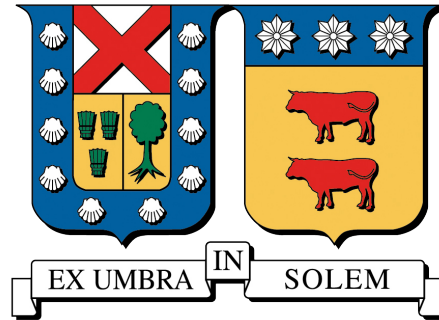


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**SIMULACIÓN, AUTOMATIZACIÓN Y PUESTA EN
MARCHA VIRTUAL DE UN SISTEMA DE
MANUFACTURA FLEXIBLE**

FRANCESCA SOFÍA DASATI CORREA

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL ELECTRÓNICA**

PROFESOR GUÍA : MANUEL OLIVARES SALINAS
PROFESOR CORREFERENTE : MAURICIO SOLÍS HERNANDEZ

FEBRERO 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Simulación, Automatización y Puesta en Marcha Virtual de un Sistema de Manufactura Flexible

Nombre del candidato(a): Francesca Dasati Correa

Carrera / Grado: Ingeniería Civil Electrónica

Campus: Casa Central Valparaíso ; **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Manuel Olivares Salinas, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 22/09/2025 ; Firma: Manuel Olivares S.

Estudiante o Candidato(a):

Fecha: 22/09/2025 ; Firma: F. Dasati

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

A mi familia, por su confianza y apoyo constante, que me han dado la fortaleza para superar cada desafío.

A mis amigos, por su apoyo incondicional y por hacer de estos años universitarios una experiencia única y llena de alegrías. Gracias por los momentos compartidos y por su paciencia infinita.

Dedicatoria

A mis padres, Viviana y Carlos, por su apoyo incondicional y por ser mis modelos a seguir, gracias por creer en mí.
Y a mi abuela, Herminia, por su amor eterno y resiliencia.

Resumen

Este proyecto desarrolla una simulación dinámica 3D en tiempo real de un Sistema de Manufactura Flexible (FMS, por sus siglas en inglés) disponible en el Centro Integrado de Manufactura y Automatización (CIMA), parte del Laboratorio de Control Industrial del Departamento de Electrónica de la Universidad Técnica Federico Santa María.

El objetivo principal es diseñar, programar y validar la automatización del sistema mediante un entorno de control virtual que replique el comportamiento del sistema físico. Para ello, se implementa un esquema de validación basado en *software-in-the-loop* (SIL), utilizando CIROS Studio en conjunto con un PLC virtual (*Control Win V3* de CODESYS), conectados a través del protocolo OPC.

La simulación integra el control lógico programado de las estaciones de transferencia y movimiento de pallets, así como la automatización de rutinas *pick and place* ejecutadas por los robots KUKA KR15 y KR6. La programación del PLC se realiza en lenguaje SFC y la programación de los robots en lenguaje KRL. Además, se implementa un sistema HMI para la supervisión y el control en tiempo real del sistema.

El resultado de este trabajo permite el desarrollo y la prueba de estrategias de automatización y control sin necesidad de utilizar el sistema físico, reduciendo riesgos y optimizando recursos. De este modo, es posible validar la puesta en marcha de la automatización programada y robotizada, y una interfaz HMI de supervisión del FMS de forma previa a su implementación física.

Palabras Clave: CIROS Studio; Automatización industrial; Robots KUKA; OPC; CODESYS; Flexible Manufacturing System (FMS)

Abstract

This project develops a real-time 3D dynamic simulation of a Flexible Manufacturing System (FMS) available at the Integrated Manufacturing and Automation Center (CIMA), part of the Industrial Control Laboratory of the Department of Electronics at Universidad Técnica Federico Santa María.

The main objective is to design, program, and validate the system's automation through a virtual control environment that replicates the behavior of the physical system. To achieve this, a validation scheme based on software-in-the-loop (SIL) is implemented, using CIROS Studio together with a virtual PLC (Control Win V3 from CODESYS), connected via the OPC protocol.

The simulation integrates the programmed logic control of the pallet transfer and movement stations, as well as the automation of pick and place routines executed by KUKA KR15 and KR6 robots. The PLC programming is done in SFC language, and the robot programming is carried out in KRL language. In addition, an HMI system is implemented for real-time monitoring and control of the system.

The outcome of this work enables the development and testing of automation and control strategies without the need to use the physical system, reducing risks and optimizing resources. In this way, it is possible to validate the deployment of the programmed and robotic automation, along with an HMI supervision interface for the FMS, prior to its physical implementation.

Keywords: CIROS Studio; Industrial automation; KUKA robots; OPC; CODESYS; Flexible Manufacturing System (FMS)

Glosario

CIROS Studio Es una plataforma de desarrollo que permite crear modelos de simulación en 3D para la automatización en tiempo real de sistemas de manufactura robotizada.

CODESYS Del inglés *Controller Development System*. Corresponde a una plataforma de software para automatización industrial. El núcleo de la plataforma es la herramienta de programación siguiendo la norma IEC 61131-3.

FMS Sistema de Manufactura Flexible en inglés.

GRAFCET Del francés *Grphe Fonctionnel de Commande Etape Transition*. Corresponde a un diagrama de estado con etapas y transiciones que describe la automatización de un sistema mediante una secuencia lógica de funcionamiento.

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

KRL En inglés (KUKA Robot Language) Lenguaje de programación utilizado para controlar robots industriales KUKA. Permite programar movimientos, secuencias de trabajo y la interacción con otros dispositivos en sistemas automatizados.

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

PLC En inglés Programmable Logic Controller. Dispositivo electrónico diseñado para automatizar procesos industriales.

Índice general

Agradecimientos	I
Dedicatoria	II
Resumen	III
Abstract	IV
Glosario	V
1. Introducción	7
1.1. Problema a resolver y objetivos	7
1.1.1. Objetivo general	8
1.1.2. Objetivos específicos	8
1.2. Requerimientos funcionales	8
1.3. Estado del arte	10
1.3.1. Sistemas didácticos de manufactura flexible	10
1.3.2. Alternativas de simulación 3D para FMS	10
1.3.3. Modelos CIROS en el Laboratorio de Control Industrial	11
1.3.4. Estrategias de integración y supervisión SIL y HIL	12
1.3.5. Estudios y proyectos de virtualización de sistemas con CIROS Studio	13
2. Componentes del FMS en CIMA	14
2.1. Cinta transportadora accionada eléctricamente	15
2.2. Pallet de 6 posiciones y cubos	16
2.3. Estaciones de transferencia electroneumáticas	17
2.4. Estaciones de procesamiento estacionarias	18
2.5. Robot KUKA KR15 y KUKA KR6	19
2.6. PLC (Programmable Logic Controller)	21
2.6.1. Tabla de entradas y salidas de la FMS	24
3. Modelamiento CIROS 3D de FMS	26
3.1. Estructura del software de modelamiento FMS CIROS	27
3.1.1. Librerías	27
3.1.2. Explorador del modelo	29
3.1.3. Asignación entradas y salidas	30
3.1.4. Opciones del modelo	31
3.1.5. Project Management	32

3.2.	Integración CIROS mediante servidor OPC Config	33
3.2.1.	Integración con sistemas de automatización PLC CODESYS	33
3.2.2.	Integración sistemas de supervisión HMI CODESYS	34
4.	Modelo CIROS del FMS e integración con PLC y HMI virtuales	35
4.1.	Componentes virtuales	35
4.1.1.	Cinta transportadora	35
4.1.2.	Pallet de 6 posiciones y cubos	36
4.1.3.	Estaciones de transferencia electroneumáticas	37
4.1.4.	Sensores	40
4.1.5.	Panel de operaciones	41
4.1.6.	Robot KUKA KR15	41
4.1.7.	Robot KUKA KR6	42
4.1.7.1.	Programa KRL de prueba	43
4.2.	Integración OPC	45
4.2.1.	Configuración servidor OPCConfig	45
4.2.2.	PLC virtual	46
4.2.3.	Tabla de entradas y salidas del modelo CIROS	50
4.2.3.1.	Programa SFC de prueba	52
4.2.4.	Pantalla virtual	53
4.2.4.1.	Programa HMI de prueba	54
5.	Experiencia virtual “Automatización de un Sistema de Manufactura Flexible”	56
5.1.	Requisitos funcionales de automatización FMS	57
5.1.1.	Requisitos automatización programada	57
5.1.2.	Requisitos automatización robotizada	59
5.2.	Programa KRL de automatización robotizada de la FMS	59
5.2.1.	Rutina KUKA KR6	60
5.2.2.	Rutina KUKA KR15	61
5.2.3.	Validación de cumplimiento de requisitos de automatización robotizada de la FMS	62
5.3.	Programa SFC de automatización programada de la FMS	64
5.3.1.	Validación de cumplimiento de requisitos de automatización programada de la FMS	70
5.4.	Programa HMI de supervisión del sistema integrado	72
5.4.1.	Validación de cumplimiento de requisitos de supervisión del sistema integrado	73
6.	Análisis de resultados y conclusiones	74
6.1.	Aportes	75
6.2.	Trabajos futuros	76
	Bibliografía	76
A.	Anexos	80

A.1. Diagrama eléctrico de conexiones de entrada	80
A.2. Diagrama eléctrico de conexiones de salida	83
A.3. Código rutina robot KR15	85
A.4. Lista de posiciones robot KR15	88
A.5. Código rutina robot KR6	89
A.6. Lista de posiciones robot KR6	91
A.7. Variables globales PLC virtual	92
A.8. Diagramas GRAFCET CODESYS	93
A.9. Especificaciones módulos PLC Glofa	95

Índice de tablas

2.1.	Datos técnicos robots.	21
2.2.	Características técnicas de la CPU.	22
2.3.	Características técnicas del módulo G4I-D22A	23
2.4.	Características técnicas del módulo G4Q-RYDA.	23
2.5.	Tabla entradas PLC - Salidas Estación de Transferencia KR6.	24
2.6.	Tabla entradas PLC - Salidas Estación de Transferencia KR15.	24
2.7.	Tabla Salidas PLC – Entradas Estación de Transferencia KR6.	25
2.8.	Tabla Salidas PLC – Entradas Estación de Transferencia KR15.	25
4.1.	Entradas Lógicas PLC.	50
4.2.	Salidas Lógicas PLC.	51
4.3.	Tabla Entradas PLC – Programa de prueba.	53
4.4.	Tabla Salidas PLC – Programa de prueba.	53
5.1.	Entradas/Salidas robot KUKA.	60

Índice de figuras

2.1. Cinta transporta en CIMA.	15
2.2. Motor disponible en CIMA.	15
2.3. Pallet de posiciones.	16
2.4. Accionamiento estación transferencia.	17
2.5. Actuadores.	18
2.6. Estación de trabajo.	18
2.7. Robot KUKA KR6.	19
2.8. Robot KUKA KR15 con séptimo eje.	20
2.9. Accesorio de garra neumática.	20
2.10. PLC LG GM4 ubicado en CIMA.	21
3.1. Acceso librerías del modelo	27
3.2. Ejemplos de modelos utilizados	28
3.3. Árbol de navegación	29
3.4. Ejemplo de personalización	30
3.5. Ejemplo asignación entradas y salidas.	30
3.6. Ejemplos de ajustes del modelo	31
3.7. Agregar nuevo proyecto	32
3.8. Project Management	32
3.9. Implementación del FMS virtual.	33
4.1. Simulación cinta transportadora.	36
4.2. Pallet con cubos.	37
4.3. Mecanismo elevador estación transferencia.	38
4.4. Mecanismo elevador estación transferencia.	39
4.5. Mecanismo elevador con geometría.	39
4.6. Sensores virtuales.	40
4.7. Panel de operaciones virtual en CIROS.	41
4.8. Robot KUKA KR15 con eje lineal.	42
4.9. Robot KUKA KR6.	42
4.10. Movimientos robot en rutina de prueba.	44
4.11. Configuraciones OPC.	45
4.12. Configuración conexión.	46
4.13. Configuración proyecto en CODESYS.	47
4.14. CODESYS Control Win V3 SysTray.	47

4.15. Configuración dispositivo en CODESYS.	48
4.16. OPC CLient en CIROS Studio.	48
4.17. Variables compartidas.	49
4.18. Ejemplo de conexión.	49
4.19. Programa SFC de prueba.	52
4.20. Componentes disponibles visualización.	54
4.21. Componentes disponibles visualización.	55
5.1. Modelo completo del FMS en CIROS.	57
5.2. Ecuación de movimiento.	58
5.3. Diagrama de estados secuencia estación de transferencia.	58
5.4. Posiciones cubos iniciales y finales estación KR6.	60
5.5. Posiciones cubos iniciales y finales estación KR15.	61
5.6. KR6 realizando rutina <i>pick and place</i>	62
5.7. KR15 realizando rutina <i>pick and place</i>	63
5.8. Diagrama GRAFCET FMS-MAIN.	66
5.9. Diagrama GRAFCET General.	68
5.10. Activación de SFC Pause.	69
5.11. Funcionamiento LED de aproximación.	70
5.12. Elevador estación KR15 en simulación.	71
5.13. HMI.	72
5.14. Prueba HMI modo manual.	73
5.15. Prueba HMI imagen del sistema.	73
6.1. Diagrama conexiones subsistemas.	77
A.1. Diagrama eléctrico conexiones correa transportadora.	80
A.2. Diagrama eléctrico conexiones robot KR6.	81
A.3. Diagrama eléctrico conexiones robot KR6.	82
A.4. Diagrama eléctrico salidas PLC.	83
A.5. Diagrama eléctrico estaciones de transferencia.	84
A.6. Lista de posiciones robot KR15.	88
A.7. Lista de posiciones robot KR6.	91
A.8. Variables globales utilizadas en CODESYS.	92
A.9. Diagrama GRAFCET general.	93
A.10. Diagrama GRAFCET MAIN.	94
A.11. Configuraciones internas módulo D22A.	95
A.12. Configuraciones internas módulo RY2A.	95

1 | Introducción

1.1. Problema a resolver y objetivos

En la actualidad, la industria requiere de sistemas de manufactura no solamente flexibles y eficientes, sino también seguros y económicamente viables para mantener competitividad. Los Sistemas de Manufactura Flexible (abreviado como FMS por sus siglas en inglés) consisten en un conjunto de máquinas interconectadas mediante un sistema automatizado de manejo y almacenamiento de materiales, controlado por un sistema computarizado. Se clasifican como flexibles debido a su capacidad para procesar diversas tareas en múltiples estaciones de trabajo de manera simultánea [1].

Estas estaciones de trabajo suelen estar automatizadas con robots industriales y sistemas computarizados, lo que permite responder con agilidad a variaciones en la demanda. Sin embargo, la implementación directa de estos sistemas conlleva posibles riesgos operacionales y costos asociados a la puesta en marcha física.

El presente proyecto se enfoca en la virtualización de un sistema de manufactura flexible, específicamente dirigido a un entorno de laboratorio universitario, con el propósito de optimizar su implementación. La creación de un entorno virtual posibilita la simulación dinámica de procesos industriales, permitiendo su evaluación, prueba y análisis antes de la aplicación física. Esto contribuye a la reducción de riesgos de daño en los equipos, así como a la minimización de costos derivados de fallos o errores durante las etapas de desarrollo y programación.

Al virtualizar el FMS, se busca diseñar, validar y optimizar el sistema en un entorno controlado, previo a su implementación física. Esto permite que los estudiantes trabajen de manera segura, comprendiendo mejor el funcionamiento del sistema y todos sus componentes asociados antes de interactuar con el sistema físico. Además, la versión virtual del FMS permite el trabajo simultáneo de múltiples grupos de estudiantes.

A continuación se presentan los objetivos generales y específicos.

1.1.1. Objetivo general

Diseñar y poner en marcha virtualmente una simulación dinámica en 3D de la experiencia de automatización de un Sistema de Manufactura Flexible, que se realiza en el Centro Integrado de Manufactura y Automatización (CIMA) como parte del Laboratorio de Control Industrial, utilizando el software CIROS Studio y su integración con un PLC virtual.

1.1.2. Objetivos específicos

- Configurar y desarrollar el FMS virtual 3D integrando componentes mecánicos y neumáticos, funcional y visualmente equivalentes al FMS físico utilizando el software CIROS Studio.
- Conectar mediante OPC, el FMS virtual con un sistema de control virtual similar al control físico.
- Diseñar la automatización programada haciendo uso de diagramas GRAFCET, y validarla por simulación mediante la conexión del FMS virtual con el sistema de control virtual.
- Documentar el proceso para su uso en futuras referencias y experiencias del Laboratorio de Control Industrial.

1.2. Requerimientos funcionales

El proyecto presenta los siguientes requerimientos funcionales correspondientes a la implementación virtual.

- El FMS debe tener conectividad OPC con un sistema de control virtual similar al físico en el Laboratorio de Control Industrial, incluyendo un PLC y un panel de operación local. Aunque este panel no existe físicamente, su funcionalidad debe ser simulada dentro del entorno virtual para permitir la interacción y control del sistema.

- La automatización del FMS virtual debe implementarse mediante un sistema de control basado en diagramas GRAFCET, asegurando la correcta secuencia de operaciones. Se debe garantizar la integración con los robots y cumplir con los requerimientos específicos establecidos por el Laboratorio de Control Industrial [2].
- Los robots KUKA KR15 y KUKA KR6 deben ejecutar de manera automática sus rutinas de movimiento utilizando lenguaje KRL. Se deben programar trayectorias y secuencias específicas para la manipulación de cubos dentro del FMS, asegurando una interacción sincronizada con el resto del sistema automatizado.
- El sistema debe contar con un sistema de supervisión que permita visualizar en tiempo real el estado de la secuencia, incluyendo el estado de sensores, componentes electropneumáticos y otros elementos del proceso. Esta supervisión debe realizarse mediante una interfaz HMI proporcionando datos relevantes para el monitoreo del sistema.

1.3. Estado del arte

1.3.1. Sistemas didácticos de manufactura flexible

En el artículo “*Teaching flexible manufacturing systems by using design and simulation program tools*” de Z.Kovacic [3], se exploran metodologías para la enseñanza de los Sistemas de Manufactura Flexible (FMS) en cursos universitarios, haciendo uso de herramientas de diseño y simulación. Se destaca cómo el uso de modelos virtuales facilita la visualización y comprensión de estos sistemas, cuya complejidad puede dificultar su estudio en entornos físicos.

El estudio hace referencia al uso de la herramienta FlexMan para la realización de simulaciones, la cual proporciona un procedimiento más ágil y menos complejo para la configuración de los parámetros de los FMS. Esta herramienta fue desarrollada por los propios autores del estudio con el propósito de mejorar el proceso de enseñanza.

Inicialmente, se requiere que los estudiantes utilicen FlexMan para diseñar el modelo virtual del sistema conforme a los requisitos establecidos. Una vez logrado el diseño y simulación, los estudiantes proceden a probar el sistema real utilizando las matrices desarrolladas anteriormente a lenguaje *Ladder* o Diagrama de Bloques Funcionales (FBD).

Finalmente, se comparan los resultados obtenidos en la simulación con los de la implementación física controlada por un PLC.

1.3.2. Alternativas de simulación 3D para FMS

Entre algunas de las alternativas de software disponibles para la simulación 3D de sistemas FMS se destacan AnyLogic y FactoryIO, ambos caracterizados por su gran flexibilidad en la personalización y diseño de modelos de simulación.

AnyLogic permite implementación lógica avanzada con programación en Java, además de poseer una biblioteca extensa de componentes. No obstante, su uso eficiente requiere conocimientos avanzados y su licencia tiene un costo elevado [4].

Por otro lado, FactoryIO se caracteriza por su facilidad de uso y una curva de aprendizaje rápida, lo que facilita el desarrollo de habilidades y la comprensión del sistema.

Además, este software está específicamente diseñado para la simulación de sistemas de automatización industrial [5, 6].

1.3.3. Modelos CIROS en el Laboratorio de Control Industrial

Actualmente el Laboratorio de Control Industrial cuenta con dos experiencias que hacen uso de CIROS Studio para su desarrollo. La primera corresponde a “Automatización de un Sistema de Transporte y Clasificación”, emplea una simulación para el control y supervisión virtual de un sistema de clasificación, compuesto por una correa transportadora principal, una correa de clasificación y dos estaciones de transferencia. En esta experiencia, la integración se realiza mediante un PLC Siemens S7-300, además de incluir un terminal de válvulas CPV DI01.

En el trabajo de Matías Herrera, "Simulación, automatización y puesta en marcha virtual de un sistema de transporte y clasificación" [7] se presenta una guía sobre el proceso de puesta en marcha virtual del sistema en CIROS Studio. Además se ofrece una descripción detallada de los distintos componentes de los software, presentes en sus librerías, junto con algunas configuraciones relevantes que deben considerarse al momento de realizar la simulación y su integración con un PLC virtual y componentes electroneumáticos.

La segunda experiencia corresponde a “Automatización Robotizada” [8], emplea un brazo articulado RV-2AJ de 5 grados de libertad, junto con un controlador RISC CR1-H3, un pallet y una bandeja de alimentación giratoria.

En este caso, CIROS Studio se utiliza para registrar las posiciones del robot y desarrollar un programa de traslado repetitivo de cubos entre el pallet estacionario y la bandeja giratoria en lenguaje *MELFA Basic IV*. Además, se incorpora una rutina de interrupción activada por un sensor de proximidad capacitivo.

1.3.4. Estrategias de integración y supervisión SIL y HIL

Los esquemas de validación *Software-in-the-Loop* (SIL) y *Hardware-in-the-Loop* (HIL) son técnicas utilizadas en la integración y validación de sistemas críticos y controladores. Cada enfoque cumple un propósito específico dentro del proceso de desarrollo y se emplean ampliamente en la industria automotriz, aunque también se aplican en otras áreas de la automatización industrial.

Una simulación *Software-in-the-Loop* (SIL) compila el código fuente y ejecuta el código como un proceso separado en una computadora. Al comparar los resultados de la simulación normal y la simulación SIL, se puede probar la equivalencia numérica del modelo y el código generado [9]. Este esquema permite probar y validar el software antes de su integración con el hardware físico, facilitando la detección de errores en etapas tempranas del desarrollo. Además, ofrece una alta capacidad de repetición en las pruebas y es fácilmente escalable, ya que no requiere modificaciones en el hardware. En el ámbito de la robótica industrial, SIL permite interpretar el código real del programa y generar movimientos exactos de los robots en un entorno simulado, optimizando el proceso de programación y prueba.

Las pruebas HIL son simulaciones en tiempo real con el hardware de control conectado a un entorno virtual en el que se simula el comportamiento del sistema real [10]. Este método permite probar la interacción del software con componentes físicos como sensores, actuadores y controladores, asegurando una validación más completa y precisa en comparación con SIL. Al incluir hardware real en la simulación, HIL permite evaluar el comportamiento del sistema en condiciones cercanas a la realidad, reduciendo costos y riesgos asociados con la validación en entornos físicos. Además, garantiza que el software interactúe correctamente con el hardware antes de su implementación final, aumentando la confiabilidad del sistema.

1.3.5. Estudios y proyectos de virtualización de sistemas con CIROS Studio

En el estudio “*Animation and Simulation Applications for Education*” de Elena Taborova [11], se investigan los fundamentos de simulación, animación y modelado aplicados a un sistema de transportadores de rodillos en 3D, desarrollado en el laboratorio de la Universidad de Ciencias Aplicadas de Saimaa. Para la simulación, se utilizó el software CIROS Studio, mientras que el control lógico del sistema fue programado en lenguaje *Ladder* utilizando SIMATIC Manager STEP7, integrándolo con un controlador lógico programable (PLC) virtual.

En esta investigación, se comparan y evalúan distintos programas basados en modelado y simulación, analizando sus características y aplicabilidad en entornos educativos e industriales. Además, se proponen mejoras para el sistema de transportadores y se plantean posibles aplicaciones futuras para optimizar su rendimiento y funcionalidad.

2 | Componentes del FMS en CIMA

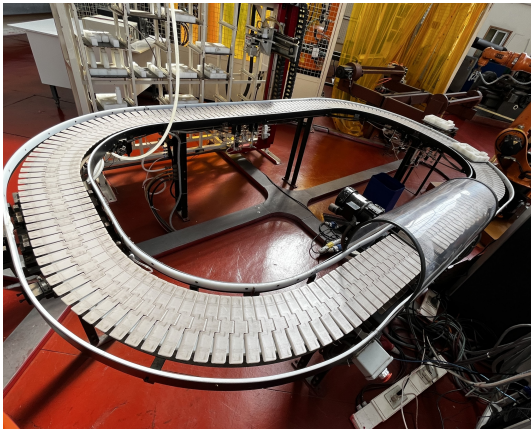
El Sistema de Manufactura Flexible (FMS) a virtualizar está compuesto por una cinta transportadora principal cerrada, diseñada para simular un entorno de manufactura industrial. Su principal función es transportar pallets entre dos estaciones de trabajo, donde se llevan a cabo operaciones automatizadas de transferencia y manipulación robotizada.

