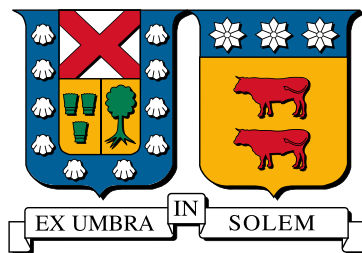


UNIVERSIDAD TÉCNICA FEDERICO SANTA
MARÍA

DEPARTAMENTO DE ELECTRÓNICA

VALPARAÍSO - CHILE



“SIMULACIÓN, AUTOMATIZACIÓN Y
PUESTA EN MARCHA VIRTUAL DE UN
SISTEMA DE TRANSPORTE Y
CLASIFICACIÓN”

MATÍAS IGNACIO HERRERA ALLENDE

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
ELECTRÓNICO.

PROFESOR GUIA: DR. MANUEL OLIVARES

PROFESOR CORREFERENTE: MSc. JORGE PONTT

NOVIEMBRE 2023

Agradecimientos

Quiero empezar agradeciendo a mis padres, Isaac y Ana, a mis hermanos y a mi familia, por todo el apoyo brindado y siempre confiar en mi, sin ellos esto no hubiese sido posible. Agradecer a mis amigos, que nos tocó pasar por diferentes desafíos durante todo este tiempo y que, sin ese constante apoyo mutuo, todo hubiera resultado más difícil. A los que comenzaron conmigo y luego nos distanciamos, a los que llegaron en la mitad y al final del camino, y a los que siempre estuvieron, solo palabras de agradecimientos para ellos. También agradecer a mi profesor guía, Manuel Olivares, por brindarme múltiples oportunidades que me ayudaron a crecer como estudiante y me permitieron convertirme en un profesional. Gracias a mis compañeros y profesor de Voleibol, que me ayudaron a practicar el trabajar en equipo para conseguir el objetivo grupal. Gracias a cada una de las personas que me aportaron hasta con lo más mínimo en mi estadía en la universidad. Gracias totales.

Simulación, automatización y puesta en marcha virtual de un sistema de transporte y clasificación

Matías Ignacio Herrera Allende

Memoria para optar al título de Ingeniero Civil Electrónico, mención Control e
Instrumentación, submención Computadores.

Universidad Técnica Federico Santa María

Profesor Guía: Dr. Manuel Olivares

Profesor Correferente: MSc. Jorge Pontt

Noviembre 2023

Resumen

En este proyecto se desarrolla una simulación dinámica 3D en tiempo real del Sistema de Transporte y Clasificación (STC), disponible en el Laboratorio de Control Industrial del Departamento de Electrónica de la UTFSM; la simulación se utiliza para diseñar y poner en marcha la automatización del sistema mediante un control virtual equivalente al sistema de control físico, aplicando así el esquema de validación software-in-the-loop (SIL). El resultado final de este trabajo, si se utiliza apropiadamente, permite acortar el tiempo de desarrollo de la automatización y control del sistema, con el cual se pueden realizar pruebas sin necesidad de utilizar el sistema físico. La simulación en tiempo real está implementada a través de PLCSIM y Ciros Studio, conectados a través de EzOPC. También se presenta la validación de la automatización realizada para el sistema virtual en el sistema físico.

Keywords: PLCSIM; Automatización; Simulación SIL tiempo real; Ez OPC, Ciros Studio

Simulation, automation and virtual commissioning of a transport and classification system

Matías Ignacio Herrera Allende

Thesis for the fulfillment of the B.S. in Electronic Engineering, major in Control & Instrumentation, minor in Computer Electronics (6 year program).

Universidad Técnica Federico Santa María

Advisor: Dr. Manuel Olivares

Co-Advisor: MSc. Jorge Pontt

November 2023

Abstract

In this project, a real-time 3D dynamic simulation of the Transport and Classification System (STC in spanish) is developed. The STC is located in the Industrial Control Laboratory of the Electronics Department at UTFSM. The simulation is employed for designing and implementing the automation of the system using a virtual control equivalent to the physical control system. This follows the software-in-the-loop (SIL) validation scheme. The ultimate outcome of this effort, when appropriately utilized, facilitates a reduction in the development time of the automation and control of the system. This enables testing without the need for the physical system. The real-time simulation is implemented through PLCSIM and Ciros Studio, connected via Ez OPC. Additionally, the validation of the automation performed on the virtual system is presented in comparison to the physical system

Keywords: PLCSIM; Automation; real-time SIL simulation; Ez OPC, Ciros Studio

GLOSARIO

Ciros Studio Es una plataforma de desarrollo que permite crear modelos de simulación en 3D para la técnica de automatización en tiempo real.

ET1 Acrónimo para Estación de Transferencia 1, asociada al STC.

ET2 Acrónimo para Estación de Transferencia 2, asociada al STC.

GRAF CET Del francés “Graphe Fonctionnel de Commande Etape Transition”. Corresponde a un diagrama de control con etapas y transiciones que describe un sistema con una secuencia lógica de funcionamiento, predefinido por variables de entrada o salida.

HH Hardware-Hardware.

HIL Hardware-In-The-Loop.

HMI Del inglés Human-Machine Interface, Se refiere a las interfaces para que operadores interactúen con los dispositivos de manera local o remota.

I/O Acrónimo en inglés de Input/Output, que se refiere a entradas y salidas.

OPC Del inglés Open Platform Communications Se refiere a un estándar para comunicación industrial que permite conectividad e interoperabilidad universal y se basa en una arquitectura Cliente-Servidor.

PG/PC La interfaz PG/PC se refiere a la conexión y comunicación entre un sistema de control programable (PC) dispositivo de programación específico llamado PG (Programming Device) o mediante una conexión directa a través de una tarjeta de interfaz PCI (PC Interface Card).

PLC En inglés Programmable Logic Controller. Los PLCs son dispositivos electrónicos que controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, procesan y reciben señales digitales y análogas y pueden aplicar estrategias de control previamente programadas en ellos.

PLCSIM Corresponde a un programa integrado en TIA Portal que permite ejecutar y comprobar el programa realizado en un PLC virtual simulado en una unidad de programación, sin la necesidad de contar con ningún equipo físico.

PROFIBUS Acrónimo en inglés de Process Field Bus, el cual es un estándar de red digital de campo abierto (bus de campo) que se encarga de la comunicación entre los sensores de campo y el sistema de control o los controladores en entornos industriales.

SIL Software-In-The-Loop.

STC Sistema de Transporte y Clasificación.

TIA Portal Del inglés “Totally Integrated Automation Portal”. Corresponde a un programa desarrollado por la empresa Siemens que integra una gran cantidad de componentes que permite, dentro de muchas opciones, programar PLCs, pantallas HMI, etc.

Índice de contenidos

| | |
|--|-----------|
| 1. Introducción | 1 |
| 1.1. Problemas a resolver y objetivos | 1 |
| 1.2. Requerimientos funcionales de la implementación virtual | 2 |
| 1.3. Estado del Arte | 2 |
| 1.3.1. Sistema de Transporte y Clasificación | 2 |
| 1.3.2. Esquemas de validación SIL y HIL | 3 |
| 2. Sistema de Transporte y Clasificación. | 5 |
| 2.1. Componentes electro-mecánicos | 7 |
| 2.1.1. Correa Principal | 7 |
| 2.1.2. Correa de Clasificación | 7 |
| 2.1.3. Motor AC | 8 |
| 2.1.4. Motor DC | 9 |
| 2.2. Componentes neumáticos | 10 |
| 2.2.1. Cilindros neumáticos | 10 |
| 2.2.2. Toberas de aspiración | 11 |
| 2.2.3. Terminal de válvulas | 11 |
| 2.2.4. Compresor | 13 |
| 2.3. Instrumentación | 13 |
| 2.3.1. Sensores | 13 |
| 2.3.2. Relés, guardamotores y automáticos | 14 |
| 2.3.3. PLC (Programmable Logic Controller) | 16 |
| 2.3.4. Tarjeta PCI Siemens CP 5611 | 22 |
| 2.3.5. HMI (Human Machine Interface) | 23 |
| 2.4. Diagrama de conexiones entre el STC, PLC y Terminal de Válvulas | 24 |
| 2.5. Requisitos funcionales de automatización lógica del STC | 25 |
| 3. Herramientas de desarrollo de esquema SIL | 26 |
| 3.1. Ciro Studio | 26 |
| 3.2. TIA Portal | 26 |

| | | |
|-----------|--|-----------|
| 3.2.1. | PLCSIM | 27 |
| 3.2.2. | PLC físico | 28 |
| 3.3. | Comunicación OPC en esquemas SIL | 28 |
| 3.3.1. | EzOPC para esquema SIL | 28 |
| 3.4. | STEP 7 | 29 |
| 3.5. | S7-Graph | 30 |
| 3.5.1. | Introducción | 30 |
| 3.5.2. | Reglas de programación | 31 |
| 3.5.3. | Elementos de una cadena secuencial | 32 |
| 3.5.4. | Acciones | 34 |
| 3.5.5. | Condiciones | 36 |
| 4. | Simulación STC en Ciroso Studio | 38 |
| 4.1. | Aplicación Ciroso | 38 |
| 4.1.1. | Librerías | 38 |
| 4.1.2. | Explorador del modelo | 40 |
| 4.1.3. | Opciones del modelo | 41 |
| 4.2. | Objetos Dinámicos y Estáticos | 43 |
| 4.2.1. | Correa Principal | 43 |
| 4.2.2. | Correa de Clasificación | 44 |
| 4.2.3. | Sensores ópticos | 44 |
| 4.2.4. | Cilindros de doble-efecto | 45 |
| 4.2.5. | Toberas de aspiración | 47 |
| 4.2.6. | Pallet y Objeto | 48 |
| 4.3. | Ambiente de simulación 3D | 49 |
| 4.4. | Dinámica del sistema | 51 |
| 4.4.1. | Grippers | 51 |
| 4.4.2. | Agregar y quitar objetos | 53 |
| 5. | Desarrollo e Implementación | 58 |
| 5.1. | Configuración de Hardware | 58 |
| 5.1.1. | Configuración red Profibus | 59 |

| | |
|---|-----------|
| 5.2. Automatización del STC | 60 |
| 5.3. Implementación en STC virtual | 63 |
| 5.3.1. Cargar programa en PLCSIM | 64 |
| 5.3.2. Enlazamiento de variables a través de EzOPC | 66 |
| 5.4. Implementación en STC físico | 69 |
| 5.5. Configuración pantalla HMI virtual | 71 |
| 6. Resultados SIL y HH | 74 |
| 6.1. Pruebas realizadas de validación física de los esquemas SIL y HH | 74 |
| 6.1.1. Prueba de validación del STC virtual - SIL | 74 |
| 6.1.2. Prueba de validación del STC físico - HH | 78 |
| 6.2. Análisis de resultados | 79 |
| 7. Conclusiones y Trabajo Futuro | 81 |
| 7.1. Conclusiones | 81 |
| 7.2. Trabajo Futuro | 83 |
| Referencias | 84 |
| 8. Anexos | 86 |
| 8.1. Planos Eléctricos [1] | 86 |
| 8.2. Proyecto nuevo y agregar dispositivos en TIA Portal | 89 |
| 8.2.1. Dispositivos en la red PROFIBUS | 89 |
| 8.2.2. Módulos PLC | 93 |
| 8.3. Programación Automatización en S7-Graph | 95 |
| 8.4. Tablas de variables PLC y HMI | 104 |

Índice de tablas

| | |
|---|----|
| 2.1. Características técnicas de la CPU | 17 |
| 2.2. Características técnicas del módulo SM 323 | 19 |
| 2.3. Características técnicas del módulo SM 334 | 21 |
| 5.4. Salidas lógicas | 63 |

| | |
|---------------------------------|----|
| 5.5. Entradas lógicas | 63 |
|---------------------------------|----|

Índice de figuras

| | |
|---|----|
| 2.1. Estaciones de Transferencia del STC | 5 |
| 2.2. Sistema de Transporte y Clasificación del LCI | 6 |
| 2.3. Correa Principal | 7 |
| 2.4. Correa de Clasificación | 8 |
| 2.5. Motor AC | 9 |
| 2.6. Motor DC | 10 |
| 2.7. Cilindro neumático de Festo | 10 |
| 2.8. Toberas de aspiración | 11 |
| 2.9. Terminal de válvulas CPV10DI01 de Festo | 12 |
| 2.10. Compresor | 13 |
| 2.11. Sensores de proximidad | 14 |
| 2.12. Tablero Eléctrico | 15 |
| 2.13. Esquema de conexiones de módulo de entradas y salidas digitales | 18 |
| 2.14. Esquema de conexiones de módulo entradas y salidas análogas | 20 |
| 2.15. PLC | 22 |
| 2.16. Tarjeta PCI Siemens CP 5611 | 23 |
| 2.17. HMI KTP700 Basic DP | 23 |
| 2.18. Diagrama de conexiones entre el STC, PLC y Terminal de Válvulas | 24 |
| 3.19. Conexión entre PLCSIM y CiroS Studio | 29 |
| 3.20. Bloques que componen un control en S7-Graph | 31 |
| 3.21. Diagrama de una etapa-transición | 32 |
| 3.22. Ejemplo de un salto | 32 |
| 3.23. Ejemplo de una rama alternativa | 33 |
| 3.24. Ejemplo de una rama simultánea | 33 |
| 3.25. Ejemplo de fin de cadena | 34 |
| 3.26. Etapa con sus acciones que debe realizar | 35 |
| 4.27. Librerías | 38 |

| | |
|---|----|
| 4.28. Ejemplo de modelos utilizados en el proyecto | 39 |
| 4.29. Explorador del modelo | 40 |
| 4.30. Ejemplo propiedades | 41 |
| 4.31. Ejemplo ajustes del modelo | 42 |
| 4.32. Correa Principal | 43 |
| 4.33. Correa Clasificación | 44 |
| 4.34. Sensores de proximidad en CiroS | 45 |
| 4.35. Frenos | 46 |
| 4.36. Extensión y Elevación | 47 |
| 4.37. Tobera de aspiración | 48 |
| 4.38. Pallet y Objeto | 48 |
| 4.39. Vista aerea STC simulado | 49 |
| 4.40. Vista desde el costado STC simulado | 50 |
| 4.41. Vista estaciones del STC | 50 |
| 4.42. Tablero con botones. | 51 |
| 4.43. Punto de agarre del pallet a la correa principal | 52 |
| 4.44. Superficie Activa en el pallet | 53 |
| 4.45. Configuración Object Source | 54 |
| 4.46. Punto de agarre del pallet a Object Source | 55 |
| 4.47. Botón A.O. para generar objeto | 55 |
| 4.48. Punto de agarre del pallet a Object Sink | 56 |
| 4.49. Botón Q.O. para quitar objeto | 57 |
| 5.50. PLC junto con sus módulos en TIA Portal | 58 |
| 5.51. Red Profibus | 59 |
| 5.52. Diagrama Grafcet de la aplicación secuencial programada | 61 |
| 5.53. Implementación del STC virtual | 64 |
| 5.54. Iniciación Simulación PLC virtual | 65 |
| 5.55. Carga avanzada | 66 |
| 5.56. OPC Client | 67 |
| 5.57. Variables compartidas | 68 |
| 5.58. Ejemplo de conexión | 68 |

| | |
|---|-----|
| 5.59. EzOPC Server | 69 |
| 5.60. Búsqueda de dispositivos a través de la red PROFIBUS | 70 |
| 5.61. Carga del programa en PLC a través de la red PROFIBUS | 71 |
| 5.62. HMI | 72 |
| 5.63. Ejemplo HMI Imagen del Sistema | 73 |
| 6.64. Detección de pallet con carga y freno izquierdo activado | 74 |
| 6.65. Pallet en ET1 y cilindros realizando la transferencia del objeto a correa de clasificación | 75 |
| 6.66. Objeto en correa de clasificación e inicio de posicionar al pallet en ET2 | 75 |
| 6.67. Objeto en ET2 esperando a ser movido hacia el pallet | 76 |
| 6.68. Objeto posicionado nuevamente en el pallet | 77 |
| 6.69. Pallet sin carga por la correa principal | 77 |
| 6.70. Funcionamiento automatización en STC físico 1 | 78 |
| 6.71. Funcionamiento automatización en STC físico 2 | 79 |
| 6.72. Funcionamiento automatización en STC físico 3 | 79 |
| 8.73. Plano eléctrico de las conexiones al PLC | 86 |
| 8.74. Plano eléctrico principal | 87 |
| 8.75. Plano eléctrico del motor DC | 88 |
| 8.76. Crear proyecto en TIA Portal | 89 |
| 8.77. Agregar CPU | 90 |
| 8.78. Agregar HMI | 90 |
| 8.79. Agregar Tarjeta CP 5611 y Terminal de Válvulas | 91 |
| 8.80. Agregar GSD Terminal de Válvulas | 92 |
| 8.81. Configuraciones para la red PROFIBUS | 93 |
| 8.82. Agregar módulos | 94 |
| 8.83. Crear bloque de función con lenguaje graph | 95 |
| 8.84. Creación bloque de datos y bloque de llamada en Main | 96 |
| 8.85. Tabla de variables PLC | 104 |
| 8.86. Tabla de variables HMI | 105 |

1. Introducción

1.1. Problemas a resolver y objetivos

En los últimos años, se han estado desarrollando sistemas virtuales para facilitar la puesta en marcha de proyectos de automatización y control industriales. Los sistemas virtuales representan simulaciones gráficas por computador que permiten la simulación de sistemas dinámicos y son controlados por dispositivos de control periféricos que pueden ser simulados o reales [2]. El principal propósito de los sistemas virtuales es permitir a los ingenieros inspeccionar y evaluar el desempeño de las aplicaciones industriales desarrolladas vía computador, para disminuir costos y sin riesgos de daños a personas o máquinas durante el desarrollo y programación de estas [3].

El objetivo principal de este trabajo consiste en desarrollar una simulación dinámica 3D en tiempo real del Sistema de Transporte y Clasificación (STC) disponible en el Laboratorio de Control Industrial, para diseñar y poner en marcha su automatización mediante un sistema de control virtual equivalente al sistema de control físico. Dado este objetivo primario se desprenden los siguientes objetivos específicos:

- Configurar y simular el STC virtual 3D en un PC con componentes mecánicas y electroneumáticas funcional y visualmente equivalentes al STC físico, en Ciro Studio.
- Conectar mediante OPC, el STC virtual con un sistema de control virtual equivalente al sistema de control físico formado por un PLC S7-300 y un Terminal de Válvulas CPV DI01.
- Diseñar la automatización del STC mediante Grafcet, implementarla en lenguaje S7-Graph y validarla por simulación mediante la conexión del STC virtual con el sistema de control virtual.
- Aplicar la automatización validada por simulación cargando el código desarrollado en el sistema de control físico conectado con el STC mediante I/O físicas, y verificar su desempeño.

1.2. Requerimientos funcionales de la implementación virtual

El trabajo presenta los siguientes requerimientos funcionales en lo que respecta a la implementación virtual:

- El STC virtual debe tener conectividad OPC con un sistema de control virtual equivalente al hardware físico disponible en el Laboratorio de Control Industrial: PLC S7-300 y Terminal de Válvulas CPV DI01 conectados en red Profibus DP.
- Automatizar el STC virtual mediante el sistema de control virtual según los requerimientos de automatización establecidos en la memoria de Angelo López [1] y en guías de laboratorio recientes, mediante diseño Grafcet y su implementación en lenguaje S7-Graph.
- Aplicar la automatización virtual del STC virtual al STC físico más el hardware de automatización físico sin mayores cambios.

1.3. Estado del Arte

1.3.1. Sistema de Transporte y Clasificación

Anteriormente ya se ha trabajado con el mismo Sistema de Transporte y Clasificación que se utiliza en este proyecto de título, específicamente Angelo López en "Instrumentación, Automatización y Supervisión de un Sistema de Transporte y Clasificación" [1]. En este trabajo Angelo se enfoca en la integración de componentes electroneumáticos y en la puesta en marcha del sistema de control, desde el ensamblaje e instrumentación, hasta la automatización. El proyecto realizado es de gran importancia para este trabajo de título, debido a que se inicia desde la base de éste, apoyándose en la documentación existente. El principal aporte de este trabajo fue entregar una herramienta didáctica para el laboratorio de control industrial, dónde los alumnos puedan trabajar con el sistema de forma que tengan un acercamiento con tecnologías de amplio uso en la industria, como son PLC, buses de campos (PROFIBUS), terminal de válvulas, entre otros.

1.3.2. Esquemas de validación SIL y HIL

Los esquemas de validación SIL (Software-in-the-Loop)[4], HIL (Hardware-in-the-Loop)[5], PIL (Processor-in-the-Loop)[6] y MIL (Model-in-the-Loop)[7] son técnicas utilizadas en el desarrollo y validación de sistemas críticos y controladores. Cada uno de estos enfoques tiene un propósito específico en el proceso de desarrollo y validación y son utilizados principalmente en la industria automotriz y aeroespacial, aunque últimamente se ha incrementado su uso en otras especialidades. Estas técnicas permiten probar y validar sistemas complejos antes de su implementación en hardware real. En específico en este trabajo de título son de importancia los esquemas de validación SIL y HIL.

- **SIL:** En SIL se ejecuta el software del sistema en un entorno de simulación virtual en un computador. Este entorno simula el comportamiento del hardware en tiempo real. El esquema se utiliza principalmente para probar y validar el software del sistema antes de que se integre con el hardware real. También es útil para verificar el comportamiento del software en diversas condiciones. Sus principales ventajas son que tiene costos reducidos, opera bajo un entorno controlado, se pueden hacer pruebas tempranas con capacidad de repetición y tiene un escalamiento sencillo. Este esquema es utilizado, por ejemplo, para programar robots industriales, con el fin que pueda interpretar el código real del programa y generar el movimiento exacto, con lo cual se obtiene una política óptima de funcionamiento y también libre de colisiones [8].
- **HIL:** En HIL se conecta el hardware de control con el sistema en un entorno virtual. Es utilizado para probar la interacción del software con componentes de hardware reales como sensores, actuadores y controladores. Esto permite verificar el comportamiento del sistema completo. Dentro de sus ventajas se encuentran que permite una validación más completa que en SIL, detección temprana de problemas de integración hardware-software y pruebas de seguridad crítica. Un ejemplo en concreto puede ser la industria aeroespacial, dónde utilizan HIL para probar el software de control de vuelo en un entorno que simula las señales de sensores y actuadores del avión. Otro ejemplo es lo que realiza Vosáhl [9], utilizando un controlador físico utilizando Raspberry Pi para controlar un péndulo de forma simulada.

Ambos enfoques son importantes en el desarrollo de sistemas críticos donde la seguridad y la confiabilidad son fundamentales. La elección de SIL y HIL depende de los objetivos de prueba y las etapas del ciclo de desarrollo del sistema. A menudo, se utilizan en conjunto para lograr una validación integral del sistema, comenzando con SIL para las pruebas de software y luego HIL para las pruebas de hardware-software y validación del sistema completo.

NOTA: Si bien la intención en un principio fue implementar ambos esquemas, SIL y HIL, finalmente éste último no fue posible, debido a que no se contaba con los recursos de hardware necesarios para su implementación. En particular, la tarjeta CP 5611 que se encuentra integrada en el PC que se utiliza en este trabajo y permite la comunicación con el sistema de control físico, no presenta el soporte requerido en software para una integración a través de OPC entre la simulación y el PLC físico.

2. Sistema de Transporte y Clasificación.

La planta sobre la cual se trabaja este proyecto, Sistema de Transporte y Clasificación (STC), es un sistema electro-neumático de selección de productos. El STC está compuesto por una banda transportadora principal con forma de riñón y una cinta transportadora recta que atraviesa a la primera por su parte más ancha. La idea es emular una línea de producción, donde dependiendo de alguna característica o evento, se desvía el producto de la línea principal hacia un proceso secundario, en la Estación de Transferencia 1 (ET1), para luego volver a la línea principal, realizado por Estación de Transferencia 2 (ET2).

En la figura 2.1 se puede ver las posiciones de las estaciones y en la figura 2.2 se puede ver el STC completo.

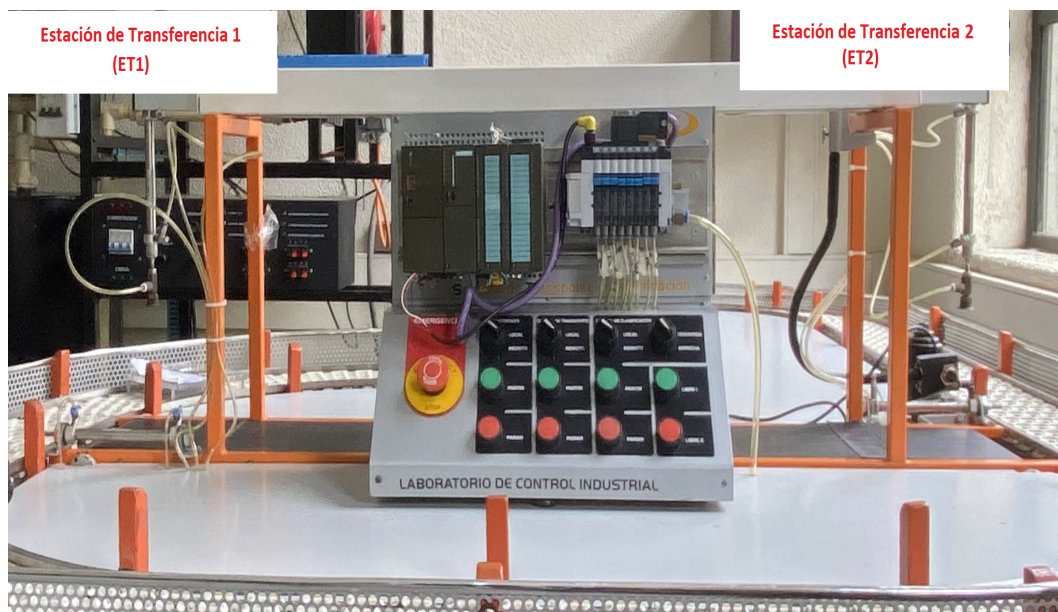
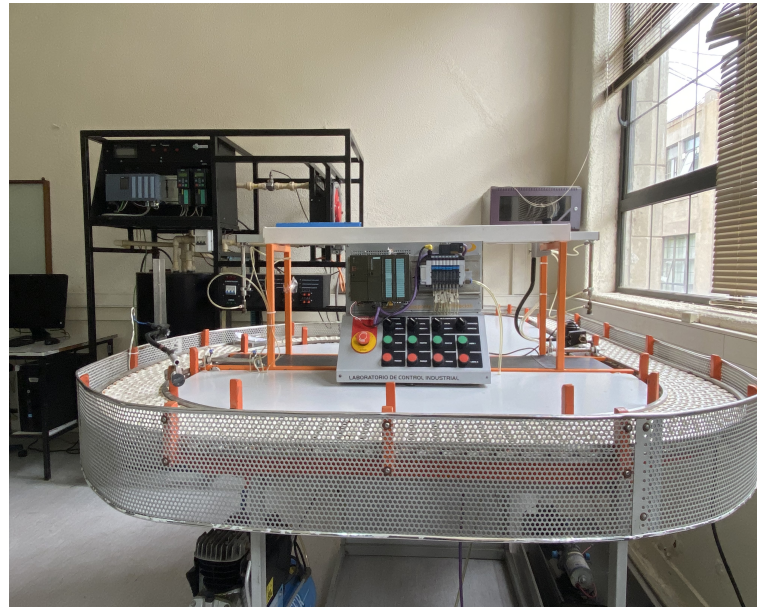
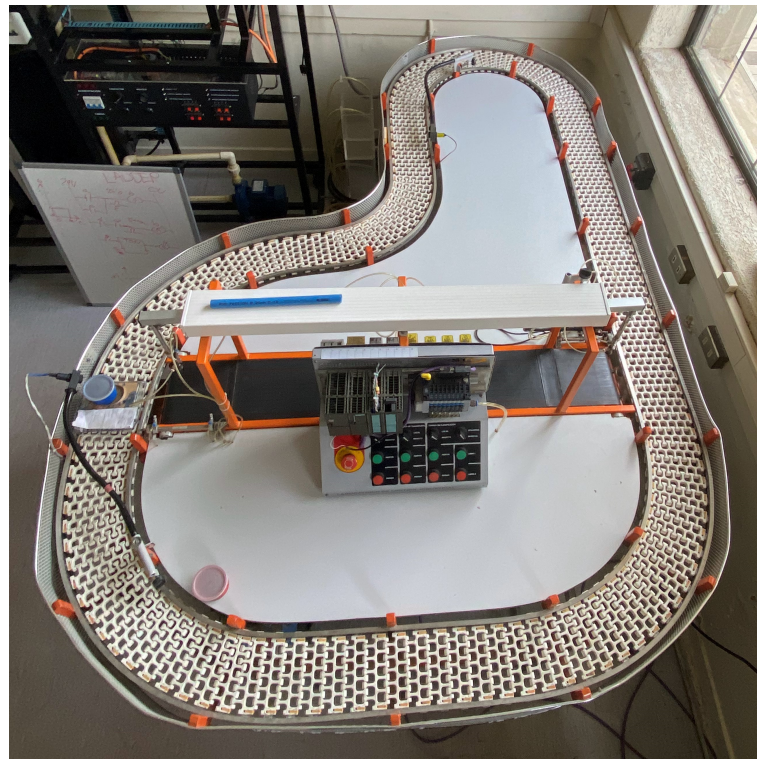


Figura 2.1: Estaciones de Transferencia del STC



(a) Vista frontal



(b) Vista desde arriba

Figura 2.2: Sistema de Transporte y Clasificación del LCI

Existen componentes de distintos tipos que permiten hacer funcionar de forma adecuada este

sistema, que se proceden a describir a continuación.

2.1. Componentes electro-mecánicos

2.1.1. Correa Principal

Corresponde a la correa que se encarga en el transporte del pallet y es accionada mediante motor de corriente alterna. Se acopla con el motor mediante un reductor, una cadena y un piñón.

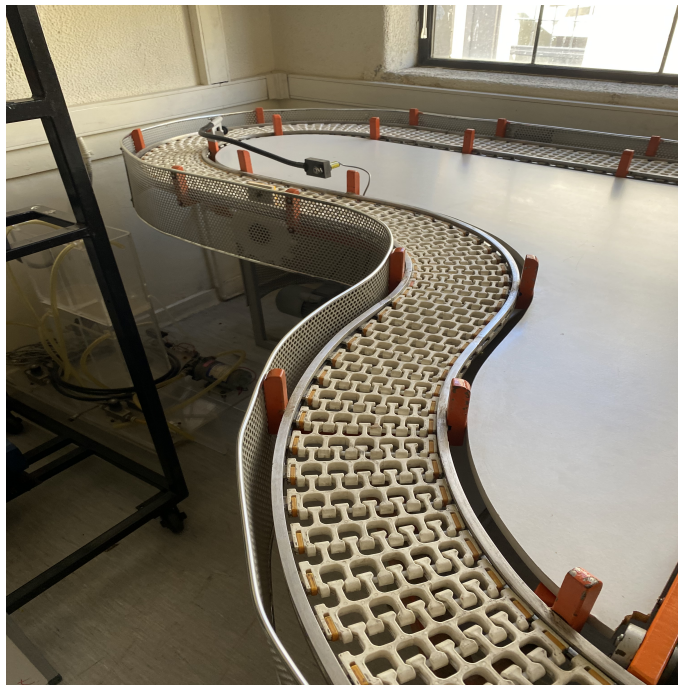


Figura 2.3: Correa Principal

2.1.2. Correa de Clasificación

Corresponde a la correa que se encarga de la clasificación, la cual lleva la carga de una estación a otra. Su accionamiento es mediante un motor de corriente continua al cual está acoplado directamente a un polín de arrastre.

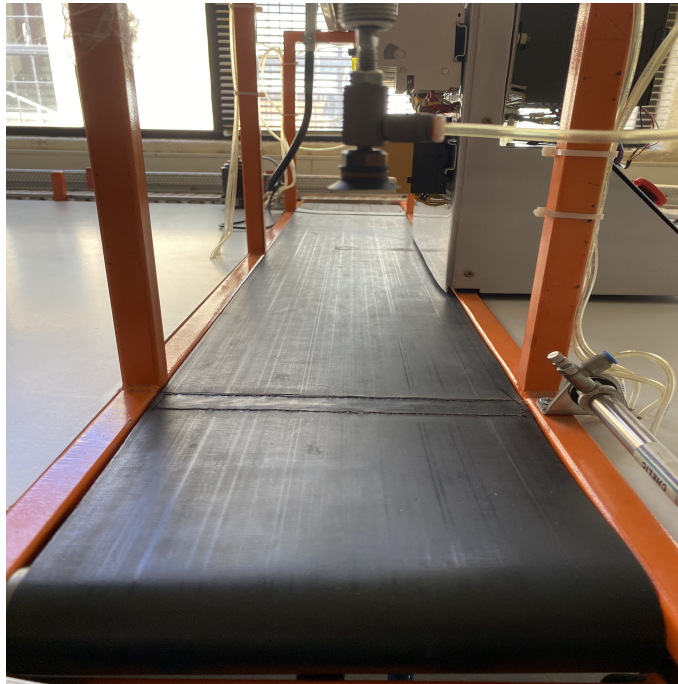


Figura 2.4: Correa de Clasificación

2.1.3. Motor AC

La *Correa de Transporte* del STC es impulsada por un motor monofásico (220 V 50 Hz) de corriente alterna marca *Siemens*. El motor posee una potencia de 0.47 KW, con un $\cos(\phi) = 0.95$ y consume una corriente nominal de 2.65 a a 1395 revoluciones por minuto.

