

# Diseño e implementación de una interfaz hombre-máquina para un banco de ensayos de cuplas programables

---

**Postulante:** Hirschmann, Juan Ignacio (jhirschmann@fi.uba.ar)

**Tutor:** Ing. Graña, Jorge Eduardo (jgraña@fi.uba.ar)  
**Cotutor:** Dr. Ing. Jurado, Alejandro Diego (ajurado@fi.uba.ar)

*Tesis presentada en cumplimiento de los requisitos del plan de estudios de la carrera de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad de Buenos Aires.*

30 de julio de 2025

---

## *Agradecimientos*

*Agradezco al departamento de energía de la Facultad de Ingeniería por proveer los equipos y el espacio para desarrollar este proyecto. Agradezco también a mis tutores de tesis, Alejandro y*

*Jorge, por haberme guiado y aconsejado en todo momento. Por último, extiendo mi agradecimiento a mis familiares y amigos, por haberme acompañado durante esta y todas las demás etapas de mi carrera.*

# Índice

<b>1. Introducción</b>	<b>8</b>
1.1. Introducción al objeto de estudio . . . . .	8
1.2. Antecedentes . . . . .	8
1.3. Objetivo general de la tesis . . . . .	9
1.4. Metodología . . . . .	9
<b>2. Componentes del banco de ensayos</b>	<b>10</b>
2.1. SIMOTION D425 . . . . .	11
2.1.1. Interfaces en SIMOTION D425 . . . . .	12
2.2. Módulo de línea SINAMICS S120 . . . . .	13
2.3. Drivers de Servomotores SINAMICS S120 . . . . .	13
2.4. Servomotores . . . . .	14
2.5. Herramientas de software . . . . .	14
2.5.1. Software de programación del banco de ensayos . . . . .	14
2.5.2. Software utilizado para el desarrollo de la interfaz gráfica . . . . .	15
<b>3. Protocolos de comunicación industrial</b>	<b>15</b>
3.1. Protocolo PROFINET . . . . .	16
3.1.1. Telegramas PROFIDrive . . . . .	16
3.2. Protocolo OPC XML-DA . . . . .	20
<b>4. Conceptos de programación en PLC</b>	<b>21</b>
4.1. Tipos de dato . . . . .	21
4.1.1. Tipos de datos elementales . . . . .	21
4.1.2. Tipos de datos definidos por usuario (UDT, por sus siglas en inglés) . . .	22
4.1.3. Tipos de dato de sistema . . . . .	23
4.1.4. Objetos de tecnología (TO, por sus siglas en inglés) . . . . .	23
4.2. Tipos de variable . . . . .	23
4.3. Programación en LADDER . . . . .	24
4.4. Programación en Structured Text (ST, por sus siglas en inglés) . . . . .	25
4.4.1. Unidades organizacionales de programa (POU, por sus siglas en inglés) .	26
4.5. Sistema de ejecución . . . . .	26
4.5.1. Estados de operación . . . . .	26
4.5.2. Niveles de ejecución . . . . .	27
<b>5. Aplicaciones del banco de ensayos</b>	<b>28</b>
5.1. Ensayos mecánicos en general . . . . .	28
5.2. Ensayos para motores para vehículos . . . . .	28
5.3. Simulación de ciclos de trabajo en máquinas-herramienta . . . . .	29
5.4. Simulación de carga para generadores mareomotrices . . . . .	30
5.5. Simulación de carga para generadores eólicos . . . . .	31
<b>6. Requisitos de diseño</b>	<b>31</b>
6.1. Interfaz con el usuario . . . . .	31
6.1.1. Seguridad del banco . . . . .	32
6.2. Cupla de simulación . . . . .	33
6.3. Requisitos de diseño - tabla de síntesis . . . . .	34

<b>7. Arquitectura de la solución</b>	<b>35</b>
7.1. Diagrama en bloques de la solución . . . . .	35
7.2. Interfaz por software . . . . .	37
<b>8. Ingeniería de la solución</b>	<b>37</b>
8.1. Programación en la unidad de control . . . . .	38
8.1.1. Algoritmo de control de flujo del programa . . . . .	38
8.1.2. Algoritmo de control de torque por saturación del lazo de velocidad . . . . .	41
8.1.3. Algoritmo de control de torque en función de la velocidad: filtrado de variables . . . . .	43
8.1.4. Algoritmo de control de torque en función del tiempo: latencia de datos . . . . .	46
8.1.5. Mecanismo de keepalive . . . . .	53
8.2. Interfaz física del banco de ensayo . . . . .	53
8.2.1. Componentes de la interfaz física . . . . .	53
8.2.2. Circuito eléctrico . . . . .	54
8.3. Interfaz por software del banco de ensayos . . . . .	55
8.3.1. Arquitectura Modelo-Vista-Controlador . . . . .	55
8.3.2. Fuentes de variables . . . . .	57
8.3.3. Ensayos de cupla en función del tiempo - envío de datos al control . . . . .	60
8.3.4. Gráficos en tiempo real - latencia de datos . . . . .	60
<b>9. Pruebas de validación del sistema diseñado</b>	<b>61</b>
9.1. Aplicación de cupla en el eje del servomotor . . . . .	61
9.2. Ecuación de cupla - velocidad . . . . .	62
9.2.1. Término constante . . . . .	63
9.2.2. Término proporcional . . . . .	65
9.2.3. Término cuadrático . . . . .	66
9.2.4. Término inercial . . . . .	67
9.3. Curva de cupla - tiempo . . . . .	68
9.3.1. Precisión temporal . . . . .	68
9.4. Duración de ensayos . . . . .	69
9.5. Seguridad del banco . . . . .	69
9.5.1. Paradas de emergencia accionadas por el usuario . . . . .	69
9.5.2. Protección contra desconexión de interfaz . . . . .	70
9.5.3. Control de torque máximo y derivada de torque . . . . .	70
9.6. Interfaz gráfica . . . . .	70
9.6.1. Ventana principal . . . . .	70
9.6.2. Ventana de avisos y alertas . . . . .	72
9.6.3. Ventana de gráficos en tiempo real para la ejecución ensayos autocontenidos	73
<b>10. Realización de un ensayo de referencia</b>	<b>73</b>
10.1. Modelo de planta motriz . . . . .	75
10.2. Perfiles de velocidad-tiempo a ensayar . . . . .	75
10.3. Escalamiento de variables . . . . .	76
10.4. Resultados . . . . .	77
10.4.1. SAE J227A . . . . .	77
10.4.2. US06 . . . . .	79

<b>11.Discusión</b>	<b>81</b>
11.1. Cumplimiento de objetivos . . . . .	81
11.2. Cumplimiento de requisitos de diseño . . . . .	81
11.3. Caracterización del sistema . . . . .	83
<b>12.Conclusión</b>	<b>83</b>
<b>13.Desarrollos a futuro</b>	<b>84</b>
13.1. Ensayos de torque en función de la velocidad con coeficientes no constantes . . . . .	84
13.1.1. Coeficientes de la ecuación torque variables con la posición . . . . .	84
13.1.2. Coeficientes de la ecuación torque variables con una variable externa . . . . .	85
13.2. Mejoras en el término inercial de la ecuación cupla-velocidad . . . . .	85
13.3. Adición de un encoder externo . . . . .	85
13.4. Aumento de la capacidad de simulación de cupla . . . . .	86
13.5. Aumento de la capacidad de simulación velocidad . . . . .	86
13.6. Interfaz web . . . . .	86
<b>14.Bibliografía</b>	<b>87</b>
<b>15.Anexo</b>	<b>88</b>
15.1. Manual de operación . . . . .	88
15.2. Programación del PLC . . . . .	126
15.3. Plano eléctrico . . . . .	191
15.4. Circuito esquemático - Placa de entradas y salidas . . . . .	195

## Índice de figuras

1. Componentes del sistema de accionamientos . . . . .	11
2. Interfaces disponibles del módulo SIMOTION D425 . . . . .	12
3. SIMOTION SCOUT V4.4 en funcionamiento . . . . .	15
4. Red LADDER . . . . .	25
5. Porción de un programa escrito en ST . . . . .	26
6. Ejemplo de la carga en el husillo de una máquina herramienta durante el mecanizado de piezas . . . . .	30
7. Tipos de generadores mareomotrices rotativos y oscilantes . . . . .	31
8. Diagrama en bloques de la solución diseñada: partes y conexiones . . . . .	35
9. Diagrama en bloques de la solución diseñada: componentes internos y señales . . . . .	36
10. Diagrama de transiciones de estado implementado en el algoritmo de control de flujo . . . . .	40
11. Diagrama de flujo para el control de torque por saturación . . . . .	42
12. Diagrama de flujo para el control de inversión de torque . . . . .	43
13. Respuesta al escalón del filtro de 91 coeficientes: en naranja la respuesta del filtro, en verde la respuesta del filtro tras aplicar el operador derivada y en azul la señal de excitación . . . . .	45
14. Respuesta al escalón filtro IIR: en naranja la respuesta del filtro, en verde la respuesta del filtro tras aplicar el operador derivada y en azul la señal de excitación . . . . .	46
15. Diagrama de buffers . . . . .	47
16. Diagrama en bloques del algoritmo con temporizadores autoreseteables . . . . .	50
17. Diagrama de flujo de la implementación del control de sincronismo . . . . .	52
19. Diagramas de diseño de la placa en la interfaz física. . . . .	55

20. Esquema del patrón de diseño MVC . . . . .	56
21. Curva de aceleración ensayada y respuesta en velocidad angular observada. . . . .	64
22. Respuesta al escalón del eje simulador únicamente con término constante de la ecuación cupla-velocidad . . . . .	65
23. Respuesta al escalón únicamente con término proporcional de la ecuación cupla-velocidad . . . . .	66
24. Respuesta al escalón únicamente con término cuadrático de la ecuación cupla-velocidad . . . . .	67
25. Respuesta al escalón únicamente con el término inercial de la ecuación cupla-velocidad . . . . .	68
26. Comparación entre la curva de cupla-tiempo efectivamente aplicada y la curva objetivo . . . . .	69
27. Ventana principal del programa . . . . .	70
28. Ejemplo de alerta al estar imposibilitada la conexión . . . . .	72
29. Acople mecánico de motores: a la izquierda el motor simulador y a la izquierda el motor de planta motriz . . . . .	74
31. Curvas de velocidad, cupla y potencia activa medidas durante el ensayo utilizando la curva SAE J227a . . . . .	78
32. Curvas de velocidad, cupla y potencia activa medidas durante el ensayo utilizando la curva US06 . . . . .	80

## Glosario

**buffer** Es una porción de memoria reservada para almacenar datos mientras estos se trasladan entre dos procesos que operan a velocidad distintas. . 47, 48, 57, 58

**DRIVE-CLiQ** Es un sistema de comunicación propietario de Siemens basado en Ethernet ideado para la comunicación de la unidad de control con periféricos.. 11, 12

**FIR** Del inglés: Finite impulse response. Es un tipo de filtro discreto cuya respuesta al impulso tiene un número de coeficiente finito. . 44, 45

**IIR** Del inglés: Infinte impulse response. Es un tipo de filtro discreto cuya respuesta al impulso tiene un número de coeficiente infinito. . 5, 45, 46, 68

**JAVA** Es un lenguaje de programación orientado a objetos, fuertemente tipado y de propósito general. . 15, 37, 56, 70

**keepalive** Del inglés: mantener vivo. Es un mecanismo para verificar que una conexión sigue abierta, aún en ausencia de tráfico. . 4, 41, 53, 70, 72, 73

**LADDER** Es un lenguaje gráfico de programación de lógica digital. Es usualmente empleado en programación industrial.. 3, 5, 24, 25, 44, 50

**MVC** Modelo-Vista-Controlador. Es un patrón de arquitectura de software usualmente empleado en aplicaciones que requieran interfaces gráficas. . 6, 37, 55, 56

**OPC XML-DA** Es un estándar de comunicación basado en XML y servicios Web. Implementa una arquitectura cliente-servidor. . 3, 13, 15, 16, 20, 21, 24, 36, 37, 47, 53, 56, 57, 60, 71, 81

**PLC** Del inglés: *programmable logic controller*. Es un controlador de lógica programable, usualmente utilizado para el procesamiento de entradas y salidas en entornos industriales.. 3, 5, 11, 17, 18, 20–27, 36, 38, 47, 48, 57–61, 85, 126

**POU** Del inglés: program organizatinal unit. Unidad de organizacional de programa. Es cualquier archivo de código susceptible de ser ejecutado. . 3, 26

**PROFIDrive** Es un perfil de comunicación entre accionamientos que corre sobre redes PROFINET o PROFIBUS.. 3, 16, 23

**PROFINET** Es un protocolo de comunicación basado en Ethernet.. 3, 15, 16

**SIMOTION D425** Es la unidad de control de la familia SIMOTION D, orientada al movimiento de precisión en múltiples ejes. . 3, 5, 9, 11, 12, 15, 21, 24, 36, 37, 51, 53, 60

**SINAMICS S120** Es un sistema modular de accionamientos de motores eléctricos. Desarrollado por Siemens para aplicaciones de alta precisión.. 3, 11–14

**ST** Del inglés: Structured Text. Es un lenguaje de programación. Es usualmente empleado en programación industrial.. 3, 5, 25, 26

## 1. Introducción

### 1.1. Introducción al objeto de estudio

Desde la segunda revolución industrial, las máquinas eléctricas fueron adquiriendo cada vez más relevancia en los procesos industriales y en la vida de las personas. En particular, se estima que los motores eléctricos consumen alrededor de un 50 % de la energía eléctrica producida a nivel mundial. Con el advenimiento de los vehículos eléctricos, se espera que esta cifra aumente.

Por este motivo, resulta de interés para la industria la caracterización de motores eléctricos, especialmente su desempeño frente a cargas no constantes o intermitentes. Frente a estas variaciones en la carga, el motor eléctrico responde de manera dinámica; lo que posiblemente afecta su consumo de energía y correcto funcionamiento.

La respuesta a estos transitorios es difícil de predecir analíticamente, por lo que se suelen determinar mediante simulaciones numéricas posteriormente corroboradas experimentalmente. La verificación empírica de esta simulación demanda ensayar el sistema mecánico real en condiciones de campo o bien simular la carga dinámica del motor en un banco de ensayos.

La primera de estas alternativas suele ser prohibitiva ya que ensayar el sistema real implica diseñarlo y construirlo, lo que deja poco margen para modificaciones de diseño en caso de ser necesario.

En el caso de la carga simulada, se debe lograr un banco de pruebas lo suficientemente versátil como para satisfacer una amplia gama de condiciones de campo, de forma tal de poder emular la carga de trabajo del motor ensayado.

Con estos fines, el laboratorio de máquinas eléctricas de la Facultad de Ingeniería de la Universidad de Buenos Aires cuenta con un banco de ensayos con máquinas sincrónicas de imanes permanentes controladas por sistemas de conversión de frecuencia (VFD, por sus siglas en inglés) que permiten programar sus características externas, a efectos de realizar estudios de comportamiento dinámico de máquinas eléctricas.

Los accionamientos adquiridos son equipos comerciales diseñados para el control de posición de motores para aplicaciones industriales. Si bien son adecuados para su destino como banco de ensayos de cuplas motoras, las herramientas de desarrollo que brinda el fabricante están fuertemente orientadas a su objetivo comercial. Esto último obliga a que el operador del banco de ensayos disponga de un gran dominio y conocimiento en el entorno de programación del fabricante para poder realizar ensayos.

Por este motivo, el banco de ensayos no cuenta con una interfaz hombre-máquina adecuada ni medidas de seguridad especiales para este propósito. Esto es indeseable para un instrumento de laboratorio y limita el uso pleno del banco de ensayos.

Frente a esta situación, se propone la creación de una interfaz hombre-máquina que evite la caída en desuso del equipo y permita la utilización del banco de ensayos de una manera segura, completa y precisa. Un desarrollo de estas características implica el diseño de software y hardware a medida.

### 1.2. Antecedentes

El banco de ensayos fue objeto de una tesis de grado en ingeniería eléctrica que lleva el nombre "Simulación de cuplas motoras y resistentes con máquinas sincrónicas de imanes permanentes controladas". Este trabajo demostró la funcionalidad del sistema de accionamientos Siemens como banco de ensayos de motores.

El autor instaló y llevó a cabo la puesta en marcha del sistema de accionamientos, documentando extensamente el trabajo. También implementó un algoritmo de aplicación de torque

senoidal y variante en función de la velocidad, demostrando la viabilidad de la utilización del sistema de accionamientos SIMOTION D425 para este fin.

Sin embargo, el método desarrollado por el autor requería iniciar el entorno de programación previo a la utilización del banco y luego iniciar una planilla en Microsoft Excel como interfaz gráfica. Necesariamente, esto requería cierto nivel de conocimiento en el entorno de programación SIMOTION D425 y es fuertemente dependiente de la versión del *software* y del sistema operativo.

Además, el mecanismo descubierto para introducir consignas de torque consistía en modificar directamente un parámetro relacionado al torque que se adicionaba a la consigna de cupla en el sistema de accionamientos. La introducción de consignas de torque sin límite de velocidad podría causar que el eje de simulación acelere indefinidamente.

Por último, las posibilidades de inserción de consignas de cupla fueron reducidas a un conjunto de funciones específicas, lo que limitó el alcance del desarrollo. Por este motivo, la característica de cupla-tiempo o cupla-velocidad del instrumento permite la aplicación del instrumento solamente en un subconjunto de casos posibles.

El autor señaló estas limitaciones como puntos de mejora para trabajos a futuro. Estas observaciones se consideraron al momento de definir los objetivos de su tesis y su alcance.

## 1.3. Objetivo general de la tesis

Implementar una interfaz hombre-máquina que, aplicada a un banco de pruebas de motores, permita simular el comportamiento dinámico de motores eléctricos bajo diferentes condiciones de carga.

Objetivo I: Implementar las características de un sistema de supervisión, control y adquisición de datos (SCADA, por sus siglas en inglés). Desarrollar un cliente para la comunicación con los elementos de control del equipo que permita la modificación de parámetros de ensayo en tiempo real.

Objetivo II: Simular cuplas variantes en función del tiempo y velocidad. Se incorporará el término de inercia con la derivada de la velocidad de operación.

Objetivo III: Implementar un sistema de visualización, almacenamiento y post procesamiento de los datos de salida.

Objetivo IV : Documentar el desarrollo mediante manuales de operación, programación y mantenimiento. Generar planos unifilares de potencia y de comando.

Objetivo V: Utilizar el sistema para caracterizar el comportamiento de un motor eléctrico frente a distintos perfiles de velocidad y torque.

## 1.4. Metodología

En primer lugar, se relevaron las necesidades del sistema de simulación requeridas por los futuros operadores. Esto contribuyó a la arquitectura general del banco de ensayos y fue considerado en todas las etapas de la planificación del banco de ensayos.

El diseño del banco luego se refinó al realizar reuniones periódicas en donde se evaluó el avance del proyecto. A raíz de estos encuentros, los operadores pensaron nuevas funcionalidades que expandían la capacidad actual del banco. Esta etapa formó parte del desarrollo inicial del banco y admitió la inclusión de nuevos requisitos, sin modificar la arquitectura diagramada.

Posteriormente, se realizaron pruebas y verificaciones de las herramientas de seguridad con las que cuenta el banco de pruebas. Durante esta instancia, se aseguró que el banco pueda ser operado sin arriesgar el equipo ni al operador, previo a realizar las pruebas de validación.

En la etapa de validación, se produjeron ensayos para verificar el desempeño del banco comparándolo con resultados teóricos y experimentales. Fue posible replicar los resultados obtenidos

por un sistema similar desarrollado por un grupo de investigación independiente.

Por último, todas las decisiones de diseño adoptadas buscan favorecer dos aspectos fundamentales de un instrumento de laboratorio. El primero de ellos es la seguridad de las personas que lo operan y presencian el ensayo.

Estrechamente ligado al primero, el segundo de estos aspectos es el diseño de un sistema determinístico, que pueda entregar información precisa, consistentemente y con comportamiento predecible y claro.

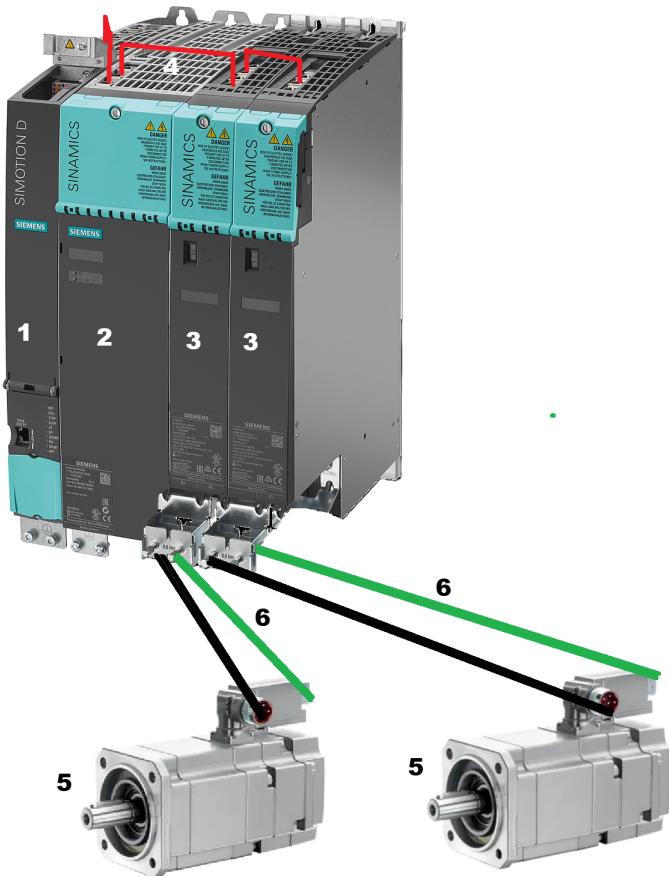
En cualquier instancia en donde se podría producir un conflicto entre el primer objetivo y el segundo, siempre se buscó favorecer la seguridad del banco de ensayos en su diseño.

## 2. Componentes del banco de ensayos

El banco de ensayos consiste en un sistema de accionamientos compuesto por un rectificador totalmente controlado, un accionamiento de servomotores de dos canales, dos servomotores y una unidad de control. De los dos servomotores, solo uno de ellos será el eje que simula las condiciones de carga. Este motor se acopla directamente al eje del motor que se requiera ensayar. Para la demostración del banco de ensayos, es posible configurar el otro de los servomotores como eje de tracción y realizar ensayos autocontenidos.

El motor simulador aceptará consignas de torque que permitan emular las condiciones de carga del dispositivo bajo prueba. De esta forma, será posible estudiar el desempeño del motor ensayado sin tener que realizar pruebas en campo o construir una versión a escala real.

Materialmente, el banco de pruebas se conforma de los elementos que se observan en la siguiente figura:



Nº	Descripción
1	Unidad de control SI-MOTION D425
2	Módulo inteligente de línea SINAMICS S120
3	Módulos de control de servomotor SINAMICS S120
4	Interfaz DRIVE-CLiQ entre módulos
5	Servomotores línea 1FK7105
6	Interfaz DRIVE-CLiQ para servomotor

Figura 1: Componentes del sistema de accionamientos

En las siguientes secciones se realizará una breve descripción de cada elemento del banco de ensayos.

## 2.1. SIMOTION D425

Es la unidad de control del sistema SIMOTION, contiene el sistema de ejecución y el software del accionamiento SINAMICS. La línea de productos SIMOTION D se encuentra fuertemente orientada al control de motores utilizando los accionamientos de la familia SINAMICS S120.

Esta unidad de control suele ser recomendada para aplicaciones de control distribuido en máquinas con un gran número de ejes y que requieran alto nivel de precisión y sincronismo.

SIMOTION D425 es la unidad central de control de movimiento de ejes. Dentro de sus componentes de software se encuentran las siguientes funcionalidades:

- Entorno de ejecución programable por el usuario (SIMOTION SCOUT).
- Diferentes niveles de ejecución de código.
- Funcionalidad controlador lógico programable (PLC, por sus siglas en inglés).
- Programación multilenguaje.
- Funciones de control de movimiento.
- Funciones de comunicación.

- Software SINAMICS S120: control a lazo cerrado de corriente, torque, velocidad para motores y control de módulo de línea.

### 2.1.1. Interfaces en SIMOTION D425

Para el intercambio de información con otros módulos o sensores, la unidad de control cuenta con un gran número de interfaces disponibles. En la figura 2 se detallan los puertos que presenta la unidad:

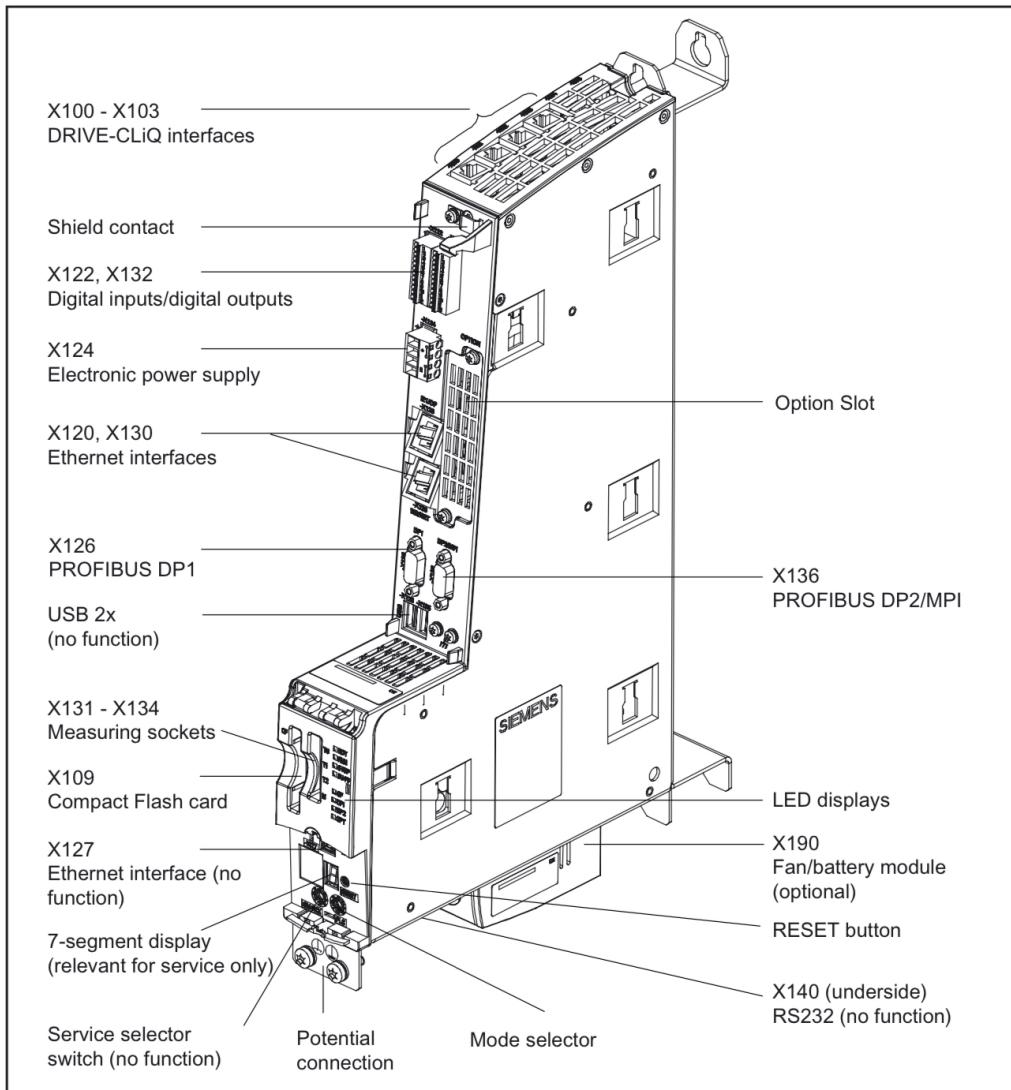


Figura 2: Interfaces disponibles del módulo SIMOTION D425

- X100-103: Interfaces DRIVE-CLiQ. DRIVE-CLiQ es una interfaz desarrollada por Siemens para la comunicación entre módulos y elementos de medición.
- X122-132: Entradas/Salidas digitales. SIMOTION D425 ofrece dos puertos de entradas y salidas digitales. En total, existen 8 salidas digitales pures y otros 8 canales digitales bidireccionales.
- X124: Fuente de alimentación interna (24V).
- X120-130: Interfaces Ethernet. En la unidad de control se encuentran 4 interfaces Ethernet industrial con una velocidad de transmisión máxima de 100Mbit/s. Estas interfaces per-

miten la creación de redes Ethernet para el intercambio de información. A través de estos puertos se realiza la programación de la unidad y el acceso al servidor OPC XML-DA.

- X126-136: Interfaces PROFIBUS DP. La unidad ofrece dos puertos PROFIBUS DP para la comunicación con otros dispositivos compatibles. Estos puertos también permiten la conexión con la unidad de programación.
- X131-134: Zócalos de medición. Son puertos de salidas analógicas, se reservan para tareas de diagnóstico.
- X109: Ranura para CompactFlash. Esta memoria es obligatoria para la operación de la unidad de control. Contiene los archivos de programa de usuario, el software SINAMICS para el control de los accionamientos, el sistema de ejecución SIMOTION y información de las licencias disponibles en el equipo.

## 2.2. Módulo de línea SINAMICS S120

El módulo inteligente de línea (SLM, por sus siglas en inglés) SINAMICS S120 implementa un rectificador totalmente controlado. Se detallan en una tabla las principales características técnicas:

Característica	Valor	Unidades
Potencia máxima <sup>1</sup>	16	kW
Factor de potencia ( $\cos(\phi)$ )	0.98	—
Maxima temperatura ambiente <sup>2</sup>	40	°C
Tension de red	$3x380 \pm 10\%$	V <sub>AC</sub>
Frecuencia de red	47 - 63	Hz
Rango de regulación	510 - 720	V <sub>DC</sub>
Corriente máxima de entrada	24.5	A <sub>AC</sub>
Corriente máxima de salida	27	A <sub>DC</sub>

Cuadro 1: Especificaciones técnicas del módulo de línea 6SL3430-6TE21-6AA3

## 2.3. Drivers de Servomotores SINAMICS S120

Para el control de los servomotores se utiliza el módulo de dos canales de la línea de drives SINAMICS S120. En la tabla a continuación se especificaciones:

Característica	Valor	Unidades
Potencia máxima	2 canales de 9,7	kW
Corriente máxima de entrada	100	A <sub>DC</sub>
Corriente máxima de salida (valor pico)	2 canales de 36	A
Corriente máxima de salida (valor medio)	2 canales de 18	A <sub>DC</sub>
Tensión de entrada	510 - 720	V <sub>DC</sub>
Tensión de salida	0 - 430	V

Cuadro 2: Especificaciones técnicas del accionamiento 6SL3120-2TE21-8AA3

<sup>1</sup>El valor también vale en sentido de regeneración

<sup>2</sup>A partir de esta temperatura es necesario aplicar un factor de degradación en la corriente y potencia máxima

## 2.4. Servomotores

Los servomotores utilizados son de la línea SINAMICS S120, en particular el modelo compacto de la serie 1FK7. A continuación se presenta una tabla que sintetiza las principales características del dispositivo:

Característica	Valor	Unidades
Potencia máxima ( $\Delta T = 100K$ )	8.17	kW
Torque estático	48	Nm
Torque dinámico	26	Nm
Inercia de rotor	0.0156	Kgm <sup>2</sup>
Velocidad máxima	3000	RPM
Corriente máxima (valor medio)	18	A
Corriente máxima (valor pico)	109	A

Cuadro 3: Especificaciones técnicas del servomotor 1FK7105-5AF71-1UGO

El fabricante ofrece el servomotor junto con su sistema de resolver de 4 polos. La resolución de este sistema es de 0,067 grados o 5400 divisiones por rotación. Todos los miembros de la familia de motores SINAMICS 1FK7, son servomotores sincrónicos ideados para ser utilizados con el sistema de drives SINAMICS S120 en aplicaciones de alto torque y precisión.

## 2.5. Herramientas de software

Esta sección se dedica a la descripción de las herramientas de software utilizadas para la programación del banco de ensayos y su interfaz gráfica

### 2.5.1. Software de programación del banco de ensayos

El banco de ensayos debe ser programado utilizando el software de desarrollo del fabricante. En este caso, el fabricante ofrece el software SIMOTION SCOUT V4.4.

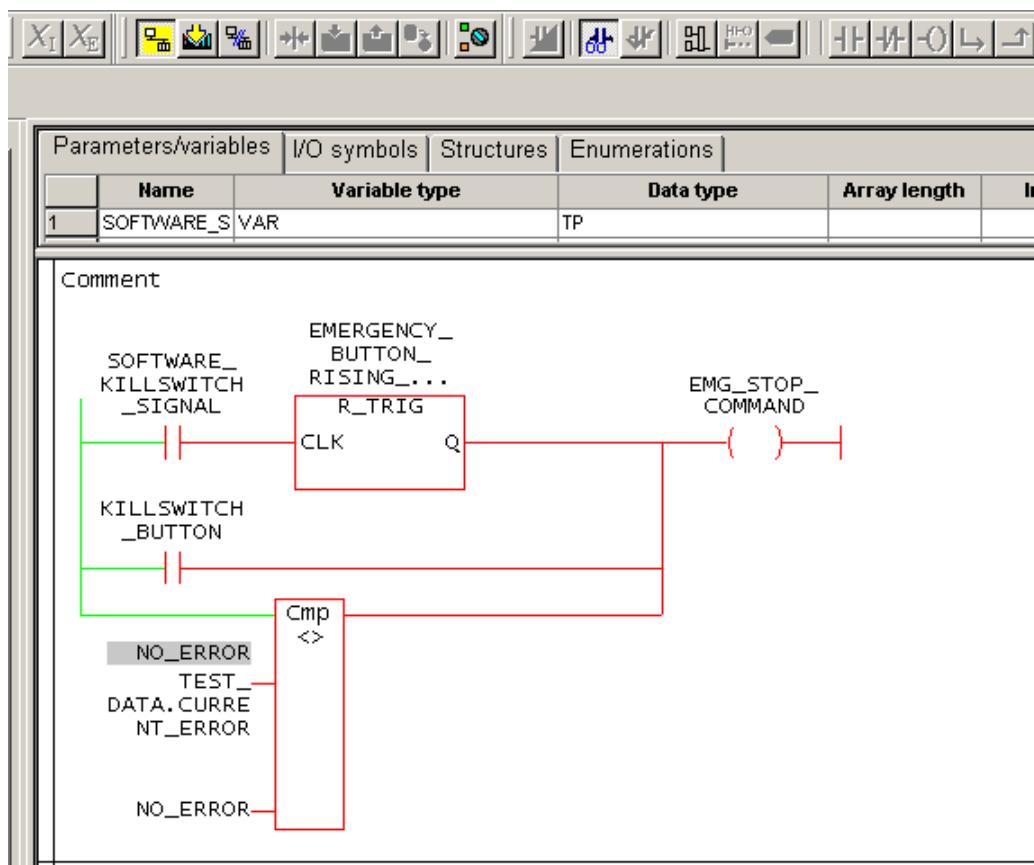


Figura 3: SIMOTION SCOUT V4.4 en funcionamiento

A través de esta herramienta es posible programar consignas de movimiento, configuraciones de comunicación y realizar la parametrización del sistema de accionamientos.

Si bien sería posible operar el banco de ensayos desde SIMOTION SCOUT, la complejidad de la interfaz gráfica haría que la operación sea propensa a errores y modificaciones no documentadas de código. Por este motivo, se suele desacoplar la interfaz de programación de los paneles de operador utilizados para el control de la máquina, agrupados bajo el término interfaz hombre-máquina (HMI, por sus siglas en inglés).

#### 2.5.2. Software utilizado para el desarrollo de la interfaz gráfica

Para implementar la interfaz hombre-máquina se utilizó el software Visual Studio Code como entorno de desarrollo integrado (IDE, por sus siglas en inglés).

Se empleó JAVA como lenguaje de programación para la interfaz gráfica y comunicación entre la computadora y el banco fue JAVA. El fabricante ofrece una librería para la comunicación OPC XML-DA en este mismo lenguaje de programación.

### 3. Protocolos de comunicación industrial

El sistema SIMOTION D425 ofrece múltiples opciones para la comunicación entre módulos. En particular, para la implementación del banco de ensayos se utilizaron 2 tipos de protocolos de comunicación: PROFINET y OPC XML-DA.

En los apartados a continuación se realizan breves descripciones de ambos protocolos.

### 3.1. Protocolo PROFINET

PROFINET es un protocolo industrial fuertemente inspirado en el estándar Ethernet Industrial introducido en las normas IEEE 802.3 y IEEE 802.3u. Al igual que Ethernet, PROFINET IO soporta medios de transmisión cableados, ópticos y aéreos. Es un estándar de comunicación que se monta sobre la capa de aplicación del modelo de ISO/OSI de TCP/IP. Esto admite la coexistencia en la misma red de otros protocolos Ethernet como OPC XML-DA.

Este estándar define protocolos de comunicación sincrónica y asincrónica entre componentes para intercambiar información sobre alarmas, datos de diagnóstico, estado de elementos de seguridad crítica y otros datos del sistema. Adicionalmente, PROFINET admite la implementación de *perfíles de comunicación*. Estos perfiles permiten implementar interfaces estandarizadas para la interacción entre diferentes equipos.

Un ejemplo de este tipo de perfiles de comunicación es PROFIDrive. PROFIDrive solamente define el mecanismo de acceso a un grupo predefinido de parámetros del accionamiento mediante telegramas estandarizados. De esta forma, es posible desarrollar software interoperable entre sistemas y admitiendo la posibilidad de implementar telegramas no estándar si la aplicación lo requiere.

#### 3.1.1. Telegramas PROFIDrive

Los telegramas PROFIDrive se utilizan para el intercambio cíclico de información. Para la implementación del banco de ensayos se utilizaron distintos tipos de telegrama.

La unidad fundamental del telegrama es el *dato de proceso* (PZD, por sus siglas en alemán) que es una palabra de 16 bits. Dependiendo del tipo de telegrama elegido, se admite una cantidad máxima de PZD, manteniendo algunas posiciones reservadas y otras disponibles para uso de usuario. En dirección de transmisión y recepción los telegramas enviados son diferentes.

Siemens implementa los telegramas PROFIDrive en su línea SINAMICS, ofreciendo la posibilidad de utilizar telegramas estándar, definidos en la norma IEC 61800-7, o bien telegramas propios.

Esto último permite desarrollar aplicaciones especiales no contempladas en el estándar IEC, aumentando la versatilidad de la capacidad instalada. Algunos de los telegramas propios del fabricante vienen aparejados con funciones pre-programadas que facilitan el desarrollo de software de máquina.

Considerado lo expuesto, y como el banco de pruebas únicamente está compuesto por productos Siemens, se optó por utilizar telegramas propietarios. De esta forma, el tiempo de desarrollo será menor. En la siguiente sección se procederá a describirlos.

##### 3.1.1.1 Telegrama 102

Para la comunicación entre el drive y la unidad de control, se eligió el telegrama 102. La estructura del telegrama es la siguiente:

	PLC → Drive	Drive → PLC
PZD1	STW1	ZSW1
PZD2	NSOLL_B	NISt_B
PZD3		
PZD4	STW2	ZSW2
PZD5	MOMRED	MELDW
PZD6	G1_STW	G1_ZSW
PZD7	-	G1_XIST1
PZD8	-	
PZD9	-	G1_XIST2
PZD10	-	
PZD11	-	-
PZD12	-	-
PZD13	-	-
PZD14	-	-
PZD15	-	-
PZD16	-	-
PZD17	-	-
PZD18	-	-

Cuadro 4: Estructura del telegrama 102

■ Señales en sentido de driver a PLC:

- ZSW1 palabra de estado del driver. Siemens define las siguientes bits para la comunicación con unidades SINAMICS.
  - ZSW1.0=1: Listo para encendido
  - ZSW1.1=1: Listo para operación
  - ZSW1.2=1: Operación habilitada
  - ZSW1.3=1: Condición de falla detectada
  - ZSW1.4=1: No coast down active (OFF2 inactivo). El eje no se detiene simplemente dejando de controlar su velocidad, si no que se lleva su consigna de velocidad a cero.
  - ZSW1.5=1: Detención rápida inactiva (OFF3 inactivo). Utilizado para paradas de emergencia.
  - ZSW1.6=1: Inhabilitación de encendido activa.
  - ZSW1.7=1: Condición de alarma detectada.
  - ZSW1.8=1: Consigna de velocidad dentro de la tolerancia especificada.
  - ZSW1.9=1: Solicitud de control de eje activa
  - ZSW1.10=1: Indica si se alcanzó el valor de referencia de tensión o frecuencia.
  - ZSW1.11=1: El sistema alcanzó su límite de corriente, potencia o cupla.
  - ZSW1.12=1: Apertura de freno de motor.
  - ZSW1.13=1: No se detectaron alarmas de motor o exceso de temperatura.
  - ZSW1.14=1: Giro de motor en sentido de las agujas de reloj.
  - ZSW1.15=1: No hay alarma en la unidad de potencia.
- Gn\_ZSW Señal de estado encoder n.
- Gn\_XISt1/2 Señales de posición del encoder n. Para un eje rotativo, es la cantidad de pulsos medidos desde el ángulo de referencia.

- MELDW palabra de mensaje. Utilizada para diagnóstico.
- NIST\_B: Consigna de velocidad (32 bit)

- Señales en sentido de PLC a driver:

- STW1: palabra de control 1 del driver. Siemens define las siguientes bits para la comunicación con unidades SINAMICS.
  - STW1.0: Detección con flanco ascendente. Deshabilitación de OFF1. OFF1 desactiva el motor frenándolo con una rampa de velocidad, no desactiva el bus de corriente continua del rectificador.
  - STW1.1=1: deshabilitación de OFF2. OFF2 desactiva el motor frenándolo inmediatamente, no desactiva el bus de corriente continua del rectificador.
  - STW1.2=1: deshabilitación de OFF3. OFF3 desactiva el motor frenándolo inmediatamente, desactiva el bus de corriente continua del rectificador.
  - STW1.3=1: Habilitación de operación.
  - STW1.4=1: Accionamiento en estado operativo.
  - STW1.5=1: Continuar generación de rampa hacia la consigna de velocidad.
  - STW1.6=1: Habilitar consignas de velocidad.
  - STW1.7: Detección con flanco ascendente. Reconocimiento de fallas.
  - STW1.10=1: Control del accionamiento mediante PLC.
  - STW1.11=1: Inversión de consigna de velocidad.
  - STW1.12=1: Desactivar el freno de motor incondicionalmente.
  - STW1.13=1: Suba de consigna de velocidad mediante potencíometro.
  - STW1.14=1: Baja de consigna de velocidad mediante potencíometro.
- STW2: palabra de control 2 del driver. Siemens define las siguientes bits para la comunicación con unidades SINAMICS.
  - STW2.0: Selección de conjunto de datos de drive, bit 0. El conjunto de datos de drive depende del tipo de control que se realice en el servomotor.
  - STW2.1: Selección de conjunto de datos de drive, bit 1.
  - STW2.2: Selección de conjunto de datos de drive, bit 2
  - STW2.3: Selección de conjunto de datos de drive, bit 3
  - STW2.4: Selección de conjunto de datos de drive, bit 4
  - STW2.7=1: Eje en modo *parking*. No se recolecta información del encoder.
  - STW2.11: Detección por flanco ascendente. Señal de cambio de gama de motor.
  - STW2.12 a 15: señal de vida para comunicación maestro-esclavo (maestro)
- NSOLL\_B: Consigna de velocidad B (32 bits).
- Gn\_STW: Palabra de control de encoder n.
- MOMRED: Reducción de torque (16 bits).

### 3.1.1.2 Telegrama 370

El telegrama 370 fue utilizado para la comunicación entre el módulo inteligente de línea (SLM, por sus siglas en inglés) y el controlador. A continuación se detalla su estructura interna:

	PLC → SLM	SLM → PLC
PZD1	E_STW1	E_ZSW1
PD2	-	-
		⋮
PD8	-	-

Cuadro 5: Estructura del telegrama 370

■ Fuentes de señal en sentido de envío

- E\_ZSWn palabra de estado para modulo de linea n. Siemens define los siguientes bits:
  - E\_ZSW.0=1: Listo para el encendido.
  - E\_ZSW.1=1: Listo para la operación.
  - E\_ZSW.2=1: Operación habilitada.
  - E\_ZSW.3=1: Condición de falla detectada.
  - E\_ZSW.4=1: OFF2 no está siendo utilizado.
  - E\_ZSW.6=1: Encendido deshabilitado.
  - E\_ZSW.7=1: Alarma detectada.
  - E\_ZSW.9=1: La unidad de control solicita el control del módulo activo de línea.
  - E\_ZSW.11=1: La precarga del bus de continua ha finalizado.
  - E\_ZSW.12=1: El contactor de línea fue accionado.

■ Receptores de señales en sentido de recepción

- E\_STW1 palabra de control para modulo de activo de línea.
  - E\_STW1.0: Detección por flanco ascendente. Activación del contactor del circuito de precarga. Se habilita la regulación pulsada de la línea potencia.
  - E\_STW1.1=1: Se deshabilita OFF2.
  - E\_STW1.3=1: Se habilita la operación.
  - E\_STW1.5=1 Se habilita la operación del módulo de línea como fuente de motor.
  - E\_STW1.6=1 Se habilita la operación del módulo de línea como fuente regenerativa.
  - E\_STW1.7: Detección por flanco ascendente. Se reconocen las condiciones de fallas.
  - E\_STW1.10: Control del modulo de línea mediante la unidad de control.

### 3.1.1.3 Telegrama 390

El telegrama 390 se utilizó para la comunicación entre la unidad de control y su puerto de entradas y salidas digitales. El telegrama presenta la siguiente estructura:

	PLC → I/O	I/O → PLC
PZD1	CU_STW1	CU_ZSW1
PZD2	A_DIGITAL	E_DIGITAL
PZD3	-	-
	⋮	⋮
PZD26	-	-

Cuadro 6: Estructura del telegrama 370

■ Fuentes de señal en sentido de envío

- CU\_ZSW palabra de estado para la unidad de control. Siemens define los siguientes bits<sup>3</sup>:
  - CU\_ZSW.3=1: Condición de falla detectada
  - CU\_ZSW.7=1: Alarma detectada.
  - CU\_ZSW.8=1: Sincronización con el tiempo de sistema alcanzada.
  - CU\_ZSW.9=1: Alarma no detectada.
  - CU\_ZSW.10=1: Condición de falla no detectada.
  - CU\_ZSW.11=1: No hay mensajes de seguridad presentes.
  - CU\_ZSW.12 a 15: Bits de señal de vida para comunicación tipo maestro-esclavo (esclavo).
- E\_DIGITAL: Puerto de entradas digitales (16 bits). Los bits de E\_DIGITAL corresponden a las distintas entradas digitales de la unidad de control.

■ Receptores de señales en sentido de recepción

- A\_DIGITAL: Puerto de salidas digitales (16 bits). Los bits de A\_DIGITAL corresponden a las distintas salidas digitales de la unidad de control.
- CU\_STW1 palabra de control para unidad de control.
  - CU\_STW1.0=1: Señal de sincronismo con el tiempo de sistema.
  - CU\_STW1.1=1: Señal de sincronización con el reloj en tiempo real del sistema.
  - CU\_STW1.2=1: Disparo de la señal frenado de emergencia. (ESR)
  - CU\_STW1.7=1: Detección por flanco ascendente. Reconocimiento de fallas.
  - CU\_STW1.10=1: Control vía PLC
  - CU\_STW1.12 a 15=1: Bits de señal de vida para comunicación tipo maestro-esclavo (maestro).

### 3.2. Protocolo OPC XML-DA

OPC XML-DA (Open process control XML Data Access) es un protocolo abierto y multi-sistema de comunicación. La arquitectura de este protocolo es tipo cliente-servidor y utiliza el protocolo SOAP para el envío de archivos XML.

Siemens en el controlador D425 implementa un servidor siguiendo la especificación para OPC XML-DA de la fundación OPC. En esta norma, se define el método de acceso a los datos de un servidor OPC. Esto permite definir una plataforma estandarizada para el intercambio de información interoperable. Se definen los siguientes servicios básicos:

<sup>3</sup>En todos los casos los bits no listados se encuentran reservados por Siemens para futuros desarrollos

- Acceso al estado de variables: GetStatus, GetStatusResponse
- Lectura de variables: Read, ReadResponse
- Escritura de variables: Write, WriteResponse
- Notificación de cambios en variables: Subscribe, SubscribeResponse
- Muestreo cíclico de variables: SubscriptionPolledRefresh, SubscriptionPolledRefreshResponse
- Cancelación de subscripción: SubscriptionCancel, SubscriptionCancelResponse
- Descubrimiento de variables: Browse, BrowseResponse
- Lectura de propiedades de variables: GetProperties, GetPropertiesResponse

Además de ello, en la especificación se describen los distintos tipos de parámetros que se esperan como argumentos de estos servicios o como su respuesta. De esta manera, OPC XML-DA define un comportamiento predecible tanto para el servidor como para el cliente.<sup>4</sup>

## 4. Conceptos de programación en PLC

En esta sección se describen los conceptos fundamentales utilizados para la programación del banco de ensayo. Si bien algunos de estos conceptos pueden trasladarse a los entornos de desarrollo de otros fabricantes, este apartado es válido solamente para el controlador SIMOTION D4255 en el entorno de programación SIMOTION SCOUT V4.4.

### 4.1. Tipos de dato

Este apartado se dedica a la descripción de las estructuras de información utilizadas. Es posible distinguir 4 tipos de dato diferentes: los tipos de dato elementales, los tipos de dato definidos por usuario, los tipos de dato de sistema y los objetos de tecnología.

#### 4.1.1. Tipos de datos elementales

El sistema de control cuenta con los siguientes tipos de dato elementales:

---

<sup>4</sup>Como no es el objeto de estudio de esta tesis, se presentaron solamente las nociones básicas del estándar OPC XML-DA. La especificación del estándar profundiza en cuestiones operativas y control de acceso que, para los fines de este estudio, aquí no se contemplan.

Tipo	Palabra reservada	Longitud en bits
<b>Tipo de dato de bit</b>		
Bit	BOOL	1
Byte	BYTE	8
Palabra	WORD	16
Palabra doble	DWORD	32
<b>Tipo de dato numérico</b>		
Entero corto	SINT	1
Entero corto sin signo	USINT	8
Entero	INT	16
Entero sin signo	UINT	16
Entero doble	DINT	32
Entero doble sin signo	UDINT	32
Número de punto flotante	REAL	32
Número de punto flotante largo	LREAL	64
<b>Tipo de dato de fecha</b>		
Incrementos temporales de 1ms	TIME	32
Fecha con incrementos temporales de 1 día	DATE	32
Hora del día con incrementos de 1ms	TIME_OF_DAY	32
Fecha y hora	DATE_AND_TIME	64
<b>Caracteres</b>		
Caracter de un byte	STRING	8

Cuadro 7: Tipos de datos elementales

#### 4.1.2. Tipos de datos definidos por usuario (UDT, por sus siglas en inglés)

El usuario puede extender la funcionalidad de los tipos de datos enumerados en la sección anterior introduciendo tipos de datos personalizados. Las posibilidades para realizar ello son a través de dos componentes:

- Enumeraciones: es un tipo de dato utilizado para agrupar un conjunto finito, entero, constante y mutuamente excluyente de valores.
- Estructuras: es un tipo de dato compuesto que permite agrupar varios tipos de datos en una única unidad.

Tanto las enumeraciones como las estructuras suelen utilizarse para aumentar la legibilidad del software. Las estructuras permiten conglomerar, bajo un mismo nombre, datos vinculados entre sí conceptualmente. Es decir que un objeto o proceso, representado por una estructura, puede ser descrito por las variables que contiene.

Las enumeraciones permiten asignar significados a un conjunto de valores fijos, agrupados bajo una misma denominación. El estado de un objeto, definido como una enumeración, puede mutar a un conjunto finito de otros estados posibles.

#### 4.1.3. Tipos de dato de sistema

Existe un gran número de tipos de dato de sistema. Por el bien de la brevedad, no serán enumerados individualmente.

Estos tipos de datos suelen ser enumeraciones o estructuras definidas por el fabricante del sistema de accionamientos. Su uso frecuentemente se vincula a tareas de sistema, configuración y uso de objetos de tecnología y modificación de parámetros de sistema de accionamiento, entre otros.

En particular, el tipo de dato de sistema StructAlarmId suele ser utilizado para la generación de mensajes de alarma.

#### 4.1.4. Objetos de tecnología (TO, por sus siglas en inglés)

Los objetos de tecnología de Siemens representan una funcionalidad tecnológica, pueden estar estrechamente ligados con el uso final que recibirá un módulo, pero también pueden representar funcionalidades básicas de este módulo. Los TO permiten agilizar el desarrollo al ofrecer funciones ya implementadas.

A modo de ejemplo: el objeto de tecnología «Axis» agrupa un servomotor y su accionamiento en un único objeto. Esta configuración permite que una misma estructura cuente con la información del encoder del motor y sus consignas de velocidad, corriente y cupla, permitiendo el acople entre la información del servomotor y la de su accionamiento.

De esta forma, es posible definir un eje de posición o de velocidad, utilizarlo y configurarlo en pocos comandos, sin tener la necesidad de interactuar directamente con los telegramas de PROFIDrive.

Naturalmente, la abstracción que provee el objeto implica una menor flexibilidad para realizar aplicaciones especiales y, además, requiere de determinadas configuraciones de comunicación.

Se presentan a continuación algunos ejemplos de objetos tecnológicos Siemens:

- Axis: Permite crear ejes lineales o rotativos con aplicaciones con consigna de velocidad o consigna de posición. Provee una abstracción para la lectura del encoder - integrado o externo- y para la mecánica del eje de posición.
- ALM: Módulo activo de línea. Permite controlar el rectificador totalmente controlado. Implementa funciones de seguridad y de encendido controlado.

Realizar la programación sin Objetos de tecnología implicaría tener que configurar el tipo de telegrama PROFIDrive y, según el comando recibido, programar las distintas respuestas del accionamiento y del servomotor escribiendo directamente en sus parámetros. Adicionalmente, para que este desarrollo sea comparable con el objeto 'Axis' se deberían programar las mismas funciones de seguridad y configuraciones posibles.

Esto último requeriría estudiar con aún mayor profundidad la documentación provista por el fabricante únicamente para alcanzar un desarrollo similar al ofrecido, demorando el desarrollo del banco de ensayos.

Para evitar esto, es conveniente estudiar los objetos de tecnología ofrecidos por el fabricante en la etapa de estudios previos del proyecto. Si resulta necesario exceder la funcionalidad del objeto tecnológico, aún es posible implementar el comportamiento deseado interactuando a más bajo nivel con el sistema de accionamientos.

### 4.2. Tipos de variable

Además de poder definir la estructura de la información con su tipo de dato, también es posible precisar su alcance. Además de ello, las variables permiten almacenar datos dinámicos,

lo que facilita la programación del dispositivo. El sistema de accionamientos SIMOTION D425 ofrece diferentes tipos de variables según su alcance. A continuación se enumeran las principales categorías:

1. Variables de sistema: Son distintas variables relacionadas al estado actual del sistema. Algunos ejemplos de variables de sistema son: estado de operación, mensajes de error, modelo de dispositivo, entre otros. Pueden consultarse mediante OPC XML-DA y suelen tener un tipo de dato de sistema.
2. Variables globales de usuario
  - a) Variables E/S: Se puede asignar nombres de variable simbólicos a las distintas direcciones de entrada/salida del dispositivo SIMOTION. Estas variables son cargadas a un servidor OPC XML-DA para su interacción con sistemas periféricos.
  - b) Variables globales de dispositivo: Son variables definidas por el usuario, pueden consultarse mediante el servidor OPC XML-DA. No están asociadas a una dirección de entrada/salida.
  - c) Variables unitarias: Variables definidas por el usuario que pueden consultarse entre programas dentro del controlador.
3. Variables locales de usuario
  - a) Variables de programa: Son variables que solamente pueden accederse dentro del programa donde fueron declaradas.
  - b) Variables de función: Son variables que pueden consultarse dentro de la función donde fueron declaradas.
  - c) Variables de bloque función: Son variables que pueden consultarse dentro de la función donde fueron declaradas. El modificador *static* permite que las variables retengan su valor luego de que el bloque función haya sido ejecutado.

### 4.3. Programación en LADDER

SIMOTION SCOUT implementa un módulo de programación de lógica LADDER en conformidad con el estándar IEC 61131-3. Los diagramas LADDER permiten implementar lógica digital para el control de una máquina utilizando una interfaz de programación gráfica. En el campo del control industrial, es uno de los lenguajes de programación más difundidos por su programación intuitiva.

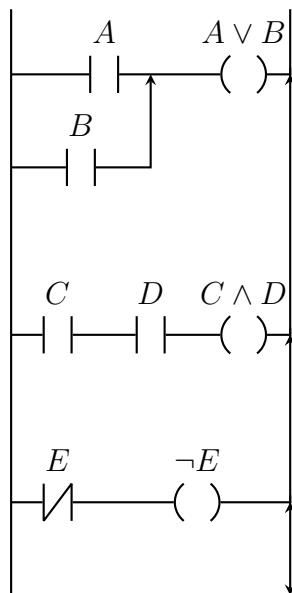


Figura 4: Red LADDER

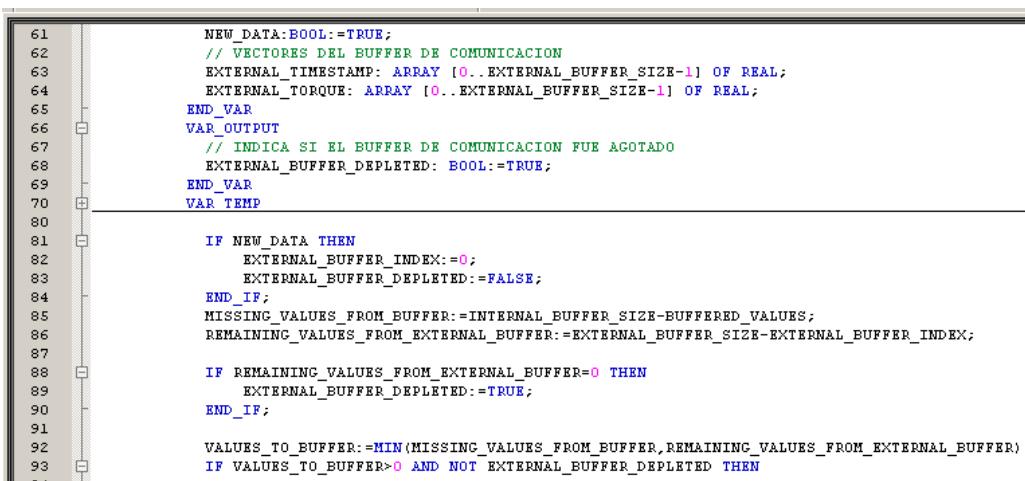
La interfaz gráfica de este lenguaje se basa en una analogía circuitual que permite definir relaciones entre variables. Los extremos del diagrama pueden pensarse como una diferencia de potencial que, al unirlos mediante un interruptor, se permite un flujo de corriente que enciende una bobina. El encendido de la bobina equivale al valor verdadero en la lógica booleana, mientras que los contactos, que pueden ser normales abiertos o normales cerrados, son equivalentes a la entrada de una compuerta lógica.

Utilizando estos bloques es posible componer circuitos lógicos arbitrariamente complejos. Además de esto, el lenguaje admite la activación de bloques función que permiten extender el campo de aplicación del lenguaje más allá de la lógica digital.

En términos generales, se suele utilizar el diagrama LADDER como control de flujo del programa y los bloques función para ejecutar tareas específicas o no digitales. Esto permite organizar el programa de PLC en subprogramas que son llamados por el programa principal de control de flujo.

#### 4.4. Programación en Structured Text (ST, por sus siglas en inglés)

Structured Text es un lenguaje de programación de alto nivel basado en PASCAL. Se encuentra especificado en el estándar IEC 61131-3.



```

61      NEW_DATA:BOOL:=TRUE;
62      // VECTORES DEL BUFFER DE COMUNICACION
63      EXTERNAL_TIMESTAMP: ARRAY [0..EXTERNAL_BUFFER_SIZE-1] OF REAL;
64      EXTERNAL_TORQUE: ARRAY [0..EXTERNAL_BUFFER_SIZE-1] OF REAL;
65      END_VAR
66      VAR_OUTPUT
67      // INDICA SI EL BUFFER DE COMUNICACION FUE AGOTADO
68      EXTERNAL_BUFFER_DEPRECATED: BOOL:=TRUE;
69      END_VAR
70      VAR TEMP
71
72
73
74
75
76
77
78
79
80      IF NEW_DATA THEN
81          EXTERNAL_BUFFER_INDEX:=0;
82          EXTERNAL_BUFFER_DEPRECATED:=FALSE;
83      END_IF;
84      MISSING_VALUES_FROM_BUFFER:=INTERNAL_BUFFER_SIZE-BUFFERED_VALUES;
85      REMAINING_VALUES_FROM_EXTERNAL_BUFFER:=EXTERNAL_BUFFER_SIZE-EXTERNAL_BUFFER_INDEX;
86
87
88      IF REMAINING_VALUES_FROM_EXTERNAL_BUFFER=0 THEN
89          EXTERNAL_BUFFER_DEPRECATED:=TRUE;
90      END_IF;
91
92      VALUES_TO_BUFFER:=MIN(MISSING_VALUES_FROM_BUFFER,REMAINING_VALUES_FROM_EXTERNAL_BUFFER);
93      IF VALUES_TO_BUFFER>0 AND NOT EXTERNAL_BUFFER_DEPRECATED THEN
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259

```

Figura 5: Porción de un programa escrito en ST

ST surge como respuesta al incremento en tráfico y procesamiento de datos en las distintas industrias. Tiene como objetivo simplificar la programación de componentes de automatización, ofreciendo funciones de comunicación, de operación entre módulos, matemáticas y estadísticas.

#### 4.4.1. Unidades organizacionales de programa (POU, por sus siglas en inglés)

En ST se define como POU a todo archivo de código susceptible de ser ejecutado, se definen las siguientes categorías :

- Program: Los programas no admiten parámetros y solamente pueden ser llamados por el sistema de ejecución. Pueden contener bloques función y funciones.
- Function (Abreviado como FC por el fabricante): Las funciones son POUs que pueden ser llamados por otros programas y admiten como entrada parámetros. Las funciones internamente operan con variables almacenadas en direcciones de memoria que se liberan luego de cada llamado a función.
- Function block (FB, por sus siglas en inglés): Los bloques función son similares a las funciones a diferencia de que pueden contener variables estáticas. Al permitir esto, las direcciones de memoria donde los FB almacenan sus datos no son liberadas luego del llamado a función. Esto permite implementar variables con permanencia dentro del POU.
- Expressions: Las expresiones en ST son un caso especial de las funciones. Obligatoriamente deben retornar una variable booleana y admiten parámetros. Se suelen utilizar como condiciones para los operadores de flujo (if, while y sus derivados).

#### 4.5. Sistema de ejecución

Siemens ofrece un sistema de ejecución versátil y organizado. El sistema de ejecución SIMOTION SCOUT se encuentra conformado por dos partes: los estados de operación del equipo y los niveles de ejecución asignados a los programas.

