

INFORME DE RESULTADOS

PROGRAMA ASTURIAS 2018-2020

TÍTULO DEL PROYECTO: Desarrollo de aplicaciones robóticas para la mejora de la productividad y la seguridad en la Industria 4.0
(ROBOTS 4.0)

REFERENCIA Nº: IDI/2018/000063

Entidad Beneficiaria:

Fundación IDONIAL (CIF G33914557)



IDONIAL
CENTRO TECNOLÓGICO



ÍNDICE

1	INTRODUCCIÓN	3
2	OBJETIVOS Y GRADO DE CUMPLIMIENTO.....	3
3	DESCRIPCIÓN DETALLADA DE LOS TRABAJOS REALIZADOS	4
3.1	HITO 1. ANTECEDENTES Y REQUISITOS.....	4
3.1.1	T1.1. Antecedentes y estado del arte	4
3.1.2	T1.2. Definición de requisitos y adecuación de los casos de uso.....	9
3.1.3	T1.3. Diseño de procesos	14
3.2	HITO 2. GEMELO DIGITAL (REALIDAD VIRTUAL PARA EL CONTROL DE ROBOTS)	24
3.2.1	T2.1. Desarrollo de los modelos matemáticos.....	24
3.2.2	T2.2. Modelado virtual de los robots	27
3.2.3	T2.3. Modelado virtual del entorno	30
3.2.4	T2.4. Diseño y desarrollo de la interfaz de usuario.....	35
3.2.5	T2.5. Integración de la aplicación.....	38
3.2.6	T2.6. Pruebas y validación.....	43
3.3	HITO 3. ROBÓTICA COLABORATIVA – INTERACCIÓN SEGURA PERSONA – MÁQUINA	48
3.3.1	T3.1 Diseño de herramientas específicas y ajuste del entorno	48
3.3.2	T3.2 Sistema de percepción para la tarea	52
3.3.3	T3.3 Sistemas de seguridad.....	55
3.3.4	T3.4 Programación del robot	57
3.3.5	T3.5 Interfaz de usuario	67
3.3.6	T3.6. Integración, pruebas y validación.....	73
3.4	HITO 4. INTERACCIÓN MÁQUINA – MÁQUINA	91
3.4.1	T4.1. Diseño de herramientas específicas y ajuste del entorno	91
3.4.2	T4.2. Sistema de sincronización	105
3.4.3	T4.3. Programación de los robots	119
3.4.4	T4.4. Diseño y desarrollo de la interfaz de usuario.....	133
3.4.5	T4.5. Integración, pruebas y validación.....	144
4	CONCLUSIONES	163

1 INTRODUCCIÓN

Actualmente, la estrategia regional, nacional y europea apuesta por la Industria 4.0 como forma de digitalizar la industria europea, garantizando la seguridad de los trabajadores e incrementando la productividad industrial. Dentro de la Industria 4.0, existen múltiples tecnologías que permitirán lograr dicha digitalización, entre las que destacan la robótica, visión artificial, realidad virtual y aumentada o simulación de procesos, entre otras. Para contribuir a ello se ha definido el proyecto ROBOTS4.0 por medio del cual se pretende consolidar y ampliar la línea de actividad del Centro relacionada con robótica para su aplicación en sectores y aplicaciones tradicionales del Centro, pero sobre todo, como herramienta para llegar a nuevas aplicaciones y sectores.

Con el proyecto ROBOTS 4.0 se desarrollarán aplicaciones robóticas para la mejora de la productividad y la seguridad en la Industria 4.0 incorporando diversas técnicas y tecnologías habilitadoras como la visión artificial, realidad aumentada y virtual. Para ello, se crearán tres demostradores tecnológicos en Robótica: Gemelo Digital en Robótica (Demo1), Robótica colaborativa persona-máquina (Demo2) y Robótica colaborativa máquina-máquina (Demo3). Los tres crearán un amplio espectro de posibilidades de cara al impacto de los resultados ya que, además, la robótica es una tecnología transversal a múltiples sectores de aplicación y casos de uso. Por este motivo, los demostradores ayudarán a continuar con líneas de actividad del Centro, además de poder ampliarlo a otras nuevas.

2 OBJETIVOS Y GRADO DE CUMPLIMIENTO

El objetivo general del proyecto ROBOTS 4.0 es el de la consolidación y ampliación de la línea de investigación de Robótica del Centro en sectores y mercados de actividad existentes actualmente en el Centro así como ampliándolo a otros nuevos sectores y mercados.

Los **objetivos técnicos específicos** del proyecto son los siguientes:

- O1. Realización de un análisis exhaustivo del estado del arte en relación a robótica colaborativa.
- O2. Desarrollo, validación y puesta en marcha del Demostrador 1. Gemelo Digital en Robótica.
- O3. Desarrollo, validación y puesta en marcha del Demostrador 2. Robótica colaborativa persona-máquina.
- O4. Desarrollo, validación y puesta en marcha del Demostrador 3. Robótica colaborativa máquina-máquina.

3 DESCRIPCIÓN DETALLADA DE LOS TRABAJOS REALIZADOS

3.1 HITO 1. ANTECEDENTES Y REQUISITOS

3.1.1 T1.1. Antecedentes y estado del arte

En esta tarea se ha realizado una búsqueda y análisis del estado del arte en relación a cada uno de los tres demostradores. A continuación, se realiza un resumen de las principales búsquedas de cada uno de ellos:

Demostrador 1. Gemelo digital: se ha realizado un estudio del estado del arte sobre este concepto, las posibilidades y obstáculos para alcanzarlo, su aplicación en diversos entornos y aplicaciones industriales, etc. Con todo ello, en la bibliografía se detallan las principales conclusiones acerca del uso del gemelo digital como herramienta. Así, la premisa sobre la que reside el gemelo digital es que consiste en dos sistemas: el sistema físico que siempre ha existido y el sistema virtual que contiene toda la información sobre el sistema físico. Puesto que la información es un remplazo de los recursos físicos inútiles, se puede usar muchos menos recursos costosos en crear, producir y operar con sistemas. A través de modelado y simulación en un espacio virtual, se puede entender mejor la forma emergente y los comportamientos del sistema así como disminuir el factor “no lo vi venir”.

Los sistemas virtuales pueden ayudar en gran medida en tratar con las cuatro categorías de comportamientos del sistema. Se puede asegurar que se obtendrá el PD (por sus siglas del inglés “Predicted Desirable”), eliminando el PU (Predicted Undesirable) y reduciendo el UU (Unpredicted Undesirable). Nótese que si se tiene el UD (Unpredicted Desirable), no hará daño aunque demuestra que no se ha entendido bien el funcionamiento del sistema. Los sistemas complejos son susceptibles de tener “accidentes normales”. Mientras que el término “comportamiento emergente” se usa, se puede afirmar que es un comportamiento que existía potencialmente al comienzo del proyecto, aunque una de las principales causas es el fallo del sistema al darle sentido. Por medio de la simulación del Gemelo Digital, permite reducir los UUs causados por interacciones humanas inesperadas. En el artículo no tratan con sistemas evolutivos: sistemas que tienen aleatoriedades construidas en ellos o cambian las reglas que gobiernan sus comportamientos de operación.

Los sistemas no estallan progresivamente hasta estar totalmente formados. El progreso se hace a través del ciclo de vida desde la creación, producción, operación y eliminación. Con los sistemas “únicamente físicos”, este desarrollo considera las “ilidades”. Mientras que si hay obstáculos más adelante, especialmente culturales, se han hecho avances notables por medio de los Test Virtuales Grieves. Se deberían esperar nuevas mejoras al respecto ya que la tecnología de computación no parece que vaya a reducir su ratio de progreso.

Finalmente, el gemelo digital y su reflejo en el mundo físico indican que potencialmente se puede usar el sistema virtual incluso cuando el sistema físico esté en operación. Los dos usos potenciales son la captura y uso de la información en uso y del sistema en funcionamiento.

El concepto de gemelo digital tiene la oportunidad de cambiar el modo en que se ve el diseño del sistema, fabricación y operación, reducir los UUs de sistemas complejos y aumentando la ingeniería de los sistemas.

En base a la información encontrada en la bibliografía, se ha visto que el concepto de gemelo digital está en auge y puede solucionar muchas limitaciones actuales a un precio reducido. Sin embargo, su preparación requiere de cierta inversión en tiempo de desarrollo, si bien es cierto que la inversión será menor que la que conlleva una implantación real para el análisis de ésta. También es importante destacar que las aplicaciones realizadas son para casos particulares, y que es importante saber que su aplicación dependerá del propósito. De hecho, dependiendo de la finalidad puede ser que el big data sea de mejor aplicación que el gemelo digital.

En el caso del demostrador 1, relativo al gemelo digital, la bibliografía ha ayudado a perfilar cuestiones a tener en cuenta para el caso de uso del proyecto:

- Considerar que la réplica virtual del entorno complementada con la sensorización de la realidad, crea el gemelo digital.
- El uso del gemelo digital por el operario, desarrolladores o inversores, por ejemplo, ayuda a optimizar los desarrollos futuros, a entrenar a operarios y a prever comportamientos.

Demostrador 2. Robótica Colaborativa Persona – Máquina: en este caso se ha realizado una revisión de la importancia que tiene la robótica para Europa y las necesidades de resolver las limitaciones actuales sobre una colaboración segura entre persona y robot. Además, el estudio del estado del arte se ha completado con una revisión de los proyectos europeos del séptimo Programa Marco (FP7) y Horizonte 2020 (H2020). Con todo ello, se ha concluido lo siguiente acerca de los aspectos a tener en cuenta a la hora de definir este demostrador.

Por la experiencia de la que dispone el equipo de proyecto, ya se conocen diversos proyectos donde los robots no suelen tener cabida pero combinados con los humanos pueden responder a las necesidades de la realidad industrial. Este es el caso del sector aeroespacial donde robots móviles trabajando con humanos pueden ayudar a implementar robots en estos entornos (proyecto VALERI) o bien la colaboración humano – robot para ensamblaje de piezas en el ámbito aeronáutico (proyecto SYMBIO-TIC). Hay otros sectores donde se trabaja con grandes piezas, como el sector naval, y se han desarrollado iniciativas donde robots móviles ayudan a realizar tareas de soldadura de ajuste final en la construcción de buques (proyectos CARLOS). Cabe destacar que en algunos proyectos más antiguos, como CARLOS, MARWIN, SMEROBOTICS o ROBO-PARTNER, se habla de colaboración persona – máquina porque unos ayudan a otros a realizar tareas pero de una manera aislada. En este caso, el enfoque está más orientado a que los robots puedan realizar las tareas implementando habilidades cognitivas humanas con su ayuda o para sustituirlos, pero se ha incluido para ver la evolución del concepto de colaboración desde hace unos años a la actualidad. También hay que destacar que hay otros proyectos relativos al uso de asistentes robóticos que colaboran con humanos para dar apoyo en el almacén. Este es el caso del proyecto SECOND-HANDS, donde es importante analizar también, al igual que en el caso de los entornos industriales, la exigencia de los requisitos normativos de aplicación.

En este ámbito es importante destacar los sistemas de visión y seguridad para garantizar la integridad de los trabajadores durante el proceso. La Organización Internacional de Normalización (ISO) ha desarrollado estándares para robots que operan en modo colaborativo¹. El cumplimiento

¹ https://ifr.org/downloads/papers/IFR_Demystifying_Collaborative_Robots.pdf

de estas normas significa que el robot puede, en principio, ser utilizado de forma segura. Sin embargo, un robot seguro no garantiza una aplicación de colaboración segura. Aunque los robots se detienen al detectar un contacto, si la parte que se manipula puede ser peligrosa cuando está en movimiento, por ejemplo debido a los bordes afilados, se requerirá algún tipo de seguridad para asegurarse de que se cumple con las normas legalmente vinculantes para la salud y la seguridad en el trabajo en su país.

En el caso del demostrador 2, relativo a la colaboración Persona – Robot (o Persona – Máquina), la bibliografía ha ayudado a determinar que el caso de uso que se defina para el proyecto, debe tener en cuenta:

- Considerar el concepto actual de colaboración humano – robot sin barreras físicas.
- Incorporación de los sensores de visión adecuados para garantizar la seguridad de los trabajadores.

Demostrador 3. Colaboración robot – robot: en este caso, se ha realizado un análisis exhaustivo del estado del arte y los proyectos de I+D asociados. Con toda esta información se ha podido realizar un análisis y conclusiones para definir el demostrador. Así, en este ámbito hay cada vez más aplicaciones pero se han observado diversas tendencias, como el uso cada vez más extendido de múltiples robots para el montaje de elementos. La colaboración de dos robots o más puede servir y ayudar a automatizar tareas tradicionalmente manuales. Esto es el caso de ensamblaje de pequeñas piezas con movimientos especiales (proyecto SARAFun) o bien para poder ser viable económicamente en lotes más cortos, respondiendo a las necesidades de las PYMEs (proyecto HORSE). A esto se le puede sumar, que si el sistema es modular y flexible, será más sencillo adaptarse al caso del usuario final y por tanto, realizar una implantación real de esta tecnología en la realidad industrial tanto de PYMEs como de grandes empresas.

Históricamente, el ensamblaje mecánico se ha realizado por medio de líneas donde las piezas se alimentan de una estación a otra, mientras que los trabajadores permanecen en su lugar realizando tareas, simples y repetitivas. Posteriormente, evolucionó de manera natural a la línea de ensamblaje robótica. Las celdas de trabajo de ensamblaje tradicionales consisten en un solo robot que realiza una tarea de inserción repetitiva con reducida incertidumbre de posición por medio de una fijación rígida. Esto provoca que sean sistemas intrínsecamente inflexibles, ya que una célula de trabajo determinada, y cualquier dispositivo asociado, deben ser rediseñados si se produce cualquier cambio significativo en el proceso de ensamblaje. El uso de múltiples robots permite soluciones de ensamblaje más ágiles que pueden adaptarse fácilmente a los cambios en los procesos de fabricación así como completar procedimientos de ensamblaje más complejos. Pero también hay que tener en cuenta que estas soluciones son intrínsecamente más difíciles de programar e integrar inicialmente y requieren capacidades avanzadas de detección y control para habilitar los algoritmos de ensamblaje.

Sistemas de visión artificial en aplicaciones de robótica colaborativa en entornos industriales: La visión artificial desempeña un papel relevante en la Industria 4.0 ya que puede integrarse en diferentes fases del proceso productivo, desplegando herramientas avanzadas y soluciones de alto valor añadido. De hecho, en el ámbito industrial cada vez se incorporan más sistemas de visión para logística, inspección en línea sin contacto, automatización de controles de calidad, trazabilidad e

integración de robots². En los últimos tiempos se ha incrementado el uso de sistemas de visión en aplicaciones relativas a la mejora de la seguridad de los trabajadores en el entorno industrial y al guiado de robots, así como en combinación con técnicas de realidad virtual y realidad aumentada para formación, control remoto o la visualización de información como asistencia. De esta forma, los sistemas de visión artificial no sólo son herramientas de selección y control de calidad, sino también pueden controlar y proporcionar información al entorno de producción, siendo claves para lograr la fabricación eficiente con cero defectos.

En el caso de robótica industrial los sistemas de visión son fundamentales ya que permiten al robot medir, entender e interactuar con lo que le rodea, aportando seguridad y flexibilidad al proceso. En un escenario colaborativo, la percepción, el razonamiento eficiente y la colaboración segura entre humanos y sistemas robóticos son componentes clave para aumentar la eficiencia en la producción. Los robots colaborativos deben operar de manera segura y eficiente en entornos que pueden ser no estructurados y en constante cambio. Las aplicaciones más habituales donde se utiliza visión en este ámbito robótico son^{10 3}: guiado y ajuste de trayectoria (visual servoing), selección y agarre de un objeto en un plano (pick&place) y ubicados aleatoriamente en una caja (bin picking), clasificación, movimiento autónomo en sistemas de guiado automático (Automatic Guided Vehicle, AGV), paletización y despaletización, alineación, medición, inspección, seguridad (detección de personas), etc. Tal es la importancia de la visión en la robótica colaborativa que SICK, uno de los principales fabricantes de sensores para la automatización de fábricas, se alió con Universal Robots en 2017 para desarrollar soluciones fáciles de integrar en aplicaciones de lectura de códigos, protección del operario y reconocimiento/posicionamiento de piezas, para abarcar desde el pick-and-place hasta procesos de control de calidad⁴.

En términos básicos, la visión en robótica implica el uso de una combinación de hardware y software acorde a cada aplicación para adquirir, procesar y analizar/interpretar los datos visuales recogidos de la escena. Posteriormente, y en consecuencia al resultado/decisión obtenido, los robots realizarán una determinada acción. Los sistemas de visión en robótica industrial se basan en general en técnicas estándar de visión por computador en las que el procesamiento se divide en cuatro pasos principales⁵: pre-procesamiento (reducción de ruido, realce de contraste, segmentación...), extracción de características (color, textura, forma, movimiento...), razonamiento/clasificación (forma, plantilla, color...) y reacción.

Actualmente, debido a los avances de los últimos años, se está incorporando además Inteligencia Artificial (AI). Esto es debido a que, aunque la visión artificial ha demostrado ser una tecnología de detección idónea y se utiliza ampliamente para identificar los objetos a manipular, existen sin embargo una serie de retos a abordar⁶: distinción entre fondo y objeto, identificación de objetos en movimiento o parcialmente ocluidos, reconocer formas cambiantes o articulaciones, reconocer

² <https://blog.robotiq.com/robot-vision-vs-computer-vision-whats-the-difference>

³ Gayubo, F., (2016). Visión Artificial en la Industria 4.0, Jornada Robótica Colaborativa, Visión Artificial y Vehículos Autónomos.

⁴ <http://www.aer-automation.com/alianza-de-universal-robots-y-sick-para-innovar-en-robotica-colaborativa>

⁵ Bergamini, L., Sposato, M., Peruzzini, M., Vezzani, R. & Pellicciari, M. (2018). Deep Learning-Based Method for Vision-Guided Robotic Grasping of Unknown Objects. Transdisciplinary Engineering Methods for Social Innovation of Industry 4.0, (7) pp. 281-290.

⁶ <https://www.machinedesign.com/motion-control/5-challenges-developing-sharp-robotic-vision>

la posición y orientación de los objetos, entre otros. En el estado de la técnica, las aplicaciones de agarre guiadas por visión se basan fundamentalmente en técnicas de Template Matching, donde un área se escanea en busca de objetos con características que coincidan con los modelos almacenados (imágenes adquiridas previamente, y asignado un agarre óptimo). Esto conduce a una fuerte reducción de la productividad del robot y la flexibilidad operativa general. En los últimos años, se están aplicando la AI y el aprendizaje automático (Machine Learning, ML) en diferentes ámbitos de robótica colaborativa⁷ entre los que destacan el desarrollo de algoritmos que permiten calcular de manera autónoma estrategias de agarre de objetos que no ha visto anteriormente. En este sentido, el uso de AI podría ser útil para abordar los retos anteriormente citados en relación a objetos no homogéneos en forma y dispuestos aleatoriamente⁸.

En la fase de adquisición de información, las tecnologías de visión artificial en robótica se dividen habitualmente en dos categorías: 2D y 3D. También se podrían considerar tecnologías para detección 1D, donde se suele utilizar un telémetro láser, cámara lineal o sensor de distancia óptico. Los sistemas 2D tienen ciertas limitaciones en cuanto a su sensibilidad a cambios de luz, alteraciones de contraste, etc. La tecnología 3D suele implicar un incremento de complejidad y requisitos computacionales en comparación con los sistemas 2D, pero presenta mayor robustez y repetibilidad ante cambios lumínicos, pudiendo inspeccionar partes con bajo contraste. Por tanto, en función de la aplicación final se utiliza la tecnología 2D, 3D o la combinación de ambas como es el caso de la solución comercial Nachi Robotics NX-AV Vision system⁹. De hecho, dependiendo del objetivo final de la aplicación y el tipo de robot, se deben tener en cuenta los siguientes factores para escoger la técnica de visión más adecuada¹⁰: precisión, distancia de trabajo, peso, seguridad, tiempo de procesamiento, condiciones del ambiente, integración con otros sistemas y presupuesto. En paralelo, se ha realizado también un estudio de las principales tecnologías 2D, 3D y de sistemas de seguridad que pueden ser de aplicación en casos como el que aplica en este proyecto.

Simulación de procesos en entornos industriales: es importante destacar que esta tecnología en sí misma ya forma parte del concepto de Gemelo Digital y por tanto, se ha tenido en cuenta dentro del estudio del estado del arte de este demostrador. No obstante, debido a su importancia dentro del proyecto, se ha decidido ampliar el estado del arte en relación al uso de la simulación de procesos a nivel industrial.

Según la información encontrada en la bibliografía¹¹, las principales ventajas y limitaciones del uso de la simulación de procesos están asociadas a lo detallado a continuación. Por un lado, la simulación de procesos permite conocer limitaciones del proceso sin necesidad de realizar la inversión en equipamiento previamente y una vez conocidas, poder optimizar el proceso.

Entre las principales **ventajas** que permite la simulación de eventos discretos, destaca:

⁷ <https://www.robotics.org/blog-article.cfm/How-Machine-Learning-is-Used-in-Collaborative-and-Industrial-Robots/94>

⁸ <https://mantec.org/vision-robotics-happy-together/>

⁹ <http://www.nachirobotics.eu/nv-ax-vision/>

¹⁰ Pérez, L., Rodríguez, Í., Rodríguez, N., Usamentiaga, R., & García, D. (2016). Robot guidance using machine vision techniques in industrial environments: A comparative review. *Sensors*, 16(3), 335.

¹¹ Estado del arte de Modelización, Simulación y Virtualización de Procesos – Oportunidades Industria 4.0 en Galicia

- Modelización de situaciones de alto nivel de complejidad con funciones moderadamente sencillas, pudiendo representar la realidad con el nivel de detalle deseado.
- Posibilidad de extraer todo tipo de estadísticas e indicadores relevantes de la operación modelada.

Además de esto, existen **limitaciones** asociadas como las encontradas en la bibliografía¹²:

- En ocasiones, problemas sencillos requieren procesos de simulación costosos y en estos casos, la simulación puede ser costosa.
- Un buen estudio de simulación puede requerir mucho tiempo y no todas las empresas disponen de ese tiempo o de los recursos necesarios para contratar a un desarrollador.
- Si bien las simulaciones y modelizaciones cada vez se parecen más a la situación real, éstas no son capaces de incluir todos los procesos y no permiten un uso fácil y eficiente de la información.

A pesar de todo ello, sigue siendo una buena alternativa en coste relativamente reducido. En este proyecto, al disponer de los recursos necesarios, tanto humanos como materiales, la simulación de procesos ha permitido optimizar los procesos de los demostradores definidos.

Igualmente, se ha realizado un estudio de las principales herramientas de simulación existentes en el mercado, proyectos asociados y casos de aplicación concretos que pudieran ser interesantes para su aplicación en el proyecto.

3.1.2 T1.2. Definición de requisitos y adecuación de los casos de uso

En este apartado, se ha realizado la definición de requisitos y adecuación de los casos de uso para cada uno de los demostradores definidos en el proyecto. A continuación, se resumen las características de cada uno de ellos:

3.1.2.1 Demostrador 1. Gemelo digital

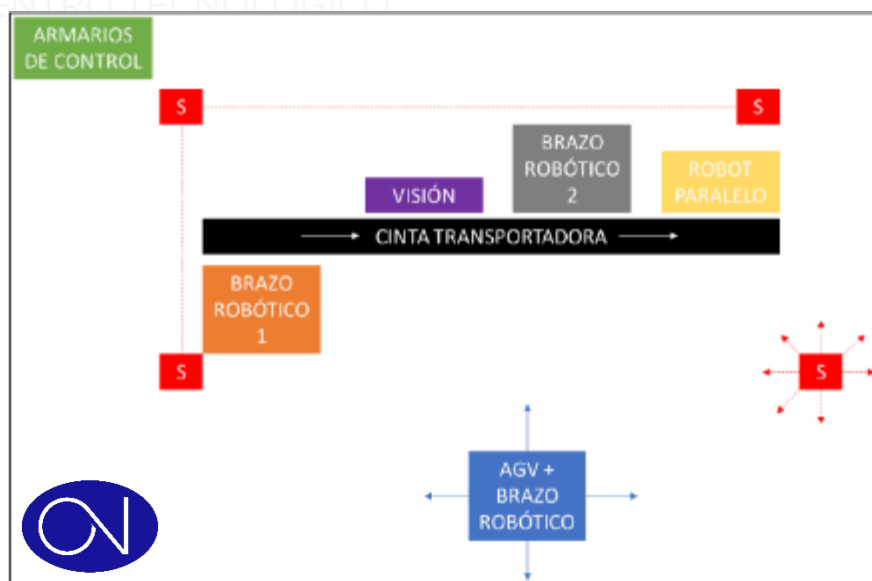
El demostrador 1 se basaba en crear una réplica virtual del Laboratorio de Robótica Colaborativa según las especificaciones iniciales. Este gemelo digital se ha determinado que esté compuesto por los siguientes elementos, esquematizados a su vez en la 1.

- **Brazo robótico 1:** robot para entornos industriales, con flexibilidad para realizar diversas opciones o tareas, orientado a aplicaciones colaborativas (limitaciones de fuerza y velocidad), estructura ligera para evitar accidentes y permitir el trabajo colaborativo seguro con personas sin barreras físicas, sensores integrados para la detección de contacto (capacidad sensitiva), disponibilidad de herramientas y facilidad de integración, capacidad de movimiento (grados de libertad) en espacios reducidos y precisión, capacidad de carga de al menos 12kg e interfaz (consola) de programación y manejo sencilla.
- **Brazo robótico 2:** robot para entornos industriales, con flexibilidad para realizar diversas opciones o tareas y con dos brazos, orientado a aplicaciones colaborativas (limitaciones de fuerza y velocidad), estructura ligera para evitar accidentes y permitir el trabajo colaborativo seguro con personas sin barreras físicas, sensores integrados para la detección de contacto (capacidad sensitiva), disponibilidad de herramientas y facilidad de

¹² Dunna, E. G., Reyes, H. G., & Barrón, L. E. C. (2006). Simulación y análisis de sistemas con ProModel. Pearson Educación.

integración, capacidad de movimiento (grados de libertad) en espacios reducidos y precisión, capacidad de carga de 0.5kg en cada brazo e interfaz (consola) de programación y manejo sencilla.

- **Brazo robótico sobre AGV:** robot y AGV para entornos industriales, flexibilidad para realizar diferentes operaciones/tareas, orientado a aplicaciones colaborativas (reducción de fuerza y velocidad), estructura ligera para evitar accidentes y permitir el trabajo colaborativo seguro con personas sin barreras físicas, sensores integrados para la detección de contacto (capacidad sensitiva) en el brazo robótico y el AGV tendrá seguridad integrada en el sistema de navegación para evitar colisiones con personas y objetos. Disponibilidad de herramientas y facilidad de integración; capacidad de movimiento (grados de libertad) en espacios reducidos y precisión; capacidad de carga del robot: 5 kg y capacidad de carga del AGV: 100 kg, autonomía del AGV de al menos 8 horas, para la instalación del AGV no debería ser necesario realizar ningún tipo de obra civil.
- **Robot paralelo:** robot para entornos industriales, alta velocidad y precisión, manipulación de piezas pequeñas, integración de sistema de visión para la localización de piezas.
- **Sistemas externos de seguridad:** sistemas ópticos certificados para entornos industriales en lugar de físicas, facilidad de programación e integración con los sistemas de control y se requerirá al menos un escáner de 270º y una barrera/cortina con espejo intermedio.
- **Sistemas externos de visión:** integración con la propia herramienta (pinza) y con el robot para la localización/identificación; en caso de tecnologías láser, la categoría deberá ser segura para las personas sin necesidad de usar gafas de protección; facilidad de programación e integración con los sistemas de control y disponibilidad de SDK.
- **Cinta transportadora:** sistema para el transporte de piezas sobre las que actuarán los robots y sobre la que se colocará el sistema de visión; facilidad de programación e integración con los sistemas de control; debe incluir encoder, sensor de presencia, etc. para sincronizar con el sistema de visión.



1. Esquema de distribución y elementos del demostrador 1. Gemelo digital

Para su recreación, y la implantación posterior del laboratorio, se ha seleccionado una zona del edificio de IDONIAL Gijón en la zona del Laboratorio de Ligeros.

3.1.2.2 Demostrador 2. Robótica colaborativa Persona – Máquina y Demostrador 3. Colaboración Robot - Robot

Como ambos demostradores estarán instalados en el “mundo real” como parte del Laboratorio de Robótica Colaborativa, los elementos hardware que los componen serán comunes. Los elementos serán los detallados en el punto anterior, si bien en este caso, la explicación se centrará en detallar el modelo particular seleccionado finalmente tras el análisis del demostrador 1. *Nótese que dichas especificaciones se han ajustado posteriormente tal y como se detalla en el caso del Demostrador 2 (HITO 3) y Demostrador 3 (HITO 4).*

3.1.2.2.1 Elementos de los demostradores 2 y 3

Los elementos robóticos que formarán estos demostradores son: Modelo de KUKA LBR iiwa – Brazo robótico 1, Modelo de ABB Yumi – Brazo robótico 2, Universal Robots UR5 sobre AGV Summit XL Steel de Robotnik – Brazo robótico sobre AGV y modelo Fanuc M-1 iA 0.5SL – robot paralelo, además de sensores de seguridad, visión o cinta transportadora, mencionados anteriormente. No obstante, los demostradores 2 y 3 se implantarán dentro del gemelo virtual (demostrador 1) para poder disponer de una copia virtual de los mismos y poder simular los procesos en este entorno. El demostrador 2 está orientado a una colaboración persona – máquina y el esquema general es el mostrado en la siguiente figura. Incluye los siguientes elementos que son los que forman dicho caso de uso: Robot UR5, CB3 y AGV Summit XL Steel. Por otro lado, el demostrador 3 orientado a la colaboración máquina – máquina, incluye los elementos mostrados en el esquema inferior.

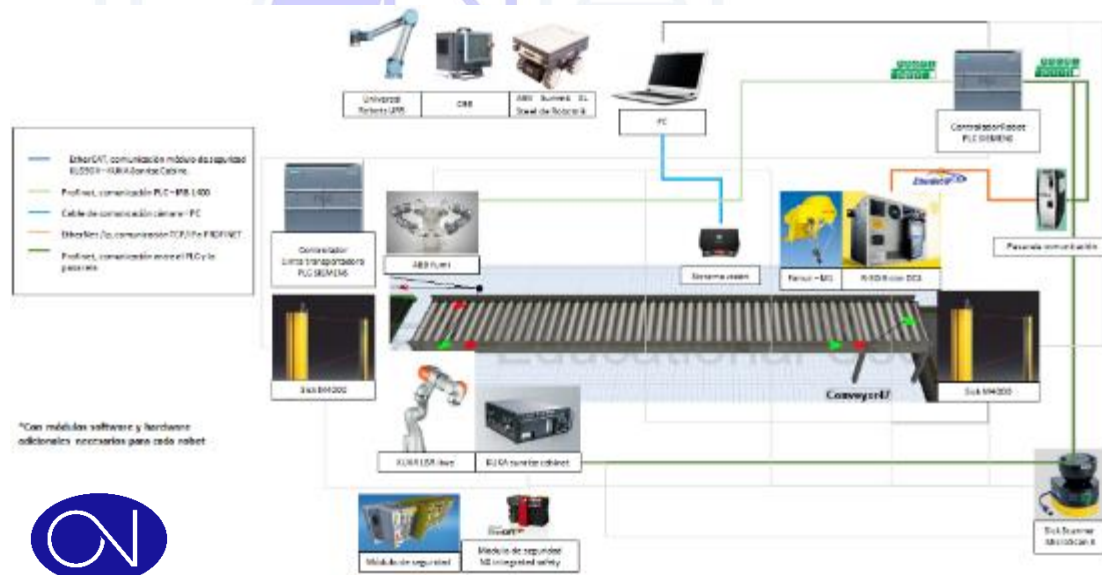


Figura 2: Diseño eléctrico general del demostrador 2 y demostrador 3 (preliminar)

Por otro lado, se han hecho un **diseño eléctrico preliminar** (Figura 2) que podría optimizarse a la hora de poner en marcha el laboratorio físico, teniendo en cuenta otros elementos del conjunto que deben tenerse en cuenta.

3.1.2.2.2 Acciones a realizar en los demostradores 2 y 3

En cuanto a las **acciones** a realizar en los demostradores, a continuación se detallan éstas para cada uno de los demostradores:

- **Acciones del Demostrador 2. Robótica Colaborativa Persona – Máquina:** el demostrador se basa en la colaboración persona – máquina en la que un operador preparará manualmente unas piezas para empaquetarlas en distintos lotes. Se han definido diversas tapas de formas y colores en las que se encuentran diversos agujeros donde insertar una tapa.

El conjunto de piezas objetivo para implementar son las detalladas a continuación:

Tabla 1. Tipos de piezas a usar en el Demostrador 2 (Fuente: elaboración propia)

Forma	Tamaño	Color	Nº de agujeros
Circular	Radio conocido	Rojo o verde	1
Triángulo equilátero	Lado conocido	Rojo o verde	3
Cuadrada	Lado conocido	Rojo o verde	2

El proceso aquí descrito simula una planta genérica de ensamblaje en la que robot y humano pueden trabajar de forma colaborativa compartiendo un mismo entorno de trabajo sin peligro alguno para la persona. Existirá un entorno virtual o aumentado, mediante el cual ambos podrán compartir información. Para el gemelo digital la información se mostrará en un panel de control. Las piezas previstas para dicho demostrador se esquematizan en la Figura 4, y se pueden ver éstas con tapa sobre los agujeros o sin ella. Las piezas de un lote se colocarán en bandejas para su almacenamiento. De este modo, se comenzaron los trabajos con un primer análisis del **proceso general** de funcionamiento del demostrador 2 mediante el que se identificaron los principales inputs y outputs del mismo, así como las diferentes etapas en las que se debiera desglosar el trabajo para su modelado y simulación posteriores. En la Figura 3 se muestran cada una de las etapas en las que se desglosó el proceso.

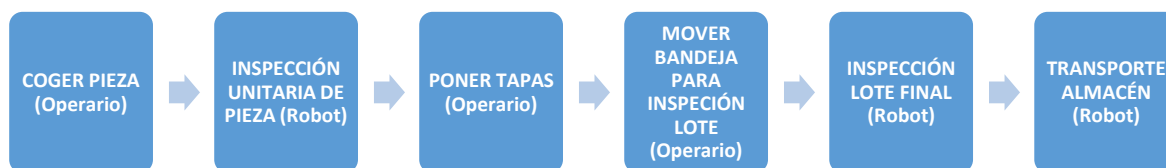


Figura 3. Principales etapas proceso demostrador 2

En base a lo anterior, se determinó la necesidad de definir 6 posibles **clases de piezas** a procesar, así como otras **clases de piezas auxiliares** que sirvieran de apoyo en el modelado, como las tapas o las bandejas en las que se procesan y transportan los lotes de dichas piezas. A continuación se muestran las posibles piezas a utilizar en el demostrador.

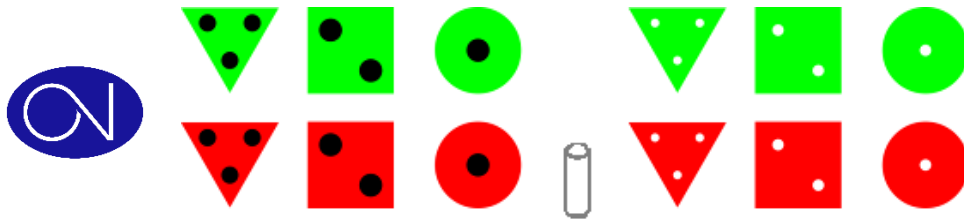


Figura 4. Posibles piezas para Demostrador 2 (Fuente: elaboración propia)

- Acciones del Demostrador 3. Robótica Colaborativa Máquina – Máquina: en este caso, el demostrador se plantea como una colaboración efectiva entre robots (máquina – máquina). Partiendo del **proceso general** de funcionamiento del demostrador 3 se identificaron los principales inputs y outputs del mismo, así como las diferentes etapas en las que se debiera desglosar el trabajo al menos para su modelado y simulación posteriores. En la siguiente *Figura 5* se muestran cada una de las etapas en las que se desglosó el proceso.

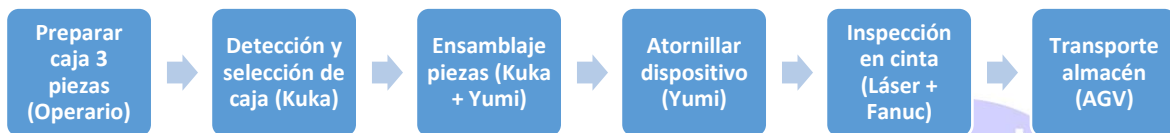
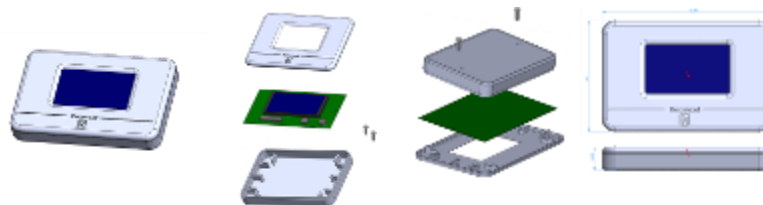


Figura 5 Principales etapas proceso demostrador 3

El modelo requirió la necesidad de definir 4 **clases de piezas** posibles a procesar (3 clases correspondientes a los componentes principales del dispositivo más 1 clase para la pieza final), así como otras **clases de piezas auxiliares** que sirvieran de apoyo en el modelado, como la caja inicial en la que el operario prepara los 3 componentes del dispositivo y el pack final que se emplee en el transporte de dispositivos desde el final de la línea hasta la zona de almacenaje.

Para ello, el demostrador se centra en realizar tareas de montaje de piezas objetivo. Teniendo en cuenta las limitaciones de peso de ciertos manipuladores, se ha seleccionado una pieza que en su conjunto no supere esos 500g. La pieza seleccionada es la mostrada en la Figura 6. Se dispondrá de una cinta transportadora, un sistema de inspección, de los equipos de visión y herramientas necesarios para cada robot y un contenedor para desechar los dispositivos mal ensamblados o con defecto.



Dispositivo #1 - Elementos

Dispositivo #1 - Montaje

Figura 6. Dispositivos tipo para el demostrador (Fuente: elaboración propia)

3.1.3 T1.3. Diseño de procesos

Se ha definido una zona única en PRODINTEC (IDONIAL Gijón, actualmente) donde instalar el Laboratorio de Robótica Colaborativa, donde se implementarán los casos de uso definidos anteriormente. Con esta información, se realiza un diseño del proceso actual manual para ensamblaje de piezas en los casos de uso estudiados y se contrasta con el análisis del proceso realizado de manera automatizada por robots, lo que permite conocer la eficiencia del proceso automatizado y optimizar el mismo, en caso de que fuera necesario, de cara a futuras versiones. Para ello se ha utilizado el software de simulación Delmia Quest. A continuación se resume el diseño de procesos por medio de la simulación en el software Delmia Quest para los casos de uso del demostrador 2 y 3. El demostrador 1 del gemelo digital está constituido por un conjunto de elementos que en realidad estarán operativos desde los casos de uso de los demostradores 2 y 3, por ello el orden en el que se muestran los resultados de las simulaciones para el diseño de procesos es 2 y 3, y una vez estos han sido optimizados, se presenta el demostrador 1.

3.1.3.1 Diseño y simulación Demostrador 2

En este apartado se recoge el planteamiento, desarrollo y resultados de la simulación del demostrador 2, realizado para diseñar el proceso de este caso de uso de manera óptima.

3.1.3.1.1 Elementos del demostrador

Para el modelado del proceso en el entorno Quest fue necesaria la creación y correspondiente programación de cada uno de los **procesos** o etapas que constituyen el proceso general, así como de las **máquinas** en las que se desarrollan dichos procesos y el **operario y/o robot** que constituyese la fuerza de trabajo en cada caso. También se han recogido los **tiempos de ciclo promedio** establecido para cada una de las operaciones implementadas. Todos los elementos definidos en el apartado anterior fueron configurados en el entorno del software de simulación de eventos discretos Delmia Quest, en el cual se tomó como referencia la **distribución de planta** de PRODINTEC, con el fin de evaluar también posibles condiciones o restricciones asociadas con la ubicación física de los demostradores y/o el espacio de almacenamiento requerido.

3.1.3.1.2 Modelos y simulación

De manera gradual se fueron definiendo diferentes escenarios o modelos de trabajo, suponiendo una distribución de tareas entre 1 operario y 1 robot colaborativo. Cada uno de estos escenarios está caracterizado fundamentalmente por la disposición física de los elementos que lo componen, diferentes lógicas de comportamiento, así como varios parámetros asociados a la gestión de la producción, como el tamaño de lote de entrada o el tamaño de lote de transferencia de una etapa a otra. En la Tabla 2 se resumen los diferentes modelos planteados y simulados durante el diseño del proceso del Demostrador 2. Una vez identificado el modelo con mejores resultados suponiendo un trabajo colaborativo entre 1 operario y 1 robot, se simuló el modelo 6, con idéntica configuración de elementos, pero suponiendo que todas las tareas son desarrolladas por un único operario, sin ayuda del robot colaborativo.

Tabla 2. Modelos de simulación– Demostrador 2

MODELO	DESCRIPCIÓN	MODELO	DESCRIPCIÓN
1	▪ 1 Operario + 1 robot	4	▪ 1 Operario + 1 robot

MODELO	DESCRIPCIÓN	MODELO	DESCRIPCIÓN
	▪ Célula de trabajo en forma de L. ▪ Almacén final zona derecha. ▪ Traslado a almacén en bandejas unitarias		▪ Célula de trabajo en forma de L. ▪ Almacén final en zona inferior. ▪ Traslado a almacén de 4 en 4 bandejas.
2	▪ 1 Operario + 1 robot ▪ Célula de trabajo en forma de L. ▪ Almacén final en zona inferior. ▪ Traslado a almacén de 3 en 3 bandejas.	5	▪ 1 Operario + 1 robot ▪ Célula de trabajo en forma de U. ▪ Almacén final en zona inferior. ▪ Traslado a almacén en bandejas unitarias.
3	▪ 1 Operario + 1 robot ▪ Célula de trabajo en forma de L. ▪ Almacén final en zona inferior. ▪ Traslado a almacén en bandejas unitarias	6	▪ 1 Operario (sin robot colaborativo) – > tiempos de ciclo superiores. ▪ Célula de trabajo en forma de U. ▪ Almacén final en zona inferior. ▪ Traslado a almacén en bandejas unitarias.

Gracias a la simulación de los modelos planteados y también a la aplicación de principios de Lean Manufacturing como la identificación del valor y la reducción de despilfarros (transportes, esperas, Sobreproceso, etc.) se pudieron evaluar diferentes alternativas de configuración e ir tomando de manera objetiva decisiones relativas a la configuración más apropiada en términos de eficiencia y productividad. A continuación se enumeran los **principales puntos evaluados** para el diseño del proceso final del Demostrador 2.

- **Optimización:** se ha trabajado en lograr un proceso de ensamblaje colaborativo humano-robot eficiente. a) En el proceso inicial el operario cogía una pieza de la forma deseada, la ensamblaba con las correspondientes tapas y posteriormente se inspeccionaba. Esta propuesta no era eficiente debido a que se estaban ensamblando tapas en piezas que podían tener un error de fabricación o que no pertenecía al lote. Esto provocaba pérdida de material en caso de desechar una pieza con tapas o pérdida de tiempo si se tuviese que quitar la tapa. Por tanto, se propone la inspección antes de realizar la inserción de tapas. b) Se muestra al operario qué pieza es la siguiente para poder realizar de forma paralela la inspección de la pieza actual por el robot y la búsqueda de la siguiente por parte del operario. c) Para evitar cuellos de botella, en el nuevo proceso se instala una bandeja de 2 posiciones (buffer) para que el operario pueda ir colocando piezas mientras el robot realiza la inspección y coloca las correctas en la bandeja del lote. d) En el proceso propuesto es el robot el que descarta las piezas para lograr una mayor eficiencia y así se evita que el operario esté esperando a recibir el resultado de la inspección para actuar en consecuencia. En este sentido, en el proceso optimizado (nuevo proceso) se añade un contenedor de piezas cogidas erróneamente por el usuario (NOK_E) para que el robot las coloque ahí, sin bloquear el proceso por un fallo de elección. Este fallo se considera que será reducido porque el operario estará viendo en todo momento qué pieza tiene que coger, por lo que se ha realizado un planteamiento de cara a minimizar este error. e) Se considera que haya dos contenedores de piezas descartadas (error dimensional NOK_D y error elección NOK_E) para lograr una mayor eficiencia, y que cuando estén llenos o haya un tiempo muerto se

vacían por un operario, colocando de nuevo en los contenedores las piezas que estaban bien pero se escogieron erróneamente. f) Se añade flexibilidad al proceso: si falta una pieza en el lote, en lugar de volver a solicitarla, se solicita al final para colaborar con el operario de una forma más fluida, sin provocar confusión en el proceso. g) La inspección final se plantea en la propia plataforma del robot porque el operario puede ir preparando la mesa para el siguiente lote mientras el robot inspecciona y si es correcto, ya no se necesita interacción por su parte logrando una mayor eficiencia.

- Reducción de movimientos y transportes: a) La colocación de los contenedores de piezas, tapas y bandejas se ha realizado con el objetivo de reducir los movimientos del trabajador, pudiendo acceder a todo el material necesario desde la misma mesa de ensamblaje. b) Se ha buscado la colocación más adecuada del robot AGV en relación al operario. c) Para reducir los movimientos del trabajador es el AGV el que se desplaza durante la fase de ensamblaje e inspección. d) Es necesario un transporte desde la zona de fabricación a la zona de contenedores o almacenamiento intermedio por lo que se ha estudiado el tiempo que conlleva ese desplazamiento y las distancias totales recorridas. Se comprueba que es un factor determinante la ubicación de los contenedores que afecta en gran medida a la disponibilidad del AGV y por tanto a la capacidad productiva de la línea. e) Es necesario un transporte desde la zona de bandejas de lotes preparadas a la zona de almacenaje de pedidos por lo que se ha analizado la distancia máxima para que el robot no provoque cuellos de botella si se lanza un nuevo pedido.
- Buena movilidad de los trabajadores: a) Se ha buscado una distribución del espacio óptima para garantizar una buena movilidad de los trabajadores y flexibilidad de movimiento de material entre puestos, teniendo en consideración el espacio disponible del laboratorio. b) En simulación se dispone de un panel pero en real se dispondrá de unas gafas donde se irá mostrando las instrucciones y estado de las piezas. De esta forma, se garantiza mayor movilidad por parte del operario.
- Posibilidad de ampliaciones futuras: a) Los tamaños de bandejas podrían adaptarse si se fabrican lotes de mayor cantidad. b) Actualmente el robot sólo se desplaza a la zona de ensamblaje y a la zona de almacenamiento de bandejas OK. Se puede plantear que en el caso de tener nuevas piezas fabricadas, en lugar de desplazarse un operario a reponer los contenedores, sea el AGV el que desplace las cajas de un lado de la instalación a otro. c) En lugar de que un operario recoja la bandeja que transporta el AGV, se podría adaptar la herramienta, la bandeja y el lugar de almacenamiento para que el brazo robótico realice el almacenamiento. d) Para optimizar el proceso, en caso de producirse solicitudes continuas de lotes, el operario podría ir apilando bandejas hechas sobre la plataforma del AGV, inspeccionando la bandeja superior cada vez que añada una.

3.1.3.1.3 Resultados y conclusiones

En la Tabla 3 se muestran los indicadores definidos para evaluar los diferentes modelos propuestos de configuración del Demostrador 2. En cada caso, se simuló un turno de trabajo de 7,5 h y, como puede comprobarse, a igualdad de recursos (operario, robot, puestos de trabajo, etc.) se puede concluir que la configuración que ofrece una mayor capacidad productiva en el Modelo 5 (470 bandejas → 62 bandejas/h). Esto es posible gracias a que en este escenario se han minimizado

fundamentalmente las distancias recorridas por el operario y el robot, lo cual permite a su vez una mayor utilización del equipo robot y también del operario, elección motivada precisamente por una mayor cantidad de piezas/bandejas procesadas en el tiempo simulado.

Tabla 3 Resultados simulación Demostrador 2

	MODELO 1	MODELO 2	MODELO 3	MODELO 4	MODELO 5	MODELO 6
Tiempo simulación (h)	7,5	7,5	7,5	7,5	7,5	7,5
Producción	352	441	450	429	470	264
Utilización operario (%)	43%	54%	55%	53%	48%	99%
Utilización robot (%)	64%	49%	54%	50%	63%	0%
Distancia recorrida operario (m)	4.508	5.663	5.757	5.497	4.424	4.987
Distancia recorrida robot (m)	5.582	1.840	2.570	2.132	3.787	0
Lead Time promedio (minutos/bandeja)	1,28	1,02	1,00	1,05	0,96	1,70

Como puede comprobarse en la tabla anterior, si comparamos los resultados del Modelo 5 con los del Modelo 6, que representa la situación actual (sin robot colaborativo), se produce un incremento de la producción del 78 % y una reducción del Lead Time promedio de bandeja procesada del 44 %.

3.1.3.1.4 Requisitos y caso de uso final tras el diseño del proceso

Tras la simulación realizada para este demostrador, se concluye que para realizar la inspección se utilizará un buffer (bandeja de dos posiciones) que servirá para ir depositando las piezas que el operario recoge para su inserción. Se dispondrá de varios contenedores de piezas, tapas, bandejas, una bandeja de lote y una de buffer, caja de NOK_D (piezas descartadas por error dimensional) y caja de NOK_E (piezas descartadas por error del operario). Teniendo en cuenta esto, el setup esquematizado y actualizado para el demostrador se muestra en la Figura 7.

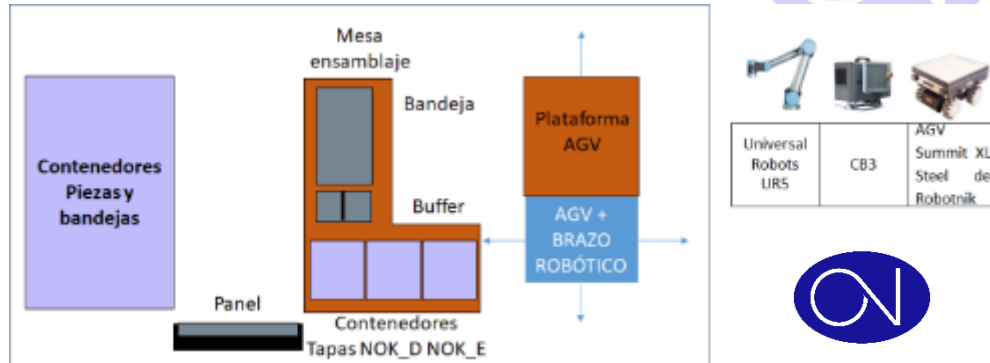


Figura 7. Setup esquematizado del Demostrador 2 (final)

Los pasos a seguir durante el proceso optimizado (tras el análisis de esta tarea 1.3) serían los siguientes, teniendo en consideración que el operario comunicará al robot, en caso de ser necesario, que ha realizado una operación por medio de un comando o botón (a valorar) para actuar en consecuencia:

- 1) **Recibir pedido:** a) El sistema recibe pedidos de lotes de piezas. Estos lotes están compuestos por una cantidad determinada de piezas de cada tipo; b) En función del pedido, el sistema calcula el número de piezas que deben fabricarse de cada tipo, además de llevar un conteo de las piezas que ya han sido introducidas en cada lote; c) Gracias al conocimiento del estado del almacén, el operario de fabricación lanza la fabricación en caso de necesitar más piezas para rellenar los contenedores existentes en la zona de ensamble. En caso contrario, si ya existen piezas suficientes, el operario de montaje recibe el aviso de acudir a la zona de montaje para empezar el pedido y el robot móvil acude a la misma zona;

- 2) **Preparar lote e inspección individual:** d) Una vez que el operario y el robot estén en la mesa de ensamblaje, el sistema indica al operario el lote a realizar y le indica la primera pieza a coger; e) Entonces, el operario coge una pieza de la forma deseada, la pone en la bandeja buffer; f) El robot se posiciona sobre la pieza, y la analiza con una cámara embarcada. Si la pieza es correcta (tipo de pieza correcto, dimensiones adecuadas y color correcto) la recoge y la deposita en la bandeja. En caso de ser incorrecta, la deposita en la caja NOK correspondiente y la añade al final de la lista de piezas a pedir al usuario: Si el defecto es corregible (si la pieza es correcta pero el color no es correcto), el robot coge la pieza, lanza un aviso y la deposita en la caja de NOK_E. Si el defecto no es corregible (si las dimensiones de fabricación no son correctas), el robot la descarta a la caja de piezas defectuosas NOK_D; g) El proceso se repite con todas las piezas del lote, facilitando el sistema al operario la información de la pieza actual y la siguiente. Una vez que ha finalizado el lote, el sistema solicita al operario las piezas que faltan en orden, repitiendo el proceso anterior hasta que el lote sea OK.
- 3) **Ensamblar tapa:** h) Una vez que el lote está preparado e inspeccionado individualmente, el operario ensambla las tapas. Mientras el operario ensambla las tapas, el robot se posiciona en la siguiente posición para la inspección del conjunto sobre su plataforma; i) Tras poner todas las tapas, el operario traslada la bandeja a la plataforma del robot e indica que el lote está listo.
- 4) **Inspección lote:** j) El robot inspecciona la bandeja. Si es OK, la traslada a la zona de bandejas OK. Si es NOK (faltan tapas): avisa al operario qué pieza es incorrecta para que lo corrija. El operario realiza el proceso e indica al robot que es OK. Si hay otra pieza NOK, repite el proceso. Si es la última pieza OK, el robot inspecciona de nuevo la bandeja. Si es OK, se traslada a la zona de bandejas OK.

Teniendo en cuenta esta distribución final definida para el demostrador, la aplicación e interfaz desarrollada han tratado de actualizarse teniendo en cuenta estas conclusiones en la medida de lo posible.

3.1.3.2 Diseño y simulación Demostrador 3

En este apartado se recoge el planteamiento, desarrollo y resultados de la simulación del demostrador 2, realizado para diseñar el proceso de este caso de uso de manera óptima.

3.1.3.2.1 Elementos del modelo de simulación

Para el modelado del proceso en el entorno Quest fue necesaria la creación y correspondiente programación de cada uno de los **procesos** o etapas que constituyen el proceso general, así como de las **máquinas** en las que se desarrollan dichos procesos. Además, se crearon y programaron otros elementos que intervienen en el proceso, entre ellos: **procesos** o etapas en los que se ha desglosado el proceso general, **máquinas** en las que se desarrollan dichos procesos y el **operario y/o robots** que constituyese la fuerza de trabajo en cada caso. En la siguiente tabla se recogen los **tiempos de ciclo promedio** establecidos para cada una de las etapas representadas del proceso, tanto en el caso de modelos en los que intervienen los robots como en el modelo representativo de una situación actual, en la que el trabajo de ensamblaje y atornillado del dispositivo es realizado por un operario.

