



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

**SIMULACIÓN DE UNA INSTALACIÓN ROBOTIZADA DE
SOLDADURA DE PIEZAS, CONTROLADA POR PLC,
GESTIONADA CON UNA INTERFAZ HOMBRE-MÁQUINA Y
COMUNICADA MEDIANTE OPC UA.**

Autor:

Villa González, Jorge

Tutor:

**Herreros López, Alberto
Departamento de Ingeniería de
Sistemas y Automática**

Valladolid, enero de 2023.



Agradecimientos

Para comenzar este proyecto quería agradecer a mi tutor D. Alberto Herreros López, por la aportación de ideas para la realización de un proyecto completo que ampliara mis conocimientos.

A Automatinfo S.L. por darme la oportunidad de trabajar y aprender de ellos, por cedermme sus equipos para la realización de este proyecto y por la aportación de proyectos similares y la documentación correspondiente para ayudarme a encauzar este trabajo.

A mis amigos, compañeros y personas que me acompañaron durante estos años por el apoyo moral y académico que me han aportado durante todos estos años. Ya que sin ellos este proceso no hubiera sido igual de llevadero.

Y en especial a mi familia, quienes viven el día a día junto a mí, quienes están en los buenos y malos momentos para levantarme el ánimo y darme confianza. Gracias por darme la oportunidad y apoyarme de manera incondicional.



Universidad de Valladolid

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Jorge Villa González



ESCUELA DE INGENIERÍAS
INDUSTRIALES



Resumen

En este Trabajo de Fin de Grado se pretende poner en práctica los conocimientos adquiridos en el grado y en las prácticas en empresa mediante el diseño y desarrollo de una instalación robotizada controlada por un autómata programable (PLC) y gestionada a nivel de usuario con una interfaz hombre-máquina cuya comunicación con la instalación se realiza mediante el estándar de comunicaciones industriales OPC UA.

Este proyecto puede suponer un punto de partida para el desarrollo de futuros proyectos debido a la variedad de tecnologías de la industria 4.0 que agrupa. Además de ser un aporte de información acerca del uso de tecnologías de la marca FANUC.

Para el desarrollo de la estación se ha utilizado el programa de simulación RoboGuide de FANUC, que incluye los robots de su catálogo para la realización de las maniobras. La instalación se controla con un PLC SIEMENS S7-1500, simulado desde la herramienta PLCSIM y programado desde TIA Portal. Por último, se añade una interfaz hombre-máquina para la gestión y supervisión de la planta, desarrollada en TIA Portal y ejecutada en WinCC.

Palabras Clave

Robótica, FANUC, RoboGuide, HandlingPro, Line Tracking, Eje Extendido, Automatización, SIEMENS, TIA Portal, WinCC, PLC, HMI, KEPServerEX, OPC UA, Servidor OPC, Visión Artificial.



Universidad de Valladolid

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Jorge Villa González



ESCUELA DE INGENIERÍAS
INDUSTRIALES



Abstract

In this Final Degree Project, it is intended to put into practice the knowledge acquired throughout the degree and the internship by designing and developing a robotic workcell, controlled by a programmable logic controller (PLC) and managed at the user level by a human-machine interface – using OPC UA industrial communications standard as a way to connect the interface to the workcell.

This project may be a starting point to developing future projects due to Industry 4.0 technologies variety included in it. Furthermore, it is a contribution of information about the FANUC technologies usage.

For developing the workcell; Roboguide, the simulation program provided by FANUC, has been used along with the robots from its catalogue. In addition, the workcell control has been assigned to a SIEMENS S7-1500 PLC, simulated from PLCSIM and programmed with TIA Portal. Finally, the human-machine user interface, which has been designed from scratch on TIA PORTAL and executed on WinCC, is added to managing and supervising the process plant.

Keywords

Robotics, FANUC, RoboGuide, HandlingPro, Line Tracking, Extended Axis, Automation, SIEMENS, TIA Portal, WinCC, PLC, HMI, KEPServerEX, OPC UA, OPC Server, Artificial Vision.



Universidad de Valladolid

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Jorge Villa González



ESCUELA DE INGENIERÍAS
INDUSTRIALES



ÍNDICE DE CONTENIDOS

CAPÍTULO 1. INTRODUCCIÓN	1
Capítulo 1.1. Motivación, origen y justificación del proyecto	1
Capítulo 1.2. Objetivos.....	2
Capítulo 1.3. Fundamentos previos.....	2
Capítulo 1.4. Alcance del proyecto	3
Capítulo 1.5. Estructura del trabajo.....	3
Capítulo 1.6. Convenciones.....	4
CAPÍTULO 2. MARCO TEÓRICO	5
Capítulo 2.1. Robótica	5
Capítulo 2.1.1. Historia de la Robótica.....	5
Capítulo 2.1.2. Robótica Industrial	7
Capítulo 2.1.3. Definición y clasificación de robots.....	8
Capítulo 2.1.4. Robots manipuladores industriales. Características	9
Capítulo 2.1.5. Robots manipuladores industriales. Ejemplos.....	10
Capítulo 2.2. Automatización	11
Capítulo 2.2.1. Automatización industrial	12
Capítulo 2.2.2. Controlador Lógico Programable (PLC)	13
Capítulo 2.2.3. Interfaz Hombre-Maquina (HMI).....	15
Capítulo 2.3. Comunicaciones industriales.....	16
Capítulo 2.3.1. Niveles de comunicación en una red industrial	16
Capítulo 2.3.2. Protocolos de comunicación industrial.....	17
Capítulo 2.3.3. OPC.....	18
Capítulo 2.4. Visión artificial	18
Capítulo 2.4.1. Detección y reconocimiento de objetos.....	20
CAPÍTULO 3. ESTADO DEL ARTE.....	21
Capítulo 3.1. Antecedentes	21
Capítulo 3.2. Análisis del problema	21
Capítulo 3.3. RoboGuide	22
Capítulo 3.3.1. HandlingPro	24



Capítulo 3.3.2. Teach Pendant.....	28
Capítulo 3.3.3. iRVision	29
Capítulo 3.4. Siemens	31
Capítulo 3.4.1. TIA Portal.....	31
Capítulo 3.4.2. PLCSim.....	35
Capítulo 3.4.3. WinCC.....	38
Capítulo 3.4.4. NetToPLCSim.....	39
Capítulo 3.5. Comunicaciones	41
Capítulo 3.5.1. KepServerEX6.....	41
Capítulo 3.5.2. Interfaz de KepServerEX6.....	42
CAPÍTULO 4. DISEÑO DE LA SOLUCIÓN ADOPTADA.....	45
Capítulo 4.1. Estudio de seguridad en entornos robotizados	45
Capítulo 4.1.1. Potenciales riesgos de la robótica industrial.....	45
Capítulo 4.1.2. Normativas de seguridad en entornos robotizados industriales	46
Capítulo 4.1.3. Medidas de seguridad de diseño en entornos robotizados industriales.....	47
Capítulo 4.2. Descripción de los elementos de seguridad.....	50
Capítulo 4.2.1. Barreras materiales	50
Capítulo 4.2.2. Barreras inmateriales	51
Capítulo 4.3. Descripción de los objetos a manipular.....	51
Capítulo 4.3.1. Placa solar	51
Capítulo 4.3.2. Elementos de fijación.....	52
CAPÍTULO 5. CONFIGURACIÓN DE LOS ENTORNOS DE TRABAJO.....	53
Capítulo 5.1. Desarrollo en HandlingPro	53
Capítulo 5.1.1. Programación	53
Capítulo 5.1.1.1. Comandos de movimiento.....	54
Capítulo 5.1.1.2. Otros comandos	55
Capítulo 5.1.1.3. RoboGuide Simulation Programm Editor.....	57



Capítulo 5.1.1.4. Line Tracking	57
Capítulo 5.1.2. Sincronización de señales	64
Capítulo 5.1.3. User frames	65
Capítulo 5.1.4. Desarrollo de la estación robotizada	66
Capítulo 5.1.4.1. Robot1	67
Capítulo 5.1.4.2. Robot2	67
Capítulo 5.1.4.3. Robot3	68
Capítulo 5.1.5. Configuración de un eje extendido	69
Capítulo 5.1.5.1. Configuración del controlador	69
Capítulo 5.1.5.2. Configuración del eje extendido en la estación	73
Capítulo 5.1.5.3. Programación de un robot con eje extendido	76
Capítulo 5.1.6. Configuración de un proceso de visión artificial	78
Capítulo 5.1.6.1. Configuración previa del robot	78
Capítulo 5.1.6.2. Cámara 2D de visión artificial	79
Capítulo 5.1.6.3. Asociación robot - cámara	81
Capítulo 5.1.6.4. Calibración de la cámara	82
Capítulo 5.1.6.5. Proceso de visión artificial	83
Capítulo 5.1.6.6. Programación de visión artificial en Teach Pendant	86
Capítulo 5.2. Desarrollo en TIA Portal	89
Capítulo 5.2.1. Interfaz hombre - máquina (HMI)	89
Capítulo 5.2.1.1. Plantilla e imágenes	90
Capítulo 5.2.1.2. Programación de la interfaz	95
Capítulo 5.2.2. Autómata programable (PLC)	99
Capítulo 5.2.2.1. Bloques de organización (OB)	100
Capítulo 5.2.2.2. Funciones (FC)	101
Capítulo 5.3. Desarrollo en KEPServerEX	102
Capítulo 5.3.1. Configuración del servidor OPC UA	102



Capítulo 5.4. Conexiones entre programas.....	106
Capítulo 5.4.1. HandlingPro	106
Capítulo 5.4.1.1. Conexión HandlingPro – Servidor OPC UA.....	106
Capítulo 5.4.1.2. Configuraciones de los controladores, señales y programas.....	109
Capítulo 5.4.2. TIA Portal.....	111
Capítulo 5.4.3. KEPServerEX.....	113
CAPÍTULO 6. PRUEBAS Y RESULTADOS.....	115
CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS.....	129
Capítulo 7.1. Conclusiones.....	129
Capítulo 7.2. Líneas futuras.....	130
BIBLIOGRAFÍA.....	131
ANEXOS.....	135



ÍNDICE DE FIGURAS

FIGURA 1. Los Pájaros de Herón. Herón de Alejandría	5
FIGURA 2. Caballero de Leonardo da Vinci. Manuscritos del ingeniero italiano	6
FIGURA 3. Telar de Jacquard y Tarjeta Perforada	6
FIGURA 4. Robot UNIMATE	7
FIGURA 5. Gama de robots PUMA	7
FIGURA 6. Robots antropomórficos	10
FIGURA 7. Robot SCARA	10
FIGURA 8. Robot cartesiano	11
FIGURA 9. Robot Delta.....	11
FIGURA 10. AGV	11
FIGURA 11. Pirámide de la automatización industrial	12
FIGURA 12. Autómatas Programables.....	14
FIGURA 13. Arquitectura interna de un PLC	14
FIGURA 14. Interfaces Hombre-Máquina	16
FIGURA 15. OPC como capa de abstracción entre Cliente y Servidor	18
FIGURA 16. Visión Artificial: montada sobre robot y fija.....	23
FIGURA 17. Dual Check Safety.....	23
FIGURA 18. Interfaz HandlingPro	24
FIGURA 19. Barra de herramientas de HandlingPro	24
FIGURA 20. Árbol del proyecto: Controladores de los robots.....	26
FIGURA 21. Árbol del proyecto: Máquinas	26
FIGURA 22. Árbol del proyecto: Accesorios	26
FIGURA 23. Árbol del proyecto: Objetos	26
FIGURA 24. Árbol del proyecto: Obstáculos	27
FIGURA 25. Árbol del proyecto: Dispositivos Externos	27
FIGURA 26. Árbol del proyecto: Sensores	27
FIGURA 27. Biblioteca de objetos CAD	28
FIGURA 28. Teach Pendant física y de simulación	29
FIGURA 29. Robot con visión artificial	29
FIGURA 30. IRVision: Robot > Internet Explorer	30
FIGURA 31. IRVision: Robot Homepage > iRVision.....	30
FIGURA 32. IRVision: Vision Setup.....	31



FIGURA 33. TIA Portal: Vista del portal	32
FIGURA 34. TIA Portal: Vista del proyecto	32
FIGURA 35. TIA Portal: Agregar dispositivo.....	33
FIGURA 36. TIA Portal: Barra de herramientas	33
FIGURA 37. TIA Portal: Área de Trabajo.....	34
FIGURA 38. TIA Portal: Herramientas de configuración	34
FIGURA 39. Ventana de Propiedades, Información y Diagnóstico	35
FIGURA 40. PLCSim: Iniciar simulación.....	35
FIGURA 41. Interfaz PLCSim y ventana de búsqueda del PLC virtual	36
FIGURA 42. Compilación y carga en PLC virtual	37
FIGURA 43. Arrancar PLC virtual.....	37
FIGURA 44. PLC virtual en funcionamiento.....	37
FIGURA 45. Interfaz PLCSim en vista extendida.....	38
FIGURA 46. HMI en WinCC	39
FIGURA 47. NetToPLCSim	39
FIGURA 48. NetToPLCSim: Interfaz.....	40
FIGURA 49. NetToPLCSim: Configuración de una estación	40
FIGURA 50. NetToPLCSim: Estación en ejecución	41
FIGURA 51. KEPServerEX	41
FIGURA 52. Interfaz de KEPServerEX6	42
FIGURA 53. KEPServerEX: OPC Quick Client	42
FIGURA 54. Marca de certificación o marcado europeo	46
FIGURA 55. Barreras materiales.....	48
FIGURA 56. Barreras inmateriales.....	48
FIGURA 57. Señalizaciones audibles y visibles.....	49
FIGURA 58. Sistemas de mando y de parada	49
FIGURA 59. Barreras materiales y dimensiones.....	50
FIGURA 60. Barreras inmateriales y dimensiones.....	51
FIGURA 61. Placa solar y dimensiones.....	52
FIGURA 62. Piezas1 y Piezas2	52
FIGURA 63. Placa solar con elementos de fijación	52
FIGURA 64. Terminación del movimiento.....	55
FIGURA 65. Ejemplo de programación en Teach Pendant	56
FIGURA 66. Programa en Simulation Program Editor	57
FIGURA 67. Opciones de software de Line Tracking	58



FIGURA 68. Line Tracking: Línea.....	58
FIGURA 69. Line Tracking: Objeto.....	58
FIGURA 70. Line Tracking: Configuración del objeto.....	59
FIGURA 71. Line Tracking: Configuración de la cinta.....	59
FIGURA 72. Line Tracking: Selección del objeto.....	59
FIGURA 73. Line Tracking: Pruebas de movimiento.....	60
FIGURA 74. Line Tracking: Line Schedule - General.....	60
FIGURA 75. Line Tracking: Line Schedule - Boundaries.....	61
FIGURA 76. Line Tracking: Line Schedule - Frame.....	61
FIGURA 77. Line Tracking: Line Track Program Header.....	62
FIGURA 78. Programación del Line Tracking.....	63
FIGURA 79. Interconexión de señales digitales.....	64
FIGURA 80. Robot1 (DO5) se coordina con Robot3 (DI4).....	65
FIGURA 81. Robot3 (DO1) se coordina con Robot2 (DI1).....	65
FIGURA 82. Celda robotizada.....	66
FIGURA 83. Robot1: R-2000iC/125L.....	67
FIGURA 84. Robot2: ARC Mate 100iD.....	67
FIGURA 85. Robot3: M-10iA/8L.....	68
FIGURA 86. Workcell Creation Wizard: Extended Axis Control.....	69
FIGURA 87. Teach Pendant de configuración de un controlador.....	69
FIGURA 88. Mantenace y configuración de un eje extendido.....	70
FIGURA 89. Eje extendido: Programa de configuración (I).....	70
FIGURA 90. Eje extendido: Programa de configuración (II).....	71
FIGURA 91. Fin de la configuración del controlador.....	72
FIGURA 92. Comprobación de la configuración del eje extendido.....	72
FIGURA 93. Raíl.....	73
FIGURA 94. Raíl en el árbol del proyecto.....	73
FIGURA 95. Asociación raíl – séptimo eje del robot.....	74
FIGURA 96. Raíl – Motor – Robot.....	74
FIGURA 97. Desactivación del movimiento de la articulación 8.....	75
FIGURA 98. TCP (bola verde) no acompaña al movimiento del raíl.....	75
FIGURA 99. TCP (bola verde) acompaña al movimiento del raíl.....	76
FIGURA 100. Registro de posición de un robot sin eje extendido.....	76
FIGURA 101. Registro de posición de un robot con eje extendido.....	77



FIGURA 102. Ejemplo de programación de un robot con eje extendido.....	77
FIGURA 103. Pos_intermedia	77
FIGURA 104. Pos_soldar	78
FIGURA 105. Opciones de software de iRVision	79
FIGURA 106. Icono de visión.....	79
FIGURA 107. Simulación de visión artificial habilitada	79
FIGURA 108. Agregar cámara 2D de visión artificial	80
FIGURA 109. Camera_BW	80
FIGURA 110. Cámara agregada al árbol del proyecto.....	80
FIGURA 111. Parámetros de la cámara de visión artificial	81
FIGURA 112. Asociación Robot-Cámara.....	81
FIGURA 113. 2D Camera Data: Camera Setup	82
FIGURA 114. 2D Camera Data: Calibration.....	82
FIGURA 115. 2D Camera Data: Márgenes de calibración.....	83
FIGURA 116. 2D Camera Data: Resultados de la calibración	83
FIGURA 117. 2-D Single View Vision Process	84
FIGURA 118. Parámetros del proceso de visión.....	84
FIGURA 119. Editor del GPM Locator Tool 1	85
FIGURA 120. Enseñanza de la pieza tipo 1.....	85
FIGURA 121. Identificación tras enseñanza	86
FIGURA 122. Ejemplo de identificación de piezas de fijación	86
FIGURA 123. Instrucciones de programación de iRVision	87
FIGURA 124. Programa de visión artificial	87
FIGURA 125. Programa de tomar objeto usando visión artificial	89
FIGURA 126. HMI: Plantilla.....	90
FIGURA 127. HMI: Imagen HOME.....	91
FIGURA 128. HMI: Imagen CONTINUA.....	91
FIGURA 129. HMI: Imagen ERRORES Y SEÑALES	92
FIGURA 130. HMI: Imagen INDIVIDUAL.....	94
FIGURA 131. HMI: Imagen AJUSTES.....	95
FIGURA 132. Ejemplos de eventos.....	96
FIGURA 133. Opciones de activación de eventos.....	96
FIGURA 134. Ejemplo de programación de eventos	97
FIGURA 135. Ejemplo de programación de visibilidad	97
FIGURA 136. Botón en reposo vs botón pulsado	98



FIGURA 137. Señalización con parpadeo	98
FIGURA 138. Ventana de propiedades: programación de elementos de una HMI	98
FIGURA 139. Ejemplo de lenguaje KOP	99
FIGURA 140. Instrucciones adicionales	100
FIGURA 141. OB de arranque	101
FIGURA 142. OB Main.....	101
FIGURA 143. FC Continua	102
FIGURA 144. FC Errores y Señales	102
FIGURA 145. FC Individual	102
FIGURA 146. Crear canal de comunicación	102
FIGURA 147. Siemens TCP/IP Ethernet	103
FIGURA 148. Nombre del canal	103
FIGURA 149. Adaptador de red.....	104
FIGURA 150. Canal creado.....	104
FIGURA 151. Añadir dispositivo	104
FIGURA 152. Identificador del PLC	104
FIGURA 153. Modelo del PLC	105
FIGURA 154. Dirección IP del PLC	105
FIGURA 155. Añadir etiqueta	105
FIGURA 156. Configuración de la etiqueta	106
FIGURA 157. Estructura de la conexión entre programas	106
FIGURA 158. Comunicaciones en HandlingPro	107
FIGURA 159. Conexión con servidor en HandlingPro	107
FIGURA 160. Conexiones externas	108
FIGURA 161. UOP in.....	108
FIGURA 162. UOP out	109
FIGURA 163. Sistema en remoto	109
FIGURA 164. Rack y Slot de las señales digitales	110
FIGURA 165. Configuración de la ejecución (I).....	110
FIGURA 166. Configuración de la ejecución (II).....	111
FIGURA 167. Configuración de la ejecución (III)	111
FIGURA 168. Asignación de dirección IP del PLC	112
FIGURA 169. Configuración de la interfaz de red de PLCSim	113
FIGURA 170. Permiso de acceso de interlocutor remoto.....	113
FIGURA 171. Inicio del servidor OPC UA.....	114



FIGURA 172. Botón de MARCHA.....	115
FIGURA 173. Secuencia de movimientos del Robot1 (I).....	117
FIGURA 174. Secuencia de movimientos de los robots 2 y 3	121
FIGURA 175. Secuencia de movimientos del Robot1 (II)	126



ÍNDICE DE TABLAS

TABLA 1. Clasificación de robots según su cronología	8
TABLA 2. Clasificación de robots según su función.....	8
TABLA 3. Clasificación de robots según su autonomía	9
TABLA 4. Ventajas y Desventajas de la Automatización Industrial	12
TABLA 5. Visión artificial según su tecnología	19
TABLA 6. Visión artificial según la calidad de imagen.....	19



Universidad de Valladolid

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Jorge Villa González



ESCUELA DE INGENIERÍAS
INDUSTRIALES



CAPÍTULO 1. INTRODUCCIÓN

Capítulo 1.1. Motivación, origen y justificación del proyecto.

La robótica es el área de conocimiento por el cual me matriculé en el grado de Ingeniería Electrónica Industrial y Automática. Durante el tramo final del grado, fui ampliando mi curiosidad gracias a asignaturas como Sistemas Robotizados y Taller de Robótica Industrial. Por otro lado, la programación de robots manipuladores resulta ser de gran importancia en entornos industriales en la actualidad; y es por ello por lo que decidí realizar un proyecto relacionado con la robótica industrial.

Por otro lado, desarrollé un gusto por la automatización industrial durante el transcurso del grado. La robótica y la automatización están muy ligadas ya que para la inclusión de robots en procesos industriales se requiere de un control automatizado de los mismos.

Es por todo ello que realicé las prácticas en Automatinfo S.L.: una empresa especializada en automatización industrial e industria 4.0. Allí me han dado la posibilidad de continuar desarrollando mis conocimientos en programación de robots FANUC y simulación off-line de procesos donde estos se incluyen. Además, en Automatinfo S.L. son expertos en autómatas programables de principales marcas como SIEMENS, SCHENIDER o ALLEN BRADLEY; por lo que he continuado mi aprendizaje en programación KOP y programación de interfaces hombre-máquina.

Uno de los principales objetivos es poner en práctica las habilidades y conocimientos adquiridos durante el grado y las prácticas en empresa con respecto a diferentes ramas de la ingeniería como son la robótica, el control, la automatización o la visión artificial.

El proyecto surge a raíz de las tareas realizadas durante las prácticas en Automatinfo S.L. Esta empresa, entre otras cosas, realiza proyectos en líneas continuas de producción que incluyen múltiples robots trabajando en sincronía. Éstos intercambian señales para trabajar de forma conjunta y requieren de técnicas de programación de trayectorias que realicen el seguimiento posicional de los elementos de la línea. Para la interacción con los operarios realizan interfaces hombre-máquina (HMI) con diversas funcionalidades para que se pueda trabajar de forma eficaz y segura.



Se plantea, pues, la realización de un proyecto de simulación similar a los que realiza la empresa que aglutine el trabajo de los robots (programados con técnicas avanzadas) con el de un PLC que se encargue de la gestión de la seguridad de la estación y del control de las señales enviadas desde una HMI.

Generalmente, cada fabricante de robots tiene sus propios autómatas para controlar la planta donde se incluyen sus robots. Sin embargo, hay ocasiones donde se tiene material de diferentes marcas que tiene que comunicarse con los propios robots u otros dispositivos. La problemática surge cuando estos no tengan una conectividad nativa.

Por lo tanto, durante el presente informe, se van a exponer y analizar los mecanismos desarrollados para la creación y control de una celda robotizada en un entorno de simulación que solventa el problema de conectividad expuesto en el párrafo anterior para emular el trabajo de la misma en un entorno industrial real.

Capítulo 1.2. Objetivos.

- Servir como herramienta de estudio de celdas robotizadas en entornos simulados de la marca FANUC. La cual se hace cada vez más presente en industrias cercanas a la Universidad de Valladolid.
- Diseño y desarrollo un proyecto de simulación que aúne distintas ramas de la ingeniería.
- Manejo y comunicación de múltiples robots que trabajen conjunta y sincronizadamente en un entorno de simulación.
- Aplicar los conceptos de programación robótica aprendidos durante las prácticas en empresa.
- Configuración y programación de un PLC simulable y permita la coordinación de la estación de trabajo.
- Diseño y programación de una interfaz de usuario que permita el manejo y control de la instalación de forma dinámica.
- Estudio de un estándar de comunicaciones que posibilite la comunicación entre los diferentes softwares.

Capítulo 1.3. Fundamentos previos.

Para la ejecución de este proyecto, se requieren conocimientos previos que abarcan diferentes ramas de la ingeniería como son la robótica industrial, la automatización y control de procesos, la visión artificial aplicada a la industria o las comunicaciones industriales. Fundamentos que se han ido adquiriendo tanto en diferentes asignaturas del grado como en las prácticas en empresa.



Estos fundamentos han de combinarse con el manejo de diferente software como RoboGuide, TIA Portal y KEPServerEX que requieren de una formación básica como la recibida en el grado para TIA Portal y un estudio autónomo como en el caso de RoboGuide y KEPServerEX.

Capítulo 1.4. Alcance del proyecto.

Como se ha indicado en anteriormente, en el siguiente proyecto se implementa una celda robotizada de simulación empleando software de diferentes marcas para su simulación y control – softwares que no proporcionan una conectividad directa.

Para coordinar ambos aspectos de la planta robotizada se emplea un software adicional que gestiona el intercambio de información entre el software de simulación y el de control, este posibilita experimentar cómo sería el comportamiento global de estos dispositivos en un entorno de simulación.

Capítulo 1.5. Estructura del trabajo.

El texto relativo al informe del proyecto está estructurado de la siguiente manera:

- Introducción teórica a las diferentes ramas de la ingeniería que conforman la temática del proyecto. De esta forma, se trata de establecer un contexto general acerca de la historia y desarrollo de las diferentes tecnologías utilizadas en la resolución del problema planteado.
- Seguidamente se analiza el problema planteado. Para buscar soluciones posibles al problema se recurre a proyectos pasados realizados en la Universidad de Valladolid. De esta manera se pueden extraer metodologías y conclusiones como orientación para la búsqueda de soluciones factibles.
- Tras el punto anterior, se profundiza en la explicación de las tecnologías mencionadas en el primer punto mediante la exposición de los diferentes softwares que se han utilizado y cómo trabajar con ellos.
- A continuación, se realiza un estudio relativo a la normativa vigente para el diseño y seguridad de instalaciones robotizadas. La normativa de seguridades tiene en cuenta los objetos que se manipulan dentro de la instalación y su inclusión dentro de la misma, por lo que estos se detallan en este punto.
- Se da paso a la explicación el desarrollo del proyecto software por software. Se exponen todas las configuraciones realizadas en cada programa paso a paso, se muestran y se analizan ejemplos de



programas tanto de robots como de controladores, y se muestra cómo poner en funcionamiento las simulaciones para realizar pruebas resultados.

- Se exponen los resultados obtenidos, se extraen una serie de conclusiones y se proponen varias mejoras o variaciones del proyecto que pueden ser interesantes desde un punto de vista didáctico.
- Finalmente, al final del informe, se localizan las referencias bibliográficas utilizadas, así como los anexos donde se incluyen todos los códigos empleados para la programación de los robots y del PLC.

Capítulo 1.6. Convenciones.

Se expone una serie de normativas de marcado que se han tenido en cuenta a la hora de la redacción del informe del proyecto, y que se pueden encontrar en el desarrollo de los siguientes capítulos.

- Aquellas palabras extranjeras que no han sido traducidas son entrecorilladas con comillas simples y remarcadas en cursiva.
- Los títulos de obras de autores se resaltan en cursiva.
- Los nombres de autores se resaltan en negrita.
- Las citas que hagan referencia a obras externas se encuentran referenciadas con el estilo IEEE.
- Tras el índice general del proyecto, se incluyen de índices relativos a la localización de figuras y tablas incluidas en este documento.

CAPÍTULO 2. MARCO TEÓRICO

CAPÍTULO 2.1. Robótica

El término 'robot' tiene sus orígenes en la palabra '*robota*', procedente del checo, cuyo significado es trabajo forzado o servidumbre. Sin embargo, el origen etimológico de la palabra se remonta a 1920, cuando el dramaturgo checo **Karel Čapek** presenta su obra *Rossum's Universal Robots (R.U.R)*. En ella aparece un robot como una herramienta diseñada aligerar la carga de trabajo que recae sobre los humanos. [1]

CAPÍTULO 2.1.1. Historia de la Robótica

A pesar de que el concepto robot parezca futurista y avanzado tecnológicamente, el origen de la robótica se remonta a los mismos orígenes de la Humanidad. Ya las antiguas civilizaciones vislumbraban la idea de crear entidades animadas que interactuaran con las personas y que poseyeran un grado de autonomía.

Parece que los **egipcios** trataban de colocar brazos mecánicos a las estatuas de sus deidades dotando a las mismas de movimiento que simulaban los propios de los seres humanos.

En la antigua **Grecia** destaca la figura de **Herón de Alejandría** (10d.C. – 70 d. C.). Este ingeniero y matemático helenístico desarrolló una variedad de autómatas basados principalmente en sistemas hidráulicos que se exhibían en los templos religiosos. Una invención destacable es la descrita en su libro *Autómata: Los Pájaros de Herón* [2]. Se trata de una fuente adornada con pájaros capaces de volar, gorjear y beber de la propia fuente.

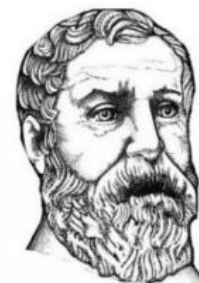
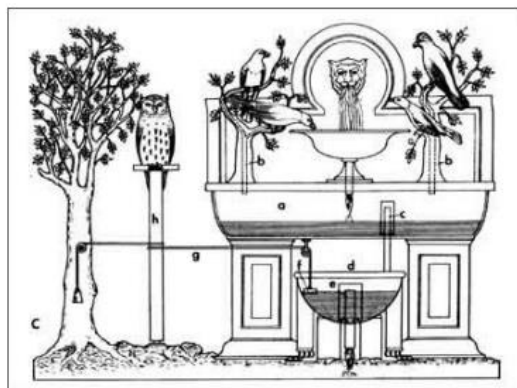


FIGURA 1. Los Pájaros de Herón. Herón de Alejandría.

Entre muchas otras cosas, **Leonardo da Vinci** (1452-1519) diseñó un autómatas humanoide en forma de caballero con capacidad de movimiento de las diferentes articulaciones del cuerpo. Dicho autómatas se basaba principalmente en un complejo de poleas como se aprecia en la figura mostrada a continuación [3].

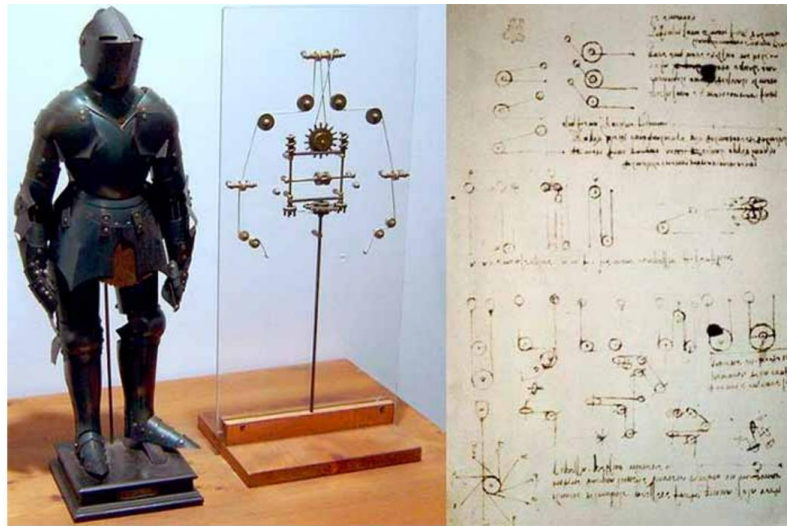


FIGURA 2. Caballero de Leonardo da Vinci. Manuscritos del ingeniero italiano.

En la revolución industrial se dio un giro a la utilidad de estas máquinas. Hasta entonces, como ya se ha dicho, los autómatas tenían una utilidad meramente lúdica o decorativa. Las máquinas tocaban música, se movían e incluso llegaban a escribir. Sin embargo, a partir de la revolución industrial, se crearon máquinas robotizadas como el *Telar de Jacquard* [4] de **Joseph Marie Jacquard** (1752-1834) cuyo fin era mejorar una tarea productiva – elevándose así la producción y el nivel de vida de la población. Este autómatas utilizaba tarjetas perforadas para tejer patrones en la tela, permitiendo cualquiera pudiera elaborar diseños complejos.



FIGURA 3. Telar de Jacquard y Tarjeta Perforada.

CAPÍTULO 2.1.2. Robótica Industrial

No es hasta el año 1954 que aparece el primer brazo robot programable con capacidad para ser utilizado para diversas tareas, es decir, el considerado primer robot industrial. Este robot fue desarrollado por el estadounidense **George Devol** (1912-2011). Se le bautizó como *UNIMATE* y fue instalado en la cadena de montaje de la factoría *General Motors* en el año 1960.

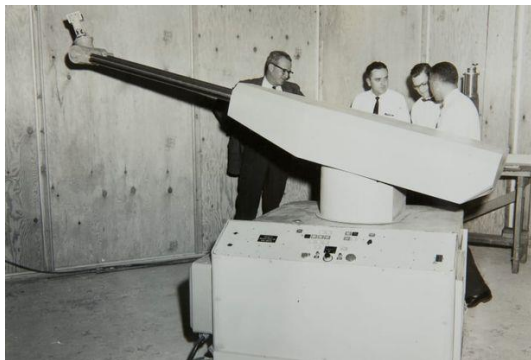


FIGURA 4. Robot UNIMATE

En la década de los años 70 se desarrollan avances que se aproximan a las bases de la robótica industrial actual. Por ejemplo, a principios de esta década, en la universidad californiana de 'Stanford University' arma un brazo robótico totalmente eléctrico (*Stanford Arm*). Por otro lado, a mediados, la empresa ABB arma el primer robot industrial eléctrico controlado por microprocesador (*IRB6*). Finalmente, la empresa 'Unimation Inc.' Lanza el robot industrial *PUMA* ('*Programmable Universal Machine for Assembly*') en el año 1978. El robot *PUMA* es el concepto de brazo multiarticulado en el que se basa la mayoría de los robots industriales actuales.



FIGURA 5. Gama de robots PUMA



CAPÍTULO 2.1.3. Definición y clasificación de los robots

No hay una definición unificada de lo que es un robot. Pero a continuación se muestran algunas representativas:

El '*Robot Institute of America*' (RIA) lo define como "manipulador multifuncional reprogramable, capaz de mover materiales, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas". [1]

Mientras que la International '*Federation of Robotics*' (IFR) lo define como: "máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que puedan posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento". [1]

Existen múltiples clasificaciones de los robots, pero a continuación se señalan tres posibles:

SEGÚN SU CRONOLOGÍA	
1º Generación	Realizan la tarea cíclicamente y de forma secuencial. No hay comunicación con el entorno y sus alteraciones.
2º Generación	Toman información de su entorno y actúan en consecuencia.
3º Generación	Se programan mediante lenguaje natural. Están dotados de cierta "inteligencia" ya que pueden planificar tareas de forma automática.

TABLA 1. Clasificación de robots según su cronología

SEGÚN SU FUNCIÓN	
Industriales	Destinados a entornos de fabricación industrial
De Servicios	Destinados a entornos fuera de instalaciones industriales. Por ejemplo, robots profesionales : médicos, militares, espaciales. Y robots de uso personal : de entretenimiento, educacionales.

TABLA 2. Clasificación de robots según su función



SEGÚN SU AUTONOMÍA	
Teleoperados	Necesitan ser controlados (a distancia) en todo momento. Ejemplo: Drones de vuelo.
Semiautomáticos	Necesitan ser controlados, pero tienen cierta autonomía. Ejemplo: Barrera de aparcamiento de coches.
Automáticos	No necesitan de un control humano. Ejemplo: Vehículos Guiados Automáticamente (AGVs).

TABLA 3. Clasificación de robots según su autonomía.

CAPÍTULO 2.1.4. Robots manipuladores industriales. Características

Los robots manipuladores industriales son aquellos dotados de sistemas mecánicos, eléctricos y electrónicos que se encuentran en entornos industriales. Son multifuncionales, es decir, realizan diversas tareas de forma automática. Las más típicas suelen ser desplazar objetos, pintarlos o aplicarles una soldadura. Dichas tareas son indicadas por los humanos – son programadas.

Estos mecanismos poseen las siguientes características que determinan la funcionalidad de estos:

- **Grados de libertad:** Se asocia al número de articulaciones del robot. A mayor número de grados de libertad, el robot será más flexible.
- **Área de trabajo:** Región del espacio alcanzable por el robot en alguna de sus configuraciones. Limitada por las características físicas de este.
- **Resolución:** Desplazamiento mínimo que puede realizar el robot.
- **Capacidad de carga:** Máximo peso en carga que un robot es capaz de manipular a una velocidad determinada considerando la configuración menos favorable.
- **Repetibilidad:** Capacidad de alcanzar el mismo punto destino desde el mismo punto inicial de forma sucesiva sin variaciones.
- **Capacidad de posicionamiento:** Exactitud entre el punto alcanzado en un movimiento y el punto real deseado.
- **Velocidad:** cómo de rápido puede moverse un robot.
- **TCP: 'Tool Center Point'** o 'Punto Central de la Herramienta'. Punto utilizado para el posicionamiento de robots en cualquier programa que implique objetivos definidos en el espacio cartesiano.
- **Puntos singulares:** Puntos del espacio de trabajo del robot sobre los que no es posible realizar una trayectoria rectilínea. Ya porque implica una velocidad infinita en el movimiento o porque el valor de los ejes en

dichos puntos es indeterminado. Se suele dar cuando dos o más ejes se alinean.

CAPÍTULO 2.1.5. Robots manipuladores industriales. Ejemplos

Los robots manipuladores más comunes en la industria son los siguientes:

Robots antropomórficos: los más típicos en entornos industriales debido a su gran versatilidad. Usualmente constan de 6 grados de libertad, lo cual permite posicionar la herramienta en una posición determinada con tres orientaciones diferentes.



FIGURA 6. Robots antropomórficos

Robots SCARA: 'Selective Compliant Assembly Robot Arm o Selective Compliant Articulated Robot Arm'. Es un robot de 4 grados de libertad que trabaja en veloces ciclos de trabajo, es decir, posee una gran velocidad de desplazamiento. Son robots de posicionamiento horizontal, muy precisos, generalmente utilizados en tareas de ensamblaje o de 'Pick&Place'.



FIGURA 7. Robot SCARA

Robots Cartesianos: constan de 3 ejes perpendiculares entre sí (X, Y, Z) que realizan únicamente movimientos lineales. Se suelen utilizar en tareas de fresado, taladrado...

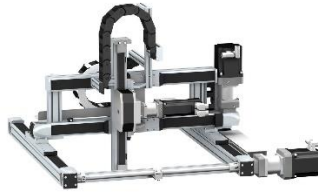


FIGURA 8. Robot Cartesiano

Robot Delta: es un tipo de robot paralelo de tres grados de libertad conformado por dos bases unidas por tres cadenas cinemáticas basadas en el uso de paralelogramos. No existen ecuaciones que modelen su configuración, por lo que se debe diseñar un modelo de la dinámica para cada robot. Se suelen utilizar en tareas de empaquetado.



FIGURA 9. Robot delta

Vehículos de Guiado Automático: (AGVs). Son vehículos que se mueven de forma autónoma principalmente destinados al transporte de materiales dentro de un entorno sin necesidad de un operario que los maneje.



FIGURA 10. AGV

CAPÍTULO 2.2. Automatización

Por automatización se entiende la realización de tareas con escasa participación humana. Este concepto se puede aplicar prácticamente a

cualquier sector de la sociedad, pero el que a nosotros nos interesa es uno más específico: Automatización Industrial.

CAPÍTULO 2.2.1. Automatización industrial

La automatización industrial se podría definir como el uso de sistemas de control y maquinaria para operar sobre procesos industriales, reduciendo así la intervención humana y aumentando la productividad y eficiencia de estos.

A continuación, se muestra una tabla con las ventajas y desventajas de la automatización industrial:

VENTAJAS	DESVENTAJAS
Aumento de la productividad	Destrucción de empleo
Ahorro de costes	Dependencia de personal más cualificado
Ventaja competitiva	Dependencia/Obsolescencia tecnológica
Mejor salud de los trabajadores	Alto coste de inversión inicial
Mayor calidad de vida	
Producción más flexible	

TABLA 4. Ventajas y Desventajas de la Automatización Industrial

En la actualidad, la automatización de los procesos productivos se ha vuelto una necesidad para poder competir en el mercado.

Dentro de la automatización de un proceso productivo, podemos distinguir diferentes niveles de automatización:

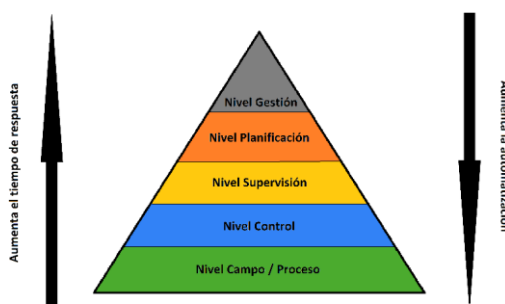


FIGURA 11. Pirámide de la automatización industrial



1. **Nivel de Campo:** el tiempo de respuesta es del orden de los nanosegundos/milisegundos. Los dispositivos principales para la automatización en este nivel son los sensores y actuadores. [5]
2. **Nivel de Control:** el tiempo de respuesta es del orden de los milisegundos/segundos. El dispositivo principal para la automatización en este nivel son los controladores lógicos programables (PLC). [5]
3. **Nivel de Supervisión:** el tiempo de respuesta es del orden de los segundos/minuto. El dispositivo principal para la automatización en este nivel son las interfaces, SCADAs... [5]
4. **Nivel de Planificación:** el tiempo de respuesta es del orden de los minutos/hora. Los dispositivos principales para la automatización en este nivel son los sistemas de ejecución de fabricación (MES). [5]
5. **Nivel de Gestión:** el tiempo de respuesta es del orden de los días/años [5]. Los dispositivos principales para la automatización en este nivel son los sistemas de planificación de recursos empresariales (ERP). [5]

CAPÍTULO 2.2.2. Controlador Lógico Programable (PLC)

Como hemos visto en el punto anterior, a nivel de control de un proceso productivo, el PLC o autómatas programables es el dispositivo más extendido en la industria.