##### 4.5.1. Estados de operación

Los estados de operación definen el nivel de control que el usuario tiene sobre el dispositivo. La situación típica es que el control ofrezca un modo de operación en donde el usuario pueda ejecutar su propio código y otros estados reservados para tareas de mantenimiento y respuesta ante errores.

- RUN: El equipo se encuentra ejecutando un programa de usuario.
- STOPU: El programa de usuario no está activo, pero los módulos de I/O y los paquetes de tecnología están activos. En este estado es posible el movimiento de ejes para realizar pruebas de funcionamiento.
- STOP: El programa de usuario no está activo, los modulos de I/O se encuentran en modo seguro y no se permite la ejecución de comandos de movimiento.
- MRES: Se reestablece el sistema a su configuración de fábrica.

Usualmente se suele operar el sistema entre los estados de STOP y RUN. Reservando los demás estados para tareas de mantenimiento y puesta en marcha.

#### 4.5.2. Niveles de ejecución

El fabricante ofrece diferentes tipos de niveles de ejecución para los programas de usuario. Los niveles de ejecución definen el momento en el que se ejecutará una tarea. El programador de la máquina debe asignar distintos programas a estos niveles de ejecución en función de las necesidades del código implementado. A continuación, se realiza un breve listado comentando los distintos niveles:

1. Controladas por tiempo: Son tareas cíclicas reiniciadas automáticamente, a intervalos de tiempo fijo, luego de la ejecución de los demás programas asignados a su nivel de ejecución.
  - a) SynchronousTask: Son tareas sincronizadas con algunos de los ciclos de reloj del sistema. Pueden tener períodos de escaneo en el orden del milisegundo, pero las instrucciones a ejecutar deben ser simples y rápidas.
  - b) TimerInterruptTasks: Son tareas que inician periódicamente a un intervalo fijo.
2. Interrupciones: Son tareas secuenciales que se inician una única vez luego del inicio del equipo.
  - a) SystemInterruptTask: se ejecutan cuando ocurre un evento de sistema<sup>5</sup>. Se reservan para errores de programación en tiempo de ejecución.
  - b) UserInterruptTasks: Se ejecutan cuando ocurren un evento definido por el usuario.
3. Round robin: Son tareas que comparten el tiempo libre luego de que el equipo haya ejecutado el resto de las tareas de sistema.
  - a) MotionTasks: Son tareas secuenciales ejecutadas con llamados a tarea.
  - b) BackgroundTasks: Son tareas cíclicas. El periodo de escaneo de las tareas es variable y se define en tiempo de ejecución.
4. StartupTask: es una tarea que se ejecuta una única vez con la transición del estado de STOP al estado de RUN.
5. ShutdownTask: es una tarea que se ejecuta una única vez con la transición del estado de RUN al estado de STOP.

---

<sup>5</sup>Siemens define cuatro eventos de sistema: ExecutionFaultTask, PeripheralFaultTask, TechnologicalFaultTask, TimeFaultTask y TimeFaultBackgroundTask. En todos los casos se asocian a condiciones de error específicas

## 5. Aplicaciones del banco de ensayos

Para definir los requisitos de diseño del banco de ensayos, fue necesario contemplar diversos casos de uso posibles. En esta sección se recopilan algunas de las aplicaciones posibles del banco de ensayos que se consideraron como base para definir las prestaciones del banco de ensayos.

### 5.1. Ensayos mecánicos en general

Aplicando conceptos de mecánica clásica, es posible interpretar la cupla resultante sobre un cuerpo como la variación de su momento angular:

$$\vec{T}_{resultante} = \frac{\partial \vec{L}}{\partial t} \quad (1)$$

Considerando únicamente el movimiento unidimensional, se puede descomponer el vector momento angular en una sola coordenada espacial. De esta forma, es posible vincular el momento de inercia del cuerpo, en esa coordenada espacial, con su variación de velocidad angular. Bajo esta simplificación, el torque resultante es:

$$T_{resultante} = J \frac{\partial \omega}{\partial t} \quad (2)$$

Bajo la hipótesis de que el torque resultante proviene de la interacción entre un torque de tracción y un torque de frenado:

$$T_{traccion} - T_{frenado} = J \frac{\partial \omega}{\partial t} \quad (3)$$

En esta ecuación, la única variable manipulada por el banco de ensayo es el torque de frenado. La cupla de tracción está definida por el dispositivo bajo ensayo, el momento de inercia del dispositivo no puede ser modificado al ser una constante física y la variación de velocidad angular es dependiente de la interacción entre cuplas.

Sin embargo, eligiendo un valor no constante para la cupla de frenado, es posible emular diferentes condiciones de carga sobre el eje de tracción. A modo de ejemplo, suponiendo que  $T_{frenado} = T'_{frenado} + J' \frac{\partial \omega}{\partial t}$  entonces es posible reescribir la ecuación de torque de la siguiente manera:

$$T_{motriz} - T'_{frenado} = (J + J') \frac{\partial \omega}{\partial t} \quad (4)$$

Esto implica que el cuerpo responderá con una nueva inercia  $I + I'$ , en donde  $I'$  potencialmente podría ser un valor negativo, mientras que la suma con  $I$  sea mayor a cero. De esta manera, es posible ensayar el dispositivo simulando distintas distribuciones de masa.

Además de esto, si  $T'_{frenado}$  fuese una función de la velocidad angular, entonces sería posible simular condiciones de carga dinámicas, como se verá en las secciones a continuación.

### 5.2. Ensayos para motores para vehículos

Es posible obtener un modelo matemático de la dinámica de un vehículo considerando la presencia de fuerzas actuantes.

Suponiendo un vehículo de dos ejes con tracción delantera con una masa  $M$ , el rodado estará sometido a una fuerza de resistencia a la rodadura en cada eje  $F_{Fr}$  y  $F_{Fd}$ . Además de ello, al mantener una velocidad distinta de cero, existirá fuerza de resistencia aerodinámica  $D_A$ . Si el vehículo estuviese sobre un plano inclinado con una pendiente  $\alpha$ , también actuaría la acción

gravitatoria proporcional al seno del ángulo de pendiente. Por último, el auto se autoimpulsaría con una fuerza  $F_T$ .

$$M \frac{dv}{dt} = F_{Td} - F_{Fd} - F_{Ft} - D_A - Mgsen(\alpha) \quad (5)$$

Es posible describir la resistencia aerodinámica en función de la densidad del aire  $\rho$ , la velocidad lineal del vehículo, su área frontal  $A_f$  y un coeficiente aerodinámico  $C_d$ .

$$M \frac{dv}{dt} = F_{Td} - F_{Fd} - F_{Ft} - \frac{1}{2} \rho A_f C_D v^2 - Mgsen(\alpha) \quad (6)$$

Adicionalmente, se puede modelar la fuerza de resistencia a la rodadura como proporcional a la fuerza normal del vehículo. La fuerza normal del vehículo será proporcional al coseno del ángulo de pendiente  $\alpha$ :

$$M \frac{dv}{dt} = F_{Td} - f_r M g \cos(\alpha) - \frac{1}{2} \rho A_f C_D v^2 - Mgsen(\alpha) \quad (7)$$

Por último, es necesario referir la contribución de estas fuerzas al eje del motor del vehículo. Para realizar ello, es necesario considerar la relación entre la aceleración angular y la sumatoria de torques,  $J \frac{d\omega_m}{dt} = \sum F \cdot R$ , y que las fuerzas se aplican en la superficie de la rueda del vehículo.

$$J \frac{d\omega_m}{dt} = T_{Td} - \frac{r_d}{i_0 i_g} f_r M g \cos(\alpha) - \frac{r_d}{i_0 i_g} \frac{1}{2} \rho A_f C_D \left( \frac{\omega_m r_d}{i_0 i_g} \right)^2 - \frac{r_d}{i_0 i_g} Mgsen(\alpha) \quad (8)$$

El momento de inercia del conjunto, motor, rueda y vehículo referido al motor puede obtenerse como:  $J = J_{motor} + \frac{J_{rueda}}{i_0^2 i_g} + \frac{M r_d^2}{i_0^2 i_g}$ . El modelo del vehículo queda definido por la siguiente ecuación:

$$(J_{motor} + \frac{J_{rueda}}{i_0^2 i_g} + \frac{M r_d^2}{i_0^2 i_g}) \frac{d\omega_m}{dt} = T_{Td} - \frac{r_d}{i_0 i_g} f_r M g \cos(\alpha) - \frac{r_d}{i_0 i_g} \frac{1}{2} \rho A_f C_D \left( \frac{\omega_m r_d}{i_0 i_g} \right)^2 - \frac{r_d}{i_0 i_g} Mgsen(\alpha) \quad (9)$$

$$T_{Td} = (J_{motor} + \frac{J_{rueda}}{i_0^2 i_g} + \frac{M r_d^2}{i_0^2 i_g}) \frac{d\omega_m}{dt} + \frac{r_d}{i_0 i_g} f_r M g \cos(\alpha) + \frac{r_d^3}{i_0^3 i_g} \frac{1}{2} \rho A_f C_D \omega_m^2 + \frac{r_d}{i_0 i_g} Mgsen(\alpha) \quad (10)$$

Eligiendo la cupla de frenado igual a  $T_{Td}$  es posible simular la carga dinámica del vehículo. Poder llevar a cabo un ensayo de este tipo permitiría recolectar información acerca del consumo de batería o combustible, la eficiencia del sistema de conversión y exigencias mecánicas y eléctricas, entre otros. Esta caracterización asistiría en el diseño mecánico del vehículo y en el diseño de su sistema de accionamientos y batería, en el caso de que se trate de un vehículo con motor eléctrico.

### 5.3. Simulación de ciclos de trabajo en máquinas-herramienta

En la industria de la manufactura sustractiva, las máquinas-herramienta suelen funcionar con cierto perfil de velocidades durante su ciclo de trabajo. Estas velocidades dependen de si el mecanizado es de desbaste grueso, desbaste fino, de terminación y también del tipo de material. Además, las máquinas suelen trabajar en producción seriada de piezas, por lo que sus ciclos de trabajo suelen repetirse durante horas.

Usualmente los materiales isotrópicos y rígidos, como el metal, suelen ofrecer una resistencia al mecanizado que depende del área del material en contacto con la herramienta, su velocidad de avance y su temperatura. Para un ciclo de trabajo en concreto, esta resistencia puede modelarse como una curva de cupla en función del tiempo. En la figura a continuación se muestra un ejemplo:

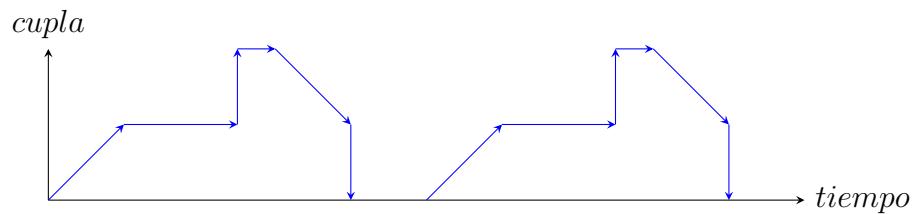


Figura 6: Ejemplo de la carga en el husillo de una máquina herramienta durante el mecanizado de piezas

La figura 6 podría explicar la interacción entre un husillo y el material a ser mecanizado. La pendiente positiva en el gráfico indica que el husillo comienza a entrar en contacto con el material gradualmente. Cuando la cupla es constante, el husillo está en contacto con la misma cantidad de material. Finalmente, las pendientes negativas en el gráfico indican que el husillo deja de estar en contacto con el material. Por lo general, los perfiles de cupla-tiempo en máquinas-herramientas suelen ser periódicos, ya que se utilizan en la producción seriada de piezas.

Usualmente, es de interés en la industria que el ciclo de trabajo de una pieza sea lo más pequeño posible, manteniendo la precisión del mecanizado. Esto obliga a acelerar el husillo a su consigna de velocidad lo más rápido posible y a desplazar el husillo sobre el material con el mayor avance posible. A raíz de esta necesidad, existe un profundo interés en nuevas estrategias de control que permitan aumentar la rigidez de la velocidad del husillo frente a las perturbaciones que ofrece el material.

Los ensayos de motores de husillo en condiciones de laboratorio permiten evaluar su desempeño dinámico frente a distintos tipos de ciclo de trabajo y condiciones de corte. Los datos recolectados pueden ser utilizados para el diseño de máquinas-herramienta y el ensayo de nuevas estrategias de control.

#### 5.4. Simulación de carga para generadores mareomotrices

El principio de la energía mareomotriz es el del aprovechamiento de la diferencia de potencial causada por el oleaje y por las mareas de un cuerpo de agua. Dado que es un área de la generación sustentable relativamente poco explorada, existe un gran número de dispositivos generadores posibles. Entre ellos es posible distinguir 2 tipos principales: rotativos y oscilantes.

Los generadores rotativos tienen un principio de funcionamiento similar al de los generadores eólicos, la principal diferencia es que deben ser invertibles y, en ocasiones, pueden estar acompañados de una estructura secundaria para concentrar el flujo de agua. A grandes rasgos, el sistema utiliza el caudal de agua que se produce durante los ciclos descendentes y ascendentes de la marea para hacer girar las aspas de un rotor sumergido, generando energía.

Los generadores oscilantes consisten en un perfil aerodinámico sumergido y acoplado a una guía perpendicular al sentido de la marea y el oleaje. Durante el ciclo ascendente, el perfil se eleva, mientras que durante el ciclo descendente, se deprime. Acoplado a un sistema mecánico adecuado, es posible transformar el movimiento lineal en rotativo rectificado, permitiendo aprovechar ambos ciclos de movimiento.

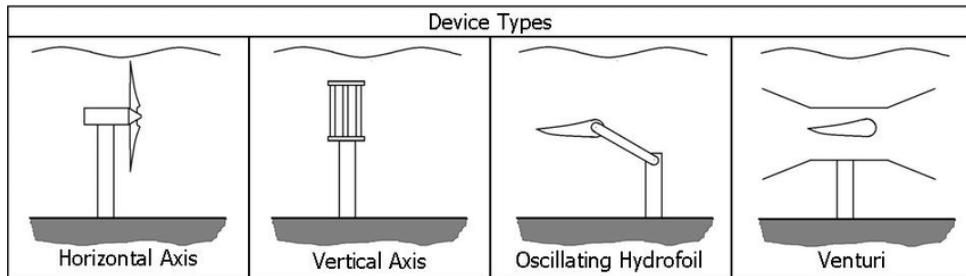


Figura 7: Tipos de generadores mareomotrices rotativos y oscilantes

En ambos casos, los generadores responden a un sistema viscoso y dinámico que les impone cierto torque que permite el movimiento de sus ejes, generando energía. El comportamiento del sistema podría ser descompuesto en dos partes: la cupla impuesta en función del tiempo y la cupla impuesta en función de la velocidad.

En función del tiempo, la cupla será la superposición de una cupla senoidal de alta frecuencia que representa el oleaje y una cupla senoidal de baja frecuencia, que simula las mareas. En función de la velocidad, el sistema ofrecerá una resistencia viscosa y una inercia que depende del diseño mecánico del sistema de aprovechamiento de energía. La velocidad de rotación será impuesta por el generador con el fin de maximizar la energía recolectada.

Luego de construir un generador mareomotriz, la validación del diseño es compleja debido a la ubicación del generador, las condiciones en campo del día de la medición y la dificultad para repetir las mediciones. Poder simular la cupla que la marea impone sobre los dispositivos, así como la característica viscosa del medio, implicaría poder realizar pruebas en laboratorio que permitirían evaluar el diseño previo a las pruebas en campo.

## 5.5. Simulación de carga para generadores eólicos

Los generadores eólicos presentan algunos de los mismos desafíos que su par mareomotriz. Ambos funcionan en un fluido viscoso y dependen de las corrientes que se producen en el medio. La generación eólica es un sector relativamente maduro con tecnologías establecidas.

Sin embargo, aún existen desarrollos novedosos en el área. Por este motivo, resulta de utilidad contar con un mecanismo que permita emular la condición de carga del generador en condiciones repetibles.

# 6. Requisitos de diseño

Con el objetivo de garantizar que el diseño del banco sea lo más compatible posible con su uso efectivo, se consultó a los futuros operadores para conocer sus necesidades y analizar el uso que podría tener el equipo.

En esta sección se tratan las exigencias de diseño del banco de ensayos producto de las conversaciones con los operadores del banco de ensayos. Los resultados de estos encuentros se resumen en la tabla 8, final de esta sección.

## 6.1. Interfaz con el usuario

Previo al desarrollo, se realizaron entrevistas con posibles operadores del banco de ensayos. A grandes rasgos, los entrevistados coincidieron en que la interfaz debe ser lo suficientemente simple como para que sea intuitiva, pero lo suficientemente compleja para admitir ensayos configurables.

Los testimonios sugirieron la importancia de una interfaz física, con pulsadores y luces, que informe condiciones básicas de operación y permita la interacción con el banco para iniciar, pausar y detener ensayos.

Durante este relevamiento se propuso la existencia de una interfaz gráfica por computadora que permita la visualización de gráficos en vivo, replique funcionalmente a la interfaz física y permita descargar los datos de simulación de al menos cinco variables: cupla electromagnética, velocidad angular, tensión eléctrica en eje simulador, corriente en eje simulador y potencia activa en eje simulador.

### 6.1.1. Seguridad del banco

Durante las entrevistas con operadores surgió como principal objetivo del banco de ensayos la seguridad del instrumento. Es importante notar que, por las características de los servomotores descritas en la tabla 1, los servomotores tienen la posibilidad de generar daños a personas y equipos si no son utilizados correctamente.

En esta sección se recopilan los principales temas discutidos y algunas de las soluciones adoptadas para mitigar y evitar los riesgos operativos durante la utilización del banco de ensayos.

#### 6.1.1.1 Atrapamiento mecánico

Debido a su característica rotativa, existe un riesgo de atrapamiento mecánico en el eje de la máquina. Con esto en mente, se planearon medidas preventivas y medidas para atenuar el daño en caso de producirse atrapamiento mecánico.

Para evitar el riesgo, se implementaron tres medidas diferentes. En primer lugar, se deberá evitar la cercanía al equipo con ropa holgada o bolsos. Con este fin, está demarcada una zona de exclusión en el piso para evitar el ingreso de personas durante un ensayo.

En segunda instancia, será necesario un sistema lumínico que indique los momentos en los cuales el eje está rotando o podría comenzar a rotar si recibe el comando de arranque. De esta forma, las personas que presencien un ensayo serán notificadas de los momentos de alto riesgo del ensayo.

Por último, será necesaria la instalación de una jaula de seguridad sobre el acople del eje simulador y el dispositivo bajo prueba. Esta medida evita el ingreso no voluntario de objetos o personas al rotor del eje simulador.

Para disminuir el daño, se deberán implementar paradas de emergencia tanto en el eje simulador como en el eje bajo prueba. Una de estas paradas de emergencia será por software, mientras que la otra estará en una botonera física, que el operador pueda portar con facilidad. De esta forma, en caso de que solamente hubiese un único operador, él podría accionar la emergencia aún estando atrapado en la máquina.

#### 6.1.1.2 Desconexión de la interfaz

Considerando que las interfaces físicas y por software contarán con paradas de emergencia, puede existir la posibilidad de que una de estas interfaces no esté en funcionamiento y el operador no lo sepa. Esto haría que, frente a una emergencia, el operador accione un botón de emergencia que no esté operativo. Esta situación retardaría el tiempo de respuesta ante una falla, lo que acabaría en un daño mayor.

Por este motivo, es necesario que el banco de pruebas pueda identificar y actuar frente a la desconexión de alguna de las interfaces.

## 6.2. Cupla de simulación

Evaluando los modelos estudiados en la sección 5, el banco de simulación necesariamente debería poder simular al menos dos tipos de condiciones de carga: las dependientes del tiempo y las dependientes de la velocidad.

Para las primeras, se definirá una ecuación de cupla resistente con coeficientes modificables según el ensayo a realizar:

$$T_{simulado} = A + B\omega + C\omega^2 + D\frac{\partial\omega}{\partial t} \quad (11)$$

Sin embargo, en la práctica los coeficientes de la ecuación deberán respetar los límites del servomotor, detallados en la tabla 3.

Además, el coeficiente D amplifica cualquier error en la medición de la velocidad, por lo que también se verá limitado en los valores que pueda adoptar. Como requisito de diseño, se estableció que el coeficiente, asociado a la inercia, deberá adoptar valores hasta 7 veces el valor de inercia de rotor.

Para las condiciones de carga dependientes del tiempo se considerará, como entrada, una tabla de pares cupla-tiempo. Esta tabla deberá admitir una cantidad de entradas arbitraria y el tiempo entre pares no necesariamente deberá ser equiespaciado. Como mínimo, el tiempo entre comandos será de 100ms.

Finalmente, se implementará un tipo de ensayo mixto que es la superposición de ensayos de tipo cupla-tiempo y ensayos de cupla-velocidad. De esta forma, el banco incorpora un nuevo tipo de ensayo que permite ampliar la frontera de posibilidades de aplicación.

### 6.3. Requisitos de diseño - tabla de síntesis

Id.	Tipo de requisito	Descripción
I.0	Interfaz	Interfaz distribuida en parte física (IF) y parte gráfica (IG)
I.1	Interfaz	Diseño intuitivo, que permita la operación con mínima experiencia
I.IF.0	Interfaz física	Interfaz física con botonera que permita iniciar, reiniciar y detener ensayos.
I.IF.1	Interfaz física	Interfaz física con luces que indiquen los estados del ensayo.
I.IG.0	Interfaz gráfica	Interfaz virtual que permita iniciar, pausar y detener ensayos
I.IG.1	Interfaz gráfica	Visualización de gráficos en tiempo real de al menos 5 variables: cupla medida, tensión en motor, corriente, potencia activa y velocidad.
I.IG.2	Interfaz gráfica	Exportación de los datos de las 5 variables en formato CSV.
I.IG.3	Interfaz gráfica	Desarrollo multisistema, mantenible y flexible.
F.0	Funcional	Admitir ensayos de cupla variable en función de la velocidad (CV) y cupla variable en función del tiempo (CT) en el eje simulador.
F.1	Funcional	Resolución temporal mínima de 150ms.
F.2	Funcional	Permitir la realización de ensayos utilizando el segundo motor del banco de ensayos como eje de tracción.
F.CV.0	Cupla vs velocidad	Admitir cuplas variables con la velocidad según lo descrito en la ecuación 11.
F.CV.1	Cupla vs velocidad	Permitir la simulación de coeficientes iniciales de al menos 7 veces la inercia del rotor del servomotor.
F.CT.0	Cupla en función del tiempo	Admitir cuplas variables con el tiempo.
F.CT.1	Cupla en función del tiempo	Admitir un tiempo mínimo entre comandos de torque de al menos 100ms.
F.CT.2	Cupla en función del tiempo	Permitir comandos de torque no equidistantes en el tiempo.
SW.0	Software general	Software mantenible, multiplataforma y escalable.
S.0	Seguridad	Aplicar límites máximos de velocidad y cupla en el eje simulador.
S.1	Seguridad	Aplicar límites máximos de derivada de cupla en el eje simulador.
S.2	Seguridad	Implementar freno de emergencia en hardware y software.
S.3	Seguridad	Prohibir la utilización del equipo con alguna de las interfaces inactivas.

Cuadro 8: Tabla de requisitos de diseño para el simulador

## 7. Arquitectura de la solución

En esta sección se exponen las decisiones de diseño adoptadas para cubrir los requisitos del sistema, junto con los fundamentos que las justifican.

Adicionalmente, se mencionan los enfoques alternativos que fueron explorados durante el proceso de diseño.

### 7.1. Diagrama en bloques de la solución

Con el fin de poder describir cada uno de los elementos de la solución adoptada y sus relaciones, se realizó el siguiente diagrama en bloques:

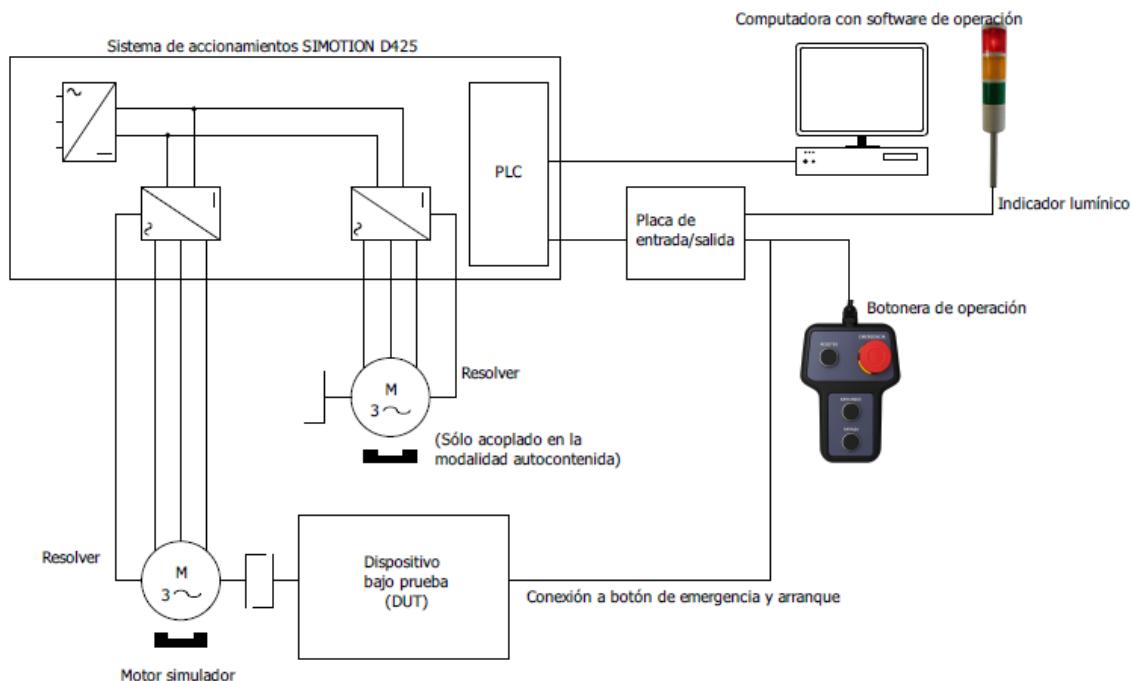


Figura 8: Diagrama en bloques de la solución diseñada: partes y conexiones

Considerando este diagrama básico, es posible definir las relaciones y señales entre sus módulos y componentes internos. En la figura 9, se describen las señales y protocolos utilizados:

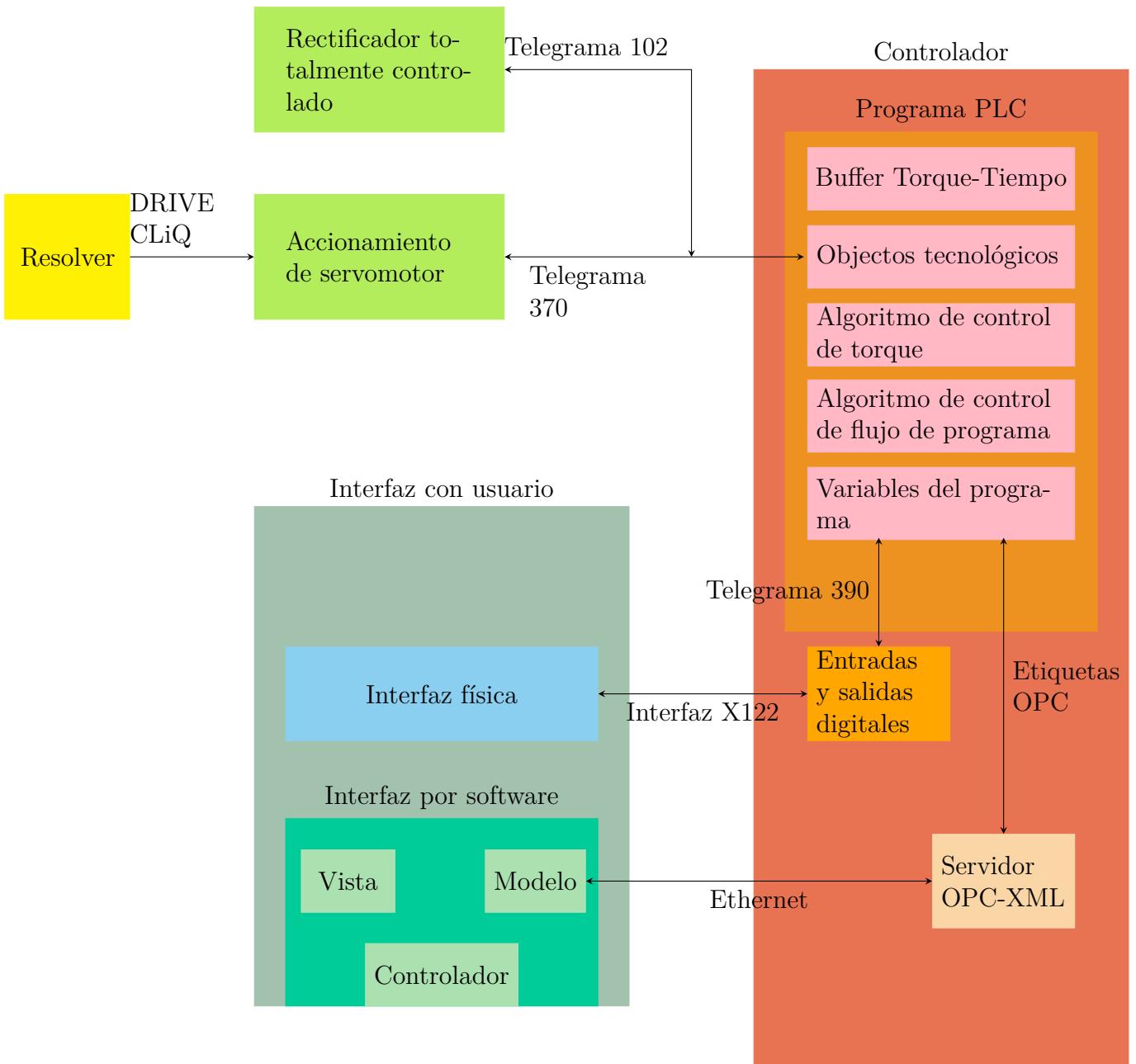


Figura 9: Diagrama en bloques de la solución diseñada: componentes internos y señales

A grandes rasgos, la solución consiste en un programa que corre dentro del SIMOTION D425 y que, usando objetos tecnológicos, controla cada elemento del sistema de accionamientos. Este programa es el encargado de aplicar las consignas de cupla-tiempo y cupla-velocidad, de forma tal de que ambos comandos se apliquen de manera precisa.

Para interactuar con el programa PLC existen dos maneras posibles: mediante las variables IO, ligadas a puertos físicos en la interfaz X122, o bien a través de las etiquetas OPC XML-DA.

Las variables IO se utilizan para el control de la botonera y el sistema de alertas lumínicas. Esta botonera permite iniciar, detener, parar en emergencia y reconocer la emergencia del equipo, mientras que las alertas lumínicas indican el estado de ejecución del ensayo.

Las etiquetas OPC XML-DA son un conjunto de variables configuradas para ser cargadas al servidor OPC XML-DA. A través de la interfaz gráfica, el usuario carga los parámetros de ensayo y estos valores se escriben en el servidor OPC XML-DA para luego ser utilizados dentro del programa PLC. Además de ello, la interfaz gráfica presenta mediciones de las distintas

variables de interés en un gráfico en tiempo real.

## 7.2. Interfaz por software

El fabricante ofrece una biblioteca en JAVA de funciones OPC XML-DA para poder conformar un cliente OPC XML-DA y comunicarse con el servidor, agilizando el desarrollo de software. JAVA es un lenguaje de programación orientado a objetos lanzado por la empresa Sun Microsystems en el año 1995, como una alternativa portable y de más alto nivel que C y C++, pero con sintaxis similar.

El código escrito en JAVA no corre directamente en el sistema operativo que lo ejecuta, sino que corre dentro de una máquina virtual de JAVA (JVM, por sus siglas en inglés). El código, al correr siempre dentro de una máquina virtual, puede funcionar en cualquier plataforma para la cual se haya desarrollado una JVM. De esta forma, la JVM funciona como un intermediario entre el sistema operativo nativo y el código JAVA, ofreciendo una interfaz estandarizada a ambos lados.

Al ser un lenguaje maduro con estas características, la adopción entre la comunidad de desarrolladores fue rápida y continua siendo una plataforma relevante a 30 años de su lanzamiento. La suma de estos factores convirtió al lenguaje en una elección clásica para la implementación de programas con interfaz gráfica.

Adoptarlo como lenguaje de programación para el software de interfaz se puede asociar con tres ventajas principales. La primera de ellas es poder utilizar la biblioteca de Siemens para la comunicación OPC XML-DA, sin tener que reescribirla en otro lenguaje. En segundo lugar, se asegura que el software desarrollado pueda ser actualizado en el futuro por personas con conocimientos generales de programación y con acceso a nuevas bibliotecas aún no desarrolladas. Por último, el diseño inherentemente multiplataforma de JAVA lo hace prácticamente inmune al cambio de sistemas operativos, lo que podría mantener en el tiempo el software desarrollado.

Con el fin de que el desarrollo pueda ser mantenido a futuro, es necesario programar el software de forma tal de utilizar patrones u estructuras estandarizadas. Si posteriormente se decide adaptar el código para incorporar nuevas funcionalidades, la persona a cargo de esta tarea no deberá estudiar el código en profundidad para interpretar sus relaciones funcionales, sino que podrá reconocer estructuras típicas en el código y rápidamente modificarlo.

Con esto en mente, la arquitectura de la interfaz de software seguirá con el patrón Modelo-Vista-Controlador (MVC). De esta forma se obtendrá un código altamente desacoplado y organizado en 3 unidades funcionales. El patrón MVC y sus variantes surgen con las primeras interfaces gráficas a principios de la década del 70 y hasta el día de hoy sigue siendo una elección popular para este tipo de desarrollos.

## 8. Ingeniería de la solución

La arquitectura del sistema, previamente detallada en la sección anterior, se deberá implementar en el controlador SIMOTION D425, en la computadora donde funciona la interfaz por software y físicamente en el panel de operador.

La implementación del sistema requiere conciliar los requisitos del desarrollo con las limitaciones técnicas del sistema de accionamientos. Necesariamente, esta tarea requiere la adopción de diversas decisiones de ingeniería para materializar la solución diseñada.

En esta sección se describe la implementación de la arquitectura del sistema y se analizan los fundamentos que sustentan las decisiones de diseño adoptadas, así como las alternativas evaluadas durante el proceso.

Se divide la ingeniería del sistema en 3 partes:

- Programación en la unidad de control
- Interfaz física del banco de ensayos
- Interfaz por software del banco de ensayos

## 8.1. Programación en la unidad de control

Dentro del programa que corre en el PLC se pueden distinguir 4 unidades fundamentales:

- Algoritmo de control de flujo de programa.
- Algoritmo de control de torque por saturación del lazo de velocidad.
- Algoritmo de cálculo de torque en función del tiempo.
- Algoritmo de cálculo de torque en función de la velocidad.

El algoritmo de control de flujo es el encargado de procesar los comandos que recibe del usuario y asegurar que el flujo del programa se ejecute de la manera diseñada. Por ejemplo: si el usuario aún no cargó los parámetros del ensayo, al presionar el botón de arranque, el sistema no debería habilitar el inicio del ensayo.

Por otra parte, el algoritmo de control de torque por saturación del lazo de velocidad se encarga de aplicar correctamente la consigna de torque independientemente del tipo de ensayo en ejecución.

A su vez, el algoritmo de cálculo de torque en función del tiempo se encarga de aplicar las consignas de cupla-tiempo de forma tal que las consignas de cupla coincidan con su instante de aplicación. Este algoritmo también comprende la gestión de una memoria intermedia que asegura la disponibilidad de pares cupla-tiempo considerando el retardo en la transmisión de datos.

Finalmente, el algoritmo de cálculo de torque en función de la velocidad computa la ecuación de cupla-velocidad. Para su cálculo, fue necesario contemplar el ruido de la velocidad medida y su impacto en los términos de la ecuación.

Estos dos últimos algoritmos alimentan el algoritmo de control de torque de a un único valor a la vez. El algoritmo de control de cupla se alimenta secuencialmente: cuando finalice el tiempo de aplicación de cupla, simplemente se avanzará a la siguiente consigna de cupla a aplicar.

### 8.1.1. Algoritmo de control de flujo del programa

Para asegurar la correcta interacción con el banco de ensayos, se modeló el comportamiento del sistema utilizando una máquina de estados finitos. De esta forma, se podrá controlar las reacciones del sistema de una manera clara y fácil de predecir.

Las máquinas de estados finitos o autómatas son un modelo matemático utilizado para controlar sistemas dinámicos y reactivos. Este tipo de sistemas requieren de comportamientos marcadamente diferentes en función de la entrada actual y las anteriores.

El autómata tiene un conjunto de estados posibles, cada estado tiene asociadas diferentes reglas de transición para acceder a los demás estados del sistema. De esta forma, el modelo es lo suficientemente flexible como para implementar comportamientos complejos utilizando una estructura sencilla y predecible.

El diseño del sistema cuenta con 5 estados básicos. A continuación, una breve descripción de su comportamiento:

- No iniciado
  - Comportamiento: Es el estado inicial del sistema. Enciende la señal lumínica de verde. Gestiona la carga de parámetros de ensayo.
- Ensayo listo
  - Comportamiento: Enciende la señal lumínica en verde de a intervalos de un segundo.
- En ejecución
  - Comportamiento: enciende la señal lumínica en amarillo de a intervalos de un segundo parpadeante. Inicia o reanuda el movimiento del eje.
- Frenado
  - Comportamiento: enciende la señal lumínica de amarillo. Detiene el movimiento del eje.
- Emergencia
  - Comportamiento: enciende la señal lumínica de rojo parpadeante. Detiene el movimiento del eje.
- Finalizado
  - Comportamiento: enciende la señal lumínica de rojo. Detiene el movimiento del eje.

Estos estados solo admiten un conjunto de transiciones posibles. Este conjunto se detalla en su diagrama de transición de estados:

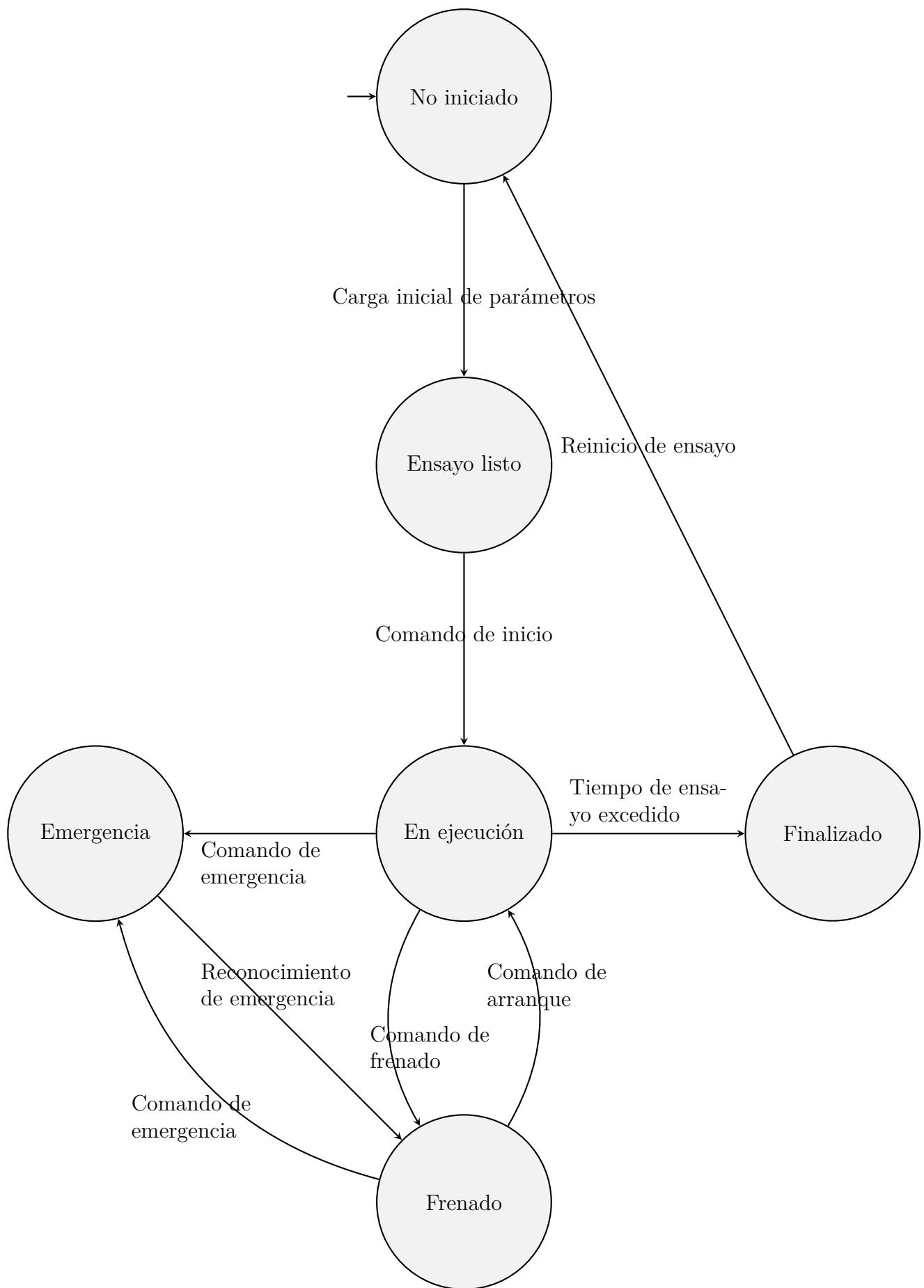


Figura 10: Diagrama de transiciones de estado implementado en el algoritmo de control de flujo

Los eventos de transición requieren de ciertas precondiciones para ser ejecutados. En la tabla 9 se describen las precondiciones de ejecución:

Evento	Precondición
Carga inicial de parámetros	Tipo de ensayo especificado
	Parámetros de ensayo recibidos
	Duración de ensayo especificada
Comando de inicio	Modulo de línea activado
	Eje habilitado
	Botón de arranque pulsado
	Botón de arranque por software pulsado
Comando de arranque	Ausencia de comando de emergencia
	Botón de arranque pulsado
	Botón de arranque por software pulsado
Comando de frenado	Ausencia de comando de emergencia
	Botón de frenado pulsado
Comando de emergencia	Botón de frenado por software pulsado
	Botón de emergencia pulsado
	Botón de emergencia por software pulsado
	Temporizador de keepalive excedido
	Desconexión de interfaz física
Reconocimiento de emergencia	Error de hardware
	Botón de reseteo pulsado
Tiempo de ensayo excedido	Ausencia de comando de emergencia
	El tiempo transcurrido en estado RUNNING excede el tiempo de ensayo prefijado
Reinicio de ensayo	Apagado de la máquina

Cuadro 9: Tabla de eventos de transición implementados en el algoritmo de control de flujo

### 8.1.2. Algoritmo de control de torque por saturación del lazo de velocidad

Dado que el dispositivo de control se encuentra diseñado para tareas de posicionamiento, su algoritmo de control de torque directo (DTC por sus siglas en inglés) no prevé la inserción adicional de consignas de cupla de manera segura. Estas se definen automáticamente en función de la consigna de velocidad o posición y otros parámetros auxiliares. Es posible abordar esta limitación del equipo utilizando una técnica llamada control de torque por saturación del lazo de velocidad.

En esta modalidad, se fijan límites para la cupla máxima y se introduce una consigna de velocidad lo suficientemente alejada del punto de operación actual,  $\omega_\infty$ . Esto hace que el control responda con torque máximo para intentar alcanzar su consigna de velocidad, lo que se conoce como saturación de lazo de control. Al poder manipular estos límites, es posible introducir una consigna de cupla de manera precisa.

Una de las principales ventajas de este método es la seguridad: la consigna de torque se aplica en simultáneo con una consigna de velocidad. Por este motivo, en caso de un error de programación, el motor no acelerará indefinidamente, sino que lo hará hasta su consigna de velocidad  $\omega_\infty$ . De esta forma, se produce una barrera de seguridad adicional que evita la fuga en velocidad del motor.

El algoritmo puede explicarse con el siguiente diagrama de flujo:

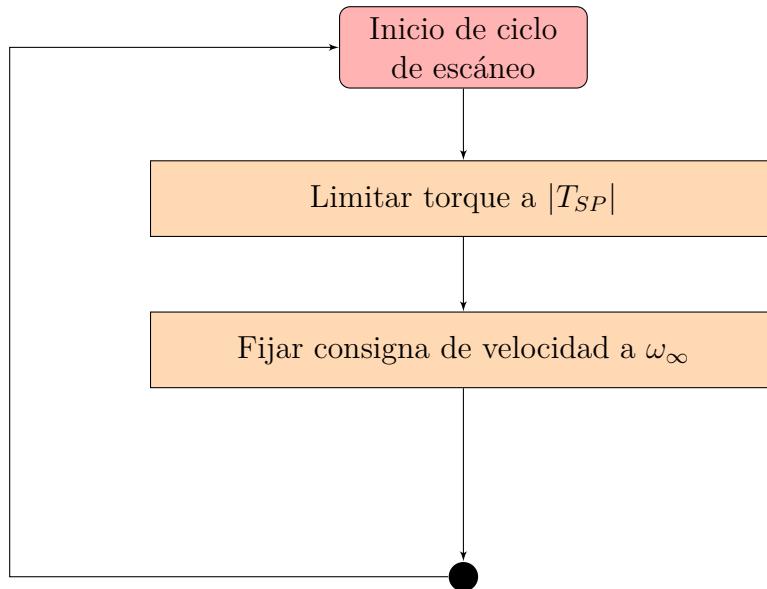


Figura 11: Diagrama de flujo para el control de torque por saturación

**Operación en los 4 cuadrantes** Observando la ecuación 11, es posible notar que el signo del torque resistente no siempre será positivo. En los casos en que el motor de ensayo se encuentre disminuyendo la velocidad, el término inercial será negativo. Lo mismo es válido para cuando se requiera realizar ensayos con torque variable en el tiempo, como los senoidales.

Por este motivo, resulta necesario adaptar el algoritmo expuesto en la figura 11 para simular inversiones de signo. Implementar esto dentro de la estrategia de control de torque por saturación no es lineal: solamente es posible limitar la cupla en módulo.

Para aplicar cuplas negativas, se debe introducir su módulo como límite de torque y su signo como el signo de consigna de velocidad, esto hará que el motor gire en dirección negativa. En el caso general, la cupla resistente debe tener el mismo signo que  $\omega_\infty$ .

A continuación se presenta un diagrama de flujo que describe el funcionamiento del control de torque en los cuatro cuadrantes:

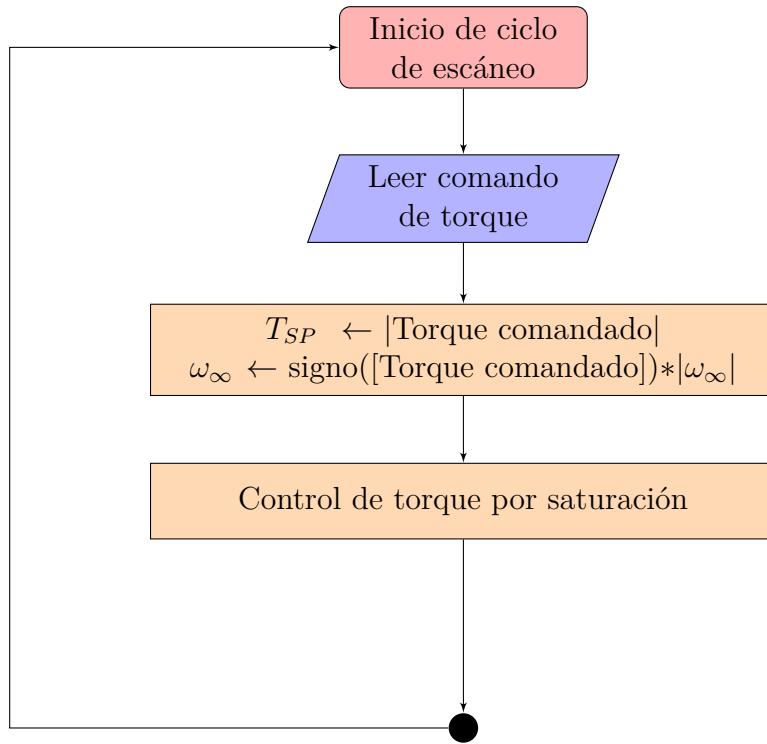


Figura 12: Diagrama de flujo para el control de inversión de torque

### 8.1.3. Algoritmo de control de torque en función de la velocidad: filtrado de variables

Considerando la ecuación 11, es posible analizar su discretización de primer orden:

$$T[k] = A + B\Omega[k] + C\Omega[k]^2 + D \frac{\Omega[k] - \Omega[k-1]}{\Delta T} \quad (12)$$

En donde  $\Delta T$  es el tiempo transcurrido entre las consignas de cupla  $T[k]$  y  $T[k-1]$ .

Suponiendo que la velocidad angular es medida con error  $\delta\omega$ , se presume que esta perturbación se observará en la cupla como un valor  $\delta t$ :

$$T[k] + \delta t_k = A + B(\Omega[k] + \delta\omega_k) + C(\Omega[k] + \delta\omega_k)^2 + D \frac{\Omega[k] + \delta\omega_k - (\Omega[k-1] + \delta\omega_{k-1})}{\Delta T} \quad (13)$$

$$\delta t_k = B\delta\omega_k + C(2\Omega[k]\delta\omega_k + \delta\omega_k^2) + D \frac{\delta\omega_k - \delta\omega_{k-1}}{\Delta T} \quad (14)$$

$$|\delta t_k| \leq B|\delta\omega_k| + C(2|\Omega[k]\delta\omega_k| + \delta\omega_k^2) + D \frac{|\delta\omega_k| + |\delta\omega_{k-1}|}{\Delta T} \quad (15)$$

Suponiendo  $\delta\omega_i = 1$  RPM,  $\Delta T \approx 20ms$  y como  $\Omega[k] < 3000$  RPM. Es posible acotar aún más la expresión:

$$|\delta t_k| \leq B + C \cdot 6001 + D \cdot 100 \quad (16)$$

Considerando el rango de valores que típicamente serán ensayados para B, C y D<sup>6</sup>, se observa que el error de medición de la velocidad angular se encuentra prácticamente dominado por el factor D.

<sup>6</sup>Para esta afirmación se espera  $B \sim 10^{-1} \frac{Nm}{RPM}$ ,  $C \sim 10^{-6} \frac{Nm}{RPM^2}$  y  $D \sim 10^{-2} \frac{Nm \cdot s}{RPM}$ .

El ruido de medición,  $\delta\omega$ , proviene de la medición del resolver del servomotor. Típicamente, el ruido de medición puede ser considerado de alta frecuencia cuando se lo compara con la velocidad angular del servomotor.

Por este motivo, para disminuir la influencia del ruido de medición, se explorarán diferentes técnicas de procesamiento de señales.

### 8.1.3.1 Filtro FIR

Aplicando conceptos clásicos de procesamiento de señales se consideró el uso de un filtro de respuesta al impulso finita (FIR, por sus siglas en inglés). Este tipo de filtros son de la forma:

$$y[n] = \sum_{i=0}^N b_i x[n-i] \quad (17)$$

En donde  $N$  es la cantidad de términos u orden del filtro,  $b_i$  es el valor de la respuesta al impulso en el instante  $i$ -ésimo,  $x[n]$  la señal de entrada e  $y[n]$  la señal de salida.

En la ecuación 17 se observa una de las desventajas del tipo promedio ponderado: al calcular la salida en el instante  $n$ , fueron necesarias las  $N - 1$  muestras anteriores. Es decir que el filtro tendrá un retardo de  $N - 1$  muestras. Suponiendo un filtro de 100 muestras y un periodo de muestreo de 15ms, consistente con el ciclo de escaneo del diagrama LADDER, entonces el retardo sería de 1,5s.

Sin embargo, también es posible notar que, como la salida del filtro es una combinación lineal de su entrada, que el filtro FIR mantiene su salida acotada siempre y cuando su entrada sea acotada (BIBO, por sus siglas en inglés). Esto se vuelve evidente al acotar la salida del filtro:

$$y[n] \leq \left| \sum_{i=0}^N b_i x[n-i] \right| \leq (N + 1) \cdot |\text{Máx}(b_i x[n])| \quad (18)$$

Usualmente el retardo podría corregirse en el post-procesamiento de la señal, pero, para aplicaciones en tiempo real, esto no es posible. Por este motivo, resulta importante reducir al mínimo la cantidad de muestras tomadas a la hora de diseñar el filtro.

Experimentalmente se observó que, para lograr un filtro FIR pasabajos, es necesario incorporar un gran número de coeficientes. Esto incrementa el retardo a niveles inaceptables. Para ilustrar este efecto, se simuló la respuesta al escalón de un filtro con frecuencia de corte 0,25hz y ancho de banda de transición de 0,25hz.

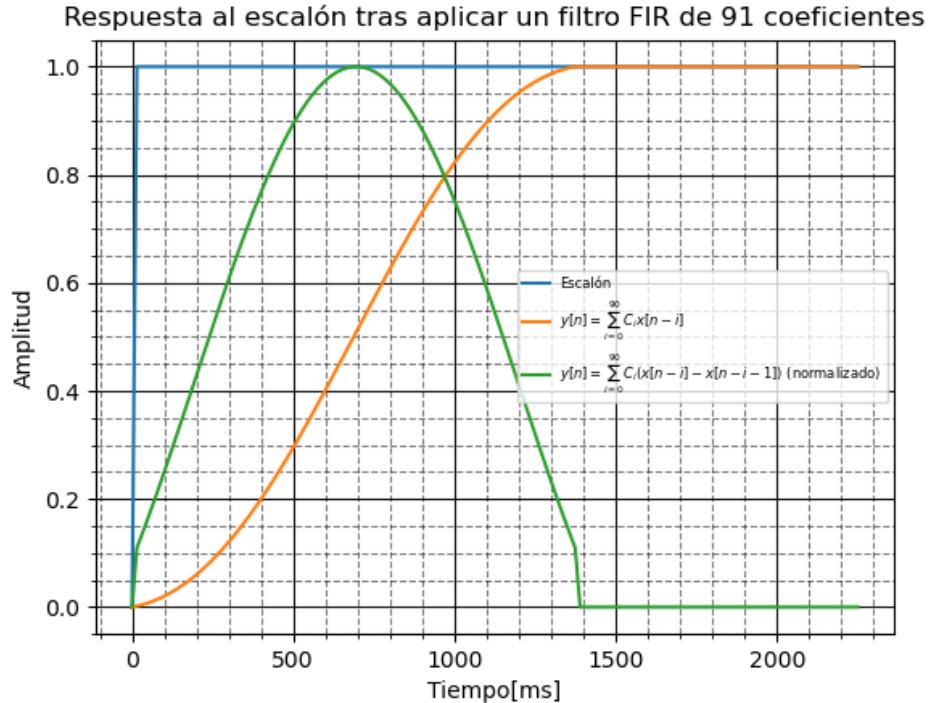


Figura 13: Respuesta al escalón del filtro de 91 coeficientes: en naranja la respuesta del filtro, en verde la respuesta del filtro tras aplicar el operador derivado y en azul la señal de excitación

En la figura se observa que, al aplicar el filtro sobre el operador derivativo, se retrasa su pico del término derivativo por aproximadamente 700ms. Aplicado para el término inercial de la ecuación cupla, se consideró que este filtro FIR produce un retardo inaceptable para su aplicación.

#### 8.1.3.2 Filtro IIR

Los filtros de respuesta infinita (IIR, por sus siglas en inglés) presentan la siguiente estructura:

$$y[n] = \sum_{i=0}^P a_i x[n-i] + \sum_{i=1}^Q b_i y[n-i] \quad (19)$$

En la ecuación puede observarse nuevamente el retardo en la salida pero también la característica recursiva del filtro. Además de ello se puede observar que el filtro no necesariamente será estable, basta con considerar el siguiente caso:

$$P = 0; \quad Q = 1; \quad a_0 = 1; \quad b_1 = 1; \quad x[i] = 1 \quad \forall i \in \mathbb{N}; \quad y[i] = 0 \quad \forall i \leq 0 \quad (20)$$

$$n = 0 \implies y[0] = 1 \cdot 1 + 1 \cdot 0 = 1 \quad (21)$$

$$n = 1 \implies y[1] = 1 \cdot 1 + 1 \cdot 1 = 2 \quad (22)$$

$$n = 2 \implies y[2] = 1 \cdot 1 + 1 \cdot 2 = 3 \quad (23)$$

$$(24)$$

Esto implica que el filtro no es BIBO estable. Es posible demostrar que los filtros IIR pueden alcanzar el mismo conjunto de especificaciones que su análogo FIR con un número menor de coeficientes.

Con el fin de utilizar la menor cantidad de términos posible, se utilizará un filtro IIR de orden 1.

$$y[n] = a_0x[n] + b_1y[n - 1] \quad (25)$$

Típicamente para filtros pasa-bajos se suele escoger  $b_1 = 1 - a_0$  y  $0 < a_0 \leq 1$  para asegurar la estabilidad del filtro.  $b_1$  se suele llamar constante de decaimiento y se vincula con la frecuencia de corte con la siguiente relación  $b_1 = \cos(\frac{\omega}{\Omega_s}) - \sqrt{2 - 2\cos(\frac{\omega}{\Omega_s}) + (1 - \cos(\frac{\omega}{\Omega_s}))^2}$ . Considerando una frecuencia de muestreo de aproximadamente 33HZ y una frecuencia de corte de 0.5Hz, la ecuación arroja una constante de decaimiento  $b_1 = 0,98$ . Este filtro tendrá un tiempo de caída de alrededor de 1,6 segundos.

Si bien es posible seleccionar otras constantes temporales que permitan una respuesta más rápida del filtro, se optó por un valor conservador con el fin de garantizar la estabilidad del mismo para su uso general. En la figura 19 se observa la respuesta al escalón del filtro:

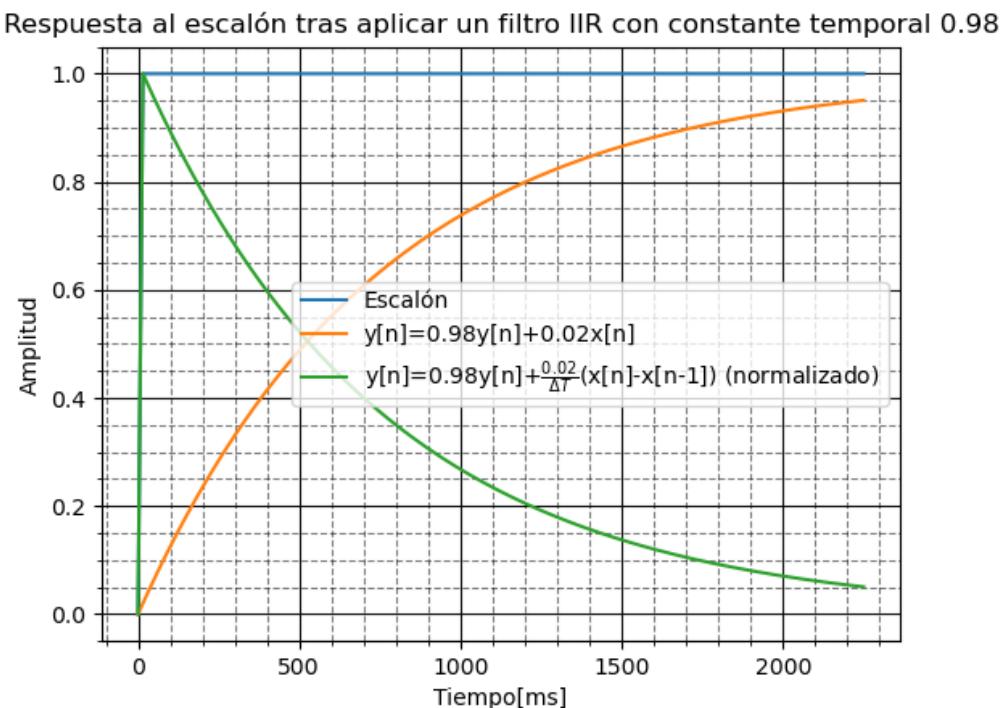


Figura 14: Respuesta al escalón filtro IIR: en naranja la respuesta del filtro, en verde la respuesta del filtro tras aplicar el operador derivativo y en azul la señal de excitación

La respuesta al escalón presenta un tiempo de caída de alrededor de 1,6 segundos. El retardo de la acción, cuando el filtro es aplicado al operador derivativo, es de aproximadamente 30 milisegundos. Se consideró que este comportamiento es adecuado para simular la inercia de un cuerpo.

#### 8.1.4. Algoritmo de control de torque en función del tiempo: latencia de datos

Como se menciona en la tabla 8, uno de los requisitos de diseño es permitir la introducción de comandos de torque con una resolución temporal de al menos 100ms. Las pruebas en campo indican una latencia en la comunicación OPC de aproximadamente de 100ms. Este retardo no es constante y puede oscilar entre los 50ms y 150ms.

Si se introdujeseen consignas de cupla de a una a la vez y estas estuviesen separadas cada 100ms, no sería posible asegurar que el tiempo de aplicación de cupla sea exactamente los

100ms requeridos. Para solventar ello, necesario implementar algún tipo de carga previa de pares cupla-tiempo.

La manera más sencilla es realizar una carga completa de los pares torque-tiempo previo a iniciar el ensayo. Luego, en tiempo de ejecución del ensayo, introducir los comandos en el instante requerido.

Sin embargo, como se mencionó en la sección 5 un ensayo podría durar potencialmente días. El almacenamiento previo de estos pares torque-tiempo obligaría a almacenar grandes volúmenes de datos en el control -excediendo su capacidad de memoria- o a utilizar técnicas de memoria dinámica, limitando su confiabilidad<sup>7</sup>.

Lo expuesto obliga a la implementación de un sistema de memorias intermedias o buffer en donde los pares torque-tiempo serán introducidos durante la ejecución del ensayo. En este esquema, se utilizan dos tipos de buffer: el primero es un buffer de comunicación, visible para la cliente OPC XML-DA, mientras que el segundo es un buffer interno.

En el buffer de comunicación se carga un número fijo de pares torque-tiempo desde la computadora. Esta operación tiene una latencia proporcional al tamaño del buffer de comunicación.

Una vez que el buffer comunicación está lleno, se cargan los valores a un buffer interno. Esta operación sucede dentro del PLC y tiene una latencia mucho menor que la de la comunicación OPC.

Una vez que se transfirieron los datos, el buffer de comunicación informa la disponibilidad para recibir un nuevo lote de datos activando una bandera de libre para recibir. Mientras tanto, el buffer interno alimenta al algoritmo de control de torque con los pares de valores recibidos.

El buffer interno se lee circularmente, es decir que funcionará como una memoria FIFO de ancho fijo. A medida que el algoritmo de control de torque vaya recibiendo datos, estos se restituirán con un dato recibido del buffer de comunicación.