TIEMPOS DE CICLO (seg)									
MODELOS CON ROBOT					MODELOS SIN ROBOT				
KUKA	KUKA+YUMI	YUMI	T_CINTA	AGV	Operario	Operario	Operario	Láser	AGV
Tiempo detección inicial	Tiempo ensamble piezas	Tiempo atornillar	Tiempo Inspección	Tiempo descarga lote	Tiempo detección inicial	Tiempo ensamble piezas	Tiempo atornillar	Tiempo Inspección	Tiempo descarga lote
3	16	20	3	5	0	20	25	3	5

Figura 8. Tiempos de ciclo - Demostrador 3

3.1.3.2.2 Modelos y simulación

El punto de partida para establecer el diseño del proceso fue el análisis pormenorizado de cómo realizar el ensamblaje colaborativo robot-robot de manera lo más eficientemente posible. Posteriormente y de manera gradual se fueron definiendo diferentes escenarios o modelos de trabajo a simular en entorno Quest con el fin de optimizar aspectos asociados con la productividad y la eliminación de tareas de no valor añadido. En este caso, las principales variantes en los modelos simulados se corresponden con cambios en la disposición de los puestos de preparación de material y al tamaño de lote de salida de dispositivos de la línea, con el fin de reducir los desplazamientos del operario encargado de la preparación de cajas/bandejas en la entrada de la línea y, por otro lado, reducir los desplazamientos necesarios a realizar por el AGV a la salida de la línea. En la Tabla 4 se resumen los diferentes modelos planteados y simulados durante el diseño del proceso del Demostrador 3.

Tabla 4. Modelos de simulación – Demostrador 3

MODELO	DESCRIPCIÓN
1	1 operario + 1 kuka + 1 yumi + AGV - Célula preparación material en forma de L - Tamaño lote final: 1
2	1 operario + 1 kuka + 1 yumi + AGV - Célula preparación material en forma de L - Se acerca operario al inicio cinta - Tamaño lote final: 1
3	1 operario + 1 kuka + 1 yumi + AGV - Célula preparación material en forma de U - Operario próximo a al inicio cinta - Tamaño lote final: 1
4	1 operario + 1 kuka + 1 yumi + AGV - Célula preparación material en forma de U - Tamaño lote final: 9
5	2 operarios (SIN ROBOTS) - Célula preparación material en forma de U, próxima a inicio de la cinta - Tamaño lote final: 1

A continuación se enumeran los **principales puntos evaluados** para el diseño del proceso final del Demostrador 3:

- Optimización:** se ha trabajado en lograr un proceso de ensamblaje colaborativo robot-robot eficiente. a) Para lograr una mayor eficiencia se han colocado dos robots enfrentados (KUKA LBR iiwa y ABB Yumi) realizando tareas de ensamblaje. b) La disposición de los

objetos a coger se plantea por medio de bandejas en las que haya una pieza de cada tipo, dispuestos de una determinada forma, como soporte para facilitar el agarre de los objetos por medio de una pinza. La preparación de esta bandeja se mostrará al operario y colocará sobre la mesa del KUKA las necesarias para realizar varios dispositivos. La preparación de las bandejas se puede realizar previamente y, a medida que se vaya realizando el ensamblaje, el operario puede seguir preparando nuevas bandejas. c) Se incluirá un sistema de visión que permitirá la detección de los objetos sobre la mesa para su agarre en el orden adecuado para el ensamblaje. Si una pieza no está en la bandeja o no es capaz de detectarla porque no se ha colocado correctamente, se avisará al usuario para que la recoloque y mientras tanto, el robot buscará otra bandeja para realizar el montaje. d) La disposición de las bandejas sobre la mesa tendrá una cierta libertad pero dentro de unos márgenes para minimizar los fallos y ajustar así los tiempos de producción. e) Se han buscado posiciones de entrega y recepción de las piezas entre robots que minimicen los tiempos y faciliten los siguientes pasos. Estos movimientos se irán perfeccionando durante las siguientes tareas. Algunos de los movimientos son: el robot iiwa dispondrá de una pinza que permitirá el agarre por el lado ancho de los objetos para que con la pinza adaptada del Yumi pueda cogerla por la parte estrecha; LBR iiwa realizará el giro conveniente de las piezas para entregárselas a Yumi de la forma adecuada para coger con una mano y realizar el ensamblaje directamente sobre el conjunto que sostiene con la otra; la primera pieza se entregará directamente a la mano de Yumi sobre la que se realizará el ensamblaje; mientras Yumi realiza el ensamblaje, iiwa buscará la siguiente pieza a montar, etc. f) La operación de atornillar requerirá un estudio profundo en las siguientes fases debido a su complejidad. Se está valorando fabricar una herramienta especial para que el robot Yumi pueda cogerla y atornillar hasta el fondo. En ese caso se contempla la opción de colocar el dispositivo sobre la cinta parada para poder coger los tornillos y la herramienta utilizando los dos brazos. Una vez atornillado, la cinta se pondrá en movimiento. En caso de atornillar directamente con las pinzas de Yumi, planteamiento inicial, éste cogerá tornillos con un brazo mientras sostiene el conjunto con el otro y posteriormente deposita el dispositivo sobre la cinta y se inicia el movimiento. g) La inspección del ensamblaje se realiza al final, considerando que las piezas ya han sido inspeccionadas anteriormente para evitar así ensamblar piezas con defectos de fabricación. Se inspecciona que el ensamblaje es correcto ya que si no han encajado las piezas, el dispositivo no tendrá un cerramiento plano si no con cierta inclinación. h) Para evitar cuellos de botella se plantea un buffer de almacenamiento al final de la línea. De tal forma que el AGV recogerá de ahí los objetos OK y realizará su traslado a la zona de almacenaje. Así, si el AGV está realizando otras operaciones, tendrá una zona donde se van estacionando temporalmente los dispositivos hasta su distribución final. i) La velocidad de la cinta, su longitud, así como la colocación del sistema de inspección se ha analizado para buscar la combinación más óptima para que el dispositivo pueda ser inspeccionado a tiempo para su descarte en caso de ser incorrecto. Este proceso se analizará en detalle en fases posteriores para lograr una máxima eficiencia.

- Reducción de movimientos y transportes: a) La simulación de los diferentes modelos planteados ha permitido establecer una disposición de elementos en la que se minimicen los movimientos y transportes de piezas. b) Los contenedores de piezas se sitúan cerca de la

zona de ensamblaje para facilitar la preparación de las bandejas por parte del operario. c) La zona de almacenaje de dispositivos ensamblados se sitúa cerca del final de la línea para reducir el desplazamiento. d) Los movimientos de ensamblaje se realizan por los robots mientras que el operario va preparando las bandejas de piezas a ensamblar.

- Buena movilidad de los trabajadores: la colocación de los robots así como la distribución de los diferentes sistemas de seguridad se ha realizado con el objetivo de garantizar una buena movilidad de los trabajadores y haciendo accesibles fácilmente ambos demostradores en el espacio disponible. De esta forma, los robots van a estar realizando operaciones a mayor velocidad cuando los operarios se encuentran fuera de las zonas de protección, bajando su velocidad si se acercan y parando el funcionamiento del robot paralelo en caso de acercarse el operario a una distancia donde pueda meter el brazo en la zona de trabajo de Fanuc M-1.
- Posibilidad de ampliaciones futuras: a) En lugar de disponer de bandejas con las tres piezas, se podría considerar que si las piezas se fabrican independientemente, se metan en cajas y se coloquen directamente sobre la mesa de acceso del KUKA LBR iiwa para que el brazo robótico realice el picking de las cajas. b) La salida de la línea debiera automatizarse para favorecer la creación de lotes adecuados para su transporte al almacén en un soporte y tamaños óptimos desde un punto de vista productivo.

3.1.3.2.3 Resultados

En la Tabla 5 se muestran los indicadores definidos para evaluar los diferentes modelos propuestos de configuración del Demostrador 3. En cada caso, se simuló un turno de trabajo de 7,5 h y, como puede comprobarse, en todos los modelos en los que intervienen robots (modelos del 1 al 4) el cuello de botella resulta ser el robot Kuka, con una saturación del 100 %, obteniéndose en todos ellos una producción similar. Sin embargo, la saturación del operario encargado de la preparación de la caja se reduce de manera significativa en a medida que acercamos el puesto de preparación a la zona de inicio de la línea (modelos 3 y 4). Por otro lado, la utilización del AGV para la descarga de la línea también se ve optimizada si creamos lotes finales de 9 dispositivos, que son el tamaño máximo que puede transportar el robot suponiendo que no se realiza el apilamiento de lotes a la salida de la cinta.

Tabla 5. Resultados simulación Demostrador 3

	SIN ROBOTS			
	Tamaño lote final = 1	Tamaño lote final = 1	Tamaño lote final = 1	Tamaño lote final = 9
	MODELO 1	MODELO 2	MODELO 3	MODELO 4
Tiempo simulación (h)	7,5	7,5	7,5	7,5
Producción (dispositivos)	1384	1369	1367	1359
Utilización operario 1 (%)	72%	52%	47%	47%
Utilización operario 2 (%)	-	-	-	-
Utilización KUKA (%)	100%	100%	100%	100%
Utilización YUMI (%)	82%	81%	81%	81%
Utilización AGV (%)	43%	43%	46%	5%
Distancia recorrida operario 1 (m)	11.796	8.564	7.803	7.803
Distancia recorrida AGV (m)	2.918	2.885	2.883	320
Lead Time promedio (min/dispositivo)	0,33	0,33	0,33	0,33

En base a lo anterior, puede afirmarse que el Modelo 4 representa la alternativa más óptima de las propuestas suponiendo la utilización de robots en la línea. Por otro lado, tal y como puede comprobarse en la tabla anterior, si comparamos los resultados del Modelo 4 con los del Modelo

5, que representaría una situación de partida (sin robots), se constata que el modelo con robots permite un incremento de la producción del 19 % y una reducción del Lead Time promedio por dispositivo procesado del 16 %. Todas las consideraciones anteriores fueron tenidas en cuenta para el diseño del proceso del demostrador 3 y el resto de tareas vinculadas al mismo.

3.1.3.2.4 Requisitos y caso de uso final tras el diseño del proceso

Tras la simulación realizada para este demostrador, se concluye que existirá un almacenaje de piezas que componen el dispositivo así como uno para los dispositivos ya montados. Tras el diseño del proceso para su optimización, el setup del demostrador se ha actualizado resultando en lo mostrado en la Figura 9.

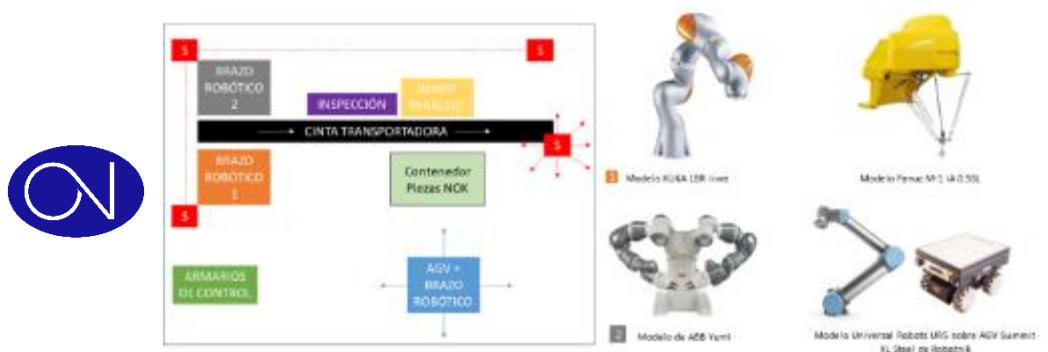


Figura 9. Setup esquematizado del Demostrador 3 (final)

Las acciones a realizar serían las del montaje del dispositivo, en las que conjuntamente varios robots realicen el montaje/ensamblaje de estos dispositivos de manera automatizada, siendo los pasos generales a seguir:

- 1) **Ensamblaje:** a) El operario deposita las piezas necesarias en la mesa de acceso del KUKA LBR iiwa para poder ensamblar varios dispositivos; b) iiwa coge la pieza superior de la bandeja; c) iiwa entrega la pieza superior a Yumi; d) iiwa coge la placa base (PCB); e) iiwa entrega la PCB al brazo libre de Yumi; f) Mientras Yumi pone la PCB sobre la anterior pieza, iiwa coge la pieza inferior; g) iiwa entrega la pieza inferior al brazo libre de Yumi; h) Mientras Yumi sitúa la pieza inferior sobre el conjunto anterior para crear el encerramiento (carcasa), iiwa vuelve a su posición inicial.
- 2) **Colocar tornillos y atornillar:** i) Yumi coloca los dos tornillos de ensamblaje y los aprieta para lograr el dispositivo final completo; j) Una vez atornillado, el dispositivo se coloca sobre la cinta y se pone en movimiento para pasar a la etapa de inspección.
- 3) **Inspección:** k) Con un sistema de triangulación láser (tipo Gocator 2350), se inspecciona que el dispositivo que pasa por esta etapa es correcto (por ejemplo, tornillos bien puestos, planitud del conjunto respecto a la cinta). Si el dispositivo es OK, sigue por la cinta hasta el final. Si el dispositivo no es correcto, Fanuc M-1 lo expulsa de la cinta a la caja de dispositivos descartados; l) Los dispositivos correctos los recogerá el AGV y los dispondrá en la zona de almacenaje.

La distribución final del demostrador 3 así como el proceso optimizado son los detallados en este apartado. Teniendo en cuenta esto, la aplicación e interfaz desarrollada ha tratado de actualizarse esta anualidad en lo posible, si bien la versión implementada totalmente se hará la próxima anualidad.

3.1.3.3 Diseño y simulación Demostrador 1

Una vez definidas las configuraciones óptimas de los demostradores 2 y 3 por separado, se procedió a la simulación de sus modelos de manera simultánea, utilizando para ello un modelo integrado por todos los elementos definidos para ambos procesos y que constituyen por tanto el entorno del demostrador 1. En este nuevo modelo se han mantenido las configuraciones previas de los procesos, aunque se han implementado pequeñas mejoras con el fin de mejorar el tránsito de los operarios del AGV. En la Figura 10 se muestra el modelo final correspondiente al demostrador 1.

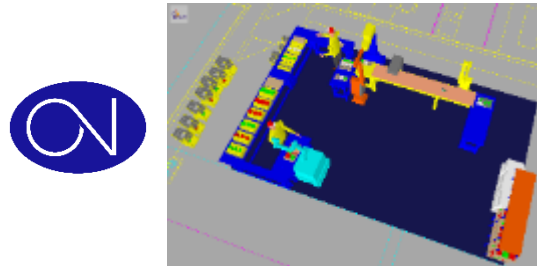


Figura 10. Modelo Quest Demostrador 1

De esta manera, se pudo constatar la viabilidad de funcionamiento de ambos procesos a partir de los recursos disponibles. Se dedicó especial atención al caso del robot colaborativo AGV, al ser un recurso compartido por ambos procesos y por tanto pudiera ser a priori un factor limitante sobre la capacidad productiva de alguna de las líneas. Según los resultados de la simulación se comprobó que el AGV podía sin problemas atender a ambos procesos, según los tiempos y los tamaños de lote fijados. Los resultados obtenidos se resumen en la Tabla 6, donde se puede comprobar que el proceso correspondiente al demostrador 3 sigue estando limitado por el ensamblado de piezas realizado por el Kuka. La capacidad productiva de ambas líneas se mantienen respecto a los resultados obtenidos con las simulaciones individuales y el robot AGV, donde ahora ya alcanza una saturación del 70 %, es capaz de atender a ambas líneas sin producir esperas.

Tabla 6. Resultados simulación Demostrador 1

	Tamaño lote final DEMO 3 = 9 Dispositivos
	MODELO 1
Tiempo simulación (h)	7,5
Producción BANDEJAS - DEMO 2	304
Producción DISPOSITIVOS - DEMO 3	1359
Utilización operario DEMO 1 (%)	39%
Utilización operario DEMO 2 (%)	60%
Utilización KUKA (%) - DEMO 3	100%
Utilización YUMI (%) - DEMO 3	81%
Utilización AGV (%) - DEMO 2 y DEMO 3	70%
Distancia recorrida operario DEMO 2 (m)	4.149
Distancia recorrida operario DEMO 3 (m)	9.915
Distancia recorrida AGV (m) - DEMO 2 y DEMO 3	6.758
Lead Time promedio DEMO 2 (min/bandeja)	1,48
Lead Time promedio DEMO 3 (min/dispositivo)	0,33

Teniendo en cuenta todas estas conclusiones, la distribución final del demostrador 1. Gemelo digital es la mostrada en la Figura 11.

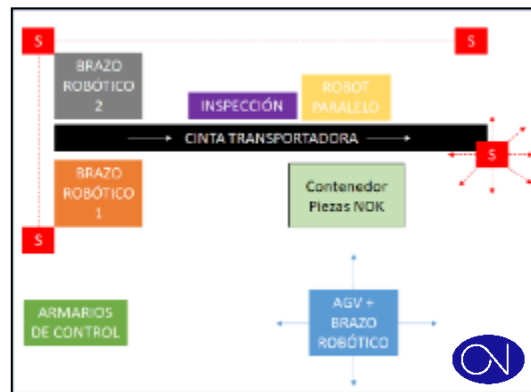


Figura 11. Setup esquematizado del demostrador 1 (final)

Teniendo en cuenta esto, la aplicación e interfaz desarrollada ha tratado de actualizarse esta anualidad en lo posible.

3.2 HITO 2. GEMELO DIGITAL (REALIDAD VIRTUAL PARA EL CONTROL DE ROBOTS)

3.2.1 T2.1. Desarrollo de los modelos matemáticos

NOTA: el desarrollo de este demostrador se ha realizado para todos los manipuladores robóticos considerados inicialmente.

En esta tarea se ha realizado el modelado matemático de los robots tipo seleccionados en la instalación, ya que para poder simular el comportamiento de éstos, se necesita un programa que, conocido el punto que se desea alcanzar, sea capaz de posicionar cada una de las partes del robot de la simulación de forma acorde a su equivalente real. Para ello se proporciona al programa una descripción matemática de cada uno de los manipuladores robóticos. En esta simulación se va a hacer uso de las ecuaciones cinemáticas, que permiten conocer los parámetros necesarios en cada uno de los actuadores del robot para alcanzar una configuración espacial determinada.

La forma de obtener estas ecuaciones viene ligada en gran medida a la configuración del robot, atendiendo a ello podemos clasificar los robots de la instalación en:

Robots móviles.

- Holonómicos:

Robots fijos.

- Cadena cinemática abierta.
- Cadena cinemática cerrada.

Para cada uno de ellos se han desarrollado las ecuaciones matemáticas de la cinemática directa e inversa de cada uno de estos robots así como de la dinámica inversa. A continuación se muestra brevemente cada uno de los robots de los cuales se han

3.2.1.1 Modelos matemáticos – cinemáticas

A lo largo de esta tarea se han desarrollado las ecuaciones de las cinemáticas directa e inversa de todos los robots tipo seleccionados en la instalación del demostrador del gemelo digital. A continuación se resumen brevemente cada uno de los robots implicados y sus características.

FANUC – cinemáticas:

El modelo M-1 iA 0.5SL de Fanuc es un robot de cadena cinemática cerrada con cuatro grados de libertad: tres para el posicionamiento de la plataforma inferior, y otro que permite la rotación de la herramienta. Su cadena cinemática está formada por una base fija que se une a una plataforma móvil mediante tres mecanismos idénticos que denominaremos “piernas”. Tres actuadores rotativos se encargan de dotar de movimiento al manipulador, cada uno de ellos accionando una de las “piernas”, las cuales cierran el lazo al unirse en la plataforma inferior. Las “piernas” están formada por un eslabón que rota en torno a uno de sus extremos por la acción de su respectivo actuador, en su otro extremo se une a un mecanismo de cuatro barras, llamaremos al punto de unión “rodilla”. Tanto la “rodilla” como el punto de unión con la plataforma móvil permiten tres grados de libertad.



Figura 12. Modelo M1-iA 0.5SL de FANUC

Plataforma móvil Robotnik – cinemáticas:

El modelo de robot Summit XL Steel de Robotnik es un robot móvil con tres grados de libertad. Es de tipo holonómico, es decir, puede moverse sin restricciones en todas las direcciones posibles de su espacio de trabajo en un momento dado. Esto es posible gracias al uso de ruedas de tipo “Mecanum”, este tipo especial de ruedas cuenta en su periferia con una serie de ruedas pasivas dispuestas formando un ángulo de 45° con respecto al eje de giro.

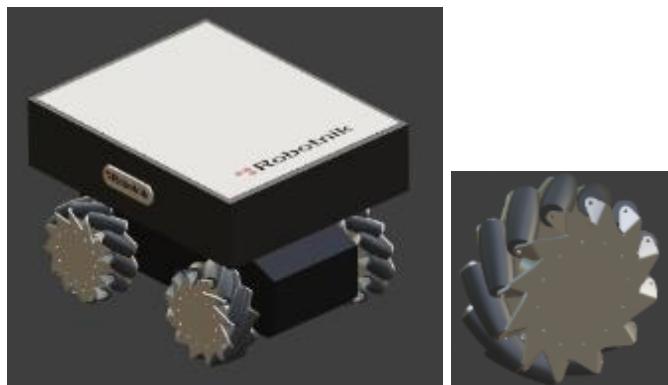


Figura 13. Detalle del robot Summit Steel de Robotnik (izqda..) y sus ruedas (dcha.)

Controlando la velocidad y el sentido de giro de cada uno de los actuadores de entrada puede obtenerse el desplazamiento deseado en cualquier posible dirección del espacio de trabajo. Se van a simplificar los cálculos cinemáticos suponiendo que:

- El robot se desplaza en una superficie totalmente plana.

- El punto de contacto entre el suelo y cada una de las ruedas no varía en ningún momento con respecto a la posición de la propia rueda.

ABB Yumi y KUKA LBR iiwa – cinemáticas:

En la instalación existen dos robots redundantes (KUKA LBR iiwa y ABB Yumi), es decir, su número de grados de libertad ofrecido es superior a los grados de libertad de su espacio de trabajo. Ambos robots cuentan con siete grados de libertad estando limitado su espacio de trabajo a seis grados de libertad (tres para posicionar la herramienta en el espacio y otros tres para orientarla). A los robots con estos grados de libertad se les suele denominar **antropomórficos** ya que suelen estar basados en los brazos humanos, que también poseen siete grados de libertad. Nótese que en el caso de ABB Yumi, hay dos brazos idénticos, por lo que se analiza un único brazo y se extrapola al otro. Ambos modelos de robots se muestran en las imágenes inferiores:



Figura 14. Modelo de KUKA LBR iiwa



Figura 15. Modelo de ABB Yumi

3.2.1.2 Modelos matemáticos – dinámicas

Como futura mejora de la simulación podrán añadirse al programa las ecuaciones dinámicas de los robots de la instalación, que permitirán tener en cuenta en la simulación las fuerzas implicadas en cada uno de los movimientos de los robots. Para el estudio de la dinámica de los manipuladores es necesario definir y estudiar primeramente características proporcionadas por los fabricantes, como son los centros de gravedad y tensores de inercia.

Podemos dividir las ecuaciones dinámicas en:

- **Dinámica directa:** conocido el par aplicado en los actuadores obtener las aceleraciones. Es útil cuando se dispone de sensores de par en los eslabones del manipulador pero no siempre ocurre.
- **Dinámica inversa:** dada una trayectoria, velocidad y aceleración del efector final obtener el par aplicado en los actuadores.

En robótica resaltan dos métodos para obtener estas ecuaciones^{13, 14, 15}:

- El **método de Lagrange-Euler**: proporciona unas ecuaciones finales bien estructuradas con los diversos pares y fuerzas que intervienen en el movimiento (inercia, centrífuga, Coriolis, gravedad). Sin embargo, es ineficiente desde el punto de vista computacional, ya que el número de operaciones crece con una potencia de cuatro en relación con el número de grados de libertad del robot.
- El **método recursivo de Newton-Euler**: se parte de la ley de conservación de par y fuerza. Se aplica habitualmente en robots de 6 grados de libertad ya que el coste computacional es mucho menor que el del método anterior.

El demostrador del gemelo digital no requiere de las ecuaciones dinámicas para la simulación por ello no se muestran estas desarrolladas.

En términos de modelado dinámico se pueden diferenciar varios tipos: robots rígidos y con articulaciones flexibles.

3.2.2 T2.2. Modelado virtual de los robots

Una vez pre-seleccionados los robots, se realiza su modelado virtual. Estos elementos se modelan / importan a Blender y se optimizan para reducir el número de vértices disponible para garantizar que estos elementos se pueden utilizar dentro de la interfaz de usuario en Unity y no ralentizan el conjunto. De lo contrario, no se dispondría de una interfaz ágil y por tanto, perdería el interés de cara a su aplicabilidad.

Cada uno de los elementos del demostrador se han virtualizado y los principales resultados de esto, se detallan a continuación. En la Figura 16 se muestra el modelo virtual del **AGV tipo**, y el soporte sobre el que irá posicionado el brazo robótico UR. Para lograr este modelo final, se ha trabajado en reducir el tamaño de los elementos, en particular del chasis. En la versión final del modelo, se observa que, aun sumando todas las mallas (plataforma, chasis y ruedas), el número de vértices es de 25528, por lo que andamos lejos de la limitación de unos 65.000 vértices por malla para exportar a Unity3D (donde se integrará la aplicación final). En este caso, se partió de modelos CAD aportados por el fabricante.

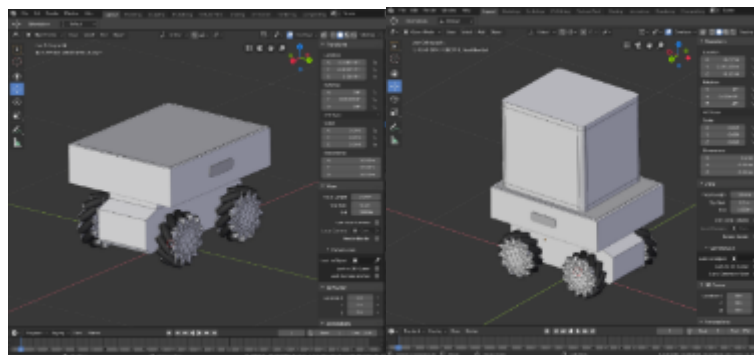


Figura 16. Modelo virtual de AGV

¹³ Barrientos A, Peñín LF, Balaguer C, Arcil R. (1997) Fundamentos de Robótica. McGraw-Hill; 28-314 p.

¹⁴ Spong MW, Hutchinson S, Vidyasagar M. (2004) Robot dynamics and control. 2nd ed. Vol. 28, Automatica. 5-303 p.

¹⁵ Kardoš J. (2010) The Simplified Dynamic Model of a ROBOT Manipulator. En: Proceedings of the 18th International Conference. p. 1-6.

En la Figura 17 se muestra el modelo final del **brazo UR** en Blender. En este caso, el modelo consta de más de 72.000 vértices, lo cual es un límite superior de los 65.000 vértices comentados. No obstante, lo que limita la aplicación en Unity es que los elementos móviles superen el valor de los 65.000 vértices. En este caso, el modelo está formado por un total de 7 mallas (la base y 6 articulaciones). Como ninguna de ellas individualmente supera el valor máximo establecido, se puede afirmar que el modelo está optimizado para su aplicación en Unity3D. Al igual que en el caso anterior, en este caso, se partía de ciertos modelos aportados por el fabricante, por lo que la virtualización fue más sencilla.

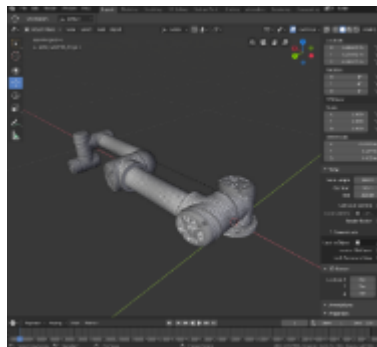


Figura 17. Modelo virtual de brazo robótico UR5

En el caso del robot **FANUC**, no se disponía de información 3D por parte del fabricante. De este modo, desde un software técnico se partió de cero, creando imágenes del robot (alzado, planta y perfil) mostradas en la Figura 18. Estas han sido importadas en Blender y se han usado como base para modelar a la escala adecuada el manipulador robótico.

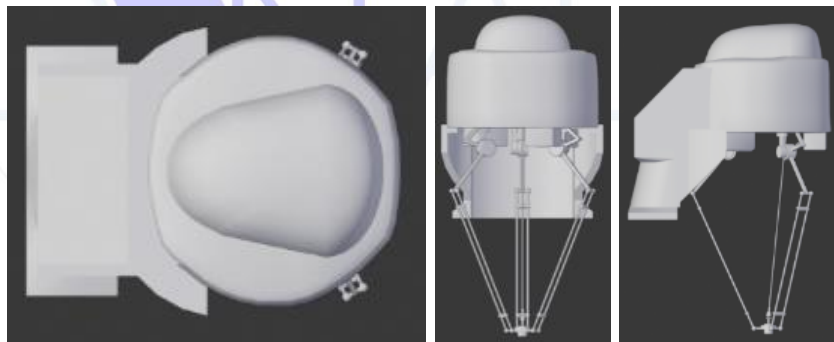


Figura 18. Alzado, planta y perfil del manipulador robótico FANUC

En la Figura 19 se ve el modelo virtual completo del manipulador robótico FANUC y éste tiene un total de 10434 vértices, por debajo del límite máximo. Este modelo está formado por una base, 2 eslabones, una base para la herramienta y una junta universal para transmitir rotación. En este caso, se ha debido invertir más tiempo en el diseño del robot 3D que en la propia optimización del modelo para reducir su número de vértices, al contrario que en los casos anteriores. Es decir, empezar el modelado directamente desde cero, permite optimizar los modelos desde el inicio aunque en total, el tiempo de modelado acabe siendo superior.

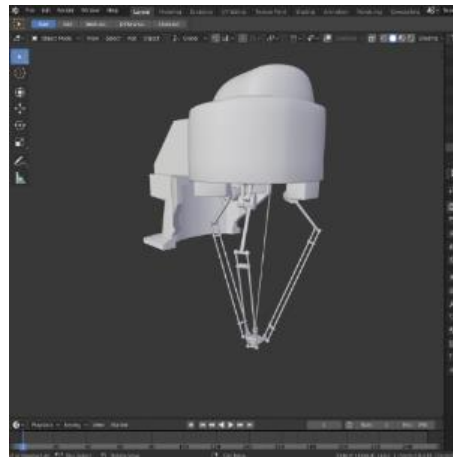


Figura 19. Modelo virtual de brazo robótico FANUC

Por otro lado, se ha desarrollado el modelo virtual del manipulador robótico **KUKA** por medio de Blender. El modelo final tiene un total de 34516 vértices, inferior al máximo permitido para garantizar que la aplicación en Unity3D funciona de manera fluida. En este caso, ese requisito se cumple incluso cuando se dispone del conjunto total considerando todas las mallas que lo forman. Dichas mallas se corresponden a la base, 6 eslabones y la interfaz para conexiones.



Figura 20. Modelo virtual de brazo robótico KUKA LBR iiwa

En el caso del brazo robótico KUKA, si bien se disponía de información aportada por el fabricante y que han facilitado la tarea de modelado, algunos de estos archivos presentaban la extensión “jt”. Esta extensión no puede ser leída directamente por Blender, por lo que tuvo que convertirse previamente en un archivo que Blender entendiera, como es el caso de “stl”. Posteriormente, tuvo que simplificarse la malla ya que las conversiones de archivos de “jt” a “stl” son poco eficientes.

Por último, se ha realizado el modelo virtual del **manipulador Robótico ABB**, cuya versión final se muestra en la Figura 23. En este caso, el modelo está formado por 237016 vértices, por encima del límite de los 65.000 vértices por malla pero se ha ajustado para que ninguna de las mallas que lo forman, supere dicho valor y por tanto, pueda ser utilizada de manera ágil en la aplicación de Unity3D. En este caso, el modelo está formado por las siguientes mallas: base y para cada uno de sus dos brazos, 7 eslabones, un efector final y dos actuadores.



Figura 21. Modelo virtual de brazo robótico ABB Yumi

En este caso, los modelos aportador por el fabricante se encuentran en formato “stp”, no compatible con Blender. Por lo tanto, tuvo que hacerse compatible con un formato que comprendiera como “stl” y posteriormente, como se ha visto para casos anteriores, optimizar la malla.

3.2.3 T2.3. Modelado virtual del entorno

Tras analizar las opciones existentes en IDONIAL Gijón (anteriormente PRODIINTEC) donde poder instalar el Laboratorio de Robótica Colaborativa, se escogió la zona llamada “Showroom” (Figura 22). Partiendo del entorno seleccionado para posicionar los robots y el resto de elementos del Laboratorio colaborativo, se ha utilizado el equipo el “Láser scanner focus3D x130HDR”, disponible en las instalaciones de PRODIINTEC (IDONIAL) para escanear la zona seleccionada. Una vez escaneada la zona, se genera una nube de puntos que debe ser procesada y texturizada de cara a poder implementarla en la interfaz de usuario de Realidad Virtual, en Unity.



Figura 22. Distintas tomas de la zona de “Showroom” seleccionado

Para poder realizar esta tarea, se ha aplicado el siguiente proceso, resultado de la experiencia previa de la entidad en este tipo de actuaciones. Así, tal y como muestra la Figura 23, el proceso para obtener el entorno de la sala de laboratorio y los principales elementos usados para obtenerlo (Hardware y software implicados):

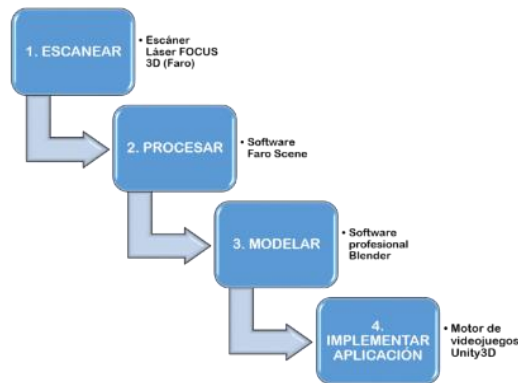


Figura 23. Proceso de obtención de la imagen del Laboratorio de Robótica Colaborativa en realidad Virtual (Gemelo Digital)

A continuación se presenta el detalle de las actividades realizadas en cada uno de los pasos del procedimiento.

PASO 1. ESCANEAR. Como se ha indicado, para el **proceso de obtención de datos** se usa el láser escáner FOCUS 3Dx con HDR de FARO, escáner sin contacto basado en tiempo de vuelo (activo). Es decir, mide el tiempo transcurrido entre la luz emitida y la recibida y utiliza dicha medida para estimar la distancia al objeto. El haz láser emitido es continuo y de potencia modulada. Su precisión depende de la modulación y frecuencia utilizada. Como característica adicional al escaneo, permite realizar fotografías con las que componer una imagen panorámica 360º con la que asignar color a la nube obtenida en el paso anterior. Una vez aplicado el láser escáner para el procesamiento completo de la sala, se obtiene una nube de puntos como la mostrada en la Figura 24.

PASO 2. PROCESAR. El software *Faro Scene* ha sido diseñado de forma específica para los escáneres láser *Focus*, permitiendo procesar y administrar datos de escaneo de manera eficiente mediante el registro en tiempo real, el reconocimiento automático de objetos, el registro de escaneo y el posicionamiento. Una vez obtenida la nube de puntos asociada al laboratorio de robótica colaborativa (paso 1), se observa que está formada por un número muy elevado de puntos, y que éste ocupa un espacio muy elevado. De este modo, se procede a reducirla para facilitar su manejo a la hora de modelar los objetos contenidos en el espacio. Para ello se utiliza el software *CloudCompare*, multiplataforma y de código abierto, que permite el procesamiento de imágenes. De esta forma, se ha obtenido una nube de puntos compuesta por menos puntos y que ocupa un menor espacio. En la Figura 24 se muestra un detalle de la nube de puntos antes y después del procesado.



Figura 24. Nube de puntos antes y después del procesamiento

PASO 3. MODELAR. El modelado se realiza con ayuda de Blender, software profesional, libre y de código abierto para la creación de películas de animación, efectos visuales, modelos 3D etc. Permite desde modelado 3D hasta simulación de partículas, movimientos o renderizados. Incluye un motor de juego integrado.

Adicionalmente, para la creación de texturas similares a las reales, se utiliza el software *Gimp*, un editor de imágenes multiplataforma libre y con permisos para cambiar su código fuente y distribuir dichos cambios. Permite la creación de texturas y mapas relacionados con ellas, a fin de incrementar el nivel de realidad en los modelos diseñados. Algunos ejemplos de esto son los mostrados en la imagen inferior acerca de los posters que se ven al fondo de la escena (según detalle de la Figura 22).



Figura 25. Posters de la escena texturizados

PASO 4. IMPLEMENTAR APLICACIÓN. En esta etapa se usa Unity3D, motor de videojuegos que permite realizar aplicaciones en las que reproducir los entornos diseñados. Presenta una gran versatilidad a la hora de definir comportamientos y funciones para objetos en tres dimensiones, ya que permite utilizar algunos predefinidos previamente en el software y además realizar nuevos comportamientos particulares a través de lenguajes de programación como C# o JavaScript, por lo que favorece la inmersión y la interacción con el entorno modelado. El detalle de este paso se detalla en el apartado 3.2.5.

Además de diseñar el propio laboratorio (sala de gran tamaño) en el que se sitúa el entorno de la aplicación, también se ha realizado el modelado virtual de los robots (apartado 3.2.2) y todo ello se ha implementado conjuntamente en la aplicación desde la tarea de integración (apartado 3.2.5) para ser validado en sus correspondientes pruebas (apartado 3.2.6). La aplicación se ha desarrollado conforme a la interfaz de usuario diseñado y desarrollada desde la tarea correspondiente (3.2.4).

Además de los robots, en base a los requisitos y especificaciones definidas en el HITO 1, se han debido modelar una serie de elementos adicionales que o bien eran necesarios para los demostradores definidos (por ejemplo, piezas) o bien eran elementos necesarios para simplificar la interacción por medio de la realidad virtual, de la simulación. Estos elementos adicionales eran necesarios para conseguir una sensación inmersiva para el usuario en el mundo virtual. En las siguientes figuras se muestran diseños de estos elementos; algunos de estos elementos han sido modificados como consecuencia del análisis del diseño de procesos.

Por un lado, se modeló la **cinta transportadora**, encargada de comunicar la cadena de montaje en los distintos demostradores. Esta se muestra en la Figura 26, y al igual que los robots, es necesaria en todos los demostradores, que serán replicados en este entorno virtual.

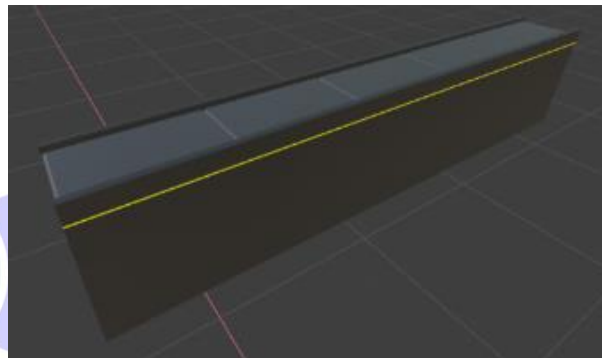


Figura 26. Modelo virtual de la cinta transportadora

Otro de los elementos comunes, es el **Panel Central Robots 4.0** que es el elemento de la simulación que permite al usuario interactuar con el entorno. Este se muestra en la Figura 27, y en la zona oscura que se observa, en su aplicación final aparecerá texto que tendrá varios objetivos: guiar al usuario en la simulación, habilitar simulaciones programadas o informar del estado de la simulación, entre otras.

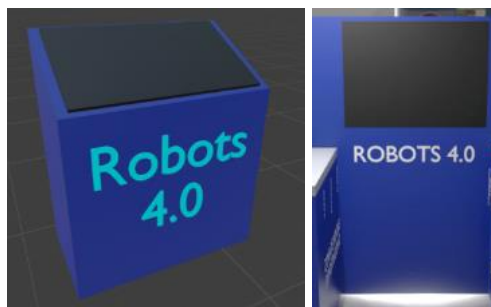


Figura 27. Modelado virtual del panel central Robots 4.0: preliminar (izqda.) y versión final (dcha.)

Para el caso del Demostrador 2. Colaboración hombre – máquina, hay una serie de elementos necesarios como es la **zona de abastecimiento** para este, mostrada en la Figura 28. En la versión final del demostrador, este elemento no aparecerá.

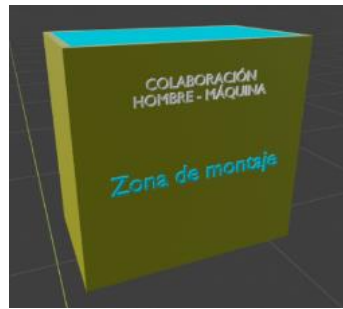


Figura 28. Zona de abastecimiento para el demostrador 2

Como parte del demostrador, hay otros elementos necesarios para asegurar que éste es capaz de reproducir el caso de uso seleccionado. Entre ellos, están las tapas (con y sin defectos) que se insertarán en las piezas y la caja donde irán estas tapas. Ambos se muestran en la Figura 29.



Figura 29. Caja de tapas (izqda.) y tapas con y sin defectos (dcha.)

Todo ello está completado con el conjunto de piezas en colores verdes y rojo, de diversas formas, y con y sin defectos. El conjunto de todas las posibles alternativas se muestra en la Figura 22.

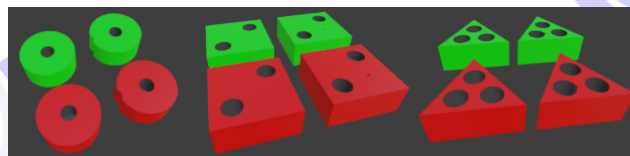


Figura 30. Piezas rojas y verdes, con y sin defectos: circulares (arriba), cuadradas (central) y triangulares (abajo)

Otros elementos adicionales especialmente asociados con el demostrador 2 son los contenedores de bandejas, piezas correctas, desechadas por control dimensional o por error corregible. De este modo, también se consigue eliminar del escenario los objetos que entren en él, almacenándolos en dichos contenedores.



Figura 31. Contenedor para piezas correctas (izqda.) y para piezas defectuosas (dcha.)

Otros elementos específicos del Demostrador 3. Cooperación máquina – máquina, donde se encontrará la **zona de abastecimiento de piezas** (pieza superior, inferior y PCBs) así como un alimentador de tornillos para el robot paralelo, para poder realizar el encerramiento del dispositivo seleccionado para el caso de estudio.

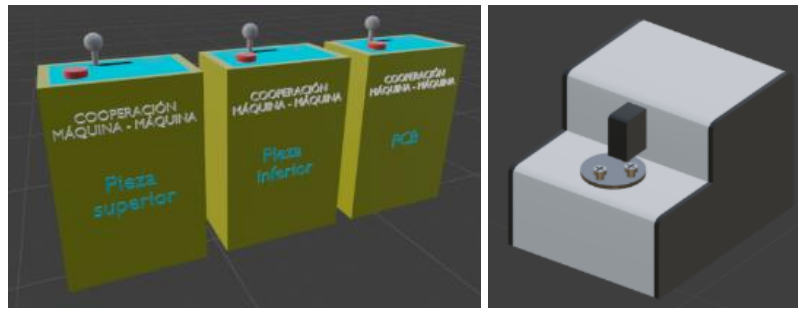


Figura 32. Zona de abastecimiento para el demostrador 3 (izqda.) y alimentador de tornillos genérico (dcha.)

3.2.4 T2.4. Diseño y desarrollo de la interfaz de usuario

La interfaz de usuario se ha realizado basándose en un entorno de visualización común para los dos demostradores físicos definidos (demostradores 2 y 3), permitiendo interactuar con diversos elementos disponibles en la escena. Esta interfaz está formada por el entorno virtual del Laboratorio de Robótica Colaborativa en el que se observan los distintos elementos modelados en el escenario seleccionado correspondiente a la zona de showroom, modelada anteriormente. Para poder interactuar con el escenario, se ha definido una interfaz preliminar que ha sido la que se ha implementado en la fase de integración (tarea T2.5). No obstante, como se ha indicado anteriormente, los resultados de la tarea de diseño de procesos han supuesto la modificación de los procesos del demostrador y, por lo tanto, se ha adaptado el diseño de la interfaz en la versión del demostrador integrada. A continuación se expone cada una de ellas. La interfaz de usuario está implementada sobre el hardware existente en las instalaciones de PRODIITEC (IDONIAL Gijón) gracias a la inversión “Nuevas tecnologías para la fábrica 4.0: realidad virtual y realidad aumentada” (expediente IDI/2016/000252) del Programa Asturias 2016 – 2017.

En la Figura 33 se muestran las funcionalidades preliminares de la aplicación de Gemelo Digital, la cual se ha estructurado en diversos entornos. Según se accede a la aplicación, se dispone del entorno de inicio donde se muestra el escenario digital completo como se ve en la Figura 45. En este entorno se observan todos los elementos que conforman el Gemelo Digital y han sido previamente virtualizados. En segundo lugar, se accede al entorno de selección del caso de estudio y que es acercándose al mando “Robots 4.0” desde donde se explican los tipos de colaboración y se puede escoger el entorno de colaboración a realizar (persona – máquina o máquina – máquina). En función del entorno seleccionado, los elementos que apliquen se activarán (cajas, robots, etc.) y el usuario podrá interactuar con el robot para el caso de uso persona – máquina o bien visualizar cómo se realiza el caso de uso de colaboración máquina – máquina en el montaje de piezas.

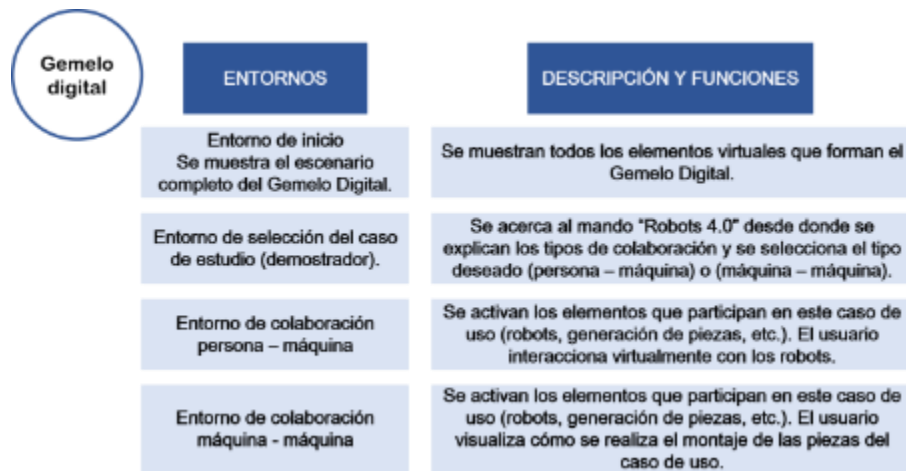


Figura 33. Funcionalidades de la interfaz de usuario (preliminar)

Como se indicaba anteriormente, como resultado de la tarea de diseño de procesos (T1.3), la interfaz se ha rediseñado. La nueva interfaz se basa en la disposición óptima de los elementos en el escenario de acuerdo a los resultados de la simulación, disposición que se muestra en la Figura 45 y Figura 46. Así, la interfaz mostrada en la Figura 34, consta de nuevo de un entorno de inicio donde se muestran todos los elementos virtuales que forman el Gemelo Digital según la distribución óptima resultante de la tarea de diseño de procesos. La interfaz desarrollada permite seleccionar el caso de estudio, en función de si es el demostrador de colaboración persona – máquina o máquina – máquina. Dicha selección se hace desde el mando "Robots 4.0", que dispone de teclado y pantalla virtuales donde visualizar los comandos. En función del caso de uso seleccionado, se dispone de un entorno de colaboración u otro:

- En el caso del entorno de colaboración persona – máquina, se observa la zona donde se sitúa el operario (persona) rodeado de mesas. En su mesa de la derecha, se encuentran las piezas así como las bandejas y según las indicaciones de la pantalla, hace unas tareas u otras. En su mesa de la izquierda, coloca primeramente la bandeja y va depositando en la bandeja buffer las piezas que el sistema solicita para ser inspeccionadas por el robot (UR5). En función del resultado, el robot las pone en la bandeja (si son correctas), y si no, desecha las piezas echándolas en el contenedor correspondiente de piezas mal fabricadas (NOK_D) o piezas mal seleccionadas por el operario (NOK_E). Una vez inspeccionadas, el operario coloca las tapas y deposita la bandeja sobre la plataforma del UR5 donde se realiza la inspección de tapas. Si no está bien, avisa al operario para que realice una operación en consecuencia pero si están bien el AGV lleva la bandeja a la zona de almacenamiento de lotes.
- En el caso del demostrador colaborativo máquina – máquina, se activa la zona de la cinta transportadora donde hay una nueva distribución de robots. En este caso, dos robots (ABB Yumi y KUKA LBR iiwa) están enfrentados y son los encargados de realizar el montaje de las piezas. A continuación, en la línea, un láser realiza el control dimensional de las piezas y determina si están o no correctamente ensambladas. Si no lo están, el robot FANUC las saca de la línea, poniéndolas en la caja de piezas incorrectas mientras que en el caso de que sean correctas, continúan por la línea hasta una zona

de estacionamiento mientras llega el AGV para recoger los dispositivos y almacenarlos en la zona correspondiente.



Figura 34. Funcionalidades de la interfaz de usuario (final)

Dichas funcionalidades han sido implementados en la aplicación tal y como se detalla en el apartado siguiente. Teniendo en consideración estos puntos se define el diagrama de flujo desde el menú principal como se muestra en la Figura 35, por lo que en dependencia de los controles modelados, el usuario puede seleccionar el entorno deseado.

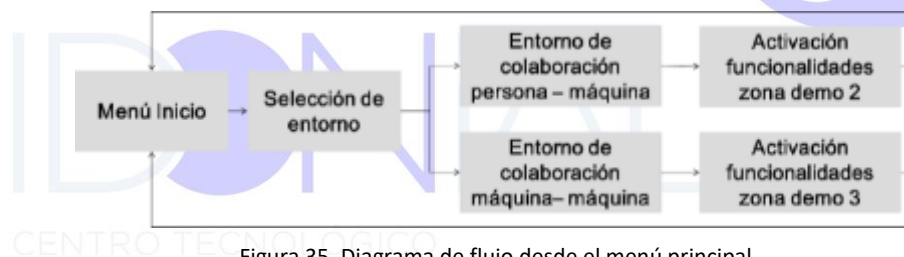


Figura 35. Diagrama de flujo desde el menú principal

Para la realización de la interfaz se realiza primeramente el diseño de wireframes de diversas ventanas para analizar el diseño desde un punto de vista estructural (Figura 36, Figura 37, Figura 38).

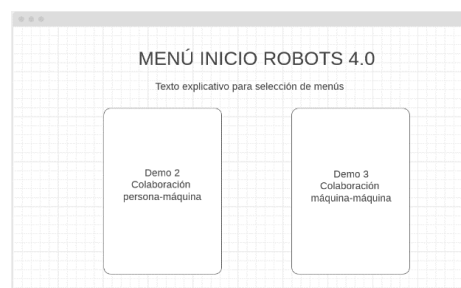


Figura 36. Wireframe del menú principal

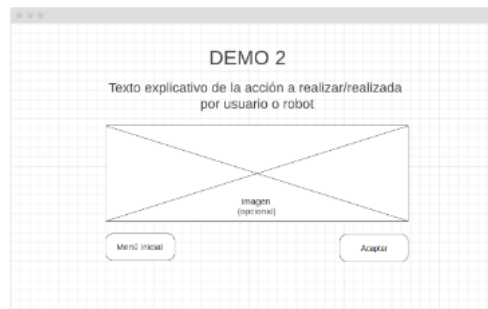


Figura 37. Wireframe del menú de la demo 2 (interacción persona-máquina)

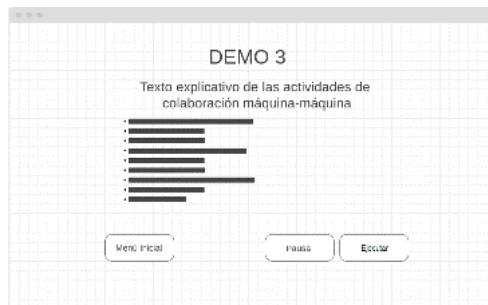


Figura 38. Wireframe del menú de la demo 3 (interacción máquina-máquina)

Una vez acordado el diseño estructural y las funcionalidades e interacciones se validan las decisiones visuales mediante un mock-up visualizando la interfaz propuesta sobre el área gráfica del panel de control Robots 4.0 (Figura 39) y posteriormente se implementan en la aplicación.



Figura 39. Mock-up del menú principal de la interfaz en el panel de control Robots 4.0

3.2.5 T2.5. Integración de la aplicación

En esta tarea se ha realizado la integración de la aplicación en su conjunto en el software Unity3D. Para la integración se ha llevado a cabo una metodología similar a la presentada por R. E. Sánchez-

Alonso¹⁶, habiendo realizado las dos primeras etapas en tareas anteriores y que sirven como entrada para la fase de integración:

- **Previo a integración:**

1. Determinación de las especificaciones: definición de las características (apariencia general y especificaciones técnicas) de cada robot, componente y equipo a modelar. Realizado en las tareas T1.2 y T2.1
2. Modelado 3D: diseño y modelado 3D de componentes (robots, escena real, nuevos componentes). Realizado en las tareas T2.2 y T2.3.

- **Integración:**

3. Montaje de la escena dentro de la plataforma virtual: todos los objetos realizados en Blender en la etapa anterior se exportan y se cargan en Unity3D, añadiendo si es necesario nuevas texturas para obtener la escena deseada. Los componentes que forman parte de la escena se pueden dividir en tres categorías: no animados (elemento decorativo que no se mueve), animados (elemento decorativo que se mueve) y objetos cuyo comportamiento físico debe ser modelado (por ejemplo, robots). Es necesario colocar los elementos en las posiciones deseadas de cada demostrador dentro de la plataforma virtual.
4. Configuración del comportamiento físico: se configuran las características físicas elementales de los componentes del robot y la escena (gravedad, detección de colisiones entre cuerpos, movilidad de articulaciones, masa, fuerzas, torque, etc.). De esta forma se evalúa que el comportamiento del modelo de robot es el deseado. Por ese motivo, inicialmente se comprueba la cinemática inversa y directa en Octave para su validación previa a su implementación en Unity3D. Octave es un software libre que permite la programación en alto nivel para el cálculo numérico.
5. Instrumentación y control virtual: se modelan los algoritmos de control que permiten representar el movimiento automatizado de un mecanismo, se pueden programar rutinas o secuencias a realizar por los robots o comportamientos de otros sensores. Esta etapa requiere también la evaluación exhaustiva del comportamiento para verificar que es el deseado.
6. Interfaz de usuario: su integración permite interactuar con la escena así como ir activando acciones a realizar con el robot y otros elementos disponibles en la escena.

Siguiendo estas etapas, en primer lugar, se han integrado los elementos modelados virtualmente dentro del escenario seleccionado y que ha sido descrito anteriormente.

Una de las primeras acciones, ha sido la de importar los elementos modelados (ya renderizados y texturizados para tener una imagen lo más realista posible), dentro del entorno virtual. Una vez importados, se ha revisado si era necesario su ajuste. A continuación se muestran algunos de los elementos modelados importados dentro del escenario para una muestra representativa de los

¹⁶ Sánchez-Alonso, R. E., Ortega-Moody, J., González-Barbosa, J. J., & Reyes-Morales, G. (2017). Uso de Plataformas para el Desarrollo de Aplicaciones Virtuales en el Modelado de Robot Manipuladores. Revista Iberoamericana de Automática e Informática industrial, 14(3), 279-287.

elementos que lo forman. En la Figura 40 donde se observa el brazo robótico UR5 de Universal Robots.



Figura 40. Brazo robótico UR5 modelado dentro del escenario virtual

En la Figura 41 se muestra el modelo M-1 iA 0.5SL de FANUC dentro del escenario virtual modelado donde se encontrará el Laboratorio.

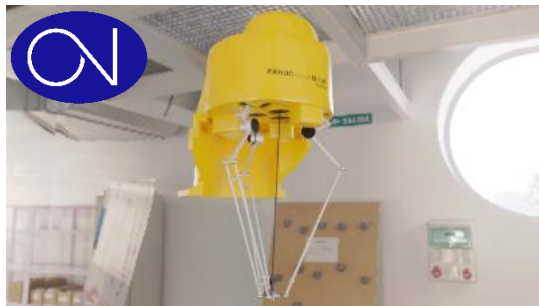


Figura 41. Modelo M-1 iA 0.5SL de FANUC dentro del escenario virtual

En la Figura 42 se muestra otro de los robots seleccionados, en este caso, el robot KUKA modelo iiwa, implementado en el escenario virtual y que irá situado en torno a la cinta transportadora, tal y como se ha descrito previamente.