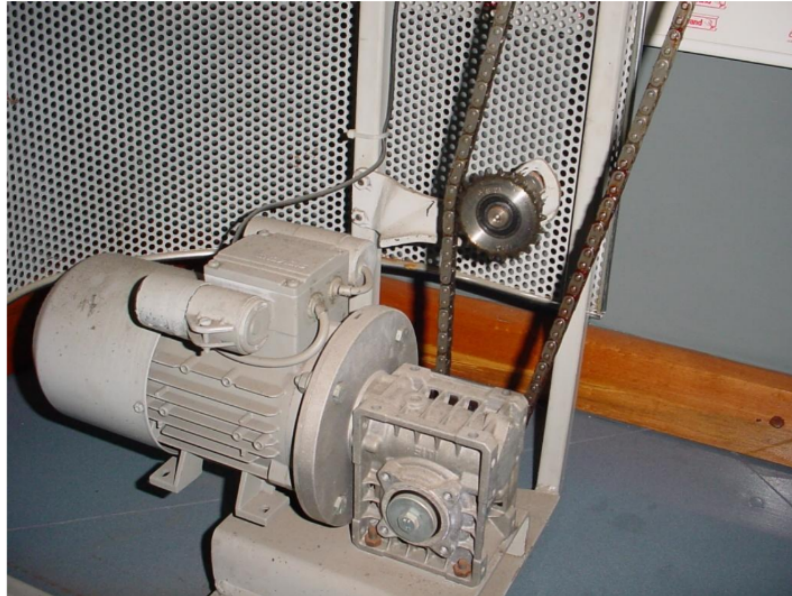


Figura 2.5: Motor AC

2.1.4. Motor DC

EL STC posee un motor DC marca *Kamco*. Este motor funciona con 12 Vdc y es capaz de generar $130\text{kg}/\text{cm}^2$ de presión. Este motor es el encargado del desplazamiento de la Correa de Clasificación. Cabe destacar que este motor puede funcionar en ambos sentidos de giro y con dos velocidades.



Figura 2.6: Motor DC

2.2. Componentes neumáticos

2.2.1. Cilindros neumáticos

El STC físico cuenta con 6 cilindros neumáticos. Dos de ellos son marca *Festo*, y los otros cuatro son marca *Chelic* de la línea 330. Según sus tamaños realizan diferentes tareas.



Figura 2.7: Cilindro neumático de Festo

2.2.2. Toberas de aspiración

Se tienen 2 toberas de aspiración F20 Cloropreno, M5 hembra, las cuales tienen la función de tomar la carga para cuando se realice la clasificación.



Figura 2.8: Toberas de aspiración

2.2.3. Terminal de válvulas

Este terminal permite mezclar variados actuadores neumáticos y conexiones eléctricas en un solo elemento de forma compacta, ver figura 2.9. Los actuadores o electroválvulas poseen formas de placas, de esta forma se pueden montar una al lado de otra, permitiendo conectar hasta 8 de ellas en un solo terminal. La conexión eléctrica puede ser individual para cada válvula o un solo bus de campo para todas las conexiones.

En este caso el terminal posee 6 válvulas monoestables de 5/2 vías, las cuales van conectadas a los cilindros neumáticos y también posee dos placas generadoras de vacío, que están conectadas a las toberas de aspiración.

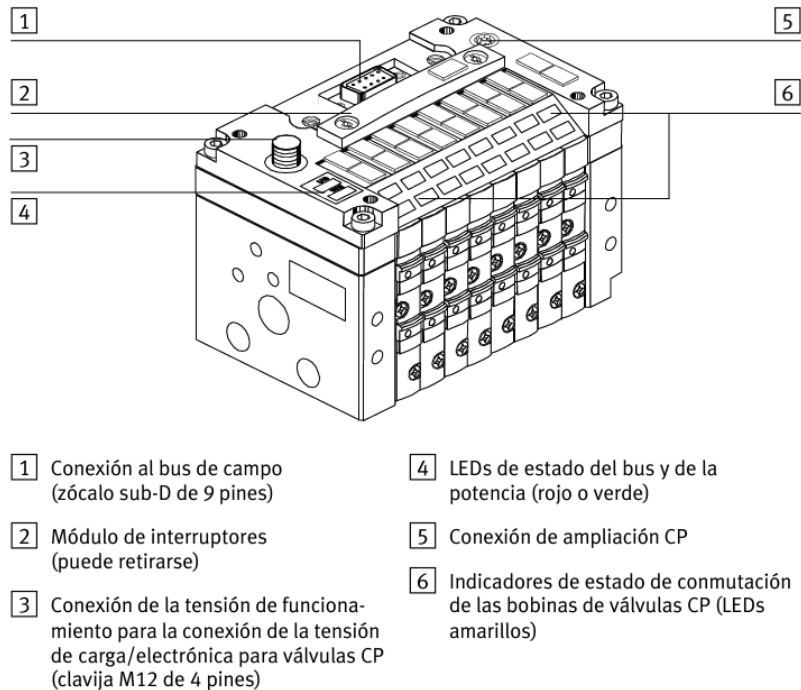


Figura 2.9: Terminal de válvulas CPV10DI01 de Festo

También, la interfaz eléctrica es un cabezal que posee un conector DB9 serial. Este cabezal está diseñado para trabajar con el protocolo Profibus DP, por lo que a través de esta interfaz puede conectarse directamente con elementos Profibus DP de Siemens, formando parte de una red mayor y por ende, se puede acceder al terminal de forma directa, por lo que para el usuario el terminal es como un módulo más del PLC.

Otras características del terminal que se pueden destacar son:

- Posee leds indicadores del estado de las válvulas, del estado del bus y de la alimentación.
- Posee un dip switch para configurar las propiedades del terminal dentro de la red profibus.
- Posee accionamiento manual de las válvulas de distribución electroneumáticas.
- Las conexiones y desconexiones neumáticas se hacen de forma rápida, ya que posee conectores del tipo "Quick-Start"(QS).

2.2.4. Compresor

Para alimentar el STC se tiene un motocompresor marca *ABAC*, su motor consume 1.5 *KW* generando 2 *HP* de potencia. Además, posee un acumulador de 24 *lts* y un regulador de presión incorporado. El motocompresor logra llegar como máximo a 8 *BAR*. Otra característica importante, es que genera un alto nivel de ruido, cercano a 99 *dB*

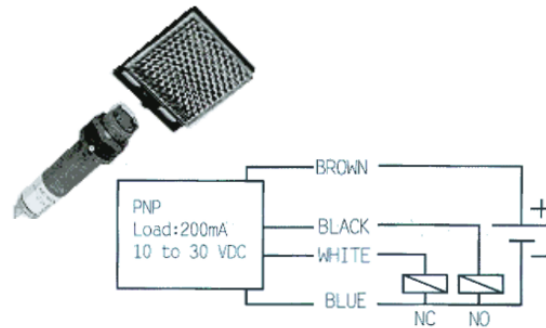


Figura 2.10: Compresor

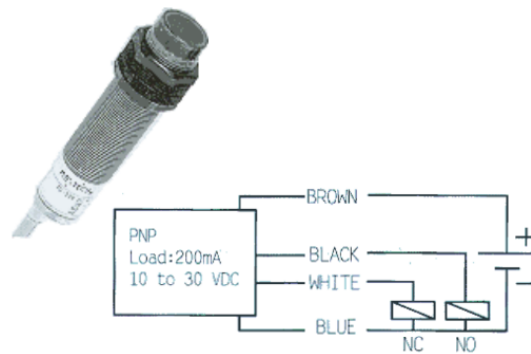
2.3. Instrumentación

2.3.1. Sensores

Se tienen 3 sensores infrarrojos PNP, dos difuso-reflectivo y uno retro-reflectivo. Tienen alcance de 100 mm, 400 mm y 2000 mm, además de que pueden ser polarizados con voltajes entre 10 y 30 Vdc.



(a) Retro-reflectivo



(b) Difuso-reflectivo

Figura 2.11: Sensores de proximidad

2.3.2. Relés, guardamotores y automáticos

El sistema presenta relés con bobinas de 24V dc para el accionamiento con PLC y de 24V ac para los comandos a través de botoneras. Se utilizan relés encapsulados de 8 y 11 pines, con bases en montaje en riel DIN. Se detalla los tipos de relés que se utilizan:

- KH-102-2C: Relé marca Koino, posee dos polos (1NC+NA), sus contactos soportan como máximo 10A, pero se recomienda solo 5A para cargas inductivas y 7.5A para cargas resistivas. Con respecto a las bobinas se tienen 4 relés con bobina de 24v dc y 1 con bobina de 220V ac.
- TRP6822: Relé marca CCD, puede resistir hasta 12A, pero se recomienda hasta 10A en cargas inductivas. Se tiene una bobina de 24V dc y otro con bobina de 220V ac. Estos relés debido al rano de corriente que manejan se utilizan para controlar el compresor.

En la figura 2.12 se puede ver el tablero eléctrico del sistema, donde se encuentran los relés de control, la regleta de conexiones, los guardamotores, los interruptores termomagnéticos, en la fila superior se pueden ver seis relés, un guardamotor y los tres automáticos, y en la fila de abajo del tablero, se puede apreciar la regleta, dos relés y un guardamotor



Figura 2.12: Tablero Eléctrico

Con respecto a los guardamotores, hay 2 de marca ABL Sursum:

- MS10: Soporta una corriente nominal de corte i_e en el rango 6.3~10A y un voltaje máximo de 690V. Por sus características se utiliza para el compresor.
- MS4: Este guardamotor puede operar en el rango de corriente comprendido entre 2.5~10A y un voltaje máximo de 690V. Se utiliza en el motor AC.

En el caso de los automáticos se tienen 3 de ellos, para corrientes de 2, 6, y 10 Amperes. Además, todos son marca ABL Sursum y son del tipo Curva C.

2.3.3. PLC (Programmable Logic Controller)

Para realizar el control del STC se cuenta con un PLC marca *Siemens* de la familia S7-300, esta línea de PLCs modulares se caracteriza por un gran nivel de adaptación a cada aplicación, ya que presenta una gran variedad de módulos de entradas, salidas y de red.

El PLC consta de una CPU, una fuente de poder y un módulo de entradas y salidas digitales.

La fuente de poder corresponde a la *PS 307 2A*, la cual es una fuente estándar de alimentación, de 220 *Vac* a 24 *Vdc* entregando un máximo de 2A.

A continuación se describen los módulos del PLC:

- **CPU 315-2DP** La principal característica de interés de esta CPU es que posee un puerto de comunicación *Profibus DP*, de esta forma se logra comunicar el terminal de válvulas Festo con la unidad de control.

En la Tabla 2.1 se detallan las principales características técnicas de la CPU.

- **Módulo de entradas y salidas digitales SM 323**

El módulo *SM 323 DI16/DO16 24 Vcc 0.5 A* tiene las siguientes propiedades:

- 16 entradas, separadas galvánicamente en grupos de 16.
- 16 salidas, separadas galvánicamente en grupos de 8.
- Tensión nominal de entrada 24 *Vcc*.
- Entradas adecuadas para interruptores y detectores de proximidad de 2, 3 y 4 hilos.
- Salidas adecuadas para electroválvulas, contactores de corriente continua y lámparas de señalización.

En la figura 2.13 se presenta un esquema de conexiones para las entradas y salidas del módulo digital. En la parte izquierda de la figura se pueden apreciar las entradas, simbolizadas con contactos de entrada. Las salidas en cambio, se pueden ver al lado derecho de la imagen, separadas galvánicamente en dos grupos.

| ESPECIFICACIONES TÉCNICAS GENERALES | |
|--|------------------------------|
| Dimensiones y peso | |
| Dimensiones A x A x A (en mm) | 40 x 125 x 130 |
| Peso aprox. | 290 gr |
| Programación con Step 7 | |
| Soporta versión | Step 7 V5.1 + SP4 o superior |
| Tensión de alimentación | |
| Voltaje nominal | 24 Vdc |
| Límite inferior | 20.4 Vdc |
| Límite superior | 28.8 Vdc |
| Consumo de corriente | |
| Consumo en vacío | 60 mA |
| Consumo máximo | 800 mA |
| Memoria | |
| Integrada (no ampliable) | 128 KB |
| Slot para Memory Card | Sí |
| Tipo Memory Card | MMC (8 MB máx.) |
| Interfaz 1 | |
| Protocolo | MPI |
| Medio físico | RS485 (serial) |
| Alimentación | 15-30 Vdc |
| Consumo | 200 mA |
| Velocidad de Tx máx | 187.5 kBit/s |
| Interfaz 2 | |
| Protocolo | Profibus DP |
| Función | Maestro/Esclavo DP |
| Medio físico | RS485 (serial) |
| Alimentación | 15-30 Vdc |
| Consumo | 200 mA |
| Velocidad de Tx máx. | 2 MBit/s |

Tabla 2.1: Características técnicas de la CPU

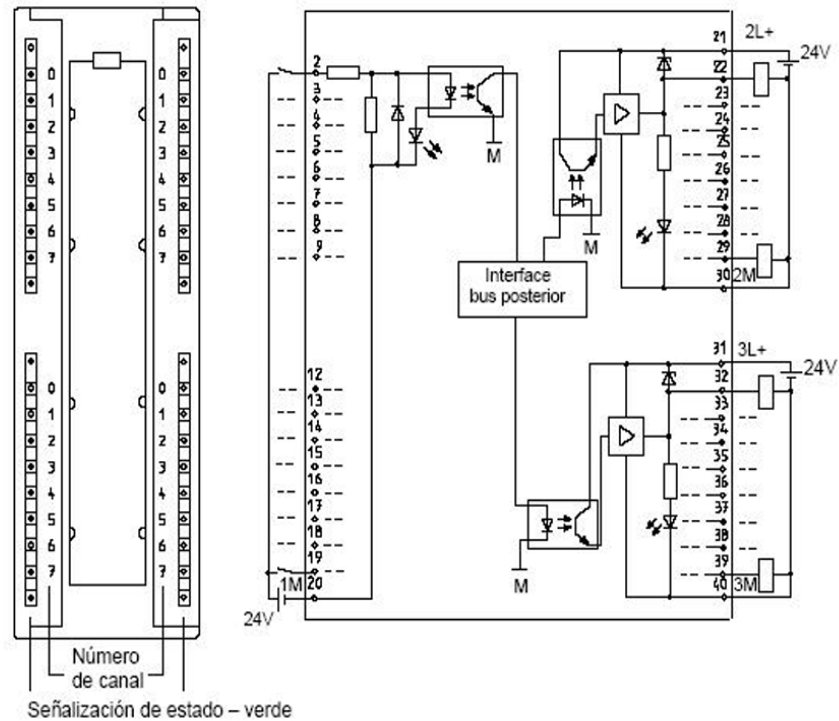


Figura 2.13: Esquema de conexiones de módulo de entradas y salidas digitales

La tabla 2.2 resume las principales características de interés para el uso del módulo.

- **Módulo de entradas y salidas análogas SM 334** El módulo *SM 334 4AI/2 AO, 12 bits* está instalado para aplicaciones futuras y tiene las siguientes propiedades:
 - 4 entradas, 2 con medición de tensión y 4 con medición de resistencia.
 - 2 salidas
 - Tensión nominal de entrada 24 Vcc.

En la figura 2.14 se presenta un esquema de conexiones para las entradas y salidas del módulo análogo.

| ESPECIFICACIONES TÉCNICAS GENERALES | |
|---|------------------------|
| Dimensiones y peso | |
| Dimensiones A x A x A (en mm) | 40 x 125 x 120 |
| Peso aprox. | 260 gr |
| Datos específicos del módulo | |
| Cantidad de entradas | 16 |
| Cantidad de salidas | 16 |
| Longitud del cable (sin pantalla) | máx. 600 m |
| Longitud del cable (con pantalla) | máx. 1000 m |
| Tensión de carga nominal (L+) | 24 Vdc |
| Potencia | 6.5 W |
| Señalización de estado | Un led verde por canal |
| Datos para la elección de un sensor | |
| Tensión nominal de entrada | 24 Vdc |
| Señal "1" | 13 a 30 Vdc |
| Señal "0" | -30 a 5 Vdc |
| Corriente de entrada (en "1") | 7 mA |
| Retardo en cambio de estado | 1.2 a 4.8 ms |
| Datos para la elección de un actuador | |
| Tensión de salida en "1" | 24 Vdc |
| Corriente de salida nominal en "1" | 0.5 A |
| Corriente de salida (límites) | 5 mA a 0.6A |
| Corriente de salida en "0" (residual) | máx. 0.5 mA |
| Carga de lámparas | máx. 5 W |
| Frecuencia de conmutación con carga resistiva | máx. 100 Hz |
| Frecuencia de conmutación con carga inductiva | máx. 0.5 Hz |
| Frecuencia de conmutación para lámparas | 10 Hz |
| Carga de lámparas | máx. 5 W |

Tabla 2.2: Características técnicas del módulo SM 323

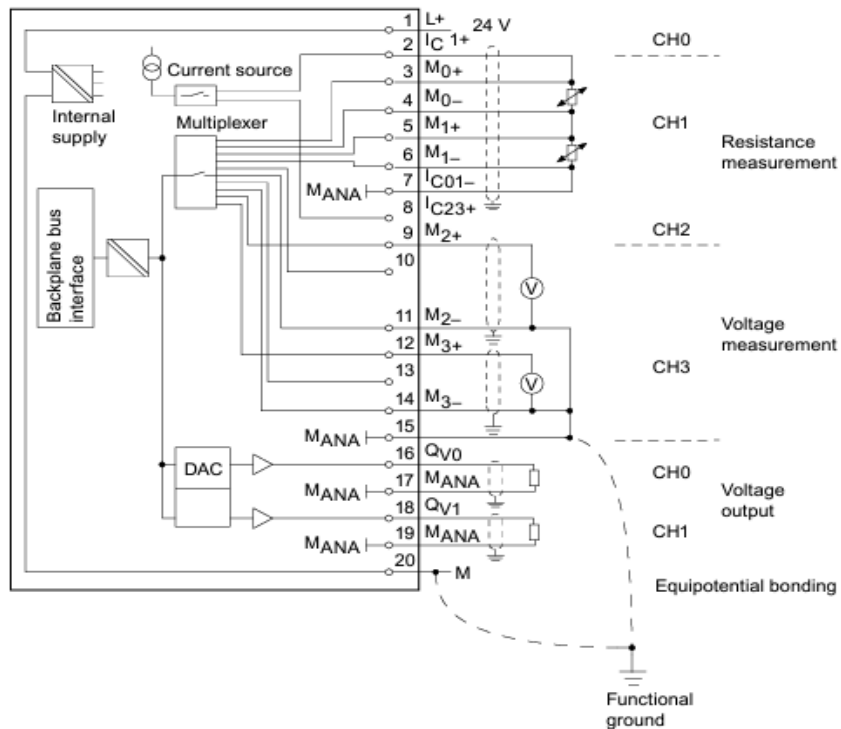


Figura 2.14: Esquema de conexiones de módulo entradas y salidas análogas

La tabla 2.3 resume las principales características de interés para el uso del módulo de entradas y salidas análogas.

En la figura 2.15 se pueden ver la fuente de alimentación, la CPU y los módulos de entradas y salidas digitales y análogas.

| ESPECIFICACIONES TÉCNICAS GENERALES | |
|---|------------------------|
| Dimensiones y peso | |
| Dimensiones A x A x A (en mm) | 40 x 125 x 117 |
| Peso aprox. | 200 gr |
| Datos específicos del módulo | |
| Cantidad de entradas | 4 |
| Cantidad de salidas | 2 |
| Longitud del cable (con pantalla) | máx. 100 m |
| Tensión de carga nominal (L+) | 24 Vdc |
| Potencia | 2 W |
| Señalización de estado | Un led verde por canal |
| Datos para entradas y salidas | |
| Tensión nominal de entrada | 20 Vdc |
| Rango de entrada tensiones 0-10 V | 100 k Ω |
| Resistencia de carga salida | 2.5 k Ω mín. |
| Formación de valor analógico para entradas | |
| Resolución con rango de rebase | 12 bit |
| Tiempo de integración parametrizable | Sí |
| Tiempo de integración (ms) | 16.67/20 ms |
| Formación de valor analógico para salidas | |
| Resolución con rango de rebase | 12 bit |
| Tiempo de conversión (por canal) | 500 μ |

Tabla 2.3: Características técnicas del módulo SM 334



Figura 2.15: PLC

2.3.4. Tarjeta PCI Siemens CP 5611

Se tiene una tarjeta PCI Siemens CP 5611 para la comunicación Profibus DP, en este caso para la comunicación entre el aparato de programación (PG) y el PLC (PC), a 12 [Mbaud] (máximo). La supervisión de elementos en la red Profibus se puede realizar a una velocidad de hasta 187.5 [kBaud]



Figura 2.16: Tarjeta PCI Siemens CP 5611

2.3.5. HMI (Human Machine Interface)

Se utiliza una HMI virtual para monitorear el sistema, permitiendo mostrar el estado del sistema a través de sensores, indicadores de luz. También es posible manipular los actuadores remotamente. El modelo de la HMI es la KTP700 Basic DP la cual cuenta con conectividad MPI/Profibus DP. En el laboratorio se cuenta con una HMI del mismo modelo, pero con conectividad Ethernet, por lo que no es compatible para utilizarse de forma física.

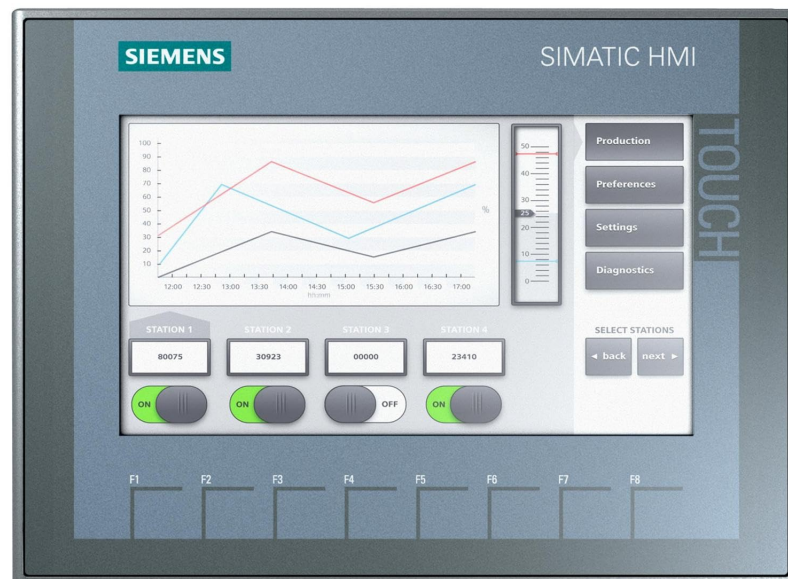


Figura 2.17: HMI KTP700 Basic DP

2.4. Diagrama de conexiones entre el STC, PLC y Terminal de Válvulas

Finalmente, una vez descrito los elementos del sistema de transporte y clasificación, junto con el sistema de control, el diagrama de conexiones entre el STC, el PLC y el Terminal de Válvulas físicos se puede ver en la figura 2.18

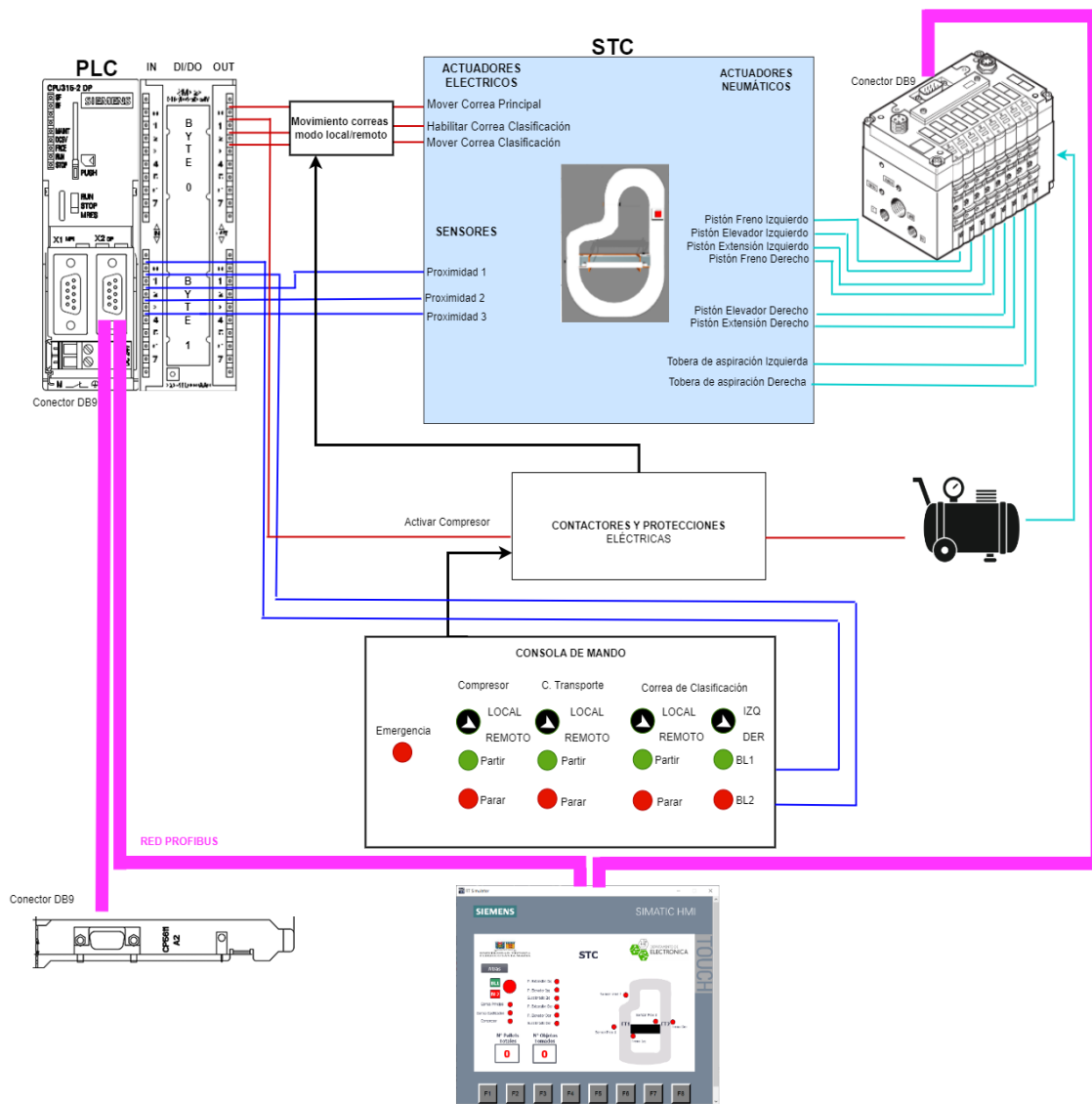


Figura 2.18: Diagrama de conexiones entre el STC, PLC y Terminal de Válvulas

2.5. Requisitos funcionales de automatización lógica del STC

Luego de conocer la descripción general del STC y sus componentes, es necesario establecer los requisitos funcionales para la automatización lógica:

- Detectar con anticipación si el pallet viene con carga o vacío por la correa principal
- Detener el pallet en la zona de transferencia de entrada si viene con carga, y desviar la carga hacia la correa de procesamiento
- Transportar la carga sobre la correa de procesamiento hasta la zona de transferencia de salida
- Volver a cargar el pallet con el objeto procesado para que continúe su desplazamiento por la correa principal
- Contar el número de pallets con carga y el número total de pallets que han circulado por la correa principal

3. Herramientas de desarrollo de esquema SIL

3.1. Ciros Studio

Es una potente plataforma de desarrollo probada en ámbito industrial que permite crear modelos de simulación en 3D para la técnica de automatización en tiempo real.

La simulación de transporte se utiliza para el diseño flexible de cualquier proceso de transporte, donde se tienen cintas de transportes con velocidad ajustables, pallets, objetos de trabajo y se pueden simular curvas de 90 y 180 grados, teniendo la opción de manipular sus posiciones y tamaños.

También permite la simulación de sensores, desde el sensor inductivo hasta la cámara, y que pueden ser replicados con sus características físicas. En este caso, el software tiene sensores ópticos, como los que tiene el STC físico.

Además en Ciros a estos componentes (objetos, sensores, actuadores) se les puede enlazar entradas y salidas fácilmente, las cuales se pueden controlar mediante distintas maneras, ya sea con un controlador que se encuentre en la misma simulación o con controladores externos, conectados a través de distintas interfaces, como por ejemplo OPC o también el conector de Festo EasyPort.

Permite la utilización de OPC, como servidor o cliente. En el caso de cliente OPC, presenta un menú de configuración para comunicación con cualquier número de servidores OPC para conexión de cualquier número de PLCs [10]. Para el servidor OPC, se configuran las variables que se quieren compartir, a través de los ajustes en el programa.

Se ahonda más sobre las características del programa en el capítulo 4.

3.2. TIA Portal

TIA Portal es un programa de Siemens que integra una gran cantidad de componentes, que permite, dentro de muchas opciones, programar controladores, programar pantallas HMI, realizar simulaciones de dispositivos, etc. En este caso, se utiliza para desarrollar la programación del PLC, tanto para la simulación (usando PLCSIM) como para el PLC físico, como también

para la simulación de una pantalla HMI.

Para este proyecto se utiliza TIA Portal v14.

3.2.1. PLCSIM

PLCSIM es un software de simulación de controladores programables desarrollado por Siemens. PLCSIM permite simular el funcionamiento de un PLC de Siemens en un entorno virtual, lo que es útil para probar y depurar programas de control antes de implementarlos en un entorno real. Algunas de las características y funciones típicas de PLCSIM incluyen:

- Emulación de hardware: PLCSIM simula el comportamiento de un PLC real, incluyendo sus entradas y salidas digitales y analógicas, así como los módulos de expansión.
- Interfaz de desarrollo (IDE): Proporciona una interfaz gráfica que permite editar y cargar los programas de control, monitorear variables y observar el comportamiento del PLC en tiempo real.
- Depuración: Permite depurar programas de control simulando diferentes escenarios y condiciones sin afectar el hardware real, lo que ayuda a encontrar y corregir errores de código.
- Pruebas de funcionamiento: Se utiliza PLCSIM para probar y validar programas de control antes de implementarlos en una máquina o proceso industrial, lo que reduce el riesgo de errores y paradas no planificadas.

Limitaciones: El entorno de simulación que presenta TIA Portal con PLCSIM es bastante bueno, sin embargo no es posible simular el terminal de válvulas a través de la red Profibus. Esto no implica que no funcione la simulación, si no que al simular el sistema a través de PLCSIM no distingue entre el PLC y el terminal de válvulas, si no que lo ve como un todo. Las variables que están ligadas al terminal si son simuladas, pero se integran a las variables del PLC.

3.2.2. PLC físico

Para la utilización del PLC físico se utiliza la misma configuración que se usa para el PLC simulado, salvo que al momento de cargar el programa se debe cambiar la interfaz PG/PC. De esto se habla y está mejor explicado en el capítulo 5. Además, en otros casos era necesario cambiar las direcciones de las entradas cuando se pasaba del modo simulación a implementación física y viceversa, en esta ocasión no es necesario realizar ese cambio.

3.3. Comunicación OPC en esquemas SIL

OPC, del acrónimo en inglés Open Platform Communications, es el estándar de interoperabilidad para el intercambio seguro y confiable de datos en el espacio de la automatización industrial y en otras industrias. Es independiente de la plataforma y garantiza un flujo fluido de información entre dispositivos de múltiples proveedores. La Fundación OPC es la responsable del desarrollo y mantenimiento de este estándar [11].

OPC se compone de una serie de especificaciones de estándares: OPC Data Access (DA) es el estándar más prolífico. OPC DA es un estándar de comunicación industrial ampliamente aceptado que permite el intercambio de datos entre dispositivos de múltiples proveedores y aplicaciones de control sin restricciones de propiedad. Un servidor OPC puede comunicar datos continuamente entre los PLC en el taller, las RTU en el campo, las estaciones HMI y las aplicaciones de software en las PC de escritorio. El cumplimiento de OPC hace posible la comunicación continua en tiempo real (incluso cuando el hardware y el software son de diferentes proveedores).

3.3.1. EzOPC para esquema SIL

EzOPC es un OPC Server de Festo que provee los drivers cliente OPC necesarios para cuatro tipos de conexiones de datos

- Para EasyPort desde Festo Didactic
- Para Step7 Simulator desde Siemens (PLCSIM)
- Para un controlador CoDeSys

- Para una simulación FluidSIM

Los datos pueden ser escritos o leídos por cualquier OPC compatible con OPC client después de la versión 2.

En el este caso de este proyecto, EzOPC permite la conexión entre TIA Portal y Ciros Studio, a través de PLCSIM.

En la Figura 3.19 se puede ver la conexión mencionada.