El esquema de cascada de buffers puede visualizarse en la siguiente figura:

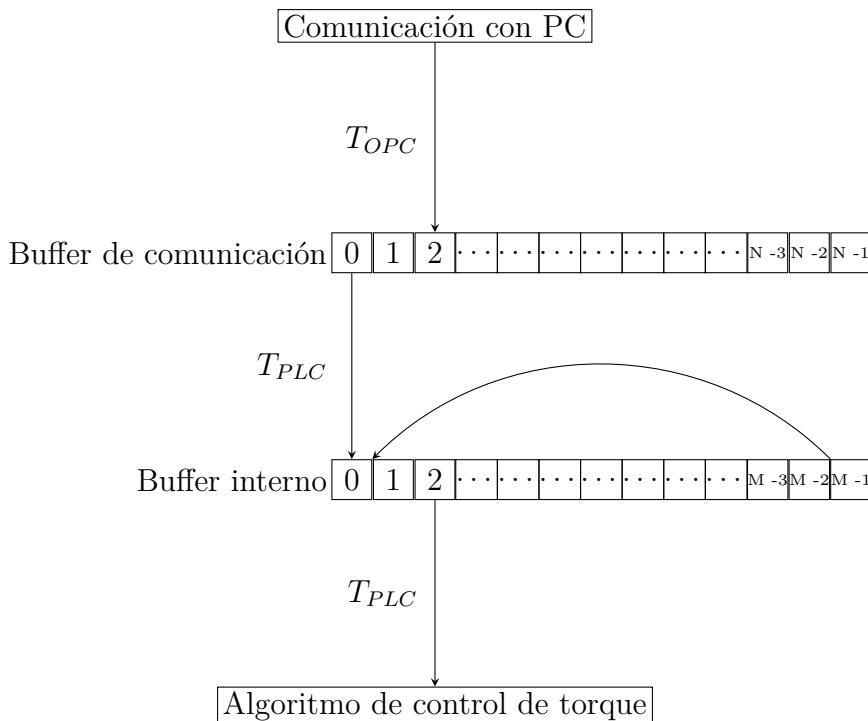


Figura 15: Diagrama de buffers

<sup>7</sup>Si bien la utilización de memoria dinámica no es una práctica inherentemente negativa, en entornos de automatización industrial se la suele evitar ya que la asignación de memoria ocurre en tiempo de ejecución. Esto hace que dependa de los recursos disponibles del sistema en el instante de asignación. En caso de que no haya memoria disponible, el programa fallará en tiempo de ejecución en un instante prácticamente aleatorio

Para poder definir el tamaño de los buffer de comunicación es necesario considerar que la transmisión de datos al buffer de comunicación ocurrirá en un tiempo  $T_{OPC}$ .

Por otro lado, el buffer interno recibe un par cupla-tiempo del buffer de comunicación en un tiempo  $T_{PLC}$ . Como el buffer interno se carga circularmente, tendrá  $M - 1$  comandos de torque disponibles para su ejecución mientras que el buffer de comunicación no se haya agotado.

El buffer de comunicación se agotará al haber transmitido  $N - 1$  comandos al buffer interno. Cuando se agota el buffer de comunicación, el buffer interno deberá poder sostener la operación del algoritmo de torque durante  $T_{OPC}$  segundos.

Durante ese período se ejecutarán -como máximo-  $\frac{T_{OPC}}{\Delta T_{min}}$  comandos de torque, en donde  $\Delta T_{min}$  es el tiempo mínimo entre comandos de cupla-tiempo. Estas instrucciones serán enviadas desde el buffer de comunicación al buffer interno en un tiempo  $\frac{T_{OPC}}{\Delta T_{min}} T_{PLC}$ . Es decir, que el buffer interno deberá tener al menos  $\frac{T_{OPC}}{\Delta T_{min}} + \frac{T_{OPC} T_{PLC}}{\Delta T_{min}^2}$  espacios para almacenar instrucciones cuando el buffer de comunicación se haya agotado.

Por su parte, el buffer de comunicación se agotará en un tiempo  $(N - 1)\Delta T_{min}$  y tardará  $T_{OPC}$  en reabastecerse. Una vez reabastecido, deberá restituir los  $\frac{T_{OPC}}{\Delta T_{min}} + \frac{T_{OPC} T_{PLC}}{\Delta T_{min}^2}$  comandos ejecutados por el buffer interno. Es decir que, luego de reabastecerse y restituir el buffer interno, el buffer de comunicación contará con  $N - 1 - \frac{T_{OPC}}{\Delta T_{min}} - \frac{T_{OPC} T_{PLC}}{\Delta T_{min}^2}$  comandos.

$$\begin{aligned} \frac{T_{OPC}}{\Delta T_{min}} + \frac{T_{OPC} T_{PLC}}{\Delta T_{min}^2} &< (M - 1) && \text{Condición para sostener la operación} \\ \frac{T_{OPC}}{\Delta T_{min}} + \frac{T_{OPC} T_{PLC}}{\Delta T_{min}^2} &< (N - 1) && \text{Condición para restituir el buffer interno} \end{aligned} \quad (26)$$

Se fijan cotas máximas para los parámetros según la siguiente tabla:

Parámetro	Valor
$T_{OPC}$	500ms
$T_{PLC}$	20ms
$\Delta T_{min}$	100ms

Cuadro 10: Valores utilizados para el tiempo de transmisión entre módulos ( $T_{OPC}$  y  $T_{PLC}$ ) y tiempo entre comandos de cupla ( $\Delta T_{min}$ )<sup>8</sup>

Luego, las inecuaciones de la ecuación 27 pueden ser evaluadas:

$$\begin{aligned} 12 < M && \text{Condición para sostener la operación} \\ 12 < N && \text{Condición para restituir el buffer interno} \end{aligned} \quad (27)$$

Eligiendo tamaños de buffer  $N = 1024$  y  $M = 4096$ , se satisfacen ambas ecuaciones. Si bien no es una condición necesaria, se buscará que  $N$  tenga un valor suficientemente grande como para reducir el tráfico en la red Ethernet. Es conveniente que  $M < N$  para que durante la carga inicial se abastezcan  $N+M$  valores y luego, en las cargas que se realizan en tiempo de ejecución del ensayo, solamente de a  $N$  valores.

<sup>8</sup>Los valores de la tabla se obtuvieron experimentalmente.  $\Delta T_{min}$  fue definido como requisito de diseño (F.CT.1). En el caso de  $T_{OPC}$ , este es el tiempo aproximado para enviar -y procesar- un bloque de información de 1500bytes, que equivale a la máxima unidad de transmisión (MTU, por sus siglas en inglés). En función del tamaño de los buffers este tiempo aumenta ya que se deben enviar más paquetes. A modo ejemplo, para un buffer de torque tiempo de 1024 espacios se deben enviar 1024 tiempos de aplicación de 32 bits y otras 1024 consignas de cupla de 32 bits, totalizando 64 Kbytes de datos. El tiempo de transmisión es de aproximadamente 60ms, pero considerando el tráfico de la red, posibles retransmisiones de datos y el procesamiento en el PLC de cada bloque recibido se tomará como cota superior 1000ms.

### 8.1.4.1 Algoritmo de control de torque en función del tiempo: sincronismo de comandos

Para poder ilustrar el funcionamiento del banco de ensayos en modo cupla-tiempo, se utilizará como caso base un ejemplo.

Suponiendo que el banco de ensayos reciba el siguiente conjunto de pares torque-tiempo:

Tiempo	Cupla
$t_0$	a
$t_1$	b
$t_2$	c

Cuadro 11: Ejemplo de entrada para el banco de ensayos en modo cupla-tiempo

Al recibir el comando de cupla  $a$ , el banco de ensayos deberá sostener la cupla en este valor durante un tiempo  $t_1 - t_0$ . En términos de la solución adoptada, se debe alimentar el algoritmo de control de torque con una consigna  $a$ . Al transcurrir el tiempo  $t_1 - t_0$ , se debe introducir la consigna  $b$  durante un tiempo  $t_2 - t_1$ . Este esquema avanza hasta terminar de aplicar todos los pares de cupla-tiempo.

Un primer acercamiento a la implementación de este algoritmo podría realizarse con un temporizador auto-reseteable con consigna temporal  $t_{i+1} - t_i$ . A grandes rasgos, la implementación seguiría el siguiente diagrama en bloques:

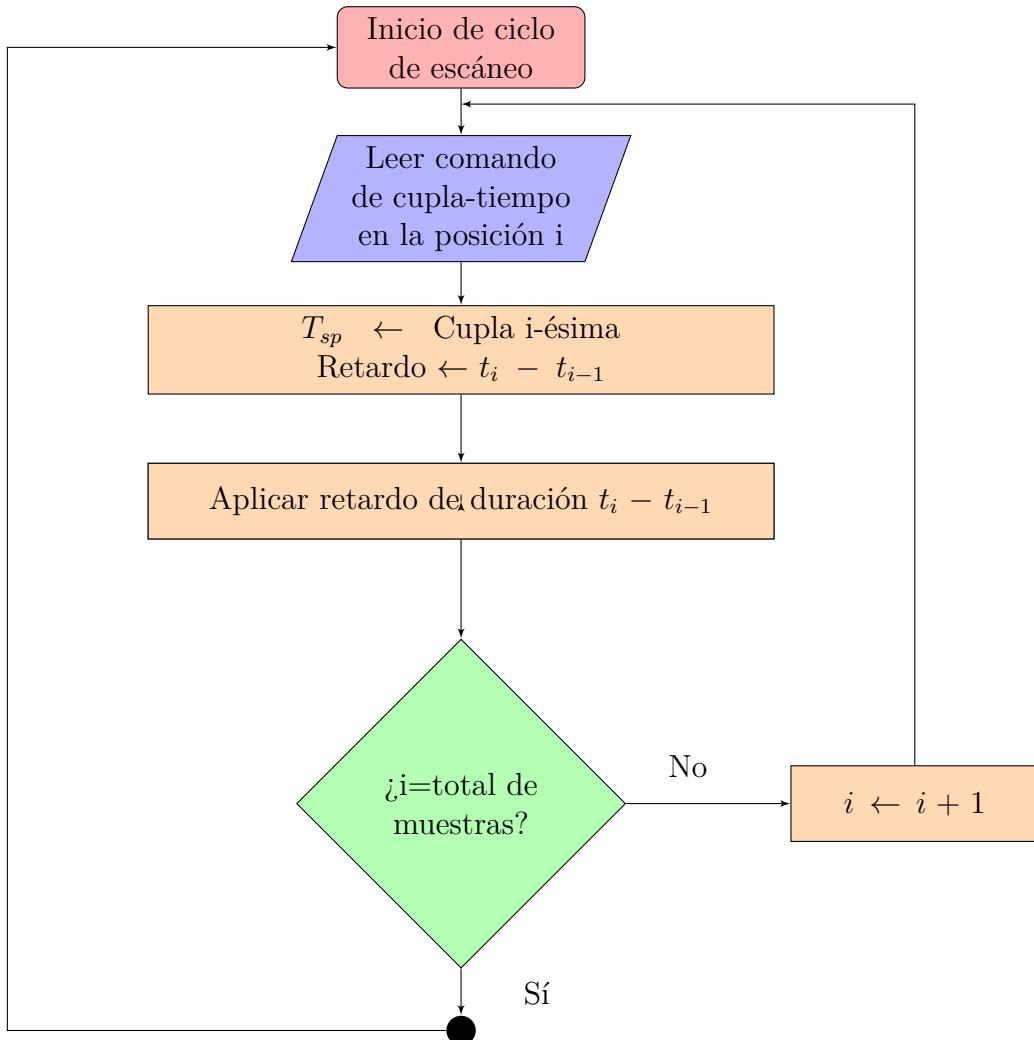


Figura 16: Diagrama en bloques del algoritmo con temporizadores autoreseteables

Sin embargo, durante la implementación de la lógica se revela la inestabilidad de los algoritmos de tipo temporizador auto-resetable. Cuando el algoritmo corre en el nivel de ejecución BackgroundTask, el tiempo de retardo puede variar en función del período de escaneo del diagrama LADDER. Esto produce que el período fijado en el temporizador sea algo mayor al esperado.

Es posible observar el efecto considerando una perturbación siempre positiva,  $\delta t$ , y un retardo homogéneo,  $\Delta T$ . La primera muestra se sostendrá durante  $\delta t + \Delta T$  segundos, esto hará que la segunda muestra tenga un desfasaje de  $\delta t$  segundos respecto al tiempo de aplicación original y, también afectada por la perturbación  $\delta t$ , se aplique durante  $\delta t + \Delta T$  segundos, provocando a su vez un desfasaje de  $2\delta t$  segundos para la aplicación de la tercera muestra. Extrapolando este razonamiento, la muestra  $i$  tendrá un desfasaje de  $(i - 1)\delta t$  respecto al tiempo de aplicación.

Un enfoque posible es corregir la perturbación a lazo abierto, estimando el valor de  $\delta t$  y restándolo del período del temporizador. Sin embargo  $\delta t$  no es constante, sino que se define en tiempo de ejecución. Además, por más adecuada que sea la estimación, el desfasaje escala según la cantidad de muestras en el vector de torque-tiempo. Esto asegura que, por más pequeño que sea, el error eventualmente sea propagado a niveles inaceptables.

La corrección a lazo cerrado, es decir, midiendo el tiempo efectivo que se realiza el retardo para hacer la corrección, también viene acompañada de la misma inexactitud del temporizador.

La solución encontrada para poder implementar el sincronismo de comandos fue la utilización de interrupciones cíclicas, que se ejecutan en el nivel de ejecución TimerTask. A diferencia

del método descrito con anterioridad, la interrupción no es autoreseteable, sino que se dispara en tiempo fijo, y la dispersión en tiempo que presenta es de media nula. De esta forma, el desfasaje que se produzca no escala con el número de muestra, evitando la propagación del desfasaje en el algoritmo.

Sin embargo, el sistema de accionamientos SIMOTION D425 no permite la configuración de estas interrupciones con período arbitrario y variable. Resulta necesario aceptar un período fijo y predefinido.

Según los requisitos de diseño, la tasa de muestreo entre los diferentes pares de torque-tiempo puede ser variable entre ensayos y no homogénea dentro del mismo vector (Requisitos F.CT.0 y F.CT.2 de la tabla 8). Esto obliga a diseñar un nuevo algoritmo que permita tasas de muestreo arbitrarias utilizando interrupciones cíclicas con período fijo.

El algoritmo diseñado sigue el siguiente diagrama de flujo:

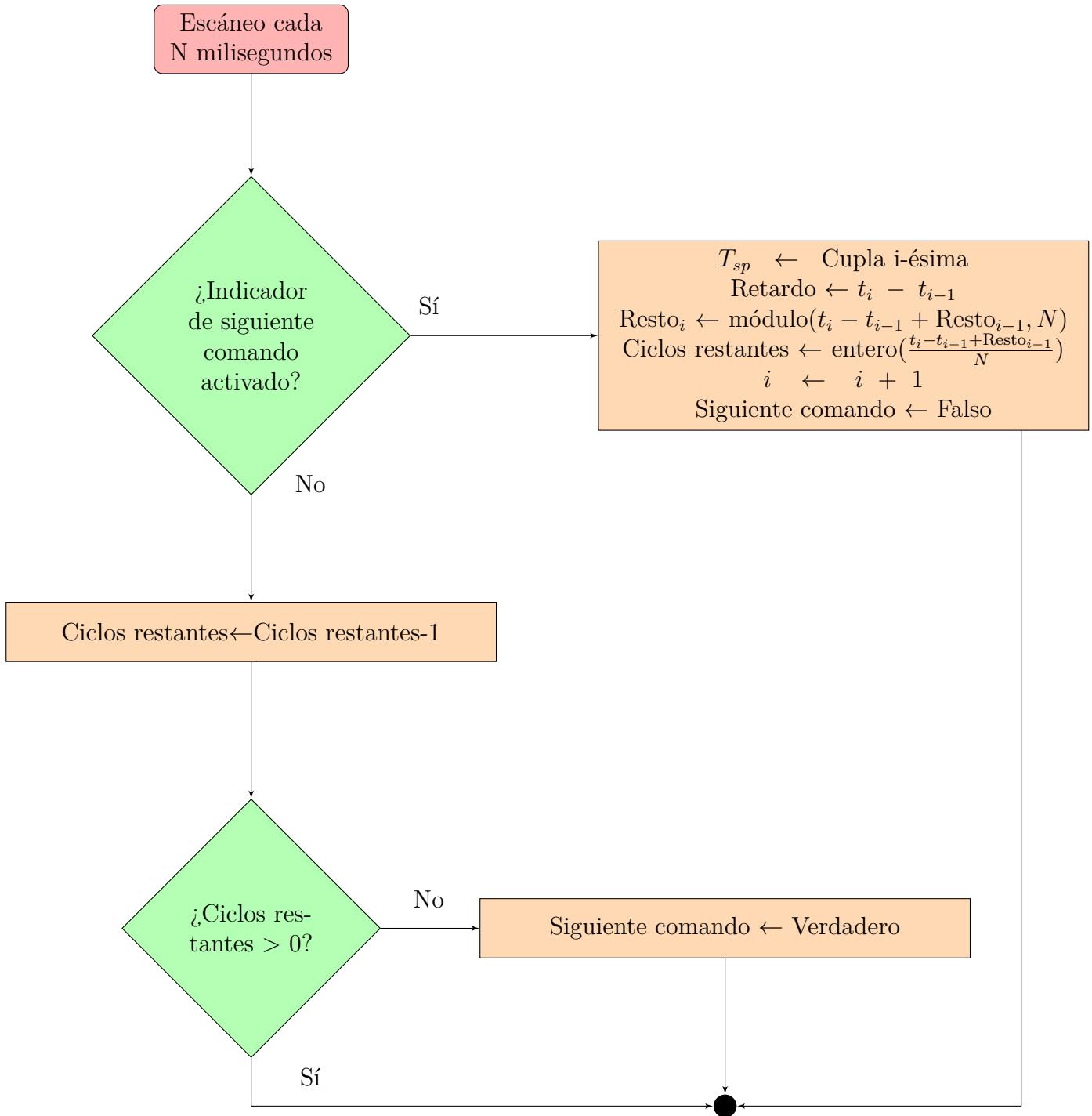


Figura 17: Diagrama de flujo de la implementación del control de sincronismo

Bajo este nuevo esquema es posible compensar el error introducido si el tiempo entre muestras y el periodo de interrupción no son múltiplos enteros. A modo de ejemplo: si  $N$  fuese 15ms y el tiempo entre muestras fuese 99ms entonces se producirá un retardo de 6 ciclos de interrupción y se reservarán 9ms como remanente para el ciclo siguiente. El ciclo siguiente se calculará un retardo de 108ms (99ms+9ms), lo que significarán 7 ciclos de interrupción y un remanente de 3ms.

Este algoritmo asegura que cada consigna de cupla se ejecutará con un error de duración menor a  $N$ . Además, la duración total del ensayo también tendrá un error menor a  $N$ .

### 8.1.5. Mecanismo de keepalive

Para poder detectar la desconexión del cliente OPC XML-DA del lado del dispositivo SIMOTION, se ideó un mecanismo fuertemente inspirado en el protocolo TCP. El objetivo es poder detectar la ausencia del cliente OPC XML-DA desde el servidor OPC XML-DA.

El mecanismo ideado simplemente consiste en un temporizador del lado del control que, cada cierta cantidad de tiempo, fija una variable 'keepalive' a su valor lógico verdadero. Se espera que, del lado de la computadora, el valor de la variable sea negado. Si, al momento de fijar la variable a verdadero, se detecta que el valor de la variable no fue negado, entonces la comunicación con la interfaz gráfica falló.

De esta forma, el dispositivo podrá ponerse en estado de emergencia si existe una falla de interfaz gráfica. Las posibles fallas podrían ser: agotamiento de batería al utilizar una computadora portátil, falta de respuesta por uso excesivo de los recursos de la computadora o desconexión de la interfaz Ethernet, entre otras.

## 8.2. Interfaz física del banco de ensayo

La interfaz física diseñada puede distinguirse funcionalmente entre entradas y salidas.

Las entradas son todas las señales que permiten que el usuario introduzca comandos al sistema. En la interfaz física, las entradas están reservadas para operaciones críticas, como la detención de emergencia, el inicio, la interrupción de ensayo y el reconocimiento de fallas.

Por su parte, las salidas de la interfaz consisten en tres señales lumínicas con los colores rojo, amarillo y verde. Estos colores están ligados a los estados de frenado, de ejecución de ensayo y de listo para ensayar respectivamente.

Si bien se podría desarrollar una única interfaz equivalente en el software, el desarrollo de la interfaz física fue motivado por la seguridad de operación. En el caso en que el operador quede atrapado en el eje, podría detener la máquina rápidamente presionando el freno de emergencia. Además de ello, al dar inicio a un ensayo no deberá retirar la mirada del equipo, lo que evitaría accidentes.

En caso de que un ensayo requiriese velocidad nula por algún periodo de tiempo, el operador podría percibir erróneamente que el ensayo se detuvo anticipadamente e intentar interactuar con el eje. Al agregar alertas lumínicas, el operador puede conocer rápidamente en qué etapa se encuentra el ensayo y bajo qué condiciones el motor podría ser susceptible al movimiento.

Luz encendida	Característica	Indicación
Roja	Fija	El ensayo ha finalizado.
Roja	Parpadeante	El sistema está en estado de emergencia.
Amarilla	Fija	El ensayo está pausado.
Amarilla	Parpadeante	El ensayo se está ejecutando.
Verde	Fija	Inicio del programa, aún no se recibieron los parámetros de ensayo.
Verde	Parpadeante	El ensayo está listo para el arranque.

Cuadro 12: Descripción de las señales lumínicas

### 8.2.1. Componentes de la interfaz física

- Fuente de alimentación externa: Para evitar el daño de la fuente de alimentación que utiliza el sistema SIMOTION D425, se alimenta la interfaz física con una fuente externa.
- Pulsador de arranque: Se utiliza para iniciar o reanudar el movimiento del eje simulador.

- Pulsador de frenado: Se utiliza para frenar el movimiento del eje simulador.
- Botón de freno de emergencia: Cableado con lógica negada, en caso de una desconexión o falla en la fuente de alimentación del circuito control se acciona el freno de emergencia.
- Pulsador de reseteo: Se utiliza para liberar la emergencia del equipo.
- Semáforo indicador de estado: El semáforo enciende su luz verde cuando se encuentra en condiciones de dar inicio al ensayo, la luz amarilla indica parpadeante indica que el ensayo está en curso y finalmente la luz roja indica una frenada de emergencia.



(a) Botonera del equipo



(b) Semáforo de señales lumínicas

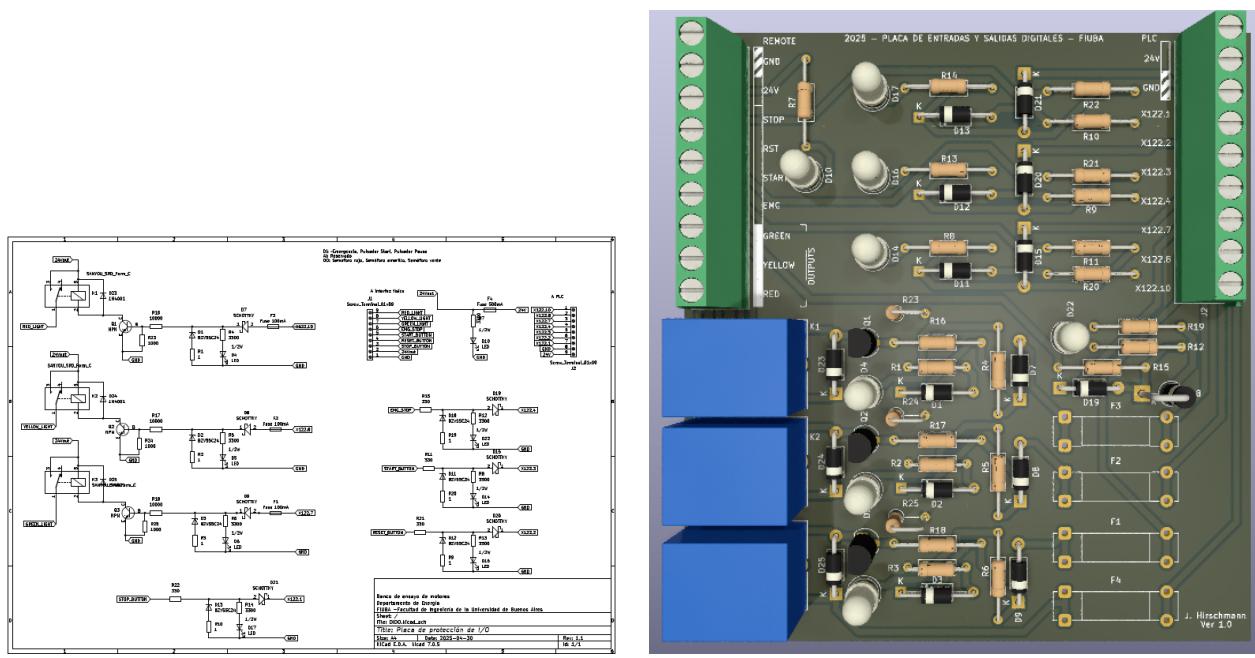
Figura 18: Elementos de la interfaz física del banco.<sup>9</sup>

### 8.2.2. Circuito eléctrico

En la placa diseñada se distinguen 3 canales de salida y otros 4 de entrada. En los canales de salida se implementaron protecciones en caso exceso de corriente, de conexión inversa y LEDs testigo para indicar el estado de la salida. En el caso de los canales de entrada, se implementaron protecciones de conexión inversa y de exceso de tensión.

A continuación se presentan los diagramas esquemáticos y de diseño de la placa electrónica:

<sup>9</sup>Ambas imágenes fueron alteradas digitalmente con fines demostrativos.



(a) Circuito esquemático

(b) Plano de placa electrónica

Figura 19: Diagramas de diseño de la placa en la interfaz física.

### 8.2.2.1 Plano eléctrico

En el anexo a este documento se adjunta una copia del plano eléctrico de la máquina completa como parte de la documentación del equipo.

## 8.3. Interfaz por software del banco de ensayos

La interfaz por software consiste en un programa que pueda presentar una ventana con la que el usuario interactuará. Los datos ingresados por el usuario deberán verse reflejados como parámetros del ensayo. Además, la interfaz proveerá gráficos en tiempo real y permitirá el almacenamiento de las variables estudiadas durante el ensayo.

### 8.3.1. Arquitectura Modelo-Vista-Controlador

Para asegurar la escalabilidad y mantenibilidad del desarrollo de software fue necesario considerar arquitecturas estandarizadas para su desarrollo. Por este motivo, se optó diseñar la interfaz de software utilizando el patrón MVC (Modelo, vista, controlador).

Este tipo de arquitecturas es una elección típica en el diseño de interfaces gráficas. Consiste en dividir la lógica de la aplicación entre tres unidades funcionales interconectadas entre sí.

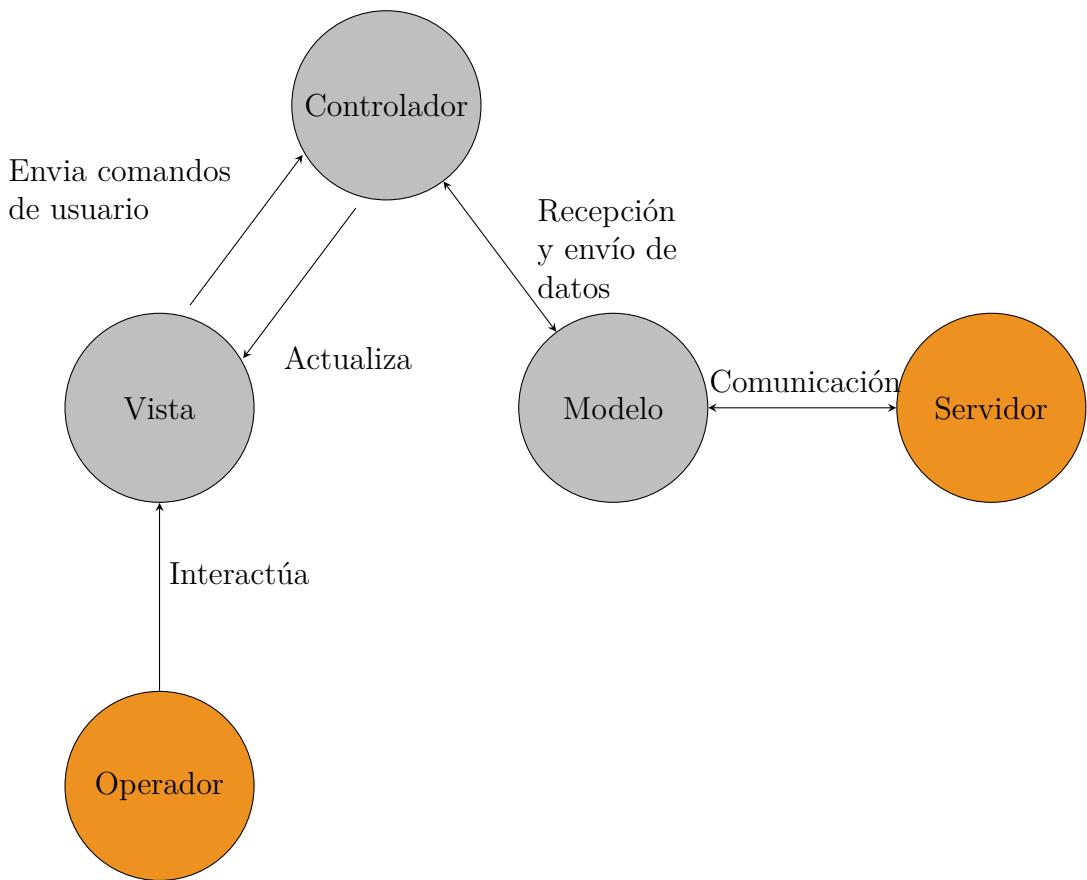


Figura 20: Esquema del patrón de diseño MVC

### 8.3.1.1 Modelo

El modelo tiene como función el control de datos y la implementación de la lógica de transmisión. En este caso, el modelo se encarga de la lectura y escritura de datos en el servidor OPC. Implementa los métodos de la especificación de OPC XML-DA descritos en la sección 3.2 y algunos métodos derivados específicos para esta aplicación.

### 8.3.1.2 Vista

La vista se encarga de la interacción con el usuario. Se ocupa de la presentación de la información obtenida mediante el modelo y de la recolección de comandos del usuario.

La implementación de la vista se realizó utilizando la biblioteca Swing de JAVA. La biblioteca permite implementar interfaces gráficas independientes de la plataforma donde se esté ejecutando el programa. La primera versión de Swing fue lanzada en abril de 1997, esto la vuelve una biblioteca madura y ampliamente divulgada entre los programadores de interfaces gráficas.

### 8.3.1.3 Controlador

El controlador enruta comandos de la vista hacia el modelo y envía datos del modelo a la vista. Su funcionamiento es mediando el intercambio de información entre ambos módulos. En esta implementación, también tiene como tarea compatibilizar los comandos recibidos por la vista y los enviados al modelo.

A modo de ejemplo, para seleccionar un ensayo de cupla en función del tiempo. El usuario carga en la vista un archivo en formato CSV con los pares de cupla-tiempo a simular. La vista

envía esta información al controlador, quien los entrega al modelo. El modelo se encarga de la lógica de transmisión hacia el controlador descrita en la sección 8.1.4. De manera recíproca, el controlador también agrupa las mediciones recibidas del modelo y las organiza en un buffer, que luego es leído periódicamente por la vista y presentado en un gráfico en tiempo real.

### 8.3.2. Fuentes de variables

Para la interacción entre el modelo y el servidor OPC XML-DA se definió un conjunto de variables a las cuales la interfaz por software accederá para interactuar con el programa PLC. En la figura 9 las variables que se enumeran a continuación llevan la leyenda 'etiquetas OPC' ya que, además de formar parte del programa PLC, también tienen una dirección en el servidor OPC-XML DA.

Tipo	Dirección	Nombre de variable	Función
Encendido	\VAR	modeOfOperation	Texto, escritura y lectura. Indica el modo de operación del CPU (usualmente _RUN o _STOP). En modo _RUN, el CPU está ejecutando el programa de usuario.
	\unit\LADDER	TEST_DATA	Estructura, escritura y lectura. Agrupa mediciones, señales externas e información del estado del ensayo
Estado	TEST_DATA	CURRENT_ERROR	Enumeración, sólo lectura. Informa el error actual del ensayo.
	TEST_DATA	CURRENT_STATE	Enumeración, sólo lectura. Informa el estado actual del ensayo.
	TEST_DATA	EXTERNAL_SIGNALS	Estructura, escritura y lectura. Agrupa señales para la interacción con el control.
	EXTERNAL_SIGNALS	SIM_CLEAR_TO_RECEIVE	Booleano, solo lectura. Señal de habilitación para recibir nuevos pares de torque-tiempo, indica que el buffer de comunicación en el PLC ya fue agotado.
	EXTERNAL_SIGNALS	DUT_CLEAR_TO_RECEIVE	Booleano, solo lectura. Señal de habilitación para recibir nuevos pares de torque-tiempo, indica que el buffer de comunicación en el PLC ya fue agotado. (Sólo en modo autocontenido)
Control de flujo	EXTERNAL_SIGNALS	SIM_SAVE_TO_BUFFER	Booleano, solo escritura. Indica al control que el buffer de comunicación recibió datos nuevos.
	EXTERNAL_SIGNALS	DUT_SAVE_TO_BUFFER	Booleano, solo escritura. Indica al control que el buffer de comunicación recibió datos nuevos. (Sólo en modo autocontenido)
	EXTERNAL_SIGNALS	ENABLE_LINEFEED	Booleano, solo escritura. Enciende el módulo de línea.
	EXTERNAL_SIGNALS	ENABLE_SIMULATOR_AXIS	Booleano, solo escritura. Permite la habilitación del eje simulador para recibir comandos, supone el encendido del módulo de línea.
	EXTERNAL_SIGNALS	ENABLE_TRACTION_AXIS	Booleano, solo escritura. Permite la habilitación del eje para recibir comandos, supone el encendido del módulo de línea. (Sólo en modo autocontenido)

Control de flujo	EXTERNAL_SIGNALS	START_BUTTON	Booleano, solo escritura. Permite iniciar un ensayo.
	EXTERNAL_SIGNALS	STOP_BUTTON	Booleano, solo escritura. Permite detener un ensayo para volver a ser reanudado.
	EXTERNAL_SIGNALS	EMERGENCY_BUTTON	Booleano, solo escritura. Cuando se activa la señal se dispara el freno de emergencia.
	\EXTERNAL_SIGNALS	KEEPALIVE	Booleano, lectura y escritura. Señal de verificación de enlace con el control.
	\EXTERNAL_SIGNALS	TORQUE_VS_TIME_BUTTON	Booleano, solo escritura. Señal de selección de tipo de ensayo, selecciona ensayo de torque-tiempo.
	\EXTERNAL_SIGNALS	TORQUE_VS_SPEED_BUTTON	Booleano, solo escritura. Señal de selección de tipo de ensayo, selecciona ensayo de torque-velocidad.
Configuración de ensayos	TEST_DATA-TORQUE-EQUATION	A	Flotante, solo escritura. Corresponde al término constante de la ecuación de cupla-velocidad.
	TEST_DATA-TORQUE-EQUATION	B	Flotante, solo escritura. Corresponde al término lineal de la ecuación de cupla-velocidad.
	TEST_DATA-TORQUE-EQUATION	C	Flotante, solo escritura. Corresponde al término cuadrático de la ecuación de cupla-velocidad.
	TEST_DATA-TORQUE-EQUATION	D	Flotante, solo escritura. Corresponde al término diferencial de la ecuación de cupla-velocidad.
	\GLOB	TORQUE_TIME_VALUES	Vector de flotantes, solo escritura, tamaño 1024 elementos. Es el canal de valores de torque en el buffer de comunicación.
	\GLOB	TIMESTAMP	Vector de flotantes, solo escritura, tamaño 1024 elementos. Es el canal de valores de tiempo en el buffer de comunicación, supone tiempo creciente.
	\GLOB	TEST_RUNTIME	Entero, solo escritura. Tiempo de ejecución previsto del ensayo en milisegundos, es informado por la interfaz gráfica al PLC.
	TEST_DATA	SIMULATOR_AXIS	Estructura, lectura. Agrupa variables relacionadas al eje de simulacion.
	TEST_DATA	TRACTION_AXIS	Estructura, lectura. Agrupa variables relacionadas al eje de traccion.

Adquisición de datos	SIMULATOR- _AXIS TRACTION- _AXIS	SPEED	Flotante, solo lectura. Es el valor de velocidad del eje en RPM.
	SIMULATOR- _AXIS TRACTION- _AXIS	TORQUE	Flotante, solo lectura. Es el valor de cupla electromagnética ejercida por el eje simulador en Nm.
	SIMULATOR- _AXIS TRACTION- _AXIS	VOLTAGE	Flotante, solo lectura. Es la tensión en $V_{RMS}$ sobre el servomotor del eje simulador.
	SIMULATOR- _AXIS TRACTION- _AXIS	CURRENT	Flotante, solo lectura. Es la corriente absoluta, medida en $A_{rms}$ , que circula por el servomotor.
	SIMULATOR- _AXIS TRACTION- _AXIS	POWER	Flotante, solo lectura. Es la potencia activa, medida en kW, que el motor consume. En modo de generación el valor es negativo.

Cuadro 13: Fuentes de variables

De esta forma, el modelo solamente tendrá acceso a un conjunto reducido de variables en el programa PLC. Esta interfaz permite la interacción controlada entre la interfaz por software y el sistema SIMOTION D425.

### 8.3.3. Ensayos de cupla en función del tiempo - envío de datos al control

Según se detalló en la sección 8.1.4, el control presenta hacia el lado de la computadora un buffer de ancho fijo para poder introducir los pares torque-tiempo.

La interfaz gráfica es la encargada de alimentar este buffer. Como el ancho de este buffer sea potencialmente mayor a la cantidad de pares de cupla-tiempo en el ensayo, será necesario recargar el buffer de ancho fijo durante la ejecución del ensayo. Solo se permitirá que la carga se realice cuando el buffer de comunicación lo solicite.

Cuando el control está libre para recibir datos, activa una bandera de libre para recepción (CTR, por sus siglas en inglés). Al observar esta bandera, la interfaz gráfica envía un nuevo lote de pares cupla-tiempo al buffer de comunicación y ordena al control los almacene en su buffer interno activando una bandera de almacenamiento en buffer interno.

Una vez que el control haya agotado los comandos disponibles en el buffer de comunicación, envía la señal CTR y el ciclo comienza hasta agotar todos los pares de cupla-tiempo del ensayo.

### 8.3.4. Gráficos en tiempo real - latencia de datos

Como se analizó en la sección 8.1.4, el controlador puede tardar alrededor de 150 ms en responder una solicitud de respuesta OPC XML-DA. Considerando que es un requisito la visualización de gráficos en tiempo real de al menos cinco variables, de ejecutarse secuencialmente cada paso de gráfico implicaría 750 ms de tiempo de cálculo. Esto limitaría la resolución mínima del gráfico a 750 ms por variable, violando el requisito de diseño F.1.

Para poder mejorar la resolución temporal mínima, es necesario implementar técnicas de programación concurrente. Cada variable será actualizada periódicamente en su propio hilo de

ejecución. Esto asegura una latencia de 150 ms por medición. Sin embargo, es posible lanzar varias mediciones en simultáneo sin necesidad de que la medición anterior haya terminado. Esto permite que la resolución temporal mejore ampliamente los 150 ms.

Cada dato tiene asociada una marca de tiempo que se calcula en la computadora que corre el cliente OPC<sup>10</sup>. Como la medición temporal se hace luego de haber recibido el valor de la variable, necesariamente esta marca temporal no será precisa y se verá afectada por la latencia en la comunicación.

Para contrarrestar el efecto, se mide en la computadora el tiempo que transcurre en solicitar un dato al PLC y recibirlo. Este tiempo se conoce como tiempo de ida y vuelta (RTT, por sus siglas en inglés). Suponiendo que el tiempo de ida y vuelta es simétrico en cada tramo, se aplica una corrección para los valores temporales medidos de  $\frac{RTT}{2}$ .

## 9. Pruebas de validación del sistema diseñado

En este apartado se detallan las distintas maneras en que se verificaron los resultados obtenidos durante el desarrollo del banco de ensayos.

Para el análisis se dividieron las diferentes pruebas en 6 áreas diferentes:

- Aplicación de cupla en el eje del servomotor: la variable manipulada en el servomotor es la cupla electromagnética en el servomotor y no la cupla mecánica presente en el eje. Se realizarán dos ensayos para poder estimar la cupla mecánica en el eje y poder caracterizar las pérdidas del sistema.
- Ecuación de cupla - velocidad: esta sección describe las pruebas realizadas para determinar que el banco efectivamente es capaz de aplicar consignas de cupla en función de la velocidad comandada. Se contrastan los resultados con sus curvas teóricas.
- Curva de cupla - tiempo: la sección se dedica a verificar que los comandos de cupla en función del tiempo son procesados de manera correcta por el banco de ensayos.
- Duración de los ensayos: en este apartado se verifica cuánto tiempo puede ejecutarse un ensayo.
- Seguridad del banco: este conjunto de pruebas se realizó para poder verificar que el banco de ensayos cumple con las condiciones de seguridad para las que fue diseñado.
- Vistas de la interfaz: en esta sección se presentan las diferentes vistas de la interfaz y los diferentes mecanismos de control de flujo con los que cuenta el banco de ensayos.

### 9.1. Aplicación de cupla en el eje del servomotor

Como variable manipulada utilizada en los algoritmos descritos con anterioridad se utilizó la cupla electromagnética en el servomotor y no la cupla mecánica aplicada en el eje de simulación. Si bien ambas variables tienen cierta relación, no coinciden debido a las pérdidas en los rodamientos del servomotor y otros efectos físicos; estas pérdidas suelen variar en función de la velocidad.

Al no contar con un torquímetro para relevar una curva de cupla mecánica en función de la cupla electromagnética, fue necesario un ensayo alternativo para caracterizar al motor.

---

<sup>10</sup>Si bien sería posible enviar la marca temporal directamente desde el PLC, esto duplica el número de variables monitoreadas aumentando la carga en el servidor OPC

El ensayo consiste en acoplar dos servomotores idénticos y fijar, en el eje de simulación, una consigna de cupla y, en el eje de tracción, varias consignas de velocidad. Durante la experiencia, se relevará la cupla electromagnética en ambos ejes y la velocidad del eje.

De esta forma, se podrá aproximar la cupla mecánica a 0 cuando la velocidad sea constante. Entonces, la diferencia entre las cuplas electromagnéticas en ambos ejes será aproximadamente el doble de las pérdidas que se producen en un servomotor.

$$\sum T_i \approx 0 \implies 0 \approx T_{tracción} - T_{freno} - 2T_{pérdida} \quad (28)$$

Conociendo la diferencia entre las cuplas electromagnéticas, será posible aproximar la cupla de pérdida en cada servomotor.

A continuación, se presentan los resultados del ensayo:

Velocidad [RPM]	Diferencia entre cuplas (Promedio)[Nm]	Cupla de pérdida [Nm]
100	1,0	0.5
200	1,0	0.6
400	1,0	0.5
800	1.2	0.6
1600	1.3	0.65
2000	1.2	0.6
2500	1.3	0.65

Cuadro 14: Medición de la cupla de pérdida

Se realizó una regresión lineal para obtener la siguiente curva de compensación:

$$T_{pérdida}[Nm] = 0.00005 \cdot \omega[RPM] + 0.53 \quad \forall \quad \omega \in [100, 2500] \quad (29)$$

La siguiente ecuación vincula la cupla mecánica ejercida por un servomotor con la electromagnética:

$$T_{mecánica} \approx T_{electromagnética} - T_{pérdida} \quad (30)$$

Se deberá aplicar esta corrección luego de realizar cualquier tipo de ensayo para poder operar con la cupla mecánica del que ejerce el servomotor eje.

## 9.2. Ecuación de cupla - velocidad

En este apartado se estudia el comportamiento del sistema al realizar ensayos del tipo cupla-velocidad con el objetivo de analizar su correcta ejecución. Con este fin, se aislaron los 4 términos en la ecuación cupla y se observó su comportamiento realizando distintas pruebas en campo.

Se estudiarán los siguientes casos:

- Término constante: la cupla aplicada deberá permanecer constante.
- Término proporcional: la cupla introducida deberá variar proporcionalmente con la velocidad angular.
- Término cuadrático: la cupla aplicada deberá ser proporcional al cuadrado de la velocidad angular.
- Término derivativo: la cupla aplicada deberá variar proporcionalmente a la derivada de la velocidad angular.

### 9.2.1. Término constante

Para corroborar la correcta aplicación de torque constante se ensayó el motor simulador en configuración *eje libre*. Esta modalidad implica que el motor se encontrará acoplado mecánicamente a un dispositivo bajo prueba inactivo. Bajo este esquema, se introducirán consignas de cupla solamente en el eje simulador. Es decir, el dispositivo bajo prueba permitirá el giro sin oponer resistencia.

De esta manera, al introducir una consigna de cupla, por lo explicado en la sección 8.1.2, el dispositivo acelerará hasta su velocidad  $\omega_\infty$ . Si la consigna de cupla fuese constante, la velocidad teóricamente debería variar proporcionalmente a la cupla asignada.

La proporción a la que varía la velocidad está dada por el momento de inercia del sistema. Considerando el sistema motor-acople-motor, es posible describir el momento de inercia del sistema como la sumatoria de momentos de inercia:

$$J_{sistema} = 2J_{rotor} + J_{acople} \quad (31)$$

El fabricante especifica una inercia en el rotor de  $0.0156Kgm^2$ . En cuanto al acople, es posible aproximar su momento de inercia al momento de inercia de un cilindro hueco con un radio interior de 1,7cm, un radio exterior de 5,5cm con un peso de 6,3Kgs.

$$J_{acople} = \frac{1}{2}M(R_{int}^2 + R_{ext}^2) = 0.0104Kgm^2 \quad (32)$$

La ecuación indica que el sistema posee un momento de inercia de  $0.0416Kgm^2$ .

Considerando lo expuesto en la ecuación 2, es posible predecir teóricamente la recta de aceleración que el eje debería seguir frente a un escalón de torque:

$$\frac{T_{mecánico}}{J_{sistema}} * t = \omega \quad (33)$$

Como se observó en la sección anterior, para computar la cupla mecánica es necesario considerar las pérdidas en el servomotor estimadas en la ecuación 29:

$$\frac{T_{electromagnético} - 2T_{Pérdida}}{J_{sistema}} * t = \omega \quad (34)$$

$$\frac{T_{electromagnético} - 0.0001 \cdot \omega - 1.02}{J_{sistema}} * t = \omega \quad (35)$$

$$\frac{T_{electromagnético} - 1.02}{\frac{J_{sistema}}{t} + 0.0001} * t = \omega \quad (36)$$

$$(37)$$

Para realizar el ensayo, se fijo la consigna de cupla electromagnética en  $3Nm$ . Utilizando el software desarrollado, se relevó la curva obtenida:

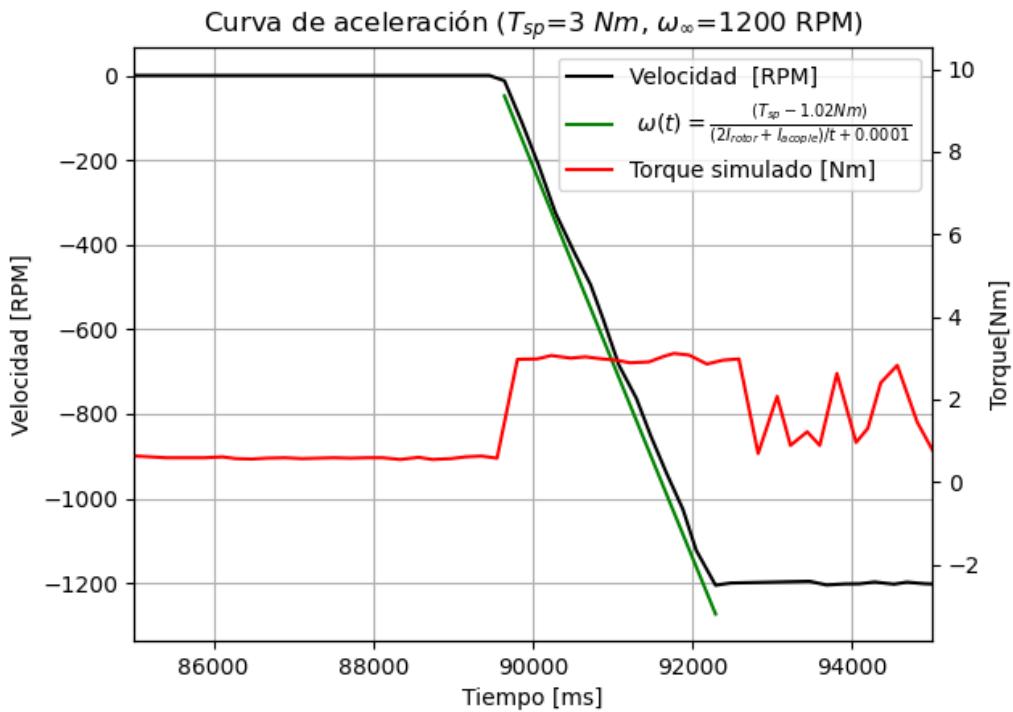


Figura 21: Curva de aceleración ensayada y respuesta en velocidad angular observada.

En la figura se observa que el modelo considerado fue capaz de predecir la medición realizada. Sin embargo, se observa un retardo de alrededor de 20ms al ser comparado con el modelo considerado.

Como medida adicional, se ensayó la respuesta a un escalón de velocidad de 100RPM con una consigna de cupla de 5Nm con un valor de  $\omega_{\infty}$  de 60RPM. El ensayo consistió en acoplar dos motores directamente, uno de estos motores es el eje simulador, mientras que el otro es el dispositivo bajo prueba. En el dispositivo bajo prueba se introduce una consigna de velocidad, mientras que en el simulador una consigna de torque

A continuación se presenta un gráfico con los resultados:

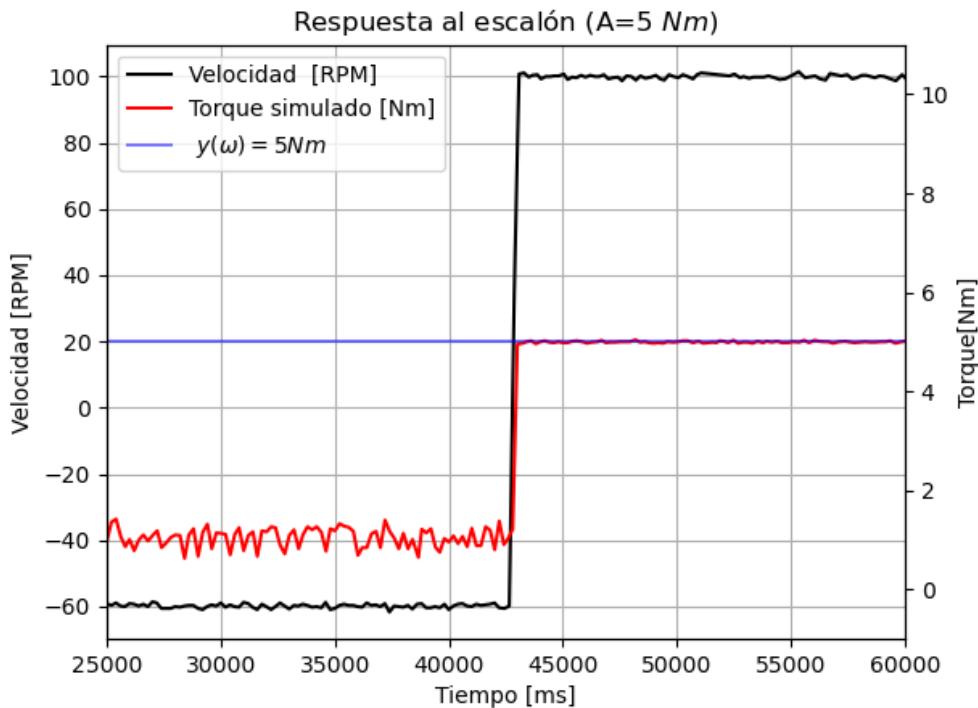


Figura 22: Respuesta al escalón del eje simulador únicamente con término constante de la ecuación cupla-velocidad

En la respuesta se observa que, previo a la introducción del escalón, el eje rota a su velocidad  $\omega_\infty$  (en este caso fijada a 60RPM) en condición de eje libre. En este estadío, el lazo de control de torque no se encuentra saturado y es comandado de forma tal que el eje mantenga la velocidad  $\omega_\infty$ . Este es el comportamiento esperado considerando lo mencionado en la sección 8.1.2 , donde se explica el mecanismo de control de torque.

Una vez que el escalón es aplicado, el eje simulador demora alrededor de 200ms en alcanzar su consigna de torque.

### 9.2.2. Término proporcional

Repitiendo la configuración del ensayo anterior, se relevó la respuesta al escalón únicamente activando el término proporcional. Se fijó el valor en  $0.191 \frac{Nm}{rad/s}$  y se ensayó la respuesta a un escalón de velocidad constante de amplitud 500 RPM. En la figura 24 se presentan los resultados:

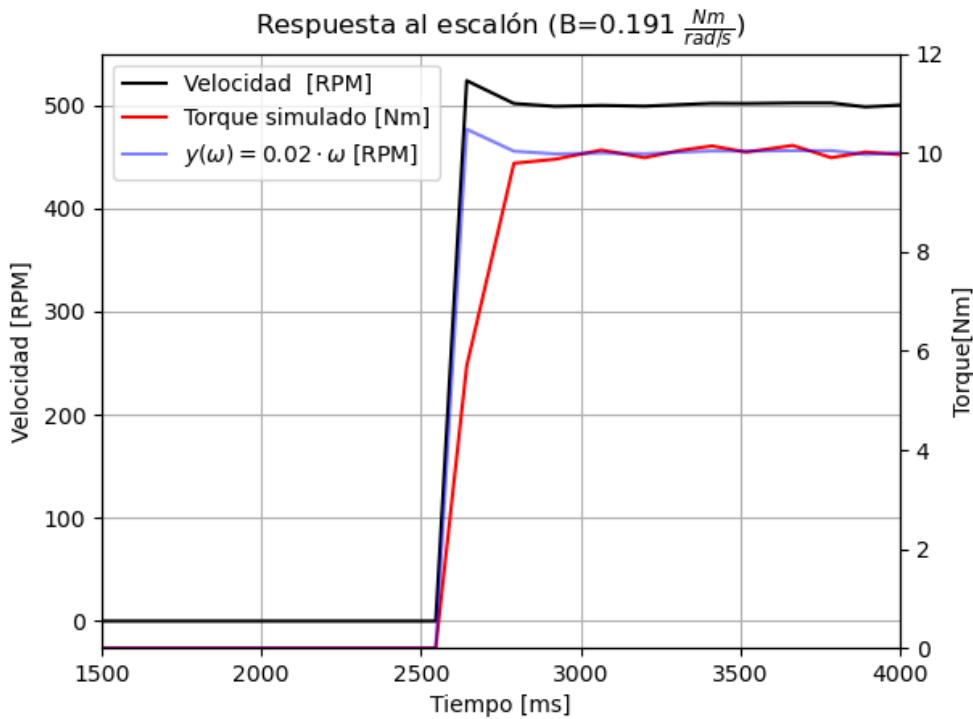


Figura 23: Respuesta al escalón únicamente con término proporcional de la ecuación cupla-velocidad

Se observa que el torque simulado responde de manera lineal, consistente con los parámetros del ensayo. Sin embargo debido al filtrado de la velocidad angular y el tiempo de cálculo de la ecuación de cupla, se produce un retardo de 150ms en la aplicación del torque simulado.

En la figura se observa un sobre-pico en la velocidad medida, la respuesta a este sobre-pico se ve atenuada en la cupla efectivamente aplicada por acción del filtrado en la variable velocidad angular.

### 9.2.3. Término cuadrático

En condiciones similares, se repite el ensayo de respuesta al escalón pero en esta instancia únicamente con el término cuadrático activo. La amplitud del escalón se eligió en 500RPM y el coeficiente del término cuadrático en  $0.036 \frac{Nm}{(rad/s)^2}$ .

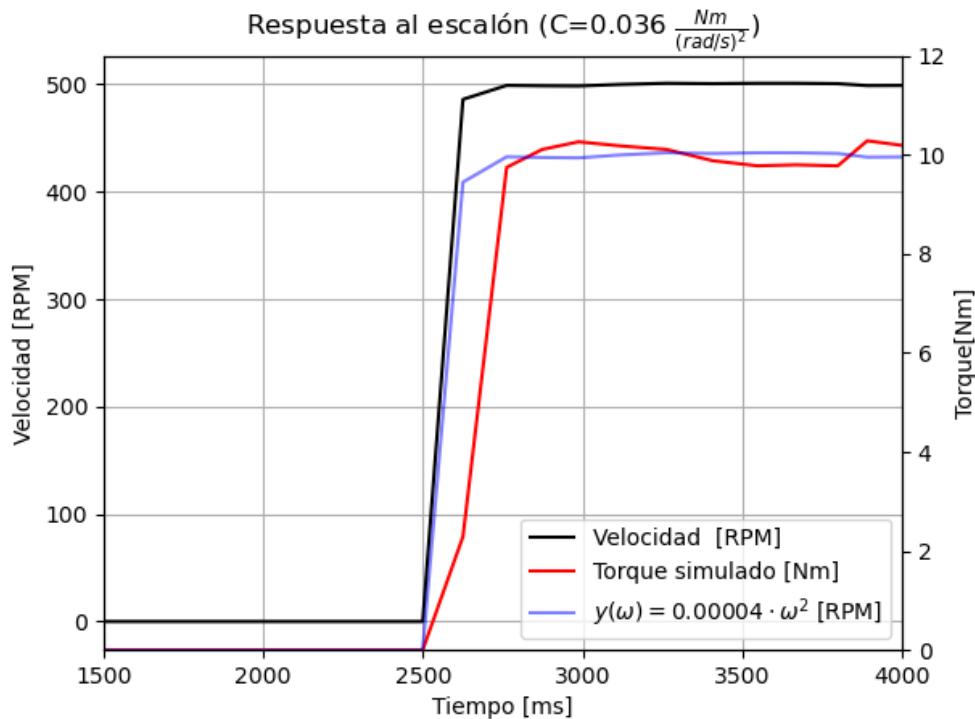


Figura 24: Respuesta al escalón únicamente con término cuadrático de la ecuación cupla-velocidad

Nuevamente se observa un retardo en torno a los 150ms. El valor medio de la respuesta sin filtrar converge al valor medio de torque simulado luego de un tiempo de estabilización.

#### 9.2.4. Término inercial

En el dispositivo bajo prueba se introdujo una consigna de velocidad de 500RPM y se fijó únicamente el término inercial con un valor de  $0.05 Kgm^2$ . Se graficó el torque efectivamente aplicado, llamado torque simulado, el torque que corresponde al término inercial, calculado según  $D \frac{\partial v}{\partial t}$ , y la velocidad del eje.

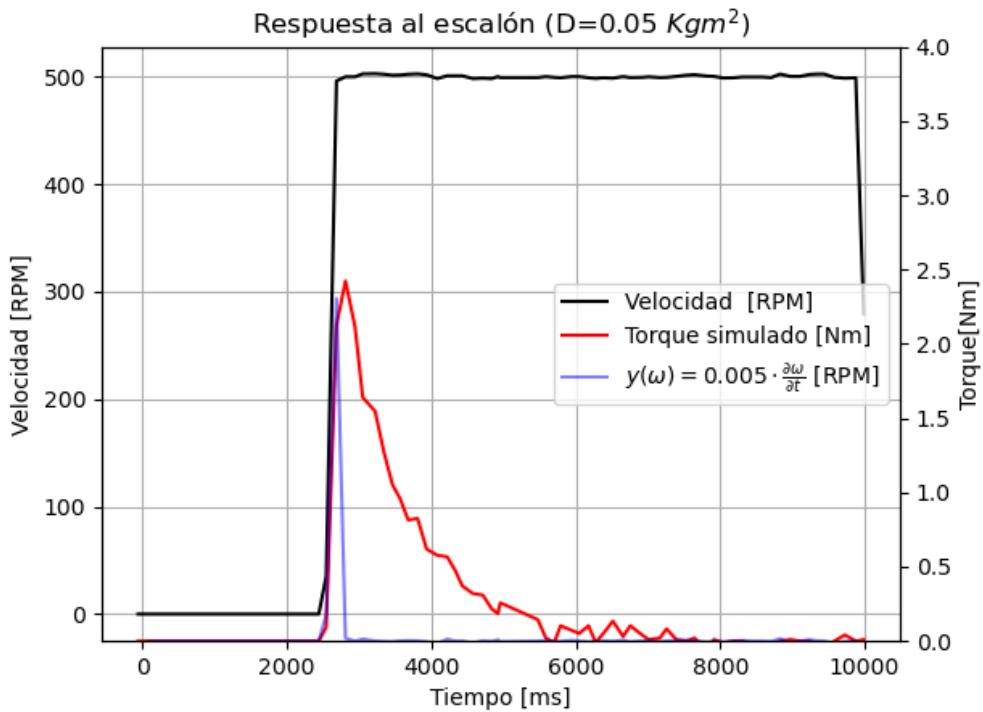


Figura 25: Respuesta al escalón únicamente con el término inercial de la ecuación cupla-velocidad

En la imagen es posible observar cómo el torque calculado presenta una respuesta más rápida que su par simulado, en retraso por aproximadamente 150ms. La señal simulada tiene un tiempo de estabilización de alrededor de 1.9 segundos.

Esto es una consecuencia directa de la aplicación de un filtro pasa-bajos. Si bien el término no filtrado ofrece una emulación de la inercia con mayor precisión, el ruido en la medición de velocidad se termina amplificando, lo que limita el rango de valores disponibles para el término D.<sup>11</sup>

### 9.3. Curva de cupla - tiempo

En concordancia con el objetivo de la sección anterior, se presentan en esta sección las pruebas de validación llevadas a cabo para corroborar el correcto funcionamiento de la aplicación de cupla en función del tiempo.

#### 9.3.1. Precisión temporal

Para estudiar el correcto sincronismo de los comandos de torque-tiempo se introdujo la siguiente consigna:

$$\begin{aligned} T(t) &= 0.01 \frac{\text{Nm}}{\text{s}} \times t \\ T(t + 115\text{s}) &= T(t) \end{aligned} \quad (38)$$

Se introdujo la ecuación discretizada con  $\Delta T = 100\text{ms}$ . Se calculó el error relativo con referencia en el comando de torque. A continuación los resultados.