Figura 42. Robot KUKA modelo LBR iiwa integrado en el escenario virtual

Una vez que todos los elementos han sido integrados dentro de la aplicación, se ha comprobado que ésta funcionaba de manera fluida y no se ralentizaba. Además, una vez integrados, debía posicionarse el conjunto según los requisitos y características detalladas en el HITO 1.

Posteriormente, los modelos matemáticos desarrollados en la tarea 2.1 de este HITO 2, se implementaron en cada uno de los manipuladores robóticos descritos anteriormente. Es decir, se implementaron 4 modelos matemáticos, y que representan un conjunto muy representativo de los distintos robots existentes en mercado por tipo de cinemática, número de articulaciones, fijo o móviles, de uno o dos brazos, etc. Como se ha comentado, se validan las cinemáticas directas e indirectas en Octave y posteriormente, una vez validadas, se implementan en Unity.

Todo ello se ha integrado dentro de la aplicación, dando lugar al Demostrador 1. Gemelo digital. La Figura 43 muestra la versión preliminar del demostrador completo instalado en la aplicación. La imagen ha sido realizada desde el software Blender de cara a que resulte más clara la visualización de todos los elementos en la aplicación.



Figura 43. Demostrador 1. Gemelo digital integrado en la aplicación (preliminar)

Además de esto, se han implementado en Unity3D los comandos que permiten al usuario interactuar con el escenario, conforme a lo descrito en el apartado 3.2.4 (tarea T2.4) así como los modelos matemáticos (tarea T2.1) que gobiernan los robots y les permitirán replicar sus movimientos reales. En la siguiente imagen se muestran ciertas imágenes de los comandos que permiten al usuario interactuar con la aplicación para desplazarse a lo largo de ella:



Figura 44. Interacción del usuario con la aplicación en Unity3D

En las siguientes imágenes (Figura 45, Figura 46) se muestra el diseño final en Blender.



Figura 45. Demostrador 1. Gemelo digital integrado en la aplicación – vista 1 (final) – Blender (arriba) y Unity3D (abajo)



Figura 46. Demostrador 1. Gemelo digital integrado en la aplicación – vista 2 (final) – Blender (arriba) y Unity3D (abajo)

En esta tarea, como se ha visto anteriormente, se han integrado las versiones preliminares de los demostradores, así como la mayoría de los ajustes resultantes de la realización de las simulaciones de la tarea 1.3 al optimizar los procesos de dichos demostradores.

3.2.6 T2.6. Pruebas y validación

Como se comentó en la fase de integración, es necesario evaluar que el comportamiento del robot y la escena es el deseado por lo que para las pruebas y validación se ha establecido una metodología iterativa de pruebas, mejoras y resolución de errores en las etapas de “Configuración del comportamiento físico e “Instrumentación y control virtual”. De esta forma, se trabaja en diversas etapas de - implementación y evaluación ante los retos de los demostradores, partiendo primeramente de un marco teórico de los modelos matemáticos de los robots:

- **Validación del comportamiento físico:**
 - Evaluación de la respuesta de los manipuladores virtuales: se validan los movimientos de los robots de forma individual, haciendo que el extremo del robot se mueva hasta alcanzar la posición y orientación del mando de realidad virtual cuando se pulse el gatillo.
 - Objetos animados: se validan los comportamientos de los objetos que pueden moverse o con los que se puede interaccionar, por ejemplo, el usuario puede coger piezas realizadas para el demostrador 2 e introducir las tapas en los agujeros.
- **Validación del escenario y control virtual:**
 - Evaluación del entorno con una disposición preliminar (Figura 47): pruebas de usabilidad para validación del entorno.
 - Evaluación del entorno con la disposición propuesta final (Figura 48): pruebas de usabilidad con diversos trabajadores para validación del entorno final.

Las pruebas con usuarios, una de las principales técnicas del Diseño Centrado en el Usuario, permiten descubrir qué problemas de usabilidad presenta el entorno virtual observando cómo lo utiliza un grupo de usuarios. Esta observación se ha combinado con el protocolo “Thinking aloud”, en el que el participante expresa verbalmente lo que piensa en cada momento para entender las acciones que realiza. Las pruebas se han realizado con cierta frecuencia ya que con esta información cualitativa se detectan problemas de usabilidad que permiten realizar mejoras de cara a optimizar la experiencia de uso de los mismos. De esta forma, las modificaciones se van realizando en paralelo a la fase de integración para realizar las mejoras durante el desarrollo, y una vez se tuvo la propuesta final gracias al equipo de desarrollo, se procedió a realizar una validación real. Los pasos seguidos para realizar la validación, incluían: toma de requisitos, planificación, definición de métricas, desarrollo de la prueba, resultados de la prueba.

A continuación se muestra un resumen de las actividades y resultados obtenidos en esta tarea. En la Figura 47 se muestra el **diseño preliminar de los demostradores 2 y 3** implementados en el demostrador 1 del gemelo digital sobre el que se han basado las primeras pruebas. Este se basa en la versión preliminar (Figura 43)

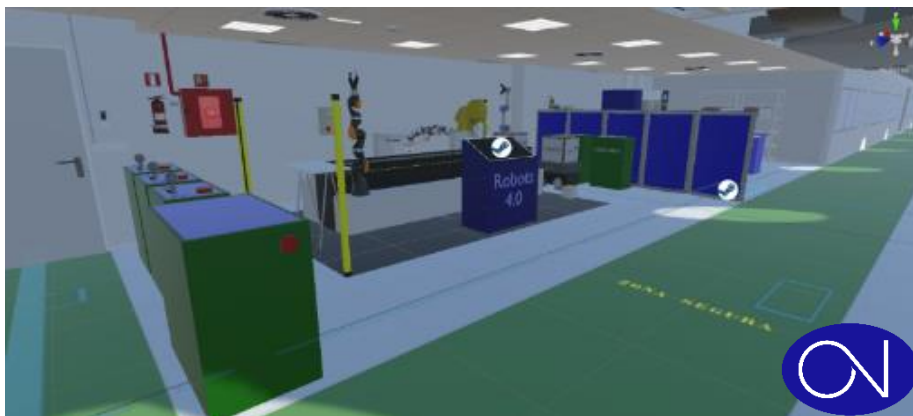


Figura 47. Gemelo digital preliminar de los Demostradores 2 (izqda.) y 3 (dcha.)

Tras una análisis de esta versión inicial, principalmente motivada por el resultado de los diseños de procesos, se ha realizado la **versión final de la aplicación** (principalmente para demo 2), que se puede ver en la Figura 48. Las pruebas en este entorno final propuesto (Figura 45, Figura 46), se focalizan en el espacio del Demostrador 2 (Figura 49) y posteriormente Demostrador 3 (Figura 50).



Figura 48. Gemelo digital final del Demostradores 2

El equipo de proyecto ha realizado los desarrollos necesarios para lograr que la versión final de la aplicación fuera la definida como tal dentro de la tarea de diseño de procesos. En la Figura 49 se muestra la vista general del Demostrador 2 según dicha versión final.



Figura 49. Vista general del Demostrador 2

Por otro lado, se muestra el detalle del Demostrador 3 implementado en la aplicación para realizar las pruebas de validación. En la Figura 50 se muestra la visión general del mismo donde se aprecian los robots que intervienen (FANUC, KUKA LBR iiwa y ABB Yumi) y otros elementos como la cinta transportadora o el láser para inspección de piezas, entre otros.



Figura 50. Vista general del Demostrador 3

De cara a la **validación del demostrador**, los operarios han utilizado la última versión disponible de la aplicación tras los ajustes realizados en la tarea T2.5 de integración. Estos operarios, como potenciales usuarios finales de la aplicación, han aportado sus mejoras de cara a posibles nuevas versiones. A continuación se detalla el flujo de la simulación para las pruebas y los pasos seguidos para navegar por la aplicación:

- **Familiarización del usuario con el entorno:** el usuario se pone las gafas y ve una representación virtual de la sala real en la que se encuentra (Laboratorio de Realidad Virtual y Realidad Aumentada). El usuario se acercará a la televisión de la misma y será donde pueda seleccionar el escenario, se le da una bienvenida y se le enseña cómo interactuar con el entorno por medio de un botón. Una vez familiarizado con el mismo seleccionará el escenario de interés donde habrá acciones relativas a las pruebas iniciales de validación de las cinemáticas y pruebas de los escenarios de los demostradores. En las siguientes imágenes se ve una selección de las distintas pruebas realizadas por el equipo para familiarizarse con el entorno. Dentro de la interfaz de la aplicación es la pantalla descrita como tutorial, donde los operarios aprenden los principales comandos de la aplicación antes de usarla ella en sí misma.
- **Pruebas iniciales de validación de cinemáticas:** en este caso, hay varios botones, uno como tutorial de interacción y otro para comprobar las cinemáticas. A continuación se detalla cada uno de ellos:
 - *Tutorial de interacción:* se abre un escenario para familiarizar al usuario con las interacciones que se va a encontrar en la simulación: 1. Agarrar y soltar objetos y 2. Teletransportarse. Se permitirá salir de este escenario de dos maneras: 1. Pulsando un botón virtual (ya debería saber cómo interactuar con estos elementos por tanto se deja su uso como opcional) ó 2. Pulsando el botón físico “menú” de su controlador (útil en el tutorial ya que le permitirá pasar de cualquier escenario al escenario de inicio).

- *Botón para comprobar cinemáticas:* se abre un submenú con las siguientes opciones: 1. Interacción con UR5, 2. Interacción con M-1 iA 0.5SL, 3. Interacción con IIWA, 4. Interacción con YUMI, 5. Interacción con SUMMIT XL STEEL. Cada botón abre un escenario que permite interactuar con el robot en cuestión. Si el usuario pulsa el gatillo dentro del espacio de trabajo del robot, su extremo se moverá hasta alcanzar la posición y orientación del controlador.

Para probar las cinemáticas, el equipo ha accedido a cada uno de los manipuladores robóticos desarrollados para realizar las pruebas.

- **Pruebas en los escenarios de los demostradores:** en este caso, hay diversos botones según los escenarios a estudiar.
 - *Botón para cargar el escenario sin robots:* se abre el laboratorio de Robótica Colaborativa vacío, el usuario solo podrá teletransportarse. Útil para conocer el estado del laboratorio previa instalación de la célula robótica.
 - *Botón para cargar el escenario con robots:* se abre el escenario con robots. Cuando el usuario se encuentre dentro de la zona de trabajo podrá seleccionar la demo deseada. Demo 2 para la interacción persona-máquina: el paso entre eventos se realiza mediante botones de la pantalla (en la aplicación real no es necesario que el usuario interactúe con la pantalla pero en la virtual es conveniente ya que es un tutorial para aprender a interactuar con la instalación real). Demo 3 para la interacción máquina – máquina: en principio, se prevé que sea similar al del demostrador anterior.

El equipo probando el escenario sin los robots, para que lo pudieran comparar con el entorno real del showroom. Era una manera de que el equipo de pruebas valorara la capacidad de inmersión del entorno y lo pudiera comparar más fácilmente con el entorno real con robots. Igualmente, se han probado los demostradores para validar su funcionalidad, facilidad de uso y utilidad. Las pruebas se han realizado a personas con diferentes perfiles (hombres, mujeres, taller, oficina, desarrolladores, operarios, etc.). Durante las pruebas los usuarios daban feedback de la aplicación, mostrando una respuesta altamente positiva cuando pasaban al escenario con todos los robots e interactuaban con los objetos. Tras pasar por los cuatro escenarios propuestos en la aplicación, se les ha pasado el cuestionario mencionado cuyas conclusiones se pueden englobar en los siguientes puntos:

1) **Adaptación al entorno:**

- a. La adaptación al entorno y manejo del dispositivo ha sido muy rápida. Consideran de gran utilidad disponer de un tutorial en un entorno con pocos objetos para aprender a manejar el dispositivo focalizándose en las funcionalidades de los botones antes de realizar las tareas del demostrador. Además, el hecho de iniciar la aplicación desde un entorno igual al que se encuentran les ha permitido adaptarse con mayor celeridad.
- b. Evalúan positivamente que haya varios escenarios por los que poder navegar según lo familiarizado que se esté con la realidad virtual y la aplicación en sí. De esta forma, pueden realizar el tutorial, probar uno a uno los robots,

interaccionar con piezas, teletransportarse, entre otras acciones antes de pasar a probar la aplicación de los demostradores.

- c. A pesar de haber realizado el tutorial, algunos usuarios han necesitado ayuda para apretar el botón adecuado del mando en alguna situación. Se ha comprobado que mencionando de palabra los botones, el usuario recuerda rápidamente las funcionalidades y puede seguir el proceso sin influir en su acciones.

2) **Usabilidad:** consideran que es muy fácil de usar, amigable, seguir las instrucciones e interaccionar con los objetos de los escenarios.

- a. Los mensajes mostrados en la pantalla son claros y se visualizan correctamente.
- b. La navegación e interacción con objetos (eventos) les ha resultado muy natural.
- c. La inmersión en el entorno ha sido total ya que han considerado todos que se encontraban en las instalaciones reales, una gran sensación de presencia en el propio escenario y que los eventos estaban sucediendo a su alrededor. La iluminación, movimientos de robots, colores y texturas las han considerado adecuadas, salvo en uno de los casos que comentaba que el acabado de los objetos del demostrador 3 no tenían la misma textura (por ejemplo, piezas en poliamida por el tema de la granulometría del acabado de estas piezas en este tipo de materiales).

3) **Seguridad, salud y efectos secundarios:**

- a. La mayor parte de los usuarios no sufren ningún efecto negativo debido al uso del dispositivo. Algunos de ellos han sufrido mareos, y puede haberse debido al uso de gafas de graduación con las propias gafas de realidad virtual.
- b. Los usuarios no han percibido ningún riesgo durante las pruebas. Cuando se acercaban a un objeto de la sala (mesa, pared...) aparece una rejilla verde mostrando así su cercanía por lo que da una elevada sensación de seguridad a la hora de moverse libremente por la sala y poder focalizarse así en la aplicación en sí misma. El único elemento que percibió un usuario a tener en cuenta durante las pruebas es el cable de gafas, para que no se enganche en las piernas, especialmente en los movimientos hacia atrás.

4) **General:**

- a. La satisfacción de uso ha sido muy alta. Los usuarios han disfrutado de la experiencia.
- b. Les ha resultado de gran utilidad poderse mover virtualmente por el laboratorio, considerando de gran potencial la aplicación en sí ya que permite ahorrar tiempo, y emplearse para entrenar, simular y controlar los diferentes elementos involucrados en los procesos de los demostradores.

Con todo ello, los usuarios se han mostrado abiertos a probar nuevas versiones de esta aplicación del gemelo digital.

3.3 HITO 3. ROBÓTICA COLABORATIVA – INTERACCIÓN SEGURA PERSONA – MÁQUINA

3.3.1 T3.1 Diseño de herramientas específicas y ajuste del entorno

Las primeras actividades realizadas en esta tarea han sido las de redefinir el caso de estudio del Demostrador 2 (y posteriormente se hará lo mismo con el Demostrador 3). Esto se ha debido al hecho de que el proyecto llevaba asociada una inversión (IDI/2018/000072) y al no haber sido concedida, el equipo del proyecto ha debido poner en marcha el plan de contingencia. Es decir, la entidad ha debido asumir la inversión del Laboratorio de Robótica con recursos propios. De esta manera, y teniendo en cuenta la búsqueda de financiación, la fecha de entrega de los fabricantes y distribuidores y la carga de trabajo del equipo, el Laboratorio de Robótica Colaborativa ha quedado compuesto por los siguientes elementos:

- Brazo robótico UR5e de Universal Robots
- Brazo robótico TM5-900 de Omron
- Escáner láser de seguridad SICK microScan3 Core
- Pinza robótica Robotiq 2F-140
- Pinza robótica Robotiq Hand-e

Además de estos elementos de inversión propia, cuenta con otros elementos que se han desarrollado y adquirido en el marco del proyecto, como son:

- Cinta transportadora
- Plataformas para colocar los robots

Con todo ello, el Demostrador 2 está formado por dos robots (modelos Omron TM5-900 y UR5e) que se posicionarán cada uno sobre una plataforma. Nótese que antes de decidirse por el uso de un modelo u otro para cada parte del Demostrador, se realizaron pruebas realizadas con ambos modelos, que se detallarán en los siguientes apartados. De cara a la interacción segura persona – máquina del Demostrador 2, el brazo robótico TM5-900 trabajará con una persona que se encontrará en la zona de trabajo colaborativo, y una vez que finalice la tarea conjunta, el robot depositará el elemento sobre una cinta transportadora que se encuentra a su lado. Este robot debe localizar e inspeccionar las piezas con las que trabajar de forma autónoma. Para garantizar la seguridad del trabajador, también habrá un elemento de seguridad posicionado en el entorno. Al final de la cinta, se encontrará el UR5e esperando a que una persona inicie la acción al llegar el elemento y le guíe durante el proceso de recogida y posicionamiento en la plataforma. Todo ello se muestra en la Figura 51.



Figura 51. Demostrador 2: planteamiento general

La Figura 52 muestra en detalle la zona de trabajo colaborativo y los diversos elementos que conforman el demostrador de una forma esquemática: 1) una mesa con diversos contenedores de piezas y bandejas; 2) una mesa de ensamblaje con los contenedores de tapas y de las piezas a desechar así como un buffer donde se irán colocando las piezas que el robot depositará sobre una bandeja; 3) una cinta transportadora donde un robot depositará la bandeja una vez que la tenga completa y un segundo robot al final de la cinta para recoger la bandeja realizada. A la derecha de la imagen se puede apreciar el diseño de las piezas y tapas con las que se ha trabajado.

Los pasos a seguir durante el proceso, son:

- 1) Preparar lote e inspección individual: a) Una vez que el operario y el robot estén en la mesa de ensamblaje, el sistema indica al operario el lote a realizar y le indica las dos primeras piezas a coger; b) Entonces, el operario coge las piezas de la forma deseada, las pone en el buffer; c) El robot se posiciona sobre el buffer, y las analiza con una cámara embarcada. Evalúa la primera pieza, si la pieza es correcta (tipo de pieza, dimensiones y color adecuados) la recoge y la deposita en la bandeja. En caso de ser incorrecta, la deposita en la caja NOK correspondiente y la añade a la lista de piezas a pedir al usuario: si el defecto es corregible (si la pieza es correcta y no se encuentra en ninguna posición de la bandeja), el robot coge la pieza y la deposita en la caja de NOK_E; si el defecto no es corregible (si las dimensiones de fabricación no son correctas), el robot la descarta a la caja de piezas defectuosas NOK_D; d) El proceso se repite con todas las piezas del lote, facilitando el sistema al operario la información de la pieza actual. En cuanto a las piezas incorrectas, el programa lleva una lista de piezas pendientes, por lo que mientras una pieza siga en esta lista, vuelve a solicitarla y sólo la eliminará si la ve correctamente en el buffer.
- 2) Ensamblar tapa: e) Una vez que el lote está preparado e inspeccionado individualmente, el operario ensambla las tapas.
- 3) Inspección lote: f) El robot se posiciona sobre la bandeja y va inspeccionando las tapas a medida que el usuario las va colocando; g) Si es OK, la posiciona en la cinta.
- 4) Distribución del lote: h) La cinta se pone en funcionamiento al detectar la bandeja posicionada; i) Al llegar la bandeja al final de la cinta, ésta se para; j) Un operario acciona al segundo robot para recoger la bandeja y posicionarla en la plataforma de forma colaborativa. De esta forma, posteriormente se puede mover la plataforma con la bandeja a la zona de distribución final o almacenaje de la instalación donde el robot podría realizar la siguiente acción.

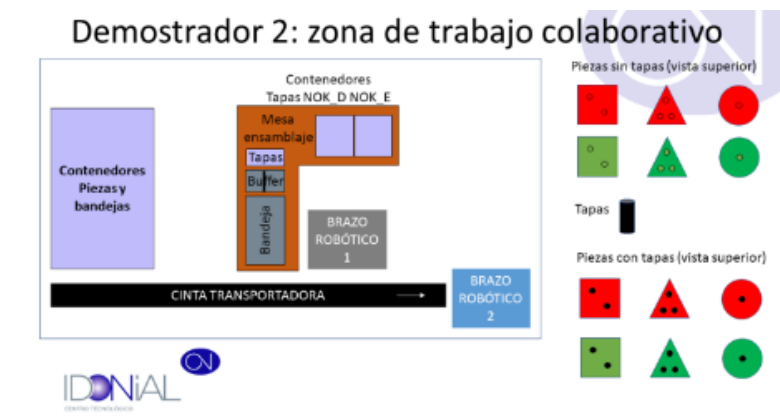


Figura 52. Demostrador 2: zona de trabajo colaborativo

Una vez se dispone de esta información actualizada conforme a las nuevas consideraciones del demostrador, se trabaja en habilitar la zona reservada para el demostrador. Por otro lado, se analiza el diseño de la arquitectura del demostrador de cara a decidir qué componentes es necesario diseñar o bien si se debe optar por soluciones comerciales. En la arquitectura general mostrada en la Figura 53 se distinguen los siguientes bloques principales:

1. **Robot 1:** está compuesto por un brazo robótico y un efector final o herramienta. Debido a que el objetivo es coger unas piezas y una bandeja (tarea pick&place), se opta por una pinza robótica como efector final. En este sentido, se ha optado tanto por los robots colaborativos como por pinzas comerciales compradas por IDONIAL como inversión propia. En el caso de los robots, se dispone tanto del UR5e como TM5-900 y unas pinzas robóticas de Robotiq, compatibles con ambos robots y cuyo rango de trabajo es adecuado para los demostradores del proyecto. El robot deberá estar colocado en una plataforma para tener acceso a la mesa de ensamblaje y a la cinta.
2. **Seguridad:** bloque encargado de la seguridad del entorno de trabajo. Se ha optado por incluir el escáner SICK comprado como inversión propia por IDONIAL, ya que colocándolo a la altura de los tobillos de un operario protege la zona de trabajo colaborativo y monitoriza el acceso.
3. **Comunicaciones:** este bloque contará con los dispositivos necesarios para garantizar la comunicación entre los diferentes componentes. Se optará por soluciones comerciales.
4. **Unidad de control:** se dispondrá de una unidad de control para controlar la ejecución del programa del robot y de las gafas de realidad aumentada (RA). Se optará por dispositivos comerciales.
5. **Realidad Aumentada (RA):** se utilizarán las gafas comerciales de RA disponibles en las instalaciones de IDONIAL.
6. **Sistema de distribución:** una vez que el robot ha finalizado la tarea de interacción con el usuario, posiciona la bandeja de piezas realizadas, a una cinta transportadora. El funcionamiento de dicha cinta estará gobernado por un PLC comercial. La cinta tendrá que tener unas dimensiones adaptadas al entorno del demostrador. Además, en el sistema de distribución se cuenta con un segundo robot que recogerá la bandeja y la depositará en su plataforma con ayuda de un operario.



Figura 53. Arquitectura general del Demostrador 2

Por tanto, los elementos clave a diseñar y fabricar son las plataformas de los robots y la cinta transportadora, mientras que para el resto se optará por soluciones comerciales.

De cara al diseño de la plataforma móvil, se tuvo en consideración que esta debe proporcionar posicionamiento y movimiento de los robots de una forma fácil y flexible. Se buscó información de diversas estaciones móviles comerciales¹⁷ para brazos colaborativos, optando finalmente por un diseño propio para disponer de mayor versatilidad. Para ello, el equipo de IDONIAL ha diseñado una plataforma de trabajo móvil que fuera válida para los dos brazos robóticos colaborativos existentes en las instalaciones de la entidad (OMRON TM5-900 y UR5e), con unos agujeros de fijación para cada uno de ellos en la placa. Otro de los elementos que se ha diseñado ha sido la cinta transportadora necesaria para crear el escenario colaborativo. Una vez diseñados los elementos, se ha procedido a su fabricación e instalación en la zona reservada para ello en IDONIAL. La Figura 54 muestra la plataforma móvil fabricada y ya posicionada en la zona del Demostrador 2.



Figura 54. Plataforma móvil fabricada

En la Figura 55 se muestra el conjunto formado por la plataforma móvil detallada anteriormente así como la cinta transportadora fabricada y montada en la parte trasera de la imagen.



Figura 55. Cinta transportadora y plataforma móvil fabricadas

Teniendo en consideración los componentes evaluados, diseñados, el entorno y la tarea a desarrollar, la Figura 56 muestra la **arquitectura final** del Demostrador 2 de ROBOTS4.0, compuesta por los siguientes bloques comunicados entre sí:

1. **Robot:** está compuesto por el modelo OMRON TM5-900 y la garra Robotiq 2F-140.
2. **Seguridad:** está formado por el SICK scanner MicroScan3 Core, que se encargará de garantizar la integridad de las personas y objetos en el entorno.
3. **Comunicaciones:** compuesto a su vez por un switch de comunicaciones y un router WiFi.

¹⁷ <http://www.ia.omron.com/products/family/3739/lineup.html>

4. **Unidad de control:** formado por la Raspberry Pi 3 B+.
5. **Realidad Aumentada:** incluye las gafas de RA Hololens que se comunican con los elementos del sistema por medio de la red WiFi.
6. **Sistema de distribución:** está formado por la cinta de distribución, accesorios, el PLC y el robot UR5e junto con la garra Robotiq Hand-e.



Figura 56. Demostrador 2: arquitectura

3.3.2 T3.2 Sistema de percepción para la tarea

El robot del demostrador que estará compartiendo la zona de trabajo con el operador durante el montaje de la bandeja debe ser capaz de inspeccionar y localizar la bandeja, el buffer y las piezas de forma autónoma. Por tanto, durante esta tarea se realizó un análisis en profundidad de los sistemas de percepción de la tarea existentes y de aquellos que pueden ser de mayor interés y aplicación para el caso de los demostradores objetivo. Así, en primer lugar, para la selección del sistema de visión se tiene que tener en consideración los aspectos a realizar de cara a la tarea de pick&place (recogida y colocación) e inspección: adquisición de imagen, preprocesamiento, análisis (extracción de características...) y decisión (reconocimiento, clasificación...) con los datos resultados del análisis. En este sentido, debido a que el brazo robótico tiene que recoger objetos y depositarlos en otro contenedor o bandeja, es necesario que el robot conozca la posición de la cámara en su sistema de referencia para relacionar la información de la imagen con el punto físico que debe alcanzar el robot. Para ello es necesario calibrar el plano del objeto que desea ubicar. Después de esto, debe convertir las posiciones de los objetos localizados en el sistema de coordenadas de la cámara al sistema de coordenadas del robot. Para facilitar este tipo de calibraciones externas se dispone de cámaras de visión integradas con el robot o con conexión Plug + Play.

El aspecto clave en una aplicación de pick&place y que influye notablemente en la decisión de elegir un sistema de visión suele ser la posición de recogida y colocación de las piezas¹⁸. Existen **diferentes alternativas a un sistema de visión** a la hora de realizar un pick&place:

- **Posición fija:** si la pieza se coloca en bandejas fijadas a una superficie o en otro tipo de accesorio que asegure una posición constante en relación con el robot simplemente es necesario enseñar las posiciones de trabajo al robot. Se podría usar un programa de paletizado sin necesidad de sistema de visión. Sin embargo, un sistema de este tipo puede

¹⁸ <https://blog.robotiq.com/vision-systems-for-collaborative-robots>

conllevar un accesorio costoso o difícil de diseñar tanto por la pieza como en casos con muchas piezas por tener que alimentar al robot continuamente sin recargar la plantilla. Otro inconveniente es cuando se tienen diferentes piezas, que implica que se necesiten varias plantillas distintas (aumentado el costo), considerando además el tiempo de cambio afectando a la eficiencia del sistema.

- **Alimentadores:** destinados a tomar artículos a granel y presentarlos al robot siempre en la misma posición. Este tipo de equipo puede ser bastante costoso y difícil de ajustar para algunas piezas, teniendo a veces atrapes de piezas. Si se cambia la pieza, es necesario modificar el sistema de alimentación.
- **Cinta:** en el caso de usar una cinta transportadora sin un sistema de visión es necesario tener una línea o punto de referencia para que la pieza siempre se encuentra en la misma posición o ubicación. En este caso, se suele necesitar un sensor de presencia para que el robot sepa cuándo ha llegado al lugar de recogida. Una vez que lo hace, el robot recogerá la pieza con la cinta en movimiento o en parado, según el tipo de robot. En caso de no disponer de una línea de referencia o las piezas no están siempre en la misma posición o en el mismo lugar en el transportador, es necesaria una cámara para ubicar la pieza en la cinta.
- **Sistema de visión:** la incorporación de un sistema de visión en la tarea de pick&place permite encontrar la ubicación y orientación de una pieza en cualquiera de las circunstancias anteriores, aportando una gran flexibilidad al sistema. Si la pieza se encuentra en una superficie plana parada o en movimiento, es posible utilizar una cámara 2D para localizar las piezas ya sea con una cámara fija o montada en el robot. También es posible incorporar una cámara 3D y hacer un escaneo de la superficie para buscar las piezas. Si las piezas cambian, se ajusta el algoritmo de visión en lugar de cambiar de plantillas o el alimentador.

En cuanto a la tarea de **control de calidad con un sistema de visión** durante la propia ejecución, es posible automatizar este proceso sin tener que parar la línea de producción o sacar la pieza de la fabricación. Las funciones de visión más comunes en inspección son coincidencia de patrones según un umbral establecido, concordancia de colores, reconocimiento óptico de caracteres, medición, detección de defectos visuales, conteo, etc. En el caso de coincidencia de patrones, es necesario hacer frente a retos como piezas superpuestas o muy juntas, contraste insuficiente, influencia de luces externas, sombras, muchas variaciones entre piezas, etc.

A la hora de **elegir el sensor más adecuado** hay que tener en cuenta que existen tecnologías 1D, 2D y 3D. Además, hay que tener en cuenta una serie de factores: resolución, tamaño del sensor, distancia de trabajo, campo de visión, distancia focal, contraste, apertura, tiempo de exposición, tasa de adquisición (frames por segundo, fps), trigger, color o blanco y negro, entre otros.

Teniendo en consideración que se dispone de los robots UR5e y TM5-900, se opta por **soluciones de visión compatibles** con ellos y además, son Plug + Play:

- **Robotiq Wrist Camera**¹⁹: cámara RGB diseñada para supervisión de máquinas, ensamblaje y pick&place mediante múltiples funcionalidades como ubicar piezas multi-objeto,

¹⁹ <https://robotiq.com/es/productos/camara-de-articulacion>

validación de color, etc. Válido para UR5e pero actualmente no confirman su funcionamiento con TM5-900.

- **Pick-It²⁰**: cámara 3D que utiliza los datos volumétricos proporcionados para reconocer las formas volumétricas dentro de su campo de visión para carga de máquinas, bin picking, pick&place, ensamblaje, despaletizado. Válido para UR5e pero actualmente no confirman su funcionamiento con TM5-900.
- **Cámara integrada del TM5-900**: cámara RGB de 5Megapixels con iluminación, integrada en el robot por lo que no es posible su uso en el UR5e. Esta cámara 2D permite la referencia automática del robot al plano de trabajo, volviendo la ubicación de objetos flexible, siendo ideal para tareas complejas de ensamblado y permitiendo que el mismo pueda trabajar en distintos puestos de trabajo sin la necesidad de accesorios mecánicos complejos (Figura 57). Así, la cámara localiza los objetos en un amplio campo de visión y la luz de mejora de la imagen permite un reconocimiento de objetos sean cuales sean las condiciones. El sistema de visión incluye funciones como el reconocimiento de patrones, la lectura de códigos de barras y la clasificación por colores, lo que facilita la inspección visual, las mediciones y la clasificación.
- **Cámara Basler acA2500-14gc y acA2500-14gm**: hardware recomendado en el datasheet del TM5-900²¹ pero no confirmado su correcto funcionamiento con UR5e.
- Existen además otras **opciones compatibles con el UR5e²²**: guiado 3D (ToF sensor O3D de IFM Elctronic GmbH, Accupick 3D de Salomon, Photoneo Locator, Smart 3D robot visión con Gocator de LMI Technologies), guiado 2D (Cognex In-Sight 2D, Dahl vision system Static DV-S, Datalogic Impact para diversas cámaras 2D, Omron Microscan Guidance, Sick Ploc2D, SICK Inspector, Visor Robotiq de SensorPart Industriesensorik), e inspección (sistema de control de movimiento de Streamline compatible con varios sensores 2D y 3D, 3D belt pick con SICK Trispector, ER-ONE Dynamic Position System de Enabled Robotics).

Teniendo en consideración que se va a trabajar con el TM5-900 y que actualmente no es posible confirmar que las cámaras Plug + Play funcionen adecuadamente para ambos robots, **se decide utilizar la cámara integrada del Omron TM5-900 que es válida para la aplicación de pick&place e inspección visual que se va a realizar en el demostrador**, teniendo en consideración la distancia de trabajo y el campo de visión mostrado en la Figura 57. Al estar esta cámara integrada en el robot no es necesario realizar ajustes de diseño en el manipulador robótico.

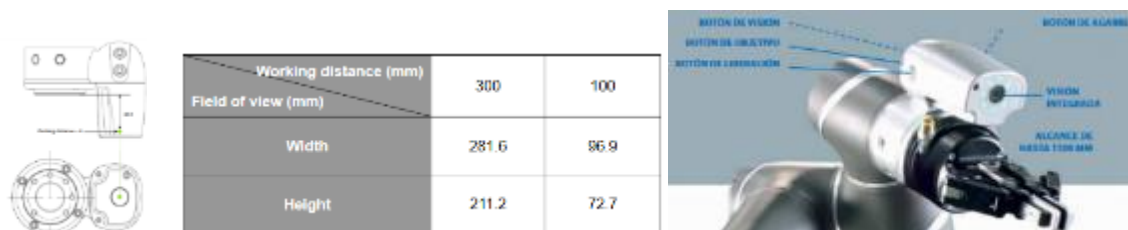


Figura 57. Detalle de la cámara integrada del TM5-900 (izqda. arriba y abajo) así como su distancia de trabajo y campo de visión (dcha. arriba)

²⁰ <https://www.pickit3d.com/applications>

²¹ <https://assets.omron.com/m/73720dec02f8c66c/original/TM5-Datasheet.pdf>

²² <https://www.universal-robots.com/plus/vision/>

Para el desarrollo de **algoritmos de percepción** hay que considerar que el Omron TM5-900 se programa en el entorno de programación TMFlow de Omron, basado en diagramas de flujo. Además, está equipado con TMvision, que es una combinación de hardware (cámara integrada en el robot - detallada anteriormente - y botón de visión) y software (o bien en su versión estándar o bien otra versión que requiere licencia y que está disponible en IDONIAL). En TMvision destacan las funcionalidades de identificación de características, localización de objetos, modo de mejora de imagen, identificación de barras, clasificación por colores o ajuste de la iluminación, entre otras. Los trabajos de visión se programan con ayuda del "Vision Designer". Este software no sólo está preparado para la cámara integrada sino que permite trabajar con tres tipos de cámaras: Eye-in-Hand (EIH), Eye-to-Hand (ETH) y Upward-Looking. TM Vision incluye una función para administrar el almacenamiento de imágenes pero requiere el TM SSD de Omron tanto para guardar imágenes sin procesar como procesadas. Una vez creado un proyecto en TMFlow, se recomienda dirigirse al desplegable derecho y seleccionar vista, así aparecerá una ventana que muestra lo que el robot ve en un determinado instante.

En resumen, en esta tarea se ha realizado un análisis de los diferentes sistemas de visión que hay disponibles y se ha determinado el más adecuado para las necesidades del demostrador. Una vez seleccionada la cámara integrada del robot Omron TM5-900, se ha analizado ésta y sus múltiples opciones para poder aplicarlas al caso del Demostrador 2. Por último, teniendo en cuenta todo ello se han planteado los pasos necesarios de los algoritmos del sistema de visión, que serán desarrollados y optimizados en las tareas posteriores, fundamentalmente en la fase de integración y pruebas (tarea 3.6.).

3.3.3 T3.3 Sistemas de seguridad

En primer lugar, en esta tarea se realiza una evaluación de riesgos del Demostrador 2, junto con el equipo de Prevención de Riesgos Laborales, para el proceso de dicho Demostrador 2. Esta actividad se ha realizado siguiendo los pasos de definición de los límites, identificación de peligros, estimación y evaluación de riesgos. Así, los **riesgos identificados (Ri)**, son:

- **Ri1.** Posibilidad de que un operario entre en contacto con el robot en movimiento.
- **Ri2.** Posibilidad de colocar los dedos en la pinza mientras la pinza está enganchando la pieza.
- **Ri3.** Posibilidad de colocar la mano entre la pieza y la bandeja en el lugar de posicionamiento.
- **Ri4.** Posibilidad de colocar la mano entre la bandeja y la cinta cuando el robot la deposite.

Cabe destacar que la estimación de riesgos de un peligro en particular se ha basado en evaluar la severidad del daño y su probabilidad de ocurrencia (exposición, ocurrencia y posibilidad de evitarlo). Para evitar estos riesgos se establecen una serie de **estrategias de mitigación (EMi)**:

- **EMi1. Setup del entorno:** Medidas complementarias de seguridad y protección en áreas abiertas.
 - Sensor de presencia.
 - Avisos en zonas de riesgo.
- **EMi2. Robot:** Medidas de diseño seguro inherente mediante medidas de seguridad por hardware y software.

- Características de seguridad activa colaborativas.
- Características colaborativas de seguridad pasiva.
- Selección de presión y fuerza.
- Programación de ruta.
- **EMi3. Sistema general:** Medidas de diseño seguro inherente.
 - Lógica del programa.
 - Posición de recuperación ante fallo.
- **EMi4. Operario:** Información de uso.
 - Medidas de sensibilización con los operarios.
 - Procedimientos operativos seguros.
 - Entrenamiento del operario.

En cuanto a las **soluciones de protección de seguridad del Emi1**, para hacer un seguimiento de la persona (operario) de cara a garantizar un entorno seguro, se ha seleccionado el escáner **SICK MicroScan3 Core**²³. Este escáner láser de seguridad está concebido para el uso en entornos industriales difíciles, funcionando correctamente incluso en presencia de suciedad, polvo y luz artificial gracias a la tecnología de exploración safeHDDM®, a la carcasa de aluminio y a un concepto de fijación avanzado (Figura 58).

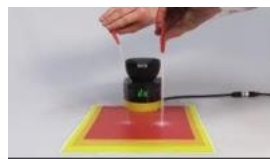


Figura 58. Imágenes representativas del escáner SICK²³

De esta forma, en las posibles **zonas de riesgo** (set-up del entorno) se dispone del escáner que detectará la presencia del operador y ralentizará o detendrá el robot según dónde se encuentre. Además, se utilizarán marcas de suelo para delimitar el espacio restringido, notificando que hay robots en funcionamiento como se puede ver en la Figura 59 donde se muestra un mensaje de “ATENCIÓN ROBOT EN MOVIMIENTO” al salir de la zona segura (pasillo verde), haciendo referencia a la zona de trabajo de los robots.

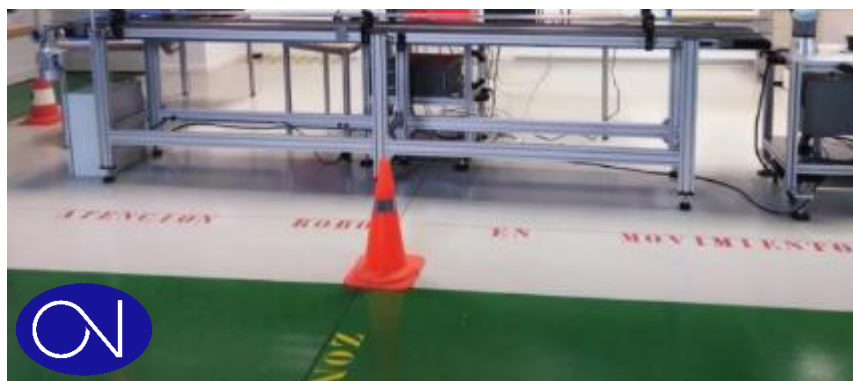


Figura 59. Detalle del mensaje de aviso de la zona de trabajo de robots

²³ <https://www.sick.com/es/es/dispositivos-de-proteccion-optoelectronicos/escaneres-laser-de-seguridad/microscan3-core/c/g295658>

En cuanto al **robot (Emi2) y el sistema general (Emi3)**, es necesario analizar las normas de seguridad ISO 10218 Parts 1&2 (Robots for industrial environments) e ISO TS/15066 (Robots and robotic devices –Collaborative Robots) para determinar que el robot es seguro de operar, mitigando así los riesgos. De hecho, es clave considerar la reducción de riesgos en colisiones con manos y cabezas.

Por tanto, en el caso del Demostrador 2, se tendrán en consideración los siguientes aspectos:

- **Efector final:** limitación de la posición de junta, codo y herramienta/efector final, orientación de herramienta/efector final limitada, se ajustará la presión de la pinza y se asegurará que no dispone de bordes afilados que puedan dañar al operador.
- **Posición del operador:** se buscará la posición más adecuada para evitar posibles puntos de atrapamiento u choque con el robot u otros objetos.
- **Robot:** fuerza, potencia, momento y velocidades limitadas en las zonas colaborativas así como monitorización en continuo. El robot TM5-900 cuenta con funciones de seguridad certificadas, incluyendo limitación de potencia y fuerza siempre activas, cumpliendo así con las normativa 13849-1 con Performance Level = d. El UR5e está equipado con una gama de funciones de seguridad incorporadas así como E/S de seguridad, señales de control digital o análogo hacia o desde la interfaz eléctrica, para conectar con otras máquinas y dispositivo de protección adicionales. Cada función de seguridad y E/S están diseñadas según la norma ISO13849-1:2008 y tienen nivel de rendimiento d (PLd) mediante una arquitectura de categoría 3. Por tanto, la seguridad de estos robots puede venir dada por hardware y software como se verá en la tarea 3.6. El software de ambos robots proporciona una parametrización sencilla de los parámetros de seguridad y además, permite añadir planos de seguridad en caso de ser necesario.

Otro aspecto identificado son los problemas de ciberseguridad. Para mitigarlos, el sistema se encontrará en una red independiente y se dispondrá de un pendrive específico para el robot, que solo puede usarse con ordenadores de la compañía, los cuales cuentan con un antivirus. Además, se evitará el acceso no autorizado a la configuración de seguridad mediante protección por contraseña.

Así, una vez se ha realizado un análisis de riesgos para el Demostrador 2 y se han seleccionado los elementos de seguridad adecuados, se realiza un primer planteamiento del posicionamiento de elementos junto con las acciones que ayuden a reducir los riesgos identificados. Este planteamiento relativo a la seguridad del entorno, se desarrollará en la fase de integración y pruebas (tarea 3.6.).

3.3.4 T3.4 Programación del robot

En el Demostrador 2, el robot de la zona de trabajo colaborativa de preparación de la bandeja debe realizar tareas de pick&place, con identificación de los objetos e inspección de forma autónoma para comprobar que el objeto o la acción realizada por el operario es correcta. El robot del sistema de distribución debe permitir estar esperando a que el operario realice una determinada acción sobre él para iniciar el proceso de pick&place, y posteriormente guiarle para indicar la posición de recogida en la cinta y dejarla en su plataforma. Por tanto, el primer robot colaborativo necesita un sistema de percepción visual y el segundo detectar fuerzas exteriores aplicadas en su efector final para ejecutar un proceso.

Para el desarrollo del programa, en primer lugar se han analizado individualmente diversas acciones con el UR5e y el Omron TM5-900 para determinar las funciones y los parámetros más adecuados

para el caso de uso seleccionado. A continuación se detalla cómo se realiza la programación con cada uno de los robots y detalles de las funciones analizadas.

3.3.4.1 UR5e

El UR5e está formado por el brazo robótico, la caja de control y un Tech Pendant con pantalla táctil de 12 pulgadas y soporte para colocarlo en la controladora. El brazo robótico es el robot colaborativo de tamaño medio de la gama Universal Robots. Tiene una carga útil de 5 kg y un radio de acción de 850mm. Dispone de sensor de fuerza y par integrado. Este sensor ultrasensible reacciona al más mínimo movimiento y dota al cobot del sentido del tacto.

UR5e - Montaje y configuración inicial del robot: tras realizar el conexionado, los pasos a seguir la primera vez que se enciende el robot son los de encender la caja de control para poder activar el brazo robótico. En su consola portátil, pulse el botón de encendido para activar la caja de control. Se espera mientras aparece en pantalla texto del sistema operativo subyacente, seguido por botones. Cuando aparezca una ventana de diálogo “No se puede proceder”, se selecciona “Ir a pantalla de inicialización” para acceder a la pantalla Inicializar. Para iniciar el robot, se presiona el botón ENCENDIDO con el LED verde para comenzar el proceso de inicialización. A continuación, el LED pasa a amarillo para indicar que el suministro eléctrico está activado y en estado Inactivo. Se presiona el botón INICIAR para liberar los frenos y el botón APAGADO con el LED rojo para apagar el brazo robótico.

UR5e - Pruebas con el gripper: para realizar la programación del robot, en primer lugar se han realizado unas pruebas básicas de funcionamiento con el robot UR5e y la pinza Hand-e (herramienta, que es compatible para ambos modelos de robot) realizando tareas de pick&place, paletizado, ensamblaje y manipulación de objetos de forma precisa. Es decir, se ha realizado la programación de las tareas unitarias que posteriormente se usarán en la validación final. Para comprobar que efectivamente esta programación se ha realizado correctamente, se realizan las pruebas unitarias de funcionamiento cuyo resultado es el que se muestra a continuación:

1. En primer lugar se realizaron pruebas de pick&place tal y como muestra la Figura 60. El robot coge la pieza de mecano sobre una pila de ruedas y la deja sobre el segundo montón de ruedas. Una vez acaba, la garra se posiciona sobre el montón de ruedas donde no se encuentra el mecano.

Pick&Place con piezas de mecano

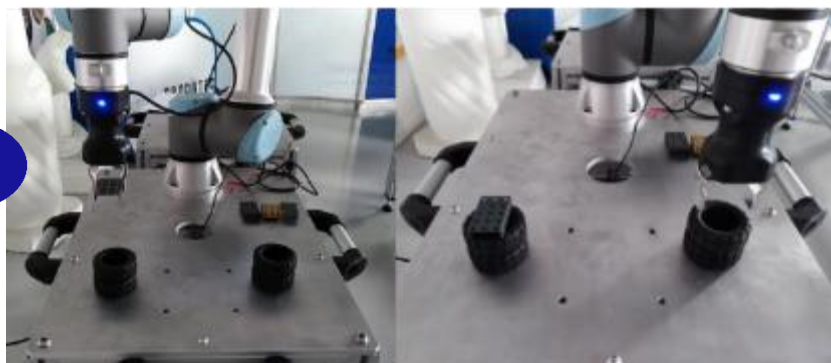


Figura 60. Pruebas de funcionamiento (I) – pick&place

2. La segunda prueba, es la de pick&place en bucle. El robot coge la pieza en una posición (1) y la lleva a una posición (2). Posteriormente, desde esta última posición (3), el robot la lleva a la posición inicial (4).
3. La tercera prueba simula un proceso de paletizado / despaletizado alterno. No se ha utilizado la plantilla de paletizado sino un script utilizando una matriz de puntos donde se han definido todos los puntos de paso uno a uno. El robot mueve una pila de piezas entre dos posiciones. En la Figura 61 se muestra cómo el robot mueve las piezas (4 ruedas) de un lado al otro, de una en una.

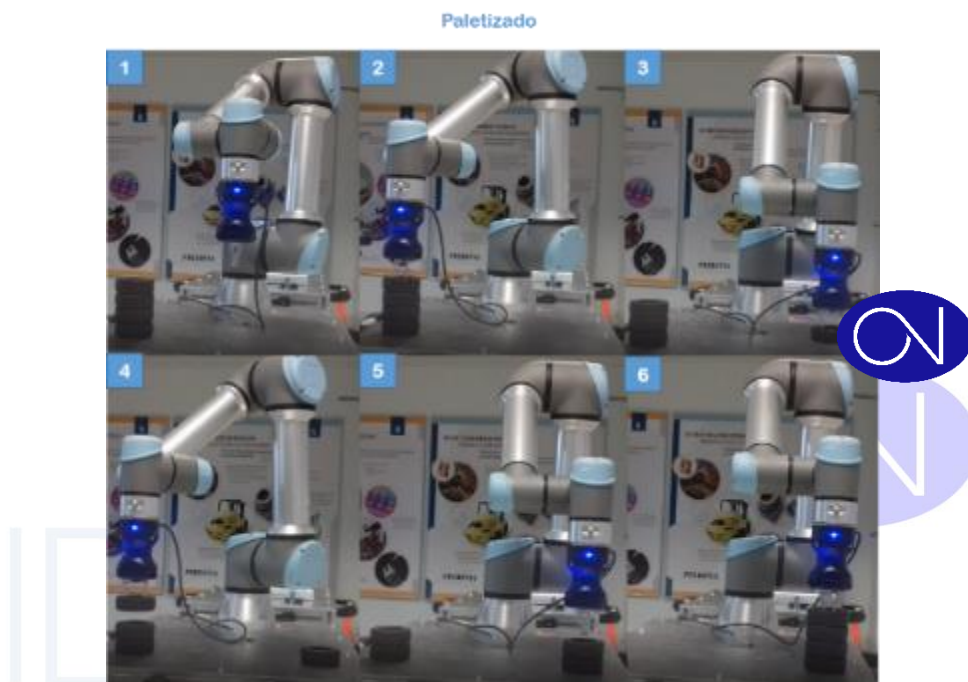


Figura 61. Pruebas de funcionamiento (III) – paletizado / despaletizado

4. La siguiente prueba es la de ensamblaje, en el que se simula un proceso sencillo de este tipo, para poder demostrar así la precisión del robot. Así, el robot coge una probeta y la introduce dentro de la columna de ruedas. Posteriormente, coge otra probeta y la introduce dentro de la anterior. La secuencia de este proceso se muestra en la Figura 62.

Ensamblaje

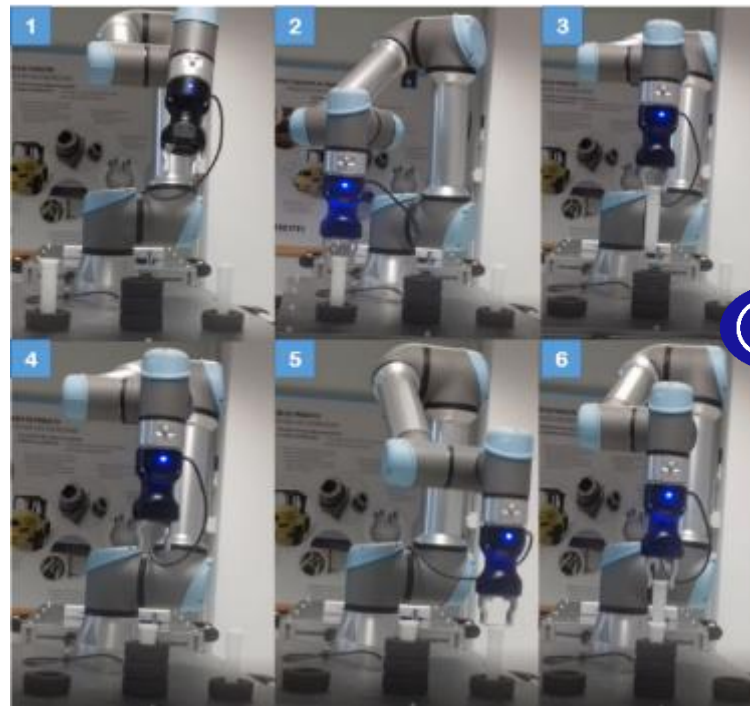


Figura 62. Pruebas de funcionamiento (IV) - ensamblaje

5. La siguiente prueba replica un proceso genérico para demostrar la capacidad del robot de manipular objetos de forma precisa. El robot coge una probeta (1), mezcla el contenido con movimientos laterales (2) y finalmente y deposita su contenido en el interior de otra probeta (3).
6. Por último, se realiza una prueba para demostrar las capacidades del robot en relación a la manipulación precisa de objetos. El robot sujeta un bolígrafo y es capaz de dibujar el logo de IDONIAL sobre una hoja de papel. La Figura 63 muestra la trayectoria seguida por el robot al realizar los trazos hasta conseguir el logo de IDONIAL, mostrado a la derecha de la imagen.

Logo



Figura 63. Pruebas de funcionamiento (VI) – manipulación precisa de objetos

7. Otra prueba de las realizadas ha sido la de que el operario puede interactuar en tiempo real con la consola de programación. Este programa alterna entre dos procesos de pick&place en base a la decisión del operario. con la consola del robot en tiempo real para realizar una serie de tareas.

UR5e - Pruebas sin el gripper: posteriormente se han realizado una serie de pruebas sin el gripper en relación a diversas opciones que ofrece el robot: cambio de plano y dirección de movimiento en función de los valores del sensor de fuerza.

1. La primera prueba sin gripper se basa en una de las opciones que ofrece el robot y está relacionada con la definición de los puntos de paso con respecto a las funciones de usuario. De este modo, se ha hecho un programa tal que se ha usado como referencia una función plano en un programa sencillo. Dicho plano coincidía con la tapa de la carpeta de las imágenes inferiores) y al modificar la función del plano, los puntos de paso del programa se ajustan automáticamente.
2. Una prueba adicional realizada ha sido la de aplicar el comando “Dirección”, que permite mover el robot en la dirección que se desee del espacio hasta que se dé una condición. En este caso, se ha programado el robot para que pare si se da alguna de estas condiciones: si se alcanza una determinada distancia o si se supera una determinada fuerza en la herramienta del robot. En la Figura 64 puede verse la dirección que sigue la herramienta del robot y como la mano del operario va a su encuentro. Una vez chocan, el robot se para.

Dirección hasta fuerza



Figura 64. Pruebas de funcionamiento (IX)

UR5e - Pruebas con plantilla: Universal Robots incorpora una amplia colección de software para simplificar la programación de aplicaciones. En estas pruebas sin garra se han utilizado las plantillas de conveyor tracking (seguimiento de cinta), paletizado y búsqueda de objetos en una dirección.

1. En este caso, se ha probado la capacidad del robot para adaptar sus movimientos en base al movimiento de una cinta transportadora, así se ve en la Figura 65.

Seguimiento de la cinta



Figura 65. Pruebas de funcionamiento (X)

2. La siguiente prueba ha sido la de realizar el paletizado con plantilla. El uso de la plantilla al programar aplicaciones de este tipo ha significado un gran ahorro de tiempo con respecto al flujo normal de trabajo donde hay que definir todos los puntos de paso, uno por uno. Se ha creado un sencillo programa que simula un proceso de paletizado de dos capas con separador, solo hace falta definir la matriz de piezas, los movimientos que se realizan sobre cada pieza, la zona donde se recogen nuevas piezas y la zona donde se recogen los separadores para colocar entre capas.
3. Otra de las pruebas realizada con plantilla ha sido la de búsqueda de piezas, permitiendo aprovechar el sensor de fuerza del robot para buscar objetos en una dirección, si no sabe la posición exacta. Un ejemplo de esta búsqueda se muestra a continuación.

UR5e - Pruebas con planos de seguridad: además, en relación a la tareas 3.3, se han realizado otras pruebas relacionadas con la seguridad del sistema en la que se han usado planos para limitar la zona de trabajo del robot.

En el primer caso se ha configurado un plano de seguridad a partir de un punto, que permite llevar al robot a una parada de protección si su TCP o su codo se acercan al límite. Es posible configurar una esfera alrededor del TCP que no pueda pasar el límite que impone el plano.