La NEMA (“*National Electrical Manufacturer Association*”) lo define PLC como: “dispositivo electrónico que utiliza memoria programable para guardar instrucciones sobre la implementación de determinadas funciones, como operaciones lógicas, secuencias de acciones, especificaciones temporales, contadores y cálculos para el control mediante módulos de entrada/salida analógicos y/o digitales sobre diferentes tipos de máquinas y de procesos”. [6]

De forma más simple, un autómatas programables es un equipo electrónico diseñado para aplicaciones industriales. Es robusto, fiable, y capaz de gestionar múltiples entradas y salidas en el duro ambiente industrial (ruido electromagnético, temperaturas extremas, suciedad, vibraciones...). Generalmente, se construye físicamente por una serie de módulos que se interconectan entre sí – CPU, Entradas/Salidas, Entradas/Salidas Seguras...



FIGURA 12. Autómatas programables

La arquitectura interna de un PLC industrial típico, se puede detallar esquemáticamente de la siguiente manera:

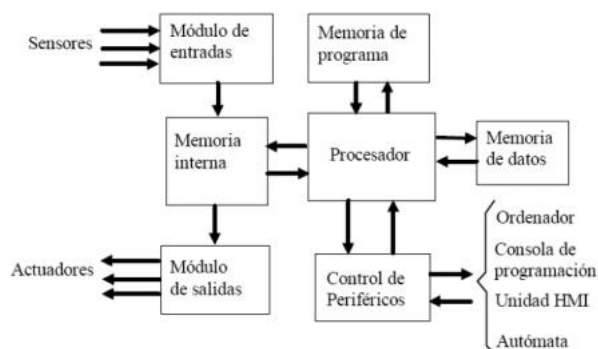


FIGURA 13. Arquitectura interna de un PLC

- **Procesador:** es el encargado de ejecutar e interpretar las instrucciones que hay en el programa almacenado en la memoria de programa, así como monitorizar el estado de las entradas del sistema por medio de los sensores.
- **Módulos de entrada/salida:** conectan el autómata y el proceso industrial a controlar, es decir, permiten la conexión directa entre el PLC con los sensores y actuadores del proceso respectivamente.
- **Bloques de memoria:** los PLC cuentan con diferentes tipos de memoria entre los que se encuentran memorias de tipo RAM, ROM o EEPROM. A continuación, se exponen los diferentes bloques de memoria que pueden encontrarse en los autómatas programables industriales atendiendo a una clasificación funcional:
 - Memoria de programa. Contiene el programa con las instrucciones que ejecuta el procesador. Se divide en dos partes, una parte ROM (fija) que contiene el programa monitor encargado de comunicar el autómata con los módulos de programación y una parte RAM con



batería (mantiene los datos, aunque se apague la alimentación) donde se almacena el programa de usuario.

- Memoria interna. Almacena los valores de las variables internas del autómatas, así como los valores de las entradas y salidas. Aunque son organizadas en bytes, cada variable es de 1 solo bit, por lo que se puede acceder a ellas de forma independiente.
- Memoria de datos. Almacena los datos de configuración y/o parámetros necesarios para el funcionamiento del autómatas y del proceso, así como también contiene datos de propósito general.

Además, cuentan con una **fuentes de alimentación** incorporada. Este módulo se encarga de suministrar la tensión que requieren los distintos módulos que forman el equipo.

Un autómatas programable tiene dos modos de funcionamiento principalmente:

- Modo **STOP**: se utiliza para la programación del PLC. El programa monitor se encarga de la comunicación del PLC con el elemento de programación con el fin de traspasar el programa seleccionado del cual se desea ejecutar el algoritmo de control.
- Modo **RUN**: este es el modo de ejecución del programa que implementa el algoritmo de control. En este modo, el PLC controla el proceso productivo. La ejecución es cíclica e indefinida. A cada ciclo de ejecución se le conoce como 'ciclo de scan'. Un ciclo de scan está formado por un conjunto de tareas que ejecuta un autómatas para controlar un proceso; donde se evalúan las entradas, se ejecuta el programa de usuario y se generan las salidas.

CAPÍTULO 2.2.3. Interfaz Hombre-Máquina (HMI)

Las HMIs son dispositivos electrónicos utilizados para interactuar con máquinas, controlar un proceso o introducir consignas (operaciones como marcha, paro, cambio de modo de funcionamiento...) a nivel de usuario.

Estos se deben programar previamente, especificando las funcionalidades de las que dispondrán los usuarios. Seguidamente se comunica con el autómatas mediante puerto serie.



FIGURA 14. Interfaces Hombre-Máquina

CAPÍTULO 2.3. Comunicaciones Industriales

La implementación del control automatizado de los procesos productivos hace que surja la necesidad de comunicar los dispositivos de la planta (sensores, actuadores) con los de control y supervisión para garantizar un óptimo funcionamiento de estos.

Los sistemas de automatización industrial se basan, pues, en robustas redes de comunicación industrial encargadas del transporte fiable de datos entre los diferentes dispositivos de automatización. Estos últimos comparten un protocolo común dentro de la red; es decir, un idioma que todos son capaces de “entender”. Por otro lado, las redes de comunicación permiten la interconexión de muchos equipos – lo cual garantiza control más versátil y completo de la planta.

Hoy en día se dispone de un gran número de protocolos de redes de comunicación industrial. Algunos son desarrollados por fabricantes específicos, mientras que otros son definidos como estándares abiertos. Esto, sin embargo, conlleva la problemática de la compatibilidad entre equipos de diferentes fabricantes a nivel de protocolo; ya que, actualmente, no existe un estándar común a todos los fabricantes que universalice la comunicación entre dispositivos.

CAPÍTULO 2.3.1. Niveles de comunicación de una red industrial

Como ya se ha mostrado anteriormente en la FIGURA 11; una factoría de procesos se puede jerarquizar, idealmente, en diferentes niveles según el grado de automatización requerida por las diferentes áreas de la factoría. Por ende, cada nivel suele utilizar diferentes redes en función de sus necesidades como volumen de datos, velocidad de respuesta, seguridad de los datos...



1. **Nivel de Campo:** la tecnología más utilizada es la red de comunicación de bus de campo (Profibus, Interbus, Fieldbus Foundation... [7]) aunque para transmisión de datos asociados a señales binarias también se usa AS-i ('*ActuatorSensor-interface*' [7]). El bus de campo permite la gestión de sensores y actuadores además de facilitar el control distribuido entre controladores y dispositivos de campo inteligentes.
2. **Niveles de Control y Supervisión:** se utilizan redes de área local (LAN) Ethernet con protocolos TCP/IP para interconectar las unidades de control a computadores. Aunque también se utilizan buses de control.
3. **Niveles de Planificación y Gestión:** estos dos niveles los podemos agrupar ya que no requieren de redes de comunicaciones industriales de bajos tiempos de respuesta y robustas. Importa el volumen de información transmitida en lugar que permita la gestión del proceso productivo. Se utilizan redes a gran escala, como puede ser WAN Ethernet.

CAPÍTULO 2.3.2. Protocolos de comunicación industrial

Los protocolos de comunicación industrial son conjuntos de condiciones o reglas que habilitan el intercambio de información entre dispositivos

Básicamente, un protocolo es el idioma que "entienden" los dispositivos de una red industrial para la correcta comunicación entre ellos.

Entre los estándares más comunes en la industria se encuentran:

- **Comunicación Serial:** se implementa con protocolos como RS232, RS422 y RS485. Este sistema facilita la comunicación tanto analógica como digital bidireccional al mismo tiempo por un mismo cableado.
- **Industrial Ethernet:** trata de incluir Ethernet a un uso en el ámbito industrial. Algunos ejemplos de Industrial Ethernet son: Ethernet/IP [7], PROFINET [7] o, EtherCAT [7].
- **DeviceNet:** red basada en Bus CAN ('*Controller Area Network*') que permite interconectar dispositivos a nivel de campo y controladores de nivel superior adoptando un protocolo CAN básico. Hay otras redes armadas sobre CAN como **CANopen**.
- **PROFIBUS:** red de campo abierto. Adecuado para tareas de comunicación complejas y aplicaciones donde el tiempo es un factor crítico. Existen tres versiones: Profibus-DP, Profibus-PA, Profibus-FMS.
- **Modbus:** protocolo de sistema abierto. Existen distintas versiones para maestro/esclavo (RTU), puerto serie (RS232 o RS485) o cliente/servidor (TCP/IP).

CAPÍTULO 2.3.3. OPC

“OPC (‘OLE (Object Linking and Embedding) for Process Control’) es un estándar de comunicación que permite la transmisión fiable de datos entre dispositivos de distintos fabricantes” [8]. En consecuencia, comunica equipos con diferentes protocolos – esto no quiere decir que sustituya a los protocolos propios de cada dispositivo. Se basa en la arquitectura Cliente-Servidor, donde el servidor almacena los datos que el cliente toma y manipula.

Para lograr la comunicación, OPC abstrae los detalles del servidor de datos y del cliente, permitiendo así que se realice el intercambio de los datos entre ellos sin necesidad de conocer sus protocolos de comunicación nativos ni su organización interna de los datos.

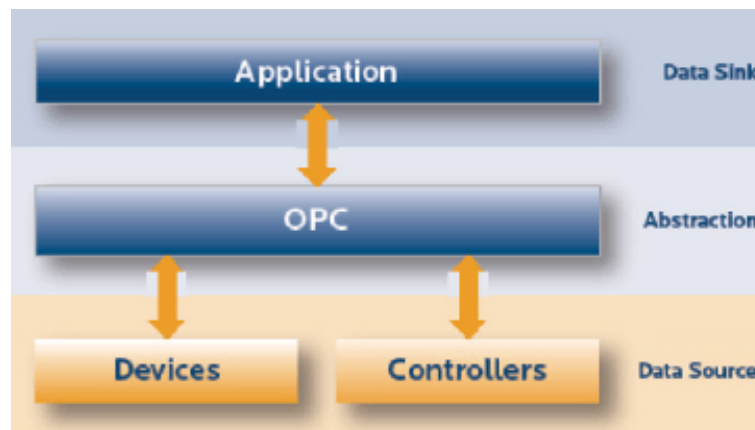


FIGURA 15. OPC como capa de abstracción entre Cliente y Servidor

CAPÍTULO 2.4. Visión Artificial

La visión artificial o visión por computadora es una tecnología que permite a dispositivos electrónicos capturar imágenes del mundo real y procesarlas en información numérica.

En otras palabras, la visión artificial permite a las máquinas “ver” qué hacen para así tomar decisiones sobre esas acciones.

La integración de visión artificial conlleva una serie de ventajas:

- Mejora la calidad de los productos, acelera y optimiza su fabricación.
- Permite una mayor precisión en los diagnósticos de errores.
- Elimina la subjetividad en los sistemas de inspección o control de calidad ya que se programa en base a unos criterios muy específicos.
- Disminuye la tasa de desperdicios.



- Se incluye en una multitud de aplicaciones: sistemas de 'Pick&Place', sistemas de medición, localización de defectos, lectura de códigos...

La visión artificial se puede clasificar atendiendo a diferentes criterios [9]:

SEGÚN SU TECNOLOGÍA	
Sensores de Visión	Son los más básicos y fáciles de integrar en un sistema automatizado. Aplicación: detección del paso de un elemento.
Cámaras inteligentes y sistemas de visión integrados	Son cámaras que cuentan con sistemas de cálculo más complejos. Tienen capacidad para procesar información y posibilidad de trabajar en remoto.
Sistemas de visión avanzados	Su <i>software</i> y su <i>hardware</i> están en un nivel superior de desarrollo y eso influye positivamente en la cantidad de datos que pueden acumular y procesar.

TABLA 5. Visión artificial según su tecnología.

SEGÚN LA CALIDAD DE IMAGEN	
Visión 1D	Opera con una señal digital que estudia la imagen línea a línea y no en su totalidad. Aplicación: detección de defectos de fabricación.
Visión 2D	Se captan imágenes en color o en escala de grises y se convierten en matrices divididas en píxeles. Aplicación: reconocimiento de caracteres como en la lectura de códigos.
Visión 3D	Se colocan cámaras en distintos lugares o sensores de láser por lo que reúnen información sobre la orientación del elemento. Aplicación: 'pick&place'.
Termografía	Sensores termográficos capta información sobre la temperatura de las piezas y se representa en una escala de colores. Aplicación: detección de fugas.

TABLA 6. Visión artificial la calidad de imagen.

La robótica, especialmente, se ha beneficiado de esta tecnología. La visión artificial dota a los robots de mayor autonomía al permitir reconocer con cierto nivel de precisión la localización y forma de los objetos de su entorno, lo cual nos conduce al siguiente punto.



Capítulo 2.4.1. Detección y reconocimiento de objetos.

Esta rama de la visión artificial estudia la detección de objetos por medio de una imagen en base a su apariencia. Entre los criterios de detección más comunes destacan la forma del objeto (reconocimiento de bordes) o su naturaleza. El proceso de reconocimiento de imágenes se suele dividir en tres partes:

1. **Proceso de Aprendizaje:** Extracción de características del contenido de una imagen: mediante una imagen del objeto a reconocer, se “enseñan” las peculiaridades del objeto al algoritmo de detección.
2. **Proceso de detección:** Una vez conocidas las características del objeto, se trata de encontrar elementos concordantes con dichas características mediante imagen.
3. **Proceso de clasificación:** finalmente los objetos detectados en el paso 2 se delimitan en cuadros donde se produce la extracción de las características de los objetos aplicando modelos matemáticos que “resuman” el contenido de la instantánea tomada. Los modelos ofrecen un resultado: la clasificación.



CAPÍTULO 3. ESTADO DEL ARTE

CAPÍTULO 3.1. Antecedentes

Anteriormente, se han llevado a cabo diferentes desarrollos de celdas robotizadas en la Escuela de Ingenierías Industriales de la Universidad de Valladolid. Se citan, a continuación, una serie de trabajos:

En el 2015, Juan Antonio Ávila Herrero diseñó una célula robotizada con fines educativos para su proyecto de fin de grado. Esta está desarrollada en el software de ABB, RobotStudio. Consta de un único robot (IRB-120) y de componentes inteligentes programados con Robot-Studio que simulen los diferentes elementos de la célula [10].

En el año 2016, Álvaro Galindo de Santos aúna la simulación de una isla robotizada educativa con el desarrollo de una aplicación Android mediante la comunicación remota a través del mecanismo socket. Desde una Tablet con dicho sistema operativo, se envían señales digitales y comandos de movimiento al robot, además de poder mantener una partida de ‘Tres en Raya’ con el mismo [11].

En 2020, José Delgado García diseñó una célula robotizada formativa basada en robots de Fanuc y autómatas de Siemens. El alumno realizó tanto el montaje físico de la misma como la programación del autómatas y el robot utilizados [12].

En el año 2022, Alberto Martínez Rodríguez comunicó mediante tecnología OPC una célula robotizada implementada en RoboGuide (Fanuc) con un autómatas programable de Siemens. Además, dicha célula era gestionada a nivel de usuario mediante una HMI desarrollada en Matlab [13].

Por último, también en ese año, Raúl Vidal del Cura realiza su trabajo de fin de grado en el que plantea una célula robotizada basada en una línea de montaje. Se compone por cuatro robots de ABB que, sincronizados, permiten simular la creación de una luminaria. De nuevo, se gestiona a nivel de usuario mediante una HMI desarrollada en Matlab [14].

CAPÍTULO 3.2. Análisis del problema

Como se ha visto anteriormente, existe una gran cantidad de trabajos e información en la escuela acerca de la programación de robots de ABB. Sin embargo, no hay gran variedad de Fanuc. Los robots de Fanuc se están volviendo fuertemente populares en industrias cercanas a la Universidad de Valladolid. Por ello, es pertinente la aportación de información acerca de la programación de estos productos y del desarrollo de entornos de simulación



que permitan el aprendizaje de los futuros robóticos sin manipulación de los equipos físicos

Este proyecto supone una aportación teórico-práctica a los programas utilizados. Programas de Fanuc, Siemens y KEPWare.

CAPÍTULO 3.3. RoboGuide

RoboGuide es un simulador 3D de robots Fanuc que permite la creación de células robotizadas, simulación offline de movimientos de cualquier robot de la marca y de programas de usuario, modificación de parámetros, trayectorias... Las configuraciones y programas finales se exportan y se cargan en los robots físicos, reduciendo el tiempo invertido en la realización de cambios de diseño en las instalaciones físicas. En consecuencia, la producción de la célula real se ve mínimamente afectada.

RoboGuide incluye diferentes herramientas según sea la tarea que debe atender el o los robots de la celda que se desea modelar. Entre ellas encontramos:

1. **HandlingPro.** Se utiliza para simulación offline de procesos de manipulación y de '*pick&place*' con el fin de estudiar el comportamiento y viabilidad del robot elegido. Además, las simulaciones ofrecen aproximaciones muy exactas acerca del tiempo de ciclo que supondría al robot real la realización del programa cargado [15].
2. **PalletPro.** Permite probar aplicaciones de paletizado y despaletizado en un entorno de simulación offline. Los datos creados se pueden cargar en el controlador del robot físico que cuente con la utilidad de paletizado (*PalletTool*) [15].
3. **WeldPro.** Se utiliza en aplicaciones donde el robot debe realizar soldaduras al arco. Permite caracterizar la trayectoria de la soldadura con diferentes patrones de movimiento, por ejemplo: línea recta, diente de sierra, círculos... [15].
4. **PaintPro.** Sirve para procesos de aplicación de pintura. Contiene funciones especiales para el control de la herramienta, patrones de pintura, delimitación de zonas a pintar, solapado, velocidad y tiempos de activación de la pistola de pintura [15].
5. **ChamferingPro.** Permite generar y simular programas de desbarbado de piezas automáticamente. Para generar las trayectorias de desbarbado, se tiene que trazar los contornos en el CAD 3D de las piezas [15].
6. **OLPCPro.** Software de desarrollo de programas de robots para el desarrollo y mantenimiento de programación KAREL (equivalente a RAPID de ABB) y Teach Pendant [15].

7. **iRPickPro**. Permite simular aplicaciones '*pick&place*' de alta velocidad [15].

Además, para mejorar las funcionalidades de los robots; RoboGuide incluye una serie de herramientas complementarias que facilitan la inclusión de visión artificial, seguridad y control movimiento.

1. **Fanuc iRVision**. Sistema de reconocimiento de imágenes 2D o 3D para manipulación en diversas situaciones: localización de piezas por tamaño, forma o posición; lectura de códigos de barras, ordenación por colores... Puede programarse sensores/cámaras de visión artificial tanto montados en el robot como fijos.



FIGURA 16. Visión Artificial: montada sobre robot y fija.

2. **Fanuc Dual Check Safety (DCS)**. Brinda una solución software inteligente de seguridad de operarios, robots y herramientas. Delimita zonas donde el robot puede o no acceder en función de las señales recibidas.

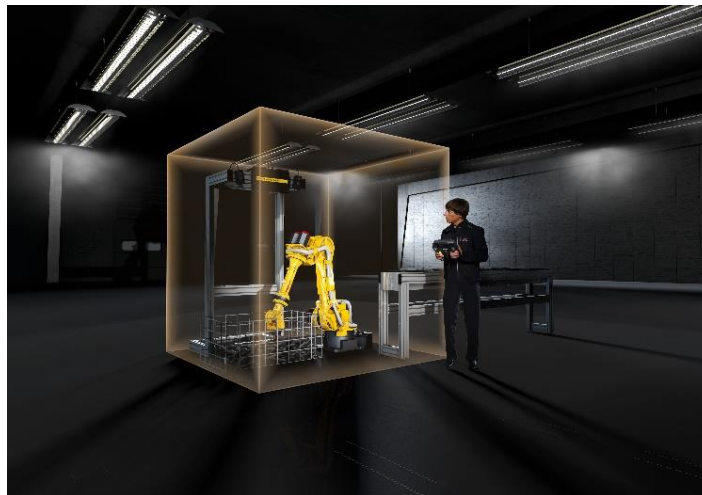


FIGURA 17. Dual Check Safety.

3. **Fanuc Multi Group Motion.** Permite el control de un eje auxiliar totalmente integrado y estandarizado como un rail, una servopinza o el control de dos o más robots trabajando de forma conjunta o un robot y un posicionador...

El proyecto se ha desarrollado en HandlingPro, por lo que solo se hará una explicación del entorno de desarrollo de dicha herramienta.

Capítulo 3.3.1. HandlingPro

Tanto HandlingPro como el resto de las herramientas de RoboGuide V9 poseen la siguiente interfaz de usuario:

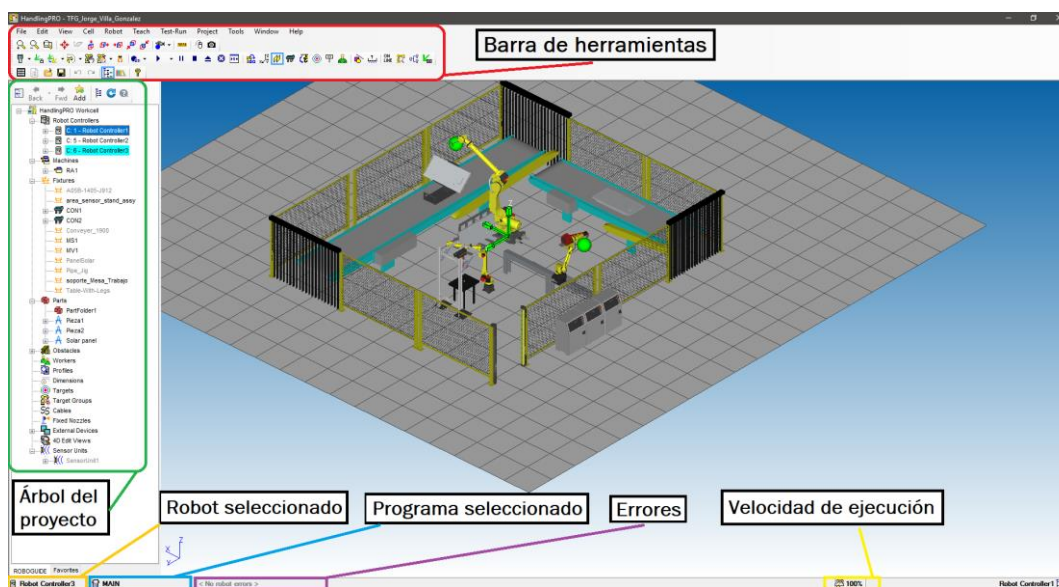


FIGURA 18. Interfaz HandlingPro.

HandlingPro ofrece la totalidad de sus robots catalogados y la totalidad de los controladores (desde versiones antiguas hasta las más recientes) para la realización de los proyectos de simulación.

Como se aprecia en la figura anterior, la interfaz está dividida en diferentes secciones según la utilidad de cada sección.

En primer lugar, la sección señalada en rojo pertenece a la barra de herramientas del software. Ofrece posibilidades que se muestran en la siguiente figura:

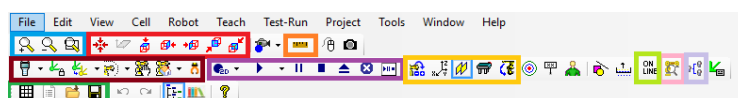


FIGURA 19. Barra de herramientas de HandlingPro.



- Remarcado en **azul** se tienen las funcionalidades tales como ‘zoom’ de la pantalla y movimiento por la misma.
- En **rojo** se tienen funcionalidades de establecimiento de perspectivas de la propia estación. E incluso, la posibilidad de centrar la perspectiva en un objetivo determinado.
- Señalado en **naranja** se tiene la regla que mide distancias, lo cual es útil a la hora de crear determinados objetos a manipular.
- En marrón se agrupan funcionalidades muy dispares. De izquierda a derecha: la primera muestra la ‘Teach Pendant’. Las dos siguientes tienen que ver con la herramienta de los robots, una mantiene su estado y la otra lo devuelve al estado anterior. Las tres siguientes tienen que ver con los mismos robots. La primera muestra el espacio de trabajo del robot seleccionado como una nube que lo envuelve. La segunda permite mover las articulaciones del robot seleccionado una a una. La tercera manda a todos los robots a una posición HOME guardada. Por último, el icono de la pinza permite abrir/cerrar la pinza del robot seleccionado.
- Recuadrado en **morado** se tienen las funcionalidades pertenecientes a la ejecución: grabación de la misma, inicio, pausa, abortar, resetear errores, parada inmediata y panel de ejecución donde se encuentran más funcionalidades de la ejecución.
- En **amarillo** también se agrupan varias funciones diferentes: movimiento del TCP según diferentes sistemas de coordenadas, mostrar barra de enseñanza rápido de puntos, mostrar límites de ‘tracking’ del robot seleccionado como dos paredes (útil para programación de ‘Line Tracking’), mostrar panel de movimiento de los transportadores y panel de movimiento del robot seleccionado a lugares determinados como un objeto, un vértice del mismo...
- En **verde** se puede conectar o desconectar el o los dispositivos externos ya programados. Esto se utiliza por ejemplo si se tiene conectado un servidor OPC, un robot externo o una CNC externa a la estación.
- En **rosa** se tiene la opción de abrir MotionPRO: una herramienta complementaria para optimización de tiempos de ciclo, reducción de consumos y mejoras de los movimientos.
- En **violeta** se tiene la funcionalidad de mostrar los programas llamados.
- Por último, en **verde oscuro** están las funciones de ir a la pantalla de inicio, crear una nueva estación, apertura de una estación ya guardada y guardado de la estación actual.

La sección remarcada en verde pertenece al árbol del proyecto. Como se puede apreciar, hay gran variedad de puntos de acceso a diferentes aspectos de la estación.

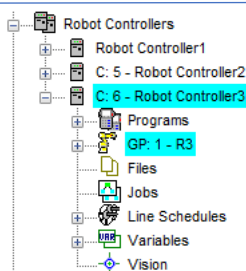


FIGURA 20. Árbol del proyecto: Controladores de los robots

En este apartado se encuentra todo lo perteneciente al control de los robots. En el apartado 'Programs' se encuentran los programas de usuario y los programas internos al robot. En GP se pueden acceder a las herramientas de las que disponga el robot, espacios de trabajo creados que le sean útiles ('uframes'), accesorios que podemos incluir al robot como cables, tubos... En 'Line Schedules' se encuentran todos los seguimientos de línea o 'Line Tracking' con los que se haya programado el robot. En 'variables' se hallan todas las variables de las que dispone el robot. Y si el robot está dotado con funcionalidades complementarias como la Visión Artificial, se mostrarán a continuación del apartado 'Variables'.



FIGURA 21. Árbol del proyecto: Máquinas

En el apartado 'Machines' se dispone de todas las máquinas que se hayan incluido en la estación: raíles, transportadores, CNCs, prensas...

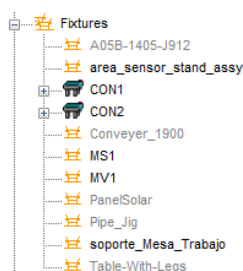


FIGURA 22. Árbol del proyecto: Accesorios

En el apartado 'Fixtures' se pueden agregar mesas, 'racks', soportes, transportadores...

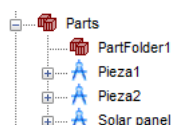


FIGURA 23. Árbol del proyecto: Objetos.

En el apartado '*Parts*' se encuentran todos los archivos CAD empleados como objetos a manipular.

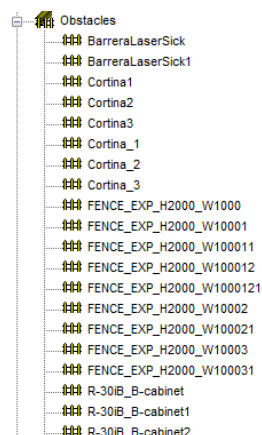


FIGURA 24. Árbol del proyecto: Obstáculos.

En el apartado '*Obstacles*' se encuentran todos los archivos CAD empleados como objetos que no son a manipular. Por ejemplo: vallas delimitadoras, decorativos...

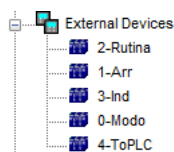


FIGURA 25. Árbol del proyecto: Dispositivos Externos.

En el apartado '*External Devices*' se puede configurar las conexiones con Servidores OPC, robots y CNCs externos.

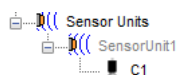


FIGURA 26. Árbol del proyecto: Sensores.

En el apartado '*Sensor Units*' se pueden configurar sensores y cámaras de visión artificial.

Finalmente, en la parte inferior de la interfaz se tienen una serie de indicadores:

- Naranja: robot seleccionado.
- Azul: programa seleccionado del robot seleccionado.
- Morado: muestra de errores del robot seleccionado.
- Amarillo: velocidad de ejecución ('*Override*').

Al crear una herramienta, una máquina, un accesorio, un objeto o un obstáculo se puede acceder a una biblioteca de objetos CAD que incluye RoboGuide. También se puede acceder a ella presionando el botón de la barra de herramientas señalado a continuación en rojo:

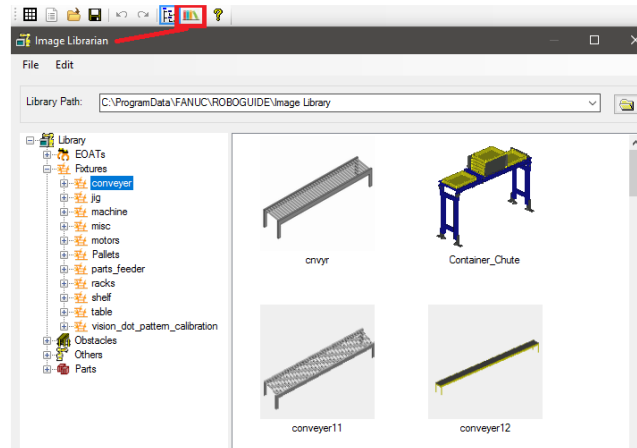


FIGURA 27. Biblioteca de objetos CAD.

CAPÍTULO 3.3.2. Teach Pendant

La *Teach Pendant (TP)* de Fanuc es una consola de mano donde el usuario interactúa con el robot a nivel de programación, notificación y tratamiento de errores, y de configuración del propio robot (entradas/salidas, registros, parámetros, configuración del sistema ...). Consta de una pantalla que muestra el contenido y una botonera en la que el usuario puede desplazarse por la *TP* y utilizar distintas funcionalidades como mostrar la configuración de los ejes, variar la velocidad de los movimientos si se está en modo manual, mover eje a eje el robot... Además, cuenta con una seta de emergencia y de una llave de encendido/apagado del terminal de enseñanza. Por la parte trasera tiene una maneta llamada '*deadman*' que siempre debe estar presionada mientras se usa la *TP* por motivos de seguridad. Si esta está en uso y se suelta el '*deadman*' se genera una emergencia – un error [16].



FIGURA 28. Teach Pendant física y de simulación.

CAPÍTULO 3.3.3. iRVision

La configuración de la visión artificial tiene un entorno aparte. Para poder acceder a él se debe hacer lo siguiente:

1º) Una vez agregada la opción de visión artificial y el sensor de visión correspondiente al robot deseado – esto se explicará en el capítulo 5.1.6. de este documento – seleccionar dicho robot. El árbol del proyecto tendrá una apariencia así:

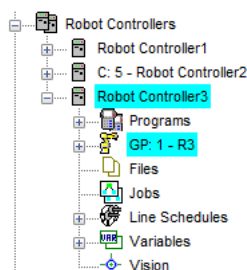


FIGURA 29. Robot con visión artificial.

2º) Se deberá acceder al navegador correspondiente dentro de la pestaña Robot. Se recomienda utilizar Internet Explorer.

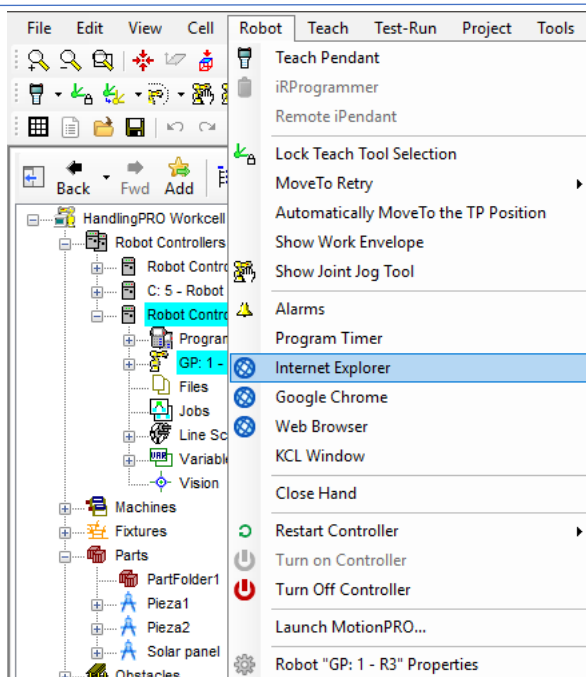


FIGURA 30. iRVision: Robot > Internet Explorer

3º) Si se han realizado los pasos correctamente. Se debería mostrar la página de bienvenida del robot con todas las funcionalidades de las que dispone. En ella se debe mostrar el apartado de iRVision.

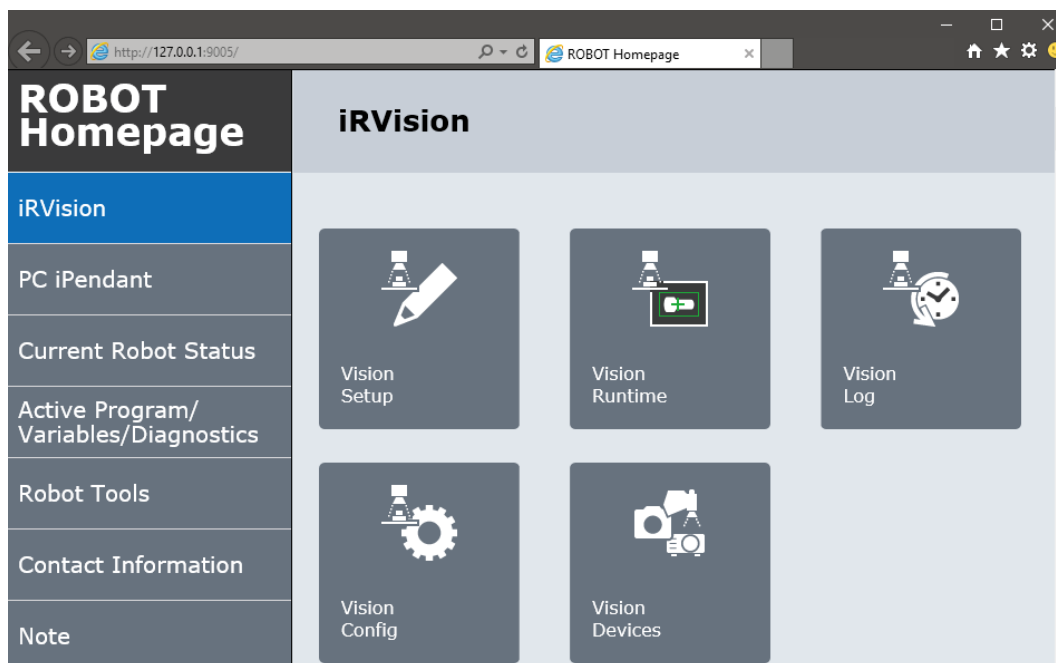


FIGURA 31. iRVision: Robot Homepage > iRVision

Como se aprecia en la figura anterior, dentro del apartado de iRvision se dispone de cinco pestañas:

1. **Vision Setup.** En esta pestaña se configura toda la visión artificial y por ello es la más importante y útil. Al acceder a ella se mostrarán tantas configuraciones como cámaras/sensores se disponga ('*Camera Data*') y tantas herramientas de procesos de visión como se hayan configurado ('*Vision Process Tools*'). En el punto 4 de este documento se abordará la configuración del '*Camera Data*' y '*Vision Process*' Tool de la imagen siguiente.

iRvision Vision Setup						
Name	Comment	Type	Created	Modified	Size	
Camera Data (1)						
CAMARAPIEZAS	Cámara para Visión Artificial	2D Camera	09-AUG-2022 09:55:08	09-AUG-2022 09:57:24	5380	
Vision Process Tools (1)						
RECONOCER_PIEZAS	Proceso para discernir entre dos tipos de piezas	2-D Single-View Vision Process	21-JUL-2022 10:16:18	09-AUG-2022 10:25:00	25251	

Below the table is a toolbar with icons for CREATE, EDIT, FILTER, COPY, DETAIL, and DELETE.

FIGURA 32. iRvision: Vision Setup

2. **Vision Runtime.** Muestra la imagen que capta la cámara de visión durante la ejecución del programa.
3. **Vision Log.** En esta pestaña se pueden ver históricos almacenados.
4. **Vision Config.** Aquí se pueden modificar aspectos como la ruta de almacenamiento de la información recopilada por la cámara/sensor, deshabilitar la muestra de las imágenes captadas por la cámara durante la ejecución...
5. **Vision Devices.** En ella se pueden ver los dispositivos de visión.

CAPÍTULO 3.4. Siemens

CAPÍTULO 3.4.1. TIA Portal

Total Integrated Automation Portal o *TIA Portal* es un potente software de la marca Siemens que permite la programación de autómatas; creación de interfaces hombre-máquina; configuración de objetos tecnológicos (PC, PC industrial, controladores embebidos...); y parametrización de accionamientos y arrancadores. TIA Portal permite interconectar todos los dispositivos

configurados si esto es posible, realizar configuraciones de seguridad avanzadas y gestionar accesos online entre otras muchas funcionalidades.

TIA Portal V15.1, la versión utilizada en este proyecto tiene la siguiente apariencia denominada “Vista del Portal”:

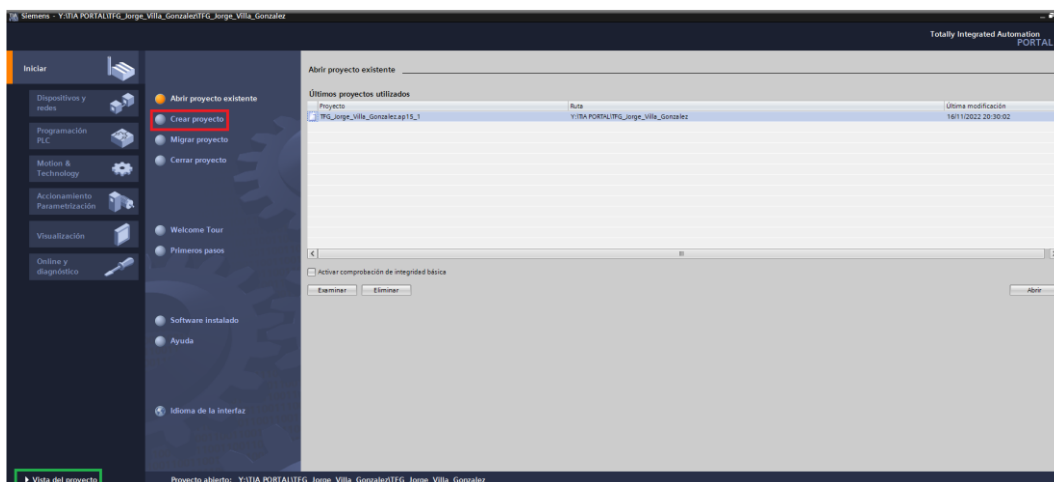


FIGURA 33. TIA Portal: Vista del portal.

Como ya se tiene mi proyecto creado, aparece la opción de la figura 33. Sin embargo, el proyecto mencionado se ha creado de cero. Por lo tanto, se debe hacer ‘click’ en la opción “Crear proyecto” (señalado en rojo). Se le dará un nombre, una ruta donde se desea almacenar, el nombre del autor y comentarios si se desea.

Una vez creado el proyecto, se pulsa sobre “Vista del proyecto” (señalado en verde) y despliega la siguiente interfaz:



FIGURA 34. TIA Portal: Vista del proyecto.

Desde esta interfaz se trabaja. Primero de todo se deberán agregar los dispositivos que se deseen utilizar. Para ello se ha de pulsar en la opción “Agregar dispositivo” situada en el árbol del proyecto (señalada en rojo en la siguiente figura).

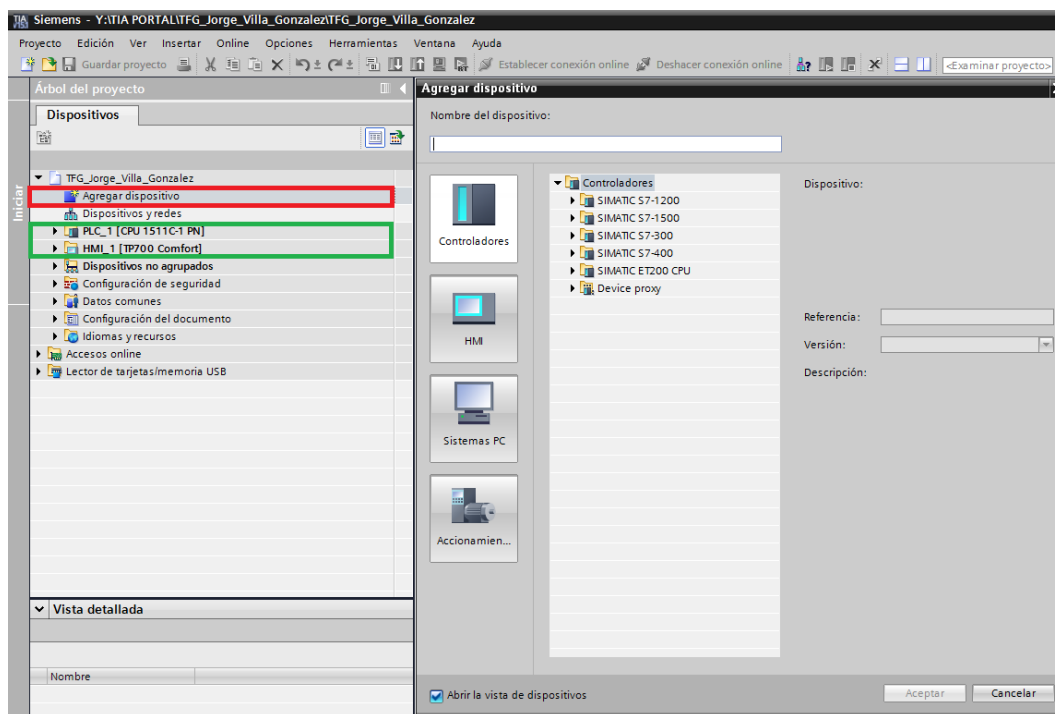


FIGURA 35. TIA Portal: Agregar dispositivo.