<sup>11</sup>Si bien la figura no da cuenta de una gran inestabilidad en el término inercial, el ensayo se condujo aplicando el torque simulado. Esto produce que el dispositivo bajo prueba responda de una manera más suave a las variaciones por el torque inercial, produciendo un menor ruido en la velocidad alcanzada. Empíricamente fue posible retirar el filtro IIR y comprobar la inestabilidad para valores de inercia superiores a  $0.2\text{Kgm}^2$

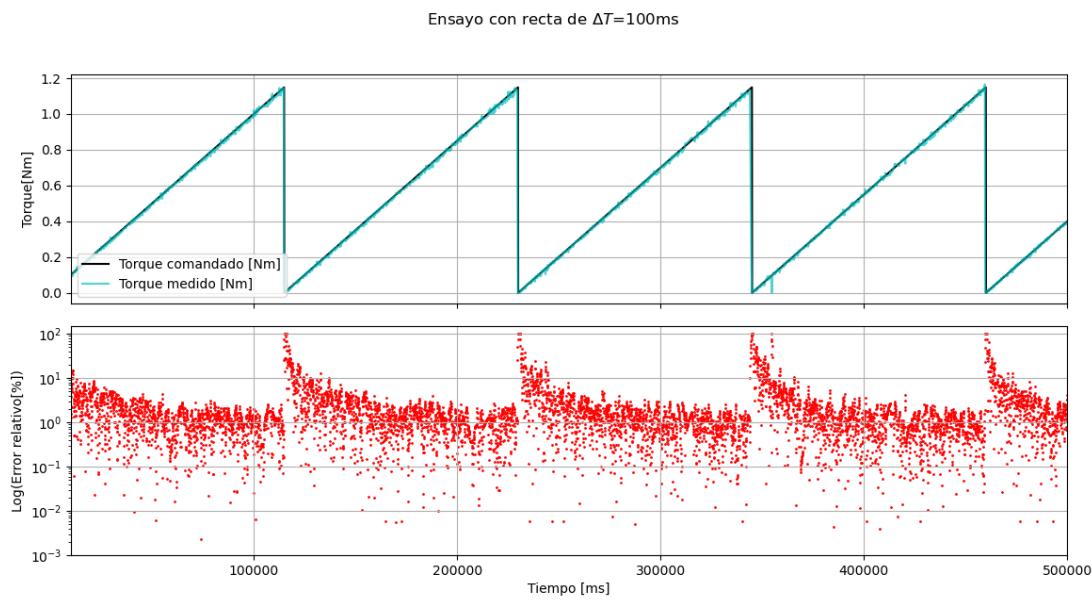


Figura 26: Comparación entre la curva de cupla-tiempo efectivamente aplicada y la curva objetivo

Se observa que, a medida que el módulo de cupla disminuye, el error relativo es mayor. Probablemente el torque medido tiene cierto retardo respecto al comando de cupla. Por este motivo, al calcular el error relativo, se produce una diferencia que es más notoria cuando la cupla comandada es cercana a cero. Este efecto puede ser observado en las transiciones bruscas de la señal, donde el error es máximo.

Además de ello, se observa que forma de onda de la curva de error es periódica. Esto indica que, independientemente del error en tiempo, el instrumento es capaz de ejecutar consignas de cupla-tiempo con alta repetibilidad.

## 9.4. Duración de ensayos

Experimentalmente fue posible realizar ensayos de hasta 40 minutos de duración y 10 mil pares cupla-tiempo. Aún no fue posible hallar empíricamente una cota máxima para la duración del ensayo ni la cantidad de puntos en un ensayo.

## 9.5. Seguridad del banco

En esta sección se dedica a corroborar el cumplimiento de los requisitos de seguridad establecidos mediante la evaluación de las medidas implementadas.

### 9.5.1. Paradas de emergencia accionadas por el usuario

Fue posible verificar el correcto funcionamiento de las paradas de emergencia accionadas por el usuario, tanto en el botón físico como en el botón de la interfaz gráfica.

Para liberar la emergencia del equipo y reanudar el ensayo, es necesario que el operador accione el botón de reconocimiento de alarma exclusivamente desde la interfaz física y luego el botón de arranque. Esta medida garantiza que el operador realice una inspección visual de la condición mecánica del eje previo a liberar la emergencia y que el sistema reciba confirmación para reanudar el movimiento.

### 9.5.2. Protección contra desconexión de interfaz

Se comprobó la efectividad del sistema de protección frente a la desconexión de la interfaz. Este sistema está compuesto por el mecanismo de keepalive, descrito en la sección 8.1.5, que permite detectar la desconexión de la interfaz a nivel de software.

Adicionalmente, se corroboró que al desconectar la interfaz física, el botón de emergencia, de contacto normalmente cerrado, dispara la emergencia del sistema.

### 9.5.3. Control de torque máximo y derivada de torque

Se corroboró que, durante cualquier tipo de ensayo, la cupla máxima y su derivada permanecen limitadas a 23 Nm y 7000 Nm/s, respectivamente. Ante el exceso de cualquiera de estos valores, el sistema activa de forma automática la condición de emergencia, deteniendo inmediatamente el movimiento del equipo.

Por motivos de seguridad, estos valores únicamente pueden modificarse accediendo al software de desarrollo, SIMOTION SCOUT.

## 9.6. Interfaz gráfica

En esta sección se recopila la evidencia que corrobora el cumplimiento de los requisitos de diseño I.IG.0 a I.IG.3 en la tabla 8.

### 9.6.1. Ventana principal

La interfaz se implementó utilizando la estructura de desarrollo JAVA Swing. La interfaz consiste en una pantalla principal y dos ventanas informativas que están ocultas inicialmente y se revelan según sea necesario.

En la pantalla principal es posible realizar la totalidad de la operación del banco de ensayos.

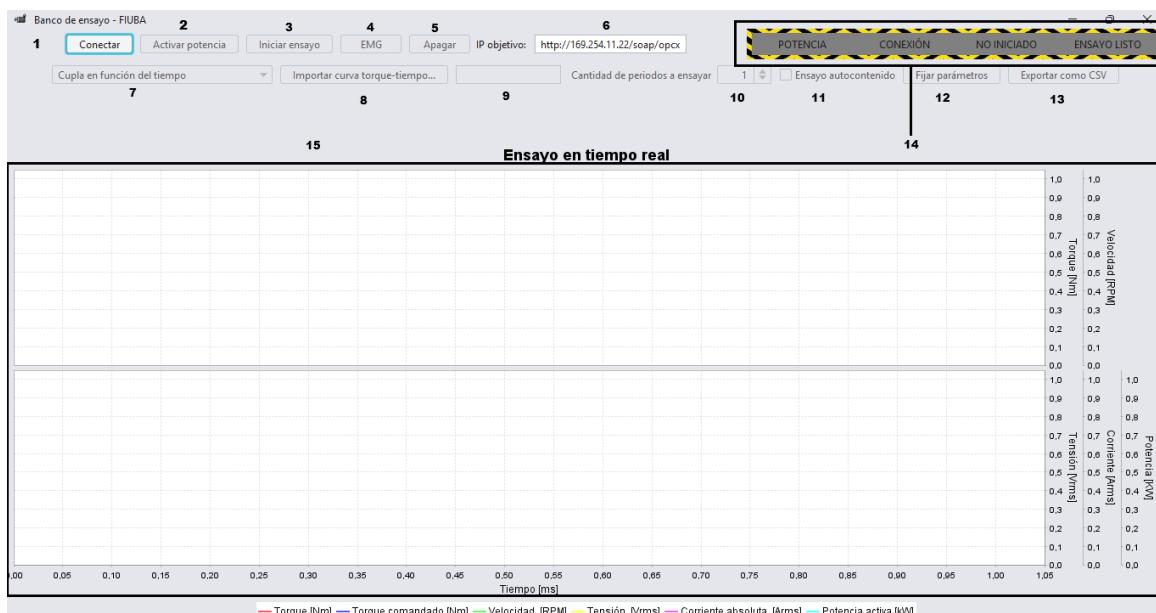


Figura 27: Ventana principal del programa

Componente	Descripción
1	Botón de conexión.
2	Botón de activación de circuito de potencia.
3	Botón de inicio de ensayo.
4	Botón de activación de emergencia.
5	Botón de apagado del sistema de accionamientos.
6	Campo de texto para introducir la dirección del servidor OPC XML-DA.
7	Selector de tipo de ensayo.
8	Botón para importar la curva de cupla-tiempo en formato CSV.
9	Campo de texto con el nombre de archivo de la curva de cupla-tiempo seleccionada.
10	Cantidad de periodos a ejecutar de la curva cupla-tiempo seleccionada.
11	Selector de ensayo autocontenido. Este modo de operación permite introducir consignas de velocidad-tiempo en el eje no utilizado del sistema de accionamientos. Solamente fue utilizado para realizar demostraciones del funcionamiento del equipo.
12	Botón para fijar los parámetros del ensayo.
13	Botón para exportar las mediciones realizadas en formato CSV.
14	Panel de indicadores. De izquierda a derecha: activación de circuito de potencia, estado de la conexión con el sistema de accionamientos, estado actual del ensayo, estado de la carga de parámetros.
15	Gráfico de las mediciones de ensayo en tiempo real.

Cuadro 15: Elementos de la pantalla principal

La ventana principal se compone de las siguientes áreas:

- Área de entrada de datos: en esta área se agrupan los controles necesarios para que el usuario interactúe con el sistema. Esta zona se dedica al control el estado del ensayo, carga los parámetros de ensayo y finalmente puede exportar estos datos para su posprocesamiento. Estos controles se habilitan o deshabilitan según el contexto.
- Área de indicadores: este grupo de controles se utilizan para informar el estado del ensayo. De manera similar a las señales lumínicas de la interfaz física, los controles en el área de indicadores cambian su color y texto en función de su estado.
- Área de gráficos en tiempo real: este área está dedicada a la visualización de datos en tiempo real. Se compone de un visor de valores instantáneos (cupla, tensión, corriente, potencia y velocidad) y un gráfico en tiempo real con sus valores históricos.

#### 9.6.1.1 Control de flujo de programa

La interfaz gráfica se diseñó para permitir solo determinadas operaciones en función de la etapa de carga de datos del sistema. A continuación, una breve descripción:

1. Conexión a la unidad de control: Es el primer control habilitado. Inicia la conexión con el servidor OPC y el sistema de keepalive. Permite el encendido del módulo de potencia y el apagado de la unidad de control.
2. Habilitación de potencia: Enciende el módulo activo de línea y habilita el movimiento en el eje. Permite la selección de ensayo, la emergencia del sistema y el apagado de la unidad de control y accionamientos.
3. Selección de ensayo: Permite elegir ensayo de cupla en función del tiempo, en función de la velocidad o ambos y también la modalidad autocontenido. En función del tipo elegido, se desbloquean controles específicos para el ingreso de parámetros de ensayos. Una vez que la entrada de usuario es validada, se habilita en envío de parámetros al Control.
4. Carga de parámetros: el botón inicia el envío de parámetros (pares torque-tiempo, coeficientes de la ecuación cupla y pares tiempo-velocidad para la modalidad autocontenido). Una vez finalizado el envío se habilita el inicio del ensayo.
5. Inicio de ensayo: Permite comenzar el movimiento en el eje simulador.
6. Interrupción de ensayo, reanudación de ensayo y emergencia: Permite controlar el movimiento del simulador, parando y reanudando el movimiento.
7. Desconexión de la unidad de control y exportación de datos recolectados: una vez que haya transcurrido el tiempo de ensayo se habilita la exportación en formato CSV de los datos recolectados, el apagado del sistema de accionamientos y desconexión del servidor OPC.

#### 9.6.2. Ventana de avisos y alertas

En el caso de que se produjera alguna condición de alarma, el software disparará una ventana informando la condición del equipo. A continuación se presenta un ejemplo de la ventana:

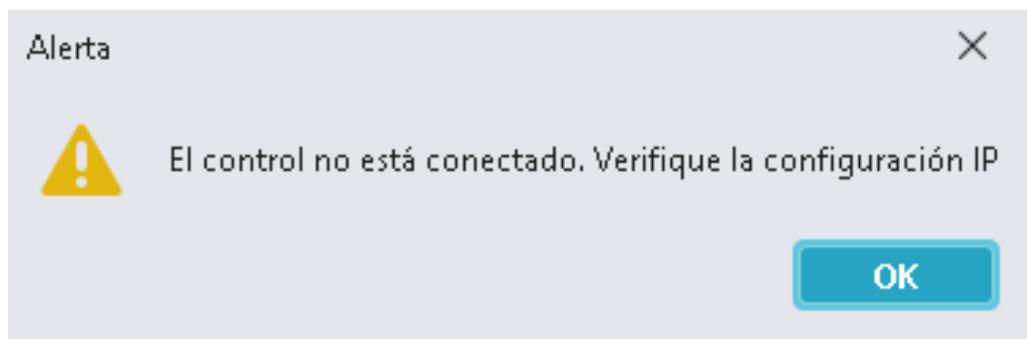


Figura 28: Ejemplo de alerta al estar imposibilitada la conexión

Esta ventana solamente se presenta si ocurre alguno de los eventos en la tabla 16:

Evento
Desconexión de la interfaz gráfica
Falla del mecanismo de keepalive
Desconexión de la interfaz física
Exceso de cupla
Exceso de derivada de cupla
Deshabilitación inesperada del rectificador
Deshabilitación inesperada del driver de servomotor
Archivo CSV no reconocido
Tiempo entre comandos de cupla o de velocidad menor a 100ms
Comando de cupla-tiempo excede los 26Nm
Formato numérico de un parámetro de ecuación cupla-velocidad no reconocido
El término inercial es demasiado grande

Cuadro 16: Eventos que disparan la ventana de alertas del sistema

#### 9.6.3. Ventana de gráficos en tiempo real para la ejecución ensayos autocontenidos

Al seleccionar la opción para realizar un ensayo autocontenido, se habilitará una ventana adicional que muestra información en tiempo real de las variables medidas en el dispositivo bajo prueba.

Al igual que la ventana en la figura 27, esta vista presenta un gráfico en tiempo real con las mediciones tomadas del eje de tracción.

## 10. Realización de un ensayo de referencia

Con el objetivo de observar la capacidad del banco como instrumento de medición, se lo empleará en la realización de un ensayo demostrativo.

El dispositivo a ensayar será el servomotor Siemens 1FK7105, este motor es idéntico al motor utilizado en el eje simulador y ya forma parte del sistema de accionamientos y se ensayarán bajo la modalidad autocontenido. Ambos motores se acoplarán mecánicamente en una configuración como muestra la figura 29:



Figura 29: Acople mecánico de motores: a la izquierda el motor simulador y a la izquierda el motor de planta motriz

El ensayo a realizar será una evaluación del motor 1FK7105 como motor de vehículo. Si bien este tipo de motor no está destinado a aplicaciones automotrices, se considera adecuado para ilustrar de manera representativa la funcionalidad del banco de ensayos.

Se comprobará el desempeño motor con un modelo de planta motriz que emula las características dinámicas de un vehículo. Durante el ensayo, se aplicarán consignas de velocidad-tiempo en el motor de tracción, mientras que el motor de simulación recibe consignas de cupla-velocidad, en forma de la ecuación cupla.

### 10.1. Modelo de planta motriz

Para modelar la planta motriz se utilizará como referencia el paper 'Electric Traction Emulator System' [21], presentado por Pérez Carmona et al. en marzo de 2025. En este artículo los investigadores proponen el siguiente modelo matemático para la dinámica de un vehículo:

$$T_m = (J_m + \frac{J_r}{i_g^2 i_0^2 \eta_t} + \frac{M R_d^2}{i_g^2 i_0^2 \eta t}) \frac{d\omega}{dt} + \frac{R_d}{i_g i_0} (f_r M g \cos(\alpha) + M g \sin(\alpha)) + \frac{\rho A_f C_D R_d^3}{2 i_g^3 i_0^3 \eta} \omega^2 \quad (39)$$

El grupo define las siguientes variables:

$$\left\{ \begin{array}{l} M : \text{Masa del vehículo} \\ \rho : \text{Densidad del aire} \\ A_f : \text{Área frontal del vehículo} \\ C_D : \text{Coeficiente de resistencia aerodinámica} \\ f_r : \text{Coeficiente de resistencia a la rodadura} \\ \omega : \text{Velocidad angular} \\ J_m : \text{Momento de inercia del eje del motor} \\ J_r : \text{Momento de inercia de las ruedas del vehículo} \\ R_d : \text{Radio de las ruedas del vehículo} \\ i_g : \text{Relación de transmisión} \\ i_o : \text{Relación de transmisión final} \\ \eta_t : \text{Eficiencia mecánica} \end{array} \right. \quad (40)$$

Bajo este esquema, la relación entre la velocidad lineal del vehículo y la velocidad angular de su motor es:

$$v = \frac{R_d}{i_g i_0} \omega \quad (41)$$

### 10.2. Perfiles de velocidad-tiempo a ensayar

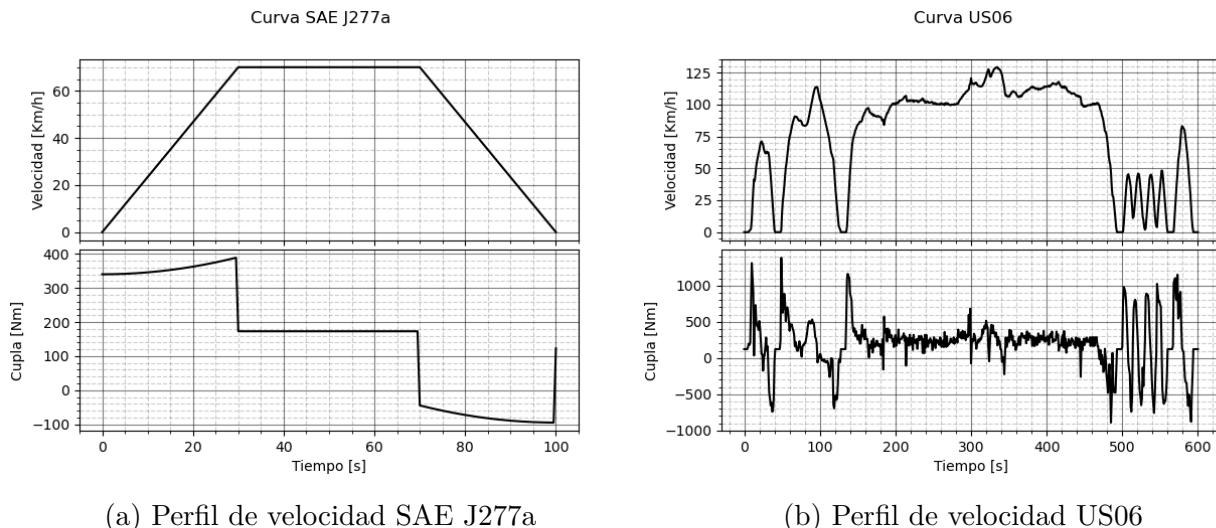
Para poder aplicar el modelo descrito en la ecuación 39, es necesario asignar valores a sus parámetros. En su publicación, los investigadores proponen la siguiente tabla:

Parámetro	Valor	Unidades
$R_d$	0.30	m
$i_0$	1	-
$i_g$	1.5	-
$\eta$	0.9	-
$f_r$	0.02	-
$\omega$	-	$\frac{1}{s}$
$A_f$	2.2	$m^2$
$C_D$	0.4	-
$M$	1500	Kgs
$g$	9.82	$\frac{m}{s^2}$
$\alpha$	1	$^\circ$
$J_r$	1	$kgm^2$
$J_m$	0.000483	$kgm^2$
$\rho$	1.225	$kg/m^2$

Cuadro 17: Parámetros de simulación

Utilizando los valores de la tabla, el modelo del vehículo y la relación entre velocidad angular y lineal descrita en la ecuación 41, es posible calcular la cupla de frenado que ejercida por el eje de simulación en función de la velocidad lineal del vehículo.

Finalmente, se escogieron dos perfiles diferentes de velocidad en función del tiempo. El primero de ellos es el perfil SAE J277a, utilizado en la caracterización de autonomía de vehículos eléctricos. El segundo es el perfil US06, un perfil de manejo agresivo utilizado para evaluar el consumo de batería.

Figura 30: Perfiles de velocidad ensayados con sus correspondientes cuplas resistivas.<sup>12</sup>

### 10.3. Escalamiento de variables

Típicamente, los vehículos eléctricos requieren de grandes torques para su locomoción. Sin embargo, el servomotor utilizado como eje de simulación únicamente cuenta con 26 Nm de cupla máxima.

<sup>12</sup>Las curvas de cupla resistiva fueron calculadas analíticamente computando la ecuación de cupla utilizando los parámetros en la tabla 18, sin aplicar escalamiento de variables.

Enfrentados a la misma problemática, los autores aplicaron una reducción de escala del problema mecánico. De esta manera, es posible reducir la cupla necesaria manteniendo la dinámica del sistema<sup>13</sup>.

Para escalar la cupla por un factor x, resulta necesario escalar  $A_f$ , el área frontal del vehículo,  $J_m$ , el coeficiente de inercia del eje del motor,  $J_r$ , el coeficiente de inercia de las ruedas del motor, y  $M$ , la masa del vehículo, por el mismo factor.

Observando la figura 30, se observa que, para el modelo de vehículo eléctrico ensayado, la curva SAE J227a produce una cupla máxima de 388Nm mientras que la curva US06 produce una cupla máxima de 1380Nm. Considerando que la cupla máxima del servomotor es de 26 Nm y que el motor ensayado, con iguales características, deberá superar la cupla resistente para alcanzar cualquier consigna de velocidad, se fijó el torque máximo resistente en 5Nm.

A raíz de ello, se obtienen dos factores de escala:  $x_{SAEJ227a} = 0.0257$  y  $x_{US06} = 0.00361$ . A continuación, se expresan los parámetros de ensayo en términos de la ecuación 11, considerando sus factores de escala:

SAE J227a		
A	3.152	Nm
B	0	Nms/rad
C	0.000137	Nms <sup>2</sup> /rad <sup>2</sup>
D	1.73	Nms <sup>2</sup> /rad
US06		
A	0.443	Nm
B	0	Nms/rad
C	0.0000192	Nms <sup>2</sup> /rad <sup>2</sup>
D	0.243	Nms <sup>2</sup> /rad

Cuadro 18: Valores de ensayo en términos de la ecuación cupla

## 10.4. Resultados

### 10.4.1. SAE J227A

Se introdujo el conjunto de pares velocidad-tiempo aparejados con los parámetros de la tabla 18. Se midieron variables en el dispositivo bajo prueba (DUT, por sus siglas en inglés) y en el motor de simulación de cupla. En la figura 31 se presentan los resultados del ensayo:

<sup>13</sup>Esta técnica supone que un comportamiento lineal del motor bajo ensayo, lo que no comprende efectos de orden superior

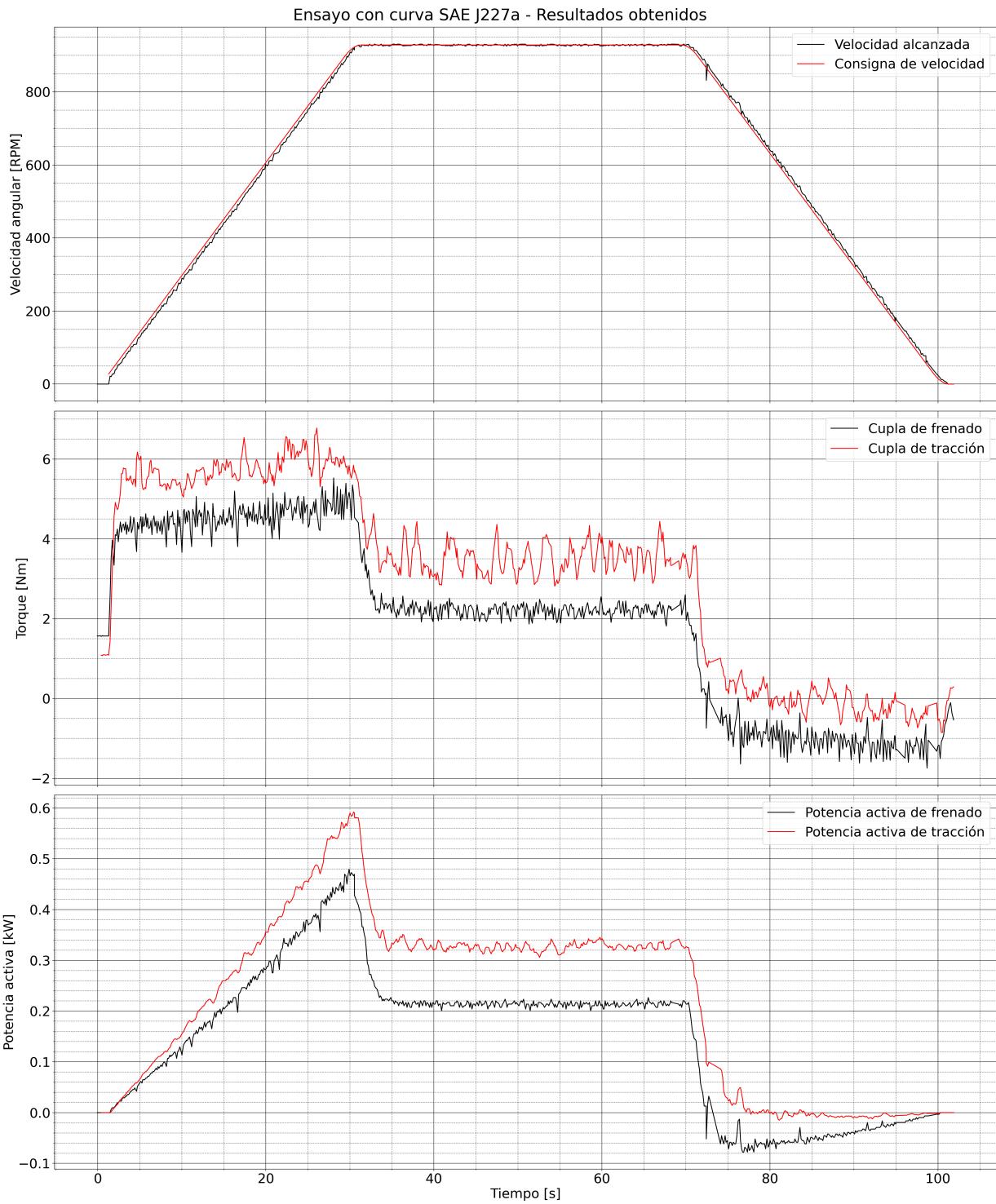


Figura 31: Curvas de velocidad, cupla y potencia activa medidas durante el ensayo utilizando la curva SAE J227a

En la figura, la cupla de frenado es la entregada por el eje de simulación y se muestra con signo positivo. Es decir que, cuando esta cupla tiene un valor positivo, se opone al movimiento del dispositivo bajo prueba. En el caso contrario, un torque de frenado negativo indica que el eje de simulación está impulsando al dispositivo bajo prueba. El mismo razonamiento se aplica para la potencia de frenado.

Durante la aceleración se observa una curva ligeramente cuadrática. Esta curva se explica por el término constante, el término cuadrático y el término inercial de la ecuación de cupla. Como la velocidad varía linealmente, entonces el término inercial es constante.

En la etapa de velocidad constante, la cupla de frenado responde tras un retardo de aproxi-

madamente 2 segundos. El retardo es producto del filtro pasa-bajos introducido para mejorar el desempeño del término inercial. La respuesta es prácticamente constante; al no variar la velocidad, el término inercial no tiene influencia.

Finalmente, durante la desaceleración se nota una inversión en la cupla de simulación. En esta etapa, la inercia se vuelve impulsora del sistema ensayado. También es posible observar la contribución cuadrática producto de la resistencia aerodinámica simulada.

En cuanto a la cupla del dispositivo simulado, esta siempre es mayor al torque simulado. Esta es una condición necesaria para poder sostener el movimiento. Sin embargo, se observa que en la etapa de desaceleración la cupla es prácticamente nula, con un pequeño sesgo negativo.

En esta instancia, el dispositivo bajo prueba aprovecha el impulso inercial del eje de simulación para poder mantener su consigna de velocidad. Cuando su cupla es negativa actúa como freno del eje de simulación.

#### 10.4.2. US06

Se discretizó la curva US06 a 200ms de separación entre muestras y se cargó en el banco de ensayos. Utilizando el segundo conjunto de parámetros en la tabla 18, se llevó a cabo el ensayo. En la figura a continuación, se graficaron los resultados:

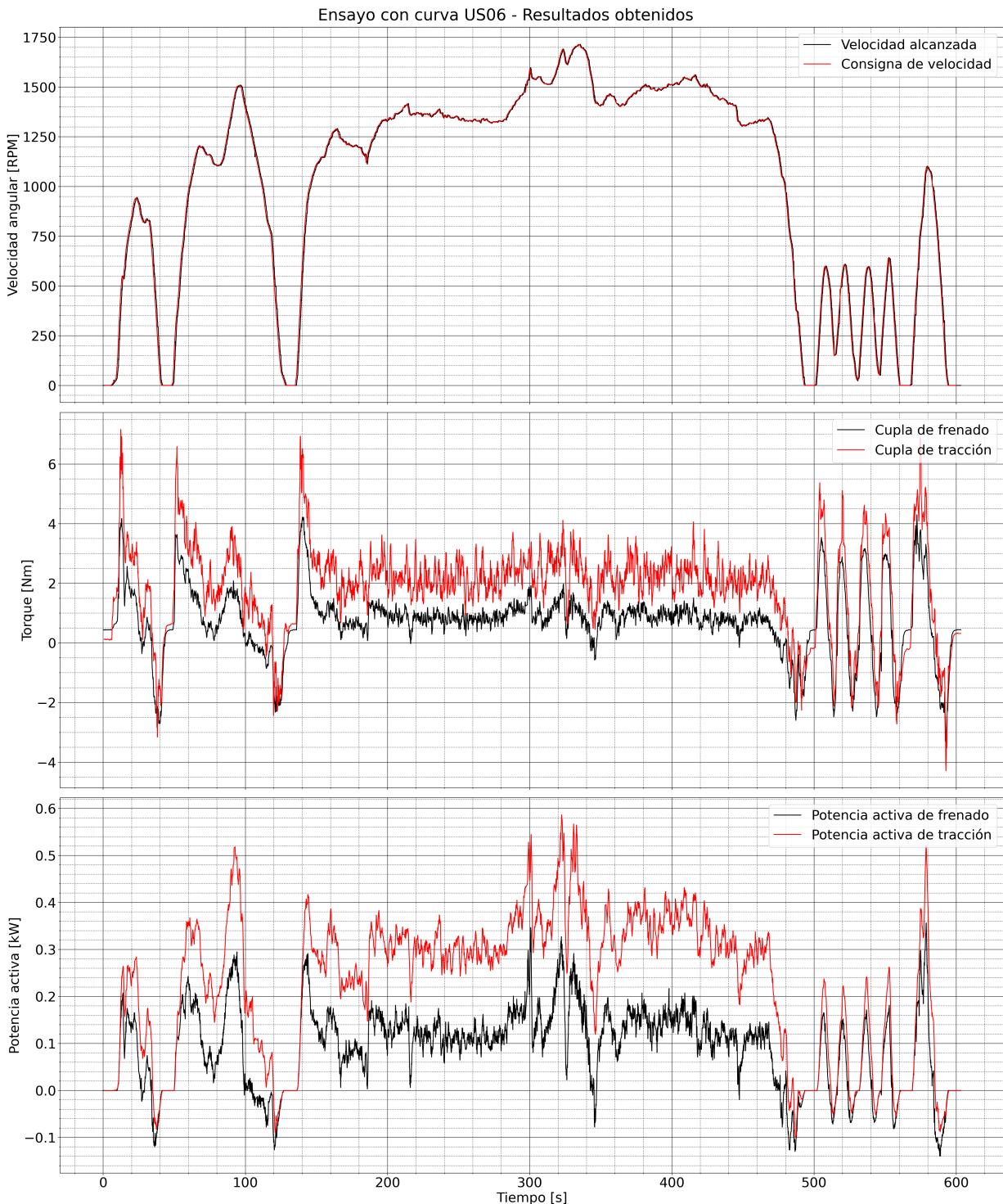


Figura 32: Curvas de velocidad, cupla y potencia activa medidas durante el ensayo utilizando la curva US06

Es posible realizar el mismo análisis cualitativo que se realizó para el ensayo anterior. En este caso, por la forma del perfil de velocidad, las variaciones son más abruptas.

La forma general de las curvas es similar a las obtenidas por el grupo de investigación en su paper 'Electric Traction Emulator System'. Además, al comparar el perfil de cupla de frenado que se presenta en la figura 30b con el perfil obtenido, la forma de onda es prácticamente una versión escalada de los resultados presentados.

Se observa que transitoriamente el dispositivo ensayado se vuelve generador de energía. Estos períodos coinciden con las desaceleraciones bruscas en la curva US06. En estos instantes, la contribución de la cupla de frenado se vuelve impulsora debido a la inercia simulada del

vehículo, lo que obliga al eje de tracción a actuar como freno para mantener la consigna de velocidad.

Esta inversión de roles, en donde el eje de tracción actúa como freno y el eje de simulación actúa como planta motriz, suele ser aprovechada en los vehículos eléctricos para recargar sus bancos de baterías, aumentando su autonomía.

## 11. Discusión

Esta sección contiene reflexiones relacionadas con el cumplimiento de los objetivos de la presente tesis, como también una síntesis de los principales resultados obtenidos. Se utilizarán métricas cuantitativas para poder realizar una evaluación objetiva y comparable con desarrollos posteriores en caso de que estos existan.

### 11.1. Cumplimiento de objetivos

El desarrollo incluyó un sistema de supervisión, control y adquisición de datos. Este sistema admite comunicación mediante un cliente OPC XML-DA que permite modificar distintas características de un ensayo y recibir sus resultados en tiempo real. El proyecto se encuentra documentado tanto en estas páginas como también en documentos técnicos complementarios y su manual de operación.

Además de ello, el sistema permite realizar 3 tipos de ensayo: de torque-tiempo, de torque-velocidad y mixtos. En el caso de torque-velocidad y mixto, se implementa una característica torque-velocidad cuadrática con incorporación de un término derivativo. Estos tipos de ensayos pueden ser ejecutados utilizando la modalidad autocontenida, pensada para realizar demostraciones rápidas.

Los ensayos pueden ser programados mediante una interfaz gráfica que permite la visualización de sus variables en tiempo real y su recolección. Se realizó un ensayo de validación y fue posible reproducir los resultados de un grupo de investigación independiente.

Los datos de estas pruebas, y todas las demás realizadas en esta tesis, se obtuvieron mediante el sistema de adquisición de datos diseñado para este fin. Esto en sí mismo valida al sistema de adquisición de datos.

Considerando lo expuesto, es posible afirmar que los objetivos de esta tesis en ingeniería electrónica fueron alcanzados. El instrumento entregado puede ser utilizado plenamente para su uso según los requisitos encomendados por los operadores del banco.

### 11.2. Cumplimiento de requisitos de diseño

A continuación, se revisitan los requisitos de diseño propuestos en la tabla 8 y se analiza su cumplimiento.

Id.	Descripción	Estado
I.0	Interfaz distribuida en parte física (IF) y parte gráfica (IG)	Cumplido
I.1	Diseño intuitivo, que permita la operación con mínima experiencia	Cumplido
I.IF.0	Interfaz física con botonera que permita iniciar, reiniciar y detener ensayos.	Cumplido
I.IF.1	Interfaz física con luces que indiquen los estados del ensayo: iniciado, detenido o emergencia.	Cumplido
I.IG.0	Interfaz virtual que permita iniciar, pausar y detener ensayos	Cumplido
I.IG.1	Visualización de gráficos en tiempo real de al menos 5 variables: cupla medida, tensión en motor, corriente, potencia activa y velocidad.	Cumplido
I.IG.2	Exportación de los datos de las 5 variable en formato CSV	Cumplido
I.IG.	Desarrollo multisistema, mantenable y flexible	Cumplido
F.0	Admitir ensayos de cupla variable en función de la velocidad (CV) y cupla variable en función del tiempo (CT) en el eje simulador	Cumplido
F.1	Resolución temporal mínima de 150ms	Cumplido
F.2	Permitir la realización de ensayos utilizando el segundo motor del banco de ensayos como eje de tracción.	Cumplido
F.CV.0	Admitir cuplas variables con la velocidad según lo descrito en la ecuación 11	Cumplido
F.CV.1	Permitir la simulación de coeficientes iniciales de al menos 7 veces la inercia del rotor del servomotor.	Cumplido
F.CT.0	Admitir cuplas variables con el tiempo.	Cumplido
F.CT.1	Admitir un tiempo mínimo entre comandos de torque de almenos 100ms.	Cumplido
F.CT.2	Permitir comandos de torque no equidistantes en el tiempo	Cumplido
SW.0	Software mantenable, multiplataforma y escalable	Cumplido
S.0	Aplicar límites máximos de velocidad y cupla en el eje simulador	Cumplido
S.1	Aplicar limites máximos de derivada de cupla en el eje simulador	Cumplido
S.2	Implementar freno de emergencia en hardware y software	Cumplido
S.3	Prohibir la utilización del equipo con alguna de las interfaces inactivas	Cumplido

Cuadro 19: Tabla de requisitos de diseño y su estado para el simulador de cupla

### 11.3. Caracterización del sistema

Considerando las diferentes pruebas de validación realizadas, se caracterizan las principales especificaciones técnicas con las que cuenta el instrumento.

Máximo comando de torque	26Nm	
Velocidad máxima	3000RPM	
$\omega_\infty$	2500RPM	
Máxima derivada de torque	7000Nm/s	
Resolución temporal de adquisición de datos	150ms	
Resolución temporal de introducción comandos de torque	Torque-tiempo 100ms	Torque-velocidad 30ms
Máximo tiempo de ensayo alcanzado	Al menos 40 minutos o 10000 pares cupla-tiempo	
Latencia en el comando de freno de emergencia	menor a 100ms	
Latencia frente a la desconexión del panel de operador	4 segundos	
Formatos de salida de datos	.CSV	

Cuadro 20: Principales resultados alcanzados

## 12. Conclusión

Cómo tesis de grado para la carrera de ingeniería electrónica en la Facultad de Ingeniería de la Universidad de Buenos Aires, se diseñó e implementó la interfaz de un banco de pruebas de motores eléctricos. El trabajo vincula diversos conceptos de ingeniería electrónica, desde la gestión de proyectos hasta la programación de software y el procesamiento de señales.

La metodología y decisiones de diseño adoptadas fueron orientadas desde el comienzo al uso del banco de ensayos como instrumento de laboratorio. Las contribuciones de los interesados al diseño del instrumento fueron fundamentales para asegurar que el banco efectivamente recibirá el uso para el que fue diseñado y pueda ser operado con facilidad.

Es una deficiencia de muchos sistemas de control industrial la dependencia de estándares, protocolos e interfaces no interoperables que fueron innovadores en su época, pero no sobrevivieron el paso del tiempo. Esto hace que el sistema sea cada vez menos confiable y costoso de mantener.

En este sentido, el software se proyectó de forma tal que pueda persistir en tiempo y entre plataformas. La programación simple y estandarizada que gobierna la lógica del banco de ensayos lo vuelve flexible a adaptaciones y nuevas funcionalidades.

La adquisición y manipulación de datos es una característica indispensable de cualquier instrumento de medición. Todas las mediciones expuestas a lo largo de esta tesis de ingeniería fueron recolectadas utilizando el sistema diseñado para tal fin. De esta manera, la experiencia de utilizar el sistema de adquisición de datos para generar las figuras en este documento, permitió su refinamiento.

El método científico como herramienta para construir conocimiento define la repetibilidad de las mediciones como calidad central de un ensayo de laboratorio. Por extensión, esta característica debe ser compartida por las condiciones iniciales del ensayo, sus procesos y por los instrumentos utilizados para realizar mediciones.

Por este motivo, el banco de ensayos debe ser capaz de producir mediciones consistentes con una varianza acotada. Para poder asegurar esto, fueron diseñadas pruebas de validación.

Durante estas pruebas se evaluó la respuesta del instrumento frente a distintos escenarios. Con los resultados de estas pruebas se puede asegurar que el instrumento genera mediciones confiables y consistentes; en todas las métricas evaluadas el error del instrumento fue repetible y acotado. Los resultados obtenidos siguen el modelo teórico con un grado alto de fiabilidad.

Una característica accesoria de un instrumento de medición es su facilidad de uso. Esto lo hará susceptible de ser empleado correcta y frecuentemente. Este punto se encuentra fuertemente ligado a la interfaz de usuario.

Para abordar esto, cada comportamiento de la interfaz de usuario fue consultado y aprobado por los futuros operadores del banco de ensayos. Como resultado de esto, se obtuvo una interfaz lo suficientemente simple como para ser intuitiva, aún admitiendo ensayos de cierta complejidad.

Adicionalmente, el instrumento se utilizó exitosamente para replicar los resultados obtenidos por un grupo independiente de investigadores. Esto es un indicio del potencial del banco de ensayos como un novedoso y versátil instrumento de medición en el ámbito de la FIUBA.

Independientemente del objetivo de este documento, es la intención de su autor que el equipo esté disponible para brindar servicio a todo aquél que desee utilizarlo. En este sentido, se documentó el desarrollo, confeccionando planos eléctricos y manuales, que complementan lo expuesto en estas páginas.

## 13. Desarrollos a futuro

A medida que las distintas etapas del banco de ensayo fueron siendo implementadas, fue posible imaginar funcionalidades que permiten extender el campo de aplicaciones del instrumento. Sin embargo, su implementación implica analizar su factibilidad con una rigurosidad que excede el objetivo de este trabajo. Se reserva esta sección para enumerar estas funcionalidades y señalar algunas consideraciones sobre su posible implementación a futuro.

### 13.1. Ensayos de torque en función de la velocidad con coeficientes no constantes

Si los coeficientes de la ecuación de torque no fuesen constantes y permitiesen variaciones en función del tiempo, de la velocidad o de la posición se podrían simular un gran número de condiciones de carga. A continuación, se hará una breve reseña de posibles aplicaciones.

#### 13.1.1. Coeficientes de la ecuación torque variables con la posición

Aplicando el banco a la locomoción, permitir variaciones en los parámetros de la ecuación torque en función de la posición habilitaría la simulación de trayectos con pendientes y coeficientes de rozamiento variantes según la distancia recorrida. Esto permitiría la simulación de terrenos específicos, con posibles aplicaciones en diseño de motores para vehículos terrestres y simulación de pistas.

El torque resistente en función de la distancia recorrida también permitiría explorar aplicaciones en el área del deporte de alto rendimiento y la rehabilitación. Los entrenamientos con peso convencionales usualmente consisten en aplicar el torque producido por un juego de pesas en suspensión y un sistema de poleas a un movimiento específico. Esto implica que el movimiento a desarrollar siempre tendrá torque constante, sin importar la anatomía de la persona que lo desarrolla.

Al admitir carga variable según el recorrido del movimiento, es posible aplicar precisamente la fuerza y evitar la sobrecarga muscular en rangos de movimiento susceptibles a lesiones o ya

lesionados. La lectura de datos del PLC permitiría conformar un perfil de cada atleta o paciente con datos anatómicos y mediciones de progreso.

### 13.1.2. Coeficientes de la ecuación torque variables con una variable externa

En aplicaciones que requieran realizar movimiento rotativo utilizando ejes largos, es necesario considerar la elasticidad del eje en la dinámica del movimiento. Al frenar la planta motriz con un eje elástico, a la inercia de la planta se le sumara la energía elástica almacenada en el eje. Si el motor que lo impulsa frenase rígidamente, entonces es posible que el motor o el sistema de transmisión sufran daños.

La inclusión de un término que modele la elasticidad del eje permitiría explorar nuevas técnicas de arranque y frenado de plantas motrices elásticas. Sin embargo, la energía elástica acumulada varía en función del material con el que se construyó el eje, su largo y el ángulo de torsión entre un extremo del eje y el otro.

Adicionar el término al banco de ensayos implicaría poder variar la cupla ejercida sobre el eje en función del ángulo de torsión. Para realizar ello, es necesario obtener el ángulo de torsión de manera independiente.

En la industria de la extracción de combustibles fósiles, es común que se requiera operar herramientas de gran longitud con una carga elevada en uno de sus extremos. Este tipo de ensayos podría ser de utilidad para ensayar técnicas de control para este tipo de maquinaria.

## 13.2. Mejoras en el término inercial de la ecuación cupla-velocidad

Si bien el tiempo de escaneo actual resultó adecuado para los fines del presente trabajo, es posible reducir aún más el tiempo de escaneo ubicando los programas desarrollados en otro nivel de ejecución del sistema de ejecución SIMOTION. Sin embargo, es posible que llevar a cabo esto implique la reescritura o refactorización del código desarrollado.

Para incrementar la estabilidad del término inercial, sería posible ubicar el código que procesa la ecuación torque en un nivel de ejecución síncrono con el ciclo de pulsos del servomotor, permitiendo ejecutar el código con un período mínimo de 250us. Esto permitiría reducir el retardo al aplicar la cupla senoidal, incrementando la fiabilidad del sistema.

Otro enfoque posible sería el de explorar técnicas de procesamiento de señales distintas a las evaluadas para suavizar la medición de la velocidad o su derivada. Algunos enfoques posibles son: la transformada wavelet, filtros adaptativos, técnicas de filtrado basadas en machine learning, entre otras.

### 13.3. Adición de un encoder externo

A lo largo del desarrollo se observaron grandes variaciones en la velocidad medida por el resolver del servomotor utilizado. El instrumento tiene una resolución de 0,067° o capacidad para alrededor de 5.400 posiciones por revolución.

Este instrumento, al ser nativamente analógico, necesita de una etapa de conversión analógico-digital para ser leído por el sistema de control. Al realizar ello, inevitablemente se introduce un error de redondeo que varía en función de la cantidad de bits disponibles para la conversión. Adicionalmente, esta conversión introduce cierto retardo en el sistema.

Por este motivo, fue necesario considerar la dispersión en la medición de velocidad a la hora de implementar el término derivativo de la ecuación torque-velocidad, obligando a utilizar un filtro promediador. Este error en la medición limita el rango posible del término D de la ecuación.

Considerando que en el mercado hay disponibles encoders ópticos, que son digitales y con más de 500.000 posiciones por revolución, sería posible mejorar ampliamente las mediciones de

velocidad del sistema agregando este instrumento al banco de ensayos. Sin embargo, además de la inversión inicial, será necesario adaptar el código a la utilización del encoder externo e incorporarlo mecánicamente al eje del servomotor. Esta modificación excede el objetivo de esta tesis, pero puede ser considerada para su implementación a futuro.

### 13.4. Aumento de la capacidad de simulación de cupla

El banco de ensayos cuenta con otro servomotor de iguales características al utilizado como eje de simulación de carga. Si fuese posible acoplar este motor mecánicamente al eje simulador y lograr que ambos ejecuten los mismos comandos de torque, entonces la capacidad de simulación de cupla prácticamente se duplicaría.

Sin embargo, realizar esto no es simple: el sincronismo de comandos debe ser preciso para poder asegurar la correcta operación del banco de ensayos. Como el dispositivo originalmente fue pensado para tareas de posicionamiento de precisión, es posible sincronizar ejes utilizando bibliotecas disponibles para este fin.

Otro mejora posible, podría ser la adición de un volante de inercia al eje de simulador. Esta modificación podría incluirse en aplicaciones donde se requiera simular valores de inercia que excedan los límites del banco de ensayos.

En ambos casos, es necesario considerar un nuevo sistema mecánico con una inercia de rotor diferente en función de cómo se implemente el acople mecánico. El cambio en el sistema mecánico podría incorporar nuevos puntos de inestabilidad, así como también efectos mecánicos como la holgura.

### 13.5. Aumento de la capacidad de simulación velocidad

Estrechamente ligado al apartado anterior, sería posible aumentar la capacidad de simulación del banco de ensayos incluyendo una caja reductora en el eje simulador.

Los motores eléctricos suelen optimizar la utilización de energía a fin de ofrecer una mayor autonomía. Por este motivo, sus motores suelen operar en la región de máxima eficiencia. Usualmente esto ocurre a altas velocidades.

Como el banco de ensayos tiene una velocidad límite de 3000 RPM, no podría acoplarse directamente al sistema de tracción para su ensayo.

Frente a ello, una alternativa posible es realizar un escalamiento de la velocidad. Este tipo de técnicas supone un comportamiento lineal del motor, lo que puede arrojar resultados imprecisos.

Se sugiere como posibilidad incluir una caja reductora que pueda ser incorporada para realizar ensayos de alta velocidad. De igual manera a lo que fue mencionado en la sección anterior, esta incorporación requiere de un análisis mecánico de estabilidad.

### 13.6. Interfaz web

En un principio se propuso desarrollar una interfaz Web en vez de una interfaz basada en un programa de escritorio. Esta idea se abandonó en pos de poder avanzar rápidamente y concentrar los esfuerzos para poder presentar un banco de pruebas completamente funcional.

La interfaz web tiene como ventaja la interoperatividad entre sistemas. Esto haría posible que, luego de configurar una red interna, los ensayos podrían ser ejecutados desde un dispositivo móvil o incluso remotamente. Esta adición implicaría un agregado de valor importante al banco de ensayo.

La ejecución de comandos de manera remota requiere de la implementación de medidas de seguridad adicionales y de un sistema control de acceso al software con una rigurosidad que excede ampliamente el objetivo de este trabajo. Por este motivo, se considera como un desarrollo a futuro.

## 14. Bibliografía

### Referencias

- [1] J. L. James Larminie, *Electric Vehicle Technology Explained*. Wiley, 2003.
- [2] *1FK7 Synchronous Motors SINAMICS S120*, Siemens, 2006.
- [3] *Simotion D4x5 Manual*, Siemens, 2007.
- [4] N. Mohan, *Power electronics : a first course*. Wiley, 2012.
- [5] *SIMOTION Structured Text*, Siemens, 2012.
- [6] *SINAMICS S120 - Booksized power sections*, Siemens, 2012.
- [7] *SINAMICS: Etapas de potencia booksized*, Siemens, 2012.
- [8] A. Hughes y B. Drury, *Electric Motors and Drives : Fundamentals, Types and Applications*. Newnes, 2013.
- [9] S. Walker, “Barriers to the Deployment of a 100 MW Tidal Energy Array in the UK,” *International Journal of Energy Engineering*, 2013.
- [10] *SINAMICS S120 Parameter List*, Siemens, 2014.
- [11] K. T. Chau, *Electric Vehicle Machines and Drives Design, Analysis and Application*. Wiley, 2015.
- [12] *SIMOTION LAD/FBD Programming and Operating Manual*, Siemens, 2015.
- [13] *Simotion IT OPC UA Programming Manual*, Siemens, 2016.
- [14] *Simotion Motion Control SIMOTION SCOUT*, Siemens, 2016.
- [15] M. Braun y W. Horn, *Object-oriented programming with SIMOTION : fundamentals, program examples and software concepts according to IEC 61131-3*. Publicis, 2017.
- [16] *Object Oriented Programming with Simotion*, Siemens, 2018.
- [17] *SINAMICS S120 web server -Creating user-defined web pages*, Siemens, 2018.
- [18] E. Vinson, A. Jurado y J. Sotelo, “Plataforma de Simulación Experimental Para Plantas Motrices de Vehículos eléctricos,” *Congreso Internacional de Distribución Eléctrica (CIDE)*, 2018.
- [19] *Simotion D4x5-2 Commissioning and Hardware Installation Manual*. Siemens Ag, Munich, Siemens, 2019.
- [20] J. Sotelo, “Simulación de cuplas motoras y resistentes con máquinas sincrónicas de imanes permanentes controladas,” *Tesis de grado en ingeniería eléctrica - Departamento de Energía - FIUBA*, 2019.
- [21] F. P. Carmona, J. J. R. Rivaz, R. O. González, O. C. Castillo, D. M. Garduño y E. P. Sánchez, “Electric Traction Emulator System,” *IEEE Access*, 2025.

## 15. Anexo

Esta sección recopila todos los documentos creados para poder documentar el funcionamiento del equipo.

### 15.1. Manual de operación

---

# **Manual de operador del banco de ensayos de motores eléctricos**

---

*Juan Hirschmann (jhirschmann@fi.uba.ar)*

Departamento de Energía FIUBA

## **Resumen**

El presente manual de operador pretende ser una guía para la utilización del banco de ensayos de motores eléctricos. Además de guiar al operador para la realización de un ensayo, se describe el funcionamiento del banco de ensayos, sus operaciones de mantenimiento y la posibilidad de añadir nuevas funcionalidades.

Este manual proporciona una guía para llevar a cabo distintos tipos de ensayos en el banco de prueba, junto con instrucciones paso a paso para llevar a cabo tareas de mantenimiento y resolución de posibles problemas.

Para información sobre el diseño del banco de ensayos y su funcionamiento, referirse a la tesis de ingeniería electrónica adjunta como documento complementario.

## Índice

<b>1. Registro de cambios</b>	<b>2</b>
<b>2. Información de seguridad</b>	<b>4</b>
<b>3. Introducción</b>	<b>4</b>
3.1. Componentes físicos del sistema . . . . .	5
3.1.1. Interfaz física . . . . .	5
3.1.2. Sistema de accionamientos . . . . .	6
3.2. Causas de emergencia en el sistema . . . . .	7
<b>4. Descripción del sistema</b>	<b>8</b>
4.1. Control de flujo de programa . . . . .	9
4.2. Descripción del Hardware . . . . .	11
4.2.1. Placa de E/S . . . . .	11
<b>5. Descripción de prestaciones</b>	<b>13</b>
5.1. Ensayos autocontenidos . . . . .	14
<b>6. Instalación</b>	<b>14</b>
6.1. Instalación eléctrica . . . . .	14
6.2. Instalación de software de operación . . . . .	14
6.2.1. Instalación de software - máquina virtual . . . . .	14
6.2.2. Instalación de software - instalador . . . . .	14
6.2.3. Configuración IP . . . . .	16
<b>7. Realización de un ensayo</b>	<b>16</b>
7.1. Común a todo tipo de ensayo . . . . .	17
7.2. Ensayo de torque-tiempo . . . . .	20
7.3. Ensayo tipo torque-velocidad . . . . .	23
7.4. Ensayo mixto . . . . .	24
7.5. Ensayo autocontenido . . . . .	24
<b>8. Resolución de problemas</b>	<b>26</b>
8.1. Indicadores LED de la unidad de control . . . . .	26
8.2. Procedimiento de carga de software . . . . .	29
8.3. Verificación de señales de LADDER . . . . .	32
8.4. Cambio de dirección IP del control . . . . .	32
8.5. Registro de modificaciones . . . . .	33
<b>9. Plano eléctrico del sistema</b>	<b>34</b>

## 1. Registro de cambios

Esta sección se reserva para futuras modificaciones en el presente manual.

Fecha	Versión	Descripción	Autor
11/01/2025	1.0	Creación del documento	J.Hirschmann

## 2. Información de seguridad

**⚠ Durante la operación del banco de ensayos estará expuesto a movimiento rotacional de alto torque. Utilice calzado de seguridad y protectores oculares para operar el banco de ensayos. No utilice vestimenta holgada ni bolsos o mochilas en la cercanía del eje de simulación. Evite accidentes.**

**⚠ Previo a energizar el banco de ensayos, realice una inspección de la unidad de accionamientos. Corrobore el estado de los cables de alimentación y señal. Verifique que el eje de simulación se encuentra mecánicamente libre y cubierto por su jaula de seguridad.**

**⚠ No opere el banco de ensayos si observa que los cables del dispositivo se encuentran desconectados o con su vaina expuesta.**

**⚠ Evite entrar en contacto con el gabinete de accionamientos durante la realización de un ensayo o cuando este se encuentre energizado. Si es necesario realizar tareas de mantenimiento en la unidad, utilice el interruptor que se encuentra del lado derecho para interrumpir el suministro eléctrico.**

**⚠ Durante la operación del banco esté preparado para accionar el freno de emergencia ante cualquier situación imprevista. No abandone la estación de operador ni se aleje de la botonera de operación.**

**⚠ Tras calcular los parámetros de ensayo, realice ensayos de prueba: introduzca los parámetros reducidos a su décima parte para asegurar su comportamiento esperado. Monitoree los valores de velocidad y cupla alcanzados. Tras confirmar que el ensayo de prueba fue exitoso incremente los parámetros de ensayo hasta alcanzar su valor final.**

## 3. Introducción

El banco de ensayos tiene como propósito la emulación de sistemas mecánicos reales. Existen tres tipos de ensayo posibles: aplicación de cupla en función del tiempo, aplicación de cupla en función de la velocidad o la superposición de ambos.

Estos ensayos pueden ser llevados a cabo en dos modalidades. La primera de ellas es ensayando un dispositivo externo, independiente del banco de ensayos. Mientras que la segunda, llamada autocontenido, permite la utilización del segundo motor del banco de ensayos como dispositivo bajo prueba.

Para poder realizar cualquier ensayo, el dispositivo a ensayar debe acoplarse mecánicamente a un eje que forma parte del motor simulador. Se deben configurar las consignas de cupla-velocidad o cupla-tiempo en función del sistema mecánico a ensayar, respetando los límites de simulación del banco de ensayos.

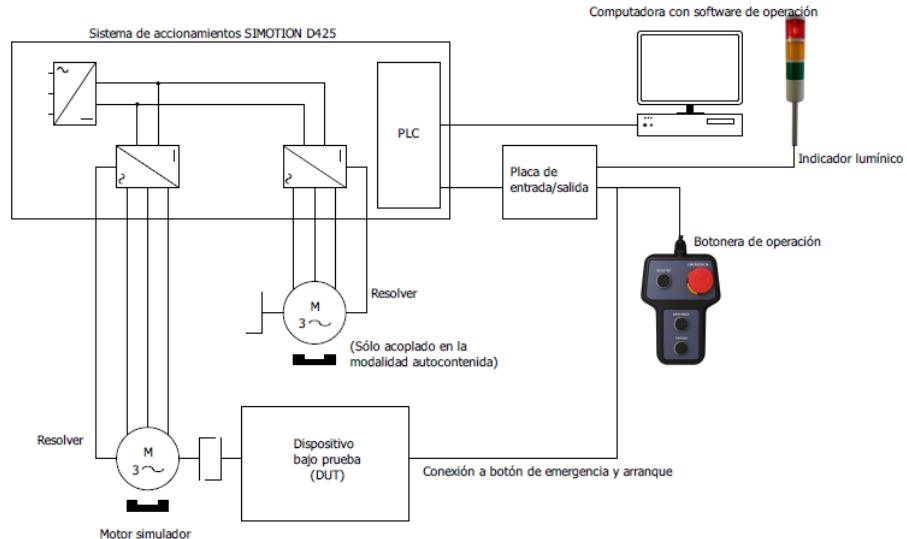


Figura 1: Diagrama en bloques de los componentes del banco de ensayos

El 'eje de simulación' ejecuta las consignas de cupla introducidas simulando diferentes condiciones de carga en el dispositivo bajo prueba. De ser necesaria, la programación del dispositivo bajo prueba debe ser realizada por el usuario en un sistema independiente, aplicando las condiciones de seguridad necesarias.

Al introducir una consigna de cupla resistiva, el banco de ensayos intentará alcanzar 2500 RPM en sentido horario, si la consigna es positiva, y en sentido antihorario, si la consigna es negativa. Si la cupla de tracción es inferior a la resistiva, entonces el eje de simulación domina el movimiento en el eje.

Al mantener esta situación, el eje de simulación mantendrá su consigna de cupla, acelerando hasta 2500 RPM. Una vez alcanzada esta velocidad, el eje de simulación aplicará una cupla inferior a la consigna con el fin de mantener su velocidad.

**⚠️ El eje de simulación está preparado para ofrecer resistencia al movimiento del dispositivo bajo prueba. No intente usar el equipo con otros fines.**

### 3.1. Componentes físicos del sistema

El banco de ensayos consiste en 2 componentes físicos: su interfaz física y el sistema de accionamientos SIMOTION D425.

#### 3.1.1. Interfaz física

La interfaz física esta conformada por un panel de operador y un sistema lumínico:



Figura 2: Elementos de la interfaz física del banco.

El panel de operador será utilizado para realizar intervenciones críticas durante el ensayo, como son: parada de emergencia, arranque, frenado y reconocimiento de emergencia.

En cuanto al sistema de alertas lumínico, indica en qué momentos el eje del equipo está en movimiento o podría iniciar su movimiento al presionar el botón de arranque. La tabla 1 describe las señales:

Luz encendida	Característica	Indicación
Roja	Fija	El ensayo ha finalizado.
Roja	Parpadeante	El sistema está en estado de emergencia.
Amarilla	Fija	El ensayo está pausado.
Amarilla	Parpadeante	El ensayo se está ejecutando.
Verde	Fija	Inicio del programa, aún no se recibieron los parámetros de ensayo.
Verde	Parpadeante	El ensayo está listo para el arranque.

Cuadro 1: Descripción de las señales lumínicas

### 3.1.2. Sistema de accionamientos

El sistema de accionamientos consiste de 4 unidades:

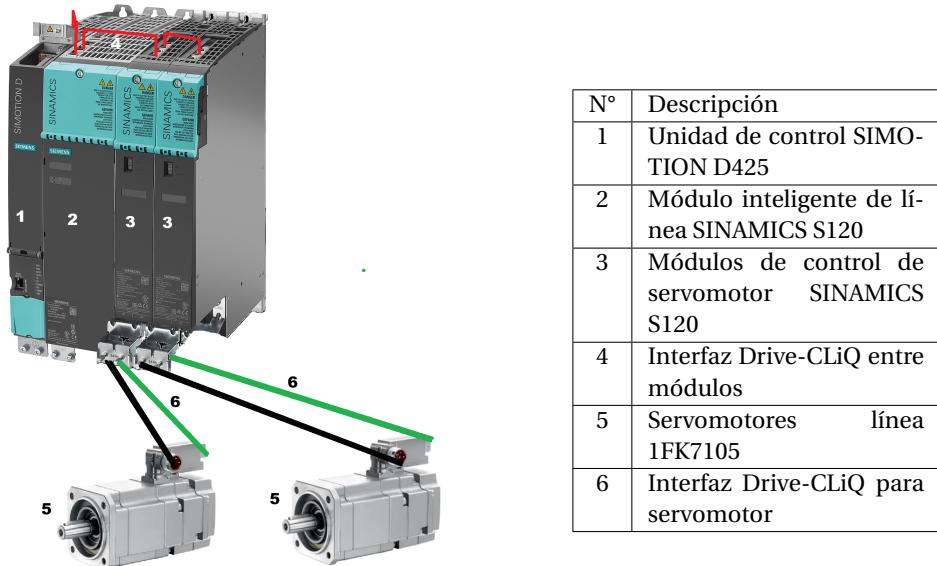


Figura 3: Componentes del sistema

### 3.2. Causas de emergencia en el sistema

El sistema puede entrar en estado de emergencia por las siguientes causas:

- **Accionamiento de botón de emergencia:** el usuario accionó el botón de emergencia en la botonera o en la interfaz gráfica.
- **Desconexión de la botonera:** la botonera no recibe tensión de la fuente externa.
- **Desconexión del control:** el sistema detectó que se perdió la conexión con el sistema de accionamientos.
- **Cupla excesiva:** el sistema detectó el valor de cupla a ejecutar es demasiado grande.
- **Derivada de torque excesiva:** el sistema detectó que la diferencia entre la consigna de cupla actual y la siguiente será demasiado grande.

**Reconocimiento de emergencias** Al producirse una emergencia, el sistema indicará la causa en la interfaz gráfica. Para poder liberar al sistema de la condición de alarma, es necesario pulsar el botón de reseteo en la botonera. Si la condición de emergencia desaparece, la señal lumínica pasará de rojo parpadeante a amarillo. El sistema reanudará su movimiento luego de presionar el botón de arranque.

