arranque directo.

3.5.4. HMI web

La interfaz web se desarrolló con la herramienta VoT integrada al software TIA Portal. Esta es una HMI que tiene como propósito ser simple y eficiente. Como se puede observar en la Figura 3.56, la interfaz se divide en 3 paneles. En el primer panel se tiene el control del proceso donde se cuenta con un botón de arranque, uno de paro y otro para hacer un reinicio, un indicador de la etapa actual de la rutina SFC y otro indicador de estado del robot. Por otra parte, se dispone un panel de monitoreo de la eficiencia energética del módulo, donde se muestran indicadores clave de rendimiento tales como el voltaje, corriente y potencia. Por último, se tiene un panel de supervisión para los actuadores, donde se muestra el estado actual de cada motor y electroválvula.

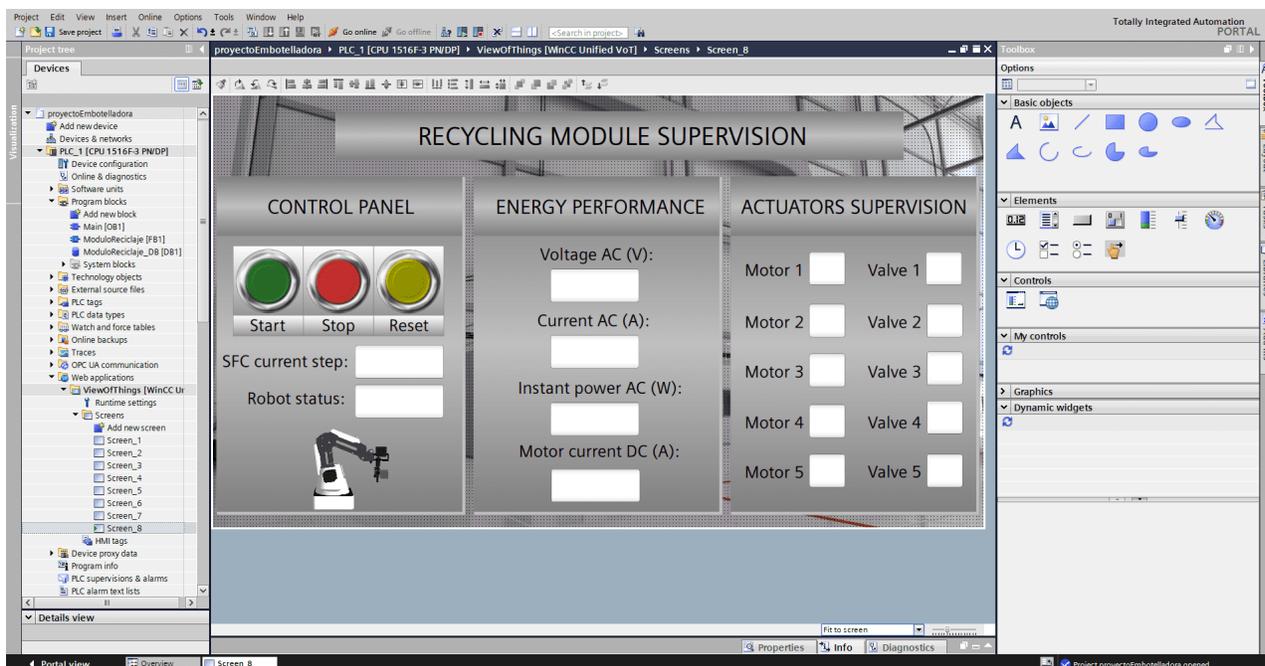


Figura 3.56 Interfaz de control y supervisión web

Capítulo 4

Resultados

4.1. Nivel de campo

Al retroceder y reflexionar sobre los primeros pasos del proyecto, se encontró un desafío inicial significativo, pues se enfrentó un prototipo el cual carecía de documentación. Más allá de identificar algunos sensores y actuadores a simple vista, se desconocían casi todos los detalles. No había información sobre su estado ni siquiera se sabía si seguían operativos. Por este motivo, la primera tarea fue llevar a cabo un reconocimiento completo de todos los dispositivos que componían el módulo, seguido de una investigación exhaustiva sobre cada elemento identificado. Una vez que quedó claro qué dispositivos estaban disponibles, se procedió a realizar un mantenimiento general. Este incluyó ajustar y alinear las bandas transportadoras, corregir el eje de los motores, reajustar los sensores, y también reubicar las estructuras plásticas y metálicas de soporte.

El siguiente desafío significativo consistió en comprender cómo se interconectaban los dispositivos, qué relaciones tenían entre sí y cómo se energizaban. Para abordar esto, se llevó a cabo una exhaustiva verificación de toda la instalación eléctrica. Esto implicó medir la continuidad de cada cable individualmente y rastrearlo a través de una maraña de cables para determinar sus puntos de conexión. Luego, utilizando esta información, se creó una tabla de ordenamiento de variables, con la cual se determinó el direccionamiento de la periferia descentralizada. Posteriormente, utilizando AutoCAD Electrical, se creó un diagrama detallado que muestra todos los dispositivos y sus conexiones eléctricas correspondientes. Este enfoque permitió tener una visión clara y sistemática de la configuración eléctrica del sistema, facilitando así cualquier tarea de diagnóstico o mantenimiento futuro.

Otro desafío se centró en el uso del brazo robótico, lo cual representó un verdadero reto dado que nunca se había trabajado con uno anteriormente, y menos integrado a un proceso específico. La primera tarea fue familiarizarse físicamente con el robot, comprendiendo tanto sus capacidades como sus limitaciones. Luego, se inició el proceso de aprendizaje de cómo programarlo, con el apoyo del software DobotStudio proporcionado por el fabricante (Figura 4.1). Este software ofrece una interfaz gráfica intuitiva que facilita controlar los movimientos del robot, programar secuencias de acciones y ejecutar tareas específicas mediante una programación visual o basada en código. Sin embargo, pese a tener un entorno de programación fácil y amigable, el robot no puede estar siempre conectado a una computadora para su operación. Por esta razón, cuenta con un modo fuera de línea que permite almacenar una rutina de movimientos predefinida, o bien, puede recibir varios comandos de movimiento a través de su puerto serie, ofreciendo así una mayor flexibilidad en su funcionamiento.

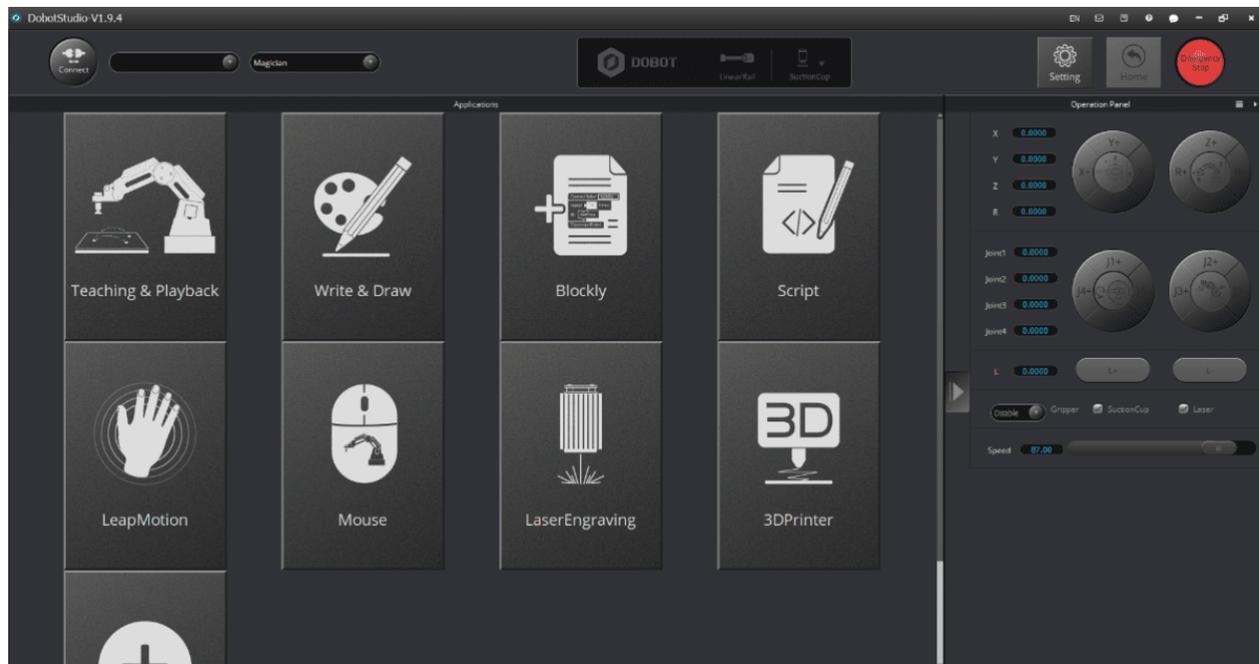


Figura 4.1 Interfaz principal del software DobotStudio

Es importante destacar que, según las recomendaciones del fabricante, se debe realizar un procedimiento de *Homing* siempre después de encender el robot. El *Homing* es un proceso de calibración diseñado para mejorar la precisión del posicionamiento. En la práctica, si no se realiza esta calibración, el robot no estará en su posición de referencia correcta, lo que ocasionará que no se alcancen las coordenadas deseadas. Para llevar a cabo este procedimiento, primero se debe encender el robot. Una vez encendido, su LED de estado será rojo debido a que, al apagarse previamente, sus límites quedaron fuera de rango y se generó una alarma. Pulsando el botón de *reset* en su interfaz física, el indicador parpadeará en color amarillo y, cinco segundos después, el LED será verde indicando que el reinicio se realizó adecuadamente. En este punto, es indispensable conectar una computadora con DobotStudio y presionar el botón de *Home* desde la interfaz gráfica, como se muestra en la Figura 4.2. Este procedimiento hará que la base del robot gire en sentido horario hasta el límite de la posición y luego regrese automáticamente al punto de referencia predeterminado. Este proceso dura unos pocos segundos y, mientras sucede, el indicador LED se volverá azul y parpadeará. Una vez terminado el proceso, se escuchará un pitido y el indicador LED volverá a verde.



Figura 4.2 Botón Home en DobotStudio

Por otra parte, se enfrentó otro desafío al implementar el transformador de corriente para medir la corriente alterna. Es importante señalar que en el diseño original del módulo, este transformador estaba previsto; pero en su lugar, solo había dos cables que se conectaban a la bornera del medidor de energía de la periferia. Inicialmente, se desconocía cómo el medidor de energía realizaba la medición de la corriente, pero después de examinar su datasheet, se comprendió que era imprescindible contar

con un transformador de corriente para esta tarea. Esto se debe a que el módulo no está diseñado para medir corrientes elevadas directamente, pero gracias al efecto de inducción electromagnética del transformador, se pueden realizar estas mediciones de manera segura. Aunque el módulo de reciclaje no demanda un alto consumo, aun así era necesario implementar un transformador.

Por esta razón, se tuvo que conseguir un transformador de corriente con la menor relación de transformación posible, y el que se logró conseguir fue uno de 30 [A] a 5 [A]. Se pasó el cable de fase de la alimentación eléctrica del módulo a través de este transformador toroidal, y finalmente se conectaron las terminales S1 y S2 del transformador a la bornera del medidor de energía. Este proceso permitió integrar de manera efectiva el transformador de corriente en el sistema, asegurando mediciones precisas y seguras de la corriente eléctrica.

Similar al problema anterior, se presentó otro desafío al implementar el transductor de corriente. En este caso, su inclusión no estaba contemplada en el diseño original, pero se decidió incorporarlo para disponer de un punto de referencia con el cual comparar los consumos de corriente en un motor. La instalación de este transductor fue relativamente sencilla, pues se soldaron algunas de sus patillas para configurarlo y aumentar la sensibilidad en las mediciones mediante el incremento de la inductancia en la bobina primaria. Una vez completada esta configuración, se conectó un cable de alimentación del motor a la patilla de entrada del transductor, y la salida se conectó directamente a la terminal correspondiente del motor. Dado que este transductor requiere alimentación externa, se utilizó una fuente independiente para suministrarle 5 [V]. Finalmente, se conectó la terminal *OUT* del transductor al módulo de entradas analógicas de la periferia para medir el consumo de corriente del motor. Con esta implementación, se obtuvo un medio confiable para monitorear y comparar los consumos de corriente de los motores, lo que contribuyó a una gestión más eficiente de la energía.

4.2. Nivel de control

Aunque la fase de puesta en marcha del control no fue la más prolongada, sí resultó ser la más desafiante debido a los diversos obstáculos que surgieron durante su desarrollo. Uno de los retos que se abordaron fue cómo controlar los movimientos del robot. Si bien ya se conocía cómo hacerlo desde la interfaz de usuario que ofrece DobotStudio, el problema surgió cuando no se disponía de ella y solo se tenían dos señales digitales provenientes del PLC para realizar todos los movimientos necesarios.

Para abordar esta cuestión, se aprovechó la pasarela de conexión instalada, recordando que esta pasarela se implementó utilizando un módulo de expansión industrial para Arduino. Además, el brazo robótico admite comandos de movimiento a través del puerto serie. En consecuencia, se desarrolló una rutina de control en código Arduino. En esta rutina, cuando se recibe un bit de activación proveniente del controlador, Arduino envía todos los comandos de movimiento necesarios a través del puerto serie para ejecutar la secuencia de movimientos deseada, a la que se llamará trayectoria. Asimismo, cuando se completa el envío de estas instrucciones, Arduino envía un bit de reconocimiento al PLC para indicar que el robot ha finalizado la ejecución de la trayectoria, como se muestra en el código 4.1.

```

1  /*****Copyright(c)*****/
2  **      Laboratorio Automatización, Facultad de Ingeniería, UNAM 2023
3  **      based on version of Shenzhen Yuejiang Technology Co., LTD.
4  **
5  **-----File Info-----
6  ** File name:      DobotDemo.ino
7  ** Latest modified Date: 26-11-2023
8  ** Latest Version:  V9
9  ** Descriptions:   for Recycling Module Supervision
10 **
11 **-----
12 ** Modify by:      Braulio Raygoza
13 ** Modified date:  13-12-2023
14 ** Version:        V10
15 ** Descriptions:   Modified initRAM and loop
16 **-----
17 *****/
18 //Libraries of dobot magician
19 #include "stdio.h"
20 #include "Protocol.h"
21 #include "command.h"
22 #include "FlexiTimer2.h"
23
24 //Libraries of IO shield
25 #include <Wire.h>
26 #include <Adafruit_ADS1X15.h>
27 #include <Adafruit_MCP23X17.h>
28
29 //Set Serial TX&RX Buffer Size
30 #define SERIAL_TX_BUFFER_SIZE 64
31 #define SERIAL_RX_BUFFER_SIZE 256
32
33 Adafruit_MCP23X17 mcp;
34 Adafruit_ADS1015 ads;
35
36
37 /*****
38 ** Global parameters
39 *****/
40 EndEffectorParams gEndEffectorParams;
41
42 PTPCoordinateParams gPTPCoordinateParams;
43 PTPCommonParams gPTPCommonParams;
44 PTPCmd          gPTPCmd;
45
46 uint64_t gQueuedCmdIndex;
47
48 /*****
49 ** Function name:      Serialread
50 ** Descriptions:      import data to rxbuffer
51 *****/
52 void Serialread()
53 {
54     while(Serial1.available()) {

```

```

55     uint8_t data = Serial1.read();
56     if (RingBufferIsFull(&gSerialProtocolHandler.rxRawByteQueue) == false) {
57         RingBufferEnqueue(&gSerialProtocolHandler.rxRawByteQueue, &data);
58     }
59 }
60 }
61
62 /*****
63 ** Function name:      Serial_putc
64 ** Descriptions:      Remap Serial to Printf
65 *****/
66 int Serial_putc( char c, struct __file * )
67 {
68     Serial.write( c );
69     return c;
70 }
71
72 /*****
73 ** Function name:      printf_begin
74 ** Descriptions:      Initializes Printf
75 *****/
76 void printf_begin(void)
77 {
78     fdevopen( &Serial_putc, 0 );
79 }
80
81 /*****
82 ** Function name:      InitRAM
83 ** Descriptions:      Initializes a global variable
84 *****/
85 void InitRAM(void)
86 {
87     gEndEffectorParams.xBias = 0;
88     gEndEffectorParams.yBias = 0;
89     gEndEffectorParams.zBias = 0;
90
91     //Set PTP Model
92     gPTPCoordinateParams.xyzVelocity = 100;
93     gPTPCoordinateParams.rVelocity = 100;
94     gPTPCoordinateParams.xyzAcceleration = 80;
95     gPTPCoordinateParams.rAcceleration = 80;
96
97
98     gPTPCommonParams.velocityRatio = 50;
99     gPTPCommonParams.accelerationRatio = 50;
100
101     gPTPCmd.ptpMode = MOVJ_XYZ;
102
103     //Home position
104     gPTPCmd.x = 0;
105     gPTPCmd.y = 260;
106     gPTPCmd.z = 155;
107     gPTPCmd.r = 90;
108
109     gQueuedCmdIndex = 0;

```

```

110
111 //IO shield outputs OFF
112 mcp.digitalWrite(0, LOW); //A0 OFF =====
113 mcp.digitalWrite(1, LOW); //A1 OFF =====
114
115
116 }
117
118 /*****
119 ** Function name:      setup
120 ** Descriptions:      Initializes Serial
121 *****/
122
123 void setup()
124 {
125     // set parameters of dobot magician
126     Serial.begin(115200);
127     Serial1.begin(115200);
128     printf_begin();
129
130     //Set Timer Interrupt
131     FlexiTimer2::set(100,Serialread);
132     FlexiTimer2::start();
133
134     // set parameters of IO shield
135     mcp.begin_I2C(0x21); // use default address 0x21
136     ads.setGain(GAIN_ONE);
137     ads.begin();
138     mcp.pinMode(0, OUTPUT); //A0 IO shield
139     mcp.pinMode(1, OUTPUT); //A1 IO shield
140     mcp.pinMode(8, INPUT); //B0 IO shield
141     mcp.pinMode(9, INPUT); //B1 IO shield
142
143     InitRAM();
144     ProtocolInit(); // initialize serial protocol
145     ProtocolProcess(); // read and execute movement instructions.
146
147
148 }
149
150 /*****
151 ** Function name:      loop
152 ** Descriptions:      Program entry
153 *****/
154 void loop()
155 {
156
157     //TRAJECTORY 1
158     while(mcp.digitalRead(9) == 0) // starts with B1 ON
159     {
160
161         //HOME_T1
162         ProtocolInit();
163         gPTPCmd.x = 0;
164         gPTPCmd.y = 260;