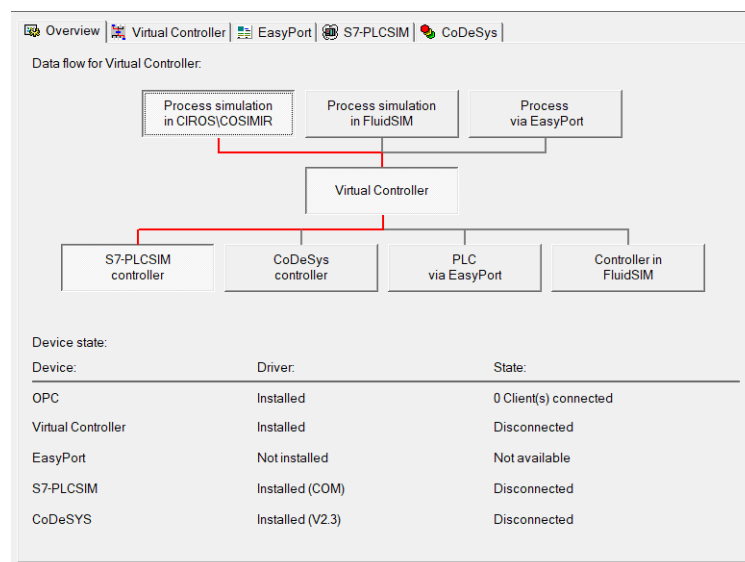


Figura 3.19: Conexión entre PLCSIM y Ciros Studio

3.4. STEP 7

STEP 7 (en TIA Portal) es el software de ingeniería para configurar la familia de controladores SIMATIC S7-1200, S7-1500, S7-300/400 y WinAC. Se nombran algunas características sobre STEP 7:

- **Programación de PLC:** STEP 7 se utiliza para crear programas de control para PLCs Siemens.
- **Lenguajes de Programación:** STEP 7 admite varios lenguajes de programación, como Ladder, S7-Graph, Texto Estructurado, Bloques de Diagrama de Funciones, entre otros.

- **Configuración de Hardware:** Además de la programación, STEP 7 se utiliza para configurar el hardware del PLC. Esto incluye la configuración de entradas y salidas digitales y analógicas, la configuración de redes de comunicación y la asignación de direcciones a los módulos del PLC.
- **Simulación:** STEP 7 también proporciona capacidades de simulación que permiten probar y depurar programas antes de cargarlos en un PLC real.
- **Diagnóstico:** La plataforma incluye herramientas de diagnóstico que ayudan a identificar y solucionar problemas en el PLC y en el sistema de control.
- **Compatibilidad:** STEP 7 es compatible con una amplia gama de PLCs Siemens, lo que lo hace versátil para diferentes aplicaciones y requisitos.

3.5. S7-Graph

3.5.1. Introducción

S7-Graph es el lenguaje de Siemens para implementar el diseño de automatismos secuenciales, que posteriormente son controlados por un PLC S7-300/400. La programación se lleva a cabo de manera gráfica mediante el uso de etapas y transiciones. En las etapas, se definen las acciones que deben ejecutarse en cada paso, mientras que las transiciones representan las condiciones que deben cumplirse para pasar de una etapa a otra.

Como se vió anteriormente, STEP 7 trabaja por medio de bloques, por lo que S7-Graph al ser parte de STEP 7 trabaja de la misma forma. De acuerdo a la complejidad del problema, el diagrama secuencial va a tomar una variada cantidad de bloques a utilizar, pero como mínimo, un control realizado con S7-Graph debe contener los siguientes bloques (Ver figura 3.20):

- Un FB de S7-Graph se describen las diferentes etapas y transiciones del control secuencial dentro de una o varias cadenas secuenciales.
- Y, un DB instancia, el cual contiene los datos y parámetros del control secuencial. Está asignado al bloque de función FB y puede ser generado por el sistema automáticamente.

- Y, para que el bloque de función FB pueda ejecutarse dentro del ciclo, debe ser llamado por un bloque lógico de nivel superior. Dicho bloque puede ser un bloque de organización (OB), una función (FC) u otro bloque de función (FB). El bloque de función FB se llama siempre como instancia individual.

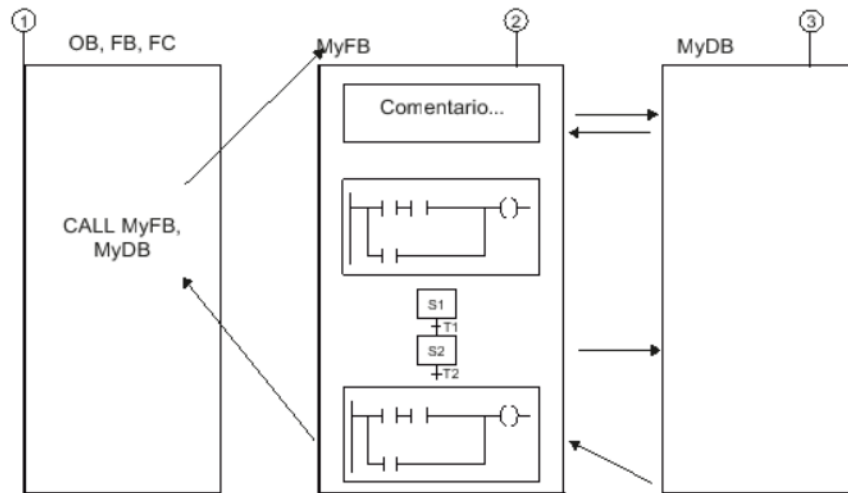


Figura 3.20: Bloques que componen un control en S7-Graph

3.5.2. Reglas de programación

- Una cadena secuencial está compuesta por una serie de etapas, las cuales se activan y desactivan dependiendo de las condiciones de transición programadas.
- Una cadena secuencial posee una o varias etapas iniciales. La cadena secuencial solo arranca una vez que se ha llamada a la etapa inicial. La etapa inicial es la etapa que se activa cuando se llama al FB que contiene el control, pero no necesariamente la etapa inicial es la primera etapa de la cadena.
- Una cadena secuencial termina con un fin de cadena, por lo que la secuencia llega a su fin o puede terminar con un salto a alguna etapa de la misma o de otra cadena secuencial, de esta forma se logra que la cadena funcione de manera cíclica.
- El control secuencial se divide en etapas y en cada una de estas etapas se programan las acciones que deben ejecutarse mientras la etapa está activa.

- Una etapa se activa si: se cumple la condición de transición que está justo antes de ella, cuando la etapa se ha definido como etapa inicial y se da inicio a la cadena, o cuando es llamada por una acción desde otra etapa.

3.5.3. Elementos de una cadena secuencial

Se presentan los diferentes elementos que se pueden encontrar al crear una cadena secuencial:

- **Etapa-Transición:** En una cadena secuencial solo se pueden insertar parejas de etapa-transición, no se pueden insertar por separado.

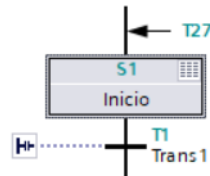


Figura 3.21: Diagrama de una etapa-transición

- **Etapa inicial:** Por media de la cual se activa la cadena.
- **Salto:** Un salto es el paso de una transición a una etapa cualquiera dentro de la misma cadena o de otra cadena del mismo FB. El inicio y la llegada del salto se representan gráficamente con una flecha (ver figura 3.22).

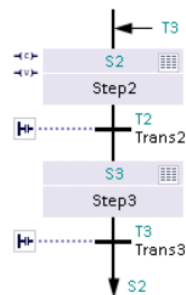


Figura 3.22: Ejemplo de un salto

- **Rama alternativa:** Las ramas alternativas están compuestas por varias ramas derivadas en paralelo, las cuales empiezan siempre con una transición. Solo se ejecutará aquella rama cuya transición sea la primera en activarse. Entonces, se puede decir que

las ramas alternativas cumplen la misma función que un OR lógico. Toda rama alternativa termina con una transición y se puede cerrar una etapa, un fin de cadena o un salto (ver figura 3.23).

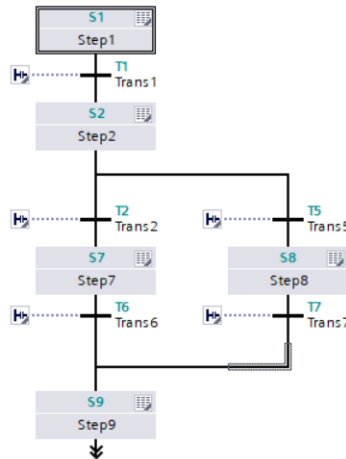


Figura 3.23: Ejemplo de una rama alternativa

- Rama simultánea:** Las ramas simultáneas se componen de varias ramas paralelas, las cuales siempre empiezan con una etapa. Las ramas derivadas se ejecutan simultáneamente. Por lo tanto, estas ramas cumplen un AND lógico. Toda rama simultánea termina con una etapa y puede cerrarse con una transición. Si todas las ramas simultáneas terminan con la misma transición, ésta no activará la próxima etapa hasta que no se hayan terminado de ejecutar todas las ramas activas (ver figura 3.24).

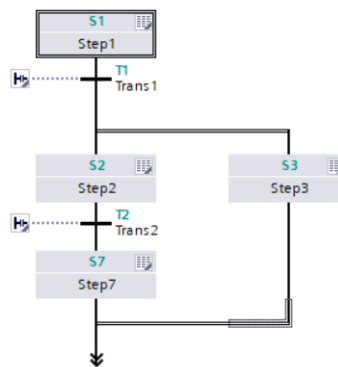


Figura 3.24: Ejemplo de una rama simultánea

- Fin de cadena:** Un fin de cadena al final de una cadena secuencial termina esta cadena sin volver a ejecutarla cíclicamente. Un fin de cadena al final de una rama derivada de una rama simultánea finaliza solo esa rama derivada. Las otras ramas simultáneas derivadas se continuarán procesando. Los fines de cadena siempre se encuentran después de una transición (ver figura 3.25).

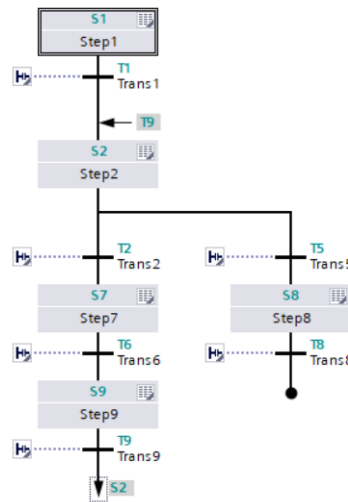


Figura 3.25: Ejemplo de fin de cadena

- Instrucciones permanentes:** Las instrucciones permanentes ofrecen la posibilidad de crear un código del programa que se ejecute antes o después del procesamiento de la cadena secuencial. Esto permite programar condiciones y llamadas de bloque que se procesen en cada ciclo con independencia de la cadena secuencial.

3.5.4. Acciones

Las acciones son programadas en las etapas, y pueden manipular entradas, salidas, variables de memoria, activar o desactivar etapas o llamar a bloques. Por lo tanto, las acciones son quienes controlan el proceso. Las acciones se ejecutan desde arriba hacia abajo una vez que se activa la etapa a la que pertenecen. Cabe destacar que se pueden crear etapas vacías, es decir, etapas sin acciones. En la figura 3.26 se puede ver una acción típica.

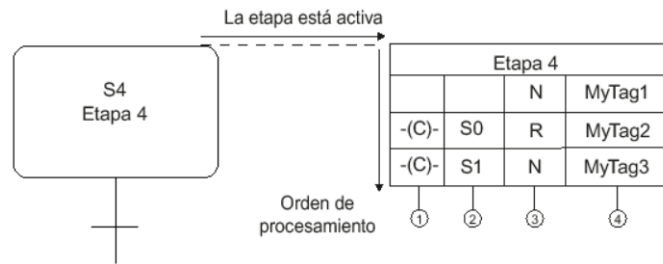


Figura 3.26: Etapa con sus acciones que debe realizar

Las acciones están compuestas por:

- **interlock (1):** es posible enlazar una acción con un interlock para influir en su ejecución.
- **un evento (opcional) (2):** un evento, por ejemplo puede ser el cambio de estado de alguna señal de interlock.
- **una operación (3):** una operación puede ser llevar a 1 (SET), RESET, ejecutar un retardo, etc.
- **y un operando o asignación (4):** es la señal de salida o interna del programa (variables de memoria) sobre la cual se ejecuta la acción.

Las acciones se clasifican en:

- **Acciones estándar:** Las acciones estándar se pueden combinar con un interlock (o enclavamiento), de esta forma, las acciones solo se ejecutarán cuando se cumplan las condiciones de interlock (además, la etapa debe estar activa), en cambio en una etapa estándar sin interlock las acciones se ejecutan una vez la etapa esté activa.
- **Acciones controladas por evento:** Las acciones se pueden combinar con eventos. Un evento es el cambio del estado de señal de una etapa, de una supervisión de etapa o de un interlock de etapa, o el acuse de un mensaje o la entrada de un registro. Cuando se combina una acción con un evento, el estado de la señal del evento se detecta por evaluación de flancos. Esto significa que las operaciones solo se pueden ejecutar en el ciclo en el que aparece (o desaparece) el evento. Cabe destacar que las acciones controladas

por eventos también pueden combinarse con interlocks, al igual que las acciones estándar. Una diferencia que poseen, es que las etapas controladas por eventos, pueden activar o desactivar a otras etapas.

- **Contadores, temporizadores y acciones varias:**

- **Contadores en acciones:** Los contadores requieren siempre un evento para determinar el instante de activación del contador. Esto significa que el contador se activará cuando se produzca el evento correspondiente. Las acciones para las que se utilizan los eventos pueden enlazarse adicionalmente con un interlock. De este modo se ejecutan únicamente si las condiciones del interlock se cumplen.
- **Temporizadores en acciones:** Al igual que los contadores, todos los temporizadores en acciones dependen siempre de un evento, es decir, el evento determina el punto de activación y/o desactivación de la operación. Opcionalmente, los temporizadores pueden combinarse con un interlock.
- **Operaciones varias:** Como la mayoría de los lenguajes de programación, S7-Graph permite la utilización de variadas operaciones, tales como operaciones lógicas entre variables, operaciones de asignación, de conversión, operaciones matemáticas, etc.

3.5.5. Condiciones

Las condiciones son estados binarios del proceso, las cuales mediante elementos KOP (esquema de contactos) o FUP (esquema lógico) entregan un resultado lógico. Este resultado lógico puede afectar a una o varias acciones de una etapa, a toda la etapa, al paso de una etapa a la siguiente o a toda la cadena.

Las condiciones pueden ser utilizadas en:

- **Una transición:** En las transiciones se programan las condiciones que controlan el paso de una etapa a la siguiente. La transición activa la siguiente etapa de la cadena secuencial cuando se cumplen las condiciones; es decir, cuando el resultado del segmento es 1, se activa la etapa siguiente a la transición. La transición no activa la siguiente etapa de la

cadena secuencial si no se cumplen las condiciones; es decir, si el resultado del segmento es 0, la etapa activa permanece activa.

- **Un interlock:** Los interlocks son condiciones que se programan para enclavar etapas y que afectan a la ejecución de las distintas acciones. Si se cumple la combinación de las condiciones, se ejecutan las acciones combinadas con interlocks.
- **Una supervisión:** La supervisión es una condición que se programa para supervisar etapas y que afecta al paso de una etapa a otra.
- **Instrucción permanente:** son condiciones que se ejecutan una vez por ciclo independiente del estado de la cadena.

4. Simulación STC en Ciros Studio

4.1. Aplicación Ciros

En la sección 3 ya se realizó una descripción general de la aplicación, en este apartado se muestran los componentes que presenta en sus librerías, como se organiza el modelo y algunas configuraciones que son importantes tener en cuenta.

4.1.1. Librerías

Las bibliotecas de modelos de CIROS proporcionan un enorme conjunto de elementos modelados listos para usar que se pueden agregar fácilmente a su modelo simplemente seleccionándolos en el cuadro de diálogo Model Libraries y presionando el botón Add. Para abrir el cuadro de diálogo Model Libraries, debe seleccionar **Modeling -> Model Libraries**. En la figura 4.27 se puede ver la ventana de las librerías.

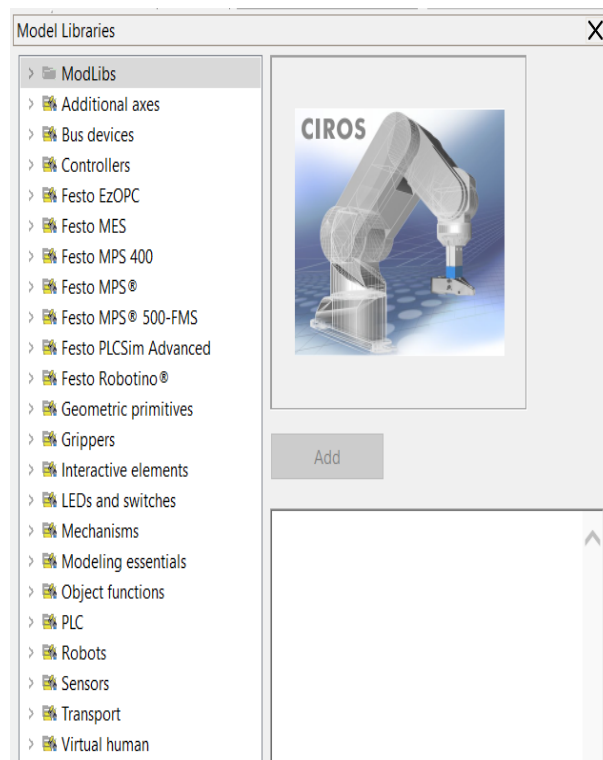
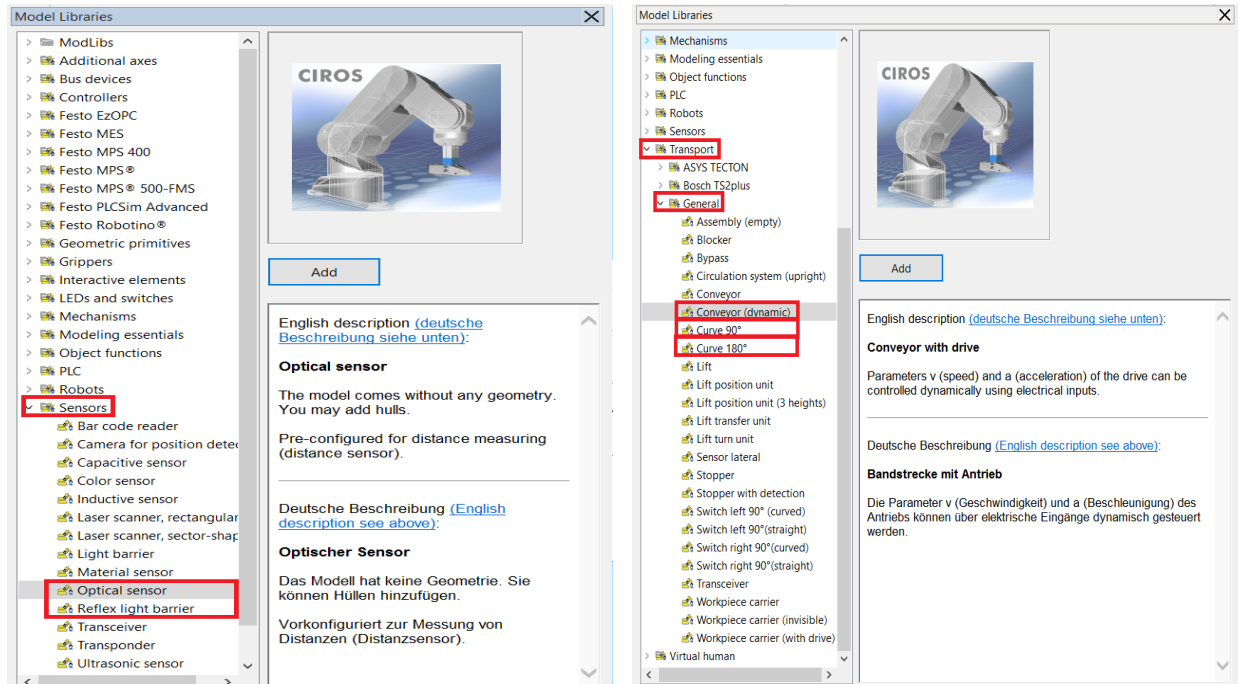


Figura 4.27: Librerías

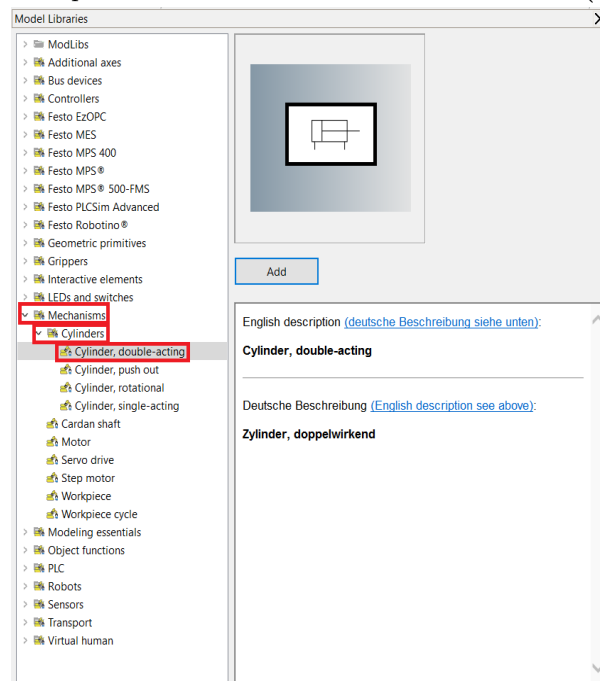
Y en la figura 4.28 se pueden ver ejemplo de los modelos que pueden ser agregados y que se

utilizan en el proyecto, como lo son el sensor óptico, la correa y el cilindro de doble efecto. Al costado derecho de cada imagen aparece una breve descripción de cada uno de los elementos.



(a) Sensor óptico

(b) Correas



(c) Cilindro de doble efecto

Figura 4.28: Ejemplo de modelos utilizados en el proyecto

4.1.2. Explorador del modelo

El explorador del modelo (Model Explorer en la aplicación) permite acceder a todos los elementos de un modelo. El acceso es posible para objetos, elementos subordinados, materiales, librerías, luces y conexiones de entradas y salidas.

La ventana del explorador del modelo está dividida en dos partes, la parte izquierda contiene el árbol de navegación con carpetas de cada elemento del modelo. La lista de elementos (element list en Ciro) en la parte derecha contiene los elementos de la carpeta seleccionada en el árbol de navegación. Para acceder se debe ir a **Modeling -> Model Explorer**.

En la figura 4.29 se puede ver la ventana abierta del explorador del modelo, donde en el costado izquierdo se encuentra el árbol del modelo y en este caso en la parte izquierda hay información de la salida de uno de los sensores.

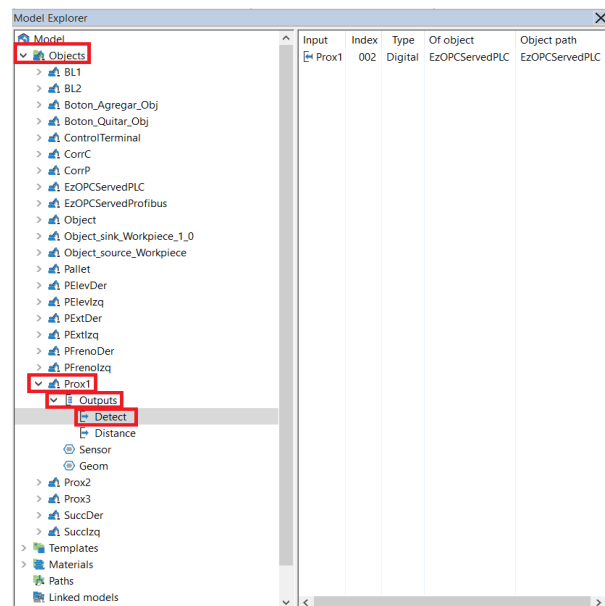


Figura 4.29: Explorador del modelo

También es posible ver las propiedades de cada elemento, las cuales se pueden modificar, para ello se debe hacer click derecho sobre el elemento que se quiere ver las propiedades en el árbol de navegación. Las propiedades que se muestran dependen de los elementos seleccionados. Las principales propiedades que son usadas son:

- **Pose:** Aquí se puede cambiar la posición y orientación de la selección actual, donde puede ser uno o más elementos. La figura 4.30a muestra esta propiedad.
- **Dimension:** Aquí se puede cambiar la dimensión de los elementos, ya sea ancho, largo o alto.
- **Visualization:** Aquí se pueden cambiar propiedades del elemento tales como el color y el material.
- **Sensor:** Esta propiedad se muestra cuando el elemento es un sensor. Aquí se puede cambiar por ejemplo el rango de medición. Lo muestra la figura 4.30b.

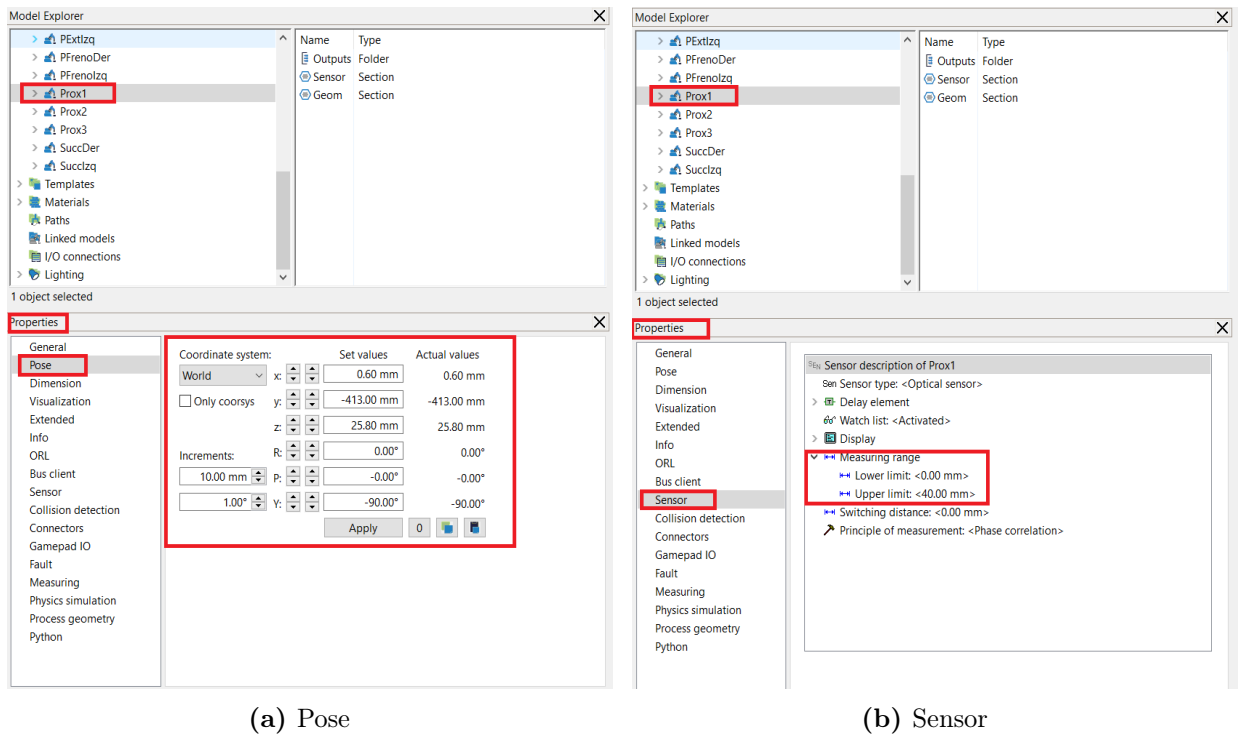
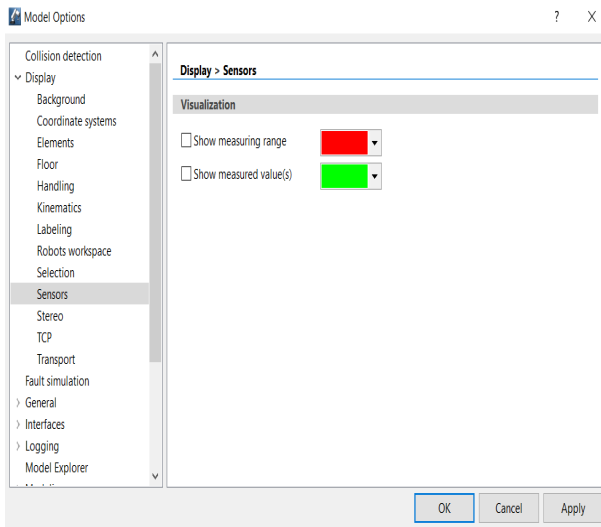


Figura 4.30: Ejemplo propiedades

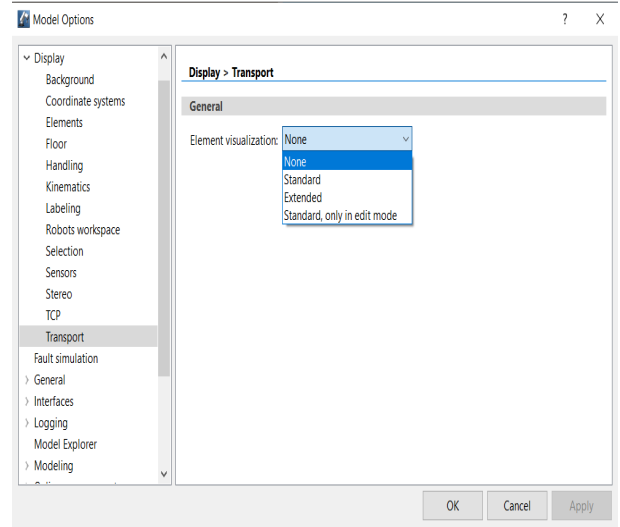
4.1.3. Opciones del modelo

En **Settings -> Model options** encontramos una ventana para poder hacer ajustes. Por ejemplo, para la simulación de sensores podemos seleccionar si queremos ver el rango de estos o no, y que la simulación muestre cuando estos están activos, como se ve en la figura 4.31a.

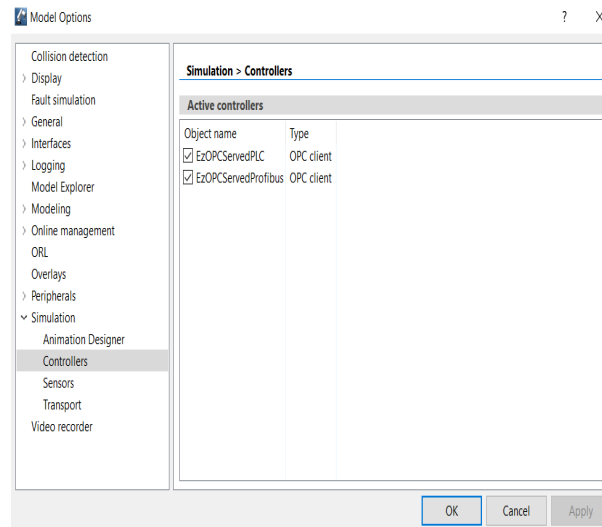
También para la simulación de transporte permite escoger ver o no cuando estén activas (standard) o incluso mayor información como los vectores de velocidad (extend), como se muestra en la figura 4.31b. También se pueden hacer ajustes con respecto al ambiente, como lo es el piso (floor) y el fondo (background). En la parte de simulación es posible seleccionar los controladores que se quieren utilizar en el modelo.



(a) Ajustes Sensores



(b) Ajustes Transporte



(c) Controladores

Figura 4.31: Ejemplo ajustes del modelo

4.2.2. Correa de Clasificación

Ciros entrega una librería con correas ya creadas. En este caso como la correa de clasificación es recta, se utiliza una correa que ya tiene geometría. Es importante recalcar que aunque ésta correa ya está creada, es posible modificar su dimensiones (largo, ancho, alto), también su visualización (colores principalmente) y parámetros como la velocidad. En la figura 4.33 se aprecia la Correa que se utiliza para la simulación.