**⚠️ Previo a reanudar el ensayo verifique que el eje se encuentra liberado y que no haya objetos que podrían resultar atrapados en su proximidad.**

- ⚠️ No desconecte ni cubra la señal lumínica.**
- ⚠️ Verifique la causa de la emergencia en la pantalla previo a liberar la emergencia.**

## 4. Descripción del sistema

A continuación se presenta un diagrama de bloques del sistema:

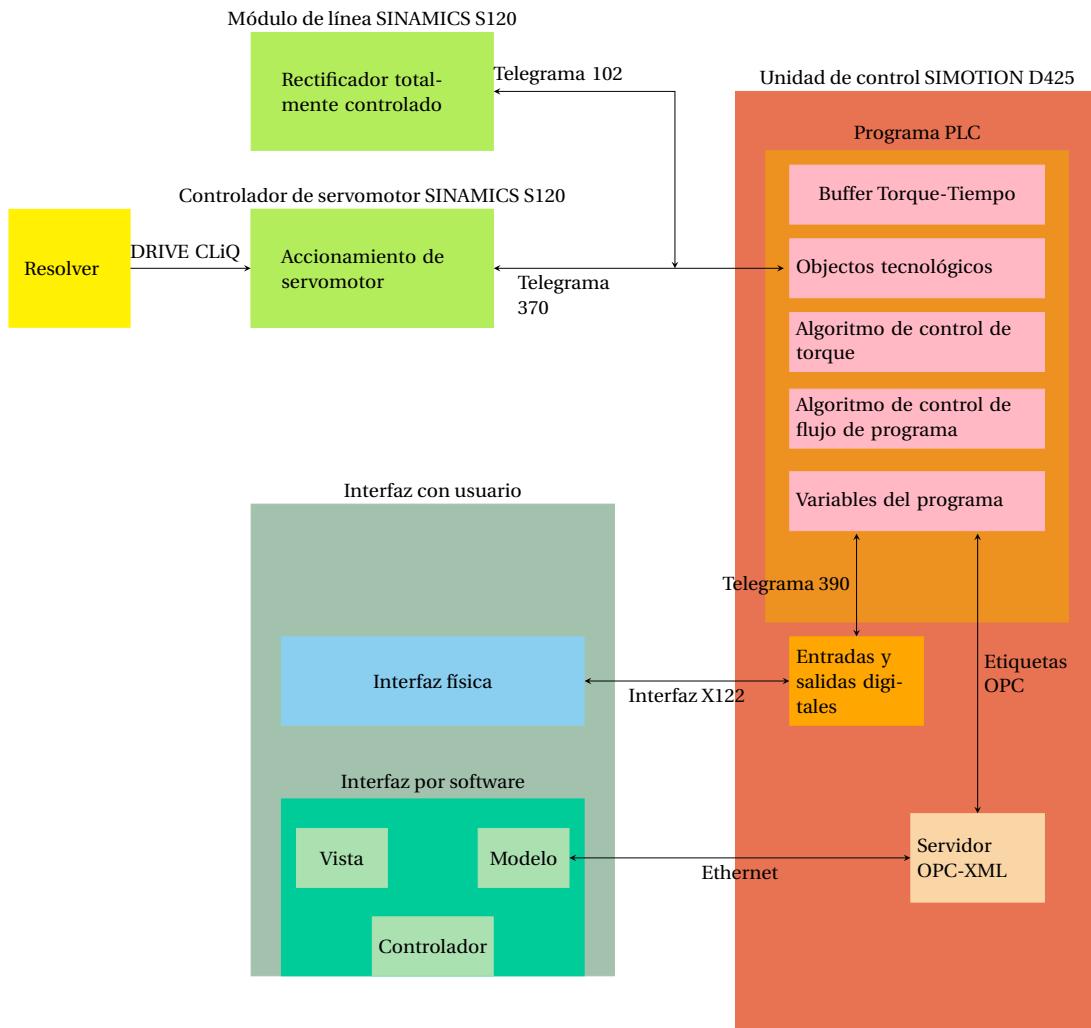


Figura 4: Diagrama en bloques del banco de ensayos

El sistema consiste en un programa que corre dentro del módulo SIMOTION D425, controlando el sistema de accionamientos utilizando objetos tecnológicos. Para interactuar con el programa PLC existen dos maneras posibles: mediante las variables IO, ligadas a puertos físicos en la interfaz X122, o bien a través de las etiquetas OPC.

Las etiquetas OPC son un conjunto de variables configuradas para ser cargadas al servidor OPC-XML. En el servidor OPC, las variables podrán ser consultadas o actualizadas por el cliente OPC-XML que conforma parte del Modelo. El modelo recibe instrucciones del Controlador de la interfaz que a su vez procesa los comandos de usuario informados por la Vista.

Para más información acerca de la implementación del banco de ensayos, referirse a la tesis de ingeniería electrónica que complementa este manual.

#### 4.1. Control de flujo de programa

El diseño del sistema cuenta con 5 estados básicos. A continuación, una breve descripción de su comportamiento:

- No iniciado
  - Comportamiento: Es el estado inicial del sistema. Enciende la señal lumínica de verde. Gestiona la carga de parámetros de ensayo.
- Ensayo listo
  - Comportamiento: Enciende la señal lumínica en verde de a intervalos de un segundo.
- En ejecución
  - Comportamiento: enciende la señal lumínica en amarillo de a intervalos de un segundo parpadeante. Inicia o reanuda el movimiento del eje.
- Frenado
  - Comportamiento: enciende la señal lumínica de amarillo. Detiene el movimiento del eje.
- Emergencia
  - Comportamiento: enciende la señal lumínica de rojo parpadeante. Detiene el movimiento del eje.
- Finalizado
  - Comportamiento: enciende la señal lumínica de rojo. Detiene el movimiento del eje.

Estos estados solo admiten un conjunto de transiciones posibles. Este conjunto se detalla en su diagrama de transición de estados:

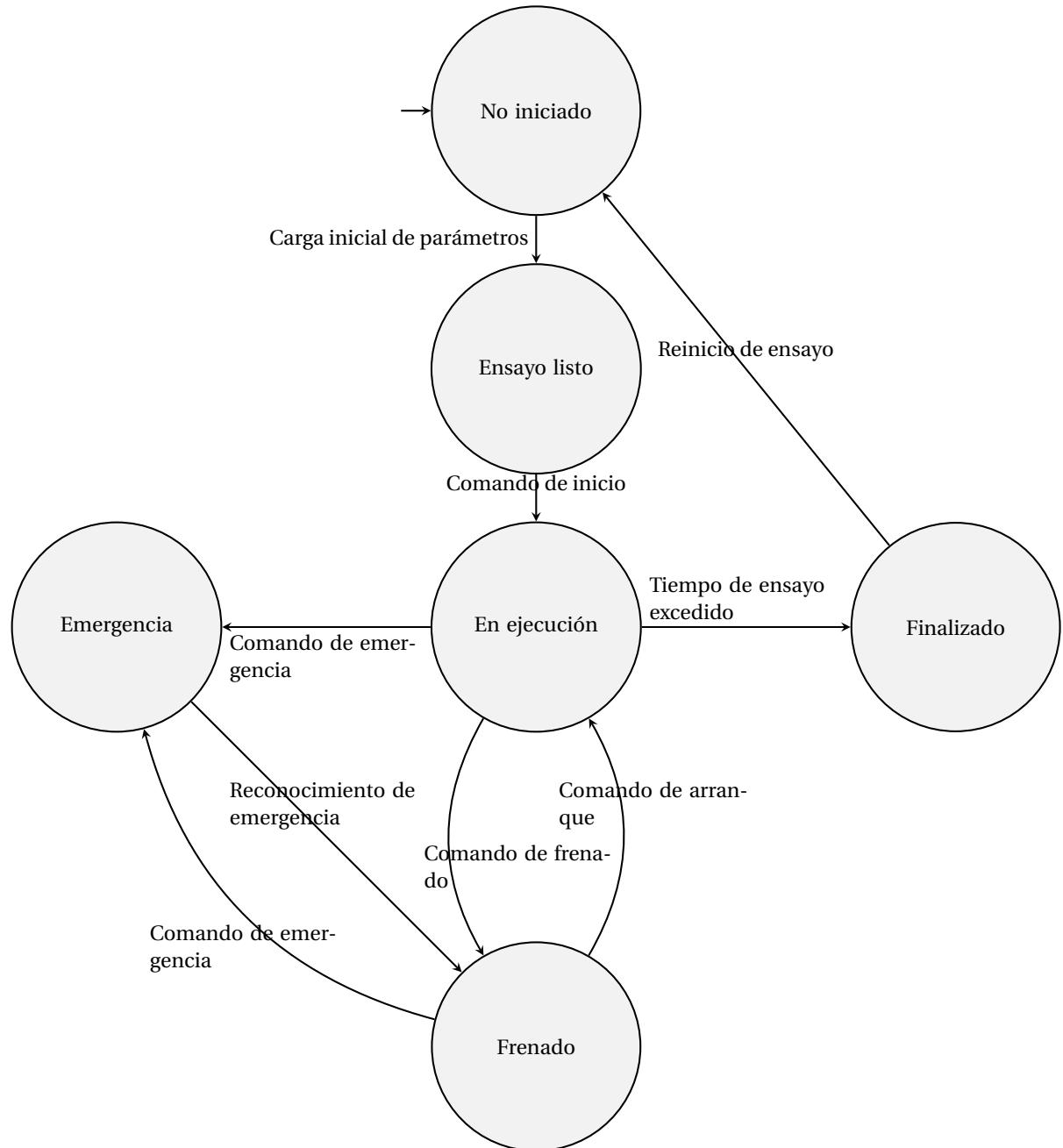


Figura 5: Diagrama de transiciones de estado en el programa de control de flujo

Los eventos de transición requieren de ciertas precondiciones para ser ejecutados. En la tabla 2 se des-

criben las precondiciones de ejecución:

Evento	Precondición
Carga inicial de parámetros	Tipo de ensayo especificado
	Parámetros de ensayo recibidos
	Duración de ensayo especificada
Comando de inicio	Modulo de línea activado
	Eje habilitado
	Botón de arranque pulsado
	Botón de arranque por software pulsado
	Ausencia de comando de emergencia
Comando de arranque	Botón de arranque pulsado
	Botón de arranque por software pulsado
	Ausencia de comando de emergencia
Comando de frenado	Botón de frenado pulsado
	Botón de frenado por software pulsado
Comando de emergencia	Botón de emergencia pulsado
	Botón de emergencia por software pulsado
	Temporizador de <i>keepalive</i> excedido
	Desconexión de interfaz física
	Error de hardware
Reconocimiento de emergencia	Botón de reseteo pulsado
	Ausencia de comando de emergencia
Tiempo de ensayo excedido	El tiempo transcurrido en estado RUNNING excede el tiempo de ensayo prefijado
Reinicio de ensayo	Apagado de la máquina

Cuadro 2: Tabla de eventos de transición en el programa de control de flujo

## 4.2. Descripción del Hardware

### 4.2.1. Placa de E/S

Además del sistema de accionamientos en la figura 3 el hardware del sistema cuenta con una placa de entradas y salidas.

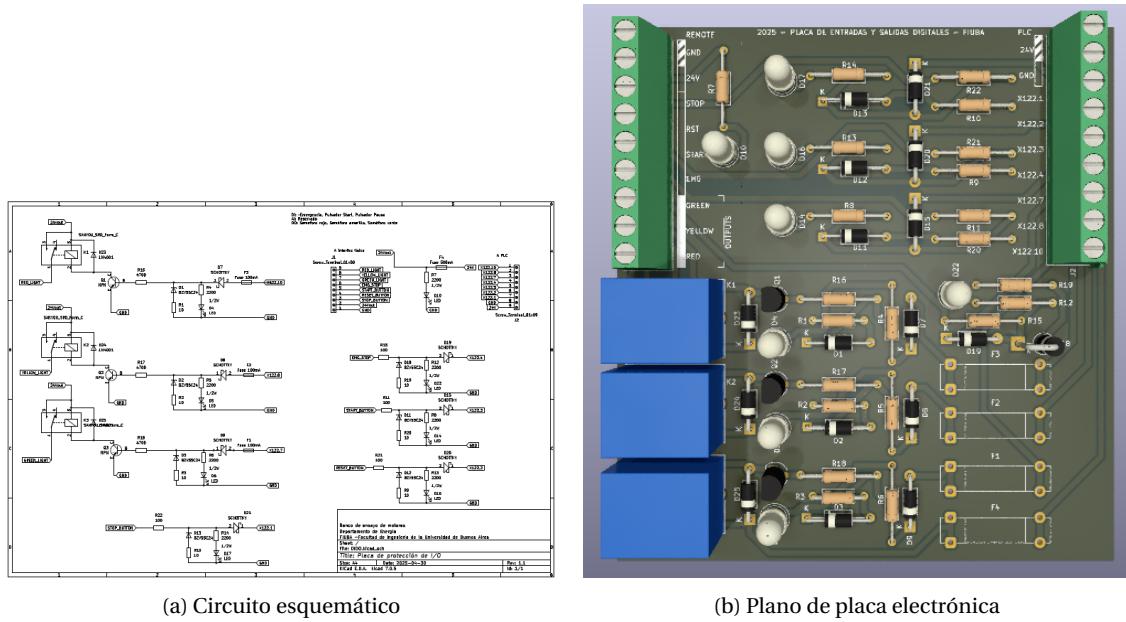


Figura 6: Diagramas de diseño de la placa en la interfáz física

La placa se encuentra conectada al puerto puerto X122 de la unidad de control. En su otro extremo, la placa se conecta con los botones de arranque, reseteo, frenado y emergencia, con la fuente de alimentación externa y con el indicador luminoso.

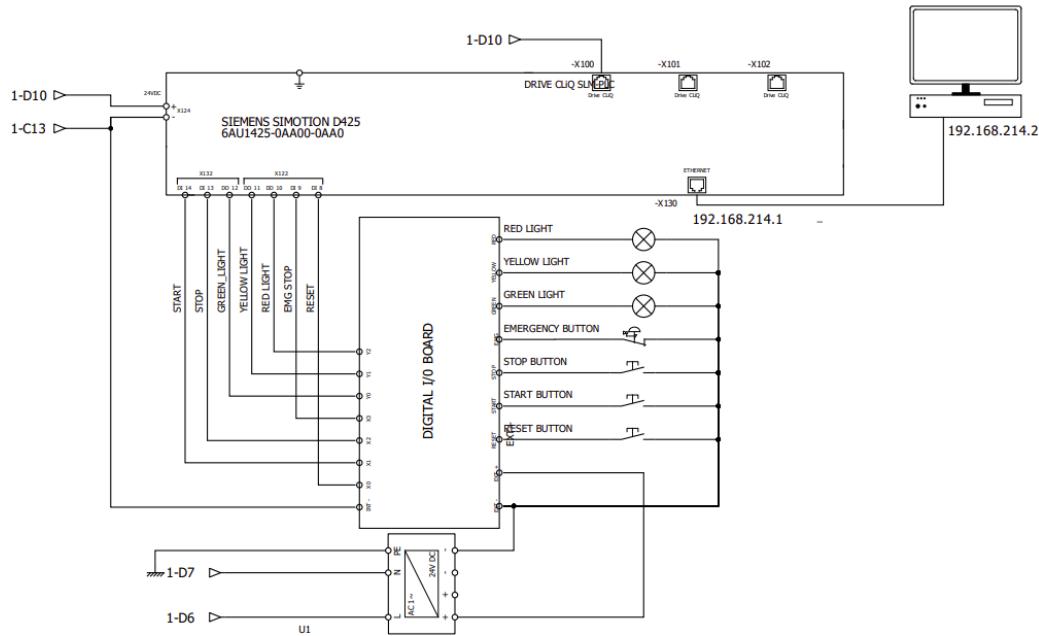


Figura 7: Detalle de la conexión de E/S

La placa incluye LEDs de verificación de estado que permiten corroborar rápidamente el estado de la alimentación, de los pulsadores y de las salidas.

## 5. Descripción de prestaciones

El banco de ensayos permite la introducción de consignas de cupla en tres régimenes distintos:

1. Cupla en función del tiempo: la consigna de cupla sigue una tabla de pares torque-tiempo cargada por el usuario
2. Cupla en función de la velocidad: la consigna de cupla sigue la ecuación  $T(\omega) = A + B\omega + C\omega^2 + D\frac{d\omega}{dt}$ , en donde el usuario introduce los parámetros A, B, C y D y  $\omega$  es la velocidad angular del motor ensayado.
3. Cupla en función del tiempo y la velocidad: la cupla es la sumatoria de los valores de la tabla torque-tiempo y la ecuación torque-velocidad.

Los datos del ensayo son recolectados y representados gráficamente en tiempo real con una resolución temporal de aproximadamente 150ms. Luego de finalizar un ensayo, el operador puede exportar los datos recolectados en formato CSV.

## 5.1. Ensayos autocontenidos

Adicionalmente, es posible llevar a cabo los tres régimenes mencionados en una modalidad 'autocontenido'. En esta modalidad, el eje simulador debe ser acoplado mecánicamente al eje vacante del sistema de accionamientos.

Este modo de operación permite introducir consignas de velocidad-tiempo en el eje vacante para poder utilizarlo como dispositivo bajo prueba. El eje de simulación responderá con su consigna de cupla dependiendo del tipo de ensayo seleccionado.

El ensayo autocontenido tiene como finalidad poder probar perfiles de velocidad-tiempo y su respuesta en el eje de simulación. Esto puede ser útil para programar el dispositivo que finalmente sea ensayado.

# 6. Instalación

## 6.1. Instalación eléctrica

El banco de ensayo se encuentra instalado en un gabinete eléctrico siguiendo el cableado expuesto en el plano eléctrico, en la sección 9. Si es necesario reemplazar el gabinete eléctrico, consultar el manual 'SIMOTION D425 commissioning and hardware installation manual'. Será necesario considerar la temperatura de trabajo, capacidad de transmisión de calor de un nuevo gabinete y la correcta puesta a tierra de las unidades.

## 6.2. Instalación de software de operación

El software de operación del banco puede obtenerse de dos maneras distintas. Una primera alternativa es utilizando una máquina virtual que ya contiene el software instalado y todo lo necesario para su uso. Esta opción permite que el usuario no modifique ciertas configuraciones de red en su computadora principal.

El segundo método es mediante un instalador de software, sin la necesidad de una máquina virtual. Este método tiene la ventaja de consumir una menor cantidad de recursos, ya que no requiere de la emulación de otra máquina.

### 6.2.1. Instalación de software - máquina virtual

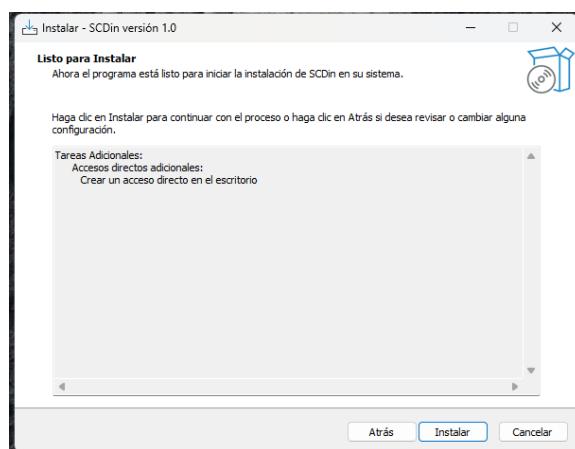
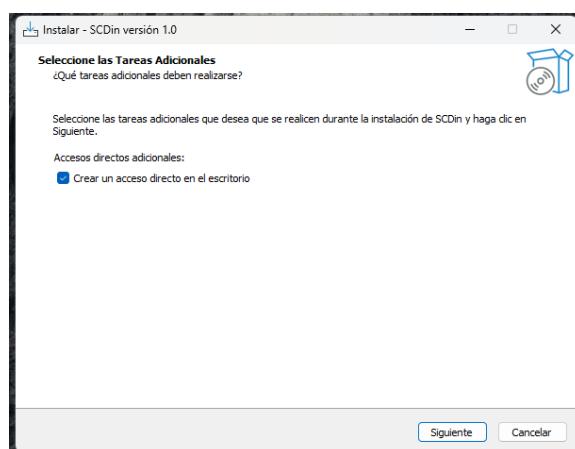
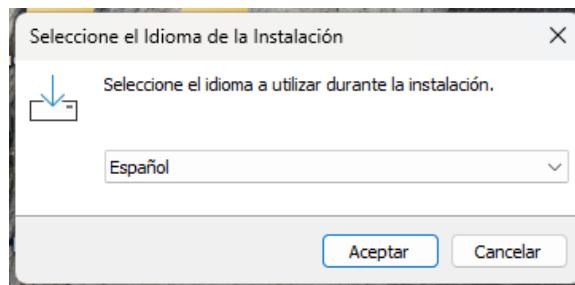
El software de operación del banco viene incluido en una máquina virtual con el nombre 'SCDIN'. Su instalación consiste en la configuración de la máquina virtual para poder adecuarla a una nueva computadora anfitriona.

### 6.2.2. Instalación de software - instalador

El software puede ser instalado en un sistema operativo usando el instalador 'SCDIn - Setup.exe'. El software requiere de la instalación de la plataforma JAVA SE JDK21 o superiores<sup>1</sup>.

A continuación, se presentan las distintas pantallas del instalador:

<sup>1</sup>Sitio web para descargas de JDK 21: <https://www.oracle.com/java/technologies/downloads/>





Al haber finalizado la instalación, se ejecutará el software SCDin. Si al abrir el software aparece una ventana con la leyenda 'This application requires a Java Runtime Environment (JRE)', significa que la aplicación no encontró la máquina virtual de JAVA instalada en la computadora.

Si efectivamente fue instalado el software JDK, entonces será necesario modificar las variables de entorno del sistema operativo, en particular la variable PATH<sup>2</sup>. La modificación consiste en agregar el directorio con los archivos ejecutables de JDK en la variable PATH para que el sistema operativo reconozca los comandos JAVA. En Windows, JDK típicamente se instala en el directorio 'C:/Program Files/Java/jdk<versión instalada>/bin'.

#### 6.2.3. Configuración IP

Es necesario configurar la computadora con la que se correrá la interfaz gráfica para que pertenezca a la misma red que la unidad de control. El puerto Ethernet (X130) de la unidad de control tiene la dirección IP 192.168.214.1. Entonces la computadora debe configurarse con una dirección IP en el mismo rango.

En Windows esto puede realizarse en 'Configuración de Ethernet'; si la red fuese solamente de dos estaciones una configuración válida sería:

- Dirección IPv4: 192.168.214.2
- Máscara IPv4: 255.255.255.0
- Puerta de enlace IPv4: 192.168.214.3

## 7. Realización de un ensayo

En esta sección se describirá cómo preparar el sistema para realizar un ensayo sobre un motor.

<sup>2</sup>Para más información consultar el siguiente enlace: <https://www.java.com/en/download/help/path.html>

## 7.1. Común a todo tipo de ensayo

Previo a realizar cualquier tipo de ensayo que no sea bajo la modalidad autocontenido, el dispositivo bajo prueba debe ser programado para seguir una consigna en un sistema de accionamiento propio, independiente del SIMOTION D425. Esta programación excede el objetivo de este texto, pero deben considerarse todas las medidas de seguridad correspondientes. Entre ellas, se debe conectar la entrada de la parada de emergencia del dispositivo bajo prueba al pulsador de parada de emergencia en la botonera del equipo. Esto debe hacerse con ambos equipos desenergizados.

**⚠ Si las paradas de emergencia no se encuentran correctamente cableadas, el motor bajo prueba no detendrá su movimiento cuando se accione la emergencia. Esto puede llevar a accidentes**

**⚠ Asegúrese de que el equipo se encuentre desenergizado previo a realizar el acople del motor bajo prueba. Luego de acoplar el motor al eje de simulación, cubra el acople con la jaula de seguridad.**

**⚠ Previo a realizar el ensayo asegúrese de que el eje este cubierto con la jaula de protección. Evite el uso de prendas sueltas.**

Energice el sistema de accionamientos girando la perilla en su extremo derecho. El sistema ahora se encuentra listo para conectarse con el software.

En la computadora conectada al sistema de accionamientos se debe abrir el programa SCDin.



Figura 8: Icono de escritorio del software SCDin

Al realizar esto, se iniciará el programa en su pantalla inicial:

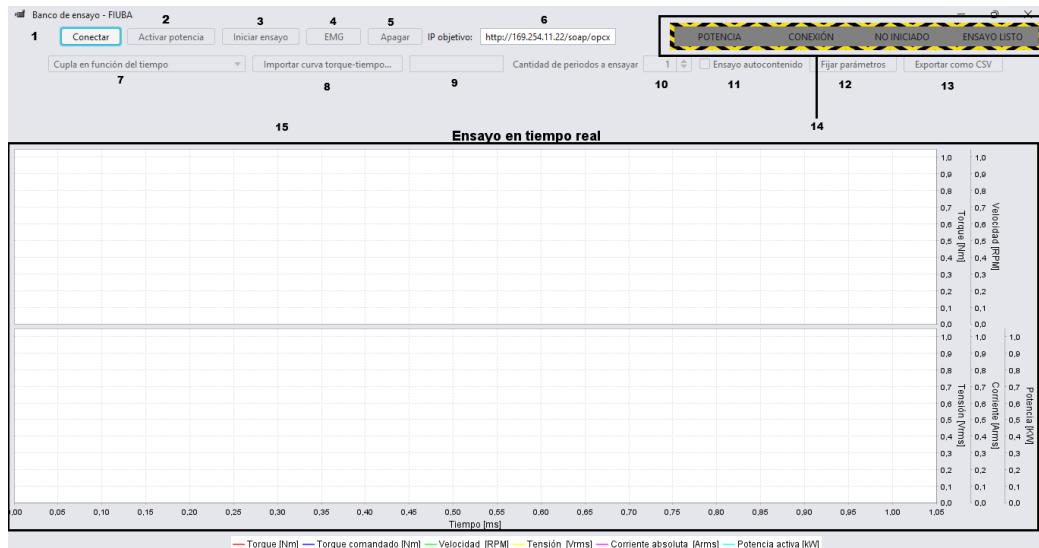


Figura 9: Pantalla incial del programa

La pantalla principal contiene los siguientes elementos:

Componente	Descripción
1	Botón de conexión
2	Botón de activación de circuito de potencia
3	Botón de inicio de ensayo
4	Botón de activación de emergencia
5	Botón de apagado del sistema de accionamientos
6	Campo de texto para introducir la dirección del servidor OPC-XML DA
7	Selector de tipo de ensayo
8	Botón para importar la curva de cupla-tiempo en formato CSV
9	Campo de texto con el nombre de archivo de la curva de cupla-tiempo seleccionada
10	Cantidad de periodos a ejecutar de la curva cupla-tiempo seleccionada
11	Selector de ensayo autocontenido
12	Botón para fijar los parámetros del ensayo
13	Botón para exportar las mediciones realizadas en formato CSV
14	Panel de indicadores. De izquierda a derecha: activación de circuito de potencia, estado de la conexión con el sistema de accionamientos, estado actual del ensayo, estado de la carga de parámetros.
15	Gráfico de las mediciones de ensayo en tiempo real

Cuadro 3: Elementos de la pantalla principal

Una vez en la pantalla inicial, se deberá cliquear el botón 'Conectar'. Si la conexión no es exitosa, una alerta con el mensaje 'El control no está conectado. Verifique la configuración IP' se mostrará en pantalla. Si la alerta se dispara, referirse a la sección 8

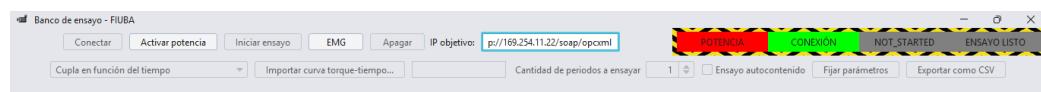


Figura 10: Franja superior del programa cuando la conexión es exitosa: en el panel de indicadores se señala el estado de conexión activa y de circuito de potencia inactivo

Al haber establecido la conexión con el control, se habilita el botón 'Activar potencia'. Tras hacer click se debería escuchar el encendido del módulo de línea y, 5 segundos después, la activación del accionamiento de servomotor.



Figura 11: Franja superior del programa cuando la conexión es exitosa: en el panel de indicadores se señala el estado de conexión activa y de circuito de potencia activo

**⚠️ Previo a activar el circuito de potencia del equipo verifique que la puerta del gabinete se encuentra cerrada. Existe riesgo de electrocución**

En esta instancia el sistema está listo para procesar los datos del ensayo. En las secciones a continuación, se detallan las instrucciones para poder realizar todos los tipos de ensayo posibles.

**⚠️ Si bien el sistema esta diseñado para identificar y prevenir fallas, no es completamente autónomo. No deje el equipo operando sin supervisión.**

**Inicio e interrupción de ensayos** Una vez realizada la carga del ensayo de los parámetros de ensayo, es posible iniciar un ensayo utilizando el botón de inicio de ensayo (elemento 3 en la tabla 3) de la interfaz gráfica o bien utilizando el botón de arranque en la botonera.

Para interrumpir el ensayo, es posible volver a clickear en el botón de inicio de ensayo -ahora con la leyenda 'Pausar ensayo'- o pulsar el botón de freno en la botonera. Para reanudar el ensayo, se deben seguir las instrucciones en el párrafo anterior.

## 7.2. Ensayo de torque-tiempo

En el caso de los ensayos de torque-tiempo, es necesario cargar los pares torque-tiempo en formato CSV. Al hacer clic en el botón 'importar curva...' se abrirá un menú para seleccionar el archivo CSV. Al seleccionar un archivo, se graficará la curva de cupla-tiempo seleccionada:



Figura 12: Vista de la pantalla principal tras seleccionar una curva de cupla-tiempo

De ser necesario, el usuario puede incrementar la cantidad de períodos a ensayar.



Figura 13: Vista de la pantalla principal tras seleccionar una curva de cupla-tiempo y extender la cantidad de periodos de ensayo a 4.

Tras verificar que el perfil de cupla-tiempo es el adecuado, es posible enviar los parámetros de ensayo al sistema de accionamientos haciendo click en 'Fijar parámetros'. En caso de que haya algún error en los

parámetros de ensayo, se presentará una alerta indicando el tipo de error. Si ese fuera el caso, se deberá corregir la condición de falla y presionar el botón nuevamente.

Suponiendo que no haya habido ningún error, el sistema se encuentra listo para iniciar el ensayo:

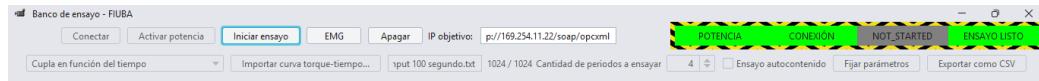


Figura 14: Franja superior del programa cuando el ensayo fue cargado correctamente: en el panel de indicadores se observa la leyenda 'ENSAYO LISTO'

Al seleccionar 'Iniciar ensayo', se comenzarán a actualizar las mediciones en pantalla y a graficar las diferentes variables de ensayo:

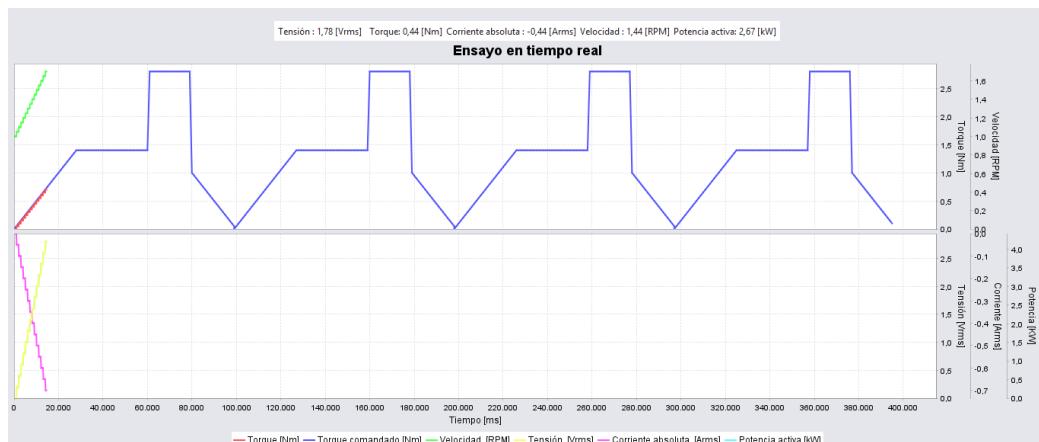


Figura 15: Panel de gráficos y mediciones en tiempo real cuando un ensayo esta en ejecución

**Consideraciones del vector cupla-tiempo** El banco de ensayos no realiza ningún tipo de interpolación entre los valores introducidos. Se mantiene el valor de cupla introducido hasta el siguiente comando de cupla-tiempo.

Para la correcta lectura del archivo CSV los valores correspondientes al tiempo de aplicación de la cupla se deberán expresar en milisegundos, mientras que la cupla se debe expresar en Nm. El archivo no debe tener encabezado. La división mínima entre comandos de cupla debe ser de 100ms o superior, no es necesario que los comandos se encuentren equiespaciados.

En la figura a continuación se observa un ejemplo de archivo aceptado:

```

1 0 ,0
2 100 ,2
3 200 ,3
4 350 ,4
5 500 ,5
6 600 ,5
7 825 ,5
8 .

```

```

9 .
10 .
11 10000,5
12 11000,5
13 12000,5
14 13000,5

```

Listing 1: Formato de archivo CSV

### 7.3. Ensayo tipo torque-velocidad

Una vez seleccionado el ensayo de tipo torque en función de la velocidad, es necesario completar los campos que corresponden a la ecuación de torque-velocidad. Estos campos son los parámetros de la ecuación cupla-velocidad y la duración del ensayo.

Para la duración del ensayo, se requiere introducir el tiempo de ensayo, en milisegundos. Este valor debe ser un número entero y superior a 100ms.

Como usualmente las consignas de velocidad referidas al eje suelen expresarse en RPM, los coeficientes de la ecuación se expresan en estos términos:

$$T(\omega) = A + B\omega + C\omega^2 + D \frac{\partial\omega}{\partial t}$$

Campo	Unidades	Rango de valores
A	Nm	0-26
B	Nm/RPM	0- $\frac{13}{\omega_{max}}$
C	Nm/RPM <sup>2</sup>	0- $\frac{13}{\omega_{max}^2}$
D	Nms/RPM	0-0.2
$A + B\omega_{max} + C\omega_{max}^2 + DMax(\frac{d\omega}{dt}) < 26Nm$		

Figura 16: Términos de la ecuación torque-velocidad y sus rangos

Para el dimensionamiento de los parámetros B y C considere la consigna de velocidad a introducir en el sistema bajo prueba. A modo de ejemplo, si se introduce una consigna con velocidad máxima 1.000RPM, entonces en su punto máximo el término lineal será 1.000 veces B y el término cuadrático será 1.000.000 de veces C. Si el término B fuese 0,0026, entonces se excedería el valor de cupla máxima si la velocidad angular supera los 1000 RPM. El mismo razonamiento es posible para el término C.

En la tabla 4, se presentan los factores de conversión en el caso en que la velocidad se exprese en radianes por segundo.

Coeficiente	Unidad	Factor de conversión	Nueva unidad
B	Nms/rad	1/9.54	Nm/RPM
C	Nms <sup>2</sup> /rad <sup>2</sup>	1/91.1	Nm/RPM <sup>2</sup>
D	Nms <sup>2</sup> /rad	1/9.54	Nms/RPM

Cuadro 4: Tabla de factores de conversión para su expresión en las unidades de la ecuación cupla

Una vez seleccionados los parámetros, se debe hacer click en 'Fijar parámetros' para continuar con el ensayo. Esto inicia la descarga de parámetros hacia el control. Cuando esta carga haya finalizado, se observará la leyenda 'ENSAYO LISTO'.



Figura 17: Franja superior del programa cuando el ensayo fue cargado correctamente.

Luego de verificar que el ensayo se podrá llevar a cabo de manera segura, es posible iniciar el ensayo haciendo click en 'Iniciar ensayo'. A partir de este momento, será posible el movimiento en los ejes.

#### 7.4. Ensayo mixto

El ensayo mixto es un tipo de ensayo en el que se ejecutan en superposición los comandos de cupla en función del tiempo y los comandos de cupla en función de la velocidad. Para su realización, es necesario cargar un archivo con las consignas de cupla-tiempo, como también definir los parámetros correspondientes a la ecuación cupla.

Dado que ambos modos actúan en simultáneo, es fundamental que la sumatoria de las cuplas generadas no supere el valor máximo permitido por el sistema, lo cual es difícil de prever antes de realizar el ensayo.

A fin de evitar un exceso de cupla, se recomienda ejecutar de manera independiente los ensayos de cupla-tiempo y de cupla-velocidad, y posteriormente realizar un ensayo mixto como combinación de ambas contribuciones. Este enfoque permite estudiar el impacto conjunto de las distintas acciones de control sobre la cupla total aplicada al sistema.

**Duración de ensayo** El ensayo mixto se ejecuta hasta que haya finalizado la ejecución de comandos de cupla-tiempo.

#### 7.5. Ensayo autocontenido

El ensayo autocontenido es una modalidad que permite ejecutar cualquier tipo de ensayo utilizando, como motor de tracción, el servomotor Siemens que ya está incluido con el banco de ensayos.

Esta modalidad tiene como objetivo poder realizar demostraciones rápidas del banco de ensayo. Esto puede ser utilizado para estudiar el comportamiento deseado del ensayo antes de programar un perfil de velocidad-tiempo en el dispositivo que se desea ensayar, entre otras cosas.

Para utilizar esta función, es necesario tildar la caja con la leyenda 'ensayo autocontenido' que se muestra como ítem 11 en la tabla 3. Esto desplegará una ventana con un gráfico de variables de ensayo, independiente del gráfico en la ventana principal del programa, y un botón que permite seleccionar un archivo de velocidad-tiempo.

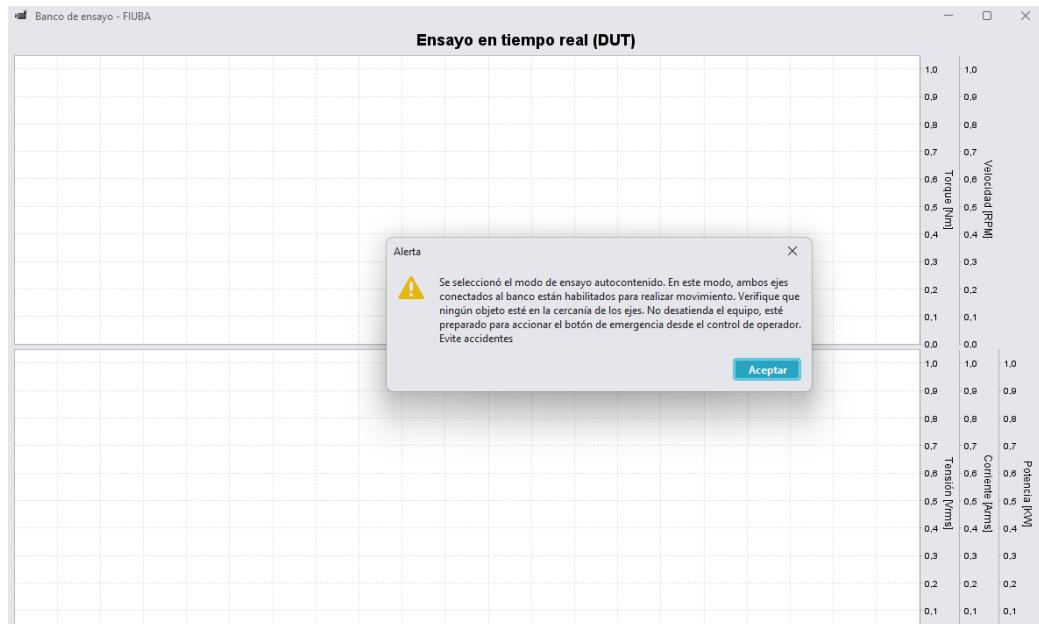


Figura 18: Ventana de advertencia en el modo de ensayo autocontenido

Al aceptar la advertencia, será necesario introducir un perfil de velocidad en función del tiempo en formato CSV. El formato del archivo deberá respetar las indicaciones en el cuadro 7.2.

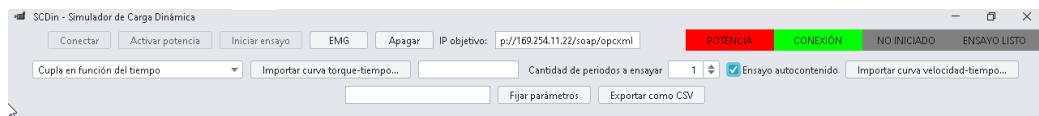


Figura 19: Franja superior del programa al seleccionar 'ensayo autocontenido'

**⚠️ En el modo de operación autocontenido, ambos ejes son capaces de producir movimiento. Verifique que ningún objeto se encuentre en la cercanía de los ejes. No desatienda el equipo durante la operación, esté preparado para accionar el botón de emergencia desde el control de operador.**

**Duración de ensayo** El ensayo autocontenido se ejecuta hasta que haya finalizado el tiempo de ejecución de su comando subyacente o se hayan ejecutado todos los pares velocidad-tiempo, lo que ocurra primero.

## 8. Resolución de problemas

**⚠️ Previo a manipular el interior del equipo asegúrese de utilizar elementos de protección personal e instrumental adecuado.**

### 8.1. Indicadores LED de la unidad de control

La unidad control D425, cuenta con dos bloques de cuatro indicadores LED para informar el estado del sistema de accionamientos.

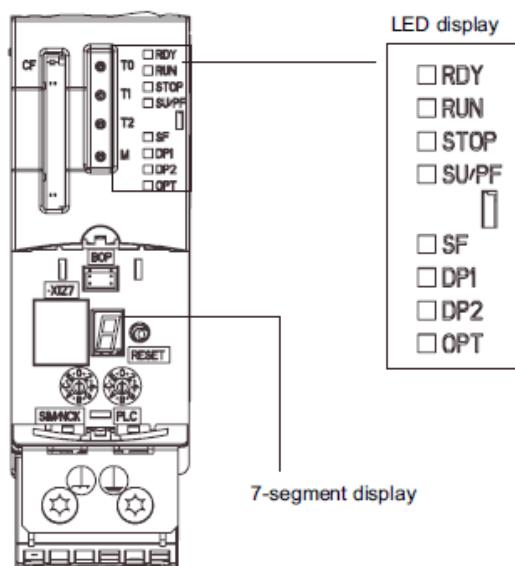


Figura 20: Ubicación de los indicadores LED en la unidad de control

Durante la operación usual, el sistema correrá en los estados RUN cuando el control se encuentre conectado al software y en STOP cuando no. El indicador SF señala un estado de error en el SIMOTION D425. Para más información, referirse al manual 'Commisioning and Hardware Installation Manual'.

Falla	Posibles causas	Verificación	Respuesta
El sistema no enciende	El sistema está desenergizado	Verifique el interruptor general del sistema de accionamientos. Verificar tensión de alimentación trifásica. Verificar tensión se alimentación del circuito control (fuente de alimentación SITOP). Verificar la presencia de LEDs encendidos en la unidad de control.	<ol style="list-style-type: none"> <li>Si hay luces encendidas en la unidad de control consultar el manual 'Commisioning and Hardware Installation Manual' en particular la sección '9.1.1 SIMOTION D4x5-2 and SINAMICS Integrated displays' para el diagnóstico de la falla.</li> <li>Si no hay luces encendidas en el módulo, verificar el estado de la fuente interna de alimentación (SITOP).</li> <li>Reemplace componentes afectados</li> <li>Realice el procedimiento de carga de software en la sección 8.2.</li> </ol>
El ensayo no inicia	La interfaz física está desconectada	Verifique que los LEDs que indican el estado de la fuente de alimentación interna y externa se encuentran encendidos.	<ol style="list-style-type: none"> <li>Verificar la conexión del Botón de emergencia</li> <li>Falla en la fuente externa de 24V.</li> <li>Falla en la fuente interna de 24V.</li> </ol>
	La interfaz por software está desconectada	Verifique que el puerto Ethernet de la computadora esté conectado al puerto X130 del control. Utilizando la línea de comandos realice un PING a la dirección IP 192.168.214.1.	<ol style="list-style-type: none"> <li>Error de configuración IP. Siga los pasos en la sección 6.2.3.</li> <li>Falla en cable Ethernet.</li> </ol>

	Falla en la unidad de control	Verificar la presencia de LEDs encendidos en la unidad de control.	<ol style="list-style-type: none"> <li>1. Verifique que el sistema no se encuentre en estado de emergencia.</li> <li>2. Si hay luces encendidas en la unidad de control consultar el manual 'Commisioning and Hardware Installation Manual' en particular la sección '9.1.1 SIMOTION D4x5-2 and SINAMICS Integrated displays' para el diagnóstico de la falla.</li> <li>3. Apague el sistema y vuelva a encenderlo.</li> <li>4. Realice el procedimiento de carga de software en la sección 8.2.</li> </ol>
Uno o más botones no responden.	Falla mecánica	Pulse el botón, la luz indicadora en la placa electrónica debería encenderse	<ol style="list-style-type: none"> <li>1. Reemplazo del botón</li> <li>2. Falla en el cableado</li> </ol>
	Falla en el puerto de entradas/-salidas digitales	Apague la máquina. Mida resistencia entre los pines de la fuente de alimentación interna. Revise la tensión de la fuente interna de 24V y las conexiones en los puertos.	<ol style="list-style-type: none"> <li>1. Fuente interna en cortocircuito</li> <li>2. Falla en fuente externa.</li> <li>3. Falla en la placa de E/S (cortocircuito).</li> <li>4. Falla en el cableado.</li> </ol>

## 8.2. Procedimiento de carga de software

En ocasiones es posible que sea necesario restaurar el software del sistema. Para realizar ello es necesario configurar la computadora según lo descrito en la sección 6.2.3.

Una vez realizado ello, es necesario abrir el software SIMOTION SCOUT V4.4.



Figura 21: Icóno de escritorio de SIMOTION SCOUT V4.4

Dentro del software se debe acceder a la opción 'open project' (abrir proyecto):

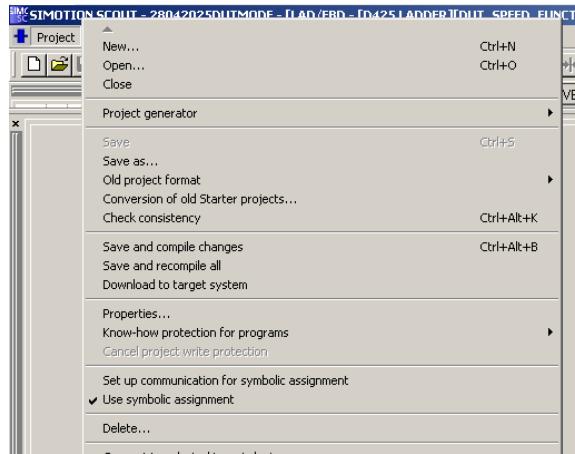


Figura 22: Menú de proyecto

Al cliquear en la opción abrir proyecto se desplegará un menú con los proyectos disponibles. El proyecto a seleccionar se llama 'SCDIN' y está ubicado en la dirección 'documentos/SCDIN'.

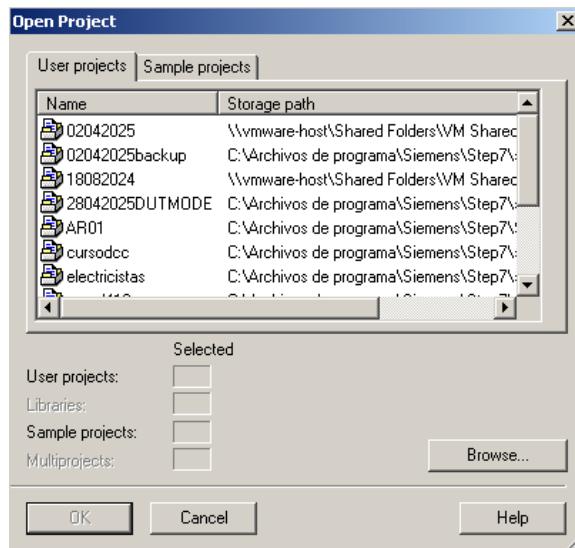


Figura 23: Menú de selección de proyecto

Luego de cargar el proyecto a SIMOTION SCOUT, es necesario descargarlo al controlador del equipo. Para hacer esto, lo primero es conectarse con la unidad utilizando el botón "connect to target device":

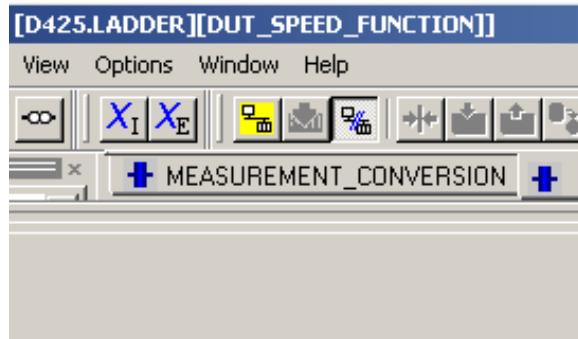


Figura 24: Botón de conexión con la unidad de control en SIMOTION SCOUT V4.4

Una vez establecida la conexión, es necesario descargar el programa de usuario en la unidad de control.

Para realizar ello, se utiliza el botón "download to target device". Luego de hacer click en el botón, se despliega el siguiente menú:

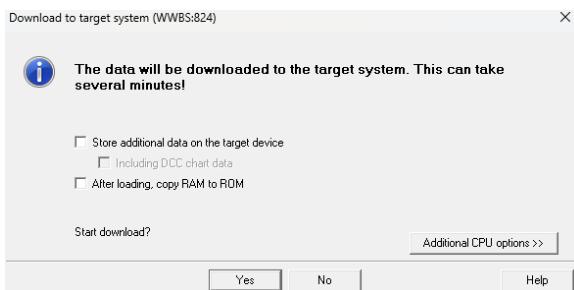


Figura 25: Menú de descarga

Clickear aceptar usando la misma configuración que se muestra en la figura anterior. El programa alertará sobre la necesidad de llevar el sistema del estado RUN al estado STOP.



Figura 26: Menú de descarga

Es necesario aceptar para poder descargar el programa de usuario en la unidad de control. Al finalizar la descarga el software dará la opción de cambiar el estado del sistema a RUN. Seleccione la opción "No":



Figura 27: Menú de descarga

El software ya fue cargado a la unidad de control. Realizar pruebas con la máquina para verificar el correcto funcionamiento de los ensayos. Repetir el procedimiento si es necesario.

### 8.3. Verificación de señales de LADDER

El entorno de desarrollo SIMOTION SCOUT permite la visualización en tiempo real de las variables de programa. Para realizar esto se debe cargar el software a la unidad de control, como indica la sección anterior, y en la ventana de edición del programa seleccionar el ícono de "program status".

Una vez realizado ello, el software que está siendo ejecutado se iluminará de rojo o verde en función de su estado lógico. También es posible observar el valor de las variables en tiempo real.

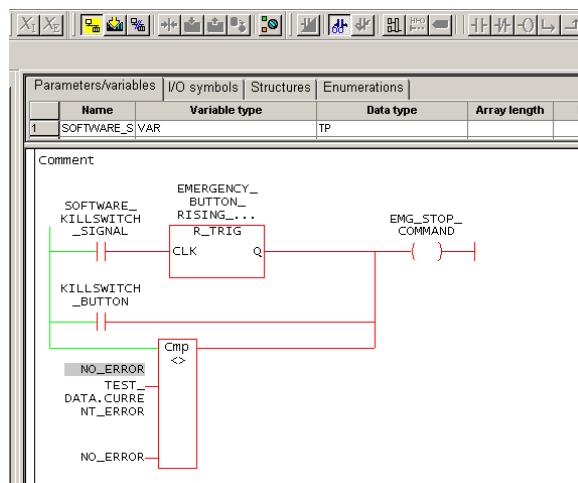


Figura 28: "Program status" siendo ejecutado

Para más información sobre este modo de operación, consultar el manual "SIMOTION SCOUT LAD/FBD programming and operating manual".

### 8.4. Cambio de dirección IP del control

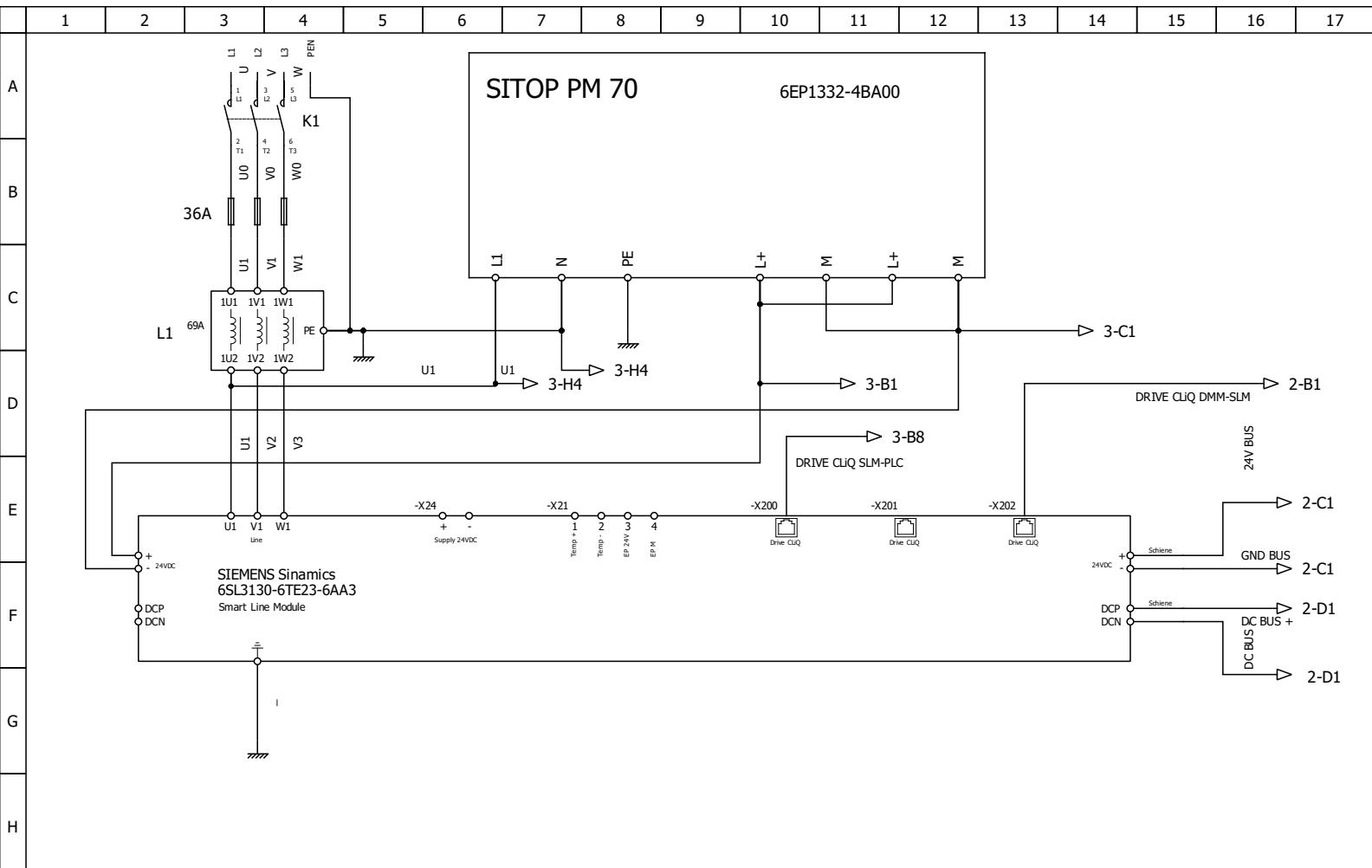
En el caso de que se requiera, es posible modificar la dirección IP del control utilizando el software de programación SIMOTION SCOUT. Para realizar esta modificación, será necesario referirse al manual 'SI-

MOTION Motion Control SIMOTION SCOUT Configuration Manual' en la sección 6.4.3: 'Connecting to the target system'.

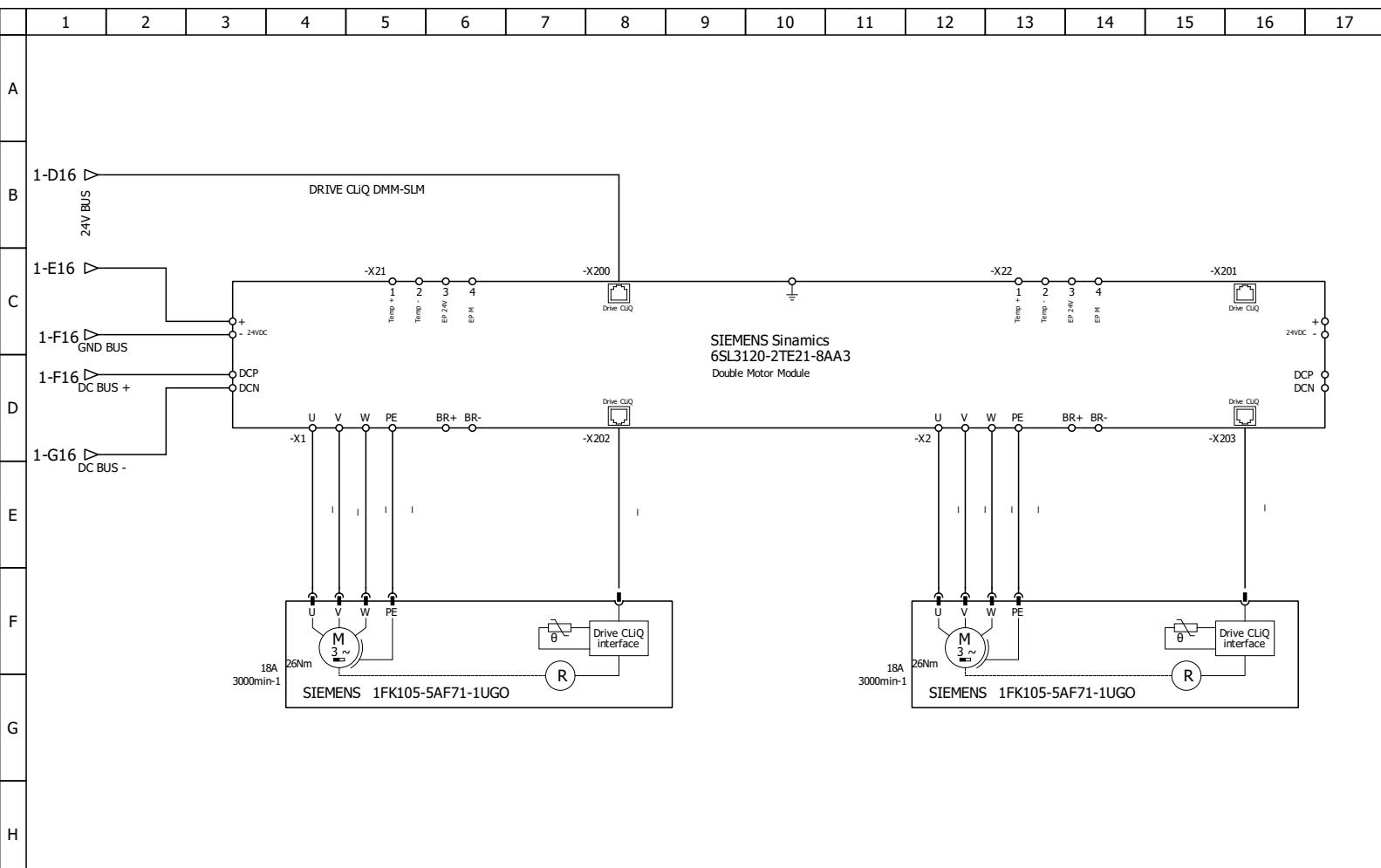
## **8.5. Registro de modificaciones**

Fecha	Motivo	Reparación	Autor

## 9. Plano eléctrico del sistema

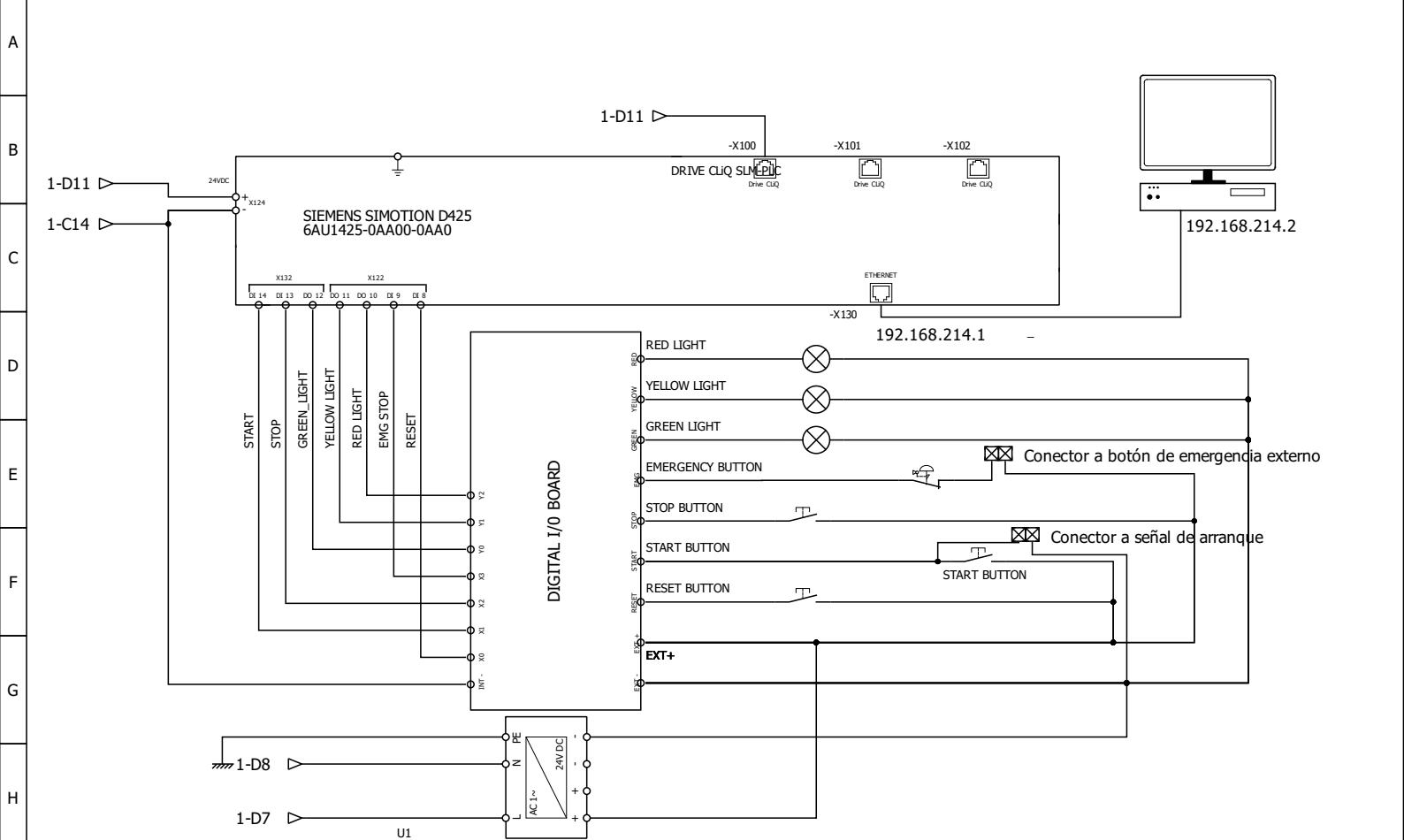


Departamento de Energía   Banco de ensayos, gabinete 8   J. Hirschmann FIUBA - Facultad de Ingeniería de la Universidad de Buenos Aires	Aprobación:   plano-electrico Wiring diagram Torque simulator test bench	SINAMICS S120 smart line module   1/3 1.0   30/4/2025   Argentina
--	--	--



	Departamento de Energía	Banco de pruebas, gabinete 8	J. Hirschmann	Aprobación:	plano-electrico
	FIUBA - Facultad de Ingeniería de la Universidad de Buenos Aires	Wiring diagram	Torque simulator test bench	SINAMICS S120 double motor module	2/3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----



Departamento de Energía | Banco de pruebas, gabinete 8 | J. Hirschmann

Wiring diagram

Aprobación: | plano-electrico

FIUBA - Facultad de Ingeniería de la Universidad de Buenos Aires

Torque simulator test bench

SIMOTION D425 connection to power and digital I/O

3/3

1.0 | 30/4/2025 | Argentina

## 15.2. Programación del PLC

1	2	3	4	5	6	7	8																		
<b>Program : Properties</b> 24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\Properties																									
A							A																		
Author							J. Hirschmann																		
Version							1.1																		
Comment							Grupo de programas para el banco de ensayos.																		
Last saved							Domingo, 06 de Julio de 2025 06:21:36 p.m.																		
Creation type:							Program																		
Program name:							RESET_VARS																		
Author							J. Hirschmann																		
Version							1.1																		
Comment							Programa para ejecutar durante el encendido y apagado del sistema. Se espera que fijen todas las variables necesarias a un estado inicial y conocido																		
Declaration table																									
Parameters/variables																									
<table border="1"> <thead> <tr> <th>Name</th><th>Variable type</th><th>Data type</th><th>Array length</th><th>Initial value</th><th>Comment</th></tr> </thead> <tbody> <tr> <td>clearBufferInsta nce</td><td>VAR</td><td>CLEARINTERNALBUFFER</td><td></td><td></td><td>Instancia de la funcion para resetear el buffer interno del eje de freno</td></tr> <tr> <td>dutClearBufferIn stance</td><td>VAR</td><td>DUTCLEARINTERNALBUFFER</td><td></td><td></td><td>Instancia de la funcion para resetear el buffer interno del eje de traccion</td></tr> </tbody> </table>								Name	Variable type	Data type	Array length	Initial value	Comment	clearBufferInsta nce	VAR	CLEARINTERNALBUFFER			Instancia de la funcion para resetear el buffer interno del eje de freno	dutClearBufferIn stance	VAR	DUTCLEARINTERNALBUFFER			Instancia de la funcion para resetear el buffer interno del eje de traccion
Name	Variable type	Data type	Array length	Initial value	Comment																				
clearBufferInsta nce	VAR	CLEARINTERNALBUFFER			Instancia de la funcion para resetear el buffer interno del eje de freno																				
dutClearBufferIn stance	VAR	DUTCLEARINTERNALBUFFER			Instancia de la funcion para resetear el buffer interno del eje de traccion																				
I/O symbols																									
<table border="1"> <thead> <tr> <th>Name</th><th>Absolute identifier</th><th>Data type</th><th>Comment</th></tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>								Name	Absolute identifier	Data type	Comment														
Name	Absolute identifier	Data type	Comment																						
Structures																									
<table border="1"> <thead> <tr> <th>Structure name</th><th>Element name</th><th>Data type</th><th>Array length</th><th>Initial value</th><th>Comment</th></tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>								Structure name	Element name	Data type	Array length	Initial value	Comment												
Structure name	Element name	Data type	Array length	Initial value	Comment																				
Enumerations																									
<table border="1"> <thead> <tr> <th>Enumeration name</th><th>Element name</th><th>Initialization value</th><th>Comment</th></tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>								Enumeration name	Element name	Initialization value	Comment														
Enumeration name	Element name	Initialization value	Comment																						

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

D

Date	06/07/2025	Editor	J. Hirschmann	Checked		24062025SCDIN-MEJORADO			Page 1
State	Modification	Date	Name	Standard					Pg. 54

F

B

C

D

E

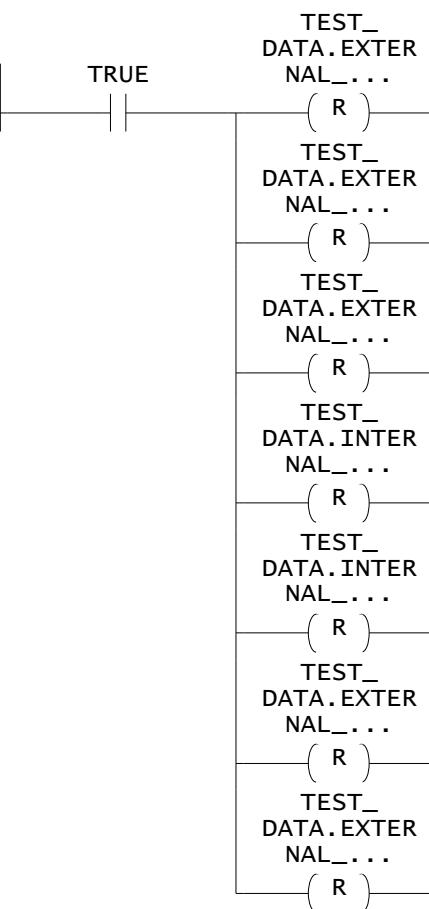
F

**RESET\_VARS - Title**

Reseteo a un valor conocido de las variables del sistema.