Cuando un pallet llega a la estación de transferencia de un robot, se activa un mecanismo automatizado de transferencia que lo desplaza hacia la estación de trabajo estacionaria correspondiente. En esta etapa, el robot ejecuta la rutina programada para el cambio de posición de los cubos dispuestos en el pallet de 6 posiciones.

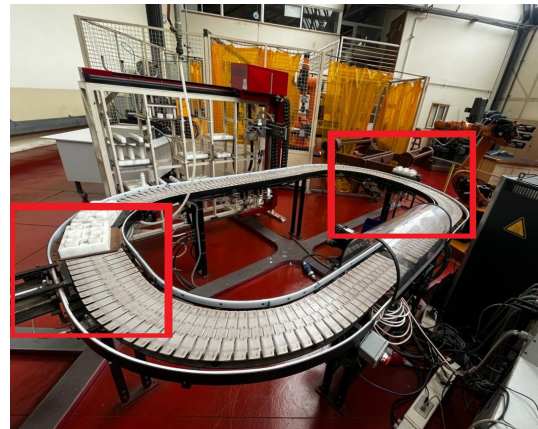
Una vez completada la tarea, el pallet es retornado a la cinta transportadora principal para continuar su recorrido a través del sistema. Este flujo continuo permite la simulación de escenarios industriales reales, facilitando el estudio de los procesos dentro de un entorno controlado.

2.1. Cinta transportadora accionada eléctricamente

La Figura 2.1a muestra la cinta transportadora ubicada en el CIMA. Por otro lado, en la Figura 2.1b se puede observar en los cuadros destacados en color rojo las posiciones de las estaciones de transferencia correspondientes a cada robot. En la derecha la del KUKA KR6 y a la izquierda del KUKA KR15.



(a) Cinta Transportadora.



(b) Estaciones de transferencia.

Figura 2.1: Cinta transporta en CIMA.

La correa de transporte es impulsada por un motor monofásico ($115V$ $60Hz$) *MAXI-TORQ Permanent Split Capacitor Gearmotor* que se muestra en la Figura 2.2, de corriente alterna. El cual posee una potencia de $1/6 - 1/7$ HP.



Figura 2.2: Motor disponible en CIMA.

2.2. Pallet de 6 posiciones y cubos

El pallet, como se muestra en la Figura 2.3 está compuesto por una placa de tecnyl de 6 ranuras distribuidas en 2 filas de 3 posiciones, y tiene la función de transportar estos cubos a lo largo de la cinta transportadora, para luego ser entregados a las estaciones de trabajo de cada robot. Las dimensiones del pallet son de 24 cm × 16 cm × 6.25 cm, cada ranura mide aproximadamente 2 cm × 4.5 cm × 4.5 cm para colocar los cubos. Cada cubo tiene dimensiones de 4 cm × 4 cm × 4 cm, lo que permite un ajuste adecuado dentro de las ranuras del pallet.

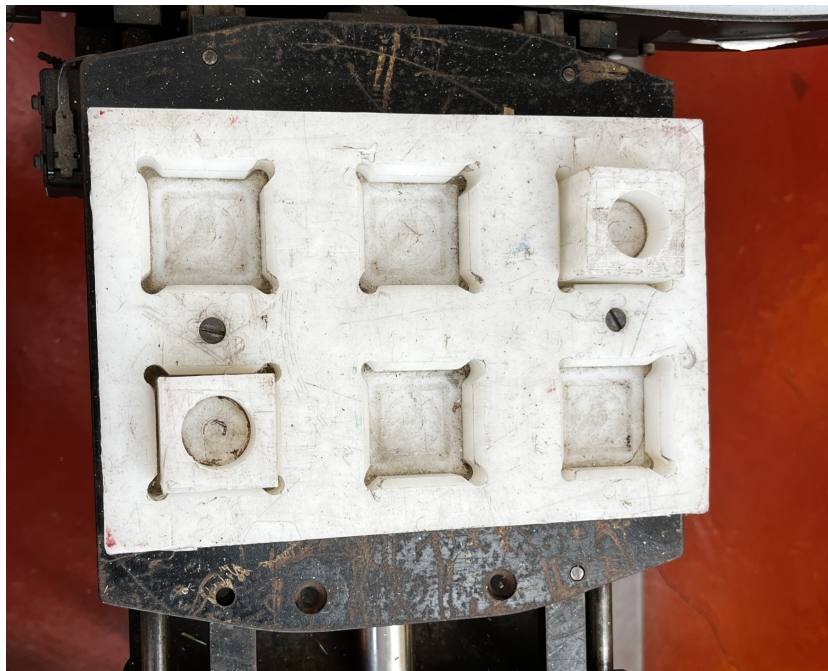


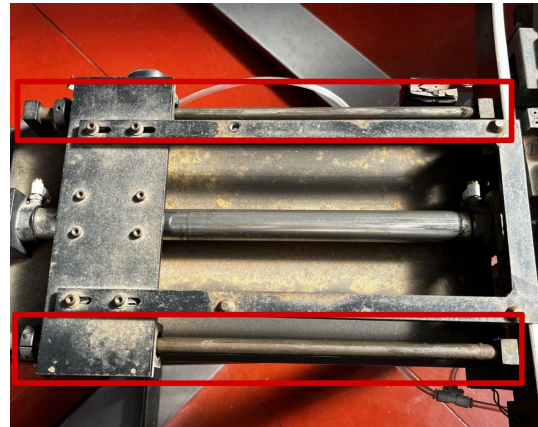
Figura 2.3: Pallet de posiciones.

2.3. Estaciones de transferencia electroneumáticas

El FMS físico está compuesto por dos estaciones de transferencia electroneumáticas. Cada estación dispone de 4 cilindros neumáticos, como se muestra en la Figura 2.4a con los cuadros destacados en verde, los cuales tienen la función de elevar el pallet con los cubos. Una vez levantado, el pallet es transportado horizontalmente mediante otros dos cilindros adicionales hasta la estación de trabajo estacionaria de cada robot, como se ilustra en la Figura 2.4b con los cuadros destacados en color rojo.



(a) Cilindros neumáticos.



(b) Bandeja de transferencia.

Figura 2.4: Accionamiento estación transferencia.

Las estaciones de transferencia están equipadas con 2 frenos frontales, los cuales tienen la función de detener el pallet durante su traslado. Además, cuentan con un freno trasero adicional controla el movimiento del pallet y, en caso de que otro pallet esté circulando por la cinta transportadora, evita que este ingrese al espacio del pallet anterior.

Esta disposición se ilustra en la Figura 2.5, donde los cuadros azules representan los frenos frontales, el cuadro verde muestra el freno de cola, y los cuadros destacados en rojo muestran los cilindros elevadores mencionados anteriormente en la Figura 2.4a.



Figura 2.5: Actuadores.

2.4. Estaciones de procesamiento estacionarias

Las estaciones de procesamiento cuentan con 2 sensores de fin de carrera que indican cuando la bandeja alcanza los límites tanto cuando entra y cuando sale de la estación de trabajo estacionaria, los cuales se muestran destacados por los recuadros rojos de la Figura 2.6.

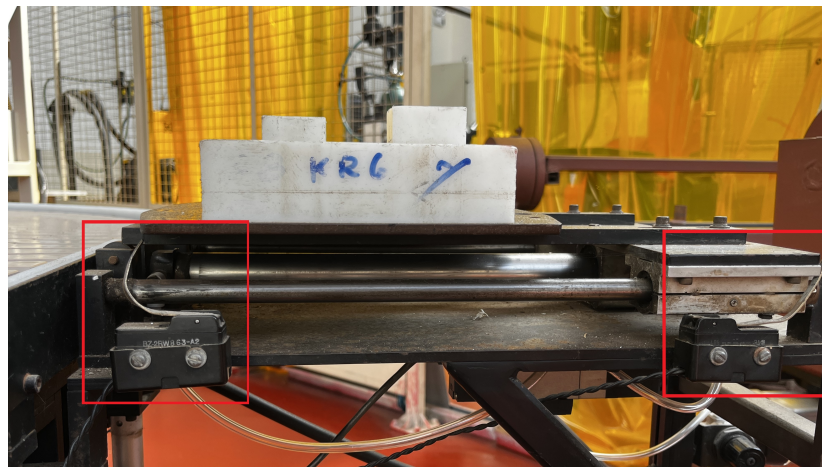


Figura 2.6: Estación de trabajo.

2.5. Robot KUKA KR15 y KUKA KR6

El FMS cuenta con 2 robots KUKA, un KR6 y un KR15 que cuenta con un séptimo eje lineal horizontal adicional, que permite el movimiento del robot de una estación de trabajo a otra. Como se muestra en las Figuras 2.7 y 2.8 respectivamente.



Figura 2.7: Robot KUKA KR6.



Figura 2.8: Robot KUKA KR15 con séptimo eje.

Ambos robots son de tipo brazo robótico de 6 ejes, y cuentan con un accesorio de garra neumática, como se muestra en la Figura 2.9.



Figura 2.9: Accesorio de garra neumática.

A continuación, en la Tabla 2.1 se presentan algunos datos técnicos, incluyendo los valores de carga útil nominal y la carga máxima total de ambos robots [12, 13].

Tabla 2.1: Datos técnicos robots.

Tipo	KR 6	KR 15
Carga útil nominal [kg]	6	15
Carga máxima total [kg]	36	25
Sistema de Accionamiento	Electromecánico, servomotores AC con mando transistorizado	

2.6. PLC (Programmable Logic Controller)

Para el control del Sistema de Manufactura Flexible, se utiliza un PLC de la marca LG, modelo GM4. Este sistema consta de una unidad central de procesamiento (CPU), dos fuentes de alimentación y dos módulos de entradas y salidas digitales.

La fuente de alimentación, modelo GM4-PA2A, convierte corriente alterna (AC) de 220 V a corriente continua (DC) de 24 V, con una capacidad de entrega de hasta 0.7 A. Además, se cuenta con un módulo de CPU GM4-CPUA que gestiona el control y la ejecución de las operaciones programadas.



(a)



(b)

Figura 2.10: PLC LG GM4 ubicado en CIMA.

A continuación, se presentan las características técnicas de la CPU [14]:

Tabla 2.2: Características técnicas de la CPU.

Especificaciones técnicas generales	
Dimensiones A x A x A	35mm x 135mm x 120mm
Peso aprox.	0.25 kg
Lenguaje de programación	Diagrama de escalera Instruction List Sequential Function Chart
Consumo de Corriente	130 mA
Memoria de programación	128K Byte
Máximo de puntos I/O	2048 puntos
Conector de Memoria	Si
Memoria de datos	32K Byte
Cantidad máxima de etapas de extensión	3
Número de bloques de programa	180 bloques
Modos de operación	RUN, STOP, PAUSE y DEBUG
Modos de reinicio	Cold, warm, hot

Este PLC también está equipado con 6 módulos de entradas digitales G4I-D22A y 6 módulos de salidas digitales G4Q-RY2A, cada uno con 16 puntos de conexión, lo que permite una integración flexible con sensores y actuadores. Además, se incluye una pantalla HMI (Interfaz Hombre-Máquina) modelo TOP-3MA para facilitar la interacción del usuario con el sistema.

Las características técnicas del módulo de entradas digitales G4I-D22A se detallan en la siguiente tabla [14]:

Tabla 2.3: Características técnicas del módulo G4I-D22A .

Especificaciones técnicas generales	
Peso aprox.	0.25 kg
Método de aislamiento	Optoacoplador
Cantidad de entradas	16 puntos
Voltaje nominal de entrada	12/24 VDC
Corriente nominal de entrada	5/11 mA
Consumo corriente interna	70 mA
Impedancia de entrada aprox.	2.2k Ω
Señalización de estado	LED se enciende en el estado ON de la entrada
Tiempo de respuesta ON-OFF	10 mseg o menos
Tiempo de respuesta OFF-ON	10 mseg o menos
Configuración común de terminales	8 puntos/común

El módulo de salidas digitales G4Q-RY2A presenta las siguientes especificaciones técnicas [14]:

Tabla 2.4: Características técnicas del módulo G4Q-RYDA.

Especificaciones técnicas generales	
Peso aprox.	0.31 kg
Método de aislamiento	Optoacoplador
Cantidad de salidas	16 puntos
Tensión de carga nominal	24 VDC
Consumo corriente interna	100 mA
Carga mínima (voltaje/corriente)	5 VDC / 1mA
Carga máxima (voltaje/corriente)	250 VAC, 125 VDC
Señalización de estado	LED se enciende en el estado ON de la salida
Frecuencia máxima de conmutación	3600 veces por hora
Tiempo de respuesta ON-OFF	12 mseg o menos
Tiempo de respuesta OFF-ON	10 mseg o menos
Configuración común de terminales	8 puntos/común

2.6.1. Tabla de entradas y salidas de la FMS

A continuación se muestran las tablas que detallan la asignación de entradas y salidas del PLC en relación con las estaciones de transferencia de ambos robots KR6 y KR15 en CIMA.

Las Tablas 2.5 y 2.6 muestran las entradas del PLC, que corresponden a las señales enviadas por sensores presentes en la estación de transferencia. Estas señales permiten detectar la presencia de piezas, posición de la bandeja y el estado de los sensores de proximidad.

Tabla 2.5: Tabla entradas PLC - Salidas Estación de Transferencia KR6.

Input	Terminal Módulo	Componente	Señal
I0.3.0	1	Sensor de Contacto	Detecta Pieza
I0.3.1	2	Sensor de Contacto	Bandeja Adelante
I0.3.2	3	Sensor de Contacto	Bandeja Atrás
I1.1.12	2	Sensor proximidad magnético	Sensor Arriba
I1.1.13	3	Sensor proximidad magnético	Sensor Abajo

Tabla 2.6: Tabla entradas PLC - Salidas Estación de Transferencia KR15.

Input	Terminal Módulo	Componente	Señal
I0.3.3	4	Sensor de Contacto	Detecta Pieza
I0.3.4	5	Sensor de Contacto	Bandeja Adelante
I0.3.5	6	Sensor de Contacto	Bandeja Atrás
I1.1.8	8	Sensor proximidad magnético	Sensor Arriba
I1.1.9	9	Sensor proximidad magnético	Sensor Abajo

Por otro lado, en las Tablas 2.7 y 2.8 muestran las salidas del PLC, las cuales controlan los actuadores de cada estación. A través de estas señales el PLC activa el freno de cola, mueve la bandeja y opera el elevador de esta última.

Tabla 2.7: Tabla Salidas PLC – Entradas Estación de Transferencia KR6.

Output	Reles Tablero	Señal
Q0.4.0	R01	Freno
Q0.4.1	R02	Control de Cola
Q0.4.2	R03	Elevador
Q0.4.3	R04	Sacar Bandeja
Q0.4.4	R05	Entrar Bandeja

Tabla 2.8: Tabla Salidas PLC – Entradas Estación de Transferencia KR15.

Output	Reles Tablero	Señal
Q0.4.5	R06	Freno
Q0.4.6	R07	Control de Cola
Q0.4.7	R08	Elevador
Q0.4.8	R09	Sacar Bandeja
Q0.4.9	R10	Entrar Bandeja

3 | Modelamiento CIROS 3D de FMS

CIROS Studio es una plataforma de desarrollo avanzada probada en el ámbito industrial, que permite crear modelos de simulación en 3D para la técnica de automatización. Puede ser aplicada de manera versátil en un entorno de aprendizaje, al poder ampliarse con mecanismos y materiales didácticos.

La plataforma facilita la implementación de diversos componentes esenciales para la simulación de sistemas de manufactura flexible, como cintas transportadoras con velocidad variable, pallets, curvas de 90 y 180 grados, además de elevadores con sistema de transferencia horizontal y robots de diversas marcas. Todos estos componentes son personalizables en cuanto a tamaño, posición y características de adaptación a las necesidades específicas de cada simulación.

Además, CIROS Studio permite la integración de entradas y salidas de manera sencilla, lo que permite el control de la simulación tanto mediante un controlador virtual presente en la plataforma, como a través de dispositivos externos. Un ejemplo de esto es la conexión mediante el protocolo OPC, que permite la operación con sistemas externos, como lo es un PLC virtual en CODESYS.

La plataforma también soporta la creación de escenarios complejos de automatización, con la capacidad de simular el comportamiento de sistemas en condiciones dinámicas, lo que la convierte en una herramienta importante para el desarrollo de soluciones industriales.

3.1. Estructura del software de modelamiento FMS CIROS

A continuación, se presentan los principales componentes disponibles en las librerías del software, detallando su función dentro de la simulación del FMS. Además, se describe la organización del modelo creado,

También se destacan configuraciones clave que deben considerarse al momento de desarrollar una simulación, como la asignación de entradas y salidas, así como la personalización de la ubicación y dimensiones de los objetos. Estos aspectos permiten ajustar el modelo a los requerimientos específicos del sistema a simular, garantizando mayor precisión y realismo en la representación del entorno virtual.

3.1.1. Librerías

Las librerías de modelos en CIROS proporcionan un amplio conjunto de elementos de modelado listos para usar. Para acceder al cuadro de diálogo **Model Libraries**, se debe ir a **Modeling > Model Libraries**, como se muestra en la Figura 3.1.

Para agregar un modelo, se debe seleccionar en el cuadro de diálogo y presionar el botón **Add**.

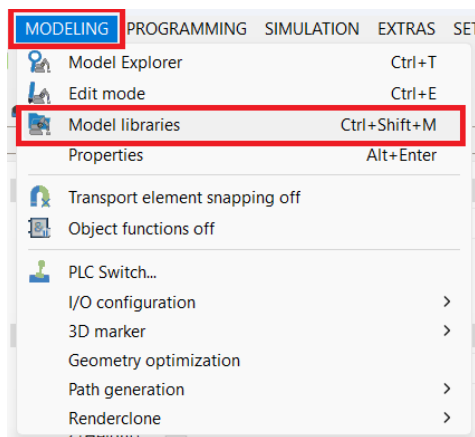
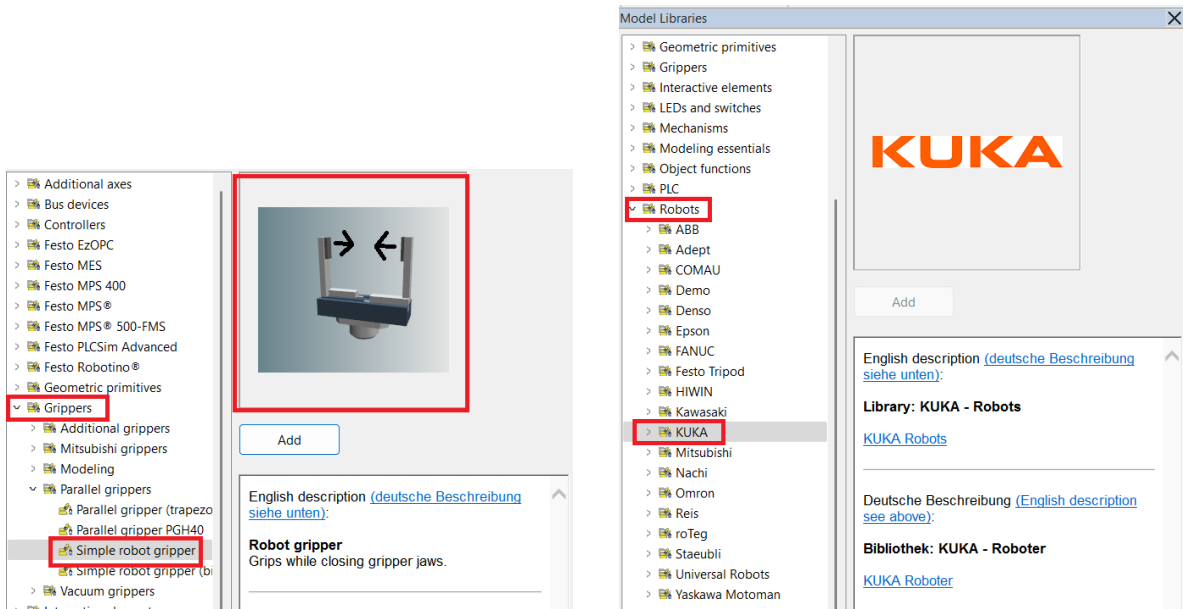


Figura 3.1: Acceso librerías del modelo

En la Figura 3.2 se muestra un ejemplo de cómo se presentan los modelos dentro del cuadro de diálogo. Estos se organizan en carpetas y subcarpetas, y al seleccionar un

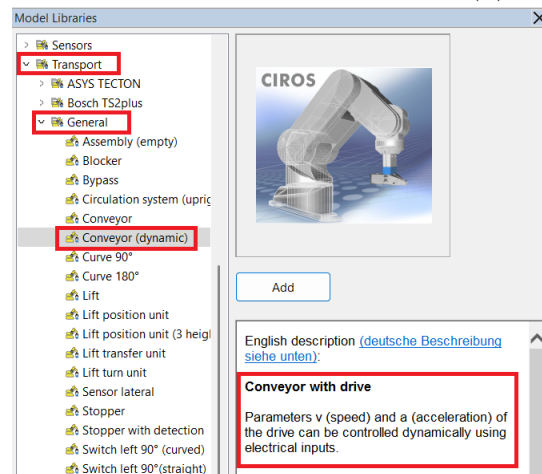
objeto, en el costado derecho se despliega una imagen junto con una breve descripción de sus características.

Las Figuras 3.2a, 3.2b y 3.2c ilustran el proceso de adición de distintos componentes clave dentro de la simulación. La Figura 3.2a muestra los pasos para agregar una garra paralela para robots, mientras que la Figura 3.2b presenta el acceso a la carpeta de robots KUKA disponibles en el software. Finalmente, la Figura 3.2c ejemplifica la incorporación de una cinta transportadora junto con su descripción.



(a) Garra paralelo

(b) Robots KUKA



(c) Cinta transportadora

Figura 3.2: Ejemplos de modelos utilizados

3.1.2. Explorador del modelo

El explorador del modelo permite acceder a todos los elementos incorporados en un modelo, incluyendo objetos, elementos subordinados, materiales, librerías y conexiones de entradas y salidas.

En la Figura 3.3 se muestra el árbol de navegación del modelo, donde se pueden visualizar las carpetas disponibles junto con sus subcarpetas, así como las secciones correspondientes a las entradas, salidas y objetos.

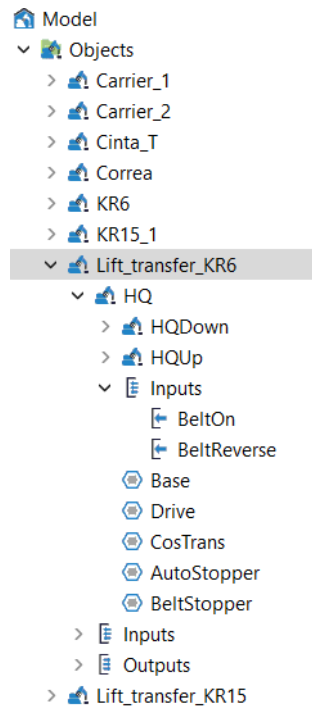
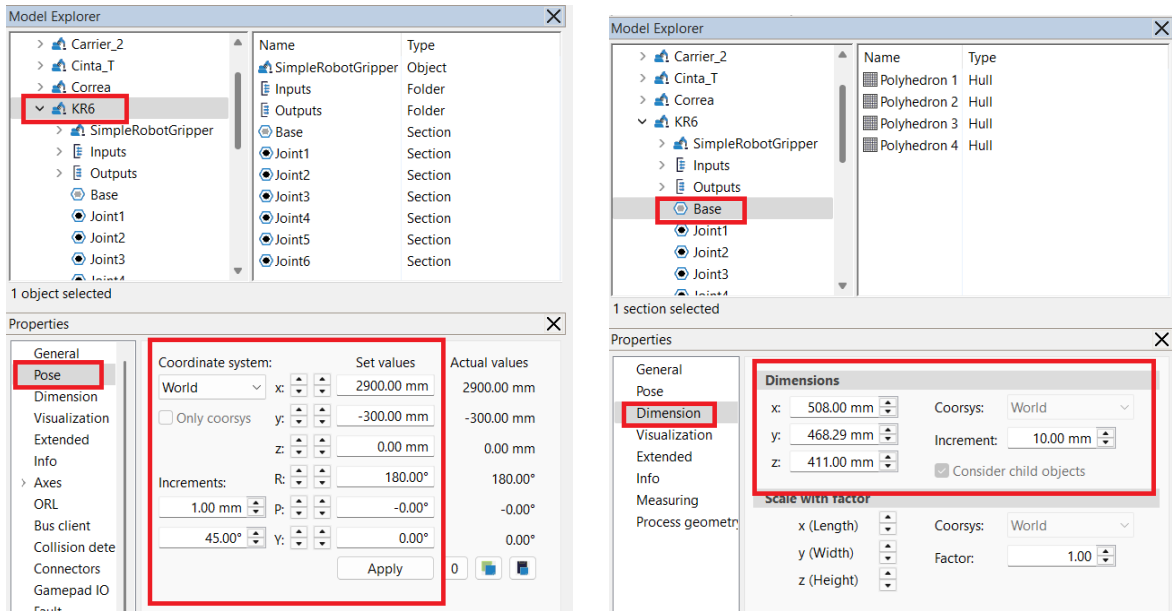


Figura 3.3: Árbol de navegación

En la Figura 3.4 se muestran ejemplos de personalización de los objetos dentro del modelo. Para modificar sus características, se debe acceder a las propiedades del objeto seleccionado y, en el panel izquierdo, elegir la categoría correspondiente.