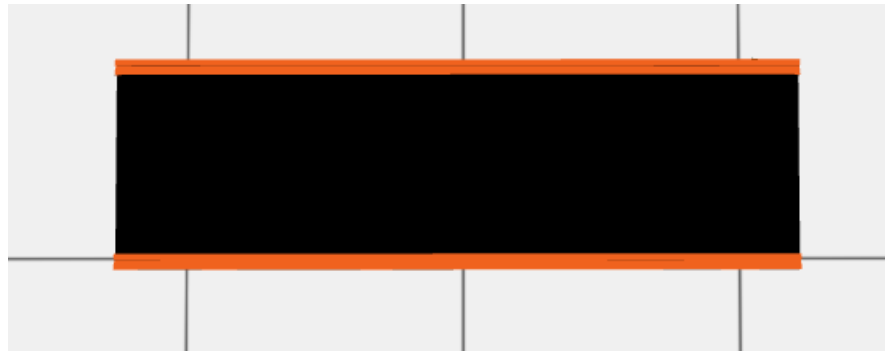
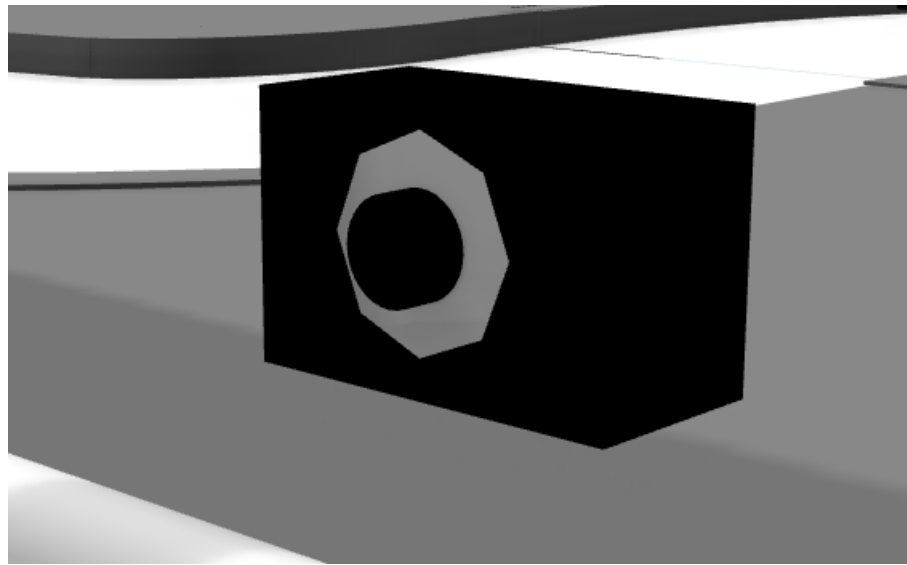


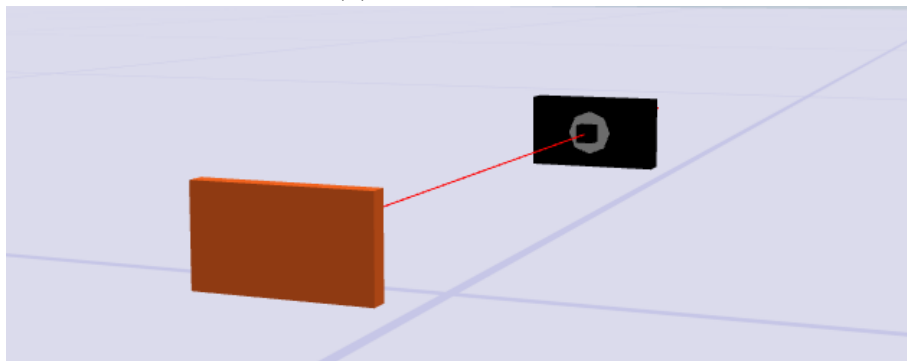
Figura 4.33: Correa Clasificación

4.2.3. Sensores ópticos

Ciros también presenta una variada librería para uso de sensores de proximidad. En este caso se utilizan tres sensores ópticos, dos de ellos difuso-reflectivo y uno retro-reflectivo. Todos los sensores, al igual que las correas utilizadas en la correa principal, no presentan geometría, si no que uno debe crearlas. Lo importante del uso de los sensores, es que uno puede manipular sus dimensiones, como también los rangos de medición, dando mayores posibilidades de adaptar el sistema. Al detectar un objeto, los sensores cambian el estado de la salida, pasando de 0 a 1. En la figura 4.34 se aprecian los sensores que se utilizan, con las geometrías ya creadas, que son semejantes al del sistema físico.



(a) Difuso-reflectivo



(b) Retro-reflectivo

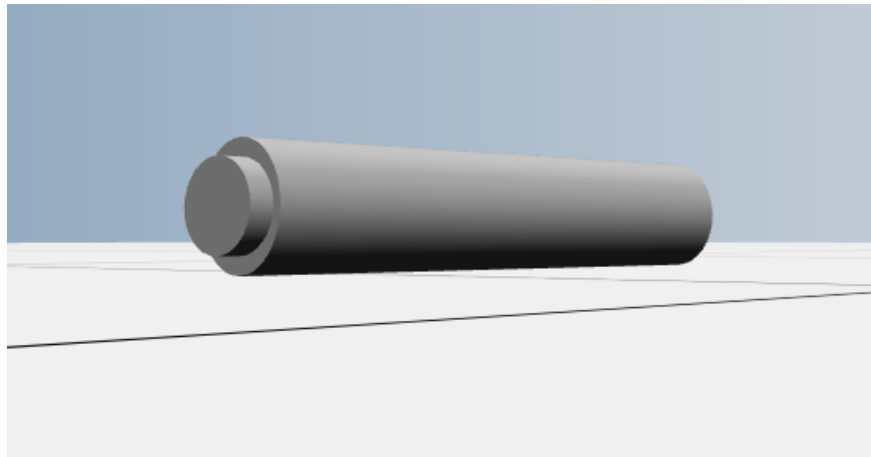
Figura 4.34: Sensores de proximidad en Ciro

4.2.4. Cilindros de doble-efecto

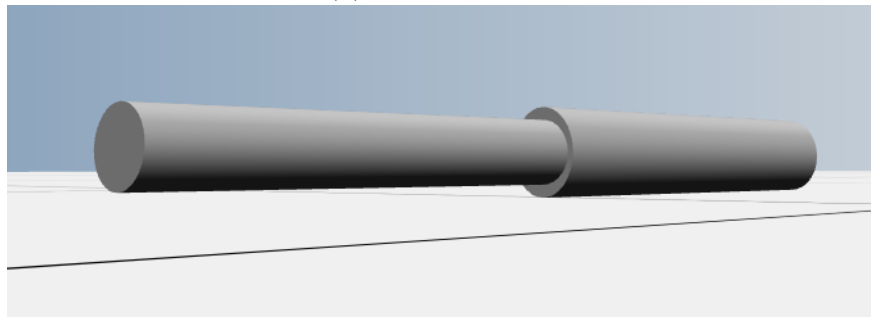
Para la simulación de los frenos, extensión y elevación, se utilizan cilindros de doble efecto. Estos cilindros existen en la librerías de Ciro, pero al igual que en otros objetos no presentan geometría, así que es necesario crearlas. Estos presentan entrada para extender y retraer, como salidas que indican en que posición se encuentra. También es posible modificar/ajustar la extensión del cilindro.

Para la realización de la simulación se utilizan 6 cilindros de doble efecto, al igual que en el sistema físico.

- **Frenos:** 2 de los cilindros se utilizan como frenos. En la figura 4.35 se puede ver uno de ellos, cuando está retraído y extendido.



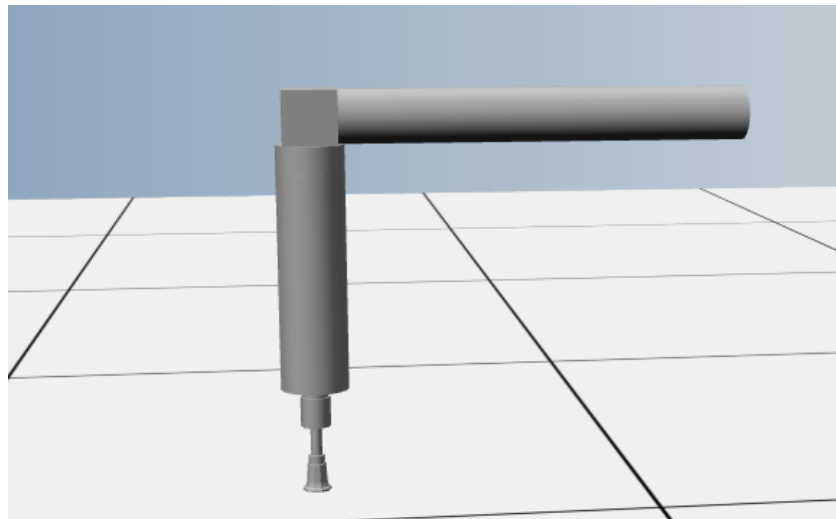
(a) Freno retraído



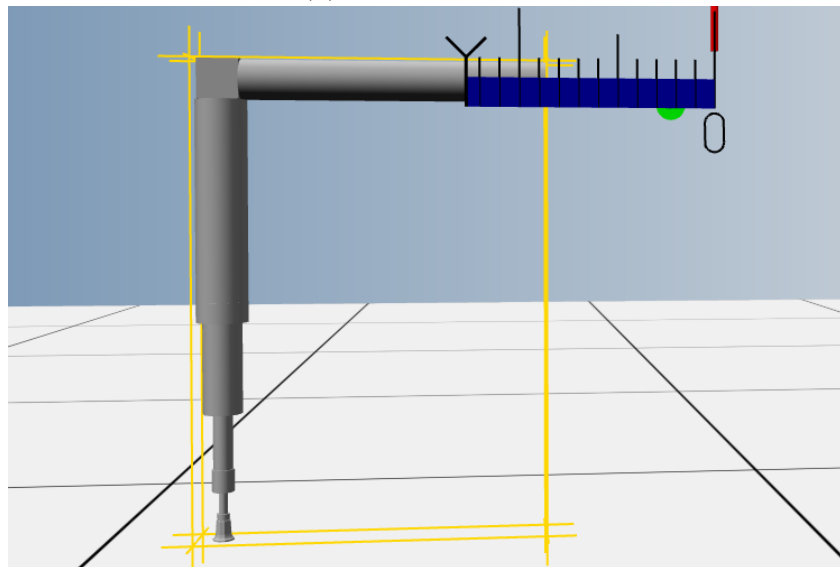
(b) Freno extendido

Figura 4.35: Frenos

- **Extensión y Elevación** Los 4 cilindros que quedan se utilizan para las estaciones de trabajo, esto es en la tarea de mover el objeto desde la correa principal a la de clasificación y viceversa. En la figura 4.36 se aprecia los cilindros que simulan la extensión y elevación en un mismo sistema, cuando están retraídos y extendidos. La regla en la figura 4.36b está para explicitar que se ha extendido el cilindro.



(a) Sistema retraído



(b) Sistema extendido

Figura 4.36: Extensión y Elevación

4.2.5. Toberas de aspiración

En este caso, las toberas de aspiración se utilizan directamente de las librerías, con la geometría ya creadas. Estos cumplen muy bien la función de tomar un objeto, pero siempre y cuando que al ser activado, el objeto debe estar muy cerca, en caso contrario el programa arrojará error. Esto es importante al momento de realizar la automatización para el sistema virtual, ya que es una limitación de la simulación, no así del sistema físico. En la figura 4.37 se aprecia la

tobera de aspiración que se utiliza.

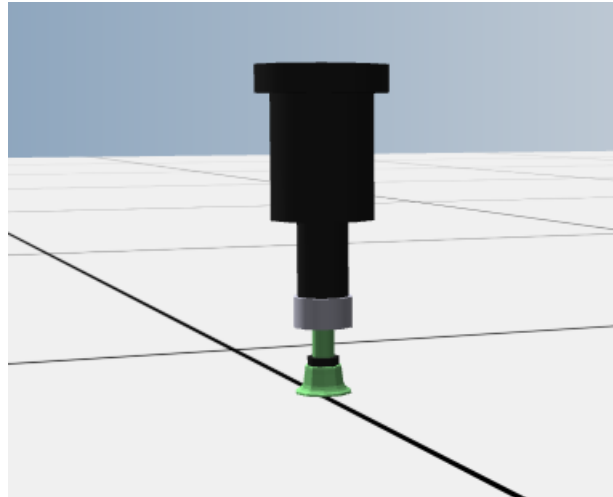


Figura 4.37: Tobera de aspiración

4.2.6. Pallet y Objeto

También se utiliza un pallet y un objeto dinámico, este último es el que emula una pieza que será clasificada. El objeto es dinámico debido a que puede ser quitado del pallet, lo cual es razonable si queremos realizar la acción de mover desde la correa principal hacia la correa de clasificación. En la figura 4.38 se aprecia el pallet junto al objeto encima.

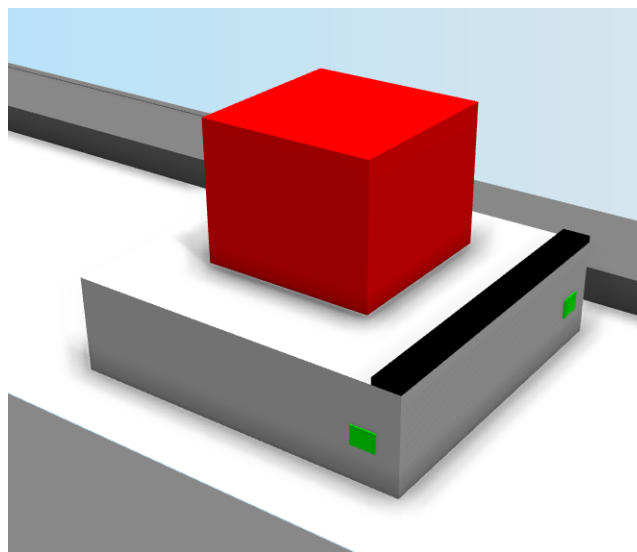


Figura 4.38: Pallet y Objeto

4.3. Ambiente de simulación 3D

Una vez creados los subsistemas por separado, se procede a la unificación de estos para crear uno solo. Además de agregar las geometrías a aquellos objetos que no la tenían, también se adaptó el sistema para que tenga un parecido razonable con el sistema físico. El resultado final del ambiente de simulación para el STC se puede ver en las figuras 4.39, 4.40 y 4.41.

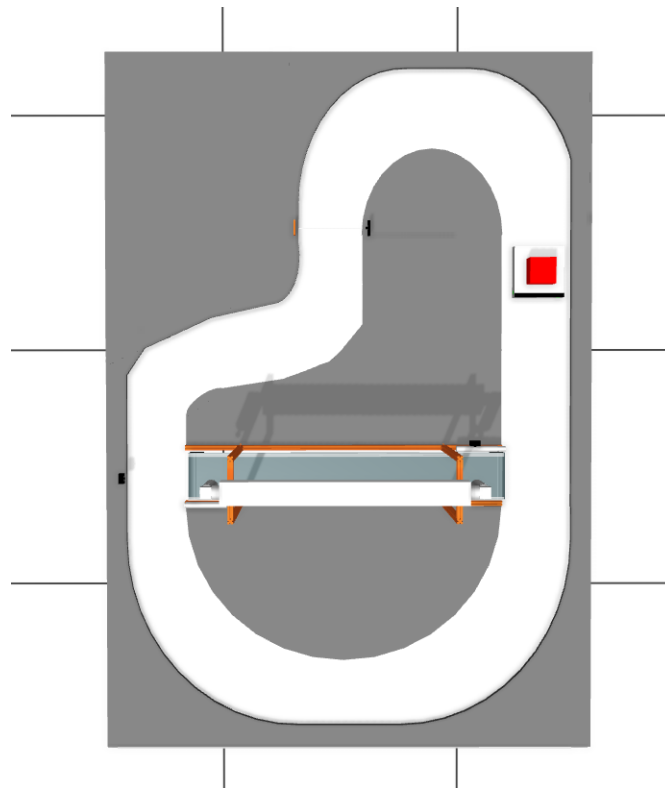


Figura 4.39: Vista aérea STC simulado

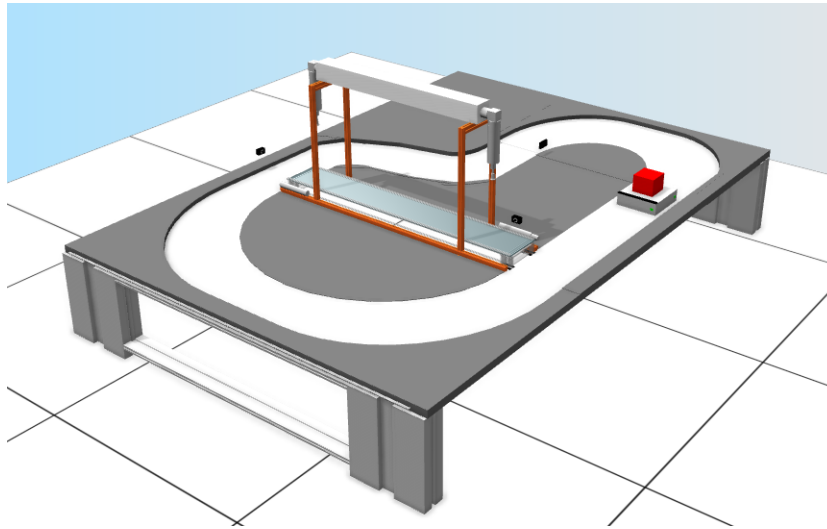


Figura 4.40: Vista desde el costado STC simulado

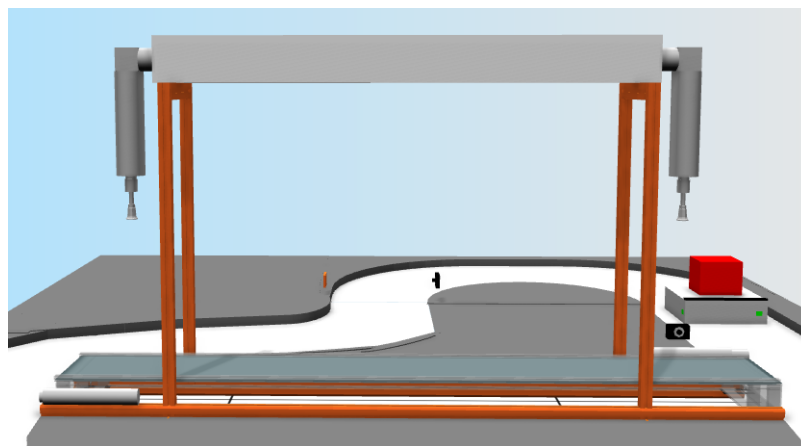


Figura 4.41: Vista estaciones del STC

Es importante señalar que no se simularon los motores y el compresor, esto debido a que las señales de activación, que en el sistema físico van directamente conectados a estos, en la simulación se puede pasar la señal de encendido a las correas en caso de los motores y, en el caso del compresor, no es necesario una fuente neumática para efecto de simular los cilindros y succionadores. Esto no afecta en nada al funcionamiento de la automatización.

También a la simulación se le agregaron unos botones, los cuales permiten interactuar con el sistema:

- **BL1:** Botón Libre 1 que permite iniciar el programa de automatización.
- **BL2:** Botón Libre 2 que permite detener el programa de automatización.
- **A.O:** Este botón permite agregar/colocar el objeto arriba del pallet.
- **Q.O:** Botón para poder retirar el objeto desde el pallet (solo si está encima del pallet).

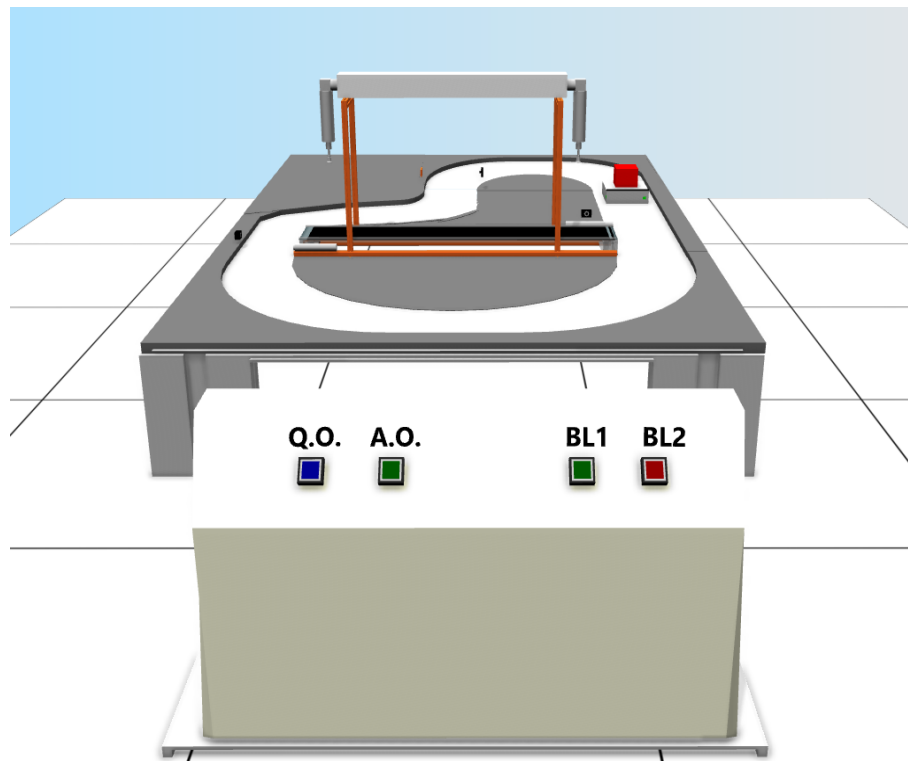


Figura 4.42: Tablero con botones.

4.4. Dinámica del sistema

Para que el sistema sea dinámico, hay ciertos elementos que tiene Ciro que lo permiten (aparte de las características que presenta cada elemento, como la velocidad de las correas por ejemplo) y se detallan a continuación:

4.4.1. Grippers

Existen diferentes formas para que otros objetos "sujeten" a otros. Estos métodos tienen diferentes prioridades. Si un objeto se puede "sujetar" de varias maneras al mismo tiempo,

siempre prevalecerá el método "más fuerte". Así, por ejemplo, un punto de agarre (Gripper Point) puede retirar un objeto que está sostenido por una superficie activa (Active Surface).

- **Puntos de Agarre** Los puntos de agarre son importantes para esta simulación.

El pallet está enlazado a la correa principal a través de este método, lo que le permite seguir su trayectoria al moverse. La figura 4.43 muestra como está enlazado el pallet en Ciro.

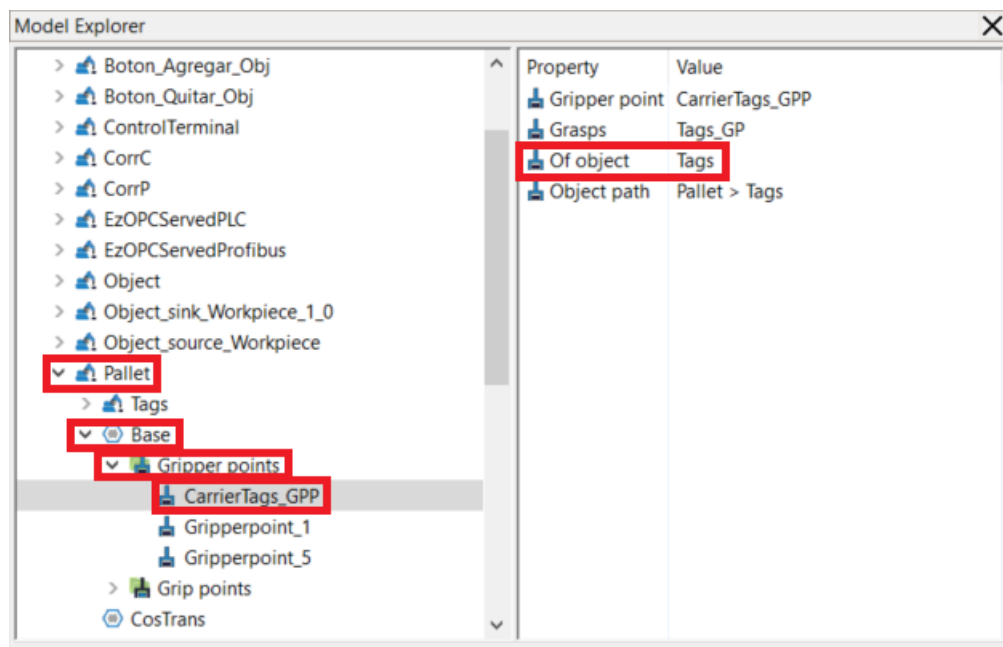


Figura 4.43: Punto de agarre del pallet a la correa principal

- **Superficie Activa**

Una superficie activa puede agarrar y mover múltiples puntos de agarre al mismo tiempo. Esto se utiliza para poder "sujetar" el objeto de trabajo en el pallet mientras este se está moviendo. Esto se puede ver en la figura 4.44

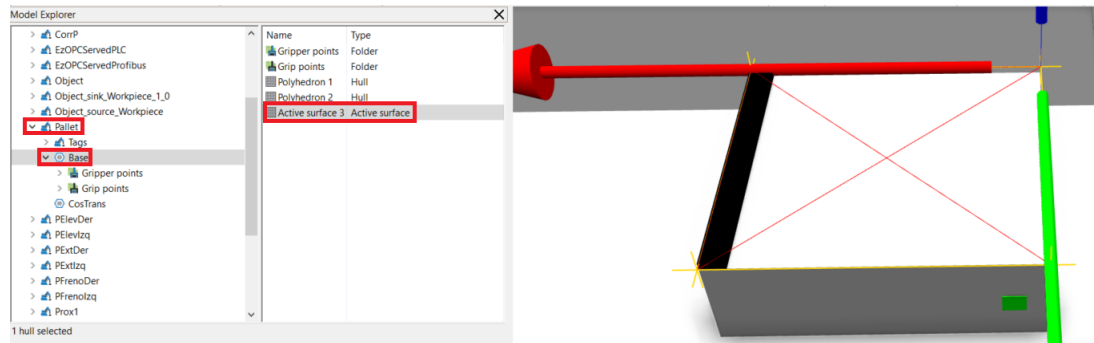


Figura 4.44: Superficie Activa en el pallet

También la correa de clasificación presenta una superficie activa, esto para poder poner el objeto encima de ella y que pueda moverlo desde ET1 a ET2.

Es posible retirar el objeto de trabajo, desde las superficies activas ,a través de las toberas de aspiración, debido a que los puntos de agarre de las toberas de aspiración tienen mayor prioridad.

4.4.2. Agregar y quitar objetos

Para poder simular que el pallet lleva carga o no, es necesario interactuar con el sistema haciendo aparecer o quitar el objeto de trabajo. Para realizar esto último, es necesario agregar al modelo lo siguiente: Object Sink y Object Source.

- **Object Source**

El mecanismo Object Source proporciona la capacidad de generar objetos basados en plantillas (templates) durante una ejecución de simulación.

El Object Source se puede configurar cambiando sus propiedades extendidas (figura 4.45). En este caso se configura que el objeto que genere sea el objeto de la figura 4.38, template que fue generado anteriormente a partir del elemento workpiece en las librerías.

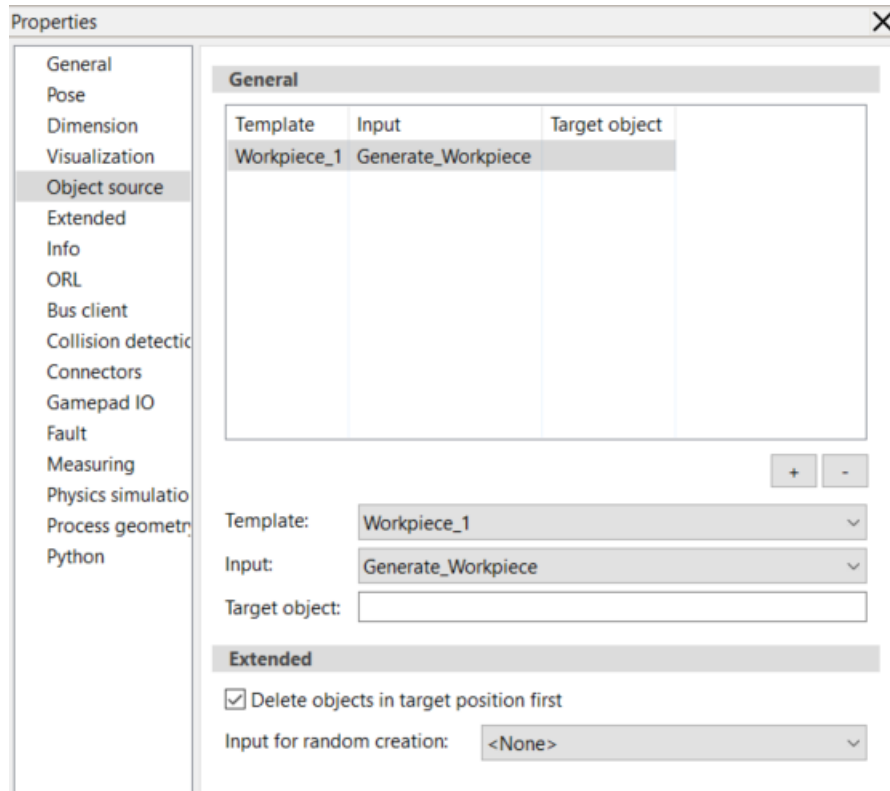


Figura 4.45: Configuración Object Source

La ubicación de Object Source es puesto en el centro del pallet, para que al ser generado el objeto se ubique encima de este. Además, el pallet tiene un punto de agarre a Object Source.

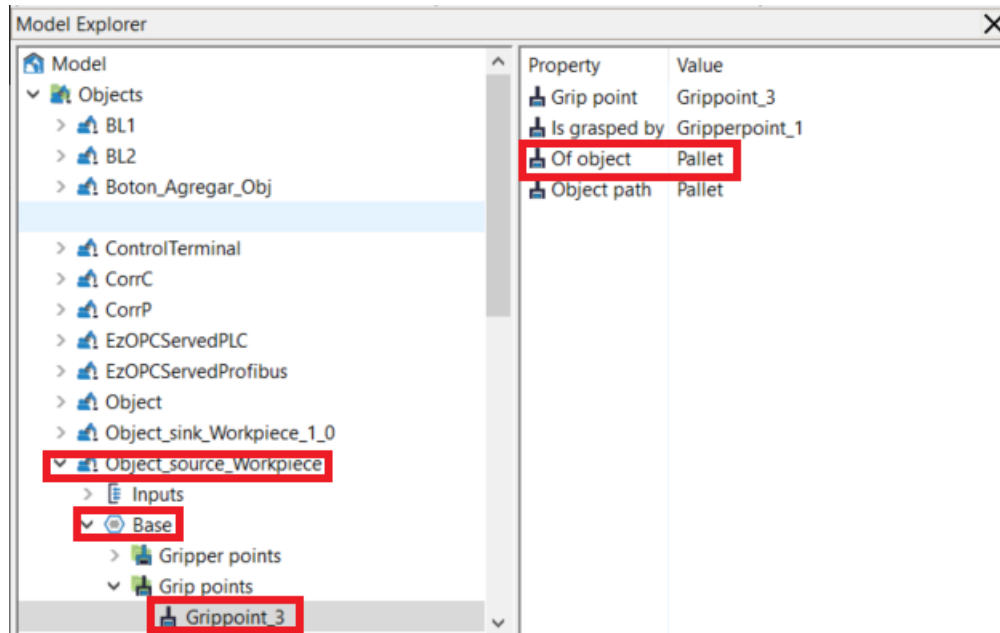


Figura 4.46: Punto de agarre del pallet a Object Source

Para agregar el objeto, como se vió anteriormente, se utiliza el botón A.O..Este botón está enlazado a la entrada de Object Source, por lo que al presionarlo se crea el objeto.

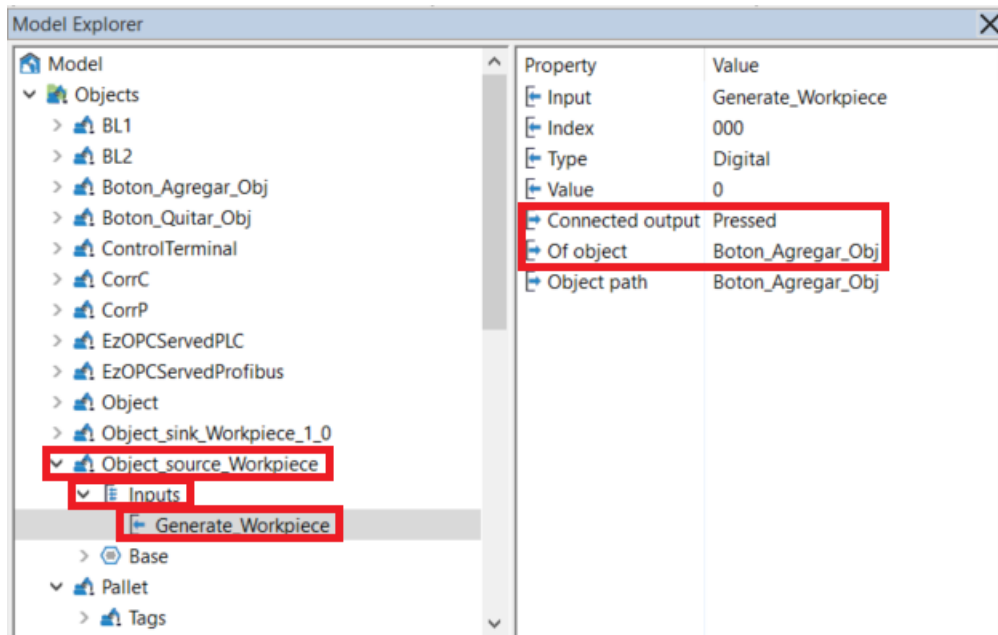


Figura 4.47: Botón A.O. para generar objeto

- **Object Sink**

Object Sink es la contraparte de Object Source. Se utiliza este mecanismo para eliminar objetos en el tiempo de ejecución. Para que el objeto sea eliminado, el objeto en cuestión debe estar enlazado a un punto de agarre. Entonces, al momento de activar la entrada del Object Sink (0 a 1) se elimina el objeto.

En la simulación del STC, se utiliza el botón Q.O. para eliminar el objeto, el cual está enlazado a la entrada de Object Sink, que también esta ubicado en el centro del pallet y tiene un punto de agarre en este.