**001 - RESETEO DE VARIABLES**

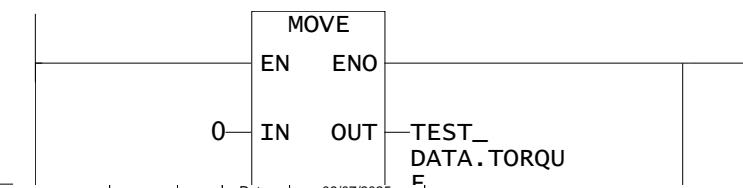
Reseteo de señales externas e internas.



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or sdesign, are reserved.

**002 - Reseteo de coeficientes de ensayo Torque-Velocidad**

Las variables se fijan a 0.

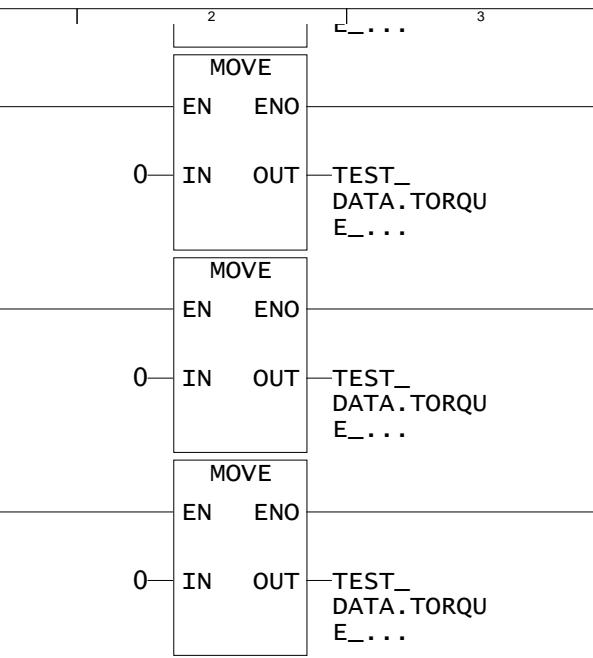


Date	06/07/2025	Editor		Checked		24062025SCDIN-MEJORADO					
State		Modification		Date		Name	Standard				

A

B

C



A

B

C

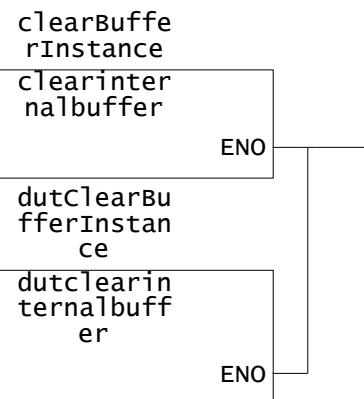
D

E

F

### 003 - Reseteo del buffer interno de pares cupla-tiempo y velocidad-tiempo.

Comment



D

E

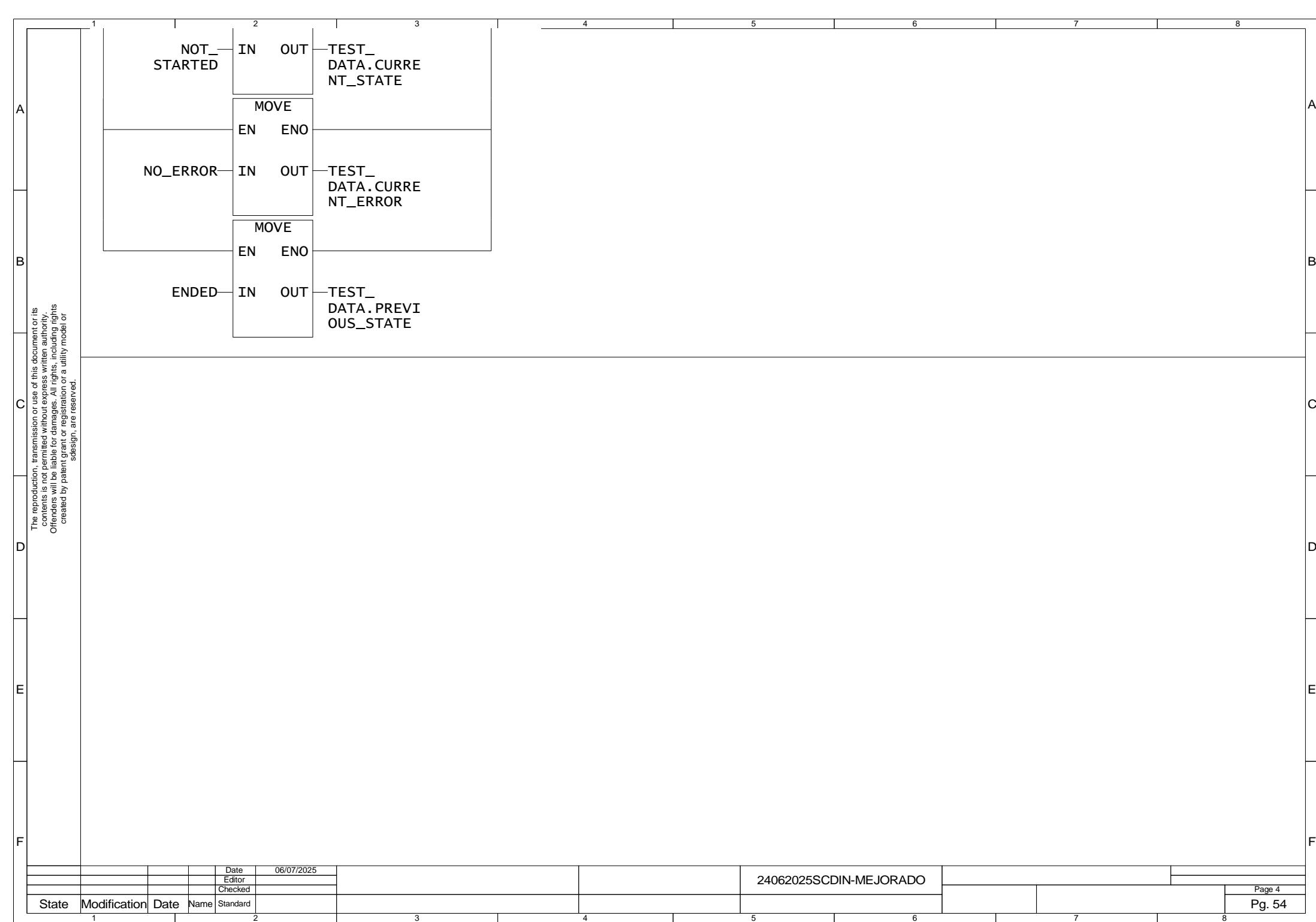
### 004 - TEST STATUS RESET

Reseteo de estados y errores de sistema.



F

State	Modification	Date	Name	Date	Editor	Checked	Comments	Page	Page
1	2	06/07/2025	Standard				24062025SCDIN-MEJORADO		Pg. 54



	1	2	3	4	5	6	7	8
A	<b>Program : Properties</b> 24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\Properties							
	<b>Author</b> J. Hirschmann							
	<b>Version</b> 1.1							
	<b>Comment</b> Grupo de programas para el banco de ensayos.							
	<b>Last saved</b> Domingo, 06 de Julio de 2025 06:21:36 p.m.							
	<b>Creation type:</b> Program							
	<b>Program name:</b> DUT_AXIS_CONTROL							
B	<b>Author</b> J. Hirschmann							
	<b>Version</b> 1.1							
	<b>Comment</b> Programas para el control del motor de traccion en el caso de los ensayos autocontenidos.							
	<b>Declaration table</b>							
C	<b>Parameters/variables</b>							
	Name	Variable type	Data type	Array length	Initial value	<b>Comment</b>		
	MCPowerInstanc eDUT	VAR	_MC_POWER			Instancia de habilitacion de eje		
	DUT_COMMAND _ERROR	VAR	BOOL			Senal de error en el comando de velocidad del eje de traccion		
	DUT_MOVE_INF O	VAR	DINT			informacion del comando de velocidad en el eje		
	DUT_SETPOINT_ IS_SET	VAR	BOOL			Flag que indica si la consigna de velocidad se fijo en el eje de freno		
D	<b>I/O symbols</b>							
	Name	Absolute identifier	Data type			<b>Comment</b>		
	Structure name	Element name	Data type	Array length	Initial value	<b>Comment</b>		
E	<b>Structures</b>							
	Enumeration name	Element name			Initialization value	<b>Comment</b>		
F	Date	06/07/2025						
	Editor							
	Checked							
State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO			
1	2	3	4	5	6	7	8	

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

A

B

C

D

E

F

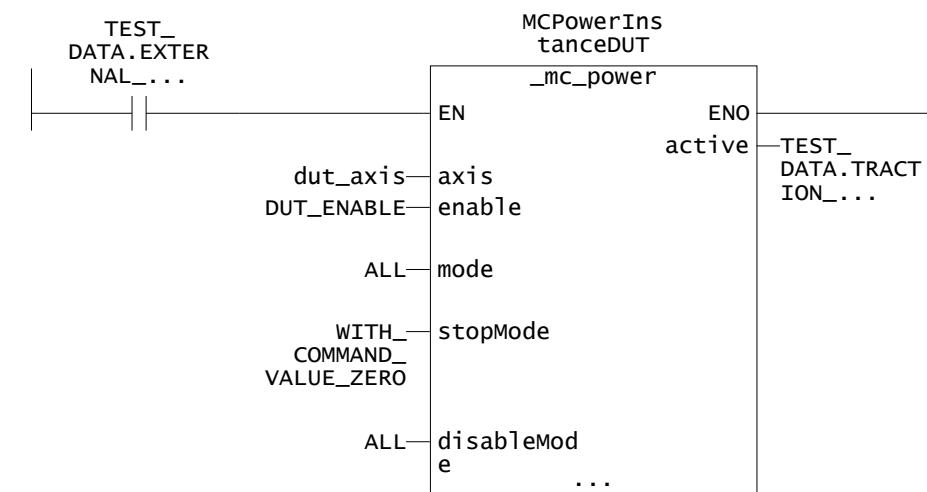
	1	2	3	4	5	6	7	8
--	---	---	---	---	---	---	---	---

## DUT\_AXIS\_CONTROL - Title

Comment

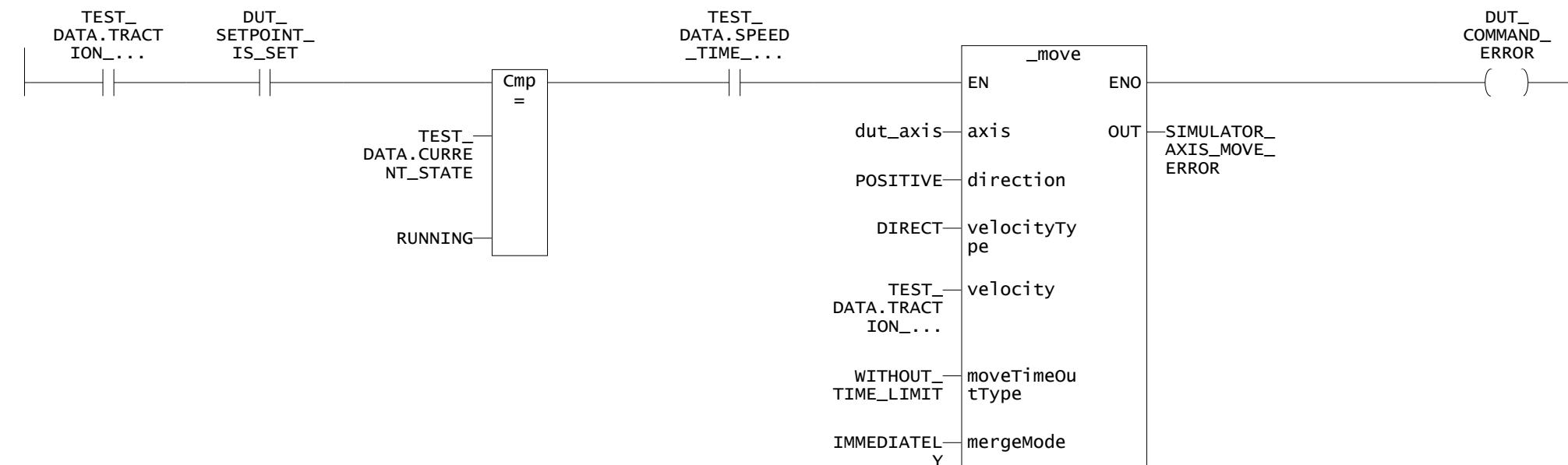
### 001 - ENABLE TRACTION AXIS

Habilita el eje de traccion si detecta la senal externa de encendido



### 002 - SPEED SETPOINT DUT AXIS

Fija el setpoint de velocidad en el eje de traccion



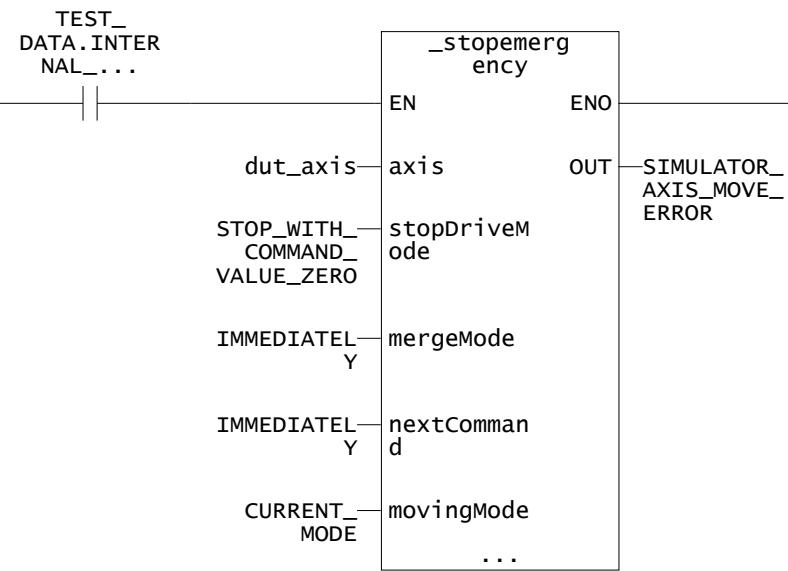
The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or design, are reserved.

Date	06/07/2025
Editor	
Checked	
Name	Standard

24062025SCDIN-MEJORADO

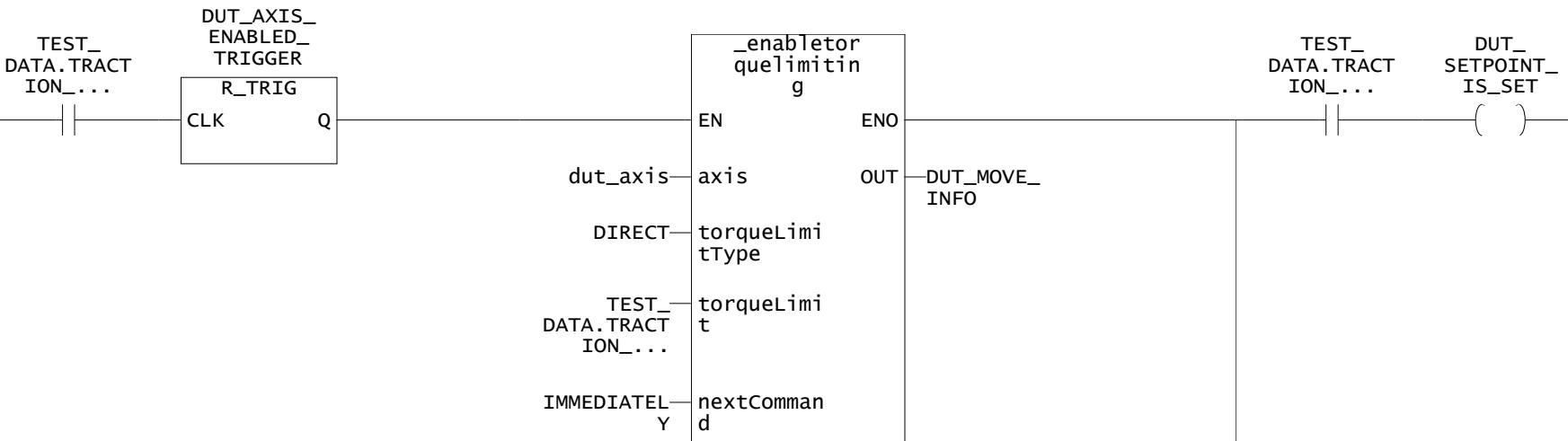
### 003 - EMG STOP DUT AXIS

Freno de emergencia del eje de traccion.



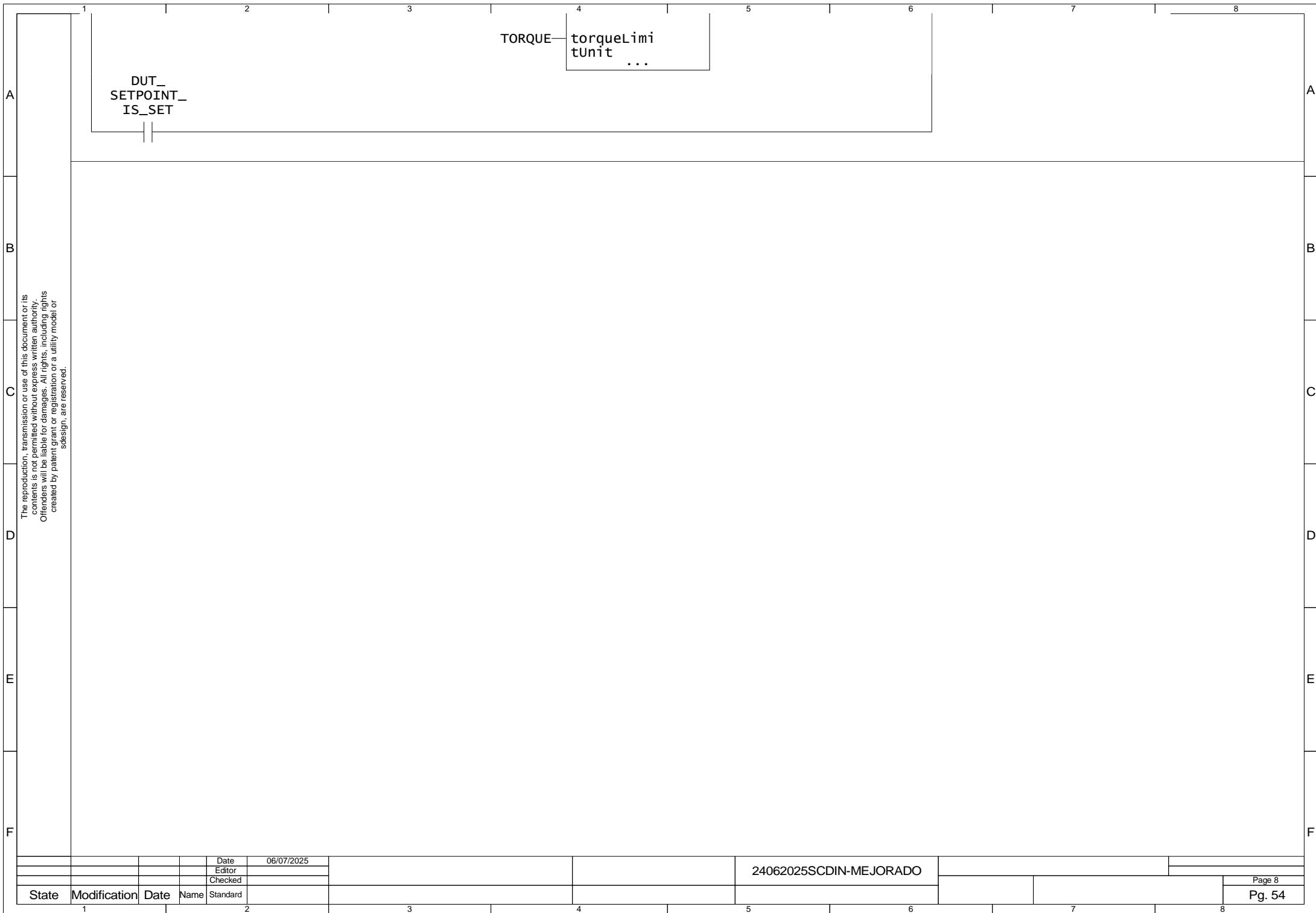
### 004 - DUT AXIS TORQUE SETPOINT

Fija la cupla maxima para el eje de traccion.



Date	06/07/2025
Editor	
Checked	
State	
Modification	
Date	
Name	Standard

24062025SCDIN-MEJORADO





	1	2	3	4	5	6	7	8		
A	Name	Variable type		Data type	Array length	Initial value	Comment			
	TORQUE_PCT	VAR		REAL			Valor porcentual de la cupla electromagnetica en el eje simulador			
	POWER_PCT	VAR		REAL			Valor porcentual de la potencia en el eje simulador			
	VOLTAGE_PCT	VAR		REAL			Valor porcentual de la tension en el eje simulador			
	SIMULATOR_SP EED_CONVERTER	VAR		REAL						
	DUT_CURRENT_PCT	VAR		REAL			Valor porcentual de la corriente del eje de traccion			
	DUT_TORQUE_PCT	VAR		REAL			Valor porcentual de cupla electromagnetica del eje de traccion			
	DUT_POWER_PCT	VAR		REAL			Valor porcentual de la potencia del eje de traccion			
	DUT_VOLTAGE_PCT	VAR		REAL			Valor porcentual de la tension del eje de traccion			
	DUT_CURRENT_CONVERTED	VAR		REAL			Corriente en Arms instantanea en el eje de traccion			
B	DUT_TORQUE_CONVERTED	VAR		REAL			Cupla en Nm instantanea en el eje de traccion			
	DUT_POWER_CONVERTED	VAR		REAL			Potencia activa instantanea en el eje de traccion			
	DUT_VOLTAGE_CONVERTED	VAR		REAL			Tension en Vrms instantanea en el eje de traccion			
	I/O symbols									
C	Name	Absolute identifier		Data type	Comment					
	Structures									
D	Structure name	Element name		Data type	Array length	Initial value	Comment			
	Enumerations									
E	Enumeration name	Element name		Initialization value		Comment				
F	Date	06/07/2025								
	Editor									
G	Checked									
	State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO			
Page 10						Pg. 54				
1	2	3	4	5	6	7	8			

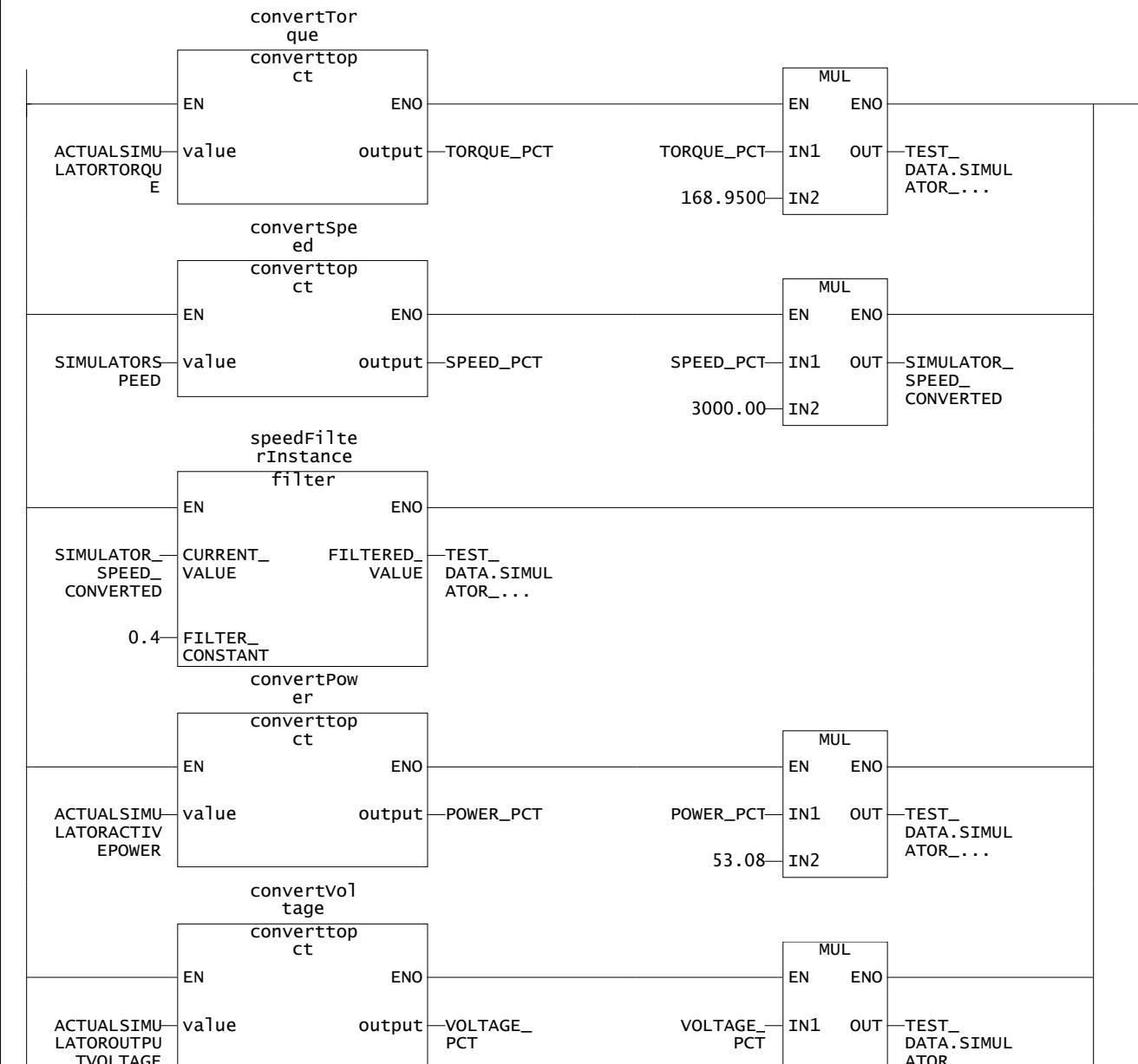
	1	2	3	4	5	6	7	8
--	---	---	---	---	---	---	---	---

## MEASUREMENT\_CONVERSION - Title

### Comment

001 - Export measurements conversion

Conversion de variables medidas en el eje simulador. Se aplica un filtro FIR en la velocidad del eje simulador.



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

D

E

F

A

B

C

D

E

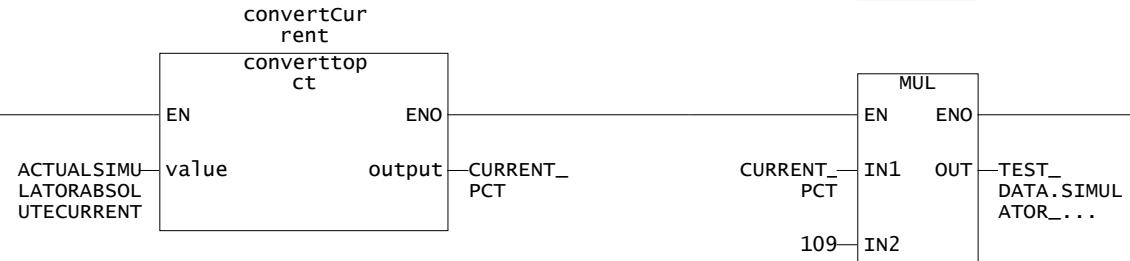
F

Date	06/07/2025	Editor		Checked				
State	Modification	Date	Name	Standard				

24062025SCDIN-MEJORADO

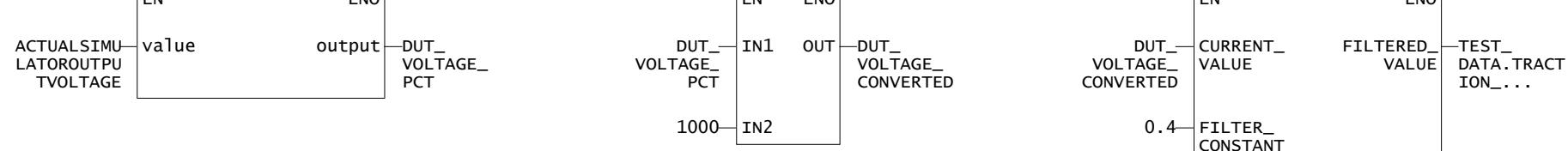
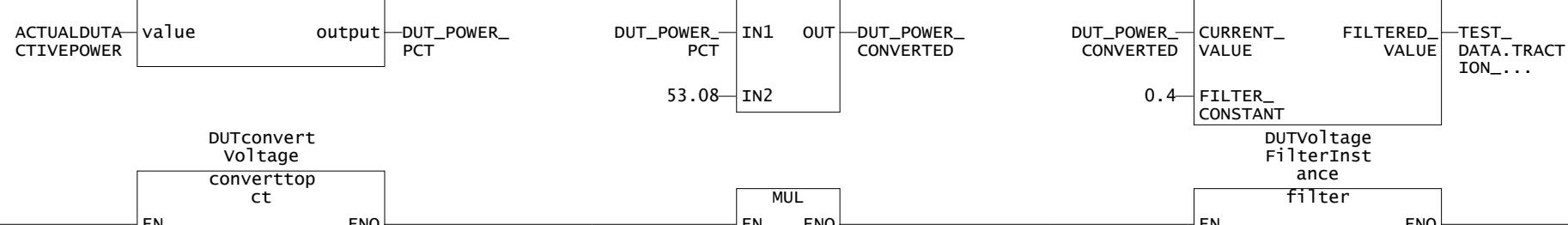
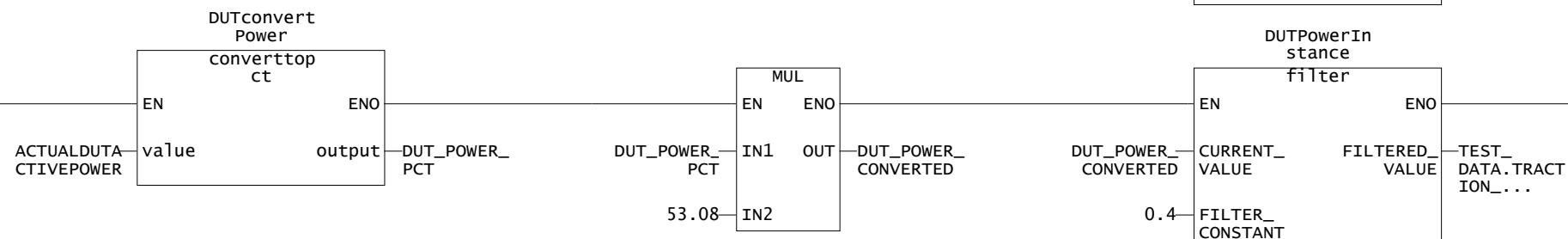
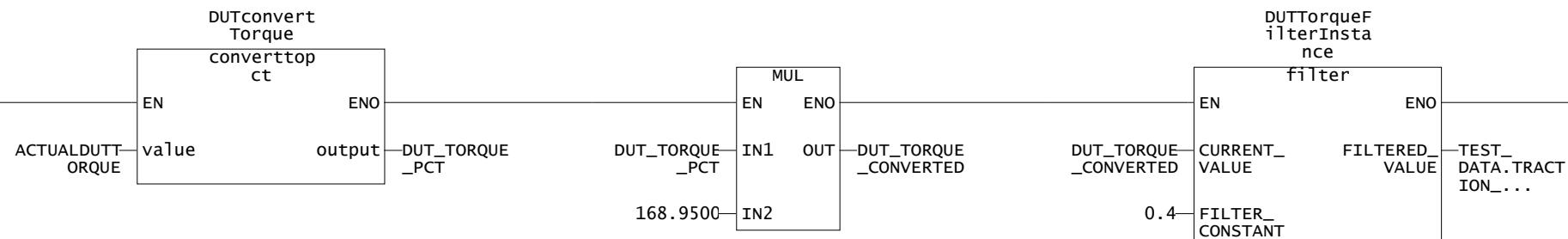
Page 11

Pg. 54

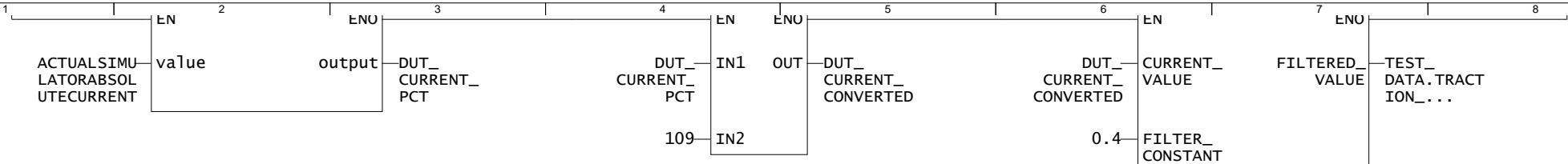


#### 002 - Export measurements conversion

Conversion de variables medidas en el eje de traccion. Se aplica un filtro FIR en la velocidad del eje simulador.



Date	06/07/2025	Editor			
Checked					
State	Standard	Modification	Date	Name	24062025SCDIN-MEJORADO



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

A

A

B

B

C

C

D

D

E

E

F

F

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		

Program : Properties  
24062025SCDIN-MEJORADO\DI425\PROGRAMS\LADDER

Properties

A	Author	J. Hirschmann	A
	Version	1.1	
	Comment	Grupo de programas para el banco de ensayos.	
	Last saved	Domingo, 06 de Julio de 2025 06:21:36 p.m.	
	Creation type:	Program	
	Program name:	TORQUE_SOURCE_HANDLER	
B	Author	J. Hirschmann	B
	Version	1.0	
	Comment	Codigo LADDER para el calculo de la funcion torque	

Declaration table

Parameters/variables

Name	Variable type	Data type	Array length	Initial value	Comment
ABS_MEASURE_D_TORQUE	VAR	REAL			Valor absoluto del torque medido
BUFFER_DEPLETION_OVERRIDE	VAR	BOOL			Senal para deshabilitar la senal de vacio de buffer. Utilizado durante el desarrollo
DIVISOR	VAR	REAL			Divisor del operador derivada.
get	VAR	GETNEXTFROMINTERNALBUFFER			instancia de la fucion para obtener valores del buffer interno
NEW_DATA_RECEIVED	VAR	BOOL		false	flag que senala la recepcion de nuevos datos
NEW_DATA_TRIGGER	VAR	R_TRIG			Deteccion por flanco ascendente
NEXT_COMMAND_TRIGGER	VAR	R_TRIG			Deteccion por flanco ascendente
PREV_FIRST_TIMESTAMP	VAR	REAL		-1.0	Valor anterior de consigna de tiempo. Cuando no se aplica una consigna anteriormente, se asume -1
PREVIOUS_TORQUE_SETPOINT	VAR	REAL			Valor anterior de setpoint de torque
put	VAR	PUTTOINTERNALBUFFER			Instancia de la funcion para guardar valores en el buffer interno
TORQUE_DERIVATIVE	VAR	REAL			Valor de la derivada de cupla en Nm/s
TORQUE_DIFFERENCE	VAR	REAL			Diferencia entre la cupla medida y su valor anterior
TORQUE_DIFFERENCE_ABS	VAR	REAL			Valor absoluto de la diferencia de cuplas
ELAPSED_TIME_MICROSECONDS	VAR	UDINT			Tiempo transcurrido desde el ultimo llamado a este POU
CURRENT_TIME_STAMP	VAR	UDINT			Consigna de tiempo actual

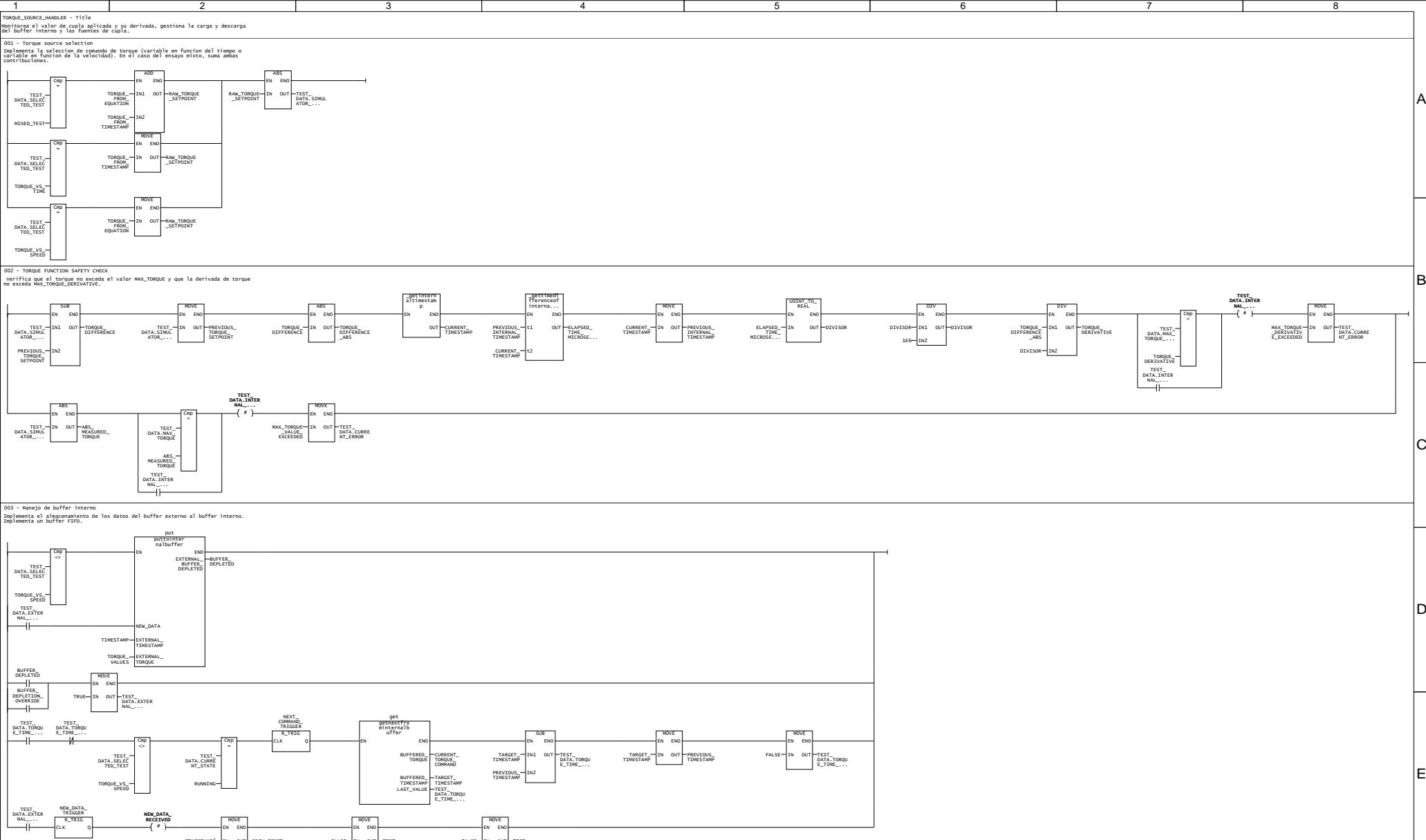
Date	06/07/2025	Editor		Checked			
State	Modification	Date	Name	Standard			

24062025SCDIN-MEJORADO

Page 14

Pg. 54

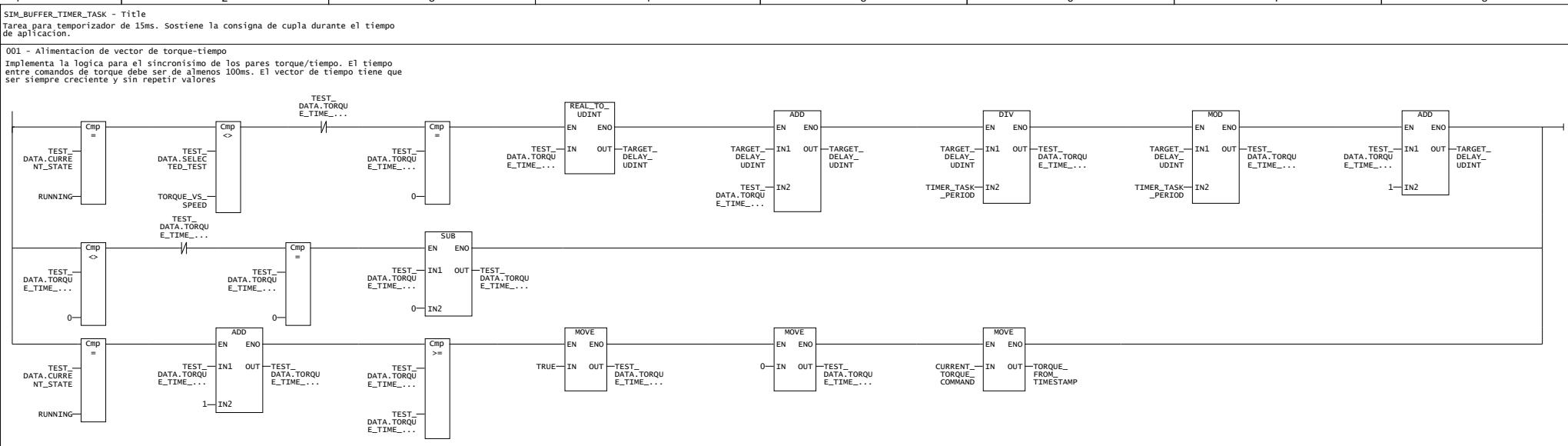




Date	06/07/2025	Editor		Checked		24062025SCDIN-MEJORADO			
State	Modification	Date	Name	Standard					Page 16
1	2	3	4	5	6	7	8	Pg. 54	

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

A	<b>Program : Properties</b> 24062025SCDIN-MEJORADO\DI425\PROGRAMS\LADDER\Properties													
	Author J. Hirschmann													
	Version 1.1													
	Comment Grupo de programas para el banco de ensayos.													
	Last saved Domingo, 06 de Julio de 2025 06:21:36 p.m.													
	Creation type: Program													
	Program name: SIM_BUFFER_TIMER_TASK													
	Author J. Hirschmann													
	Version 1.1													
	Comment Gestión del sincronismo de pares torque-tiempo. Se espera que el programa corra en una tarea ciclica de 15ms.													
Declaration table														
B	<b>Parameters/variables</b>													
	Name	Variable type	Data type	Array length	Initial value	Comment								
	TARGET_DELAY _UDINT	VAR	UDINT			Tiempo en milisegundos del retardo actual a aplicar								
C	TIMER_TASK_P ERIOD	VAR	UDINT		15	Tiempo en milisegundos del periodo de la interrupcion ciclica								
	<b>I/O symbols</b>													
Name		Absolute identifier	Data type		Comment									
<b>Structures</b>														
Structure name		Element name	Data type	Array length	Initial value	Comment								
<b>Enumerations</b>														
Enumeration name		Element name	Initialization value			Comment								
D														
E														
F														
		Date 06/07/2025												
		Editor												
		Checked												
State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO			Page 17						
								Pg. 54						



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Program : Properties  
24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\

### Properties

A	Author	J. Hirschmann	A
	Version	1.1	
	Comment	Grupo de programas para el banco de ensayos.	
	Last saved	Domingo, 06 de Julio de 2025 06:21:36 p.m.	
	Creation type:	Program	

B	Program name:	KEEPALIVE_CHECK	
	Author	J. Hirschmann	
	Version	1.0	
	Comment		B

Declaration table					
Parameters/variables					
Name	Variable type	Data type	Array length	Initial value	Comment
KEEPALIVE_TIMER	VAR	TON			Instancia de temporizador para detectar la desconexion de interfaz grafica

C	TRIGGER_DISCONNECT	VAR	BOOL		Flag de desconexion de interfaz	C
---	--------------------	-----	------	--	---------------------------------	---

I/O symbols					
Name	Absolute identifier	Data type	Comment		

Structures					
Structure name	Element name	Data type	Array length	Initial value	Comment

Enumerations					
Enumeration name	Element name	Initialization value		Comment	

D						D
---	--	--	--	--	--	---

E						E
---	--	--	--	--	--	---

F						F
---	--	--	--	--	--	---

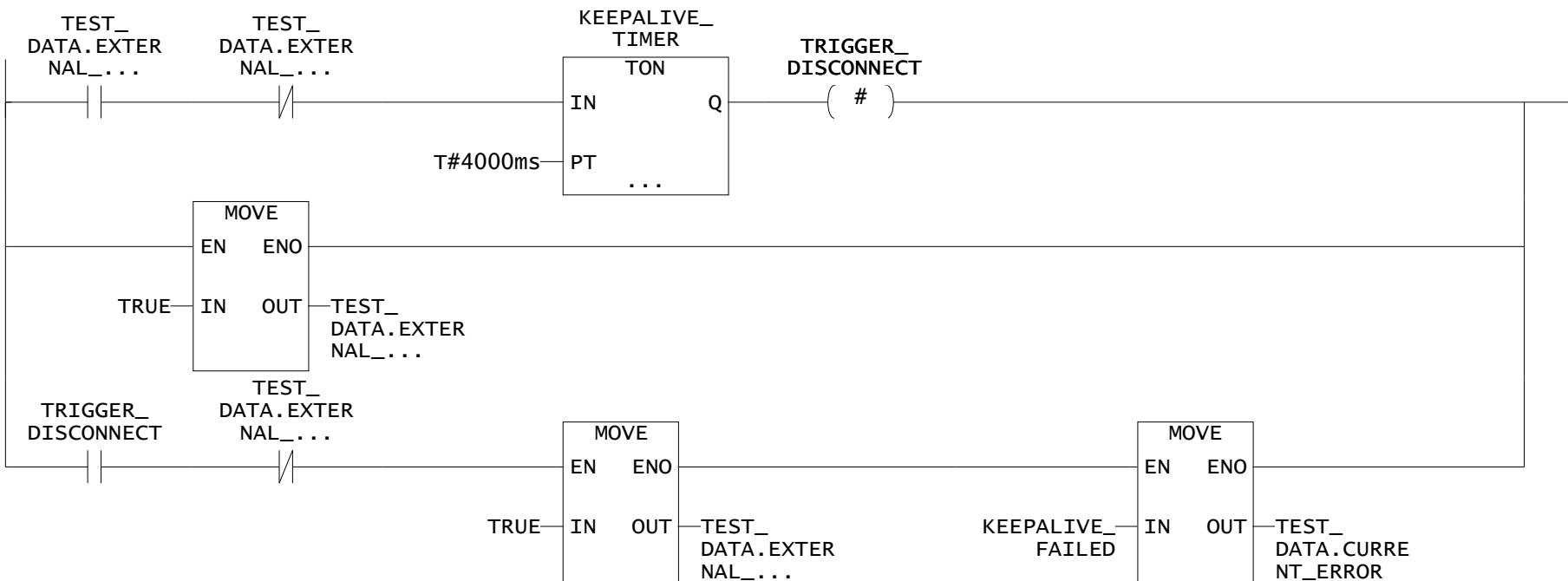
Date	06/07/2025					
Editor						
Checked						
State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO	Page 19

**KEEPALIVE\_CHECK - Title**

Programa para la detección de desconexión del dispositivo. La desconexión se dispara cuando la variable KEEPALIVE es verdadera por más de 2 segundos.

**001 - Manejo de keepalive**

Se espera que la variable KEEP\_ALIVE sea seteada a falso por la interfaz gráfica.



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

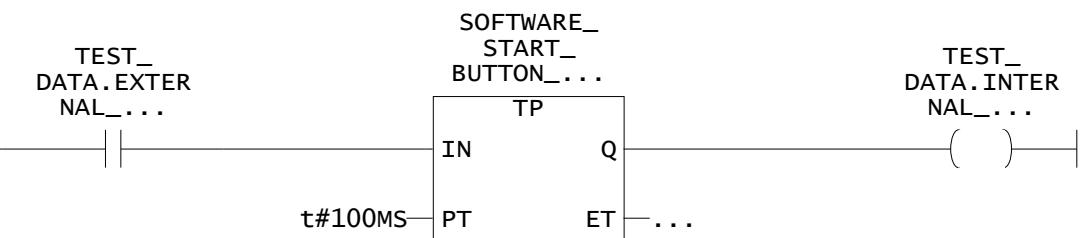
	1	2	3	4	5	6	7	8
	Program : Properties 24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\							
A		Properties						
	Author		J. Hirschmann					
	Version		1.1					
	Comment		Grupo de programas para el banco de ensayos.					
	Last saved		Domingo, 06 de Julio de 2025 06:21:36 p.m.					
	Creation type:		Program					
	Program name:		SIGNAL_SOURCES					
B	Author		J. Hirschmann					
	Version		1.0					
	Comment							
C	Declaration table							
	Parameters/variables							
	Name	Variable type	Data type	Array length	Initial value	Comment		
	SOFTWARE_ST ART_BUTTON_H OLD	VAR	TP					
	START_SIGNAL_ _TRIGGER	VAR	R_TRIG					
	SOFTWARE_ST OP_BUTTON_HO LD	VAR	TP					
	STOP_SIGNAL_ _TRIGGER	VAR	R_TRIG					
	RESET_BUTTON_ FALLING_EDG E	VAR	F_TRIG					
	RESET_BUTTON_ RISING_EDGE	VAR	R_TRIG					
	SOFTWARE_KIL LSWITCH_HOLD	VAR	TP					
	EMERGENCY_B UTTON_RISING_ EDGE	VAR	R_TRIG					
E	TEST_ON_SHUT DOWN	VAR	BOOL					
F	I/O symbols							
	Name	Absolute identifier	Data type		Comment			
	Structures							
	Structure name	Element name	Data type	Array length	Initial value	Comment		
	Enumerations							
	Enumeration name	Element name		Initialization value		Comment		
	Date	06/07/2025						
	Editor							
	Checked							
	State	Modification	Date	Name	24062025SCDIN-MEJORADO			
				Standard				
	1	2	3	4	5	6	7	8
	Page 21							
	Pg. 54							

**SIGNAL\_SOURCES - Title**

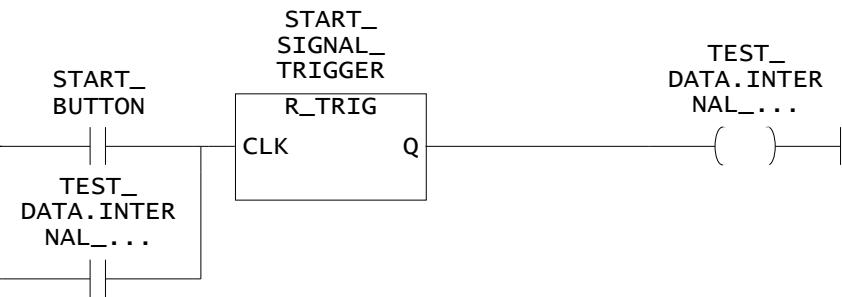
Gestiona las señales internas del sistema

**001 - SOFTWARE START BUTTON PROCESSING.**

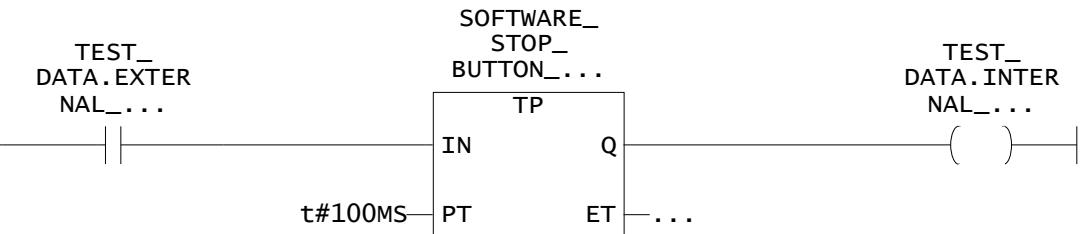
Al recibir una señal de arranque por software, la convierte en un pulso de 100ms de señal interna.

**002 - START COMMAND OUTPUT**

La señal START\_COMMAND se envía como un pulso y considera como fuentes el botón físico y el botón por software.

**003 - SOFTWARE STOP BUTTON PROCESSING.**

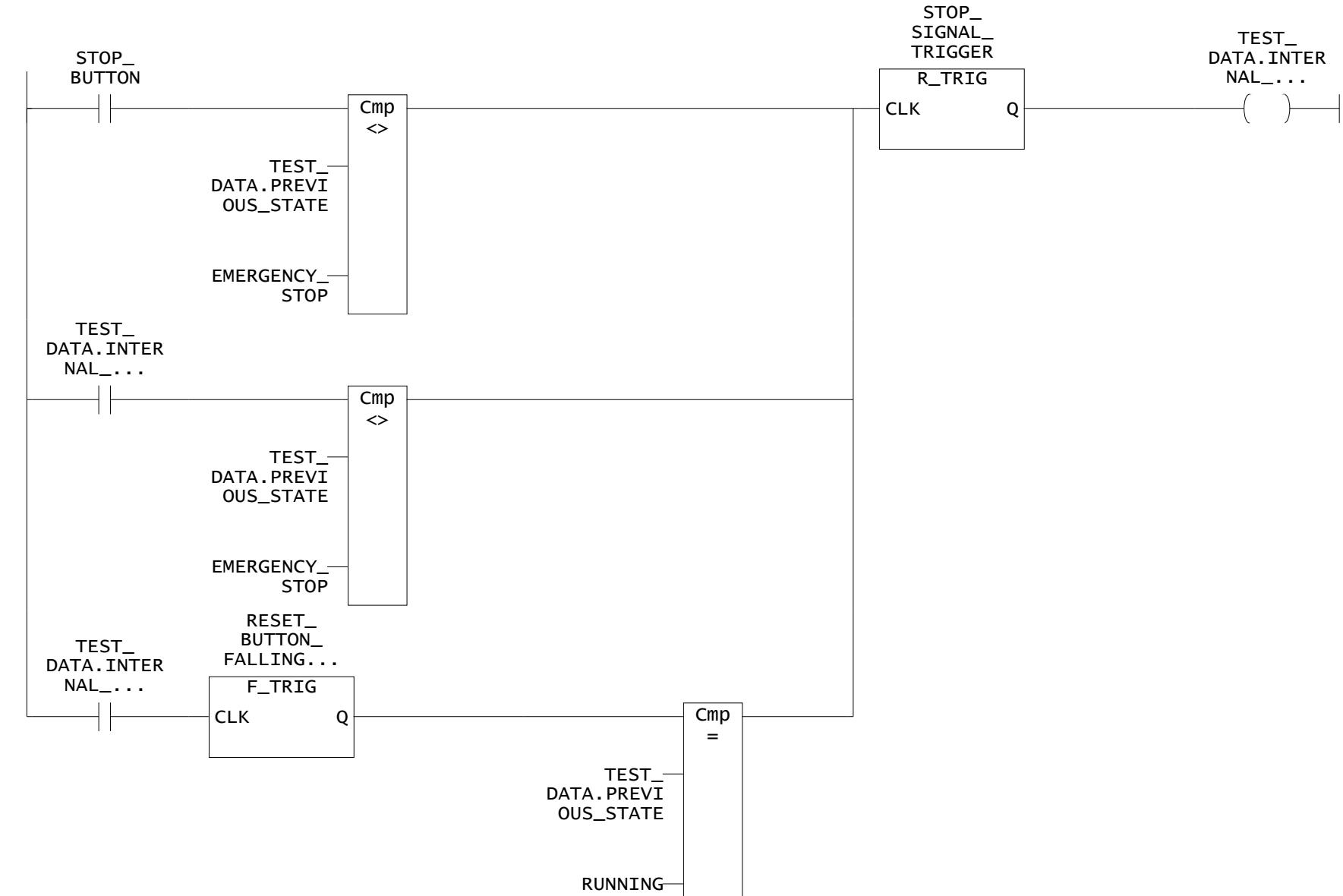
Al recibir una señal de frenado por software, la convierte en un pulso de 100ms de señal interna.