La posición o ubicación de los objetos puede ajustarse de manera precisa, como se observa en la Figura 3.4a, permitiendo no solo modificar su ubicación en el espacio, sino también rotarlo según los ejes disponibles.

Además, es posible personalizar las dimensiones del objeto para adaptarlo a los requerimientos de la simulación, como se muestra en la Figura 3.4b.



(a) Pose

(b) Dimensión

Figura 3.4: Ejemplo de personalización

3.1.3. Asignación entradas y salidas

Para establecer la conexión entre las entradas y salidas de los componentes, se debe arrastrar una salida hacia una entrada o viceversa, según corresponda. Si la conexión se realiza correctamente, al seleccionar una carpeta de entrada/salida, en el panel izquierdo del explorador del modelo se desplegará una lista con los componentes vinculados a dicho objeto. Este proceso se ilustra en la Figura 3.5.

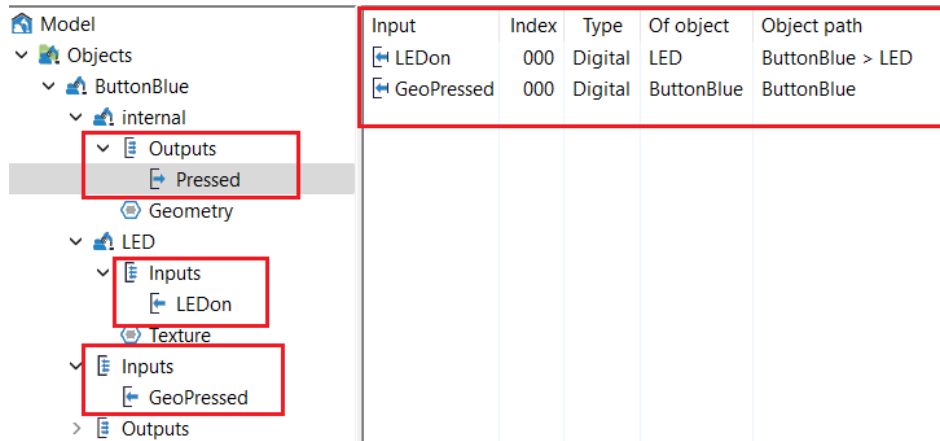
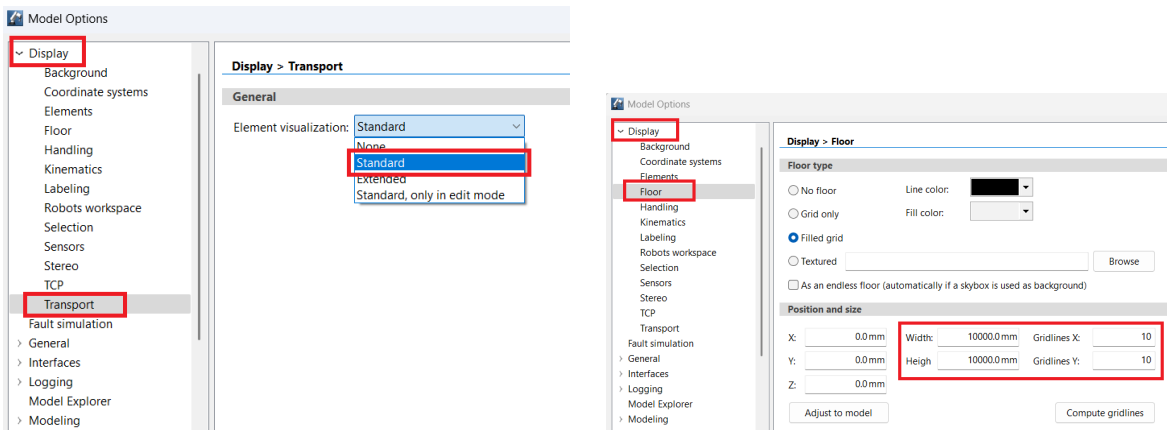


Figura 3.5: Ejemplo asignación entradas y salidas.

3.1.4. Opciones del modelo

En la ventana de opciones del modelo, a la que se puede acceder desde **Settings** > **Model Options**, es posible editar varios ajustes para la simulación. Por ejemplo, en la Figura 3.6a se muestra la configuración que permite activar o desactivar la visualización de los componentes de transporte, así como mostrar las curvas y formas de estos componentes.

Además, se pueden realizar ajustes relacionados con el piso de la simulación, como se observa en la Figura 3.6b. En esta sección, es posible editar las dimensiones del piso, incluyendo su ancho y largo, así como configurar las cuadrículas del suelo y su color.



(a) Ajuste transporte

(b) Ajuste piso

Figura 3.6: Ejemplos de ajustes del modelo

3.1.5. Project Management

Para acceder a esta ventana, se debe seguir la ruta **Programming > Project Management**, donde se pueden gestionar los programas de los robots y otros archivos. Al agregar un nuevo proyecto, se despliega una ventana que permite seleccionar el lenguaje de programación a utilizar, como se muestra en la Figura 3.7, y se crea un archivo .PJRX asociado al proyecto.

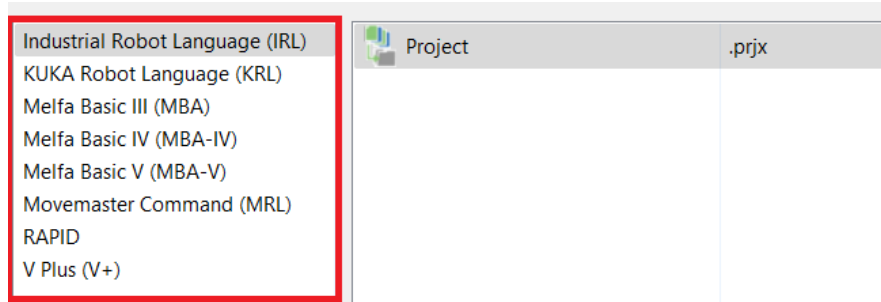


Figura 3.7: Agregar nuevo proyecto

En la ventana de **Project Management**, también se pueden ajustar los controladores a utilizar para programar los robots del proyecto, así como agregar y modificar las carpetas de código de programa, las listas de posiciones y los archivos de datos, como se muestra en la Figura 3.8.

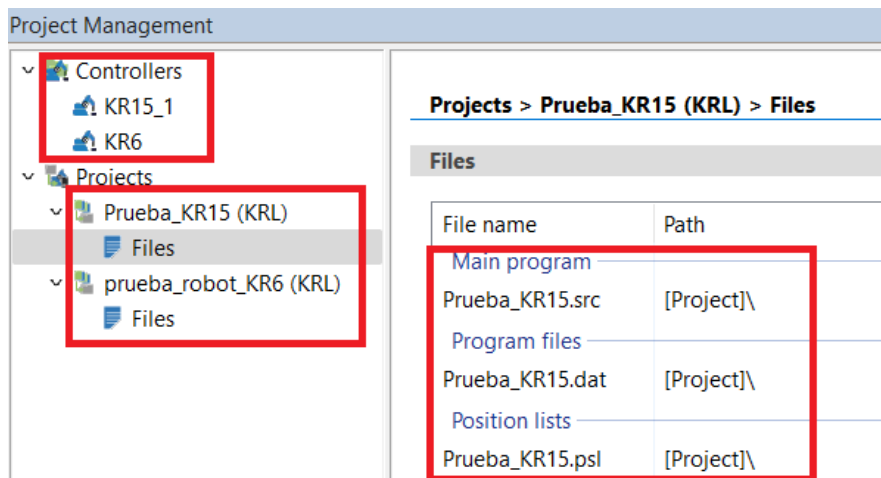


Figura 3.8: Project Management

3.2. Integración CIROS mediante servidor OPC Config

CODESYS es un entorno de desarrollo utilizado para la programación de controladores lógicos programables, basado en el estándar internacional IEC 61131-3. Para la integración del modelo de manufactura flexible, se utiliza la versión proporcionada por FESTO, la cual incluye herramientas especializadas para la configuración, programación, puesta en marcha y mantenimiento de soluciones de automatización neumática y eléctrica [15].

Este software no solo permite la programación de controladores, sino también el diseño y la programación de pantallas HMI. En este caso, se utiliza para desarrollar la programación del PLC mediante el controlador virtual *CODESYS Control Win V3* de FESTO.

3.2.1. Integración con sistemas de automatización PLC CODESYS

Para implementar la automatización en el FMS virtual, es necesario iniciar la simulación del PLC en CODESYS, utilizando el *Control Win V3*. Posteriormente, se deben enlazar las variables mediante el protocolo OPC, lo que permite la conexión entre CODESYS y CIROS Studio.

El diagrama de la implementación virtual se muestra en la Figura 3.9, donde las flechas indican el intercambio de valores de las variables del sistema.

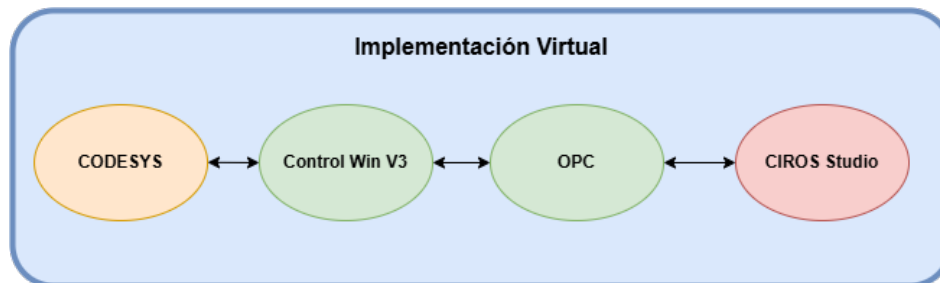


Figura 3.9: Implementación del FMS virtual.

Para la vinculación de las variables a través de OPC, se utiliza la herramienta CODESYS OPC Config, diseñada para configurar la comunicación OPC y facilitar la comunicación entre dispositivos y software industrial. Esta herramienta permite la conexión entre CODESYS y CIROS Studio, asegurando el intercambio de datos en tiempo real.

CODESYS OPC Config actúa como un puente de comunicación, permitiendo que el PLC virtual comparta y reciba datos de la simulación en CIROS Studio a través del protocolo OPC. Para lograr esto, se debe configurar correctamente el servidor OPC, definiendo la instancia de comunicación y los accesos de lectura y escritura. Además, es necesario establecer la conexión con el PLC, especificando la dirección del controlador y lista de variables a intercambiar. Finalmente se debe realizar la asignación de variables, enlazando las señales del modelo de CIROS con las variables del PLC virtual.

En la sección 4.2 se describe a detalle los pasos a seguir para lograr la correcta integración OPC.

3.2.2. Integración sistemas de supervisión HMI CODESYS

La integración de un sistema de supervisión HMI en CODESYS permite a los operadores supervisar y controlar el sistema automatizado de manera eficiente. En este entorno, las pantallas se diseñan con el objetivo de incluir botones de control y visualizaciones en tiempo real del estado del sistema. Estos elementos permiten que el usuario interactúe con el sistema de forma intuitiva y monitoree su funcionamiento fácilmente.

La conexión entre el HMI y el PLC se realiza a través de variables definidas en CODESYS, asegurando que los datos se actualicen en tiempo real en la interfaz del usuario.

Cuando es necesario implementar un modo de operación manual, la interfaz HMI permite controlar los mecanismos de manera independiente, como cintas transportadoras o elevadores. Esta funcionalidad permite verificar el correcto funcionamiento de las conexiones y los mecanismos sin depender de la secuencia automatizada.

4 | Modelo CIROS del FMS e integración con PLC y HMI virtuales

Para la simulación, se utilizan componentes de las librerías disponibles en CIROS Studio, como correas, unidades de elevación, sensores capacitivos, objetos, pallets, pulsadores, interruptores, selectores y robots KUKA. Además, se emplean elementos de la librería de misceláneos para replicar con mayor fidelidad el diseño físico original del sistema.

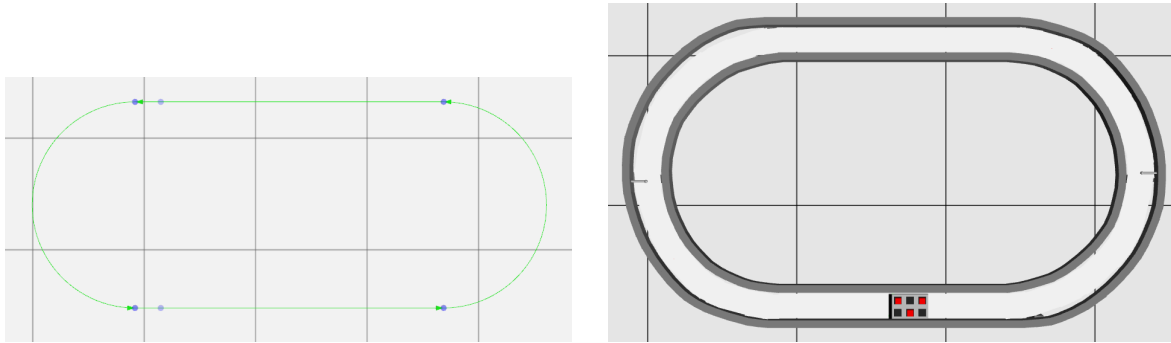
4.1. Componentes virtuales

A continuación se presentan los componentes utilizados en CIROS Studio al momento de crear la simulación.

4.1.1. Cinta transportadora

En la Figura 4.1a se muestra la cinta transportadora en vista superior en su diseño base, aún sin geometría definida. Su estructura está compuesta por cuatro secciones: dos rectas y dos curvas de 180 grados. La velocidad de la cinta es ajustable, lo que permite una simulación más fiel al comportamiento de la correa física. El control de activación y desactivación se gestiona a través de una señal binaria ON/OFF, enviada desde el PLC.

En la Figura 4.1b se presenta una vista superior de la cinta transportadora con su diseño completo y un pallet sobre ella en posición HOME, es decir, donde empieza el programa. Lo que permite visualizar su disposición dentro del entorno simulado.



(a) Estructura cinta transportadora desde arriba.

(b) Cinta transportadora desde arriba con geometría.

Figura 4.1: Simulación cinta transportadora.

4.1.2. Pallet de 6 posiciones y cubos

Se utiliza un pallet con seis huecos de superficies activas y tres cubos como objetos dinámicos, los cuales se pueden observar en la Figura 4.2. Se consideran objetos dinámicos a los cubos, ya que pueden ser retirados del pallet, en este caso, mediante las garras de los robots.

En la simulación, los huecos del pallet real se representan como posiciones de color oscuro con propiedades de superficie activa. Esto permite que un objeto dinámico, al estar ubicado sobre un área activa, se desplace junto con el pallet a lo largo de la cinta transportadora.

Para permitir la manipulación de los cubos por parte de los robots, cada cubo cuenta con un *grip point* ubicado en el centro de su geometría, un punto diseñado para facilitar su agarre. Las garras de los robots KUKA KR6 y KR15 están equipadas con *grripper points*, los cuales han sido diseñados para alinearse y acoplarse con los *grip points* de los cubos. Al conectarse estos puntos, se logra simular la acción de agarre en la simulación, permitiendo que los robots sujeten los cubos y los trasladen de posición.

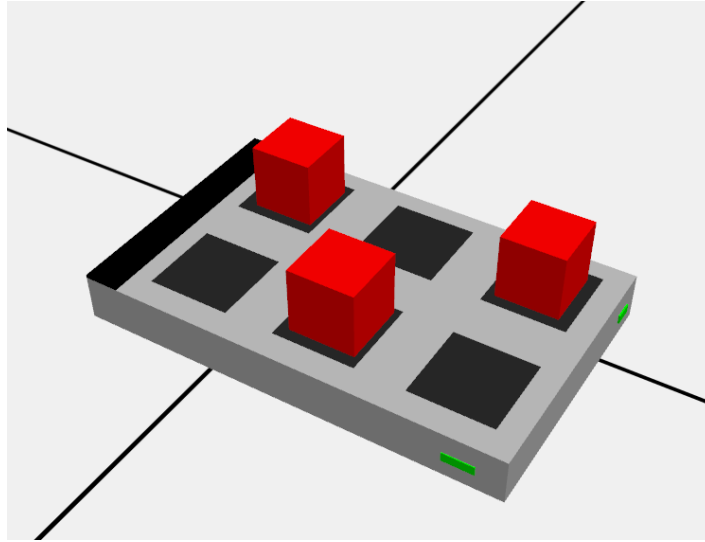


Figura 4.2: Pallet con cubos.

4.1.3. Estaciones de transferencia electroneumáticas

El sistema físico utiliza varios cilindros para elevar el pallet y, posteriormente, transportarlo horizontalmente hacia la estación de trabajo estacionaria de cada robot. Para simular este mecanismo en CIROS Studio, se utilizó el componente *Lift transfer unit with stopper*.

Este componente puede montarse sobre la cinta transportadora y utilizarse para transferencias laterales. Cuenta con tres posiciones en el eje Z: abajo, medio y arriba.

- En la posición abajo, el pallet continúa su recorrido por la cinta transportadora sin interrupciones.
- En la posición medio, el pallet se detiene temporalmente, cumpliendo la función de freno.
- En la posición arriba, si se activa el movimiento de la cinta interna del elevador, el pallet es transportado lateralmente.

El accionamiento de elevación y desplazamiento horizontal son del tipo monoestable, es decir, se mantiene la posición hasta que se detenga la señal de activación y al detenerse esta se devuelve a su estado original. Mientras que el mecanismo de freno se activa de manera automática al estar en la posición medio.

La Figura 4.3a muestra la estructura básica del componente, donde se destacan los

stoppers con forma de esfera. Los *stoppers* de los extremos detienen el pallet en la cinta interna del elevador cuando este se encuentra en la posición arriba, mientras que el *stopper* central detiene el pallet si el elevador está en la posición medio.

En la Figura 4.3b se observa el elevador con su geometría, compuesta por dos cilindros: uno vertical y otro horizontal. El cilindro vertical controla el movimiento de elevación del pallet cuando se activa la señal correspondiente, mientras que el cilindro horizontal permite el desplazamiento lateral del pallet hacia la estación de trabajo del robot.

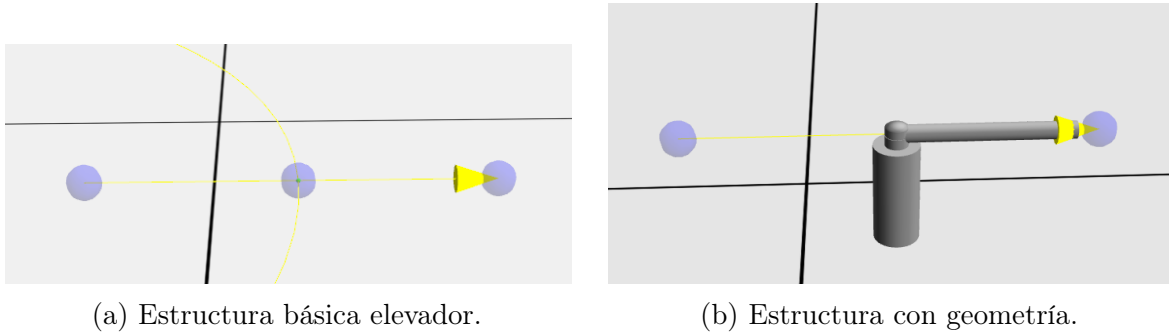
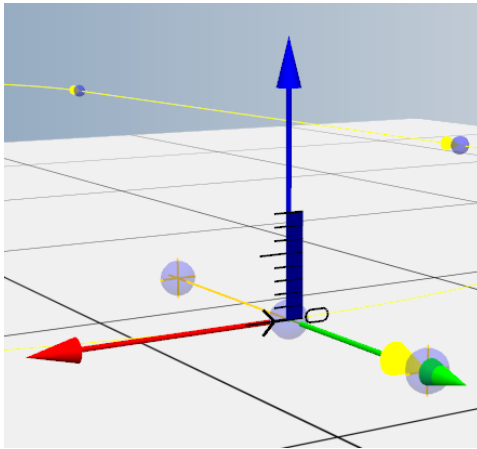


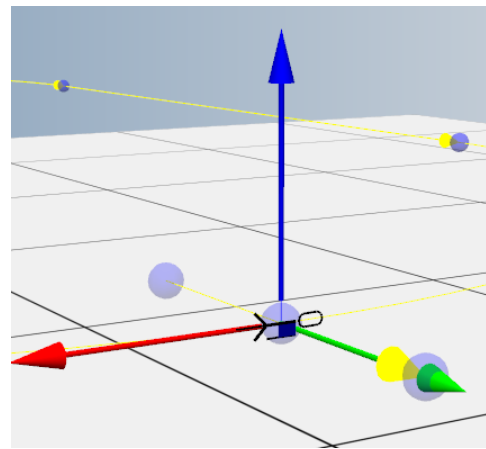
Figura 4.3: Mecanismo elevador estación transferencia.

La Figura 4.4 muestra la estructura de los cilindros encargados de elevar y descender el pallet hacia la estación de trabajo. En la Figura 4.4a, se observa el eje vertical del cilindro elevador, el cual tiene una longitud de 80 mm. Por otro lado, la Figura 4.4b muestra el eje del cilindro descendente, cuya longitud es de 10 mm.

Este último tiene una menor longitud, ya que su función es únicamente desbloquear el *stopper* del componente, sin necesidad de descender el pallet por debajo del nivel de la cinta transportadora.



(a) Eje elevador arriba.

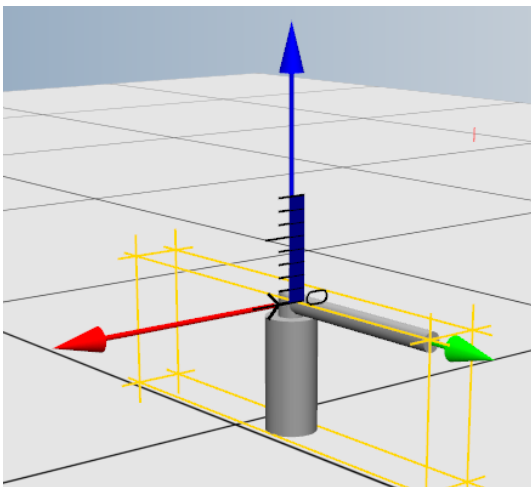


(b) Eje elevador abajo.

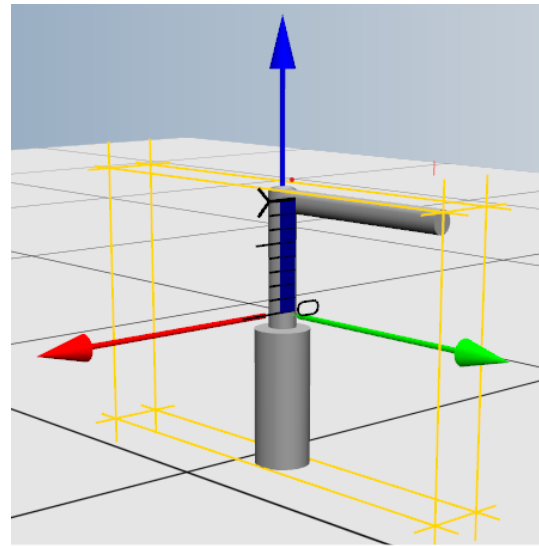
Figura 4.4: Mecanismo elevador estación transferencia.

La Figura 4.5a muestra la estructura final del elevador. Cuando se activa la señal de extensión, el cilindro se extiende y eleva el pallet mediante el mecanismo ilustrado en la Figura 4.5b. Donde luego se transporta horizontalmente por la sección horizontal mostrada en la Figura 4.5b.

Cabe mencionar que en la simulación no se implementaron los frenos frontal y de cola presentes en el sistema físico en CIMA, esto debido a que el componente *Lift transfer unit with stopper* ya incorpora internamente un mecanismo de detención que cumple la misma función, por lo que no es necesario agregar otro extra.



(a) Eje elevador con geometría.



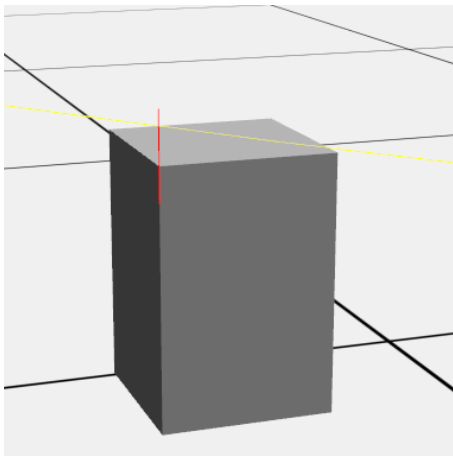
(b) Eje elevador extendido con geometría.