Se dispone de una gran variedad de dispositivos. Deberán seleccionarse los que se quieren agregar, uno por uno, especificando la versión del firmware deseada. Cuando se hayan agregado, se verán en el propio árbol del proyecto como se señala en la figura 35 de color verde; y estarán listos para su configuración/programación.

La Vista del Proyecto, mostrada en la figura 34, está dividida en diferentes secciones cuyas funcionalidades más importantes se detallan a continuación:

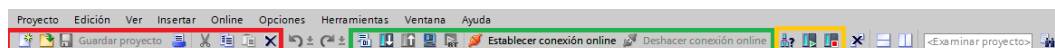


FIGURA 36. TIA Portal: Barra de Herramientas.

- **Barra de Herramientas.** Señalado en rojo en la figura anterior se tienen todas las funciones relacionadas con el proyecto: crear, abrir, guardar, eliminar, copiar... En verde se señalan las funcionalidades que tienen que ver con la simulación o funcionamiento real del dispositivo: compilar, cargar en dispositivo, iniciar simulación, establecer/deshacer

conexión online donde se puede visualizar el valor de las variables del dispositivo en tiempo real. En amarillo se señalan funciones que tienen que ver con la puesta en marcha/pausa del dispositivo además de la visualización de dispositivos accesibles para cargar los programas realizados.

- **Árbol del Proyecto.** Sencillamente, aglutina los puntos de acceso a ventanas de modificación del proyecto. Desde el árbol del proyecto se abre la ventana que permite agregar dispositivos, interconectarlos, configurar los dispositivos agregados, su seguridad...

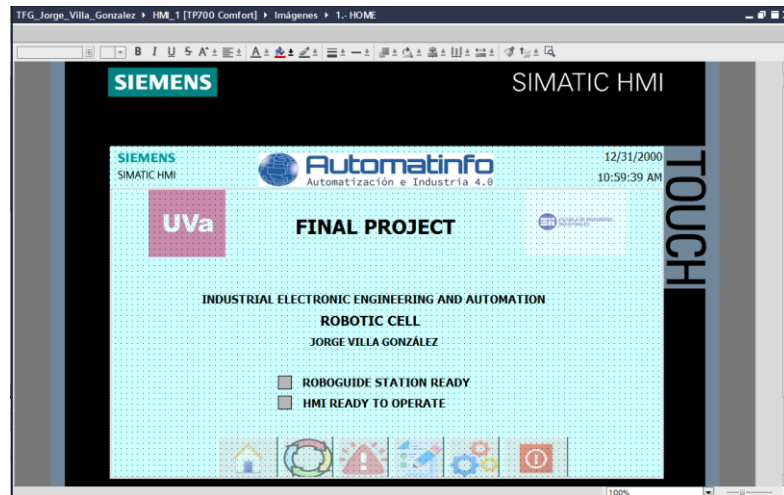


FIGURA 37. TIA Portal: Área de Trabajo.

- **Área de Trabajo.** Esta es la zona donde se despliegan las ventanas de modificación del proyecto. Se podrá programar PLCs, configurar HMIs, crear variables, etc.

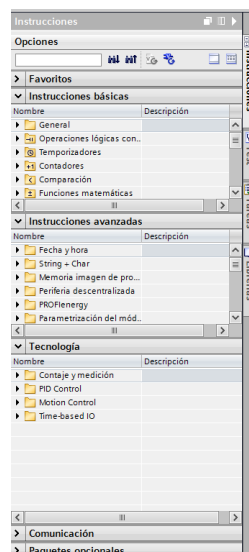


FIGURA 38. TIA Portal: Herramientas de configuración.

- **Herramientas de Configuración.** En esta zona aparecen infinidad de funciones según el dispositivo con el que se esté trabajando. Por ejemplo, si se está programando un PLC; aparecerán instrucciones. Si se está configurando un HMI; aparecerán opciones para agregar elementos a la pantalla que se está modificando...

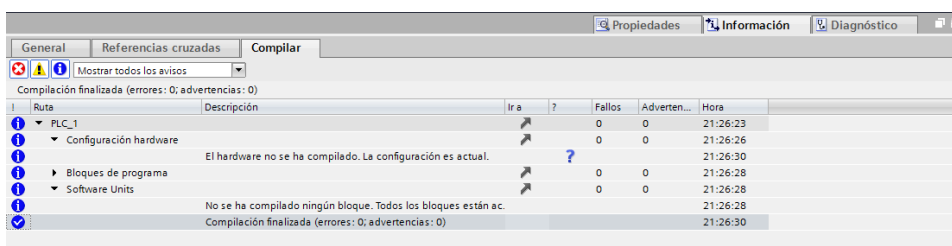


FIGURA 39. Ventana de Propiedades, Información y Diagnóstico.

- **Ventana de Propiedades, Información y Diagnóstico.**
 - **Propiedades:** permite parametrizar un autómata, añadir eventos a un elemento de una HMI, configurar la seguridad de un PC Industrial...
 - **Información:** muestra los avisos y errores que se puedan dar durante la compilación y ejecución del proyecto, información de referencias cruzadas, avisos que lanza TIA Portal como que se ha abierto un proyecto...
 - **Diagnóstico:** muestra el estado de los dispositivos, si se han conectado satisfactoriamente, avisos...

CAPÍTULO 3.4.2. PLCSim

Se podría querer testear un proyecto realizado en TIA Portal en el que se incluyese un autómata programable. Tras haberlo programado podría surgir la problemática de no disponer de un PLC físico en el que cargar el programa desarrollado para realizar las pruebas pertinentes. PLCSim es una herramienta de TIA Portal que ataja este problema mediante la simulación de una copia virtual del autómata programable utilizado en el proyecto.

Para acceder a PLCSim, primeramente, se debe de tener instalada dicha herramienta ya que no viene de serie con TIA Portal. Una vez escrito un programa que correrá en el PLC virtual, se debe seleccionar el PLC en el árbol del proyecto y, posteriormente, hacer *click* en el siguiente icono de la barra de tareas que está señalado en rojo:

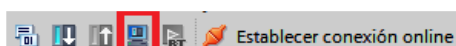


FIGURA 40. PLCSim: Iniciar simulación.

Entonces se abrirán dos ventanas:

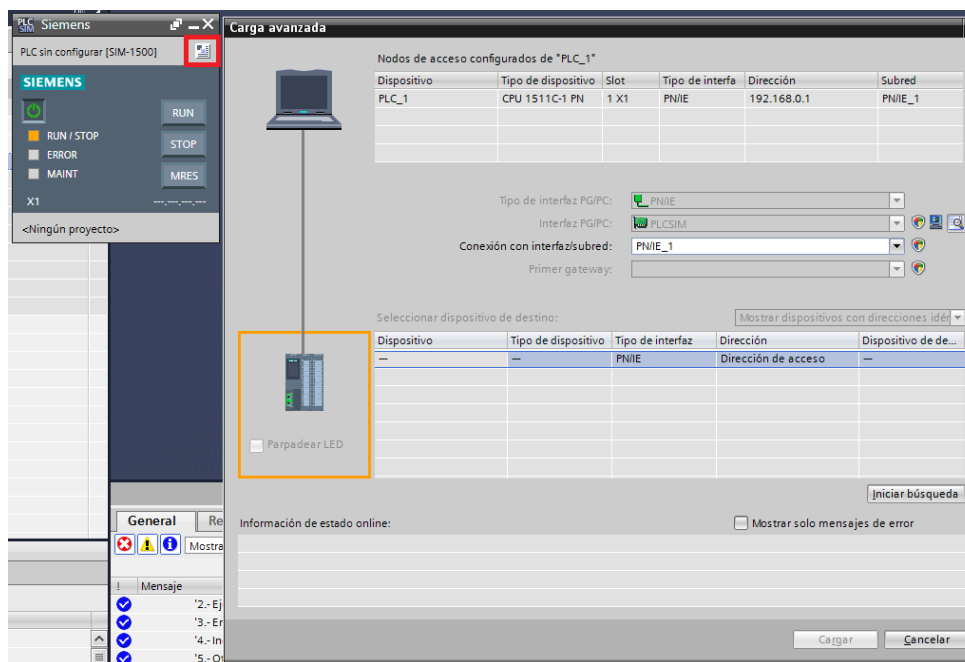


FIGURA 41. Interfaz PLCSim y ventana de búsqueda del PLC virtual

La ventana de la izquierda es la interfaz de PLCSim en una vista comprimida. Para desplegar la interfaz en vista extendida se debe pulsar sobre el botón señalado en rojo. En esta vista extendida se podrán cargar las variables del PLC además de modificar sus correspondientes valores. Para ello se deberá crear un nuevo proyecto dentro de PLCSim que estará directamente vinculado con el proyecto en TIA Portal.

La ventana de la derecha es la de carga avanzada. En ella se realiza la búsqueda del PLC virtual. Para ello se especifica el tipo interfaz de red, en este caso Profinet; la interfaz PG/PC que será PLCSim ya que se desea simular; y la conexión con la subred donde se haya conectado dicho PLC virtual.

Cuando se haya encontrado y seleccionado, TIA Portal compilará el proyecto si no se ha compilado antes de iniciar la simulación. Además se abrirá la vista preliminar de carga en la que, si la compilación fue satisfactoria, se dará la opción al usuario de cargar el programa en el PLC virtual como se muestra en la figura siguiente.

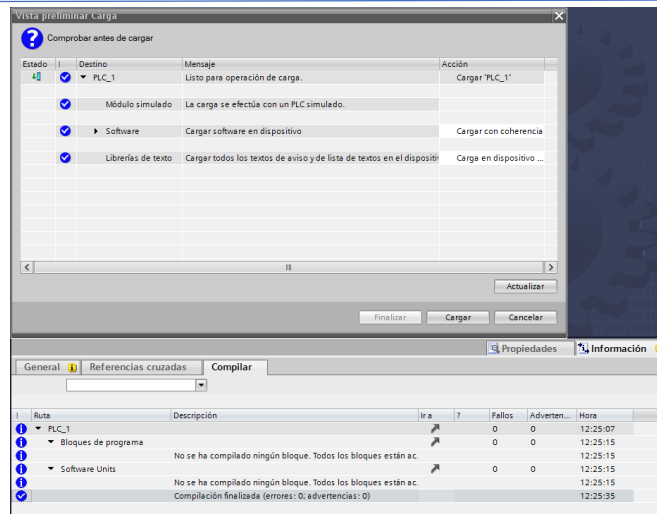


FIGURA 42. Compilación y carga en PLC virtual.

Tras la carga, se da la opción de arrancar el PLC virtual.

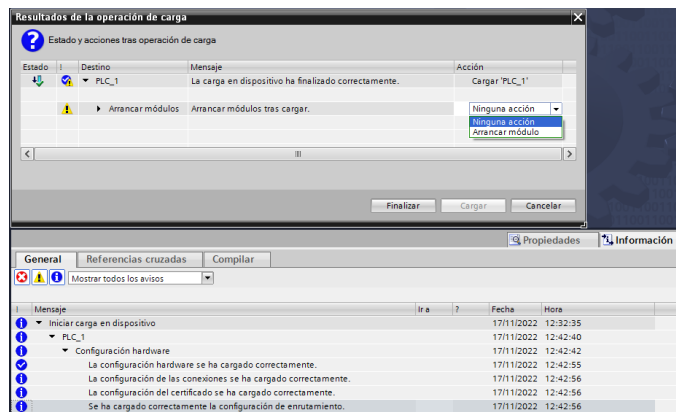


FIGURA 43. Arrancar PLC Virtual.

Tras arrancarlo, la interfaz de PLCSim tendrá el siguiente aspecto:

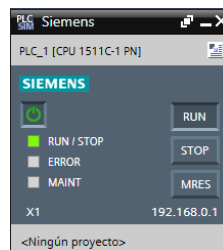


FIGURA 44. PLC Virtual en funcionamiento.

Si se despliega la vista extendida y se crea un nuevo proyecto, la interfaz tendrá este aspecto:

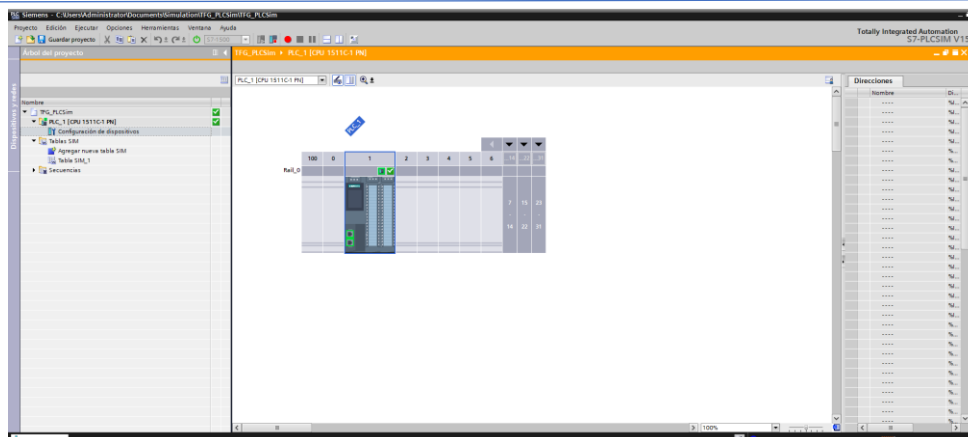


FIGURA 45. Interfaz PLCSim en vista extendida.

Desde aquí se podrán importar las variables para modificar sus valores, y visualizarlas en tiempo real.

CAPÍTULO 3.4.3. WinCC

Esta herramienta de TIA Portal es el análogo de PLCSim para interfaces hombre-máquina. Si no se dispone de una HMI física, WinCC permite simular una copia virtual.

Para abrir WinCC se debe haber instalado previamente, después se debe haber agregado y configurado una HMI al proyecto de TIA Portal. Hecho esto, se selecciona la HMI en el árbol del proyecto y, posteriormente, hacer 'click' en el icono de la barra de tareas que está señalado en rojo mostrado en la figura 40.

TIA Portal realiza la compilación de los cambios realizados (si lo hay) y lanza el programa WinCC donde se ejecuta la HMI que el usuario configuró. A diferencia de PLCSim; WinCC no cuenta con una interfaz propia donde crear un proyecto vinculado al desarrollado en TIA Portal, sino que es un programa donde se ejecuta la HMI con todas las funcionalidades que el programador configura.

Es importante recalcar que para que las consignas lanzadas desde la HMI que corre en WinCC tengan efecto sobre el PLC virtual, ambos deben estar simulándose simultáneamente además de estar conectados en la misma red. Esta conexión hará que las variables del PLC virtual tengan unas variables análogas en la HMI. Por lo tanto, si una se modifica desde cualquier lado de la conexión, la análoga también lo hará. Todo esto se explicará más en profundidad en el capítulo 5 de este documento.

A continuación se muestra una imagen de como se ve WinCC:

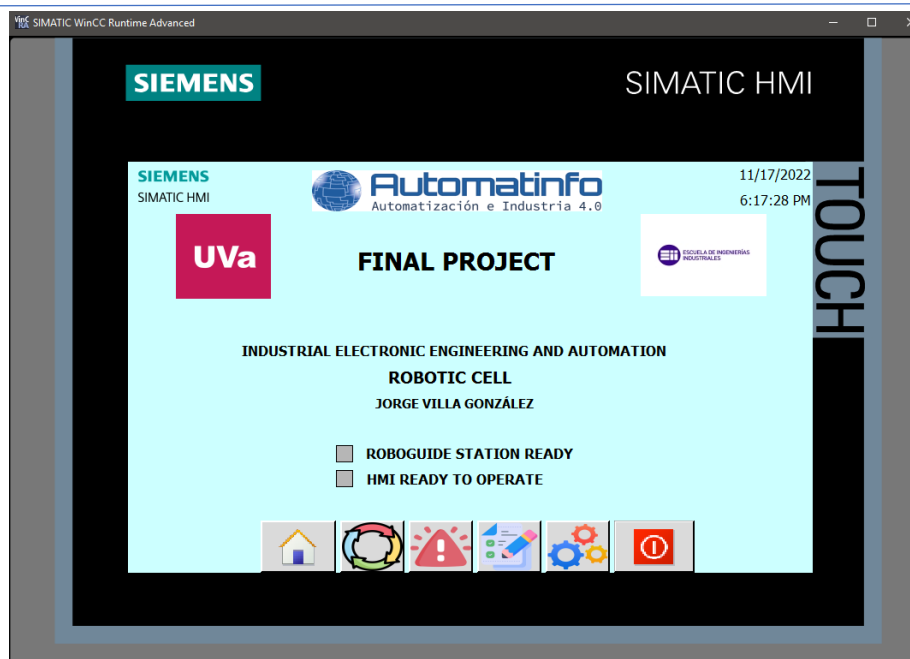


FIGURA 46. HMI en WinCC.

CAPÍTULO 3.4.4. NetToPLCSim

NetToPLCSim es una extensión de software que permite, a través de comunicación TCP/IP, acceder a la simulación del PLC (en PLCSim) desde la red a la que está conectada el PC donde se ejecuta la simulación utilizando la interfaz de red del propio PC. Esto permite simular aplicaciones cliente sin necesidad del PLC real.

De forma más sencilla: NetToPLCSim crea una vía de comunicación con el PLC de simulación regido por el protocolo de comunicación TCP/IP.

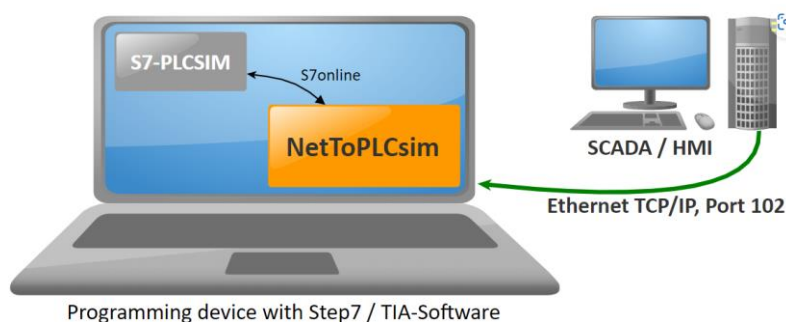


FIGURA 47. NetToPLCSim

De esta forma, la comunicación con el servidor OPC de KEPServerEX es posible siempre que se configure de la forma correcta.

NetToPLCSim tiene una interfaz muy sencilla:

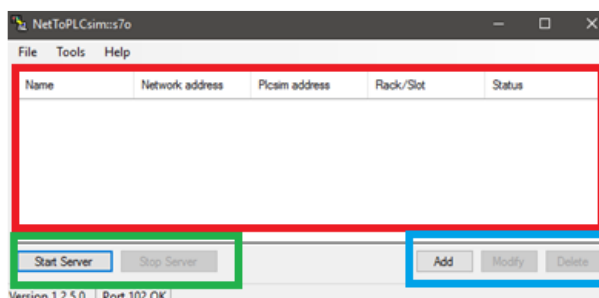


FIGURA 48. NetToPLCSim: Interfaz.

El área señalada en **rojo** mostrará las denominadas “estaciones” y sus respectivas configuraciones. Estas estaciones se refieren a los PLCs de simulación a los que se desea acceder. Mientras que las configuraciones hacen alusión al nombre que se le haya dado a la estación, la dirección IP de la red, la del propio PLC de simulación dentro de la red, el ‘Rack’ y ‘Slot’ en el que se tiene el PLC y el estado en el que se encuentra la estación desde el punto de vista de NetToPLCSim. Esto es importante ya que se puede tener el PLC ejecutando la simulación mientras la estación de NetToPLCSim está parada.

El área señalada en **azul** tiene la funcionalidad de crear, modificar o eliminar una estación. Cuando se trata de añadir o modificar una estación se abre el siguiente cuadro de diálogo en el que se deben introducir todos los parámetros del PLC al que se desea acceder:

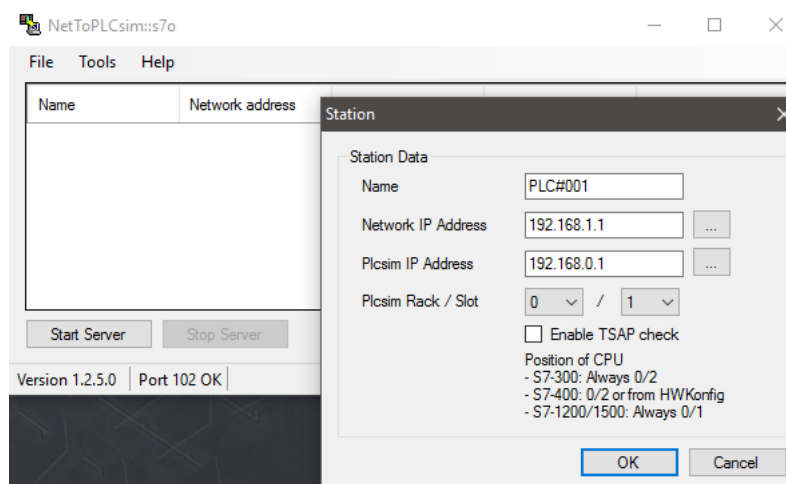


FIGURA 49. NetToPLCSim: Configuración de una estación.

El área señalada en **verde** permite ejecutar o parar la estación seleccionada.

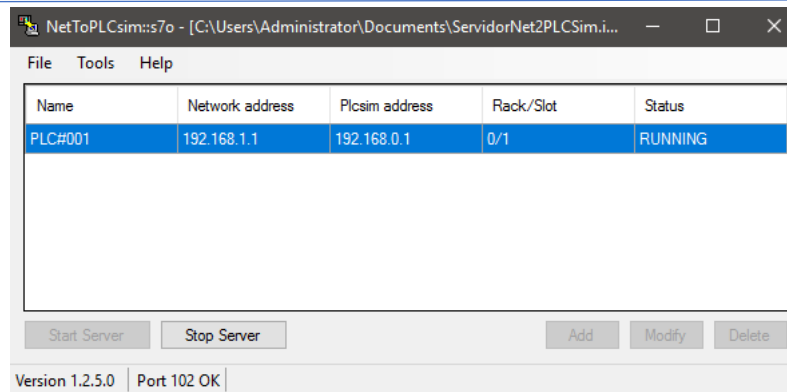


FIGURA 50. NetToPLCSim: Estación en ejecución.

CAPÍTULO 3.5. Comunicación OPC UA.

Los softwares anteriormente mencionados no tienen conectividad nativa, por lo que es necesario un mecanismo que permita el intercambio de datos para crear una aplicación eficiente y completa.

CAPÍTULO 3.5.1. KEPServerEX6

KEPServerEX proporciona una única fuente de datos de automatización industrial. Permite conectar, monitorizar y controlar diversos dispositivos de automatización y aplicaciones de software. KEPServerEX aprovecha el estándar OPC y protocolos IT para proporcionar a los usuarios una única fuente de datos industriales.

Dispone de múltiples drivers lo que permite una alta escalabilidad, pudiendo conectar casi cualquier dispositivo o sistema según se van incrementando las necesidades del proyecto. Todo ello con un interfaz común, lo que mejora su usabilidad.

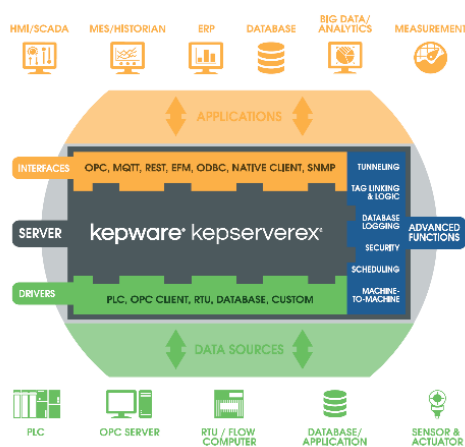


FIGURA 51. KEPServerEX.

CAPÍTULO 3.5.2. Interfaz de KEPServerEX6

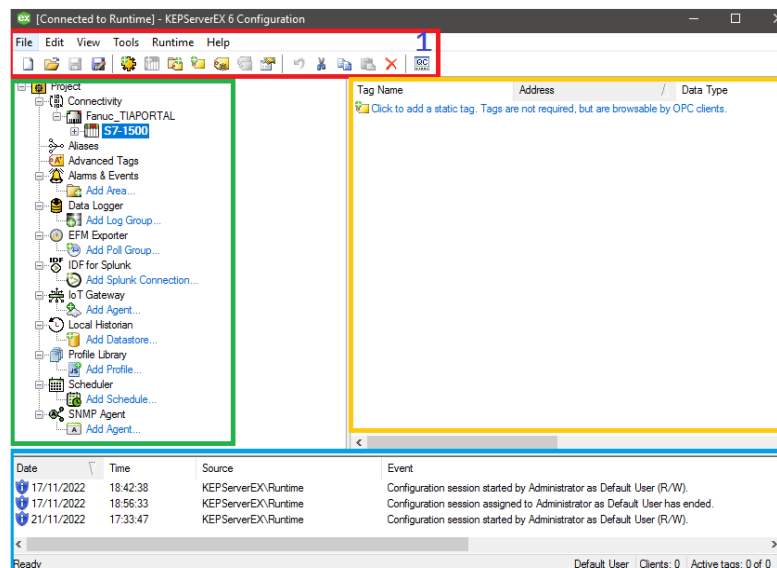


FIGURA 52. Interfaz de KepServerEX6

La interfaz del programa consta de diferentes secciones:

Señalado en **rojo** se tiene la barra de herramientas. En ella el usuario puede guardar, abrir, y crear un proyecto nuevo. Se podrá crear un nuevo canal, definir nuevos grupos de etiquetas, definir nuevas etiquetas y mostrar propiedades del elemento seleccionado. Además, en esta misma sección se tiene la funcionalidad “Quick Client” (señalada con un 1). Esta es la más significativa pues permite correr un cliente OPC que se conecta al servidor creado por el usuario. Así, se puede visualizar, en tiempo real, el valor de las etiquetas y señales que maneja el servidor además su correcto funcionamiento.

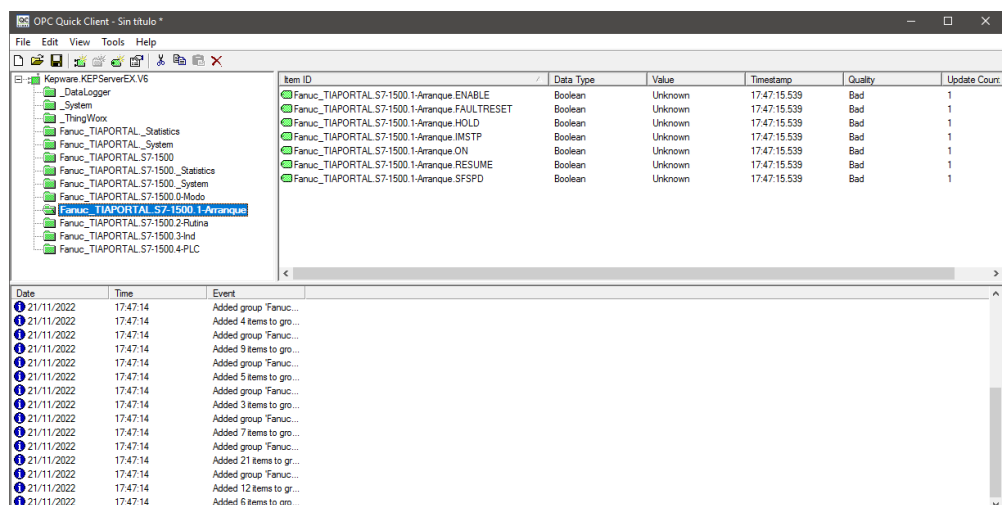


FIGURA 53. KEPServerEX: OPC Quick Client



En **verde** se tiene el árbol del proyecto. Aquí se tienen puntos de acceso a diferentes aspectos de la configuración del servidor OPC. El punto más relevante de cara al proyecto es el de 'Connectivity' ya que ahí se configuran los drivers y etiquetas de los dispositivos a configurar.

En **azul** se señala el registro de eventos. Esta sección es puramente informativa. Señala fallos de conectividad, estado del servidor, eventos, número de clientes OPC conectados, etiquetas activas...

Por último, en **amarillo** se remarca la zona de trabajo. En esta región se configura el servidor atendiendo a las diferentes necesidades del usuario.



Universidad de Valladolid

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Jorge Villa González



ESCUELA DE INGENIERÍAS
INDUSTRIALES



CAPÍTULO 4. DISEÑO DE LA SOLUCIÓN ADOPTADA

Como ya se ha mencionado anteriormente en este documento; este proyecto se ha realizado desde cero teniendo en cuenta lo aprendido durante el período de prácticas en empresa. Gracias a esto y a trabajar en un entorno de simulación, no se han tenido en cuenta especificaciones de un proyecto real. Sencillamente se ha pretendido llevar a la práctica los conocimientos adquiridos durante este período además de explotar al máximo las funcionalidades de los programas expuestos en el capítulo anterior.

A pesar de no tener en cuenta especificaciones de proyectos reales, se ha realizado un estudio de seguridades en entornos robotizados que influyen en el diseño de la estación en HandlingPro.

CAPÍTULO 4.1. Estudio de seguridad en entornos robotizados.

El trabajo con robots supone un potencial peligro para aquellos que operan en el entorno robotizado. Por ello, surge la necesidad de crear una regulación para evitar accidentes que pueda causar la interacción operario-robot.

CAPÍTULO 4.1.1. Potenciales riesgos de la robótica industrial.

Riesgo por instalación inadecuada. La instalación incorrecta de un sistema robótico y el hecho de no tener en cuenta las medidas de comprobación constituyen riesgos potenciales y hacen que el entorno de trabajo sea peligroso.

Riesgo de acceso a la instalación. Los operarios podrían no conocer los parámetros de seguridad para entrar en zonas restringidas donde trabajan los robots (falta de formación o información).

Riesgo de atrapamiento. Las trayectorias del robot pueden producir atrapamientos de los trabajadores entre el brazo robótico y los elementos del entorno.

Riesgo por contacto mecánico. Sucede cuando hay contacto entre el operario y el brazo robótico. Se puede dar en diferentes situaciones: cuando se están realizando pruebas de funcionamiento y las medidas de seguridad externas al robot están desactivadas; cuando el operario entra en la zona de operación del robot en pleno funcionamiento automático, cuando el robot supera los límites preestablecidos debido a interferencias electromagnéticas o errores de software, errores en el control del robot...



Riesgo eléctrico. Los trabajadores se exponen a corrientes eléctricas debido a que los robots están conectados a fuentes de energía, a que estos conservan la energía de otra manera, a un mal aislamiento en los cables...

Riesgo por proyección de materiales. En una instalación robotizada pueden salir eyectadas piezas, chispas, motas de polvo y restos de materiales durante el transcurso del proceso.

Una vez vistos los diferentes riesgos laborales relacionados con la robótica industrial se destaca que estudios realizados por el Instituto de Investigaciones de Seguridad en el Trabajo de Tokio revelan que el 90% de los accidentes en líneas robotizadas ocurren durante las operaciones de mantenimiento, ajuste, programación... Mientras que sólo el 10% ocurre durante el funcionamiento normal de la línea.

Capítulo 4.1.2. Normativas de seguridad en entornos robotizados industriales.

Seguidamente se citan las normativas europea y española vigentes ya que son las que competen a este trabajo. En concreto, las medidas que se deben tomar en el diseño de una instalación robotizada.

Primeramente, la **Directiva 2006/42/CE** del Parlamento Europeo y del Consejo [17], sirve como marco de referencia en cuanto al uso de maquinaria.

Toda maquinaria que cumpla los requisitos técnicos mínimos especificados en esta directiva irá acompañada del marcado europeo, necesario para su comercialización.



FIGURA 54. Marca de certificación o marcado europeo.

Las diferentes normativas estatales de referencia se basan en la directiva anterior. Estas normativas tratan de dar solución a los riesgos presentes en el uso de robots y sistemas industriales robotizados. La normativa española se divide en dos partes:

UNE-EN ISO 10218-1:2012: Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 1: Robots. [18]



En esta parte se detallan las directrices para realizar un diseño seguro; las medidas de seguridad pertinentes; y las instrucciones para un uso correcto de robots industriales.

UNE-EN ISO 10218-2:2011: Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración. [19]

En esta segunda parte se clarifican las pautas para la correcta integración de sistemas robotizados industriales según se define en la primera parte de la norma para una o varias celdas de robots industriales. Dentro del apartado integración se incluyen aspectos informativos de diseño, fabricación, instalación, funcionamiento, mantenimiento y retirada de servicio de robots o celdas robotizadas.

Se espera que en un futuro esta normativa sea sustituida por la **PNE-prEN ISO 10218** [20] que será dividida en dos partes como la actual.

Capítulo 4.1.3. Medidas de seguridad de diseño en entornos robotizados industriales.

En cuanto a la fase de diseño de una celda robotizada, se deben tomar una serie de medidas para reducir el riesgo de accidente. Entre ellas se encuentran:

- **Barreras de acceso a la instalación.** Se emplean barreras de seguridad que delimitan el entorno de trabajo restringiendo el acceso a personas. En caso de que algún operario o cualquier persona entre dentro de la celda deberá producirse parada inmediata. Estas pueden ser de dos tipos:
 - **Barreras materiales:** la norma **UNE-EN ISO 14120:2016** [21] dictamina que estas deben tener una altura de entre 1.80 y 2.00 metros siempre que sea posible. También establece el tipo de mallado que deben conformar las vallas para evitar la intrusión de cualquiera de las extremidades de personas en la instalación.



FIGURA 55. Barreras materiales.

- **Barreras inmateriales:** estas barreras generalmente suelen formarse gracias a dispositivos fotoeléctricos o dispositivos láser. Se instalan cuando el empleo de barreras materiales sea ineficiente o cuando su dimensionado presente imposibilidad para el cumplimiento de la norma. Su función es la misma: deben restringir el acceso a personas de forma que, si cualquier persona interrumpe el o los haces de luz de la barrera, se produce una parada inmediata.

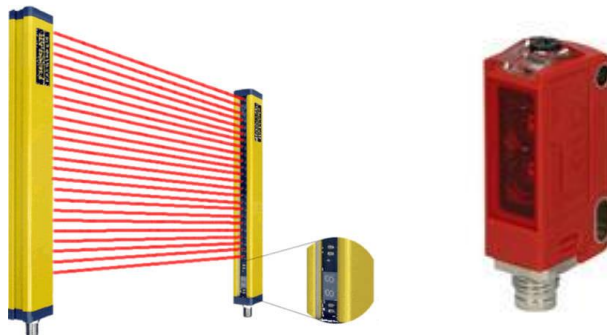


FIGURA 56. Barreras inmateriales

- **Dispositivos de control de acceso.** El acceso a la instalación suele estar restringido con dispositivos de protección como sensores de proximidad o de presencia, escáneres de área láser, dispositivos de acceso mediante código... Esto evita la intrusión de personal no autorizado en la instalación robotizada.
- **Dispositivos de señalización.** Han de instalarse tanto mecanismos audibles (sirenas) como visibles (balizas luminosas) que indiquen el estado de la instalación en todo momento.



FIGURA 57. Señalizaciones audibles y visibles.

- **Dispositivos de intercambio de elementos.** Si el operador requiere introducir o retirar elementos de la instalación, se utilizarán dispositivos que garanticen realizar estas operaciones de forma segura. Los mecanismos generalmente utilizados son: mesas giratorias, cintas transportadoras, cortinas foteléctricas...
- **Movimientos condicionados.** Si el operario debe entrar a la instalación, situándose en la zona de acción del robot cuando este se encuentra en estado de funcionamiento automático, se debe realizar la programación de tal forma que el robot no realice ningún movimiento en esos instantes de tiempo.
- **Zonas de reparación.** Deben existir zonas de reparación y mantenimiento fuera de la región de trabajo del robot pero que sean alcanzables por este. Mientras el operario esté en las zonas de reparación, el robot no realizará ningún movimiento.
- **Sistemas de mando.** Alrededor de la instalación se deben incluir dispositivos de mando y elementos de accionamiento no alcanzables por ningún elemento o piezas de las máquinas que permitan al operario realizar arranques de forma segura o rearmar la estación tras una parada de emergencia. En consecuencia, se evitan situaciones peligrosas, cumpliendo así con los requisitos de seguridad.
- **Sistemas de parada.** Toda instalación debe estar dotada de dispositivos que permitan la parada del funcionamiento de las máquinas – con o sin corte de suministro eléctrico de las propias máquinas.



FIGURA 58. Sistemas de mando y de parada.

CAPÍTULO 4.2. Descripción de los elementos de seguridad.

Puesto que el proyecto se ha realizado a nivel de simulación, no se han implementado todas las medidas de seguridad mencionadas en el punto anterior. Varias de ellas se han implementado en la interfaz hombre-máquina que se ha desarrollado como, por ejemplo: señalización, sistemas de mando y sistemas de parada. Se ha hecho de esta forma ya que la interfaz será la vía de interacción directa con la estación robotizada de simulación.

Sin embargo, se han implementado los dos tipos de barreras de seguridad explicados anteriormente en este documento.

CAPÍTULO 4.2.1. Barreras materiales.

En la biblioteca de elementos CAD de HandlingPro se dispone de barreras de seguridad cuyas dimensiones se pueden modificar. El robot manipulador más grande del que se dispone es el R-2000iC/125L cuya cota máxima vertical respecto al suelo es de alrededor de 3.5 metros según Fanuc. Sin embargo, dicha altura nunca se alcanza. Por lo tanto, se ha establecido que las barreras materiales de seguridad midan 2.00m de alto – el máximo por normativa.

Estas barreras serán utilizadas para delimitar el perímetro de la estación. Dejando un espacio libre para la entrada/salida de operarios. Tienen un mallado en forma de rombos que no permite el paso de extremidades.

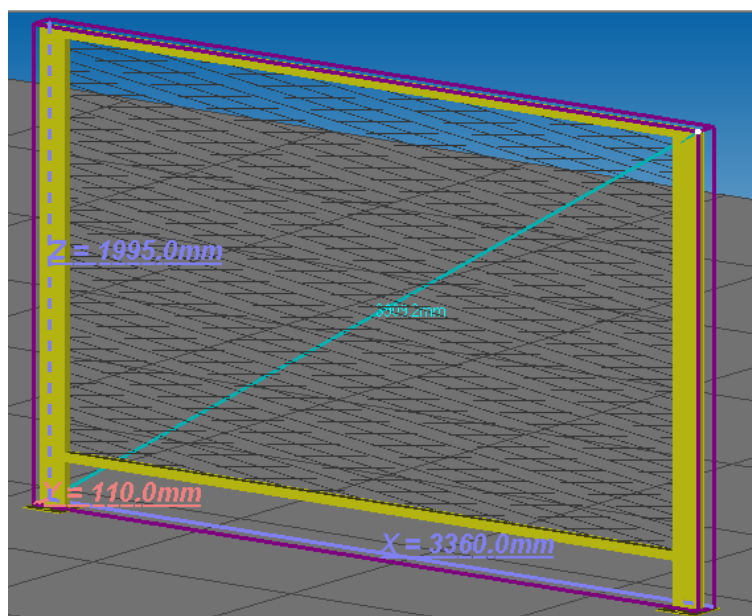


FIGURA 59. Barreras materiales y dimensiones.

CAPÍTULO 4.2.2. Barreras inmateriales.

HandlingPro no dispone de cortinas láser que se empleen como barreras de seguridad. Sin embargo, permite la inclusión de elementos CAD externos. Por ello, se ha importado la siguiente barrera inmaterial.

Esta se localiza en el espacio libre dispuesto para la entrada/salida del personal de trabajo. La cortina láser está pensada para que se active una señal de parada en caso de que una persona la atraviese.

Como las dimensiones de este dispositivo se pueden modificar a nivel de simulación, su altura se ha establecido para que abarque aproximadamente la totalidad de la altura de las vallas de materiales de seguridad.

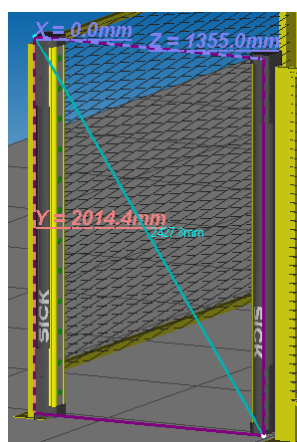


FIGURA 60. Barreras inmateriales y dimensiones.

CAPÍTULO 4.3. Descripción de los elementos a manipular.

En el proyecto se dispone de dos tipos de elementos a manipular: placas solares y elementos de fijación que serán soldados a estas.

CAPÍTULO 4.3.1. Placa Solar.

Se dispone de un único tipo de placa solar a la que le serán soldadas 4 fijaciones. Al igual que las barreras inmateriales, su archivo CAD ha sido importado a HandlingPro. Se modifican sus dimensiones para que quepa en la cinta transportadora que llegada a la estación; y en el soporte situado de la línea de salida en el que, tras las soldaduras, se coloca la placa ya tratada.

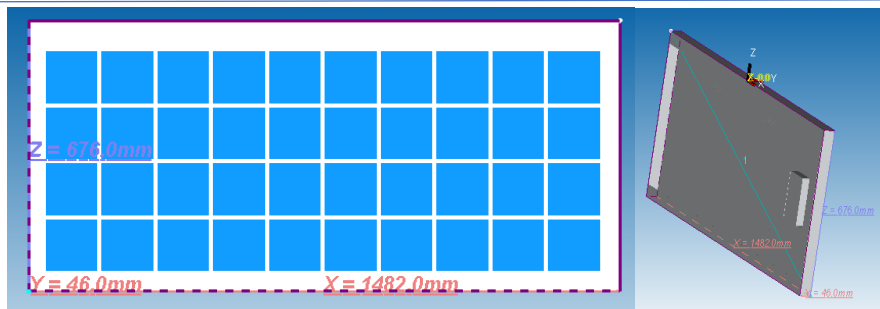


FIGURA 61. Placa solar y dimensiones.

CAPÍTULO 4.3.2. Elementos de fijación.

Como los elementos anteriormente descritos, se han importado dos tipos de piezas de sujeción de la placa solar a la estación. A un tipo se les bautiza como “Piezas1” y al otro como “Piezas2” que corresponden a las piezas de sujeción inferiores y superiores respectivamente. La utilidad de esta distinción se explicará en el capítulo 5 de este documento.