**004 - STOP COMMAND OUTPUT**

La señal STOP\_COMMAND se envía como un pulso y considera como fuentes el botón

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard			24062025SCDIN-MEJORADO	
1	2	3	4	5	6	7	8	

1 FÍSICO, el botón por software y el tramo descendente de la señal de reset



#### 005 - RESET COMMAND OUTPUT

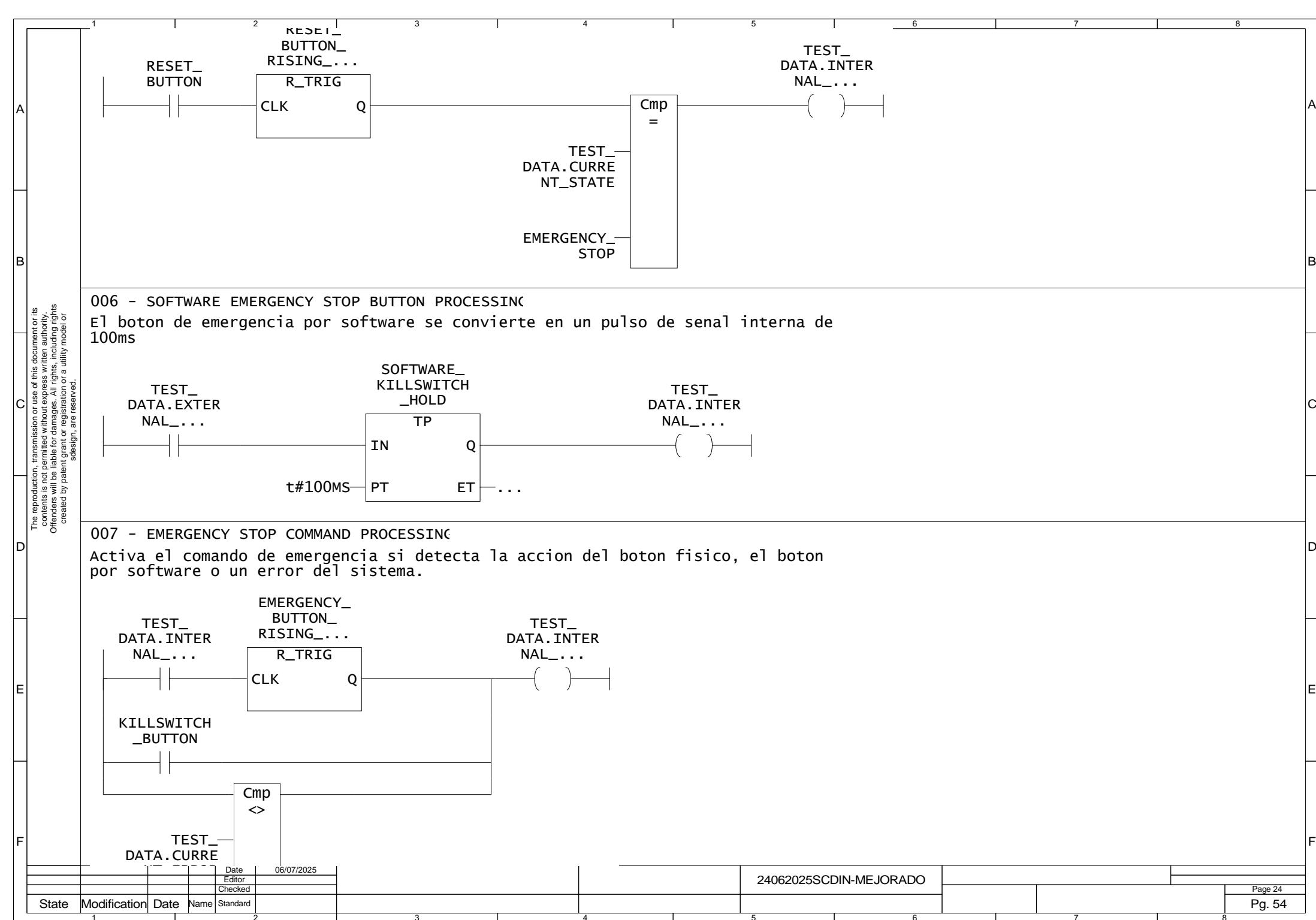
La señal RESET\_COMMAND se envía como un pulso y considera como fuentes solamente el botón físico.

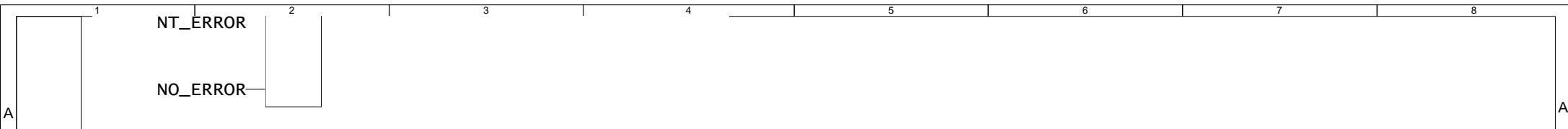
			Date	06/07/2025	RESET
			Editor		
			Checked		
State	Modification	Date	Name	Standard	

24062025SCDIN-MEJORADO

Page 23

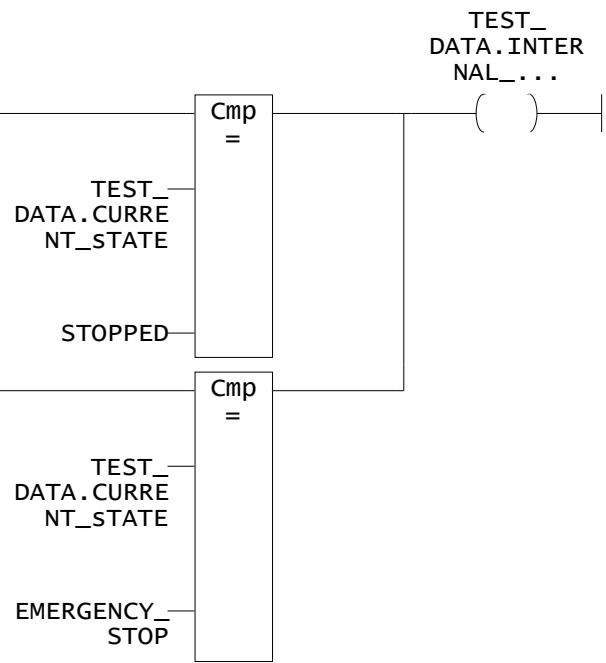
Pg. 54





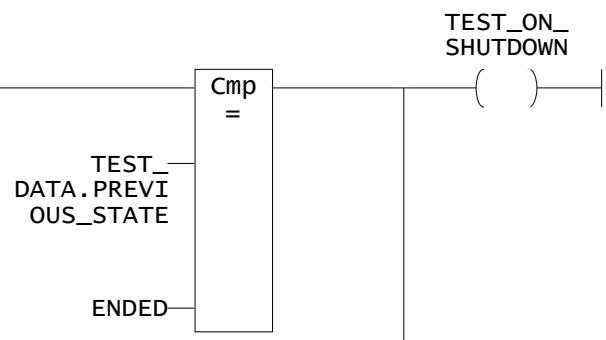
#### 008 - TEST ON HOLD SIGNAL SOURCE

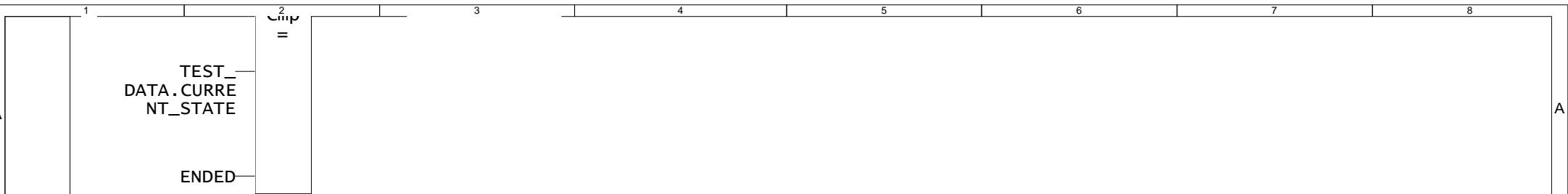
Cuando el ensayo está detenido, la variable TEST\_ON\_HOLD se activa



#### 009 - TEST ON HOLD SIGNAL SOURCE

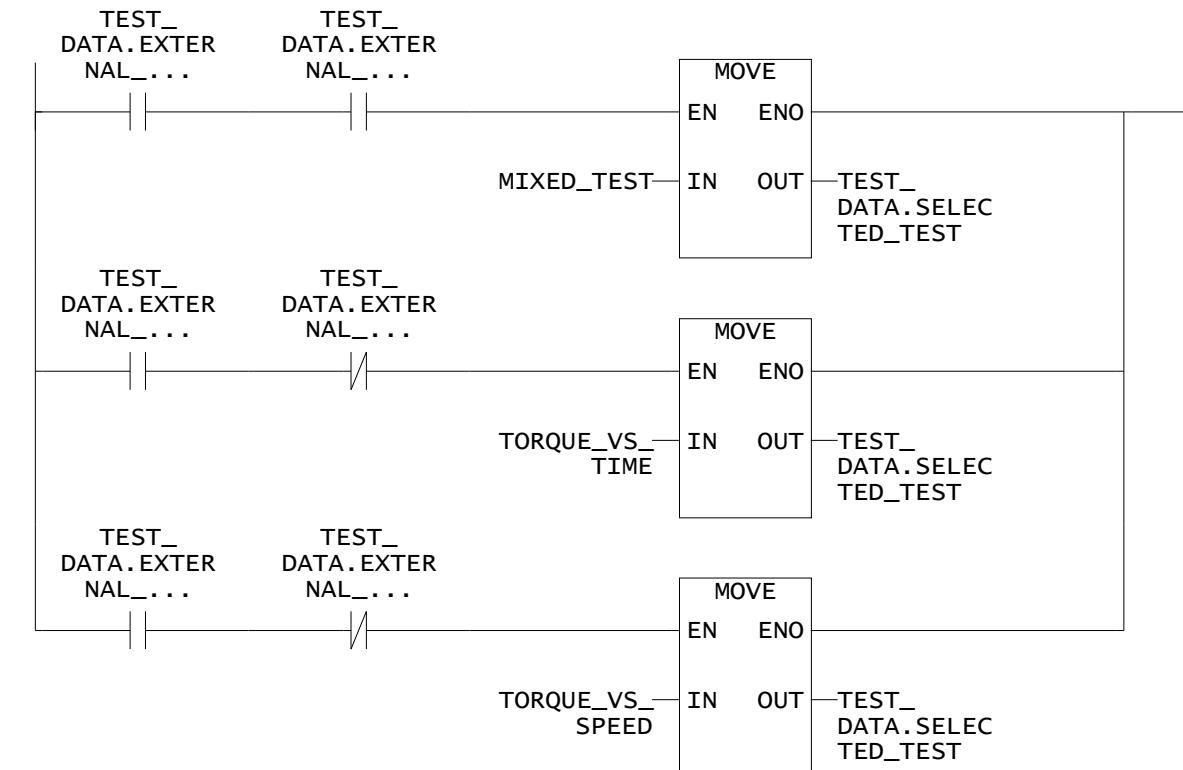
Cuando el ensayo finalizo, la variable TEST\_ON\_SHUTDOWN se activa





#### 010 - TEST SELECTION SETTING

En función de las señales externas fija el tipo de ensayo seleccionado.



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or design, are reserved.

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		

1                    2                    3                    4                    5                    6                    7                    8

Page 26  
Pg. 54

	1	2	3	4	5	6	7	8
A	Program : Properties 24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\Properties							
B	Author Version Comment Last saved Creation type: Program name: Author Version Comment	J. Hirschmann 1.1 Grupo de programas para el banco de ensayos. Domingo, 06 de Julio de 2025 06:21:36 p.m. Program STATE_BEHAVIOUR J. Hirschmann 1.1 Implementacion del comportamiento de los estados de sistema.						
C	Parameters/variables STOPPED_TEST_TRIGGER STARTED_TEST_TRIGGER EMG_STOP_TRIGGER	Declaration table Name Variable type Data type Array length Initial value Comment	VAR R_TRIG VAR R_TRIG VAR R_TRIG				Deteccion por flanco ascendente Deteccion por flanco ascendente Deteccion por flanco ascendente	
D	I/O symbols Structures Enumerations	Name Absolute identifier Data type Comment						
E								
F								
	Date Editor Checked	06/07/2025			24062025SCDIN-MEJORADO			
	State Modification Date Name	Standard						
	1	2	3	4	5	6	7	8

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or sdesign, are reserved.

A

B

C

D

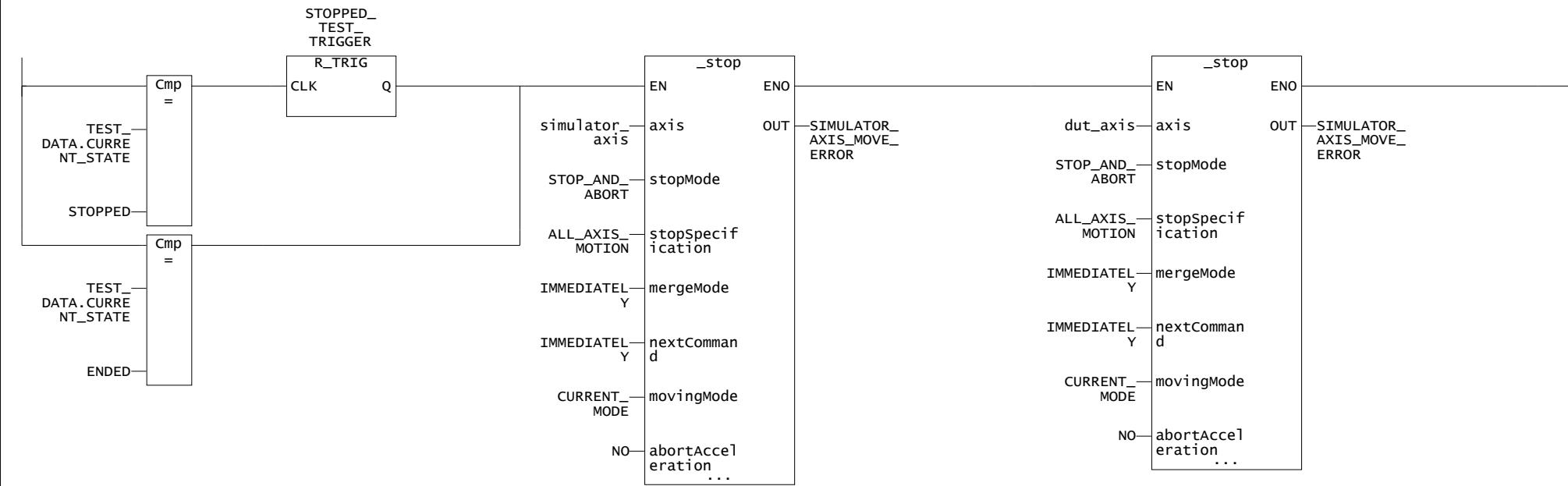
E

F

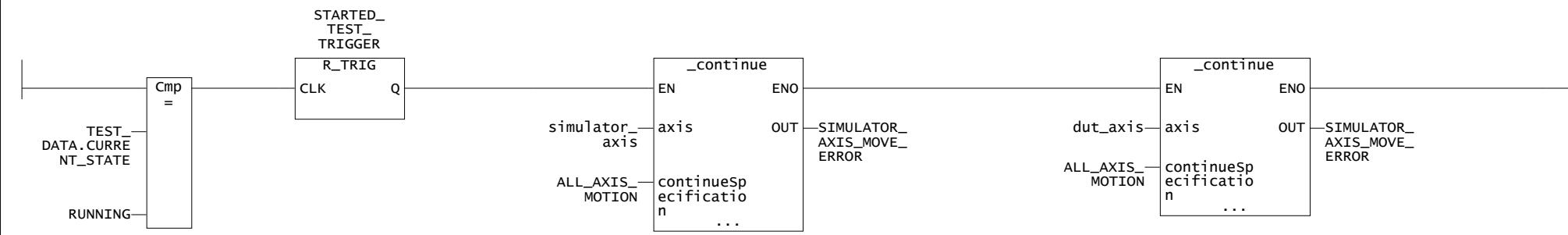
1 2 3 4 5 6 7 8

STATE\_BEHAVIOUR - Title  
Programa para el control del comportamiento de estado

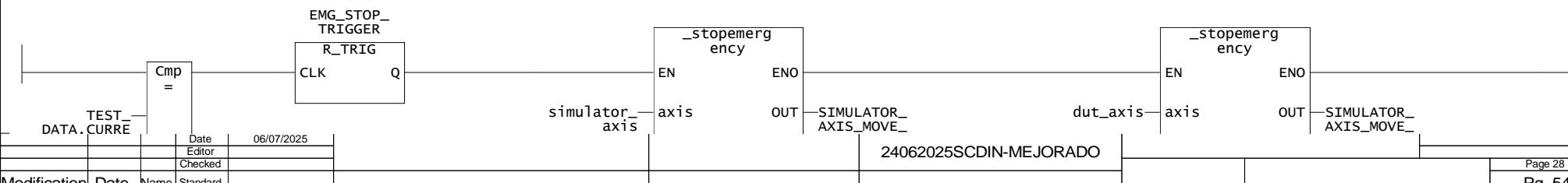
001 - STOP STATE BEHAVIOUR  
Durante el frenado, ambos ejes se detienen.



002 - TEST RUNNING STATE BEHAVIOUR  
En el inicio del ensayo se activa el comando \_continue.



003 - EMERGENCY STOP STATE BEHAVIOUR  
En el inicio del estado de emergencia se activa la emergencia en ambos ejes.



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or sdesign, are reserved.

D

A

B

C

D

E

F

E

F

1

2

3

4

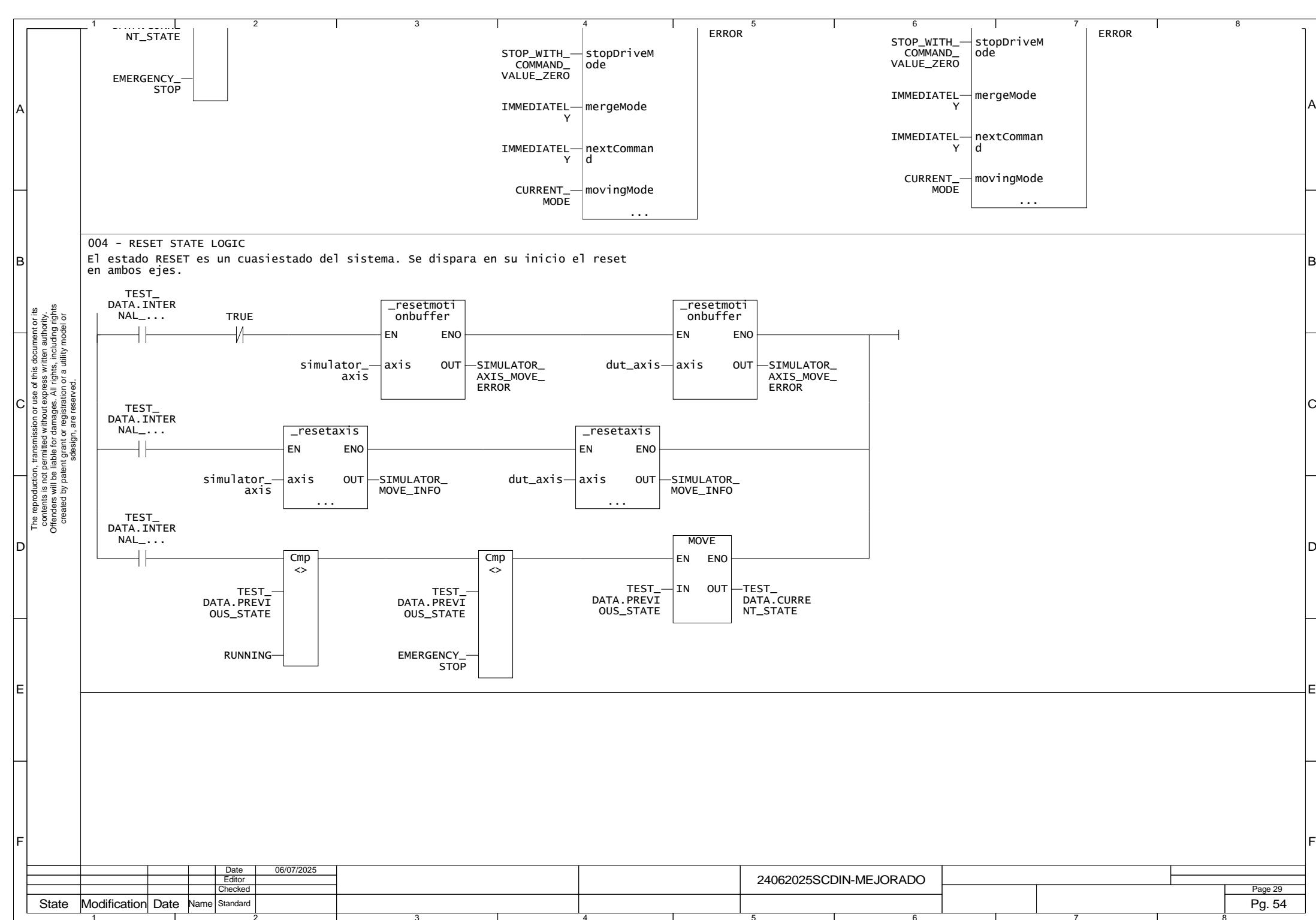
5

6

7

8

Date	06/07/2025	Editor		Checked	
State	Modification	Date	Name	Standard	
					24062025SCDIN-MEJORADO



**Program : Properties**  
**24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER**
**Properties**

A	Author	J. Hirschmann	A
	Version	1.1	
	Comment	Grupo de programas para el banco de ensayos.	
	Last saved	Domingo, 06 de Julio de 2025 06:21:36 p.m.	
	Creation type:	Program	
	Program name:	ENABLE_POWER	
B	Author	J. Hirschmann	B
	Version	1.0	
	Comment		

**Declaration table**

<b>Parameters/variables</b>					
Name	Variable type	Data type	Array length	Initial value	Comment
SLM_ERROR	VAR	BOOL			Flag de detección de error en modulo de linea
SET_SIMULATO_R_SETPOINT	VAR	BOOL			Flag de fijacion de la consigna de velocidad del eje de simulacion
AXIS_ENABLE_DELAY	VAR	TON			Retardo para habilitar los ejes
SIMULATOR_AXI_S_ERROR	VAR	BOOL			Flag de error en el eje simulador
ERROR_ID	VAR	DWORD			Palabra con la descripcion de error en el modulo de linea. Usado durante el desarrollo
STATUS_MODE	VAR	BOOL			
IS_BUSY	VAR	BOOL			Verdadero si el eje de simulacion esta ocupado
STATUS	VAR	BOOL			
RESETINSTANC_E	VAR	_MC_RESET			Instancia de funcion de reseteo del eje
ENABLE_FALLING_EDGE	VAR	F_TRIG			Dteccion por flanco descendente
DISABLE_AXIS_DELAY	VAR	TOF			Retardo para deshabilitar el eje
MCPowerInstance	VAR	_MC_POWER			Instancia de habilitacion de eje
ALMInstance	VAR	_LINEMODULE_CONTROL			Instancia de habilitacion de modulo de linea

<b>I/O symbols</b>					
Name	Absolute identifier	Data type	Comment		

<b>Structures</b>					
Structure name	Element name	Data type	Array length	Initial value	Comment
<b>Enumerations</b>					
Enumeration name	Element name		Initialization value		Comment

		Date	06/07/2025			
		Editor				
		Checked				
State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO	
						Page 30
						Pg. 54

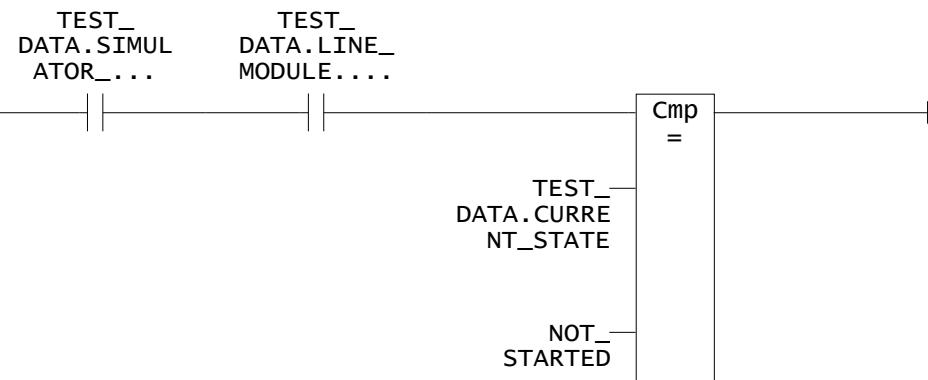
1 2 3 4 5 6 7 8

## ENABLE\_POWER - Title

Comment

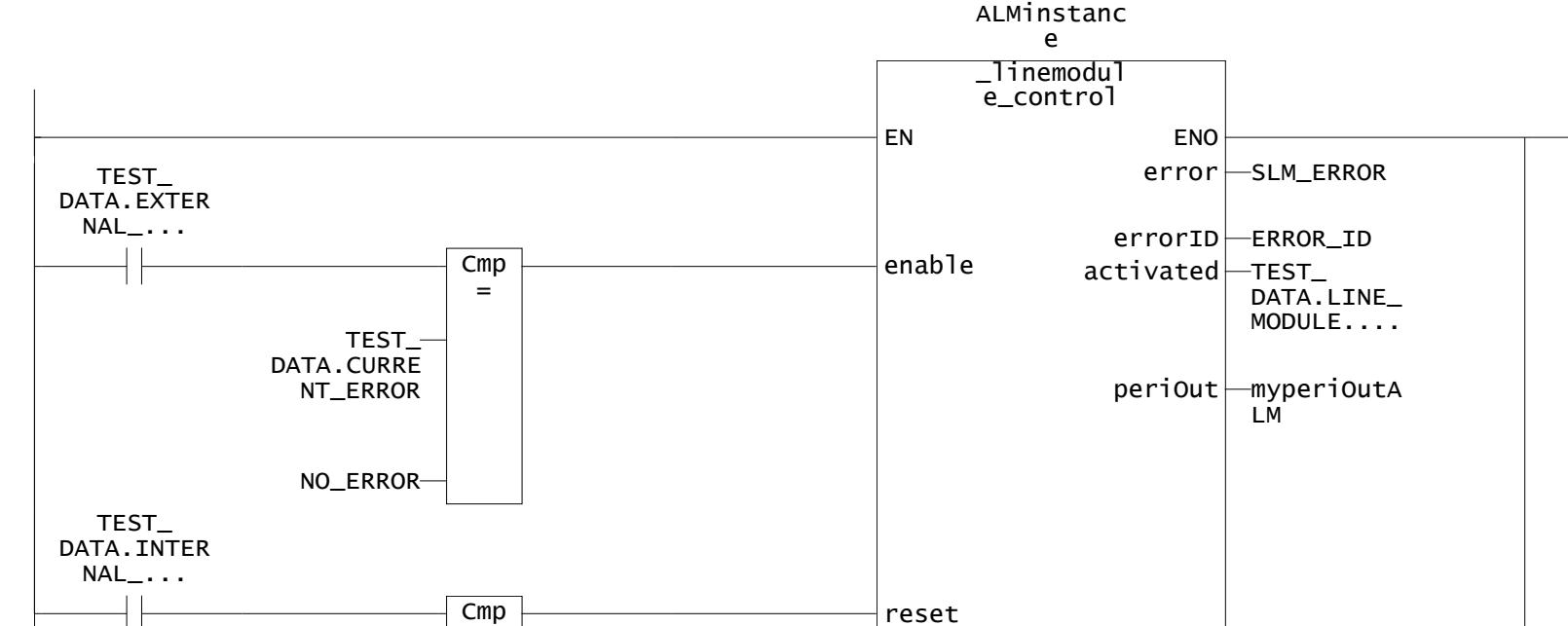
## 001 - READY\_TO\_START STATE TRANSITION

Se transiciona al estado READY\_TO\_START únicamente desde el estado NOT\_STARTED en cuanto se habiliten los elementos de potencia.



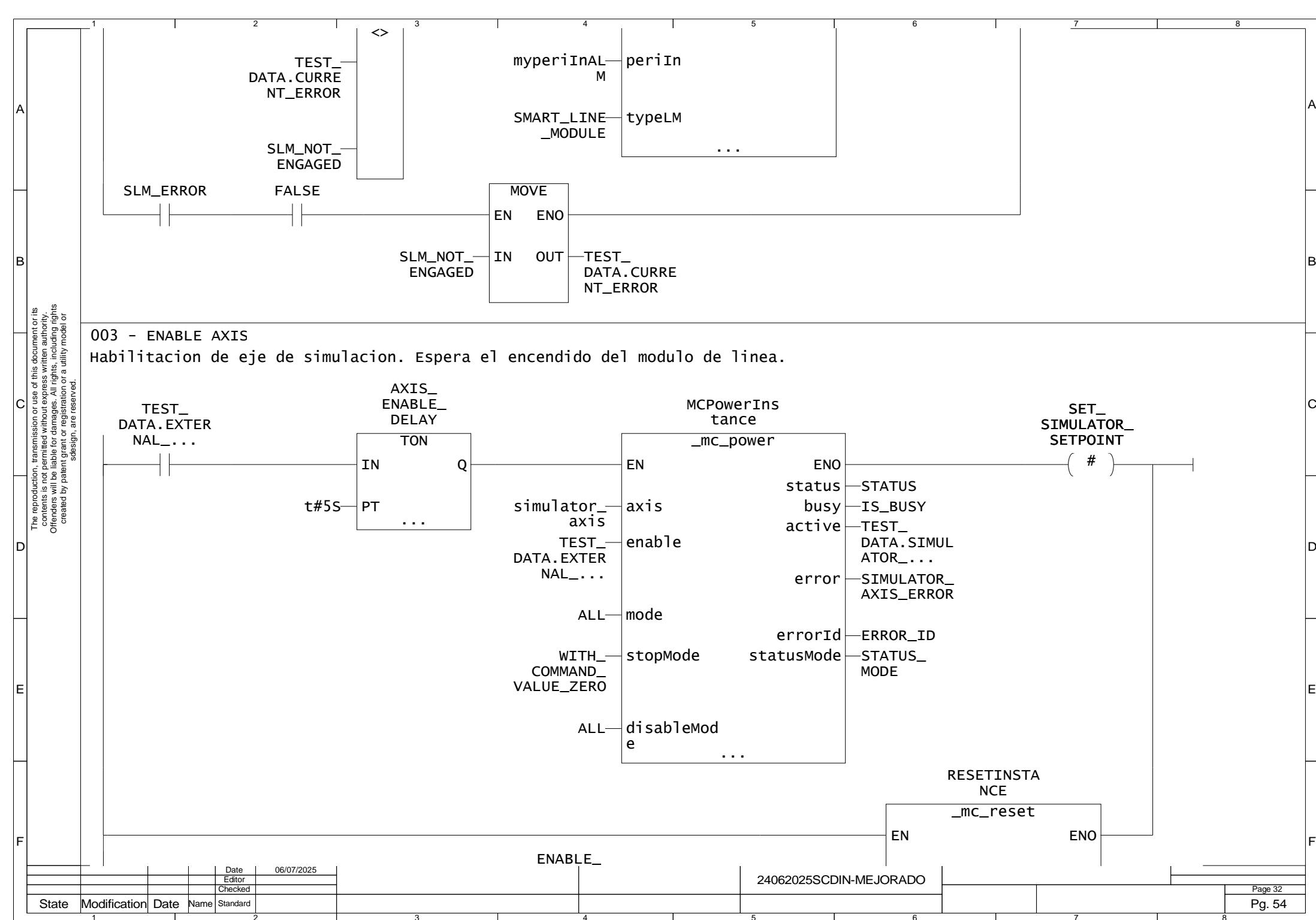
## 002 - Activacion del m[odulo de linea

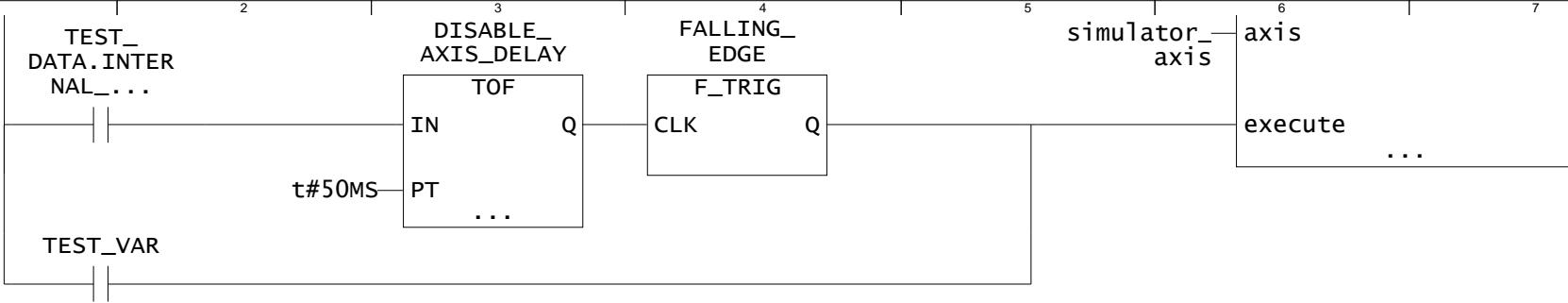
Comment



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

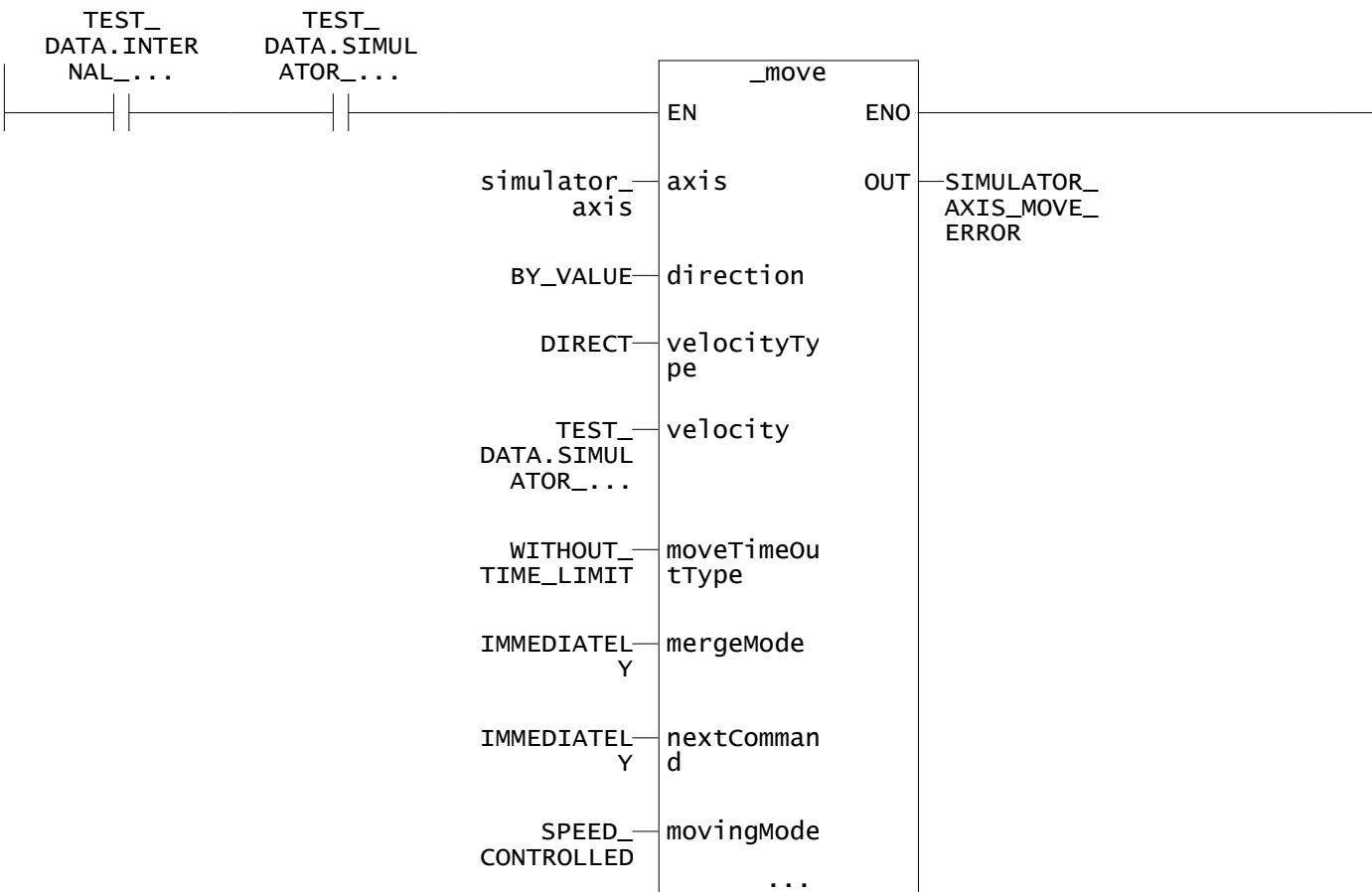
Date	06/07/2025	Editor		Checked		24062025SCDIN-MEJORADO	Page 31	Pg. 54
State	Modification	Date	Name	Standard				
1		2		3	4	5	6	7





#### 004 - SIMULATOR AXIS SPEED SETPOINT

Fija el setpoint de velocidad para el eje simulador una unica vez.



				Date	06/07/2025				
				Editor					
				Checked					
State	Modification	Date	Name	Standard					
1	2	3	4	5	6	7	8		

24062025SCDIN-MEJORADO

--	--	--	--

Page 33

Pg. 54

1	2	3	4	5	6	7	8
<b>Program : Properties</b> 24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\Properties							

Properties	
Author	J. Hirschmann
Version	1.1
Comment	Grupo de programas para el banco de ensayos.
Last saved	Domingo, 06 de Julio de 2025 06:21:36 p.m.
Creation type:	Program
Program name:	SEMAPHORE_CONTROL
Author	J. Hirschmann
Version	1.1
Comment	Administracion de las senales luminicas del sistema.

Declaration table					
Parameters/variables					
Name	Variable type	Data type	Array length	Initial value	Comment
BLINK_OFF_TIMER2	VAR	TOF			Temporizador para apagado
BLINK_ON_TIMER2	VAR	TON			Temporizador para encendido
BLINK_OFF_TIMER3	VAR	TOF			Temporizador para apagado
BLINK_ON_TIMER3	VAR	TON			Temporizador para encendido
BLINK_OFF_TIMER	VAR	TOF			Temporizador para apagado
BLINK_ON_TIMER	VAR	TON			Temporizador para encendido

I/O symbols					
Name	Absolute identifier	Data type	Comment		
<b>Structures</b>					
Structure name	Element name	Data type	Array length	Initial value	Comment
<b>Enumerations</b>					
Enumeration name	Element name		Initialization value	Comment	

Date	06/07/2025						
Editor							
Checked							
State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO		

A

B

C

D

E

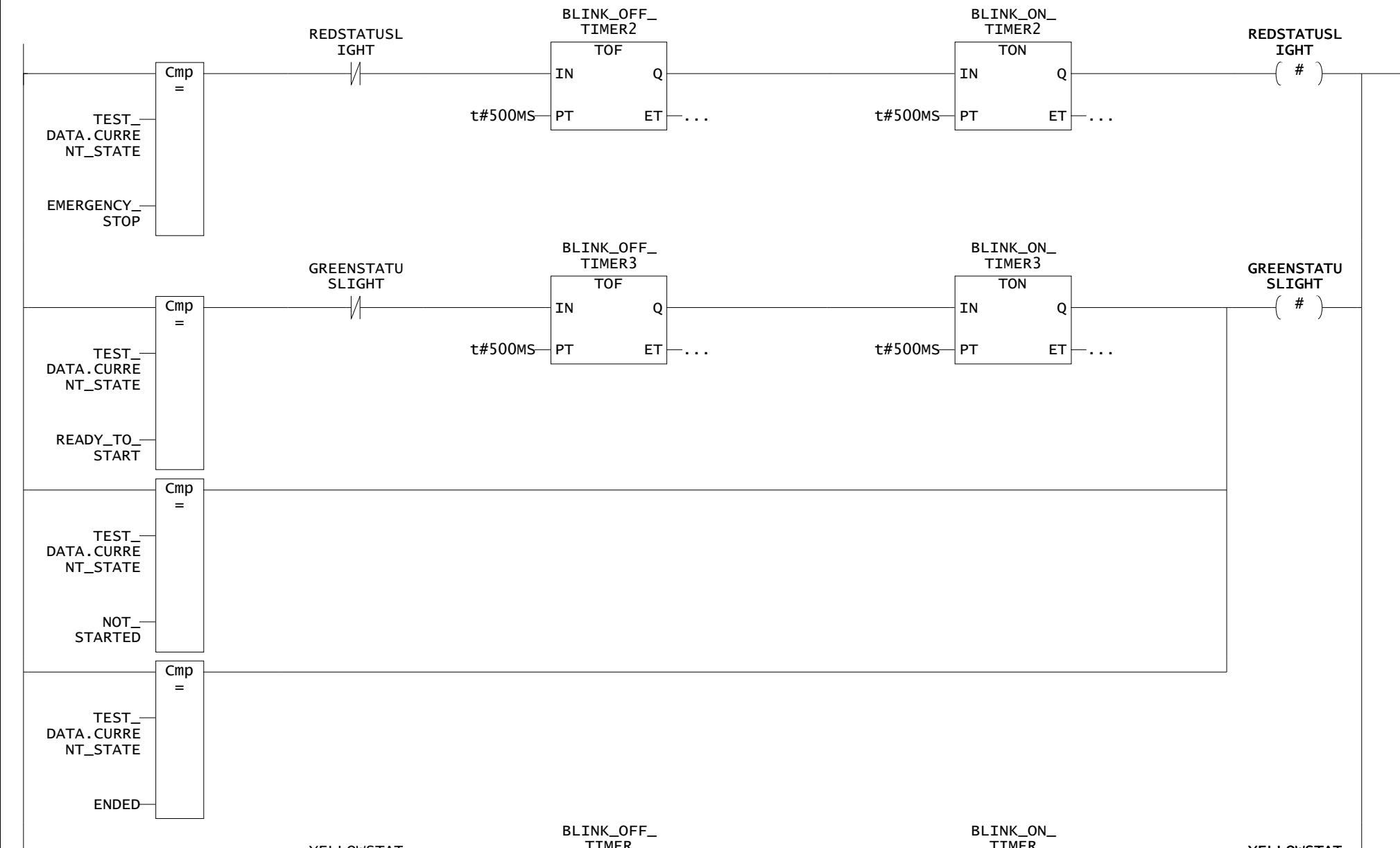
F

# SEMAPHORE\_CONTROL - Title

Control de las señales luminica del sistema.

## 001 - SEMAPHORE LIGHT CONTROL

Control de señales visuales



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

Date	06/07/2025		
Editor			
Checked			
State	Modification	Date	Name
			Standard

YELLOSTAT

BLINK\_OFF\_TIMER

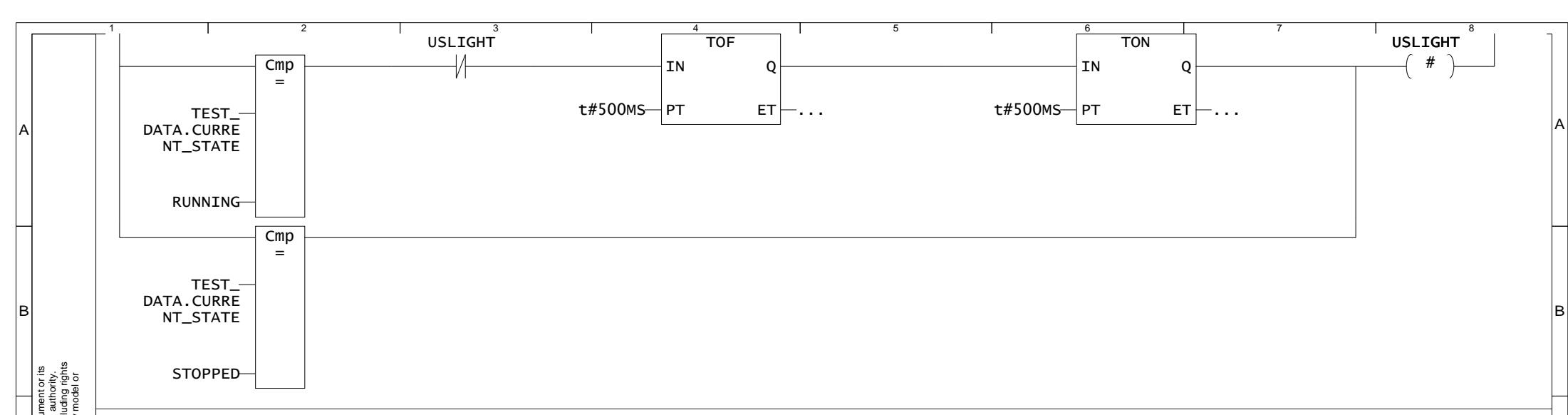
24062025SCDIN-MEJORADO

BLINK\_ON\_TIMER

YELLOSTAT

Page 35

Pg. 54



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

C

D

E

F

A

B

C

D

E

F

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		

	1	2	3	4	5	6	7	8
	Program : Properties 24062025SCDIN-MEJORADO\425\PROGRAMS\LADDER\							
A	Properties							
	Author	J. Hirschmann						
	Version	1.1						
	Comment	Grupo de programas para el banco de ensayos.						
	Last saved	Domingo, 06 de Julio de 2025 06:21:36 p.m.						
	Creation type:	Program						
	Program name:	STATE_TRANSITIONS						
B	Author	J. Hirschmann						
	Version	1.0						
	Comment							
	Declaration table							
	<b>Parameters/variables</b>							
C	Name	Variable type	Data type	Array length	Initial value	Comment		
	ELAPSED_TEST_RUNTIME	VAR	TIME			Tiempo transcurrido de ensayo		
	TEST_STOP_TIME	VAR	TIME			Tiempo de interrupcion del ensayo		
	TEST_RUNTIME_HOLD	VAR	TON			Temporizador con la duracion remanente del ensayo		
	TEST_RUNTIME_TIME	VAR	TIME			Tiempo remanente del ensayo		
	TEST_ON_HOLD_TRIGGER	VAR	R_TRIG			Deteccion por flanco ascendente		
	REMAINING_TEST_TIME	VAR	TIME			Tiempo remanente de ensayo		
D	TEST_WAS_STOPPED	VAR	BOOL			Flag de deteccion de interrupcion de ensayo		
	TEST_INITIAL_RUNTIME	VAR	TIME			Tiempo de duracion inicial del ensayo		
	<b>I/O symbols</b>							
	Name	Absolute identifier	Data type		Comment			
	<b>Structures</b>							
E	Structure name	Element name	Data type	Array length	Initial value	Comment		
	<b>Enumerations</b>							
	Enumeration name	Element name		Initialization value		Comment		
F								
	Date	06/07/2025						
	Editor							
	Checked							
	State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO		
	Page 37							
	Pg. 54							
1	2	3	4	5	6	7	8	

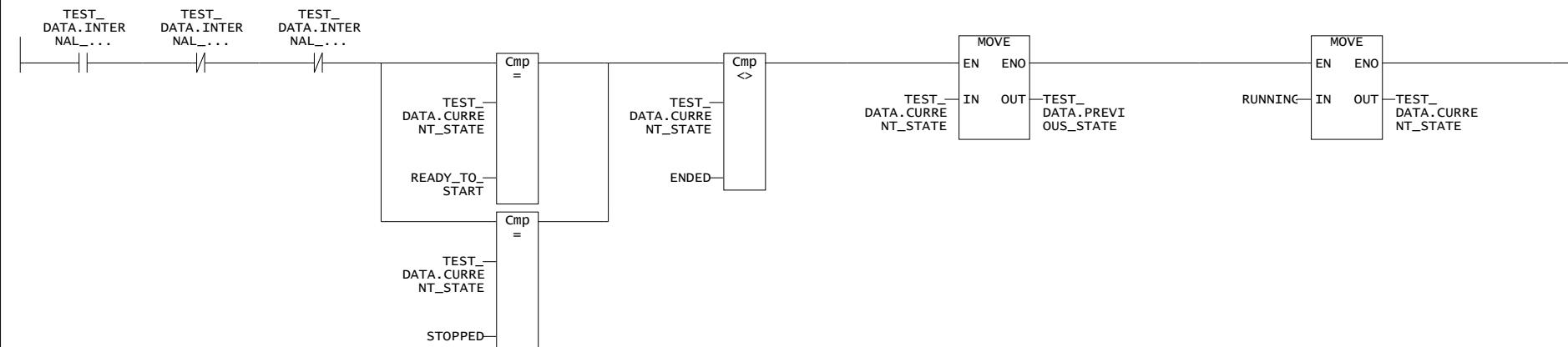
1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

### STATE\_TRANSITIONS - Title

Programa para controlar la transición de estados de máquina

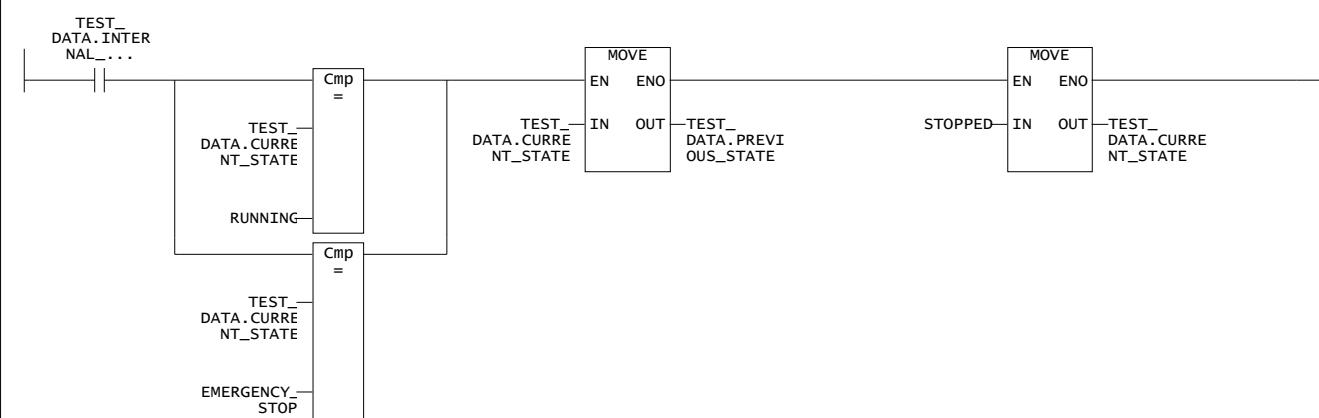
#### 001 - TEST RUNNING STATE TRANSITION

Para transicionar al estado RUNNING, se debe haber recibido un comando de encendido en ausencia de comandos de frenado.



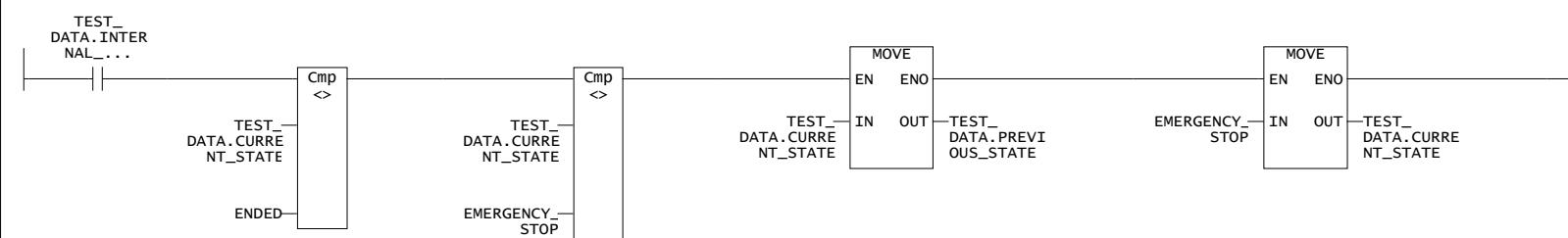
#### 002 - TEST STOPPED STATUS TRANSITION

Para transicionar al estado STOPPED, se debe haber recibido un comando de frenado.



#### 003 - EMERGENCY COMMAND STATE TRANSITION

Para transicionar al estado EMERGENCY\_STOP, se debe haber recibido un comando de frenado de emergencia.



Date	06/07/2025
Editor	
Checked	

24062025SCDIN-MEJORADO

State	Modification	Date	Name	Standard
-------	--------------	------	------	----------

A

B

C

D

E

F

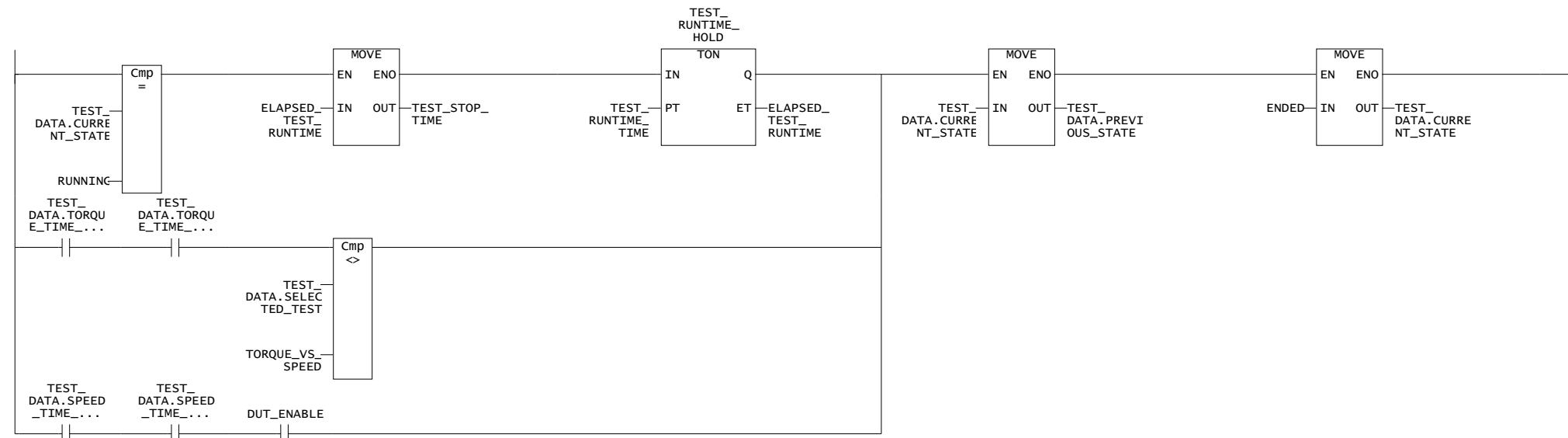
Page 38

Pg. 54

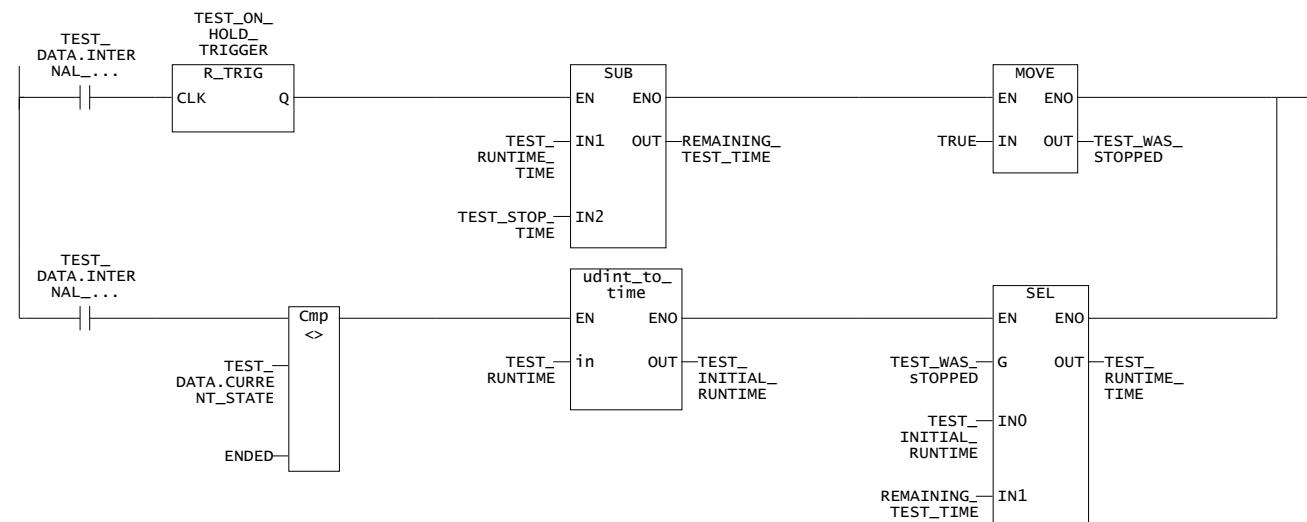
1 2 3 4 5 6 7 8

**004 - TEST ENDED STATE TRANSITION**

Para transicionar al estado ENDED el tiempo de ensayo tiene que haber concluido.  
 La transición solamente considera el tiempo de ensayo efectivo, es decir en modo RUNNING.

**005 - RECALCULATE TIMEOUT ON TEST HOLD**

Red auxiliar para recalcular el tiempo efectivo restante de ensayo cuando se frena el ensayo.

**006 - INITIAL RUNTIME LOAD**

Carga inicial del tiempo total de ensayo antes de iniciar el ensayo. Este valor se utiliza para calcular detener el ensayo después de transcurrir el tiempo establecido.

		Date	06/07/2025				
State	Modification	Date	Name	Standard			
					24062025SCDIN-MEJORADO		

A

B

C

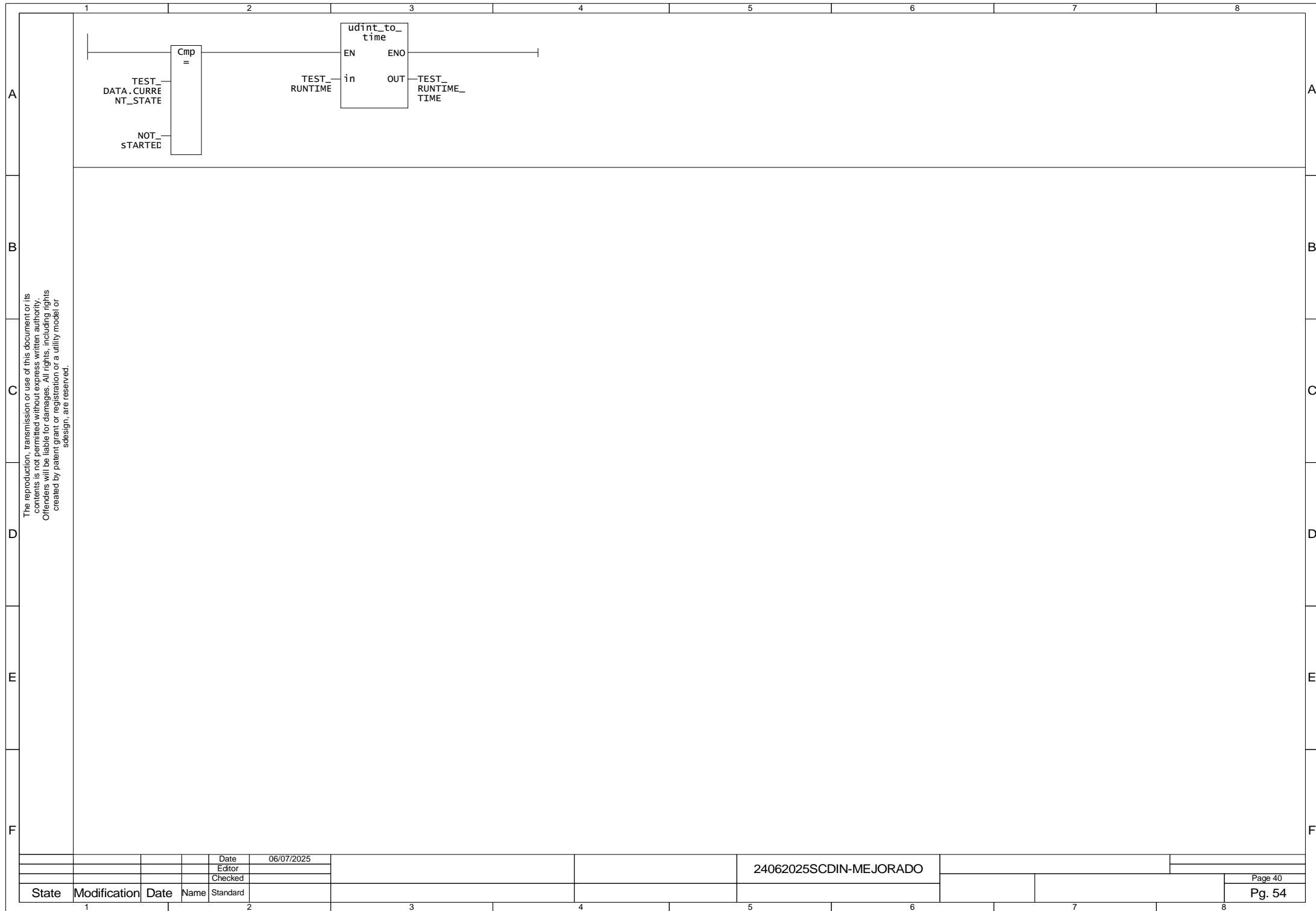
D

E

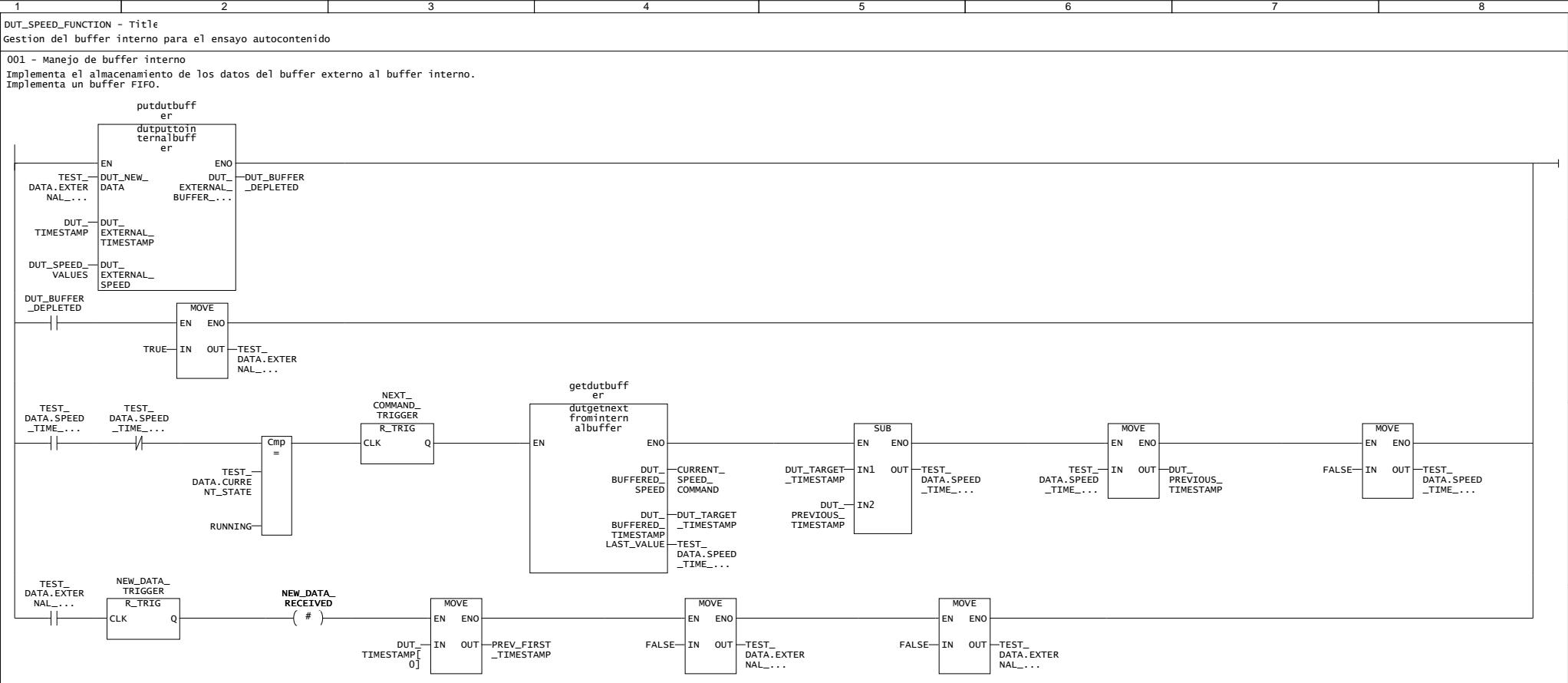
F

Page 39

Pg. 54







The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		



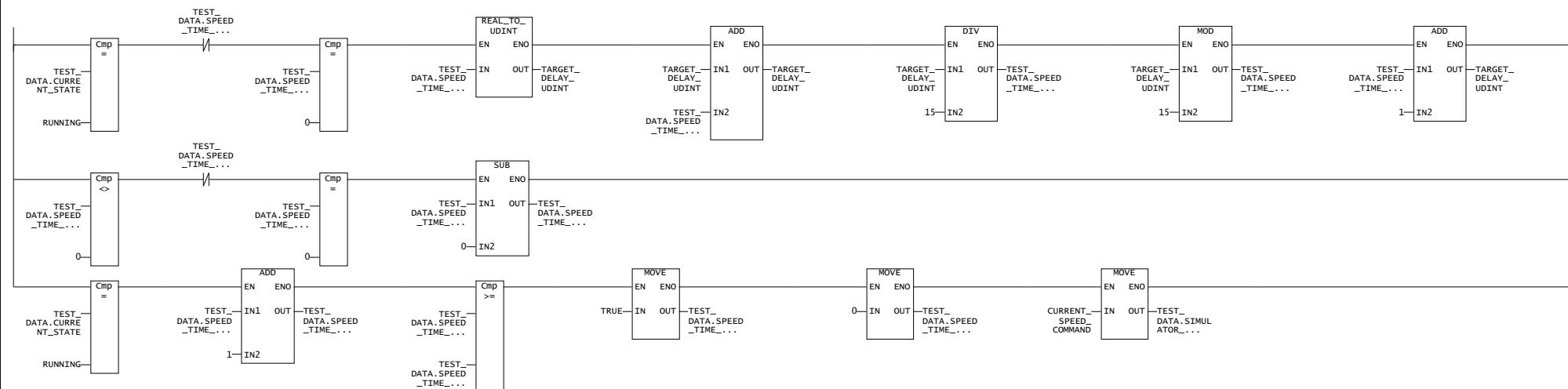
1 2 3 4 5 6 7 8

## DUT\_BUFFER\_TIMER\_TASK - Title

Comment

001 - Alimentación de vector de torque-tiempo

Implementa la lógica para el sincronismo de los pares torque/tiempo. El tiempo entre comandos de torque debe ser de al menos 100ms. El vector de tiempo tiene que ser siempre creciente y sin repetir valores



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

D

E

F

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		

Page 44

Pg. 54

1

2

3

4

5

6

7

8

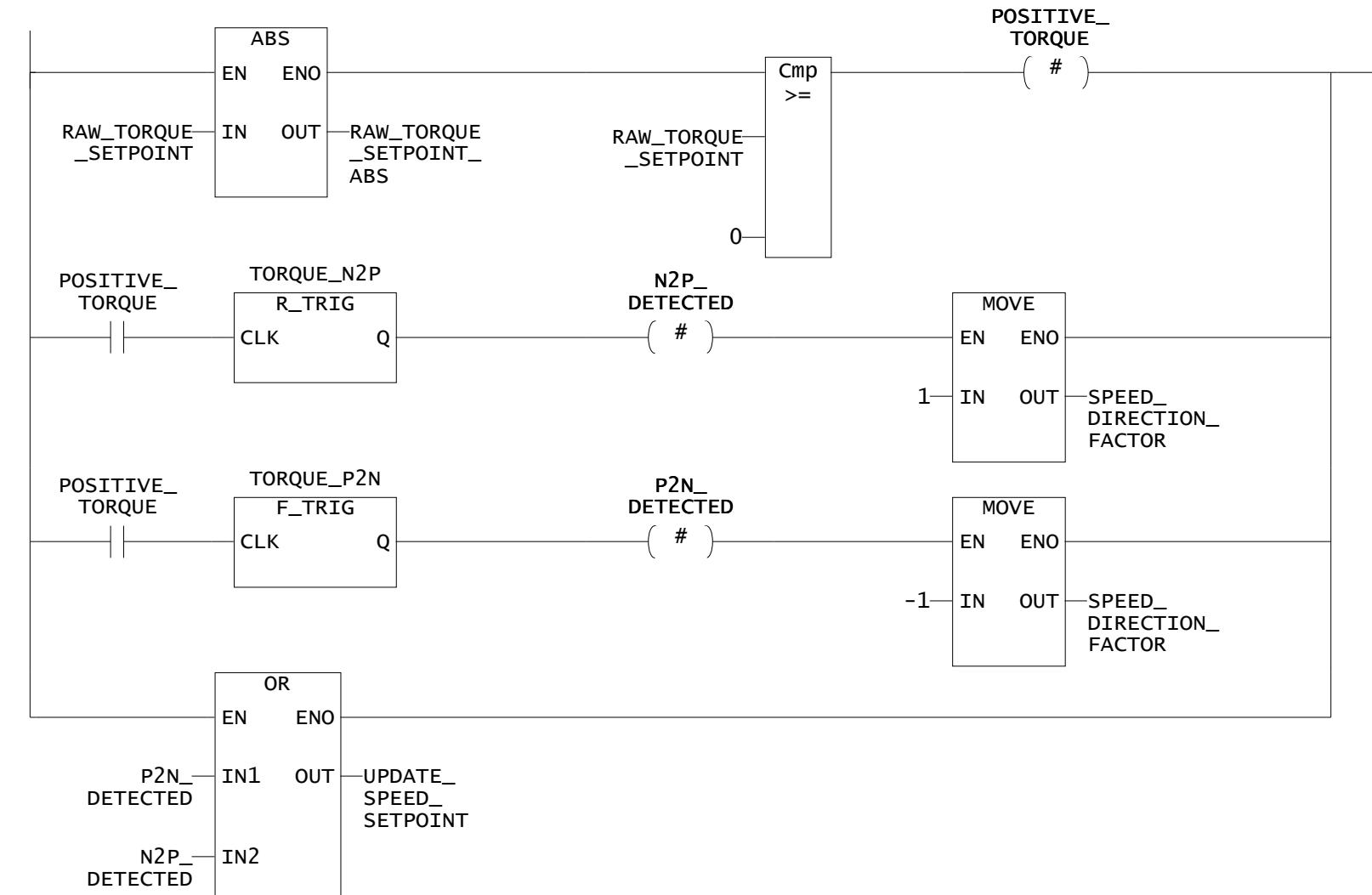


## SPEED\_REVERSAL\_TIMER\_TASK - Title

## Comment

## 001 - SPEED REVERSAL ON NEGATIVE TORQUE

Implementa la reversion de setpoint de velocidad del eje simulador en presencia de una consigna de torque negativa.



