

Jornadas de Automática

Perspectivas en integración de sistemas robóticos y autómatas programables mediante ROS2

Cruz, Juan Diego*, Oña, Edwin Daniel, Martínez, Santiago

Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911, Leganés, Spain.

To cite this article: Cruz, J. D., Oña, E.D., Martínez, S. 2026. Perspectives on the integration of robotic systems and programmable logic controllers using ROS2. *Jornadas de Automática*, 47.
<https://doi.org/10.17979/ja-cea.2026.47.13875>

Resumen

La integración de ROS 2 con controladores lógicos programables (PLC) se ha abordado tradicionalmente mediante protocolos industriales genéricos (MODBUS TCP, OPC UA, MQTT), que introducen latencias de red incompatibles con el control cinemático cerrado. Frente a este enfoque, los principales fabricantes de automatización (Siemens, Beckhoff, Phoenix Contact, Bosch Rexroth, Schneider Electric y Rockwell) han desarrollado conectores nativos que alojan ROS 2 en el mismo hardware del PLC mediante hipervisores y memoria compartida. Este artículo presenta una revisión comparativa de seis arquitecturas de integración nativa, analizando su mecanismo de comunicación, latencia, determinismo y madurez tecnológica, y contrastándolas con los protocolos industriales convencionales. Se identifica la interconexión de particiones independientes comunicadas por medio de memoria compartida como el patrón arquitectónico óptimo, ejemplificado por el conector SIMATIC ROXSIE de Siemens y el hipervisor TwinCAT/BSD de Beckhoff. Finalmente, se discuten las implicaciones de esta arquitectura para Laboratorios Inteligentes (*Self-Driving Labs*), donde la coexistencia de PLCs deterministas y robots heterogéneos basados en ROS 2 exige garantías simultáneas de tiempo real y seguridad funcional.

Palabras clave: ROS 2, PLC, conectores nativos, ROXSIE, TwinCAT/BSD, ctrlX CORE, memoria compartida, integración IT/OT, tiempo real, laboratorios autónomos

Perspectives on the integration of robotic systems and programmable logic controllers using ROS2

Abstract

Historically, the integration of ROS 2 with Programmable Logic Controllers (PLCs) has been achieved using standard industrial protocols (such as MODBUS TCP, OPC UA, MQTT). However, these conventional approaches introduce network latencies that preclude strict closed-loop kinematic control. Addressing this constraint, leading automation manufacturers (Siemens, Beckhoff, Phoenix Contact, Bosch Rexroth, Schneider Electric, and Rockwell) have developed native integration solutions. These architectures co-execute ROS 2 and the PLC runtime on a single hardware platform through the implementation of hypervisors and shared memory techniques. This paper provides a comparative review of six native integration architectures, assessing their communication mechanisms, latency, determinism, and technological maturity in contrast to traditional industrial protocols. The interconnection of independent execution partitions via shared memory is identified as the optimal architectural pattern, a paradigm best illustrated by Siemens' SIMATIC ROXSIE connector and Beckhoff's TwinCAT/BSD hypervisor. Ultimately, the implications of this architecture for Self-Driving Labs are discussed, where the coexistence of deterministic PLCs and heterogeneous ROS 2-based robots demands simultaneous guarantees of real-time and functional-safety.

Keywords: ROS 2, PLC, native connectors, ROXSIE, TwinCAT/BSD, ctrlX CORE, shared memory, IT/OT integration, real-time, self-driving labs

*Autor para correspondencia: 100575537@alumnos.uc3m.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

1. Introducción

La manufactura contemporánea y los entornos de investigación de vanguardia se enfrentan a la necesidad de unir dos mundos computacionales históricamente aislados: las Tecnologías de la Información (IT) y las Tecnologías de Operación (OT). El dominio OT, gobernado por controladores lógicos programables (PLC), proporciona el determinismo y la seguridad funcional necesarios para ejecutar ciclos de control predecibles, típicamente entre 1 ms y 10 ms, cumpliendo niveles SIL 3 o PL e según la IEC 62061 e ISO 13849 (Thramboulidis, 2015). Sin embargo, estos sistemas resultan cognitivamente rígidos para procesar datos no estructurados como visión artificial o planificación dinámica de trayectorias.

El dominio IT ha consolidado plataformas abiertas como ROS 2, que incorpora el *middleware* DDS (*Data Distribution Service*) y generaliza el acceso a algoritmos de localización y mapeo simultáneo (SLAM), control adaptativo y orquestación distribuida de flotas (ROS-Industrial Consortium, 2024). La integración directa de ambos mundos mediante protocolos industriales genéricos —MODBUS TCP, OPC UA, MQTT— introduce latencias de red de hasta 500 ms y *jitter* inadmisibles para el control directo de actuadores (Profanter et al., 2019).