```

```
165     gPTPCmd.z = 155;
166     gPTPCmd.r = 90;
167     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
168     ProtocolProcess();
169     delay(1000);
170
171     //POSITIONING1_T1
172     ProtocolInit();
173     gPTPCmd.x = 27.5;
174     gPTPCmd.y = 178;
175     gPTPCmd.z = 70;
176     gPTPCmd.r = 180;
177     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
178     ProtocolProcess();
179     delay(1000);
180
181     //GRIP_T1
182     ProtocolInit();
183     gPTPCmd.x = 27.5;
184     gPTPCmd.y = 178;
185     gPTPCmd.z = 37;
186     gPTPCmd.r = 180;
187     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
188     SetEndEffectorGripper(true, true, &gQueuedCmdIndex); // Grip
189     ProtocolProcess();
190     delay(3000);
191
192     //POSITIONING2_T1
193     ProtocolInit();
194     gPTPCmd.x = 20;
195     gPTPCmd.y = 178;
196     gPTPCmd.z = 37;
197     gPTPCmd.r = 180;
198     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
199     ProtocolProcess();
200     delay(1000);
201
202     //POSITIONING3_T1
203     ProtocolInit();
204     gPTPCmd.x = 20;
205     gPTPCmd.y = 178;
206     gPTPCmd.z = 68;
207     gPTPCmd.r = 180;
208     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
209     ProtocolProcess();
210     delay(1000);
211
212     //BACKHOME_T1
213     ProtocolInit();
214     gPTPCmd.x = 0;
215     gPTPCmd.y = 260;
216     gPTPCmd.z = 155;
217     gPTPCmd.r = 90;
218     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
219     ProtocolProcess();
```

```
220     delay(1000);
221
222     //POSITIONING4_T1
223     ProtocolInit();
224     gPTPCmd.x = 215;
225     gPTPCmd.y = 86;
226     gPTPCmd.z = 169;
227     gPTPCmd.r = 92;
228     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
229     ProtocolProcess();
230     delay(1000);
231
232     //UNGRIP_T1
233     ProtocolInit();
234     gPTPCmd.x = 215;
235     gPTPCmd.y = 86;
236     gPTPCmd.z = 144;
237     gPTPCmd.r = 92;
238     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
239     SetEndEffectorGripper(false, true, &gQueuedCmdIndex); // Ungrip
240     ProtocolProcess();
241     delay(3000);
242
243     //UPMOVE_T1
244     ProtocolInit();
245     gPTPCmd.x = 215;
246     gPTPCmd.y = 86;
247     gPTPCmd.z = 169;
248     gPTPCmd.r = 92;
249     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
250     ProtocolProcess();
251     delay(1000);
252
253     //HOME_T2
254     ProtocolInit();
255     gPTPCmd.x = 152;
256     gPTPCmd.y = -142;
257     gPTPCmd.z = 171;
258     gPTPCmd.r = 2;
259     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
260     ProtocolProcess();
261
262     mcp.digitalWrite(1, HIGH); // A1 IO shield ON
263     delay(5000);
264     mcp.digitalWrite(1, LOW); // A1 IO shield OFF
265 }
266
267 //TRAJECTORY 2
268 while(mcp.digitalRead(10) == 0) // starts with B2 ON
269 {
270
271     //HOME_T2
272     ProtocolInit();
273     gPTPCmd.x = 152;
274     gPTPCmd.y = -142;
```

```
275     gPTPCmd.z = 171;
276     gPTPCmd.r = 2;
277     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
278     ProtocolProcess();
279     delay(1000);
280
281     //GRIP_T2
282     ProtocolInit();
283     gPTPCmd.x = 152;
284     gPTPCmd.y = -142;
285     gPTPCmd.z = 73;
286     gPTPCmd.r = 2;
287     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
288     SetEndEffectorGripper(true, true, &gQueuedCmdIndex); // Grip
289     ProtocolProcess();
290     delay(2000);
291
292     //BACKHOME_T2
293     ProtocolInit();
294     gPTPCmd.x = 152;
295     gPTPCmd.y = -142;
296     gPTPCmd.z = 171;
297     gPTPCmd.r = 2;
298     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
299     delay(1000);
300     ProtocolProcess();
301
302     //POSITIONING1_T2
303     ProtocolInit();
304     gPTPCmd.x = -41;
305     gPTPCmd.y = -276;
306     gPTPCmd.z = 140;
307     gPTPCmd.r = -1.5;
308     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
309     ProtocolProcess();
310     delay(1000);
311
312
313     //UNGRIP_T2
314     ProtocolInit();
315     gPTPCmd.x = -41;
316     gPTPCmd.y = -276;
317     gPTPCmd.z = 68;
318     gPTPCmd.r = -1.5;
319     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
320     SetEndEffectorGripper(false, true, &gQueuedCmdIndex); //Ungrip
321     ProtocolProcess();
322     mcp.digitalWrite(2, HIGH); // A2 IO shield ON
323     delay(3000);
324     mcp.digitalWrite(2, LOW); // A2 IO shield OFF
325
326     //UPMOVE_T2
327     ProtocolInit();
328     gPTPCmd.x = -41;
329     gPTPCmd.y = -276;
```

```
330     gPTPCmd.z = 140;
331     gPTPCmd.r = -1.5;
332     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
333     ProtocolProcess();
334     delay(1000);
335
336     //BACKHOME_T2
337     ProtocolInit();
338     gPTPCmd.x = 152;
339     gPTPCmd.y = -150;
340     gPTPCmd.z = 171;
341     gPTPCmd.r = 2;
342     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
343     ProtocolProcess();
344     delay(1000);
345
346     //BACKHOME_T1
347     ProtocolInit();
348     gPTPCmd.x = 0;
349     gPTPCmd.y = 260;
350     gPTPCmd.z = 155;
351     gPTPCmd.r = 90;
352     SetPTPCmd(&gPTPCmd, true, &gQueuedCmdIndex);
353     ProtocolProcess();
354     delay(6000);
355
356 }
357
358 }
```

Código 4.1 Rutina de control de movimientos para el robot

Como se puede observar y en resumen, con el comando *ProtocolInit* se inicializa el protocolo de comunicaron serie. Con el comando *gPTPCmd* se definen las coordenadas del efector final, estas coordenadas están dadas en milímetros. Con el comando *SetPTPCmd* se establecen las coordenadas en el vector de posición. Y finalmente, con el comando *ProtocolProcess* se leen y envían las instrucciones de movimiento por el puerto serie.

Una vez dominado el control de movimiento del robot, es crucial destacar el procedimiento para adquirir las coordenadas de los puntos específicos. Esta tarea se simplifica enormemente con la asistencia de DobotStudio. Al conectar el robot a esta interfaz, basta con posicionar el efector final en la ubicación requerida y observar las coordenadas que se despliegan en el panel de operaciones como se muestra en la Figura 4.3. Dichas coordenadas se presentan tanto en el sistema cartesiano como en el de articulación, proporcionando así una comprensión completa de la posición del robot.



Figura 4.3 Panel de operación en DobotStudio

Por otra parte, hay que destacar la importancia de tener una buena organización y asignación de las IP dentro de la red, ya que tanto el PLC, como la periferia, y hasta la computadora del diseñador requieren de una, y de no estar correctamente asignadas no habrá comunicación entre dichos dispositivos. Para el caso de la computadora del diseñador se asignó la IP 192.168.1.67 de manera manual desde la terminal de comandos de Windows como se muestra en Figura 4.4.

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.4291]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>netsh interface ip set address name="Ethernet" source=static addr=192.168.1.67 mask="255.255.255.0"
store=persistent