En el segundo caso se ha configurado un plano de seguridad que activa el modo reducido del robot cuando el codo o el TCP lo traspasan (Figura 66).

Plano de seguridad con modo reducido



Figura 66. Pruebas de funcionamiento (XIV)

3.3.4.2 Omron TM5-900

El Omron TM5-900 está formado por el brazo robótico, la caja controladora y un mando o stick (Figura 67, izqda.). El brazo robótico es el robot colaborativo de tamaño medio de la gama TM de Omron. Tiene una carga útil de 4 kg, un radio de acción de 900mm e incorpora una cámara a color de 5Mpx integrada. El Omron TM5-900 dispone del entorno de programación TMFlow de Omron, basado en diagramas de flujo. Además, está equipado con TMvision. Para su programación se puede conectar a la caja controladora una pantalla, teclado y ratón o bien utilizar un PC comunicándose mediante ethernet. Para la programación son de gran utilidad los botones de VISION, POINT, FREE y GRIPPER de su módulo final (Figura 67, dcha.). A continuación se resumen algunos de los pasos a realizar para la programación de una determinada acción.

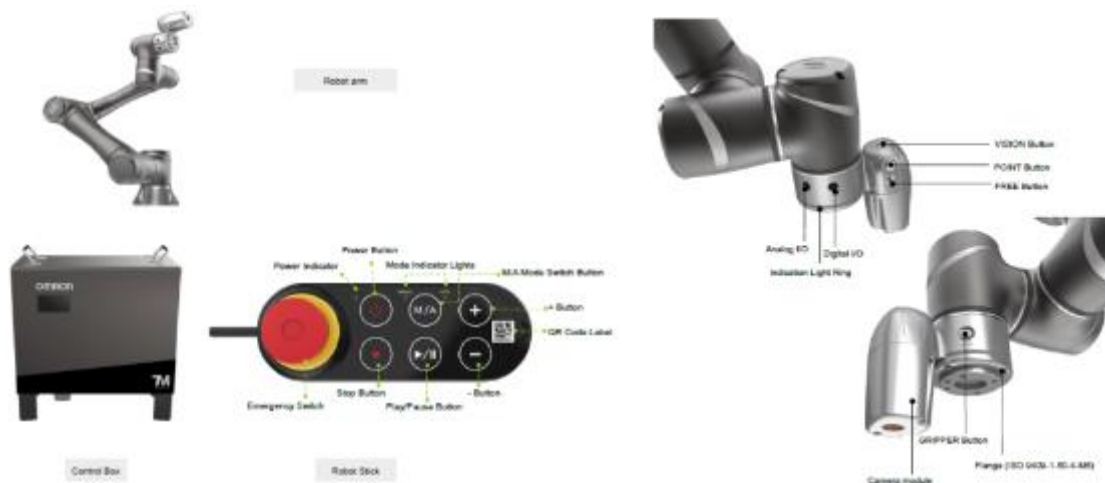


Figura 67. Visión general del sistema Omron TM5-900 (izqda.) y detalle de los componentes de los módulos finales del Omron TM5-900

TM5-900 - Montaje y configuración inicial del robot: tras realizar el conexionado, los pasos a seguir la **primera vez que se enciende el robot** son los detallados a continuación. En primer lugar, se pulsa seta de emergencia del stick del robot, se conecta la controladora y se pulsa el botón de encendido en el stick. Una vez arrancado, se libera la seta de emergencia. Tres LEDs del stick comienzan a parpadear, el anillo de LED del robot alterna entre rojo y azul claro (este comportamiento indica la entrada en modo de arranque seguro). Se pulsa el botón de movimiento libre en el último eslabón del robot, se espera a que se liberen los frenos y se lleva el robot a la posición de arranque deseada. Por último, se pulsa el botón de parada del stick un par de segundos, el robot entra en un proceso de calibración, todas las articulaciones se mueven ligeramente para calibrarse. Cuando el proceso de calibración termina correctamente el anillo de LED del robot pasa a un azul fijo.

El robot puede operarse en **modo colaborativo y no colaborativo**, cada uno de estos modos puede funcionar en **modo manual o automático**. En caso de que se seleccione la opción de **modo manual**, esto se sabe porque el anillo LED pasa a verde. En este modo puede editarse el proyecto, los ajustes del robot y del sistema mientras se modifica el programa. Es útil para programar y enseñar al robot, puede variarse la velocidad, modificar la secuencia de ejecución, entre otras. Si se programa una aplicación colaborativa el anillo LED parpadea alternando entre verde y morado. En caso de que se trabaje en **modo automático**, el anillo LED estaría en color azul, en este modo solo se puede ejecutar el programa tal y como se ha configurado. En caso de que el robot se encuentre en modo automático en una aplicación colaborativa el anillo LED parpadea alternando entre color azul y morado. Para pasar de modo automático a modo manual, se pulsa el botón M/A del stick. Para pasar de modo manual a modo automático mantener pulsado el botón M/A del stick hasta escuchar un pitido y a continuación presionar secuencialmente “+ - + + - “.

TM5-900 - Programación con diagrama de flujo: se puede realizar la programación por medio del uso del movimiento libre y el botón para guardar puntos, pero se puede programar todo mediante diagramas de flujo en TMflow. De hecho, se pueden combinar ambas formas, guardando los puntos con el botón y editando el tipo de punto en el diagrama.

Así, los pasos para generar una aplicación sencilla de pick&place son los detallados a continuación. En primer lugar, se mueve el cobot a una posición segura con ayuda del botón de movimiento libre, además de crear un nuevo proyecto y pulsar sobre el nodo de comienzo. En segundo lugar, se

arrastra un nodo de punto directamente debajo del nodo de comienzo. Al tener el nodo anterior seleccionado se unen automáticamente. Para cambiar el nombre del punto, se seleccionaría el nodo y se editaría dicho punto, seleccionando a continuación la opción de administrar punto y cambiar el nombre. A continuación, se movería el robot a la posición de pick con ayuda del botón de movimiento libre y se bajaría otro nodo de punto al flujograma, renombrando el punto a Pick. Se seleccionaría waypoint para dividir el movimiento en un movimiento eficiente hacia un punto de aproximación y un movimiento lineal hacia el punto de pick, partiendo del punto de aproximación. En porcentaje de altura, se seleccionaría 100% ya que este movimiento producirá dicho cambio. Arrastrando un nodo de movimiento bajo el punto anterior e introduciendo un valor de -200 mm en el campo de Z, se crea otro nodo de punto y se renombra como Place. De nuevo, se cambia el porcentaje de altura a 100%. Por último, se añade un nuevo nodo de movimiento y se vuelve a poner el valor de Z a -200mm y se conecta el segundo nodo de movimiento con el punto de Pick.

TM5-900 - Conectar un gripper: el TM5-900 está pensado para trabajar con multitud de herramientas. El caso más sencillo es su uso junto con una herramienta Plug + Play, que permite una rápida y sencilla conexión con el robot. Los pasos a seguir para añadir un gripper Plug + Play son: en primer lugar se debe formatear un USB en NTFS, llamarlo "TMROBOT" (mayúsculas sin espacios) y descargar la última versión disponible de los drivers que se puede encontrar en la web de Omron. Posteriormente, descomprimir y copiar el fichero TM_export en la raíz del USB e introducir dicho USB en la controladora. Posteriormente, debe dirigirse al menú principal, seleccionar sistema e Import/Export. Entonces, a través de Import y de TMComponent, se accede a importar. Seleccionando Component y todos los archivos que aparecen, se importan. Los nuevos nodos referentes al gripper están listos para usarse.

Con estos pasos aparecerán unos nuevos nodos disponibles que se usan de la siguiente forma: se crea un nuevo proyecto, se arrastra el nodo de configuración de gripper que ha aparecido y si el nodo no se conecta automáticamente, debe dirigirse a propiedades en el nodo de configuración, seleccionar la opción "Initialize_or_Not" y seleccionar variables. Posteriormente, se selecciona la ecuación y se cambia el valor false a true. De esta forma el nodo podrá inicializar el gripper. En el nodo de configuración de gripper pueden ajustarse los parámetros de inicialización del gripper. En la configuración de release puede modificarse la fuerza, velocidad y posición. Lo mismo puede configurarse para la configuración de grip. Finalmente, en los ajustes de puerto serie puede seleccionarse el número de puerto. Una vez en este punto, se arrastra un nodo de gripper abierto bajo el nodo de OK y se arrastra un nodo de gripper cerrado bajo el nodo anterior. Finalmente, se ejecuta el programa con ayuda del stick del robot.

Para usar el botón de gripper, deben dirigirse al menú y entrar en ajustes, seleccionando botón de gripper y componente personalizado. Se comprueba que se está usando el componente adecuado en la cogida y la dejada y se prueba a pulsar el botón de gripper del extremo del robot para comprobar si se desencadena la acción seleccionada en el gripper.

TM5-900 – Comunicación Modbus: para establecer comunicación con el PC es posible utilizar sockets TCP, comunicación serie RS-232, Modbus TCP y Modbus RTU. A continuación se muestra un ejemplo de cómo comunicarse entre un PC y el robot mediante Modbus TCP. Se supone que se quiere llevar al robot a una posición de home tras una parada de emergencia siguiendo una

trayectoria no lineal para evitar una colisión. Con Modbus se puede conocer la posición del TCP actual para poder definir la trayectoria a seguir en consecuencia.

Para ello, las acciones a realizar son las descritas a continuación. En TMflow, se debe dirigir al menú y seleccionar propiedades, Modbus y el tipo de comunicación Modbus, para este ejemplo TCP (empleando Ethernet con direccionamiento IP). Si se quiere limitar el número de dispositivos disponibles para la comunicación puede aplicarse un filtro de IP. En la documentación del robot puede comprobarse el código para leer la componente en Z del TCP con respecto a la base del robot (dirección en base decimal 7053 ~ 7054 del registro de entrada 04). Una vez habilitada la comunicación Modbus, volver a la ventana principal del proyecto y en la esquina superior derecha seleccionar el menú de funciones y en la ventana que se despliega, seleccionar ModbusDev y añadir Modbus TCP. El robot está habilitado por defecto con el nombre localhost y Modbus TCP. En los ajustes puede verse la dirección IP local y el puerto del robot. Para añadir señales Modbus, se selecciona la opción de añadir configuración. Para el ejemplo añadir un nombre de señal para la componente en Z (por ejemplo, tcp_z), slave ID a 1, tipo de señal como registro de entrada, dirección de comienzo a 7053, el tipo de dato en este caso es float y el orden de datos es AB CD (big endian). La tabla con los códigos para todas las opciones disponibles puede encontrarse en el menú de configuración de Modbus. A continuación, se inicializan tres variables de coma flotante:

- pos_z: inicializada a 0 para el valor que se acaba de solicitar.
- home_z: inicializada a 250 (altura en mm segura a la que debe llevarse el robot ande mandarlo a la posición de home con respecto a la base del robot).
- difference_z: diferencia entre la altura segura y la altura actual del TCP obtenida mediante Modbus inicializada a cero.

Para asignar valores a las variables recién creadas es necesario arrastrar un nodo de set al flujograma. Seleccionar la variable pos_z, seleccionar el operador y asignar la función modbus_read() en el localhost, seleccionar la señal definida previamente e insertar. Seleccionar la variable difference_z, seleccionar el operador e introducir la expresión $\text{var_pos_z} - \text{var_home_z}$. Por último, añadir un nodo de movimiento que moverá el TCP a lo largo del eje z empleando para ello la información contenida en la variable difference_z.

TM5-900 – Calibración automática: para la programación de tareas de pick&place identificando objetos, se siguieron primeramente los pasos de calibración automática indicados en T3.3. Para ello se coloca el patrón proporcionado por Omron en el plano de acción y se lanza el proceso de calibración automático. El robot va moviéndose, sacando fotos y procesando la imagen para determinar los parámetros de calibración automáticamente. En la Figura 68 se muestran los pasos en la parte inferior, y arriba se ve la detección del patrón (izqda.) así como el resultado (dcha.).



Figura 68. Prueba de calibración automática

TM5-900 – Pruebas preliminares de seguimiento de TM Landmark: Para la programación de tareas de pick&place identificando lugares de recogida o dejada, se siguieron los pasos indicados en T3.3 de pick&place con nodo de visión y un TM Landmark proporcionado por el fabricante. En la Figura 69 se muestra cómo el robot, en primer lugar, busca donde se encuentra el landmark para realizar el pick y posteriormente se orienta correctamente para acercarse al punto correspondiente en relación al landmark y no al punto absoluto definido previamente.

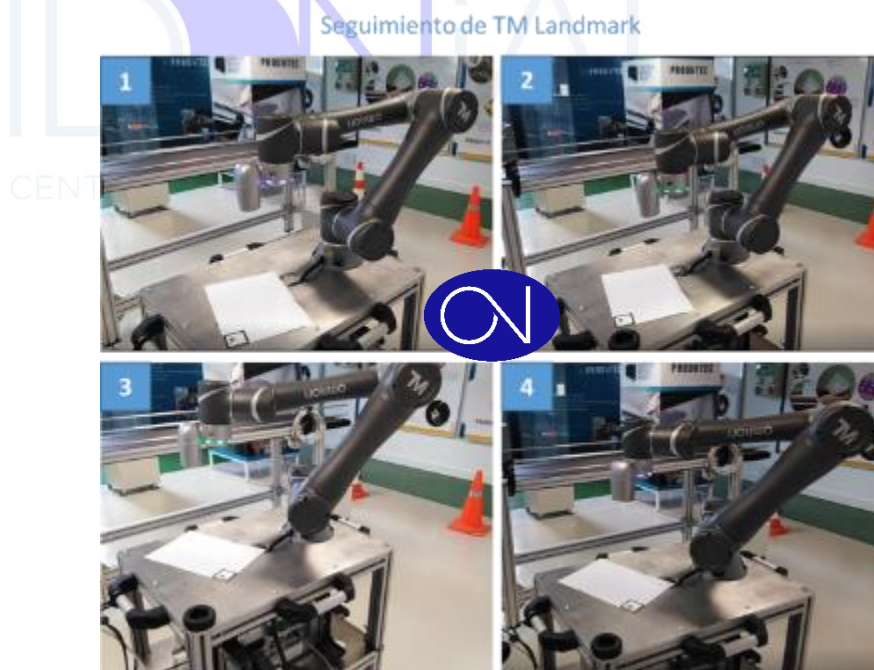


Figura 69. Prueba de pick&place sin garra identificando el TM Landmark para realizar el acercamiento al punto definido en función de la posición del marcador

De este modo, en esta tarea inicialmente se han desarrollado diversos programas con el UR5e y el Omron TM5-900 para comprobar diversas operaciones utilizando los sensores que incorporan. A la vista de las funcionalidades que dispone cada uno de los robots, se decide disponer del Omron

TM5-900 para la tarea de trabajo en la zona colaborativa de preparación de la bandeja usando la cámara y el UR5e para el sistema de recogida de la bandeja al final de la cinta utilizando su sensor de fuerza y el comando dirección. Posteriormente, se realizó la programación preliminar en ambos robots, siendo desarrollada ampliamente para su aplicación final en la tarea de validación (tarea 3.6.), donde se detallará el programa realizado.

3.3.5 T3.5 Interfaz de usuario

En primer lugar, hay que tener en cuenta que el demostrador tendrá dos interfaces de usuario en función del hardware asociado. Por un lado, se dispondrá de una interfaz adecuada para Realidad Aumentada (RA) y por otro lado, se utilizará una Raspberry. Esta última funcionará como unidad de control y enviará a las gafas de RA la información que tiene que mostrar. El proceso de visionado de RA se construye a partir de una serie de acciones o etapas esenciales que se suceden una detrás de otra en el tiempo: reconocimiento de objetos (el dispositivo captura la realidad física del usuario y reconoce los elementos que componen ese entorno, por ejemplo, códigos QR), seguimiento de objetos y representación de contenidos. De esta forma en las gafas se irán mostrando las instrucciones y, además, otra información de interés acerca del entorno gracias al reconocimiento de QR.

Por tanto, para que la interfaz sea operativa en el entorno del demostrador, lo primero que se debe realizar es el posicionamiento en el mismo de códigos QR con un determinado contenido almacenado en su matriz de puntos. La Figura 70 muestra los distintos lugares donde se posicionan los códigos QR para poder interactuar con la interfaz de Realidad Aumentada: en color azul se indican los códigos QR que identificarán el área de trabajo, en verde se identifica la mesa donde se encuentra el material y en amarillo, se posiciona el QR que identifica la zona colaborativa.



Figura 70. Demostrador 2 – interfaz de usuario: códigos QR (RA)

La interfaz realizada por RA, mostrará texto, un 3D del objeto con el que interactuar, acciones entre objetos e irá indicando con flechas las diferentes acciones a realizar relacionadas con depositar bandejas, recoger piezas o tapas, etc. Más adelante se detalla el contenido y detalle de la interfaz de RA y cómo esta se complementa con la interfaz de la Raspberry.

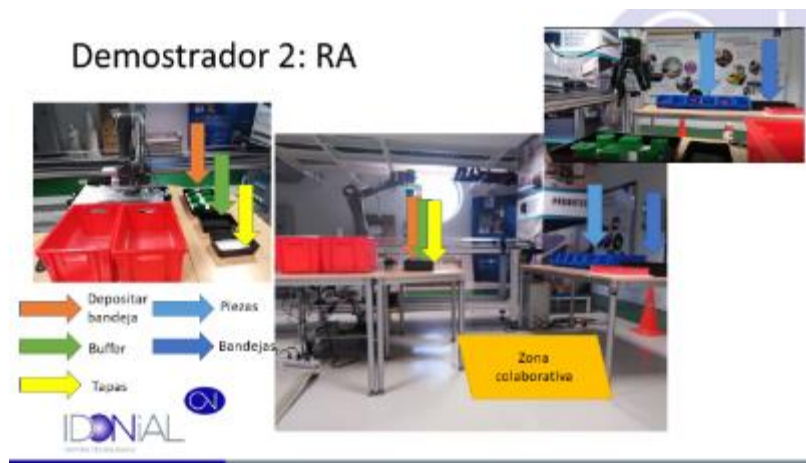


Figura 71. Demostrador 2 – interfaz de usuario: RA

La Figura 72 muestra el wireframe y mockup de la interfaz para Raspberry. Hay un encabezado para el título y el logo de la entidad, en la parte central, hay una zona reservada para una imagen en la zona izquierda y el derecha para texto (zona central) y para un botón en la zona inferior.

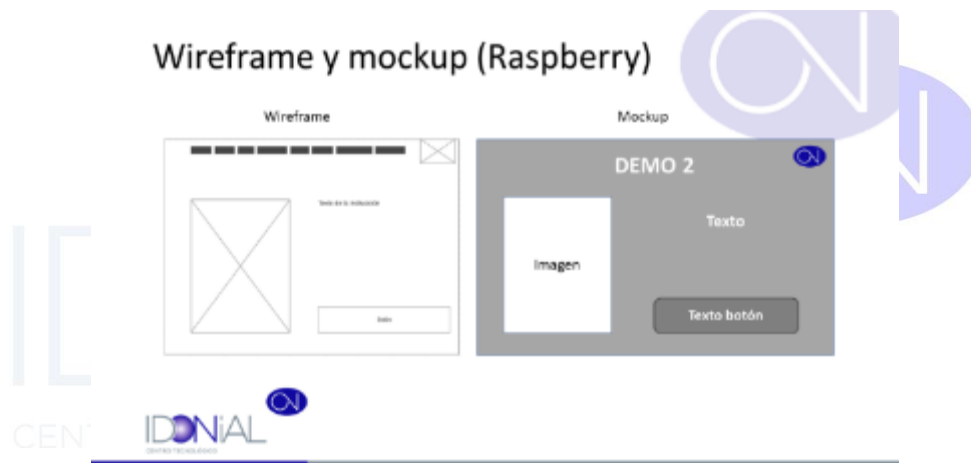


Figura 72. Demostrador 2 – interfaz de usuario: wireframe y mockup (raspberry)

Se ha definido un flowchart para un caso básico (Figura 73) en el que se parte de la pantalla de inicio; se indica al operario que coja de la bandeja y la deposite en la zona de trabajo marcada; se pide que coloque una pieza circular roja en una posición libre del buffer; se inspecciona la pieza y en función de si está OK o NOK, se sigue el proceso o se vuelve atrás hasta que esté OK; una vez que se tiene la bandeja completa adecuadamente, se introducen las tapas; se inspeccionan las tapas y si están OK, se sigue el proceso, sino se repite; si todo está OK, el proceso continúa posicionando el robot la bandeja en la cinta; una vez depositada, finaliza el proceso.

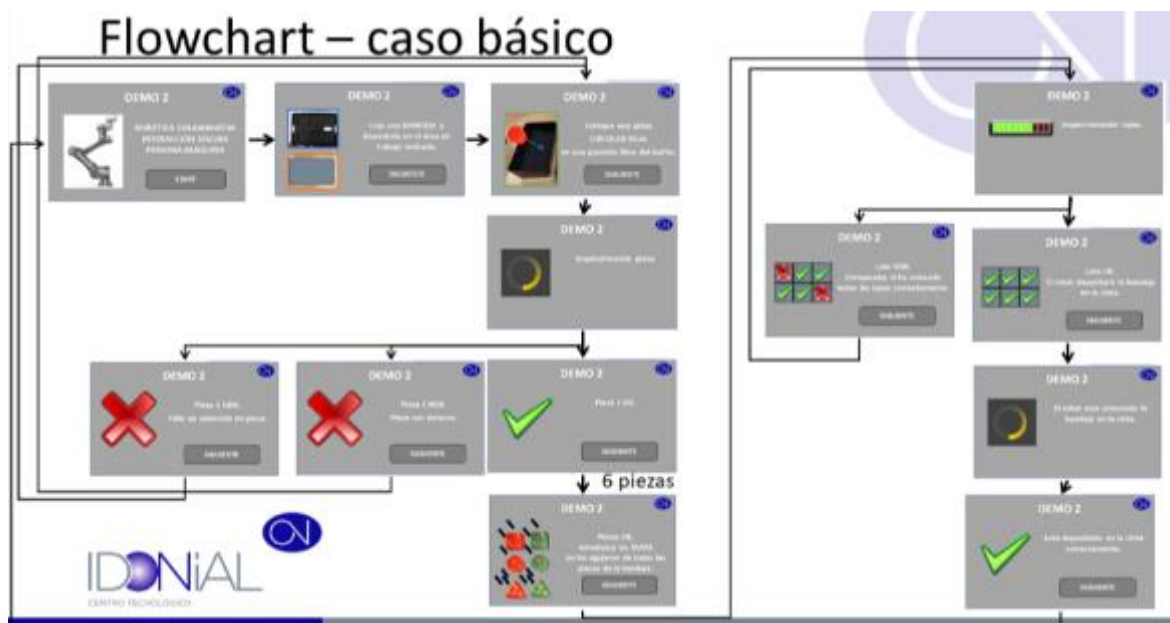


Figura 73. Demostrador 2 – interfaz de usuario: flowchart para caso básico

A continuación se muestran cada una de las pantallas de la interfaz de usuario, comparando la interfaz del Demostrador 1 de gemelo digital realizada en Realidad Virtual (RV), con la interfaz realizada en este Demostrador 2 para RA y Raspberry. La Figura 74 muestra el detalle de la pantalla de inicio del demostrador, comparando la estructura de la misma para RA y Raspberry con la versión en RV. Una vez se pulsara al botón de “Start” se pasaría a la siguiente pantalla mostrada en la Figura 75.

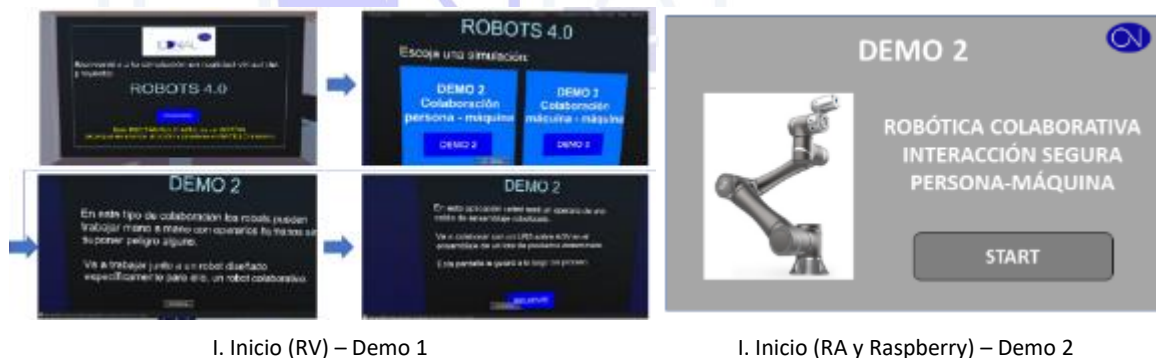
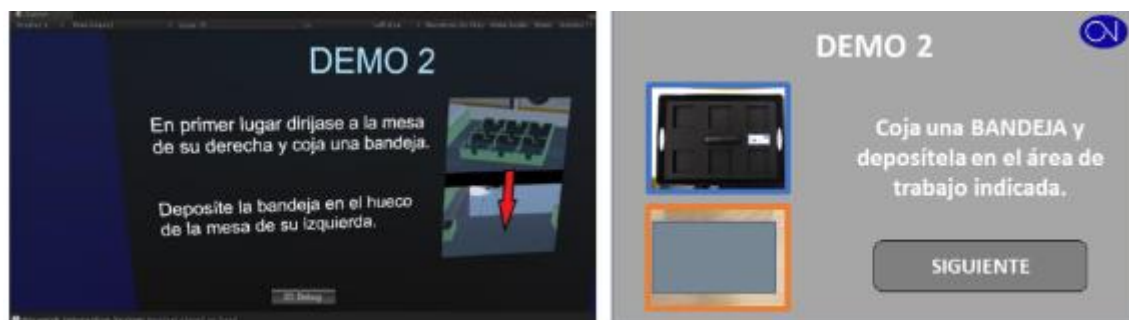


Figura 74. Demostrador 2 – interfaz de usuario: pantalla de inicio (RV vs RA o Raspberry)

La Figura 75 detalla la siguiente pantalla, acerca de la colocación de la bandeja. Se ve la comparativa entre la interfaz de realidad virtual (izqda.) y en la derecha, la de la Raspberry donde se indica en color azul, que se debe coger una bandeja y dejarla en la zona de trabajo marcada en naranja.



II. Colocación bandeja (RV)- Demo 1

II. Colocación bandeja (Raspberry) – Demo 2

Figura 75. Demostrador 2 – interfaz de usuario: colocación bandeja (RV vs Raspberry)

En la Figura 76 se muestra la misma pantalla de colocación de bandeja pero en la interfaz de Realidad Aumentada. En este caso, también se ve una comparativa de esta interfaz con la de Realidad Virtual definida en el demostrador de Gemelo Digital.



II. Colocación bandeja (RV)- Demo 1

II. Colocación bandeja (RA) – Demo 2

Figura 76. Demostrador 2 – interfaz de usuario: colocación bandeja (RV vs RA)

La Figura 77 muestra la interfaz de usuario donde se ve por un lado, la versión de realidad virtual (izqda.) de la pantalla de introducir en el buffer las piezas frente a la imagen mostrada en la Raspberry (dcha.). Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.



III. Introducir en buffer (RV) – Demo 1

III. Introducir en buffer (Raspberry) – Demo 2

Figura 77. Demostrador 2 – interfaz de usuario: introducir en buffer (RV vs RA)

La Figura 78 muestra la inspección en buffer y colocación. En este caso, no hay un equivalente en la interfaz de RV por eso, no se muestra una comparativa. A continuación se detalla cada una de las pantallas de la interfaz: (1) Pantalla para cuando hay piezas en el buffer y las inspecciona; (2) Se mostrará este resultado tras procesar la pieza y mientras tanto, el robot mueve la pieza del buffer a su posición en la bandeja; (3) Se mostrará este resultado tras procesar la pieza y mientras tanto, el robot se mueve para desechar la pieza. Si una pieza es errónea, se almacena en su posición y al acabar la bandeja se vuelve a las piezas NOK; (4) Se mostrará este resultado tras procesar la pieza. El robot se mueve mientras tanto para desechar la pieza. Si es una pieza errónea se almacena su posición, y al acabar la bandeja se vuelve a las piezas NOK. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.



Figura 78. Demostrador 2 – interfaz de usuario: introducir en buffer (Raspberry)

La Figura 79 muestra la pantalla correspondiente a introducir las tapas, indicando los agujeros en los que los deben posicionar. Además, se ve la comparativa con la interfaz de RV. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.

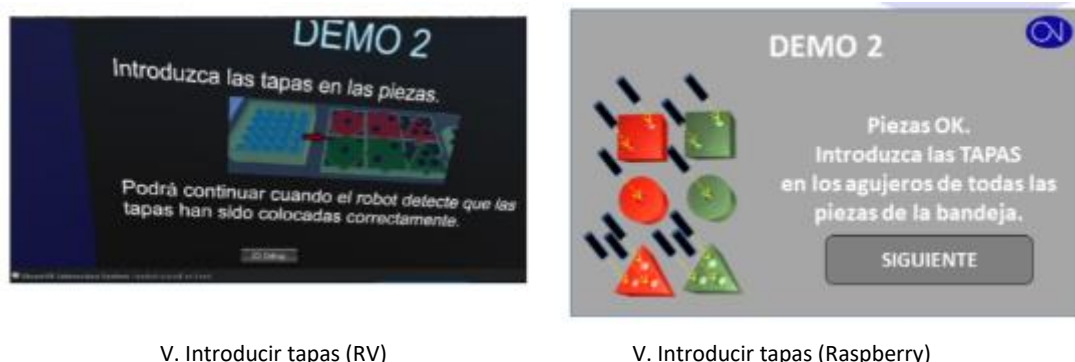


Figura 79. Demostrador 2 – interfaz de usuario: introducir tapas (RV vs. Raspberry)

La Figura 81 muestra la pantalla de inspección de tapas, que aparece mientras se realiza dicho proceso. En este caso, no hay equivalente en RV y por ello no aparece una comparativa. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.



VI. Inspección de tapas (1)

Figura 80. Demostrador 2 – interfaz de usuario: inspección de tapas (Raspberry)

En la siguiente pantalla de inspección de tapas, se ve una imagen u otra en función del resultado que dé la inspección de las piezas. Así pues, si el resultado de la inspección indica que el lote completo está OK, el robot depositará la bandeja en la cinta y se verá la imagen mostrada en la Figura 81. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.

En este caso, se ve que en el demostrador de RV, la pantalla tenía su equivalente.



V. Inspección de tapas (RV)



V. Inspección de tapas (Raspberry) (2)

Figura 81. Demostrador 2 – interfaz de usuario: inspección de tapas (RV vs. Raspberry)

Si alguna de las piezas no estuviera OK, la interfaz indicaría “Lote NOK” y pediría al operario que comprobara si ha colocado las tapas correctamente, tal y como se muestra en la Figura 82. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.



VI. Inspección de tapas (3)

Figura 82. Demostrador 2 – interfaz de usuario: Inspección de tapas (Raspberry)

Por último, cuando se tenga el “Lote OK” y el robot esté depositando la bandeja en la cinta, la interfaz es como se muestra en la Figura 83. No hay una comparativa con RV pues en el Demostrador 1, esta pantalla no se consideraba. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.



VII. Depositar bandeja en cinta (1)

Figura 83. Demostrador 2 – interfaz de usuario: Depositar bandeja en cinta (Raspberry)

Una vez deposite la bandeja en la cinta, la interfaz da el mensaje mostrado en la Figura 84 y posteriormente, volverá a mostrarse la pantalla inicial. No hay comparativa con RV porque en el Demostrador 1, esta pantalla no existía. Para el caso de la interfaz en Realidad Aumentada, sería similar a la Raspberry pero viéndose el escenario real, de modo similar a cómo se mostró para la colocación de la bandeja.



VII. Depositar bandeja en cinta (2)

Figura 84. Demostrador 2 – interfaz de usuario: Depositar bandeja en cinta (Raspberry)

Esta propuesta de interfaz es preliminar y podrá sufrir modificaciones durante la fase de integración, pruebas y validación, adaptándose al entorno real y a la propia experiencia de usuario durante el proceso.

3.3.6 T3.6. Integración, pruebas y validación

Durante esta tarea se han integrado todos los elementos que conformarán el Demostrador 2. Se ha dispuesto por un lado del UR5e con la pinza Hand-e y por otro, del Omron TM5-900 con cámara integrada, la pinza robótica Robotiq 2F-140 y el SICK scanner MicroScan3 Core, dispositivos de inversión propia. El resto de componentes necesarios para la integración global han sido adquiridos: diversos elementos para conformar el setup, el sistema de comunicaciones y la unidad de control; hardware asociado a la cinta y hardware asociado al robot. Según se iban recibiendo se realiza su instalación, conformando así el laboratorio: cinta, robots y setup general (Figura 85).



Figura 85. Setup con todos los elementos instalados (II)

En cuanto a los objetos con los que interaccionar durante el Demostrador 2 cabe destacar los siguientes pasos realizados para su preparación previa a la instalación:

- **Piezas:** se ha realizado el diseño de las piezas acorde al tamaño de la pinza con la que se van a coger y posteriormente se han hecho por fabricación aditiva, AM, y se han pintado con unos sprays rojo y verde para tener el acabado buscado. En cuanto al diseño cabe destacar que se suavizaron los bordes de los agujeros para facilitar la inserción de las tapas y que la parte inferior, se realizó con nervios, para aligerar su peso aportando robustez a la pieza. Para confirmar que las piezas fabricadas eran adecuadas, según se tuvieron preparadas, se realizaron unas pruebas unitarias básicas de pick&place con detección. Se validó que tanto su acabado, color, tamaño y peso era adecuado para la aplicación, no siendo necesario realizar modificaciones en su diseño ni adaptaciones en la pinza. Además, se realizó en PLA una pieza con error para validar la detección de piezas erróneas. Dicha pieza se imprimió en una impresora FDM en las instalaciones de IDONIAL y posteriormente se pintó usando el mismo spray que para las fabricadas por AM.
- **Tapas:** se cortaron unas varillas de diámetro 10,6mm en 40 trozos de 30 mm para disponer de tapas suficientes para las piezas fabricadas.
- **Bandejas de piezas:** se adaptaron las bandejas compradas (tamaño acorde a la cinta y a las piezas a colocar) incorporando espuma y un asa para que el robot pueda coger la bandeja y depositarla en la cinta (Figura 86). Se realizaron cortes a la espuma para dotar a la de 6 posiciones para colocar las piezas. Además, se añadió el logo de IDONIAL para la detección de la bandeja, pudiendo colocar también el TM landmark para pruebas.
- **Buffer:** se adaptó la bandeja comprada (tamaño acorde a las piezas a colocar) incorporando espuma cortada para disponer de dos posiciones para colocar las piezas. Además, se añadió el logo de IDONIAL para la detección del buffer, pudiendo colocar también el TM landmark para pruebas.
- **Bandeja para las tapas:** se adaptó la bandeja comprada incorporando una base de Foamboard blanco.



Figura 86. Bandejas de piezas con 6 posiciones

Una vez preparados e instalados los elementos que conforman el Demostrador 2, son necesarios los siguientes pasos para su integración, realizando en primer lugar pruebas unitarias y posteriormente, en conjunto:

- **Cinta:** integración de baliza y sensores de presencia con la cinta, configurando velocidad de arranque y parada.
- **Seguridad:** integración del SICK escáner con el robot y configuración de parámetros de seguridad de los robots.
- **Sistema de comunicaciones, unidad de control y robot Omron TM5-900:** configuración del sistema de comunicaciones.
- **Integración de los robots tras la instalación y configuración de todos los componentes.**
- **Pruebas y validación finales del conjunto.**

3.3.6.1 Cinta

En el caso de la cinta, se realizó el armario pertinente para su control, se integró la baliza, la seta de emergencia y los sensores de presencia, configurando velocidad de arranque y parada al detectar un objeto. El armario dispone en su exterior de un interruptor para puesta en marcha, otro para elegir el sentido de giro de la cinta, un pulsador de rearme y dos leds luminosos, así como un cable extendido con un potenciómetro para controlar la velocidad de la cinta. En cuanto al sistema de control de la cinta, se muestran todas las entradas que recibe el controlador sobre el estado de la cinta transportadora y las señales de control:

- El PLC recibe las señales de posición de cada detector y, en función de las mismas, envía la orden de parada de la cinta. Una vez transcurrido el tiempo programado éste se encarga de volver a enviar la señal de activación del motor. La salida de activación del motor Q0.0 del controlador está condicionada por el sistema de emergencia, evitando el movimiento de la cinta si el usuario pulsa la seta de emergencia.
- La dirección de movimiento de la cinta es controlada a través de la salida de la CPU Q0.1 (en nivel bajo la cinta mantiene su dirección pre-configurada y en estado alto se considera cambio de sentido).
- La velocidad de la cinta es controlada a través de la entrada analógica de control 10VDC, que recibe la señal directamente del potenciómetro ubicado en el exterior del armario eléctrico.
- El variador es el encargado de regular la red trifásica acorde a la potencia necesaria por el motor. De esta forma, acciona unos relés de estado acorde a la activación de la cinta.

- La baliza de estado recibe las señales de estado directamente del variador. El verde fue asignado al pin correspondiente a la activación del motor y por el contrario el rojo se conectó al pin de parada del motor.

Considerando esta arquitectura de control, se ha programado el PLC para disponer de las siguientes funcionalidades asociadas a la cinta:

- La cinta se pone en movimiento a los 3 segundos de detectar un objeto el sensor de presencia central, para dar tiempo al brazo robótico a retirarse. Pasada esa temporización se activa la marca "Accionamiento_Marcha". Dicha marca, a su vez, activa la marca "Arranque_Cinta", que está directamente ligada a la salida "Cinta_Start" y pone la cinta en movimiento. Una vez que el objeto llega a cualquiera de los señores laterales la cinta se detiene ya que se activa la marca "Accionamiento_Paro" que es la que resetea la marca "Arranque_Cinta".
- La velocidad de la cinta es configurable mediante un potenciómetro.
- La rampa de deceleración del driver de la cinta viene configurada por defecto a 10 segundos y se ha reducido a 5 segundos de tal forma que, al ser detectada la bandeja por los sensores de presencia de los extremos, la cinta se para y el objeto queda posicionado en el extremo de la cinta sin caerse sea cual sea su velocidad. El sensor fotoeléctrico permite una detección estable independientemente del color y reflectancia.
- El sentido de giro de la cinta es preestablecido mediante un selector colocado en el armario eléctrico. Se ha escogido que la bandeja avance hacia la posición del UR5e pero es posible cambiar el sentido en caso de necesitarse.
- La luz verde de la baliza de señalización se activa al ponerse la cinta en movimiento, por lo que mientras está parada, la luz roja permanece encendida. De este modo el operario sabrá si la cinta se encuentra en tensión porque aparecerá iluminada en rojo o si está en movimiento porque se iluminará en verde.
- Como medida de protección se ha colocado una seta de emergencia que permite detener el funcionamiento de la cinta en caso de una situación de peligro. Al accionar la seta de emergencia, se para la cinta y es necesario pulsar el rearme de la seta y el pulsador del armario para que se ponga otra vez en funcionamiento.
 - Al estar la seta S1 inactiva, permite el paso de corriente. Si es presionado el pulsador de rearme S2, la corriente produce la excitación de la bobina de un relé, el cual mantendrá cerrado el contacto K1, permitiendo el funcionamiento de la cinta. En caso de que la seta de emergencia sea presionada, corta el paso de corriente, provocando la parada.

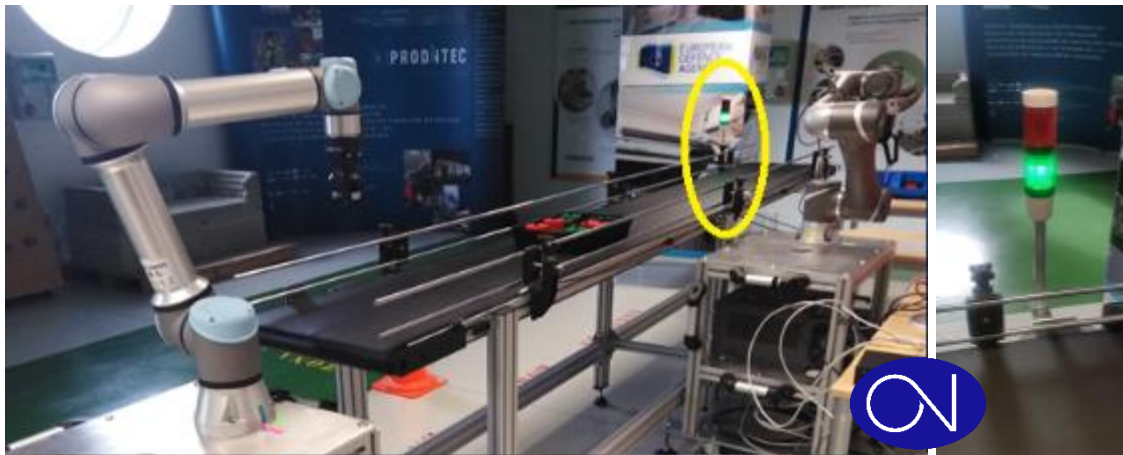


Figura 87. Ejemplo de baliza con luz verde estando la cinta en movimiento (remarcada en amarillo)



Figura 88. Ejemplo de baliza con luz roja estando la cinta parada (remarcada en amarillo)

3.3.6.2 Sistemas de seguridad

Como se indicó en la tarea 3.3 en el Demostrador 2 se han aplicado diversas medidas de seguridad: setup (aviso en el suelo sobre la zona de trabajo e integración de un escáner SICK para protección horizontal de zona de peligro), configuración adecuada de los robots y del sistema general, así como información adecuada a los operadores mediante unas indicaciones previas e información del proceso por medio de la interfaz.

Respecto al escáner **SICK MicroScan3 Core**, se colocó en la plataforma del Omron TM5-900 a una altura por debajo de las rodillas del operador y se conectó a su controladora para monitorizar la zona de trabajo colaborativo, realizando parada de protección o reduciendo su velocidad en función de dónde se encuentre el operario. Para la conexión se ha tenido en consideración que el robot cuenta con 6 entradas redundantes de seguridad.

Los campos de protección se configuraron con el propio software del escáner (Figura 90). De esta forma, el escáner utiliza dos campos de protección relacionados con la velocidad del robot: un área de advertencia (amarillo) y un área de protección (rojo). En la Figura 89 se puede apreciar el espacio que cubre cada área en el setup, siendo la zona de protección aquella que corresponde al espacio

más cercano al robot (bajo la mesa y la zona entre la mesa de ensamble y la cinta) mientras que la zona de advertencia corresponde a la zona de trabajo colaborativo.



Figura 89. Campos de protección en el setup

De esta forma, si una persona entra en el área de advertencia durante el proceso en curso, el robot reduce su velocidad entrando en modo colaborativo (Figura 90, centro). Dado que en ese momento el robot se está moviendo con mayor lentitud, tendría un tiempo de parada más corto en el caso de parada de protección o emergencia, por lo que el sistema de seguridad activa el campo de protección que tiene un área más pequeña. Si se entra en esta área, el robot realiza una parada de protección (Figura 90, dcha.). Si la persona abandona esta zona, el robot reanuda su movimiento automáticamente, al principio a una velocidad reducida y monitorizada por el campo de protección, de modo que luego puede reanudar la velocidad de trabajo original con la protección que brinda el área de advertencia una vez que salga de las zonas monitorizadas (Figura 90, izqda.).



Figura 90. Configuración del escáner SICK (abajo) y visualización de su salida (arriba) en función de si hay algo en el campo de monitorización de advertencia (amarillo, centro) o protección (rojo, dcha.)

En cuanto a los **robots**, en el caso del **TM5-900** se realizó la siguiente configuración de seguridad:

- **Hardware:** existen dos mecanismos de seguridad disparados por señales externas. Se utilizará la seta del mando de control como parada de emergencia y una parada de protección en función de la señal proporcionada por el SICK escáner.

- **Parada de emergencia:** puede accionarse con la seta del mando de control o mediante la entrada redundante de seguridad de parada de emergencia. Cuando se acciona se activan los frenos del robot y se desactiva la alimentación de los motores una vez detenidos (parada de categoría 1 acorde a ISO 60204-1).
- **Parada de protección:** puede accionarse con la entrada redundante de seguridad habilitada a tal fin. Cuando se activa, el robot se detiene pero la alimentación se mantiene activa (parada de categoría 2 acorde a ISO 60204-1), en ese momento comienza un proceso de monitorización continua que comprueba que no se produzcan movimientos no intencionados, lo que desencadenaría una parada de categoría 0 (se desactiva la alimentación de forma inmediata).
- **Software:** mecanismos de seguridad modificables con ayuda del entorno de programación. Cabe destacar: ajustes de seguridad y espacio de operación.

3.3.6.3 Sistema de comunicaciones entre unidad de control y robot Omron TM5-900

Una vez configurada la seguridad del sistema, el siguiente paso en la integración ha sido la configuración del sistema de comunicaciones (switch y router), de la Raspberry y del robot. Durante la ejecución de las tareas del Demostrador 2, el robot se comunicará con una Raspberry para conocer su siguiente movimiento en base a la información que recoge del entorno. Las tareas serán mostradas tanto en la propia Raspberry como en unas gafas de RA Hololens como se explicó en la tarea anterior.

La controladora del TM5-900 puede funcionar como cliente o como servidor, permitiendo acceder a información específica del robot desde el propio programa o desde un dispositivo externo, asimismo también puede modificarse la información en él contenida. Por este motivo, para implementar la arquitectura de comunicación, primeramente es necesario establecer quién será el cliente / servidor entre el robot y la Raspberry.

Para elegir quién será el cliente / servidor, una vez configurados e integrados todos los componentes de comunicación, se han realizado varios **programas para validar la comunicación Modbus TCP**. A la vista de las pruebas se decide **disponer de la Raspberry como servidor, Modbus TCP Slave, y el robot como cliente** (siempre será el robot el que decida cuándo quiere recibir o enviar información), Modbus TCP Master. Esta elección permitirá de cara a futuro crear un esclavo para cada robot de la instalación o compartir un mismo servidor entre varios robots, pudiendo leer y escribir en un espacio común.

Por tanto, la Raspberry será la unidad de control, actuando como servidor del robot, indicando qué bandeja se debe realizar y en base a la información que recoge del entorno indicará el siguiente movimiento a realizar por parte del usuario o el robot. El **programa desarrollado en la Raspberry** primeramente se desarrolló como aplicación de consola, de gran utilidad durante las pruebas, y posteriormente se desarrolló la interfaz que se mostrará posteriormente. Este programa es multiplataforma, ya que aunque corre en Ubuntu Core, ha sido probado tanto en Windows como en Ubuntu durante las pruebas de validación de comunicaciones con el robot (utilizando un portátil para ello).

En paralelo al desarrollo de la programación de la Raspberry, se ha creado un **programa en el robot Omron TM5-900** para validar la integración. Este programa en TMFlow permite validar la ejecución del programa desarrollado en C#, el cual actualiza continuamente el valor de una variable que lee

de un registro de Modbus. Según el valor leído (que se actualiza modificando el registro desde la propia Raspberry), el robot sabe qué pieza debe coger y así permite mover las piezas entre el buffer de dos posiciones y la bandeja de seis posiciones, así como desechar las piezas mal colocadas. Se configuran una serie de bobinas/registros de solo lectura o lectura/escritura según la acción a realizar. Cuando el robot realiza un escaneo del buffer de dos posiciones tiene que escribir la información extraída en el servidor. Para ello se ha decidido emplear un registro de lectura y escritura para ambas posiciones, el cual se descompone en dos bytes (uno relacionado con una de las posiciones del buffer de dos posiciones y el otro con la otra posición). El robot también debe leer del servidor la posición de la que debe coger una pieza del buffer de dos posiciones y la posición donde debe dejarla (en la bandeja de 6 posiciones o en las bandejas para piezas erróneas/defectuosas).

3.3.6.4 Integración del robot Omron TM5-900

Se ha integrado el robot Omron TM5-900 teniendo en consideración la arquitectura planteada. Con este robot el operario debe preparar el lote que se le indique por la interfaz. Para ello debe colocar la bandeja en la zona de trabajo, seleccionar la pieza indicada y colocarla en el buffer. El robot inspeccionará que la pieza es OK y que corresponde a la indicada por el sistema. Si es correcta, la cogerá y la colocará en la bandeja. En caso de que no sea correcta, la moverá a la caja correspondiente (incorrecta o fallo de fabricación). Una vez posicionadas todas las piezas, el operario colocará las tapas y el robot las irá inspeccionando para confirmar que están todas las tapas colocadas. Si todo es correcto, el robot cogerá la bandeja y la posicionará en la cinta. Durante el proceso se mostrarán mensajes por la interfaz al usuario para facilitar toda la operación y guiarle durante las etapas. Por otra parte, si el usuario se encuentra en la zona de advertencia el robot trabajará en modo colaborativo. En caso de que el usuario se acerque al robot por el espacio disponible entre la mesa de ensamblaje y la cinta, el robot se parará ya que la persona se encuentra en la zona de protección. Esta medida está fundamentalmente pensada para cuando el robot traslada la bandeja a la cinta para asegurarse que durante ese movimiento no haya nada en la trayectoria.

Buscando la máxima flexibilidad y robustez en el proceso de interacción humano – máquina, se ha desarrollado un programa que proporciona al usuario cierta libertad para colocar la bandeja, las piezas y que sea robusto frente a cambios de las posiciones del buffer o de la bandeja durante el proceso. Para ello se han utilizado diversas funcionalidades del sistema de visión integrado y se han dotado a la bandeja y al buffer de unos marcadores o Landmark con el logo de IDONIAL. De forma general, el diagrama de flujo del programa se muestra en la Figura 91 y se puede resumir en los siguientes bloques, explicando posteriormente los subprogramas implementados:

1. Configuración inicial: en primer lugar, el programa comprueba que el robot se encuentra en la posición de HOME, en caso contrario el robot se mueve a ella. Tras dirigirse a la posición de HOME y realizar una configuración inicial de la pinza (subprograma gripperConf), el robot establecerá comunicación con el servidor, quedando así a la espera de que se le solicite realizar una operación. Para ello realiza un switch sobre el valor leído de un determinado registro del servidor: escanear el buffer, comprobar las tapas o default (el robot vuelve a leer el registro del servidor, entrando en bucle hasta obtener una acción válida).

2. Escanear buffer: en el primer caso, el robot se desplaza a una posición sobre la mesa de ensamblaje y realizará la adquisición de una imagen para detectar dónde se encuentra el buffer buscando un Landmark del logo de IDONIAL ubicado sobre él. Una vez detectado el logo, se posicionará sobre el buffer en referencia a dicho marcador y analizará las piezas que hay en el buffer, escribiendo la información en el servidor al terminar. Tras el procesado, accederá al servidor de nuevo y leerá la posición de recogida (en caso de detectar pieza en el buffer) y de dejada (sobre la bandeja o sobre los contenedores de piezas incorrectas o mal seleccionadas). Cabe destacar los siguientes pasos del programa:
 - a. Esperar señal de la unidad de control para realizar SCAN: el robot levanta un flag para informar al servidor de que ha comenzado la comunicación. El robot entra en un bucle del que saldrá cuando el flag de escaneo se ponga a TRUE y en ese momento se llamará al subprograma scanBuffer.
 - b. Escanear buffer: el robot indica al servidor el estado del buffer. Se resetean las variables auxiliares del escaneo anterior y se actualiza su valor con la información actual.
 - c. Escribir resultado en el servidor: la información anterior se pasa a DWord y se escribe en el registro de lectura/escritura habilitado a tal fin.
 - d. Actualizar posición de recogida y dejada: el robot espera un pequeño lapso de tiempo para asegurar que el servidor actualiza y lee correctamente las posiciones de recogida y dejadas. Posteriormente el robot lee del servidor las posiciones de recogida y de dejada. Si la posición de recogida no se corresponde con una de las dos posiciones del buffer significa que no hay piezas en el buffer y el programa vuelve al punto donde se escanea el buffer.
3. Pick&Place de la pieza: una vez dispone de la posición y orientación de la pieza en el buffer, el robot la coge (subprograma pickSeq) y la deposita en el lugar que corresponde (subprograma placeSeq). Se distinguen estos dos pasos:
 - a. Recoger pieza e indicárselo al servidor: el robot recoge la pieza de la posición indicada y le comunica al servidor que tiene la pieza.
 - b. Posicionarse sobre posición de dejada: el robot se dirige a la posición de dejada. Si tiene que depositar la pieza sobre la bandeja entonces realizará un proceso de visión para localizar el Landmark con el logo de IDONIAL que se encuentra sobre la bandeja y posicionar la pieza siempre en referencia a dicho marcador. En caso de que tenga que desechar la pieza el robot se mueve a una posición fija (las cajas de piezas incorrectas están colocados en una posición prefijada).
 - c. Depositar pieza e indicárselo al servidor: una vez colocado sobre el lugar de dejada el robot soltará la pieza y se lo indicará al servidor para que pueda actuar en consecuencia.
4. Tapas listas: en caso de que deba comprobar tapas, buscará la bandeja (con ayuda del Landmark ubicado sobre ella), y se posicionará sobre las dos posiciones de un extremo quedando a la espera de que las tapas se introduzcan de forma correcta. El proceso se repite para las dos posiciones centrales y finalmente para las dos posiciones restantes (subprograma checkCovers).

5. Pick&place bandeja: una vez detecta que todas las tapas se han introducido de manera correcta vuelve a referenciar la bandeja con ayuda del Landmark para poder coger la bandeja por el asa de forma precisa y depositarla sobre la cinta transportadora (subprograma PickPlaceTray).

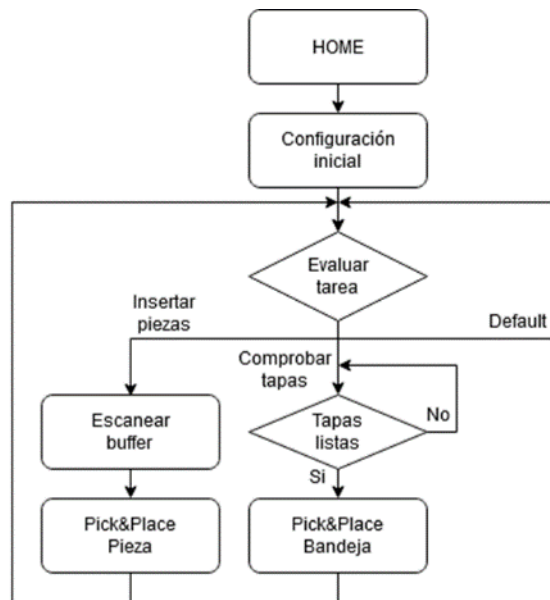


Figura 91. Diagrama de flujo general del programa del Omron TM5-900

En base a estos bloques, la aplicación del robot se ha dividido en un programa principal donde comienza el proceso y subprogramas llamados desde el programa principal, a los cuales se ha hecho referencia anteriormente:

- gripperConf: configuración inicial de la pinza ya que necesita realizar un proceso de apertura y cierre para comprobar internamente que su funcionamiento es correcto. Además se configura la apertura, fuerza y velocidad iniciales.
- checkConnect: subprograma al que se llama antes de realizar cualquier operación de lectura/escritura por Modbus. En este programa se intenta establecer comunicación con el servidor, actualizando una variable booleana en consecuencia. Si la comunicación falla, el robot realiza un proceso de desconexión, avisando al usuario del error y regresando a la posición de HOME.
- scanBuffer: el robot se posiciona sobre el buffer con ayuda de un Landmark. Tras escanear el buffer se actualiza una variable que almacena la información referente al tipo de piezas detectadas en las dos posiciones del buffer.
- pickSeq: el robot lee por Modbus la posición de recogida. Si la posición de recogida es válida, se mueve a dicha posición orientándose según el tipo de pieza en el buffer. En el momento que coge la pieza, informa al servidor.
- placeSeq: el robot lee por Modbus si la pieza tiene que ser desechada (por error o mala presentación), o introducida en la bandeja. En los dos primeros casos el robot lleva la pieza a los contenedores habilitados a tal efecto (posiciones fijas). Si la pieza es correcta busca la bandeja con ayuda del landmark y posiciona las piezas en referencia a él (corrige la posición

de dejada en función del tipo de pieza que lleva consigo). Al depositar la pieza se lo comunica al servidor.