Frente a esta limitación estructural, los principales fabricantes de autómatas programables han adoptado un enfoque distinto, integrando ambos entornos en el mismo *hardware* mediante hipervisores y memoria compartida, en lugar de interconectar ROS 2 y el PLC mediante red.

En este artículo se revisan de forma comparativa las soluciones proporcionadas por los principales fabricantes de autómatas programables (Siemens, Beckhoff, Phoenix Contact, Bosch Rexroth, Schneider Electric y Rockwell) para ejecutar de forma nativa comunicación mediante ROS 2. La integración de PLCs y robots mediante comunicación ROS 2 es de interés en aplicaciones industriales donde se requieren control de procesos con funciones de programación que normalmente no están disponibles en un PLC. Por ello, se presenta como caso de uso el desarrollo de Laboratorios Inteligentes (*Self-Driving Labs*, SDL), un entorno donde coexisten equipos específicos, sensores, y robots de diferente morfología.

La organización del artículo es la siguiente: la Sección 2 introduce los fundamentos de la integración IT/OT y las limitaciones de los protocolos de red. La Sección 3 analiza las arquitecturas nativas para ejecutar ROS 2 en PLCs. La Sección 4.1 presenta las perspectivas en el uso de ROS 2 para integrar dispositivos PLCs y robots, describiendo el caso de aplicación en Laboratorios Inteligentes. La Sección 5 recoge las conclusiones.

2. Fundamentos de la integración IT/OT

El dominio OT constituye el sistema nervioso autónomo de la planta industrial. Los PLCs, los relés de seguridad y los módulos de entradas/salidas están diseñados para trabajar en condiciones exigentes y totalmente controladas donde una parada de emergencia detiene los actuadores con una latencia garantizada. Sin embargo, la arquitectura estática de los PLCs convencionales carece de la capacidad de procesar datos no estructurados necesarios para la toma de decisiones dinámica,

como los modelos de IA o la visión computacional (Thramboulidis, 2015).

ROS 2, sucesor de ROS 1 con soporte nativo para sistemas distribuidos, se ha consolidado como la plataforma de referencia para el software robótico industrial (ROS-Industrial Consortium, 2024). Su *middleware* DDS introduce mecanismos de calidad de servicio (QoS) que mejoran el determinismo respecto a su predecesor, aunque estudios recientes confirman que persisten desafíos de latencia (*jitter*) y de gestión de memoria dinámica que impiden su uso nativo para ciclos de control por debajo del milisegundo (Puck et al., 2021; Casini et al., 2025).

La interconexión tradicional ROS 2–PLC mediante MODBUS TCP, OPC UA o MQTT (Tabla 1) hereda esta limitación. Cada mensaje atraviesa la pila completa del modelo OSI (*Open Systems Interconnection*), con serialización, copiado por sockets y tránsito de red, penalizando la latencia efectiva (Maruyama et al., 2016). El potencial de ROS 2 para la colaboración entre múltiples empresas mediante *middleware* estándar ha sido analizado por Daubaris et al. (2024), quienes documentan casos de uso reales con vehículos guiados automáticamente, si bien limitados a tareas de coordinación de alto nivel, no de control cinemático cerrado.

Tabla 1: Protocolos industriales genéricos para integración IT/OT.

Protocolo	Latencia	Determinismo
MODBUS TCP	10–100 ms	No
MQTT	50–500 ms	No
OPC UA (TSN)	1–10 ms	Soft RT
DDS (ROS 2)	0.1–5 ms	Soft/Hard*
PROFINET IRT	<1 ms	Hard RT

*Depende de la configuración de QoS (Casini et al., 2025)

Frente a la interconexión por red, la literatura técnica señala los mecanismos IPC (*Inter-Process Communication*) basados en memoria compartida como la solución arquitectónica óptima, al eliminar los tiempos de serialización y copiado por *sockets* (Maruyama et al., 2016). Este paradigma exige que ROS 2 y el *runtime* del PLC coexistan en el mismo equipo físico y tenga una capa de virtualización directamente ejecutada sobre el hardware, como por ejemplo hipervisores de Tipo 1 o gestores de contenedores con soporte de tiempo real, habilitando el aislamiento de núcleos (*CPU shielding*) mediante las cuales un conjunto de núcleos del procesador se dedican al escaneo lógico del PLC y a la gestión del bus de campo, mientras que el conjunto restante se destinan al ejecución de los nodos asíncronos de ROS 2. Las siguientes secciones analizan cómo seis fabricantes han materializado este paradigma.