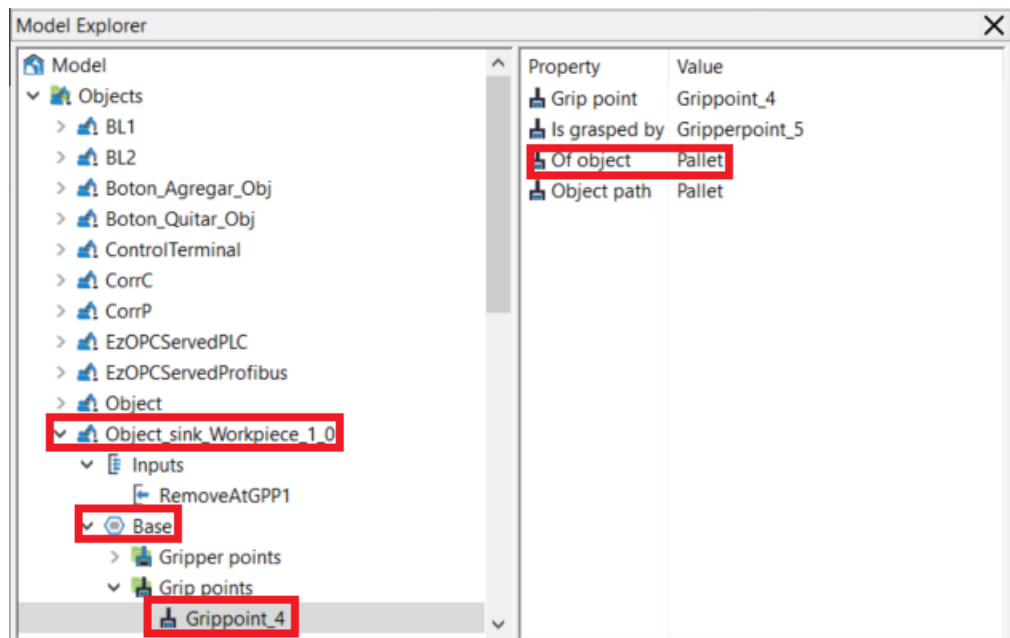


Figura 4.48: Punto de agarre del pallet a Object Sink

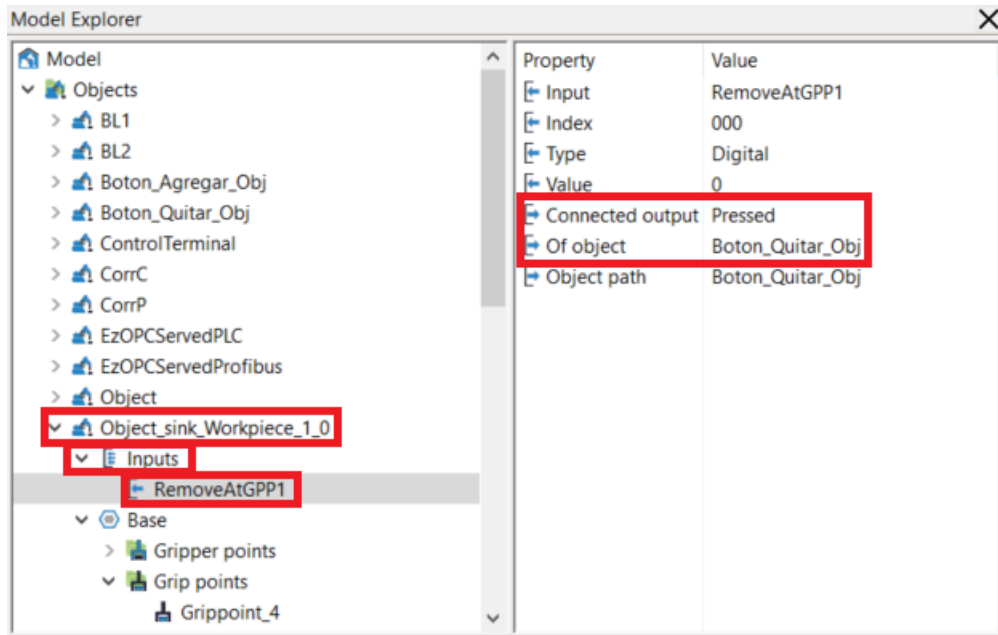


Figura 4.49: Botón Q.O. para quitar objeto

5. Desarrollo e Implementación

Ya que se conocen las herramientas de desarrollo para el esquema SIL, se procede a realizar la automatización del sistema, precisamente la configuración de hardware y la programación del PLC para el control del STC. Luego, se implementa esta automatización en el STC virtual y el STC físico.

5.1. Configuración de Hardware

Al trabajar en un proyecto de automatización en TIA Portal, es necesario configurar en el software los elementos físicos a utilizar para realizar el control, ya sea si es simulado o no, es decir, se debe establecer el modelo de CPU a utilizar, los módulos adicionales de entradas y salidas, las interfaces de comunicación, la periferia descentralizada (Profibus DP), entre otros.

Una vez que se ha creado el proyecto, entonces se deben agregar los dispositivos. Para la CPU 315-2DP se debe agregar desde **Agregar dispositivo**, en el árbol del proyecto. Los demás dispositivos se agregan desde el **Catálogo de hardware**, estos son la tarjeta de comunicación CP 5611, el Terminal de Válvulas CPV DI01 y la HMI KTP700 Basic DP. Estos 4 dispositivos integran la red Profibus.

También desde el **Catálogo de hardware** se pueden agregar la fuente de alimentación del PLC y el módulo de entradas/salidas.

En la figura 5.50 se puede apreciar el PLC junto con sus módulos.

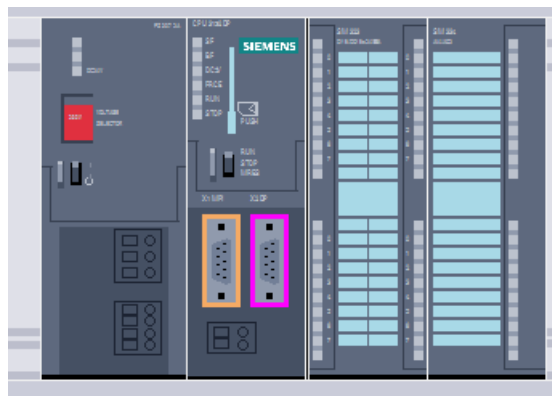


Figura 5.50: PLC junto con sus módulos en TIA Portal

5.1.1. Configuración red Profibus

Se debe configurar los parámetros de red de cada dispositivo, correspondiente a la dirección y velocidad de transferencia de bits. En este proyecto, el PLC actuará como maestro DP y por su parte, el cabezal de válvulas Festo funcionará como esclavo DP.

Configuración de parámetros de red Profibus DP:

- CP5611 Profibus DP: Dirección 0, 1.5 Mbps
- CPU CP315-2DP: Dirección 2, 1.5 Mbps
- HMI KTP500 Basic DP: Dirección 4, 1.5 Mbps
- CPV DI01: Dirección 6, 1.5 Mbps

Una vez hechas estas configuraciones, se tiene la siguiente red de conexión Profibus

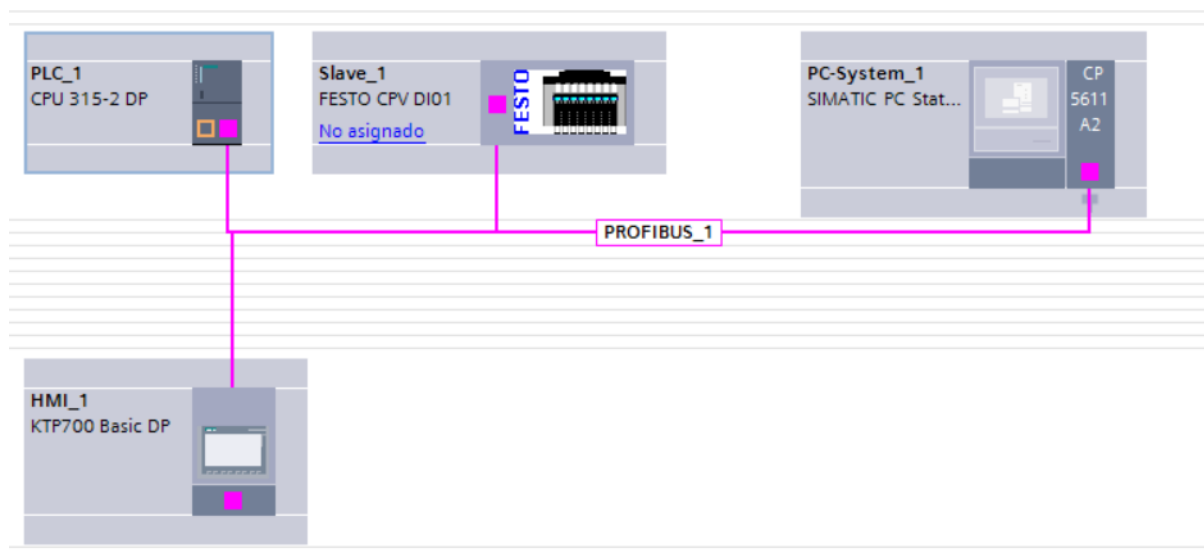


Figura 5.51: Red Profibus

Para el detalle de como se agregan los dispositivos al proyecto y la configuración de la red Profibus vea 8.2.1.

5.2. Automatización del STC

Para comenzar, es esencial crear una secuencia lógica de operación que el PLC seguirá meticulosamente, avanzando paso a paso para llevar a cabo las transiciones de un estado a otro. Esta labor se lleva a cabo mediante el diseño de la automatización utilizando un diagrama Grafset, el cual define claramente los diferentes estados que la rutina de programación debe cumplir. Por lo tanto, la primera tarea consiste en dividir la rutina en múltiples estados bien definidos. Luego, se sabe que el cambio de estados viene ligado con que si se cumplen o no las transiciones, por lo que se deben crear las diferentes condiciones que establecen el cambio de estados.

Luego de dividir los estados y definir las transiciones se llega a un diagrama como el mostrado en figura 5.52. La rutina de programación consta de 11 macro estados, donde algunos de ellos al programar en S7-Graph se dividen en más estados, por lo tanto el diagrama presentado es más bien funcional, pero que describe todas las opciones y funcionalidades que posee la rutina de programación que se implementa.

El diagrama secuencial se lee de arriba hacia abajo, donde cada estado posee las acciones que debe ejecutar escritas a su costado y las transiciones aparecen con una línea horizontal entre etapas.

El inicio de la rutina comienza al pulsar el botón de inicio de programa, pasando al estado S1 en donde se enciende el compresor y se activa la correa de transporte. Luego, el programa se divide en tres ramas alternativas, por un lado si se presiona el botón fin de programa, se apagan todos los elementos (esto en el estado S4) y la rutina vuelve al estado inicial. Otra alternativa es que si se activa el sensor en ET1, quiere decir que el pallet está vacío, por lo que en el estado S3 se aumenta el contador de pallet y vuelve al estado S1. En caso que se detecte que un pallet lleva carga, se pasa al estado S2, donde se acciona el freno en ET1 (cilindro neumático). Luego de un tiempo, se pasa al estado S5 en donde se apaga la correa de transporte. Aquí se espera un segundo timer, pasando al estado S6 que es un estado de espera. En este punto, el pallet está detenido en ET1, si se activa el sensor en ET1 la secuencia pasa al estado S7, donde el objeto es tomado utilizando los componentes neumáticos. Luego de que se cumpla un tiempo y se ha tomado el objeto desde ET1, se pasa al estado S8, donde se deja

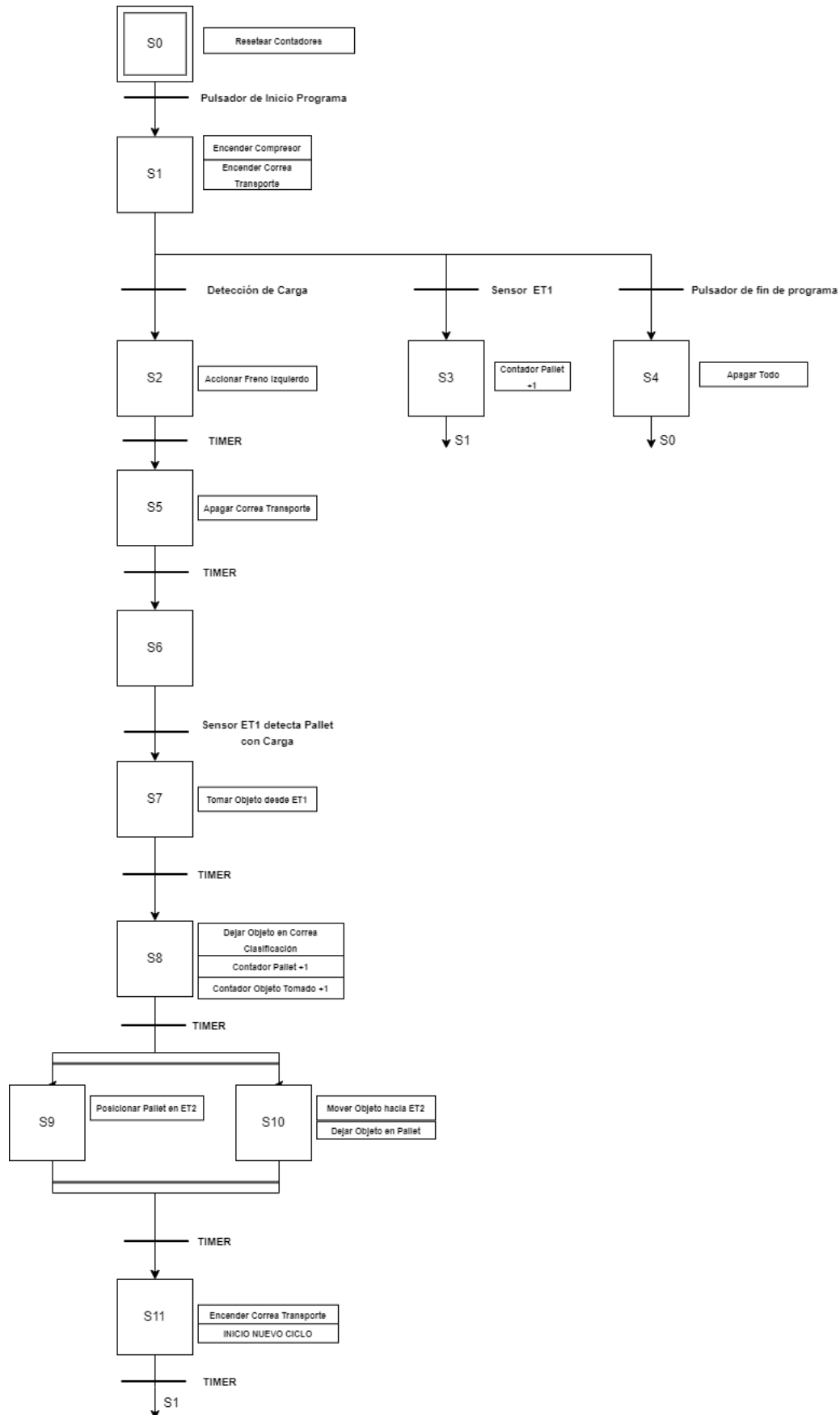


Figura 5.52: Diagrama Grafset de la aplicación secuencial programada

el objeto en la correa de clasificación y además se aumentan en una unidad los contadores de objetos tomados y pallets que han circulado por la correa principal. Luego de un timer la rutina se divide en 2 ramas simultáneas (suceden a la vez), donde el estado S9 posiciona el pallet en ET2 encendiendo la correa de transporte y activando el freno en ET2, para que una vez que llegue a ET2 apague la correa de transporte; y el estado S10 a su vez mueve a través de la correa de clasificación el objeto desde ET1 a ET2, una vez llega a ET2 (detectado por el sensor en ET2) se posiciona el objeto en el pallet a través de los mecanismos neumáticos. Cuando finalizan ambos estados simultáneos y ha pasado un tiempo se pasa al estado S11, donde se libera el freno en ET2 y se activa la correa de transporte nuevamente iniciando así un nuevo ciclo.

Resumiendo la rutina implementada realiza lo siguiente:

- Se detecta si viene un pallet con carga y se activa el freno en ET1.
- Los pallets que no traen objeto son detectados en ET1 y se aumenta el contador de pallets totales que han pasado por la correa principal.
- Si el pallet trae objeto, lo toma y lo lleva hasta ET2, a través de la correa de clasificación.
- Luego se suelta el freno de ET1 y se activa el freno de ET2. De esta forma se mueve el pallet hacia la ET2.
- Una vez puesto el objeto en la correa de clasificación, se incrementa el contador de pallets con objeto.
- Cuando se detecta que el objeto que se mueve a través de la correa de clasificación llega a ET2, se activa la transferencia hasta el pallet.

Para la programación detallada en S7-Graph, con todos los estados de la rutina, junto a transiciones y acciones, se puede revisar el Anexo (8.3).

Además, para la programación en las tablas 5.4 y 5.5 se tiene información con respecto a entradas y salidas, que en este caso las direcciones no cambian entre el sistema simulado y el sistema físico.

| Accesorios | Salidas lógicas | Descripción |
|------------|-----------------|---|
| MovCorrP | Q0.0 | Habilita movimiento de la correa de transporte |
| ActComp | Q0.1 | Activa el compresor de aire |
| HabCorrC | Q0.2 | Habilita movimiento de la correa de clasificación |
| MovCorrC | Q0.3 | Define movimiento de la correa de clasificación |
| PFrenoIzq | Profibus Q2.0 | Activa pistón de freno izquierdo |
| PElevIzq | Profibus Q2.2 | Activa pistón de elevación izquierdo |
| PExtIzq | Profibus Q2.4 | Activa pistón de extensión izquierdo |
| PFrenoDer | Profibus Q2.6 | Activa pistón de freno derecho |
| PElevDer | Profibus Q3.0 | Activa pistón de elevación derecho |
| PExtDer | Profibus Q3.2 | Activa pistón de extensión derecho |
| SuccIzq | Profibus Q3.4 | Activa tobera de aspiración izquierda |
| SuccDer | Profibus Q3.6 | Activa tobera de aspiración derecha |

Tabla 5.4: Salidas lógicas

| Accesorios | Entradas lógicas | Descripción |
|------------|------------------|---|
| BL1 | I1.0 | Botón pulsador libre "verde", normalmente abierto |
| BL2 | I1.1 | Botón pulsador libre "rojo", normalmente cerrado |
| Prox1 | I1.2 | Sensor de proximidad infrarrojo difuso |
| Prox2 | I1.3 | Sensor de proximidad infrarrojo reflectivo |
| Prox3 | I1.4 | Sensor de proximidad infrarrojo difuso |

Tabla 5.5: Entradas lógicas

5.3. Implementación en STC virtual

Para implementar la automatización en el STC virtual, se debe iniciar la simulación del PLC en TIA Portal, en este caso a través de PLCSIM, luego se debe enlazar las variables a través de EzOPC, conectando PLCSIM con Ciro's Studio. El diagrama de la implementación del STC virtual se puede ver en la figura 5.53, donde las flechas indican que se comparten el valor de las variables del sistema.

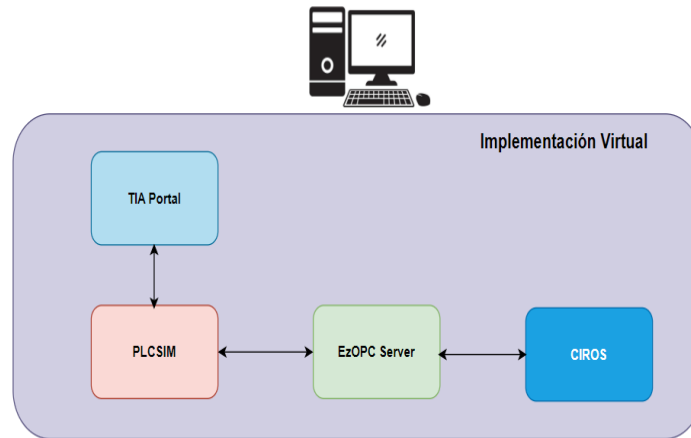
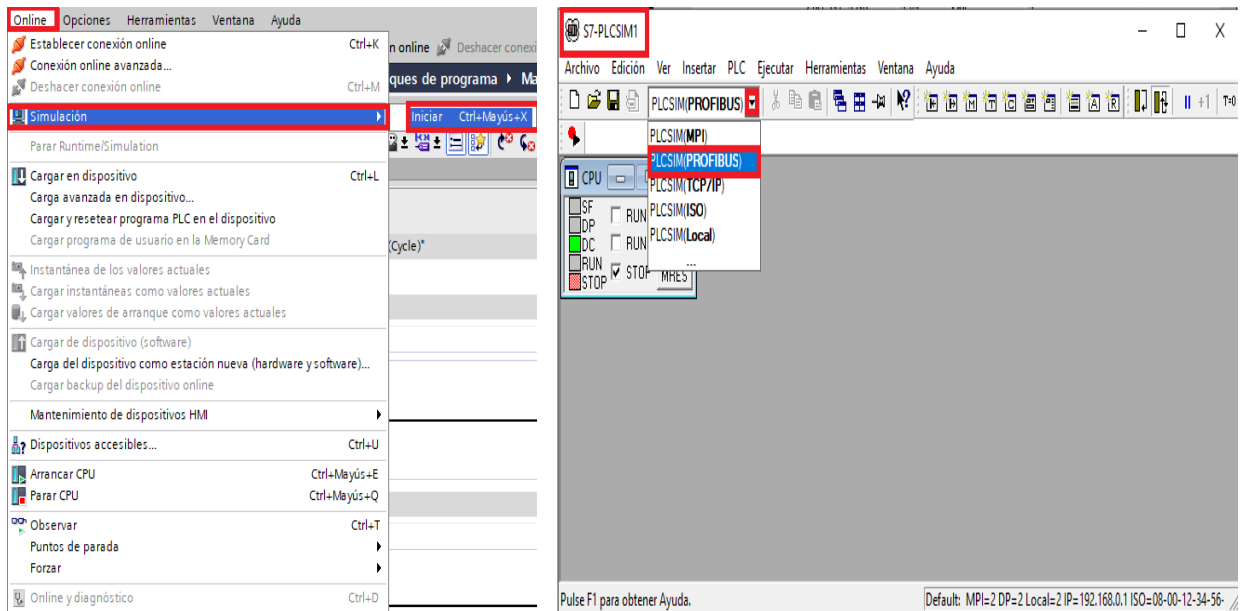


Figura 5.53: Implementación del STC virtual

5.3.1. Cargar programa en PLCSIM

Una vez iniciada la simulación del PLC virtual, se inicia el proceso de carga del programa en este. Para iniciar la simulación, en la barra de herramientas en la parte superior izquierda del software TIA Portal hay que seguir la siguiente serie de pasos: **Online -> Simulación -> Iniciar**, como lo muestra la figura 5.54a. Luego, se abrirá una ventana con el PLC virtual, dónde hay que escoger la interfaz que se utilizará, en este caso **Profibus**, esto se puede ver en 5.54b.



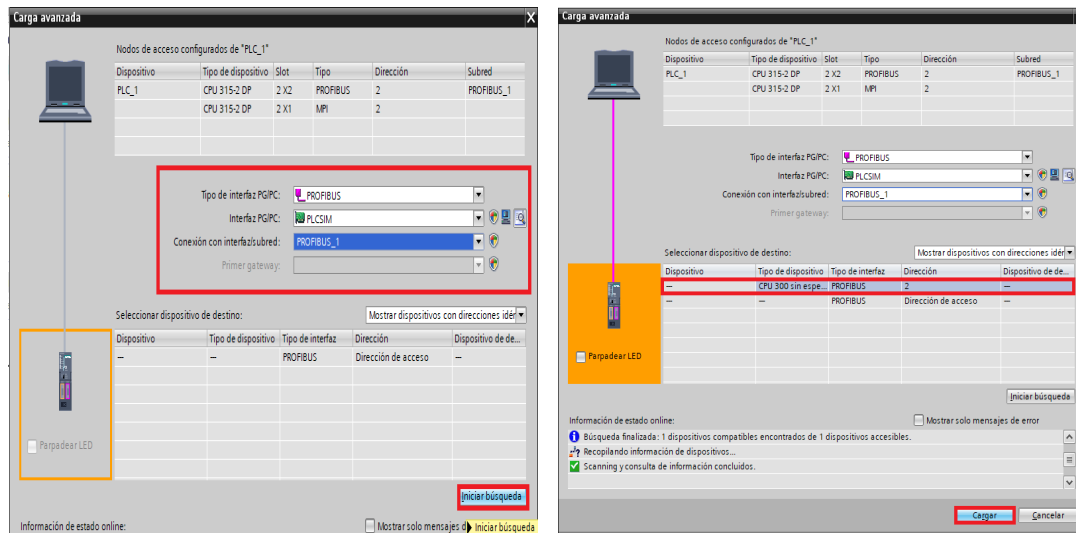
(a) Inicio simulación

(b) Configuración en PLC virtual

Figura 5.54: Iniciación Simulación PLC virtual

Una vez se haya hecho los cambios en el PLC virtual, hay que escoger el tipo de interfaz PG/PC (PROFIBUS), la interfaz debe ser a través de PLCSIM y se debe escoger la red PROFIBUS_1, que es la red en que están enlazados los dispositivos. Finalmente se debe dar a **Iniciar búsqueda**.

Una vez terminada la búsqueda, aparece en la red el PLC virtual al cual se le puede cargar el programa, para esto hay que hacer click en **Cargar**.



(a) Configuración interfaz PG/PC

(b) Carga del programa

Figura 5.55: Carga avanzada

Con estos pasos realizados, se tiene cargado el programa en el PLC virtual.

5.3.2. Enlazamiento de variables a través de EzOPC

Para poder enlazar las variables a través de OPC, específicamente a través de EzOPC server, se debe agregar al árbol desde las librerías el Objeto **EzOPC** desde **Festo EzOPC**. Una vez agregado, en las propiedades del objeto, hay que dar click en **OPC Client** y se abrirá una ventana como la de la figura 5.56. Aquí en server name hay que escoger **FestoDidactic.EzOPC.1** para luego seleccionar **Connect items**.

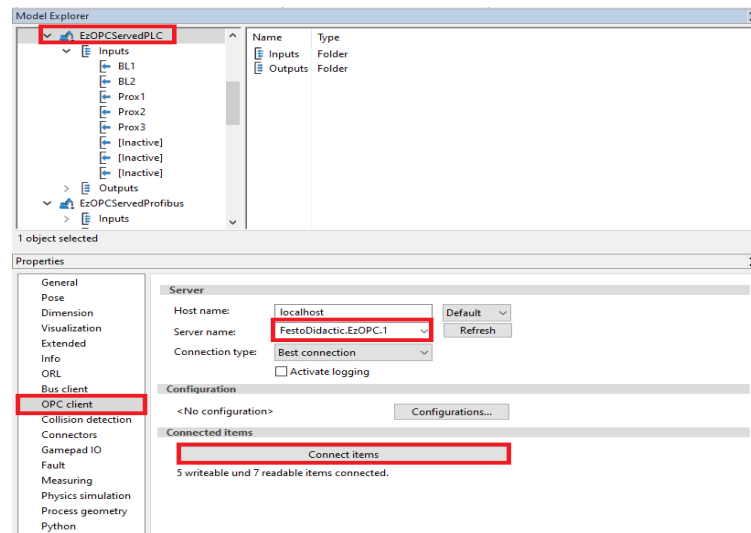


Figura 5.56: OPC Client

Una vez que se selecciona **connect items**, se abre otra ventana. En la parte izquierda se tienen las entradas y salidas para el cliente, donde están identificadas por sus nombres. Para conectar las variables en CiroS con PLCSIM, hay que dar click en **Connect -> PLCSIM -> Bits**, donde se desplegará las entradas y salidas del PLC virtual. No es muy intuitivo el asociar entradas y salidas debido a que la nomenclatura es un poco confusa. En este caso las entradas vienen asociadas con la letra E y las salidas con la letra A, y cada una está asociada al bit que corresponda. Por ejemplo la dirección %Q0.0 viene dada por AB0Bit00. Esto se puede ver en la figura 5.57

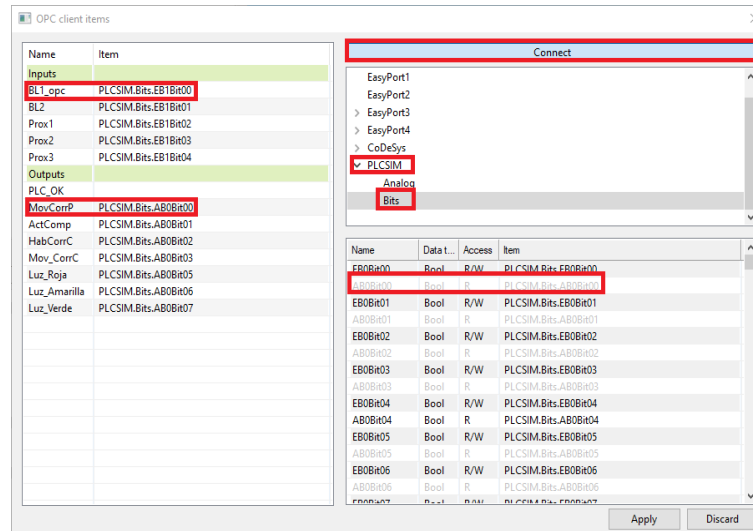


Figura 5.57: Variables compartidas

Así es cómo se asocia cada variable entre CiroS y PLCSIM. Luego, se puede observar en la figura 5.58 como están asociadas las variables con el servidor y a qué están conectadas en la simulación. Por ejemplo BL1_opc está asociada con el botón virtual BL1 de la simulación y BL1_opc está asociada con la dirección EB1Bit00, que corresponde a la dirección %I1.0 en las tablas de variables en TIA Portal.

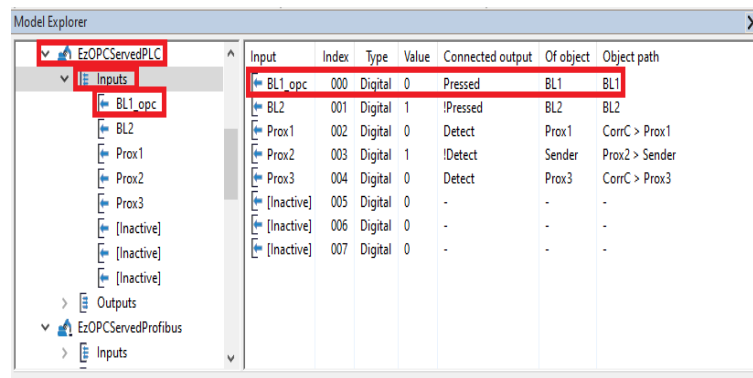
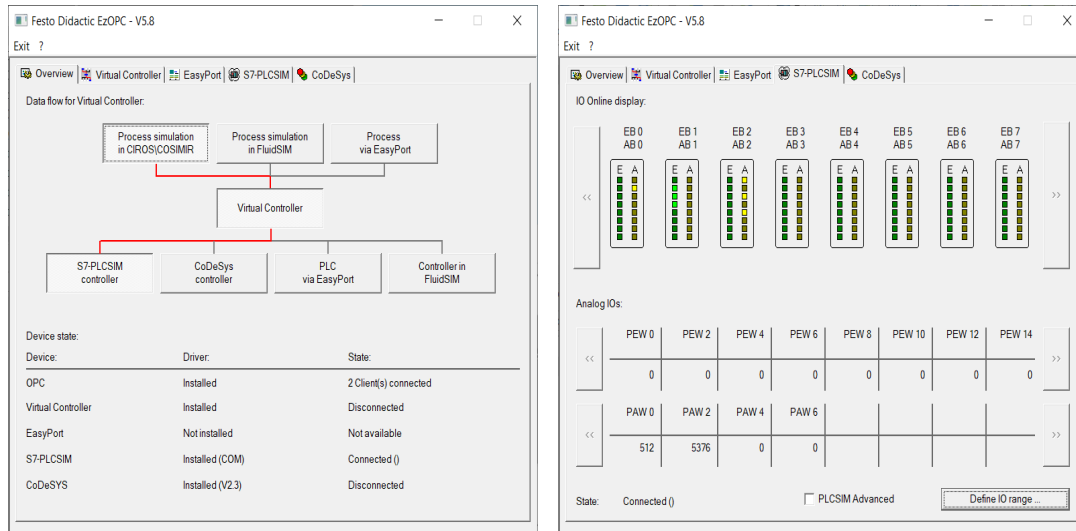


Figura 5.58: Ejemplo de conexión

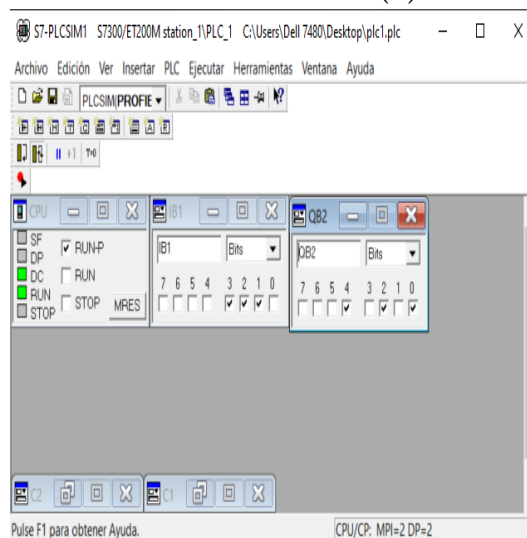
En la figura 5.59 se puede ver la conexión exitosa entre CiroS studio y PLCSIM. También muestra que existen dos clientes (esto porque en CiroS se configuró para que se conectase a través de dos clientes, uno para las variables del PLC y otro para las variables del Terminal de Válvulas), como también el monitoreo de las señales en el PLC virtual que está en EzOPC

server. Es importante decir que la actualización de las variables es de 10 milisegundos, lo que permite una simulación en tiempo real del STC.