- **checkCovers:** proceso realizado en tres fases. En primer lugar, se analizan las dos posiciones de un extremo, luego las centrales y finalmente las del otro extremo. Además, de esta forma la aplicación es más robusta frente a diferentes condiciones de iluminación. El robot avisa al servidor cuando detecta todas las tapas.
- **pickPlaceTray:** el robot vuelve a referenciar la bandeja en base a su landmark, de esta forma será capaz de coger la bandeja por el asa de forma precisa y depositarla en la cinta transportadora. El robot informa al servidor tanto en el instante en que coge la bandeja como en el que la deposita.

El programa del robot solo podrá funcionar si es capaz de comunicarse correctamente con un **servidor ModbusTCP** (unidad de control) y para ello se han realizado una serie de ajustes.

Por otra parte, teniendo en consideración los **módulos del software de visión** integrados en TMFlow (TMVision) y los pasos generales de implementación explicados en la tarea 3.2., se desarrollaron los algoritmos del sistema de percepción con la cámara integrada (EyeInHand), habiendo realizado la calibración. Como se ha comentado en el programa existen cuatro puntos donde la visión es un factor clave:

- Reconocimiento del patrón del buffer: este nodo de visión es del tipo “fixed positioning” ya que interesa, además de localizar el patrón del buffer, conocer su ubicación en el espacio para poder definir puntos respecto al marcador. Todos los nodos de visión de tipo “fixed positioning” usados en esta aplicación comparten una misma calibración previa.
- Reconocimiento del patrón de la bandeja: misma secuencia de operaciones que en la búsqueda del patrón del buffer. Este nodo de visión también es del tipo “fixed positioning” por lo que crea un sistema de referencia sobre el objeto que ha encontrado.
- Reconocimiento de piezas en el buffer: la búsqueda de las piezas que hay sobre el buffer se realiza en varios nodos de visión en serie, todos ellos comparten la misma estructura.
- Chequeo de tapas: estos nodos de visión son del tipo AOI, es decir, se reducen únicamente a un análisis de la imagen por lo que no generan ningún sistema de referencia asociado. La comprobación de que se han introducido las tapas en las piezas de forma correcta se realiza buscando un elemento en la imagen analizando las posiciones de la bandeja de dos a dos.

En el apartado de pruebas y validación final se mostrarán imágenes del proceso.

3.3.6.5 Integración del robot UR5e

Se ha integrado el UR5e con el objetivo de dotar de la mayor flexibilidad al Demostrador 2 en su etapa del sistema de distribución mediante interacción humano - máquina. Con el UR5e, el operario debe realizar el pick&place de la bandeja posicionada en la cinta para que ésta sea colocada en la plataforma del robot (Figura 92). Para ello debe guiar al robot UR5e para indicarle la posición de recogida de la bandeja, una vez que la cinta se detiene, y posteriormente, una vez que el robot se haya girado hacia su plataforma, el operario debe nuevamente guiar al robot para indicarle la posición de depósito. Para guiarle debe coger el brazo robótico por la herramienta, en su extremo cercano a la junta del efector final y realizar una pequeña fuerza hacia abajo.



Figura 92. Pasos del pick&place con el UR5e

Hay que tener en consideración que la cinta puede tener diferentes velocidades, por lo que cuando la bandeja es detectada por el sensor de presencia, la cinta se parará con una determinada curva de desaceleración no dependiente de la velocidad y por tanto, la posición de la bandeja cuando se detiene no es fija. Además, el posicionado de la bandeja por parte del Omron TM5-900 puede sufrir ligeras rotaciones. De esta forma, es necesaria la colaboración con un humano para el pick&place, considerando que la bandeja puede estar colocada en diferentes posiciones de la cinta, que aunque su variación es pequeña, hace que su posición no sea fija. Por otra parte, al colocar la bandeja en la plataforma, es posible mover posteriormente el robot a otro lugar de la instalación para realizar la tarea de almacenaje o distribución del lote realizado. También, podría utilizarse para ir depositando varios elementos para su posterior distribución.

En la Figura 93 se muestra el **diagrama de flujo** del programa realizado, que se puede resumir en los siguientes bloques:

1. Configuración inicial: el robot se posiciona encima de la cinta a la espera de que el operario realice una fuerza cerca de la unión del extremo de la herramienta junto a la muñeca. Se realizará también la configuración inicial de la pinza.
2. Detección de fuerza: se evalúa constantemente la fuerza detectada por el sensor de fuerza (force()) y cuando supera un determinado umbral, el robot se mueve ya que considera que el operador ya está guiándole a la posición de recogida de la bandeja.
3. Movimiento en z hasta una determinada altura: el robot se mueve hacia abajo, con la guía del operario, hasta una determinada altura. La bajada del robot se consigue mediante la combinación de dos comandos:
 - a. Plantilla fuerza, que permite aplicarla en distintos ejes. En este caso, se aplica una fuerza de cero en x, y, z y rz permitiendo así la rotación de la herramienta en x e y.
 - b. El comando de direction/until para mover la herramienta en el eje z hasta alcanzar una determinada altura.
4. Recogida de la bandeja: una vez que el robot detecta que ha llegado a la altura prefijada, cierra la pinza, recogiendo así la bandeja.
5. Movimiento del robot para ir a la posición de dejada: el robot realiza la trayectoria prefijada para levantar la bandeja y realizar el giro hacia su plataforma.
6. Detección de fuerza: una vez en la posición inicial de dejada, como en el caso anterior, se evalúa constantemente la fuerza detectada por el sensor de fuerza. Cuando supera un determinado umbral, el robot se mueve ya que considera que el operador ya está guiándole a la posición de recogida de la bandeja.

7. Movimiento en z hasta una determinada altura: el robot se mueve hacia abajo, con la guía del operario, hasta una determinada altura. La bajada del robot se consigue mediante la combinación de fuerza y direction/until como en el caso anterior.
8. Dejada de la bandeja: una vez que el robot detecta que ha llegado a la altura prefijada, abre la pinza, dejando así la bandeja sobre la plataforma.
9. Movimiento a posición de reposo: una vez finalizado el proceso va a posición de reposo a la espera de la siguiente acción (siguiente bandeja, otro lugar de la planta, etc.)

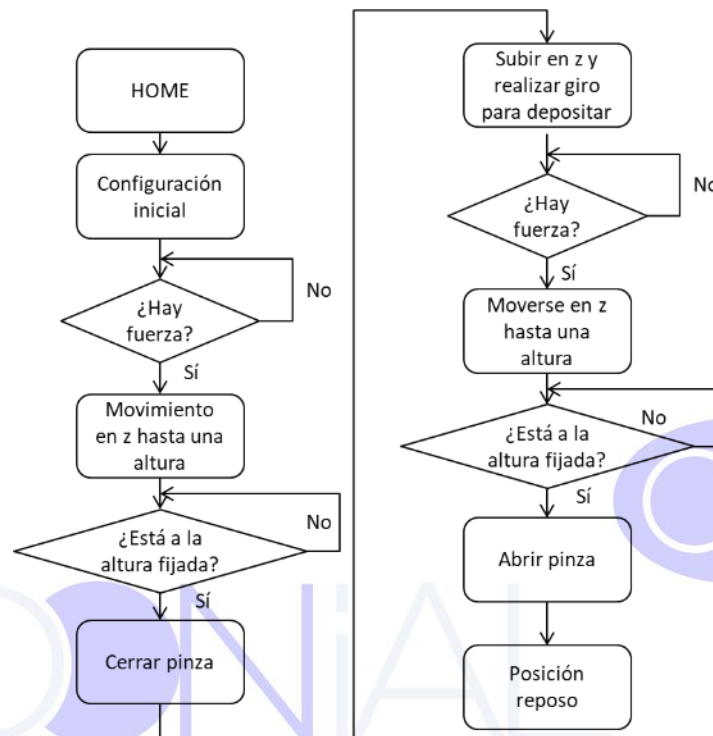


Figura 93. Diagrama de flujo del programa del UR5e

En el siguiente apartado de pruebas y validación final se mostrarán imágenes del proceso.

3.3.6.6 Pruebas y validación final

En primer lugar, se validó la **configuración de los campos de protección y advertencia monitorizados** con el SICK para la colaboración con el TM5-900. Como se puede apreciar en la Figura 94, la pantalla del SICK escáner cambia de color según la posición del usuario, la velocidad del robot se modifica y el color de su anillo varía:

- Usuario fuera de las zonas monitorizadas (arriba): SICK y anillo del robot con luz verde (en modo manual). En caso de encontrarse en modo automático, la luz del robot sería azul.
- Usuario en la zona de advertencia (abajo izqda.): SICK en amarillo, el robot disminuye su velocidad y su anillo LED parpadea alternando entre verde y morado debido a que se pasa a modo colaborativo. En caso de encontrarse en modo automático, la luz del robot sería azul y morada.
- Usuario en la zona de protección (abajo dcha.): SICK en rojo, el robot se para y su anillo LED pasa a color morado con un breve parpadeo en verde indicando que la pausa de protección se ha realizado estando en modo manual. Cuando el usuario sale de la zona, el robot vuelve

a ponerse en movimiento con la velocidad determinada según el modo en que se encuentre debido a la posición del usuario.

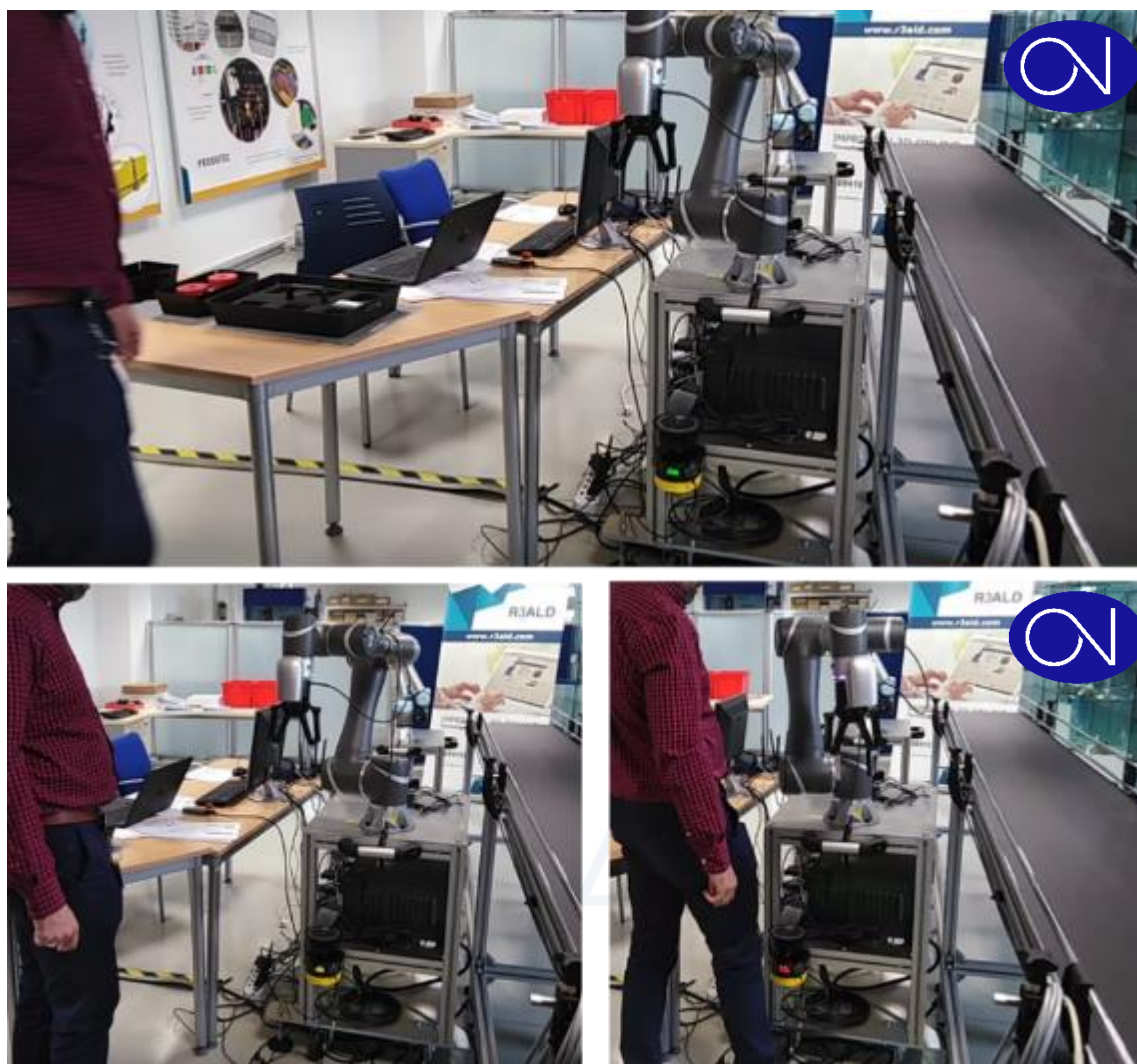


Figura 94. Validación de la integración del SICK escáner y el Omron TM5-900, monitorizando un área de advertencia (abajo izqda.) y de protección (abajo dcha.)

Posteriormente se colocó la mesa de ensamblaje en su sitio y se realizaron las pruebas del sistema completo, teniendo el setup mostrado en la Figura 95. La Raspberry se encuentra en la mesa de ensamblaje colaborativo para facilitar la interacción con el usuario.



Figura 95. Setup del Demostrador 2

A continuación se detalla el proceso completo relacionado con el Omron TM5-900 y se muestran algunas imágenes del mismo a modo explicativo:

- Colocación de piezas en el buffer por parte del operador mientras que el robot las recoge y las coloca en la bandeja al ser correctas. Se ha apreciado que, mientras que el robot está trabajando, existe un tiempo de inactividad del operario durante el pick&place de las dos últimas piezas, por lo que se considera que puede ser un tiempo útil para otras tareas como por ejemplo, revisar el número de piezas disponible, actualizar las órdenes de fabricación y así no quedarse sin stock para las próximas bandejas.
- Una vez que la bandeja contiene todas las piezas correctas, el operador introduce las tapas y mientras tanto el Omron TM5-900 comprueba si las tapas están puestas, inspeccionando de dos en dos las posiciones de la bandeja.
- Una vez que todas las tapas están puestas, el Omron TM5-900 detecta la bandeja y la coge por su asa para depositarla en la cinta (Figura 96). Si el operario se cruzase en su trayectoria, el robot se pararía debido a que estaría entrando en la zona de protección monitorizada por el SICK escáner.
- Una vez que la bandeja está depositada en la cinta, el robot vuelve a su posición de inicio para la siguiente preparación de bandeja. Mientras tanto, cuando la bandeja es detectada por el sensor de presencia central y la cinta se pone en movimiento. En este caso el operador va a ejecutar las operaciones en el sistema de distribución por lo que acude a la zona de trabajo del UR5e.

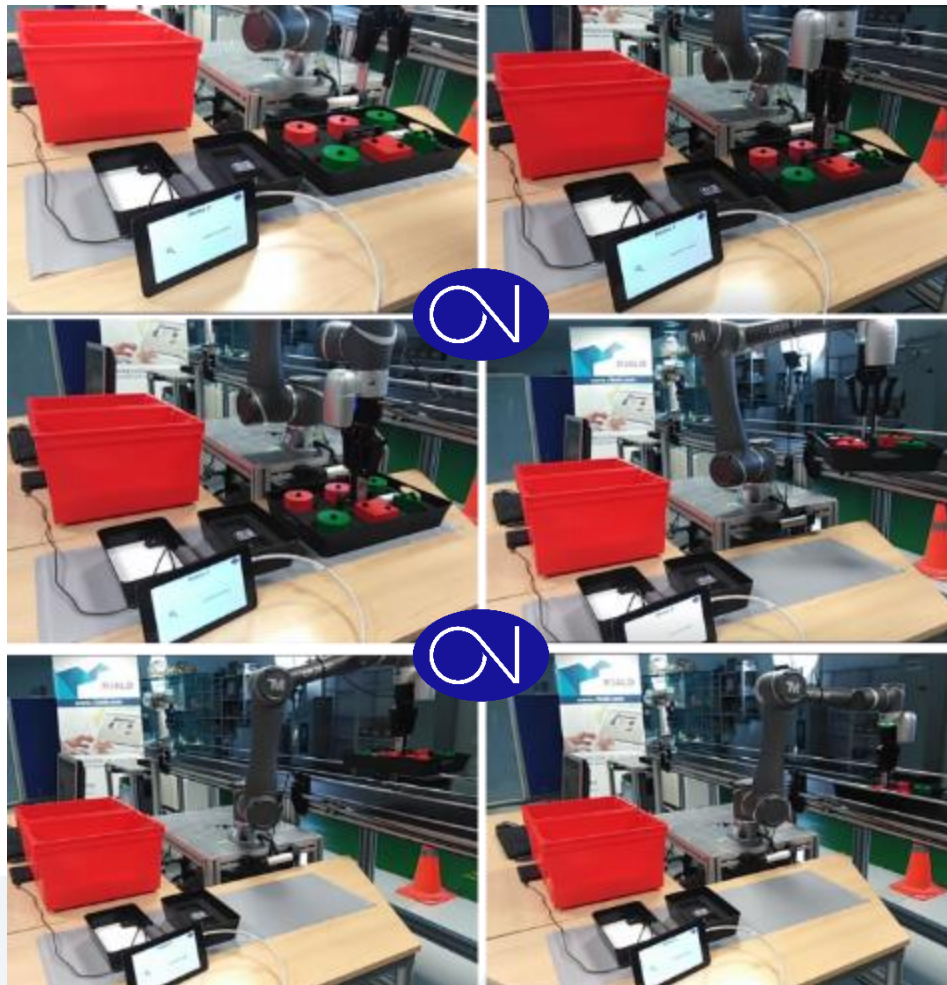


Figura 96. Pick&place de la bandeja

Durante el proceso se muestran en la Raspberry las instrucciones que el operario debe seguir. Tras la pantalla inicial, se le muestran al operario los pasos en relación a la bandeja (depositar en la mesa de ensamblaje) y a las piezas del lote que tiene que preparar (colocar la pieza solicitada en el buffer). El operario tiene libertad para colocar la pieza en cualquier posición del buffer. Si el buffer tiene una sola posición disponible, se solicitará una nueva pieza al operario. Si las dos posiciones del buffer están disponibles, se le mostrará las dos piezas siguientes a colocar. En la Figura 97, a la izquierda, se muestran las instrucciones una vez finalizado el lote y, a la derecha, la imagen de la bandeja real en ese momento. Una vez finalizado el proceso se vuelve a mostrar la pantalla inicial. Se ha decidido simplificar el número de pantallas mostradas al usuario respecto a la propuesta en la tarea 3.5 ya que el operador solicitó durante las pruebas una interacción más ágil con el robot. De esta forma, si el robot no indica nada, las piezas son correctas y el usuario puede seguir con el proceso de una forma más rápida.

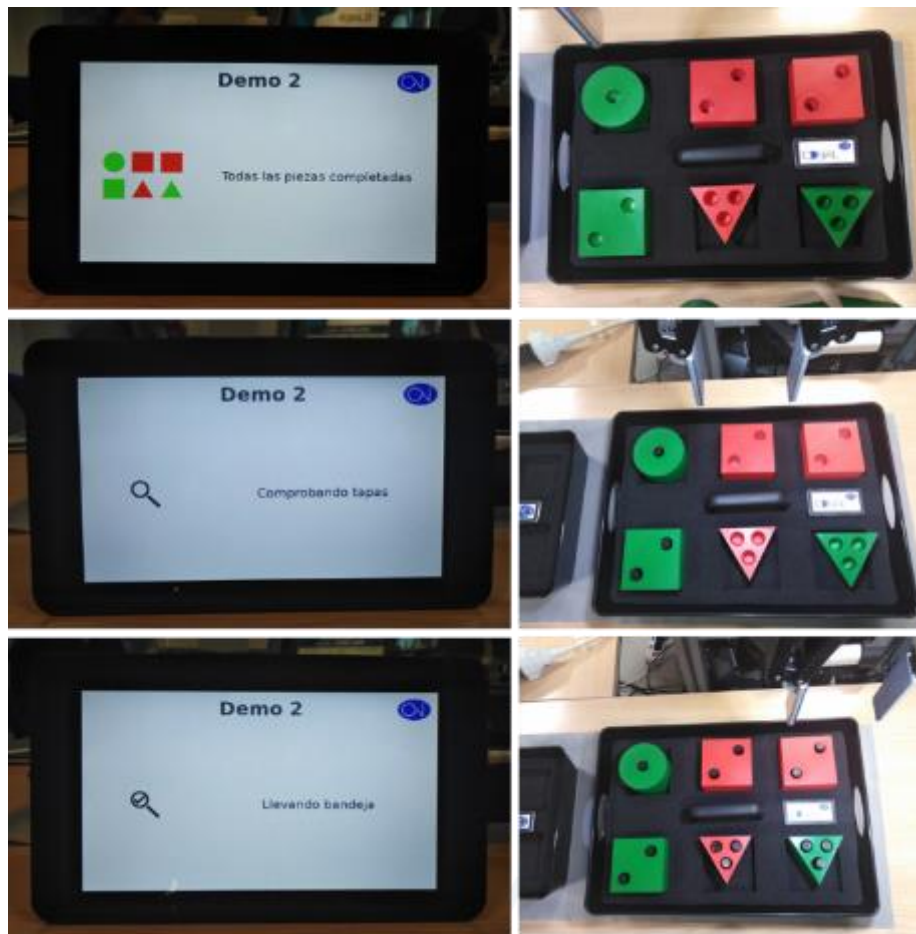


Figura 97. Instrucciones mostradas en la Raspberry (izqda.) e imagen de la bandeja real en ese momento (dcha.)

En las imágenes siguientes se muestra cómo sería la interacción en caso de que el usuario utilizase las gafas de Realidad Aumentada (RA). Al ponerse el usuario las gafas de RA, lo primero que tiene que realizar es buscar los activadores QR. Estos marcadores del mundo real se utilizan para reconocer el entorno físico y asociar la información virtual que se debe añadir. De esta forma, al detectar los códigos QR en el entorno, se establece un sistema de coordenadas en la ubicación real del marcador y muestra la información pertinente asociada a su posición.

A modo de ejemplo, en la parte superior izquierda de la Figura 98 se puede apreciar que cuando el usuario mira a la zona de las cajas de almacenaje, aparecen unas flechas moviéndose para indicarle que ahí se encuentran las piezas a coger (el detalle de lo que visualiza el usuario se puede ver en la parte inferior de la imagen). En la parte derecha de la Figura 98 se muestra el caso cuando el usuario mira a la zona donde se encuentra la bandeja. De la misma forma, aparece una flecha moviéndose de arriba hacia abajo para indicarle que ahí se encuentra la bandeja a coger y depositar en la zona de trabajo gracias a la interfaz de RA.



Figura 98. Interacción del usuario con gafas de realidad aumentada (II): zona de las cajas de almacenaje (izqda.) y zona donde se encuentra la bandeja (dcha.)

Una vez que la bandeja está depositada en la cinta y ésta se pone en movimiento, la bandeja se va desplazando, hasta que llega al sensor de presencia del extremo, que es el instante en el que se da la orden de parada a la cinta. Mientras tanto el UR5e permanece en reposo a la espera del operador, el cual ya se ha desplazado de la zona de trabajo con el Omron TM5-900 a la zona del UR5e.

Para realizar la operación de recogida de la bandeja, al parar la cinta el operador se acerca al UR5e por el lateral contrario al pasillo verde, y coge al robot por la herramienta (cerca de la junta del efector final) ejerciendo una ligera fuerza hacia abajo. En ese instante, el robot empieza a descender en z, permitiendo rotaciones en su efector final para que el operador guíe la pinza hasta el asa. Una vez que llega a la altura prefijada, se cierra la pinza y levanta la bandeja.

Posteriormente, el robot realiza el giro de la junta de su base en el sentido contrario al que se encuentra el operador. Una vez posicionado sobre su plataforma, espera a que el operador coja al robot por la herramienta (cerca de la junta del efector final) ejerciendo una ligera fuerza hacia abajo. Como anteriormente, en ese momento el robot empieza a descender en z, permitiendo rotaciones en su efector final para que el operador guíe al robot hasta la posición de dejada. Una vez que llega a la altura prefijada, se abre la pinza por lo que la bandeja queda depositada y el robot se eleva para posicionarse en reposo a la espera de la siguiente acción.

3.4 HITO 4. INTERACCIÓN MÁQUINA – MÁQUINA

En este hito se ha desarrollado el Demostrador 3 del proyecto, el cual se basa en la colaboración máquina – máquina. Dicho demostrador se ha desarrollado según los requisitos definidos en el HITO 1 y aprovechando parte de los equipos e instalaciones realizadas en el Demostrador 2 en el HITO 3. A continuación se detallan las principales actividades y resultados realizados en cada una de las tareas de este hito.

3.4.1 T4.1. Diseño de herramientas específicas y ajuste del entorno

En este hito, una de las primeras acciones a realizar fue la revisión del Demostrador 3 definido para este caso. A continuación se resumen las principales características asociadas a este demostrador:

- Se plantea como demostrador el uso de una colaboración efectiva entre robots (máquina – máquina) para montaje de dispositivos. → Para ello se dispone de 2 robots, UR5e (UNIVERSAL ROBOTS) y TM5-900 (OMRON); cada uno de ellos con plataforma móvil intercambiable entre los dos robots; 2 grippers (Robotiq 2F-140 y Robotiq Hand-e), compatibles para cada robot y acople en cada robot; 1 escáner láser de seguridad de SICK microScan3 Core como dispositivo de protección de la celda. Además de estos elementos, cuenta con otros desarrollados y adquiridos en el marco del proyecto como la cinta transportadora, armario de control, etc.
- Dispositivos para montar → son unos dispositivos formados por carcasa (pieza superior e inferior) así como una placa intermedia y encerramiento realizado por medio de tornillos.
- Proceso de montaje de dispositivos → en base a los dispositivos descritos anteriormente, el proceso de montaje de los mismos se realiza entre ambos robots aprovechando además que el TM5-900 tiene una cámara integrada. Como se explicará en apartados posteriores, se plantea un proceso de montaje donde el Robot 1 es UR5e y Robot 2 es Omron TM5-900. Una vez montado, el dispositivo se coloca sobre una cinta transportadora. Con esta información, se revisa el proceso de montaje de los dispositivos para el diseño final de dispositivo escogido y el equipamiento disponible finalmente para realizarlo.

Con esta definición, a continuación se han considerado una serie de aspectos adicionales como validar el tamaño mínimo de tornillo con el que puede trabajar de manera adecuada el robot UR5e. Para ello se deberá analizar si se puede crear una herramienta para que coja un destornillador o llave Allen. Igualmente, se debe analizar el modo en que se colocan los elementos y se sacan las piezas al final de la cinta (caída en cesta, moverlas poco a poco...). También ha habido otros elementos auxiliares que han debido revisarse / rediseñarse, como por ejemplo las conexiones eléctricas y el armario eléctrico asociado.

Con este replanteamiento del demostrador a causa de los motivos indicados, se indican las principales acciones realizadas y cómo resulta el estado final del demostrador.

3.4.1.1 Definición del escenario del Demostrador 3 y proceso de montaje

La Figura 99 muestra los dos robots colaborativos de los que se dispondrá (Omron TM5-900 y Universal Robots UR5e), las garras o grippers que les permitirán realizar tareas (Robotiq 2F-140 y Hand-e) así como el escáner láser de seguridad SICK MicroScan3 Core. La distribución del escenario que permitirá realizar las pruebas será la formada por una cinta transportadora, dos robots

colaborativos enfrentados donde cada uno de ellos tiene una garra, un sistema de seguridad y los armarios de control de todo el sistema.



Figura 99. Demostrador 3: planteamiento general

De esta forma, en la arquitectura general mostrada en la Figura 100 se distinguen los siguientes bloques principales:

1. **Robot 1:** está compuesto por el brazo robótico UR5e instalado sobre una plataforma y un efector final o herramienta.
2. **Robot 2:** está compuesto por el brazo robótico Omron TM5-900 sobre una plataforma y un efector final o herramienta.
3. **Seguridad:** bloque encargado de la seguridad del entorno de trabajo. Se ha optado por incluir al menos el escáner SICK, ya que colocándolo a la altura de los tobillos de un operario monitoriza el acceso.
4. **Comunicaciones:** este bloque contará con los dispositivos necesarios para garantizar la comunicación entre los diferentes componentes.
5. **Unidad de control:** se dispondrá de una unidad de control para controlar la ejecución del programa de la célula.
6. **Sistema de distribución:** una vez que los robots han finalizado, se posiciona el dispositivo sobre una cinta transportadora.

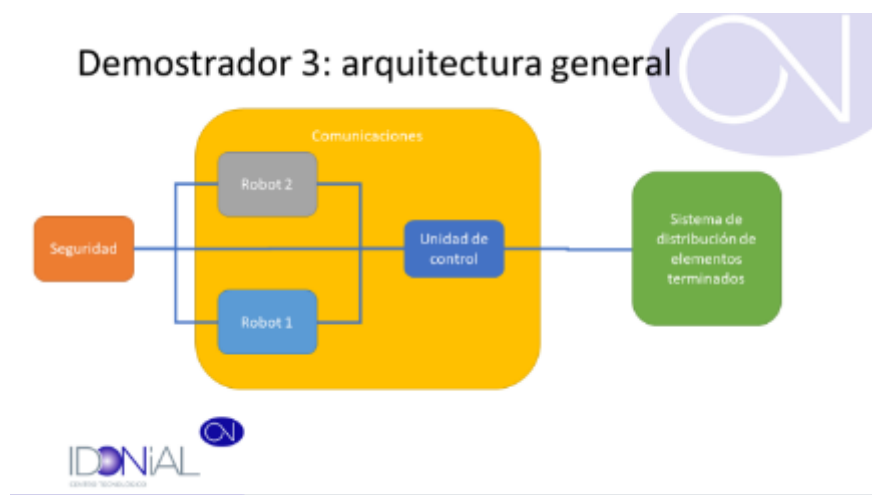
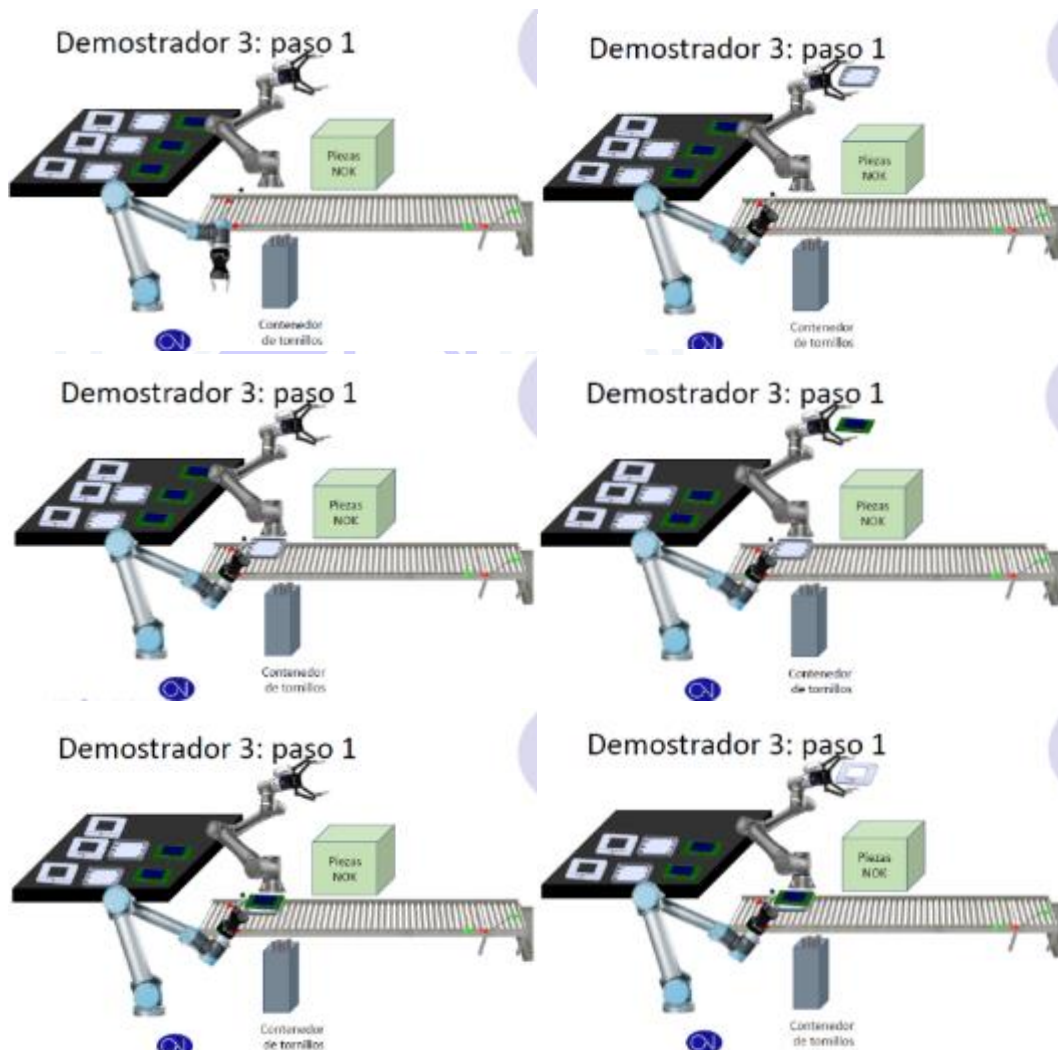


Figura 100. Arquitectura general del Demostrador 3

En base a estos dispositivos descritos anteriormente se ha planteado un proceso de montaje que consta de las siguientes fases: 1. Ensamblaje, 2. Atornillado y 3. Inspección. A continuación se describen las tareas que se han planteado para cada una de estas fases, donde el Robot 1 es UR5e y Robot 2 es Omron TM5-900:

1) Ensamblaje: se detallan a continuación las tareas que forman esta etapa, que a su vez se muestra en la Figura 101.

- Robot 2 coge la pieza inferior de las bandejas.
- Robot 2 entrega pieza a robot 1.
- Robot 2 coge PCB de las bandejas (PCB colocadas en vertical) y la pone sobre pieza sujeta por robot 1.
- Robot 2 coge la pieza superior de las bandejas y la pone sobre la pieza+PCB sujeta por robot 1.
- Robot 2 coge el dispositivo del robot 1, le da la vuelta y lo coloca en una posición determinada para que el robot 1 coloque los tornillos.



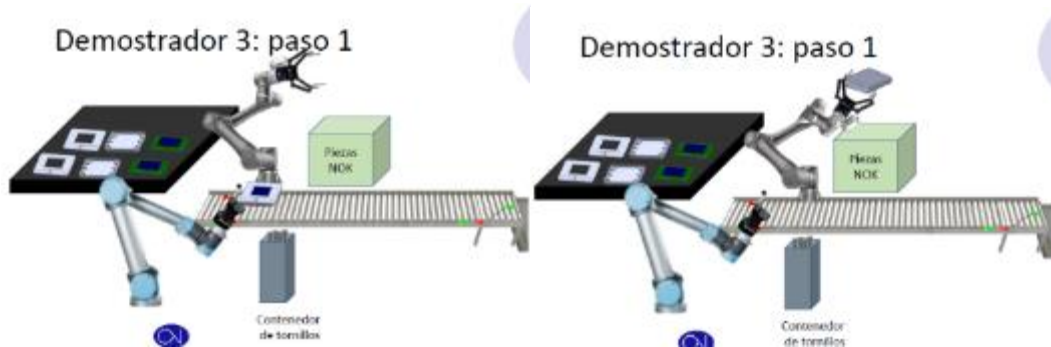


Figura 101. Ensamblaje - tareas

2) Atornillado: se detallan a continuación las tareas que forman esta etapa, que a su vez se muestra en la Figura 102.

- Robot 1 coge tornillo de una posición fija mientras tanto, lo presenta en el dispositivo cuando éste está en la posición fijada por el robot 2 y aprieta (si se realiza con destornillador es necesario pensar cómo cogerlo, y en caso de no utilizarlo, cómo realizar el atornillado directamente con la garra).
- Robot 1 coge segundo tornillo de una posición fija, lo pone sobre el dispositivo sujetado por el robot 2 y aprieta (si se realiza con destornillador es necesario pensar cómo cogerlo, y en caso de no utilizarlo, cómo realizar el atornillado directamente con la garra).

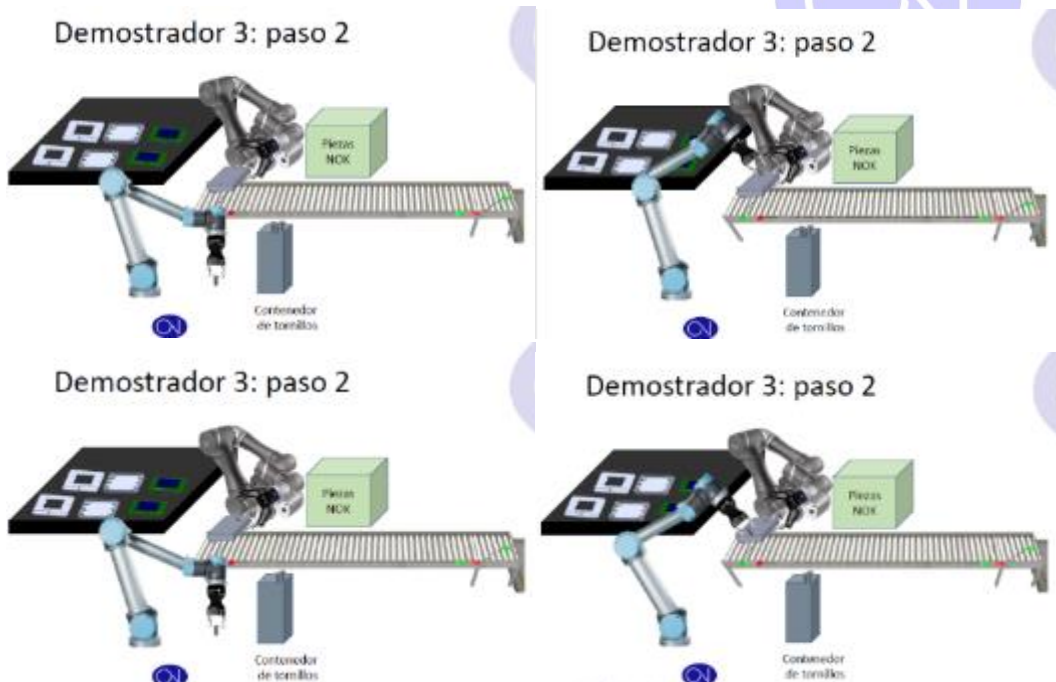




Figura 102. Atornillado - tareas

3) Inspección: se detallan a continuación las tareas que forman esta etapa, que a su vez se muestra en la Figura 102.

- Robot 1 coge el dispositivo agarrado por robot 2 y lo coloca en la cinta.
- Robot 2 saca foto para inspeccionar si está bien (tamaño, apoyado plano sobre cinta) y la guarda a modo de trazabilidad.
- Si está mal, el robot 2 lo coge de la cinta y lo echa a la caja de NOK
- Si está OK, se mueve la cinta una posición para liberar el espacio de disposición del siguiente dispositivo, mientras el robot 2 coge la siguiente pieza a montar.



Figura 103. Inspección - tareas

Partiendo del proceso y el dispositivo indicados, se ha analizado la viabilidad de realizar dicho proceso con el equipamiento disponible para ver si es necesario diseñar herramientas específicas o adquirir elementos de apoyo. A medida que se ha ido desarrollando este demostrador y conociendo las limitaciones del sistema de agarre, por ejemplo, el dispositivo ha debido de revisarse y ajustarse a las capacidades de los manipuladores robóticos y herramientas diseñadas o desarrolladas en el marco de esta tarea. El detalle y análisis de los diversos dispositivos analizados se recoge en el apartado 3.4.1.2. Al igual que para el dispositivo, ha sido necesario revisar el proceso de montaje para el nuevo dispositivo y posibles nuevos ajustes del escenario, tal y como se recoge en el apartado 3.4.1.3.

En cuanto al tipo de dispositivos, inicialmente se definieron unos dispositivos con una carcasa realizada por fabricación aditiva, con una placa base (PCB) intermedia y un encerramiento por medio de dos tornillos. En la Figura 104 se muestra el diseño inicial del dispositivo pero, a medida que se ha ido desarrollando este demostrador y conociendo las limitaciones del sistema de agarre,

por ejemplo, este dispositivo ha debido de revisarse y ajustarse a las capacidades de los manipuladores robóticos y herramientas diseñadas o desarrolladas en el marco de esta tarea. El detalle y análisis de los diversos dispositivos analizados se recoge en el apartado 3.4.1.2.

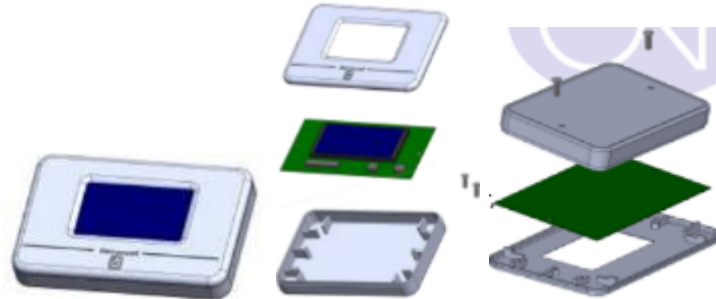


Figura 104. Dispositivo para montaje del Demostrador 3

3.4.1.2 Revisión del dispositivo

En este caso, la primera labor realizada ha sido la de buscar un dispositivo que pudiera ser fabricado por medio de impresión 3D en las instalaciones de IDONIAL y dentro del cual se pudieran posicionar una serie de placas comerciales disponibles en la entidad (por ejemplo, Arduino o Raspberry Pi). El cierre del conjunto se realizará por medio de tornillos, y para realizar la selección final se han realizado una serie de pruebas, tal y como se detalla en la tarea 4.3.

Tras haber analizado diversos diseños, con sus ventajas y desventajas, haber realizado pruebas de montaje manual y un estudio de viabilidad, se han seleccionado un determinado diseño de carcasa y bases para la presentación de las diversas partes que conforman el dispositivo a montar y una herramienta de agarre específicos para este demostrador, tal y como se describe a continuación.

Así, para el Demostrador 3 se han diseñado varias piezas con ayuda del programa Blender y que posteriormente se han fabricado con una impresora de FDM usando PLA. De esta forma, por un lado, las piezas de las que consta el dispositivo y que se han diseñado y fabricado, son:

- Carcasa:
 - Parte superior.
 - Parte inferior.
 - Soporte.

Por otro lado, los elementos adicionales que se requieren para realizar el montaje en los manipuladores robóticos, son:

- Base:
 - Base para el UR5e.
 - Base para el Omron TM5-900.
- Dedos personalizados para la garra Hand-E usada en el UR5e:
 - Dedo izquierdo.
 - Dedo derecho.

A continuación se resumen los principales resultados asociados a cada uno de estos elementos.

Carcasa - diseño:

Como se ha indicado anteriormente, para llegar a seleccionar esta carcasa se ha hecho un estudio exhaustivo de las diversas opciones. Finalmente, se toma como referencia el siguiente modelo para una Raspberry Pi 3B+ (Figura 105). La carcasa consta de tres piezas: una superior, una inferior, y un soporte auxiliar que viene muy bien para la secuencia de ensamblaje propuesta para los robots.

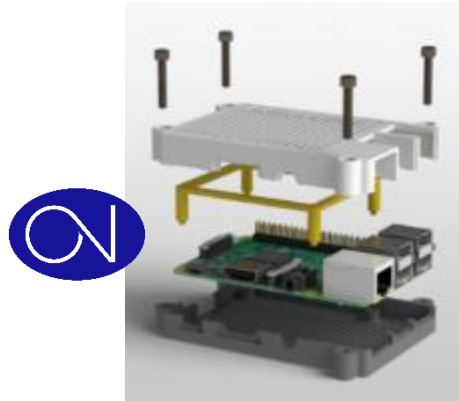


Figura 105. Modelo de carcasa de referencia²⁴

Inicialmente para validar se han realizado las piezas en una impresora de baja resolución y se ha comprobado que el modelo es viable. Posteriormente, se han realizado las mismas pruebas con una impresora de FDM de mayor precisión tal y como se detalla a continuación.

En primer lugar se realizó la fabricación de la parte superior de la carcasa donde hubo que adaptar el diseño de la misma. Este nuevo diseño ha incluido los agujeros para tornillos de métrica M5, más grandes que los del diseño inicial (más fáciles de manejar con las garras del robot). Estos agujeros no están completamente roscados (solo se tendrá rosca en la parte inferior para reducir el tiempo de ensamblado). Además, se aumentó ligeramente la altura de la carcasa para poder introducir un soporte más sólido (se verá en el apartado del soporte) para cerrar la carcasa por encima (tapando los conectores). En cuanto a la parte inferior de la carcasa, al igual que para el caso de la parte superior, los agujeros para atornillar del diseño inicial eran demasiado pequeños para poder trabajar de forma más fiable con las garras del robot y por ello, se ha decidido usar tornillos de métrica M5 por lo que ha sido necesario modificar el diseño. Además, como la rosca en el PLA puede estropearse fácilmente se ha modificado el diseño para introducir unas tuercas de acero inoxidable y asegurar que el roscado con el robot se realiza de forma correcta. Para validar las dimensiones del agujero de la tuerca, la rosca de la parte inferior y las dimensiones del tornillo seleccionado, se ha fabricado un dado auxiliar simulando uno de los puntos de sujeción de la pieza. Una vez validados los parámetros en el dado, se ha modificado el diseño original por el siguiente:

²⁴ <https://www.thingiverse.com/thing:3361218>



Figura 106. Carcasa – parte inferior

Para el caso del soporte, el diseño inicial presentaba varios problemas. En primer lugar era muy endeble por lo que sujetarlo con el robot de forma firme es más complicado. Para solucionar el problema se ha cambiado el diseño por uno con una mayor rigidez, que el robot pueda coger con sus garras con mayor facilidad y firmeza (Figura 107).



Figura 107. Soporte – diseño intermedio

El problema principal de los soportes es el pequeño diámetro de los cilindros que se insertan en la Raspberry. Al imprimir la pieza es necesario tratarla para quitar irregularidades o lijarla en ciertos puntos ya que los cilindros no son perfectos (debido a su pequeño tamaño), si no se hace con mucho cuidado, pueden romperse. Así, se ha usado un nuevo diseño para mejorar la firmeza en la sujeción (Figura 108).

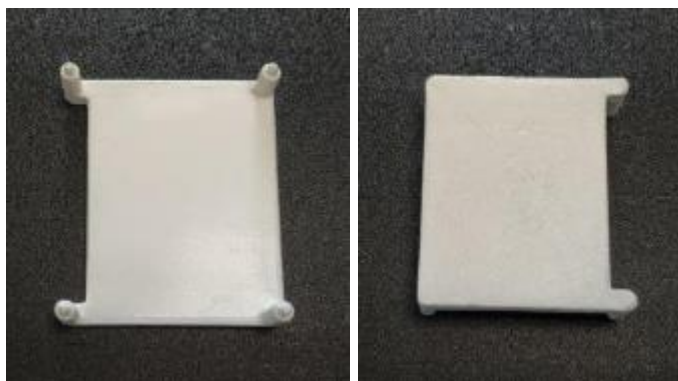


Figura 108. Soporte – diseño final

Carcasa - montaje:

Una vez se dispone de las versiones finales del diseño de la carcasa (Figura 109), los pasos del proceso de montaje son los que se explican a continuación. Se coloca el soporte sobre la PCB (Figura 110), a continuación se introduce la placa con el soporte en la parte inferior de la carcasa (Figura 111). Posteriormente se posiciona la parte superior de la carcasa sobre el conjunto obtenido del paso anterior (Figura 112). Por último, se atornillan los cuatro tornillos para unir la parte superior e inferior de la carcasa (Figura 113).

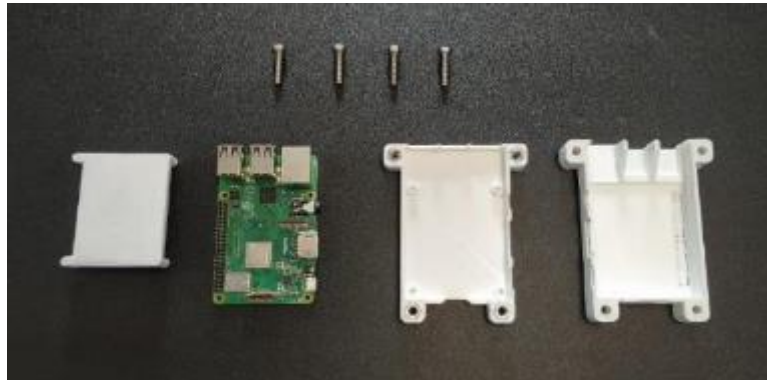


Figura 109. Montaje de carcasa – paso 1: elementos iniciales



Figura 110. Montaje de carcasa – paso 2: posicionamiento de soporte

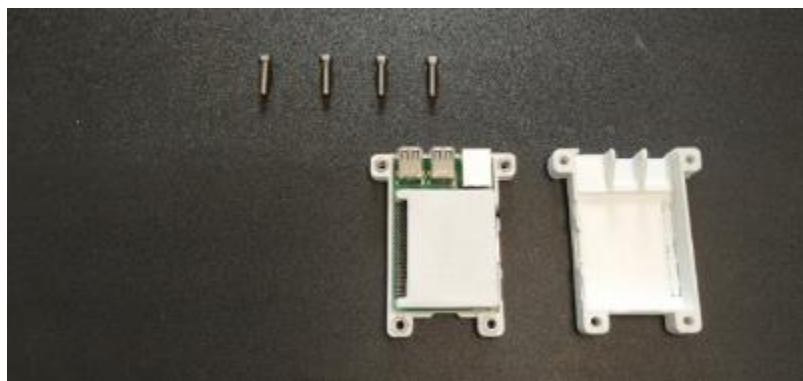


Figura 111. Montaje de carcasa – paso 3: posicionamiento de parte inferior de carcasa

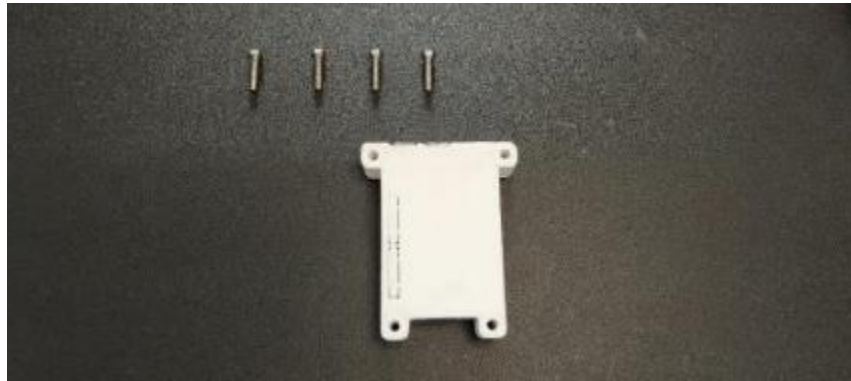


Figura 112. Montaje de carcasa – paso 4: posicionamiento de parte superior de carcasa

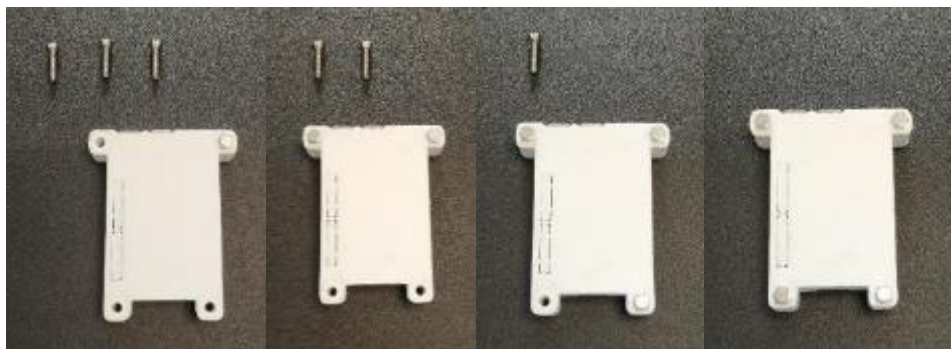


Figura 113. Montaje de carcasa – paso 5: atornillado

Base para el UR5e - diseño:

Las bases para las piezas facilitarán el trabajo a los robots, siendo un factor clave en el caso del UR5e ya que trabajará con posiciones fijas. Se han realizado varios diseños hasta obtener el adecuado para la aplicación.

El diseño final de la base para las piezas del UR5e consta de soportes para los tornillos, para la pieza superior de la carcasa, para el retenedor y para añadir un landmark de calibración que permitirá añadir una secuencia de calibrado para que la posición de inserción del TM5-900 se adapte a la del UR5e (Figura 114, izquierda). Debido a que los carritos de los robots son simétricos, presentan cuatro agujeros en la parte opuesta a la que ocupa el robot, que se puede aprovechar para insertar la base con las piezas (Figura 114, derecha). En la Figura 115 se puede ver la base del UR5e instalada en el carrito con todas las piezas colocadas con las que este robot se relaciona: el soporte y la parte superior de la carcasa, los cuatro tornillos, y el landmark de calibrado.



Figura 114. Base para el UR5e – parte superior (izqda.) e inferior (dcha.)



Figura 115. Base para el UR5e instalada en el carrito

Base para el TM5-900 - diseño:

Se ha adaptado el modelo anterior para el caso del TM5-900. En este caso se han añadido soportes para la parte inferior de la carcasa y la Raspberry, que son los objetos con los que interactúa el robot. También se ha incluido una cavidad para añadir un landmark, de forma que el robot pueda sacar provecho de la cámara integrada, localizando la base con las piezas y accediendo a ellas sin necesidad de colocarlas en una posición fija. Esta base (Figura 116) se puede colocar sobre el carrito, una bandeja o dejarla directamente sobre la mesa en una posición alcanzable por el robot.

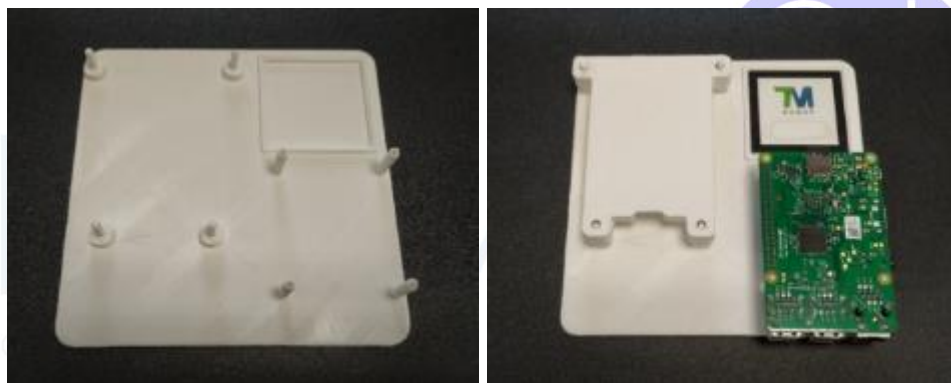


Figura 116. Base para el Omron TM5-900

Dedos personalizados para la garra Hand-E usada en el UR5e - diseño:

La abertura de la pinza Hand-E es de 50mm. Debido a que el UR5e tiene que manejar objetos de un tamaño muy variable (tornillos y carcasa), se ha detectado que los dedos que vienen por defecto con la garra Hand-E no eran adecuados para abarcar los dos casos. Aprovechando la ventaja que proporciona esta garra de poder realizar dedos personalizados e integrarlos en ella, se ha realizado el diseño de unos dedos que se adaptan adecuadamente al trabajo específico del robot en el Demostrador 3. Se han realizado varios diseños y pruebas para poder obtener un diseño que se adaptase correctamente a los objetos a manejar y con la rigidez necesaria para la aplicación. El diseño final fabricado se muestra en la Figura 117. Cabe destacar los siguientes puntos en el diseño de los dedos:

- Se ha añadido un burlete para mejorar la sujeción de las piezas.
- La longitud de los dedos para sujetar las piezas de la carcasa abarca más de la mitad de la pieza más grande a sujetar, por lo que es más complicado que la pieza se mueva.