3. Arquitecturas ROS 2 por fabricante PLC

Para identificar las arquitecturas que implementan los principales fabricantes de autómatas programables para ejecutar ROS 2, se ha elaborado una búsqueda sistemática de artículos científicos e información disponible por los fabricantes. La investigación se centro en bases de datos de alto impacto como la IEEE Xplore, ScienceDirect y arXiv, complementándose con portales oficiales de desarrolladores in-

Tabla 2: Comparativa de arquitecturas de integración ROS 2/PLC por fabricante.

Fabricante	Arquitectura híbrida	Mecanismo de comunicación	Latencia típica	Determinismo OT	Grado de madurez
Siemens (ROXSIE)	Hipervisor Tipo 1 sobre IPC SIMATIC. Partición Linux (ROS 2) + partición S7-1500 Software Controller aisladas.	Memoria compartida inter-particional (canal RIB). Generación automática de código SCL y paquete ROS 2 desde YAML.	<1 ms 1 kHz garantizado	Hard RT (partición aislada)	Moderado-Alto. Producto comercial reciente (2024); documentación pública y developer portal activos.
Beckhoff (TwinCAT/BSD)	Hipervisor sobre FreeBSD 13. VM Linux para ROS 2 con <i>device passthrough</i> de GPU (CUDA/OpenVINO).	ADS Sum-Commands: consolida múltiples variables en 2 transacciones/ciclo (1 lectura + 1 escritura).	<1 - 2 ms	Hard RT (TwinCAT nativo)	Muy alto. Ecosistema maduro; drivers <code>beckhoff_ads_driver</code> validados en producción.
Phoenix Contact (PLCnext)	Yocto Linux con kernel RT. ROS 2 en contenedor Docker gestionado por Podman. Descargable desde PLCnext Store.	gRPC sobre Global Data Space (GDS). Hilos independientes por tipo de mensaje; soporta 200 Hz configurable.	<10 ms (asíncrono)	Soft RT (aislamiento de carga)	Muy alto. Soporte oficial en GitHub; limitación en ARM32 (compilación desde fuente).
Bosch Rexroth (ctrlX CORE)	Ubuntu Core 22 con kernel PREEMPT-RT. Aplicaciones empaquetadas como <i>Snaps</i> ; SDK oficial para ROS 2.	ctrlX Data Layer en memoria compartida. Nodos C++/Python leen y escriben variables del PLC a 200 Hz.	~5 ms 200 Hz validado	Fuerte/Suave (PREEMPT-RT)	Muy alto. SDK nativo publicado; benchmarks con <code>geometry_msgs</code> documentados.
Schneider E. (EcoStruxure / Intel ECI)	Maquinas virtuales (KVM/ACRN) sobre hardware Intel. Linux PREEMPT-RT o Xenomai Dovetail como guest RT.	IVSHMEM (memoria compartida entre VMs). Apoyado por Intel TCC y TSN para anular interrupciones espurias.	<5 ms (depende HW)	Fuerte (TCC/TSN Intel)	Moderado. Complejidad de hardware elevada; requiere hardware Intel específico.
Rockwell (ControlLogix)	Enfoque perimetral: PLC aislado + PC Linux externo. Sin hipervisor compartido. ROS 2 en maquina separada.	Edge Gateway de red: puente entre DDS (eProsima Fast DDS) y OPC UA o MQTT sobre EtherNet/IP (CIP).	10–50 ms (depende de red)	Depende del protocolo de red	Alto en SCADA/MES; bajo para control cinemático conjunto. Sin soporte nativo IPC compartido.

IPC: Industrial PC; IVM: Inter-VM; HRT: Hard Real-Time; SRT: Soft Real-Time.

dustriales (aerOS Consortium, 2023; ROS-Industrial Consortium, 2024; Siemens AG, 2024b). Las palabras clave utilizadas incluyeron combinaciones de: “ROS 2”, “PLC”, “Cyber-Physical Systems (CPS)”, “OPC UA”, “shared memory” y tecnologías comerciales específicas como “SIMATIC ROXSIE”, “TwinCAT” y “ctrlX AUTOMATION”. Por último se filtró priorizado estudios y casos de uso industriales enfocados en la latencia, el uso de hipervisores y la creación de gemelos digitales (Profanter et al., 2019; Puck et al., 2021; Nguyen et al., 2023), los resultados de esta búsqueda se resumen en la Tabla 2, y se describen a continuación.