(a) Conexión Ciros Studio con PLCSIM

(b) PLC virtual en EzOPC



(c) PLCSIM

Figura 5.59: EzOPC Server

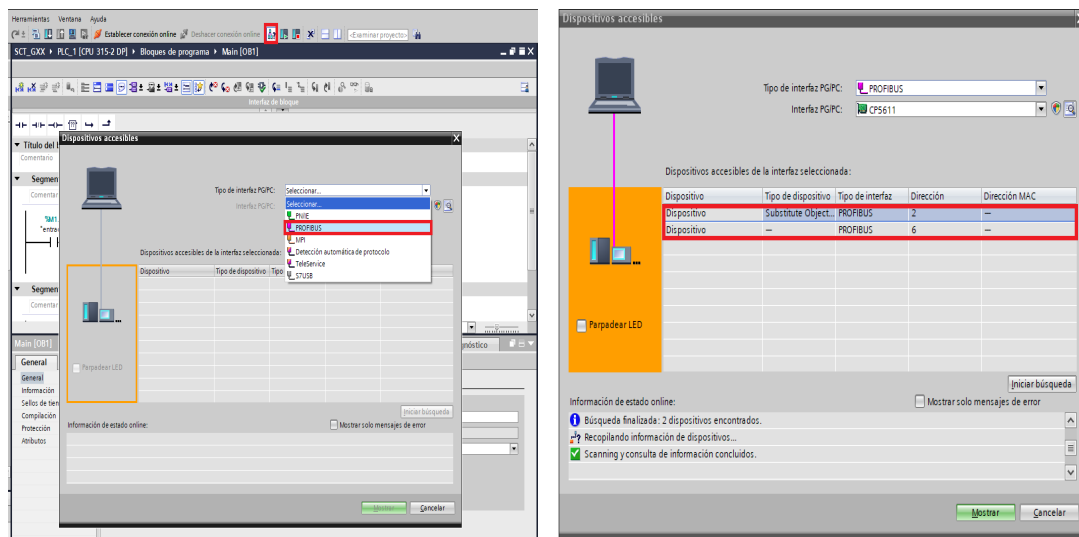
5.4. Implementación en STC físico

La automatización que se implementa en el STC físico es la misma que se implementa en el STC virtual, por lo que no se realizan cambios.

Para implementar la automatización en el STC físico, se debe cargar el programa al PLC

físico a través de la red PROFIBUS. Para ello primero hay que asegurarse de que están bien conectados los conectores a los dispositivos, es decir, al PLC, el terminal de válvulas y a la tarjeta CP 5611 en el PC y que además estos están energizados. Para comprobar que los dispositivos están disponibles para ser utilizados, en TIA Portal damos click en **Dispositivos disponibles**, donde se abrirá una ventana como la que muestra la figura 5.60a. En el **Tipo de interfaz PG/PC** hay que escoger **PROFIBUS**, en **Interfaz PG/PC** se escoge **CP 5611** y luego se debe dar click en **Iniciar búsqueda**.

Aquí debiesen aparecer los dispositivos, PLC y terminal de válvulas, reconocidos por la direcciones que fueron configuradas anteriormente, como se puede ver en figura 5.60b



(a) Configuración interfaz PG/PC

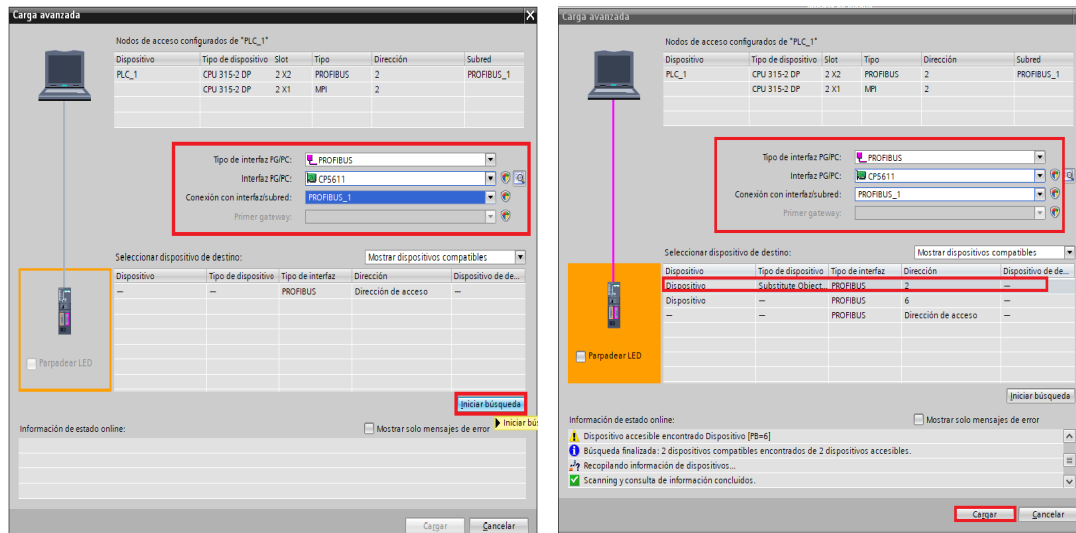
(b) Dispositivos encontrados

Figura 5.60: Búsqueda de dispositivos a través de la red PROFIBUS

Luego, en TIA Portal se debe ir a la barra de herramientas en la parte superior izquierda de la pantalla y dar click en **Online -> Cargar y resetear programa PLC en el dispositivo**, esto para la primera carga, después se puede cargar directamente con **Cargar en dispositivo**.

Se abrirá una ventana como la que se muestra en figura 5.61a, aquí se debe escoger la misma configuración que en el paso de dispositivos disponibles, es decir, en **Tipo de interfaz PG/PC** hay que escoger **PROFIBUS**, en **Interfaz PG/PC** se escoge **CP 5611** y en **Conexión con interfaz/subred** se escoge **PROFIBUS**. Luego, se debe **Iniciar búsqueda**. Si las configuraciones están bien hechas, debiesen aparecer el PLC y el terminal de válvulas.

Aquí se carga el programa en el PLC, que tiene la dirección 2, como se muestra en la figura 5.61b.



(a) Configuración interfaz PG/PC

(b) Dispositivos encontrados

Figura 5.61: Carga del programa en PLC a través de la red PROFIBUS

Una vez hechas estás configuraciones, se abrirán una serie de ventanas para terminar la carga en el dispositivo. Una vez finalizado el sistema está listo para ser ocupado.

5.5. Configuración pantalla HMI virtual

Una pantalla HMI es una pantalla que integra la interfaz humano-máquina, lo que posibilita la supervisión y el control remoto de ciertos sistemas. En esta situación, la pantalla HMI virtual desempeña la tarea de supervisión de los sensores y actuadores del sistema, permitiendo monitorear qué está sucediendo en tiempo real.

Para el diseño de la pantalla en TIA Portal, se pueden incorporar toda clase de elementos disponibles en las bibliotecas de gráficos, así como también imágenes externas.

Se utilizan 4 imágenes en la pantalla. La primera es la principal y permite al usuario desplazarse a las otras 3: La imagen del sistema, dónde se ve el estados de los actuadores y sensores, al igual que los contadores. La imagen que está asociada al Terminal de Válvulas, que permite manipular remotamente los pistones y toberas de aspiración. Finalmente, también se tiene

la imagen Actuadores/Sensores, donde se monitorea aparte los sensores y se puede operar remotamente los actuadores que no están en Terminal de Válvulas, además de los botones BL1 y BL2.

En la figura 5.62 se pueden ver las pantallas de la HMI virtual.

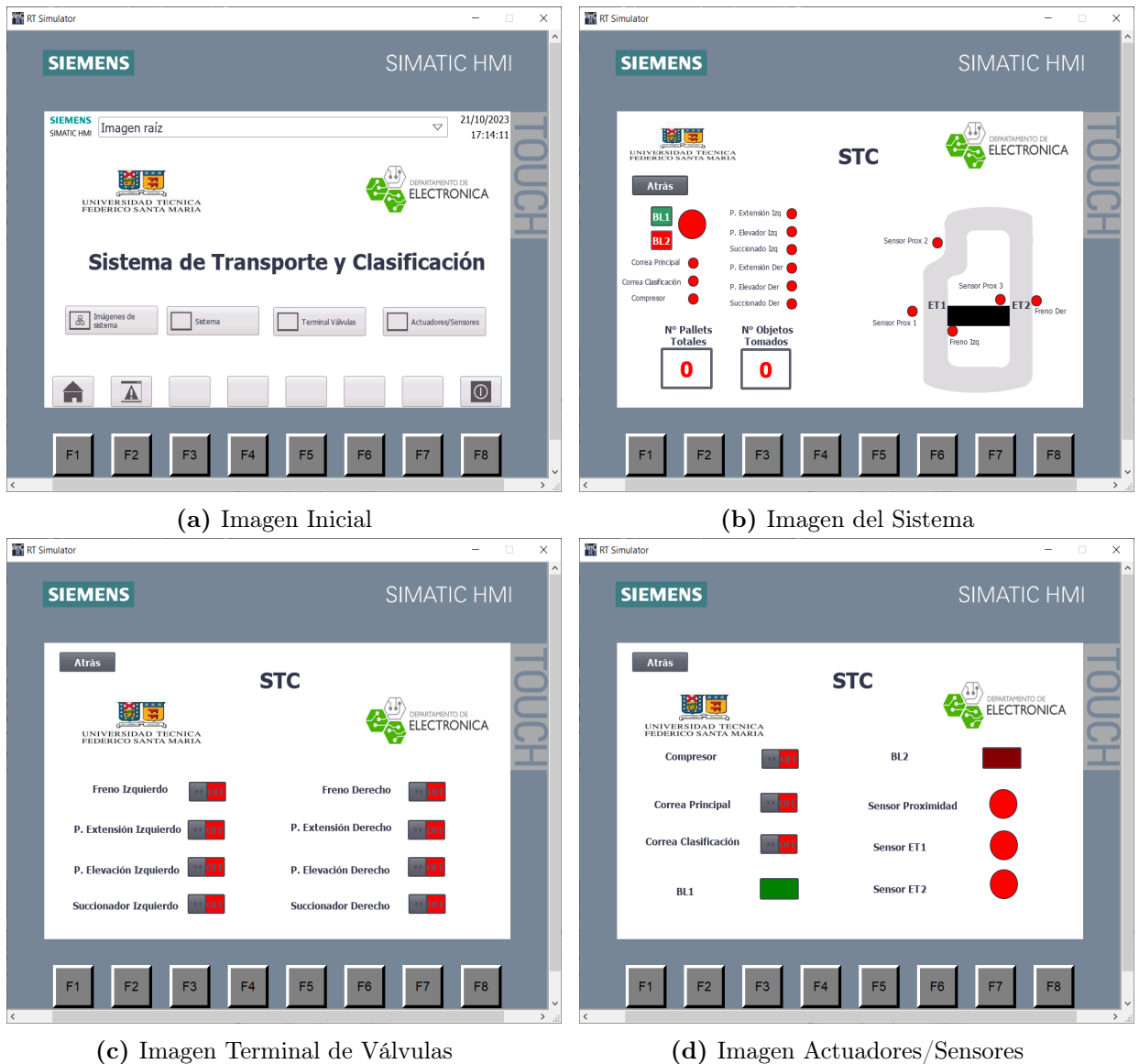


Figura 5.62: HMI

Además, en la figura 5.63 se puede ver el funcionamiento de la HMI, en este caso muestra cuándo el objeto llega a la ET2 y es detectado por el sensor en ET2. Se aprecia que el freno

derecho está activo, al igual que el sensor y el pisto de elevación derecho. También se ve que ya se han tomado 3 objetos desde ET1 que han sido transportados a ET2 y que el número total de pallets que han circulado por la correa principal son 4, esto desde que se inicio el programa.

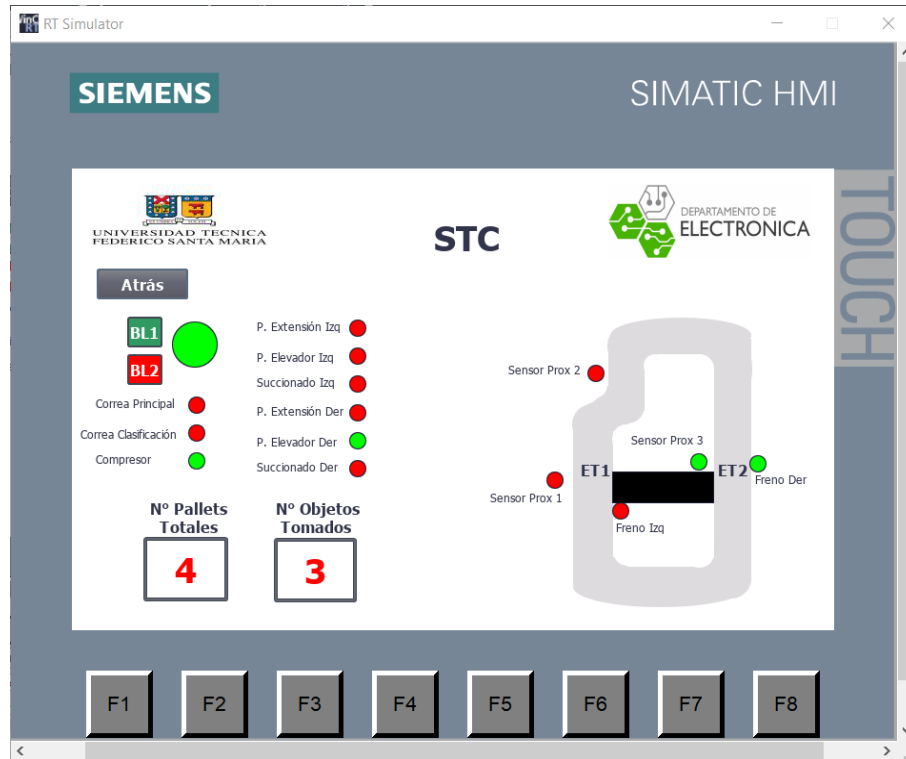


Figura 5.63: Ejemplo HMI Imagen del Sistema

6. Resultados SIL y HH

6.1. Pruebas realizadas de validación física de los esquemas SIL y HH

6.1.1. Prueba de validación del STC virtual - SIL

Para la validación del STC virtual se tiene el PLCSIM cargado con la programación realizada en S7-Graph, conectado a través de EzOPC a Ciro's Studio. Para iniciar la simulación del sistema se da a play en la aplicación de Ciro's.

Una vez se tiene ambos, PLCSIM e iniciada/habilitada la simulación en Ciro's, el programa está a espera de dar inicio a la automatización, que empieza pulsando el botón BL1, ya sea a través de la HMI o el botón que se encuentra en Ciro's.

Se procede a mostrar como se ve en la simulación en diferentes instantes del funcionamiento del sistema. En la figura 6.64 se puede ver cuando un pallet con carga ya fue detectado y el freno izquierdo fue activado.

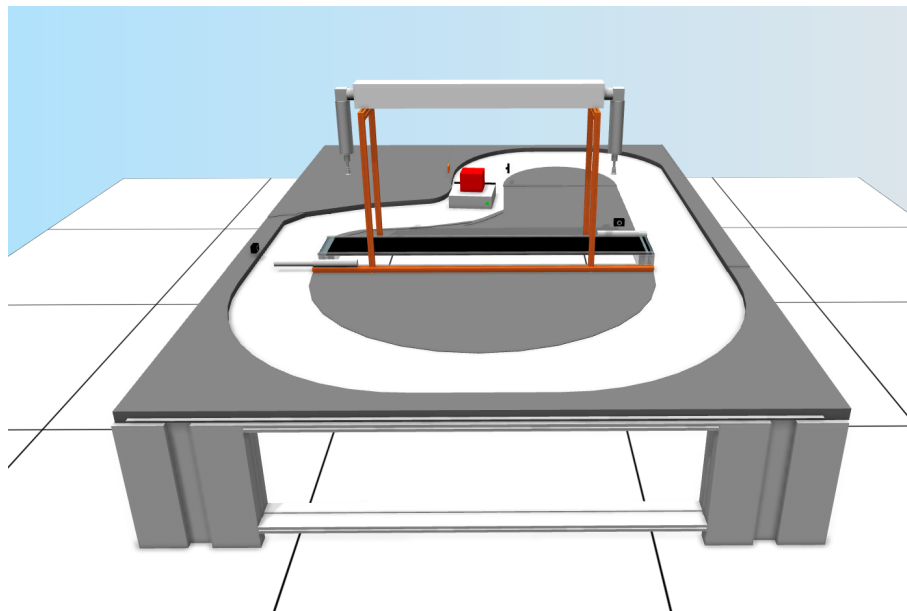


Figura 6.64: Detección de pallet con carga y freno izquierdo activado

Luego, en la figura 6.65, se observa cuando el pallet ha llegado a ET1 y están activos los cilin-

dros de doble efecto correspondientes para realizar la transferencia a la correa de clasificación.

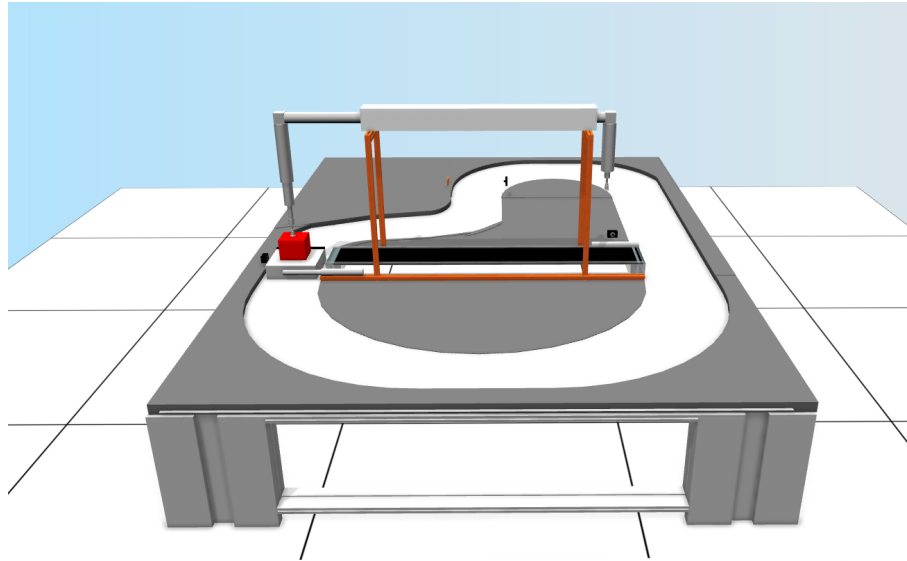


Figura 6.65: Pallet en ET1 y cilindros realizando la transferencia del objeto a correa de clasificación

En la figura 6.66 se ve que el objeto ya ha sido transferido a la correa de clasificación y se ha liberado el freno izquierdo, por lo que el pallet comienza a ser movido hacia ET2.

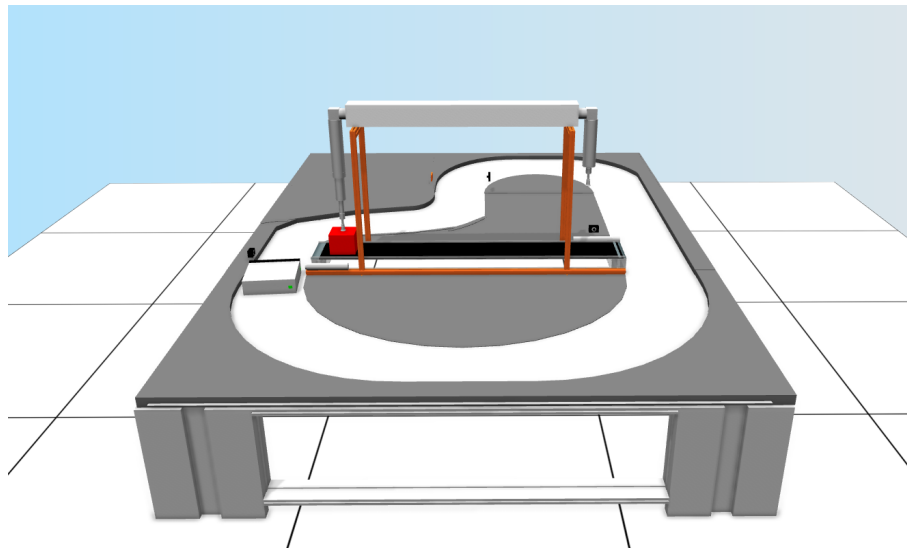


Figura 6.66: Objeto en correa de clasificación e inicio de posicionar al pallet en ET2

En la figura 6.67 se muestra cuando el objeto es detectado por el sensor en ET2, por lo que comienza el traslado del objeto hacia el pallet, que ya está posicionado. También se puede ver activo el freno derecho.

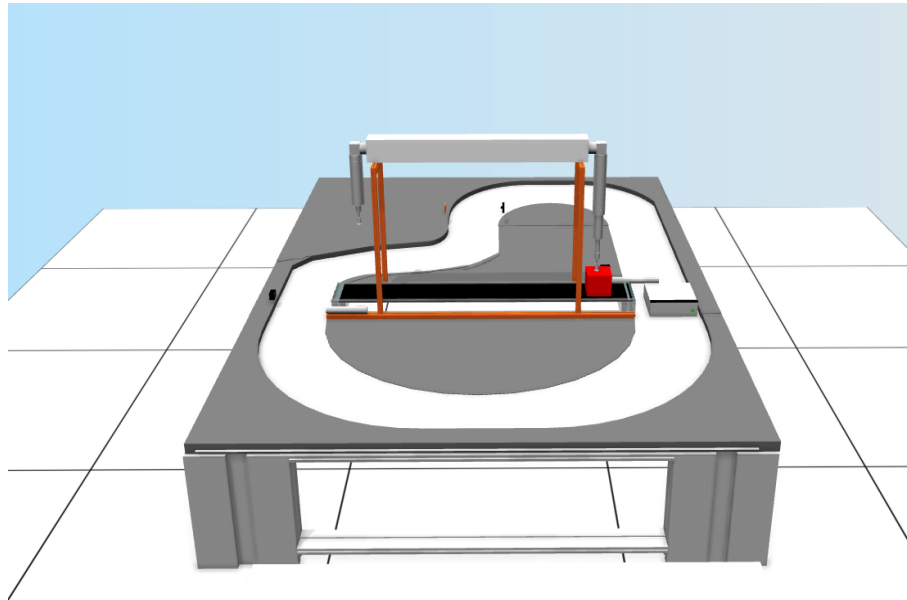


Figura 6.67: Objeto en ET2 esperando a ser movido hacia el pallet

Finalmente, el objeto es posicionado nuevamente en el pallet, como lo muestra la figura 6.68, y el sistema está listo para comenzar otro ciclo.

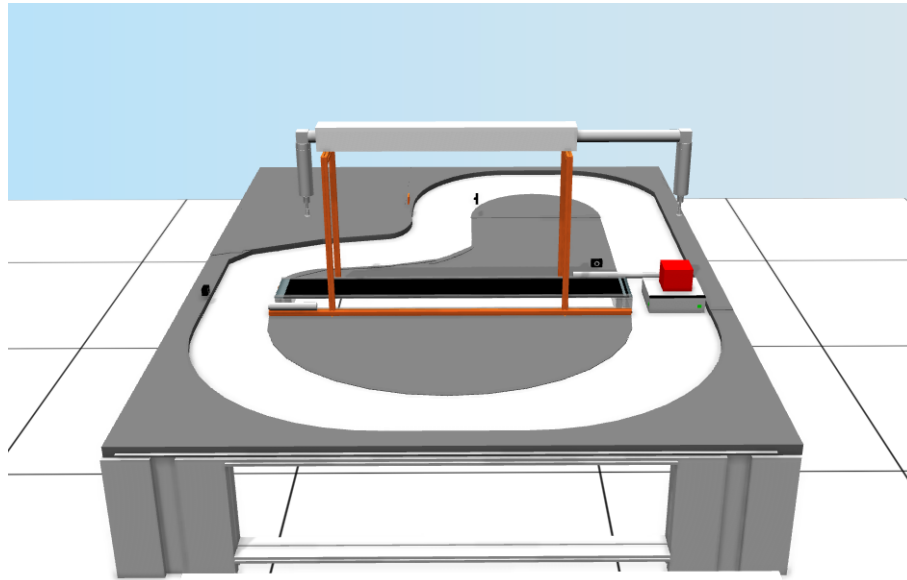


Figura 6.68: Objeto posicionado nuevamente en el pallet

También es posible ver que si el pallet no contiene carga, no se activa el freno en ET1 y continua por la correa principal, esto en la figura 6.69.

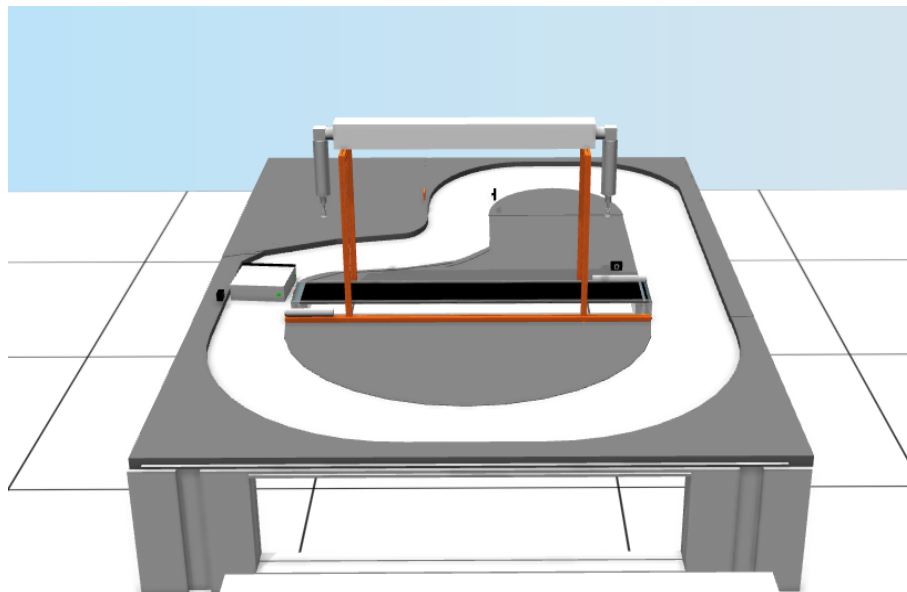


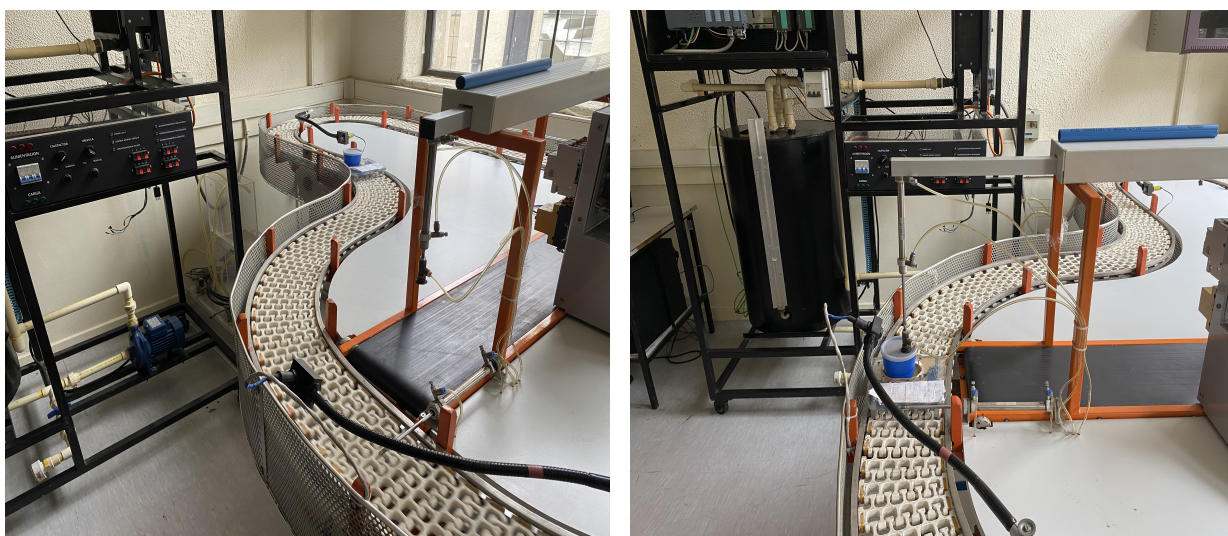
Figura 6.69: Pallet sin carga por la correa principal

6.1.2. Prueba de validación del STC físico - HH

Para realizar las pruebas en el sistema físico, primeramente, se debe realizar la carga del programa de la automatización en el PLC físico, a través de la red PROFIBUS configurada previamente.

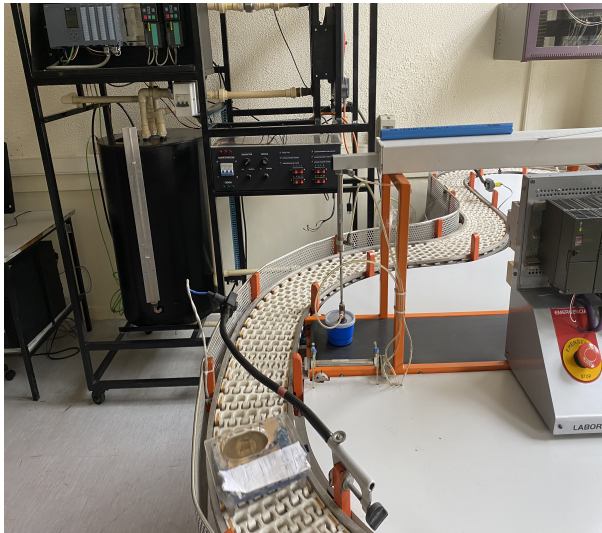
Se da a inicio a la automatización pulsando el botón libre 1 que se encuentra en el panel físico (BL1), con lo que comienza a moverse la correa principal.

Se muestra en la figura 6.72 las mismas instancias mostradas para la validación virtual, con el fin de comparar.

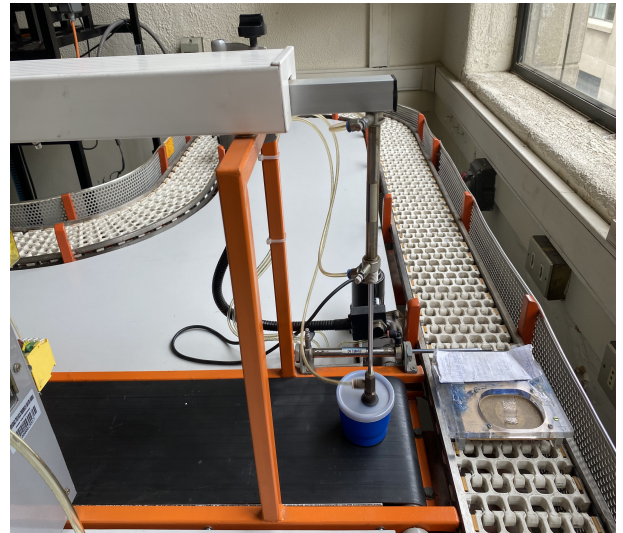


(a) Detección de pallet con carga y freno izquierdo activado (b) Pallet en ET1 y cilindros realizando la transferencia del objeto a correa de clasificación

Figura 6.70: Funcionamiento automatización en STC físico 1

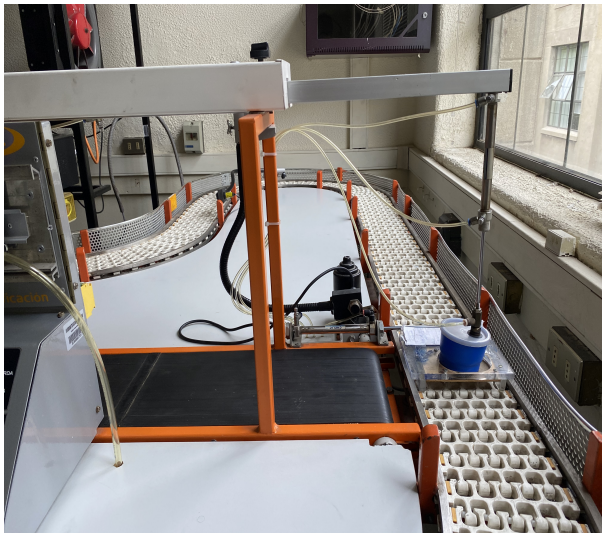


(a) Objeto en correa de clasificación e inicio de posiciónar al pallet en ET2

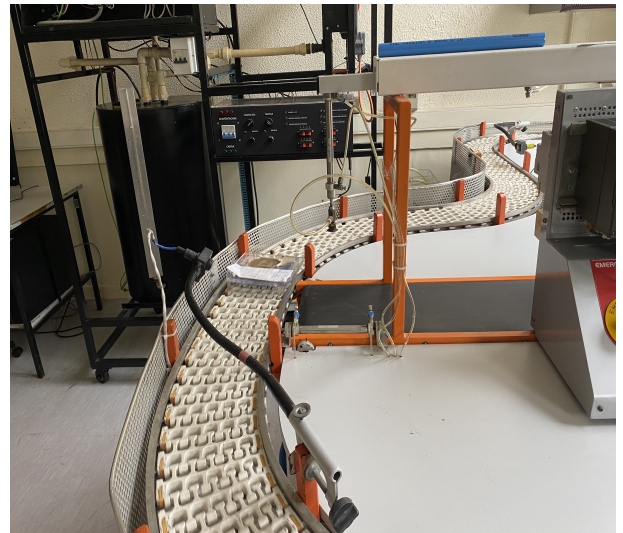


(b) Objeto en ET2 esperando a ser movido hacia el pallet

Figura 6.71: Funcionamiento automatización en STC fisico 2



(a) Objeto posicionado nuevamente en el pallet



(b) Pallet sin carga por la correa principal

Figura 6.72: Funcionamiento automatización en STC fisico 3

6.2. Análisis de resultados

La automatización del sistema es lograda en la simulación 3D, cumpliendo con los requisitos funcionales para el sistema de transporte y clasificación, todo esto realizado en tiempo real.