- La pieza se ha diseñado de forma tal que pudiese imprimirse hacia arriba en la cama de la impresora sin necesidad de añadir ningún soporte.
- La separación entre la zona para coger las piezas de la carcasa y los dedos que recogen los tornillos se ha reducido para aumentar la rigidez, realizando unos agujeros pasantes que atraviesan la pieza.
- Se ha añadido una cavidad en la zona encargada de insertar los tornillos para que al buscar el agujero, el tornillo no se mueva (Figura 118, izquierda).
- Se han añadido dos niveles en la zona encargada de recoger las piezas de la carcasa. De esta forma se puede ejercer presión sobre el retenedor sin moverlo si se sujeta con el primer nivel, a la vez que se puede sujetar firmemente la parte superior de la carcasa si se emplea el segundo nivel (Figura 118, derecha).

En la Figura 119 se muestran los dedos incorporados perfectamente en la garra instalada en el UR5e.

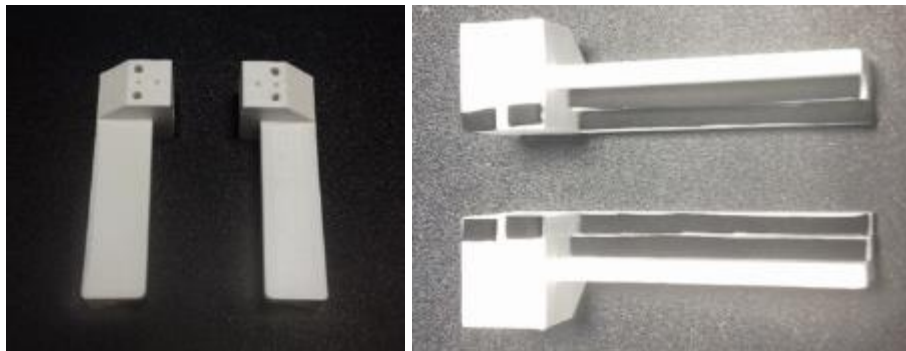


Figura 117. Dedos personalizados para la garra Hand-E

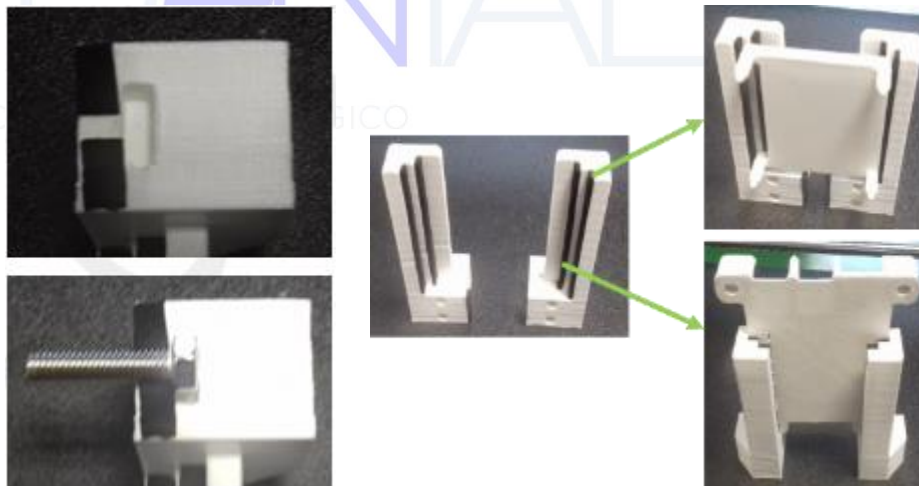


Figura 118. Detalle del diseño de los dedos, resaltando la cavidad para los tornillos (izqda.) y los dos niveles de sujeción (dcha.)

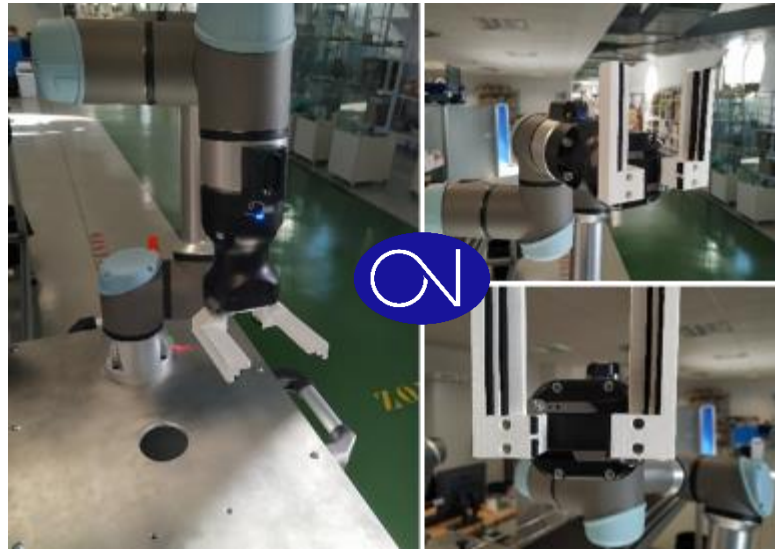


Figura 119. UR5e con la garra Hand-E con los dedos personalizados

3.4.1.3 Revisión del escenario y proceso de montaje

Como se comentó anteriormente, una vez redefinido el dispositivo ha sido necesario revisar el proceso de montaje y posibles nuevos ajustes del escenario en base al diseño de la arquitectura del demostrador. De esta forma se ha analizado el diseño de la arquitectura del demostrador de cara a decidir si es necesario diseñar más herramientas específicas o ajustar el entorno. Para ello se consideran los componentes ya disponibles del escenario anterior de interacción persona-máquina (Demostrador 2), y los elementos diseñados y fabricados, detallados anteriormente relacionados con el propio dispositivo, las bases para colocar sus diversas partes para que los robots los vayan cogiendo y los dedos personalizados para la garra Hand-e. De esta forma, teniendo en consideración los componentes evaluados, diseñados, el entorno y la tarea a desarrollar, la Figura 120 muestra la **arquitectura final** del Demostrador 3 de ROBOTS4.0, compuesta por los siguientes bloques comunicados entre sí – muchos de los cuales han sido comprados como inversión propia de IDONIAL tal y como se indicó en la descripción del Demostrador 2:

1. **Robot 1:** está compuesto por el brazo robótico UR5e, el efector final Robotiq Hand-e con los dedos personalizados comentados anteriormente y la plataforma móvil.
2. **Robot 2:** está compuesto por el brazo robótico Omron TM5-900, el efector final Robotiq 2F-140 y la plataforma móvil.
3. **Seguridad:** bloque encargado de la seguridad del entorno de trabajo. Se ha optado por incluir una seta de emergencia y el SICK scanner, ya que colocándolo a la altura de los tobillos de un operario se monitoriza el acceso a la célula.
4. **Comunicaciones:** este bloque contará con los dispositivos comerciales necesarios para garantizar la comunicación entre los diferentes componentes, ya disponibles del Demostrador 2.
5. **Unidad de control:** se dispondrá de una unidad de control para controlar la ejecución del programa del robot mediante un PLC y HMI a incorporar en el armario de control.
6. **Sistema de distribución:** una vez que los robots han finalizado la tarea de interacción con el usuario, se posiciona el dispositivo montado a una cinta transportadora.



Figura 120. Demostrador 3: arquitectura

Por tanto, se optará por soluciones comerciales para aquellos nuevos componentes necesarios como el HMI y los asociados a la seguridad, armario de control y sistema de distribución. Una vez se dispone de esta información actualizada conforme a las nuevas consideraciones del demostrador, se trabaja en habilitar la zona reservada para el demostrador.

Teniendo en consideración los puntos anteriores, en base a la arquitectura planteada y el dispositivo redefinido, el proceso final de montaje constará de las tres fases planteadas inicialmente 1. Ensamblaje, 2. Atornillado y 3. Inspección, que se explican de forma general a continuación:

1) Ensamblaje: para que los robots puedan ponerse en marcha tienen que estar habilitados por el PLC. Si los robots están habilitados por el PLC realizarán un proceso de inicialización y se detendrán hasta que el otro robot esté habilitado e inicializado, momento en el cual comenzarán a moverse y actualizarán su estado. Cada robot contará con una variable que permitirá reiniciar el flujo de trabajo y otra que indicará cuándo el proceso de reinicio ha terminado.

- 1) Soporte y Raspberry (la PCB): si los dos robots están habilitados e inicializados comenzarán a moverse a la vez.
 - a. El TM5-900 recogerá la Raspberry.
 - b. El UR5e recogerá el soporte de la carcasa.
 - c. Ambos se dirigirán al punto de ensamblado. Como el UR5e se encargará de la inserción del soporte, espera a que el TM5-900 le indique que se encuentra en posición, momento en el que inserta la pieza y lo indica.
- 2) Inserción de Raspberry:
 - a. Cuando el UR5e inserta el soporte, el TM5-900 libera la Raspberry e informa de ello.
 - b. Cuando la Raspberry ha sido liberada, el UR5e vuelve a la posición de inserción (al insertar la pieza se habrá movido ligeramente), a la vez que el TM5-900 se dirige a recoger la parte inferior de la carcasa.
- 3) Carcasa inferior:
 - a. El TM5-900 lleva la parte inferior de la carcasa a la posición de inserción e informa de ello.
 - b. Momento en el que el UR5e inserta el soporte y la Raspberry en la parte inferior e informa al terminar.

- 4) Ambos robots deben de moverse justo a la vez para cambiar la orientación del conjunto (soporte, Raspberry y pieza inferior), de forma que el UR5e quede libre y el TM5-900 sujetándolo.
 - a. El TM5-900 indicará al UR5e que quiere iniciar el movimiento sincronizado. Cuando el UR5e se encuentra a la espera de la orden de movimiento sincronizado y la recibe, resetea la variable y comienza a moverse.
 - b. El TM5-900 comienza a moverse cuando detecta que la variable se ha reseteado.
- 5) Carcasa superior: una vez el movimiento sincronizado termina, el UR5e:
 - a. Suelta el conjunto.
 - b. Se dirige a recoger la parte superior de la carcasa.
 - c. La lleva hasta el punto de inserción.
 - d. La inserta en el conjunto.

2) Atornillado:

- 6) Una vez insertada la parte superior el UR5e se dirige a por un tornillo, lo lleva hasta el punto de inserción y lo atornilla, repitiendo el proceso cuatro veces, una por tornillo.

3) Inspección:

- 7) El UR5e sujeta el dispositivo e informa de ello.
- 8) Cuando el TM5-900 detecta que el UR5e está sujetando el dispositivo lo suelta e informa de ello.
- 9) Cuando el UR5e detecta que el TM5-900 ha soltado el dispositivo lo deposita en la cinta transportadora (de esta forma la parte superior del dispositivo queda mirando hacia arriba), e informa de ello.
- 10) Cuando el TM5-900 detecta que el dispositivo está sobre la cinta transportadora captura una imagen de trazabilidad e informa al PLC de que debe llevarse la pieza, poniéndose en marcha la cinta.

Existen determinados puntos a explorar en cuanto al atornillado, la posición ideal de recogida, intercambio de piezas, etc. En la tarea 4.2 (apartado 3.4.2) se realizará una revisión del diseño eléctrico y el armario eléctrico. En la tarea 4.3 (apartado 3.4.2.1) se realiza un estudio detallado de este proceso para llevar a cabo la definición definitiva y el desarrollo del programa del Demostrador 3, cuya integración se explica en la tarea 4.5 (apartado 3.4.5).

3.4.2 T4.2. Sistema de sincronización

En esta tarea se ha partido de la arquitectura planteada para el Demostrador 3 en la Figura 120. En base a ello se propone el sistema de control detallado en la Figura 121, considerando que debe existir una comunicación entre los robots y el PLC para garantizar el control sincronizado del conjunto. Además, para que esta demo se pueda llevar a cabo es totalmente necesario establecer un mecanismo de comunicación entre ambos robots y así evitar retardos innecesarios, de forma que uno pueda conocer el estado del otro y viceversa. Para ello se va a utilizar el protocolo de comunicación industrial que comparten ambos robots, Modbus TCP/IP. De esta forma, se aprovecha la capacidad de ambos robots de funcionar como servidor Modbus para almacenar las variables referentes a cada uno en su propio servidor y actuando el otro robot como cliente para consultarlas en caso necesario. Así ambos robots funcionarán de forma más autónoma. En cuanto a la comunicación del PLC con cada uno de los robots:

- El PLC se comunicará con el Omron TM5-900 mediante Modbus TCP/IP ya que de entre los protocolos de comunicación incluidos en el Omron (Modbus TCP/IP, Ethernet y RS232), éste es el más escalable. El protocolo permite usar el robot como cliente y/o servidor, de esta forma el PLC puede acceder a información del robot aunque el programa aún no esté en ejecución e incluso controlarlo (seleccionar programa, ejecutar programa, pausar programa, reanudar programa...).
- El PLC se comunicará con el UR5e por PROFINET (mayor escalabilidad de cara a futuro). De todas formas, se hicieron pruebas también de conectividad en Modbus TCP/IP, considerando el PLC como doble server.

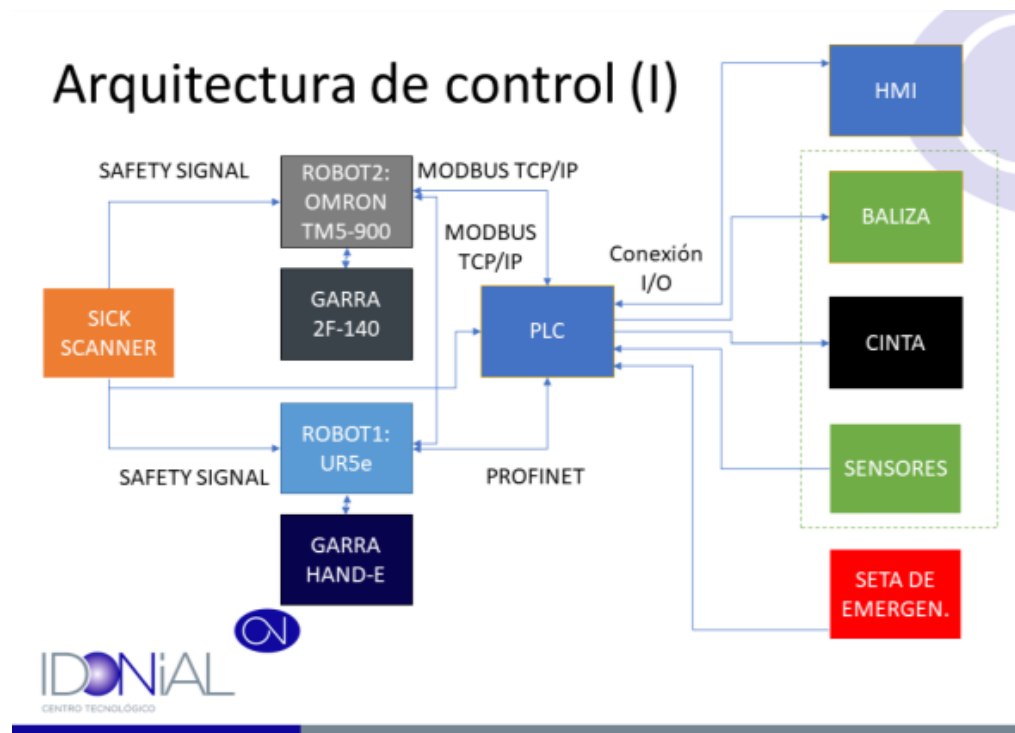


Figura 121. Demostrador 3: arquitectura de control

Por tanto, el PLC se encarga del control del conjunto, gestionando el flujo de trabajo de los robots. En las siguientes imágenes se ve el proceso general de las tres fases planteadas inicialmente 1. Ensamblaje, 2. Atornillado y 3. Inspección. La interacción de ambos robots con el PLC es parecida, el PLC da la orden de inicio de programa, si todo está correcto el programa de ambos robots comienza, pasando de la posición de Home a la posición de inicio. Durante esta primera etapa también se aprovecha para inicializar las garras y algunas variables de programa.

Teniendo en cuenta el proceso general simplificado, se observa que una vez los robots llegan a la posición de inicio se quedan a la espera de una orden del PLC. El PLC puede solicitar comenzar un nuevo proceso de ensamblado o la finalización del programa:

- 1) Si se solicita un nuevo ensamblado ambos robots comenzarán el proceso comunicándose entre sí de forma directa (comunicación robot-robot) a través de Modbus TCP/IP. Ambos robots exponen a través de sus propios servidores el estado particular del ensamblaje para cada uno de ellos, actuando el otro robot como cliente para conocer el estado y pudiendo quedar uno de ellos a la espera del otro, por ejemplo, si uno de los programas se pausa, el

otro se ejecutará con normalidad y cuando eventualmente necesite información del otro robot quedará a la espera.

El único punto donde no se puede seguir esta filosofía es en la parte donde ambos robots deben realizar el movimiento síncrono. En este caso se usará un acuse de recibo para asegurar que ambos robots están ejecutando el programa, el TM5-900 solicita la orden de movimiento síncrono (se usa el servidor del UR5e), cuando el UR5e recibe la información confirma su recepción, momento en el cual el movimiento de ambos robots se desencadena. Para evitar que el PLC pueda parar el programa durante este movimiento se añade una variable auxiliar para indicar al PLC si es seguro realizar una parada ya que, al tener el PLC el control total sobre la ejecución de los programas, se podría realizar una parada que podría desencadenar en desperfectos en la pieza. Eso es debido a que si la ejecución del programa tras la pausa no se realiza al mismo tiempo podría darse el caso de que un robot intentase mover la pieza y el otro no.

- 2) Si se solicita finalizar el programa el robot se mueve a su posición de Home y el programa termina.

Cuando un robot termina el proceso de ensamblaje vuelve a la posición de inicio y vuelve a quedar a la espera de la orden del PLC, repitiéndose la secuencia hasta que el PLC solicite la finalización del programa. Este sistema asegurará el control sincronizado entre los robots, el PLC y la cinta, pero podrá verse interrumpido debido al sistema de seguridad (SICK scanner y seta de seguridad). Además, los sensores instalados en la cinta pueden determinar su parada o imposibilitar una determinada acción porque la cinta se encuentre ocupada ya que serán quienes determinen la información de los dispositivos sobre la cinta. La activación de la baliza estará sincronizada con el funcionamiento de la cinta.

A continuación se detallan aspectos relativos al diseño eléctrico, sistema de comunicaciones, integración con los robots, etc. En las siguientes secciones, además, se profundizará en la programación de los robots, el sistema de seguridad, la cinta y las variables de comunicación entre ellos. Se partirá de este flujo general planteado, pero podrá sufrir modificaciones a medida que avancen los desarrollos, considerando estas actividades como una especie de tareas previas a la integración y que serán reajustadas posteriormente en la tarea 4.5.

3.4.2.1 Diseño eléctrico y desarrollo del armario eléctrico

En esta tarea una de las actividades llevadas a cabo ha sido la de realizar el diseño eléctrico y desarrollo del armario eléctrico, revisando el armario utilizado en el Demostrador 2 y actualizándolo según los requisitos detectados en esta tarea. Esto supuesto la incorporación del panel táctil Siemens KTP400, la preparación del armario por la integración del SICK con el resto de componentes y la incorporación de otros elementos necesarios. En primer lugar se ha revisado la alimentación de los dispositivos (red trifásica 230 VAC para el motor de la cinta y red continua 24 VDC para el resto):

- El controlador responsable de enviar las señales de activación de movimiento de la cinta se alimenta a una tensión continua de 24 VDC. Los foto-detectores, distribuidos a lo largo de la cinta transportadora para detectar objetos, se alimentan a 24 VDC. También son alimentados a 24 VDC el HMI, el SICK scanner y la baliza de señalización.
- Los dispositivos encargados del control de la señal trifásica entregada al motor de potencia se alimentan a una tensión continua de 24 VDC, integrados en el variador.

Se ha integrado el HMI que servirá de interfaz en el Demostrador 3, realizando las modificaciones pertinentes en el armario. De hecho, ha sido necesario utilizar un armario eléctrico de mayores dimensiones disponible en las instalaciones. Se han revisado los relés de seguridad necesarios, como por ejemplo para impedir el movimiento de la cinta cuando el usuario pulsa la seta de emergencia. Cabe destacar que después de una parada de emergencia para desarmar la seta de emergencia el usuario debe hacer uso de la llave de desarme y el usuario debe pulsar el botón de rearme ubicado en la parte frontal del armario.

Por otra parte, se ha realizado el esquema del control donde se recogen todas las entradas que recibe el controlador de estado de la cinta transportadora y las señales de control. El controlador (PLC) recibe las señales de posición de cada detector y en función de las mismas envía la orden de movimiento o parada de la cinta. El variador es el encargado de regular la red trifásica acorde a la potencia necesaria por el motor, ésta acciona unos relés de estado acorde a la activación de la cinta. En el Demostrador 2 la salida de activación del motor Q0.0 del controlador está condicionada por el sistema de emergencia y la dirección de movimiento de la cinta es controlada a través de la salida de la CPU Q0.1. En este caso, a nivel bajo la cinta mantiene su dirección pre-configurada y en estado alto se considera cambio de sentido; por otra parte, la velocidad de la cinta es controlada a través de la entrada analógica de control 10VDC, que recibe la señal directamente del potenciómetro ubicado en el exterior del armario eléctrico. En cambio, en el Demostrador 3 la activación, velocidad y dirección de la cinta podrá ser seleccionada a través del HMI, podrá verse influenciada por el estado dado por el SICK y de los robots, por lo que será necesario realizar las modificaciones pertinentes en esta tarea y durante la integración. Por tanto, en la tarea T4.5 se explicará en detalle la integración del PLC, el SICK, los robots y la cinta.

La baliza de estado recibe las señales de estado directamente del variador, el verde fue asignado al pin correspondiente a la activación del motor y por contrario el rojo se conectó al pin de parada del motor. Las señales emitidas a la baliza necesitan estar referenciadas a 0V como el resto de dispositivos.

Con todo ello, en el Demostrador 3 se dispone inicialmente del siguiente armario eléctrico montado (Figura 122), disponiendo en su exterior de un pulsador de rearme y el HMI. Nótese que este ha sufrido una importante actualización con respecto al desarrollado en el Demostrador 2 (anualidad 2019).

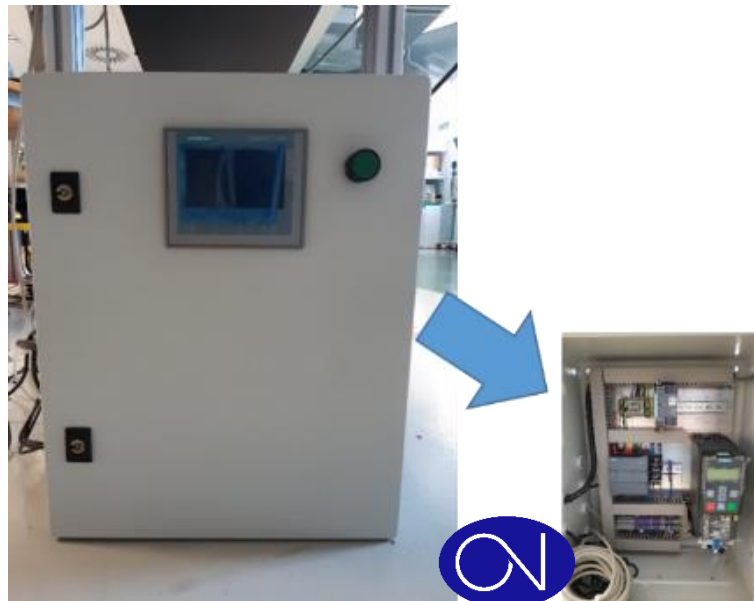


Figura 122. Armario eléctrico del Demostrador 3 – detalle (versión inicial)

3.4.2.2 Sistema de comunicaciones

Una vez configurada la seguridad del sistema y la cinta, el siguiente paso en la integración ha sido la configuración del sistema de comunicaciones y los robots. Primeramente se han realizado las modificaciones pertinentes en el armario eléctrico para incluir en la red de control al PLC y a los dos robots.

Durante la ejecución de las tareas del Demostrador 3, el PLC se encarga de gestionar el flujo de trabajo de los robots ejecutándose el proceso según unos parámetros iniciales seleccionados por el operador a través del HMI. Como se comentó en 3.4.2, el PLC se comunicará con el TM5-900 mediante Modbus TCP/IP y con el UR5e mediante Profinet, mientras que los robots se comunican entre ellos por medio de Modbus TCP/IP. Se realizó la configuración del sistema de comunicaciones del PLC en TIA Portal, creando un servidor y un cliente para acceder al servidor del Omron TM5-900 (Figura 123, izquierda y centro) y la conexión con UR5e (Figura 123, derecha). En el caso del TM5-900 para que el PLC pueda controlarlo debe estar en modo automático e inactivo, en el caso del UR5e debe encontrarse en modo de control remoto y se debe especificar la funcionalidad de las variables de Profinet a través del archivo de instalación del robot.

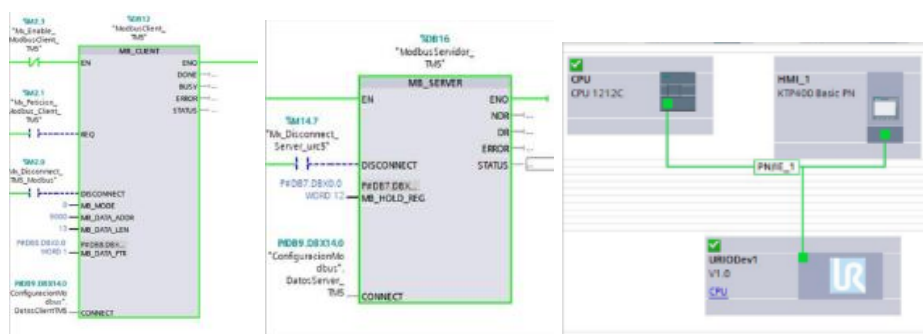


Figura 123. Definición en TIA Portal de las comunicaciones del PLC con los robots Omron TM5-900 (izqda. y centro) y UR5e (dcha.)

Teniendo en consideración el funcionamiento simplificado explicado en 3.4.2, el PLC da la orden de inicio de programa, y si todo está correcto el programa de ambos robots comienza, pasando de la

posición de Home a la posición de inicio, inicializando las garras y algunas variables de programa. Una vez los robots llegan a la posición de inicio se quedan a la espera de una orden del PLC, que puede ser un nuevo proceso de ensamblado o la finalización del programa. En el apartado 3.4.2.1 se habló del flujograma de los robots así como las variables utilizadas para la comunicación entre ambos robots. A continuación se detallan las variables de control y estado necesarias para la comunicación entre los robots y el PLC.

Para controlar el Omron TM5-900 se hacen uso de una serie de variables de control y estado del propio servidor del robot así como de una serie de variables específicas del proyecto en el espacio de memoria de variables de usuario:

- Señales de estado del robot. Se usan para saber si el robot está en estado de error, el programa está en ejecución, el usuario está editando algún programa, el proyecto está en pausa, otro dispositivo tiene el control sobre el robot, el robot se encuentra en modo colaborativo o si la seta de emergencia se ha pulsado (todas a excepción del control de la luz que no es necesaria).
- Señales para consultar proyecto seleccionado y cambiar a un nuevo proyecto.
- Señales para controlar la ejecución del programa. Se usa la señal de modo automático o manual para saber si se puede ejecutar el programa desde el PLC y la de Play/Pausa para ejecutar y pausar el programa.

En la Figura 124 se muestran las variables de comunicación entre el TM5-900 y el PLC definidas en el espacio de memoria de variables de usuario.

SERVIDOR MODBUS TM5-900			
Dirección	Tipo	Señal	Descripción
9001	DO	StopProgram	0: Si robot esperando orden y señal a 0 el robot sigue esperando orden. 1: Si robot esperando orden y señal a 1 el programa termina.
9002	DO	StartWorking	0: Si robot esperando orden y señal a 0 el robot sigue esperando orden. 1: Si robot esperando orden y señal a 1 comienza secuencia de ensamblado.
9003	DO	ReadyToStart	0: Robot no está esperando orden. 1: Robot esperando orden.
9004	DO	RobotWorking	0: La secuencia de ensamblado no se está ejecutando. 1: Secuencia de ensamblado en ejecución.
9020	DO	CanPause	0: Punto crítico donde se desaconseja pausar el robot. 1: El programa del robot puede pausarse sin problema.
9000	RO	RobotSpeed	Velocidad deseada en el robot en porcentaje (0-100).
9001	RO	RobotState	Estado del TM5-900: - 0: Estado por defecto. - 1: Inicializando. - 2: Esperando señal del PLC. - 3: Esperando señal del UR5e. - 4: Movimiento síncrono. - 5: Raspberry. - 6: Parte inferior de la carcasa. - 7: Tomando imagen.

Figura 124. Variables definidas para la comunicación robot Omron TM5-900 - PLC

El PLC también actúa como servidor de Modbus, pudiendo los clientes consultar información de los sensores o mover la cinta con las variables mostradas en la Figura 125.

SERVIDOR MODBUS PLC			
Dirección	Tipo	Señal	Descripción
00	RO	MoveConveyor	0: No se ha solicitado mover la cinta. 1: Se ha solicitado mover la cinta.
00	DI	LeftSensor	0: El sensor izquierdo no detecta objeto. 1: El sensor izquierdo está detectando un objeto. * Sensor de la izquierda visto desde el puesto de programación.
01	DI	MidSensor	0: El sensor central no detecta objeto. 1: El sensor central está detectando un objeto.
02	DI	RightSensor	0: El sensor derecho no detecta objeto. 1: El sensor derecho está detectando un objeto. * Sensor de la derecha visto desde el puesto de programación.
07	DI	MovingConveyor	0: La cinta no está en movimiento. 1: La cinta está en movimiento.

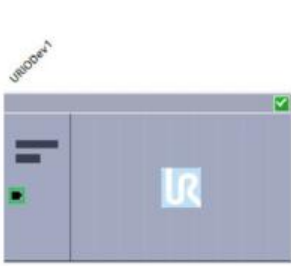
Figura 125. Variables definidas para la comunicación PLC – robot Omron TM5-900

Las variables de comunicación entre el UR5e y el PLC se muestran en la Figura 126.

PROFINET UR5e - PLC			
Dirección	Tipo	Señal	Descripción
00	GPbi	Pro_Brake_Start	El flanco de subida libera frenos e inicia programa de arranque.
01	GPbi	Pro_StartProg	El flanco de subida se usa como rearme tras pausa.
02	GPbi	Pro_PauseProg	El flanco de subida pausa el programa.
03	GPbi	Pro_StopProg	0: Si robot esperando orden y señal a 0 el robot sigue esperando orden. 1: Si robot esperando orden y señal a 1 el programa termina.
04	GPbi	Pro_StartWorkin	0: Si robot esperando orden y señal a 0 el robot sigue esperando orden. 1: Si robot esperando orden y señal a 1 comienza secuencia de ensamblado.
05	GPbi	Pro_LeftSensor	0: El sensor izquierdo no detecta objeto. 1: El sensor izquierdo está detectando un objeto. * Sensor de la izquierda visto desde el puesto de programación.
06	GPbi	Pro_MidSensor	0: El sensor central no detecta objeto. 1: El sensor central está detectando un objeto.
07	GPbi	Pro_RightSensor	0: El sensor derecho no detecta objeto. 1: El sensor derecho está detectando un objeto. * Sensor de la derecha visto desde el puesto de programación.
08	GPbi	Pro_MovingConv	0: Cinta transportadora parada. 1: Cinta transportadora en movimiento.
09	GPbi	Pro_FreeDrive	Cuando esta variable está activa el robot entra en estado de movimiento libre.
00	GPii	Pro_RobotSpeed	Velocidad deseada en el robot en porcentaje (0-100).
00	GPbo	Pro_RdyToStart	0: Robot no está esperando orden. 1: Robot esperando orden.
01	GPbo	Pro_RobotWorkin	0: La secuencia de ensamblado no se está ejecutando. 1: Secuencia de ensamblado en ejecución.
02	GPbo	Pro_CanPause	0: Punto crítico donde se desaconseja pausar el robot. 1: El programa del robot puede pausarse sin problema.
00	GPio	Pro_RobotState	Estado del UR5e: - 0: Estado por defecto. - 1: Inicializando. - 2: Esperando señal del PLC. - 3: Esperando señal del TMS-900. - 4: Movimiento síncrono. - 5: Soporte. - 6: Carcasa superior. - 7: Tornillos. - 8: Ensamblado finalizado. - 9: Depositando pieza.

Figura 126. Variables definidas para la comunicación robot UR5e – PLC

La Figura 127 corresponde al archivo que vincula el robot UR5 y el PLC a través de la comunicación PROFINET. A la derecha de la imagen se pueden visualizar las tarjetas de comunicación configuradas para el robot UR5 y el direccionamiento de variables compartidas. Cada tarjeta de red tiene su propia funcionalidad y espacio (“mapeado PROFINET”).



Módulo	Rack	Slot	Dirección	Dirección
UR5Dev1	0	0		
Interface	0	0 X1		
1_T2O State_1	0	1	1...32	
2_T2O IO_1	0	2	68...135	
3_T2O Joints_1	0	3	136...239	
4_T2O TCP_1	0	4	240...315	
5_T2O General Purpose Bit R...	0	5	33...40	
6_T2O General Purpose Int R...	0	6	316...411	
7_T2O General Purpose Floa...	0	7	412...507	
8_O2T Robot IO_1	0	8		1...24
9_O2T General Purpose Regi...	0	9		25...124
10_O2T General Purpose Re...	0	10		125...224

Figura 127. Archivo que vincula el robot UR5 y el PLC a través de la comunicación PROFINET

Una vez definidas estas variables en los robots, así como las variables para la comunicación entre ambos (mostradas en la Figura 135), se realizó su integración, validando primeramente con un PLC simulado los cambios de estado de las variables para posteriormente realizar la integración del conjunto con el PLC real como se detallará en próximos apartados.

3.4.2.3 Sistemas de seguridad

En el caso del Demostrador 3, al igual que para el Demostrador 2, se han utilizado distintos elementos de seguridad, tales como el set-up o instalación – incluyendo avisos en el suelo y a incorporación del escáner SICK para protección de la zona de peligro –, configuración de los robots y el sistema completo, una breve formación de los operarios, incluyendo las indicaciones dadas por medio de la interfaz.

Todos estos elementos, deben de tenerse en cuenta a la hora de desarrollar el sistema de sincronización de todo el conjunto. En este sentido, los diferentes sistemas de seguridad mencionados se incluirán dentro del sistema de sincronización que asegure el correcto funcionamiento de todo él. En este sentido, el detalle de las actividades realizadas para lograrlo, se realiza dentro de la tarea 4.5 sobre integración, pruebas y validación (apartado 3.4.5).

3.4.2.4 Sincronización del robot Omron TM5-900

Primeramente, para validar el planteamiento del flujograma diseñado en la Figura 134 y Figura 136 se ha creado una primera versión del programa en TMFlow para validar el flujo del proceso. Por este motivo se han implementado los movimientos pero sin coger realmente objetos a los que se hace mención durante el texto, validando así unitariamente cada paso y posteriormente en conjunto. Los ajustes en las tareas de pick&place adaptadas a los objetos se hará en la programación final durante las pruebas y validación final, que es donde se mostrarán las imágenes del proceso, mostrando aquí unas imágenes como ejemplo representativo de la programación del movimiento síncrono primeramente cada robot por separado (Figura 129 y Figura 131) y por último, sincronizando ambos (Figura 133).

Para la validación del programa base del Omron TM5-900 las señales que le manda el UR5e por Modbus se han forzado con ayuda de un programa auxiliar creado en el UR5e para ese fin. Las señales del PLC se han simulado con ayuda del programa “EasyModbus” en modo servidor ejecutándose en un portátil dentro de la red. De esta forma, el programa base del TM5-900 sigue los siguientes pasos, mostrando en las Figura 128 y Figura 129 un ejemplo representativo:

1) Ensamblaje:

- El robot se mueve de su posición de Home (posición en la que se mantienen los robots cuando no están ejecutando un programa) a la de inicio (posición a la que vuelve el robot tras terminar de ensamblar un dispositivo de la Demo 3).
- El robot quedará a la espera de que el PLC le dé la orden de ensamblar un nuevo dispositivo (simulado con EasyModbus).
- Una vez el robot recibe la orden desde el PLC simulado, recoge la Raspberry y la lleva al punto de inserción.
- El robot informa a través de su servidor de Modbus que tiene la Raspberry en el punto de inserción, el UR5e al leer dicha información procederá a insertar el soporte en la Raspberry, informando a través de su propio servidor de Modbus cuándo ha terminado. La variable se ha forzado con ayuda del programa auxiliar del UR5e.
- Una vez el TM5-900 recibe la información, suelta la Raspberry y se dirige a recoger la parte inferior del dispositivo.
- Tras recoger la parte inferior el TM5-900 comprueba que el UR5e ha vuelto a la posición de inserción (al insertar el soporte en la Raspberry el UR5e se mueve ligeramente, aprovechando el momento en el que el TM5-900 se dirige a recoger la parte inferior para regresar a la posición de inserción). Para continuar con el programa se fuerza mediante el programa auxiliar del UR5e la variable con la que informa de que se encuentra en la posición de inserción.

- Si el UR5e ya está en la posición de inserción el TM5-900 mueve la parte inferior hasta la posición de inserción, en caso contrario se ha producido un error y el TM5-900 queda a la espera para evitar que los dos robots choquen.
- Cuando el TM5-900 llega a la posición de inserción informa mediante su servidor Modbus de que la pieza inferior está en posición y el UR5e se encargará de insertar el conjunto soporte + Raspberry en la parte inferior de la carcasa. Cuando el UR5e termina la inserción informa a través de su servidor Modbus. Se fuerza la variable con ayuda del programa auxiliar.
- Cuando el TM5-900 detecta que la parte inferior ha sido insertada informa al UR5e de que quiere realizar el movimiento síncrono, quedando a la espera de que el UR5e baje la variable para realizar el movimiento. Cuando el UR5e recibe la orden de movimiento síncrono baja la variable y comienza a moverse. Se fuerza la bajada de la variable con ayuda del programa auxiliar.
- Cuando el TM5-900 detecta que la variable ha bajado comienza el movimiento síncrono para cambiar la orientación del dispositivo.

2) Atornillado:

- El TM5-900 queda a la espera de que el UR5e inserte la parte superior de la carcasa, los cuatro tornillos, y sujete el dispositivo. Una vez el UR5e se encuentra sujetando el dispositivo informa de ello a través de su servidor Modbus. Se fuerza la variable con ayuda del programa auxiliar del UR5e.

3) Inspección:

- Cuando el TM5-900 detecta que el UR5e ya está sujetando el dispositivo lo suelta y deja libre la cinta.
- Una vez el TM5-900 libera la cinta, informa a través de su servidor de Modbus. Cuando el UR5e detecta que la cinta está libre depositará el dispositivo sobre la cinta y vuelve a su posición de inicio, informando en su servidor de Modbus cuando deja libre la cinta. Se fuerza la variable con ayuda del programa auxiliar.
- Cuando el TM5-900 detecta que la cinta está libre se posiciona sobre el dispositivo y toma una imagen para la trazabilidad del producto. La imagen se guarda en el disco SSD de Omron (TMROBOT_VisionImages), considerando a la hora de guardar las imágenes el proyecto en cuestión, el nombre del nodo de visión y la fecha/hora de ejecución.
- Cuando el TM5-900 toma la imagen solicita al PLC el movimiento de la cinta.
- Finalmente, el TM5-900 vuelve a la posición de inicio.

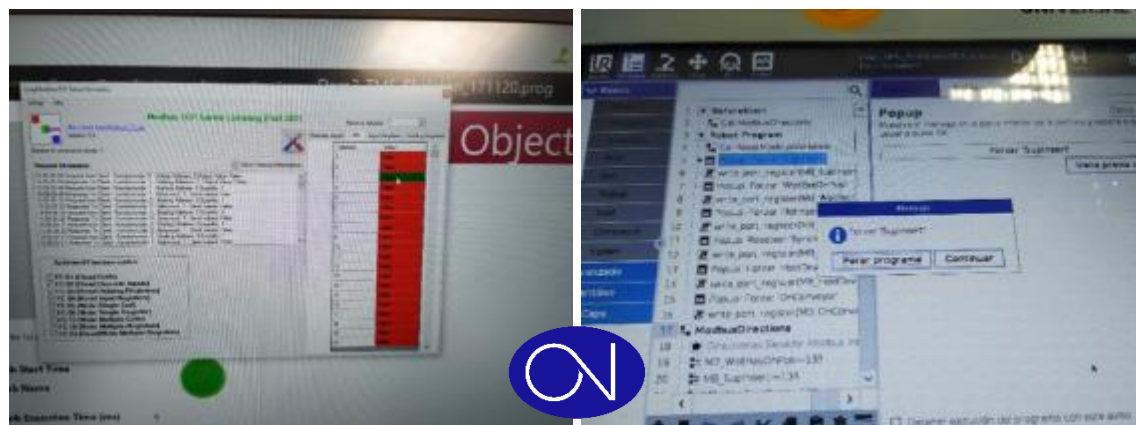


Figura 128. Ejemplo representativo de la simulación del PLC con el programa EasyModbus (izqda.) simulando la orden de ensamblar un nuevo dispositivo y el programa auxiliar del UR5e (dcha.) en el momento de informar de que ha terminado de insertar el soporte en la Raspberry



Figura 129. Ejemplo representativo de pruebas con el programa base del Omron TM5-900 – paso de realización del movimiento síncrono

Una vez validado el programa base, se procede a la integración con el PLC como se detalla en próximos apartados, validando la comunicación y sincronización de todo el sistema, de la misma forma que se ha hecho simulando los cambios de estado de las variables. La adaptación de las posiciones de agarre y dejada de cada componente, se verá en detalle en las pruebas y validación final.

3.4.2.5 Sincronización del robot UR5e

Como en el otro robot, para validar el planteamiento del flujograma diseñado en la Figura 134 y Figura 136 se ha creado una primera versión del programa con el Tech Pendant para validar el flujo del proceso. Por este motivo se han implementado los movimientos pero sin coger realmente objetos a los que se hace mención durante el texto, validando así unitariamente cada paso y posteriormente en conjunto. Los ajustes en las tareas de pick&place adaptadas a los objetos se hará en la programación final durante las pruebas y validación final.

Para la validación del programa base del UR5e las señales que le manda el TM5-900 por Modbus se han forzado con ayuda de un programa auxiliar creado en el TM5-900 para ese fin, las señales del PLC se han simulado con ayuda del programa “EasyModbus” en modo servidor ejecutándose en un portátil dentro de la red. De esta forma, el programa base del UR5e sigue los siguientes pasos, mostrando en las Figura 130 y Figura 131 un ejemplo representativo:

1) Ensamblaje:

- En primer lugar, el robot se mueve de su posición de Home (posición en la que se mantienen los robots cuando no están ejecutando un programa) a la de inicio (posición a la que vuelve el robot tras terminar de ensamblar un dispositivo de la Demo 3).
- El robot quedará a la espera de que el PLC le dé la orden de ensamblar un nuevo dispositivo (simulado con ayuda de “EasyModbus”).
- El robot cogerá el soporte y lo llevará al punto de inserción.
- Una vez en el punto de inserción el UR5e se mantiene a la espera de que el TM5-900 posicione la Raspberry en el punto de inserción. Se fuerza la variable con ayuda del programa auxiliar del TM5-900.
- Cuando el UR5e detecta que la Raspberry está en posición inserta el soporte en ella (se mueve ligeramente hacia adelante hasta que las piezas encajan).
- Una vez insertado el soporte, el UR5e informa de ello a través de su servidor Modbus. Cuando el TM5-900 lo detecta, libera el conjunto e informa a través de su servidor Modbus, simulando la variable con el programa auxiliar del TM5-900.
- Cuando el UR5e detecta que el conjunto está libre se desplaza nuevamente a la posición de inserción (de esta forma los robots no colisionarán cuando el TM5-900 lleve la siguiente pieza al punto de inserción).
- Cuando el TM5-900 coloca la parte inferior de la carcasa en el punto de inserción informa a través de su servidor Modbus, forzando la variable con ayuda del programa auxiliar del TM5-900.
- Cuando el UR5e detecta que la parte inferior está en la posición de inserción inserta el conjunto soporte+Raspberry en ella.
- Una vez se ha insertado la parte inferior el UR5e queda a la espera de que el TM5-900 le solicite realizar el movimiento síncrono para cambiar la orientación del conjunto. Se fuerza mediante el programa auxiliar del TM5-900 la solicitud de movimiento síncrono.
- Cuando el UR5e detecta la solicitud de movimiento síncrono baja la variable para informar al TM5-900 de que está procesando la orden y realiza el movimiento.
- Al terminar el movimiento síncrono el UR5e suelta el conjunto y recoge e inserta la parte superior.

2) Atornillado:

- Tras colocar la parte superior, el robot recoge y enrosca en el conjunto el primer tornillo.
- Recoge y enrosca el segundo tornillo.
- Recoge y enrosca el tercer tornillo.
- Recoge y enrosca el cuarto tornillo.
- Una vez insertados los cuatro tornillos el robot sujeta el dispositivo completo e informa de ello mediante su servidor Modbus. En este punto ambos robots se deberían encontrar sujetando el dispositivo. El UR5e queda a la espera de que el TM5-900 suelte el dispositivo y deje la cinta libre. Se fuerza la variable con ayuda del programa auxiliar del TM5-900.

3) Inspección:

- Cuando el UR5e detecta que el TM5-900 ha soltado el dispositivo y la cinta está libre deposita el dispositivo en la cinta siempre y cuando el sensor de posición central de la cinta no esté activado, y la cinta no se encuentre en movimiento (esta información se consulta al PLC).
- Finalmente, el UR5e se retira a su posición de inicio. Al liberar la cinta informa mediante su servidor Modbus para que el TM5-900 pueda entrar a tomar la imagen.

Para simplificar la programación inicialmente se ha empleado un registro por flag (depuración más sencilla ya que cada registro tiene el nombre del flag correspondiente y se puede ver su valor directamente añadiendo las variables de Modbus en la instalación del UR5e). Una vez terminado el programa se plantea almacenar el valor de cada flag en un determinado bit dentro de cada registro, de forma que cada registro pueda almacenar 16 flags (más eficiente).

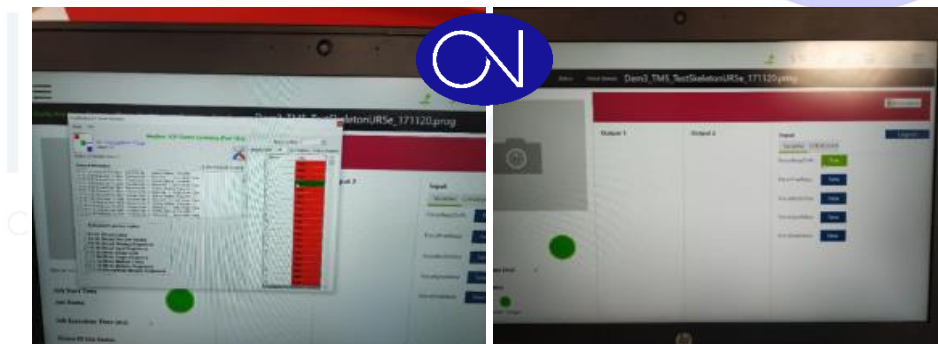


Figura 130. Ejemplo representativo de la simulación del PLC con el programa EasyModbus (izqda.) simulando la orden de ensamblar un nuevo dispositivo y el programa auxiliar del Omron TM5-900 (dcha.) en el momento de informar de que el TM5-900 ha posicionado la Raspberry en el punto de inserción



Figura 131. Ejemplo representativo de pruebas con el programa base del UR5e – paso de realización del movimiento síncrono

Una vez validado el programa base individualmente, se procede a la integración con el programa base del Omron TM5-900. De esta forma, al lanzarse a la vez, se observa la interacción real entre ambos sin necesidad de forzar variables, aunque se sigue simulando el PLC con EasyModbus, mostrando en las Figura 132 y Figura 133 un ejemplo representativo:

1) Ensamblaje:

- Inicialmente ambos robots se mueven a sus respectivas posiciones de inicio.
- Una vez los robots están en la posición de inicio se mantienen en ella hasta que el PLC los habilita (señal simulada con ayuda del servidor de prueba de EasyModbus).
- Una vez se habilitan ambos robots cada uno recoge la pieza correspondiente y la lleva a la posición de inserción. El TM5-900 recoge la Raspberry y el UR5e el soporte.
- Una vez ambos robots están en posición el UR5e inserta el soporte en la Raspberry.
- El UR5e informa de que ha insertado el soporte en la Raspberry y el TM5-900, que estaba a la escucha, se dirige a recoger la parte inferior de la carcasa.
- Mientras el TM5-900 ha ido a recoger la pieza inferior de la carcasa, el UR5e aprovecha a volver a la posición de inserción (para corregir la desviación que se produce cuando inserta el soporte en la Raspberry). Cuando el TM5-900 recoge la parte inferior y se dispone a llevarla al punto de inserción comprueba que el UR5e está en la posición de inserción, ya que en caso contrario se habrá producido un error y es posible que los robots colisionen si el TM5-900 lleva la pieza al punto de inserción. Si el UR5e informa de que se encuentra en

el punto de inserción el programa continúa, llevando el TM5-900 la pieza inferior al punto de inserción.

- Cuando el TM5-900 coloca la pieza inferior en el punto de inserción el UR5e introduce el conjunto soporte+Raspberry en ella e informa de ello. Cuando el TM5-900 detecta que la pieza inferior está insertada solicita al UR5e realizar un movimiento síncrono para cambiar la orientación del robot. Cuando el UR5e recibe la solicitud de movimiento síncrono baja la variable y comienza el movimiento. Cuando el TM5-900 detecta que la variable ha bajado comienza el movimiento.
- Una vez el movimiento síncrono termina el UR5e suelta el conjunto, recoge la parte superior de la carcasa y la inserta en el conjunto.

2) Atornillado:

- Tras colocar la parte superior, el UR5e recoge y enrosca el primer tornillo.
- El UR5e recoge y atornilla el segundo tornillo.
- El UR5e recoge y atornilla el tercer tornillo.
- El UR5e recoge y atornilla el cuarto tornillo.
- Una vez el UR5e ha insertado los cuatro tornillos sujeta el dispositivo e informa de ello a través de su servidor Modbus.

3) Inspección:

- Cuando el TM5-900 detecta que el UR5e se encuentra sujetando el dispositivo, suelta el conjunto y deja libre la cinta.
- Una vez el TM5-900 ha soltado el dispositivo y dejado libre la cinta, informa de ello a través de su servidor Modbus.
- Cuando el UR5e detecta que el TM5-900 ha dejado libre la cinta deposita el conjunto sobre la cinta transportadora (siempre y cuando el sensor central no detecte nada y la cinta no esté en movimiento).
- Una vez depositado el conjunto sobre la cinta el UR5e se dirige a su posición de inicio, en el momento que deja libre la cinta informa a través de su servidor Modbus, en ese momento, el TM5-900, que se encuentra a la escucha, se dirige a tomar una captura del dispositivo con fines de trazabilidad.
- Una vez el TM5-900 toma la captura solicita al PLC mover la cinta, siempre y cuando las fotocélulas de los extremos no detecten ningún objeto (simulado mediante EasyModbus).
- A continuación, el TM5-900 se dirige a su posición de inicio.
- En este momento ambos robots se encuentran en su posición de inicio, a la espera de que el PLC les mande insertar un nuevo dispositivo o terminar el programa.
- En caso de terminar el programa, los robots pasan de su posición de inicio a su posición de Home, y el programa de ambos termina.

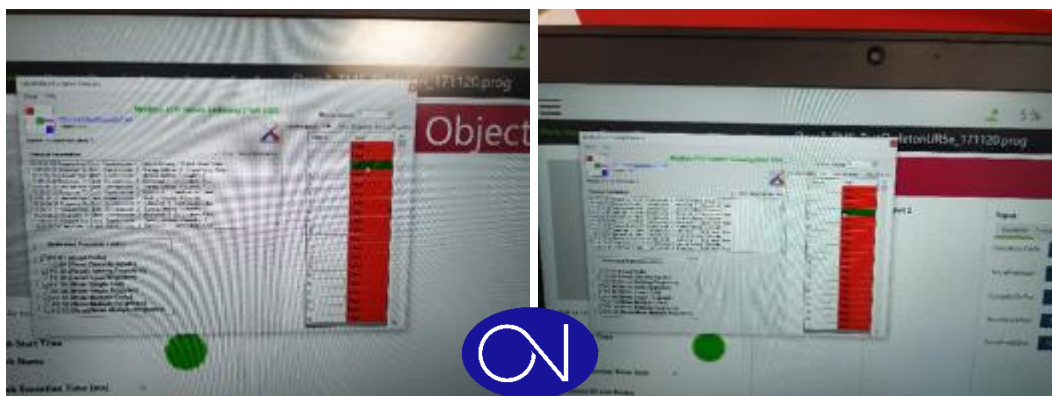


Figura 132. Ejemplo representativo de la simulación del PLC con el programa EasyModbus simulando la orden de ensamblar un nuevo dispositivo para cada uno de los robots



Figura 133. Ejemplo representativo de pruebas con los programas bases de ambos robots integrados – paso de realización del movimiento síncrono

Una vez validado el programa base de ambos robots, se procede a la integración con el PLC real (detallado en el próximo apartado), validando la comunicación y sincronización de todo el sistema, de la misma forma que se ha hecho simulando los cambios de estado de las variables. La adaptación de las posiciones de agarre y dejada de cada componente, se verá en detalle en las pruebas y validación final.

3.4.3 T4.3. Programación de los robots

En esta tarea, a partir de los desarrollos realizados en las tareas anteriores, se procede a programar los robots para garantizar una colaboración efectiva entre ellos. En primer lugar, se detallará el flujograma que deben seguir cada uno de los robots existentes en el demostrador y

posteriormente, se mostrarán los pasos llevados a cabo para realizar la programación de aquellas tareas más específicas del demostrador. Para la programación de los robots se utilizó el Teach Pendant para el UR5e y el entorno de programación TMFlow en el caso de Omron TM5-900. Respecto a los programas-base realizados en la tarea anterior existen una serie de modificaciones:

- Se dispone de un nodo de calibración manual en el que el TM5-900 emplea la cámara para localizar un landmark que le presenta el UR5e y en función del mismo recalcular las posiciones de interacción entre robots.
- Cuando el TM5-900 se encuentra en la posición de recogida de piezas emplea la cámara para localizar el landmark contenido en la misma y poder calcular en función de él la posición de recogida de las distintas piezas.
- Se han modificado varias posiciones, entre ellas la de recogida de piezas del TM5-900, de forma que sea más sencillo para el usuario colocar la bandeja con las piezas.
- Se hace uso del sensor de par-fuerza del UR5e para insertar las piezas y los tornillos.

3.4.3.1 Flujograma de los robots

En la Figura 134 se muestra el flujograma general de los pasos que deben seguir los robots en el desarrollo de este demostrador. El PLC será el encargado de ejecutar los programas de los robots. La programación del PLC e integración con los robots se detallará en la sección 3.4.5.