3.1. Siemens: ROXSIE y el canal RIB

El SIMATIC Connector for ROS (ROXSIE) divide el procesamiento de un IPC industrial —como el SIMATIC BX-39A— entre el SIMATIC Industrial OS (Linux, ejecutando ROS 2) y el S7-1500 Software Controller, mediante un hipervisor de Tipo 1 [citesiemens2024roxsie](#). El intercambio de datos elude las capas de red mediante el *Realtime Information Backbone* (RIB), un segmento de memoria compartida

inter-particional capaz de alcanzar 1 kHz de sincronización con latencias inferiores a 1 ms. El flujo de ingeniería es declarativo: a partir de un fichero YAML, ROXSIE genera simultáneamente el paquete C++ de ROS 2 y los bloques SCL para TIA Portal (Zoitl et al., 2024). El ecosistema se completa con ROXSITA, que extrae especificaciones AsyncAPI de las interfaces ROS 2 activas (Siemens AG, 2024a), y con ROS# para validación mediante gemelo digital en Unity (Siemens AG, 2019).

3.2. Beckhoff: hipervisor TwinCAT/BSD y ADS

Beckhoff sustituyó la dependencia de Windows CE por TwinCAT/BSD, un sistema operativo basado en el núcleo FreeBSD 13 que integra de forma nativa el entorno de tiempo real TwinCAT 3 (Beckhoff Automation GmbH, 2024). El TwinCAT/BSD Hypervisor ejecuta máquinas virtuales Linux para alojar ROS 2 con aislamiento absoluto, ofreciendo *device passthrough* que asigna explícitamente GPU y buses USB a la partición Linux capacidad crítica para delegar el proce-

samiento de visión artificial (CUDA, OpenVINO) sin impactar la sincronización de microsegundos de la red EtherCAT (Beckhoff Automation GmbH, 2024). El enlace de comunicación se realiza mediante *Automation Device Specification* (ADS): el paquete `beckhoff_ads_driver` emplea *ADS Sum-Commands*, consolidando múltiples variables del PLC en solo dos transacciones por ciclo de control. Un estudio reciente que implementa un *framework* de gemelo digital sobre TwinCAT 3 y ROS 2 mediante NVIDIA Isaac Sim confirma la idoneidad de esta arquitectura para sistemas mecatrónicos de elevación y manipulación de carga (Nguyen et al., 2023).

3.3. Phoenix Contact: PLCnext y el puente gRPC

El ecosistema PLCnext opera sobre un núcleo Yocto Linux con soporte de tiempo real, habilitando la ejecución nativa de C/C++, C# y Python concurrente con la lógica de control clásica. El PLCnext-ROS-bridge se despliega como nodo ROS 2 dentro de un contenedor Docker, conectando mediante *remote procedure calls* (gRPC) con el *Global Data Space* (GDS) del controlador. A partir de un fichero de descripción de interfaz (IDF) en YAML, el sistema sintetiza automáticamente los tópicos de ROS 2, alcanzando latencias deterministas de pocos milisegundos a frecuencias configurables de hasta 200 Hz. Una limitación documentada es la falta de soporte oficial de binarios ROS 2 para arquitecturas ARM32, que requiere compilación desde código fuente.

3.4. Bosch Rexroth: ctrlX CORE y el Data Layer

ctrlX CORE se basa en Ubuntu Core 22 con aplicaciones empaquetadas como *Snaps*, fusionando la gestión de software de servidores en la nube con el *edge computing* industrial. El elemento unificador es el ctrlX Data Layer, un *broker* de mensajes en tiempo real que mapea todas las variables del sistema. Bosch Rexroth ofrece un SDK oficial para ROS 2 que permite crear nodos en C++ o Python que interactúan con el PLC mediante memoria compartida (Bosch Rexroth AG, 2025). Validaciones publicadas por el propio fabricante demuestran que nodos publicadores de ROS 2 (`geometry_msgs/msg/Vector3`) transmiten datos a través del Data Layer hacia la memoria global del PLC, procesándolos en un ciclo de escaneo de 5 ms y devolviéndolos a la red DDS con una frecuencia estable de 200 Hz.