Figura 4.5: Mecanismo elevador con geometría.

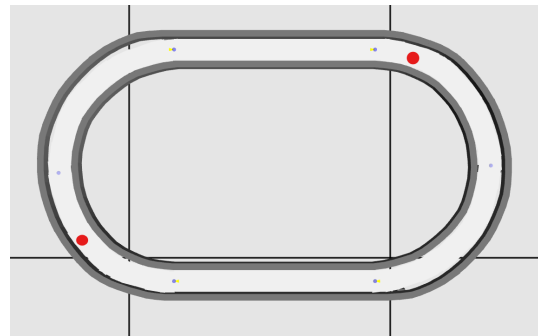
4.1.4. Sensores

El sistema físico emplea dos sensores de contacto para detectar la aproximación de los pallets a las estaciones de transferencia, uno para cada estación. Sin embargo, debido a que CIROS Studio no incluye este tipo de sensores en sus librerías, se toma la decisión de reemplazarlos con sensores capacitivos de corto alcance. En este caso, el rango de medición es de 40 mm, como se ilustra con la línea roja en la Figura 4.6a.

En la Figura 4.6b, se muestran las posiciones de los sensores en la implementación virtual, marcadas con círculos rojos. Estos se ubican por debajo de la cinta transportadora, de manera que el rango de detección permita identificar el paso del pallet cuando este pasa por encima.



(a) Sensor capacitivo.



(b) Ubicaciones sensores.

Figura 4.6: Sensores virtuales.

4.1.5. Panel de operaciones

Para activar y desactivar el programa, se incluye un panel de operaciones equipado con pulsadores, un interruptor, un selector de dos posiciones y dos LEDs, como se muestra en la Figura 4.7.

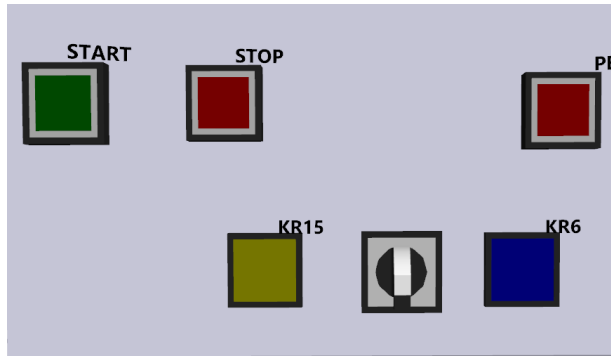


Figura 4.7: Panel de operaciones virtual en CIROS.

El panel cuenta con pulsadores de *START* y *STOP*, los cuales inician y detienen el ciclo del programa. Además, dispone de un interruptor de parada de emergencia (P.E.) que permite interrumpir el ciclo en cualquier momento, manteniéndolo detenido hasta que se desactive la señal del interruptor.

El selector de posiciones permite elegir cuál robot estará activo al iniciar el ciclo. Finalmente, los LEDs se encienden cuando un pallet activa el sensor que indica su aproximación a la estación de transferencia de cada robot y se apagan una vez que el pallet llega a dicha estación.

4.1.6. Robot KUKA KR15

Para la simulación, se incorpora un robot KUKA KR15 con un séptimo eje lineal, como se muestra en la Figura 4.8. Este eje lineal tiene una longitud de 3.2 m, aunque el recorrido útil para el desplazamiento del robot es de 2.7 m.

Gracias a este eje adicional, el robot puede moverse entre su posición inicial y la segunda estación de trabajo, ampliando así su espacio de operación y permitiendo una mayor flexibilidad en la ejecución de tareas dentro del sistema.

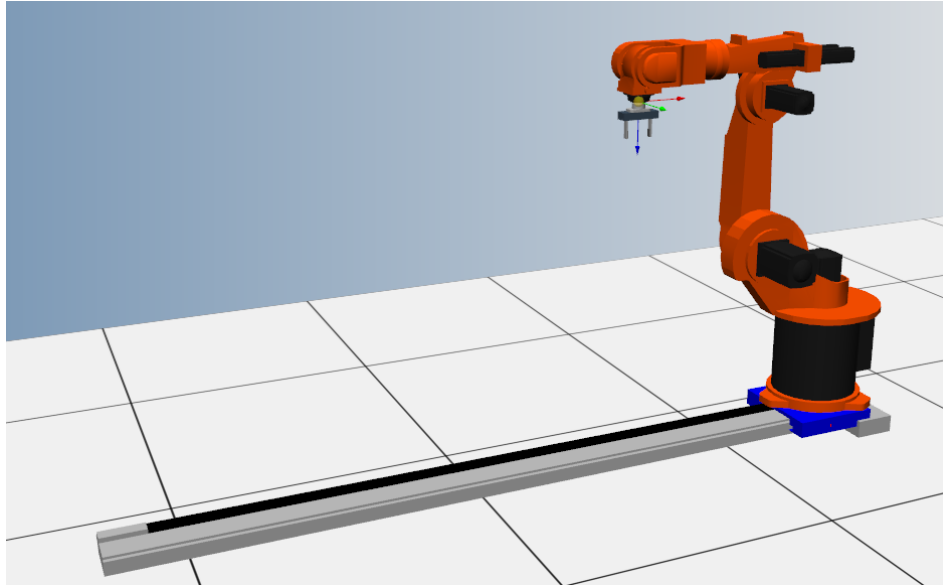


Figura 4.8: Robot KUKA KR15 con eje lineal.

4.1.7. Robot KUKA KR6

En la Figura 4.9 se muestra el robot KUKA KR6 utilizado en la simulación.

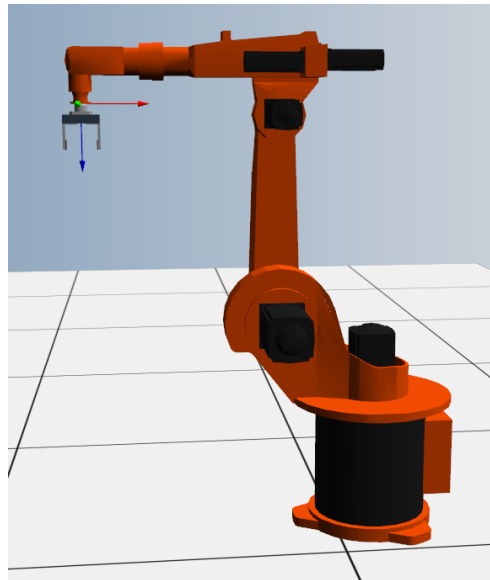


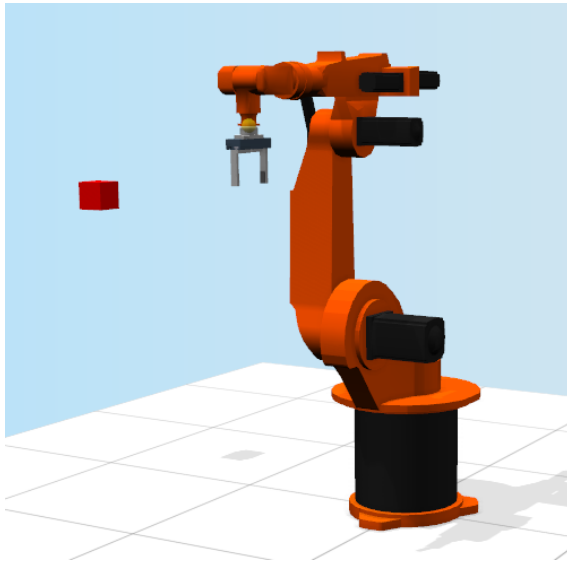
Figura 4.9: Robot KUKA KR6.

4.1.7.1. Programa KRL de prueba

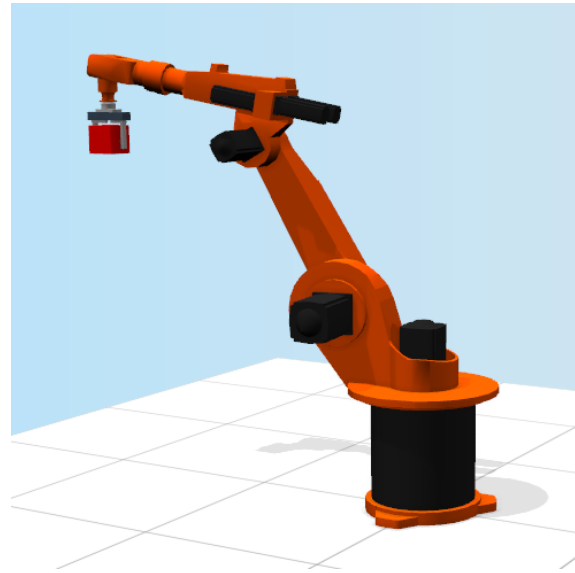
Se lleva a cabo una prueba con el robot KUKA KR6, con el propósito de verificar el correcto funcionamiento del programa desarrollado en KRL, el cual se detalla a continuación. La rutina consiste en desplazar el robot desde la posición “HOME” hasta la posición “POS1”, para luego descender ligeramente hasta “POS2”, activar la garra, elevar el cubo hasta “POS1” y finalmente trasladarlo de regreso a “HOME”, donde se procede a soltarlo.

```
1 DEF prueba_robot_solo ()
2   LIN HOME
3   WAIT SEC 2
4   LIN POS1
5   WAIT SEC 2
6   LIN POS2
7   $OUT[0] = TRUE ;Activar gripper
8   WAIT SEC 2
9   LIN POS1
10  WAIT SEC 1
11  LIN HOME
12  $OUT[0] = FALSE ;Desactivar gripper
13 END
```

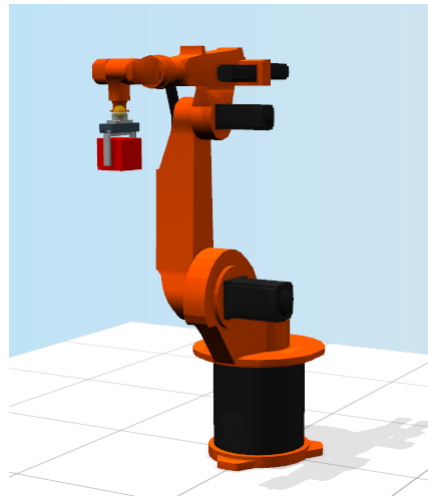
En la Figura 4.10 se ilustran los movimientos ejecutados por el robot durante la rutina de prueba. La Figura 4.10a muestra la posición inicial “HOME” antes de iniciar la secuencia. Posteriormente, en la Figura 4.10b, se observa el robot sosteniendo el cubo en la posición “POS1”, y finalmente, en la Figura 4.10c, se aprecia el estado final de la rutina, con el robot ubicado en “HOME” y el cubo aún en su garra.



(a) Posición HOME.



(b) Posición POS1 con el cubo en la garra.



(c) Posición HOME con el cubo en la garra.

Figura 4.10: Movimientos robot en rutina de prueba.

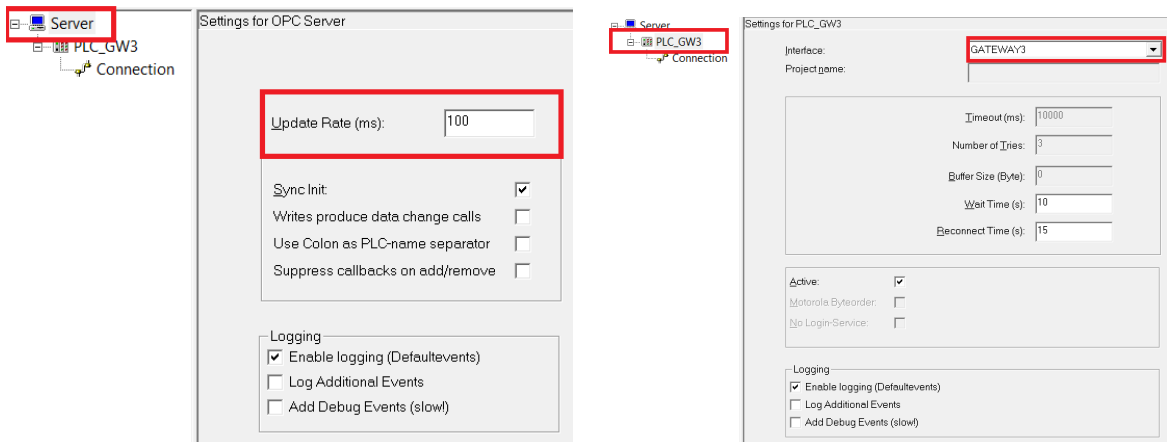
4.2. Integración OPC

A continuación se describen las configuraciones necesarias para realizar la integración mediante OPC Config de CIROS Studio y CODESYS.

4.2.1. Configuración servidor OPCConfig

Una vez abierta la aplicación OPC Config, es necesario configurar los parámetros del servidor, el PLC y la conexión, tal como se muestra en las Figuras 4.11a, 4.11b y 4.12. El objetivo es asegurar una actualización rápida de los datos, por lo que se define este valor en 100 *ms*.

Se debe establecer el tiempo de espera antes de reintentar la conexión. Finalmente, se deben configurar la interfaz y el puerto a utilizar para completar la conexión.



(a) Configuración Servidor.

(b) Configuración PLC.

Figura 4.11: Configuraciones OPC.



Figura 4.12: Configuración conexión.

4.2.2. PLC virtual

Para la simulación, se utiliza el PLC virtual Control Win V3, que viene incluido en CODESYS y permite ejecutar y probar programas sin necesidad de un PLC físico. Este se selecciona al momento de crear el archivo en CODESYS, donde también se define el lenguaje de programación a utilizar.

En este caso, se elige utilizar *Sequential Function Chart* (SFC), un lenguaje gráfico basado en diagramas de estado, diseñado para la programación la automatización de procesos secuenciales. Este lenguaje facilita la estructuración del código mediante pasos y transiciones, lo que mejora la organización y comprensión del programa, especialmente en sistemas automatizados con múltiples estados operativos, como es en este caso [16].

Una vez configurado el PLC virtual, CODESYS permite su simulación y visualización de la ejecución del programa, la activación de variables y la respuesta del sistema en tiempo real. La Figura 4.13 ilustra la configuración del controlador en CODESYS.

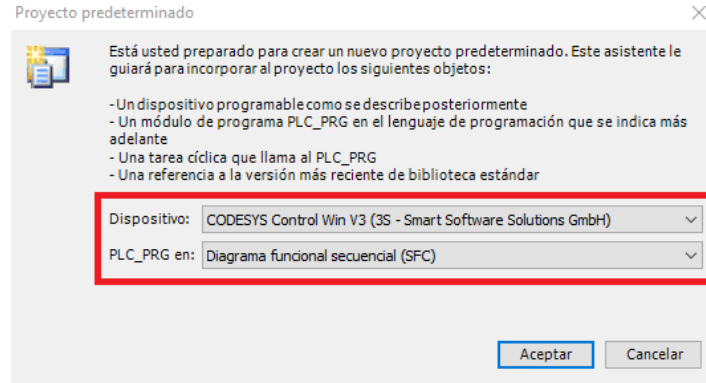


Figura 4.13: Configuración proyecto en CODESYS.

A continuación, se debe encender el PLC virtual en el computador. Para ello, se debe abrir la aplicación **CODESYS Control Win V3 SysTray** y buscar su ícono en la barra de tareas. Al hacer clic derecho sobre el ícono, se despliega un menú, como se muestra en la Figura 4.14, donde se debe seleccionar la opción **Start PLC** para activarlo.

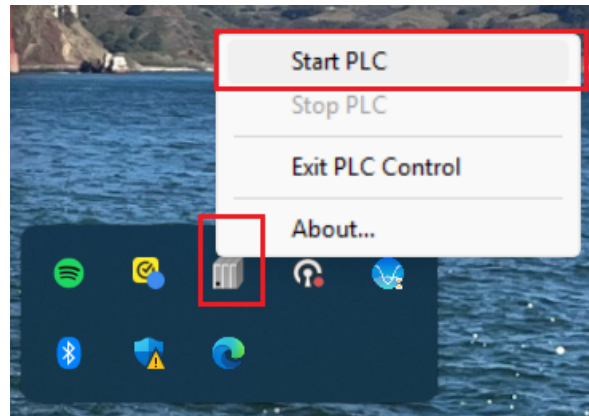


Figura 4.14: CODESYS Control Win V3 SysTray.

Posteriormente, para establecer la comunicación entre el PLC Virtual Control Win V3 y CODESYS se debe acceder a la ventana **Device** dentro del árbol de dispositivos. Luego, es necesario hacer doble clic en el nombre del computador a utilizar, lo que activa la conexión y cambia el nombre a negritas. Esto se muestra en la Figura 4.15.

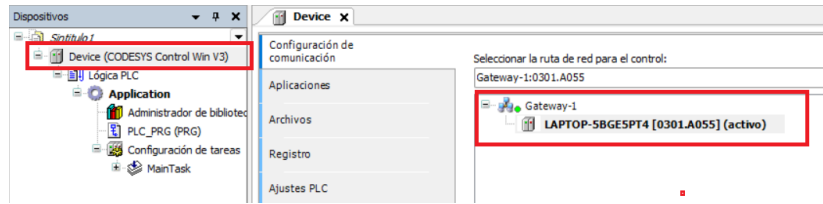


Figura 4.15: Configuración dispositivo en CODESYS.

Para completar la carga de variables a través de OPC, se debe agregar el objeto **OPCServedPLC** desde la carpeta **PLC** en CIROS. Una vez añadido, en sus propiedades se debe acceder a la opción **OPC Client**, abriendo la ventana mostrada en la Figura 4.16. Es importante asegurarse de que el servidor seleccionado sea **CoDeSys.OPC.DA** y luego hacer clic en **Connect Items**.

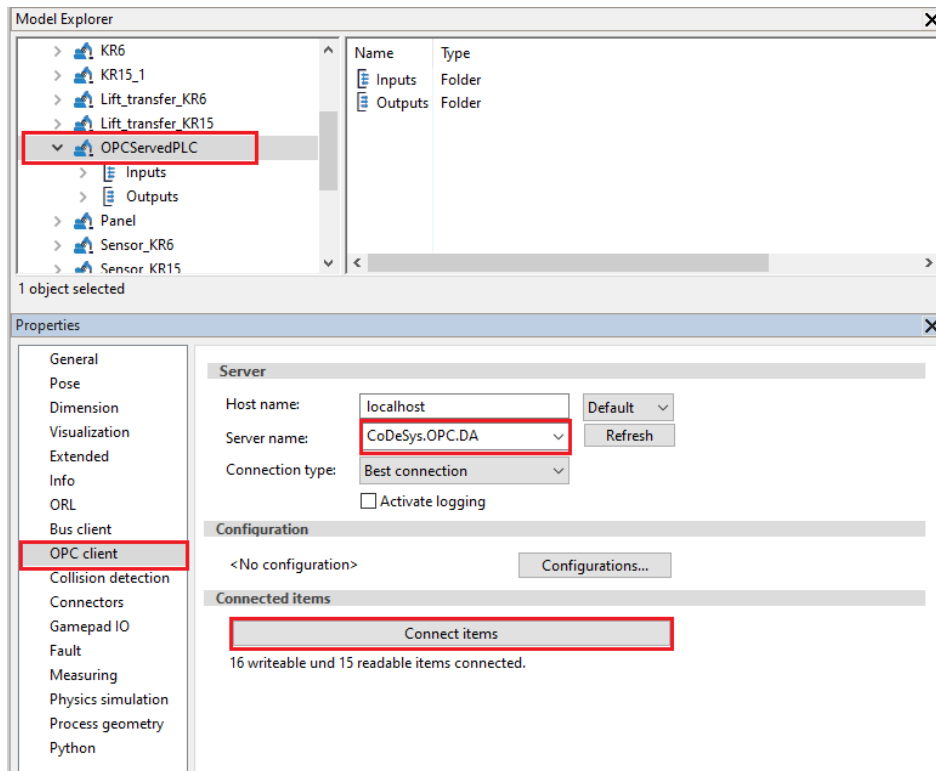


Figura 4.16: OPC CLient en CIROS Studio.

Al presionar el botón, se desplegará otra ventana. En la parte izquierda de esta, se mostrarán las entradas y salidas del cliente OPC en CIROS Studio. Para conectar las variables, se debe hacer clic en **Connect > PLC_GW3 > Application > GVL**. Esto desplegará una lista con las variables de CODESYS, las cuales ya estarán vinculadas con las entradas y salidas de CIROS. Finalmente, se debe presionar **Apply** para confirmar

la conexión. Este proceso se ilustra en la Figura 4.17.

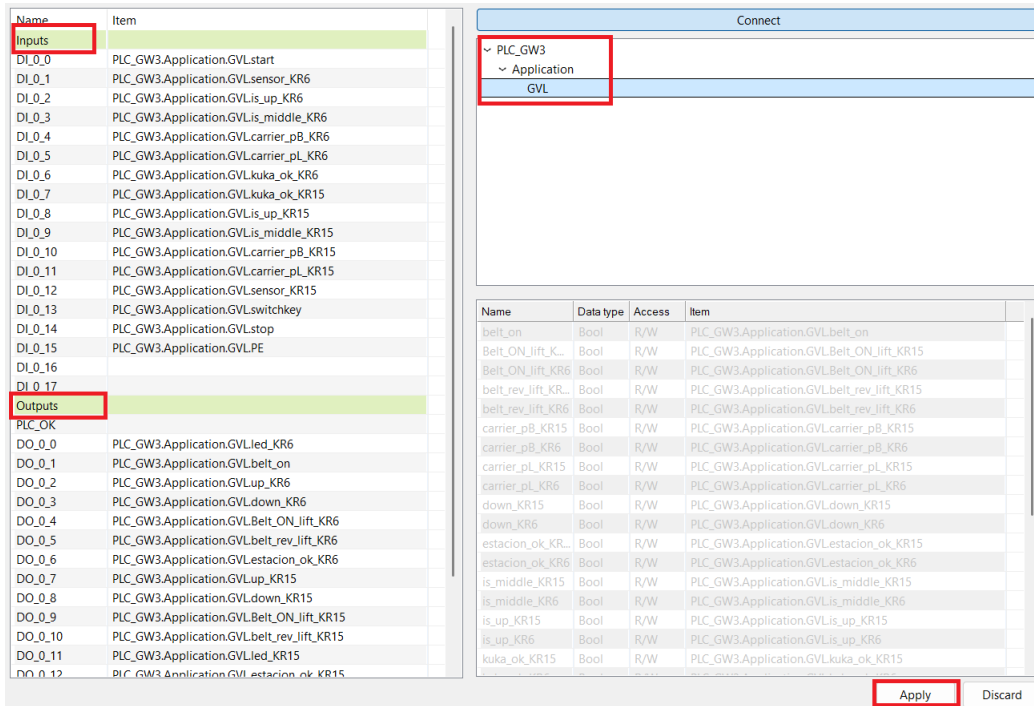


Figura 4.17: Variables compartidas.

En la Figura 4.18 se muestran las conexiones entre las variables del servidor y las de la simulación. Por ejemplo, la entrada (DI_0_0) está vinculada al botón **START** de la simulación.

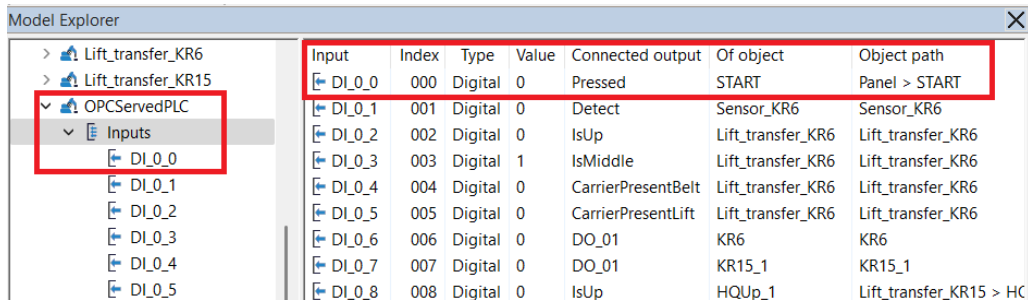


Figura 4.18: Ejemplo de conexión.

4.2.3. Tabla de entradas y salidas del modelo CIROS

A continuación se presentan las tablas 4.1 y 4.2, de entradas y salidas lógicas del PLC, junto con sus descripciones y señales correspondientes en el sistema .

Tabla 4.1: Entradas Lógicas PLC.