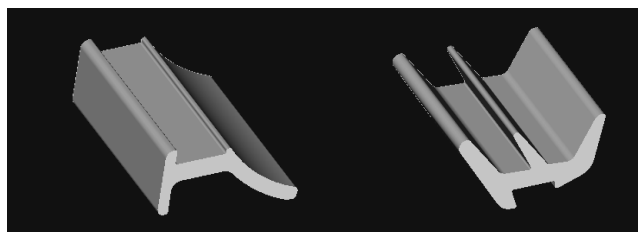


FIGURA 62. Piezas1 y Piezas2

La forma de las piezas es compleja por lo que no se detalla el valor de sus dimensiones, además de que carece de relevancia puesto que son significativamente más pequeñas que la placa solar. Las piezas soldadas a la placa se ven de la siguiente forma:

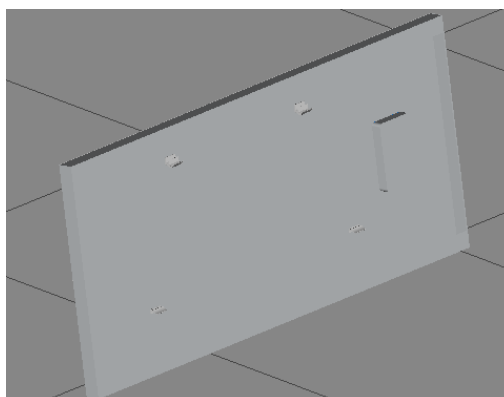


FIGURA 63. Placa solar con elementos de fijación.



CAPÍTULO 5. PROGRAMACIÓN Y CONFIGURACIÓN DE LOS ENTORNOS DE TRABAJO.

Se desea desarrollar una estación robotizada que simule el proceso de adhesión de elementos de fijación a una placa solar ya armada por medio de soldadura.

Se dispone de dos tipos de elementos de fijación: un tipo para la parte superior de la placa y otro para la parte inferior, que deben ser reconocidos mediante un proceso de visión artificial. Se realiza la soldadura de un total de cuatro piezas de fijación que corresponden a dos para la parte superior y dos para la inferior.

Las placas solares sin piezas entran en la estación por medio de una cinta transportadora citada como “cinta de llegada”. Una vez tratadas, saldrán de la celda por otra cinta transportadora citada como “cinta de salida”. Por esta cinta de salida, entra el soporte en el que se colocará la placa tras haber realizado las cuatro soldaduras.

Se considera que hay suficientes piezas en la propia estación de simulación; de forma que, cuando se coloquen 4 piezas, reaparecerán otras 4.

La célula robotizada es gestionada a nivel de usuario por una interfaz hombre-máquina dotada con funciones como: realización de movimientos de los robots en modo manual, lanzamiento del modo automático, solicitud de paradas...

Para la gestión de la seguridad de la estación y de la interfaz se tiene un autómatas programable S7-1500 de Siemens.

CAPÍTULO 5.1. Desarrollo en HandlingPro.

A continuación, se detalla todo lo relacionado con el desarrollo de la celda robotizada de simulación.

CAPÍTULO 5.1.1. Programación.

Como se explicó en el capítulo 3.3 de este documento; existen dos alternativas de programación para los robots de Fanuc: programación en Teach Pendant o programación KAREL.

En este proyecto se ha optado por la programación en Teach Pendant, ya que es la que he podido aprender durante el periodo de prácticas en empresa.



Este tipo de programación permite la creación de rutinas de movimiento basadas en puntos a los que se desea mandar al robot. Esto quiere decir que, para realizar una secuencia de movimientos, hay que posicionar al robot en los puntos a los que se desea llegar y registrarlos. En consecuencia, el robot realiza la cinemática inversa y registra la configuración de sus ejes necesaria para llegar a esos puntos guardados en memoria.

Adicionalmente, la programación en Teach Pendant tiene multitud de sentencias para realizar bucles, tomar decisiones en función del valor de señales, incluir técnicas de programación avanzadas como el '*Line Tracking*'...

CAPÍTULO 5.1.1.1. Comandos de movimiento

Estos son los comandos más esenciales dentro de un programa. Su estructura básica consta de cinco elementos más las posibles opciones que el programador requiera:

J	P	[1]	100%	FINE	OPTIONS
(1)	(2)	(3)	(4)	(5)	(6)

El primer elemento (1) expresa el tipo de movimiento que realiza el robot. Existen varios tipos de movimiento:

- J: movimiento cómodo.
- L: movimiento lineal.
- C: movimiento circular.

El segundo elemento (2) designa el tipo de posición al que se quiere acceder. A decisión del programador:

- P: posición (local al programa).
- PR: registro de posición (global a todos los programas).

En el tercero (3) se tiene el distintivo de la posición o registro al que se quiere acceder.

El cuarto (4) establece la velocidad a la que se requiere realizar el movimiento. Esta velocidad se puede expresar de muchas formas:

- X %: porcentaje respecto a los ejes.
- X inch/min: pulgadas por minuto.
- X deg/sec: grados por segundo.
- X mm/sec: grados por segundo.
- X cm/min: grados por segundo.
- seg: segundos.

El elemento quinto (5) establece la terminación del movimiento, es decir, la tolerancia con respecto al punto guardado:

- FINE: sin tolerancia, se desea llegar exactamente al punto guardado.
- CNTX: tolerancia continua (X va de 0 a 100).

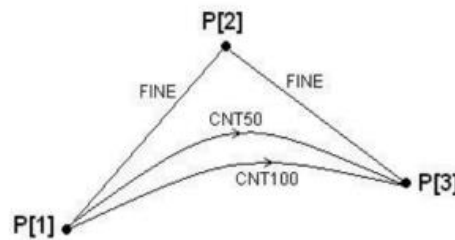


FIGURA 64. Terminación del movimiento.

El sexto elemento (6) y posteriores sirven para añadir opciones al movimiento. Algunas opciones son:

- **Offset, PR[X]:** se mueve al punto con un offset almacenado en un registro de posición.
- **Coord:** se especifican las coordenadas sobre las que está definido el punto.
- **Frames:** especifica el espacio de trabajo sobre el que se sitúa el punto.

CAPÍTULO 5.1.1.2. Otros comandos.

Para dotar al programa de mejores prestaciones se tienen una serie de instrucciones. Entre las de uso más frecuente destacan:

- **Salto incondicional.** Para realizar un salto incondicional dentro de un programa se tienen dos sentencias: **JMP LBL[X]** con la que se realizará el salto a la etiqueta con el distintivo X (**LBL[X]**). Resulta útil para realizar bucles.
- **Salto condicional.** Se realiza un salto (o bucle) a una marca del propio programa solo si se cumple la condición impuesta. En caso contrario se continua la ejecución del programa. Las sentencias de salto condicional son **IF** y **SELECT**.
- **Llamadas a subrutinas.** Con la sentencia **CALL** se podrán realizar llamadas a otros programas que se hayan desarrollado. La subrutina se ejecuta completamente y tras su finalización, se retoma el programa origen en la siguiente línea de código de la de la propia llamada.

- **Esperas.** Permite pausar la ejecución de un programa durante un cierto tiempo (**WAIT[tiempo]**) o hasta que se cumpla una condición (**WAIT[condición]**).
- **Registros.** Son variables reales (32 bits) o enteros, de acceso global desde cualquier programa. En ellos es posible almacenar el resultado de una operación aritmética, o introducir un valor ya sea una constante, un valor procedente de alguna señal de entrada-salida, otro registro o un elemento de un registro de posición. La sentencia es **R[X]**.

Un ejemplo sencillo que integra varias de las sentencias anteriormente descritas es la secuencia de movimientos para tomar un elemento del entorno:

```
PICKUP 6/6
1:L P[1] 3000mm/sec FINE
  : Tool_Offset,PR[2:OffsetZ]
2:L P[1] 100mm/sec FINE
3: WAIT .50(sec)
4: CALL PICKUP_SIM
5:L P[1] 100mm/sec FINE
  : Tool_Offset,PR[2:OffsetZ]
[End]
```

FIGURA 65. Ejemplo de programación en Teach Pendant.

Primero se realiza un movimiento de aproximación. Este es un **movimiento lineal a máxima velocidad** hacia el punto donde está situado el elemento a coger; pero con una **desviación vertical**, almacenada en el **registro de posición número 2**: “OffsetZ”. Esta desviación se realiza respecto al eje Z del sistema de coordenadas de la herramienta (**Tool Offset**). Si se elige la opción offset, el sistema de coordenadas varía, y como consecuencia el punto al que llega el TCP del robot podría ser diferente.

Seguidamente se realiza otro **movimiento lineal** a una **velocidad muy inferior** para ir exactamente (**FINE**) al punto donde se recogerá el elemento.

Antes de tomarlo, se realiza una **pausa** de medio segundo.

Para tomarlo, se **llama a una subrutina** donde está el código que permite realizar la animación del movimiento de tomar el objeto. Esta subrutina se explica en el capítulo 5.1.1.3. ya que no se realiza en la propia Teach Pendant.

Tras tomar el objeto, se vuelve al **punto de aproximación** a una **velocidad muy moderada de forma lineal**.

CAPÍTULO 5.1.1.3. RoboGuide Simulation Programm Editor.

Este editor es una herramienta aparte dentro de HandlingPro que permite realizar programas del mismo tipo que los de la propia Teach Pendant – por lo que pueden ser modificados en esta – con la ventaja de simular las animaciones de acciones del robot como la tomar o dejar elementos del entorno.

En consecuencia, se puede evaluar de forma “offline” la correcta ejecución de un ciclo de trabajo, evitar la colisión de los objetos tomados con otros elementos del entorno, ajustar los tiempos de llegada de los elementos, variar la velocidad de las cintas transportadoras...

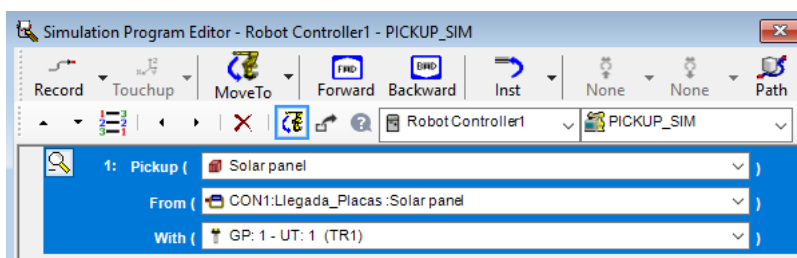


FIGURA 66. Programa en Simulation Program Editor.

Esta es una subrutina que únicamente cuenta con una instrucción (aunque se pueden incluir comandos de movimiento, pausas, saltos incondicionales...). Es la mencionada en el capítulo 5.1.1.2. Aquí se programan las animaciones específicas como la de este ejemplo: Tomar un elemento.

En este caso el Robot1 tomará un panel solar que viaja sobre la cinta transportadora de entrada (CON1) con la herramienta TR1 (Tool Robot 1) que es una ventosa de vacío de la que se dispone. Todo ello se especifica en la instrucción ‘Pickup’ como se muestra a en la figura 66.

CAPÍTULO 5.1.1.4. Line Tracking [22].

En el proyecto se ha aplicado esta técnica avanzada de programación de robots basada en que el robot realiza un seguimiento de la posición de uno o varios elementos dentro de una o varias líneas de producción. De este modo, se hace innecesaria la parada de la línea para tomar o dejar el elemento que va por ella – ahorrando, así, tiempo de ciclo.

En este caso en concreto, se realiza Line Tracking en dos situaciones:

- 1) En la toma de la placa solar sin tratar proveniente de la cinta de entrada.
- 2) En la colocación de la placa ya tratada sobre el soporte que viaja por la cinta de salida.

Aprovechando el mostrado en la figura 65 del capítulo 5.1.1.2. se explicará el seguimiento de línea para tomar un objeto. Pero previamente han de realizarse algunas configuraciones previas:

Antes de nada, el robot debe incluir las opciones de software de *'tracking'* para poder hacer el seguimiento de línea:

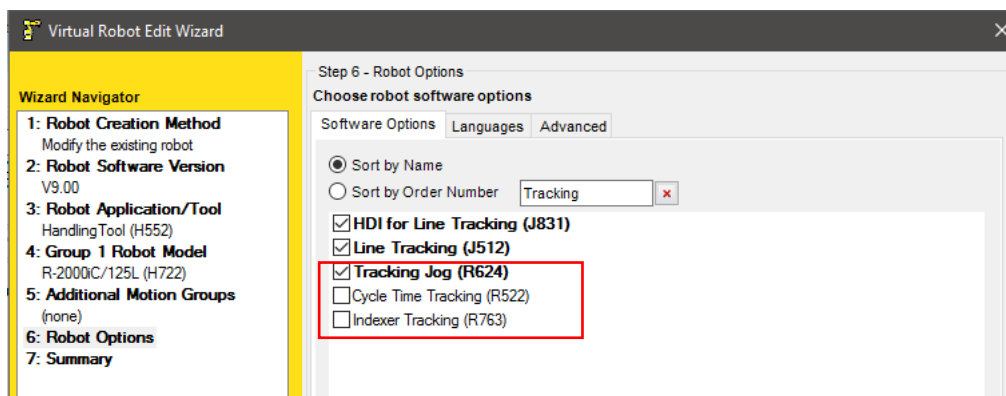


FIGURA 67. Opciones de software de Line Tracking.

Paso 1. En el apartado *'Fixtures'* del árbol del proyecto, hacer *'click'* derecho y añadir línea (*'add line'*). Para ello se elige la cinta según convenga de entre las de la librería de RoboGuide o alguna importada. Tras hacerlo, en dicho apartado aparecerá algo así:



FIGURA 68. Line Tracking: Línea.

Paso 2. Hacer *'click'* derecho y añadir el objeto que viaja por la línea (*'add link'*). Aparecerá algo así en el apartado *'Fixtures'*.



FIGURA 69. Line Tracking: Objeto.

Paso 3. Ahora se debe configurar todo lo relacionado con el objeto haciendo doble *'click'* en "Link1". En la pestaña "General" se le puede dar un nombre al objeto, especificar qué robot monitoreará su posición, el *'Line Schedule'* asociado, la posición donde se sitúa el *'encoder'* que se conecta al robot...

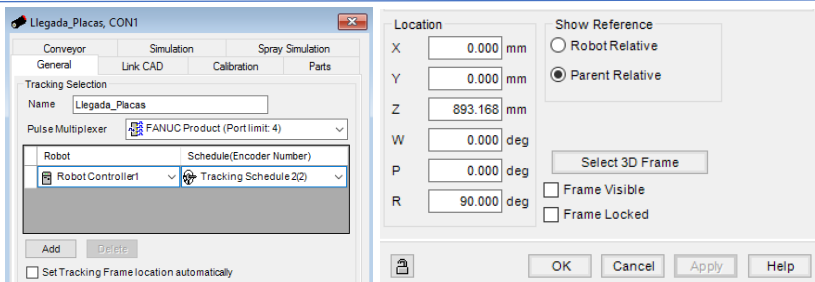


FIGURA 70. Line Tracking: Configuración del objeto.

Paso 5. En la pestaña “Conveyor” se modifica todo lo relacionado con la cinta transportadora: si de ella se toman (*‘infeed’*) o se dejan objetos (*‘outfeed’*), estipular el punto de llegada y de salida de objetos, tiempos de llegada de los objetos, velocidad de la cinta (que sea distinta de 0)...

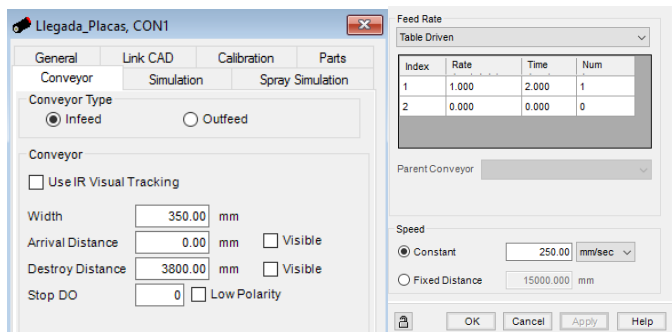


FIGURA 71. Line Tracking: Configuración de la cinta.

Paso 6. En el paso 4 se configuró el objeto, pero no se le agrega la opción de ser tomado/dejado. Para ello se accede a la pestaña “Parts” y se selecciona el objeto que se desea. Así será accesible a la hora de realizar la animación en RoboGuide Simulation Programm Editor. De esta forma se verá así dicha pestaña:

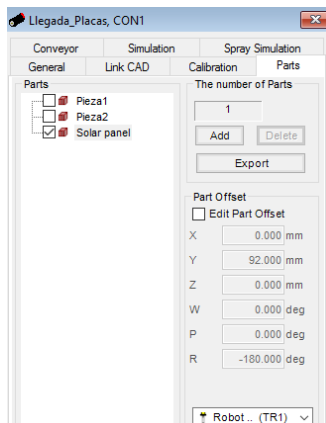


FIGURA 72. Line Tracking: Selección del objeto.

Cuando se hayan terminado estos pasos, se debe probar que el movimiento de la línea y el del objeto siguen la misma dirección. Para ello se debe alinear el eje X de la línea con el eje X del objeto haciendo que sus sentidos coincidan.

Para probar el buen funcionamiento: **Barra de Herramientas**→**Icono de cinta transportadora**→**Seleccionar la línea**.

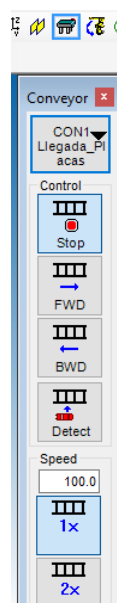


FIGURA 73. Line Tracking: Pruebas de movimiento.

Paso 7. Ahora se debe configurar el 'Line Schedule' seleccionado en el paso 3. Para ello: **Árbol del proyecto**→**Robot Controllers**→**Controlador del robot seleccionado en el paso 3**→**Line Schedules**→**Línea seleccionada en el paso 3**. Para configurarla, se abre el siguiente cuadro de diálogo:

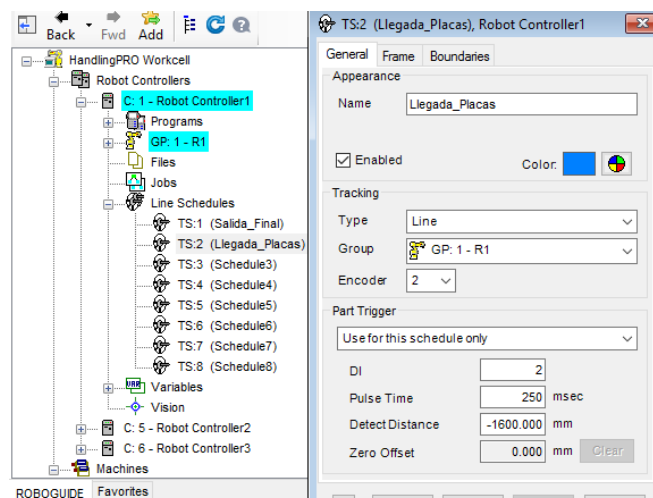


FIGURA 74. Line Tracking: Line Schedule - General.

Paso 8. En la pestaña “General” se especifica tipo de seguimiento (lineal, circular), el robot que lo realiza, el encoder del robot que monitorea la posición del objeto, la señal digital de entrada del robot que activará el seguimiento, el tiempo que se mantiene activa esta entrada digital (debe ser mayor que cero), y la distancia a la que se activará la dicha señal.

Paso 9. En la pestaña “Boundaries” se delimita la región del transportador en la que es posible realizar un seguimiento (dentro del propio alcance del robot). Esta región queda definida por el límite aguas arriba o ‘*upstream*’ y el límite aguas abajo o ‘*downstream*’. El límite aguas arriba se debe tener en cuenta como el límite por donde entra el objeto a la región de seguimiento. Mientras que el límite aguas abajo es por donde sale. Más allá de esos límites, el robot no realiza seguimiento y se producirá un error.

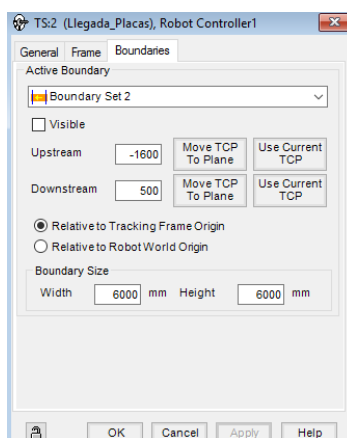


FIGURA 75. Line Tracking: Line Schedule - Boundaries.

Paso 10. En la pestaña “Frame” se varía la orientación de los límites antes mencionados. De modo que, sus ejes X queden alineados con el eje X de la cinta transportadora y del objeto que viaja por ella.

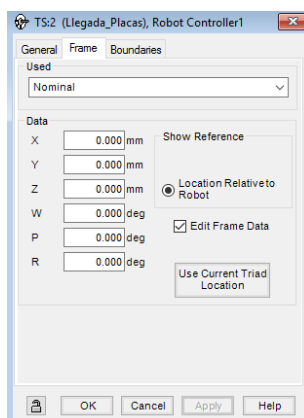


FIGURA 76. Line Tracking: Line Schedule - Frame.

Paso 11. Según el tipo de cinta que se haya seleccionado en el paso 5, se debe crear un programa de animación de movimiento de tomar o dejar un objeto. En este caso, se creó el mostrado y explicado en la figura 66 del capítulo 5.1.1.3. relativo a tomar un objeto. Pero esto solo genera esa animación – a no ser que se incluyan instrucciones de movimiento del robot.

Paso 12. Como no se incluyeron comandos de movimiento en esa subrutina, se debe crear un programa en Teach Pendant que integre esa animación con movimientos reales del robot. Un ejemplo de ese programa, es el mostrado y explicado en la figura 66 del capítulo 5.1.1.3.

Cuando se cree el programa, hay que establecer una configuración especial. Para ello, se selecciona ‘Select’ → ‘>’ → ‘Detail’ → ‘Next’. Una vez ahí, la Teach Pendant muestra el denominado “Line Track Program Header” donde se seleccionará el *Line Schedule* (1) seleccionado en el paso 3 y el *Boundary Set* (3) creado en el paso 9.

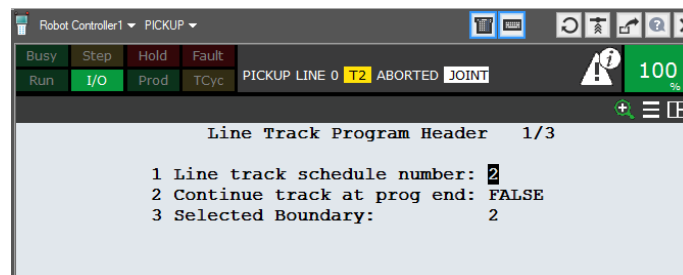


FIGURA 77. Line Tracking: Line Track Program Header.

Al haber configurado así el programa, **sólo** se podrán programar **movimientos lineales** (en caso de que se realice un seguimiento de una línea recta; si la línea es curva, solo se permitirán **movimientos circulares**). En caso de programar un movimiento cómodo, a la hora de la ejecución, producirá un error. Así pues, es conveniente usar este tipo de programas como rutinas a las que se llame desde un programa principal donde sí se pueden realizar movimientos cómodos.

Con todas las configuraciones previas listas, se procede a la programación del ‘Line Tracking’.

```
LINETRACKING 2/17
1: LINE[2] ON
2:
3: J PR[1:Home] 100% FINE
4:
5: LBL[30]
6: WAIT DI[2:LINE TRACKING ENTRADA]
7: =ON
8: LINECOUNT[2] R[3]
9: LINERATE[2] R[4]
10: IF R[4]<1, JMP LBL[30]
11: SETTRIG LNSCH[2] R[3]
12: SELBOUND LNSCH[2] BOUND[2]
13:
14: CALL PICKUP
15:
16: J PR[1:Home] 100% FINE
[End]
```

FIGURA 78. Programación del Line Tracking.

Línea 1. Primeramente, se activa la línea correspondiente; en este caso, la 2.

Línea 3. Se manda el robot exactamente a la posición registrada como “Home” con un movimiento cómodo a máxima velocidad.

Línea 6. El robot queda a la espera de que se active la entrada digital (DI2) configurada en el paso 8 que activa el seguimiento – por ello, su duración del pulso de debe ser mayor que cero; si es cero nunca se activa.

Línea 7. Se arranca a contar los pulsos del encoder asociado a la línea y se almacenan en un registro.

Línea 8. Seguidamente, se registra el ‘Linerate’ con ello se podrá verificar que el objeto se mueve en sentido correcto. Si esto no se cumple, se almacenarían números negativos en el Registro 4; y, en consecuencia, no se debe hacer el seguimiento. Entonces, si esto ocurre, se debe volver a esperar a que se reactive la señal digital 2 y verificar que el movimiento es el adecuado.

Esta verificación se realiza por medio de un bucle formado por las **líneas 9 y 5**: siempre que el sentido del movimiento sea erróneo, vuelve a esperar a un nuevo objeto.

Línea 11. Si el sentido del movimiento es el adecuado, continúa la ejecución hasta esta línea. Se establece el “disparador” del ‘Line Schedule’ configurado en el paso 7 (en este caso el 2) y se le asocia el registro que almacena los pulsos del ‘encoder’ con la que se monitorea la línea.

Línea 12. Se asocia el 'Boundary set' configurado en el paso 9 (en este caso el 2) al 'Line Schedule'.

Línea 14. Se llama a la subrutina de recogida del elemento que viaja sobre la línea explicada en el capítulo 5.1.1.2. La pausa que se realiza de medio segundo de duración sirve para que, en la animación de coger la placa solar, el robot siga al objeto antes de tomarlo y genere esa sensación de seguimiento de la línea. Como ya se ha mencionado, los movimientos únicamente pueden ser lineales porque la línea creada era recta (Ver figura 65).

Línea 16. Una vez tomado el objeto, se retorna a la posición 'Home' mediante un movimiento cómodo.

CAPÍTULO 5.1.2. Sincronización de señales.

En ocasiones, es necesario sincronizar las acciones de los robots para que puedan cooperar en la realización de las tareas evitando colisiones.

Existen dos alternativas posibles para llevar a cabo la coordinación:

- 1) Sincronizar los robots mediante la programación del PLC.
- 2) Interconectar señales de entrada y salida en HandlingPro.

Como la estación es de simulación y no se requiere un control tan robusto del proceso como el que se debería dar en una planta real, y como este documento servirá para aportar información sobre el uso del software de Fanuc; se opta por la segunda opción.

Num Points	Output Dev	Type	Start Point	I/O Tag	Input Dev	Type	Start Point	I/O Tag
1	Robot Controller1	DO	5	DO[5]	Robot Controller3	DI	4	DI[4]
1	Robot Controller3	DO	1	DO[1]	Robot Controller2	DI	1	DI[1]
1	Robot Controller2	DO	3	DO[3]	Robot Controller1	DI	6	DI[6]
1	Robot Controller3	DO	2	DO[2]	Robot Controller2	DI	2	DI[2]
1	Robot Controller3	DO	3	DO[3]	Robot Controller2	DI	3	DI[3]
1	Robot Controller3	DO	4	DO[4]	Robot Controller2	DI	4	DI[4]
1	Robot Controller3	DO	5	DO[5]	Robot Controller2	DI	5	DI[5]

FIGURA 79. Interconexión de señales digitales.

Un ejemplo: en el proyecto se tienen tres robots como se detallará en el capítulo 5.1.4. Tras haber hecho el Line Tracking para tomar la placa solar, el robot 1 activa una señal digital conectada a otra del robot 3 para iniciar el proceso de visión artificial. Posteriormente, para la realización de la soldadura de los elementos de fijación, el robot 2 y el robot 3 han de coordinarse. Uno coloca la pieza sobre la placa, y el otro la suelda a esta.

The screenshot shows two windows from a PLC simulation. The left window, titled 'Robot Controller1', displays a ladder logic program for 'RUTINAAUT' with line 13 highlighted. The right window, titled 'Robot Controller3', displays a ladder logic program for 'RUTINAAUT' with line 6 highlighted. Both windows show a status bar with '50%' and 'ABORTED JOINT'.

```
Robot Controller1 - RUTINAAUT
Run Step Hold Fault SRVO-037 IMSTP input (Group:1)
RUTINAAUT LINE 0 12 ABORTED JOINT 50%
RUTINAAUT 13/48
12: SETTRIG LNSCH[2] R[3]
13: SELBOUND LNSCH[2] BOUND[2]
14: CALL PICKUP
15:
16: J P[3] 100% FINE
17: J P[2] 100% CNT25
18: CALL DROP
19: DO[5]=ON
20: J P[2] 100% FINE
21: J P[3] 100% CNT25
22:

Robot Controller3 - RUTINAAUT
Run Step Hold Fault SRVO-037 IMSTP input (Group:1)
RUTINAAUT LINE 0 12 ABORTED JOINT 50%
RUTINAAUT 6/16
1: LBL[400]
2: J @PR[1:HOME] 100% CNT25
3:
4: DO[1]=OFF
5: DO[2]=OFF
6: WAIT DI[4:VA & COLOCA AUT]=ON
7: IF R[21:Piezas1Colocadas]>0 OR
: R[22:Piezas2Colocadas]>0 OR
: R[5-ID]<>0, CALL BORRAR_REGISTROS
8:
9: CALL VISIONPIEZAS
```

FIGURA 80. Robot1 (DO5) se coordina con Robot3 (DI4).

The screenshot shows two windows from a PLC simulation. The left window, titled 'Robot Controller3', displays a ladder logic program for 'VISIONPIEZAS' with line 28 highlighted. The right window, titled 'Robot Controller2', displays a ladder logic program for 'RUTINAAUT' with line 16 highlighted. Both windows show a status bar with '50%' and 'ABORTED JOINT'.

```
Robot Controller3 - VISIONPIEZAS
Run Step Hold Fault SRVO-037 IMSTP input (Group:1)
VISIONPIEZAS LINE 0 12 ABORTED JOINT 50%
VISIONPIEZAS 28/40
27: JMP LBL[30]
28:
29: LBL[20]
30: CALL PICKUP2
31: CALL DROPOFF2
32: IF R[20:PiezasQuedan]<>0,
: JMP LBL[50]
33: LBL[30]
34: DO[1]=ON
35: LBL[100]
36:

Robot Controller2 - RUTINAAUT
Run Step Hold Fault SRVO-037 IMSTP input (Group:1)
RUTINAAUT LINE 0 12 ABORTED JOINT 50%
RUTINAAUT 16/49
9: J PR[11:pos_intermedia] 100%
: CNT100
10:
11: WAIT DI[1:IR POSSOLDAR AUT]=ON
12: J PR[2:pos_soldar] 100% FINE
13:
14: IF (DI[2:SAI AUT]=ON) THEN
15: R[2:uframe]=1
16: CALL SOLDARARRIBA
17: R[1:contador]=R[1:contador]+1
```

FIGURA 81. Robot3 (DO1) se coordina con el Robot2 (DI1).

CAPÍTULO 5.1.3. User frames.

Un 'user frame' es un sistema de coordenadas creado por el programador sobre los que es posible expresar un punto al que mover el robot. Resultan especialmente útiles a la hora de realizar movimientos sobre superficies u objetos con orientaciones "complicadas" ajenos a un robot.

Para crear un 'user frame' existen diferentes métodos:

- **Tres puntos.** Se debe desplazar al robot a tres posiciones diferentes de la superficie u objeto que marcan: el origen de la orientación (origen del sistema de coordenadas), la dirección X, y la dirección Y. Se forma automáticamente el triedro directo en la posición donde se registró el origen de la orientación.
- **Cuatro puntos.** Este método se utiliza en procesos de visión artificial en los que se utilizan patrones de puntos para la calibración de la cámara. Se debe desplazar al robot a cuatro posiciones diferentes que marcan: el origen de la orientación, la dirección X, la dirección Y, y origen del sistema (comúnmente el centro del patrón de puntos). Se forma automáticamente el triedro directo en la posición donde se registró el origen del sistema (el último punto almacenado).

- **Entrada directa.** El menos recomendable ya que se necesitan saber con exactitud las coordenadas de los puntos que conforman el triedro directo e introducirlas por Teach Pendant.
- **Manual.** Este método solo es válido a nivel de simulación. Para crearlo se sigue la ruta **Árbol del proyecto>Robot Controllers>GP. X>User Frames.** Haciendo doble 'click' sobre el que se desea modificar, se podrá variar su posición y orientación.

CAPÍTULO 5.1.4. Desarrollo de la estación robotizada.

Como se aprecia en la figura siguiente, la instalación cuenta con tres robots de la marca Fanuc encargados de realizar el proceso.

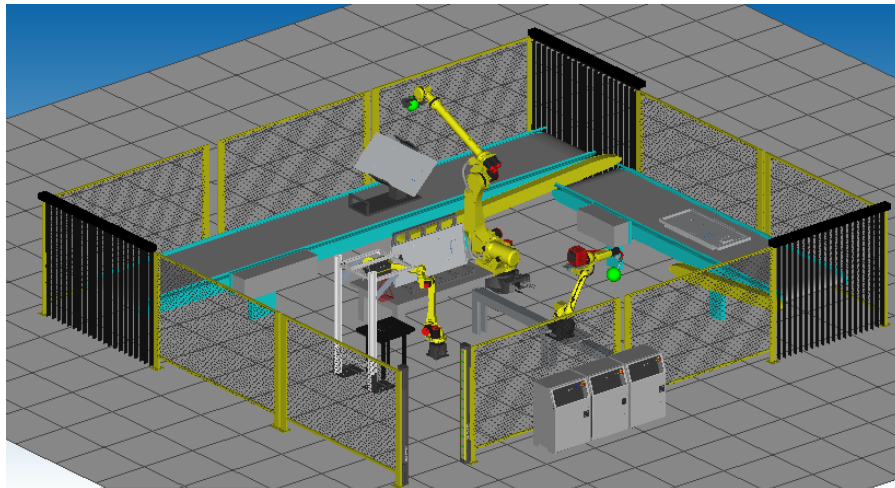


FIGURA 82. Celda robotizada

Se tienen dos cintas transportadoras: una de llegada de placas solares sin tratar, y otra de salida de placas ya tratadas.

Dos mesas de trabajo: una en la que reaparecen los elementos de simulación, y otra que cuenta con un soporte donde apoyar las placas solares para la realización de la soldadura.

La cámara de visión artificial 2-D con su correspondiente soporte.

Cuenta con los elementos de seguridad descritos en el capítulo 4.2.

Y otros elementos de “decoración” como las cortinas por donde pasan las cintas transportadoras y los controladores de los robots.

CAPÍTULO 5.1.4.1. Robot1.

Se trata de un **R-2000iC/125L** [23]. Un robot de 6 grados de libertad de brazo largo capaz de manipular cargas de hasta 125 kilogramos. Su brazo largo le aporta un gran alcance (3 metros aproximadamente). Todo ello le hace ser una buena alternativa para aplicaciones de carga pesada que requieran una repetibilidad excelente.



FIGURA 83. Robot1: R-2000iC/125L.

Este robot es utilizado para transportar las placas solares. Emplea la técnica de '*Line Tracking*' para tomarlas de la cinta transportadora de entrada y para dejarlas en la cinta transportadora de salida después de que se realicen las soldaduras.

CAPÍTULO 5.1.4.2. Robot2.

El segundo robot es un **ARC Mate 100iD** [24]. Este es un robot de soldadura por arco constituido por 6 ejes. Su diseño curvado le permite operar con facilidad en celdas de soldadura compactas. Tiene un alcance de alrededor de 1.5 metros y una buena repetibilidad.



FIGURA 84. Robot2: ARC Mate 100iD.

Este robot tiene un séptimo eje conformado por un raíl lineal de 2.5 metros de longitud. Este le permite desplazarse horizontalmente; evitando así las posibles colisiones con el tercer robot, puesto que el alcance del robot 2 y el del 3 no son lo suficientemente amplios como para que puedan cooperar en la zona de tratamiento. Mediante el raíl, el Robot2 se alejará de la mesa de soldadura en

los instantes en los que el Robot3 coloca las piezas de fijación. Este eje extendido se explicará más detalladamente en el capítulo 5.1.5.

Puesto que las soldaduras se efectúan en cuatro posiciones diferentes de la placa, se han creado cuatro '*User Frames*' – uno por cada pieza. De este modo, se desplaza el robot por los contornos de las piezas siguiendo trayectorias rectilíneas que simulan la soldadura de estas a la placa solar. En función de la pieza que haya colocado el Robot3 sobre la placa, se cambia al '*user frame*' correspondiente y se efectúa la soldadura. Otra ventaja que ha aportado la creación de los '*user frames*' es que se ha conseguido reutilizar código. Como las piezas tienen diferentes dimensiones, se ha programado una rutina de soldadura de piezas1 y otra de piezas 2 en lugar de programar cuatro secuencias de movimiento prácticamente iguales.

CAPÍTULO 5.1.4.3. Robot3.

El tercer robot es un **M-10iA/8L** [25]. Este robot tiene seis grados de libertad que tiene un alcance de 2 metros aproximadamente. Resulta ser idóneo para manipulación de materiales ligeros como pueden ser los elementos de fijación de este proceso productivo.



FIGURA 85. Robot3: M-10iA/8L.

Se conecta a una cámara de visión artificial 2D para poder discernir entre las piezas de fijación de la placa solar (ver capítulo 5.1.6.). Esta cámara no va montada sobre el robot, si no que está fija en un soporte. Como maneja un proceso de visión artificial, tiene asociado un '*user frame*' creado mediante el método de cuatro puntos sobre un patrón de puntos utilizado en la calibración de la cámara.

Se coordina con el Robot2 por medio del intercambio de señales para evitar colisiones y para comunicar qué tipo de pieza se ha colocado sobre la placa para su posterior soldadura.

CAPÍTULO 5.1.5. Configuración de un eje extendido.

Como ya se ha mencionado, al Robot2 se le ha añadido un séptimo grado de libertad basado en un raíl que le permite desplazarse longitudinalmente. Este proceso tiene varias etapas.

CAPÍTULO 5.1.5.1. Configuración del controlador.

Primeramente, el robot al que se desea agregar un eje adicional debe tener agregada la opción de software “*Extended Axis Control*”.

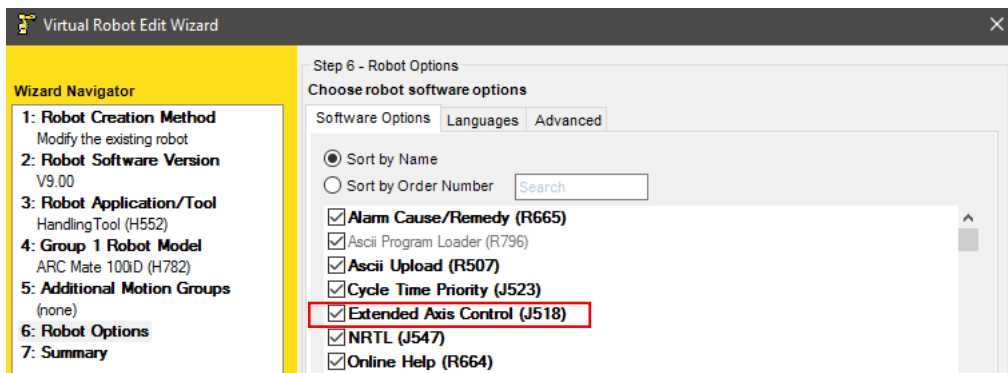


FIGURA 86. Workcell Creation Wizard: Extended Axis Control.

Una vez hecho esto, se debe configurar el controlador del robot. Para ello, se realizará un **reinicio controlado del controlador**. Para llevarlo a cabo, se sigue la siguiente ruta: **Barra de Herramientas > Robot > Restart Controller > Controlled Start**.



FIGURA 87. Teach Pendant de configuración de un controlador.

Se muestra una Teach Pendant (TP) especial en la que se modifica el número de ejes del robot para que posteriormente reconozca el eje agregado. Se pulsa sobre el botón **Menú** de la TP que muestra una serie de opciones, y se selecciona la opción **Maintenance**.

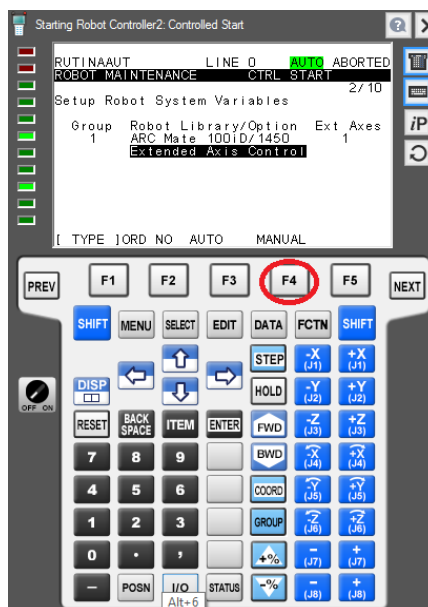


FIGURA 88. Maintenance y configuración del eje extendido.

Se posiciona el cursor sobre 'Extended Axis Control' como se muestra en la figura 88 y se pulsa F4 para configurarlo manualmente. Se selecciona el grupo que hace alusión al robot. Después, se selecciona la opción de añadir un eje (como máximo se pueden añadir 3) y se escribe el número de eje que se desea añadir – en la figura siguiente ya se tiene creado uno en E1, por lo que solo se podría seleccionar el 2 o el 3.

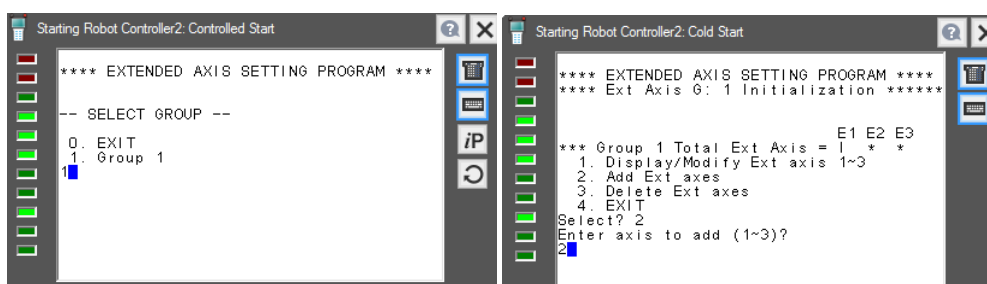


FIGURA 89. Eje extendido: Programa de configuración (I).

Posteriormente, se elige un método de introducción de los parámetros del séptimo motor y se pasa a introducir todos los datos según va pidiendo el programa de configuración. En este proyecto se utilizó el 'Enhanced method'.

Hecho esto, se vuelve a la pantalla mostrada a la derecha en la figura 89. Y se podrá mostrar todo lo configurado si se selecciona la opción 1: *Display/Modify Ext Axis*.