3.5. Schneider Electric y Rockwell

Schneider Electric colabora con Intel mediante la arquitectura *Edge Controls for Industrial* (ECI), que facilita hipervisores KVM/ACRN y extensiones de tiempo real (PRE-EMPT_RT, Xenomai Dovetail). La partición EcoStruxure intercambia datos con la partición ROS 2 a través de memoria compartida entre máquinas virtuales (IVSHMEM), apoyada en tecnologías Intel TSN y CAT para anular interrupciones espurias.

Rockwell Automation, en cambio, mantiene el núcleo del PLC aislado de sistemas operativos abiertos, optando por un enfoque perimetral de *Edge Gateway*: los robots ROS 2 se ejecutan en un PC externo y se comunican con el ControlLogix mediante puentes entre DDS y OPC UA o MQTT sobre EtherNet/IP. En implementaciones industriales documentadas, la latencia de un enclavamiento de seguridad disparado desde un

nodo de percepción ROS 2 hasta el PLC vía OPC UA alcanza 10 ms, siendo aceptable para navegación móvil, pero insuficiente para bucles de control cinemático cerrado.

4. Perspectivas en integración PLCs y ROS2

Después de revisar las arquitecturas que los principales fabricantes de PLCs proponen para integrar ROS 2 dentro de un autómata programable, se observan dos grandes tendencias en el mercado: (1) la consolidación determinista mediante hipervisores y memoria compartida, orientada a garantizar el tiempo real estricto en lazos de control cerrados; y (2) la virtualización flexible basada en contenedores (Apps/Snaps), diseñada para priorizar el despliegue rápido y la escalabilidad de algoritmos de inteligencia artificial en el entorno *Edge*. Por un lado, Beckhoff y Siemens proporcionan las arquitecturas más maduras para el control estricto, empleando hipervisores y comunicación colocalizada (cero latencia de red). Dentro de este grupo, ROXSIE destaca por su capacidad de generar código C++ y SCL certificado por Siemens, y TwinCAT/BSD por la robustez de ADS y el uso de *device pass-through* para aceleración de hardware.

Por otro lado, enfoques como ctrlX CORE y PLCnext priorizan la accesibilidad y el despliegue rápido. Al integrar Snaps y contenedores Docker, facilitan enormemente la implementación de algoritmos de IA, asumiendo a cambio un leve aumento en la latencia de ejecución. Finalmente, la topología perimetral de Rockwell demuestra ser excelente para la supervisión SCADA/MES, pero insuficiente para un control determinista de alta frecuencia. Los seis enfoques quedan resumidos en la anterior Tabla 2.

A nivel de comunicaciones el ecosistema ROS 2 gestiona la interacción de nodos mediante tres tipos de mensajes *tópicos* (modelo publicador/subscriptor asíncrono), *servicios* (modelo petición/respuesta síncrono) y *acciones* (ejecución de tareas prolongadas con retroalimentación continua). Sin embargo al integrarse con la arquitectura clásica de un PLC, se observa una tendencia hacia el uso de tópicos; el bus RIB de Siemens, por ejemplo soporta de forma nativa y casi exclusiva este mecanismo. Esto se debe a que el flujo continuo y asíncrono de los tópicos se mapea de manera transparente con la naturaleza cíclica de ejecución del autómata. Por el contrario, los servicios y acciones presentan limitaciones para el control de bajo nivel. Su estructura síncrona o bloqueante obligaría a que el PLC, dentro de su ciclo de escaneo, esperara la respuesta de un nodo o la finalización de una tarea prolongada, lo que excedería el tiempo de ciclo permitido (*watchdog timeout*) y en consecuencia, una parada de emergencia del sistema.

Como síntesis de estas evaluaciones, la Figura 1 propone una arquitectura de referencia óptima. Esta se estructura en: una partición IT (ROS 2 en Linux) y una OT (PLC) comunicadas por memoria compartida, un entorno aislado para la seguridad funcional, y una capa Edge/Cloud que asume la orquestación global y el soporte del gemelo digital.

4.1. Caso de aplicación: Laboratorios Inteligentes

Una aplicación que exige la integración simultánea de múltiples sistemas automatizados y robóticos heterogéneos es el desarrollo de Laboratorios Inteligentes o *Self-Driving Labs*