C:\Windows\system32>
```

Figura 4.4 Configuración manual de IP en una computadora

Por otra parte, la creación de un proyecto en TIA Portal planteó un desafío, especialmente en las primeras ocasiones. Si bien dar de alta el controlador es un proceso simple que se realiza apenas al crear el proyecto, la configuración de la periferia descentralizada resulta más compleja. Esto se debe a que cada módulo que conforma la periferia debe ser seleccionado y configurado adecuadamente, pues incluso el uso de una versión incorrecta de firmware puede ocasionar serios problemas de comunicación, lo que dificultará trabajar con la periferia de manera eficiente. Es esencial prestar atención a estos detalles para garantizar un correcto funcionamiento del sistema de control.

El primer paso para configurar correctamente un proyecto es ir a la vista de proyecto y agregar un nuevo dispositivo, donde se selecciona el CPU no especificado 1500 (6ES7 5XX). En el recuadro amarillo emergente se selecciona la opción de detectar y se comienza la búsqueda del PLC. Una vez detectado el hardware, se procede a establecer la conexión con el dispositivo como se muestra en la Figura 4.5.

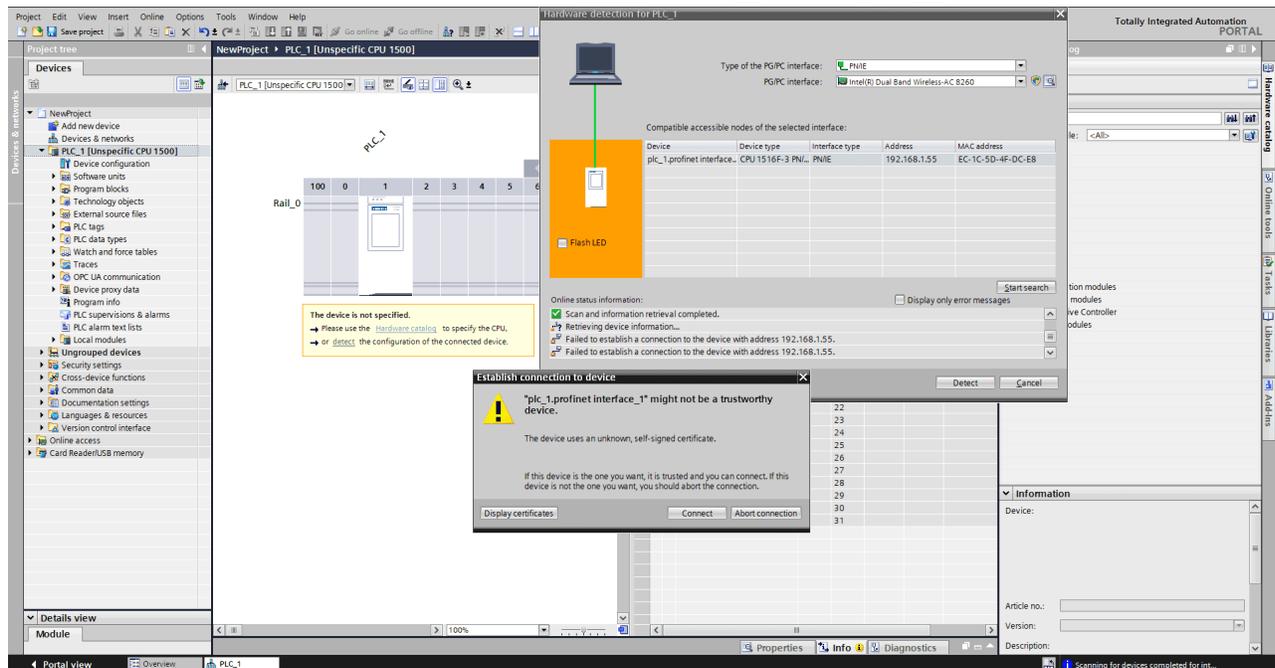


Figura 4.5 Selección, identificación y conexión con el PLC

Hecho lo anterior, se procede a establecer las configuraciones de seguridad del controlador. Primero se debe desactivar la opción *Protects the PLC configuration data from TIA Portal*, como se muestra en la Figura 4.6

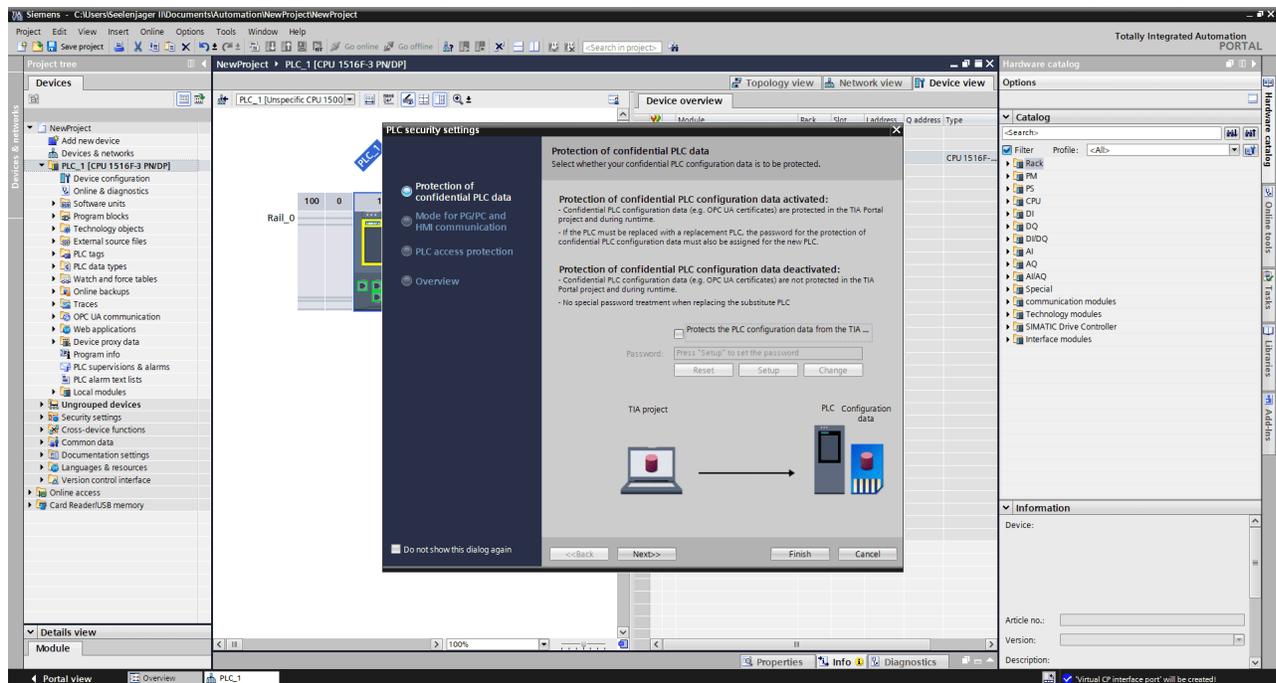


Figura 4.6 Configuración de protección

Después, se cambia el nivel de acceso a *full access inc. fail-safe (no protection)* y se finaliza la configuración inicial, como se observa en la Figura 4.7

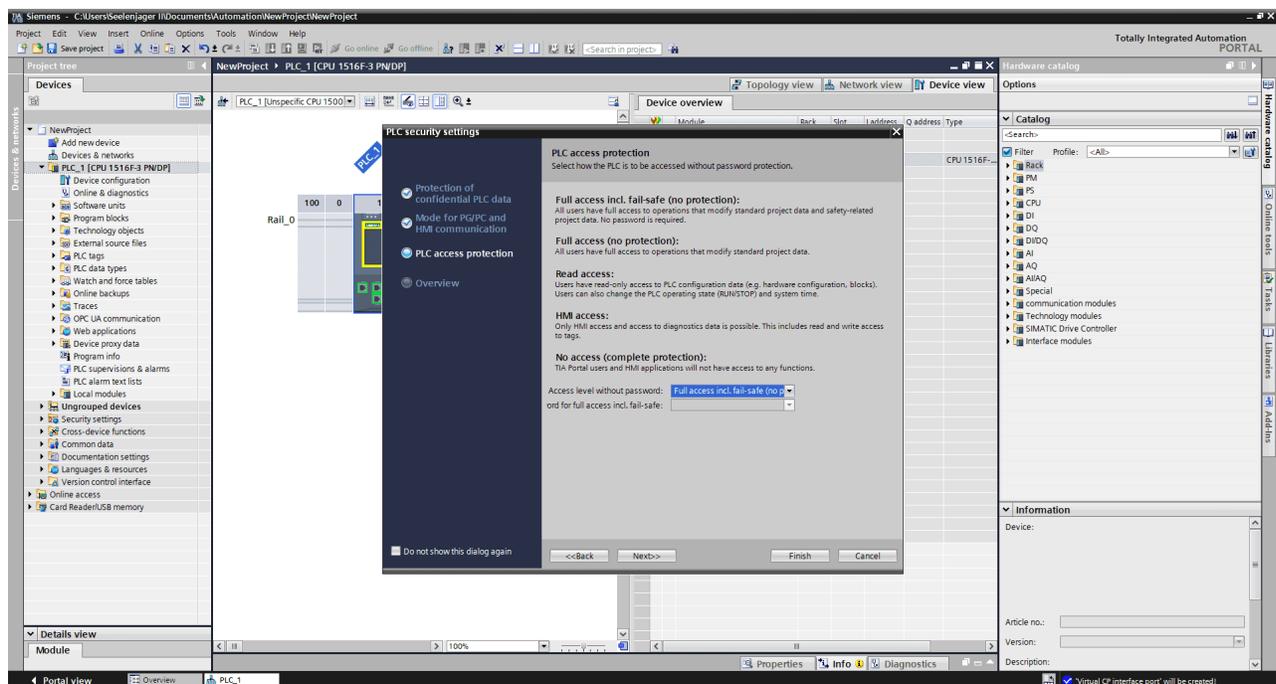


Figura 4.7 Configuración de acceso

Posteriormente, se procede a configurar a la periferia. En la *Network view* se accede al catalogo de hardware y se busca el módulo 6ES7 155-6AU00-0CN0 con versión de firmware 3.3, como se muestra en la Figura 4.8

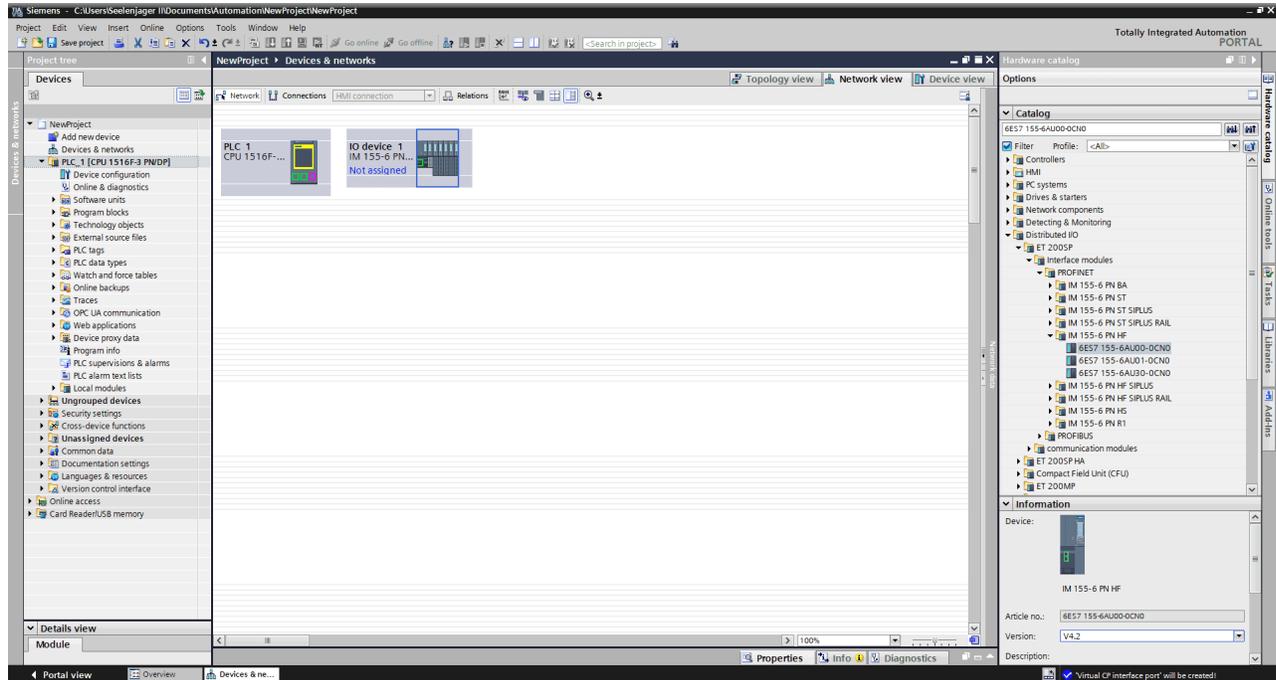


Figura 4.8 Configuración de la periferia

Es de suma importancia asignar una IP sobre la misma red a este dispositivo. Por ello, en propiedades del dispositivo se le asigna la IP 192.168.1.64, como se observa en la Figura 4.9

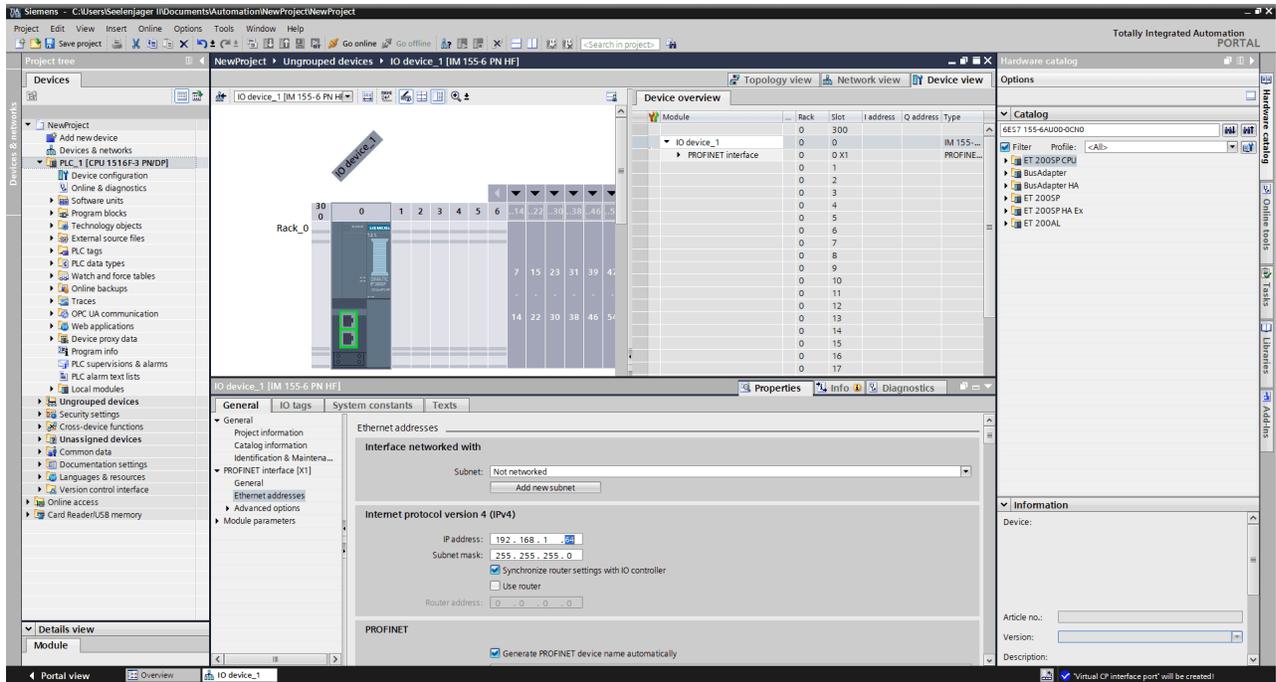


Figura 4.9 Configuración de IP de la periferia

Después, en el catalogo de hardware se busca y agregan 2 módulos de entradas digitales 6ES7 131-6BF00-0CA0, como se muestra en la Figura 4.10

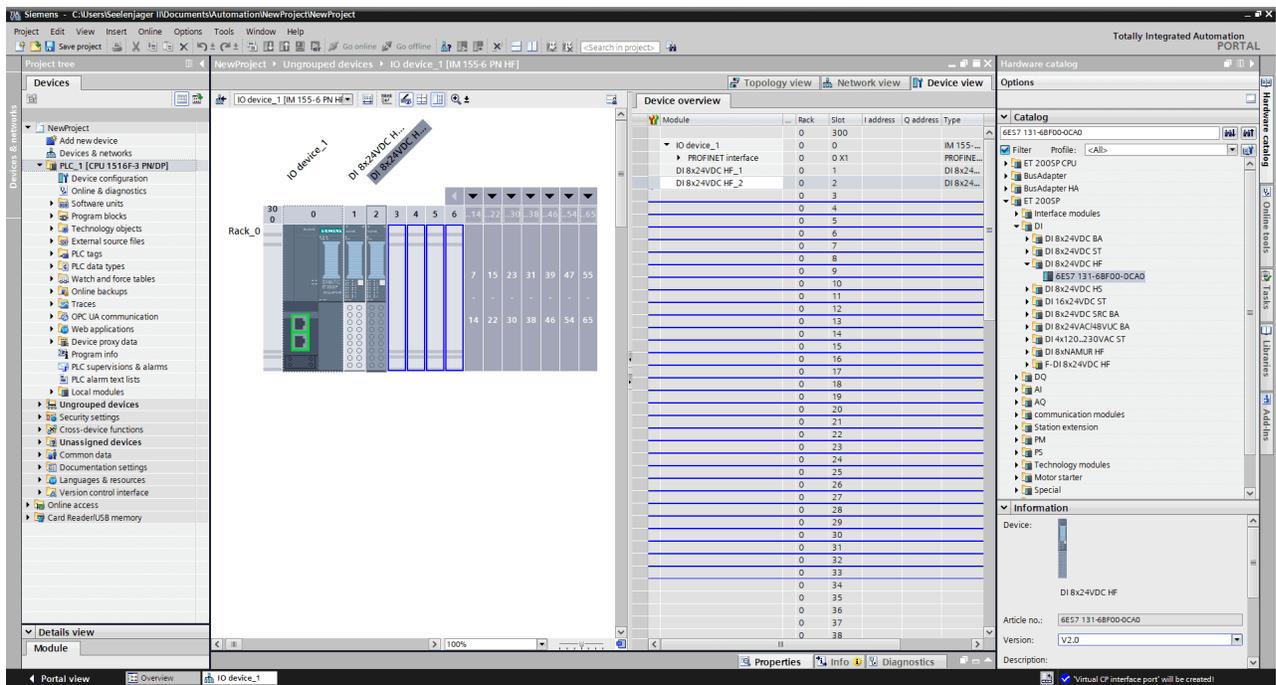


Figura 4.10 Configuración de los módulos de entradas digitales

También se busca y agrega un módulo de salidas digitales 6ES7 132-6BH00-0BA0, al cual se le activa el grupo potencial desde sus propiedades, como se visualiza en la Figura 4.11

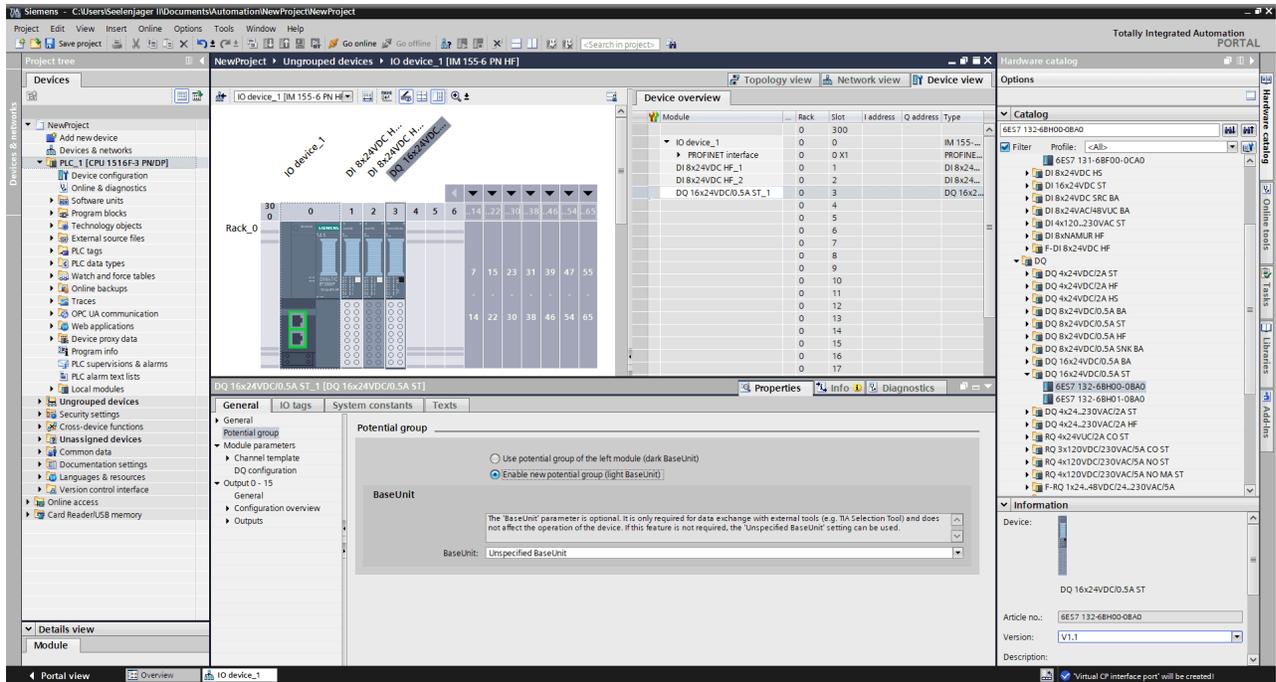


Figura 4.11 Configuración del módulo de salidas digitales

También se busca y agrega un módulo de salidas analógicas 6ES7 135-6HD00-0BA1, como se muestra en la Figura 4.12

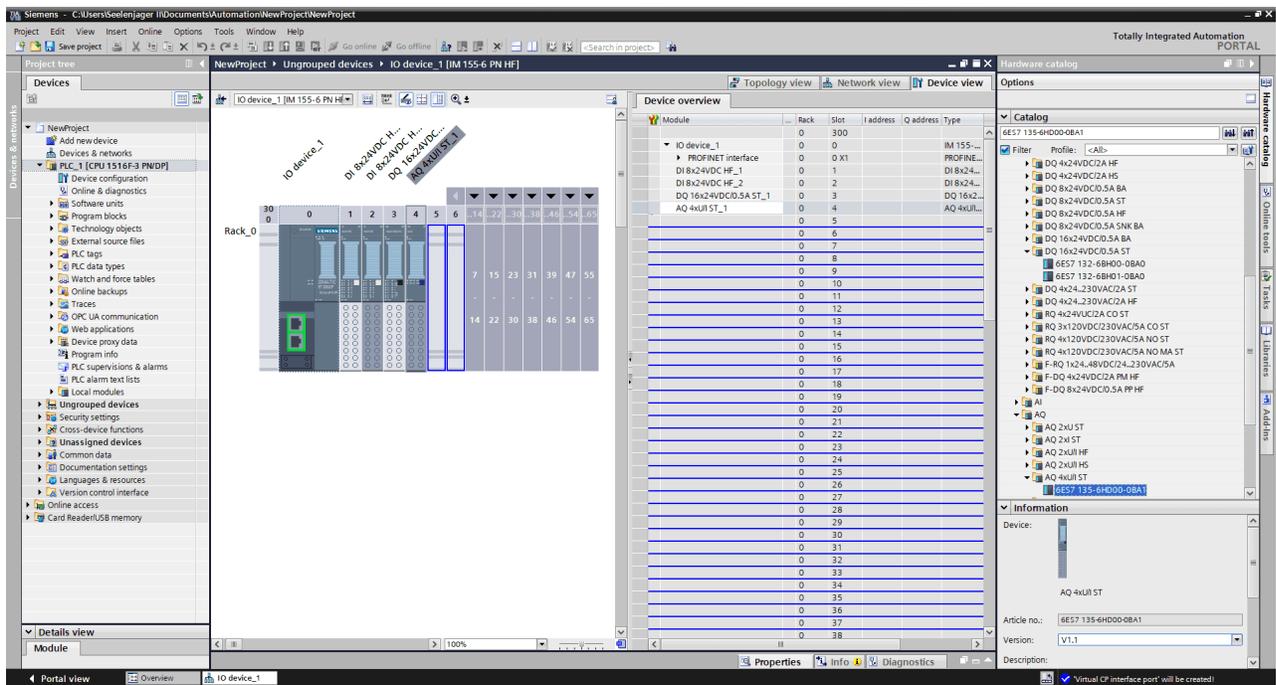


Figura 4.12 Configuración del módulo de salidas analógicas

También se busca y agrega un módulo medidor de energía 6ES7 134-6PA20-0BD0, como se observa en la Figura 4.13

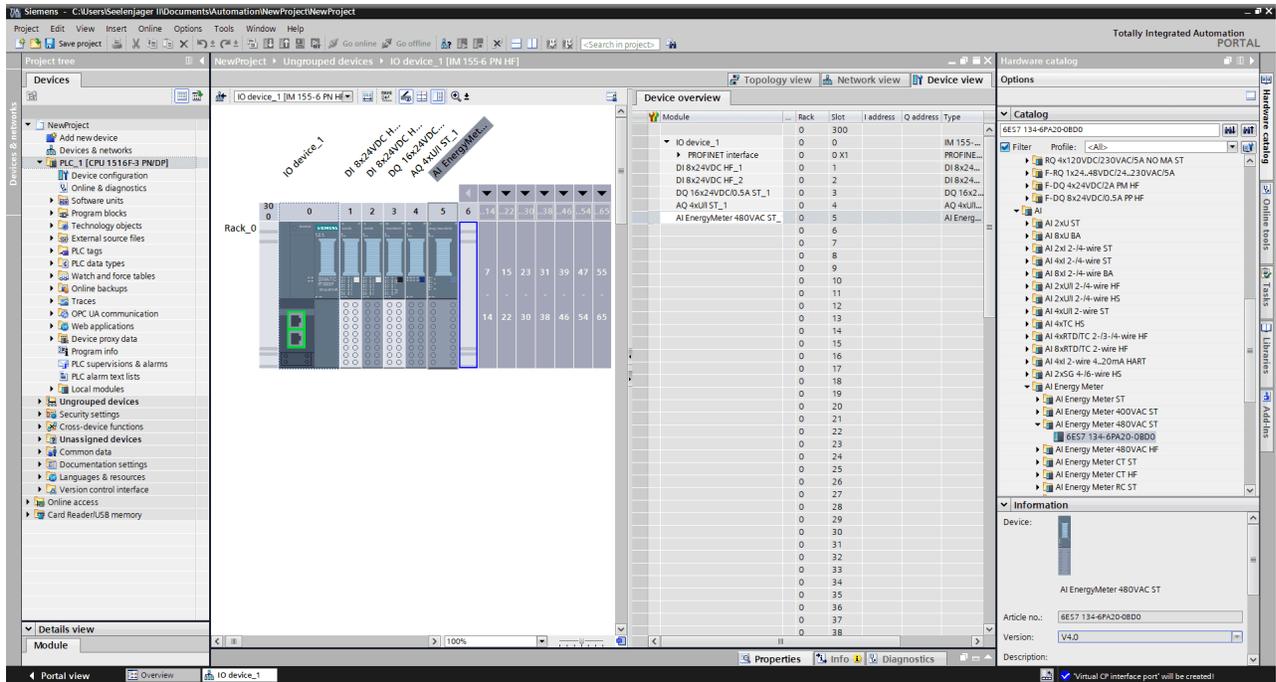


Figura 4.13 Configuración del módulo medidor de energía

Por último, se busca y agrega un módulo de entradas analógicas 6ES7134-6HB00-0CA1, al cual también se le activa el grupo potencial desde sus propiedades, como se aprecia en la Figura 4.14

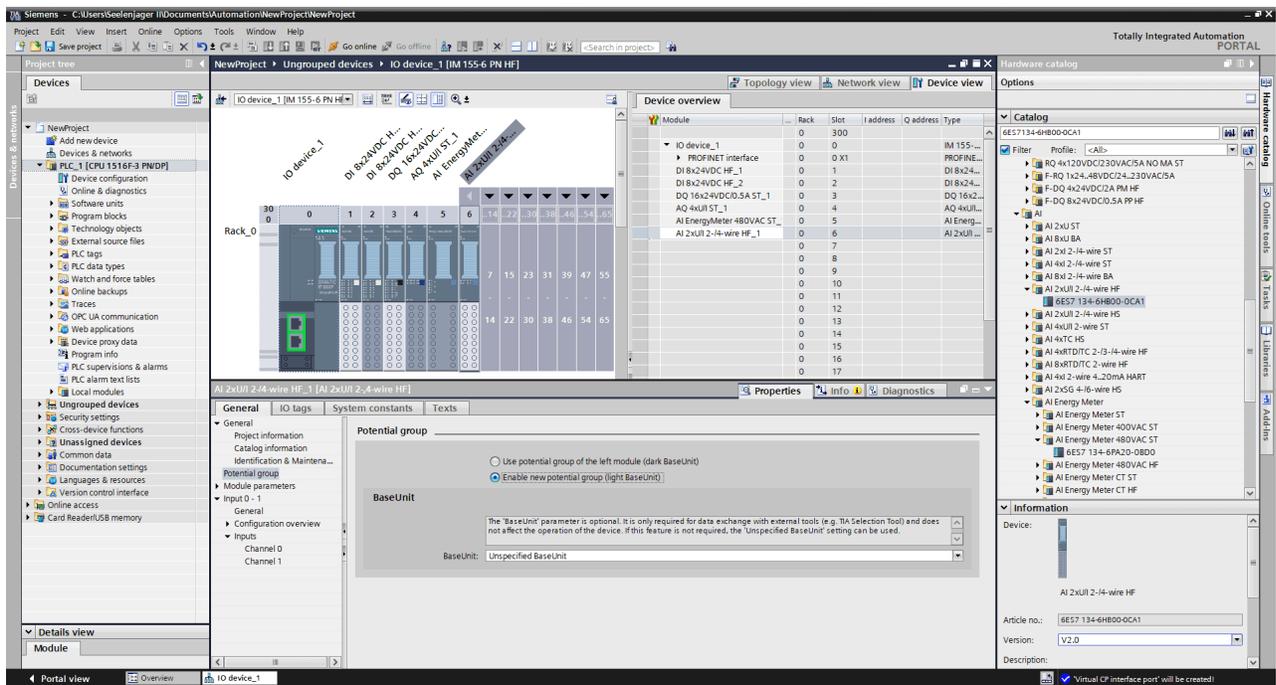


Figura 4.14 Configuración del módulo de entradas analógicas

Posteriormente, para comunicar al controlador con la periferia, se accede a la *Network view* y se cambia la conexión *Not assigned* por *PLC_1.PROFINET interface_1*, como se muestra en la Figura 4.15

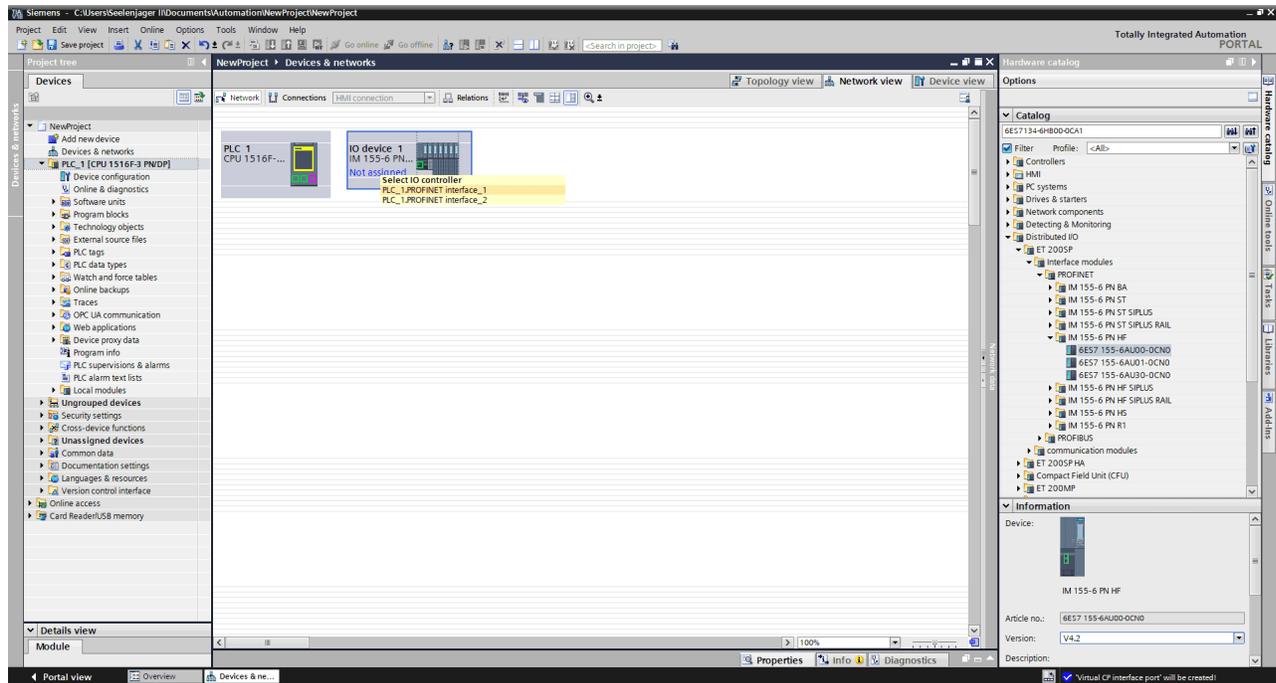


Figura 4.15 Comunicación del PLC y la periferia