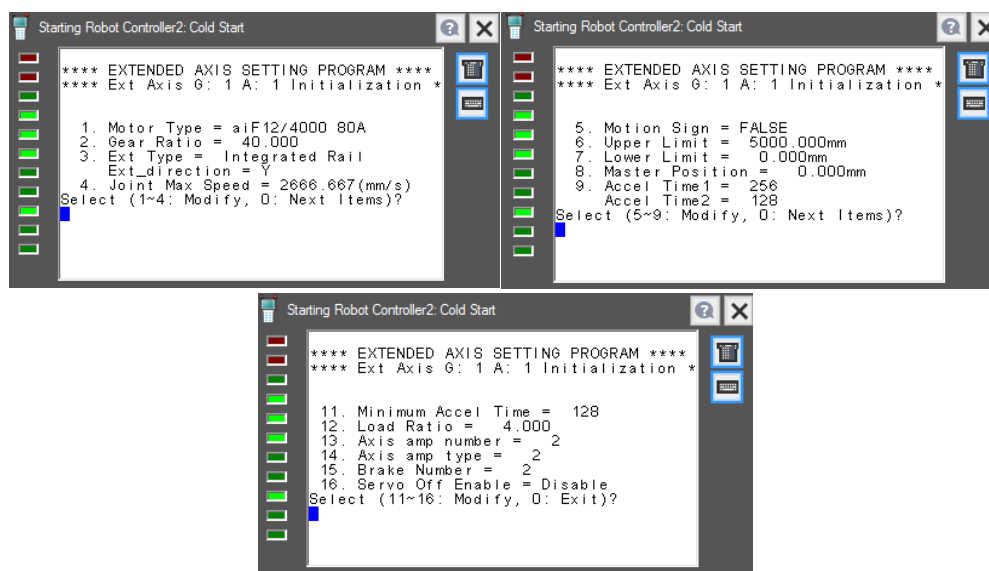


FIGURA 90. Eje extendido: Programa de configuración (II).

Algunas consideraciones importantes que se muestran en las figuras anteriores y que se tienen en cuenta en la creación del séptimo eje de simulación:

- El eje extendido agregado es de tipo raíl.
- La dirección del movimiento es la del eje Y del raíl (movimiento lateral).
- El raíl tendrá una longitud máxima de 5 metros.

Cuando se haya concluido la configuración del controlador se pulsa en el botón de funciones de la Teach Pendant y se selecciona la función 1: Inicio en frío.

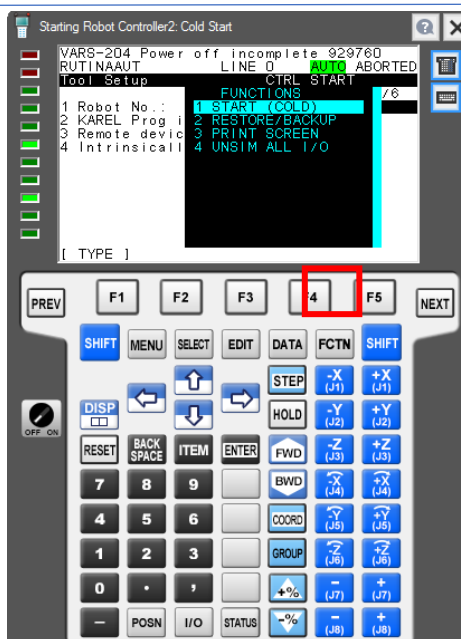


FIGURA 91. Fin de la configuración del controlador.

Esto reinicia el controlador del robot, con la configuración ya preparada.

Se puede comprobar si todo ha ido bien pulsando en el botón POSN de la Teach Pendant del controlador del robot al que se le ha agregado la configuración anterior.

Si todo fue correctamente, debería aparecer la configuración de todos los ejes del robot; incluido el que se acaba de configurar. En este caso, al tener siete ejes, debe mostrar la posición de los siete grados de libertad. En la figura siguiente se muestra la posición actual del robot. Los cuadros son editables y permiten un cómodo registro de posiciones como los que se muestran en las figuras 101, 103 y 104 del capítulo 5.1.5.3: Programación de un robot con eje extendido.

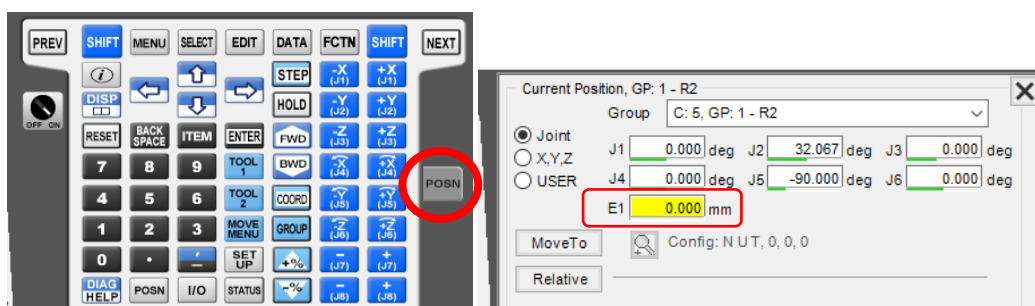


FIGURA 92. Comprobación de la configuración del eje extendido.

CAPÍTULO 5.1.5.2. Configuración del eje extendido en la estación.

A continuación, se detalla la configuración del raíl sobre el que se monta el Robot2 y que servirá como séptimo eje aportándole un desplazamiento lateral.

Primero se crea el propio raíl desde el apartado 'Machines' del árbol del proyecto. Haciendo 'click' derecho, se selecciona 'Add Machine' y después 'CAD Library'. De la biblioteca de elementos CAD, en el apartado 'Positioners', se pueden elegir diseños de raíles preparados para el montaje de robots.

En este caso se escogió el siguiente raíl de 2.5 metros de Fanuc cuya distancia es suficiente para evitar colisiones con el Robot3:

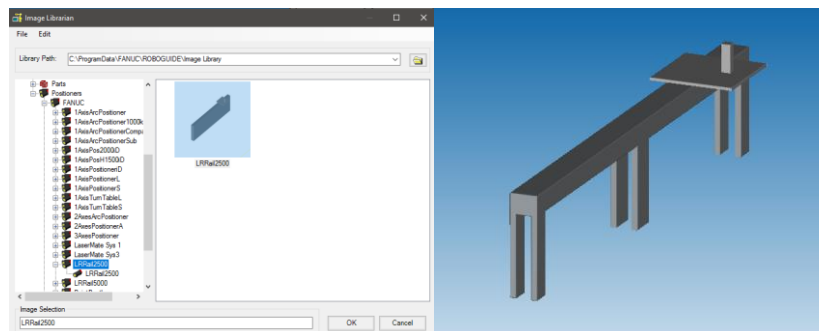


FIGURA 93. Raíl.

Tras incorporarlo a la estación, aparecerá en el apartado 'Machines' del árbol del proyecto. Consta del eje principal (señalado en azul en la figura 94) con un motor asociado (señalado en rojo).

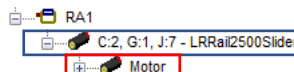


FIGURA 94. Raíl en el árbol del proyecto.

Si se selecciona el eje principal, aparece un cuadro de diálogo con diferentes pestañas. Únicamente interesa la pestaña 'Motion'. Aquí se selecciona el robot que se montará sobre el raíl, y el eje del robot al que se asocia ese raíl – en este caso el séptimo.

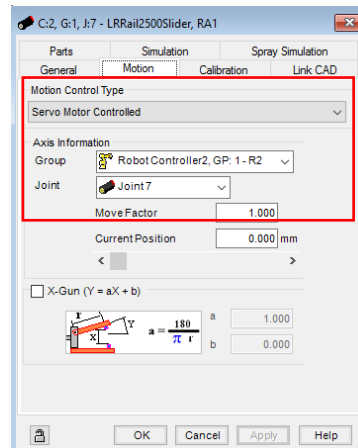


FIGURA 95. Asociación raíl – séptimo eje del robot.

Se aplican esta configuración, y se selecciona el motor asociado al raíl. Se hace ‘click’ derecho y se selecciona la opción **Attach Robot** y se selecciona el robot deseado – en este caso el Robot2.

Automáticamente, el robot se monta sobre la base desplazable del raíl. Y sobre el árbol del proyecto se muestra un nuevo ‘link’ que representa al robot:

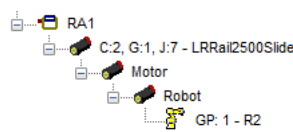


FIGURA 96. Raíl – Motor – Robot.

Este nuevo ‘link’ genera un nuevo movimiento a mayores: el de un hipotético octavo eje en este caso. Este movimiento se debe desactivar ya que solo se desea crear un eje extendido y no dos – y esto ya se ha ajustado (ver figura 95). Para ello, se hace doble ‘click’ en el apartado “Robot” mostrado en la figura 96.

Para ello, se accede a la pestaña ‘Motion’. Y se realiza el siguiente ajuste: no asociar el movimiento del octavo eje a ningún robot. Entonces en el apartado ‘Group’ se debe seleccionar la opción ‘None’ como se muestra en la figura 97.

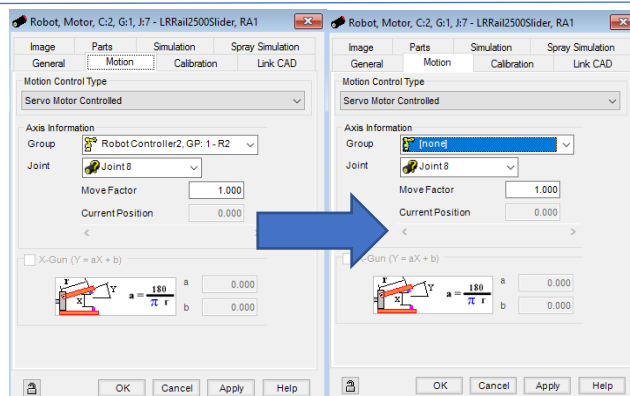


FIGURA 97. Desactivación del movimiento de la articulación 8.

Es posible que, al adjuntar el robot al raíl, se monte el robot de forma que al desplazarlo por el mismo; su TCP no acompañe el movimiento (error que me encontré y que aborta la ejecución de un programa). Esto se puede comprobar desplegando la Teach Pendant con el robot en cuestión seleccionado. Se pulsarán los botones (J7) + y (J7) - para realizar pruebas de desplazamiento de la base del robot por el raíl y para la posterior programación del robot.

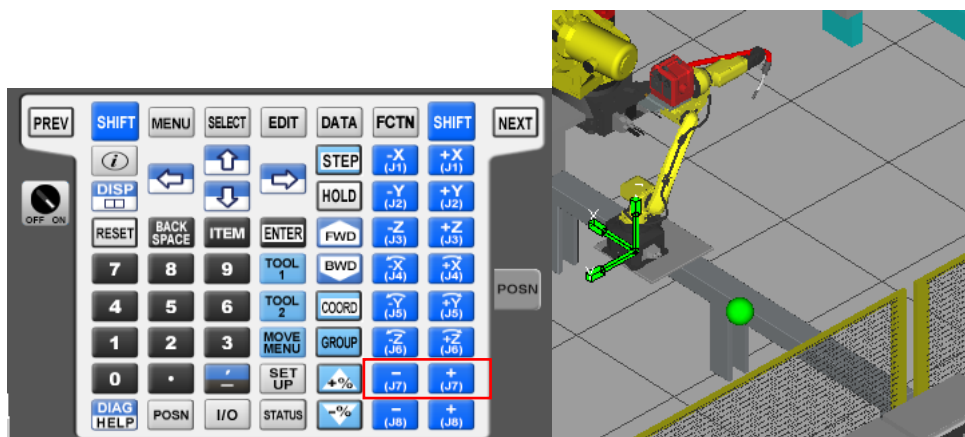


FIGURA 98. TCP (bola verde) no acompaña al movimiento del raíl.

Este error es debido a que, en la configuración del control de eje extendido, se establece una dirección de movimiento. En este caso la dirección Y como se puede ver en la primera imagen de la figura 86, apartado 3: EXT_direction. Y la dirección Y del séptimo motor no es coincidente con la dirección del movimiento sobre el raíl.

Para corregir este error, se debe modificar el eje de coordenadas mostrado en la figura 98. Se hace doble 'click' en el apartado "Motor" mostrado en la figura 96. Acceder a la pestaña 'Link CAD' del cuadro de diálogo y alinear el eje elegido con la dirección del movimiento del raíl (en este caso con el eje Y).

Así, si se realiza la comprobación del movimiento con la Teach Pendant, se puede ver que el TCP acompaña al movimiento del robot sobre el raíl.

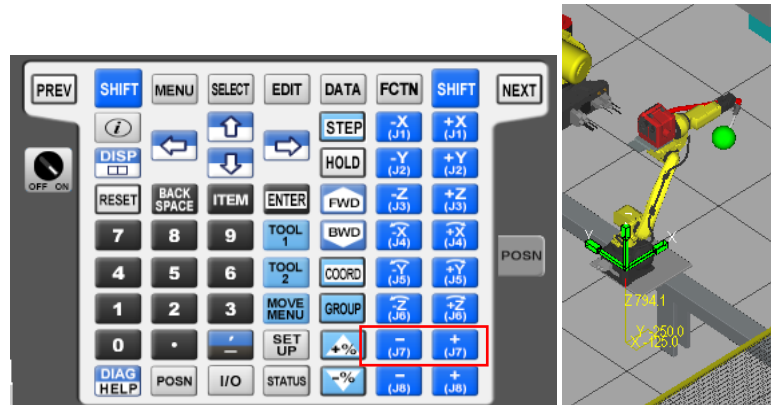


FIGURA 99. TCP (bola verde) acompaña al movimiento del raíl.

CAPÍTULO 5.1.5.3. Programación de un robot con eje extendido.

Efectuadas las configuraciones pertinentes, solo resta programar los movimientos del robot a lo largo del raíl.

La programación del robot se realiza de la misma manera que si no se tuviera el eje extendido (usando los comandos de movimiento señalados en el capítulo 5.1.1.1.). Esto es debido a que, desde el punto de vista del controlador del robot, el eje extendido está completamente integrado en el propio robot.

Por lo tanto, a la hora de programar comandos de movimientos, incluyan desplazamientos a lo largo del raíl o no, la mecánica no cambia: se desplaza el robot hasta el punto deseado y se registra la configuración de los ejes del robot en dicho punto.

La única diferencia con respecto a un robot que no tenga un eje extendido es la propia configuración de los ejes del robot registrada en memoria. A continuación se muestra un ejemplo:

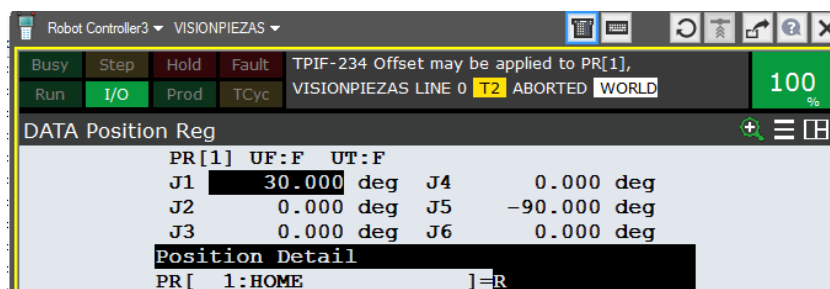


FIGURA 100. Registro de posición de un robot sin eje extendido.

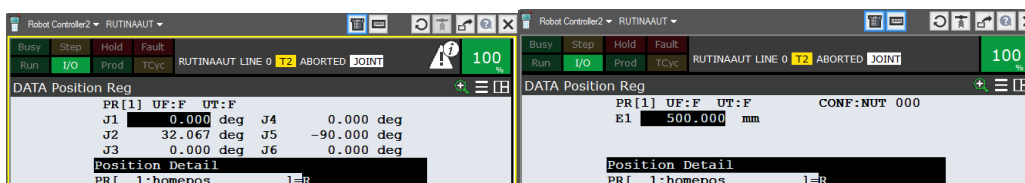


FIGURA 101. Registro de posición de un robot con eje extendido.

La figura 100 representa la configuración de los ejes del Robot3 en su posición 'Home' – un robot de 6 grados de libertad sin eje extendido. En consecuencia, al registrar un punto, se guarda la configuración de los seis ejes que conforman el robot.

Mientras que en la figura 101 se representa la configuración de los ejes del Robot2 en su posición 'Home' – un robot de 6 grados de libertad con el eje extendido configurado en los capítulos 5.1.5.1. y 5.1.5.2. En total 7 grados de libertad. En consecuencia, al registrar un punto, se guarda la configuración de los siete ejes.

De esta manera se puede realizar una traslación por el raíl a la vez que una rotación de cualquiera de los otros seis ejes del robot con un único comando de movimiento. Se muestra un ejemplo de programación a continuación:

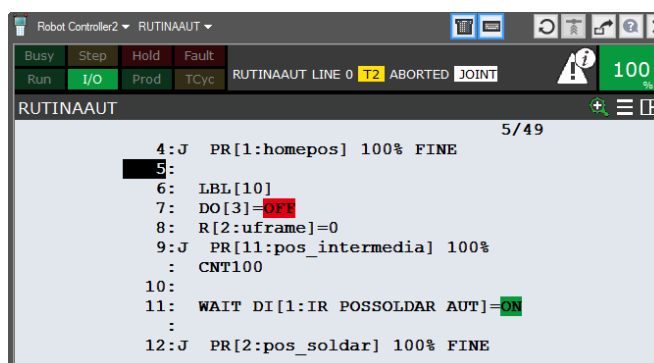


FIGURA 102. Ejemplo de programación de un robot con eje extendido.

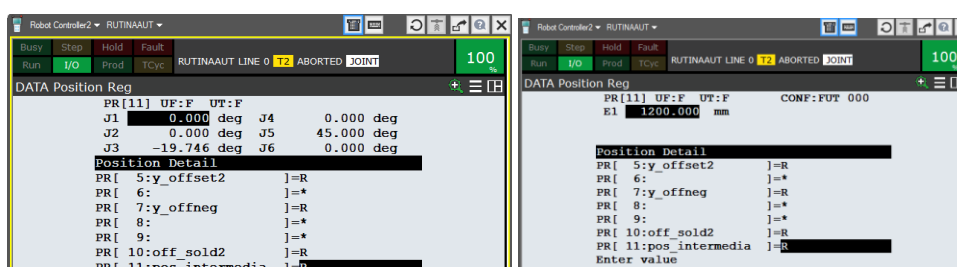


FIGURA 103. Pos_intermedia.

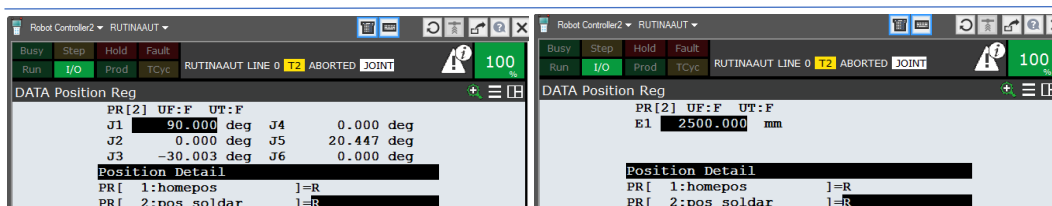


FIGURA 104. Pos_soldar.

Si se analiza el ejemplo de programación de la figura 102, se observa que sólo con un comando de movimiento se puede componer una traslación por el raíl junto con una o varias rotaciones del resto de los ejes. Se hace la suposición de que el robot está en *PR1: homepos*.

Línea 4. El robot no se mueve pues se ha supuesto que ya está en dicha posición.

Línea 9. Si se comparan la figura 101 y la 103, se aprecia que para ir de *PR1:homepos* a *PR11: pos_intermedia*; se debe trasladar el robot 0.7 metros a lo largo del eje extendido. Además las articulaciones J2, J3 y J5 deben rotar. Todo ello realizado con un único comando de movimiento cómodo.

Línea 12. Cuando se autorice, el robot se desplaza con un movimiento cómodo de la *PR11: pos_intermedia* a la *PR2: pos_soldar*. Si se observan las figuras 103 y 104, se puede entender que este desplazamiento se compone de una traslación de 1.3 metros a lo largo del raíl y de rotaciones de las articulaciones J1, J3 y J5.

Como conclusión, se entiende que la posición del motor que mueve el raíl es controlada por el mismo controlador que maneja el robot. En consecuencia, no es necesario programar dicho motor aparte; este ya forma parte del propio robot.

CAPÍTULO 5.1.6. Configuración de un proceso de visión artificial.

A continuación, se detallan los pasos a seguir para configurar un proceso de reconocimiento de objetos mediante visión artificial como el implementado en el Robot3 de la estación robotizada de simulación. En este caso se trata de un proceso de visión 2D; dónde la cámara está fijada en un soporte.

CAPÍTULO 5.1.6.1. Configuración previa del robot.

El primer paso es configurar el robot que integra el proceso de visión artificial. El robot debe incluir las opciones de software '*iRVision*'. De esta forma, podrá realizar procesos de visión artificial. En este caso, como el proceso es de visión 2D, se ha añadido la opción '*iRVision 2D Pkg*'.

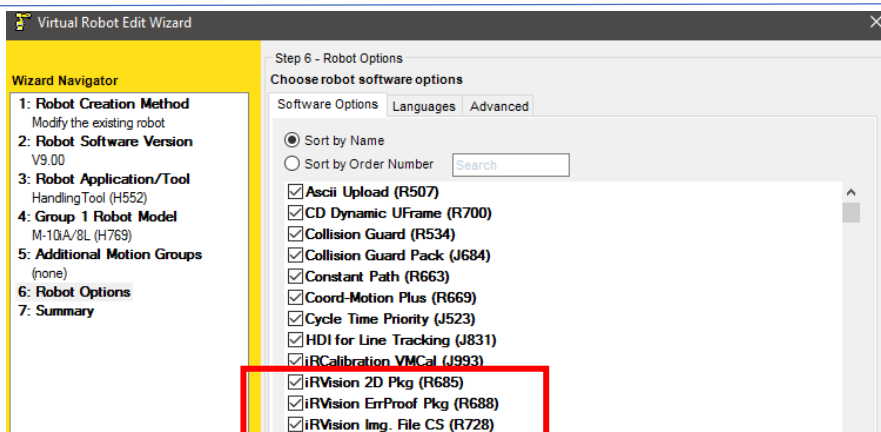


FIGURA 105. Opciones de software de iRVision.

Con esto seleccionado, en **Árbol del proyecto**>**Robot Controllers**>**Robot con visión** debería aparecer el icono de 'Vision':

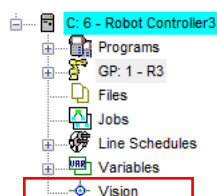


FIGURA 106. Icono de visión.

Haciendo 'click' derecho sobre este icono, se abre un cuadro con opciones. Para habilitar la simulación de visión artificial, se debe seleccionar en la opción 'Enable Vision Simulation' y reiniciar el controlador del robot.

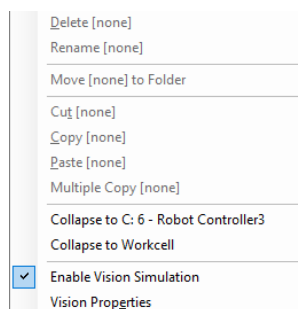


FIGURA 107. Simulación de visión artificial habilitada.

CAPÍTULO 5.1.6.2. Cámara 2D de visión artificial.

El siguiente paso es incluir la cámara 2D de visión artificial en la estación de simulación. Para incluirla, se hace 'click' derecho en el apartado 'Sensor Units' del árbol del proyecto y se sigue la siguiente ruta:

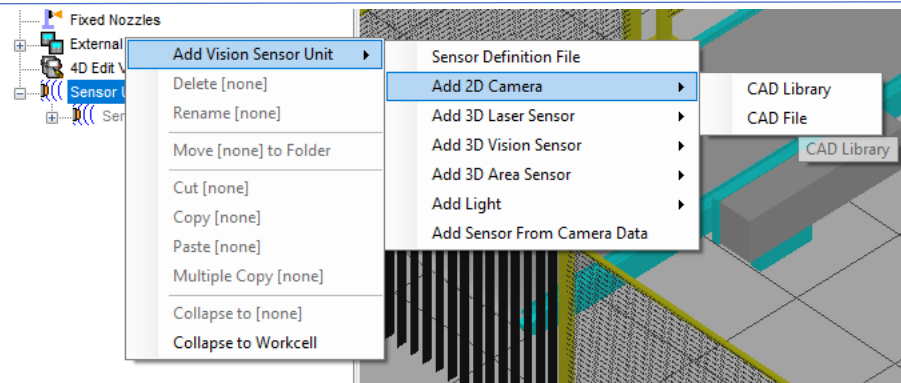


FIGURA 108. Agregar cámara 2D de visión artificial.

En este punto se puede agregar un archivo de la biblioteca de HandlingPro u otro archivo compatible del que se disponga.

En este caso, se seleccionó una cámara de las de la biblioteca de HandlingPro que no distingue colores. Se seleccionó esta porque para discernir entre los dos tipos de piezas de fijación de la placa solar, el criterio de discriminación es la forma de dichas piezas y no su color.

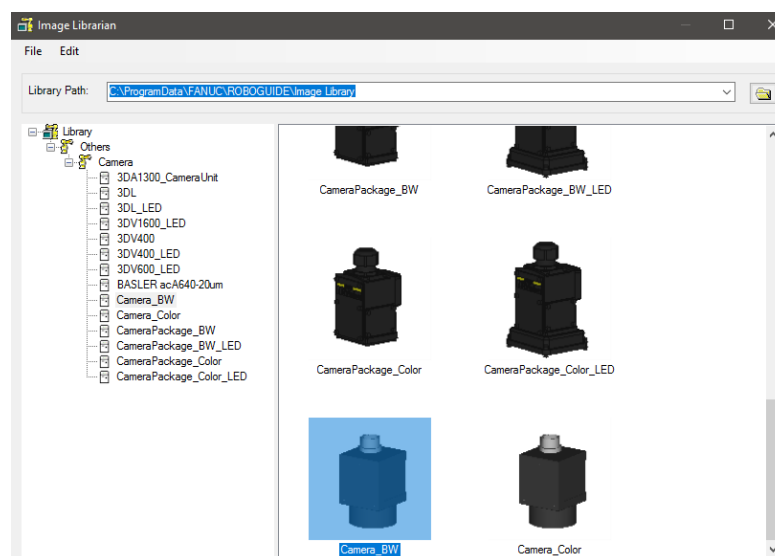


FIGURA 109. Camera_BW.

Incorporada la cámara, en **Árbol del proyecto**>**Sensor Units** se muestra lo siguiente:



FIGURA 110. Cámara agregada al árbol del proyecto.

Si se hace doble 'click' en C1 mostrado en la imagen anterior, se abre un cuadro de diálogo con múltiples pestañas.

Tanto en la pestaña 'Setting' como en la pestaña 'View' se pueden modificar ciertos parámetros de la cámara para que haga mayor o menor 'zoom', es decir, que abarque más o menos espacio a la hora de visualizar el proceso. Se deja en manos del programador la configuración de estos parámetros según sus necesidades.

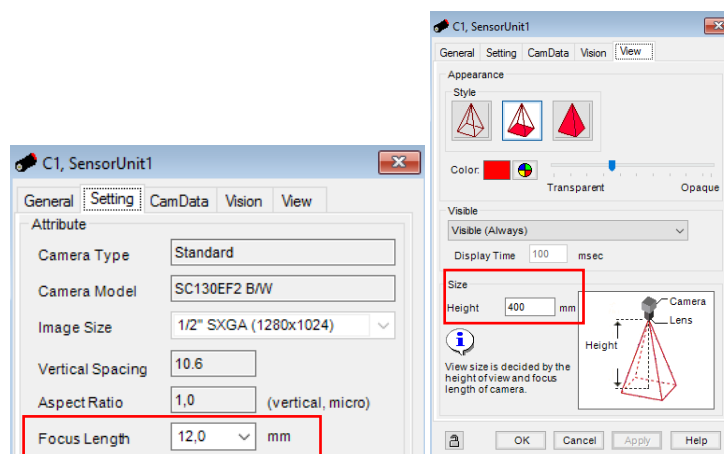


FIGURA 111. Parámetros de la cámara de visión artificial.

CAPÍTULO 5.1.6.3. Asociación robot-cámara.

Lo siguiente es conectar la cámara de visión al robot. Esto se consigue en el cuadro de diálogo que se abre haciendo doble 'click' sobre el icono 'Vision' mostrado en la figura 106.

En la pestaña 'General' se asocia la cámara al robot como se muestra a continuación:

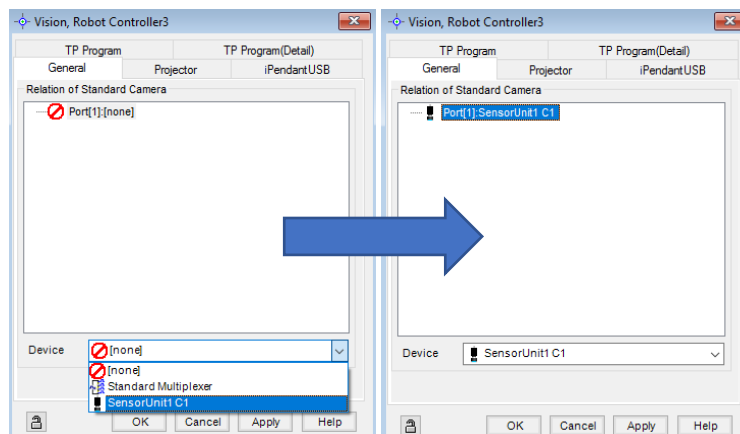


FIGURA 112. Asociación Robot-Cámara.

CAPÍTULO 5.1.6.4. Calibración de la cámara.

Cuando la cámara esté conectada al robot, se puede realizar la calibración de la cámara en 'iRVision Vision Setup' (ver figura 32). Para la calibración se necesita un patrón de puntos cuya distancia entre puntos sea conocida como ya se ha mencionado anteriormente en este documento; y crear un '2D Camera Data' donde realizar la calibración como tal.

Paso 1: Camera Setup. En este paso se selecciona la cámara de la estación y el tipo de calibración.

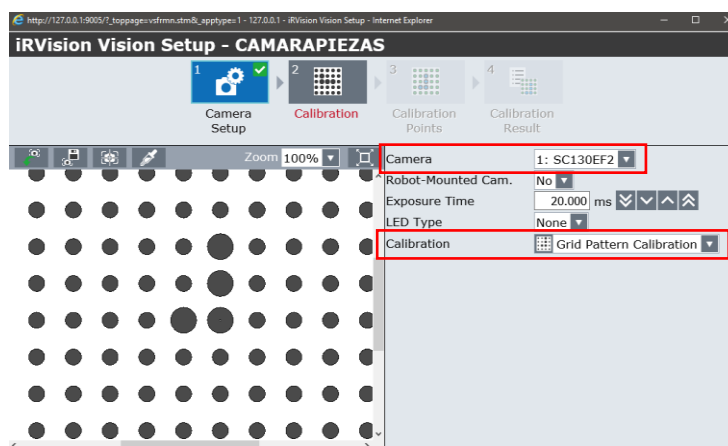


FIGURA 113. 2D Camera Data: Camera Setup.

Paso 2: Calibration. En este paso se selecciona el 'user frame' de visión creado, el espacio entre puntos del patrón de calibración, la proyección del plano de la cámara, y se estipulan unos márgenes de calibración al pulsar en 'find' cuando esté todo listo ('set').

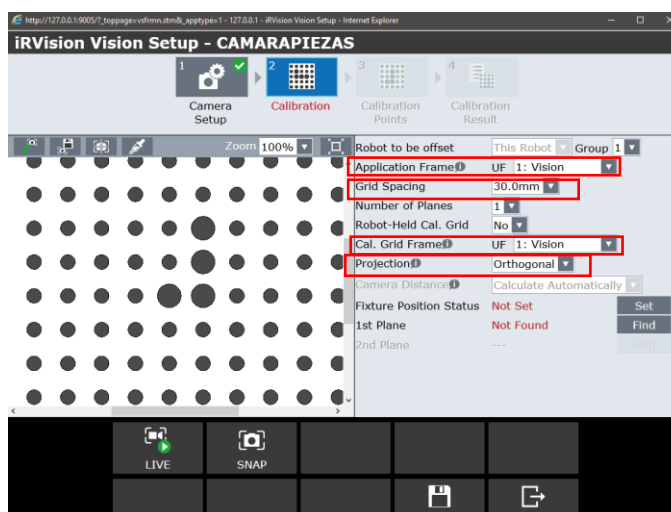


FIGURA 114. 2D Camera Data: Calibration.

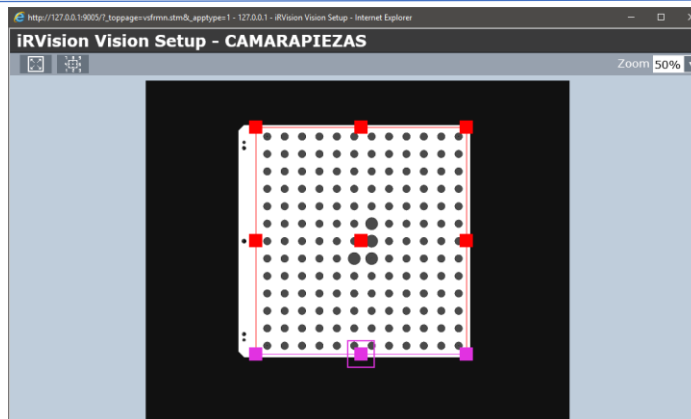


FIGURA 115. 2D Camera Data: Márgenes de Calibración.

Cuando se pulse OK al establecer los márgenes de calibración, se realiza esta automáticamente. El resultado satisfactorio de la calibración se ve así:

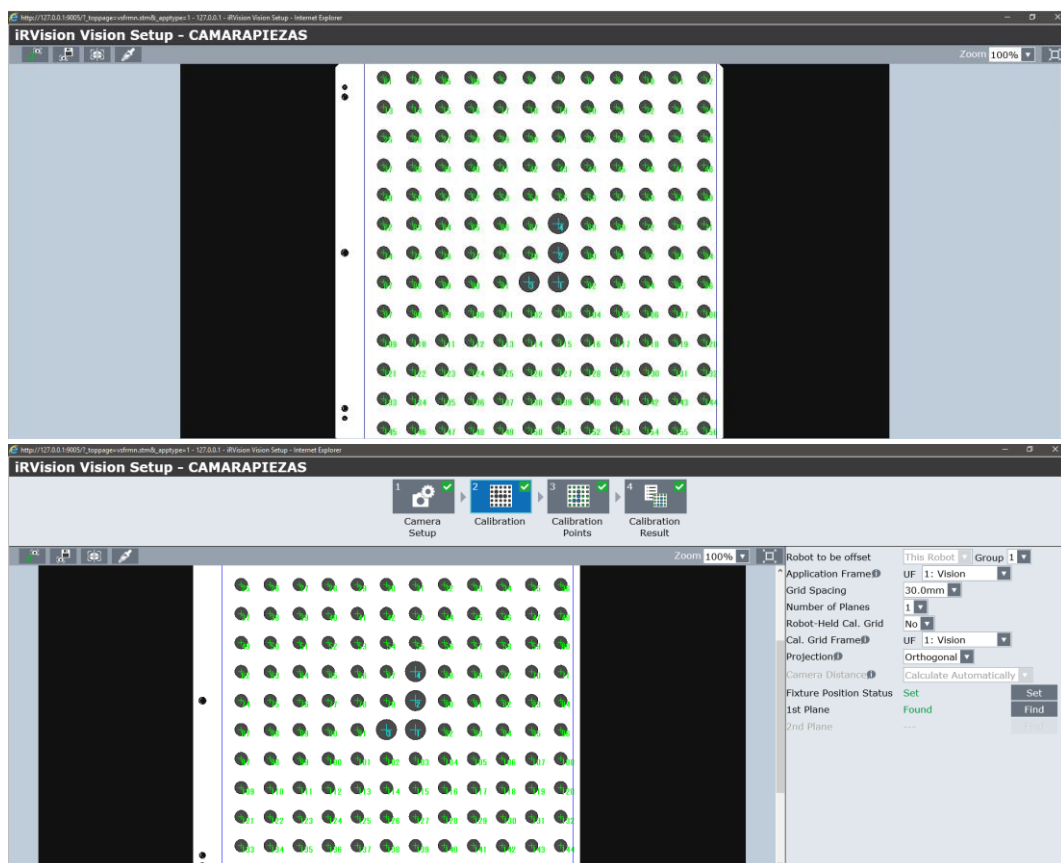


FIGURA 116. 2D Camera Data: Resultados de la calibración.

CAPÍTULO 5.1.6.5. Proceso de visión artificial.

Cuando la cámara esté calibrada, se debe enseñar al robot como discernir las piezas de fijación de la placa.

Esto se consigue creando un 'Vision Process Tools'. Existen varios tipos, pero como se ha comentado anteriormente, la cámara es de dos dimensiones que va fija en un soporte. En consecuencia, la cámara solo proporciona una vista o perspectiva. Así pues, el tipo de 'Vision Process Tools' a crear debe ser un '2-D Single View Vision Process'.

El entorno de configuración es el siguiente:

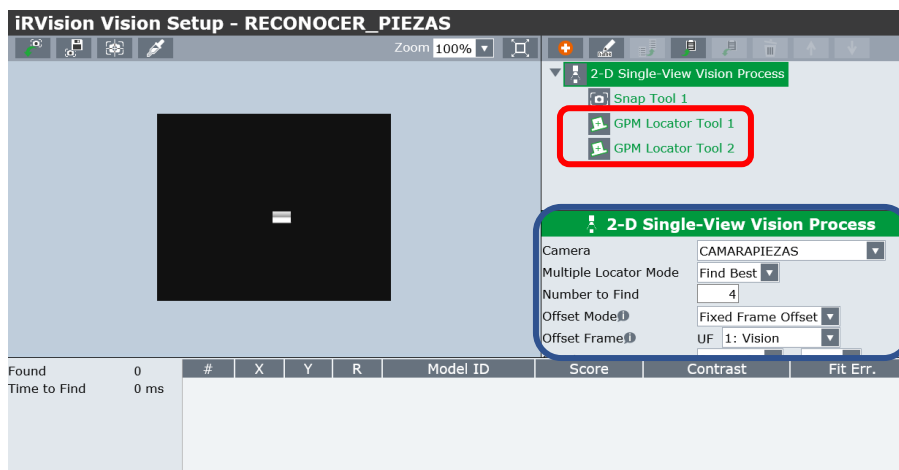


FIGURA 117. 2-D Single View Vision Process.

Los parámetros del proceso se editan en la zona señalada en azul en la figura anterior. A continuación, se muestra la configuración utilizada en el proyecto.

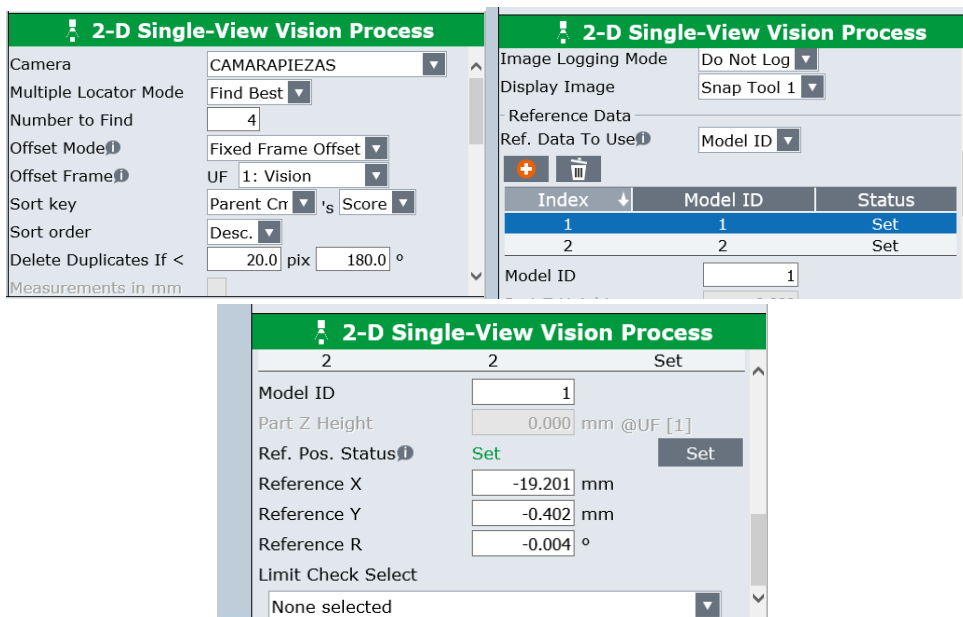


FIGURA 118. Parámetros del proceso de visión.

Los parámetros más relevantes de la figura 116, son los denominados 'Reference Data to Use' en los que se ha utilizado 'Model ID' (ver segunda imagen de la figura 116).

Estos identificadores de los modelos son relativos a los dos tipos de piezas de fijación de las que se dispone. La configuración de estos parámetros es sencilla: se pulsa sobre el icono con el signo (+) que se ve en la segunda imagen de la figura 116 y se le asigna un índice igual al número de modelo. Después, se desplaza el TCP del robot al lugar donde está el modelo de la pieza en la célula de simulación y se guarda esa referencia de posición pulsando en el botón 'Set' de la tercera imagen de la figura 116. Esta referencia será muy útil a la hora de reconocer piezas del mismo tipo que estén en distintas posiciones – ya que, en la estación, la posición de reaparición de las piezas en cada ciclo de trabajo ha sido programada para ser completamente aleatoria para incrementar la necesidad de una visión artificial robusta.

Posteriormente, cuando se realiza la identificación de las piezas en pleno funcionamiento de la estación, el propio robot las categoriza entre piezas del tipo 1 o del 2 según el identificador del modelo.

Una vez configurado el proceso de visión, se debe enseñar al robot cómo son las piezas de fijación por medio de la configuración de un 'GPM Locator Tool' por cada tipo de pieza – ver figura 117, señalización roja.

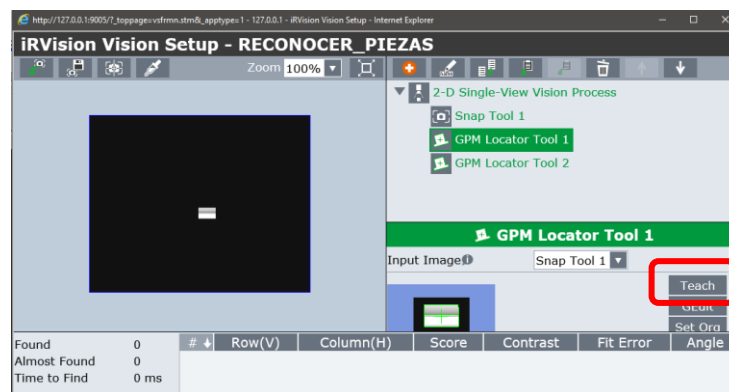


FIGURA 119. Editor del GPM Locator Tool 1.