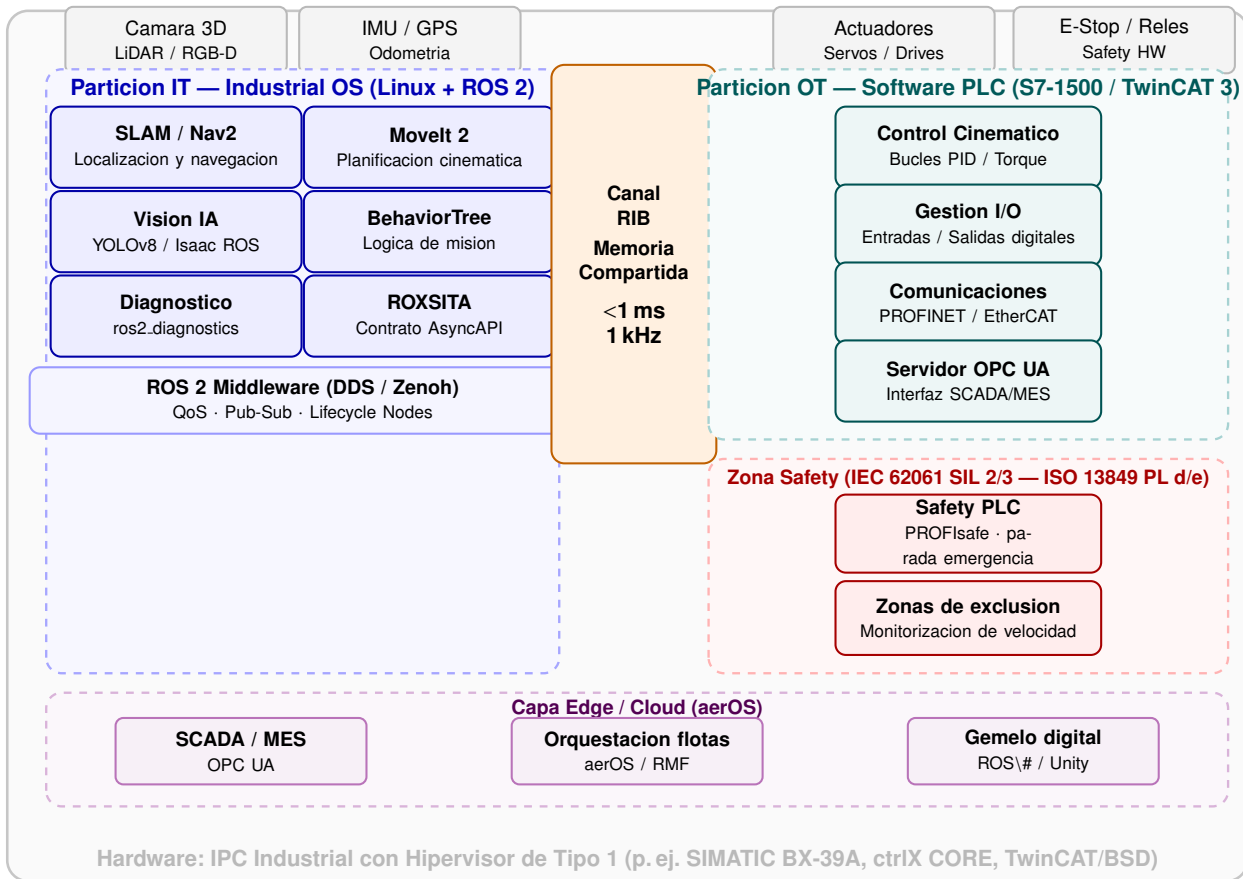


Figura 1: Arquitectura de automatización definida por software para la integración ROS 2/PLC basada en el bus de tiempo real RIB y aislamiento de seguridad funcional. Basado en aerOS Consortium (2023), Siemens AG (2024b) y Gramß et al. (2025).

(SDL) (Tobias and Wahab, 2025). En un SDL orientado al descubrimiento de materiales, los PLCs controlan con precisión determinista reactores químicos y sistemas de dosificación, mientras coexisten brazos colaborativos y robots móviles autónomos que deben manipular y transportar muestras de forma dinámica (Chen and Xu, 2025). ROS 2 actúa como middleware unificador que disuelve las barreras de los ecosistemas cerrados de fabricante, ofreciendo una API común que permite a algoritmos de IA orquestar flujos de trabajo cambiantes sin reprogramar pasarelas para cada hardware nuevo (Figura 2).