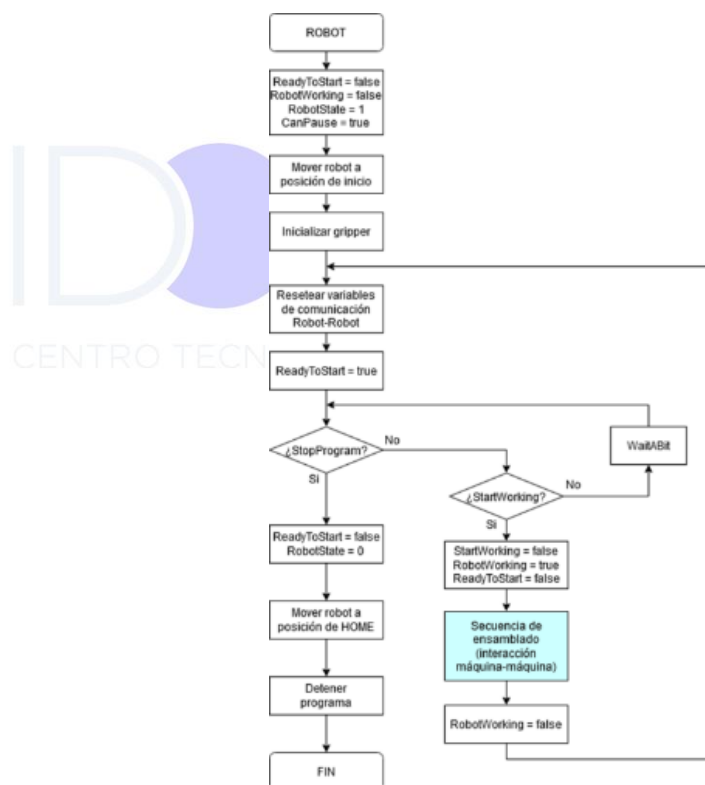


Figura 134. Flujograma general de los robots

Cuando los robots reciben la orden realizan un proceso de inicialización y pasan de la posición de “Home” a la posición de “inicio”, donde quedan a la espera de una señal del PLC para: a. Ensamblar un dispositivo; b. Terminar el programa. Si se elige la opción a) el bloque de “Secuencia de ensamblado” se ejecuta. En la Figura 136 se muestra el flujograma general de los pasos detallados de este bloque que deben seguir cada uno de los robots, de manera sincronizada, para lograr

realizar el montaje del dispositivo asociado a este Demostrador 3. Estos pasos están relacionados con las tres fases planteadas en el apartado 3.4.1.3: 1. Ensamblaje, 2. Atornillado y 3. Inspección y para la comunicación entre robots se utilizan las variables definidas en la Figura 135.

1) Ensamblaje:

- 1) Soporte y Raspberry (la PCB): si los dos robots reciben la orden de ensamblar un dispositivo comenzarán a moverse a la vez.
 - a. Aproximar Raspberry (TM5-900): el TM5-900 recogerá la Raspberry.
 - b. Aproximar soporte (UR5e): el UR5e recogerá el soporte de la carcasa.
 - c. Ambos se dirigirán al punto de ensamblado. Como el UR5e se encargará de la inserción del soporte, espera a que el TM5-900 le indique que se encuentra en posición ("RaspOnPos"), momento en el que inserta la pieza y lo indica ("SupInsert").
- 2) Inserción de Raspberry:
 - a. Insertar soporte (UR5e) y soltar Raspberry (TM5-900): cuando el UR5e inserta el soporte, el TM5-900 libera la Raspberry e informa de ello ("FreeRasp").
 - b. Mover a posición de ensamblaje (UR5e) y recoger parte inferior (TM5-900): cuando la Raspberry ha sido liberada, el UR5e vuelve a la posición de inserción (al insertar la pieza se habrá movido ligeramente), a la vez que el TM5-900 se dirige a recoger la parte inferior de la carcasa.
- 3) Carcasa inferior:
 - a. Aproximar parte inferior (TM5-900): el TM5-900 lleva la parte inferior de la carcasa a la posición de inserción e informa de ello ("BotOnPos").
 - b. Insertar parte inferior (UR5e): momento en el que el UR5e inserta el soporte y la raspberry en la parte inferior e informa al terminar ("BotInsert").
- 4) Se ha llegado al punto crítico del programa ya que ambos robots deben de moverse justo a la vez para cambiar la orientación del conjunto (soporte, Raspberry y pieza inferior), de forma que el UR5e quede libre y el TM5-900 sujetándolo.
 - a. El TM5-900 indicará al UR5e que quiere iniciar el movimiento sincronizado ("SyncMove"). Cuando el UR5e se encuentra a la espera de la orden de movimiento sincronizado y la recibe, resetea la variable ("SyncMove=0") y comienza a moverse.
 - b. Movimiento sincronizado en ambos robots: el TM5-900 comienza a moverse cuando detecta que la variable se ha reseteado ("SyncMove==0").
- 5) Carcasa superior: una vez el movimiento sincronizado termina, el UR5e recoge e inserta la parte superior sobre el dispositivo sujetado por el TM5-900:
 - a. Suelta el conjunto.
 - b. Se dirige a recoger la parte superior de la carcasa.
 - c. La lleva hasta el punto de inserción.
 - d. La inserta en el conjunto.

2) Atornillado:

- 6) Recoger e insertar tornillo x4 (UR5e): una vez insertada la parte superior el UR5e se dirige a por un tornillo, lo lleva hasta el punto de inserción y lo atornilla, repitiendo el proceso cuatro veces, una por tornillo.

3) Inspección:

- 7) Sujetar dispositivo (UR5e): el UR5e sujeta el dispositivo e informa de ello ("HoldDevice").
- 8) Soltar dispositivo (TM5-900): cuando el TM5-900 detecta que el UR5e está sujetando el dispositivo lo suelta e informa de ello ("FreeDevice").
- 9) Depositar en cinta (UR5e): si la cinta está disponible ("ConveyorRdy"), cuando el UR5e detecta que el TM5-900 ha soltado el dispositivo lo deposita en la cinta transportadora (de esta forma la parte superior del dispositivo queda mirando hacia arriba), e informa de ello ("OnConveyor").
- 10) Trazabilidad (TM5-900): cuando el TM5-900 detecta que el dispositivo está sobre la cinta transportadora captura una imagen de trazabilidad e informa al PLC de que debe llevarse la pieza ("MoveConveyor").

SERVIDOR MODBUS TM5-900			
Dirección	Tipo	Señal	Descripción
9005	DO	RaspOnPos	0: La raspberry aún no está en posición de ensamblado. 1: Raspberry en posición de ensamblado.
9006	DO	WaitSupInsert	0: Espera no iniciada/terminada. 1: Esperando que el soporte se inserte en la Raspberry.
9007	DO	FreeRasp	0: El gripper no ha liberado la Raspberry. 1: El gripper ha liberado la Raspberry.
9008	DO	BotOnPos	0: La pieza inferior aún no está en posición de ensamblado. 1: Pieza inferior en posición de ensamblado.
9009	DO	WaitBotInsert	0: Espera no iniciada/terminada. 1: Esperando que la pieza inferior se inserte correctamente.
9010	DO	WaitHoldDevice	0: Espera no iniciada/terminada. 1: Esperando que el otro robot sujete el dispositivo.
9011	DO	FreeDevice	0: TM5-900 aún se encuentra sujetando el dispositivo. 1: TM5-900 ha liberado el dispositivo.
9012	DO	WaitOnConveyor	0: Espera no iniciada/terminada. 1: Esperando a que el dispositivo se encuentre sobre la cinta.

SERVIDOR MODBUS UR5e			
Dirección	Tipo	Señal	Descripción
133	RO	WaitRaspOnPos	0: Espera no iniciada/terminada. 1: Esperando que la Raspberry se encuentre en posición de ensamblado.
134	RO	SupInsert	0: El soporte aún no se ha insertado en la Raspberry. 1: Soporte insertado en la Raspberry.
135	RO	WaitFreeRasp	0: Espera no iniciada/terminada. 1: Esperando que el gripper del otro robot libere la Raspberry.
136	RO	WaitBotOnPos	0: Espera no iniciada/terminada. 1: Esperando que la parte inferior se encuentre en posición de ensamblado.
137	RO	BotInsert	0: La parte inferior aún no se ha insertado en el conjunto. 1: La parte inferior se ha insertado en el conjunto.
138	RO	HoldDevice	0: UR5e aún no está sujetando el dispositivo. 1: UR5e sujetando el dispositivo.
139	RO	WaitFreeDevice	0: Espera no iniciada/terminada. 1: Esperando que el otro robot libere el dispositivo.
140	RO	OnConveyor	0: El dispositivo aún no se ha colocado sobre la cinta transportadora. 1: Dispositivo colocado sobre la cinta transportadora.
141	RO	SyncMove	0: Movimiento sincronizado no solicitado. 1: Solicitar movimiento sincronizado del robot. * El robot lo devuelve a cero al comenzar el movimiento.
142	RO	WaitSyncMove	0: Espera no iniciada/terminada. 1: Esperando orden de movimiento sincronizado.
143	RO	WaitConveyorRdy	0: Espera no iniciada/terminada. 1: Esperando que la cinta transportadora esté lista.

Figura 135. Variables definidas para la comunicación robot - robot

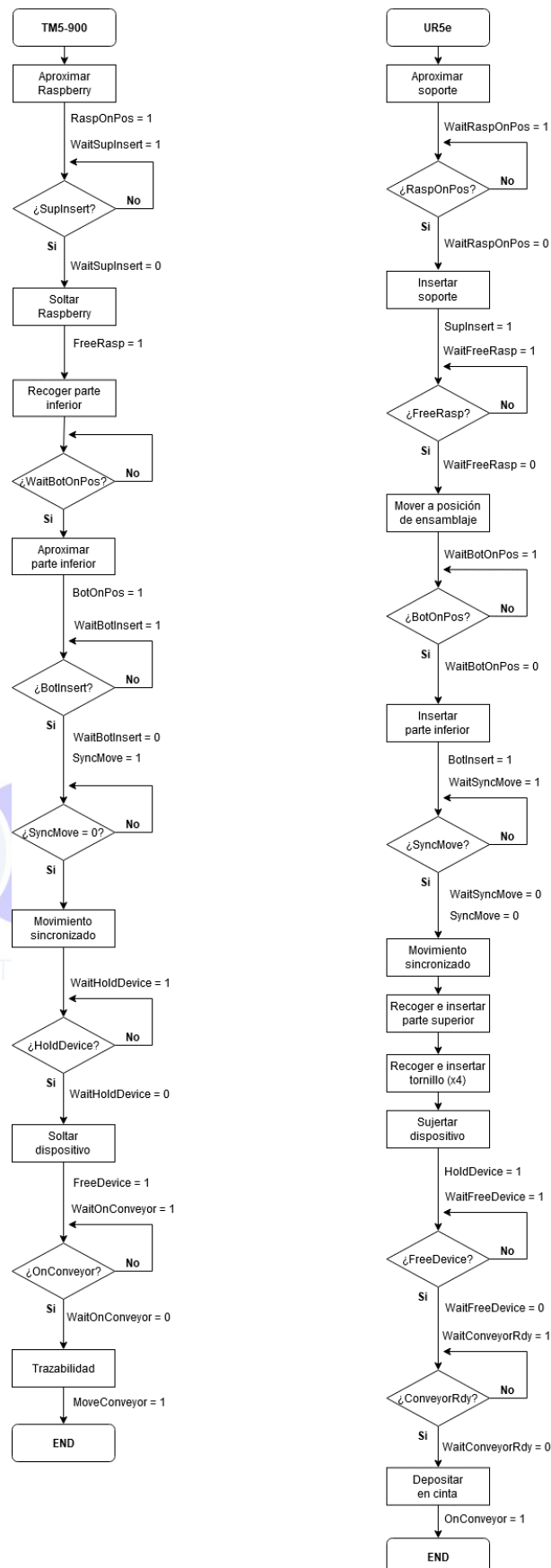


Figura 136. Flujograma general de la secuencia de ensamblaje, atornillado e inspección de los robots

En los siguientes apartados se detalla la programación de pasos clave dentro de dicho flujograma, como es la comunicación entre ambos manipuladores y el proceso de atornillado para el montaje final de los dispositivos.

3.4.3.2 Pruebas de comunicación entre UR5e y OMRON TM5-900

Como ya se ha indicado, en el Demostrador 3 se aborda el ensamblado de un dispositivo electrónico mediante el uso de dos robots colaborativos (un UR5e y un Omron TM5-900). Para que ambos robots puedan coordinar sus movimientos de una forma flexible y eficiente es necesario establecer una comunicación robusta entre ambos. En este apartado se abordan las pruebas de comunicación realizadas inicialmente para garantizar su correcto funcionamiento de cara a su implementación posterior en el programa del Demostrador 3.

Los principales protocolos de comunicación industrial que incorpora por defecto el UR5e son Modbus TCP, IP/Ethernet y ProfiNet, mientras que los del TM5-900, son Modbus TCP / RTU. Modbus TCP es un protocolo de comunicación muy extendido en la industria por ser un protocolo abierto y de fácil implementación, que máquinas de diversos fabricantes suelen tener como denominador común. Como ambos robots cuentan con Modbus TCP/IP se hará uso de dicho protocolo para la comunicación máquina-máquina del Demostrador 3.

Para la configuración, se debe tener en cuenta que Modbus TCP es un protocolo construido sobre TCP/IP donde un servidor atiende las solicitudes de uno o varios clientes. Los controladores de ambos robots pueden configurarse como servidor de Modbus TCP, además permiten crear clientes de Modbus TCP que podrán iniciar comunicación con servidores de Modbus TCP. El primer paso para poder establecer una comunicación es comprobar que ambos robots se encuentran dentro de una misma red y pueden “verse” entre sí, en caso de que solo se necesite la comunicación entre los dos robots puede conectarse directamente un cable ethernet desde la controladora del UR5e y la del TM5-900, en nuestro caso, como la red incluye más dispositivos, hacemos uso de un switch y router. Posteriormente, a cada uno de los robots se le asigna una dirección de red y una vez realizada esta acción se realiza la configuración de red en ambos robots.

Para poder comprobar el correcto intercambio de información entre ambos robots, se han realizado dos programas de mímica, uno que permite al UR5e copiar los movimientos del TM5-900, y otro que permite al TM5-900 copiar los movimientos del UR5e.

Para poder realizar estos programas el primer paso es determinar la información que necesita un robot para poder imitar al otro, en este caso, aprovechando que ambos robots tienen una estructura mecánica similar, se ha decidido emplear el ángulo de cada una de las articulaciones (hay que tener en cuenta por tanto que los puntos en los que se encuentra el TCP del robot en cada instante no serán iguales respecto a sus bases, dependerá de la configuración espacial de cada robot).

El siguiente paso es comprobar si los servidores de Modbus TCP de cada robot cuentan con la información de interés y como se codifica dicha información en cada uno de ellos:

- En el caso del UR5e podemos acceder a los valores absolutos de cada una de las articulaciones en las direcciones.
- En el caso del TM5-900 se puede comprobar que la información de cada ángulo aparece codificada en dos registros consecutivos, cada registro tiene 16 bits por lo que la

información se codifica en 32 bits, para obtener el valor de interés será necesario hacer la conversión de 32 bits a un float32 (información en grados).

A continuación, al no estar trabajando con robots de la misma marca, estos no tienen que compartir necesariamente las mismas referencias (cosa que pasa realmente). Para poder establecer una relación en los ángulos de las articulaciones de ambos robots es útil comprobar la posición cero de cada uno de ellos y ver si coinciden.



Figura 137. Posición cero del UR5e (izqda.) y del TM5-900 (dcha.).

Se comprueba rápidamente que las articulaciones 1, 2 y 4 presentan un desfase de 90° entre un robot y otro. Lo siguiente que hay que observar es si la dirección positiva que toma un robot como referencia para cada articulación se corresponde con la que toma el otro robot. En el caso del UR5e en la siguiente imagen se ha dibujado una flecha en cada articulación, el sentido positivo de giro se puede obtener aplicando la regla de la mano derecha sobre dichas flechas (Figura 139).

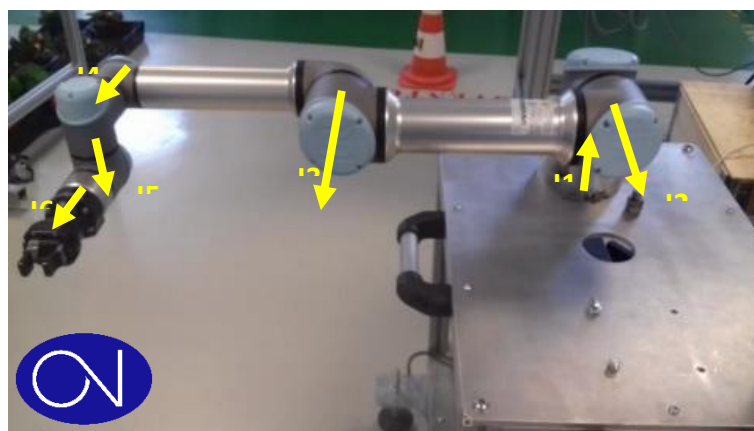


Figura 138. Articulaciones del robot UR5e y direcciones

Se observa en la Figura 139 la dirección positiva que toma el TM5-900 como referencia para cada articulación.



Figura 139. Articulaciones del robot TM5-900 y direcciones

En base a las imágenes anteriores se puede concluir que las articulaciones 2, 3 y 4 de ambos robots tienen sentidos de giro positivo opuestos. Con dicha información se determinan las ecuaciones que relacionan cada uno de los ángulos de las articulaciones de cada robot. Además, a la hora de ejecutar los programas hay que tener en cuenta otra diferencia entre ambos robots, los ángulos de giro permitidos para cada uno tampoco son iguales. El UR5e puede mover todas sus articulaciones en el intervalo $[-360^\circ, 360^\circ]$ mientras que el TM5-900 está más limitado en función de la articulación referida (por ejemplo, $J1 \rightarrow [270^\circ, -270^\circ]$, $J2 \rightarrow [180^\circ, -180^\circ]$, etc.). Por ello, se deben establecer una serie de restricciones de los ángulos de las articulaciones para limitar los movimientos del UR5e en el TM5e-900. En caso contrario el TM5-900 se bloqueará al intentar llevarlo a una posición que no puede alcanzar.

Como se ha comentado, para validar la comunicación entre ambos robots se han realizado dos programas de mímica, uno que permite al UR5e copiar los movimientos del TM5-900 y otro al revés. El primer caso es un poco más complejo de implementar, pero los resultados son mucho mejores ya que el UR5e es capaz de seguir los movimientos del TM5-900 de forma fluida. A continuación se explican en detalle las pruebas realizadas.

3.4.3.2.1 UR5e replicando al TM5-900

El principal problema en este caso es la diferencia en el tratamiento de la información de ambos robots, en el caso del UR5e los registros que interesan para esta aplicación almacenan por un lado el ángulo de la articulación en valor absoluto, y por otro lado un registro que permite saber si el ángulo es positivo o negativo. El TM5-900 por el contrario almacena la información en dos registros consecutivos de 16 bits que puede transformar directamente en un float de 32 bits. El UR5e no es

capaz de transformar la información almacenada en dos registros consecutivos en un float32 de forma directa, por tanto, se ha optado por crear un servidor con ayuda de Visual Studio Code, que se encuentra a la escucha del UR5e. A cada registro se le ha dado un nombre donde se puede comprobar a que articulación pertenece y si el registro contiene los bits más significativos o los bits menos significativos. En el UR5e se añade una unidad Modbus que se configura para comunicarse con el servidor Modbus TCP del TM5-900, y se añaden las direcciones de la información de interés del servidor de Modbus TCP del TM5-900 a la lista de señales. El UR5e leerá los registros de interés del servidor Modbus TCP del TM5-900 y enviará la información en un array al servidor. El servidor, cuando recibe la información, la procesa y la envía de vuelta al UR5e en un formato que pueda entender, actualizando un array global que se pueda acceder desde el programa principal. Una vez el UR5e obtiene la información en un formato que puede entender, se ha hecho uso de una función de URScript que permite realizar un movimiento fluido de las articulaciones del robot, aun cuando la información llega de forma discreta.

En el programa principal inicialmente se establece comunicación con el servidor (definiendo la IP y el puerto), se mueve el robot con ayuda de un movej a la primera posición obtenida y, a partir de ahí se entra en un bucle encargado de llamar a la función **“servoj”** con la información de las articulaciones, que permite mover el UR5e de forma fluida. Una vez se conecta un cliente se decodifica la información (que llega en un string). Se aplica la conversión de ángulos y se aprovecha para pasarlos a radianes, ya que es la información que procesa el comando usado para mover el UR5e. Se codifica la información en un string que el UR5e pueda entender y se envía la información procesada al cliente (UR5e).

En la siguiente imagen se muestran diversas instantáneas en las el robot UR5e imita los movimientos del robot TM5-900. A este último, el operario le mueve libremente con sus manos.

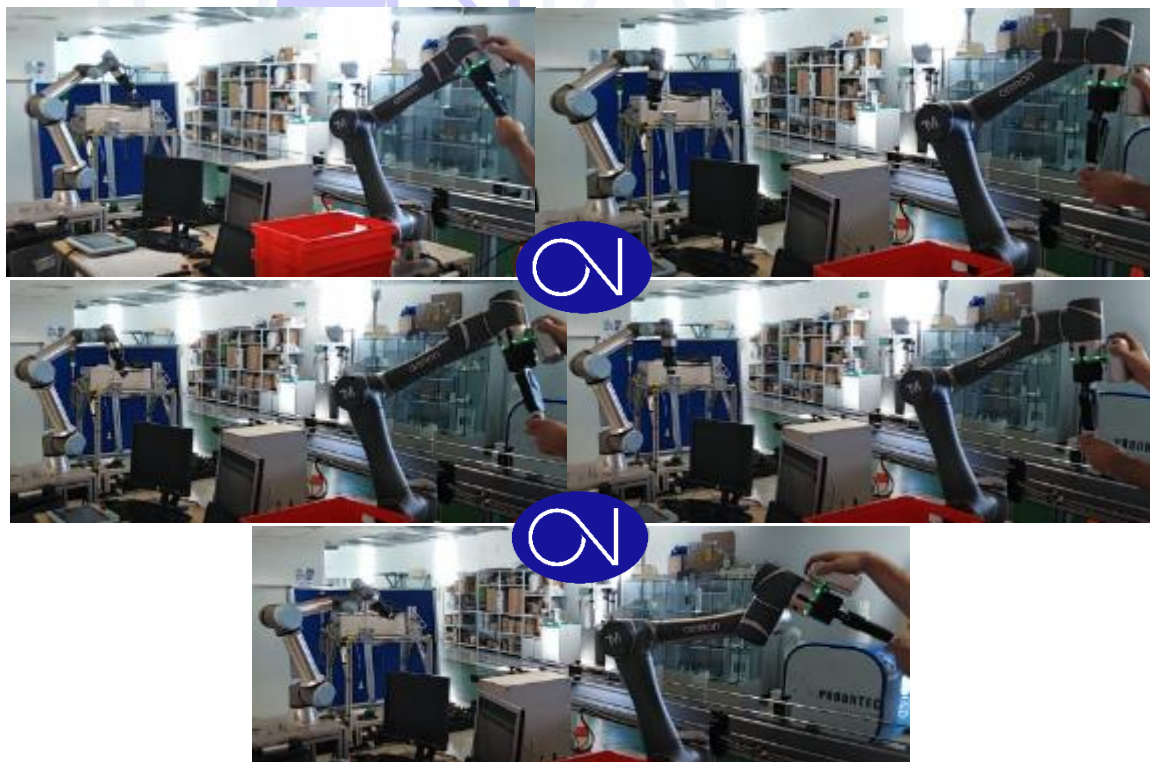


Figura 140. Pruebas mímica UR5e – TM5-900

3.4.3.2.2 TM5-900 replicando al UR5e

En este caso es importante destacar que el UR5e puede llegar a puntos que el TM5-900 no puede. Por tanto, si se lleva el UR5e a uno de esos puntos, el TM5-900 no podrá alcanzarlo y se bloqueará, para evitarlo hay que asegurar que los ángulos de todas las juntas del UR5e se encuentran en todo momento dentro del intervalo de ángulos válido calculado al final del apartado anterior.

Como se ha activado anteriormente el servidor Modbus TCP del UR5e, únicamente se necesita crear un programa en el TM5-900 que se comuniqué con él para obtener la información necesaria, que la decodifique y la transforme en información que pueda usar directamente, y que finalmente emplee dicha información para mover sus articulaciones a las posiciones de interés.

Como se ha comentado previamente, para conocer la información que se puede obtener del servidor Modbus TCP del UR5e se puede acceder a su tabla de Modbus, en la que se puede encontrar la información de interés en las siguientes direcciones:

Una vez localizada la información de interés, se crea un cliente de Modbus TCP en el proyecto del TM5-900 que acceda al servidor del UR5e, tal y como se detalló anteriormente. Asimismo, se añaden las señales de interés para simplificar posteriormente el acceso a su valor (todas configuradas como registros de entrada en este caso).

El TM5-900 es capaz de decodificar la información previa, no obstante, no se ha encontrado ninguna función que permita mover el robot de una forma suavizada a partir de información discreta de sus articulaciones. Por ello se ha decidido hacer uso de un nodo "Move", que permite realizar un movimiento de las articulaciones indicando el ángulo que se desea girar en función del ángulo actual de las articulaciones, como consecuencia, al mover el UR5e, el TM5-900 se moverá de forma no fluida, es decir, obtendrá los valores deseados, realizará los cálculos oportunos, y moverá el robot a dicha posición. Si durante ese proceso se sigue moviendo el UR5e el TM5-900 no lo tendrá en cuenta, solo volverá a recalcular la posición una vez alcance la posición previa.

Para ello, se ha comenzado creando un hilo encargado de leer los datos de interés del servidor Modbus TCP y transformarlos en información útil (Figura 141). El hilo consta de dos partes (Figura 141), analizándose el signo del dato en la primera parte. Si éste es positivo se accede al registro que codifica el valor del ángulo, se hace la conversión de miliradianes a grados y se almacena el valor en la posición correspondiente de un array habilitado a tal efecto. Si el ángulo es negativo se sigue el mismo procedimiento con la salvedad de que al valor almacenado en el array se le deben de restar 360°. En la siguiente parte se transforma la información anterior en información que el TM5-900 pueda usar directamente para mover sus propias articulaciones. Para ello, en primer lugar se aplica la conversión de grados que se calculó al inicio del apartado. El siguiente paso es calcular la diferencia existente entre la posición actual de las articulaciones del robot (para lo cual se accede al servidor Modbus TCP del propio TM5-900), y la posición deseada (obtenida del servidor Modbus TCP del UR5e y procesada para hacerla entendible por el TM5-900). Esta información se almacena en un array que se consultará desde el programa principal.

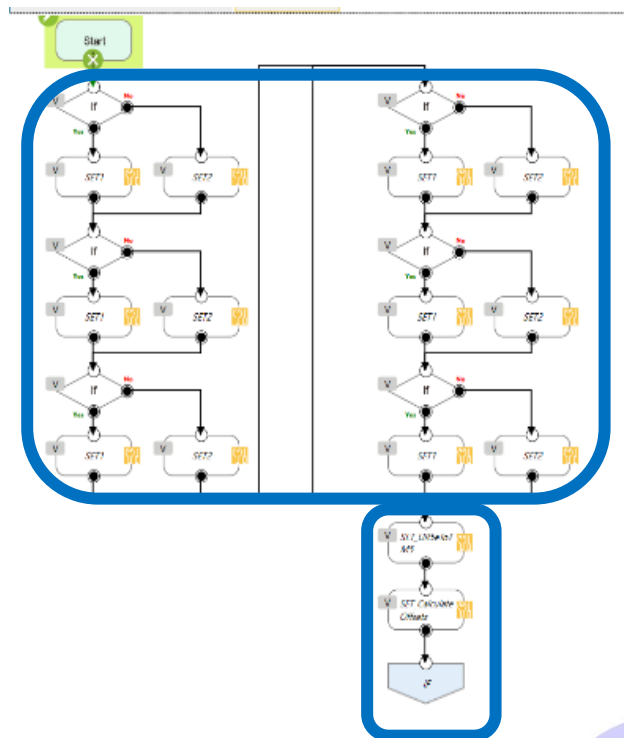


Figura 141. Hilo para lectura de datos

Posteriormente, el programa principal utiliza la información obtenida en el hilo para mover el robot a las posiciones de interés (Figura 142). En el programa principal se crea un bucle donde cada iteración se almacena el valor deseado para las articulaciones en variables independientes (el nodo “Move” no admite arrays). Y se añade un nodo Move configurado como movimiento de articulación al que se le pasa la información previa.

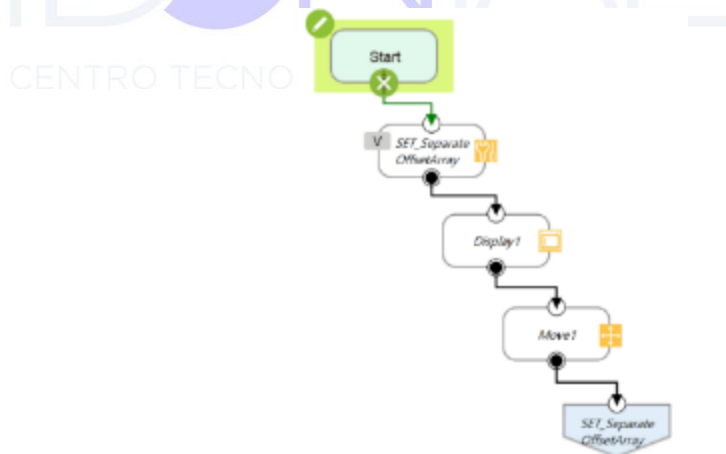


Figura 142. Programa principal y movimiento del robot

Al ejecutar el programa el TM5-900 comenzará a moverse siguiendo los movimientos del UR5e, no siendo necesario ejecutar ningún programa en el UR5e por lo que se puede hacer la prueba moviéndolo con ayuda del botón de movimiento libre. En resumen, el resultado sería como el del caso anterior pero al revés, es decir, el robot TM5-900 imitaría al UR5e al moverse.

3.4.3.3 Pruebas de atornillado

Tal y como se ha indicado anteriormente, en este Demostrador 3 se aborda el ensamblaje de un dispositivo electrónico mediante el uso de dos robots colaborativos (un UR5e y un TM5-900). El ensamblaje se ha decidido realizar mediante atornillado, lo que simplifica la reutilización del material para posteriores ejecuciones de la demo.

Para ello, en primer lugar, se realiza un estudio de viabilidad del proceso de atornillado en base al material disponible. Así, se cuenta con los manipuladores robóticos equipados con la pinza adaptable Hand-E y la pinza 2F-140, ambas de Robotiq. Esto plantea una serie de limitaciones ya que, por un lado, las pinzas están diseñadas para mover objetos, no para atornillar, existen multitud de efectores finales diseñados para ello contra los que una pinza no puede competir. Por otro lado, los dos robots están muy limitados en cuanto al giro que son capaces de realizar en su última articulación, el UR5e está limitado a giros de $\pm 360^\circ$ y el TM5-900 a giros de $\pm 270^\circ$.

Teniendo en cuenta el proceso a desarrollar y estas limitaciones, se ha decidido realizar el atornillado con el UR5e por varios motivos. El sensor de Par/Fuerza integrado en el UR5e permitirá añadir funcionalidades de búsqueda de tornillo o atornillar con un determinado par. Al tener el UR5e mayor ángulo de giro en su última articulación el atornillado será más rápido y además, gracias a URScript se simplifica la programación a más bajo nivel. En cuanto a las pinzas, la 2F-140 es más grande, por tanto, es mejor destinarla a sujetar las piezas, dejando la Hand-E con el UR5e para el proceso de atornillado.

Por otro lado, se va a atornillar directamente con la pinza ya que sujetar un destornillador con las pinzas aumentará en gran medida la complejidad de la solución ya que, aunque se realice un soporte específico, ligeros deslizamientos del destornillador provocarán que sea imposible alcanzar el tornillo. Además, habrá que asegurar que el tornillo se imante fuertemente al destornillador y se pueda mover sin caerse, habrá que alejarse y acercarse al tornillo cada vez que se quiera dar un nuevo giro aumentando las posibilidades de error, etc.

Por tanto, una vez decidido que se va a atornillar directamente con la pinza Hand-E y el UR5e, se realizan una serie de pruebas de viabilidad del proceso para comprobar si es posible realizar el atornillado. Para ello se ha seleccionado un conjunto de pernos y tuercas de distintos tamaños (Figura 143).



Figura 143. Pernos y tuercas para pruebas de viabilidad

En primer lugar, se ha desarrollado un script encargado de realizar un movimiento en espiral de parámetros configurables para buscar el punto de inserción. Posteriormente, para las pruebas se utiliza además el Omron TM5-900 con la función de mímica desarrollada, permitiendo replicar en el UR5e el movimiento del TM5-900. Así, mediante el TM5-900 se indica la posición donde se quiere

coger el perno a insertar (posición 1 o 2) y una vez recibida la señal, el UR5e recoge el perno indicado (Figura 144 (a)) y lo sitúa sobre la tuerca (Figura 144 (b)). Entonces, empieza bajar hasta hacer contacto con ella, tras lo cual comienza a girar mientras aplica una fuerza descendente (Figura 144 (c)). Debido a las limitaciones de giro del robot será necesario soltar el tornillo al aproximarse al límite de la última articulación, deshacer el giro, y volver a atornillar, repitiendo el proceso el número necesario de veces (Figura 144 (d) y (e)).

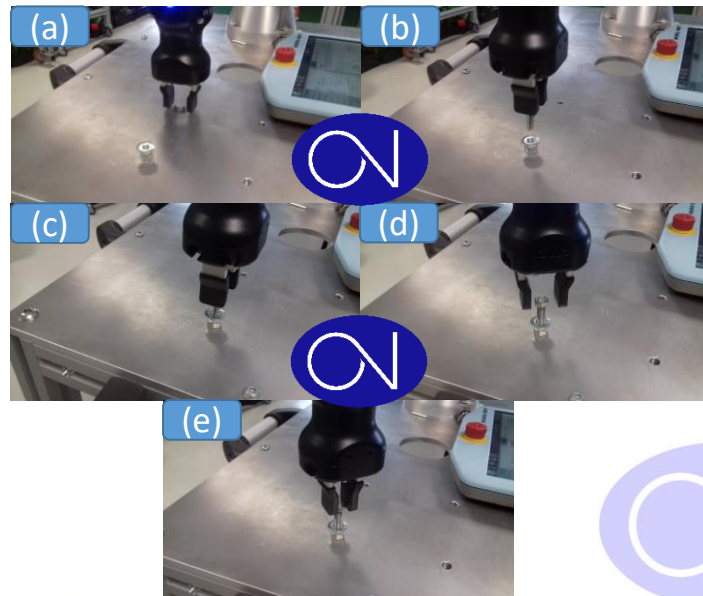


Figura 144. Pasos para las pruebas de atornillado

Gracias a estas pruebas se concluye que el proceso de atornillado puede realizarse con el material disponible, no obstante, no será un proceso eficiente, aunque si permitirá observar la flexibilidad de estos robots. Así el equipo de trabajo ha identificado ciertas recomendaciones a tener en cuenta a la hora de realizar estas actividades:

- Usar pernos en lugar de tornillos, con parte inferior plana, simplificará el proceso de búsqueda del punto de inserción y reducirá la posibilidad de error.



Figura 145. Pernos vs Tornillos

- Cuanto más grande sea el perno más sencillo será el proceso y se reducirá la posibilidad de error. Se recomienda seleccionar un perno de dimensiones mayores que las del perno más pequeño de las pruebas.

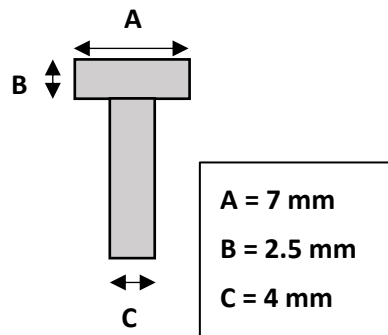


Figura 146. Dimensiones mínimas de perno

- Hay que tener en cuenta que, a mayor superficie de contacto con las pinzas mejor, por tanto, mejor usar pernos con cabeza cuadrada o rectangular para reducir la posibilidad de error.
- El proceso de búsqueda del punto de inserción podrá observarse mejor si se dispone de cavidades roscadas únicamente en la parte inferior, de forma que ambos procesos se puedan observar independientemente, primero la inserción y luego el atornillado. En la siguiente imagen se muestra un ejemplo.

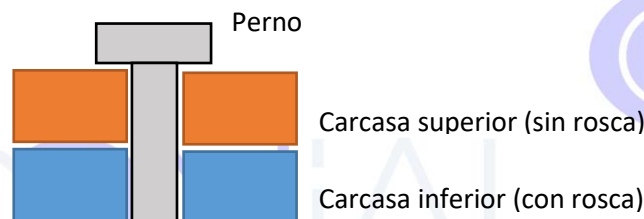


Figura 147. Ejemplo de montaje con atornillado

- El proceso de atornillado dependerá en gran medida de la calidad de la rosca del perno y del dispositivo electrónico.
- Para evitar perder mucho tiempo en el atornillado se recomienda seleccionar un tornillo corto de paso elevado.

Programación de los robots:

A continuación se explica cómo se ha realizado la programación de los robots para realizar el proceso de atornillado. Por un lado, atendiendo al robot Omron TM5-900 se ha puesto en práctica la comunicación entre ambos robots a partir de Modbus estudiada previamente, de forma que es el TM5-900 el encargado de dar al UR5e la orden inicio y el punto al que debe dirigirse para recoger el perno. Para ello se ha creado un sencillo programa en TMFlow que escribe el valor que indicamos en una de las direcciones libres del servidor Modbus, y muestra por pantalla el valor de dicho registro, repitiéndose el proceso en bucle.

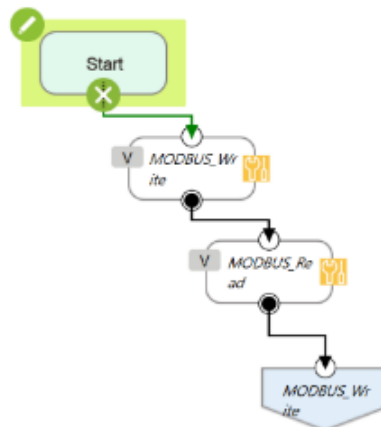


Figura 148. Programa TM Flow del TM5-900 para el atornillado con el UR5e

Para permitir al usuario modificar en tiempo real el valor de la variable hay que dirigirse a la opción de **“Display”** e introducir las entradas y salidas de interés. En el caso de las salidas se puede seleccionar un tiempo de refresco y en el de las entradas una contraseña. De esta forma al ejecutar el programa se mostrarán las salidas y, tras introducir la clave, se habilitarán las entradas para asignarles valores en tiempo de ejecución. En la sección de Modbus de la Instalación del UR5e se ha añadido la señal encargada de seleccionar el tornillo que se empleará en el programa.

En la secuencia de inicio del programa del UR5e se ha definido un script para realizar el movimiento en espiral para la búsqueda del punto de inserción, un parámetro para comprobar si el TM5-900 envía una orden de movimiento y se inicializa el gripper. El robot se mueve a una posición de “Home”, y accede al servidor de Modbus TCP del TM5-900, si la orden toma el valor 1 o 2 se dirige a la posición “P_PickScrew1” o “P_PickScrew2” respectivamente.

Si se recibe una orden válida la evaluación del If (paso final del párrafo anterior), se vuelve True y el programa puede continuar; el robot mueve el tornillo hacia arriba para sacarlo del orificio donde se ha introducido y lo posiciona sobre la tuerca, tras ello se mueve siguiendo la dirección Z descendente hasta que detecta una pequeña colisión (choca con la tuerca).

El siguiente paso es el atornillado (en el programa final se podrá añadir previo al atornillado el script de movimiento en espiral para buscar el orificio de entrada en caso necesario). Para atornillar se aplica una fuerza en Z y en RZ (dependerá del tornillo seleccionado y la velocidad deseada), liberando los ejes en X e Y para facilitar la tarea, una vez el robot se acerca al límite de la última articulación se suelta el perno, se deshace el giro, y se vuelve a repetir el proceso el número necesario de veces hasta terminar la tarea.

Para el script del movimiento en espiral, se ha introducido dentro de un nodo de fuerza para que, a la vez que se realiza el movimiento en espiral, se realice una pequeña fuerza descendente que facilitará el proceso de búsqueda del punto de inserción.

3.4.4 T4.4. Diseño y desarrollo de la interfaz de usuario

En primer lugar, cabe destacar que para el Demostrador 2 existía una colaboración persona – máquina que se basaba en el uso de una Raspberry con su pantalla de visualización y una interfaz de Realidad Aumentada. Mientras que en el Demostrador 3, se ha incorporado una interfaz del operador Hombre-Máquina (HMI) en el armario de control para poder interactuar desde éste con todo el entorno de la celda multirobótica. Se ha diseñado y realizado una interfaz de operador capaz

de brindar en tiempo real la información esencial sobre el estado general del sistema, los datos de operación de componentes e información de seguridad.

Para el diseño de la interfaz del operador, primeramente se elaboraron unos mockups para validar los menús deseados y el funcionamiento. Una vez validado, se desarrolló la interfaz utilizando TIA Portal versión 14SP1. Para el desarrollo, al crear el proyecto, se deben de añadir y conectar entre sí los dispositivos que se van a utilizar, en este caso, el panel táctil HMI Siemens KTP400 conectado al PLC 1212 DC/DC/DC. Además, se debieron configurar las direcciones de cada uno de los dispositivos y tipo de comunicación. Se definen las variables del PLC, la secuencia lógica necesaria, se establece la tabla de variables que actuarán sobre HMI y se relacionan con el PLC.

La interfaz diseñada y desarrollada se compone de varias pantallas:

- **Inicio** (Figura 149): el usuario puede seleccionar de forma sencilla e intuitiva desde la pantalla de inicio el Demostrador que se pondrá en funcionamiento, lanzando así el programa del PLC adecuado para cada caso.
 - Demo 2: se lanzará la pantalla “Demo 2 general” al pulsar el botón DEMO 2.
 - Demo 3: se lanzará la pantalla “Demo 3 general” al pulsar el botón DEMO 3.



Figura 149. Pantalla inicial de la interfaz diseñada para el panel táctil HMI Siemens KTP400

- **Demo 2 general** (Figura 150): carga del programa automático del demostrador 2. En este caso además el usuario puede configurar la dirección de la cinta y ponerla en marcha o pararla manualmente.
 - Poner la cinta en movimiento al pulsar START.
 - Parar la cinta al pulsar STOP.
 - Control de la dirección de la cinta mediante los botones de flecha izquierda o derecha.
 - Además, el dibujo del setup simula diferentes estados del proceso:
 - Baliza: se activará la luz verde cuando está en funcionamiento la cinta (Figura 150, derecha). En caso contrario, aparecerá con la luz roja iluminada (Figura 150, izquierda).
 - Cinta: según la dirección de movimiento seleccionada aparecerá el botón de la flecha correspondiente remarcada en verde una vez que se ponga en movimiento. La velocidad de la cinta es constante y corresponde a la velocidad media (9000 rpm). De forma automática la cinta se pondrá en

movimiento cuando se detecte un objeto en el sensor central y se parará al detectar objeto una fotocélula de los extremos.

- Fotocélulas de la cinta: se activarán si detectan un dispositivo pasando por el sensor. Se cambiará el color del sensor correspondiente a un azul más claro y aparecerá un objeto sobre la cinta en la posición activa (Figura 151).
- Si se intenta cambiar el sentido del movimiento de la cinta cuando está en funcionamiento, saldrá una ventana emergente para indicar que primeramente el usuario debe parar la cinta (Figura 152). Al cabo de 3 segundos desaparece el mensaje y vuelve a verse la vista principal de la Demo 2 para que el usuario realice la acción oportuna.
- Campo de protección: si existe un objeto en los campos de seguridad definidos en el SICK scanner aparecerá un aviso (Figura 153).
- En el caso de querer salir de la demo se ha de pulsar el botón EXIT, ubicado en la parte superior derecha. Al pulsar, se procederá a parar el funcionamiento de la Demo 2. El usuario debe confirmar la acción de salida de la demo a través del botón ACEPTAR de un popUp (Figura 154). En caso de aceptar la operación, la cinta se parará si estaba en movimiento y se accederá a la pantalla Inicio para seleccionar nuevamente la demo deseada.

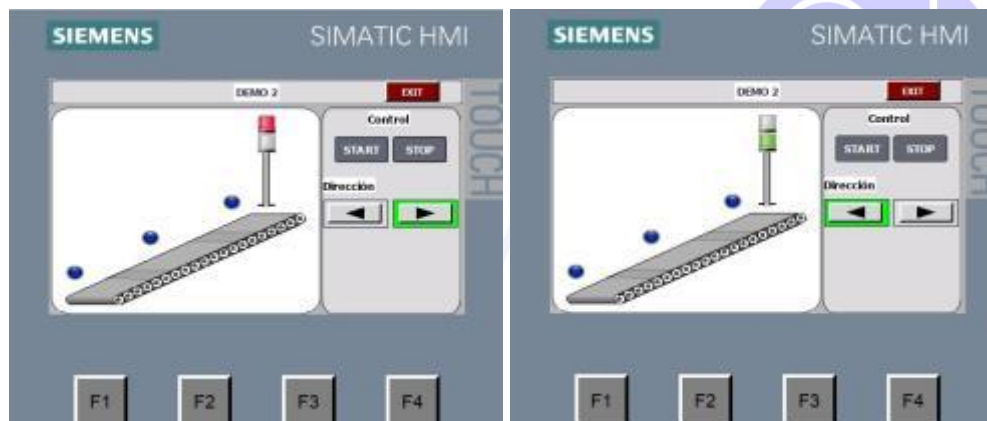


Figura 150. Vista de la pantalla “Demo 2 general” con la cinta parada (izqda.) y en funcionamiento (dcha.)

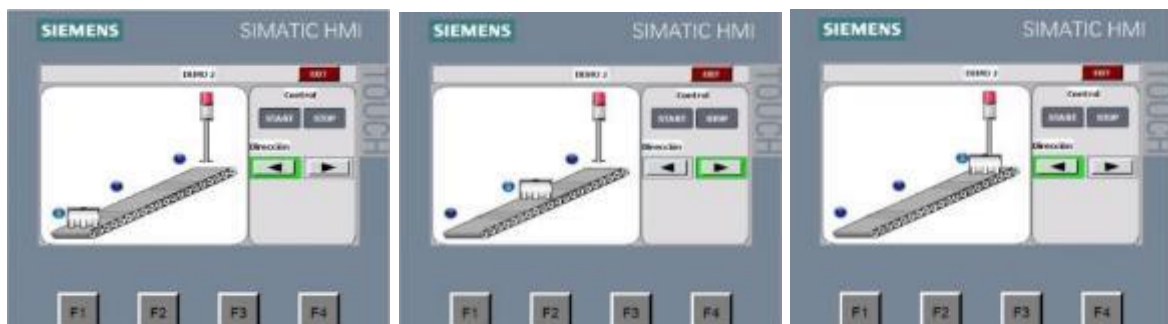


Figura 151. Vista de la pantalla “Demo 2 general” con las fotocélulas activas izquierda, central y derecha

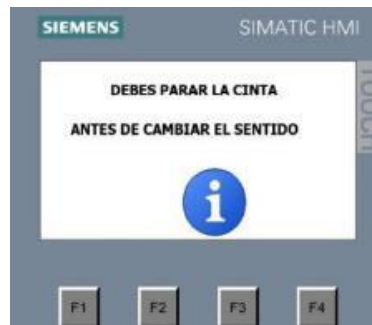


Figura 152. Vista del aviso mostrado como ventana emergente en la pantalla “Demo 2 general”

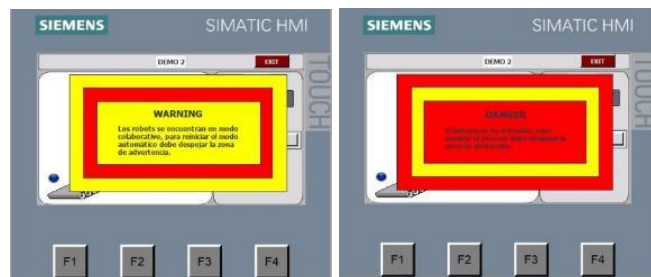


Figura 153. Vista de la pantalla “Demo 2 general” cuando hay un objeto en la zona de advertencia (izqda.) o en la de protección (dcha.)

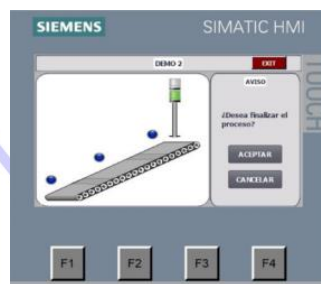


Figura 154. Vista de la pantalla “Demo 2 general” con el popUp para volver al Inicio

- **Demo 3 general** (Figura 155, izquierda): el usuario puede configurar determinadas características de funcionamiento de algunos de los componentes que conforman el Demostrador 3 (cinta y robots), iniciar el proceso de montaje de dispositivos indicando el número deseado a montar, visualizar información relacionada con el proceso (número de dispositivos que quedan por montar, los robots, cinta, fotocélula, baliza y seguridad) y parar el funcionamiento. A continuación se explican los bloques donde se realizan o muestran estas acciones.
 - Bloque de ESTADO UR5:
 - Si el estado del robot es OK la luz verde estará a ON (Figura 156), en caso contrario se activará la luz roja.
 - Estado del UR5e: preparado, inicio, trabajando o error.
 - Velocidad de funcionamiento del U5e en %.
 - Bloque de VELOCIDAD CINTA: muestra la velocidad de la cinta en rpm.
 - Bloque de ESTADO TM5:
 - Si el estado del robot es OK la luz verde estará a ON (Figura 156), en caso contrario se activará la luz roja.
 - Estado del TM5-900: preparado, inicio, trabajando o error.

- Velocidad de funcionamiento del TM5-900 en %.
- Bloque de Control:
 - Lanzar ejecución al pulsar START.
 - Parar ejecución al pulsar STOP.
 - Control de la dirección de la cinta mediante los botones de flecha izquierda o derecha.
 - Elección de la velocidad de la cinta mediante los botones de BAJA (400 rpm), MEDIA (900 rpm) o ALTA (1300 rpm).
 - Elección del número de dispositivos a ensamblar. El campo es editable por el usuario.
 - Cuadro informativo sobre el número de dispositivos restantes a realizar. Campo no editable por el usuario.
- En caso de existir algún aviso, éste aparecerá a la derecha sobre el bloque de control (Figura 155, derecha).
- Si existe un error del sistema aparecerá un botón intermitente en la esquina superior izquierda (Figura 164), el cual permite acceder a la ventana “DEMO 3 – Registro de alarmas” (Figura 163). En dicho punto se detallará más sobre ello.
- Configuración del UR5e al pulsar sobre la imagen del UR5e (se lanza la pantalla “DEMO 3 – Estado y configuración UR5”).
- Configuración del Omron TM5-900 al pulsar sobre la imagen del Omron TM5-900 (se lanza la pantalla “DEMO 3 – Estado y configuración TM5”).
- Además, el dibujo del setup simula diferentes estados del proceso:
 - Baliza: se activará la luz verde cuando está en funcionamiento la cinta (Figura 156, derecha). En caso contrario, aparecerá con la luz roja iluminada (Figura 156, izquierda).
 - Cinta: cuando se activa, aparecerá una flecha indicando la dirección de movimiento (Figura 156, derecha).
 - Fotocélulas de la cinta: se activarán si detectan un dispositivo pasando por el sensor. Se cambiará el color del sensor correspondiente a un azul más claro y aparecerá un objeto sobre la cinta en la posición activa (Figura 157).
 - Campo de protección: si existe un objeto en los campos de seguridad definidos en el SICK scanner aparecerá un aviso (Figura 158). Además se mostrará en amarillo la zona de advertencia (reducción velocidad) o en rojo la zona de protección (parada).
- En el caso de querer salir de la demo se ha de pulsar el botón EXIT, ubicado en la parte superior derecha. Al pulsar, se procederá a parar el funcionamiento de la Demo 3. El usuario debe confirmar la acción de salida de la demo a través del botón ACEPTAR de un popUp. Si los robots están en funcionamiento saltará un aviso indicando que no es posible parar el programa. En caso de aceptar la operación el usuario accederá a la pantalla Inicio para seleccionar nuevamente la demo deseada.



Figura 155. Vista inicial de la pantalla “Demo 3 general” (izqda.) y con un aviso (dcha.)



Figura 156. Vista de la pantalla “Demo 3 general” con los robots en funcionamiento (izqda.) y con la cinta en marcha (dcha.)



Figura 157. Vista de la pantalla “Demo 3 general” con las fotocélulas activas izquierda, central y derecha



Figura 158. Vista de la pantalla “Demo 3 general” cuando hay un objeto en la zona de advertencia (izqda.) o en la de protección (dcha.)

- **DEMO 3 – Estado y configuración UR5** (Figura 159): en esta pantalla se muestra el estado del robot UR5e y los parámetros que son posibles configurar.
 - Bloque de STATUS: visualización del estado del robot UR5e.
 - Estado: puede ser uno de los siguientes estados dormido, inicialización, síncrono, espera CPU, espera TM5, soporte, carcasa superior, tornillos, ensamblado finalizado, depositando pieza.
 - Velocidad: velocidad del robot en %.
 - Inicio programa: si el inicio es OK la luz verde estará a ON, en caso contrario se activará la luz roja.
 - Preparado: si el robot está preparado la luz verde estará a ON, en caso contrario se activará la luz roja.
 - Trabajando: si el robot está trabajando la luz verde estará a ON, en caso contrario se activará la luz roja.
 - Seta de emergencia: si la seta de emergencia está activada la luz verde estará a ON, en caso contrario se activará la luz roja.
 - Modo colaborativo: si el robot está en modo colaborativo la luz verde estará a ON, en caso contrario se activará la luz roja.
 - Ejecutando: si el robot está ejecutando una acción la luz verde estará a ON, en caso contrario se activará la luz roja.
 - Error: si está todo OK la luz verde estará a ON, pero si hay algún error en el robot se activará la luz roja. Este aviso de error genérico puede activarse por varias condiciones como que el programa no este ejecutándose después de la orden y x segundos, las propias setas de emergencia de los robots, errores de comunicación, etc. Para obtener una descripción más detallada del error se debe acceder al alarmero pulsando sobre el botón intermitente de la esquina superior izquierda (Figura 160, izquierda), el cual permite acceder a la ventana “DEMO 3 – Registro de alarmas”. En dicho punto se detallará más sobre ello.
 - SICK: si se encuentra en verde es porque no hay nada ni en el campo de protección ni en el de advertencia (Figura 161, izquierda). Si se activa la luz naranja hay algo en el campo de advertencia (Figura 161, centro) y si se activa la luz roja es porque hay algo en el de protección (Figura 161, derecha), mostrándose el botón de error en la esquina superior izquierda.
 - Bloque de CONTROL:
 - Velocidad: permite al usuario introducir la velocidad del robot deseada en % y salvarla, pulsando en Guardar. En caso de no ser un valor correcto, aparecerá un mensaje en el lado derecho de la pantalla (Figura 159, derecha).
 - Control del programa: la visualización de los botones PLAY o PAUSE depende del estado de ejecución de los programas de los robots, es decir, el botón PLAY conmuta a un estado invisible y se visualizaría solo el botón PAUSE si el robot está ejecutando ya el programa, y viceversa. La visualización del botón PAUSE también está condicionado por los

programas de los robots, ya que hay un momento en el que se encuentran sincronizados trabajando sobre la misma pieza, es en este momento cuando no se puede detener el programa.