Después, se compila el hardware y software de la periferia, ambos como (*rebuild all*), como se observa en la Figura 4.16

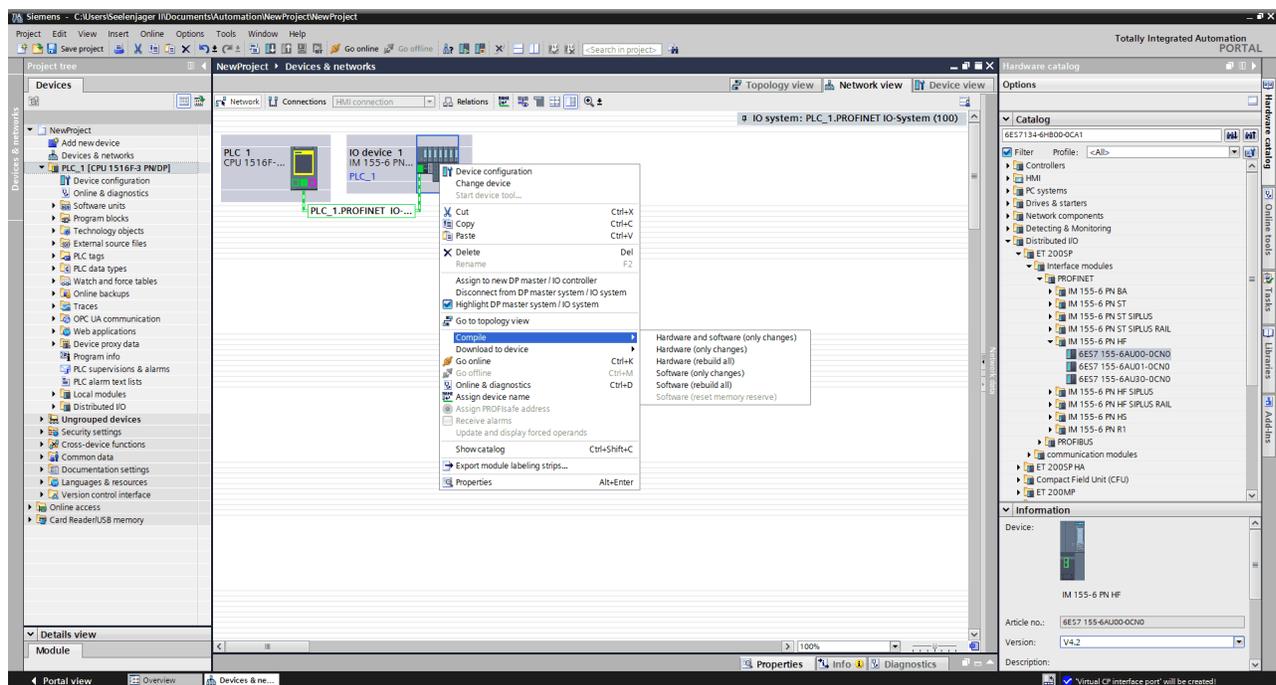


Figura 4.16 Compilación del hardware y software de la periferia

Nuevamente se repite el proceso de compilación anterior, pero ahora con el PLC, como se aprecia en la Figura 4.17

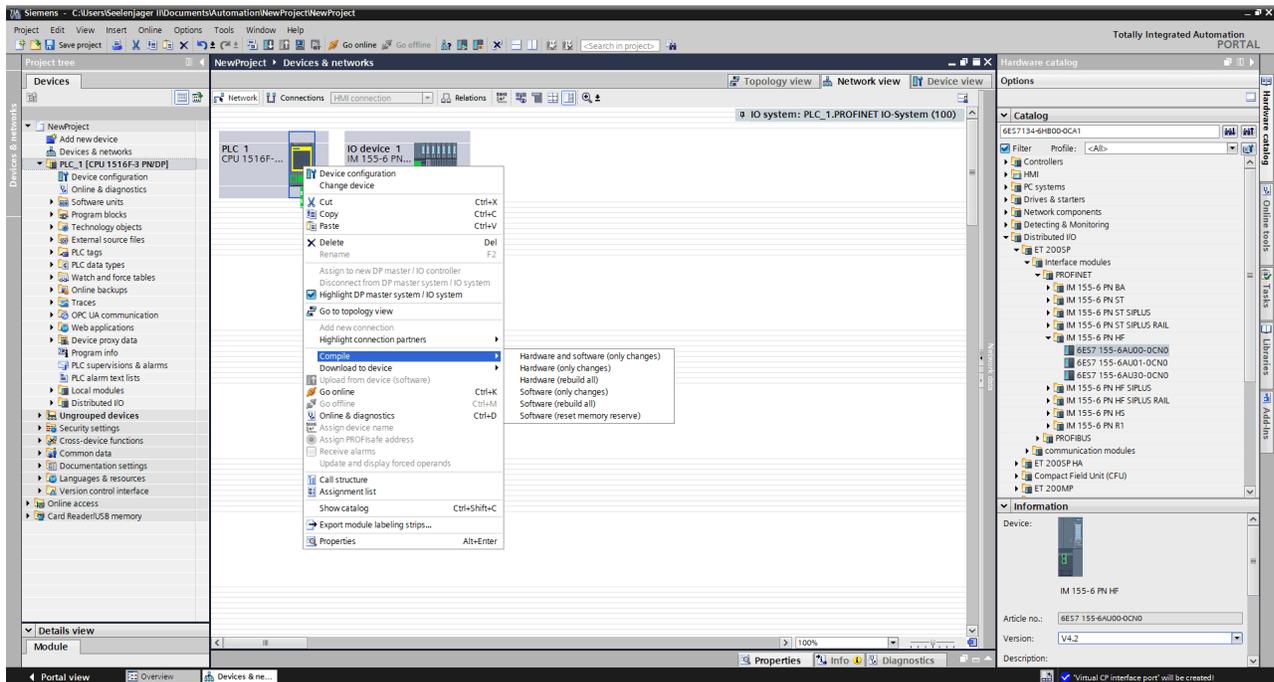


Figura 4.17 Compilación del hardware y software del PLC

Posteriormente, se realiza la descarga de la configuración de hardware de la periferia, como se visualiza en la Figura 4.18

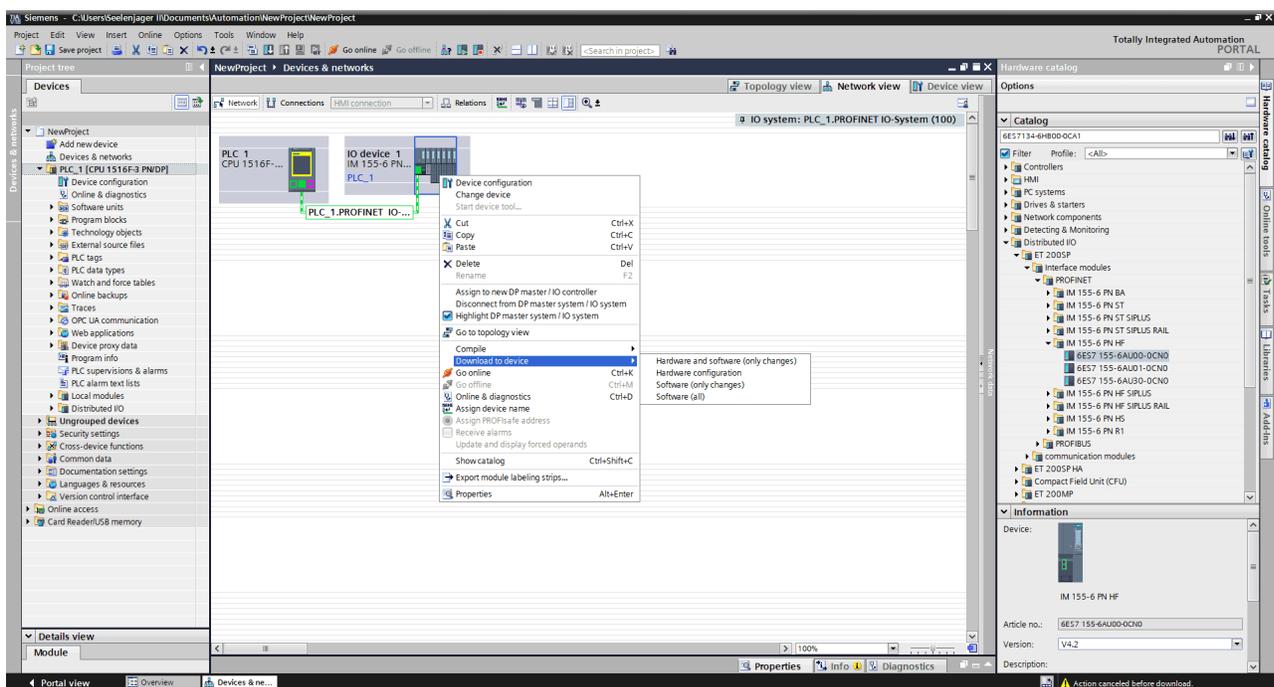


Figura 4.18 Descarga de la configuración de hardware de la periferia

Con lo anterior se desplegará una ventana donde se buscará y seleccionará al PLC para realizar la descarga, como se muestra en la Figura 4.19

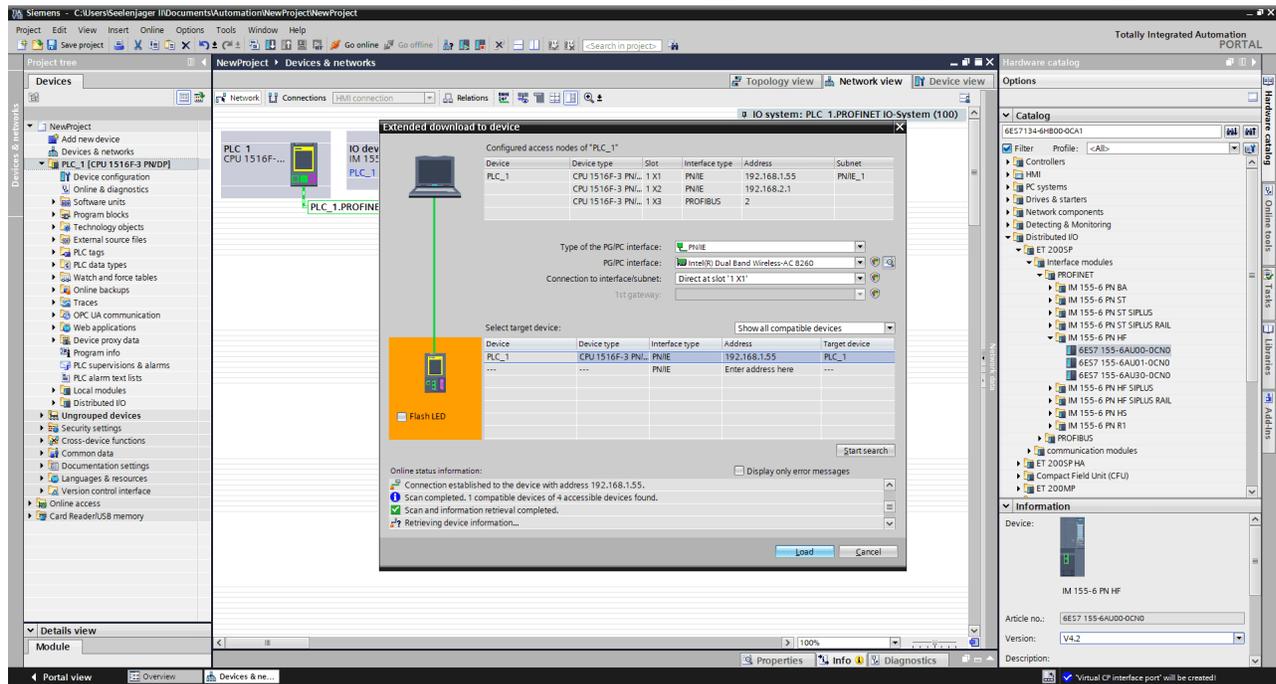


Figura 4.19 Descarga al PLC

Después, se hace la descarga de la configuración de hardware del controlador, como se observa en la Figura 4.20

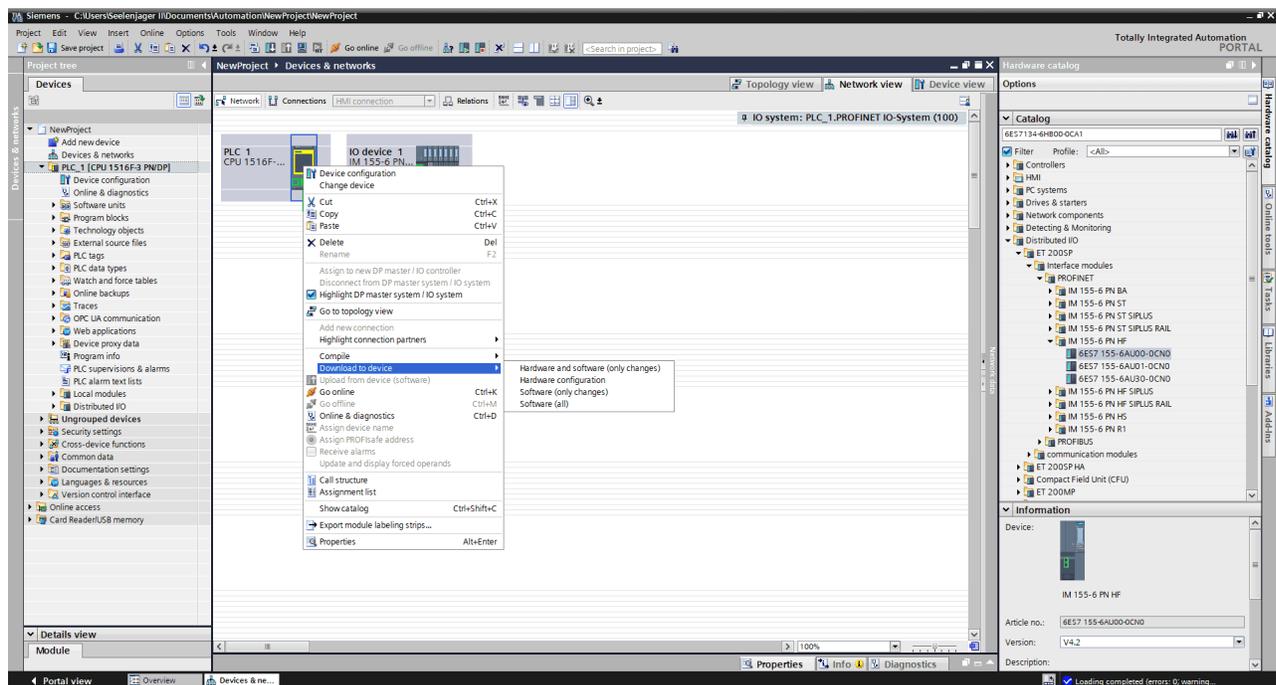


Figura 4.20 Descarga de la configuración de hardware del PLC

Posteriormente, se lleva al PLC a línea, como se aprecia en la Figura 4.21

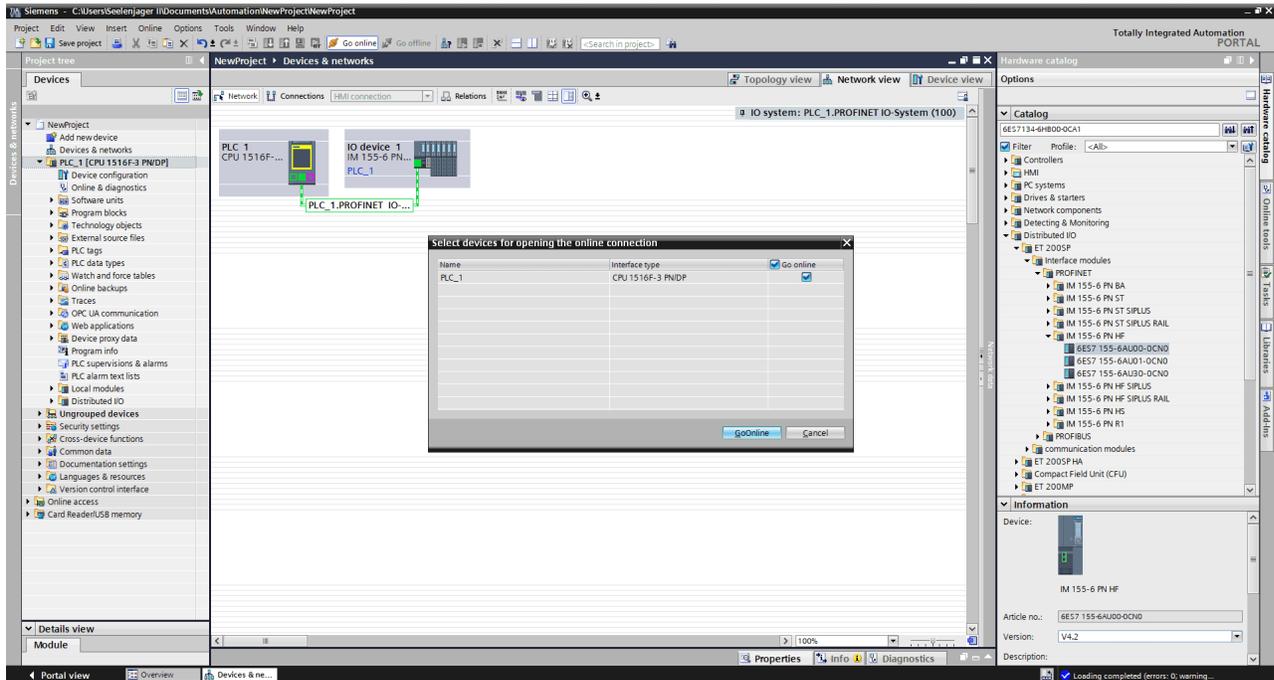


Figura 4.21 PLC en línea

Una vez con el controlador en línea, se realiza un *upload* confirmando la acción *Checks performed before upload from device*, como se muestra en la Figura 4.22

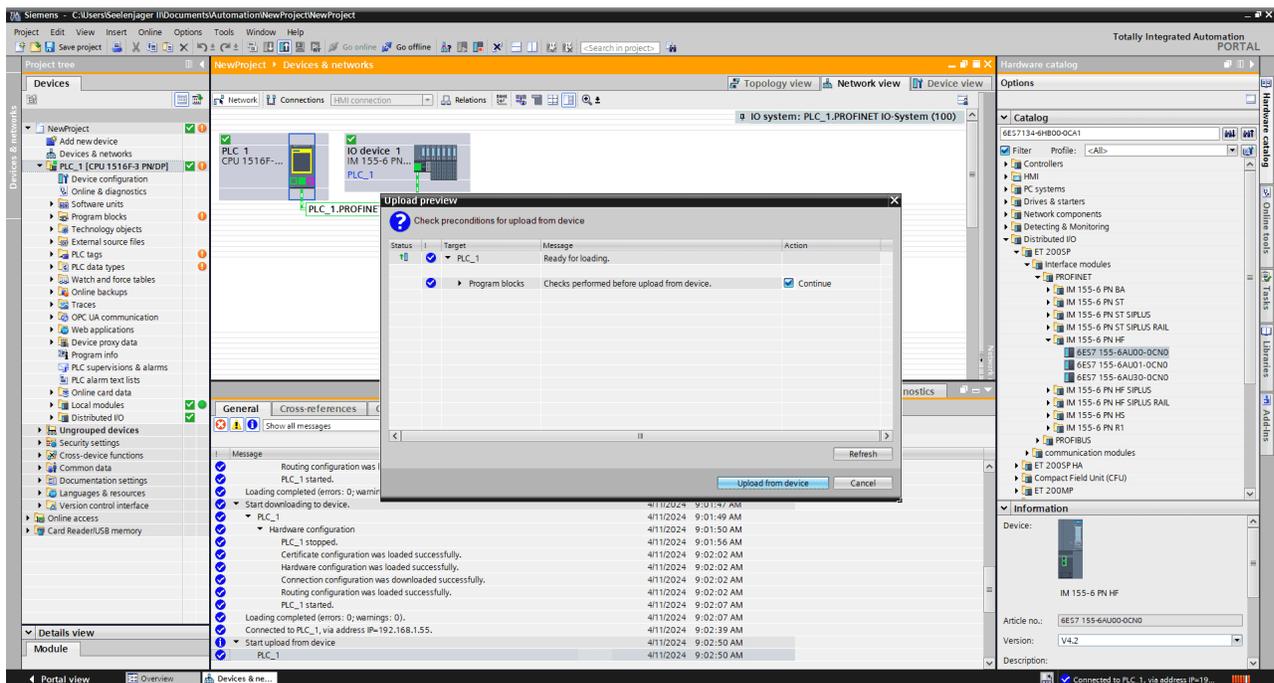


Figura 4.22 Upload en el PLC

De esta manera se consigue un proyecto totalmente configurado y listo para trabajar, como se observa en la Figura 4.23

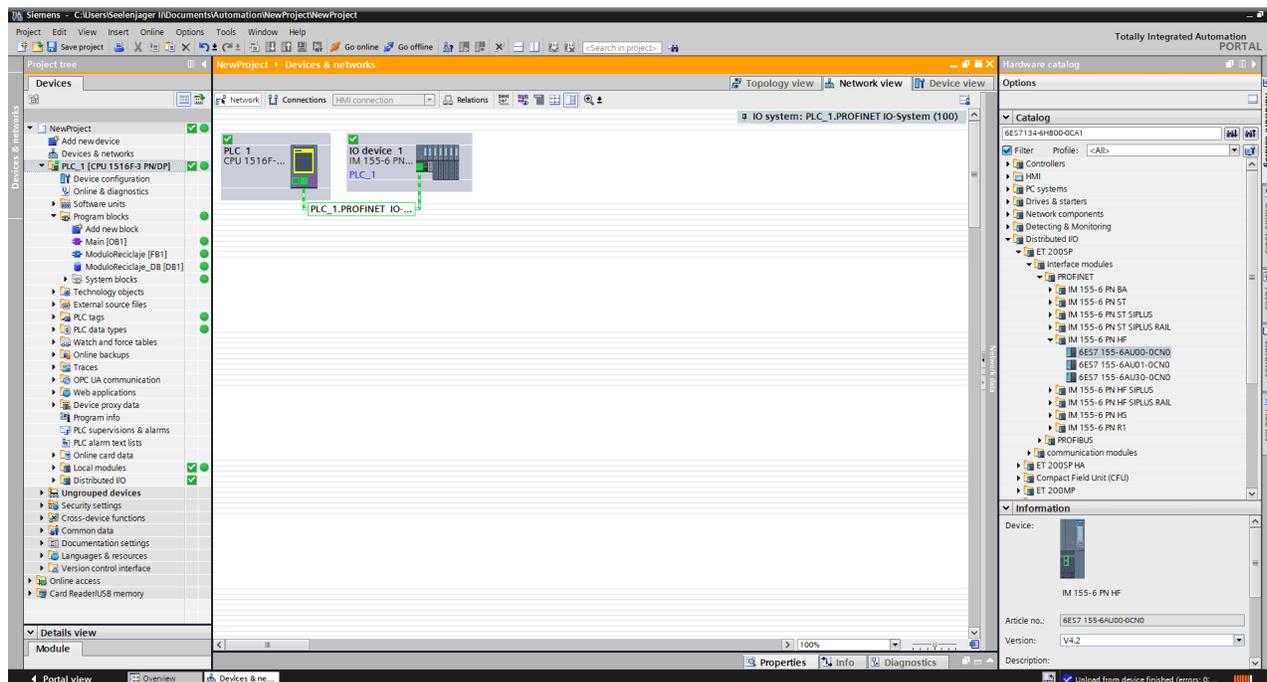


Figura 4.23 Configuración del proyecto terminada

Otra configuración pertinente en el proyecto es la de la interfaz de supervisión web. Se debe recordar que esta se implementa en un servidor web alojado en el sistema del controlador. Por esta razón, primero se debe habilitar la casilla de servidor web en la pestaña de propiedades del PLC, como se ilustra en la Figura 4.24.

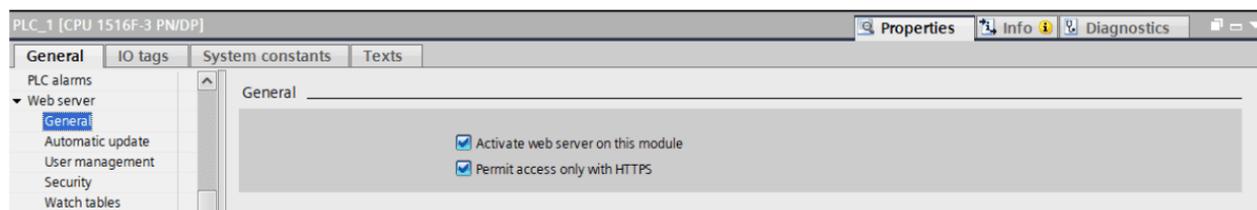


Figura 4.24 Activación del servidor web en el PLC

Posteriormente se accede a la gestión de usuarios y se deben crear los perfiles que podrán acceder a la interfaz. Hay que destacar que es importante definir que privilegios y accesos tendrá cada usuario, así como su debida contraseña, como se visualiza en la Figura 4.25.

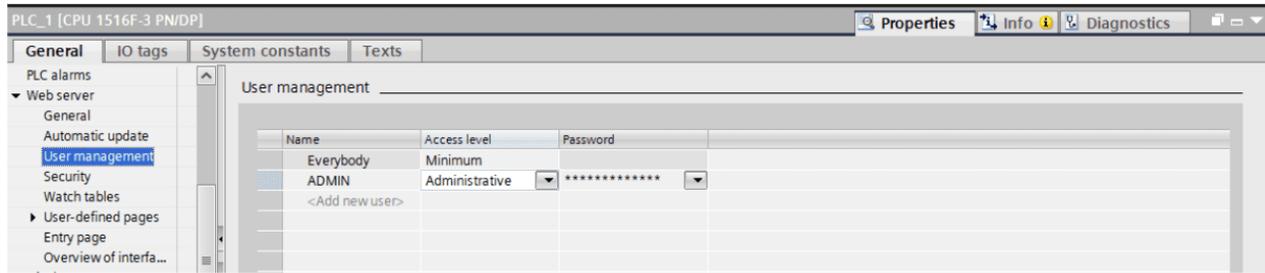


Figura 4.25 Creación de usuarios

Después, en la pestaña de protección y seguridad del controlador, se encuentra el administrador de certificados. Aquí se exportará el certificado web, como se muestra en la Figura 4.26.

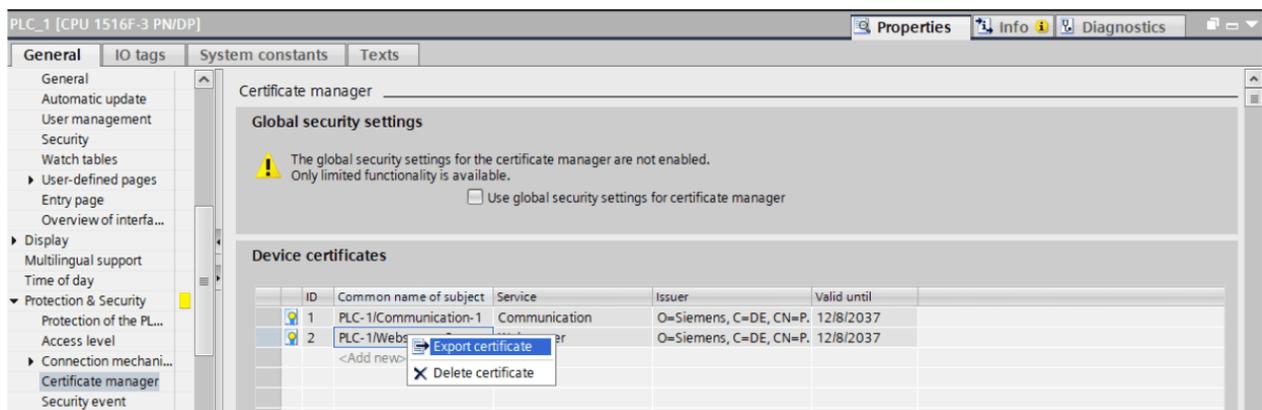
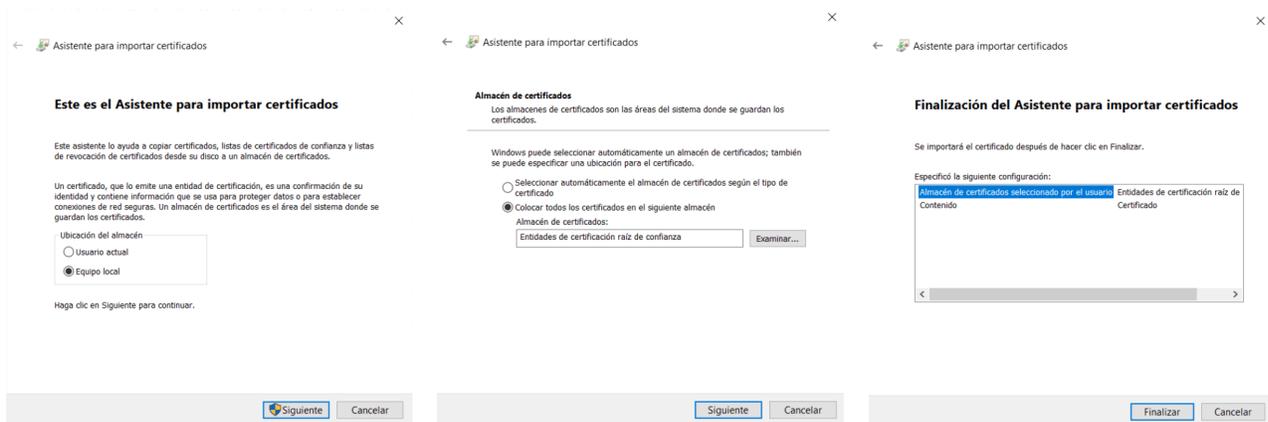


Figura 4.26 Exportación del certificado

Posteriormente, se instala el certificado de manera local. Figura 4.27a. Después en el almacén de certificados, se selecciona el certificado de *Entidades de certificación raíz de confianza*, como se observa en la Figura 4.27b, este certificado provee del protocolo HTTPS a la página web haciéndola segura y confiable. Y por ultimo, se finaliza la instalación del certificado, como se visualiza en la Figura 4.27c.



(a) Instalación de manera local

(b) Selección del certificado

(c) Finalizar instalación

Figura 4.27 Instalación del certificado web

Por otra parte, es importante destacar que gracias al sensor de corriente y la rutina de control de movimiento implementados, se lograron realizar un par de pruebas donde se comparó la corriente consumida por un motor con y sin el control de arranque suave. Como se puede apreciar en la Figura 4.28, se observa el comportamiento de la corriente medida en un motor cuando se activa el arranque directo y cuando se utiliza el arranque suave. En este último caso, se puede notar que desaparece el pico máximo de corriente cuando inicia el movimiento, y en su lugar se genera una rampa en el consumo de corriente.