Salidas Sistema	Entradas Lógicas PLC	Descripción
start	DI_0_0	Pulsador START presionado
sensor_KR6	DI_0_1	Sensor detecta pallet acercándose a estación de trabajo KR6
is_up_KR6	DI_0_2	Elevador estación KR6 está en posición “up”
is_middle_KR6	DI_0_3	Elevador estación KR6 está en posición “middle”
carrier_pB_KR6	DI_0_4	Pallet en posición de elevador estación KR6
carrier_pL_KR6	DI_0_5	Pallet elevado por elevador estación KR6
kuka_ok_KR6	DI_0_6	Rutina KR6 terminada
kuka_ok_KR15	DI_0_7	Rutina KR15 terminada
is_up_KR15	DI_0_8	Elevador estación KR15 está en posición “up”
is_middle_KR15	DI_0_9	Elevador estación KR15 está en posición “middle”
carrier_pB_KR15	DI_0_10	Pallet en posición de elevador estación KR15
carrier_pL_KR15	DI_0_11	Pallet elevado por elevador estación KR15
sensor_KR15	DI_0_12	Sensor detecta pallet acercándose a estación de trabajo KR15
switchkey	DI_0_13	Selector en posición “1”
stop	DI_0_14	Pulsador STOP presionado
PE	DI_0_15	Interruptor parada de emergencia activado

Tabla 4.2: Salidas Lógicas PLC.

Entradas Sistema	Salidas Lógicas PLC	Descripción
led_KR6	DO_0_0	Enciende LED proximidad KR6
belt_on	DO_0_1	Habilita movimiento correa transportadora
up_KR6	DO_0_2	Eleva elevador estación KR6
down_KR6	DO_0_3	Baja elevador estación KR6
belt_on_lift_KR6	DO_0_4	Habilita movimiento estación transferencia KR6
belt_rev_lift_KR6	DO_0_5	Habilita movimiento reversa estación transferencia KR6
estacion_ok_KR6	DO_0_6	Pallet estación de trabajo KR6
up_KR15	DO_0_7	Eleva elevador estación KR15
down_KR15	DO_0_8	Baja elevador estación KR15
belt_on_lift_KR15	DO_0_9	Habilita movimiento estación transferencia KR15
belt_rev_KR15	DO_0_10	Habilita movimiento reversa estación transferencia KR15
led_KR15	DO_0_11	Enciende LED proximidad KR15
estación_ok_KR15	DO_0_12	Pallet estación de trabajo KR15
PE_KR6	DO_0_13	Señal parada de emergencia KR6
PE_KR15	DO_0_14	Señal parada de emergencia KR15

Todas las salidas lógicas del sistema son del tipo monoestable, puesto que su activación depende exclusivamente de la señal correspondiente. Una vez desactivada dicha señal, la salida retorna automáticamente a su estado inactivo.

4.2.3.1. Programa SFC de prueba

Para realizar las primeras pruebas de conexión entre CODESYS y CIROS Studio, se desarrolla un programa de prueba, como se muestra en la Figura 4.19. El funcionamiento del programa consiste en que, al presionar el botón de activación, el pallet comienza a desplazarse sobre la cinta transportadora. Después de pasar el sensor de aproximación y transcurridos 8 segundos, la cinta se detiene. Posteriormente, al presionar nuevamente el botón, la cinta inicia un movimiento en reversa durante 5 segundos hasta detenerse. Finalmente, el sistema queda en espera de una nueva activación para reiniciar el ciclo.

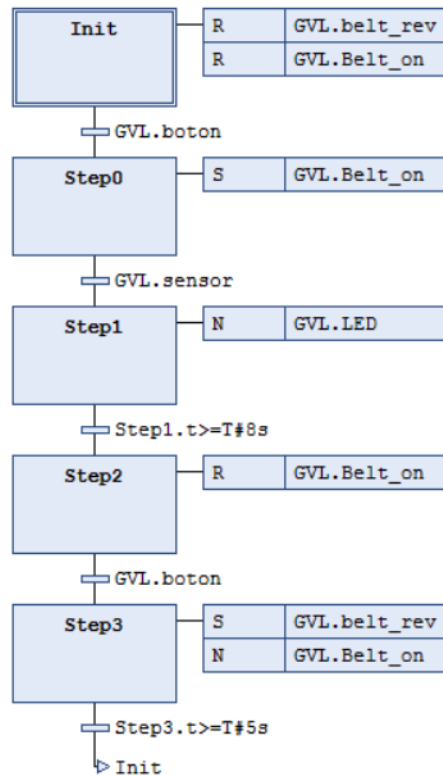


Figura 4.19: Programa SFC de prueba.

A continuación, en las Tablas 4.3 y 4.4, se presentan las señales utilizadas junto con su correspondiente dirección en el PLC virtual.

Tabla 4.3: Tabla Entradas PLC – Programa de prueba.

Salida Sistema	Entradas lógicas PLC	Descripción
boton	DI_0_0	Botón activación presionado
sensor	DI_0_1	Señal sensor de aproximación

Tabla 4.4: Tabla Salidas PLC – Programa de prueba.

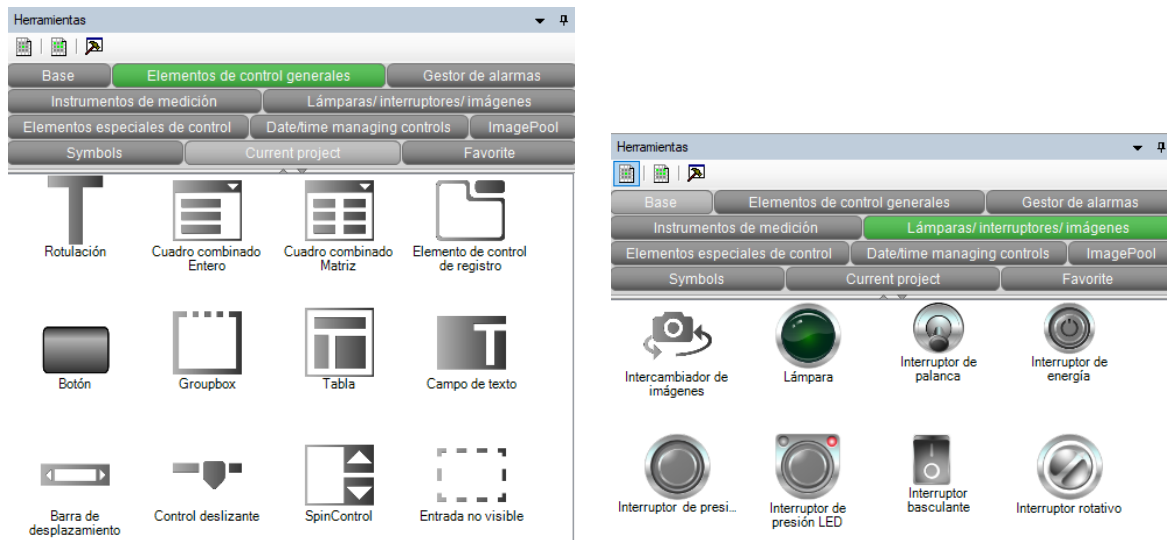
Entrada Sistema	Salida lógicas PLC	Descripción
LED	DO_0_0	LED sensor
belt_on	DO_0_1	Correa transportadora ON
belt_rev	DO_0_1	Correa transportadora reversa

4.2.4. Pantalla virtual

Una pantalla HMI (Interfaz Hombre-Máquina, por sus siglas en inglés) es una herramienta que permite la supervisión y el control de determinados sistemas. En este caso, la HMI virtual facilita el monitoreo en tiempo real de los sensores y actuadores del sistema, proporcionando una visualización detallada de su estado y funcionamiento.

CODESYS permite incorporar diversos elementos en sus visualizaciones, como lámparas, interruptores, botones y cuadros de texto, entre otros. La Figura 4.20a muestra algunos elementos de control generales, como cuadros de texto, botones simples, tablas y barras de desplazamiento.

Por otro lado, en la Figura 4.20b se presentan distintos diseños disponibles para lámparas, interruptores de palanca, indicadores de energía y presión, entre otros.



(a) Componentes de control general.

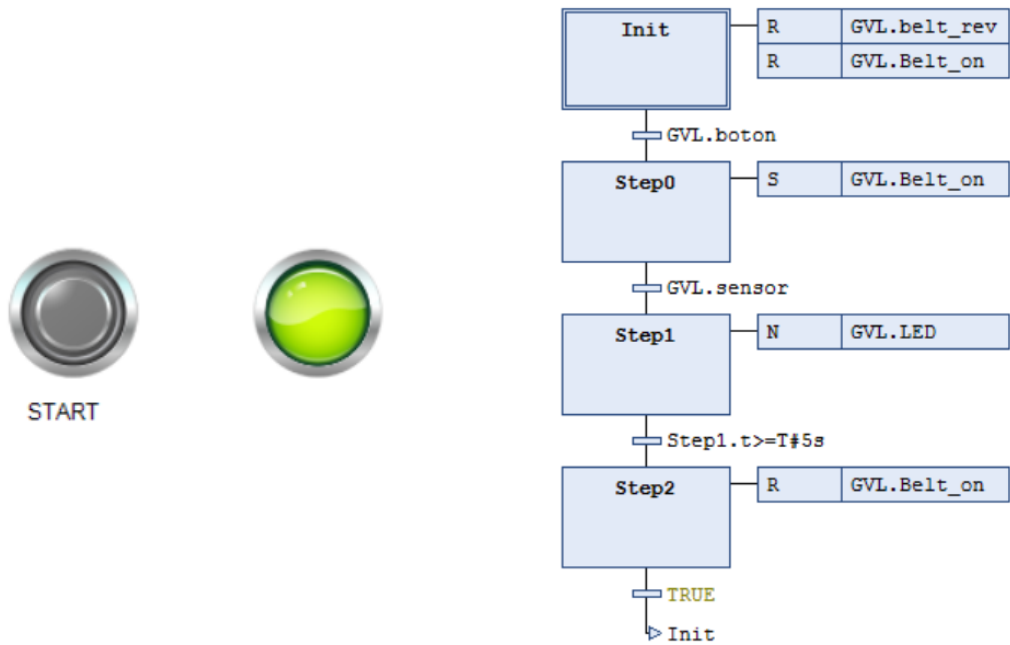
(b) Diseños de lámparas e interruptores.

Figura 4.20: Componentes disponibles visualización.

4.2.4.1. Programa HMI de prueba

Al momento de realizar las primeras pruebas de conexión del HMI en CODESYS y CIROS Studio, se desarrolla un modelo de prueba, como se muestra en la Figura 4.21. El funcionamiento consiste en un botón que active la cinta transportadora, y un LED que indique si esta está encendida o no, como se observa en la Figura 4.21a.

En la Figura 4.21b se presenta el programa en lenguaje SFC utilizado para la prueba del HMI, el cual se basa en la prueba previamente ilustrada en la Figura 4.19, con una implementación simplificada.



(a) Prueba diseño HMI.

(b) Programa SFC para prueba de HMI.

Figura 4.21: Componentes disponibles visualización.

5 | Experiencia virtual “Automatización de un Sistema de Manufactura Flexible”

Conociendo las herramientas de desarrollo del esquema completo, se procede a la automatización del sistema, integrando tanto la automatización programada del proceso como la automatización robotizada para ambos robots. En esta etapa, es fundamental asegurar la sincronización entre los distintos componentes para garantizar la precisión y el correcto funcionamiento del sistema.

La automatización programada se encarga de gestionar la lógica de control del sistema, la cual se implementa mediante un PLC que regula el flujo de trabajo. Este control incluye la activación de la cinta transportadora, la detección de objetos en ella y la comunicación con los robots, permitiendo la coordinación eficiente de todas las operaciones.

Por otro lado, la automatización robotizada consiste en la programación y control de los robots KUKA, responsables de manipular los cubos en los pallets. A través de la integración con el PLC, los robots reciben señales que indican cuándo deben ejecutar sus rutinas, permitiendo la realización de las tareas de *pick and place*, que se refiere al proceso de manipular y colocar productos en otro lugar de manera automatizada [17].

La combinación de ambas automatizaciones permite la creación de un entorno simulado completamente funcional. Esto facilita la validación del sistema en un entorno virtual, asegurando que el diseño cumpla con los requisitos operativos.

En la Figura 5.1 se muestra el modelo virtual completo desarrollado en CIROS Studio, el cual integra los componentes descritos previamente en el Capítulo 4, y sobre el cual se desarrolla la automatización del sistema.

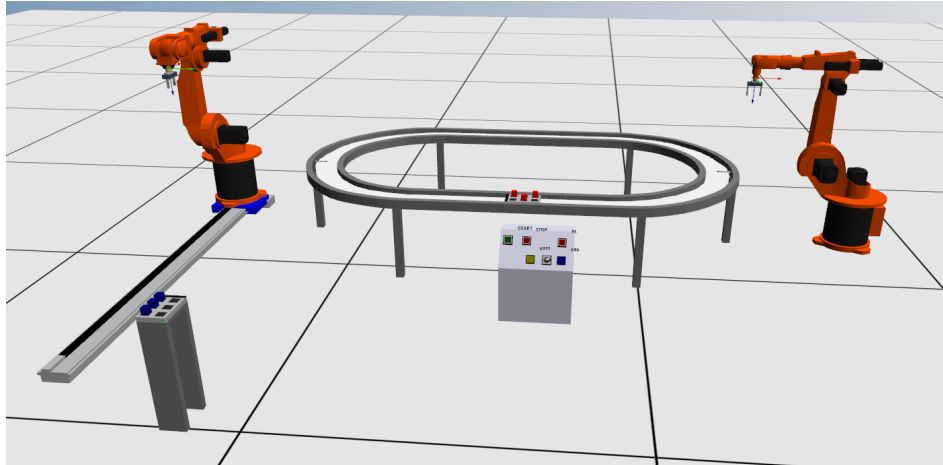


Figura 5.1: Modelo completo del FMS en CIROS.

5.1. Requisitos funcionales de automatización FMS

Es fundamental establecer los requisitos funcionales de la automatización lógica del Sistema de Manufactura Flexible (FMS) para garantizar su correcto funcionamiento. Los requisitos definidos son los siguientes:

5.1.1. Requisitos automatización programada

Corresponden a las acciones gestionadas por el PLC, encargadas de controlar el flujo de trabajo y la interacción entre los distintos componentes del sistema:

- Detectar la aproximación del pallet a la zona de transferencia por la cinta transportadora mediante un sensor “detecta pieza”.
- Detener el pallet en la zona de transferencia, activando el freno y mecanismo de control de cola de la estación.
- Elevar el pallet y posteriormente activar el mecanismo de transporte de bandeja para posicionarlo en la estación de trabajo del robot industrial.
- Una vez finalizada la rutina del robot, devolver la bandeja a su posición original y descender el pallet nuevamente a la cinta transportadora.
- Desactivar el freno y control de cola, permitiendo que el pallet continúe su desplazamiento por la cinta transportadora y permitir el ingreso de un pallet.

A partir de los requisitos funcionales de automatización del FMS, se obtiene la ecuación de movimiento correspondiente a la secuencia mencionada, como se muestra en la Figura 5.2. La cual es válida para cualquiera de las dos estaciones de transferencia del sistema físico.

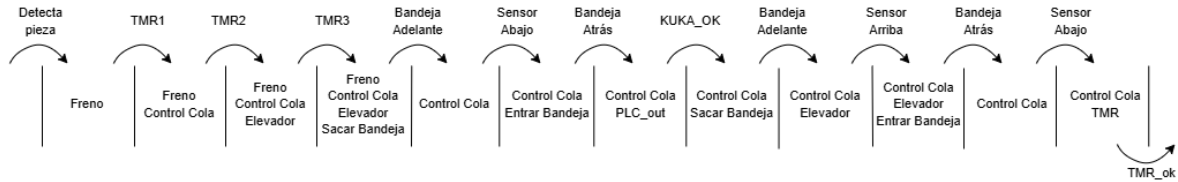


Figura 5.2: Ecuación de movimiento.

A partir de esta ecuación de movimiento, se desarrolla el diagrama de estados asociado, como se observa en la Figura 5.3.

En este modelo se agrupan los estados no bloqueantes, permitiendo la simplificación de la secuencia sin perder los detalles sobre las condiciones relevantes del proceso. En cada estado se detallan las acciones principales y las condiciones que habilitan el avance al siguiente estado de la secuencia. Es importante destacar que las condiciones que deben cumplirse dentro de cada estado se producen de forma secuencial y no en paralelo. Cada acción como la finalización de un temporizador o la activación de un sensor, ocurre en un orden definido antes de ejecutar la siguiente acción. Las señales de sacar y entrar bandeja son biestables, ya que permanecen activas hasta recibir una orden de desactivación. Mientras que las señales freno, control de cola y elevador son monoestables, y se desactivan automáticamente al cambiar de estado activo.

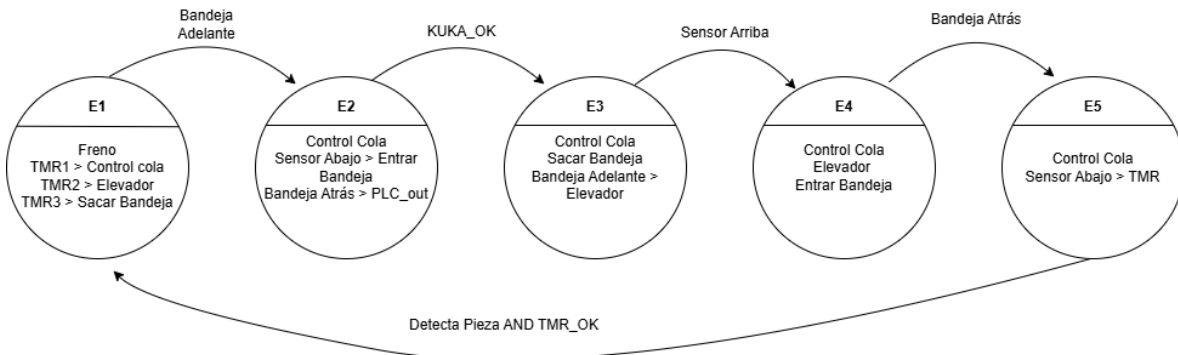


Figura 5.3: Diagrama de estados secuencia estación de transferencia.

5.1.2. Requisitos automatización robotizada

Corresponden a las acciones ejecutadas por cualquiera de los 2 robots KUKA, encargados de manipular los objetos dentro del sistema:

- Esperar la señal del PLC indicando la presencia de un pallet en la estación de trabajo.
- Ejecutar rutina *pick and place*, tomando cubos del pallet y colocándolos en una nueva posición dentro del mismo.
- Enviar una señal al PLC una vez finalizada la rutina para continuar con el proceso
- Monitorear continuamente el sistema en caso de interrupción de la rutina por la activación del interruptor de parada de emergencia.

5.2. Programa KRL de automatización robotizada de la FMS

Los programas que definen las rutinas de movimiento para los robots KUKA KR6 y KUKA KR15 se encuentran detallados en los Anexos A.3 y A.5. A continuación, se presentan los aspectos generales de cada rutina.

Cada robot ejecuta una secuencia de movimientos predefinida en lenguaje KRL (*KUKA Robot Language*), basada en las señales recibidas desde el PLC. Estas rutinas incluyen instrucciones de *pick and place* y sincronización con la estación de transferencia. El KR6 manipula los cubos en su estación de trabajo, mientras que el KR15 usa su eje lineal para trasladarlos entre estaciones.

La Tabla 5.1 presenta las señales de entrada y salida utilizadas por los robots. En ambos casos, las conexiones son idénticas, diferenciándose únicamente por el sufijo correspondiente en el PLC. Entre las señales de entrada, se incluye una que informa al robot sobre la presencia de un pallet en la estación de trabajo, lo que activa la rutina *pick and place*. Adicionalmente, se dispone de una señal de parada de emergencia, la cual, al activarse, interrumpe la rutina en ejecución.

Por otro lado, las señales de salida permiten el control de la apertura y cierre de la garra, así como la señal de notificación de finalización de la rutina *pick and place*.

Tabla 5.1: Entradas/Salidas robot KUKA.

Entradas Robot	Descripción
DI_00	Pallet en estación de trabajo
DI_01	Parada de emergencia
Salidas Robot	Descripción
DO_00	Cerrar garra
DO_01	Rutina completada

5.2.1. Rutina KUKA KR6

En la Figura 5.4 se muestran las posiciones iniciales y finales de los cubos en la estación de trabajo del robot KR6. Los cubos comienzan ubicados en una distribución específica dentro del pallet, y mediante la rutina *pick and place*, el robot los re posiciona en sus ubicaciones finales, que consisten en los mismos espacios iniciales, pero con los cubos rotados en sentido horario.

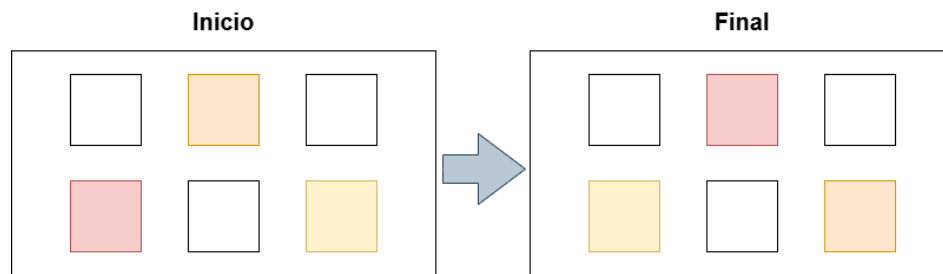


Figura 5.4: Posiciones cubos iniciales y finales estación KR6.

5.2.2. Rutina KUKA KR15

La Figura 5.5 muestra las posiciones iniciales y finales de los cubos en las dos estaciones de trabajo asignadas al robot KR15. La primera estación se encuentra en la cinta transportadora, mientras que la segunda está ubicada a unos metros de distancia, permitiendo que el robot utilice su eje lineal para trasladar los cubos entre ambas estaciones.

Para ejecutar este proceso, se sigue una rutina *pick and place* predefinida en el código del programa en KRL. Al igual que en la rutina del robot KR6, los cubos finalizan ocupando la misma disposición que tenían al inicio, pero en una nueva ubicación dentro del entorno de trabajo. Específicamente, los cubos que se encuentran inicialmente en la estación 1 terminan en la estación 2, mientras que los cubos de la estación 2 ocupan las posiciones originales de la estación 1.

Por este motivo, en la Figura 5.5 se utilizan colores diferentes para distinguir las posiciones de cada cubo, facilitando la visualización del intercambio entre ambas estaciones.

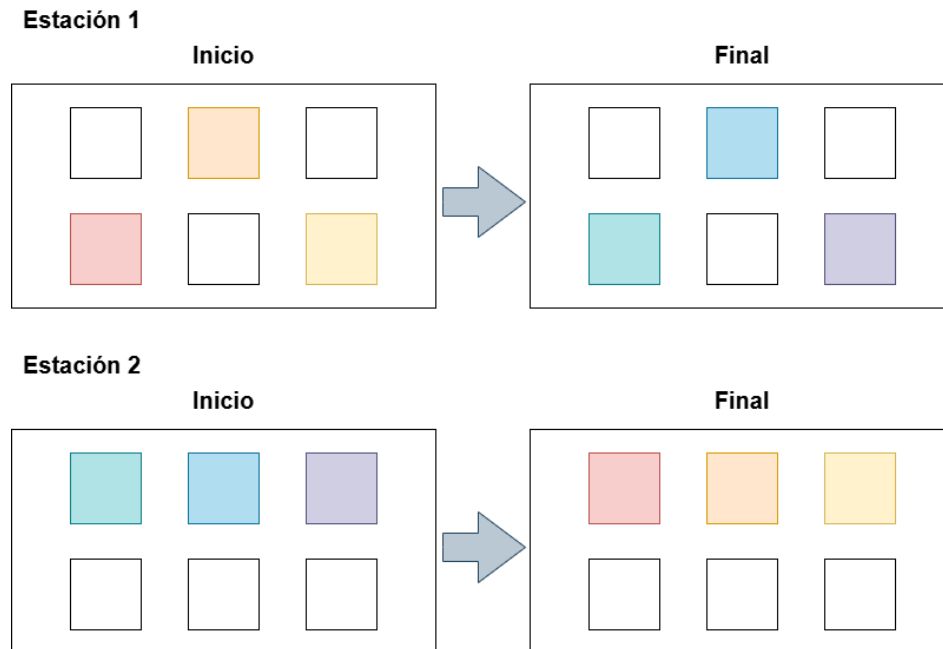


Figura 5.5: Posiciones cubos iniciales y finales estación KR15.

5.2.3. Validación de cumplimiento de requisitos de automatización robotizada de la FMS

Para la validación virtual de la automatización robotizada de la FMS, una vez cargado el PLC virtual en CODESYS con la programación en SFC y establecida la conexión con CIROS Studio mediante OPC, se inicia la simulación presionando **Iniciar simulación** en CODESYS y **Play** en CIROS Studio.

Una vez que se tiene habilitada la simulación en CIROS Studio y en CODESYS, el sistema queda en espera de que se presione el botón **START**.

Se presentan distintos momentos de la simulación para ilustrar el funcionamiento de las rutinas de los robots. En la Figura 5.6 se muestra el robot KR6 ejecutando el intercambio de cubos en el pallet. Se observa cómo el pallet está ubicado en la estación de trabajo, elevado por el elevador correspondiente y desplazado horizontalmente hacia el robot.