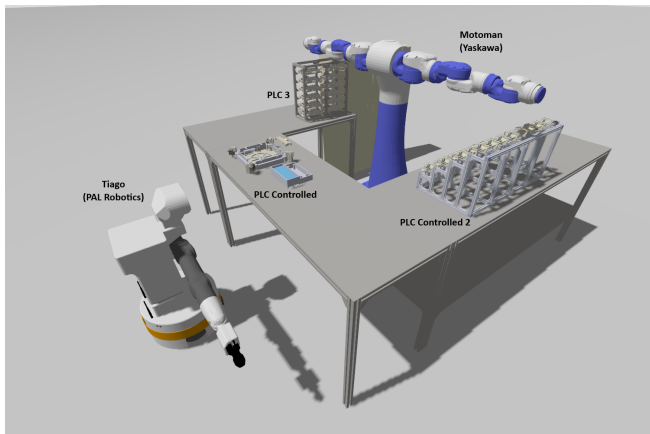


Figura 2: Representación del gemelo estructural del Brain Lab en Gazebo Harmonic, mostrando la integración de los robots TIAGo y Motoman SDA10F (Nazarie et al., 2026).

Esta simbiosis enfrenta barreras en los dominios del tiempo real y la seguridad funcional. Mientras los PLCs operan bajo *hard real-time* con ciclos de milisegundos para garantizar la estabilidad del proceso químico, ROS 2 introduce comunicación probabilística dependiente de la infraestructura de red cuando no se emplea un conector nativo: una fluctuación de latencia inalámbrica puede retrasar la confirmación de acoplamiento esperada por el PLC, deteniendo la línea por *timeout* o provocando derrames. A esto se suma que ROS 2 carece de certificaciones de seguridad funcional, por lo que las lógicas críticas de parada de emergencia ante riesgos químicos deben permanecer confinadas en el PLC de seguridad, obligando a duplicar arquitecturas de control. Resultados recientes en retos de manipulación móvil colaborativa —como el *WARA Robotics Mobile Manipulation Challenge*— confirman que la coordinación robusta de robots heterogéneos en espacios de laboratorio compartidos con humanos exige precisamente este tipo de arquitectura híbrida (WARA Robotics Consortium, 2025).

Los conectores nativos descritos en la Sección 3 ofrecen, frente al uso de protocolos genéricos, una vía directa para resolver esta fricción. Al colocar ROS 2 y el PLC en el mismo hardware, la latencia de acoplamiento deja de depender de la calidad de la red inalámbrica, reduciendo el riesgo de *timeout* en operaciones críticas de dosificación o manipulación de muestras sensibles.

5. Conclusiones

Los protocolos industriales genéricos (MODBUS TCP, OPC UA, MQTT) son estructuralmente inadecuados para el control cinemático cerrado ROS 2–PLC, al introducir latencias y *jitter* de red incompatibles con el tiempo real. Frente a ello, los conectores nativos de fabricante —que colocan ROS 2 y el PLC en el mismo hardware mediante hipervisores y memoria compartida— representan el estado del arte de la integración IT/OT. ROXSIE (Siemens) y TwinCAT/BSD (Beckhoff) ofrecen actualmente la mayor madurez y el menor determinismo de latencia, mientras que ctrlX CORE (Bosch Rexroth) y PLCnext (Phoenix Contact) priorizan la accesibilidad de desarrollo mediante contenedores. Ninguna de estas soluciones está, sin embargo, ampliamente depurada ni estandarizada entre fabricantes: cada conector permanece atado al ecosistema de hardware propio, lo que limita la interoperabilidad en entornos multi-fabricante como los Laboratorios Inteligentes.

El presente trabajo tiene como principal limitación la ausencia de validación experimental propia sobre hardware real, restringiendo la generalización cuantitativa de los resultados, que se basan en literatura técnica y documentación de fabricante. Como trabajo futuro se propone la evaluación experimental de latencias en un banco de pruebas que integre PLCs, robots colaborativos e industriales (ABB GoFa, Yaskawa) y periféricos de inspección visual en línea, característicos de un Laboratorio Inteligente, permitiendo contrastar empíricamente las arquitecturas ROXSIE y TwinCAT/BSD bajo condiciones de carga realistas.

Agradecimientos

La investigación que ha conducido a este resultado ha recibido financiación de los proyectos “iRoboCity2030-CM, Robótica inteligente para ciudades sostenibles” (TEC-2024/TEC-62) y “FotoArt5.0-CM, Laboratorios inteligentes para la ciencia del futuro” (TEC-2024/TEC-308), financiados por los Programas de Actividades I+D en Tecnologías de la Comunidad de Madrid.