Los resultados de las pruebas indican que el consumo promedio utilizando el arranque directo fue de 1.53 [A], mientras que al emplear la rutina de control para el arranque suave, el consumo promedio fue de 1.48 [A] durante los 12 segundos de prueba en cada caso. Esto representa una reducción en el consumo de corriente de 50 [mA]. Aunque a primera vista pueda parecer una mejora modesta, es importante tener en cuenta que estos resultados se aplican a motores pequeños. Sin embargo, al extrapolar este ahorro a motores de mayor potencia, como los utilizados en las líneas de producción industrial, el impacto es significativo. Por ejemplo, en un motor industrial de 10 veces más potencia, que implemente este arranque suave 1000 veces al día, se podría lograr una reducción de hasta 500 [A], lo que representa un ahorro energético considerable.

Por otra parte, es importante destacar que el comportamiento fluctuante de la señal de corriente a plena carga se debe a que los motores no están colocados sobre una superficie fija, sino que son flotantes, ya que están sostenidos por un lado de su carcasa acoplada a la banda transportadora, lo que genera perturbaciones e inestabilidad en el movimiento del eje, tal y como se observa en la Figura 3.2.

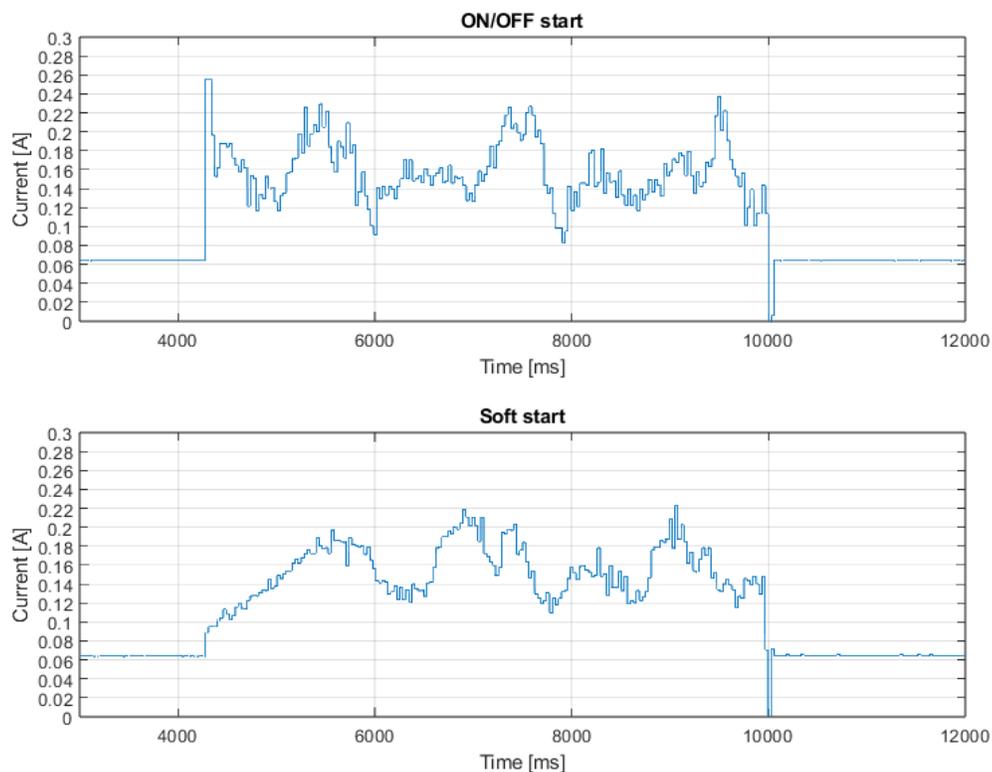


Figura 4.28 Comparación de corriente con arranque suave y directo

4.3. Nivel de supervisión

La etapa de implementación a nivel supervisión resultó ser fluida y sin contratiempos significativos. Esto permite afirmar que la interfaz web diseñada fue un éxito, como se muestra en la Figura 4.29. Ahora, es posible monitorear e incluso controlar el proceso desde una laptop o cualquier dispositivo móvil con un navegador web, tal como se puede apreciar en la Figura 4.30. Esto se logra gracias a la herramienta VoT, la cual agrega un gran valor al proceso, acercándolo un paso más hacia una Industria 4.0. Considerando su naturaleza descentralizada, esta herramienta ofrece múltiples ventajas al no estar vinculada a una arquitectura específica. En consecuencia, es posible acceder a la interfaz desde prácticamente cualquier dispositivo con un navegador web.

Es importante destacar que este tipo de interfaces desarrolladas con VoT son seguras. Al igual que con las interfaces tradicionales, se requiere de un usuario y contraseña para poder acceder, y cada usuario es configurado previamente con sus respectivos privilegios. Además, el servidor web cuenta con el Protocolo Seguro de Transferencia de Hipertexto (HTTPS, por sus siglas en inglés), basado en el uso de certificados de Capa de Puertos Seguros (SSL, por sus siglas en inglés) y Seguridad de la Capa de Transporte (TLS, por sus siglas en inglés), que autentican la identidad del sitio web y cifran la información transmitida entre el usuario y el servidor.

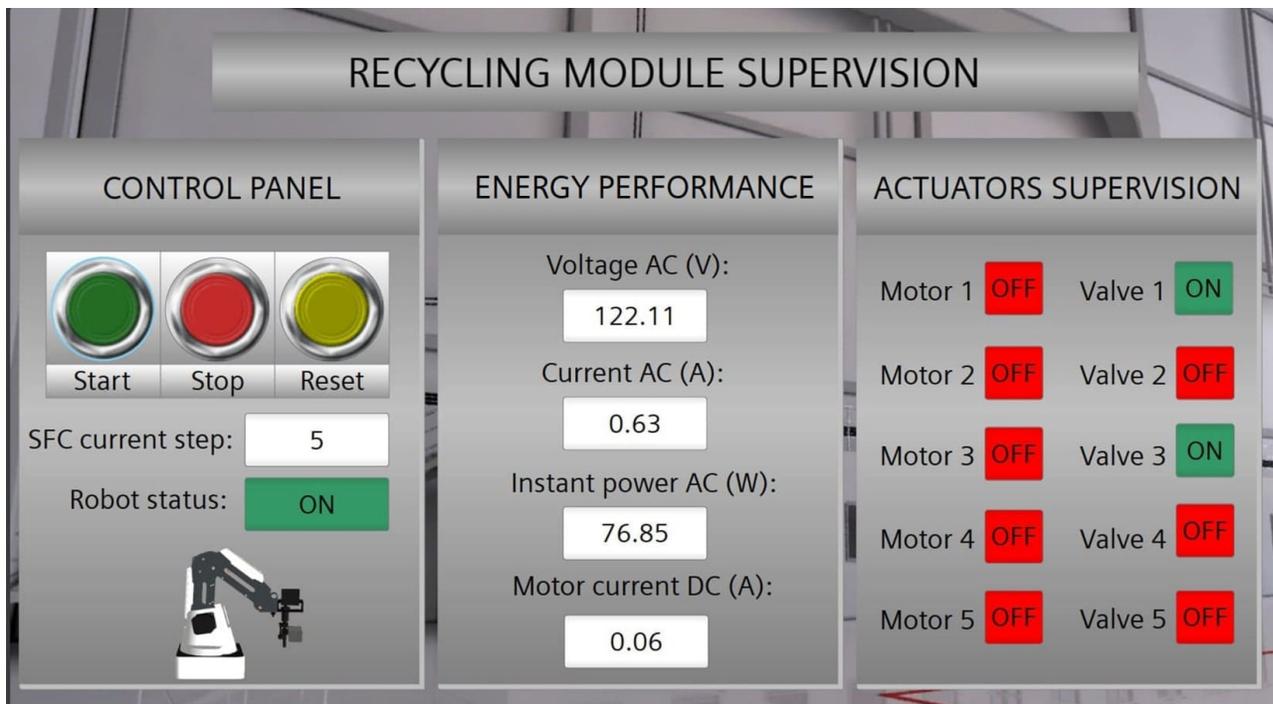


Figura 4.29 HMI web resultante



Figura 4.30 HMI web en dispositivos móviles

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo se presenta un sistema de control de naturaleza secuencial para una plataforma experimental de un proceso de embotellado. La resolución de la problemática planteada ha sido un proceso integral que ha permitido avanzar significativamente hacia la automatización y optimización del cuarto módulo del prototipo. Por un lado, se ha logrado un entendimiento detallado del funcionamiento y los componentes del módulo de reciclaje mediante un exhaustivo levantamiento técnico, que incluye la documentación de los dispositivos y conexiones eléctricas.

Por otra parte, la arquitectura de control diseñada e implementada ha demostrado ser confiable y versátil, facilitando la interconexión y programación del PLC y la periferia descentralizada, así como la integración de futuros dispositivos de automatización. Hablando de la estrategia de control desarrollada en lenguaje secuencial SFC y diagrama de escalera LD, garantizan un funcionamiento correcto y seguro del sistema, al establecer una secuencia de etapas y transiciones bien definidas de acuerdo a la metodología GRAFCET, así como congruencia y compatibilidad entre ellos al estar estandarizados por la IEC 61131-3. Además, esta arquitectura de control tiene la capacidad de evaluar otras estrategias, como la implementada en el control de movimiento para las bandas transportadoras.