## 002 - SIMULATOR SPEED SETPOINT ENABLE

Admite la fijacion de setpoint inicial y la modificacion en caso de cambios en el signo de la ecuacion torque

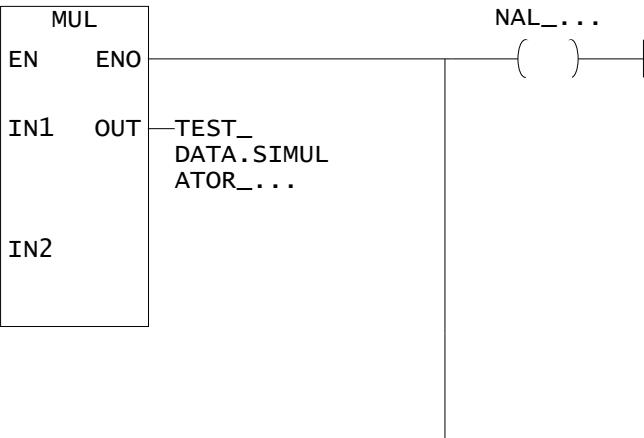
		Date	06/07/2025				
		Editor					
		Checked					
State	Modification	Date	Name	Standard			
1		2		3	4	5	6

24062025SCDIN-MEJORADO

Page 46

Pg. 54

1 2 3 4 5 6 7 8

UPDATE\_  
SPEED\_  
SETPOINTTEST\_  
DATA.INTER  
NAL\_...TEST\_  
DATA.MAX\_  
SIMULAT...SPEED\_  
DIRECTION\_  
FACTORTEST\_  
DATA.INTER  
NAL\_...TEST\_  
DATA.INTER  
NAL\_...TEST\_  
DATA.SIMUL  
ATOR\_...

The reproduction, transmission or use of this document or its contents is not permitted without express written authority.  
Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or design, are reserved.

A

A

B

B

C

C

D

D

E

E

F

F

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		

	1	2	3	4	5	6	7	8			
<b>Program : Properties</b> 24062025SCDIN-MEJORADO\DI425\PROGRAMS\LADDER\Properties											
Properties											
Author					J. Hirschmann	A					
Version					1.1	A					
Comment					Grupo de programas para el banco de ensayos.	A					
Last saved					Domingo, 06 de Julio de 2025 06:21:36 p.m.	A					
Creation type:					Program	A					
Program name:					TORQUE_EQNATION	A					
Author					J. Hirschmann	B					
Version					1.0	B					
Comment					Programa para calcular la ecuacion de cupla y aplicar la limitacion	B					
Declaration table											
Parameters/variables											
Name	Variable type	Data type	Array length	Initial value	Comment						
FILTER_INSTANCE	VAR	FILTER			Instancia de filtro FIR						
RAW_ELAPSED_TIME_MICROSECONDS	VAR	UDINT		1	Tiempo transcurrido desde el ultimo llamado de este POU						
FILTERED_SPEED_PREVIOUS_VALUE	VAR	REAL			Valor anterior de la velocidad del eje simulador						
TORQUE_EQ_INSTANCE	VAR	TORQUEEQUATION			Instancia de ecuacion cupla-velocidad						
FILTERED_SPEED	VAR	REAL			Velocidad medida en el eje simulador filtrada						
I/O symbols											
Name	Absolute identifier	Data type			Comment						
Structures											
Structure name	Element name	Data type	Array length	Initial value	Comment						
Enumerations											
Enumeration name	Element name			Initialization value	Comment						
E											
F											
Date	06/07/2025										
Editor											
Checked											
State	Modification	Date	Name	Standard							
24062025SCDIN-MEJORADO											
Page 48											
Pg. 54											

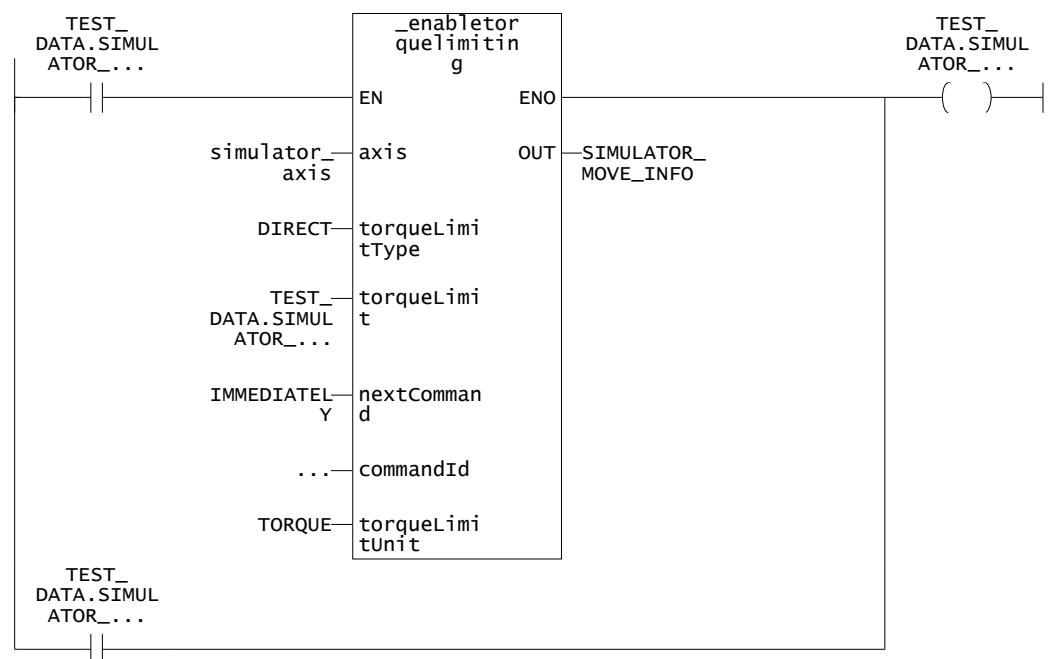
	1	2	3	4	5	6	7	8
--	---	---	---	---	---	---	---	---

## TORQUE\_EQUATION - Title

Comment

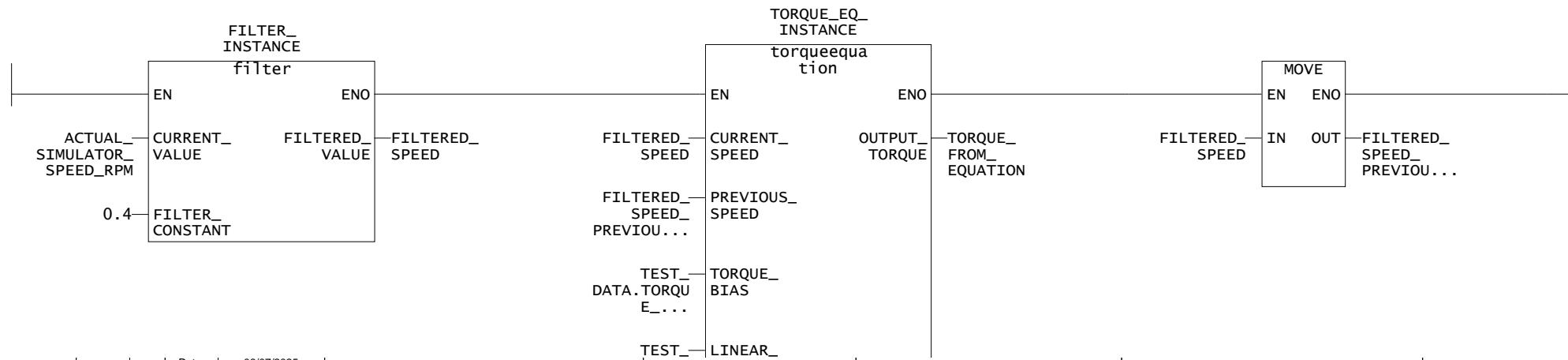
### 001 - TORQUE LIMIT SIMULATOR AXIS

Limita el torque en el eje simulador, espera que el eje este activo y que el setpoint de velocidad este fijado



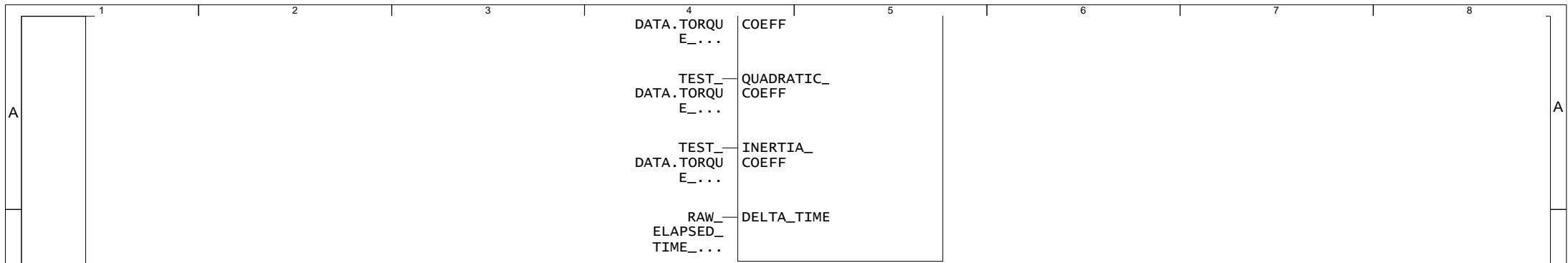
### 002 - Torque equation calculation

Calcula la ecuacion cupla-velocidad utilizando la velocidad filtrada.



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration or a utility model or sdesign, are reserved.

Date	06/07/2025	Editor		Checked		24062025SCDIN-MEJORADO	Page 49
State	Modification	Date	Name	Standard			Pg. 54



The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

A  
B  
C  
D  
E  
F

A

B

C

D

E

F

			Date	06/07/2025				
			Editor					
			Checked					
State	Modification	Date	Name	Standard			24062025SCDIN-MEJORADO	

Page 50

Pg. 54

1 2 3 4 5 6 7 8



	1	2	3	4	5	6	7	8
	Structure name	Element name	Data type	Array length	Initial value	Comment		
A		TEST_ON_HOLD	BOOL		FALSE	VERDADERO SI EL EJE NO ESTA STOP O EMERGENCIA Y EN EL ENSAYO SE HA INICIADO		
		ENABLE_SPEED_SETPOINT	BOOL		FALSE	SEÑAL QUE PERMITE FIJAR UNA NUEVA CONSIGNA DE VELOCIDAD EN EL EJE SIMULADOR	A	
		EXCESS_TORQUE	BOOL		FALSE	FLAG QUE SE ACTIVA POR CUPLA EXCESIVA		
		EXCESS_TORQUE_DERIVATIVE	BOOL		FALSE	FLAG QUE SE ACTIVA POR DERIVADA DE CUPLA EXCESIVA		
	SOFTWARE_INTERFACE	EMERGENCY_BUTTON	BOOL		FALSE	BOTON DE EMERGENCIA POR SOFTWARE		
		START_BUTTON	BOOL		FALSE	BOTON DE INCIO POR SOFTWARE		
B		STOP_BUTTON	BOOL		FALSE	BOTON DE FRENO POR SOFTWARE		
		TEST_READY	BOOL		FALSE	SEÑAL DE CARGA DE PARAMETROS COMPLETADA	B	
		ENABLE_SIMULATOR_AXIS	BOOL		FALSE	SEÑAL DE HABILITACION DE EJE SIMULADOR		
		ENABLE_TRACTION_AXIS	BOOL		FALSE	SEÑAL DE HABILITACION DE EJE DE TRACCION		
C		ENABLE_LINEFEED	BOOL		FALSE	SEÑAL DE HABILITACION DE MODULO DE LINEA		
		TORQUE_VS_SPEED_BUTTON	BOOL		FALSE	SELECCION DE ENSAYO TORQUE-TIEMPO		
		TORQUE_VS_TIME_BUTTON	BOOL		FALSE	SELECCION DE ENSAYO TORQUE-VELOCIDAD	C	
		KEEPALIVE	BOOL		FALSE	SEÑAL DE CONTACTO CON LA INTERFAZ POR SOFTWARE		
		KEEPALIVE_OVERRIDE	BOOL		FALSE	SEÑAL DE DESHABILITACION DE MECANISMO KEEPALIVE		
		DUT_SAVE_TO_BUFFER	BOOL		FALSE	SEÑAL PARA ALMACENAR LOS DATOS DEL BUFFER DE COMUNICACION AL BUFFER INTERNO		
D		SIM_CLEAR_TO_RECEIVE	BOOL		FALSE	SEÑAL DE LIBRE PARA RECIBIR NUEVOS DATOS AL BUFFER DE COMUNICACION	D	
		DUT_CLEAR_TO_RECEIVE	BOOL		FALSE	SEÑAL DE LIBRE PARA RECIBIR NUEVOS DATOS AL BUFFER DE COMUNICACION		
		SIM_SAVE_TO_BUFFER	BOOL		FALSE	SEÑAL PARA ALMACENAR LOS DATOS DEL BUFFER DE COMUNICACION AL BUFFER INTERNO		
E	TORQUE_EQUATION_PARAMS	A	REAL		0	TERMINO CONSTANTE DE LA ECUACION DE CUPLA	E	
		B	REAL		0	TERMINO LINEAL DE LA ECUACION DE CUPLA		
		C	REAL		0	TERMINO CUADRATICO DE LA ECUACION DE CUPLA		
		D	REAL		0	TERMINO INERCIAL DE LA ECUACION DE CUPLA		
F	TIME_VARIABLES	CURRENT_TIMER_CYCLES	UDINT		0	CANTIDAD DE CICLOS A ESPERADOS EN ENSAYO		
		NEXT_COMMAND_UNAVAILABLE	BOOL		FALSE	FLAG DE INEXISTENCIA DE PROXIMO COMANDO DE TIEMPO	F	
		TIMER_CYCLES_TO_WAIT	UDINT		0	CANTIDAD DE CICLOS A ESPERAR EN ENSAYO		
		Date	06/07/2025					
		Editor						
		Checked						
State	Modification	Date	Name	Standard		24062025SCDIN-MEJORADO		
1	2	3	4	5	6	7	8	

1	2	3	4	5	6	7	8
Structure name		Element name	Data type	Array length	Initial value	Comment	
A			PREVIOUS_CYCLE_COMPENSATION	UDINT	0	VALOR EN MS A COMPENSAR DEL CICLO ANTERIOR	
			GET_NEXT	BOOL	FALSE	COMANDO PARA OBTENER EL PROXIMO VALOR DEL BUFFER INTERNO	
			TARGET_DELAY	REAL	0	TIEMPO DE MANTENIMIENTO DE CUPLA ACTUAL	
	CONTEXT_DATA		CURRENT_STATE	TEST_STATE	NOT_STARTED	ESTADO ACTUAL DEL ENSAYO	
			SELECTED_TEST	TEST_TYPE	NOT_SELECTED	TIPO DE ENSAYO SELECCIONADO	
			CURRENT_ERROR	ERROR_TYPE	NO_ERROR	TIPO DE ERROR ACTUAL	
			PREVIOUS_STATE	TEST_STATE	ENDED	ESTADO ANTERIOR DEL ENSAYO	
B			SIMULATOR_AXIS	AXIS		EJE DE SIMULACION	
			TRACTION_AXIS	AXIS		EJE DE TRACCION PARA ENSAYOS AUTOCONTENIDOS	
			LINE_MODULE	LINEFEED		RECTIFICADOR DE LINEA	
			INTERNAL_SIGNALS	SIGNALS		SENALES INTERNAS DEL SISTEMA	
			EXTERNAL_SIGNALS	SOFTWARE_INTERFACE		SENALES DE INTERFAZ CON SOFTWARE OPC	
			TORQUE_EQUATION	TORQUE_EQUATION_PARAMS		PARAMETROS DEL ENSAYO VELOCIDAD-CUPLA	
			TORQUE_TIME_PARAMS	TIME_VARIABLES		PARAMETROS DE ENSAYO TORQUE-TIEMPO	
C			SPEED_TIME_PARAMS	TIME_VARIABLES		PARAMETROS DE ENSAYO AUTOCONTENIDO	
			MAX_TORQUE	REAL	23	Cota de torque maximo	
			MAX_TORQUE_DERIVATIVE	REAL	7000	Maximo valor para la derivada de torque [Nm/s]	
			MAX_SIMULATOR_SPEED	REAL	2500	MAXIMO VALOR DE VELOCIDAD EN MODULO PARA EL EJE SIMULADOR	
	Enumerations						
			Enumeration name		Element name		Initialization value
							Comment
D			TEST_STATE		RUNNING		NOT_STARTED
			STOPPED				
			ENDED				
			NOT_STARTED				
			EMERGENCY_STOP				
			READY_TO_START				
			TEST_TYPE		NOT_SELECTED		
E							
			TORQUE_VS_TIME				
			TORQUE_VS_SPEED				
			MIXED_TEST				
			NOT_SELECTED				
			ERROR_TYPE		NO_ERROR		
F							
			SLM_NOT_ENGAGED				
			AXIS_NOT_ENABLED				
			EMERGENCY_STOP_ENGAGED				
			MAX_TORQUE_DERIVATIVE_EXCEEDED				
			MAX_TORQUE_VALUE_EXCEEDED				
			NO_ERROR				
		Date	06/07/2025				
		Editor					
		Checked					
State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO		
1	2	3	4	5	6	7	8

	1	2	3	4	5	6	7	8											
A	Enumeration name		Element name		Initialization value		Comment												
			TECH_FAULT																
			PERI_FAULT																
			TIMER_FAULT																
			KEEPALIVE_FAILED																
	Connections																		
B	Type	Name		Namespace															
	Program/unit	SPEED_FILTER																	
	Program/unit	BUFFER																	
	Program/unit	CONVERT																	
	Program/unit	DUTBUFFER																	
C	Program/unit	TORQUE_EQUATION																	
	IMPLEMENTATION (source-internal declaration)																		
	Parameter																		
	Name	Variable type		Data type	Array length	Initial value	Comment												
	CURRENT_TORQUE_COMMAND	VAR_GLOBAL		REAL			Comando de cupla actual, es una señal previa a la consigna de cupla.												
	TORQUE_FROM_TIMESTAMP	VAR_GLOBAL		REAL		0	Valor de torque en función del tiempo												
D	CURRENT_SPEED_COMMAND	VAR_GLOBAL		REAL			Comando de velocidad actual, es una señal previa a la consigna de cupla.												
	TORQUE_FROM_EQUATION	VAR_GLOBAL		REAL			Valor de torque en función de la velocidad												
	SIMULATOR_MODE_INFO	VAR_GLOBAL		DINT															
	RAW_TORQUE_SETPOINT	VAR_GLOBAL		REAL			Valor signado de torque												
	I/O symbols																		
E	Name	Absolute identifier		Data type	Comment														
	Structures																		
F	Structure name	Element name		Data type	Array length	Initial value	Comment												
	Enumerations																		
	Enumeration name	Element name		Initialization value		Comment													
Connections																			
G	Type	Name		Namespace															
	Date	06/07/2025																	
	Editor																		
	Checked																		
	State	Modification	Date	Name	Standard	24062025SCDIN-MEJORADO													
Page 54								Pg. 54											

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or sdesign, are reserved.

	1	2	3	4	
A	1 ===== =====				
B	2 //FIUBA 3 //DEPARTAMENTO DE ENERGIA 4 //Copyright 2025 All Rights Reserved 5 //----- 6 // project name: SCDIN 7 // file name: 240620205SCDIN 8 // system: D425 9 // version: SIMOTION SCOUT V4.4 10 // application: SIMULADOR DE CARGA DINAMICA 11 // functionality: Ecuacion cupla-velocidad 12 //----- 13 // change log table: 14 // version date expert in charge changes applied 15 // 16 //=====		A		
C	17 ===== 18 INTERFACE 19     FUNCTION_BLOCK torqueEquation; 20 END_INTERFACE 21 22 IMPLEMENTATION 23 24     FUNCTION_BLOCK TorqueEquation 25         // CALCULA LA ECUACION DE CUPLA 26         VAR CONSTANT 27             // CONSTANTE DEL FILTRO FIR DE PRIMER ORDEN EN EL TERMINO DERIVATIVO 28             FILTER_CONSTANT: REAL:= 0.98; 29         END_VAR 30         VAR 31             FILTERED_INERTIA:REAL; 32         END_VAR 33 34         VAR_INPUT 35             // VELOCIDAD ACTUAL EN RPM 36             CURRENT_SPEED:REAL:=0; 37             // VELOCIDAD ANTERIOR EN RPM 38             PREVIOUS_SPEED:REAL:=0; 39             // TERMINO CONSTANTE DE LA ECUACION [NM] 40             TORQUE_BIAS:REAL:=0; 41             // TERMINO LINEAR DE LA ECUACION [NM/RPM] 42             LINEAR_COEFF:REAL:=0; 43             // TERMINO CUADRATIVO DE LA ECUACION [NM/RPM**2] 44             QUADRATIC_COEFF:REAL:=0; 45             // TERMINO INERCIAL DE LA ECUACION [NM*S/RPM] 46             INERTIA_COEFF:REAL:=0; 47             // TIEMPO TRANSCURRIDO ENTRE LLAMADOS A FUNCION EN MICROSEGUNDOS. 48             DELTA_TIME:UDINT:=1; 49 50         END_VAR 51 52         VAR_OUTPUT 53             // CUPLA CALCULADA [NM] 54             OUTPUT_TORQUE:REAL; 55         END_VAR 56 57         VAR_TEMP 58             CONSTANT_TERM:REAL; 59             LINEAR_TERM:REAL; 60             QUADRATIC_TERM:REAL; 61             DELTA_TIME_SECONDS:REAL:=1;// 62             INERTIA_TERM:REAL; 63             SPEED_DIFF:REAL; 64         END_VAR 65 66         DELTA_TIME_SECONDS := UDINT_TO_REAL(in := DELTA_TIME); 67         CONSTANT_TERM:=TORQUE_BIAS; 68         LINEAR_TERM:=LINEAR_COEFF*ABS(CURRENT_SPEED); 69         QUADRATIC_TERM:=QUADRATIC_COEFF*(CURRENT_SPEED**2); 70         SPEED_DIFF:=CURRENT_SPEED-PREVIOUS_SPEED; 71 72         INERTIA_TERM:=-INERTIA_COEFF*(SPEED_DIFF)/DELTA_TIME_SECONDS; 73         FILTERED_INERTIA:=FILTERED_INERTIA*FILTER_CONSTANT+(1-FILTER_CONSTANT)*INERTIA_TERM;		B	C	
D		Date	06/07/2025		D
E		Editor			E
F		Checked			F
State	Modification	Date	Name	Standard	
24062025SCDIN-MEJORADO					
				Page 1	
				Pg. 2	
1	2	3	4		

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

	1	2	3	4
A				A
B				B
C	The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.			C
D				D
E				E
F				F
	Date	06/07/2025		
	Editor			
	Checked			
State	Modification	Date	Name	Standard
24062025SCDIN-MEJORADO				
				Page 2 Pg. 2
1	2	3	4	

	1	2	3	4
A	1 ===== 2 //FIUBA 3 //DEPARTAMENTO DE ENERGIA 4 //Copyright 2025 All Rights Reserved 5 //----- 6 // project name: SCDIN 7 // file name: 240620205SCDIN 8 // system: D425 9 // version: SIMOTION SCOUT V4.4 10 // application: SIMULADOR DE CARGA DINAMICA 11 // functionality: Filtro FIR 12 //----- 13 // change log table: 14 // version date expert in charge changes applied 15 // 16 ======		A	
B	17 INTERFACE 18 19 FUNCTION_BLOCK Filter; 20 21 END_INTERFACE 22 23 24 25 IMPLEMENTATION 26 27 FUNCTION_BLOCK Filter 28 //IMPLEMENTA UN FILTRO FIR DE PRIMER GRADO CON CONSTANTE DECAIMIENTO FILTER_CONSTANT 29 30 VAR_INPUT 31 // VALOR ACTUAL 32 CURRENT_VALUE:REAL; 33 // CONSTANTE DE DECAIMIENTO 34 FILTER_CONSTANT:REAL; 35 END_VAR 36 37 VAR_OUTPUT 38 //VALOR FILTRADO 39 FILTERED_VALUE:REAL; 40 END_VAR 41 FILTERED_VALUE:=FILTERED_VALUE*FILTER_CONSTANT+(1-FILTER_CONSTANT)*(CURRENT_VALUE); 42 FILTERED_VALUE:=FILTERED_VALUE; 43 END_FUNCTION_BLOCK 44 45 46 47 48 END_IMPLEMENTATION 49		B	
C	The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or sdesign, are reserved.		C	
D			D	
E			E	
F			F	
	Date	06/07/2025		
	Editor			
	Checked			
State	Modification	Date	Name	Standard
	24062025SCDIN-MEJORADO			Page 1
				Pg. 1

	1	2	3	4
A	1 ===== 2 // (FIUBA) 3 // (DEPARTAMENTO DE ENERGIA) 4 // (c)Copyright 2025 All Rights Reserved 5 //----- 6 // project name: SCDIN 7 // file name: 240620205SCDIN 8 // system: D425 9 // Version: SIMOTION SCOUT V4.4 10 // application: SIMULADOR DE CARGA DINAMICA 11 // functionality: Buffer interno del eje simulador 12 //----- 13 // change log table: 14 // version date expert in charge changes applied 15 // 16 //=====		A	
B	17 18 INTERFACE 19 20 FUNCTION_BLOCK putToInternalBuffer; 21 FUNCTION_BLOCK getNextFromInternalBuffer; 22 FUNCTION_BLOCK clearInternalBuffer; 23 24 25 END_INTERFACE 26 27 // **** 28 // * IMPLEMENTATION section * 29 // ****		B	
C	30 31 IMPLEMENTATION 32 VAR_GLOBAL CONSTANT 33 INTERNAL_BUFFER_SIZE: INT :=4096; 34 EXTERNAL_BUFFER_SIZE: INT :=1024; 35 END_VAR 36 37 VAR_GLOBAL 38 //VECTORES DEL BUFFER INTERNO 39 INTERNAL_TIMESTAMP : ARRAY [0..INTERNAL_BUFFER_SIZE-1] OF REAL; 40 INTERNAL_TORQUE : ARRAY [0..INTERNAL_BUFFER_SIZE-1] OF REAL; 41 //INDICE ACTUAL DEL BUFFER 42 CURRENT_INDEX: INT:=0; 43 // CANTIDAD DE ELEMENTOS CARGADOS EN BUFFER 44 BUFFERED_VALUES:INT:=0; 45 // INDICE DEL BUFFER EXTERNO 46 EXTERNAL_BUFFER_INDEX:INT:=0; 47 END_VAR 48 49 50 51 FUNCTION_BLOCK putToInternalBuffer 52 //Carga en el buffer interno (FIFO) 53 // El programa carga valores a un buffer interno de INTERNAL_BUFFER_SIZE 54 // posiciones. Estos valores se cargan con los pares del buffer externo 55 // en funcion de la cantidad de datos procesados. Si el buffer interno tiene 56 // 100 posiciones vacias o ya procesadas entonces recuperara 100 posiciones 57 // del buffer interno. Cuando todos los valores del buffer interno fueron leidos 58 // la funcion pondra en verdadero la señal EXTERNAL_BUFFER_DEPLETED. 59 VAR_INPUT 60 // FLAG QUE INDICA SI SE RECIBIO UNA ACTUALIZACION DEL BUFFER DE COMUNICACION 61 NEW_DATA:BOOL:=TRUE; 62 // VECTORES DEL BUFFER DE COMUNICACION 63 EXTERNAL_TIMESTAMP: ARRAY [0..EXTERNAL_BUFFER_SIZE-1] OF REAL; 64 EXTERNAL_TORQUE: ARRAY [0..EXTERNAL_BUFFER_SIZE-1] OF REAL; 65 END_VAR 66 VAR_OUTPUT 67 // INDICA SI EL BUFFER DE COMUNICACION FUE AGOTADO 68 EXTERNAL_BUFFER_DEPRECATED: BOOL:=TRUE; 69 END_VAR 70 VAR_TEMP 71 // CANTIDAD DE VALORES A RESTITUIR DEL BUFFER INTERNO 72 MISSING_VALUES_FROM_BUFFER:INT:=0; 73 // CANTIDAD DE VALORES QUE FALTAN PROCESAR DEL BUFFER EXTERNO		C	
D		Date	06/07/2025	
E		Editor		
F		Checked		
State	Modification	Date	Name	Standard
240620205SCDIN-MEJORADO				
			Page 1	
			Pg. 3	

	1	2	3	4			
A	74            REMAINING_VALUES_FROM_EXTERNAL_BUFFER:INT:=0; 75            // CANTIDAD DE VALORES A CARGAR 76            VALUES_TO_BUFFER:INT:=0; 77            i:INT:=0; 78            INTERNAL_BUFFER_LOAD_INDEX:INT:=0; 79            END_VAR 80 81            IF NEW_DATA THEN 82              EXTERNAL_BUFFER_INDEX:=0; 83              EXTERNAL_BUFFER_DEPLETED:=FALSE; 84            END_IF; 85            MISSING_VALUES_FROM_BUFFER:=INTERNAL_BUFFER_SIZE-BUFFERED_VALUES; 86            REMAINING_VALUES_FROM_EXTERNAL_BUFFER:=EXTERNAL_BUFFER_SIZE-EXTERNAL_BUFFER_INDEX; 87 88            IF REMAINING_VALUES_FROM_EXTERNAL_BUFFER=0 THEN 89              EXTERNAL_BUFFER_DEPLETED:=TRUE; 90            END_IF; 91 92 93            VALUES_TO_BUFFER:=MIN(MISSING_VALUES_FROM_BUFFER,REMAINING_VALUES_FROM_EXTERNAL_BUFFER); 94            IF VALUES_TO_BUFFER>0 AND NOT EXTERNAL_BUFFER_DEPLETED THEN 95 96              FOR i := 0 TO VALUES_TO_BUFFER-1 BY 1 DO 97                 INTERNAL_BUFFER_LOAD_INDEX:=ABS(CURRENT_INDEX+BUFFERED_VALUES-1); 98                 INTERNAL_BUFFER_LOAD_INDEX:=(INTERNAL_BUFFER_LOAD_INDEX) MOD (INTERNAL_BUFFER_SIZE); 99 100            INTERNAL_TORQUE[INTERNAL_BUFFER_LOAD_INDEX]:=EXTERNAL_TORQUE[EXTERNAL_BUFFER_INDEX]; 101            MISSING_VALUES_FROM_BUFFER:=MISSING_VALUES_FROM_BUFFER-1; 102            EXTERNAL_BUFFER_INDEX:=EXTERNAL_BUFFER_INDEX+1; 103            BUFFERED_VALUES:=BUFFERED_VALUES+1; 104            END_FOR; 105 106            END_IF; 107            END_FUNCTION_BLOCK 108 109            FUNCTION_BLOCK getNextFromInternalBuffer 110            // Extrae un par del buffer interno (FIFO) 111            VAR 112              // INDICE DEL VALOR A ENTREGAR 113              NEXT_INDEX: INT:=0; 114              // ETIQUETA TEMPORAL A ENTREGAR 115              NEXT_TIMESTAMP:REAL:=0; 116            END_VAR 117            VAR_OUTPUT 118              // TORQUE 119              BUFFERED_TORQUE:REAL; 120              // TIEMPO 121              BUFFERED_TIMESTAMP:REAL; 122              // INDICA QUE NO HAY SIGUIENTE VALOR A ENTREGAR 123              LAST_VALUE:BOOL:=FALSE; 124            END_VAR 125 126            IF CURRENT_INDEX=INTERNAL_BUFFER_SIZE THEN 127              CURRENT_INDEX:=0; 128            END_IF; 129            NEXT_INDEX:=CURRENT_INDEX+1; 130            BUFFERED_TORQUE:=INTERNAL_TORQUE[CURRENT_INDEX]; 131            BUFFERED_TIMESTAMP:=INTERNAL_TIMESTAMP[CURRENT_INDEX]; 132            CURRENT_INDEX:=CURRENT_INDEX+1; 133            BUFFERED_VALUES:=BUFFERED_VALUES-1; 134            NEXT_TIMESTAMP:=INTERNAL_TIMESTAMP[NEXT_INDEX]; 135            LAST_VALUE:=FALSE; 136 137            IF NEXT_TIMESTAMP-BUFFERED_TIMESTAMP<=0 THEN 138              LAST_VALUE:=TRUE; 139 140            END_IF; 141            END_FUNCTION_BLOCK 142            FUNCTION_BLOCK clearInternalBuffer 143            // INICIALIZA EL BUFFER INTERNO. 144            VAR_TEMP 145              i:INT; 146            END_VAR	A	B	C	D	E	F
		Date	06/07/2025				
		Editor					
		Checked					
State	Modification	Date	Name	Standard			
24062025SCDIN-MEJORADO					Page 2		
					Pg. 3		
1	2	3	4				

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or sdesign, are reserved.

	1	2	3	4
A	147       FOR i := 0 TO INTERNAL_BUFFER_SIZE-1 BY 1 DO 148           INTERNAL_TORQUE[i]:=0.0; 149           INTERNAL_TIMESTAMP[i]:=0; 150           CURRENT_INDEX:=i; 151 152       END_FOR; 153       i:=0; 154       CURRENT_INDEX:=0; 155       BUFFERED_VALUES:=0; 156       EXTERNAL_BUFFER_INDEX:=0; 157     END_FUNCTION_BLOCK 158 159     // ****= 160     // * PROGRAM * 161     // ****= 162 163 164   END_IMPLEMENTATION 165 166		A	
B			B	
C	The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.		C	
D			D	
E			E	
F			F	
	Date	06/07/2025		
	Editor			
	Checked			
State	Modification	Date	Name	Standard
24062025SCDIN-MEJORADO				
			Page 3	
			Pg. 3	
1	2	3	4	

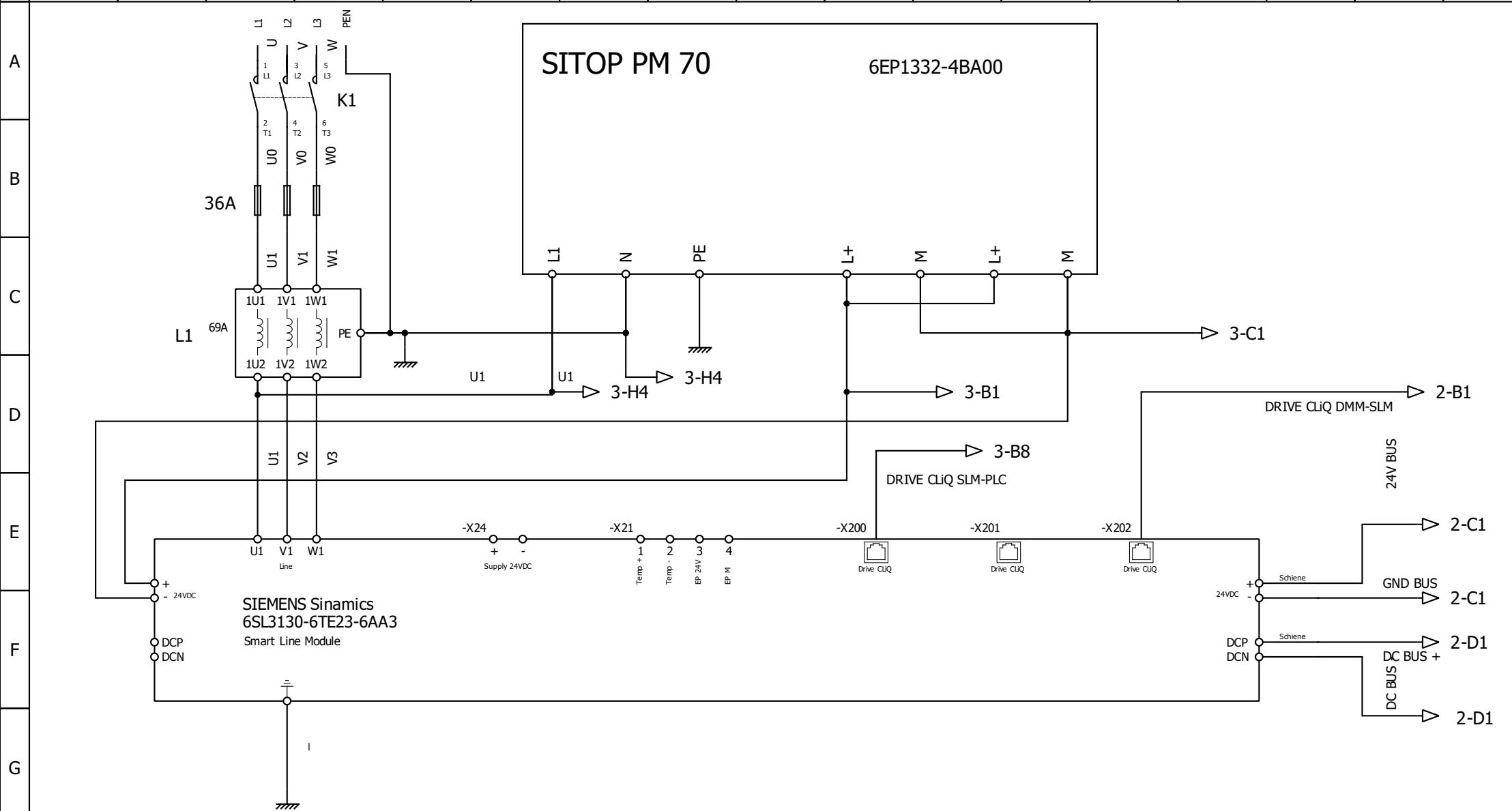
	1	2	3	4
A	1 ===== 2 //FIUBA 3 //DEPARTAMENTO DE ENERGIA 4 //Copyright 2025 All Rights Reserved 5 //----- 6 // project name: SCDIN 7 // file name: 240620205SCDIN 8 // system: D425 9 // version: SIMOTION SCOUT V4.4 10 // application: SIMULADOR DE CARGA DINAMICA 11 // functionality: Conversion de mediciones 12 //----- 13 // change log table: 14 // version date expert in charge changes applied 15 // 16 //=====		A	
B	17 18 INTERFACE 19 20 FUNCTION_BLOCK convertToPct; 21 END_INTERFACE		B	
C	22 23 IMPLEMENTATION 24 25 VAR_GLOBAL CONSTANT 26 REFERENCE_VALUE: REAL :=16384.0; 27 END_VAR		C	
D	28 29 FUNCTION_BLOCK convertToPct 30 //CONVIERTA VALORES BINARIOS EN PUNTO FLOTANTE DE 32 BITS. 31 VAR_INPUT 32 // VALOR A CONVERTIR. 33 value: INT; 34 END_VAR 35 VAR_OUTPUT 36 // VALOR CONVETIDO 37 output: REAL; 38 END_VAR 39 40 VAR_TEMP 41 // VALORES AUXILIARES INTERMEDIOS 42 auxDint:DINT; 43 auxReal:REAL; 44 END_VAR 45 auxDint:= INT_TO_DINT(in :=value ); 46 auxReal :=DINT_TO_REAL(in :=auxDint ); 47 auxReal:= auxReal/REFERENCE_VALUE; 48 output:=auxReal; 49 END_FUNCTION_BLOCK 50 51 END_IMPLEMENTATION		D	
E	52 53		E	
F			F	
		Date 06/07/2025		
		Editor		
		Checked		
State	Modification	Date	Name	Standard
	24062025SCDIN-MEJORADO			Page 1
				Pg. 1

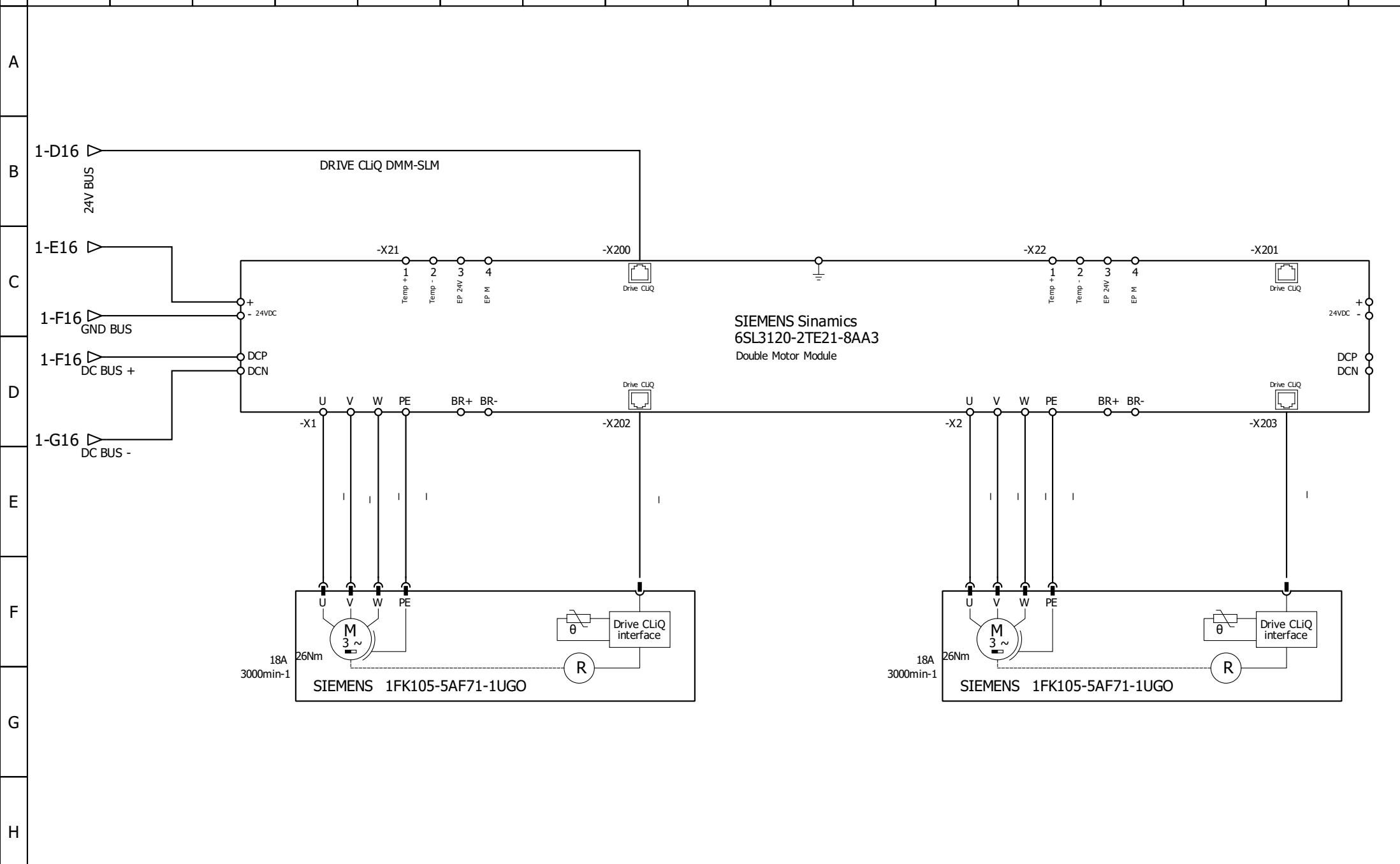
	1	2	3	4
A	1 ===== 2 // (FIUBA) 3 // (DEPARTAMENTO DE ENERGIA) 4 // (c)Copyright 2025 All Rights Reserved 5 //----- 6 // project name: SCDIN 7 // file name: 240620205SCDIN 8 // system: D425 9 // version: SIMOTION SCOUT V4.4 10 // application: SIMULADOR DE CARGA DINAMICA 11 // functionality: Buffer interno del eje de traccion 12 //----- 13 // change log table: 14 // version date expert in charge changes applied 15 // 16 //=====		A	
B	17 18 INTERFACE 19 20 FUNCTION_BLOCK dutPutToInternalBuffer; 21 FUNCTION_BLOCK dutGetNextFromInternalBuffer; 22 FUNCTION_BLOCK dutClearInternalBuffer; 23 24 25 END_INTERFACE 26 27 // ***** 28 // * IMPLEMENTATION section * 29 // *****		B	
C	30 31 IMPLEMENTATION 32 VAR_GLOBAL CONSTANT 33 DUT_INTERNAL_BUFFER_SIZE: INT :=4096; 34 DUT_EXTERNAL_BUFFER_SIZE: INT :=1024; 35 END_VAR 36 37 VAR_GLOBAL 38 39 //VECTORES DEL BUFFER INTERNO 40 DUT_INTERNAL_TIMESTAMP : ARRAY [0..DUT_INTERNAL_BUFFER_SIZE-1] OF REAL; 41 DUT_INTERNAL_SPEED : ARRAY [0..DUT_INTERNAL_BUFFER_SIZE-1] OF REAL; 42 //INDICE ACTUAL DEL BUFFER 43 DUT_CURRENT_INDEX: INT:=0; 44 // CANTIDAD DE ELEMENTOS CARGADOS EN BUFFER 45 DUT_BUFFERED_VALUES:INT:=0; 46 // INDICE DEL BUFFER EXTERNO 47 DUT_EXTERNAL_BUFFER_INDEX:INT:=0; 48 END_VAR 49 50 51		C	
D	52 FUNCTION_BLOCK dutPutToInternalBuffer 53 //Carga en el buffer interno (FIFO) 54 // El programa carga valores a un buffer interno de INTERNAL_BUFFER_SIZE 55 // posiciones. Estos valores se cargan con los pares del buffer externo 56 // en funcion de la cantidad de datos procesados. Si el buffer interno tiene 57 // 100 posiciones vacias o ya procesadas entonces recuperara 100 posiciones 58 // del buffer interno. Cuando todos los valores del buffer interno fueron leidos 59 // la funcion pondra en verdadero la senal EXTERNAL_BUFFER_DEPRECATED. 60 VAR_INPUT 61 // FLAG QUE INDICA SI SE RECIBIO UNA ACTUALIZACION DEL BUFFER DE COMUNICACION 62 DUT_NEW_DATA:BOOL:=TRUE; 63 // VECTORES DEL BUFFER DE COMUNICACION 64 DUT_EXTERNAL_TIMESTAMP: ARRAY [0..DUT_EXTERNAL_BUFFER_SIZE-1] OF REAL; 65 DUT_EXTERNAL_SPEED: ARRAY [0..DUT_EXTERNAL_BUFFER_SIZE-1] OF REAL; 66 END_VAR 67 VAR_OUTPUT 68 // INDICA SI EL BUFFER DE COMUNICACION FUE AGOTADO 69 DUT_EXTERNAL_BUFFER_DEPRECATED: BOOL:=TRUE; 70 END_VAR 71 VAR_TEMP 72 // CANTIDAD DE VALORES A RESTITUIR DEL BUFFER INTERNO 73 DUT_MISSING_VALUES_FROM_BUFFER:INT:=0;		D	
E		Date	06/07/2025	
F		Editor		
		Checked		
State	Modification	Date	Name	Standard
	24062025SCDIN-MEJORADO			Page 1
				Pg. 3

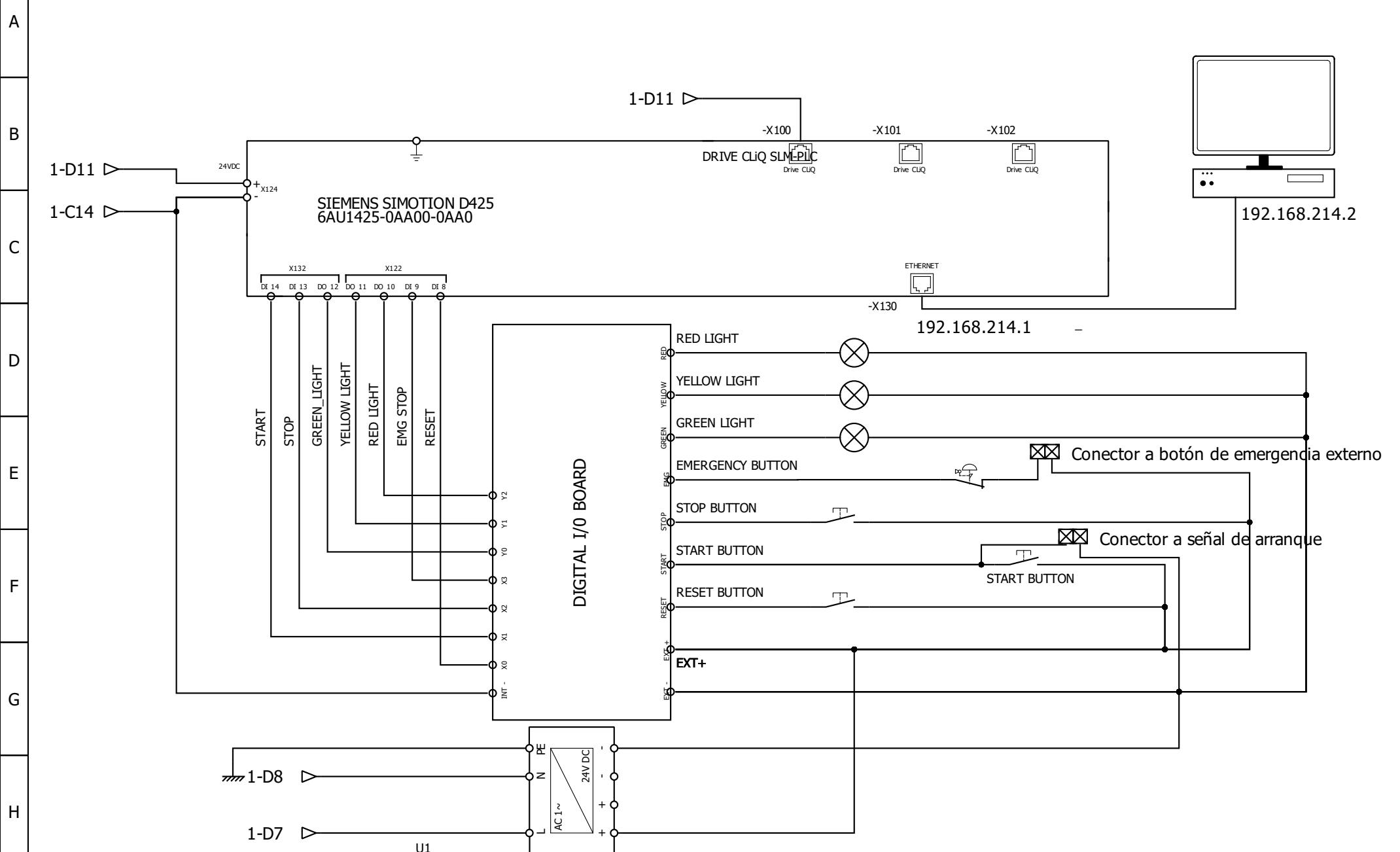
	1	2	3	4
A	74 // CANTIDAD DE VALORES QUE FALTAN PROCESAR DEL BUFFER EXTERNO 75 DUT_REMAINING_VALUES_FROM_EXTERNAL_BUFFER:Int:=0; 76 // CANTIDAD DE VALORES A CARGAR 77 DUT_VALUES_TO_BUFFER:Int:=0; 78 i:Int:=0; 79 DUT_INTERNAL_BUFFER_LOAD_INDEX:Int:=0; 80 END_VAR 81 82 IF DUT_NEW_DATA THEN 83 DUT_EXTERNAL_BUFFER_INDEX:=0; 84 DUT_EXTERNAL_BUFFER_DEPRECATED:=FALSE; 85 END_IF; 86 //Cantidad de valores que se procesaron del buffer y no se restituyeron 87 DUT_MISSING_VALUES_FROM_BUFFER:=DUT_INTERNAL_BUFFER_SIZE-DUT_BUFFERED_VALUES; 88 // Valores que faltan procesar del buffer externo 89 DUT_REMAINING_VALUES_FROM_EXTERNAL_BUFFER:=DUT_EXTERNAL_BUFFER_SIZE-DUT_EXTERNAL_BUFFER_IN DEX;			A
B	90 91 IF DUT_REMAINING_VALUES_FROM_EXTERNAL_BUFFER=0 THEN 92 //Pido nuevos datos 93 //SUPONE QUE EL PROXIMO LLAMADO A FUNCION SERA CON DATOS NUEVOS 94 DUT_EXTERNAL_BUFFER_DEPRECATED:=TRUE; 95 END_IF; 96 97 DUT_VALUES_TO_BUFFER:=MIN(DUT_MISSING_VALUES_FROM_BUFFER,DUT_REMAINING_VALUES_FROM_EXTERN AL_BUFFER); 98 IF DUT_VALUES_TO_BUFFER>0 AND NOT DUT_EXTERNAL_BUFFER_DEPRECATED THEN 99 100 FOR i := 0 TO DUT_VALUES_TO_BUFFER-1 BY 1 DO 101 DUT_INTERNAL_BUFFER_LOAD_INDEX:=ABS(DUT_CURRENT_INDEX+DUT_BUFFERED_VALUES-1);//SIEMPRE CARGO ADELANTE DEL INDICE 102 DUT_INTERNAL_BUFFER_LOAD_INDEX:=(DUT_INTERNAL_BUFFER_LOAD_INDEX) MOD (DUT_INTERNAL_BUFFER_SIZE); 103 DUT_INTERNAL_SPEED[DUT_INTERNAL_BUFFER_LOAD_INDEX]:=DUT_EXTERNAL_SPEED[DUT_EXTERNAL_BUFFER INDEX]; 104 DUT_INTERNAL_TIMESTAMP[DUT_INTERNAL_BUFFER_LOAD_INDEX]:=DUT_EXTERNAL_TIMESTAMP[DUT_EXTERNAL _BUFFER_INDEX]; 105 //DEBERIA IR DE CURRENT_INDEX-i, pero evitar valores negativos 106 DUT_MISSING_VALUES_FROM_BUFFER:=DUT_MISSING_VALUES_FROM_BUFFER-1; 107 DUT_EXTERNAL_BUFFER_INDEX:=DUT_EXTERNAL_BUFFER_INDEX+1; 108 DUT_BUFFERED_VALUES:=DUT_BUFFERED_VALUES+1; 109 END_FOR; 110 END_IF; 112 END_FUNCTION_BLOCK		B	
C	113  FUNCTION_BLOCK dutGetNextFromInternalBuffer 114 // Extrae un par del buffer interno (FIFO) 115 VAR 116 //INDICE DEL VALOR A ENTREGAR 117 NEXT_INDEX: INT:=0; 118 //ETIQUETA TEMPORAL A ENTREGAR 119 NEXT_TIMESTAMP:REAL:=0; 120 END_VAR 121 VAR_OUTPUT 122 // TORQUE 123 DUT_BUFFERED_SPEED:REAL; 124 // TIEMPO 125 DUT_BUFFERED_TIMESTAMP:REAL; 126 // INDICA QUE NO HAY SIGUIENTE VALOR A ENTREGAR 127 LAST_VALUE:BOOL:=FALSE; 128 END_VAR 129 IF DUT_CURRENT_INDEX=DUT_INTERNAL_BUFFER_SIZE THEN 130 DUT_CURRENT_INDEX:=0; 131 END_IF; 132 NEXT_INDEX:=DUT_CURRENT_INDEX+1; 133 DUT_BUFFERED_SPEED:=DUT_INTERNAL_SPEED[DUT_CURRENT_INDEX]; 134 DUT_BUFFERED_TIMESTAMP:=DUT_INTERNAL_TIMESTAMP[DUT_CURRENT_INDEX]; 135 DUT_CURRENT_INDEX:=DUT_CURRENT_INDEX+1; 136 DUT_BUFFERED_VALUES:=DUT_BUFFERED_VALUES-1; 137 NEXT_TIMESTAMP:=DUT_INTERNAL_TIMESTAMP[NEXT_INDEX]; 138 LAST_VALUE:=FALSE; 139		C	
D		Date	06/07/2025	
E		Editor		
F		Checked		
State	Modification	Date	Name	Standard
24062025SCDIN-MEJORADO				
			Page 2	
			Pg. 3	

	1	2	3	4
A	140 141     IF NEXT_TIMESTAMP-DUT_BUFFERED_TIMESTAMP<=0 THEN 142         LAST_VALUE:=TRUE; 143 144         END_IF; 145     END_FUNCTION_BLOCK 146     FUNCTION_BLOCK dutClearInternalBuffer 147         //INICIALIZA EL BUFFER INTERNO. 148         VAR_TEMP 149             i:INT; 150         END_VAR 151     FOR i := 0 TO DUT_INTERNAL_BUFFER_SIZE-1 BY 1 DO 152         DUT_INTERNAL_SPEED[i]:=0.0; 153         DUT_INTERNAL_TIMESTAMP[i]:=0; 154         DUT_CURRENT_INDEX:=i; 155 156     END_FOR; 157     i:=0; 158     DUT_CURRENT_INDEX:=0; 159     DUT_BUFFERED_VALUES:=0; 160     DUT_EXTERNAL_BUFFER_INDEX:=0; 161     END_FUNCTION_BLOCK 162 163     // **** 164     // * PROGRAM * 165     // **** 166 167 168     END_IMPLEMENTATION 169 170		A	
B			B	
C	The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.		C	
D			D	
E			E	
F			F	
		Date	06/07/2025	
		Editor		
		Checked		
State	Modification	Date	Name	Standard
24062025SCDIN-MEJORADO				
			Page 3	
			Pg. 3	

### 15.3. Plano eléctrico







#### 15.4. Circuito esquemático - Placa de entradas y salidas