Referencias

- aerOS Consortium, 2023. Autonomous, scalable, trustworthy, intelligent European meta Operating System for the IoT edge-cloud continuum. Tech. Rep. HORIZON-CL4-2021-DIGITAL-EMERGING-01-20, ID 101069732, CORDIS EU.
URL: <https://cordis.europa.eu/project/id/101069732>
- Beckhoff Automation GmbH, 2024. TwinCAT/BSD hypervisor: simultaneous execution of virtual machines and real-time applications. Documentación técnica.
URL: <https://www.beckhoff.com/en-en/products/automation/twincat-bsd-hypervisor/>
- Bosch Rexroth AG, 2025. ctrlX AUTOMATION software development kit for ROS 2. GitHub Repository.
URL: <https://github.com/boschrexroth/ctrlx-automation-sdk-ros2>
- Casini, D., Chen, J.-J., Li, J., Teper, H., 2025. A survey of real-time support, analysis, and advancements in ROS 2. arXiv preprint 2511.
- Chen, J., Xu, Q., 2025. Artificial intelligence-driven autonomous laboratory for accelerating chemical discovery. Chemical Synthesis 5 (4).
URL: <https://www.oaepublish.com/articles/cs.2025.66>
DOI: 10.20517/cs.2025.66
- Daubaris, P., Tuovinen, A.-P., Linkola, S., Vicente-Chicote, C., Mikkonen, T., Mäkitalo, N., 2024. ROS2 as an enabler for a business ecosystem: industrial collaboration case study. In: 8th IEEE International Conference on Robotic Computing (IRC). pp. 72–79.
DOI: 10.1109/IRC61757.2024.10680751
- Gramß, F., Kreutz, M., Bazan, R., Braunroth, C., Sancho, A., 2025. Orchestration of heterogeneous experimental machines via ROS 2 for automated integration of industrial automation equipment. Science and Technology of Advanced Materials: Methods.
DOI: 10.1080/27660400.2025.2611510
- Maruyama, Y., Kato, S., Azumi, T., 2016. Exploring the performance of ROS 2. In: Proceedings of the 13th International Conference on Embedded Software (EMSOFT).
DOI: 10.1145/2968478.2968502
- Nazarie, H. P., Oña, E. D., Rosas-Cervantes, V., de la Peña O’Shea, V., Copaci, D., 2026. Un marco metodológico para desarrollar gemelos digitales en laboratorios autónomos. Simposios del Comité Español de Automática (CEA) 2 (2).
- Nguyen, V. H., Kyrki, T., Mattila, J., 2023. A ROS 2 and TwinCAT based digital twin framework for mechatronics systems. Robotics and Computer-Integrated Manufacturing.
DOI: 10.13140/RG.2.2.10248.97287
- Profanter, S., Tekat, A., Dorofeev, K., Rickert, M., Knoll, A., 2019. OPC UA versus ROS, DDS, and MQTT: performance evaluation of Industry 4.0 protocols. In: IEEE International Conference on Industrial Technology (ICIT).
DOI: 10.1109/ICIT.2019.8755050
- Puck, L., Keller, P., Schnell, T., Plasberg, C., Tanev, A., Heppner, G., Roennau, A., Dillmann, R., 2021. Performance evaluation of real-time ROS 2 robotic control in a time-synchronized distributed network. In: IEEE 17th International Conference on Automation Science and Engineering (CASE). IEEE, pp. 1–8.
- ROS-Industrial Consortium, 2024. ROS-Industrial: open-source software for industrial automation. Web.
URL: <https://rosindustrial.org>
- Siemens AG, 2019. ROS# (ros-sharp): ROS–Unity integration library. GitHub Repository.
URL: <https://github.com/siemens/ros-sharp>
- Siemens AG, 2024a. ROXSITA: ROS AsyncAPI specification extractor. GitHub Repository.
URL: <https://github.com/siemens/roxsita>
- Siemens AG, 2024b. SIMATIC Connector for ROS — ROXSIE: overview and documentation. Developer Portal.
URL: <https://developer.siemens.com/roxsie/overview.html>
- Thramboulidis, K., 2015. A cyber-physical system-based approach for industrial automation systems. Computers in Industry 72, 92–102.
DOI: 10.1016/j.compind.2015.04.006
- Tobias, A. V., Wahab, A., 2025. Autonomous ‘self-driving’ laboratories: a review of technology and policy implications. Royal Society Open Science 12 (7), 250646.
DOI: 10.1098/rsos.250646
- WARA Robotics Consortium, 2025. The first WARA robotics mobile manipulation challenge: lessons learned. In: ABB Corporate Research Workshop. Västerås, Suecia, arXiv:2505.06919.
- Zoiti, A., Rooker, M., Sünder, C., Strasser, T., 2024. Integrating PLCs and robots into the ROS 2 ecosystem. In: IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).
DOI: 10.1109/ETFA61755.2024.10710873