Haciendo 'click' sobre el botón 'Teach' se enseña al robot la forma de la pieza al detalle:

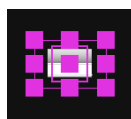


FIGURA 120. Enseñanza de la pieza tipo 1.

Tras la enseñanza de la pieza se pulsa en 'Snap+Find' y se identifica la pieza:

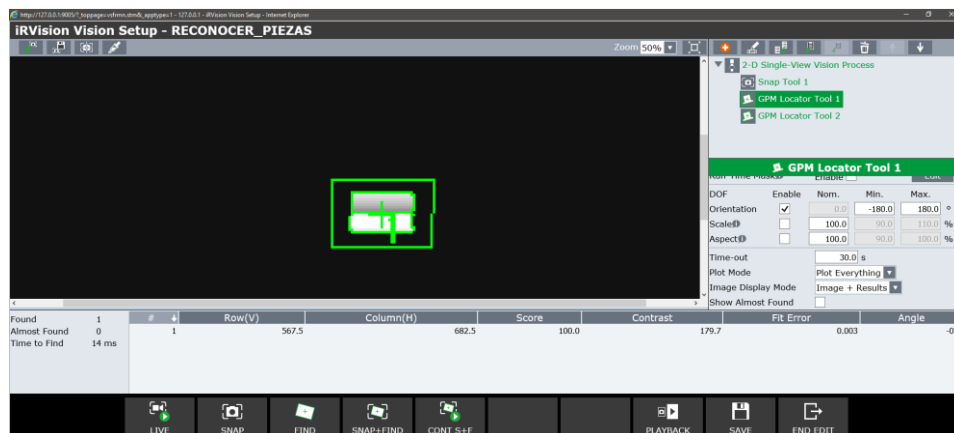


FIGURA 121. Identificación tras enseñanza.

Este proceso se repite para el otro tipo de piezas. Y cuando los dos tipos se hayan enseñado, el robot puede trabajar de forma autónoma. Seguidamente, se muestra un ejemplo de reconocimiento en pleno funcionamiento:

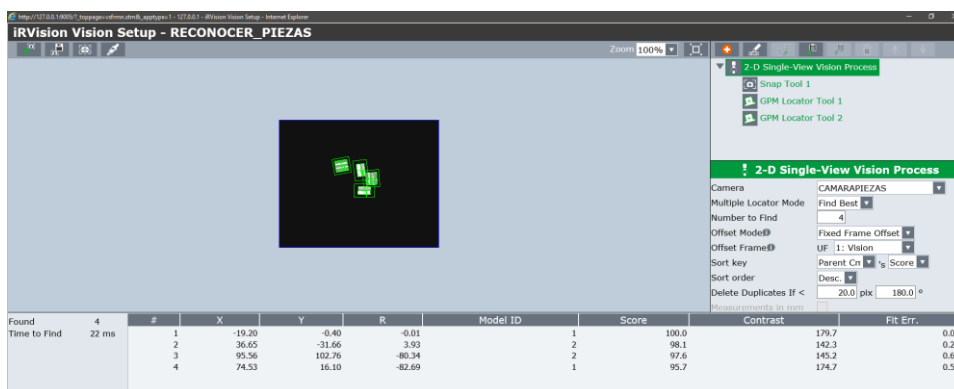


FIGURA 122. Ejemplo de identificación de piezas de fijación.

Como se aprecia en la figura 122, la identificación da cierta información relevante referente a cada pieza:

- Referencias X, Y, R donde debe ir el robot a recogerlas.
- Model ID.
- Puntuación en porcentaje de la identificación: a más se parezca al modelo enseñado, mejor puntuación.

CAPÍTULO 5.1.6.6. Programación de visión artificial en Teach Pendant.

El capítulo anterior se centra en la enseñanza del reconocimiento de las piezas de fijación. Sin embargo, para que el robot realice correctamente la tarea

recogida de piezas tras dicha identificación, se debe programar una rutina en la Teach Pendant.

Por suerte, al haber incluido la opción de software ‘iRVision 2D Pkg’, el robot cuenta con instrucciones relativas a la programación de procesos de visión artificial:

VISION 1/2	VISION 2/2
1 RUN_FIND	1 VR[].MODELID
2 GET_OFFSET	2 VR[].MES[]
3 GET_PASSFAIL	3 VR[].FOUND_POS[]
4 GET_NFOUND	4 VR[].OFFSET
5 SET_REFERENCE	5 VR[].ENC
6 OVERRIDE	6 GET_READING
7 CAMERA_CALIB	7

FIGURA 123. Instrucciones de programación de iRVision.

Con varias de estas se han realizado el programa de identificación de piezas de fijación. A continuación, se muestra y explica la parte específica de la visión del programa en cuestión:

```

VISIONPIEZAS
2/37
1: UTOOL_NUM=2
2: UFRAME_NUM=1
3: OVERRIDE=100%
4:
5: VISION RUN_FIND
   : 'RECONOCER_PIEZAS'
6: VISION GET_NFOUND
   : 'RECONOCER_PIEZAS' R[10]
7:
8: R[20:PiezasQuedan]=R[10]
9: IF R[10]=0,JMP LBL[100]

17/37
10: LBL[10]
11:
12: VISION GET_OFFSET
   : 'RECONOCER_PIEZAS' VR[R[20]]
   : JMP LBL[100]
13: R[10]=R[10]-1
14: IF R[10]<>0,JMP LBL[10]
15:
16: LBL[50]
17: DO[1]=OFF
18: R[5]=VR[R[20]].MODELID
  
```

FIGURA 124. Programa de visión artificial.

Línea 1. Se especifica la herramienta que utiliza el robot.

Línea 2. Se especifica el ‘user frame’ de visión que ha sido creado.

Línea 3. La velocidad de ejecución será máxima – 100%.



Línea 5. Se ejecuta el proceso de reconocimiento de piezas explicado en el capítulo 5.1.6.5.

Línea 6. El número total de piezas identificadas en el proceso de visión se almacena en un registro.

Línea 8. Se duplica el registro 10 de la línea 6 para un uso posterior en la línea 12 y en la 18.

Línea 9. Si no hay piezas a identificar se realiza un salto a la etiqueta 100.

Línea 10. Etiqueta que marca el inicio de un equivalente a bucle do while de C++.

Línea 12. Se guarda el offset de una pieza reconocida en un registro de visión – puesto que, en cada ejecución, las piezas reaparecen en un lugar diferente.

Línea 13. Guardado el offset, se resta 1 de las piezas que quedan.

Línea 14. Condición del bucle. Mientras haya piezas cuya posición ha de ser guardada, se realiza el bucle que comprende las líneas 10-14. Cuando no haya más piezas se pasa a la siguiente línea.

Línea 18. Se guarda el '*model ID*' de una pieza en un registro.

Las líneas siguientes hasta la última implementan el control del flujo del programa, y la toma de decisiones en función de la información recopilada en las líneas anteriormente explicadas (llamadas a rutinas y activación de señales de coordinación con otros robots).

Dentro de esas rutinas, las más importantes de cara a la visión artificial es la de tomar un objeto que se identifica – ya que la posición donde haya aparecido es aleatoria. En consecuencia, en el proceso de visión artificial se registra un desplazamiento respecto del origen registrado ('*offset*'). Un ejemplo es el siguiente:


```
PICKUP1
1/9
1: UFRAME_NUM=1
2:L PR[3:REF_PIEZA1] 2000mm/sec
: CNT25 Tool_Offset, PR[2:offsetZ]
: VOFFSET, VR[R[20]]
3:L PR[3:REF_PIEZA1] 1000mm/sec
: FINE VOFFSET, VR[R[20]]
4: CALL PICKUP_SIM1
5:L PR[3:REF_PIEZA1] 1000mm/sec
: CNT25 Tool_Offset, PR[2:offsetZ]
: VOFFSET, VR[R[20]]
6:J @PR[1:HOME] 100% FINE
```

FIGURA 125. Programa de tomar objeto usando visión artificial.

Línea 1. Aunque sea redundante, se vuelve a especificar el ‘user frame’ de visión creado.

Línea 2. Se envía al TCP del robot en un movimiento lineal prácticamente a la posición de referencia de la pieza con dos ‘offsets’ diferentes: uno vertical para realizar una aproximación, y el propio de la pieza calculado en el proceso de visión artificial. Así se sitúa la pinza del robot justo encima de la pieza.

Línea 3. En ese punto se desciende linealmente para agarrar la pieza con la pinza del robot.

Línea 4. Se produce la animación de agarre del elemento.

Línea 5. Se asciende linealmente de nuevo con la pinza agarrando el objeto hasta el punto de la línea 2.

Línea 6. El robot vuelve a la posición de ‘HOME’ con el objeto tomado.

CAPÍTULO 5.2. Desarrollo en TIA Portal.

En TIA Portal se ha creado una interfaz hombre-máquina para la gestión de la planta robotizada a nivel de usuario junto con un PLC que controla la seguridad de dicha planta y el correcto uso de las funcionalidades de la interfaz.

CAPÍTULO 5.2.1. Interfaz hombre-máquina (HMI).

Se ha utilizado una TP700 Comfort [26] de 7 pulgadas de la marca Siemens con conexión Profinet. La pantalla real es totalmente táctil – no tiene ningún botón. Aunque en este proyecto solo se emplea una copia simulada que consta de varias secciones (las denominadas “imágenes” en TIA Portal) con diferentes usos.

CAPÍTULO 5.2.1.1. Plantilla e Imágenes.

Todas las imágenes de la interfaz están montadas sobre una plantilla para ahorrar trabajo en la personalización de cada imagen. Es decir, todas las imágenes tienen en común los elementos que aparecen en la plantilla y los eventos o animaciones asociados a dichos elementos:

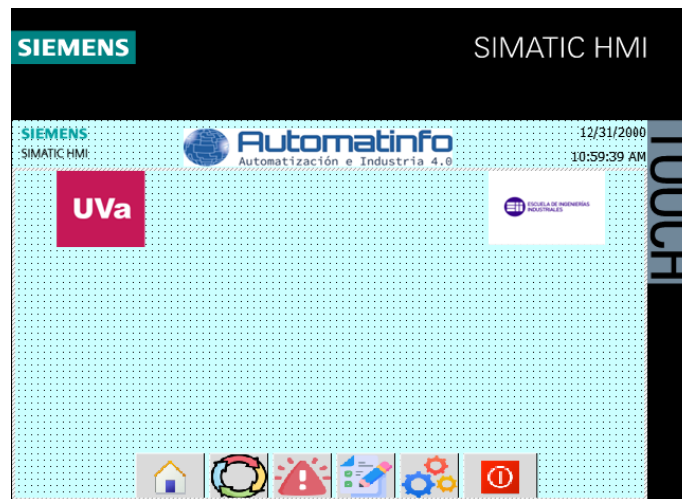


FIGURA 126. HMI: Plantilla.

La plantilla incluye varios 'logos': los propios de la pantalla de siemens, el de la Universidad de Valladolid, el de la Escuela de Ingenierías Industriales de la Universidad de Valladolid, y el de la empresa donde he realizado las prácticas – Automatinfo S.L.

En la esquina derecha se tiene la fecha y hora actuales que se actualizan automáticamente en el momento en que se ejecuta la simulación de la interfaz. Según el idioma seleccionado en la interfaz, el formato de la fecha varía (DD/MM/AAAA o MM/DD/AAAA).

Por último, en la zona inferior, están localizados 6 “botones virtuales” con diferentes funciones. De izquierda a derecha:

- Botón de “**HOME**”: al presionar este botón, el contenido de la pantalla cambia al propio de la imagen “HOME”.
- Botón de “**Ejecución Continua**”: al presionar este botón, el contenido de la pantalla cambia al propio de la imagen “CONTINUA”.
- Botón de “**Errores y Señalización**”: al presionar este botón, el contenido de la pantalla cambia al propio de la imagen “ERRORES Y SEÑALES”.
- Botón de “**Ejecución Individual**”: al presionar este botón, el contenido de la pantalla cambia al propio de la imagen “INDIVIDUAL”.

- Botón de “**Ajustes**”: al presionar este botón, el contenido de la pantalla cambia al propio de la imagen “AJUSTES”.
- Botón de “**Apagado**”: al presionar este botón, concluye el ‘runtime’ de la interfaz; es decir, se termina la simulación.



FIGURA 127. HMI: Imagen HOME.

La imagen “HOME” es la de bienvenida a la interfaz. Incluye textos y dos señalizadores donde que indican al usuario el estado de la estación y de la interfaz con respecto a la estación. Cuando estos dos indicadores parpadeen con una luz verde, entonces se podrá comenzar a gestionar la celda robotizada desde la HMI.

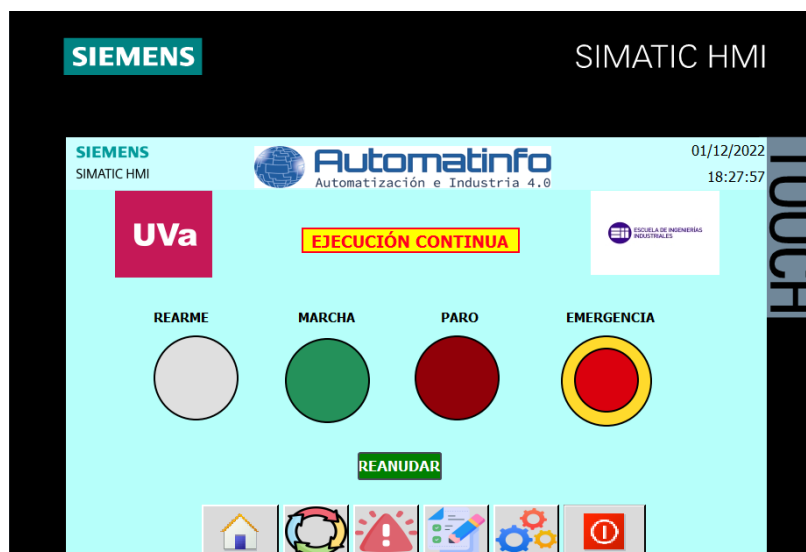


FIGURA 128. HMI: Imagen CONTINUA.

La imagen “CONTINUA” es la perteneciente a la gestión del modo automático de la planta. Consta de 5 botones:

- REARME: ha de pulsarse tras parada de emergencia.
- MARCHA: empleado para cambiar de modo manual a automático.
- PARO: realiza una parada controlada del proceso.
- EMERGENCIA: simula una seta de emergencia.
- REANUDAR: normalmente permanece invisible, pero cuando se ha producido una parada controlada, aparece en la imagen. Reanuda la simulación en el punto donde se había producido la parada controlada.



FIGURA 129. HMI: Imagen ERRORES Y SEÑALES.

La imagen “ERRORES Y SEÑALES” es meramente informativa. Indica tanto los posibles errores que se puedan dar en la planta como información acerca de la estación.

Errores:

- **Visión: Num Piezas < 4.** Podría darse que el resultado del proceso de visión artificial fuera erróneo y no reconociera las 4 piezas de fijación.
- **Barrera de Seguridad.** Si algún operario cruzara la cortina laser de la estación, se activa este error.
- **Estación con Faltas.** Generalmente ocurre al conectar la estación al servidor OPC. Algunas variables cruciales (IMSTP, SFSPD, Enable) podrían no cargan correctamente.

- **Parada de Emergencia.** Se enciende al pulsar cualquier seta de emergencia. Además, se muestra un icono en forma de señal amarilla con un texto que dice: Rearmar!
- **2 o más Robot Seleccionados.** Se da cuando en modo manual se tienen seleccionados más de un robot.
- **Error en la Selección de Comandos.** Salta cuando se produce cualquier contradicción en la selección de órdenes, por ejemplo: pulsar marcha y paro a la vez.

Señalizaciones:

- **Programa en Ejecución.** Luce únicamente si los programas de los robots se están ejecutando – ya que la simulación de la estación puede haberse iniciado sin haber arrancado los robots.
- **Programa Parado.** Brilla si se ha realizado una pausa controlada del programa.
- **Modo Continuo.** La estación funciona en modo automático.
- **Modo Individual.** La estación funciona en modo manual.



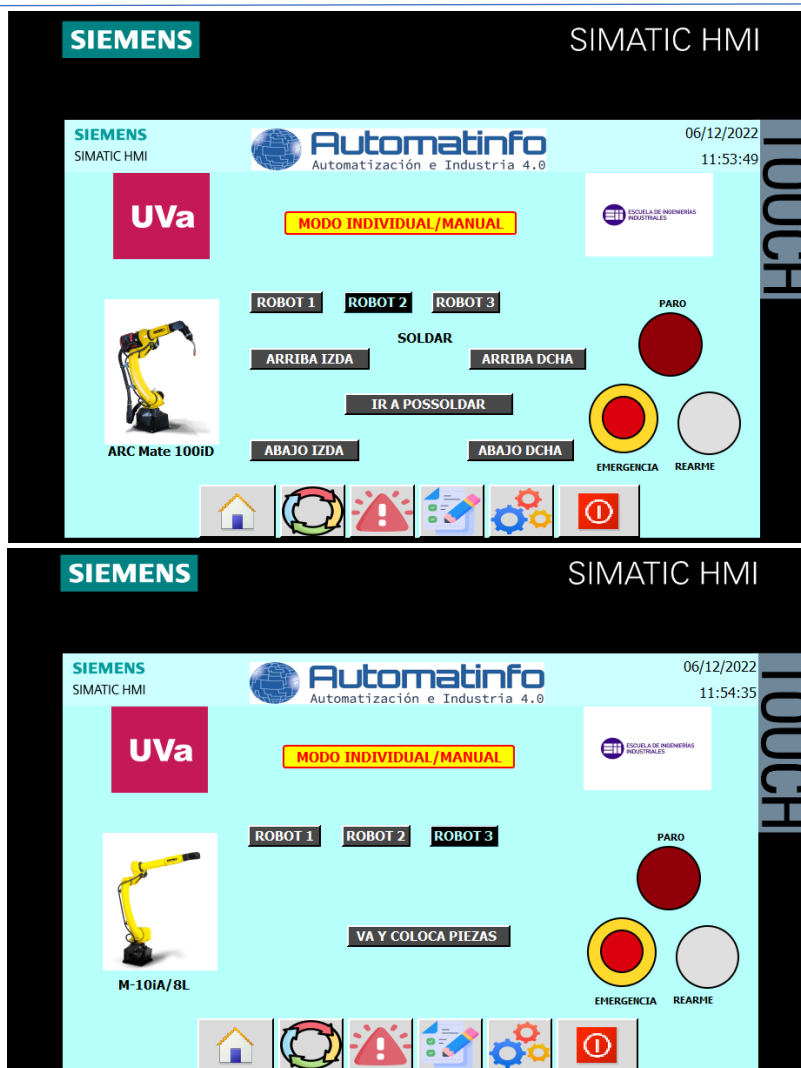


FIGURA 130. HMI: Imagen INDIVIDUAL.

La imagen “INDIVIDUAL” es la más compleja ya que, en ella, se prueban los posibles movimientos que puede realizar cada robot. Se ha programado la visibilidad de los elementos de esta imagen para que, en función del robot seleccionado, se muestren las opciones correspondientes y la imagen que muestra la apariencia del robot junto con su nombre. Además, cuenta con los botones para realizar pausas (controlada y de emergencia), rearmar la estación y reanudar la ejecución de un movimiento tras pausa controlada.

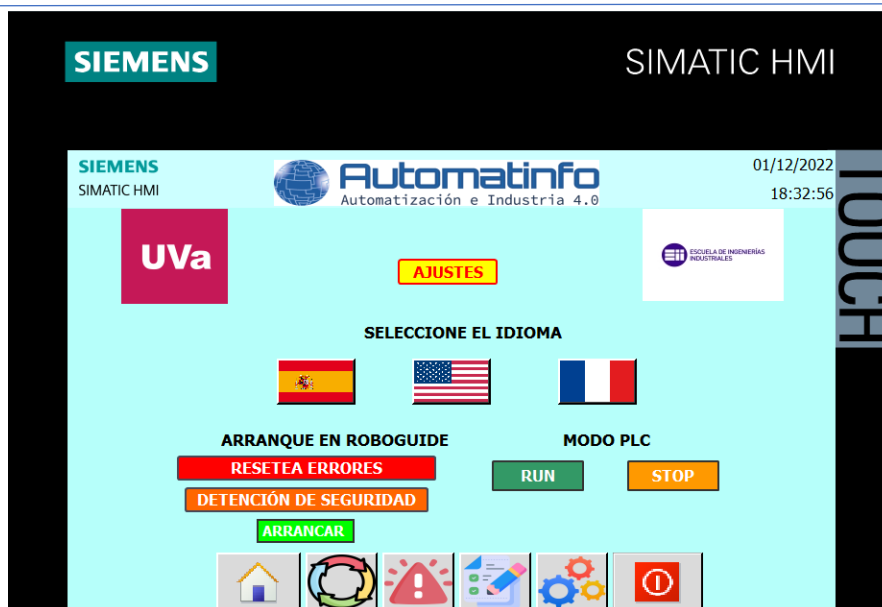


FIGURA 131. HMI: Imagen AJUSTES.

Por último, la imagen “Ajustes” es la que permite el cambio de idioma entre los tres disponibles: español, inglés y francés. Adicionalmente, permite modificar el estado del PLC: RUN (en ejecución) o STOP (parado). Por último, desde esta imagen se realiza el arranque remoto de la estación en HandlingPro de RoboGuide.

Para arrancar en remoto, se debe haber iniciado la simulación en HandlingPro. Sin embargo, esto no inicia la ejecución de los programas. Para iniciarla o para reiniciarla tras rearmar la estación debido a una parada de emergencia, se deben pulsar los siguientes botones en el siguiente orden:

- 1) **Restea Errores.** Para eliminar cualquier error que haya en la estación. Si no lo hay, esta acción no tiene repercusión en la estación.
- 2) **Detención de Seguridad.** Si había algo ejecutándose por error (no debería suceder nunca), se para por seguridad.
- 3) **Arrancar.** Se inicia/reinicia la ejecución de los programas en **modo manual siempre** – incluido, si antes de la parada de emergencia, la estación estaba en modo de funcionamiento automático. Esto es así por seguridad aunque la planta sea de simulación.

CAPÍTULO 5.2.1.2. Programación de la interfaz.

Cuando se tienen todos los elementos dispuestos sobre las diferentes imágenes de la interfaz, hay que dotarles de funcionalidades para poder gestionar la estación robotizada. En esta interfaz se han programado tres aspectos:

1) **Programación de eventos.** Por eventos se entienden las funcionalidades directamente asociadas a los botones que componen las imágenes de la interfaz.

TIA Portal ofrece gran variedad de eventos:

ActivarBit	BorrarBúferDeAvisos	ControlarSmartServer
ActivarBitDeVariable	BorrarBúferDeAvisosProtocol	CopiarFichero
ActivarBitMientrasTeclaPulsada	BorrarFichero	CopiaSeguridadSistemaDeArchivosRAM
ActivarImagen	BorrarRegistro	DecrementarVariable
ActivarImagenAnterior	BorrarRegistroMemoria	DefinirEstadoVariableDeReceta
ActivarImagenConNúmero	CalibrarPantallaTáctil	DefinirModoPLC
ActivarModoDeInformeAviso	CambiarConexión	DefinirVariable
ActivarPantallaDeLimpieza	CambiarConexiónEIP	DesactivarBit
ActivarVisorCódigoPLC	CargarRegistro	DesactivarBitEnVariable
ActualizarVariable	CerrarSesión	DesplazarYEnmascarar
AjustarFechaHoraPLC	CerrarTodosLosFicheros	EnviarCorreoElectrónico
AjustarHorarioVerano	Codificar	EscalaLineal

FIGURA 132. Ejemplos de eventos.

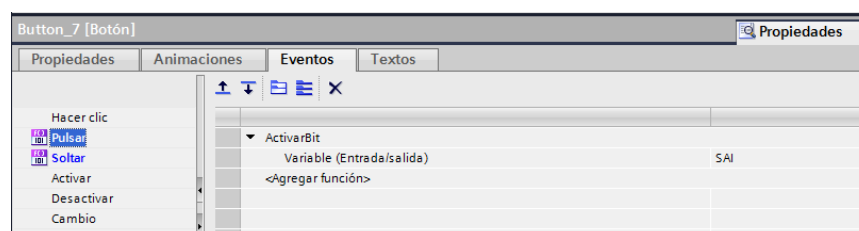
Los eventos utilizados en esta interfaz son los relacionados con activación (puesta a 1) y desactivación (puesta a 0) de bits; el de cambio de imagen que se relacionan con los botones de la parte inferior de la interfaz; el de selección de idioma; el de selección del modo del PLC; y el de finalización de la simulación de la interfaz (*'runtime'*).

Estos eventos se pueden producir en diferentes momentos:



FIGURA 133. Opciones de activación de eventos.

Cabe la posibilidad de producirse un evento al pulsar un botón y producirse otro diferente al desactivar ese mismo botón – aportando gran libertad a la hora de programar la interfaz. Se muestra, a continuación, un ejemplo de esto:



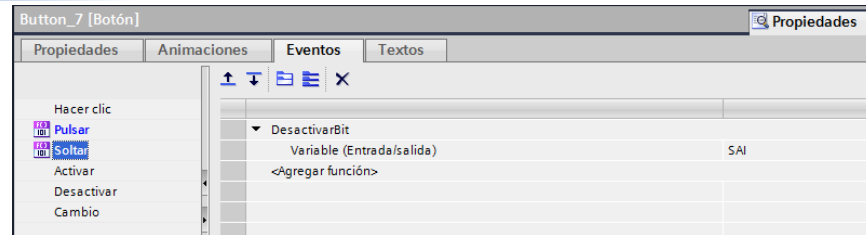


FIGURA 134. Ejemplo de programación de eventos.

- 2) **Programación de visibilidad.** Para poder ocultar elementos dentro de la interfaz – como se hace con los botones de “Reanudar” tras pausa controlada; o como se hace con los diferentes elementos de la imagen “INDIVIDUAL”, dónde se muestra el contenido en función del robot seleccionado – ha de programarse su visibilidad.

Para ello, la visibilidad de los elementos de la interfaz utilizada en este proyecto depende del valor de diferentes variables según convenga. Estas variables pueden ser “Bool” o no (“Integer”, “Word”, etc.). Sin embargo, siempre hay que dar un rango de valores de esas variables. Por ejemplo: si se desea que un elemento sea visible cuando una variable “Bool” esté activa; hay que seleccionar la variable, darle el rango desde 1 hasta 1, y seleccionar la opción “Visible”.

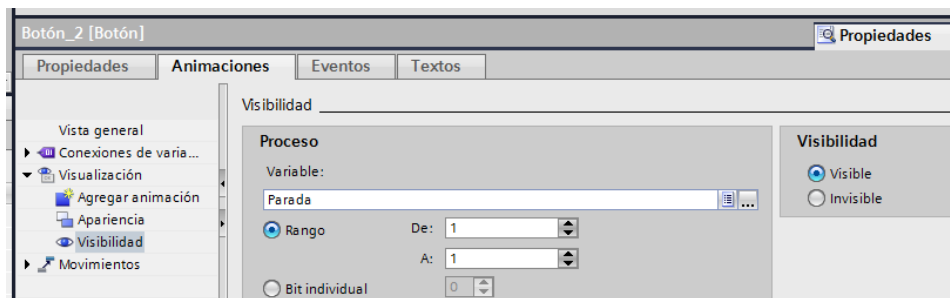


FIGURA 135. Ejemplo de programación de visibilidad.

Si, por ejemplo, se desea que un elemento sea invisible en un cierto rango de valores de una variable “Integer”; ha de seleccionarse la variable, especificar el rango deseado y marcar la opción “Invisible”.

- 3) **Programación de apariencia.** La apariencia de ciertos elementos se puede modificar. En este proyecto se modifica la apariencia de los botones de mando (marcha, paro, arranque, mover robot en modo individual...) al ser pulsados. Esto se ha hecho así para asegurar al operario de que ha pulsado/soltado un botón.

ROBOT 1

ROBOT 1

FIGURA 136. Botón en reposo vs botón pulsado.

Por otro lado, cualquier señalización de la interfaz está construida sobre un elemento llamado “Rectángulo” cuya apariencia es modificada. TIA Portal da la opción de parpadeo – por lo que no es necesario activar las marcas de ciclo de TIA Portal: variables “Bool” que se activan y desactivan periódicamente.

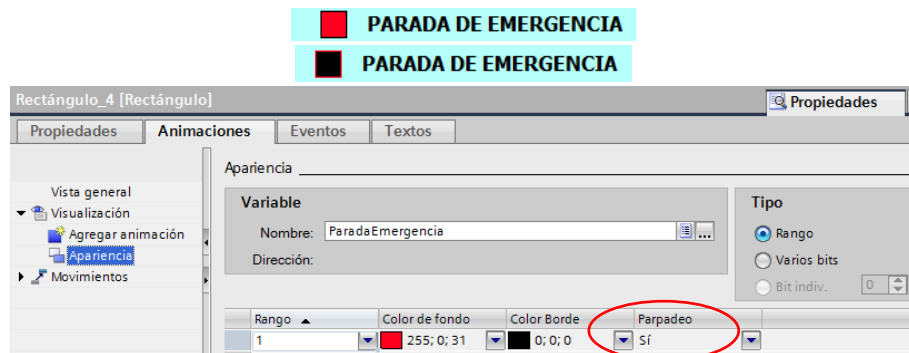


FIGURA 137. Señalización con parpadeo.

Para acceder a la programación de los tres aspectos anteriores, se deberá seleccionar el elemento a programar. Después se accede a la ventana de propiedades de la parte inferior de la interfaz de TIA Portal. Ahí están las tres pestañas relativas a la programación de los elementos de la HMI.

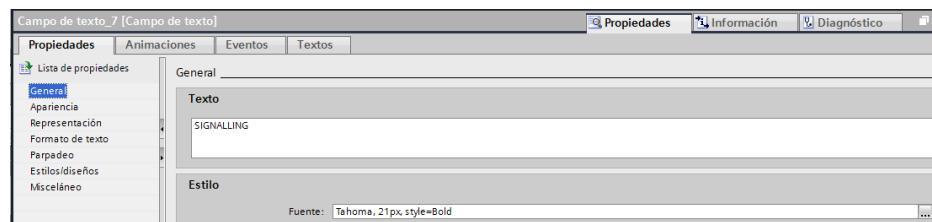


FIGURA 138. Ventana de propiedades: programación de elementos de una HMI.

Surge un problema con la programación de eventos: no se implementa ninguna lógica. Sencillamente, se activan y desactivan variables, lo que podría conllevar contradicciones de mando como marchar y parar a la vez o seleccionar más de un robot a la vez y moverlos. Por ello se requiere de un PLC que controle que las órdenes enviadas desde la HMI no produzcan las contradicciones mencionadas entre otras tareas que debe llevar a cabo.

CAPÍTULO 5.2.2. Autómata programable (PLC).

El PLC utilizado en el proyecto es un S7-1500 [27] de Siemens con conexión Profinet que permite la conexión con la TP700 Comfort. Al igual que en esta, el autómata es de simulación.

Aunque hay varias opciones de cara a la programación de los autómatas en TIA Portal, generalmente (y en este proyecto) se emplea el lenguaje de programación KOP. Este lenguaje se basa en el diagrama de escalera o 'ladder' que incluye segmentos compuestos por contactos y bobinas. En función del valor de los contactos (abierto o cerrado) se activan o no las bobinas. De esta manera se puede implementar la lógica de control de un proceso.

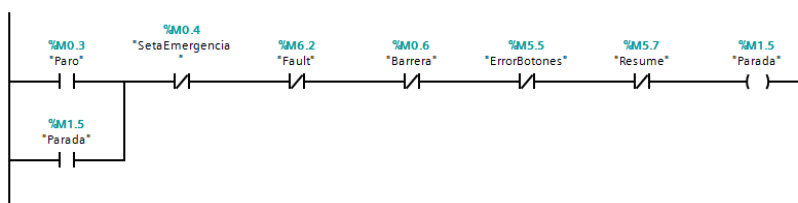


FIGURA 139. Ejemplo lenguaje KOP.

Existen diferentes tipos de variables:

- **Entradas (%I).** Las entradas son variables que lee el autómata desde el exterior. Se emplean como contactos en la programación.
- **Marcas (%M).** Las marcas son variables internas del PLC. Pueden ser empleadas como contactos y como bobinas.
- **Salidas (%Q).** Las salidas son variables que escribe el autómata en función de las entradas y marcas. El valor de las salidas solo se modifica una vez en un "ciclo de scan", por lo que, cada salida se incluye una única vez en todo el programa. De lo contrario, el valor que adquieran estas salidas puede ser erróneo. Se emplean como bobinas en la programación.

TIA Portal incluye elementos para asignar un valor a una variable, realizar cálculos aritméticos, temporizar la ejecución del programa, o contabilizar el número de veces que ocurre un evento. Todas estas instrucciones son accesibles desde el área de trabajo (ver figura 38) o pulsando sobre el siguiente icono:

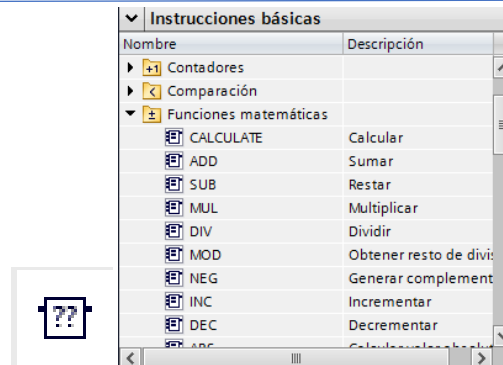


FIGURA 140. Instrucciones adicionales.

Todos estos elementos que sirven de base se agrupan en diferentes bloques según las necesidades de la programación. Los existen varios tipos de bloques:

- **Bloques de organización (OB).** Los bloques de organización se emplean para ejecutar los programas de usuario. En función del tipo de OB, su ejecución puede ser cíclica o activada por eventos (arranque, interrupciones...).
- **Bloques de función (FB).** Los bloques de función agrupan segmentos de programación a los que el bloque de organización llama en función de ciertas condiciones que se den en el programa. Tienen asociado un bloque de datos (DB).
- **Funciones (FC).** Son prácticamente lo mismo que los FB con una simple diferencia: no tienen asociado un bloque de datos.
- **Bloques de Datos (DB).** Los bloques de datos almacenan datos del programa. Resultan útiles para registrar parámetros de un proceso.

La programación del PLC del proyecto se ha realizado en base a los siguientes dos tipos de bloques de programación.

CAPÍTULO 5.2.2.1. Bloques de organización (OB).

En este proyecto solo se han utilizado dos OB:

- 1) **OB100 - Arranque:** este bloque únicamente se ejecuta una vez. Se ejecuta al iniciar la simulación autómatas programable únicamente.

En el proyecto se emplea para activar algunas variables que son estrictamente necesarias para que los robots de HandlingPro puedan funcionar.

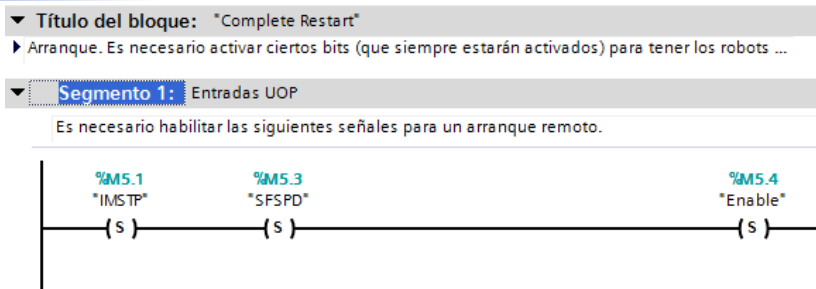


FIGURA 141. OB de arranque.

- 2) **OB1 – Main**: este bloque es el pilar de los programas de usuario. Se ejecuta cíclicamente, por lo que es empleado para incluir segmentos de programación importantes o realizar llamadas a otros bloques.

En este trabajo realiza las llamadas a todas las funciones e incluye algunos segmentos comunes a varias imágenes de la HMI como las paradas y la reanudación del movimiento. E incluye funcionalidades que deben de ser programadas pertenecientes a la imagen de ajustes – ya que no se ha creado una función para esta imagen puesto que se compondría de un solo segmento de programación: el arranque remoto de la estación robotizada.

El código completo se adjunta en el apartado de anexos: ANEXO A. Bloques de programación de TIA Portal.

▶ Segmento 1: Arranque de la estación
▶ Segmento 2: Parada controlada de la estación
▶ Segmento 3: Reanudación tras parada controlada
▶ Segmento 4: Parada de Emergencia
▶ Segmento 5: Llamada a Ejecución Continua
▶ Segmento 6: Llamada a Errores y Señalización
▶ Segmento 7: Llamada a Individual

FIGURA 142. OB Main.

CAPÍTULO 5.2.2.2. Funciones (FC).

Prácticamente se incluye una función por imagen de la HMI. Desde las funciones, se gestiona la lógica de control de las órdenes enviadas desde la interfaz para que no ocurran contradicciones de mando y la estación funcione como se espera. Estas son llamadas desde el 'Main' incondicionalmente puesto que gestionan el funcionamiento de la interfaz hombre-máquina.

El código completo se adjunta en el apartado de anexos (ANEXO A. Bloques de programación de TIA Portal.)

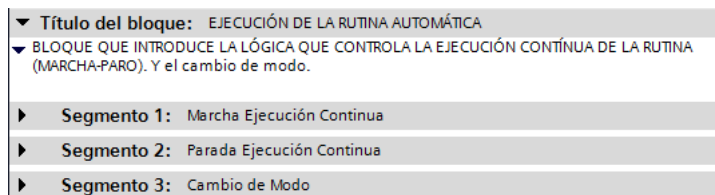


FIGURA 143. FC Continua.

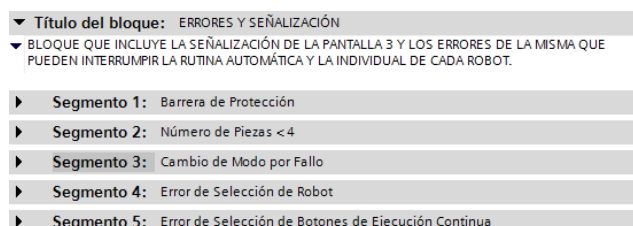


FIGURA 144. FC Errores y Señales.



FIGURA 145. FC Individual.

CAPÍTULO 5.3. Desarrollo en KEPServerEX.

Como ya se ha mencionado anteriormente, en este proyecto se ha configurado un servidor OPC para interconectar datos entre el PLC y la HMI de simulación en TIA Portal y la estación robotizada en HandlingPro puesto que estos dos softwares no tienen conectividad nativa.

CAPÍTULO 5.3.1. Configuración del servidor OPC UA.

Para comenzar con la configuración del servidor, se debe crear un canal de comunicación haciendo 'click' en 'New Channel' situado en la barra de herramientas.



FIGURA 146. Crear canal de comunicación.

En este punto se abre un asistente que irá solicitando toda la información relativa al canal de comunicación que empleará el servidor OPC.

Paso 1 – Elección del canal. Se necesita saber qué tecnología de comunicación se requiere. En este caso, se debe seleccionar la opción “**Siemens TCP/IP Ethernet**” ya que se desea tener acceso a las variables PLC de simulación. Como ya se explicó, la vía de acceso al PLC de simulación es a través de NetToPLCSim de Siemens. Esta extensión de software creaba una vía de comunicación utilizando el protocolo TCP/IP (puerto 102).

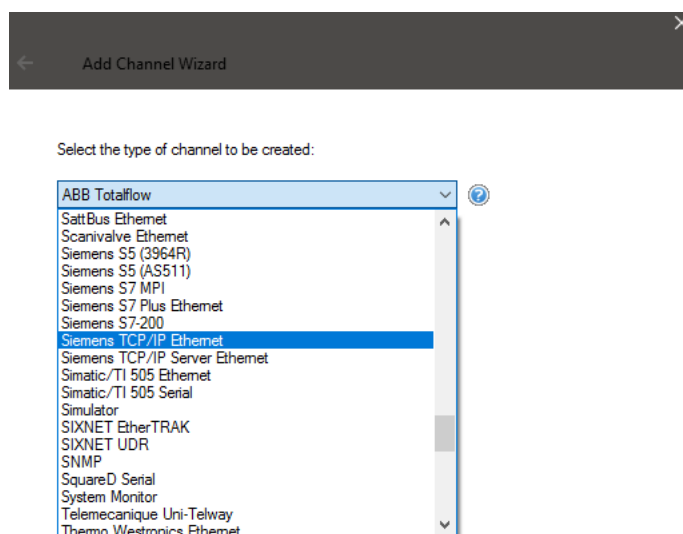


FIGURA 147. Siemens TCP/IP Ethernet.

Paso 2 – Identidad del canal. Sencillamente, se le da un nombre al canal.

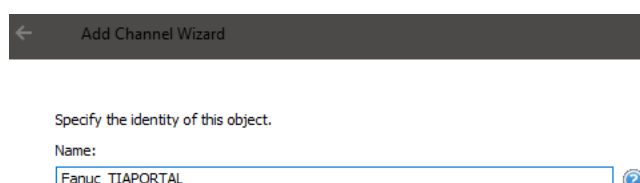


FIGURA 148. Nombre del canal.

Paso 3 – Adaptador de red. Se selecciona el adaptador de red asociado al PLC. En este caso, el del PLC es de simulación:

Binding	Adapter Name
	Default
10.0.4.15	Intel(R) PRO/1000 MT Desktop Adapter #3
169.254.236.139	Intel(R) PRO/1000 MT Desktop Adapter
172.25.212.2	Intel(R) PRO/1000 MT Desktop Adapter #2
192.168.1.1	Siemens PLCSIM Virtual Ethernet Adapter

FIGURA 149. Adaptador de red.

Los siguientes pasos se mantienen por defecto hasta que el asistente muestre el resumen de la configuración. Se pulsa finalizar para terminar con la configuración del canal de comunicación. Aparece lo siguiente en el árbol del proyecto:

Identification	
Name	FANUC_TIAPORTAL_
Description	
Driver	Siemens TCP/IP Ethernet
Diagnostics	
Diagnostics Capture	Disable
Tag Counts	
Static Tags	0
Ethernet Settings	
Network Adapter	Siemens PLCSIM Virtual Ethernet ...
Write Optimizations	
Optimization Method	Write Only Latest Value for All Tags
Duty Cycle	10

Finalizar Cancelar

FIGURA 150. Canal creado.