Por otra parte, un aspecto importante a destacar de la interfaz de control y supervisión implementada en un servidor web es que proporciona una herramienta intuitiva, flexible, ágil y sobre todo descentralizada, para monitorear y gestionar al proceso en tiempo real. Esto permite iniciar, detener o reiniciar la rutina de control del proceso, así como visualizar y cuantificar variables de interés como los indicadores clave de desempeño, tal como los voltajes y corrientes consumidos, desde cualquier parte del piso de planta gracias a la visualización web. Además, es importante destacar la descentralización de esta interfaz de supervisión, ya que forma parte de la actual tendencia de adoptar arquitecturas abiertas que permitan desde la visualización hasta la gestión de información proveniente del proceso. Esta descentralización permitió implementar una HMI web que es accesible desde cualquier dispositivo móvil, independientemente de su marca o sistema operativo. Únicamente se necesita tener acceso a internet y un navegador web, todo ello sin incurrir en gastos adicionales de software y hardware, aprovechando únicamente las capacidades nativas del controlador. También, es relevante mencionar que esta herramienta contempla la digitalización, abriendo así la puerta hacia una industria 4.0 que busca crear entornos de producción más eficientes, flexibles y conectados.

Finalmente, el resultado de esta investigación ha culminado en la creación de una filosofía operativa que permite comprender clara y plenamente las funciones del proceso en su totalidad, incluyendo qué hace, dónde comienza y termina, así como las interrelaciones entre los módulos. Esta comprensión,

junto con la documentación generada, coloca al equipo de trabajo del laboratorio en una posición mucho más sólida para continuar automatizando al resto de módulos.

5.1. Trabajo a futuro

Una vez demostrada la correcta implementación y funcionamiento del sistema de control y supervisión, junto con la viabilidad de emplear esta plataforma como herramienta de evaluación para otras estrategias de control, resulta natural considerar la posibilidad de introducir nuevas mejoras que optimicen la eficiencia y seguridad del proceso.

En este sentido, se perciben numerosas posibilidades de mejora para este cuarto módulo mediante la integración de diversos dispositivos y herramientas. Por ejemplo, la incorporación de una cámara con ayuda de algoritmos de visión artificial permitiría la implementación de un sistema capaz de identificar con precisión las etapas del proceso, así como detectar factores de riesgo que pudieran interferir en el mismo, como la presencia de una mano dentro del área de trabajo. Ante tal eventualidad, el sistema podría detener el proceso como medida de seguridad y generar alarmas para alertar sobre la situación.

Otro aspecto a mejorar sería la pasarela de conexión, dado que el actual sistema, que depende únicamente de dos variables booleanas y el módulo de expansión industrial, para controlar los movimientos del robot, podría comprometer la robustez del sistema de control. En este sentido, la integración de un módulo de comunicación serie, como el 6ES7137-6AA01-0BA0, en la periferia descentralizada, permitiría enviar todos los comandos de movimiento directamente desde el PLC hacia el brazo robótico, sin la necesidad de un sistema intermedio como lo es la pasarela. Lo anterior podría permitir otras mejoras, como la capacidad de modificar los destinos de los movimientos directamente desde la HMI.