Además se pudo validar el funcionamiento de los cilindros de doble efecto, toberas de aspiración y sensores, los cuales se replican con bastante exactitud con respecto al del sistema físico, esto en cuanto a la dinámica y la detección de objetos, como también la visualización (forma física, colores y posición).

Para el caso de la implementación de la automatización en el sistema de transporte y clasificación físico, el resultado es bastante bueno. Esto debido principalmente a que fue probado primero en la simulación. Lo importante es que no fue necesario realizar ningún cambio para poder implementar la automatización, esto incluye a la programación en S7-Graph y las variables del sistema, logrando así una carga directa hacia el PLC físico escogiendo la adecuada interfaz PG/PC.

El sistema es flexible, si se quiere cambiar por ejemplo la posición de los sensores en el sistema físico, es posible replicar esto en la simulación de manera sencilla. Esto podría ser necesario realizarlo en caso de que se quiera cambiar alguno de los requisitos funcionales por ejemplo, ameritando modificar el modelo virtual.

Hay que dejar en claro algunos detalles: la simulación puede fallar en algunos casos. Si bien es bastante robusta durante la gran parte del tiempo, en ocasiones tiene fallas, un ejemplo es un error que arroja el programa indicando que el objeto no está lo suficientemente cerca de la tobera de aspiración, por lo que se interrumpe la simulación, cuando el objeto en realidad si lo está. También, si se le hace mucho zoom a la simulación, en ocasiones se ralentiza esta, por lo que ocurren errores como que el pallet no llegue a posicionarse correctamente a alguna de las estaciones, esto debido a que en el PLC sigue corriendo el programa, incluido los timers y no distingue si la simulación está más lenta o no.

Generalmente estos errores ocurrieron cuando el computador donde estaban siendo ejecutados los programas estaba consumiendo muchos recursos, por lo que la dedicación a estos era menor. En el caso de uso de solo los programas de simulación y control no hubieron problemas.

7. Conclusiones y Trabajo Futuro

7.1. Conclusiones

Con respecto al STC virtual, se logró configurar y simular el sistema con un modelo en 3D en uno de los computadores en el Laboratorio de Control Industrial, el cual contiene componentes mecánicas y electroneumáticas funcionales y visualmente equivalentes al del STC físico. Esto realizado en Ciro Studio.

También fue posible conectar mediante OPC el STC virtual con el sistema de control virtual, el cual está formado por un PLC S7-300 y un Terminal de Válvulas CPV DI01, aunque éste último no se simula como tal, pero es igual parte del control, asociado a través de las variables con el PLC virtual.

Se logró diseñar la automatización del STC mediante Grafcet, para luego ser implementada en el lenguaje S7-Graph. Dicha automatización se validó con éxito en primera instancia en el STC virtual a través de la simulación, para luego ser implementada en el STC físico, esto cargando el código desarrollado en el sistema de control físico conectado con el STC mediante I/O físicas, validando también su desempeño.

Con respecto a los aportes realizados, se logra desarrollar una simulación dinámica 3D en tiempo real del Sistema de Transporte y Clasificación disponible en el Laboratorio de Control Industrial, la cual se puede utilizar para diseñar y poner en marcha la automatización del sistema mediante un sistema de control virtual equivalente al sistema de control físico, utilizando herramientas que son ampliamente utilizadas en la industria como lo son TIA Portal de Siemens y el protocolo OPC.

En otros trabajos de títulos que se realizaron anteriormente, implicó la utilización de Net-ToPLCSIM para poder conectarse entre el sistema de control y el servidor OPC. En este trabajo no fue necesario de un software intermedio, donde la conexión fue directa entre PLC-SIM y EzOPC, por lo que facilita el desarrollo de la implementación del sistema de forma virtual.

En la actualidad, las simulaciones de sistemas son ampliamente utilizadas en la industria, por lo que es importante que los estudiantes, en este caso del Laboratorio de Control Industrial,

puedan interactuar/trabajar con este tipo de herramientas, por lo que el resultado final de este trabajo puede ayudar en su formación.

7.2. Trabajo Futuro

Teniendo en cuenta que en el proyecto realizado no fue posible implementar el esquema HIL, un trabajo futuro podría ser implementarlo. Para ello será necesario cambiar la tarjeta de conexión Profibus PCI Siemens CP 5611 que tiene el PC del laboratorio por una versión más reciente, esto debido a que los softwares actuales ya no tienen soporte para ésta, imposibilitando la conexión a través de OPC entre el PLC físico y la simulación del STC.

La capacidad que tiene Ciro es grande, lo realizado en este trabajo de título solo es una pincelada de todo lo que se puede realizar en el programa, por lo que si se desea implementar un sistema más complejo, agregando más sensores por ejemplo o agregando más etapas donde interactúen otros elementos, perfectamente puede ser replicado en la simulación. Con esta idea en mente una mejora al sistema sería la implementación de un panel de operación manual en el STC virtual, para verificar la funcionalidad de los elementos sin la necesidad de conectarse primero a través del PLC virtual.

También, hoy en día la industria está tomando un enfoque hacia el IIoT (Industrial Internet of Things), que se refiere a la red colectiva de dispositivos conectados y a la tecnología que facilita la comunicación entre los dispositivos y la nube, así como entre los propios dispositivos, enfocada a la industria. Por lo que sería interesante poder implementar este tipo de tecnologías en esta experiencia, pudiendo así por ejemplo monitorear o interactuar con el sistema a través de la nube.

Tener estos temas en consideración será importante para avanzar en la realización de gemelos digitales, los cuales son el futuro de la industria 4.0.

Referencias

- [1] A. López, “Instrumentación, automatización y supervisión de un sistema de transporte y clasificación,” in *Departamento de Electrónica, UTFSM*, 2010. [Online]. Available: <https://catalogo.usm.cl/cgi-bin/koha/opac-detail.pl?biblionumber=100014>
- [2] B. Riera and B. Vigário, “Home i/o and factory i/o: a virtual house and a virtual plant for control education,” in *AC-PapersOnLine*,, 2017.
- [3] I. Salvatierra, “Simulation software for automation industry, bachelor degree,” in *University of G avle*, 2018.
- [4] MathWorks, “Software-in-the-loop simulation,” [accedido el 05-12-2023]. [Online]. Available: <https://es.mathworks.com/help/ecoder/software-in-the-loop-sil-simulation.html>
- [5] —, “Pruebas hardware-in-the-loop (hil) para electrónica de potencia,” [accedido el 05-12-2023]. [Online]. Available: <https://es.mathworks.com/solutions/electrification/hardware-in-the-loop.html>
- [6] —, “Processor-in-the-loop simulation,” [accedido el 06-12-2023]. [Online]. Available: <https://es.mathworks.com/help/ecoder/processor-in-the-loop.html>
- [7] —, “What is model-based testing?” [accedido el 05-12-2023]. [Online]. Available: <https://es.mathworks.com/discovery/model-based-testing.html>
- [8] F. Jaensch, A. Csiszar, J. Sarbandi, and A. Verl, “Reinforcement learning of a robot cell control logic using a software-in-the-loop simulation as environment,” in *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, 2019, pp. 79–84.
- [9] M. Vosáhl, “Hil simulator for analysis and demonstration of smart control algorithms for overhead cranes,” in *Master’s thesis, University of West Bohemia*., 2020.
- [10] Festo, “Página de ciros studio,” [accedido el 30-10-2023]. [Online]. Available: <https://www.didacticsservices.co.uk/products/ciros>
- [11] What is opc? [accedido el 30-10-2023]. [Online]. Available: <https://opcfoundation.org/about/what-is-opc/>



DEPARTAMENTO DE
ELECTRONICA
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



8. Anexos

8.1. Planos Eléctricos [1]

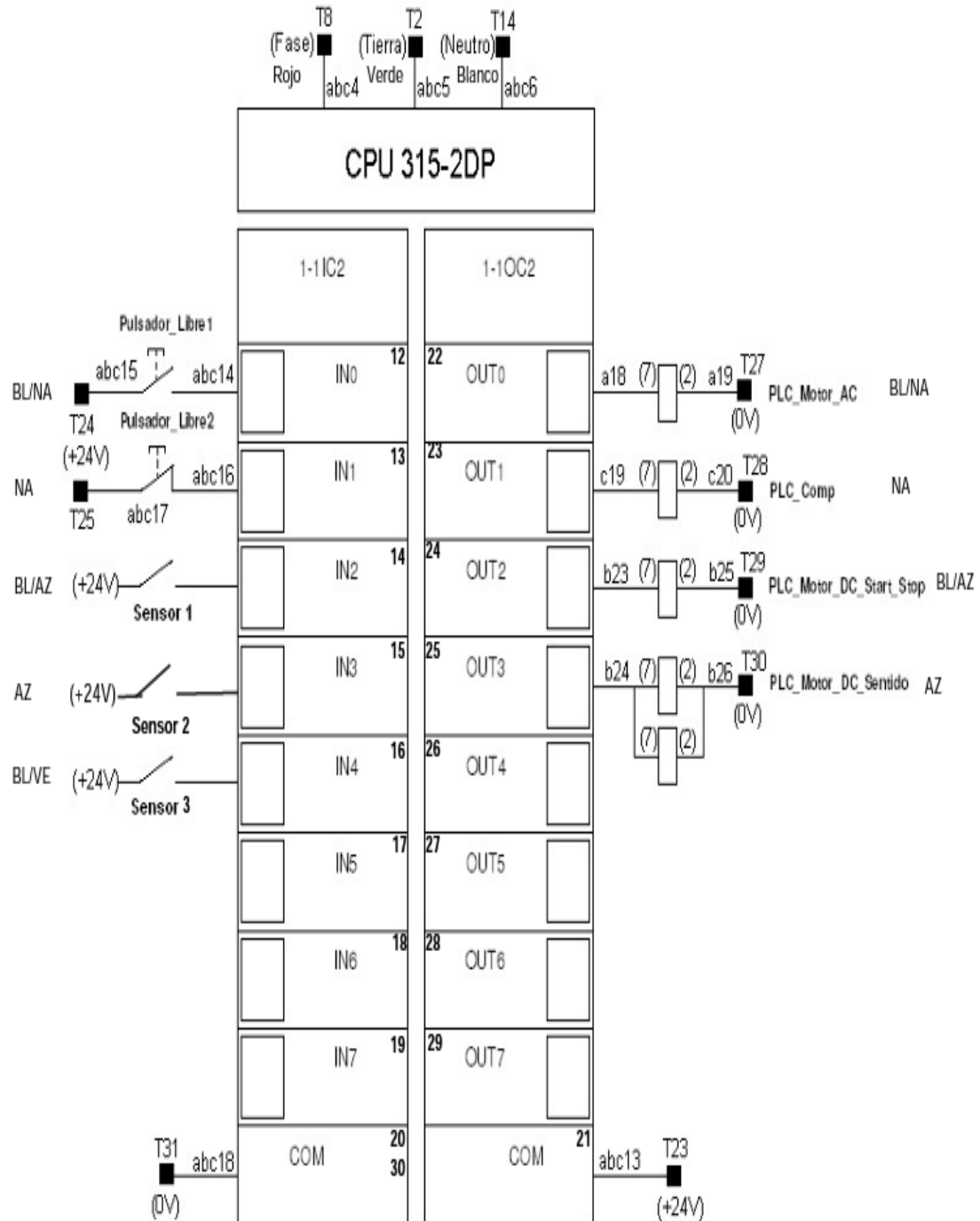


Figura 8.73: Plano eléctrico de las conexiones al PLC

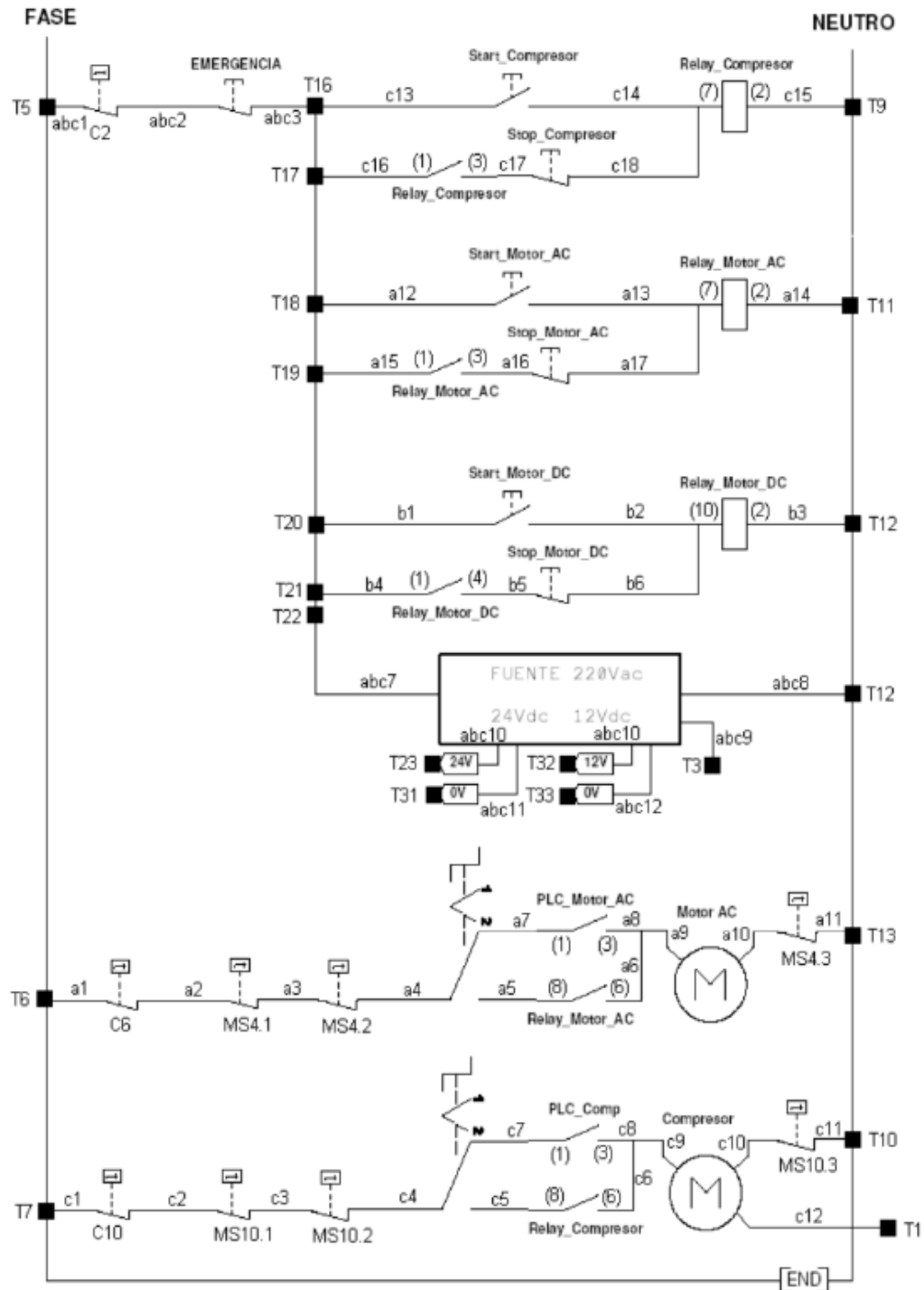


Figura 8.74: Plano eléctrico principal

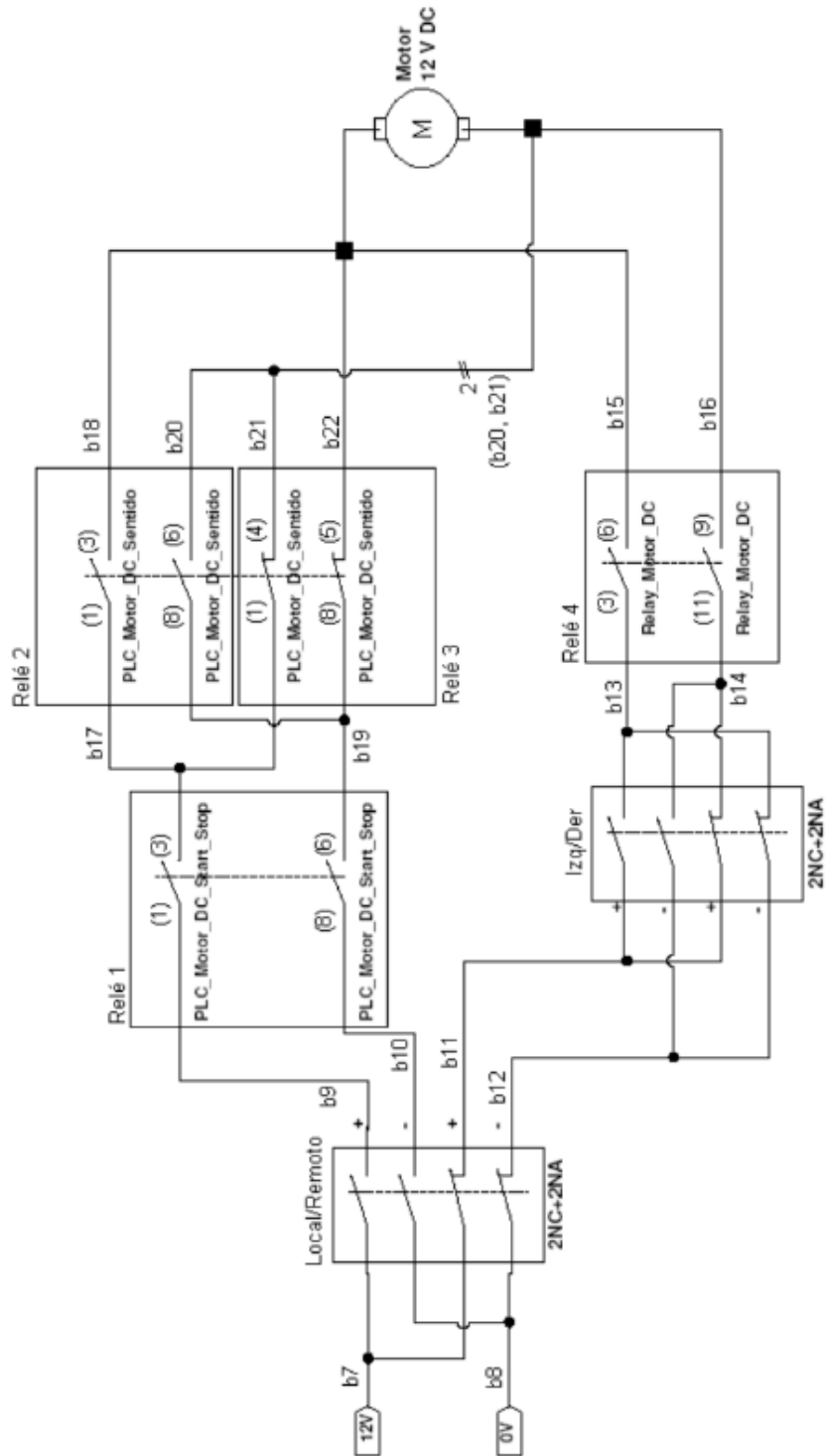


Figura 8.75: Plano eléctrico del motor DC

8.2. Proyecto nuevo y agregar dispositivos en TIA Portal

Lo primero que se debe hacer es abrir el programa TIA Portal. Una vez que se ha abierto se debe seguir **Iniciar** -> **Crear proyecto**, donde hay que asignar un nombre al proyecto y escoger una ruta para guardarlo. Finalmente se debe dar click en **Crear**.

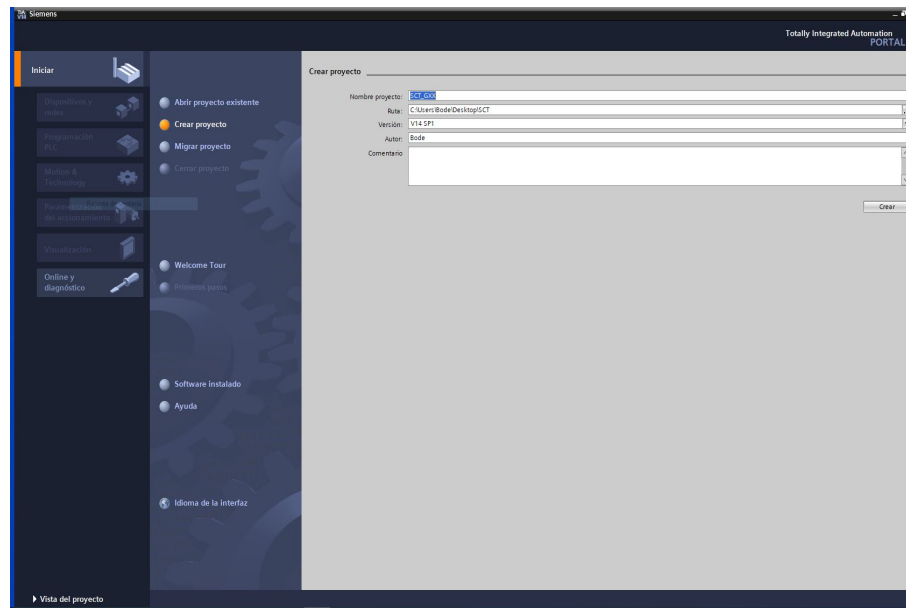


Figura 8.76: Crear proyecto en TIA Portal

Luego, se abrirá una nueva ventana, donde hay que dar click en **Vista del proyecto** para seguir la configuración. Se continua agregando los dispositivos de la red PROFIBUS.

8.2.1. Dispositivos en la red PROFIBUS

Para agregar la CPU, en el árbol del proyecto se debe dar click en **Agregar dispositivo** y en la librería de los Controladores se debe escoger la **CPU 315-2DP** que es la CPU que se utiliza para este proyecto y en específico es la **6ES7 315-2AG10-0AB0**. Una vez que se selecciona se da click en **Aceptar**. Al agregar el dispositivo, se crea un bloque **Main [OB1]** y varias ramificaciones para trabajar y configurar.

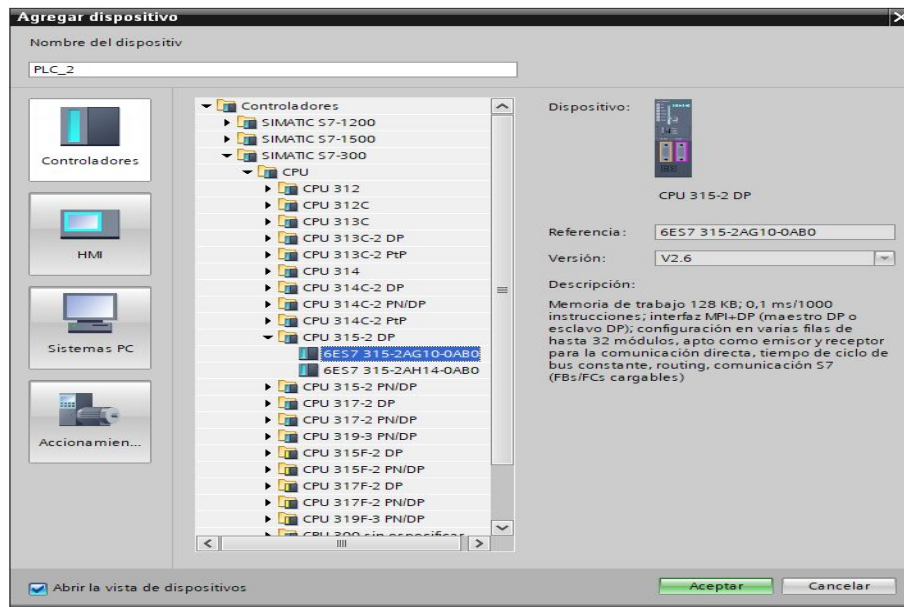


Figura 8.77: Agregar CPU

Al igual que para el PLC, en **Agregar dispositivo** se debe agregar la HMI KTP700 Basic DP (6AV2-123-2GA03-0AX0)(figura 8.78), la cual permite conexión MPI/PROFIBUS. La HMI utilizada de forma virtual es el mismo modelo de la HMI que se encuentra en el laboratorio, salvo que esta última es para conexiones con Ethernet.

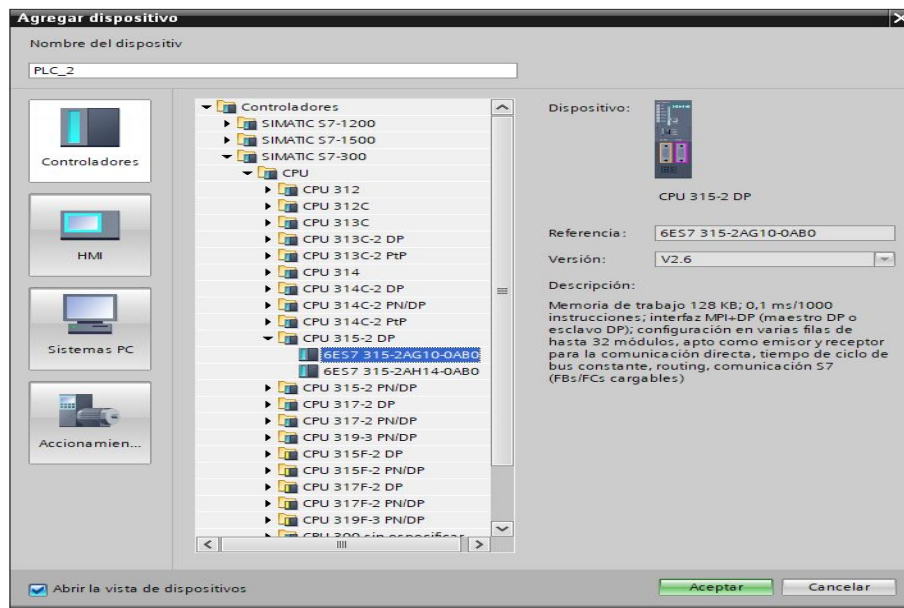
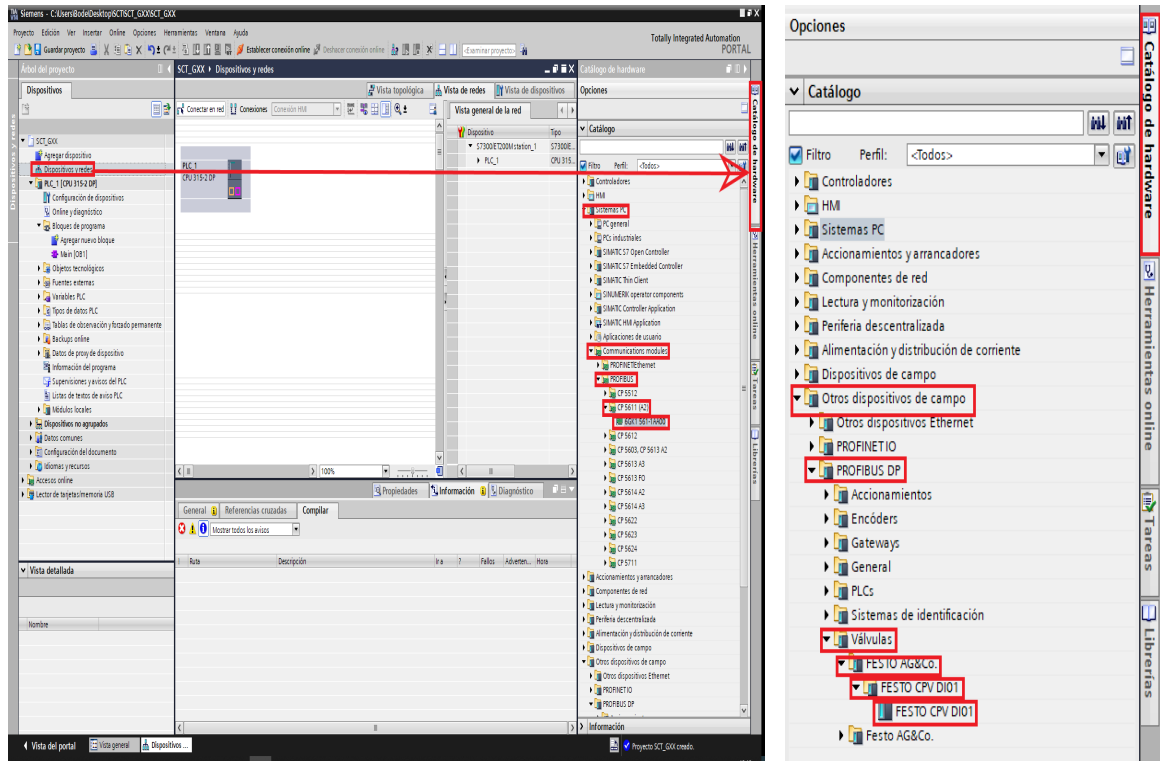


Figura 8.78: Agregar HMI

Para continuar con la configuración de la red, se deben agregar también el **Terminal de Válvulas** y la **Tarjeta CP 5611**.

Para agregarlos se debe ir a **Dispositivos y redes** en el árbol del proyecto. Al costado superior derecho de la pantalla hay que expandir la pestaña **Catálogo de hardware**. Para agregar la tarjeta se debe realizar los siguientes pasos: **Sistemas PC -> Communications modules -> PROFIBUS -> CP 5611(A2) -> 6GK1 561-1AA00**. En el caso del Terminal de Válvulas: **Otros dispositivos de campo -> Válvulas -> FESTO AG&Co. -> FESTO CPV DI01**. Esto se puede ver en la figura



(a) Tarjeta CP 5611

(b) Terminal de Válvulas

Figura 8.79: Agregar Tarjeta CP 5611 y Terminal de Válvulas

Agregar Archivo GSD En caso de que el Terminal de Válvulas no aparezca, se debe agregar manualmente a TIA Portal, esto debido a que no es de marca Siemens, si no de FESTO. Para ello se debe seguir los siguientes pasos:

- Descargar o tener el archivo GSD (General Station Description) del Terminal de Válvulas

a utilizar.