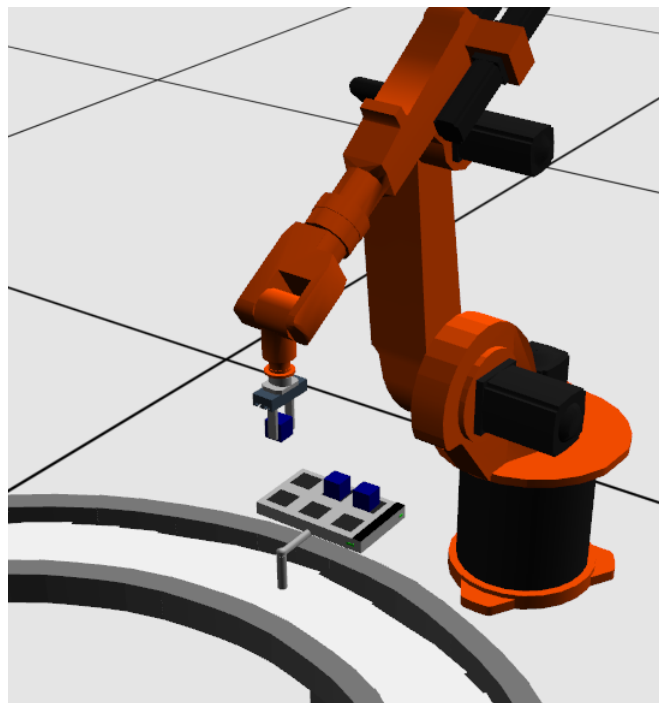


Figura 5.6: KR6 realizando rutina *pick and place*.

En las Figuras 5.7a y 5.7b se muestra el robot KR15 desplazándose a lo largo de su eje lineal, transportando cubos entre las estaciones de trabajo

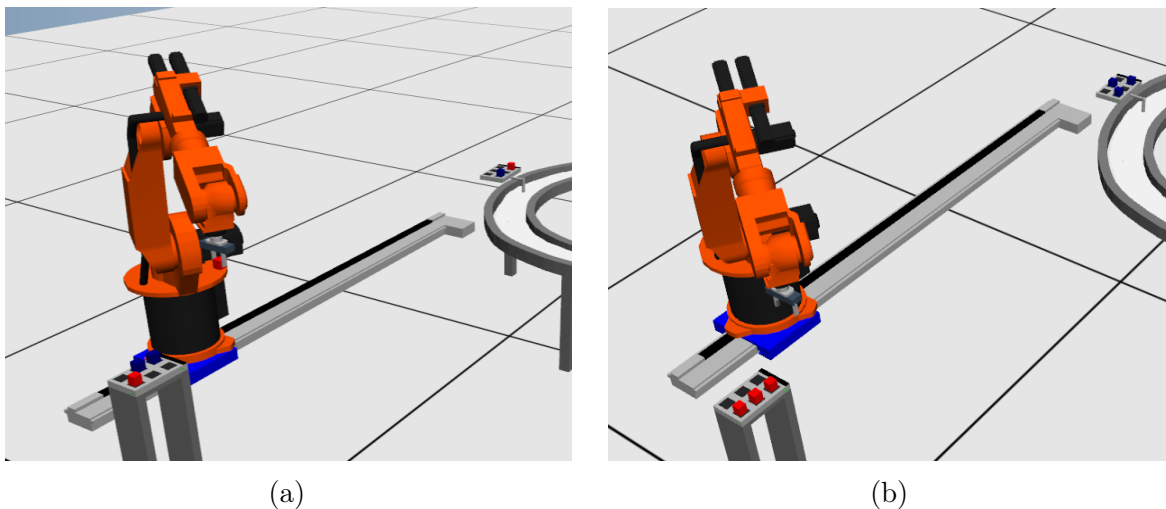


Figura 5.7: KR15 realizando rutina *pick and place*.

5.3. Programa SFC de automatización programada de la FMS

Para lograr la automatización programada de la FMS, es necesario definir una secuencia lógica de operación que el PLC ejecutará paso a paso, permitiendo la transición ordenada entre los distintos estados del sistema. Esta secuencia se diseña mediante un diagrama GRAFCET, el cual establece los estados y las condiciones necesarias para la transición entre ellos. Ese diseño se basa en el diagrama de estados mencionado previamente, adaptándolo a los mecanismos a utilizar en la simulación.

El primer paso consiste en dividir la rutina en múltiples estados bien definidos y luego determinar las condiciones de transición entre ellos. Las condiciones definidas deben cumplirse para que el sistema avance de un estado a otro.

Luego de dividir los estados y transiciones, se obtiene un diagrama representado en la Figura 5.8. La rutina está compuesta por 16 estados; después de los dos primeros, el diagrama se divide en 2 ramas alternativas correspondientes a la operación de las estaciones de transferencia de los robots.

El diagrama secuencial se interpreta de arriba hacia abajo. Cada estado contiene las acciones a ejecutar escritas a su costado, mientras que las transiciones están representadas por líneas horizontales entre los estados.

La rutina comienza al presionar el botón START, lo que activa el programa y conduce al estado E1, donde se inicia el movimiento de la cinta transportadora. Este y otros comandos del sistema pueden ejecutarse tanto desde los botones virtuales del modelo en CIROS Studio (Figura 4.7), como desde la interfaz HMI desarrollada en CODESYS (Figura 5.13b). Ambas opciones permiten iniciar y detener el ciclo automático, activar la parada de emergencia además de seleccionar cuál robot se encuentra activo.

Posteriormente, el diagrama se divide en dos ramas alternativas. Si el selector se encuentra en la posición “1”, se avanza al estado E2, activando la rutina del robot KR15. Para ello, se desactiva el elevador de la estación de transferencia del KR6 y se verifica que el elevador de la estación del KR15 esté en posición de espera.

Una vez que el pallet pasa por el sensor de aproximación en la estación de transferencia del KR15, se enciende un LED indicando su activación. Al detectar la presencia del pallet, el elevador se activa, lo eleva y transporta horizontalmente hasta su posición final en la estación de trabajo del robot, activando la señal “Estación OK”, lo que indica al

robot que debe iniciar su rutina *pick and place*.

Una vez que el robot finalice su rutina y active la señal “KUKA OK”, se activa el movimiento de retorno del pallet hacia la cinta transportadora. En este punto, el sistema se presenta con dos opciones: si se presiona el pulsador STOP, la cinta se detiene y el sistema regresa al estado E0; si no se presiona el pulsador en un período de 10 segundos, el sistema retorna al estado E1 y se reevalúa la posición del selector para decidir la siguiente rama a seguir.

Si el selector está en posición “0”, se sigue la rama correspondiente al robot KR6, repitiendo la misma secuencia descrita anteriormente, pero aplicada a su estación de transferencia.

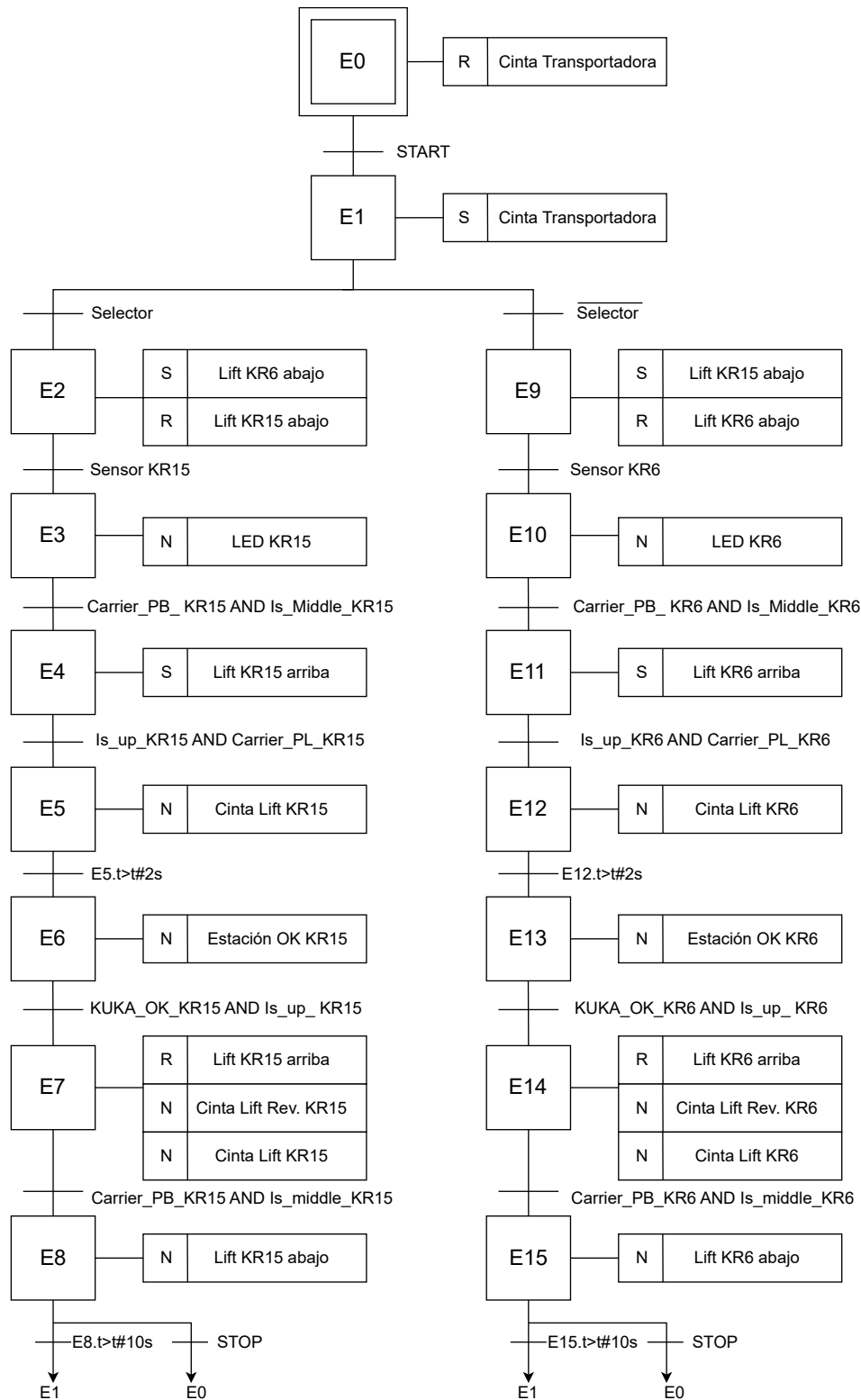


Figura 5.8: Diagrama GRAFCET FMS-MAIN.

Para asegurar el correcto funcionamiento de la interrupción por parada de emergencia y la habilitación del modo de operación manual, se implementa un diagrama GRAFCET general, como se muestra en la Figura 5.9, compuesto por cuatro estados: INIT, MAIN, MANUAL y STOP. La transición entre INIT y MAIN es siempre verdadera, por lo que, al iniciar el programa, el sistema avanza automáticamente al estado MAIN.

Por defecto, el modo automático está activado, permitiendo la ejecución de la rutina programada sin intervención manual. En el estado MAIN, el sistema opera normalmente ejecutando la rutina descrita previamente. Cabe señalar que la selección entre los modos de operación automático y manual sólo puede realizarse desde la interfaz HMI. Si se presiona el interruptor de parada de emergencia mientras el modo automático está activo, el sistema realiza la transición al estado STOP, deteniendo la cinta transportadora y activando la señal de interrupción a los robots, lo que provoca que estos detengan su rutina hasta que la señal de parada de emergencia sea desactivada.

Para acceder al modo manual, es necesario activarlo explícitamente, deshabilitando así el modo automático. Cuando esto ocurre, el sistema realiza la transición al estado MANUAL, donde se desactiva la secuencia programada y se permite operar los componentes individualmente a través de los botones en el HMI.

En este estado, si se activa la parada de emergencia, como se mencionó anteriormente, todos los mecanismos, incluidas las rutinas de los robots, se detendrán hasta que se desactive la señal de parada de emergencia. Si el modo manual se desactiva, el sistema volverá al estado MAIN.

Cuando la señal del interruptor de parada de emergencia es desactivada, el diagrama retorna al estado MAIN, permitiendo que el proceso continúe desde el punto en el que se encontraba al momento de la interrupción.

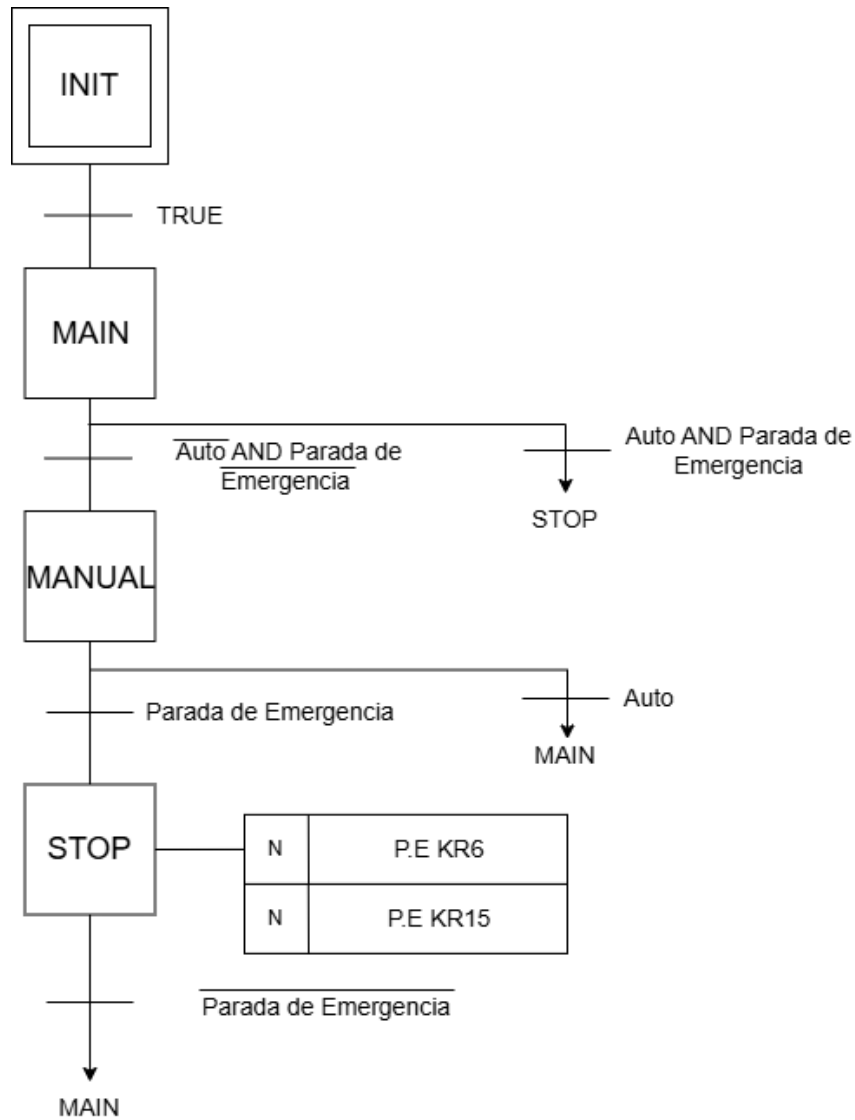


Figura 5.9: Diagrama GRAFCET General.

Este mecanismo se logra utilizando la bandera disponible en CODESYS **SFC Pause**, la cual se declara en el software de la siguiente manera:

Listing 5.1: Código SFC en CODESYS

```

1 PROGRAM MAIN_PRG
2 VAR_INPUT
3     SFCPause : BOOL;
4 END_VAR
    
```

El uso de **SFC Pause** permite que el sistema detenga la secuencia programada sin perder la información del estado activo. Es decir, al activarse la señal de parada de emergencia PE, el valor de **SFCPause** pasa a ser verdadero, lo que provoca que el GRAFCET detenga su avance, manteniendo activos los estados que estaban en ejecución al momento de la interrupción, pero evitando que se evalúen nuevas transiciones. Esta herramienta es fundamental para garantizar que el proceso pueda reanudarse exactamente desde el punto en que se encontraba una vez desactivada la señal de parada de emergencia, sin necesidad de reiniciar la secuencia completa.

Esto se implementa como se muestra en la Figura 5.10, la cual representa la acción asociada a la salida del estado MAIN cuando se cumple la transición de parada de emergencia hacia el estado STOP. En esta acción, se activa la variable **SFCPause**, lo que permite detener la secuencia programada hasta que se restablezcan las condiciones normales de operación.

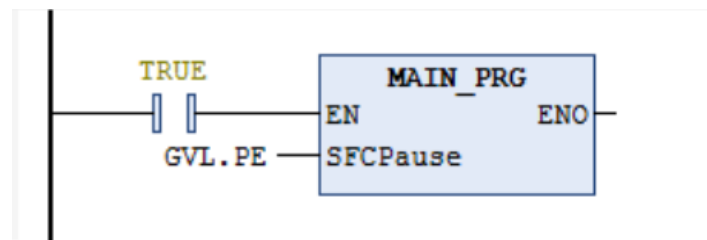
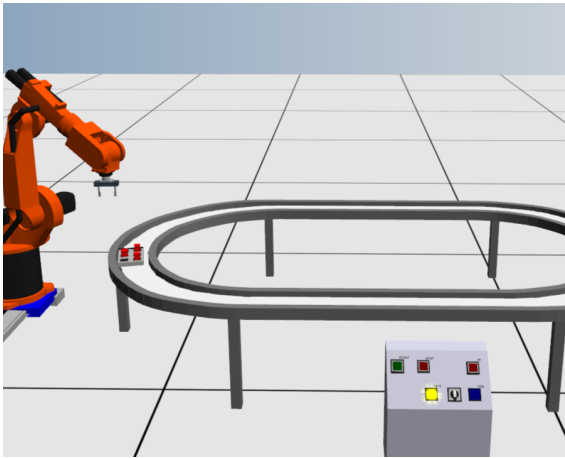


Figura 5.10: Activación de SFC Pause.

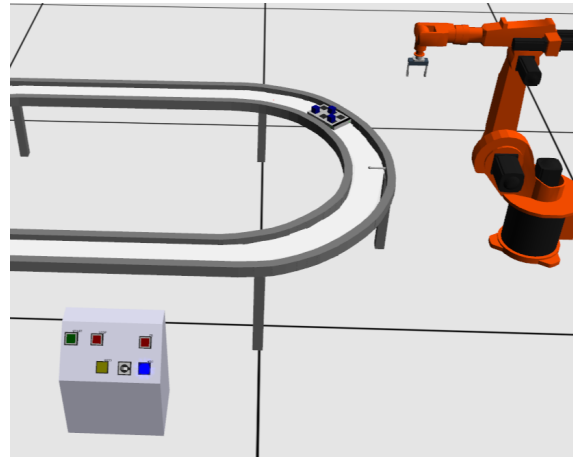
5.3.1. Validación de cumplimiento de requisitos de automatización programada de la FMS

De manera similar a la validación del cumplimiento de los requisitos de automatización robotizada, esta validación de requerimientos presenta distintos momentos de la simulación para ilustrar el funcionamiento del sistema.

En las Figuras 5.11a y 5.11b se observan los LEDs de indicación de aproximación de los pallets, activando la secuencia de movimiento del elevador para la transferencia de los pallets desde la cinta transportadora hacia las estaciones de trabajo correspondientes.



(a) LED aproximación encendido estación KR15.



(b) LED aproximación encendido estación KR6.

Figura 5.11: Funcionamiento LED de aproximación.

En la Figura 5.12 se observa el pallet siendo elevado y transportado por el elevador de la estación de transferencia KR15, para que el robot pueda realizar su rutina de intercambio de cubos. Cabe recordar que no se simula el sistema de freno ni control de cola, ya que el componente *Lift transfer unit with stopper* incorpora internamente estas funciones.

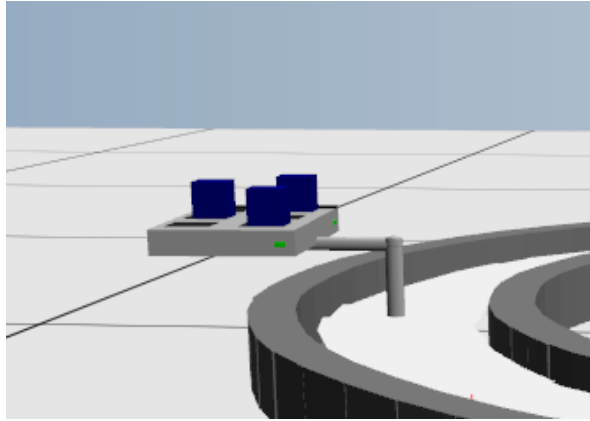


Figura 5.12: Elevador estación KR15 en simulación.

5.4. Programa HMI de supervisión del sistema integrado

La pantalla se diseña con tres imágenes: la primera, mostrada en la Figura 5.13a, es la imagen principal, desde la cual el usuario puede desplazarse a las otras dos imágenes. La segunda imagen, en la Figura 5.13b, muestra el sistema y permite activar los pulsadores del panel de simulación de manera remota, así como ver los estados de los sensores de proximidad. La tercera imagen, representada en la Figura 5.13c, permite la operación manual del sistema, habilitando el control de los actuadores para ser activados o desactivados según sea necesario.

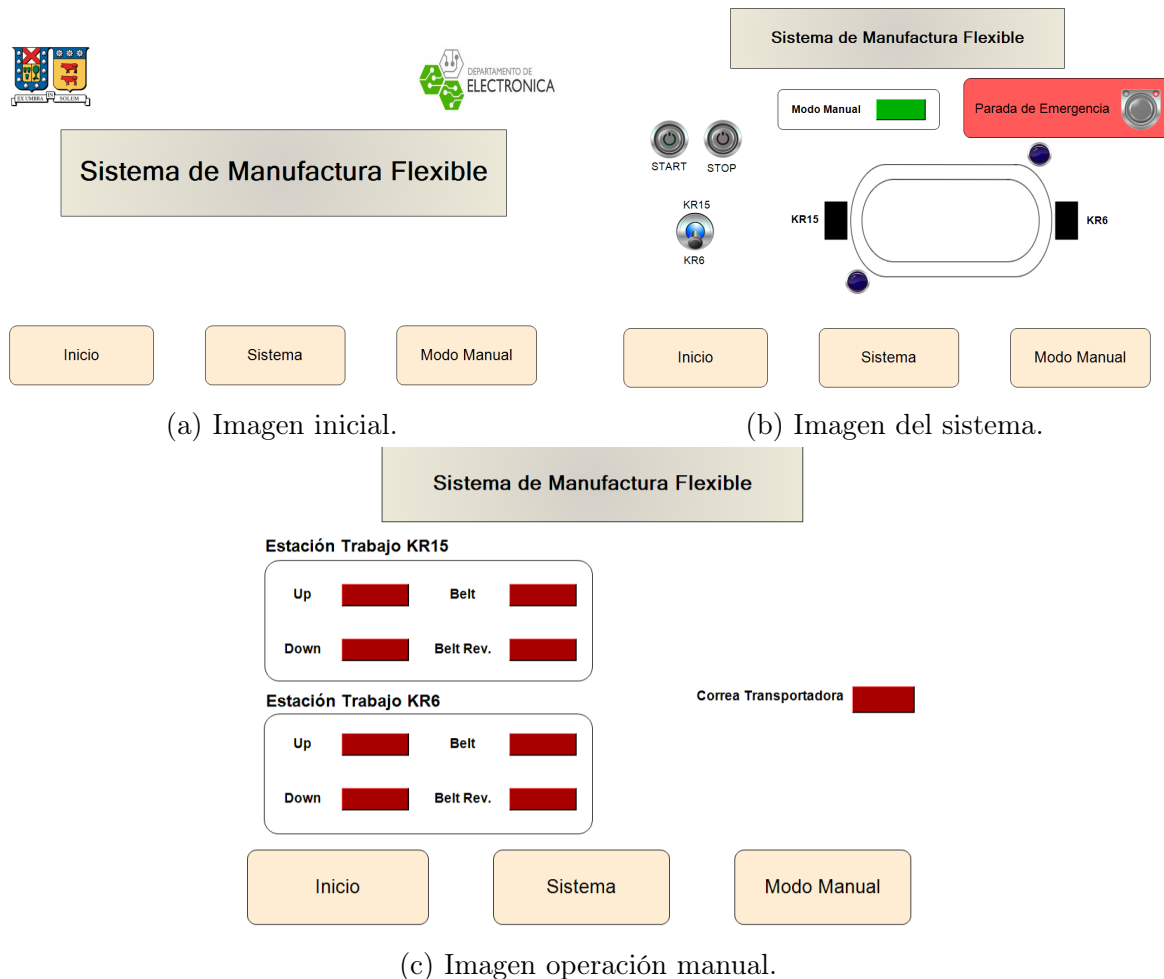


Figura 5.13: HMI.