Seguidamente se pasa a configurar el servidor pulsando sobre la opción 'New Device' de la barra de herramientas. En este punto se abre un asistente que irá solicitando toda la información relativa al PLC con el que se desea conectar.



FIGURA 151. Añadir dispositivo.

Paso 1 – Identidad del PLC. Sencillamente se le da un nombre.

Specify the identity of this object.

Name:

S71500

FIGURA 152. Identificador del PLC.

Paso 2 – PLC. Se selecciona el modelo del PLC de Siemens con el que se desea conectar. En este caso, un S7-1500.

Select the specific type of device associated with this ID. Options depend on the type of communications in use.

Model:

S7-200	▼	?
S7-200		
S7-300		
S7-400		
S7-1200		
S7-1500		
NetLink: S7-300		
NetLink: S7-400		

FIGURA 153. Modelo del PLC.

Paso 3 – Dirección IP del PLC. Esta debe ser la que se le asignado desde TIA Portal.

Specify the device's driver-specific station or node.

ID:

 ?

FIGURA 154. Dirección IP del PLC.

Los siguientes pasos se mantienen por defecto hasta que el asistente muestre el resumen de la configuración. Se pulsa finalizar para terminar con la configuración del PLC.

Lo que resta es añadir las variables que se desean interconectar entre programas. En el proyecto se han creado diferentes carpetas con variables por motivos de organización.

Para añadir las variables, se debe seleccionar el PLC creado y pulsar sobre el icono de “Crear etiqueta” de la barra de tareas. Se abre un cuadro de diálogo para la configuración de la etiqueta:



FIGURA 155. Añadir etiqueta.

Se debe especificar un nombre – conviene que coincida con el nombre asignado a la variable en TIA Portal. Se puede añadir una descripción.

Después se especifica la dirección de memoria de la variable. Esta debe ser exactamente igual a la dirección asignada en TIA Portal para poder acceder a su valor. Por ejemplo: si en TIA Portal se tiene una variable ‘Bool’ llamada “Marcha” almacenada en la dirección %M1.7 con efecto en la estación de HandlingPro; en el servidor se debe crear una etiqueta cuya dirección es M1.7.

El tipo de dato se puede especificar, pero si las variables son ‘Bool’ (como todas las variables empleadas en el proyecto) esto se ajusta automáticamente. En

este caso, como todas las variables se leen y escriben desde ambos lados de la comunicación, el acceso es de Lectura/Escritura. Y el último parámetro marca el tiempo de refresco del valor de la etiqueta.

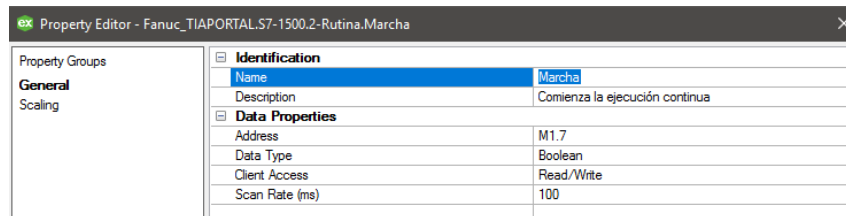


FIGURA 156. Configuración de la etiqueta.

Cuando todas las etiquetas estén incluidas, se puede dar por concluida la configuración del servidor OPC UA.

CAPÍTULO 5.4. Conexiones entre programas.

Desarrollada la estación robotizada, programados los robots, el PLC y la HMI, y configurado el servidor OPC UA; únicamente resta interconectar todos los programas para lograr el funcionamiento deseado.

La estructura de la conexión es la siguiente:

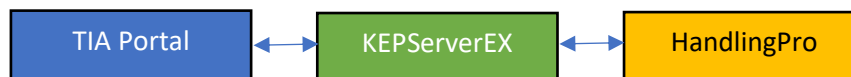


FIGURA 157. Estructura de la conexión entre programas.

Tanto desde TIA Portal como desde HandlingPro se accede a ciertas variables cuyo valor se almacena en el servidor OPC de KEPServerEX. Estos valores se pueden modificar desde cualquier punto de la comunicación.

Pero para que esta comunicación sea efectiva, los diferentes programas necesitan ser configurados para poder comunicarse con “el exterior”.

CAPÍTULO 5.4.1. HandlingPro.

CAPÍTULO 5.4.1.1. Conexión HandlingPro – Servidor OPC UA.

Después de la figura 25 se explicó que el apartado de ‘External Devices’ servía para configurar la conexión con un servidor OPC. Para ello se hace ‘click’ derecho sobre el apartado mencionado, se selecciona la opción de añadir dispositivo, y finalmente se selecciona la opción deseada. En este caso un servidor OPC.

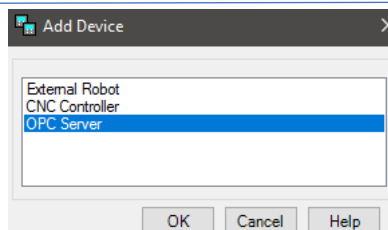


FIGURA 158. Comunicaciones en HandlingPro.

Se abre el siguiente cuadro de diálogo donde se le da un nombre al servidor con el que se conecta, se elige el software sobre el que se ha montado el servidor OPC, se especifica la ruta de acceso a las variables, el producto/interfaz en caso de que fuera necesario, y el tiempo de refresco del valor de las etiquetas a las que se accede.

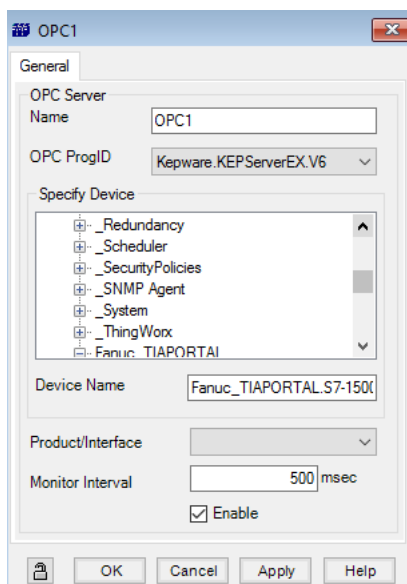


FIGURA 159. Conexión con servidor en HandlingPro.

En caso de tener varias carpetas con variables en el servidor, se deben incluir tantos servidores en HandlingPro como carpetas haya. Esto se debe a que cada carpeta tiene asociada una ruta diferente.

Al haber incluido todas las carpetas con las variables provenientes del servidor OPC, estas variables ya son conectables con las señales de entrada y salida de los robots. Para conectar estas variables, con las señales de los robots se hace 'click' derecho sobre el apartado de 'External Devices' y se selecciona la opción de 'External I/O Connection'.

	Bit Len.	Sign	Output Device	Type	Number	Value	Input Device	Type	Number	Value
▶	1	Unsign	1-Arr	ON	---	---	Robot Controller1	UI	18	---
	1	Unsign	2-Rutina	SetaEmer...	---	---	Robot Controller1	UI	4	---
	1	Unsign	3-Ind	R1_DI7_...	---	---	Robot Controller1	DI	7	---
	1	Unsign	3-Ind	R1_DI8_...	---	---	Robot Controller1	DI	8	---
	1	Unsign	3-Ind	R1_DI9_...	---	---	Robot Controller1	DI	9	---
	1	Unsign	0-Modo	R1_DI10...	---	---	Robot Controller1	DI	10	---
	1	Unsign	1-Arr	IMSTP	---	---	Robot Controller1	UI	1	---
	1	Unsign	1-Arr	HOLD	---	---	Robot Controller1	UI	2	---
	1	Unsign	1-Arr	SFSPD	---	---	Robot Controller1	UI	3	---
	1	Unsign	1-Arr	ENABLE	---	---	Robot Controller1	UI	8	---
	1	Unsign	1-Arr	IMSTP	---	---	Robot Controller2	UI	1	---

Operation: Remove [Monitor Interval: 100 msec] Connect

FIGURA 160. Conexiones externas.

Hay señales muy importantes para el correcto funcionamiento de los robots y para que la estación pueda funcionar en remoto. Estas señales son las denominadas UOP ('User Operator Panel'). Hay dos tipos de señales UOP: las de entrada y las de salida.

- **Señales UOP de entrada.** Son señales digitales de entrada desde la perspectiva del robot que manejan el funcionamiento del robot: selección de programas por los métodos RSR, PNS y STYLE (9-17), arranque(18)/parada(2)/ reanudación(6) / aborto(4) de la ejecución de un programa, reseteo de faltas (5), robot a HOME (7). Y hay tres señales que si no están activas, los robots no se moverán ya que se generará un error por cada variable no activa. Estas variables son las que se activan en la OB de arranque de la programación del PLC: IMSTP (1), SFSPD (3) y Enable (7).

Para visualizarlas se debe seguir la ruta **MENU > 5 I/O > UOP > IN/OUT.**

I/O UOP In			
	#	STATUS	1/18
UI[1]	ON	[*IMSTP]
UI[2]	ON	[*Hold]
UI[3]	ON	[*SFSPD]
UI[4]	OFF	[Cycle stop]
UI[5]	OFF	[Fault reset]
UI[6]	OFF	[Start]
UI[7]	OFF	[Home]
UI[8]	ON	[Enable]
UI[9]	OFF	[RSR1/PNS1/STYLE1]
UI[10]	OFF	[RSR2/PNS2/STYLE2]
UI[11]	OFF	[RSR3/PNS3/STYLE3]

Sorted by port number.

FIGURA 161. UOP in.

- **Señales UOP de salida.** Son señales digitales de salida desde la perspectiva del robot que sirven para informar del estado de los propios robots y de los programas que ejecutan.

Para visualizarlas se debe seguir la ruta **MENU > 5 I/O > UOP > IN/OUT.**

I/O UOP Out	#	STATUS		1/20
	UO[1]	OFF	[Cmd enabled]
	UO[2]	ON	[System ready]
	UO[3]	OFF	[Prgrunning]
	UO[4]	OFF	[Prgrpaused]
	UO[5]	OFF	[Motion held]
	UO[6]	OFF	[Fault]
	UO[7]	OFF	[At perch]
	UO[8]	ON	[TP enabled]
	UO[9]	OFF	[Batt alarm]
	UO[10]	OFF	[Busy]
	UO[11]	OFF	[ACK1/SNO1]

Sorted by port number.

FIGURA 162. UOP out.

Pero para que los robots puedan funcionar en modo remoto no sólo hay que conectar sus señales con las del servidor OPC. Se requiere de unas configuraciones en sus respectivos controladores y programas para que esta conexión sea efectiva.

CAPÍTULO 5.4.1.2. Configuraciones de los controladores, señales y programas.

A nivel de sistema, hay que configurar todos los controladores en modo remoto. Para ello se sigue la ruta **MENU > NEXT > 6 SYSTEM > 5 CONFIG.** Una vez ahí, se desciende hasta el punto 43, donde se seleccionará la opción 'Remote':

System/Config		51/61
41	Signal if OVERRIDE = 100	DO[0]
42	Hand broken :	<*GROUPS*>
43	Remote/Local setup:	Remote
44	External I/O(ON:Remote):DI	[12]

FIGURA 163. Sistema en remoto.

Por otro lado, las señales digitales de interconexión deben ser configuradas de forma que sean accesibles para el canal de configuración. Por lo tanto, su 'Rack' y 'Slot' debe ser 89 y 1 respectivamente. Esta configuración es la indicada en los manuales del fabricante para la conexión de señales mediante tecnología Ethernet.

#	RANGE	RACK	SLOT	START	STAT.
1	DO[XXXXXXXXXX 1- 512]	89	1	1	ACTIV

#	RANGE	RACK	SLOT	START	STAT.
1	DI[XXXXXXXXXX 1- 512]	89	1	1	ACTIV

FIGURA 164. Rack y Slot de las señales digitales.

La configuración normalmente para un proyecto de simulación se realiza con 'Rack' y 'Slot' 0 y 1 respectivamente, pero esta es la configuración utilizada para las señales de entrada UOP, por lo tanto, se ha cambiado la configuración para evitar superposición de señales.

Finalmente, queda configurar los programas que van a ejecutar los robots. En este caso, hay tres programas (uno por cada robot) llamados "Main" que combinan el modo manual con el modo automático en función de la elección del usuario. Cada robot ejecutará su "Main", cuyo modo de ejecución debe ser configurado:

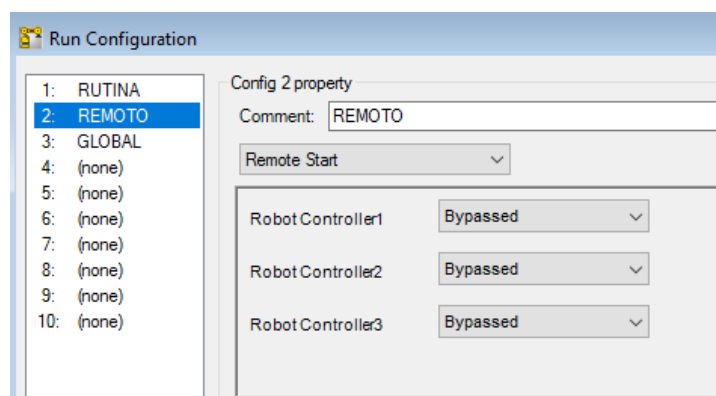


FIGURA 165. Configuración de la ejecución (I)

Se crea un modo de ejecución con arranque remoto, en el que el programa de cada robot debe ser especificado a nivel de selección de programa. Esta especificación se hace siguiendo la siguiente ruta: **MENU > 6 SETUP > 1 PROG. SELECT**. En este caso, el modo de selección del programa es OTHER.

```
Prog Select
1 Program select mode: OTHER
2 Production start method: UOP
Production checks:
3 At home check: DISABLED
4 Resume position toler.: ENABLED
5 Simulated I/O: DISABLED
6 General override < 100%: DISABLED
7 Prog override < 100%: DISABLED
8 Machine lock: DISABLED
9 Single step: DISABLED
10 Process ready: DISABLED
```

FIGURA 166. Configuración de la ejecución (II)

Si se accede a los detalles del modo de selección del programa, se podrá especificar el programa a ejecutar:

```
Prog Select
OTHER Program select mode is enabled.
1/1
With this selection method, the
application must set the variable
$shell_wrk.$cust_name to the desired
program name.
$shell_wrk.$cust_name : MAIN
```

FIGURA 167. Configuración de la ejecución (III)

Con esto, el funcionamiento en remoto desde la estación está completamente configurado.

Sin embargo, si se comienza la simulación, pero no se han conectado las variables con el exterior, no se podrá comunicar la estación con el servidor OPC UA.

Así que han de conectarse las variables de HandlingPro con las del servidor. Para ello, se pulsará el botón de 'Connect' de la esquina inferior derecha de la figura 160 antes de iniciar la simulación.

CAPÍTULO 5.4.2. TIA Portal.

Para que el servidor OPC pueda acceder a las variables del PLC, la dirección IP de este debe ser exactamente igual a la especificada en KEPServerEX (ver figura 154).

Para asignarle este mismo valor, se debe pulsar dos veces en el apartado "Configuración de Dispositivos" del árbol del proyecto.

Posteriormente, se debe acceder a la pestaña “General” de la ventana de propiedades. Una vez ahí, se navega hasta el apartado de “Direcciones Ethernet”.

Y finalmente, en el apartado de “Protocolo IP” se podrá asignar la dirección IP del autómata.

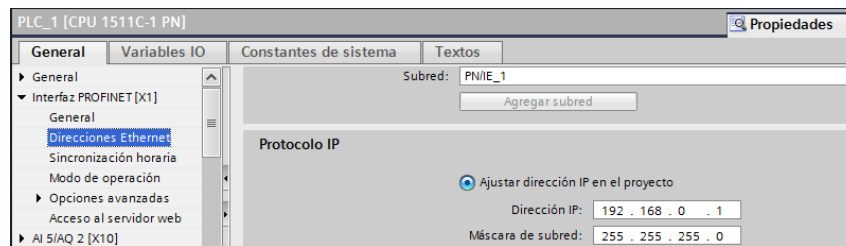


FIGURA 168. Asignación de dirección IP del PLC.

Por otro lado, en el capítulo 3.4.4. se había dicho que NetToPLCSim “permite acceder a la simulación del PLC (en PLCSim) desde la red a la que está conectada el PC donde se ejecuta la simulación **utilizando la interfaz de red del propio PC**”. Así pues, se configura esta interfaz de red del PC.

En el apartado de “Conexiones de red” del panel de control del PC, se puede acceder a todos los adaptadores de red Ethernet de las que dispone el ordenador. Por supuesto, debería aparecer un adaptador virtual asociado a PLCSim.

- 1) Se hace doble ‘click’ sobre la red correspondiente. Y se abre el cuadro de diálogo del estado de dicha red.
- 2) Se pulsa sobre el botón de “Propiedades”. Y se abre el cuadro de diálogo de las propiedades de la red.
- 3) Se selecciona y se marca el ‘checkbox’ de la opción ‘Internet Protocol Version 4 (TCP/IPv4)’. Hecho esto, con esa opción seleccionada, se pulsa sobre el botón de “Propiedades”. Y se abre el cuadro de diálogo de las propiedades del protocolo de la red.
- 4) En este punto, se usa una dirección IP fija, que aparecerá en KEPServerEX cuando se creó el canal de comunicación (ver figura 149).

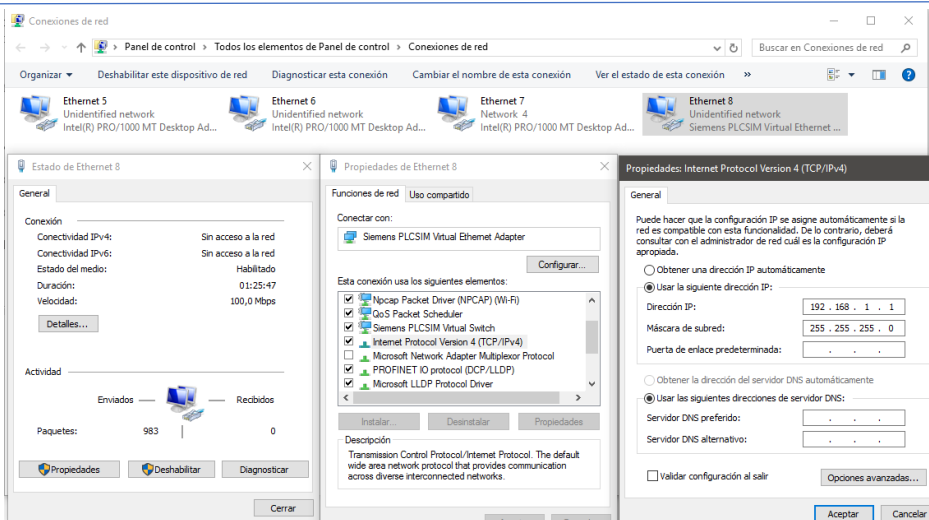


FIGURA 169. Configuración de la interfaz de red de PLCSim.

Por último, para que la simulación del PLC virtual no de errores de protección y seguridad – ya que se trata de acceder a valores de variables guardadas en la memoria del PLC desde un agente externo – hay que seleccionar la siguiente opción de permiso de lectura/escritura desde un interlocutor remoto:

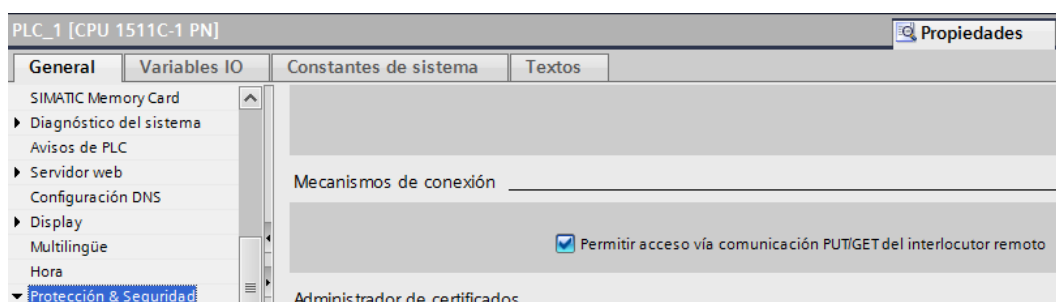


FIGURA 170. Permiso de acceso desde interlocutor remoto.

Para llegar a activarlo, se debe seguir los siguientes pasos:

- 1) En el árbol del proyecto, seleccionar “Configuración de Dispositivos”.
- 2) En la ventana de propiedades, seleccionar la pestaña “General”.
- 3) En la opción “Protección & Seguridad”, navegar hasta el apartado de “Mecanismos de conexión”.
- 4) Marcar el ‘checkbox’ de la figura 172.

Hecho esto, la conexión de TIA Portal con el servidor OPC será posible.

CAPÍTULO 5.4.3. KEPServerEX.

Por último, para que toda la comunicación sea efectiva, el servidor OPC UA de KEPServerEX debe estar corriendo.

Si nunca se ha ejecutado tras su creación, se debe seleccionar la opción 'Reinitialize' de la figura 173 para que todos los cambios se graben y entre en funcionamiento el servidor. Con cada cambio que se haga en el servidor también se debe seleccionar esta opción para guardar esos cambios.

Si no se han realizado cambios, sencillamente se pulsa la opción 'Connect'.

El servidor comenzará a correr con cualquiera de estas dos formas. Y la conexión será posible.

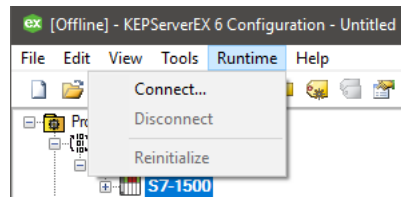


FIGURA 171. Inicio del servidor OPC UA.

CAPÍTULO 6. RESULTADOS.

Se detalla más en profundidad qué maniobras se pretenden llevar a cabo y en qué orden. Nada más se inicia la simulación, se debe arrancar la ejecución de los tres programas “Main” (uno por cada robot) desde la HMI. Hecho esto; comienzan las maniobras:

- 1) Siempre se parte en modo manual tras el arranque. En este modo se podrán hacer los diferentes movimientos de cada robot para realizar un testeo de la correcta ejecución de cada movimiento.

Para ello se debe seleccionar un único robot de entre los tres disponibles en la imagen “INDIVIDUAL” de la HMI. Tras la selección de un robot y un movimiento, se deberá esperar a su conclusión para volver a seleccionar otro movimiento con dicho robot – los mandos no se acumulan en una cola para su ejecución automática tras la finalización del mando anterior.

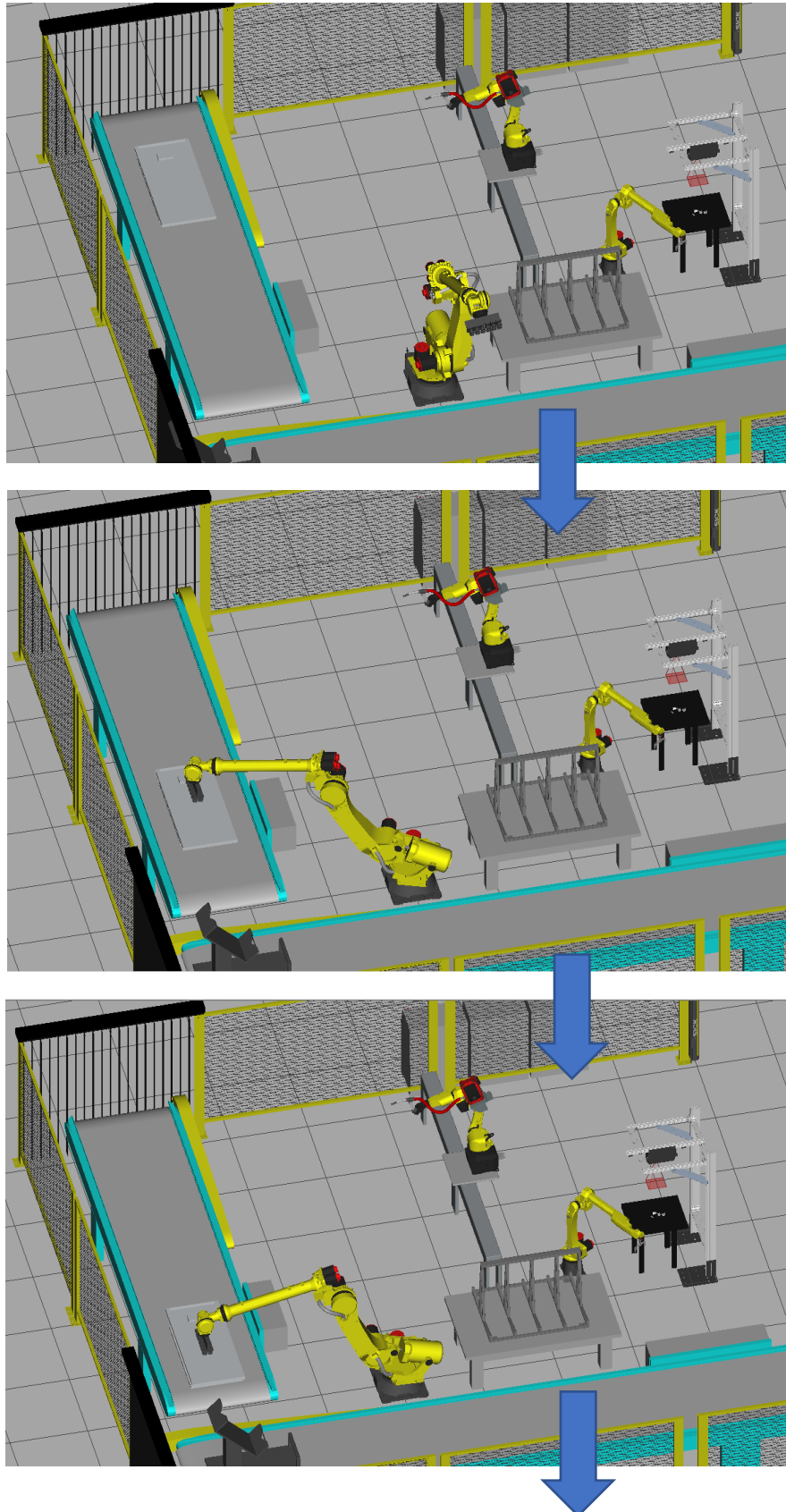
Sin embargo, en este modo manual, diferentes robots pueden realizar un movimiento “al mismo tiempo”. Pero primero se debe seleccionar un robot, ordenar su movimiento, deseleccionar dicho robot, seleccionar otro robot diferente y ordenar su movimiento. De esta forma se pueden estar ambos (o los tres si se desea) en funcionamiento “simultáneamente” en este modo.

- 2) Una vez se concluyan las pruebas que el usuario quiera realizar en modo manual. Se podrá cambiar a modo automático en cuanto se desee. Este cambio se realiza pulsando sobre el botón “**MARCHA**” de la imagen “CONTINUA” de la HMI.



FIGURA 172. Botón de MARCHA.

- 3) Una vez en este modo, el orden de las maniobras es el que sigue:
 - El Robot1 espera a que una placa solar llegue a su posición de recogida. Hace ‘*Line Tracking*’ a la línea de llegada para tomarla. Va a su posición de ‘HOME’ con la placa ya agarrada. Deposita la placa sobre el soporte de la mesa de soldadura. Ordena al Robot3 comenzar con el proceso de visión artificial. Y retorna a su posición de reposo. Se adjuntan los resultados de esta maniobra a continuación:



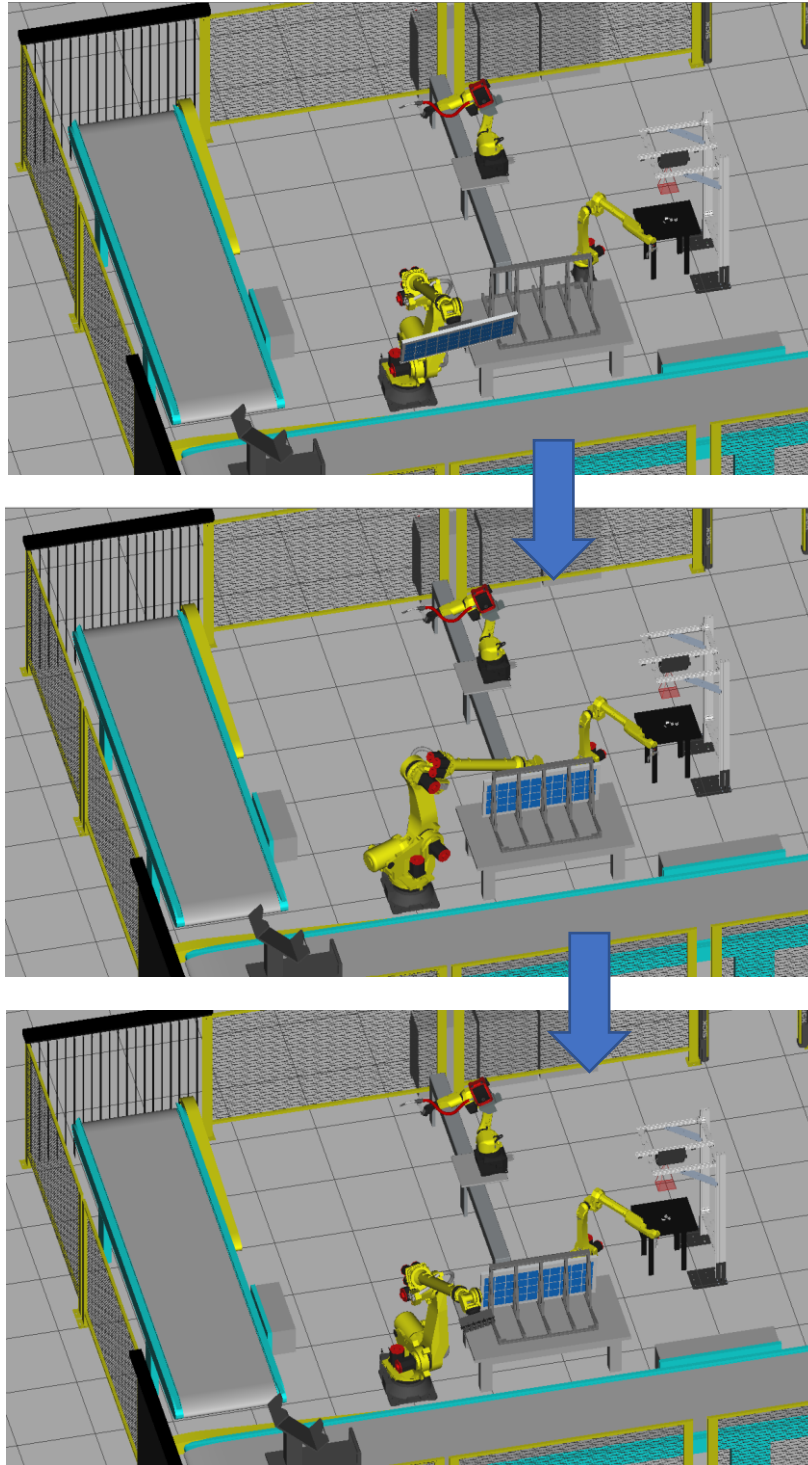
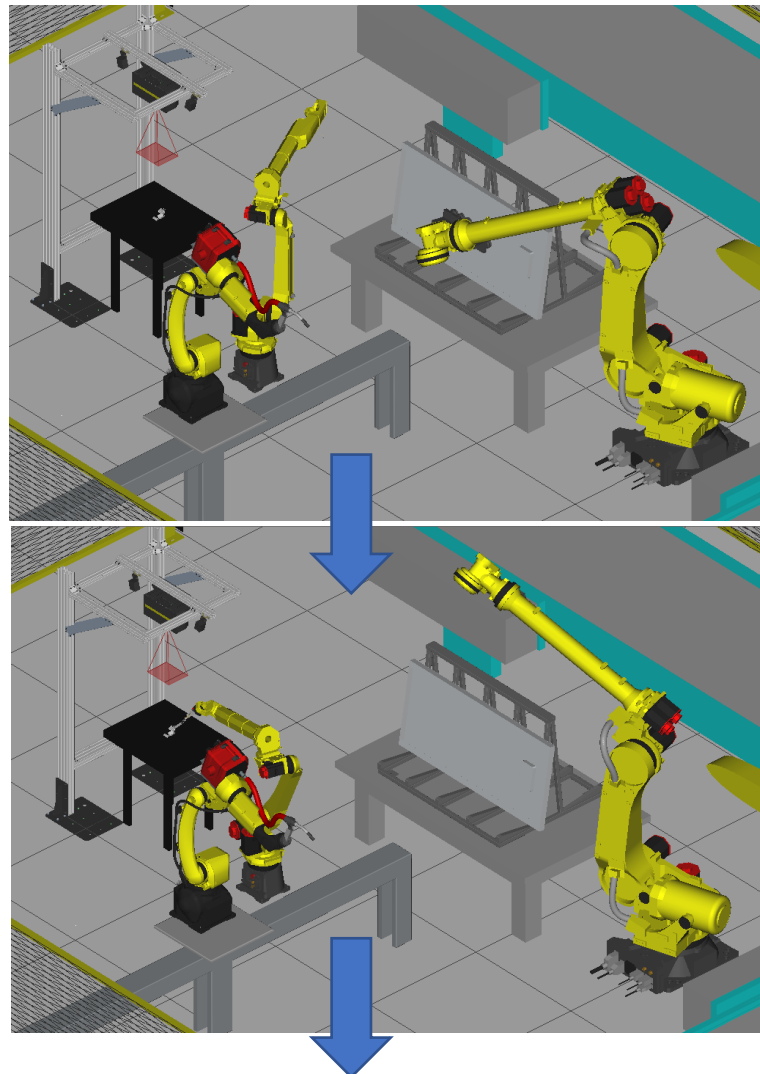


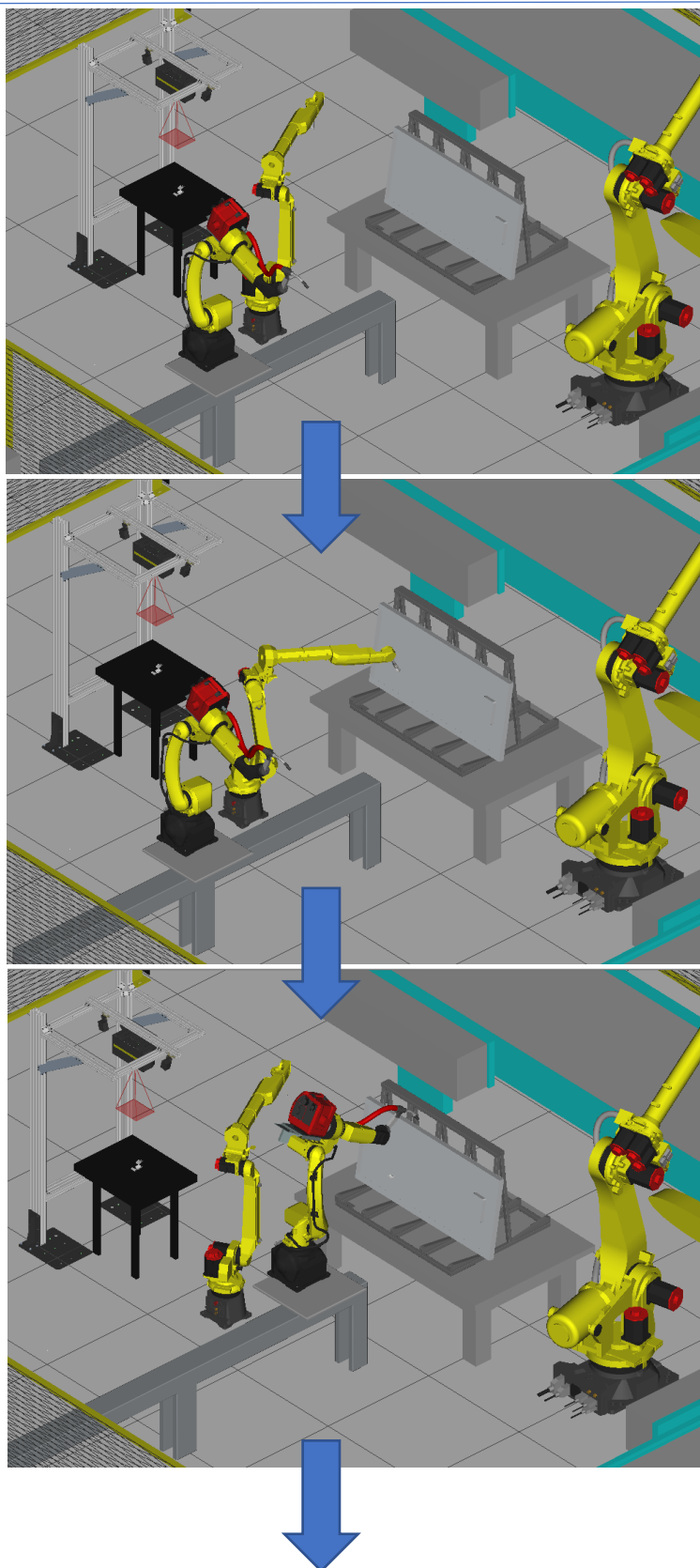
FIGURA 173. Secuencia de movimientos del Robot1 (I).

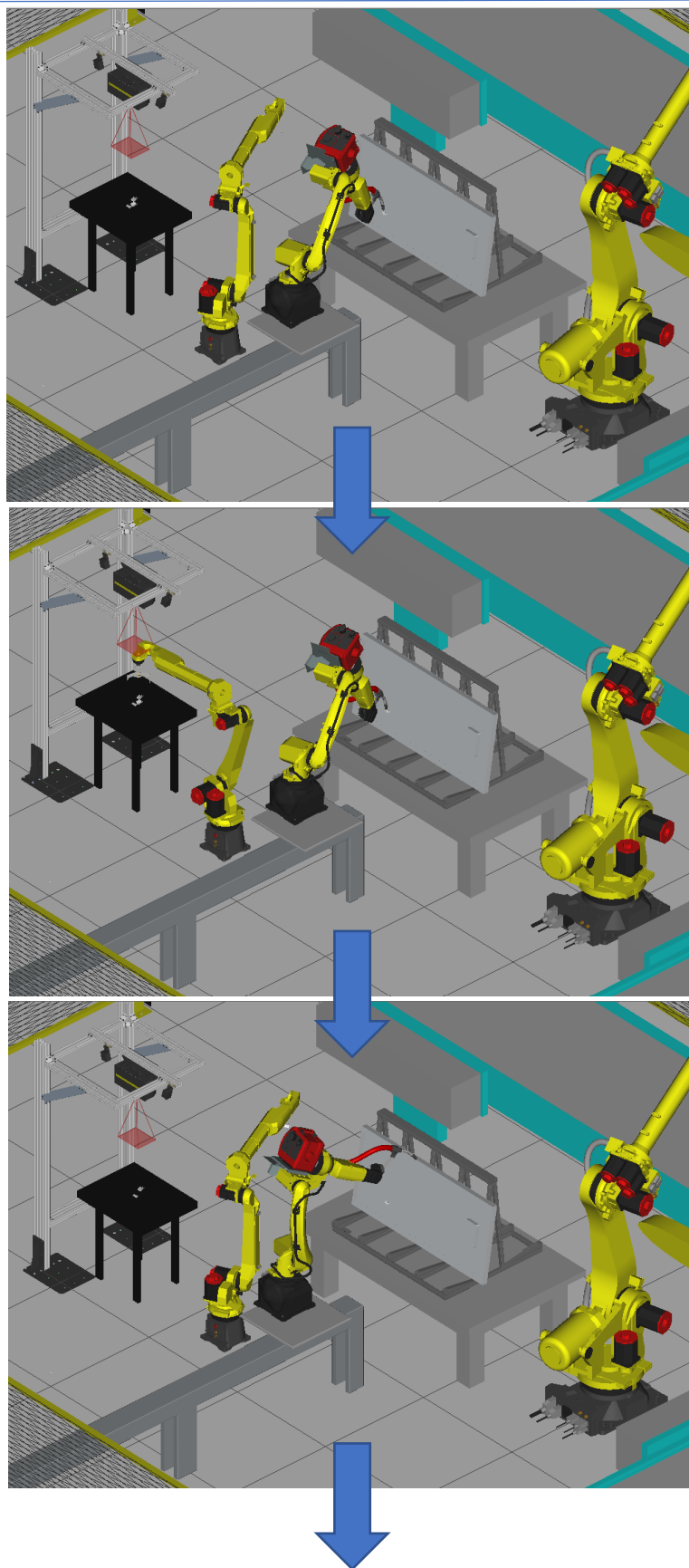
- Cuando la placa esté colocada en el soporte de la mesa de soldadura, el Robot3 (estando en su posición de reposo) inicia el proceso de visión artificial de reconocimiento de las cuatro piezas de fijación. Este proceso solo se realiza cuando las cuatro están sobre la mesa. Cuando las ha

reconocido, recoge una. Vuelve a su posición de 'HOME'. La deposita en su posición respectiva. Le comunica al Robot2 qué pieza debe soldar y le da la orden de soldadura. Finalmente, mientras el Robot2 va a soldar, el Robot3 retorna a su posición de reposo evitando una colisión con el Robot2.