- En TIA Portal realizar lo siguiente: **Opciones -> Administrar archivos de descripción de dispositivos -> examinar hacia carpeta donde se encuentra el GSD -> Aceptar -> Instalar.**

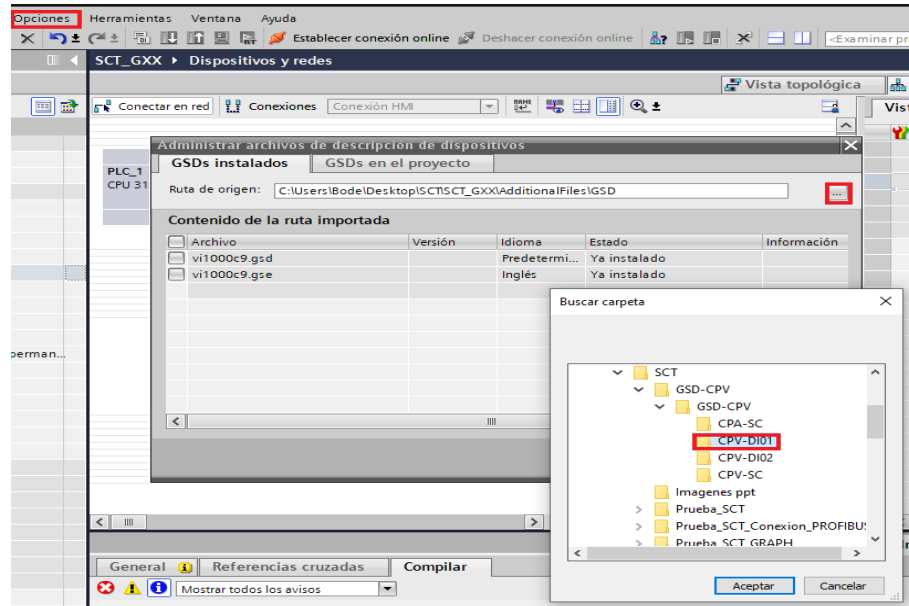


Figura 8.80: Agregar GSD Terminal de Válvulas

Finalmente se tiene la red que se puede ver en 5.51, con las siguientes configuraciones:

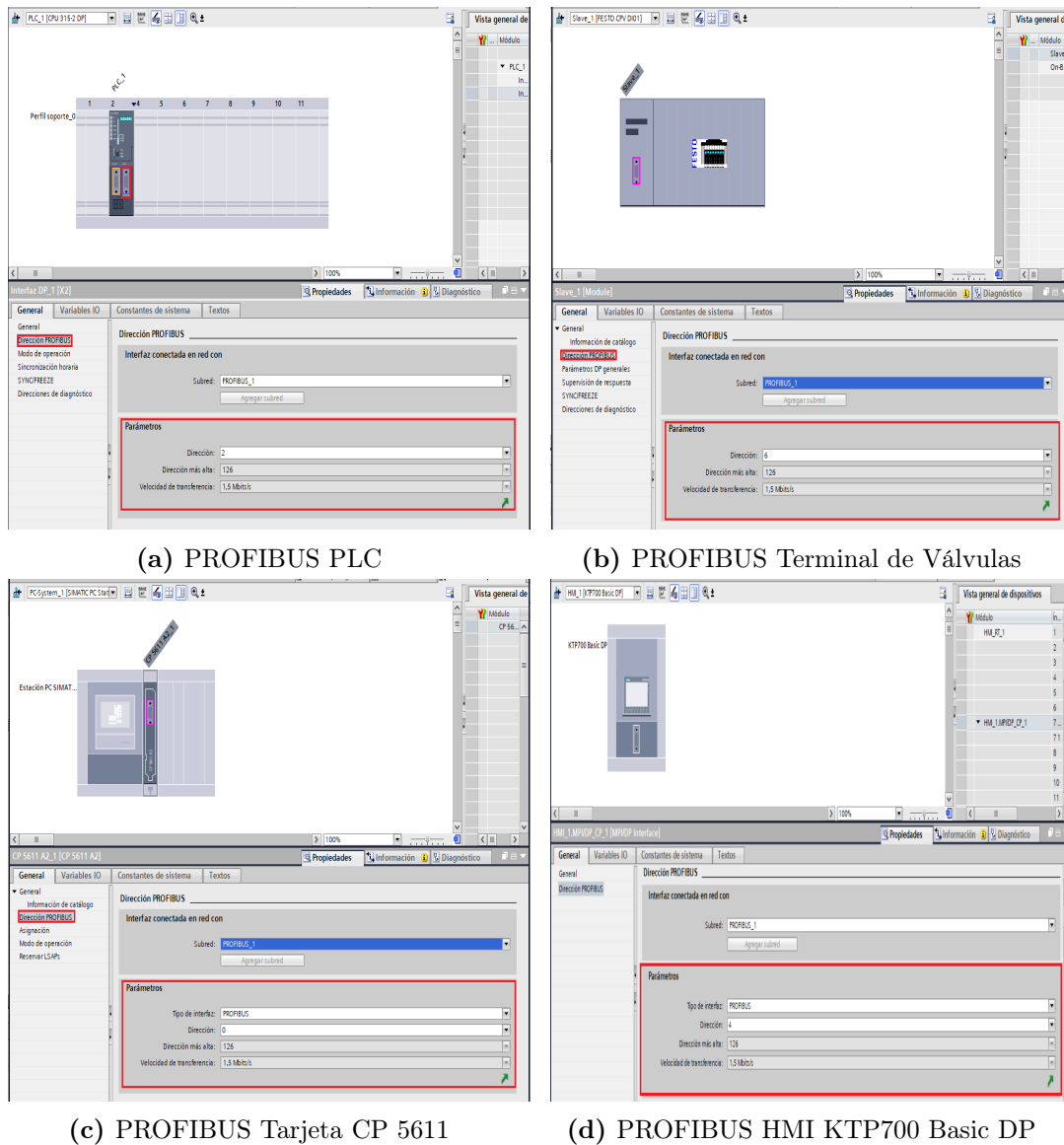


Figura 8.81: Configuraciones para la red PROFIBUS

8.2.2. Módulos PLC

Ahora se agrega al rack (bastidor o backplane), los módulos que el PLC tiene instalado. Se debe explorar en el menú de la derecha, en catálogo de hardware, para encontrar el módulo exacto (fijarse en el código único) y arrastrarlo hasta el bastidor.

Un dato importante es la ubicación que pueden tomar los módulos en el bastidor virtual. La CPU está asignada a la posición 2 del bastidor, lo cual no se debe confundir con la dirección

PROFIBUS asignada a la CPU. Por convención la fuente o PS se le asigna la posición 1, a la CPU la 2, la 3 al módulo de interfaz de I/O remota (no disponible, por lo que se deja libre), y desde la 4 en adelante a los módulos de E/S y de comunicación en el orden que corresponda. Hay que tener presente lo anterior al agregar los módulos. Por lo que los módulos quedan como:

- PS 307 2A
- CPU 315-2DP
- SM 323 DI16/DO16xDC24V
- SM 334 AI4/AO2x12bit

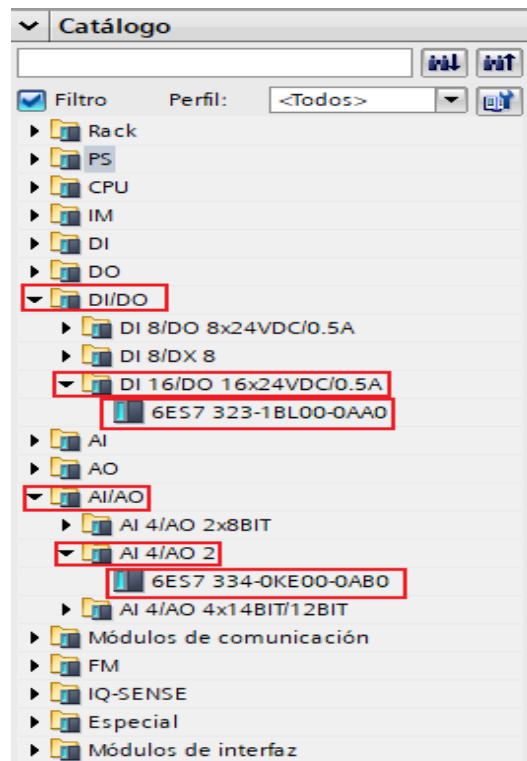


Figura 8.82: Agregar módulos

Se aclara que el módulo SM 334 AI4/AO2x12bit se agrega debido a que está presente en el PLC físico, pero no se utiliza. Finalmente, el PLC queda como en la figura 5.50

8.3. Programación Automatización en S7-Graph

Para iniciar la programación se debe crear un bloque de función, en este caso en lenguaje Graph. Para ello en el árbol del proyecto dr debe ir a **Bloques de programa -> Agregar nuevo bloque**. Luego se abrirá una ventana como la que muestra la figura 8.83. Hay que dar click en **Bloque de función** y escoger el lenguaje **Graph**, para luego dar en **Aceptar**.

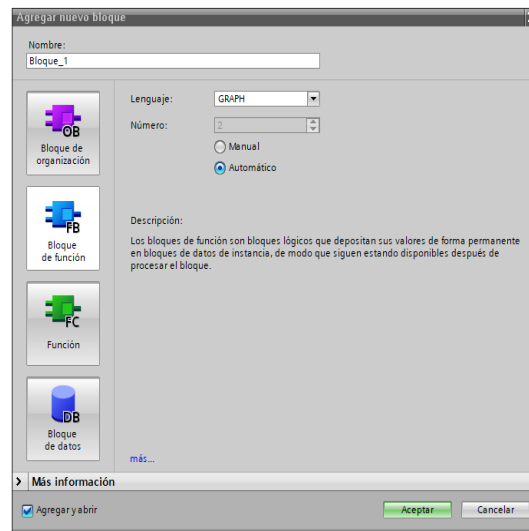
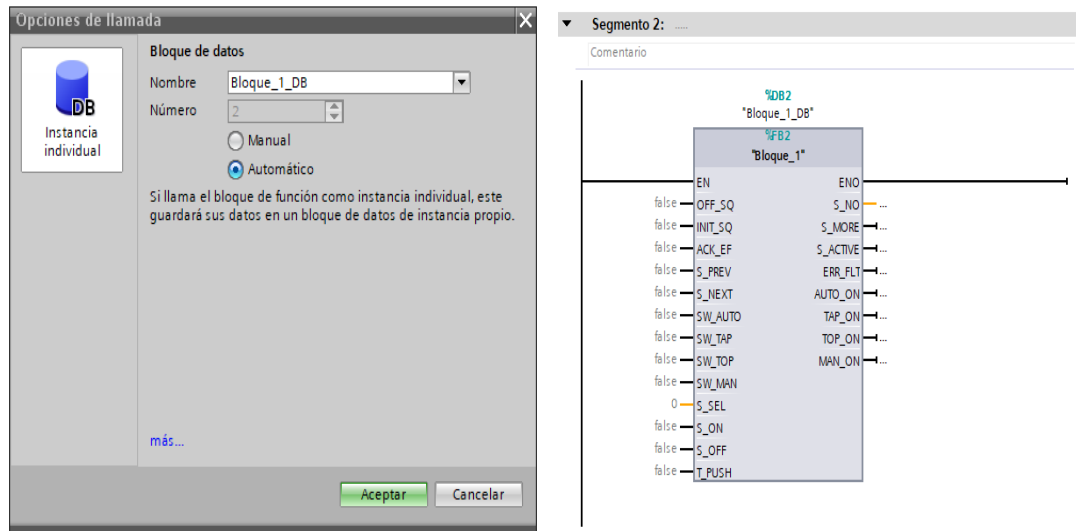


Figura 8.83: Crear bloque de función con lenguaje graph

Una vez creado el bloque de función, para acceder a él desde el programa principal, hay que arrastrar el bloque desde el árbol del proyecto a un segmento del Main. Aquí se abrirá una ventana para crear un bloque de datos, asociado al bloque de función graph. La figura 8.84 muestra la ventana para crear el bloque de datos y como queda el Main con el bloque de llamada para interactuar con el bloque de función graph.

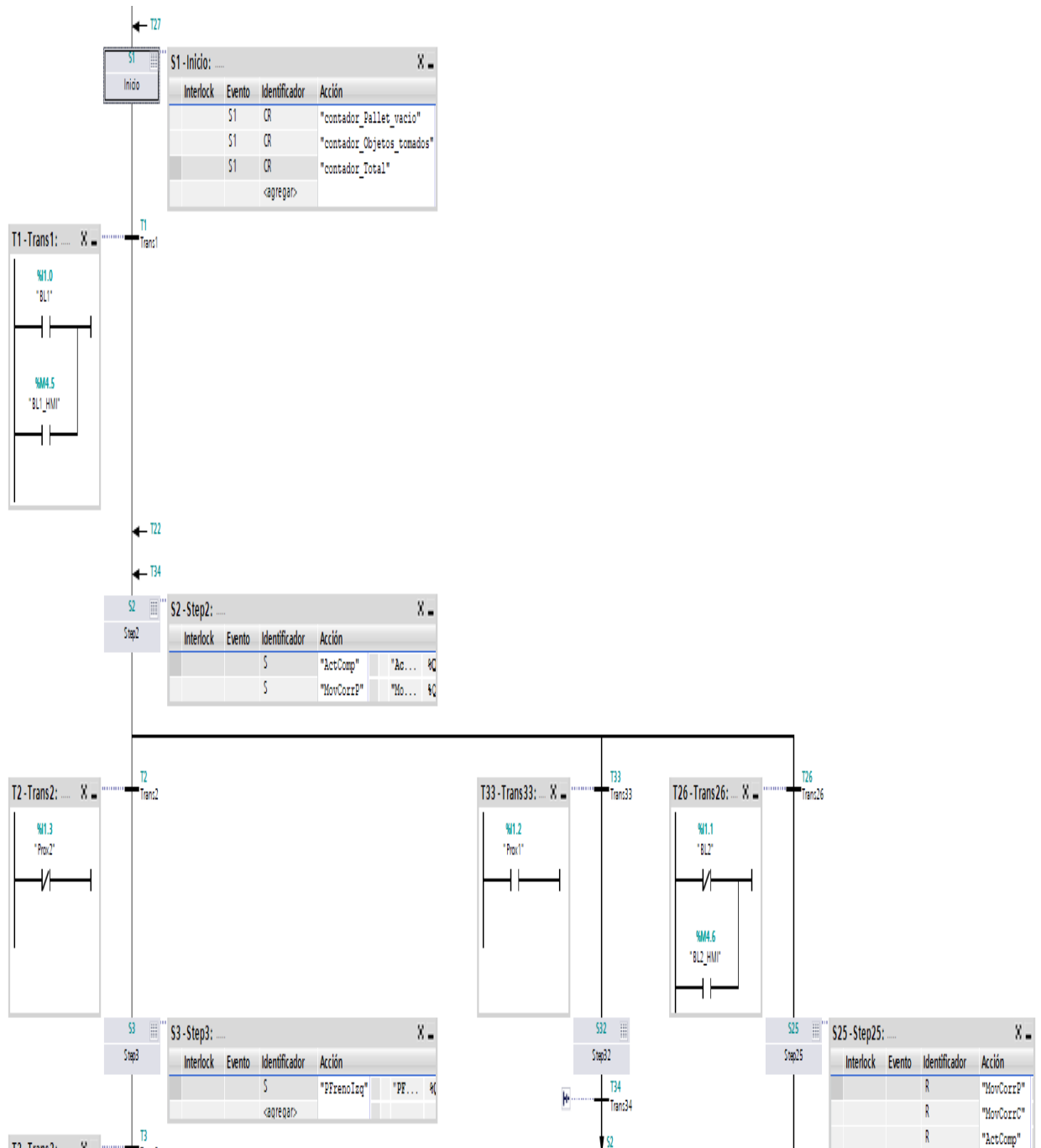


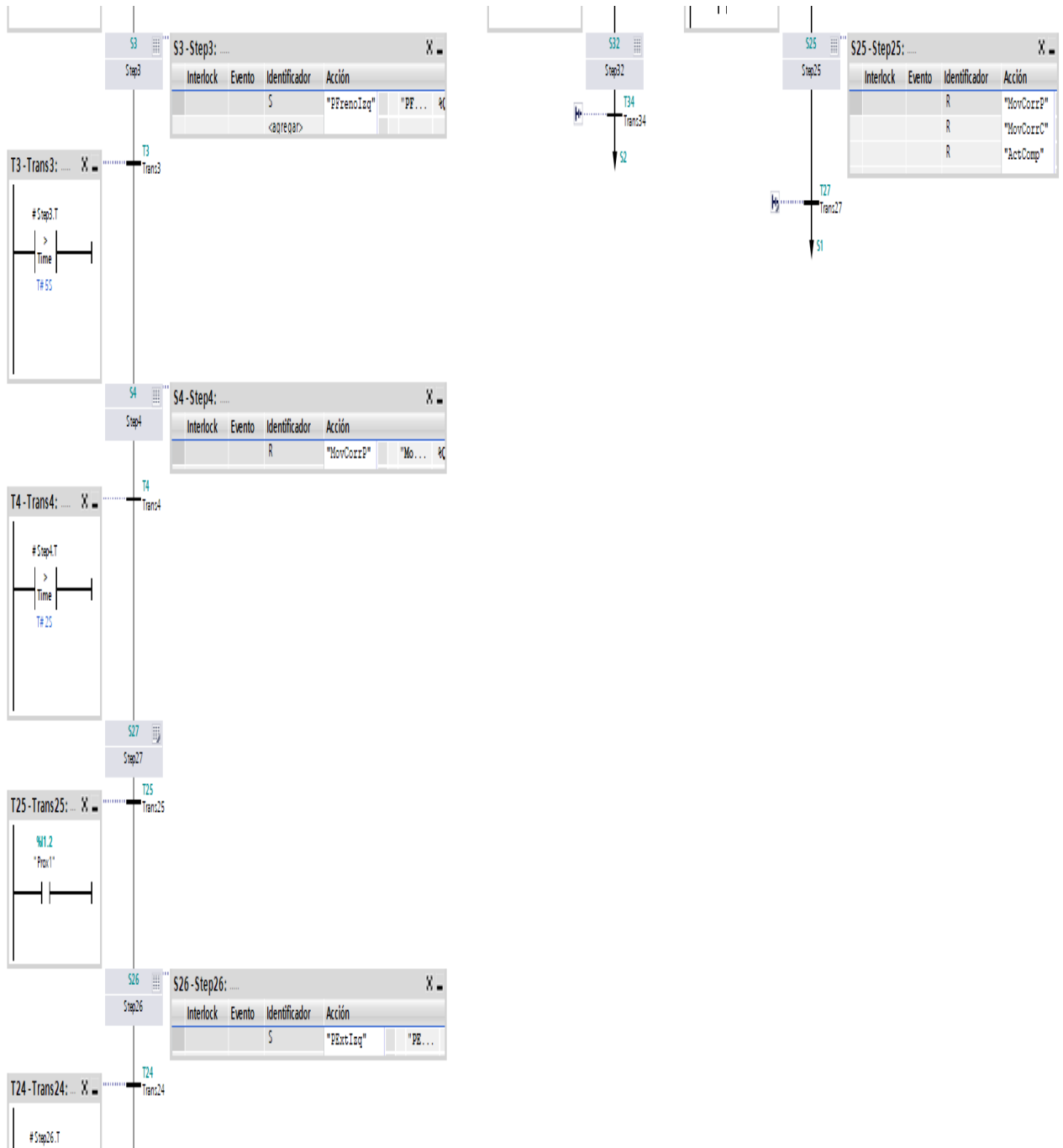
(a) Bloque de datos

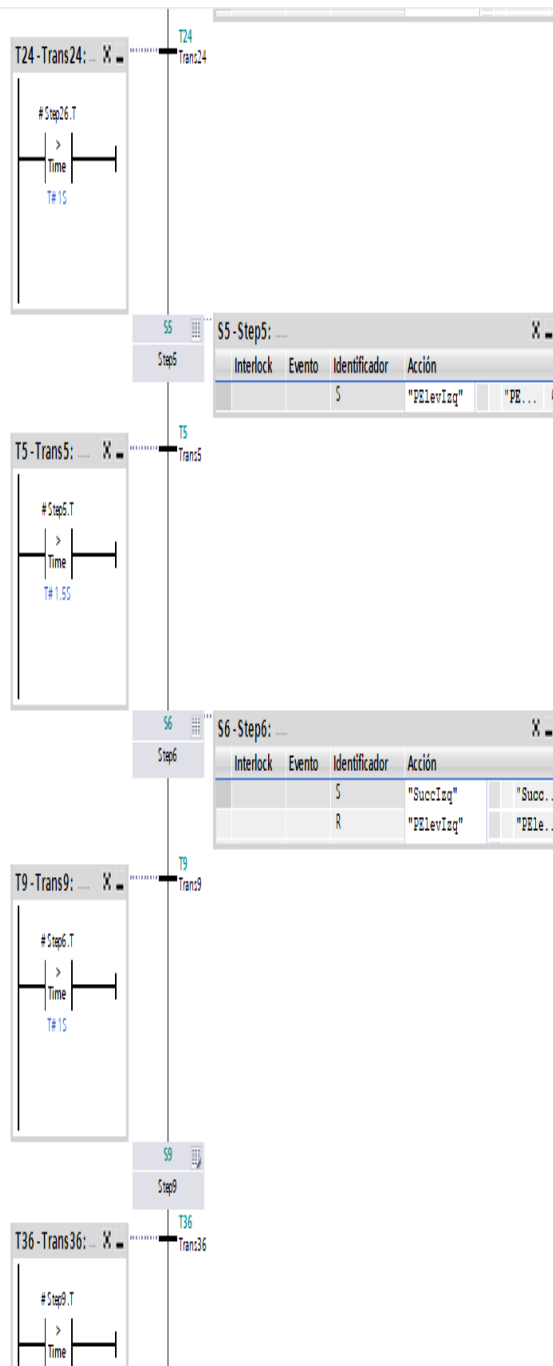
(b) Main

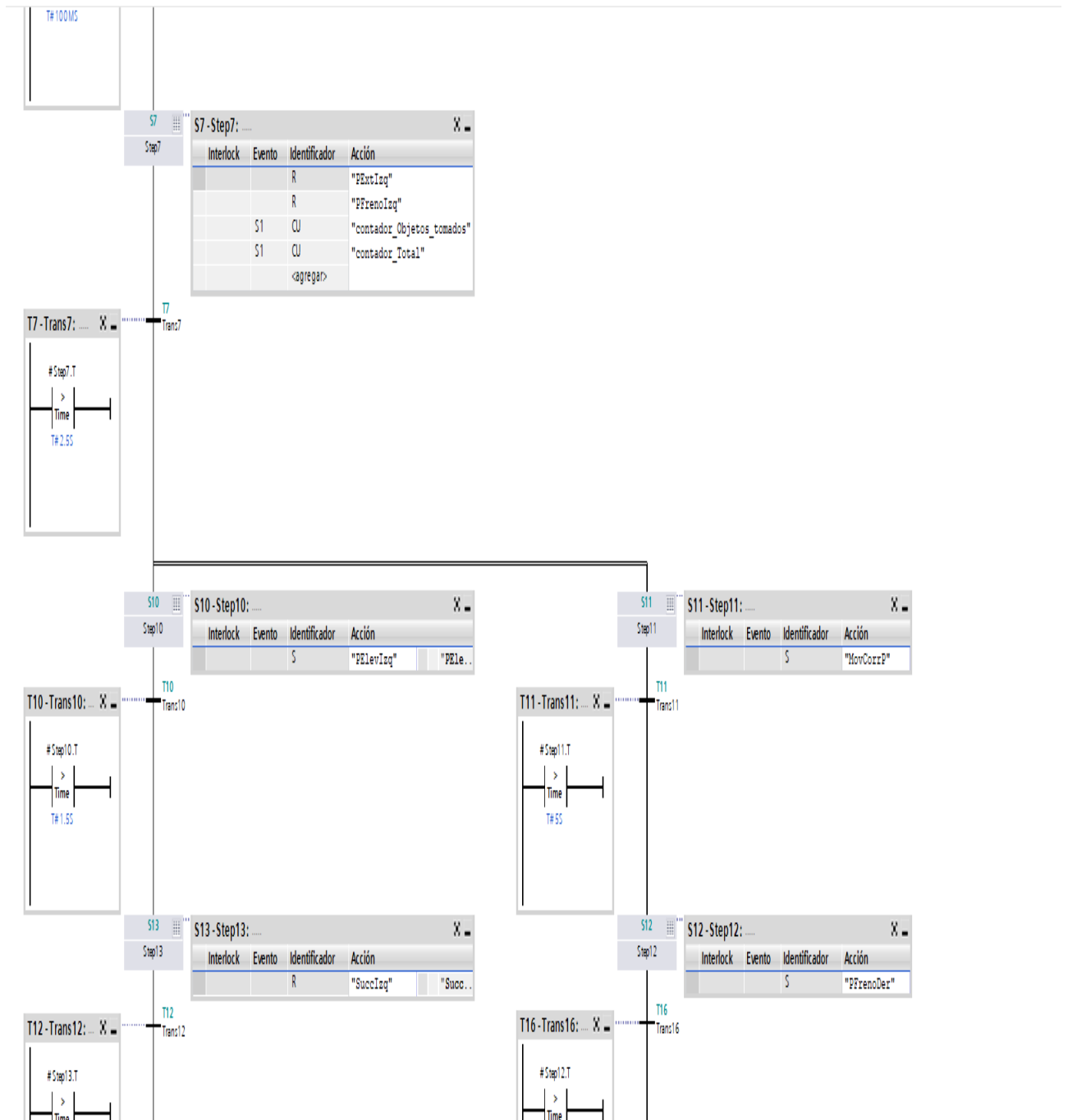
Figura 8.84: Creación bloque de datos y bloque de llamada en Main

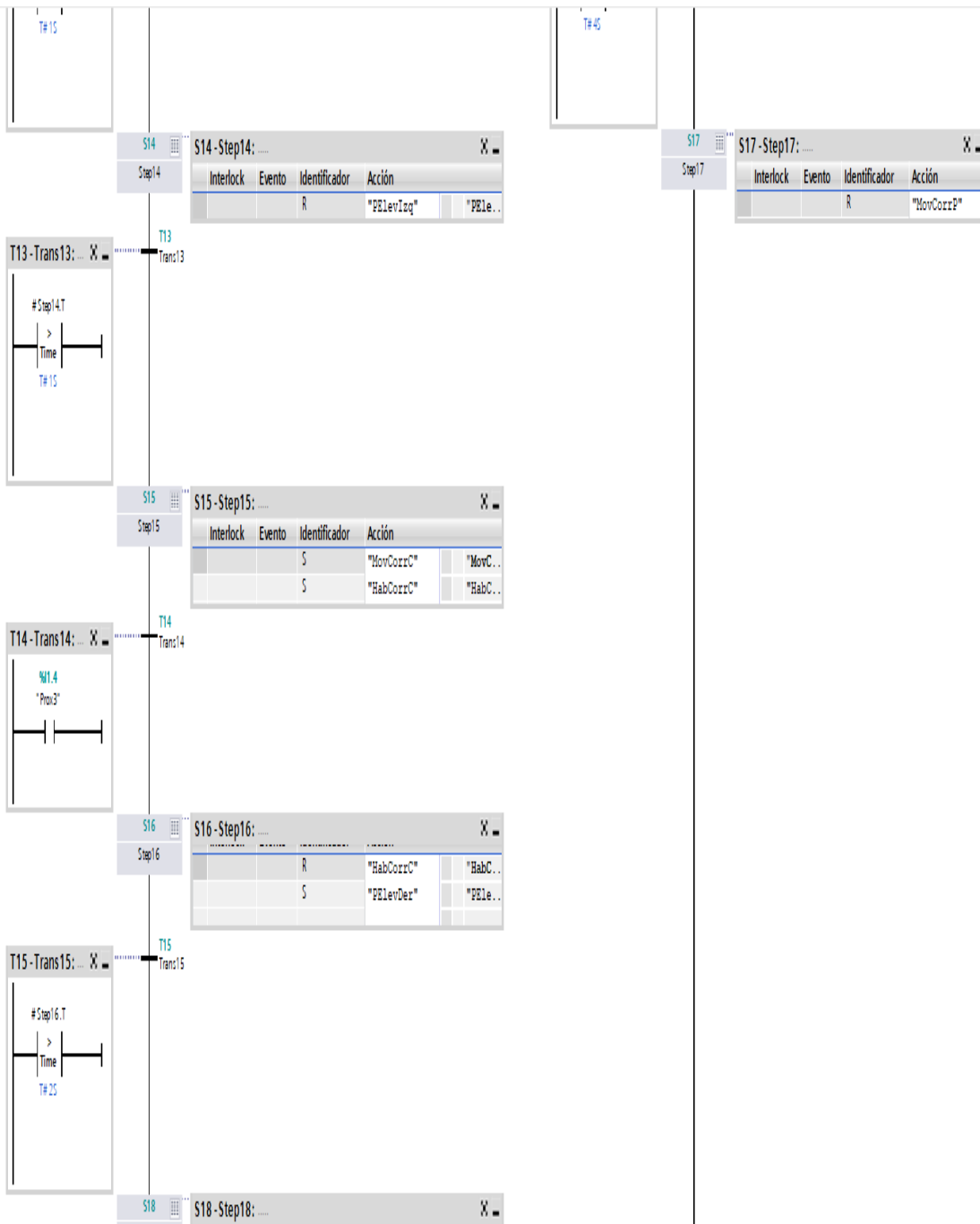
Luego se tiene la programación realizada en el bloque de función con lenguaje graph.

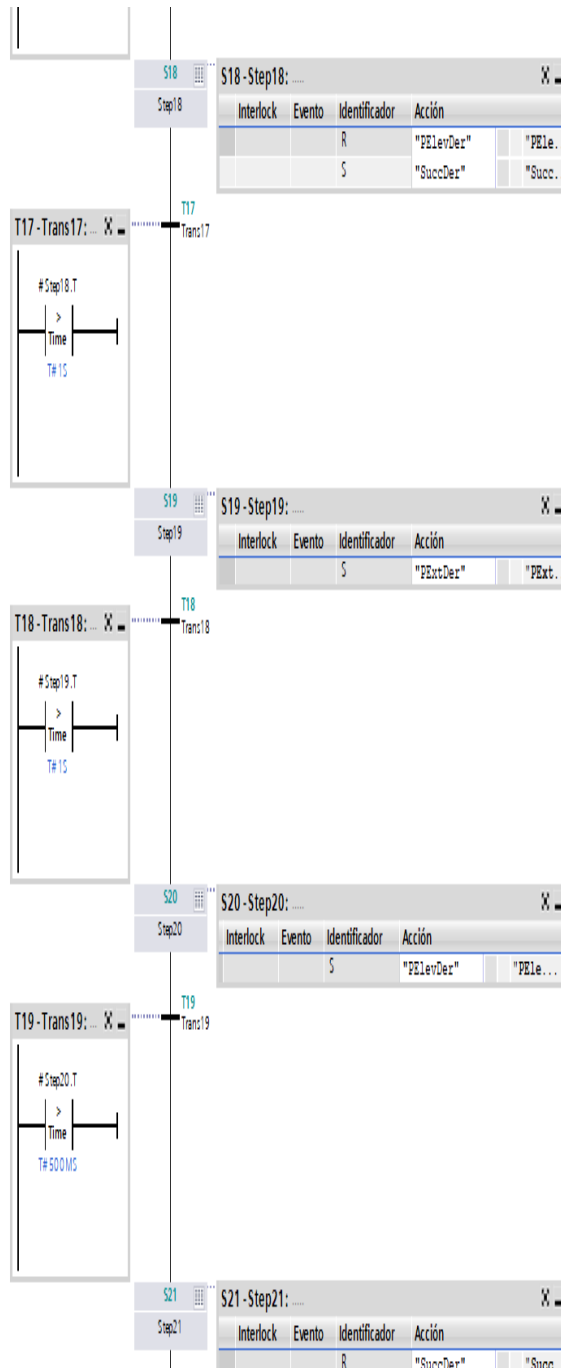














8.4. Tablas de variables PLC y HMI

| Variables PLC | | | | | | | |
|---------------|--------------------------|------------------------|---------------|-----------|---------|-------------------------------------|-------------------------------------|
| | Nombre | Tabla de variables | Tipo de datos | Dirección | Rema... | Acces... | Visibl... |
| | BL1 | Tabla de variabl... | Bool | %I1.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | BL2 | Tabla de variables e.. | Bool | %I1.1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | Prox1 | Tabla de variables e.. | Bool | %I1.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | Prox2 | Tabla de variables e.. | Bool | %I1.3 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | Prox3 | Tabla de variables e.. | Bool | %I1.4 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | MovCorrP | Tabla de variables e.. | Bool | %Q0.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | ActComp | Tabla de variables e.. | Bool | %Q0.1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | HabCorrC | Tabla de variables e.. | Bool | %Q0.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | MovCorrC | Tabla de variables e.. | Bool | %Q0.3 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PFrenolzq | Variables_PROFIBUS | Bool | %Q2.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PElevlzq | Variables_PROFIBUS | Bool | %Q2.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PExtlzq | Variables_PROFIBUS | Bool | %Q2.4 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PFrenoDer | Variables_PROFIBUS | Bool | %Q2.6 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PElevDer | Variables_PROFIBUS | Bool | %Q3.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PExtDer | Variables_PROFIBUS | Bool | %Q3.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | SuccIzq | Variables_PROFIBUS | Bool | %Q3.4 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | SuccDer | Variables_PROFIBUS | Bool | %Q3.6 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | contador_Pallet_vacio | Tabla de variables e.. | Counter | %C0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | contador_Objetos_tomados | Tabla de variables e.. | Counter | %C1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | BL1_HMI | Tabla de variables e.. | Bool | %M4.5 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | BL2_HMI | Tabla de variables e.. | Bool | %M4.6 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | contador_Total | Tabla de variables e.. | Counter | %C2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | Inicio | Tabla de variables e.. | Bool | %M5.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <Agregar> | | | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Figura 8.85: Tabla de variables PLC

| Variables HMI | | | | | | |
|-------------------------|-----------------------------|---------------|---------------------|----------------|--------------------------|-----------|
| Nombre | Tabla de variables | Tipo de datos | Conexión | Nombre del PLC | Variable PLC | Dirección |
| ActComp | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | ActComp | %Q0.1 |
| BL1 | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | BL1 | %I1.0 |
| BL1_HMI | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | BL1_HMI | %M4.5 |
| BL2 | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | BL2 | %I1.1 |
| BL2_HMI | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | <No definido> | %M4.6 |
| Contador_pallet_totales | Tabla de variables estándar | Counter | HM_Conexión_1 | PLC_1 | contador_Total | %C2 |
| HabCorrC | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | HabCorrC | %Q0.2 |
| Inicio | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | Inicio | %M5.0 |
| MovCorrC | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | MovCorrC | %Q0.3 |
| MovCorrP | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | MovCorrP | %Q0.0 |
| N_Objetos_tomados | Tabla de variables estándar | Counter | HM_Conexión_1 | PLC_1 | contador_Objetos_toma... | %C1 |
| N_pallet_vacios | Tabla de variables estándar | Counter | HM_Conexión_1 | PLC_1 | contador_Pallet_vacio | %C0 |
| Número_imagen_variable | Tabla de variables estándar | UInt | <Variable intern... | | <No definido> | |
| PElevDer | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | PElevDer | %Q3.0 |
| PElevIzq | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | PElevIzq | %Q2.2 |
| PExtDer | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | PExtDer | %Q3.2 |
| PExtIzq | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | PExtIzq | %Q2.4 |
| PFrenoDer | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | PFrenoDer | %Q2.6 |
| PFrenoIzq | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | PFrenoIzq | %Q2.0 |
| Prox1 | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | Prox1 | %I1.2 |
| Prox2 | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | Prox2 | %I1.3 |
| Prox3 | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | Prox3 | %I1.4 |
| SuccDer | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | SuccDer | %Q3.6 |
| SuccIzq | Tabla de variables estándar | Bool | HM_Conexión_1 | PLC_1 | SuccIzq | %Q3.4 |
| <Agregar> | | | | | | |

Figura 8.86: Tabla de variables HMI