Otra forma de mejorar al módulo sería integrar un sensor fotoeléctrico en la pinza del brazo robótico. Esta adición aumentaría la fiabilidad del sistema de control, ya que el sensor sería capaz de detectar anomalías relacionadas en el transporte de las botellas con el robot. Por ejemplo, podría alertar si la pinza no logra sujetar las botellas o si éstas se caen mientras el robot las traslada de un punto a otro. El sensor avisaría al controlador, que junto a la rutina de control, tomaría las medidas necesarias para abordar la situación. En resumen, esta mejora garantizaría la seguridad del proceso frente a este tipo de imprevistos.

Finalmente, con todo este trabajo también se busca promover la generación de conocimiento dentro del Laboratorio de Automatización con la implementación de prácticas para las materias afines.

Apéndice A

Diagramas eléctricos

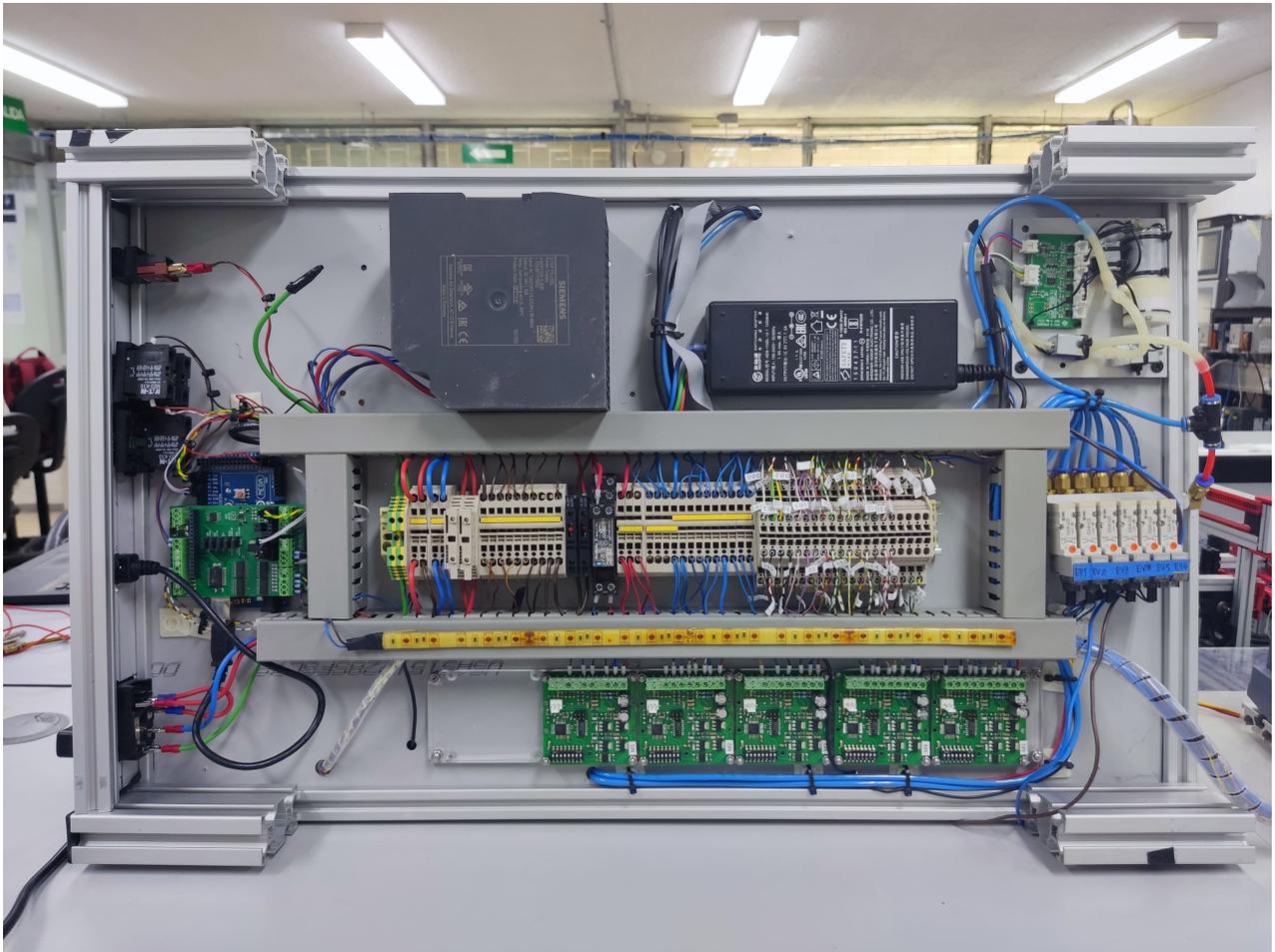


Figura A.1 Vista de conexiones inferiores de los dispositivos del módulo 4

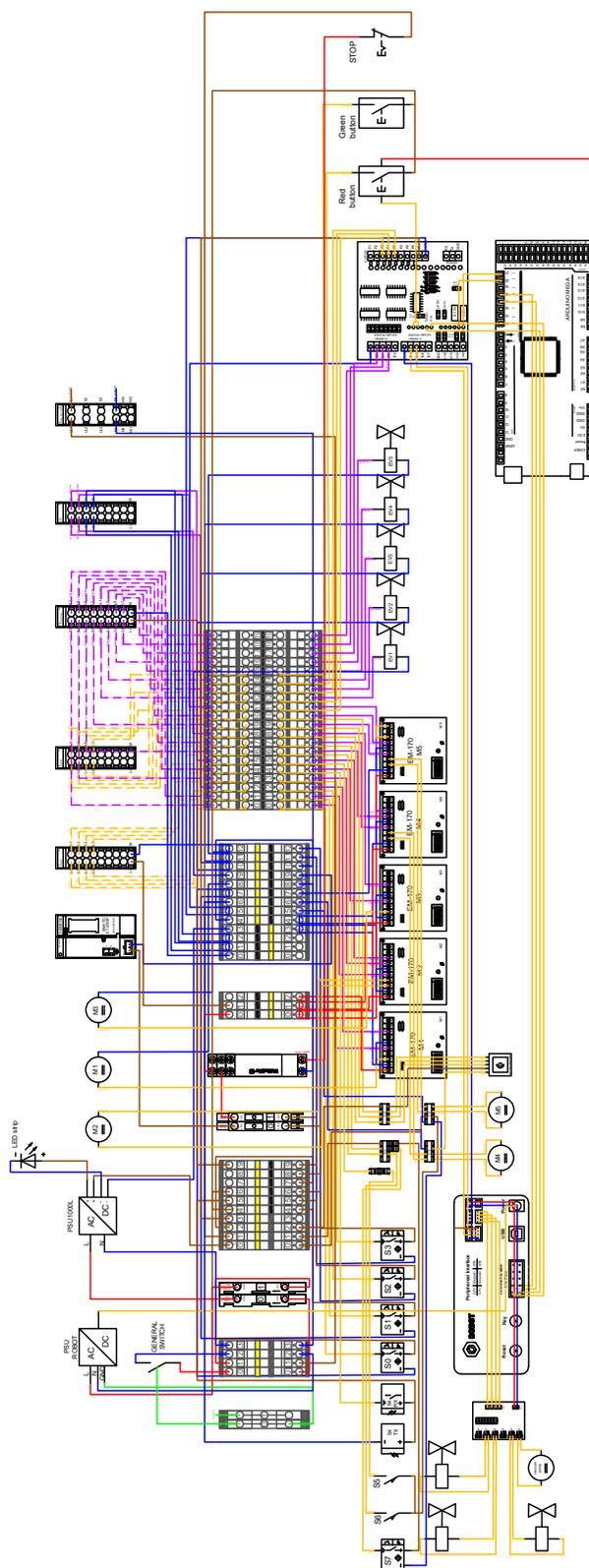


Figura A.2 Diagrama de conexiones generales del módulo 4

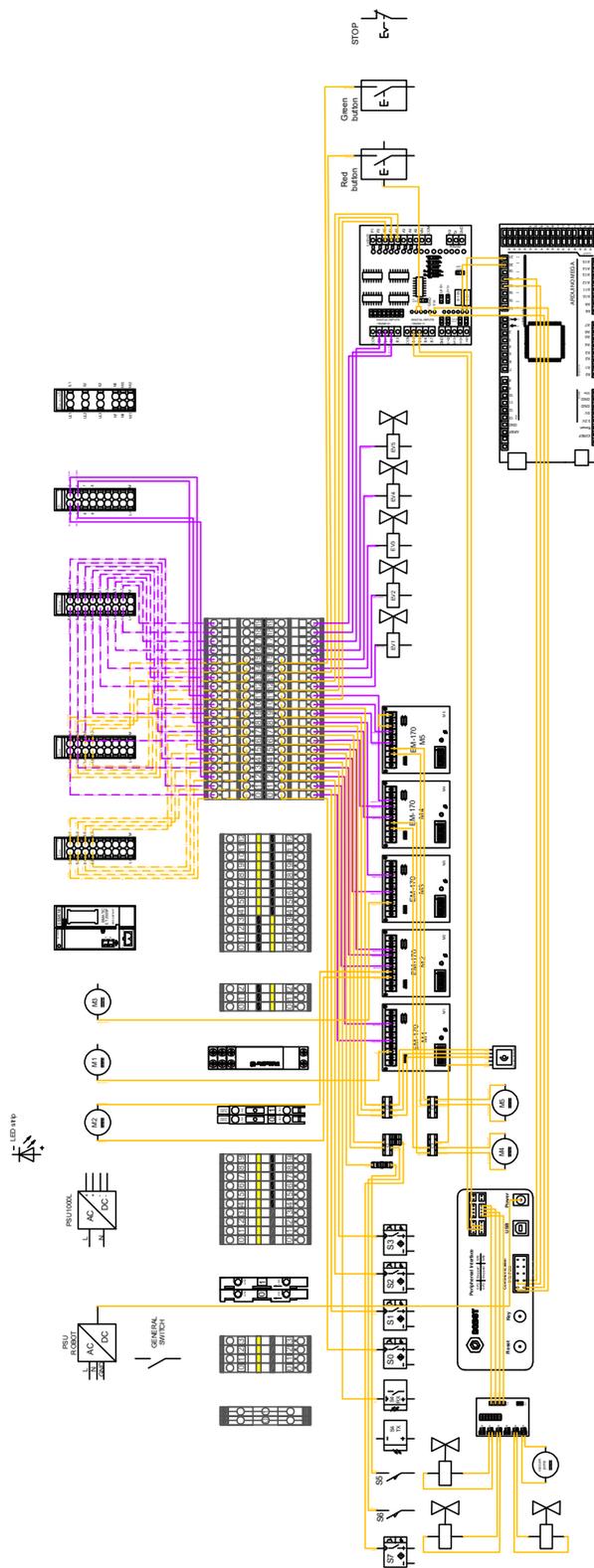


Figura A.3 Diagrama de conexiones de dispositivos del módulo 4

Apéndice B

Rutinas de control

B.1. Diagrama escalera

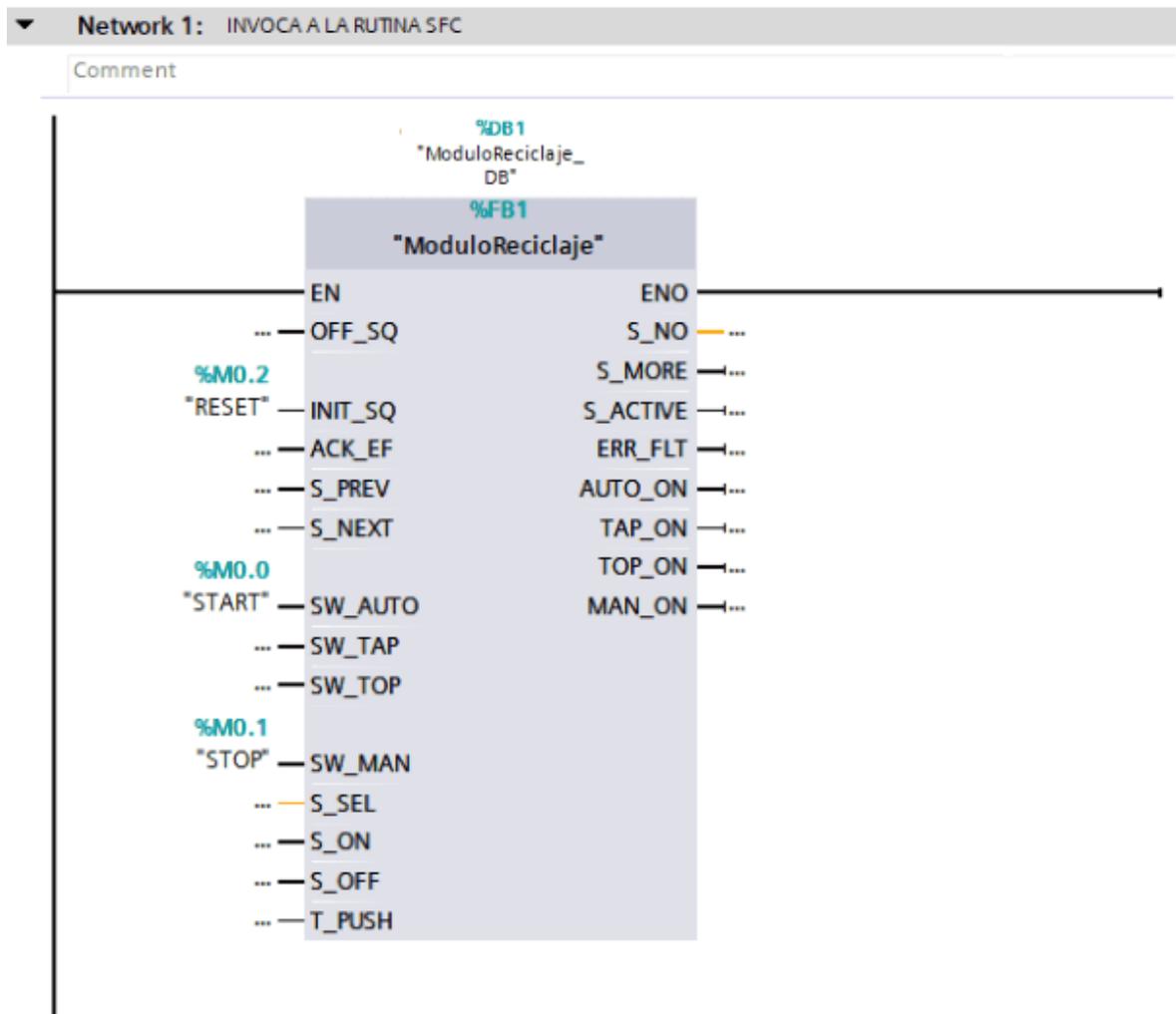


Figura B.1 Rutina LD parte 1

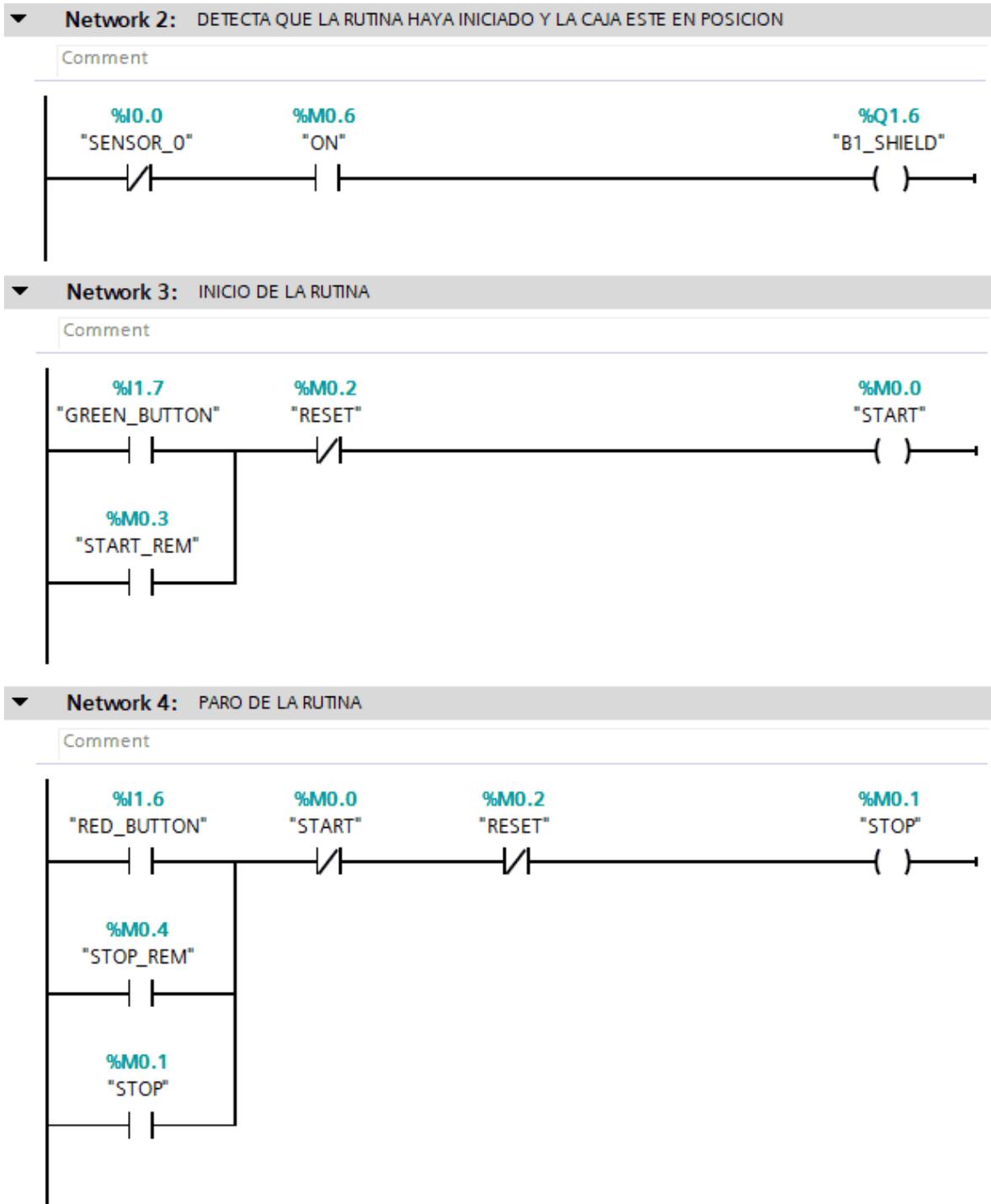


Figura B.2 Rutina LD parte 2

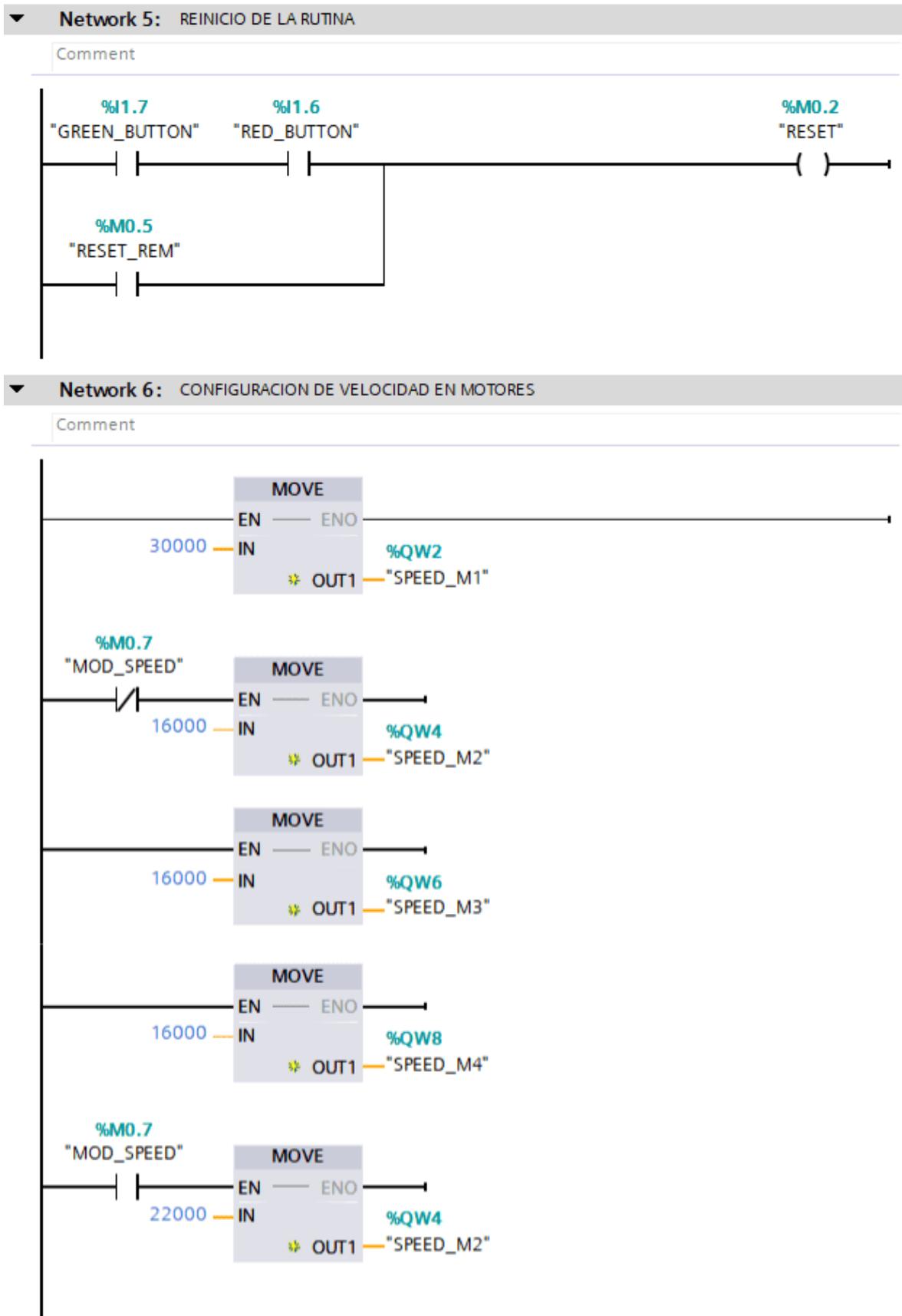


Figura B.3 Rutina LD parte 3

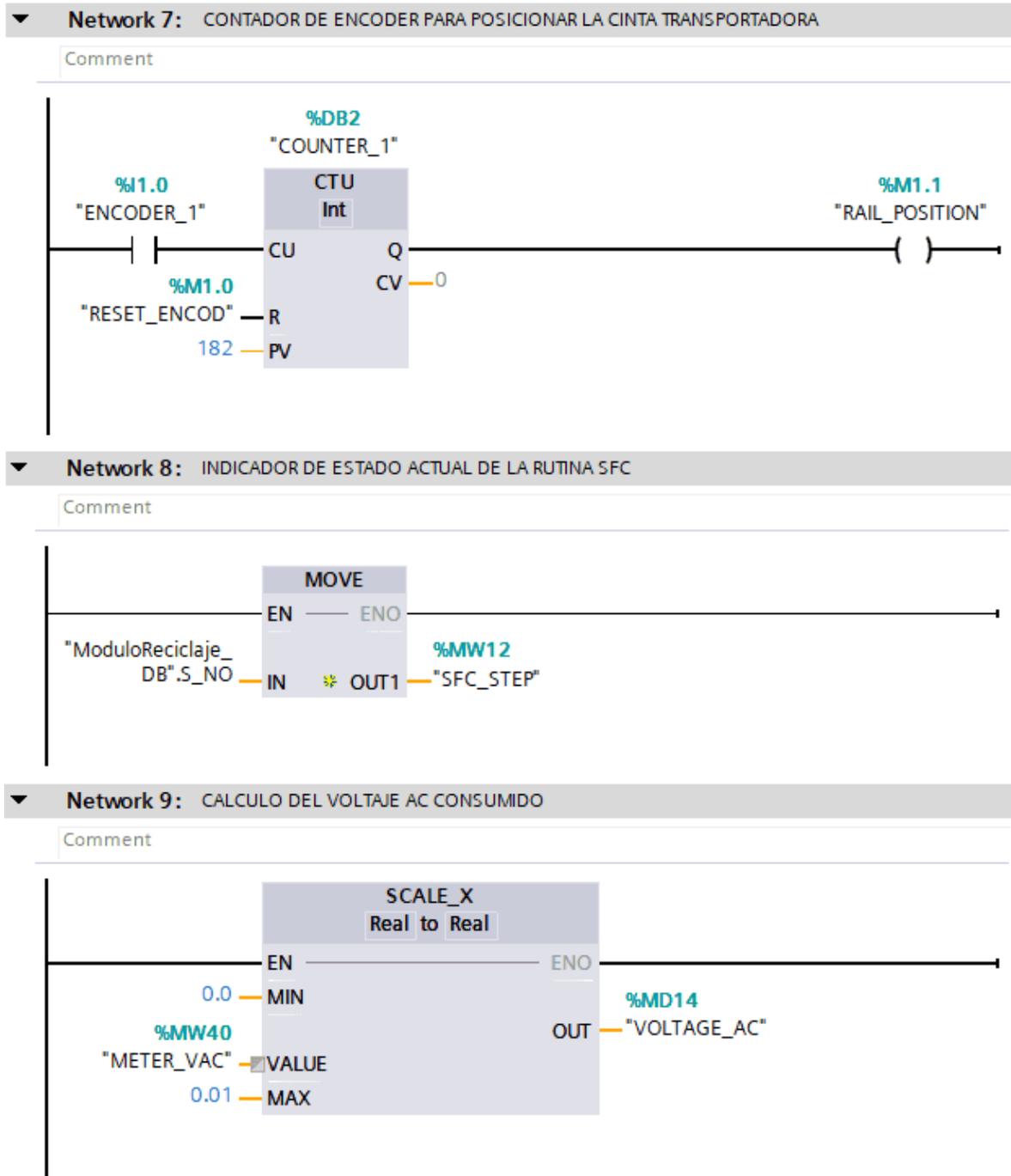


Figura B.4 Rutina LD parte 4

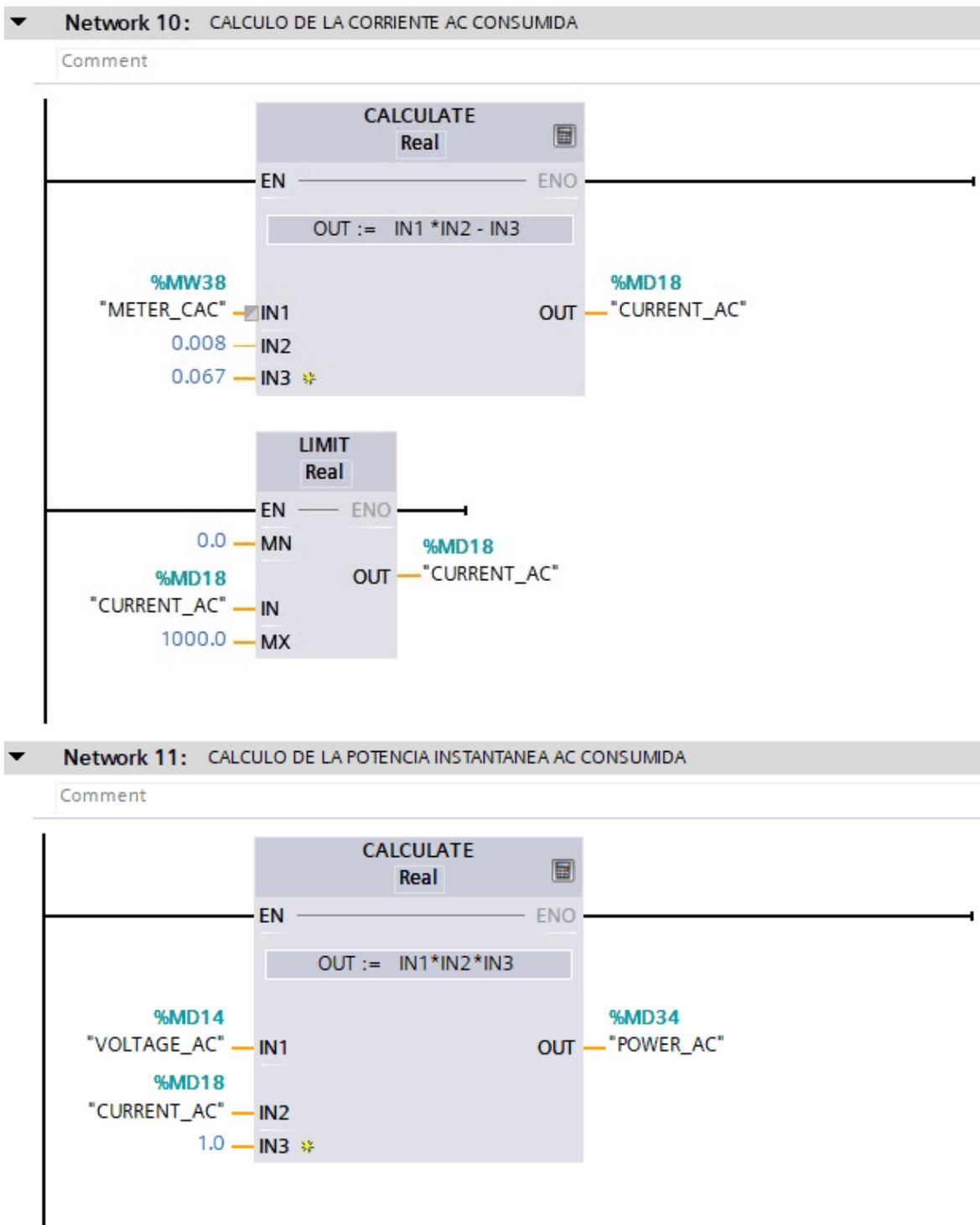


Figura B.5 Rutina LD parte 5

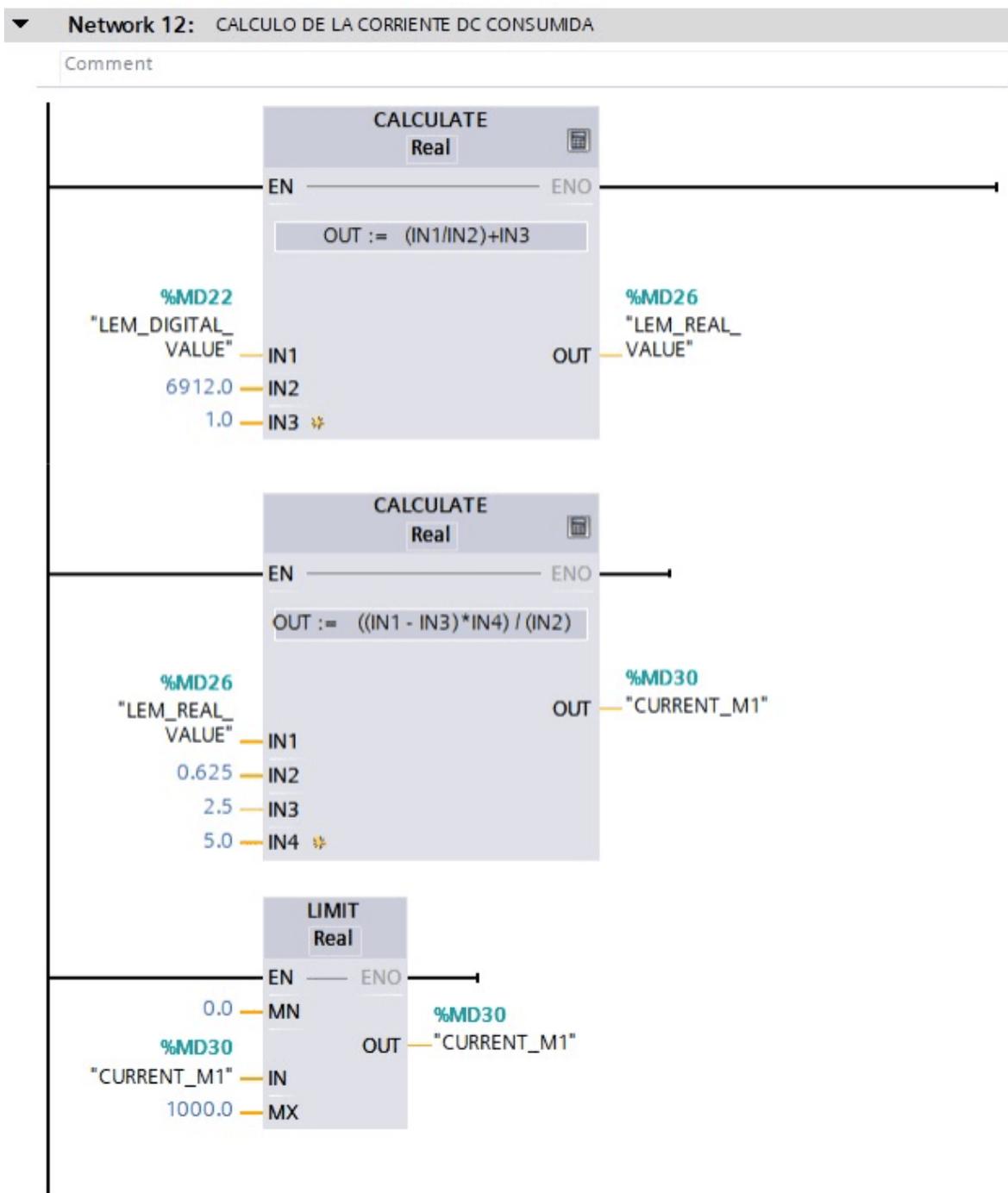


Figura B.6 Rutina LD parte 6

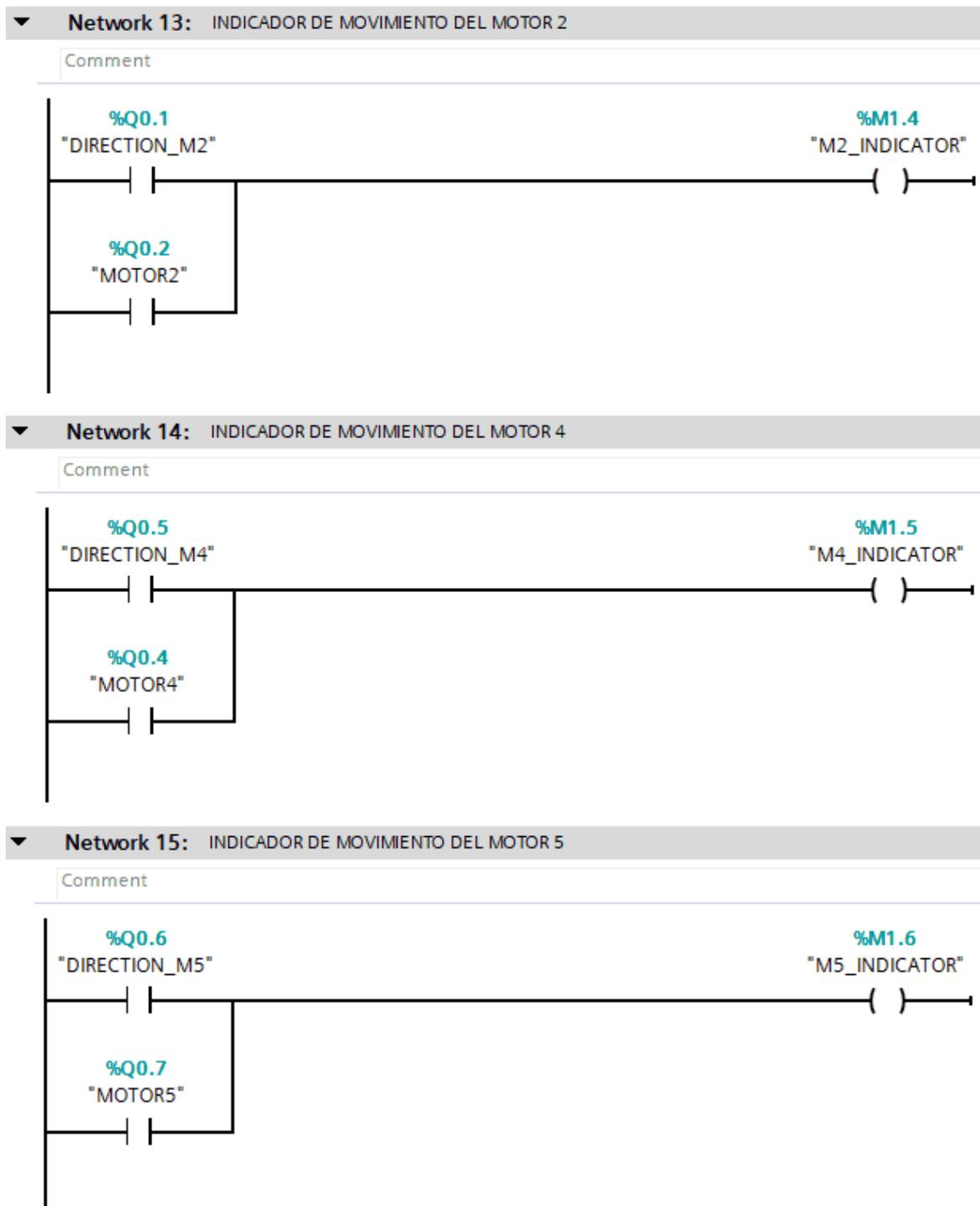


Figura B.7 Rutina LD parte 7

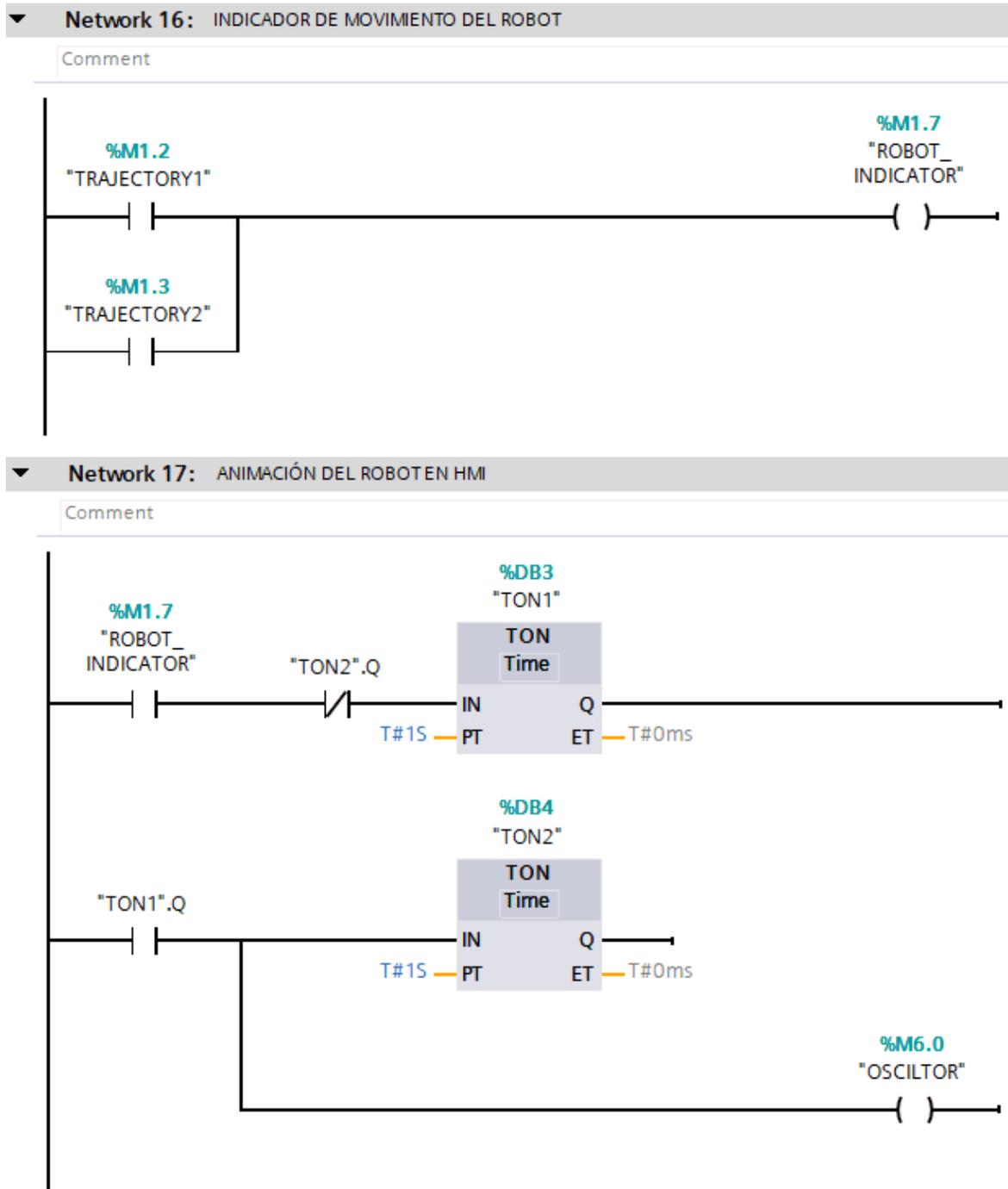


Figura B.8 Rutina LD parte 8

B.1.1. Arranque suave

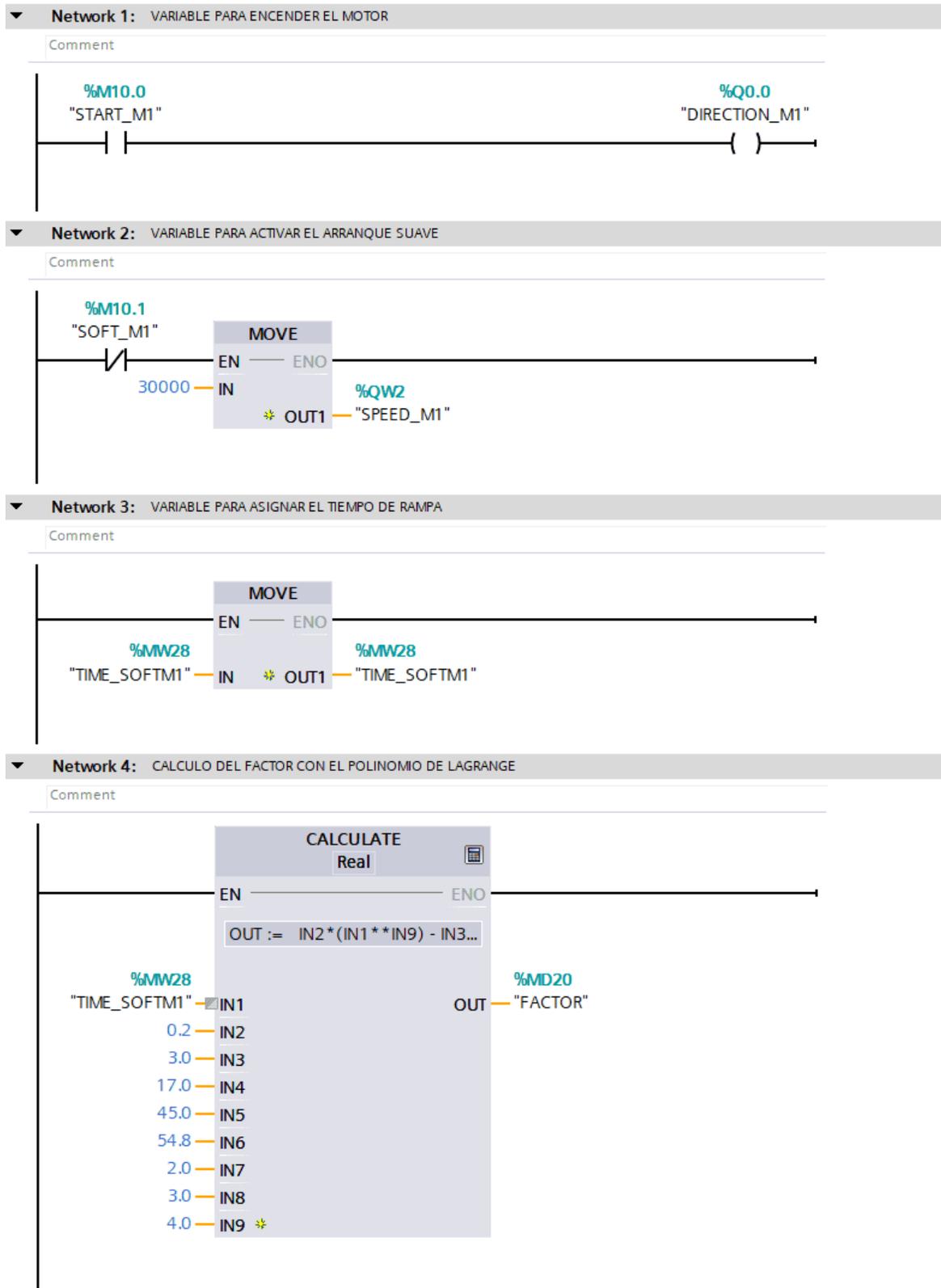


Figura B.9 Arranque suave parte 1

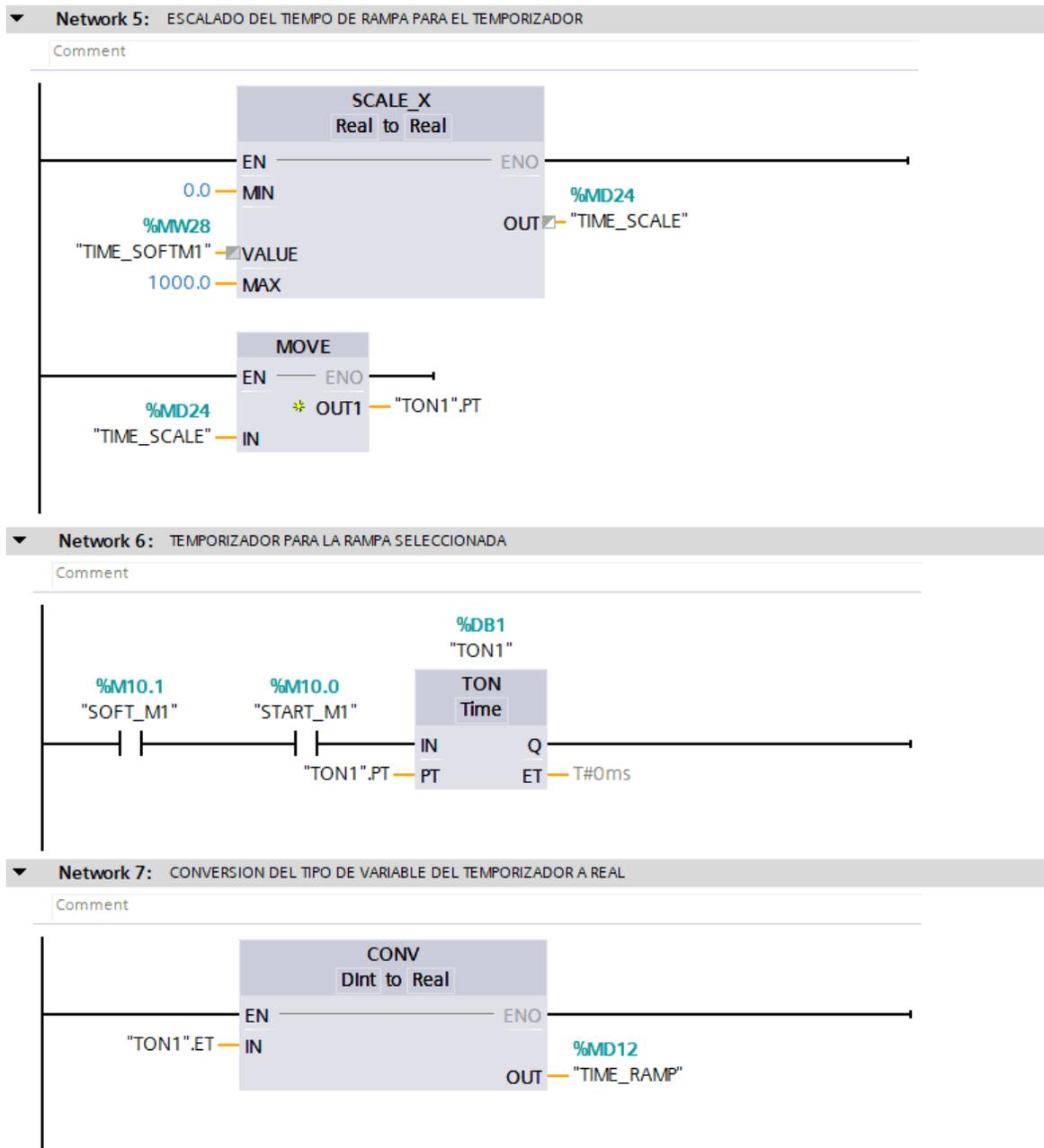


Figura B.10 Arranque suave parte 2

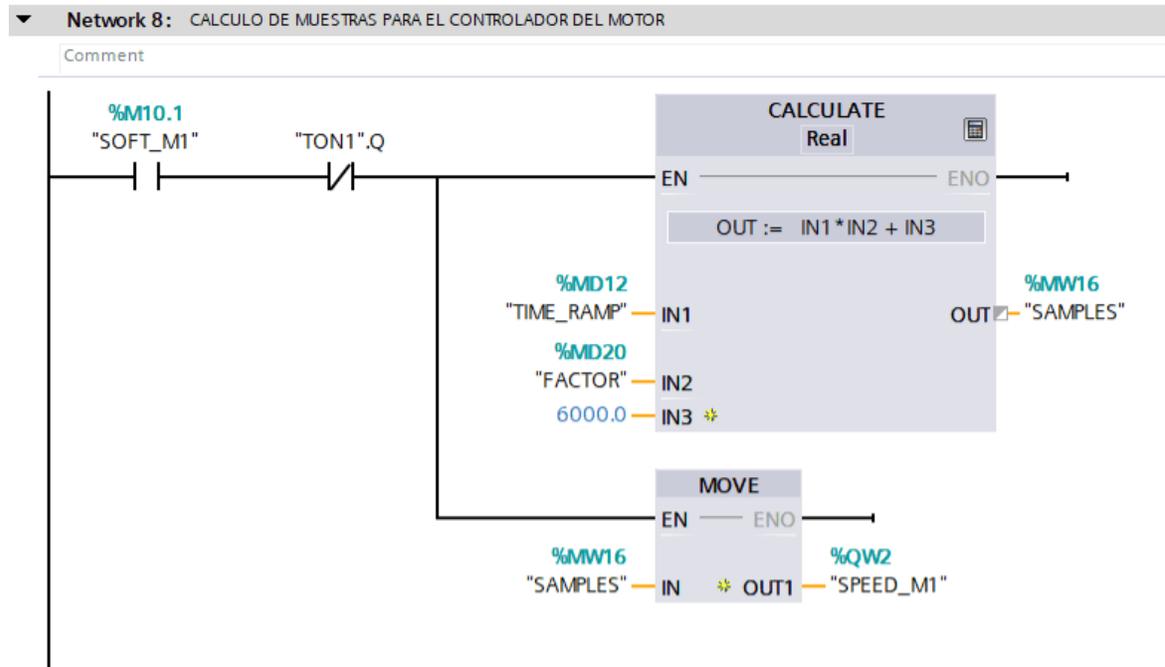


Figura B.11 Arranque suave parte 3

B.2. Gráfico de funciones secuenciales

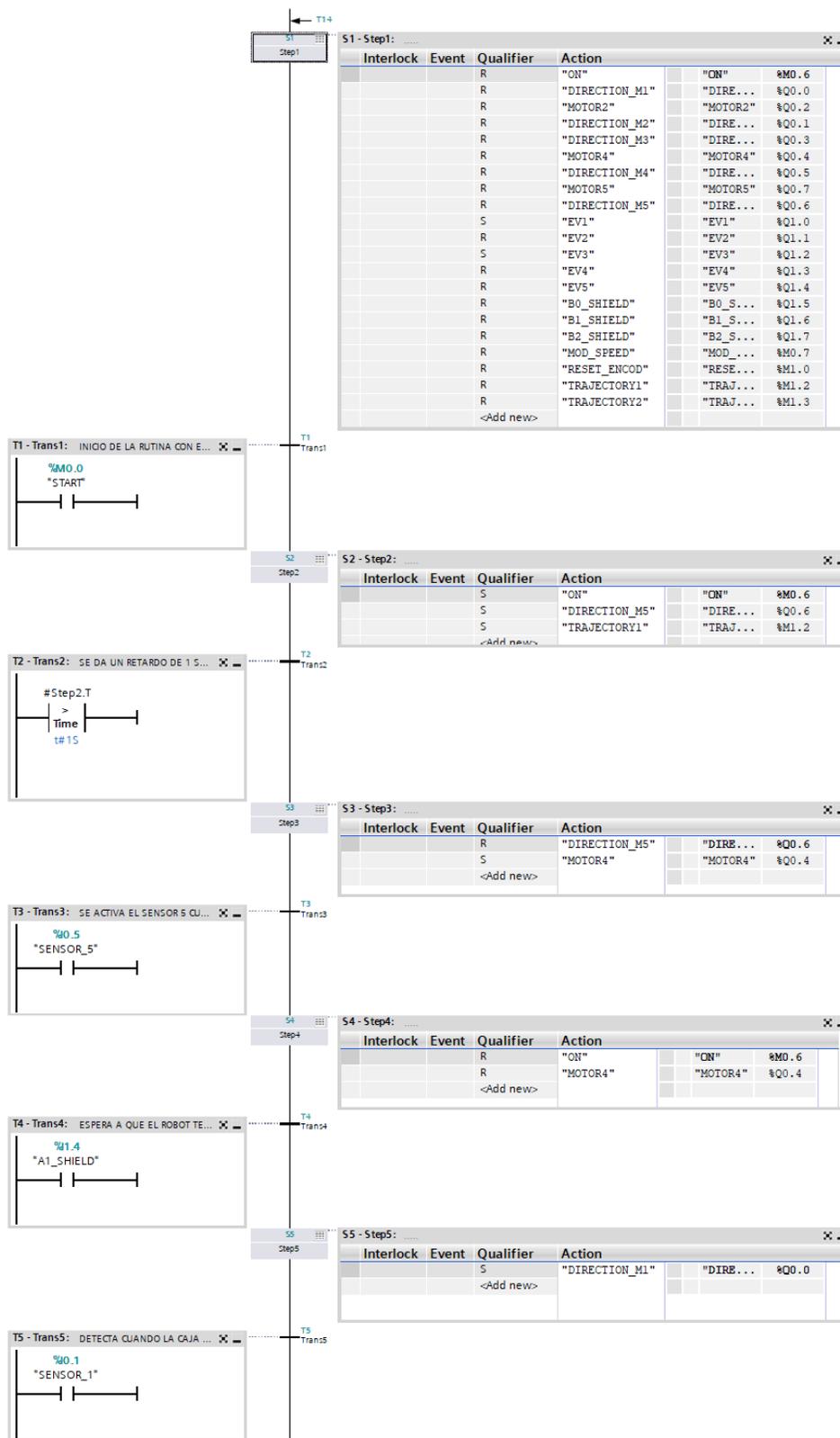


Figura B.12 Rutina SFC parte 1



Figura B.13 Rutina SFC parte 2



Figura B.14 Rutina SFC parte 3

Referencias

- [Castellanos y Solaque, 2010] Castellanos, J. & Solaque, L. (2010). Modelado con redes de Petri e implementación con GRAFCET de un sistema de manufactura flexible con procesos concurrentes y recursos compartidos. In *Universidad Militar Nueva Granada*. (Citado en página 15.)
- [Dixsen, 2024] Dixsen (2024). Low voltage current transformer [Datasheet]. (Citado en página 22.)
- [Dobot, 2019] Dobot (2019). Dobot Magician User Guide. In *Shenzhen Yuejiang Technology Co., Ltd.* (Citado en página 19.)
- [Dorf, 2015] Dorf, R. C. (2015). The Industrial Electronics Handbook. Industrial communication systems. In *University of California*. (Citado en página 14.)
- [Electromen, 2024] Electromen (2024). EM-170 DC-MOTOR CONTROLLER 12-24V 1.5A [Datasheet]. (Citado en página 21.)
- [Gabor y Livint, 2021] Gabor, G. & Livint, G. (2021). Solution for Monitoring and Remote Control Based on Modbus RTU and S7-1200 PLC. In *2021 International Conference on Electromechanical and Energy Systems (SIELMEN)* (pp. 462–465). (Citado en página 5.)
- [García Barroso, 2015] García Barroso, J. A. (2015). Automatización de un Proceso Industrial 7 (Caso Práctico-Arquitectura de Control). (Citado en página 11.)
- [García Moreno, 2014] García Moreno, E. (2014). Automatización de procesos industriales. In *2014 Universitat Politècnica de València*. (Citado en páginas 1 y 2.)
- [Hacienda, 2023] Hacienda, S. (2023). El nearshoring es la oportunidad de impulsar el crecimiento económico de México, señala Ramírez de la O. In *2023 Secretaría de Hacienda y Crédito Público*. (Citado en página 3.)
- [Hassan Reza, et al., 2021] Hassan Reza, M. N., Agamudai Nambi Malarvizhi, C., Jayashree, S., & Mohiuddin, M. (2021). Industry 4.0–Technological Revolution and Sustainable Firm Performance. In *2021 Emerging Trends in Industry 4.0 (ETI 4.0)* (pp. 1–6). (Citado en página 2.)