5.4.1. Validación de cumplimiento de requisitos de supervisión del sistema integrado

Para validar los requisitos de supervisión del sistema integrado, se presentan dos casos de funcionamiento del HMI. En la Figura 5.14 se muestra el cambio de color de los botones correspondientes a la operación manual, específicamente los botones para subir el elevador y activar la cinta del elevador de la estación de trabajo KR15, así como para activar la cinta transportadora general.

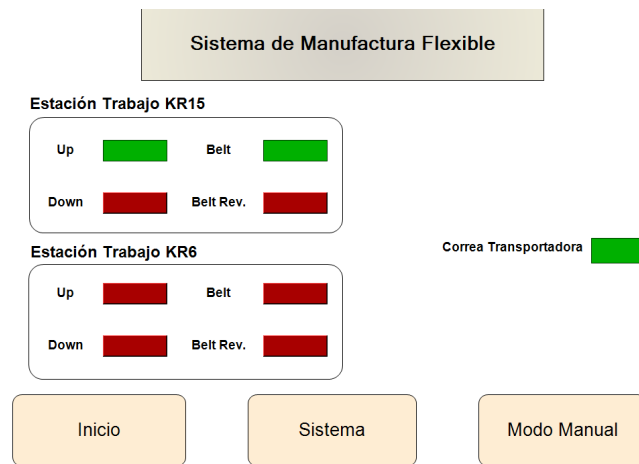


Figura 5.14: Prueba HMI modo manual.

En la Figura 5.15 se presenta la imagen del sistema en funcionamiento con el modo manual desactivado, mostrando la estación de trabajo KR6 activada y el sensor de proximidad de dicha estación encendido.

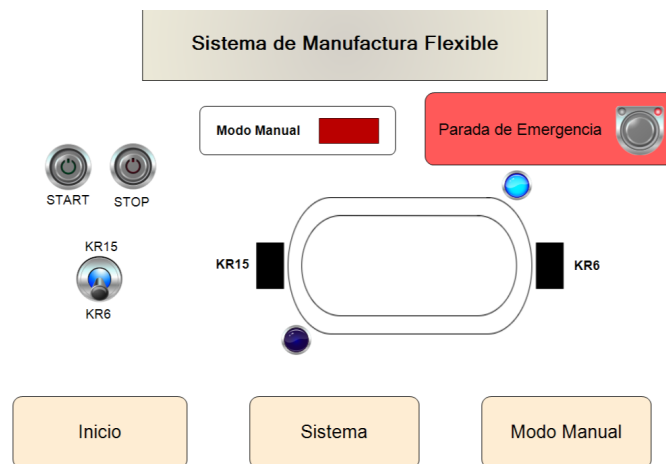


Figura 5.15: Prueba HMI imagen del sistema.

6 | Análisis de resultados y conclusiones

Se implementó exitosamente la automatización programada y robotizada del sistema de manufactura flexible en la simulación 3D de CIROS Studio, cumpliendo con los requisitos funcionales establecidos por el Laboratorio de Control Industrial. La validación en tiempo real permitió comprobar el funcionamiento de las distintas partes del sistema, asegurando la funcionalidad de la secuencia programada y la ejecución de los estados.

Se diseñó y validó la automatización a través de un diagrama GRAFCET en el PLC virtual, permitiendo la sincronización entre los robots, el elevador y la cinta transportadora. Además, los sensores desempeñaron un papel clave en la detección de la posición de los pallets, facilitando la activación de los frenos de estos en su debido momento.

Durante el diseño del mecanismo de transferencia de los pallets, se realizaron ajustes en la configuración de los actuadores. Se reemplazaron los múltiples cilindros para la elevación y el desplazamiento horizontal por un componente disponible en las librerías de CIROS Studio, el cual integra ambas funciones, simplificando la configuración del modelo. Si bien el programa SFC utilizado en la simulación no es idéntico al que se implementaría físicamente, la lógica de automatización se mantiene equivalente.

En términos de integración, la conexión entre CODESYS y CIROS Studio permitió validar la interacción entre la lógica de control y la simulación virtual, asegurando la funcionalidad de las instrucciones en cada etapa del proceso.

Se comprobó la estabilidad de la comunicación entre ambas plataformas, garantizando la sincronización del sistema y la ejecución de las rutinas de automatización. Este enfoque se relaciona con la metodología *Software-in-the-loop*, que permite probar y validar software antes de su implementación física, lo que optimiza el proceso de desarrollo y ayuda a reducir posibles errores.

Se estableció la conexión directa entre CODESYS y CIROS Studio utilizando el PLC virtual Control Win V3 mediante protocolo OPC. A diferencia de otros proyectos que requieren múltiples programas para la conexión, este método simplificó la implementación del sistema virtual.

Se desarrolló una interfaz HMI virtual para la supervisión del FMS. Incluye un panel de pulsadores y LEDs que indican el estado del sistema, además de permitir la operación manual de los mecanismos cuando sea necesario.

6.1. Aportes

Se desarrolló una simulación dinámica 3D en tiempo real del sistema de manufactura flexible disponible en CIMA, como parte de la experiencia del Laboratorio de Control Industrial. Esta simulación permite la puesta en marcha de la automatización del sistema mediante un sistema de control virtual, equivalente al sistema físico, utilizando herramientas ampliamente utilizadas en la industria, como CODESYS y el protocolo OPC.

En proyectos anteriores, CIROS Studio se ha utilizado para simular un Sistema de Transporte y Clasificación, programando y automatizando la secuencia de control de componentes electroneumáticos. Asimismo, en otra ocasión, se empleó este programa para programar los movimientos en una rutina de un robot RV2AJ, registrando sus posiciones y desarrollando una rutina de interrupción. Ambos enfoques han permitido explorar distintas aplicaciones de automatización en entornos virtuales. En este proyecto, se integraron ambos enfoques en una única simulación, diseñando un entorno virtual capaz de controlar de manera integral un sistema de manufactura flexible. Se implementó un sistema que gestiona la cinta transportadora, los sensores y el mecanismo de elevación, además de programar rutinas de movimiento para dos robots KUKA. Como resultado, se obtuvo una simulación donde todos estos elementos operan en conjunto de manera sincronizada, permitiendo la validación del sistema de automatización.

Este trabajo representa un avance en la simulación y validación de sistemas industriales para el Laboratorio de Control Industrial, proporcionando una plataforma para la enseñanza, el desarrollo y la optimización de procesos de manufactura flexible en entornos académicos e industriales.

6.2. Trabajos futuros

Durante el desarrollo de este proyecto, se planteó la idea de implementar el modelo en un esquema HIL, lo cual no fue posible debido a las limitaciones de los softwares y del PLC actualmente disponible en CIMA. Como trabajo a futuro, se propone la posibilidad de reemplazar el PLC físico por uno más moderno y compatible con la simulación, lo que permitiría realizar la conexión de manera efectiva.

Como base para el desarrollo de estas futuras integraciones, se propone un esquema conceptual que representa la interacción entre los distintos entornos de simulación y control utilizados, CIROS Studio, CODESYS y los componentes físicos disponibles en CIMA. Este esquema, mostrando en la Figura 6.1 ilustra una posible arquitectura de integración HIL/SIL en la que se combinan componentes virtuales como los componentes de CIROS Studio, el programa SFC e interfaz HMI en CODESYS junto con el PLC físico en CIMA.

Por otro lado, CIROS Studio permite simular diversos entornos y situaciones, lo que abre la posibilidad de agregar un panel de operación que permita controlar manualmente el modelo, sin necesidad de una conexión directa con un PLC en primera instancia. Además, se podría mejorar el modelo mediante la incorporación de más sensores o mecanismos, con el fin de aumentar su complejidad y acercarlo más a la simulación del sistema físico real disponible en el CIMA.

También se propone como mejora futura el modelado explícito de los mecanismos de freno y control de cola en las estaciones de transferencia de ambos robots. Esta incorporación permitiría una representación más realista del comportamiento físico, mejorando su valor como herramienta de validación.

Adicionalmente, se sugiere la integración virtual de un sistema automático de almacenamiento y recuperación (AS/RS, por sus siglas en inglés) al modelo desarrollado en esta memoria. Esto permitiría extender las funcionalidades del sistema de manufactura flexible.

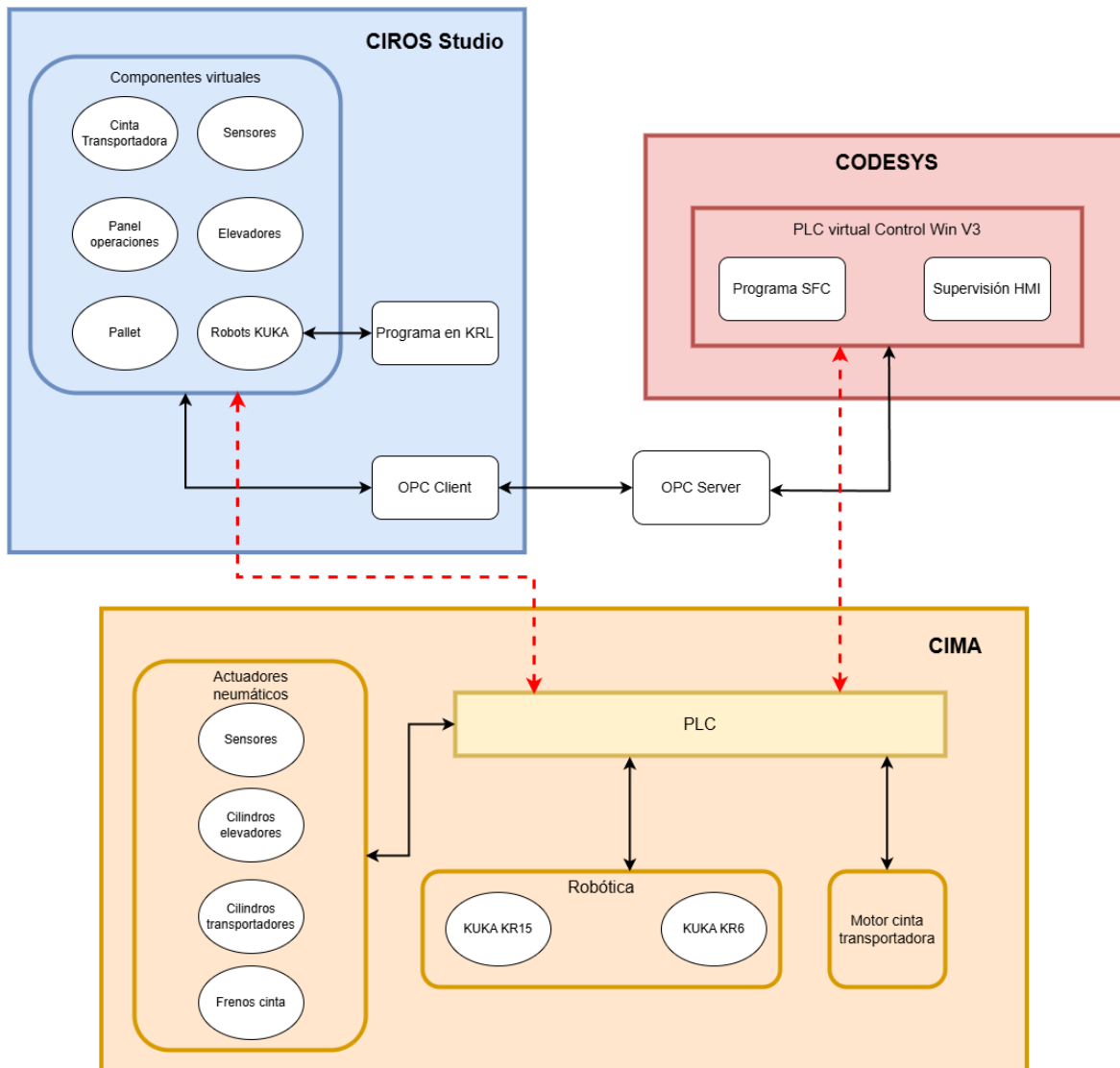


Figura 6.1: Diagrama conexiones subsistemas.

Bibliografía

- [1] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, 4th ed. Pearson, 2015.
- [2] “Documentacion Experiencia Automatización de una Celda de Manufactura Flexible (CIMA),” n.d.
- [3] Z. Kovacic, S. bogdan, N. Smolic-Rocak, and B. Birgmajer, “Teaching flexible manufacturing systems by using design and simulation program tools,” in *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, vol. 1, 2003, pp. 47–51 vol.1.
- [4] Anylogic. Accessed: 2024-05-19. [Online]. Available: <https://www.anylogic.com/>
- [5] Factory io. Accessed: 2024-05-19. [Online]. Available: <https://factoryio.com/>
- [6] Factory io. Accessed: 2024-05-19. [Online]. Available: <https://www.programacionmultidisciplinar.com/factory-io/>
- [7] M. Herrera, “Simulación, automatización y puesta en marcha virtual de un sistema de transporte y clasificación,” Memoria de título de grado, Universidad Técnica Federico Santa María, 2023.
- [8] “Información Experiencia 4 - Automatización Robotizada,” n.d.
- [9] MathWorks, “Software-in-the-loop simulation,” 2025, accessed: 2025-02-13. [Online]. Available: <https://www.mathworks.com/help/ecoder/software-in-the-loop-sil-simulation.html>
- [10] —, “Pruebas hardware-in-the-loop (hil) para sistemas de electrónica de potencia,” 2025, accessed: 2025-02-13. [Online]. Available: <https://es.mathworks.com/solutions/electrification/hardware-in-the-loop.html>

- [11] E. Taborova, “Animation and simulation applications for education,” Master’s thesis, Saimaa University of Applied Sciences, 2015. [Online]. Available: <https://www.theseus.fi/handle/10024/92734>
- [12] *Especificaciones técnicas Robots KR6, Kr16, KR16-L6.*, KUKA, 2003, accessed: 2025-02-04. [Online]. Available: <https://www.construnario.com/ebooks/9365/especificaciones/kr%206/files/kr%206.pdf>
- [13] *Datos técnicos Robots KR15*, KUKA, 1998, accessed: 2025-02-04. [Online]. Available: https://www.cemausa.com/docs/ro_me_01_13_02_es.pdf
- [14] L. I. Systems, *GLOFA-GM4: Programmable Logic Controllers User Manual*, LG Industrial Systems, incluye especificaciones técnicas de los módulos G4Q-RY2A y G4I-D22A. [Online]. Available: http://www.ehaegypt.com/uploads/GM4_06201yq4.pdf
- [15] “Codesys programming - support | festo usa,” https://www.festo.com/us/en/e/support/programming-id_21578/, January 2025, accessed: 2025-01-13.
- [16] *Sequential Function Chart (SFC)*, Siemens, 2010, accessed: 2025-02-13. [Online]. Available: <https://www.automation.siemens.com/sce-static/learning-training-documents/pcs7/v7-0/p01-07-sequential-function-chart-rc1012-en.pdf>
- [17] A. Lightstead, “What is a pick and place robot? uses and types,” *robotics247*, 2022, accessed: 2025-02-19. [Online]. Available: https://www.robotics247.com/article/what_is_a_pick_and_place_robot_uses_and_types

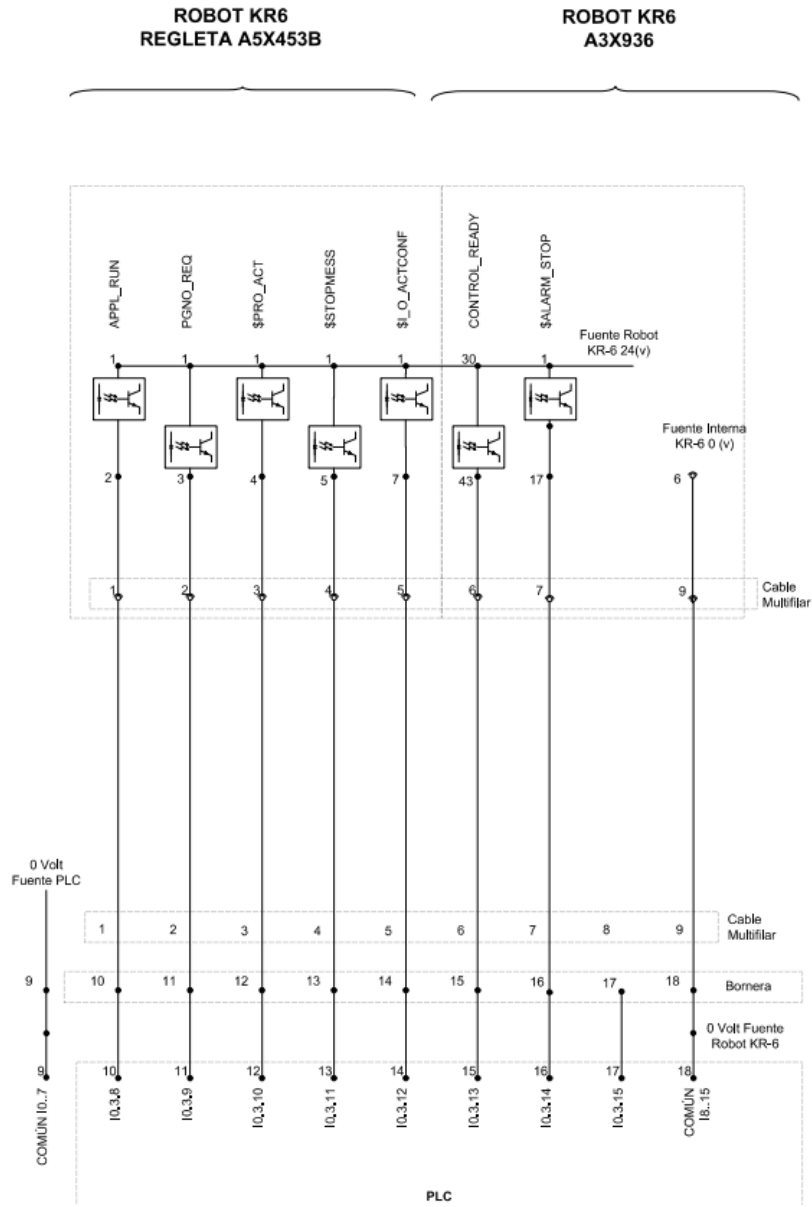


Figura A.2: Diagrama eléctrico conexiones robot KR6.

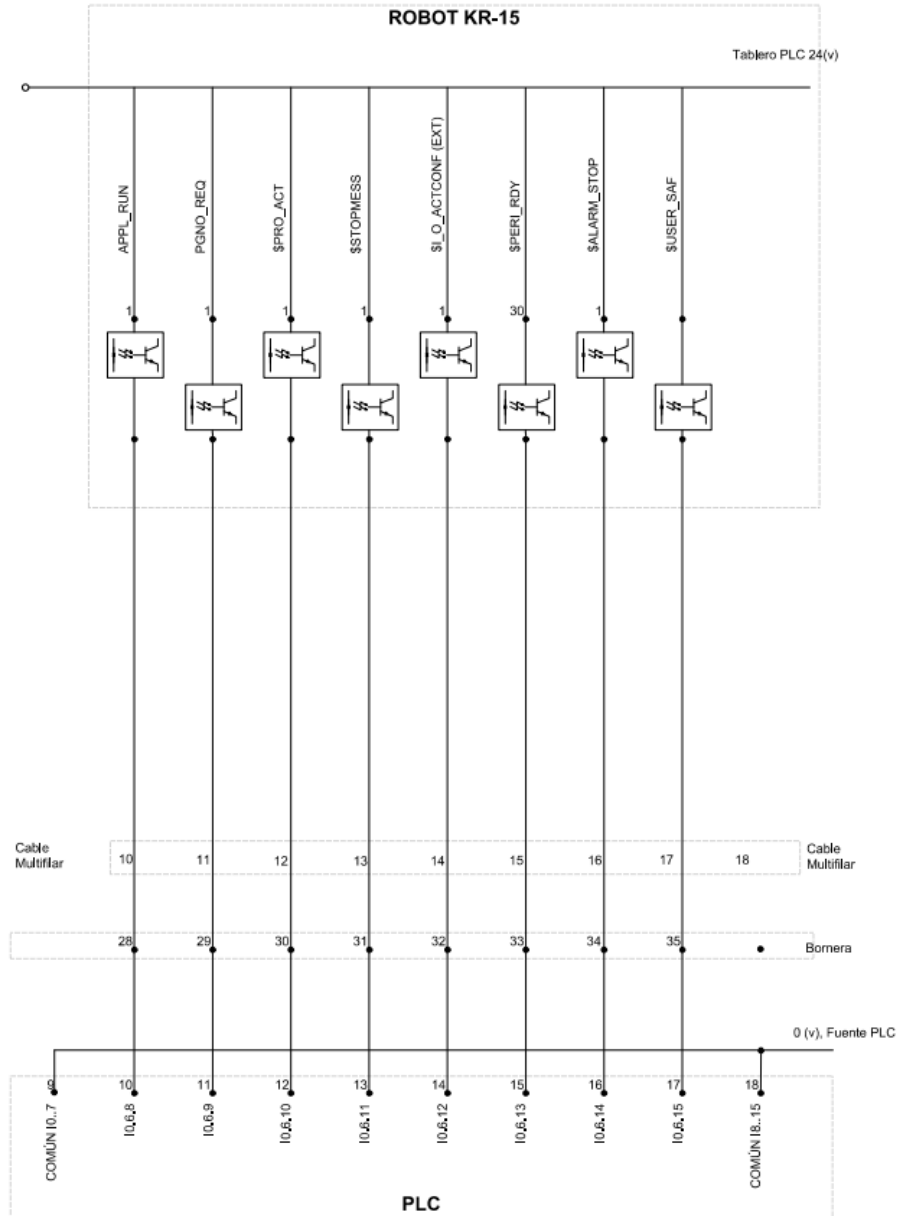


Figura A.3: Diagrama eléctrico conexiones robot KR6.

A.2. Diagrama eléctrico de conexiones de salida

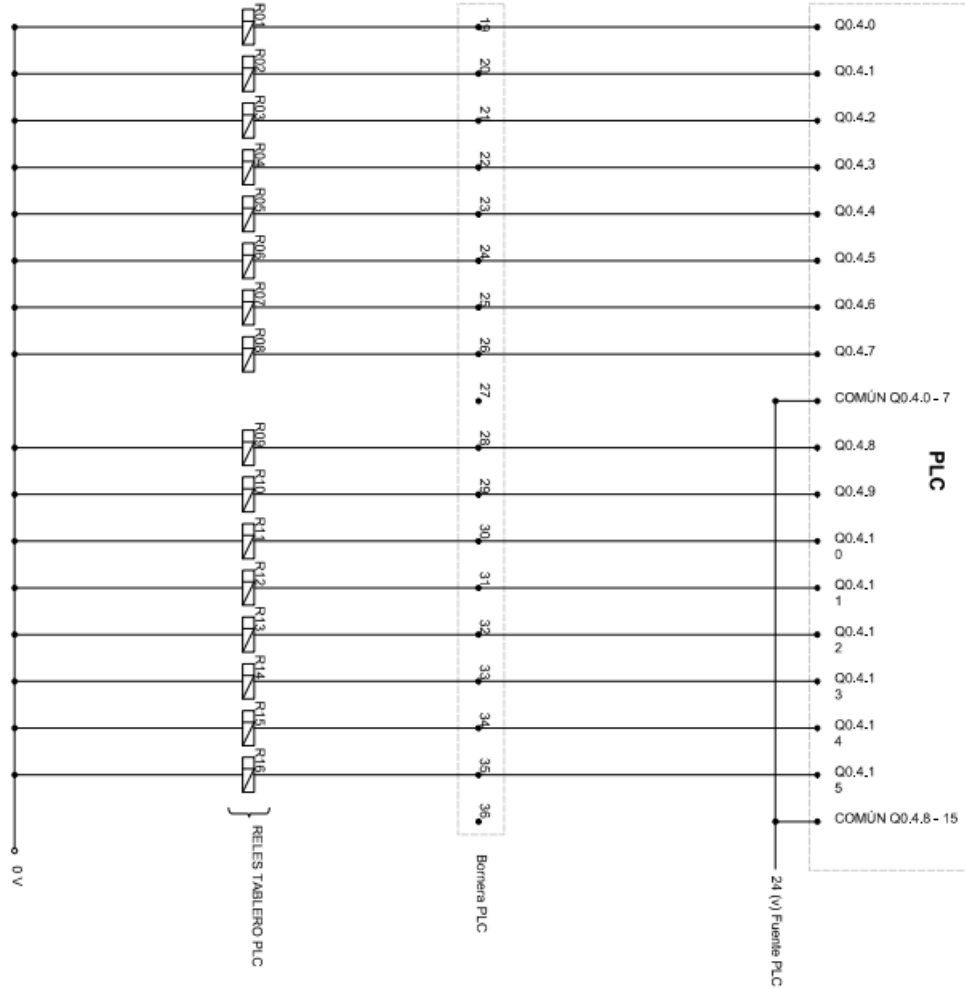


Figura A.4: Diagrama eléctrico salidas PLC.