- Movimiento libre: permite activar el movimiento libre (ON) para así no tener que mantener pulsado el botón de la controladora. Debido al riesgo del movimiento libre del robot, este botón requiere del registro del usuario (usuario y contraseña) ya que es una funcionalidad destinada a mantenimiento, no a nivel usuario (Figura 160, derecha).



Figura 159. Vista de pantalla “DEMO 3 – Estado y configuración UR5” (izqda.) y con aviso (dcha.)



Figura 160. Vista de la pantalla “DEMO 3 – Estado y configuración UR5” con el símbolo de alarma activo, conmutando entre amarillo y rojo (izqda.) y al activar el movimiento libre (dcha.)



Figura 161. Vista de la pantalla “Demo 3 general” con el SICK en los tres estados: verde (izqda.), naranja (centro) y rojo (dcha.)

- DEMO 3 – Estado y configuración TM5** (Figura 162): en esta pantalla se muestra el estado del robot Omron TM5-900 y los parámetros que son posibles configurar.
 - Bloque de STATUS: visualización del estado del robot TM5-900.

- Estado: puede ser uno de los siguientes estados dormido, inicialización, síncrono, espera CPU, espera UR5, raspberry, carcasa inferior, captura imagen.
- Velocidad: velocidad del robot en %.
- Inicio programa: si el inicio es OK la luz verde estará a ON, en caso contrario se activará la luz roja.
- Preparado: si el robot está preparado la luz verde estará a ON, en caso contrario se activará la luz roja.
- Trabajando: si el robot está trabajando la luz verde estará a ON, en caso contrario se activará la luz roja.
- Seta de emergencia: si la seta de emergencia está activada la luz verde estará a ON, en caso contrario se activará la luz roja.
- Modo automático: si el robot está en modo auto la luz verde estará a ON, en caso contrario se activará la luz roja.
- Ejecutando: la luz verde estará a ON si el PLC y los robots han realizado bien la secuencia de arranque y ambos robots están ejecutando el programa. En caso contrario se activará la luz roja.
- Error: si está todo OK la luz verde estará a ON, pero si hay algún error en el robot se activará la luz roja. Este aviso de error genérico puede activarse por varias condiciones como que el programa no este ejecutándose después de la orden y x segundos, las propias setas de emergencia de los robots, errores de comunicación, etc. Para obtener una descripción más detallada del error se debe acceder al alarmero pulsando sobre el botón intermitente de la esquina superior izquierda (Figura 162, derecha), el cual permite acceder a la ventana “DEMO 3 – Registro de alarmas”. En dicho punto se detallará más sobre ello.
- SICK: si se encuentra en verde es porque no hay nada ni en el campo de protección ni en el de advertencia. Si se activa la luz naranja hay algo en el campo de advertencia y si se activa la luz roja es porque hay algo en el de protección.
- Bloque de CONTROL:
 - Velocidad: permite al usuario introducir la velocidad del robot deseada en % y salvarla, pulsando en Guardar. En caso de no ser un valor correcto, aparecerá un mensaje en el lado derecho de la pantalla.
 - Control del programa: la visualización de los botones PLAY o PAUSE depende del estado de ejecución de los programas de los robots, es decir, el botón PLAY conmuta a un estado invisible y se visualizaría solo el botón PAUSE si el robot está ejecutando ya el programa, y viceversa. La visualización del botón PAUSE también está condicionado por los programas de los robots, ya que hay un momento en el que se encuentran sincronizados trabajando sobre la misma pieza, es en este momento cuando no se puede detener el programa.



Figura 162. Vista de pantalla “DEMO 3 – Estado y configuración TM5” (izqda.) y con error (dcha.)

- **DEMO - Registro de alarmas** (Figura 163): en la pantalla del alarmero se visualizan todas las alarmas del proceso (el estado del SICK, el estado de los robots y el tipo de fallo que pudiera suceder así como un texto explicativo asociado a cada alarma). Se puede acceder pulsando en F4 o a través del botón de aviso de cualquiera de las pantallas, el cual aparece en la zona superior izquierda. Con el fin de resaltar el estado de alarma este botón conmuta a color amarillo cada 700 ms (Figura 164). Pueden existir las siguientes alarmas:
 - Errores registrados específicos del TM5-900 (Figura 165): modo manual, selección de programa, ejecución de programa, pausa de programa, parada de programa, detección SICK, seta de emergencia, orden de trabajo, error general, control de ejecución, editando programa.
 - Errores registrados específicos del UR5e (Figura 166): alimentación, selección de programa, ejecución de programa, pausa de programa, parada de programa, detección SICK, seta de emergencia, orden de trabajo, liberación de frenos.
 - Errores generales (Figura 167): seta de emergencia PLC, movimiento cinta CPU, detección SICK PLC o aviso detección SICK PLC.

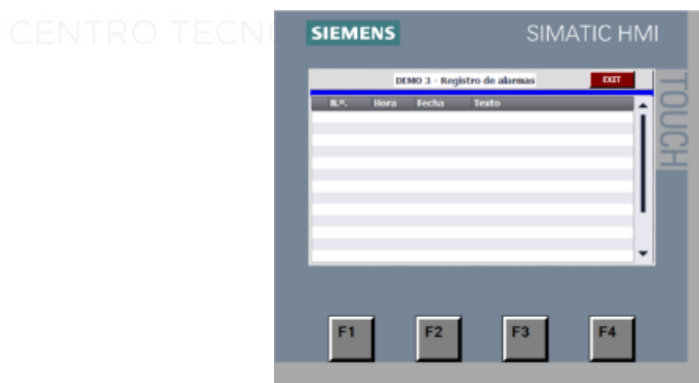


Figura 163. Vista de pantalla “DEMO 3 – Registro de alarmas”

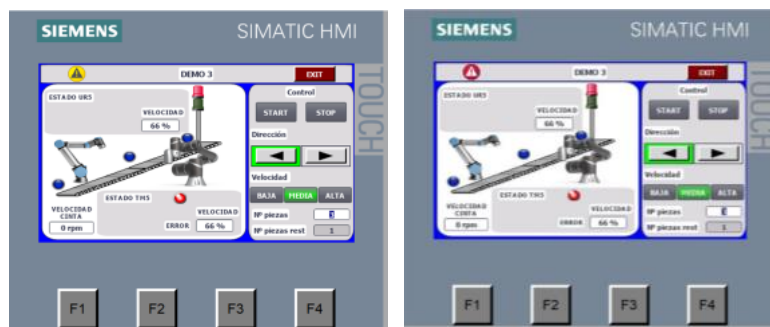


Figura 164. Vista de la pantalla “DEMO 3” con el símbolo de alarma activo, conmutando entre amarillo y rojo

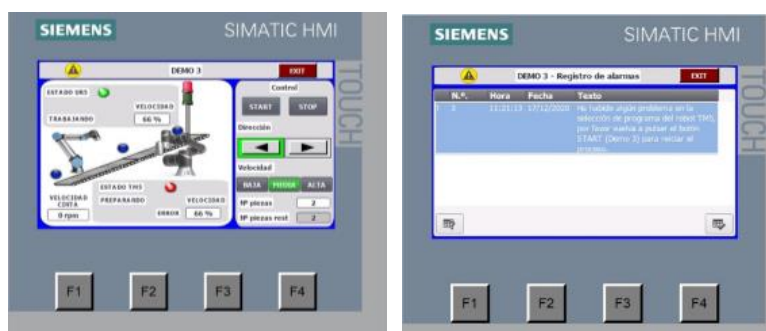


Figura 165. Vista de pantalla “DEMO 3” con el Omron TM5-900 en error (izqda.) y “DEMO 3 – Registro de alarmas” con el error selección de programa de TM5-900 (dcha.)



Figura 166. Vista de pantalla “DEMO 3” con el UR5e en error (izqda.) y “DEMO 3 – Registro de alarmas” con el error frenos y ejecución de programa de UR5e (dcha.)

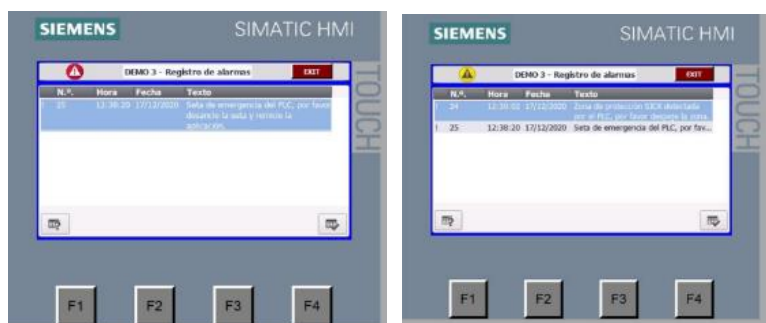


Figura 167. Vista de pantalla DEMO 3 – Registro de alarmas” con el error de seta de emergencia (izqda.) y detección de presencia en la zona de protección (dcha.)

TIA Portal ofrece posibilidad de realizar la simulación del programa del autómatas junto con el programa de una pantalla Basic Panel HMI utilizando PLCSim. Por tanto, para comprobar el correcto funcionamiento de la interfaz, se ha validado primeramente en simulación depurando el programa

y validando la correcta experiencia del operador. Una vez validado, se prueba su funcionamiento con el hardware físico (Figura 168), validando el desarrollo realizado con varios operadores, garantizando así una experiencia de usuario exitosa durante las pruebas a realizar en la siguiente tarea T4.5 de integración, pruebas y validación.



Figura 168. Detalle del armario encendido con la pantalla inicial de la interfaz desarrollada

3.4.5 T4.5. Integración, pruebas y validación

Durante esta tarea se procedió en primer lugar a la integración de todos los elementos desarrollados en las tareas anteriores. En este caso, además, hay que destacar que el Demostrador 3 desarrollado en este HITO 4 cuenta con múltiples elementos existentes gracias a las tareas realizadas en el Demostrador 2. Sin embargo, esto no impide que se deban realizar ajustes que permitan optimizar el Demostrador 3 asociado a este hito. Por este motivo se ha dispuesto por un lado del UR5e con la pinza Hand-e y dedos personalizados y por otro, del Omron TM5-900 con cámara integrada, la pinza robótica Robotiq 2F-140 y el SICK scanner MicroScan3 Core, dispositivos de inversión propia. Se revisó la integración del resto de componentes ya existentes que conforman el setup y se fueron integrando los nuevos a medida que se iban recibiendo: baliza, cinta, sensores, el sistema de comunicaciones y la unidad de control, hardware asociado a la cinta y hardware asociado al robot. Se colocaron los robots enfrentados a ambos lados de la cinta en torno al sensor central de detección de presencia (Figura 169), manteniendo el escáner colocado en la plataforma del Omron TM5-900 (Figura 170).

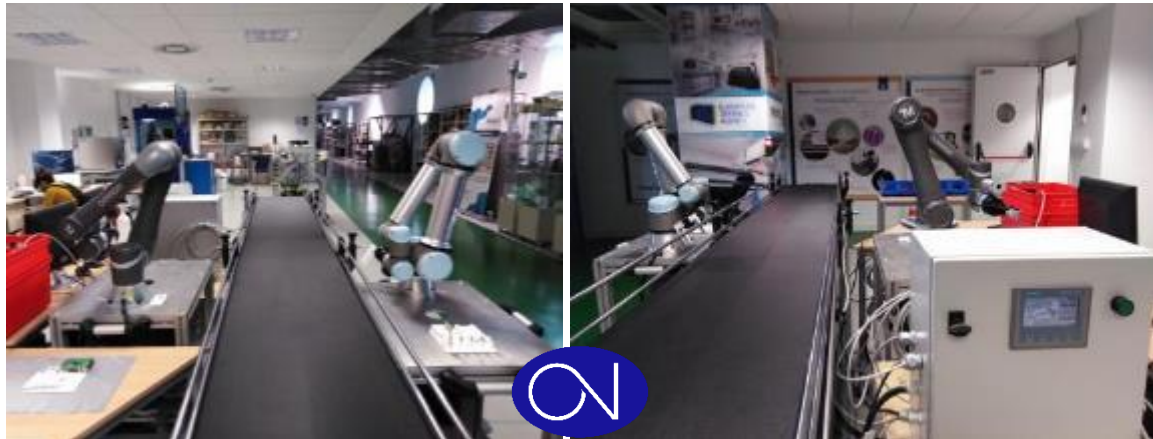


Figura 169. Imagen del setup final del Demostrador 3 (I)



Figura 170. Imagen del setup final del Demostrador 3 (II)

En cuanto a los objetos con los que interactuar durante el Demostrador 3 caben destacar los siguientes pasos realizados en la instalación:

- Preparación del dispositivo a montar: impresión de piezas de la carcasa diseñada y explicada en T4.1, y comprobación manual de inserción correcta de Raspberry y tornillos.
- Base para el UR5e (Figura 171): colocación de la base en la plataforma móvil del UR5e y los componentes con los que va a interactuar el robot (cuatro tornillos, pieza superior de la carcasa, retenedor y un landmark de calibración).
- Base para el Omron TM5-900 (Figura 172): colocación de la base sobre una bandeja tipo la del Demostrador 2 pero sin cortes en la espuma. Colocación sobre la base de los componentes con los que va a interactuar el robot (parte inferior de la carcasa, la Raspberry y un landmark). El conjunto se colocará accesible desde el TM5-900.



Figura 171. Imagen del setup final del Demostrador 3: base para el UR5e



Figura 172. Imagen del setup final del Demostrador 3: base para el Omron TM5-900

Una vez preparados e instalados los elementos que conforman el Demostrador 3, son necesarios los siguientes pasos para su integración, realizando en primer lugar pruebas unitarias y posteriormente, en conjunto:

- **Seguridad:** integración del SICK escáner con el PLC, la cinta y ambos robots. Además, se ha realizado la configuración de parámetros de seguridad de los robots.
- **Cinta:** actualización de configuración acorde a su integración en el sistema de seguridad y control de velocidad.
- **Sistema de comunicaciones entre PLC y robots:** configuración del sistema de comunicaciones y la secuencia lógica del PLC.
- **Integración de los robots y configuración de todos los componentes** realizado como parte de la tarea asociada al sistema de comunicaciones entre PLC y robots.
- **Pruebas y validación finales del conjunto.**

3.4.5.1 Sistemas de seguridad

Como en el Demostrador 2 se han aplicado diversas medidas de seguridad: setup (aviso en el suelo sobre la zona de trabajo e integración de un escáner SICK para protección horizontal de zona de peligro), configuración adecuada de los robots y del sistema general, así como información adecuada a los operadores mediante unas indicaciones previas e información del estado de la célula por medio de la interfaz.

Respecto al escáner SICK MicroScan3 Core, en el Demostrador 3 su rango de protección engloba a ambos robots y al funcionamiento de la cinta. Como en el Demostrador 2 los campos de protección (advertencia y protección) se configuraron con el propio software del escáner. El escáner se colocó en la plataforma del Omron TM5-900 a una altura por debajo de las rodillas de un operador monitorizando el área de funcionamiento de los robots, realizando parada de protección o reduciendo la velocidad de robots y cinta en función de si algún operario se acerca a la célula mientras el sistema está en funcionamiento. En este sentido, a la hora de integrar los dos robots y la cinta en el sistema de seguridad, se observaba que era más escalable y flexible que el escáner SICK MicroScan3 Core tuviese su propia fuente y así no depender de tener conectado el Omron TM5-900 todo el tiempo. Por tanto, se procedió a proporcionar al SICK de una fuente de alimentación propia, actualizando así el conexionado del armario y del SICK en una caja eléctrica propia con respecto a lo detallado en la tarea 4.1. A continuación, se resumen las principales modificaciones e integraciones realizadas con respecto a esto:

- Caja del SICK según lo desarrollado en el Demostrador 2 (Figura 174): en ese caso la alimentación del SICK se realizaba desde la controladora del propio Omron TM5-900 y la supervisión solo afectaba a dicho robot por lo que para la conexión de las zonas de protección y advertencia se tuvo en consideración que el robot cuenta con 6 entradas redundantes de seguridad (Figura 173). De esta forma, en la caja sólo era necesario disponer de los relés y relés de seguridad necesarios, así como los cables asociados para las conexiones a la controladora (safety y power).

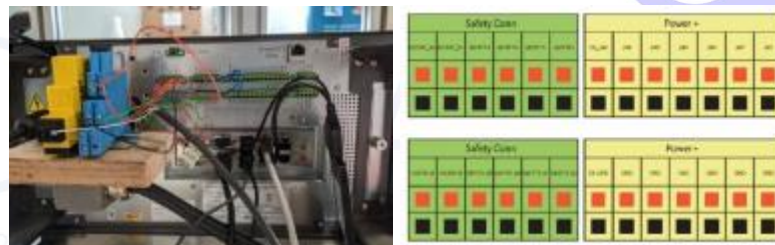


Figura 173. Asignación de terminales del cable de conexión del SICK con los relés y la controladora del Omron Tm5-900 en el Demostrador 2



Figura 174. Caja SICK según el conexionado del Demostrador 2

- Caja SICK ajustada al Demostrador 3 (Figura 175), incluyendo las siguientes modificaciones: las mangueras provenientes de la caja de control del Omron TM5-900 se han cambiado por unas más largas por si el día de mañana se quiere mover de ubicación el dispositivo SICK; los cables han sido etiquetados para que concuerden con los nuevos esquemas eléctricos;

se ha introducido una fuente de alimentación de 24 VDC para que el dispositivo fuese independiente de las cajas de control de los robots, de esta manera ya no es necesario activarlos para que el dispositivo SICK funcione; se ha incorporado el conexionado necesario para la conexión con el UR5e así como con el PLC para incorporar la cinta en el sistema de protección; se han tenido que incorporar los relés y relés de seguridad pertinentes. Debido a ello se ha utilizado una nueva caja eléctrica de mayor tamaño, disponible en las instalaciones. Esto ha supuesto las modificaciones pertinentes en las controladoras de los robots y en el armario eléctrico.



Figura 175. Caja SICK actual – Demostrador 3

- Armario eléctrico actualizado para incorporar el SICK scanner como entrada al PLC y como dispositivo de seguridad. Como se ha comentado anteriormente, en el Demostrador 2 sólo influía sobre el Omron TM5-900 (por lo que sólo estaba conectado a su controladora), pero en el Demostrador 3 la protección del SICK engloba a ambos robots y al funcionamiento de la cinta. Esto ha implicado varios cambios en el armario y los cables han sido etiquetados para que concuerden con los nuevos esquemas eléctricos. Respecto al sistema de protección asociado a la cinta, la señal SICK se cableó en serie con la seta de emergencia (pero sin necesidad de rearmar). De esta forma, en el caso de acercarse un operario a la zona de protección del SICK scanner, la cinta se detiene y si por el contrario el usuario se aleja de dicha zona, ésta se rearma sola sin necesidad de actuar ni sobre el HMI ni sobre el botón de rearme. En la Figura 176 se muestra el armario actual, con el detalle a la derecha de los relés asociados al SICK.



Figura 176. Armario actual – Demostrador 3

En cuanto a la seguridad asociada a los **robots**, en ambos casos se ha realizado la siguiente configuración:

- **Hardware:** existen dos mecanismos de seguridad disparados por señales externas. Se utilizará la seta del mando de control como parada de emergencia y una parada de protección en función de la señal proporcionada por el SICK escáner.
 - **Parada de emergencia:** puede accionarse con la seta del mando de control o mediante la entrada redundante de seguridad de parada de emergencia. Cuando se acciona se activan los frenos del robot y se desactiva la alimentación de los motores una vez detenidos (parada de categoría 1 acorde a ISO 60204-1).
 - **Parada de protección:** puede accionarse con la entrada redundante de seguridad habilitada a tal fin. Cuando se activa, el robot se detiene pero la alimentación se mantiene activa (parada de categoría 2 acorde a ISO 60204-1), en ese momento comienza un proceso de monitorización continua que comprueba que no se produzcan movimientos no intencionados, lo que desencadenaría una parada de categoría 0 (se desactiva la alimentación de forma inmediata).
- **Software:** mecanismos de seguridad modificables con ayuda del entorno de programación. Cabe destacar: ajustes de seguridad y espacio de operación.

De esta forma, como en el Demostrador 2, si una persona entra en el área de advertencia del SICK scanner durante el proceso en curso, el robot en funcionamiento reduce su velocidad entrando en modo colaborativo. Si se entra en el área de protección del SICK, el robot que se encuentre en funcionamiento realiza una parada de protección (Figura 177). Si la persona abandona esta zona, el robot reanuda su movimiento automáticamente, al principio a una velocidad reducida y monitorizada por el campo de advertencia, de modo que luego puede reanudar la velocidad de trabajo original una vez que salga de la zona monitorizada. Por otra parte, como se explicará a continuación, al acercarse un operario a la zona de protección del SICK scanner, la cinta se detiene y si por el contrario el usuario se aleja de dicha zona, ésta se rearma sola sin necesidad de actuar ni sobre el HMI ni sobre el botón de rearme.

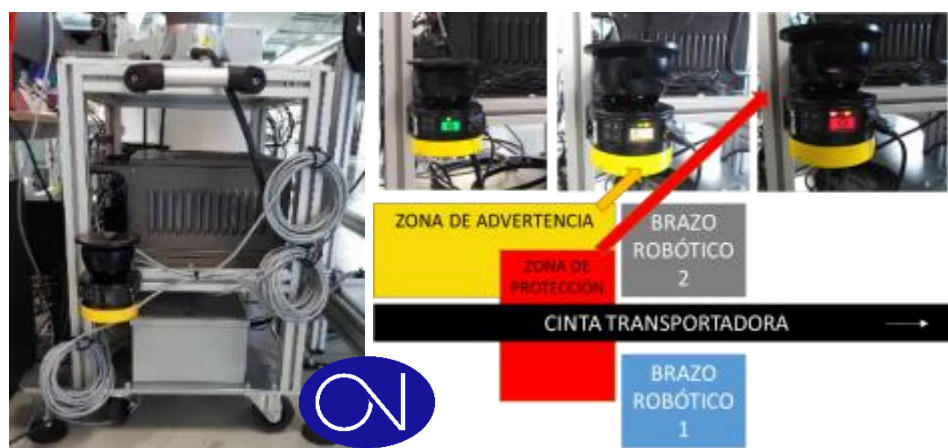


Figura 177. Esquema representativo de las zonas de advertencia y protección en el Demostrador 3

3.4.5.2 Cinta

Como se comentó en T4.1, en el Demostrador 3 la activación, velocidad y dirección de la cinta podrá ser seleccionada a través del HMI, además de poder verse influenciada por el estado dado por el

SICK y de los robots por lo que ha sido necesario realizar una serie de modificaciones en el armario y consecuentemente en el control.

Como se ha explicado en el apartado anterior, respecto a la incorporación de la cinta en el sistema de seguridad monitorizado por el SICK, se actualizó el conexionado pertinente en el armario eléctrico. Además, como se tenían las entradas del SICK duplicadas, se quitó dicha duplicidad y se ha usado esa entrada para la confirmación de movimiento de la cinta (cable proveniente del variador).

En relación a la activación, control de la velocidad y dirección de la cinta a través del HMI, en la T4.1 se comentaron las actualizaciones realizadas en el armario para incorporar el HMI de Siemens. En esta tarea de integración se han realizado las actualizaciones necesarias del conexionado para poder llevar a cabo el control de velocidad la cinta por medio del PLC en lugar del potenciómetro (que estaba alimentado a la tensión suministrada por el propio variador, 10 VDC), así como gestionar desde el PLC el sentido de giro y la activación en lugar de usar un interruptor. De esta forma, al eliminar el potenciómetro y los interruptores, se han actualizado las entradas y salidas del controlador sobre el estado de la cinta transportadora y las señales de control, realizando las modificaciones oportunas en el driver de la cinta:

- Como no se dispone de una tarjeta de salidas analógicas se ha decidido configurar el driver de control de la cinta a 3 velocidades distintas, es decir, se puede seleccionar velocidad baja (400 rpm), media (900 rpm) y alta (1300 rpm). El variador es el encargado de regular la red trifásica acorde a la potencia necesaria por el motor, ésta acciona unos relés de estado acorde a la activación de la cinta.
- De esta forma, la velocidad de la cinta es controlada por medio de la combinación de dos señales de salida del PLC al variador, permitiendo así seleccionar una de esas tres velocidades pre-configuradas:
 - 00: parado, no hay velocidad seleccionada.
 - 01: velocidad baja (400 rpm).
 - 10: velocidad media (900 rpm).
 - 11: velocidad alta (1300 rpm).
- La dirección de movimiento de la cinta es controlada a través de 2 salidas del PLC al variador para definir (START/STOP a la izquierda o derecha):
 - Sentido horario: hacia la derecha mirando desde el lado del programador.
 - Sentido antihorario: hacia la izquierda desde el lado del programador.
- Se utiliza una señal analógica desde el variador al PLC para que el PLC realice la lectura de la velocidad de la cinta para transmitírsela a los robots por la red. La señal transmitida por el variador es de 0-10VDC.
- La baliza de estado recibe las señales de estado directamente del variador por previa configuración del variador. El verde fue asignado al pin correspondiente a la activación del motor y por el contrario el rojo se conectó al pin de parada del motor.

Considerando esta arquitectura de control, se ha programado el PLC para disponer de las siguientes funcionalidades asociadas a la cinta, teniendo en consideración que se puede seleccionar la opción de Demostrador 2 o Demostrador 3 desde el HMI como se explicó en 3.4.4:

- El PLC recibe las señales de posición de cada detector y, en función de las mismas y del paso del proceso en el que se encuentre, envía la orden de parada, activación o no de la cinta:
 - En la DEMO2 la cinta se pone en movimiento a los 3 segundos de detectar un objeto el sensor de presencia central, para dar tiempo al brazo robótico a retirarse. Pasada esa temporización se activa la marca “Accionamiento_Marcha”. Dicha marca, a su vez, activa la marca “Arranque_Cinta”, que está directamente ligada a la salida “Cinta_Start” y pone la cinta en movimiento a velocidad media. Una vez que el objeto llega a cualquiera de los sensores laterales la cinta se detiene ya que se activa la marca “Accionamiento_Paro” que es la que resetea la marca “Arranque_Cinta”. Si hay un objeto en alguno de los extremos, la cinta no se pone en movimiento.
 - En la DEMO3 la cinta se pone en movimiento cuando el robot Omron TM5-900 envía la orden de activación una vez que ha realizado la foto del objeto depositado en la cinta. Para ello se activa Qx_Order_Move_TM5 (dirección %DB10.DBX15.3, variable almacenada en el bloque de datos DB10 correspondiente a “ControlCintaD3”) y la cinta se pone en movimiento. Una vez que se ha desplazado lo suficiente para liberar el espacio donde deposita el robot el dispositivo, la cinta se detiene ya que se activa Qx_Order_Stop_Move (dirección %DB10.DBX15.4, correspondiente al bloque funcional DB10 “ControlCintaD3”). Si hay un objeto en alguno de los extremos, la cinta no se pone en movimiento. En el caso de que además el sensor central detecte que hay un objeto, el PLC no permitirá que el robot deposite un nuevo dispositivo sobre la cinta. Cabe destacar que debido a la envergadura del demostrador, se han utilizado bloques de datos en lugar de marcas. De esta forma cada bloque de datos almacena las variables correspondientes a las diferentes funciones Modbus TCP/IP (carpeta ControlModbusTM5 PROFINET URC5) y son diferenciadas de las variables utilizadas para el control del movimiento de la cinta (carpeta Data).
- La salida de activación del motor del controlador está condicionada por el sistema de emergencia, evitando el movimiento de la cinta si el usuario pulsa la seta de emergencia. Es necesario liberar la seta, y pulsar el botón de rearme frontal del armario para que se ponga otra vez en funcionamiento.
- En el caso de acercarse un operario a la zona de protección del SICK scanner, la cinta se detiene y si por el contrario el usuario se aleja de dicha zona, ésta se rearma sola sin necesidad de actuar ni sobre el HMI ni sobre el botón de rearme.
- La velocidad de la cinta si se selecciona DEMO2 es la media y en el caso de DEMO3 el usuario debe seleccionar en el HMI una de las tres posibles. En el caso de que el operario no seleccione ninguna velocidad, el HMI emitirá un aviso (popUp) previamente al inicio del proceso.
- La rampa de deceleración del driver de la cinta está a 5 segundos de tal forma que si uno de los sensores de presencia de los extremos detecta un objeto, la cinta se para y el objeto queda posicionado en el extremo de la cinta sin caerse sea cual sea su velocidad. El sensor fotoeléctrico permite una detección estable independientemente del color y reflectancia.

- El sentido de giro de la cinta se selecciona desde el HMI. De manera semejante a la selección de la cinta sucede con la selección de velocidad. En este caso como la dirección de la cinta se selecciona a través de dos salidas digitales no existe una dirección por defecto.
- La luz verde de la baliza de señalización se activa al ponerse la cinta en movimiento, por lo que mientras está parada, la luz roja permanece encendida. De este modo el operario sabrá si la cinta se encuentra en tensión porque aparecerá iluminada en rojo o si está en movimiento porque se iluminará en verde.
- Cabe destacar una serie de avisos en la DEMO3:
 - El usuario debe introducir el número de piezas a ensamblar en el HMI. En el caso de que no lo haga, el sistema no se iniciará y mostrará un aviso por pantalla.
 - Si el usuario está controlando el Omron TM5-900 a través de la pantalla, el proceso no se inicia, si no que emite nuevamente la señal de aviso para liberar el control del robot. De manera similar sucede si el operario está modificando el programa del robot.
 - En el caso de que alguno de los robots se encuentren en error el sistema tampoco se inicia indicando al operario que debe solventar el error previamente a comenzar el proceso.

3.4.5.3 Sistema de comunicaciones

El sistema de comunicaciones se ha actualizado basándose en los desarrollos realizados dentro de la tarea 4.2 (apartado 3.4.2) y teniendo en cuenta el resto de elementos incluidos en esta tarea.

3.4.5.4 Integración de los robots

Se ha integrado los robots teniendo en consideración la arquitectura planteada, considerando el flujograma general presentado en la Figura 136 en la que los robots deben seguir cada uno de los bloques de manera sincronizada para lograr realizar el montaje del dispositivo asociado a este Demostrador 3. Como se ha indicado, si el usuario se encuentra en la zona de advertencia el robot trabajará en modo colaborativo. En caso de que el usuario se acerque al robot por el espacio disponible entre la mesa y la cinta, el robot se parará ya que la persona se encuentra en la zona de protección. Esta medida está fundamentalmente pensada para asegurarse que durante los movimientos entre la mesa y la cinta no haya nada en la trayectoria.

El diagrama de flujo del programa final del UR5e se muestra en la Figura 178, detallando algunas de los subprogramas en la Figura 179. Se definieron una serie de variables auxiliares para facilitar el trabajo con señales Modbus en funciones de URScript sin necesidad de definir las en el archivo de instalación. En el caso del Omron Tm5-900, su diagrama de flujo del programa final se muestra en la Figura 180, así como el detalle de algunas de los subprogramas en Figura 181. Los pasos de ambos robots están relacionados con las tres fases planteadas en el apartado 3.4.1.3: 1. Ensamblaje, 2. Atornillado y 3. Inspección, explicadas en 3.4.3. Como se indicó, en el caso del Omron TM5-900 se han utilizado diversas funcionalidades del sistema de visión integrado y se han dotado a las bases de unos marcadores o Landmark. En el caso del soporte del Omron, el marcador sirve para localizar la base con las piezas y acceder a ellas sin necesidad de colocarlas en una posición fija. En el caso de la base del UR5e, el landmark sirve para calibración, recogiendo el UR5e y presentándolo al Omron para que la posición de inserción del TM5-900 se adapte a la del UR5e. En

la integración final la secuencia de calibrado en el TM se ha añadido como un nodo que puede conectarse con el nodo inicial cuando se quiera hacer la calibración manualmente (Figura 182).

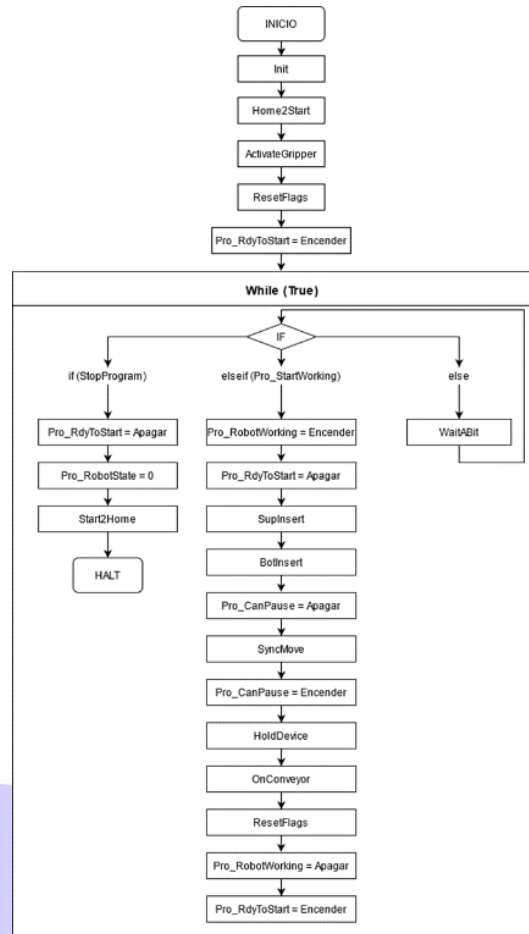


Figura 178. Flujograma simplificado del programa final del UR5e

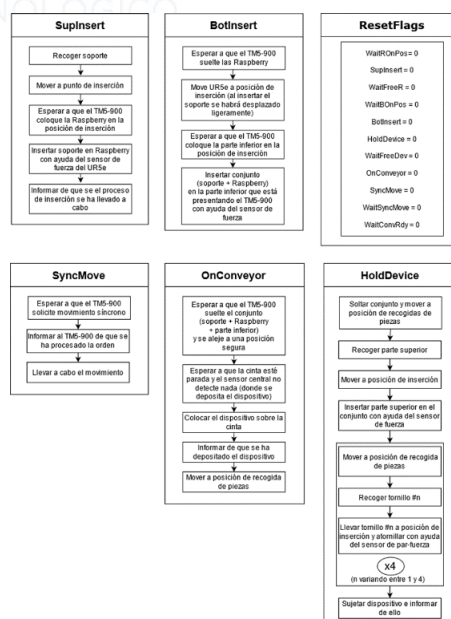


Figura 179. Detalle de subprogramas del flujograma final del UR5e

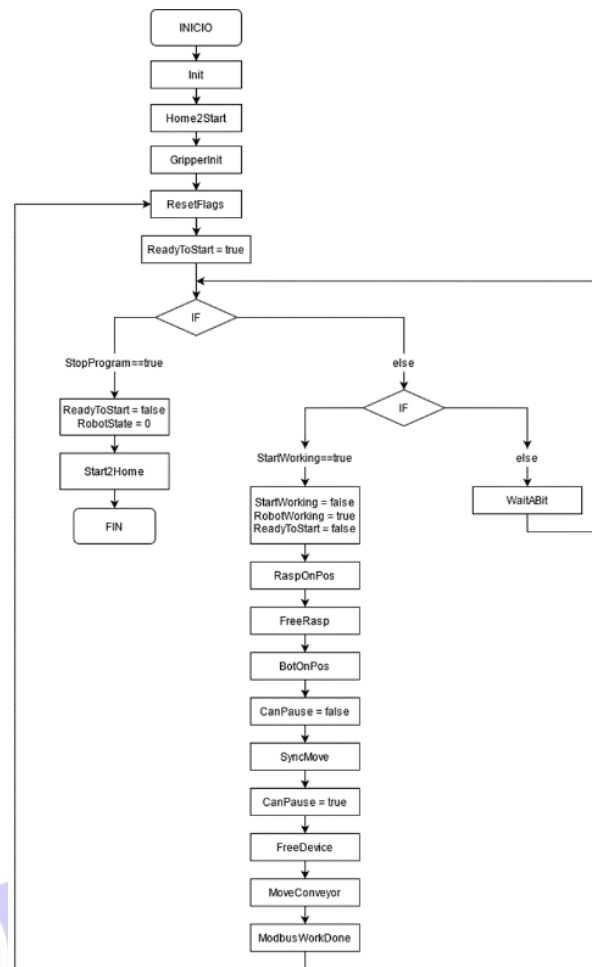


Figura 180. Flujograma simplificado del programa final del Omron TM5-900

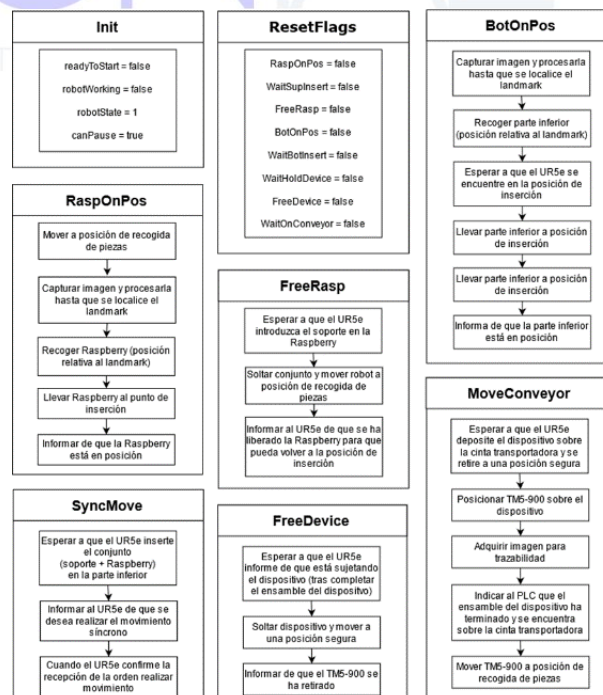
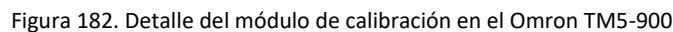


Figura 181. Detalle de algunas de las subrutinas del flujograma del Omron TM5-900



El PLC gestiona el flujo de trabajo de los robots, siendo parecida la interacción de ambos robots con el PLC. Como se explicó, el PLC da la orden de inicio de programa, y si todo está correcto el programa de ambos robots comienza, pasando de la posición de Home a la posición de inicio, inicializando las garras y algunas variables de programa. Una vez los robots llegan a la posición de inicio se quedan a la espera de una orden del PLC, que puede ser un nuevo proceso de ensamblado o la finalización del programa. Por tanto, el PLC tiene la capacidad de:

- Se definieron en el PLC las variables de comunicación con el TM5-900 y con el UR5e. Una vez definidas estas variables e implementado el flujo de trabajo en el PLC, se validó la comunicación y sincronización de todo el sistema. Además, en esta tarea se han realizado las integraciones necesarias asociadas a los robots desarrolladas desde la tarea T4.2.

Página 155 de 163

armario sale el mensaje cuando hay un obstáculo en la zona de advertencia o en la de protección (Figura 153 y Figura 183), tal y como se esperaba.

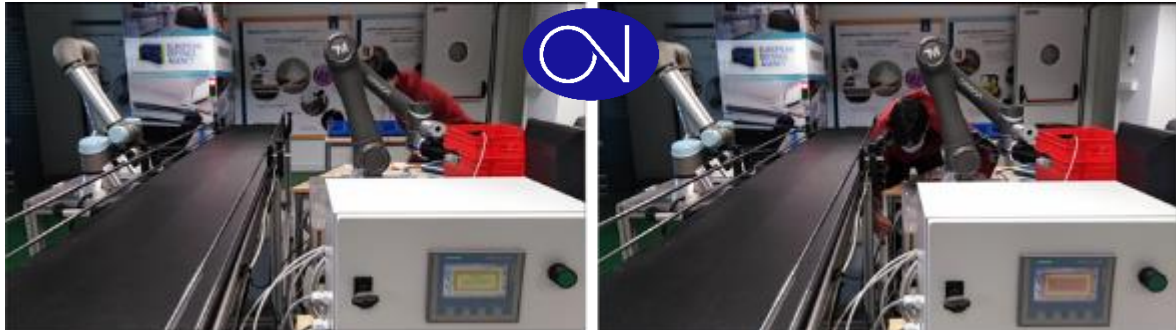


Figura 183. Validación de la aparición de los mensajes de advertencia (izqda.) y peligro (dcha.) en el HMI del armario cuando hay un obstáculo en las zonas monitorizadas por el escáner

Posteriormente se realizó la calibración de ambos robots para conocer la posición del UR5e frente al Omron TM5-900, adaptándose así la posición de inserción a la ubicación actual del UR5e. En la Figura 184 se muestra cómo se posicionan los dos robots para que la cámara del Omron TM5-900 pueda visualizar el landmark sujeto por el UR5e. La Figura 185 recoge el proceso programado en TMFlow (con TMvision) y “Vision Designer” para la detección del marcador y obtención de su posición respecto al robot Omron.



Figura 184. Imágenes representativas de la calibración de ambos robots



Figura 185. Detalle del proceso de calibración en TMFlow (equipado con TMvision) y Vision Designer

Posteriormente se revisa la configuración del gripper y estado de cada robot para iniciar las pruebas para la validación del Demostrador 3. Así, se realizaron las pruebas del sistema completo para validar la realización del montaje del dispositivo asociado a este Demostrador 3 en base a los flujogramas y programas comentados en tareas anteriores. Para ello, primeramente el usuario debe asegurarse de colocar los componentes necesarios en la zona de trabajo del robot (Figura 186), sobre la base específica diseñada para cada uno de los robots.





Figura 186. Colocación de los componentes a ensamblar

En el HMI del armario, si el usuario no se encuentra en la interfaz del Demostrador 3, debe lanzarla seleccionando el botón “DEMO 3” (Figura 187), revisar que el estado de los robots sea correcto (Figura 188, Figura 189) y configurar el modo de funcionamiento de la cinta (dirección y velocidad) así como indicar el número de dispositivos a ensamblar. La Figura 190 muestra la configuración de la cinta seleccionada y cómo se introduce el valor 1 actualizándose el valor 0 a 1 en ambos campos asociados de piezas.



Figura 187. Selección de Demo 3 en la interfaz

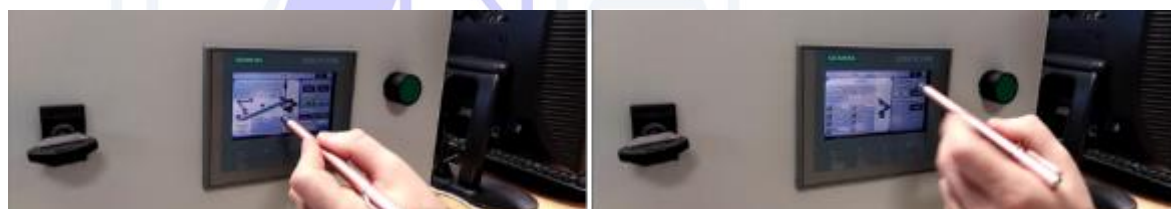


Figura 188. Revisión del estado del robot Omron TM5-900



Figura 189. Revisión del estado del robot UR5e



Figura 190. Configuración del modo de funcionamiento de la cinta y el número de dispositivos a ensamblar, en este caso, un dispositivo

Una vez todo arrancado y configurado, se pulsa el botón START para iniciar el proceso de ensamblaje (Figura 191), en este caso de 1 pieza.



Figura 191. Detalle de la configuración seleccionada (izqda.) y acción de iniciar la Demo 3 (dcha.)

Cuando los robots reciben la orden del PLC realizan un proceso de inicialización y pasan de la posición de "home" (Figura 192, izquierda) a la posición de "inicio" (Figura 192, derecha) para comenzar el proceso de montaje del dispositivo. Como se aprecia en la Figura 192, primeramente el UR5e va a la posición inicial (centro) y posteriormente se mueve el Omron TM5-900 (derecha).

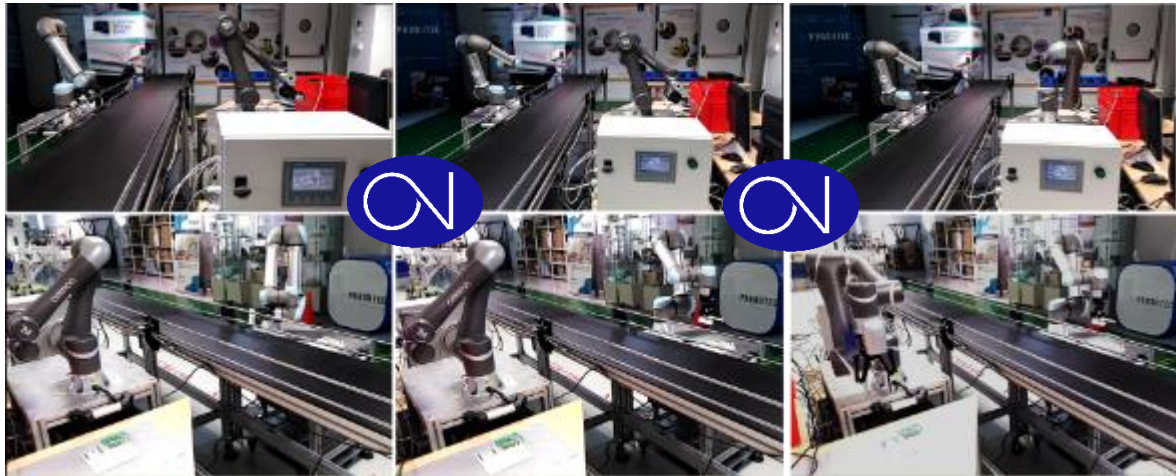


Figura 192. Secuencia de inicio de los robots, mostrando capturas del mismo instante desde puntos de vista enfrentados: desde el armario (arriba) desde el extremo de la cinta con baliza (abajo)

Una vez inicializados correctamente comienzan los dos robots a realizar de forma sincronizada las tres fases comentadas en tareas anteriores: 1. Ensamblaje, 2. Atornillado y 3. Inspección. Cuando el TM5-900 se encuentra en la posición de recogida de piezas emplea la cámara para localizar el landmark de su base y poder calcular en función de él la posición de recogida de las distintas piezas. De esta forma, puede cogerlas sin necesidad de colocar la base en una posición fija. En la Figura 193 se muestra el proceso desde la captura de la imagen, pasando por la aplicación de diversos filtros (realce del contraste, suavizado) hasta la detección del marcador, mostrando como etapa final el tiempo de ejecución. En cambio, el UR5e al no disponer de cámara, funciona con posiciones prefijadas. Cabe destacar que se hace uso del sensor de par-fuerza del UR5e para insertar las piezas y los tornillos.

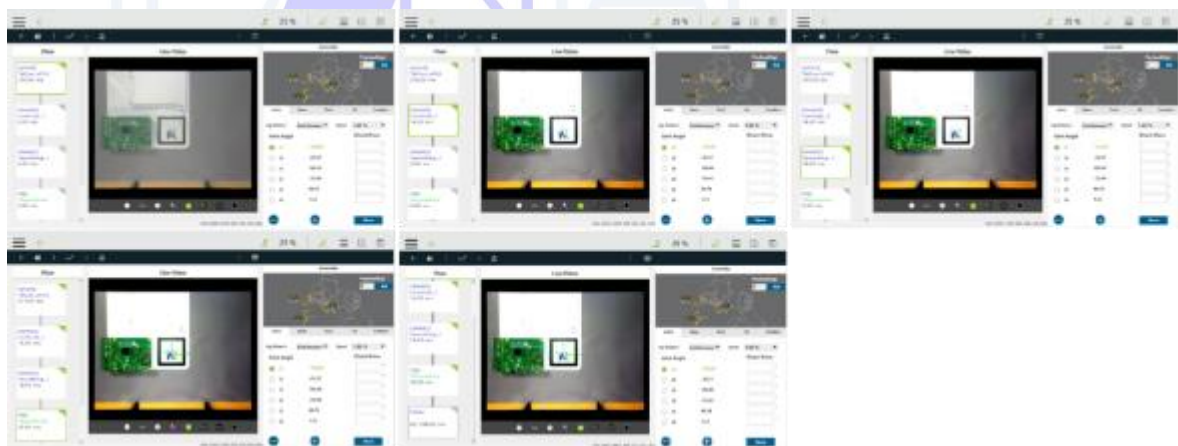


Figura 193. Proceso de detección del landmark para la localización de la base del Omron TM5-900

A continuación se muestran una serie de imágenes representativas del proceso de montaje de un dispositivo:

1) Ensamblaje:

1) Soporte y Raspberry (la PCB):

- a. Una vez que el TM5-900 ha localizado el marcador, el robot recoge la Raspberry, ya que ya conoce la posición de todos los objetos colocados en su base.
- b. Mientras tanto el UR5e recoge el soporte de la carcasa.

- c. Ambos se dirigen al punto de ensamblado, esperando el UR5e a que el TM5-900 le indique que se encuentra en posición.
- 2) Inserción de Raspberry:
 - a. El UR5e inserta el soporte y el TM5-900 suelta la Raspberry.
 - b. Cuando la Raspberry ha sido liberada, el UR5e se mantiene en la posición de inserción, a la vez que el TM5-900 se dirige a recoger la parte inferior de la carcasa.
- 3) Carcasa inferior:
 - a. El TM5-900 lleva la parte inferior de la carcasa a la posición de inserción.
 - b. El UR5e inserta el soporte y la raspberry en la parte inferior.
- 4) Cambiar la orientación del conjunto de forma sincronizada.
- 5) Carcasa superior: una vez el movimiento sincronizado termina, el UR5e recoge e inserta la parte superior sobre el dispositivo sujetado por el TM5-900 (Figura 194):
 - a. El UR5e suelta el conjunto.
 - b. El UR5e se dirige a recoger la parte superior de la carcasa. La lleva hasta el punto de inserción. La inserta en el conjunto y la suelta para realizar el atornillado.



Figura 194. Proceso de ensamblaje: etapa 5

2) Atornillado:

- 6) El UR5e recoge e inserta los 4 tornillos sobre el conjunto sujeto por el Omron TM5-900 (Figura 195): una vez insertada la parte superior, el UR5e se dirige a por un tornillo, lo lleva hasta el punto de inserción, lo coloca en el agujero correspondiente, lo empuja para asegurarse que ha encontrado correctamente y lo atornilla girando varias veces según su ángulo máximo, repitiendo el proceso cuatro veces, una por tornillo.



Figura 195. Atornillado del primer tornillo

3) Inspección:

- 7) El UR5e sujeta el dispositivo.
- 8) El TM5-900 suelta el dispositivo.
- 9) El UR5e deposita el conjunto en la cinta transportadora, de esta forma la parte superior del dispositivo queda mirando hacia arriba (Figura 196).
- 10) El TM5-900 captura una imagen de trazabilidad (Figura 197) cuando el objeto está sobre la cinta y el UR5e se ha apartado. Tras ello, informa al PLC de que debe llevarse la pieza por lo que la cinta se pone en movimiento a la velocidad y en la dirección definida al lanzar la Demo, dejando libre la zona de trabajo de los robots.



Figura 196. Proceso de inspección: etapas 7-9



Figura 197. Resultado del proceso de trazabilidad: imagen adquirida (izqda.) y xml generado (dcha.)

Al haber indicado al inicio del proceso que solo había que ensamblar un dispositivo, los robots vuelven a su posición home durante el movimiento de la cinta (Figura 198) y la interfaz se actualiza indicando que hay 0 dispositivos restantes a realizar (Figura 199). En caso de existir otro dispositivo a ensamblar se repite la secuencia de ensamblado, ejecutándose tantas veces como el usuario haya indicado que hay dispositivos a ensamblar y restando uno en los dispositivos restantes. Si el usuario desea parar debe pulsar STOP en la interfaz.

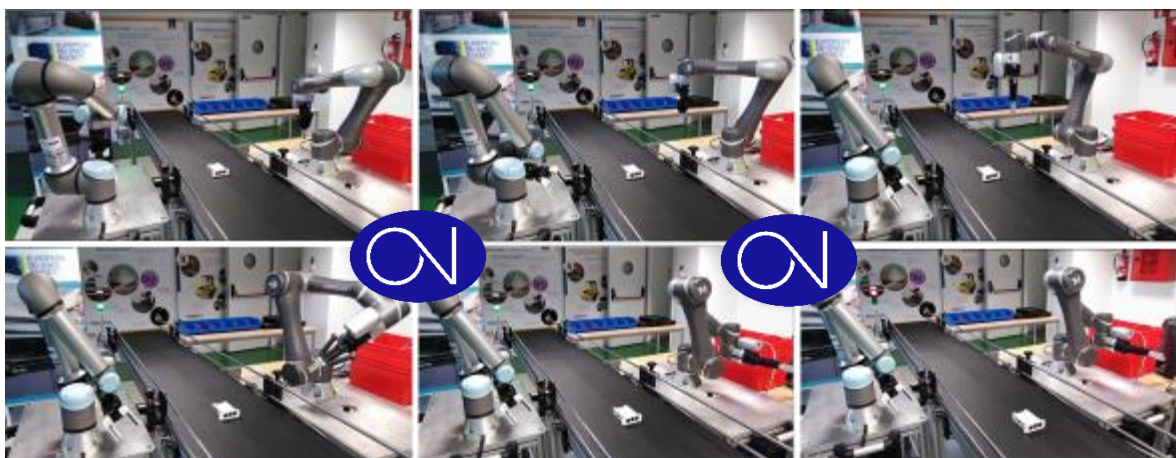


Figura 198. Secuencia de movimiento del dispositivo sobre la cinta y movimiento de los robots a su posición de home



Figura 199. Fin de la ejecución de montaje de un dispositivo

En las numerosas pruebas realizadas se ha comprobado la correcta ejecución de todo el proceso de ensamblaje así como el comportamiento adecuado de reducción de velocidad o parada de los elementos cuando había un obstáculo en la zona de advertencia o de protección.



4 CONCLUSIONES

A continuación se resumen las principales conclusiones asociadas a las actividades realizadas en este proyecto. En primer lugar cabe destacar que durante el HITO 1 se realizó un importante estado del arte que ayudó al equipo a conocer aún más la aplicación de la robótica colaborativa en el ámbito industrial, además de comprender mejor el concepto de gemelo digital. Por otro lado, en dicho hito se definieron los requisitos de los tres demostradores a desarrollar, ayudándose de la tecnología existente en el Centro de simulación de procesos, de cara a optimizar los escenarios definidos para cada demostrador.

En el HITO 2 se ha logrado crear un gemelo digital del Laboratorio de Robótica Colaborativa sobre el que se basan los tres demostradores definidos en el HITO 1. En este sentido, se han creado los modelos virtuales de los diferentes manipuladores robóticos así como elementos adicionales y necesarios del escenario; se han desarrollado las ecuaciones cinemáticas y dinámicas de los robots y se han implementado; se ha diseñado y desarrollado una interfaz de usuario desde la cual los usuarios puedan interactuar con el gemelo digital. Por último, todo ello se integró, validó y probó y gracias al feedback aportado por los usuarios, se han podido definir líneas de mejora.

En el HITO 3 e HITO4, se han desarrollado demostradores reales asociados a la robótica colaborativa. El primero de ellos, estaba más enfocado a la colaboración persona – máquina, mientras que el segundo lo estaba hacia la colaboración máquina - máquina. Para el desarrollo de ambos demostradores, el planteamiento ha sido similar si bien cada uno se ha debido ajustar a su condición particular: 1. Analizar y revisar el proceso asociado a cada uno de ellos y los elementos necesarios y diseñar, adquirir o fabricar estos; 2. Definir, desarrollar e implementar los sistemas de percepción para la tarea (Demostrador 2) y de sincronización (Demostrador 3); 3. Programar el o los robot(s) para cada escenario según el demostrador de interés; 4. Desarrollar la interfaz de usuario, que obviamente en el caso del Demostrador 2 tiene una mayor relevancia; 5. Integración, pruebas y validación. En ambos casos, ha sido necesaria la última tarea para poder realizar un buen ajuste de todos los elementos y para comprobar con los operarios el grado de aceptación y la identificación de mejoras futuras. En este sentido, para el caso del Demostrador 2 relativo a colaboración persona – robot, esto ha sido esencial.