- [Hollifield, et al., 2008] Hollifield, B., Oliver, D., Nimmo, I., & Habibi, E. (2008). The high performance HMI handbook. (Citado en página 16.)
- [IEC, 2003] IEC, I. E. C. (2003). IEC 61131-3 Programmable controllers - Part 3: Programming languages. In *International Standard*. (Citado en página 15.)
- [INEGI, 2016] INEGI (2016). Estadísticas a propósito de la Industria automotriz. In *Asociación mexicana de la industria automotriz*. (Citado en página 2.)
- [Karl-Heinz y Tiegelkamp, 2001] Karl-Heinz, J. & Tiegelkamp, M. (2001). IEC 61131-3: Programming Industrial Automation Systems. In *Springer*. (Citado en página 15.)
- [LEM, 2017] LEM (2017). Current Transducer LTS 15-NP [Datasheet]. (Citado en páginas 23 y 44.)

- [Li, *et al.*, 2023] Li, H., Dong, C., Jia, X., Xiang, S., & Hu, Z. (2023). Design of Automatic Sorting and Transportation System for Fruits and Vegetables Based on Dobot Magician Robotic Arm. In *2023 2nd International Symposium on Control Engineering and Robotics (ISCER)* (pp. 229–234). (Citado en página 6.)
- [Luejai, *et al.*, 2019] Luejai, W., Srisongkham, W., & Kanthiya, P. (2019). Demonstration of Production System and Automation Controller for Industrial 4.0 using PLC. In *2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 774–777). (Citado en página 3.)
- [Pardo, 2015] Pardo, G. (2015). Protocolos de Comunicación Industrial. In *Tecnología en mecánica industrial, Universidad ECCI*. (Citado en página 12.)
- [Pigan y Metter, 2008] Pigan, R. & Metter, M. (2008). Automating with PROFINET. In *Industrial Communication based on Industrial Ethernet*. (Citado en página 14.)
- [Quiroz Cuenca, 2021] Quiroz Cuenca, S. (2021). Condiciones del nearshoring en México 2023. In *2021 Universidad Autónoma del Estado de México*. (Citado en página 3.)
- [Ramanathan, 2014] Ramanathan, R. (2014). The IEC 61131-3 programming languages features for industrial control systems. In *2014 World Automation Congress (WAC)* (pp. 598–603). (Citado en páginas 1 y 2.)
- [Rugged Circuits, 2024] Rugged Circuits, RC, R. C. (2024). NEW 24V Industrial I/O Shield PLC Replacement V2 [Datasheet]. (Citado en página 20.)
- [Siemens, 2013] Siemens (2013). SIMATIC S7-1500 Sistema de automatización. (Citado en página 16.)
- [Siemens, 2020] Siemens (2020). SIMATIC PCS 7 – Sequential control systems. In *Siemens Automation Cooperates with Education (SCE)*. (Citado en páginas 14 y 15.)
- [Siemens, 2021] Siemens (2021). WinCC Unified View of Things (VoT) -Getting started-. In *WinCC Unified / v17 / View of Things*. (Citado en páginas 24 y 25.)
- [Siemens, 2023a] Siemens (2023a). Configuring topology. In *PROFINET with STEP 7*. (Citado en páginas 12 y 13.)
- [Siemens, 2023b] Siemens (2023b). Industrial Ethernet Switches SCALANCE X-300 Instrucciones de servicio. In *SIMATIC NET*. (Citado en página 18.)
- [Siemens, 2023c] Siemens (2023c). SIMATIC S7-1500/ET 2000MP Sistema de automatización. (Citado en página 16.)
- [Siemens, 2024a] Siemens (2024a). SCALANCE X308-2LD. (Citado en página 18.)
- [Siemens, 2024b] Siemens (2024b). SIMATIC ET 200SP. (Citado en páginas 17, 24, 25 y 26.)
- [Siemens, 2024c] Siemens (2024c). TIA Portal (Totally Integrated Automation Portal). (Citado en página 24.)
- [Sánchez-Chávez y Martell-Chávez, 2019] Sánchez-Chávez, I. Y. & Martell-Chávez, F. (2019). Automatización multinivel de celda de manufactura robotizada utilizando máquinas de estados finitos. In *Ingeniería Investigación y Tecnología*. (Citado en página 11.)
- [Toon, 2002] Toon, K. (2002). IEC 61131-3 in safety applications. In *IEE The Application of IEC 61131 in Industrial Control (Ref. No. 2002/060)* (pp. 7/1–7/4). (Citado en página 2.)
- [Xiaodong, *et al.*, 2015] Xiaodong, Z., Jie, Z., & Ke, L. (2015). Design and implementation of control system for beer fermentation process based on SIMATIC PLC. In *The 27th Chinese Control and Decision Conference (2015 CCDC)* (pp. 5653–5656). (Citado en página 4.)