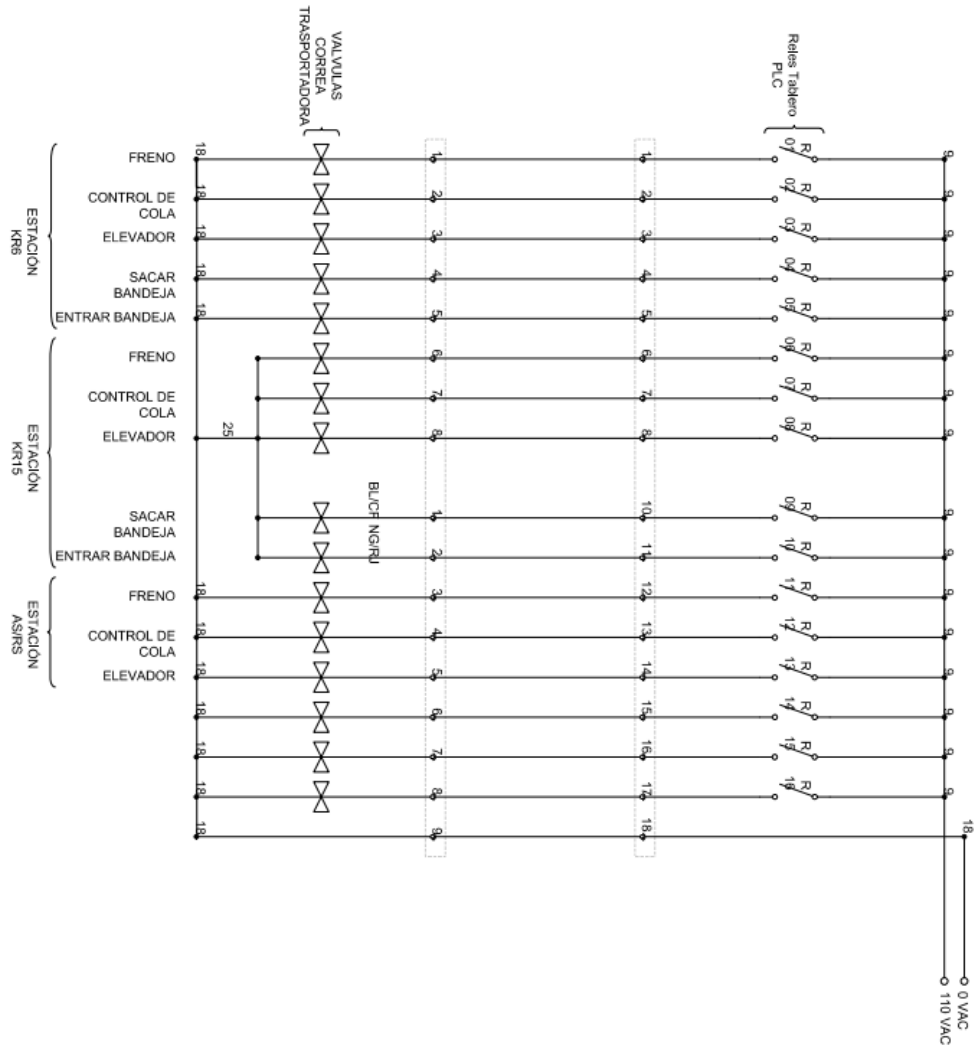


Figura A.5: Diagrama eléctrico estaciones de transferencia.

A.3. Código rutina robot KR15

```
1 DEF Prueba_KR15 ()
2     INTERRUPT DECL 10 WHEN $IN[1] == TRUE DO PAUSE_ROUTINE_KR15()
3         ; Declara la interrupción para la señal IN[1]
4     INTERRUPT ON 10 ;ativa la interrupción
5     LOOP
6         $OUT[1]=FALSE ;para asegurarse que la señal está apagada
7         WAIT FOR $IN[0] == TRUE ;espera la señal del carrier en la
8             estacion de trabajo
9         LIN POS1
10        WAIT SEC 1
11        LIN POS2
12        $OUT[0]=TRUE ;cierra gripper
13        WAIT SEC 2
14        LIN POS1
15        WAIT SEC 1
16        PTP POS3
17        LIN POS4
18        $OUT[0]=FALSE ;abre gripper
19        WAIT SEC 2
20        LIN POS3
21        PTP POS5
22        WAIT SEC 1
23        LIN POS6
24        $OUT[0]=TRUE ;cierra gripper
25        WAIT SEC 2
26        LIN POS5
27        WAIT SEC 1
28        PTP POS1
29        LIN POS2
30        $OUT[0]=FALSE ;abre gripper
31        WAIT SEC 2
32        LIN POS1
33        WAIT SEC 1
34        PTP POS7
35        LIN POS8
36        $OUT[0]=TRUE ;cierra gripper
37        WAIT SEC 2
38        LIN POS7
39        WAIT SEC 1
40        PTP POS9
41        LIN POS10
```

```
40      $OUT[0]=FALSE ;abre gripper
41      WAIT SEC 2
42      LIN POS9
43      PTP POS11
44      WAIT SEC 1
45      LIN POS12
46      $OUT[0]=TRUE ;cierra gripper
47      WAIT SEC 2
48      LIN POS11
49      PTP POS7
50      WAIT SEC 1
51      LIN POS8
52      $OUT[0]=FALSE ;abre gripper
53      WAIT SEC 2
54      LIN POS7
55      PTP POS13
56      WAIT SEC 1
57      LIN POS14
58      $OUT[0]=TRUE ;cierra gripper
59      WAIT SEC 2
60      LIN POS13
61      PTP POS15
62      WAIT SEC 1
63      LIN POS16
64      $OUT[0]=FALSE ;abre gripper
65      WAIT SEC 2
66      LIN POS15
67      PTP POS17
68      WAIT SEC 1
69      LIN POS18
70      $OUT[0]=TRUE ;cierra gripper
71      WAIT SEC 2
72      LIN POS17
73      WAIT SEC 1
74      PTP POS13
75      LIN POS14
76      $OUT[0]=FALSE ;abre gripper
77      WAIT SEC 2
78      LIN POS13
79      WAIT SEC 1 ; ACA TERMINA EL CAMBIO
80      PTP POS3
81      LIN POS4
82      $OUT[0]=TRUE ;cierra gripper
83      WAIT SEC 2
```

```
84     LIN POS3
85     LIN POS5
86     WAIT SEC 1
87     LIN POS6
88     $OUT[0]=FALSE ;abre gripper
89     WAIT SEC 2
90     LIN POS5
91     WAIT SEC 1
92     LIN POS9
93     LIN POS10
94     $OUT[0]=TRUE ;cierra gripper
95     WAIT SEC 2
96     LIN POS9
97     LIN POS11
98     WAIT SEC 1
99     LIN POS12
100    $OUT[0]=FALSE ;abre gripper
101    WAIT SEC 2
102    LIN POS11
103    WAIT SEC 1
104    LIN POS15
105    LIN POS16
106    $OUT[0]=TRUE ;cierra gripper
107    WAIT SEC 2
108    LIN POS15
109    WAIT SEC 1
110    LIN POS17
111    LIN POS18
112    $OUT[0]=FALSE ;abre gripper
113    WAIT SEC 2
114    LIN POS17
115    PTP HOME
116    $OUT[1]=TRUE ;manda señal kuka_ok
117    WAIT SEC 5
118    ENDLOOP
119    END
120    DEF PAUSE_ROUTINE_KR15()
121        INTERRUPT OFF 10 ; Desactiva la interrupción para evitar
            múltiples activaciones
122        LIN HOME ; Va a la posición HOME
123        WAIT FOR $IN[1] == FALSE ; Espera a que la señal se desactive
124        INTERRUPT ON 10 ; Reactiva la interrupción
125    END
```

A.4. Lista de posiciones robot KR15

Name	Position	Orientation
HOME	900.0, 0.0, 1340.0	-180, -0, -180, 0.0
POS1	380.5, 832.6, 1000.0	-86, 0, 180, 0.0, R, A, N
POS2	380.5, 832.6, 720.0	-86, 0, 180, 0.0, R, A, N
POS3	651.2, 956.5, 1000.0	-90, -0, 180, 2652.6
POS4	651.2, 956.5, 720.0	-90, 0, -180, 2652.6, R, A, N
POS5	651.2, 876.0, 1000.0	-90, 0, 180, 2652.6
POS6	651.2, 876.0, 720.0	-90, 0, 180, 2652.6, R, A, N
POS7	311.5, 748.5, 1000.0	-85, 0, 180, 0.0
POS8	311.5, 748.5, 720.0	-85, 0, 180, 0.0
POS9	573.7, 956.5, 1000.0	-90, 0, 180, 2652.6
POS10	573.7, 956.5, 720.0	-90, 0, 180, 2652.6, R, A, N
POS11	573.7, 876.0, 1000.0	-90, 0, 180, 2652.6
POS12	573.7, 876.0, 720.0	-90, 0, 180, 2652.6, R, A, N
POS13	224.9, 822.2, 1000.0	-85, 0, 180, 0.0
POS14	224.9, 822.2, 720.0	-85, 0, 180, 0.0
POS15	489.8, 956.5, 1000.0	-90, 0, 180, 2652.6
POS16	489.8, 956.5, 720.0	-90, 0, 180, 2652.6, R, A, N
POS17	489.8, 876.0, 1000.0	-90, 0, 180, 2652.6
POS18	489.8, 876.0, 720.0	-90, 0, 180, 2652.6, R, A, N

Figura A.6: Lista de posiciones robot KR15.

A.5. Código rutina robot KR6

```
1  DEF prueba_robot_KR6 ()
2
3  INTERRUPT DECL 10 WHEN $IN[1] == TRUE DO PAUSE_ROUTINE_KR6()
4      ; Declara la interrupción para la señal IN[1]
5  INTERRUPT ON 10 ; Activa la interrupción
6
7  LOOP
8  $OUT[1]=FALSE ;para asegurarse que la señal está apagada
9  WAIT FOR $IN[0] == TRUE ;espera la señal del carrier en la
10     estacion de trabajo
11  LIN POS3
12  WAIT SEC 1
13  LIN POS4
14  $OUT[0]=TRUE ;cierra gripper
15  WAIT SEC 2
16  LIN POS3
17  WAIT SEC 1
18  LIN POS7
19  WAIT SEC 1
20  LIN POS8
21  $OUT[0]=FALSE ;abre gripper
22  WAIT SEC 2
23  LIN POS7
24  LIN POS1
25  WAIT SEC 1
26  LIN POS2
27  $OUT[0]=TRUE ;cierra gripper
28  WAIT SEC 2
29  LIN POS1
30  LIN POS3
31  WAIT SEC 1
32  LIN POS4
33  $OUT[0]=FALSE ;abre gripper
34  WAIT SEC 2
35  LIN POS3
36  LIN POS5
37  WAIT SEC 1
38  LIN POS6
39  $OUT[0]=TRUE ;cierra gripper
40  WAIT SEC 2
41  LIN POS5
```

```
40     LIN POS1
41     WAIT SEC 1
42     LIN POS2
43     $OUT[0]=FALSE ;abre gripper
44     WAIT SEC 2
45     LIN POS1
46     LIN POS7
47     WAIT SEC 1
48     LIN POS8
49     $OUT[0]=TRUE ;cierra gripper
50     WAIT SEC 2
51     LIN POS7
52     LIN POS5
53     WAIT SEC 1
54     LIN POS6
55     $OUT[0]=FALSE ;abre gripper
56     WAIT SEC 2
57     LIN POS5
58     LIN HOME
59     $OUT[1]=TRUE ;manda señal kuka_ok
60     WAIT SEC 5
61     ENDLOOP
62 END
63
64 DEF PAUSE_ROUTINE_KR6()
65     INTERRUPT OFF 10 ; Desactiva la interrupción para evitar
66     múltiples activaciones
67     LIN HOME ; Va a la posición HOME
68     WAIT FOR $IN[1] == FALSE ; Espera a que la señal se desactive
69     INTERRUPT ON 10 ; Reactiva la interrupción
70 END
```

A.6. Lista de posiciones robot KR6

Name	Position	Orientation
HOME	900.0, 0.0, 1355.0	-180, -0, 180
POS1	978.1, -51.0, 1000.0	178, -0, -180
POS2	978.1, -51.0, 851.9	178, 0, -180
POS3	899.0, 27.3, 1000.0	178, -0, 180
POS4	899.0, 27.3, 851.9	178, -0, 180
POS5	978.1, 103.4, 1000.0	178, 0, -180
POS6	978.1, 103.4, 851.9	178, 0, -180
POS7	899.0, 103.4, 1000.0	178, 0, 180
POS8	899.0, 103.4, 851.9	178, -0, -180

Figura A.7: Lista de posiciones robot KR6.

A.7. Variables globales PLC virtual

```

1  VAR_GLOBAL
2
3  //inputs
4      start           :BOOL;//boton activación
5      switchkey      :BOOL;//switch elige cual estación va a funcionar
6      stop           :BOOL;//boton stop
7      PE             :BOOL;//parada de emergencia
8      auto           :BOOL;//activar modo automatico
9
10     sensor_KR6      :BOOL;//sensor detecta carrier acercándose
11     is_up_KR6       :BOOL;//lift en posicion up
12     is_middle_KR6   :BOOL;//lift en posicion middle
13     carrier_pB_KR6  :BOOL;//carrier presente en belt
14     carrier_pL_KR6  :BOOL;//carrier en el lift(elevado)
15     kuka_ok_KR6     :BOOL;//KR6 terminó rutina
16
17     sensor_KR15     :BOOL;//sensor detecta carrier acercándose
18     is_up_KR15      :BOOL;//lift en posicion up
19     is_middle_KR15  :BOOL;//lift en posicion middle
20     carrier_pB_KR15 :BOOL;//carrier presente en belt
21     carrier_pL_KR15 :BOOL;//carrier en el lift(elevado)
22     kuka_ok_KR15    :BOOL;//KR15 terminó rutina
23
24 //outputs
25     belt_on         :BOOL;//activar todas las secciones conveyor
26
27     led_KR6         :BOOL;//led sensor
28     up_KR6          :BOOL;//eleva lift
29     down_KR6        :BOOL;//baja lift
30     Belt_ON_lift_KR6 :BOOL;//señal belt on del lift
31     belt_rev_lift_KR6 :BOOL;//belt reverse del lift
32     estacion_ok_KR6  :BOOL;//carrier en estación de trabajo
33     PE_KR6          :BOOL;//interrupcion rutina
34
35     led_KR15        :BOOL;//led sensor
36     up_KR15         :BOOL;//eleva lift
37     down_KR15       :BOOL;//baja lift
38     Belt_ON_lift_KR15 :BOOL;//señal belt on del lift
39     belt_rev_lift_KR15 :BOOL;//belt reverse del lift
40     estacion_ok_KR15 :BOOL;//carrier en estación de trabajo
41     PE_KR15        :BOOL;//interrupcion rutina
42
43 //PARADA EMERGENCIA
44     WasBeltOn       : BOOL := FALSE;
45
46 END_VAR

```

Figura A.8: Variables globales utilizadas en CODESYS.

A.8. Diagramas GRAFCET CODESYS

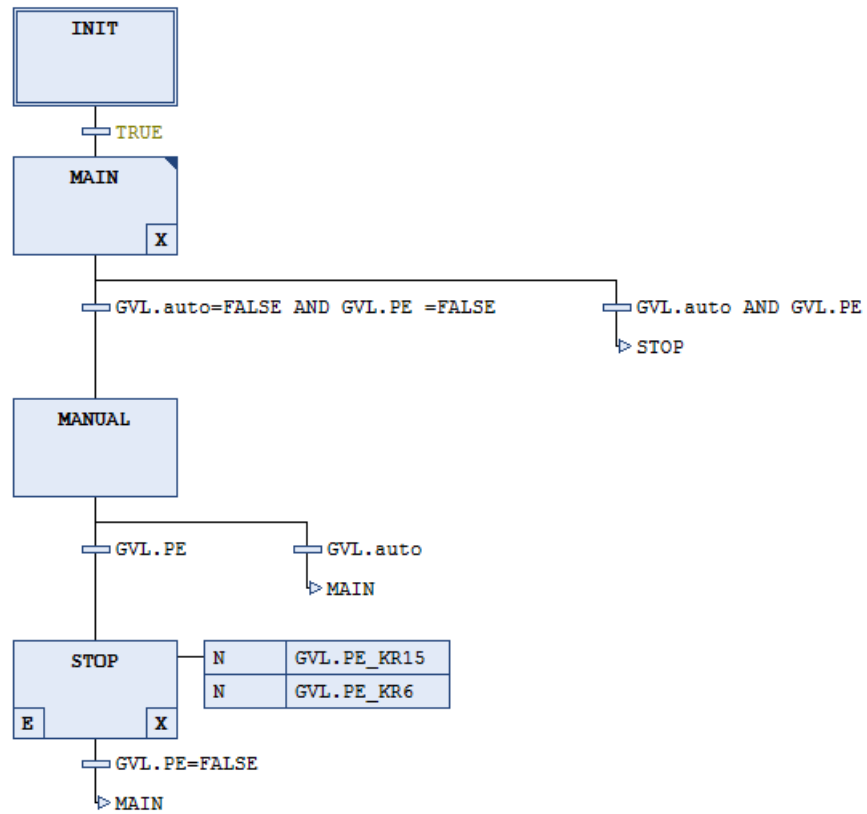


Figura A.9: Diagrama GRAFCET general.

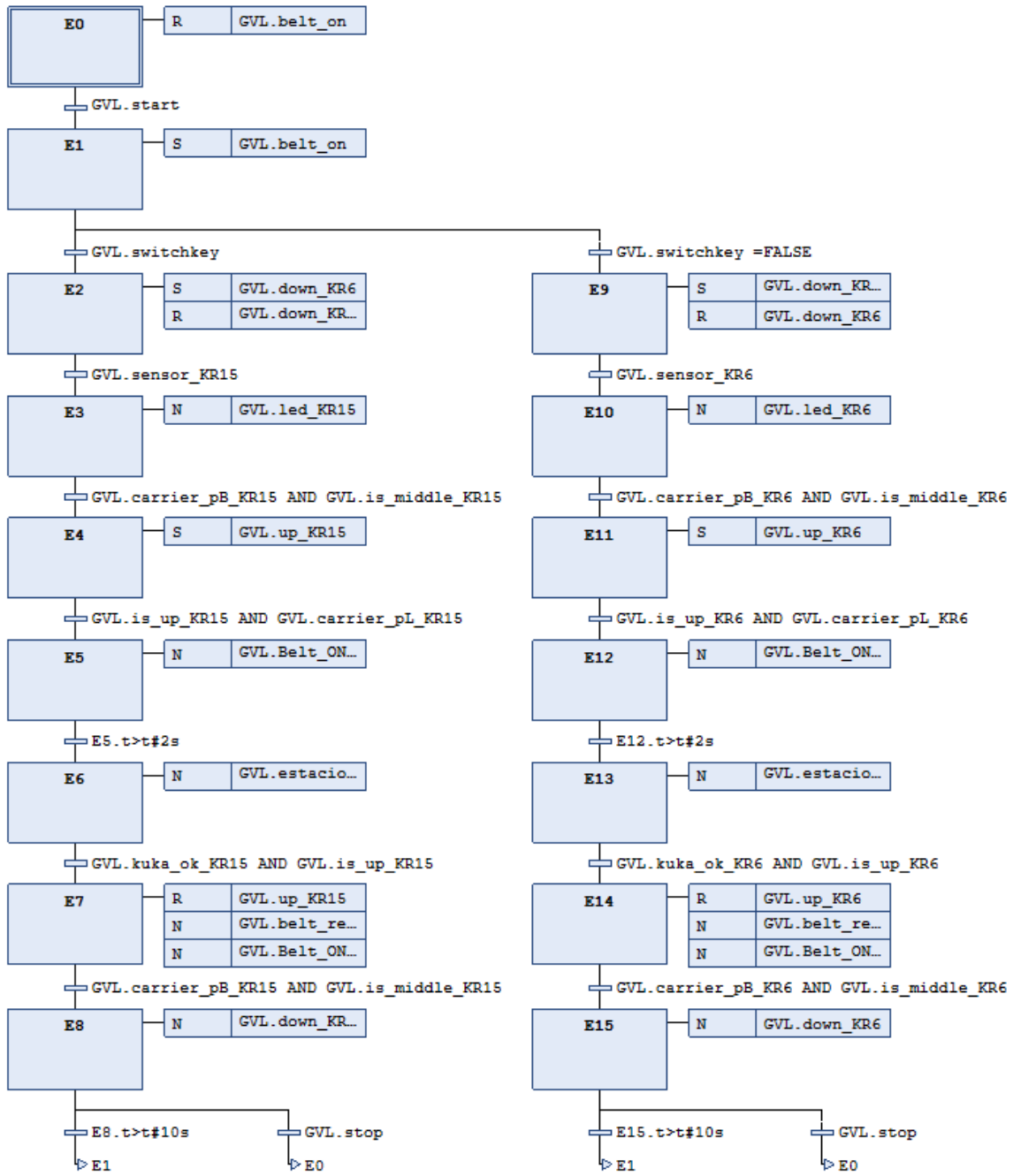


Figura A.10: Diagrama GRAFCET MAIN.

A.9. Especificaciones módulos PLC Glofa

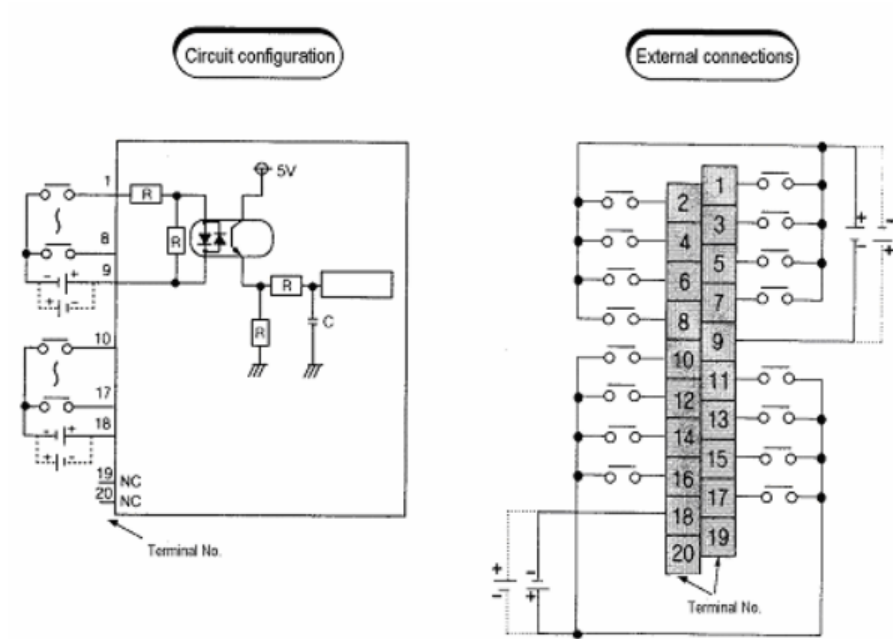


Figura A.11: Configuraciones internas módulo D22A.

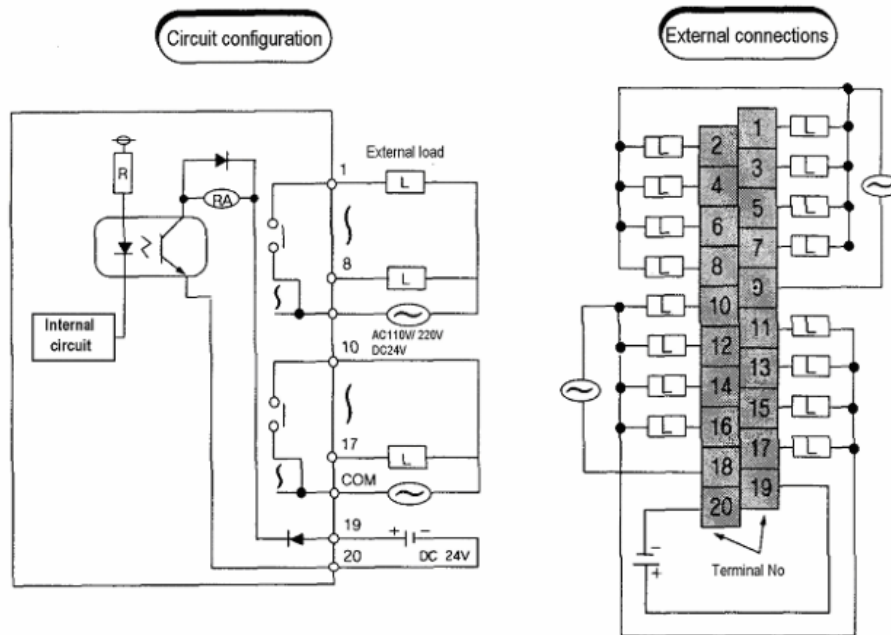


Figura A.12: Configuraciones internas módulo RY2A.