- Cuando al Robot2 se le ordena soldar, este ya estaba esperando dicha orden en una posición intermedia del carril ("pos_intermedia") a la que se desplazó tras realizar el cambio a modo automático. Tras recibir la orden de soldadura, el Robot3 se desplaza por el carril hasta la denominada "pos_soldar". Como el Robot2 ya le ha comunicado el lugar donde debe soldar, este va a la posición origen del 'user frame' creado para esa pieza en particular. Realiza cuatro movimientos lineales por el contorno de la pieza simulando la soldadura. Retorna a "pos_soldar". Ordena al Robot3 que tome otra pieza de las reconocidas. Y retorna a "pos_intermedia". Se adjuntan los resultados de esta maniobra a continuación:







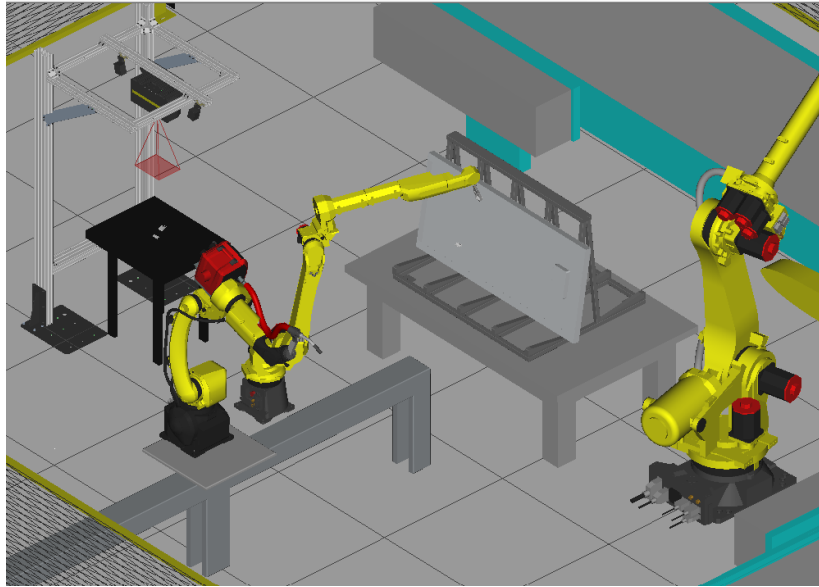
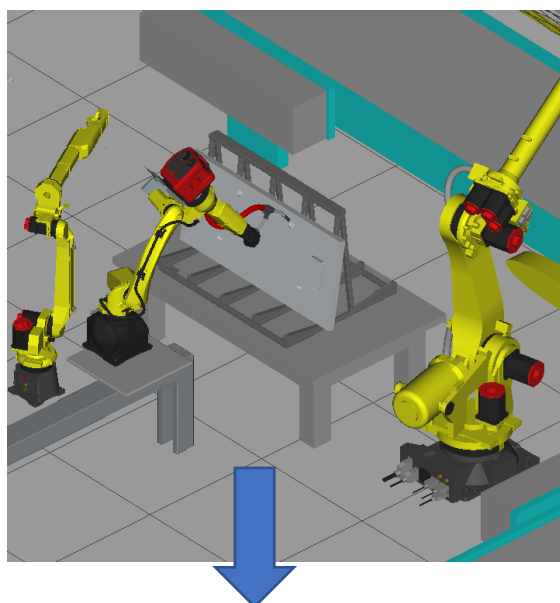
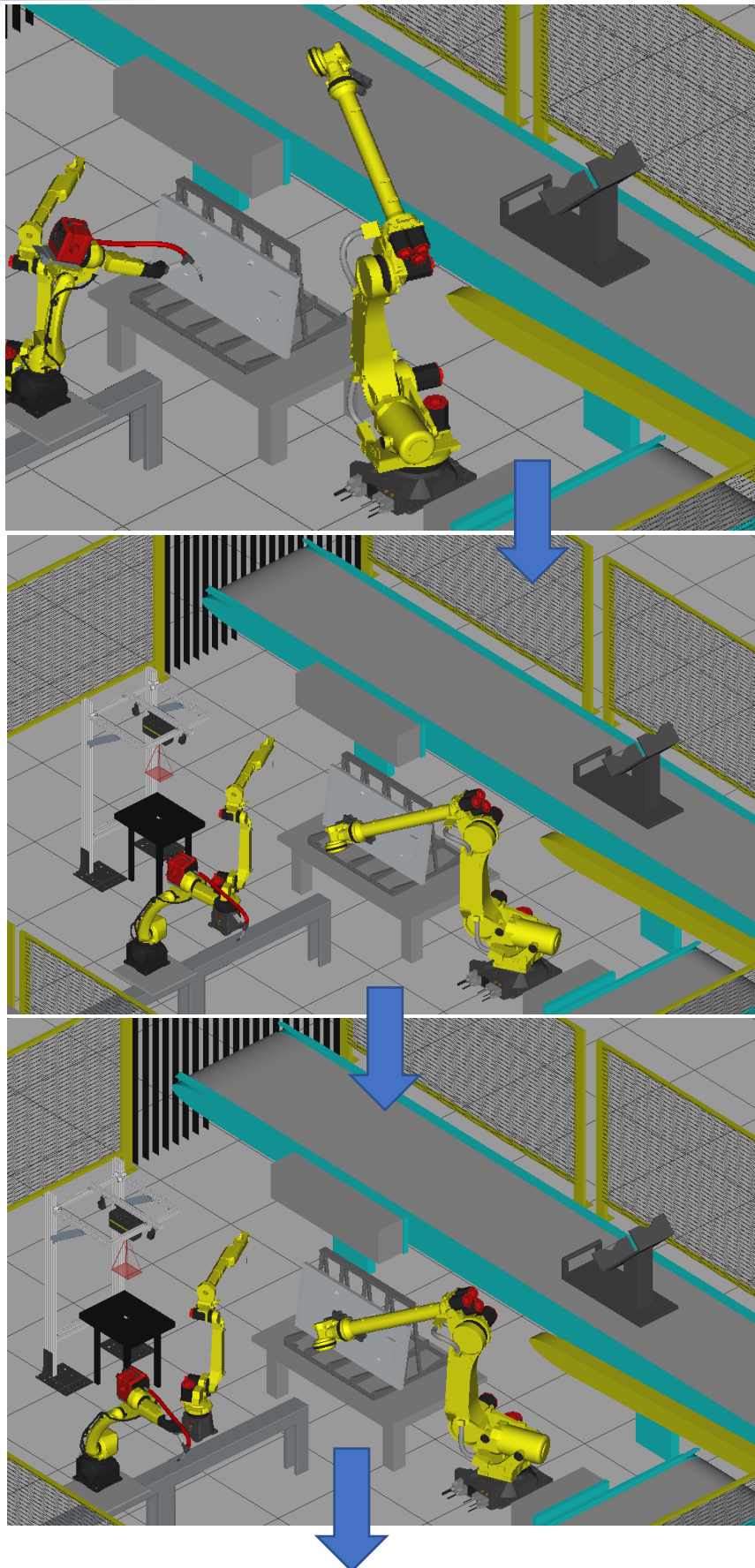
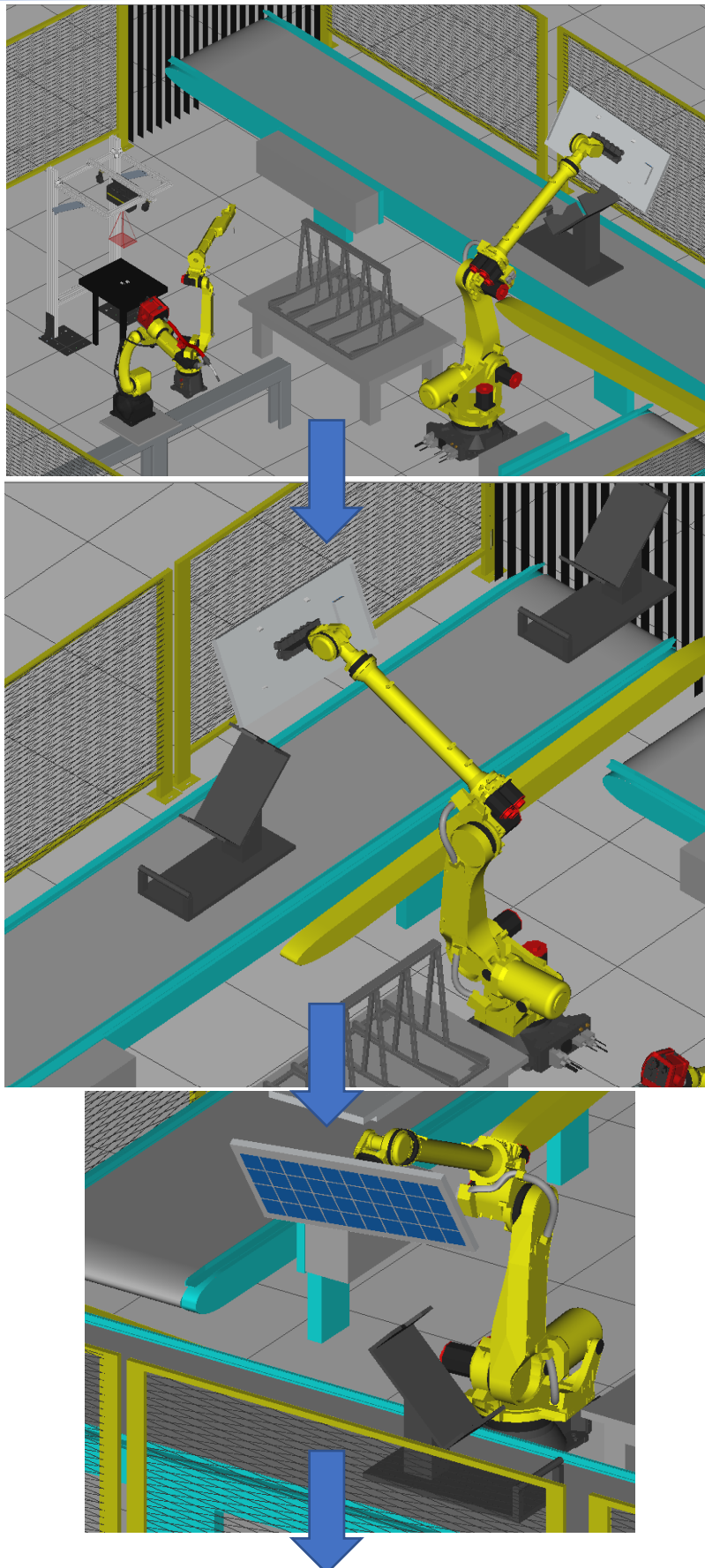


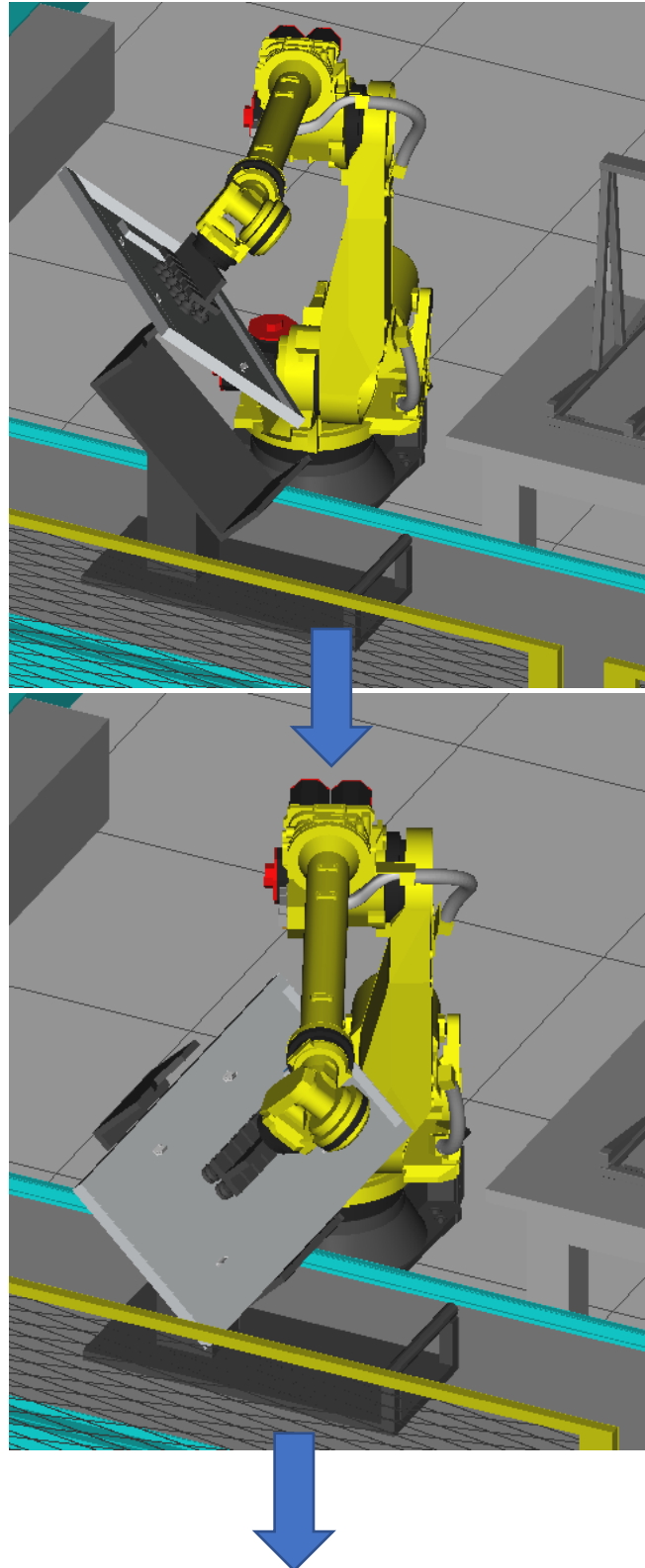
FIGURA 174. Secuencia de movimientos de los robots 2 y 3.

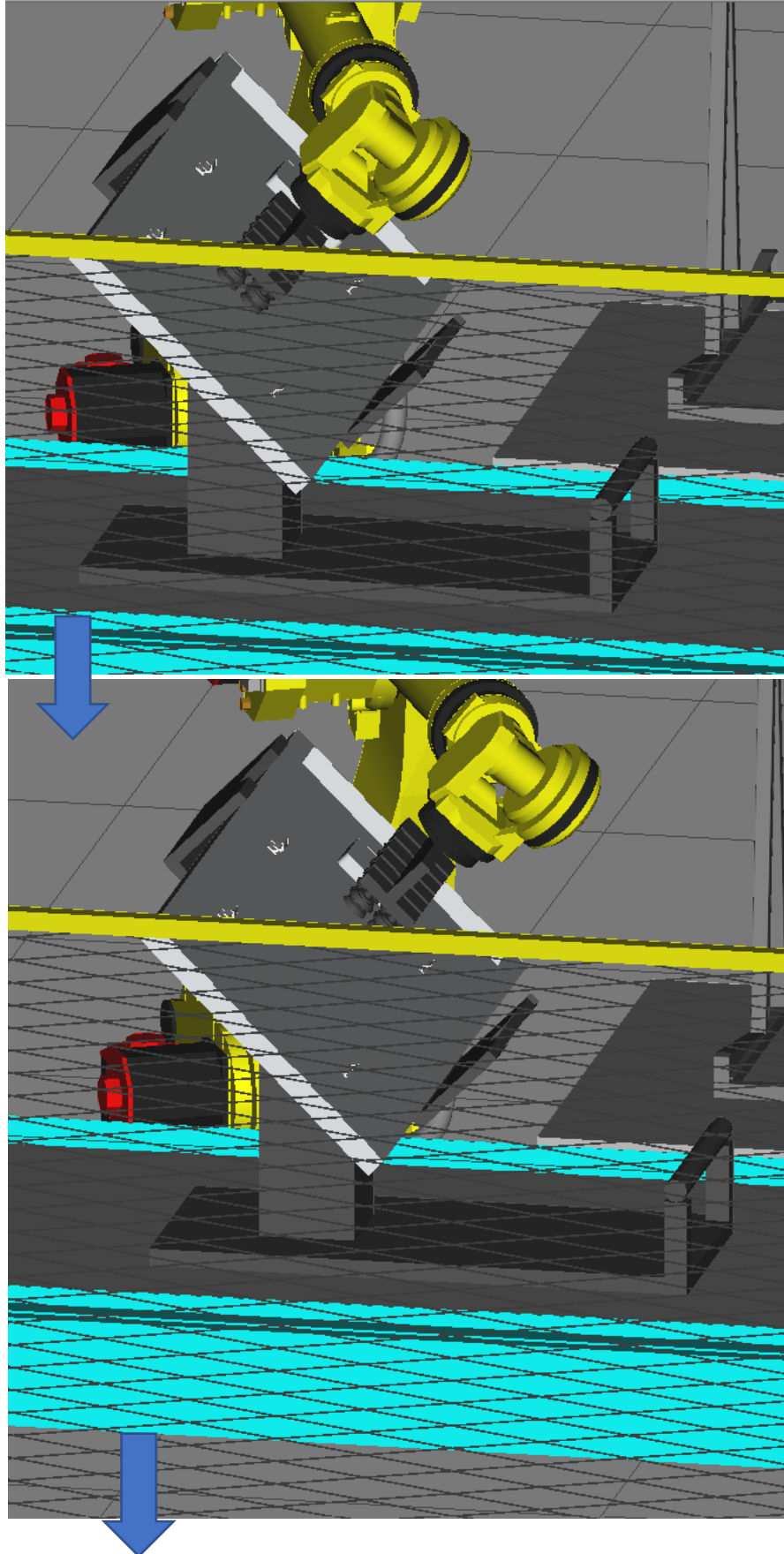
- Los dos puntos anteriores se repiten hasta que no quedan piezas que soldar.
- Así pues, cuando se ha terminado la soldadura de la última pieza, el Robot2 ordena al Robot1 tomar la placa ya tratada.
- El Robot1, que esperaba en su posición de reposo, va a tomar la placa situada en la mesa de soldadura. La toma y vuelve a la posición de reposo. Con ella agarrada, espera a que el soporte que viene por la línea de salida esté en el punto de depositar la placa sobre él. Para ello realiza 'Line Tracking' a la línea de salida. Deposita la placa, y vuelve a su posición 'HOME'. Se adjuntan los resultados de esta maniobra a continuación:











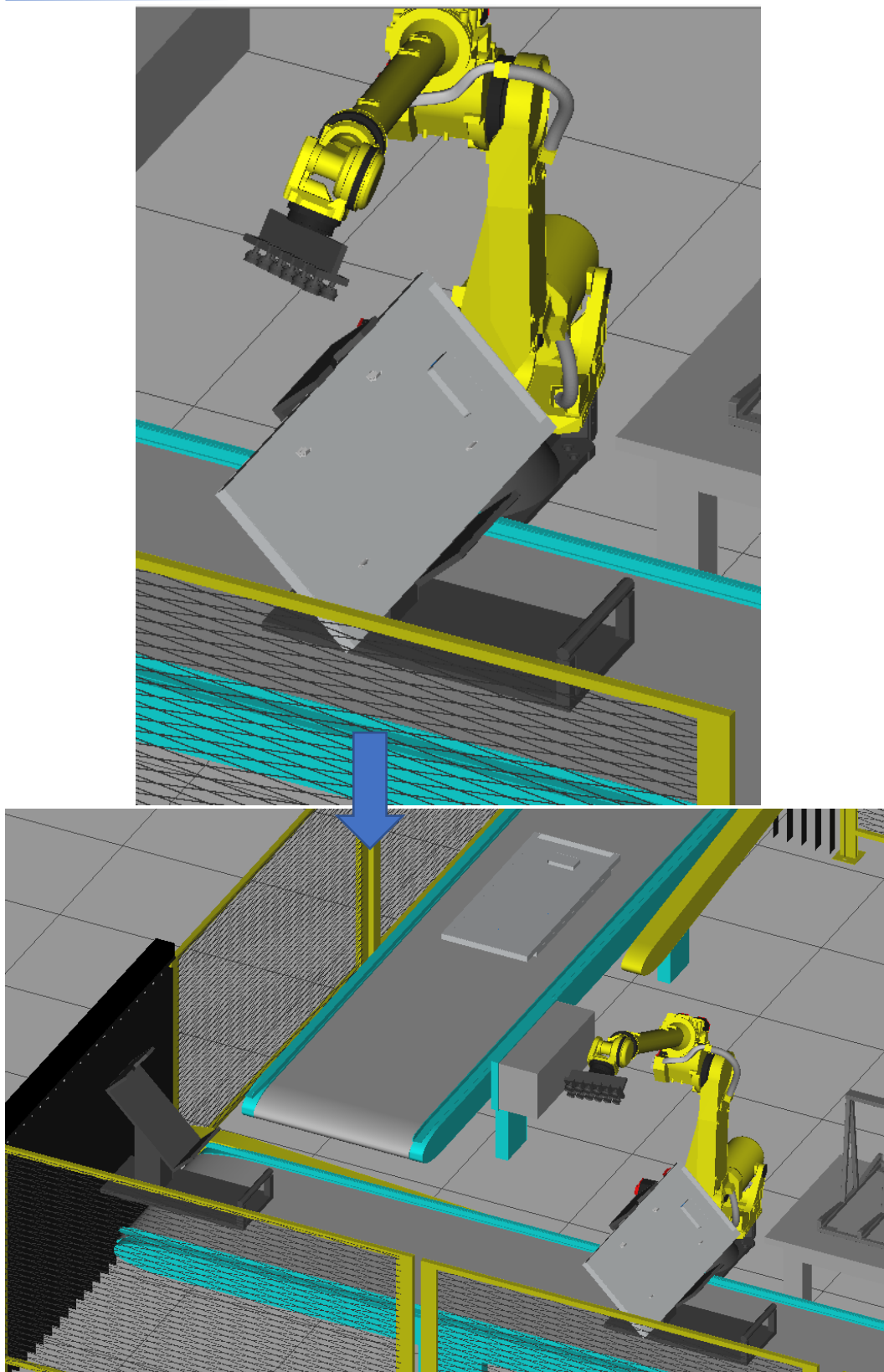


FIGURA 175. Secuencia de movimientos del Robot1 (II).



- Este sería el final de un ciclo de trabajo. Se volvería al punto inicial donde el propio Robot1 espera la llegada de otra placa.
- 4) Este modo automático está pensado para realizarse de continuo en una planta real. Por lo que, para volver al modo manual, se debe realizar una parada de emergencia debe haber sucedido un error.





CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS.

CAPÍTULO 7.1. Conclusiones.

A partir del trabajo realizado se pueden sacar varias conclusiones. La primera es la consecución de los objetivos fijados al inicio del documento:

Se ha proporcionado una herramienta que posibilita el estudio de celdas robotizadas simuladas que trabajan en sintonía con software de diferentes fabricantes con los que no tienen conectividad nativa. Para ello se ha empleado una estándar de comunicaciones común a estos (OPC UA).

Mediante la herramienta KEPServerEX, se ha podido tener una toma de contacto con el mundo de las comunicaciones industriales. Se ha creado un servidor muy sencillo que emplea la tecnología Ethernet como canal de comunicación utilizando el modelo TCP/IP. El servidor permitió el intercambio de datos entre la célula robotizada y sus dispositivos de control.

Se ha desarrollado una estación robotizada compleja en HandlingPro de Fanuc que coordina el funcionamiento de varios robots para la realización de un proceso industrial ficticio; pero que resulta de gran utilidad académicamente debido a la gran variedad de aspectos interesantes que se han tenido en cuenta a la hora del desarrollo del proyecto: técnicas avanzadas de programación de robots como el '*Line Tracking*', implementación de un eje extendido, intercambio de señales entre robots para una coordinación eficiente, visión artificial y el uso de marcos de trabajo para facilitar la ejecución de movimientos.

Esta estación se controla con un autómata de Siemens que utiliza el software TIA Portal para su configuración. Así que se han recordado conceptos de programación KOP en este software adquiridos durante el grado.

A mayores la estación se gestiona a nivel de usuario con una HMI también de la marca Siemens que, en consecuencia, se debe configurar en el software TIA Portal. Este software aporta una muy amplia variedad de funcionalidades y elementos de los cuales se han empleado unos pocos en este proyecto. Gracias a esta variedad, las posibilidades en cuanto a la creación y programación de interfaces es más que notable – lo que potencia la creatividad de los alumnos.

El proyecto engloba varias de las tecnologías de la industria 4.0 como la robótica, la visión artificial, gemelos digitales y la interconectividad entre dispositivos.



Este mismo documento se puede emplear como aporte para el aprendizaje en el desarrollo de celdas robotizadas empleando software de la marca Fanuc, ya que en la Universidad de Valladolid no hay una variedad al nivel del de las celdas robotizadas desarrolladas en el entorno de la marca ABB.

Por último, el proyecto puede servir como punto de partida para futuros proyectos sobre el que realizar cambios y mejoras para ampliar el conocimiento sobre las herramientas utilizadas. Estos pueden darse en el modo de programación de los robots, en las funcionalidades de la HMI...

CAPÍTULO 7.2. Líneas futuras.

En relación con los cambios y mejoras que se pueden dar en base al trabajo realizado mencionado anteriormente, varios ejemplos podrían ser:

- 1) **Seguimiento de línea circular.** El '*Line Tracking*' utilizado en el proyecto es únicamente lineal. Sin embargo, dentro de HandlingPro existen cintas transportadoras de geometría curva a las cuales se les puede hacer un seguimiento.
- 2) **Lenguaje de programación.** Podría cambiarse el tipo de programación de los robots al lenguaje KAREL. Y del propio PLC, ya que TIA Portal permite la programación de sus autómatas en varios lenguajes como AWL, GRAFCET, SCL...
- 3) **Implementación de entornos seguros.** Esta mejora de la estación es gracias a la herramienta '*Dual Check Safety*', la cual no se ha utilizado en este proyecto, se pueden crear zonas de seguridad en torno a cada robot.
- 4) **Ampliación de la estación.** Tras esta parte del proceso, se podría realizar un paletizado de las placas ya tratadas por medio de un robot que primeramente examine mediante visión artificial que las placas no sufren ningún daño por ninguna de sus caras.



BIBLIOGRAFÍA

- [1] A. Barrientos, FUNDAMENTOS DE ROBÓTICA, Madrid: McGraw-Hill, 1997.
- [2] Federación de Enseñanza de C.C.O.O. de Andalucía, «www.feandalucia.ccoo.es,» [En línea]. Available: <https://www.feandalucia.ccoo.es/docu/p5sd7206.pdf#:~:text=Los%20P%C3%A1jaros%20de%20Her%C3%B3n.%20En%20el%20a%C3%B1o%202022,por%20encontrarles%20aplicaci%C3%B3n%202.4.-%20EL%20TEMPLO%20DE%20HER%C3%93N..> [Último acceso: 29 Octubre 2022].
- [3] «krikpit.com,» [En línea]. Available: <https://krikpit.com/caballero-autmata/#:~:text=El%20Caballero%20aut%C3%B3mata%20%28a%20veces%20tambi%C3%A9n%20llamado%20robot,no%20se%20sabe%20si%20se%20hizo%20o%20no..> [Último acceso: 19 Octubre 2022].
- [4] Wikipedia, «es.wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Telar_de_Jacquard. [Último acceso: 19 octubre 2022].
- [5] A. Brunete, P. San Segundo y R. Herrero, Introducción a la Automatización Industrial, Madrid, 2020.
- [6] A. E. Murillo Sánchez, «CTIN,» 25 10 2013. [En línea]. Available: <http://www.ctinmx.com/que-es-un-plc/>. [Último acceso: 17 Noviembre 2022].
- [7] J. Pérez Turiel, *Buses de Campo y Ethernet Industrial*, Valladolid, 2021.
- [8] OPC, «opcfoundation.org,» [En línea]. Available: <https://opcfoundation.org/about/what-is-opc/>. [Último acceso: 22 octubre 2022].
- [9] UNIR - La Universidad en Internet, «www.unir.net,» [En línea]. Available: <https://www.unir.net/ingenieria/revista/vision-artificial/>. [Último acceso: 5 Noviembre 2022].
- [10] J. A. Ávila Herrero y A. Herreros López, «UVaDOC,» Septiembre 2015. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/14441>. [Último acceso: 25 Noviembre 2022].



- [11] Á. Galindo de Santos y A. Herreros López, «UVaDOC,» Enero 2016. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/17056>. [Último acceso: 25 Noviembre 2022].
- [12] J. Delgado García y A. Herreros López, «UVaDOC,» Mayo 2020. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/41133>. [Último acceso: 25 Noviembre 2022].
- [13] A. Martínez Rodríguez y A. Herreros López, «UVaDOC,» Junio 2022. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/53922>. [Último acceso: 28 Agosto 2022].
- [14] R. Vidal del Cura y A. Herreros López, «UVaDOC,» Junio 2022. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/53931>. [Último acceso: 15 Agosto 2022].
- [15] FANUC, «www.fanuc.eu,» [En línea]. Available: <https://www.fanuc.eu/es/es/robots/accesorios/roboguide>. [Último acceso: 11 junio 2022].
- [16] InfoPLC, «www.infopl.net,» [En línea]. Available: <https://www.infopl.net/descargas/47-fanuc-robotica/1973-programaci%C3%B3n-b%C3%A1sica-de-robots-fanuc-seat>. [Último acceso: 13 Junio 2022].
- [17] «www.boe.es,» Parlamento Europeo y del Consejo, 9 Junio 2006. [En línea]. Available: <https://boe.es/buscar/doc.php?id=DOUE-L-2006-81063>. [Último acceso: 28 Julio 2022].
- [18] UNE, «une.org,» ISO, 5 Febrero 2014. [En línea]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=N0049289>. [Último acceso: 28 Julio 2022].
- [19] UNE, «une.org,» ISO, 28 Septiembre 2016. [En línea]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0048668>. [Último acceso: 28 Julio 2022].
- [20] UNE - Una Norma Española, «www.une.org,» ISO, [En línea]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/proyecto?c=P0054712>. [Último acceso: 28 Julio 2022].
- [21] UNE - Una Norma Española, «www.une.org,» ISO, 6 Abril 2016. [En línea]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=norma-une-en-iso-14120-2016-n0056446>. [Último acceso: 17 Agosto 2022].



- [22] FANUC, Line Tracking Setup and Operations Manual, 2007.
- [23] FANUC, «www.fanuc.eu,» [En línea]. Available:
<https://www.fanuc.eu/es/es/robots/p%c3%a1gina-filtro-robots/serie-r-2000/r-2000ic-125l>. [Último acceso: 15 junio 2022].
- [24] FANUC, «www.fanuc.eu,» [En línea]. Available:
<https://www.fanuc.eu/es/es/robots/p%c3%a1gina-filtro-robots/soldadura-por-arco/arcmate-100id>. [Último acceso: 15 Junio 2022].
- [25] FANUC, «www.fanuc.eu,» [En línea]. Available:
<https://www.fanuc.eu/es/es/robots/p%c3%a1gina-filtro-robots/serie-m-10/m-10id-8l>. [Último acceso: 15 Junio 2022].
- [26] SIEMENS, «mall.industry.siemens.com,» [En línea]. Available:
<https://mall.industry.siemens.com/mall/es/es/Catalog/Product/6AV2124-0GC01-0AX0>. [Último acceso: 10 Septiembre 2022].
- [27] SIEMENS, «new.siemens.com,» [En línea]. Available:
<https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html>. [Último acceso: 16 Junio 2022].



Universidad de Valladolid

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Jorge Villa González



ESCUELA DE INGENIERÍAS
INDUSTRIALES

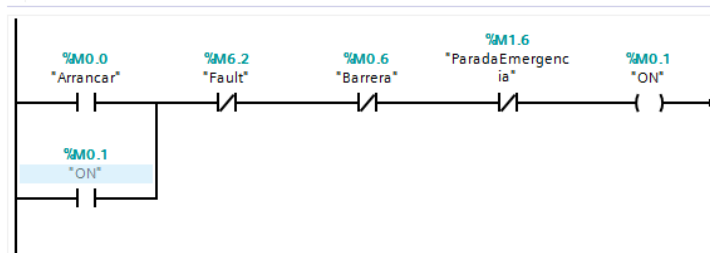
ANEXOS

ANEXO A. BLOQUES DE PROGRAMACIÓN DE TIA PORTAL.

Main (OB1)

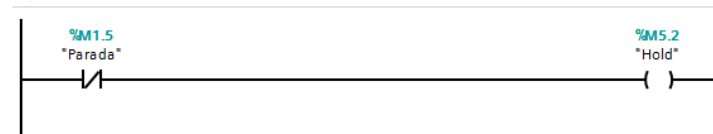
▼ Segmento 1: Arranque de la estación

- ▼ Cuando se pulse el botón de arrancar, la estación se ejecutará. El estado de ejecución se mantiene, solo se interrumpe si hay algún problema. Cualquiera de estos abortará la ejecución.



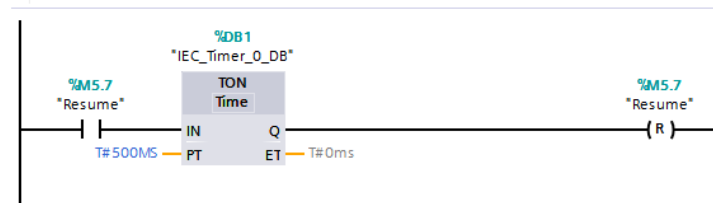
▼ Segmento 2: Parada controlada de la estación

- ▼ Cuando se pulse el botón de "paro" y se entre en estado de "parada", caerá la variable "Hold*" de Roboguide. Y el programa se detendrá. Se podrá reanudar.



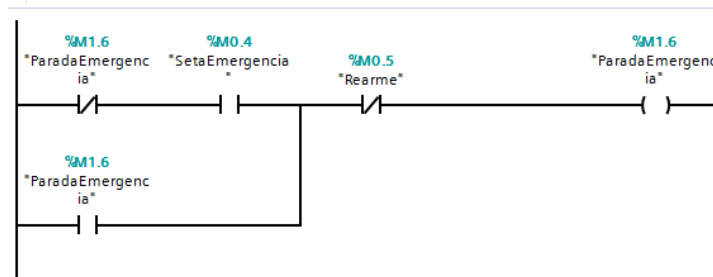
▼ Segmento 3: Reanudación tras parada controlada

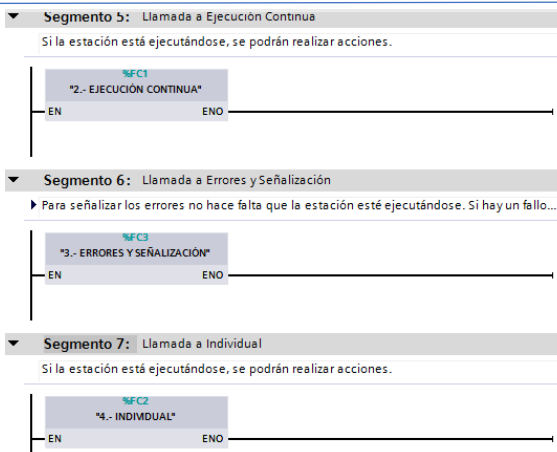
- ▼ Cuando estemos en parada controlada, el programa estará en pausa (UO4 Roboguide -> ON). Pero al pulsar "Resume", esta variable debe cambiar a OFF.



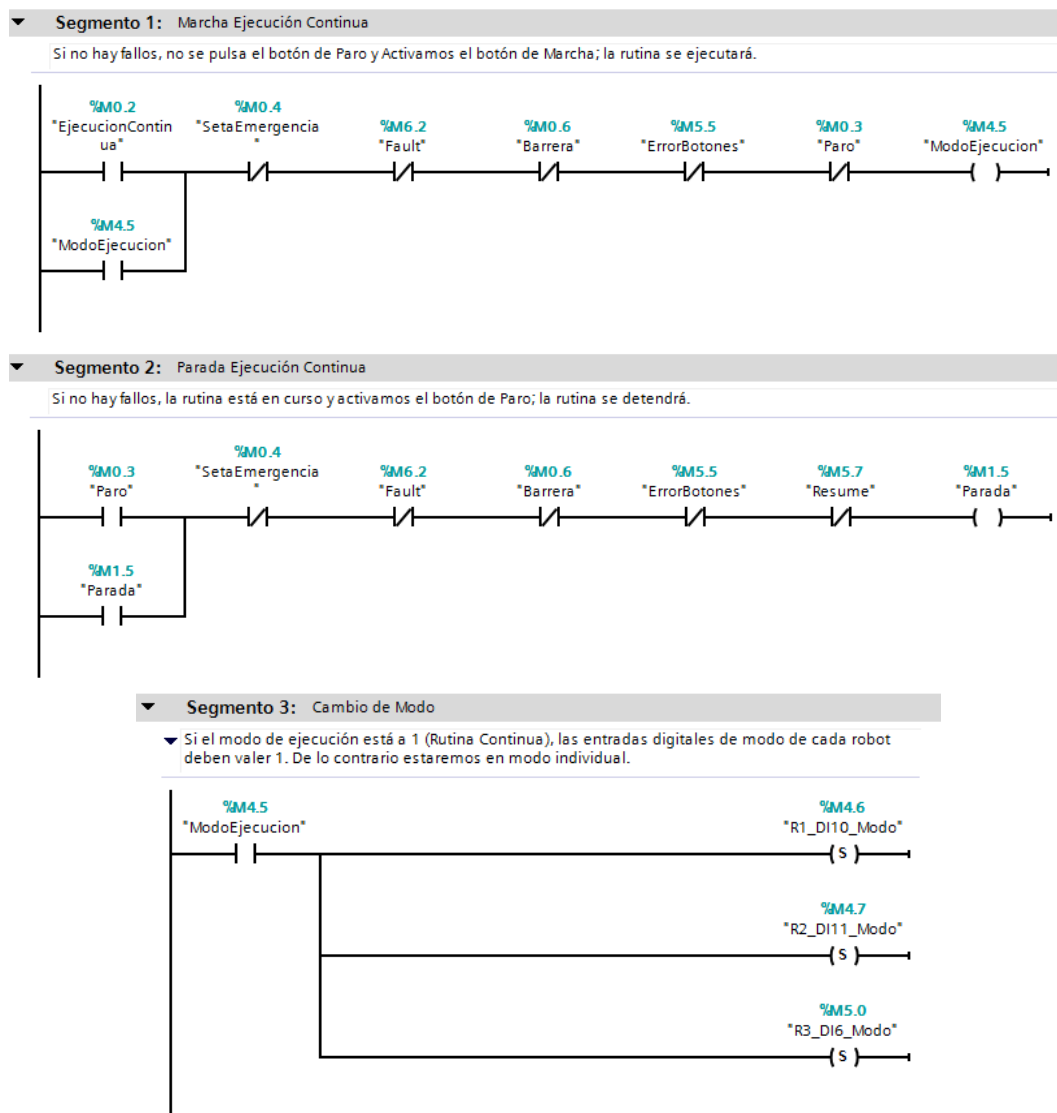
▼ Segmento 4: Parada de Emergencia

- ▶ Si no estamos en parada de emergencia (parado o funcionando) y se pulsa la seta, se enclavará h...

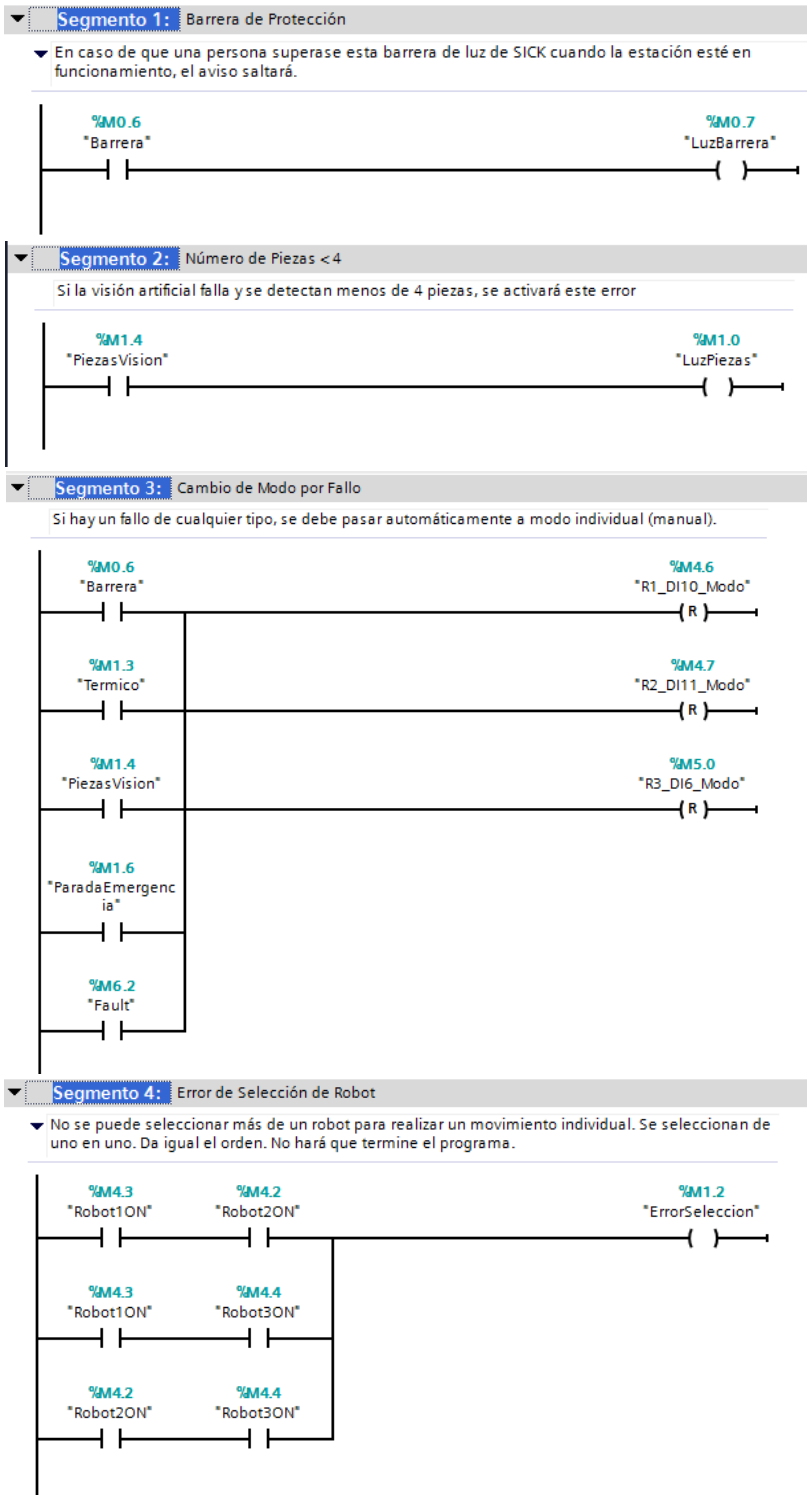




Ejecución Continua (FC1)

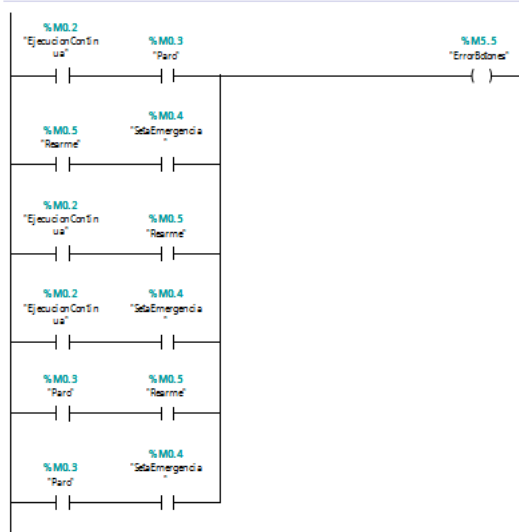


Errores y Señales (FC2)



Segmento 5: Error de Selección de Botones de Ejecución Continua

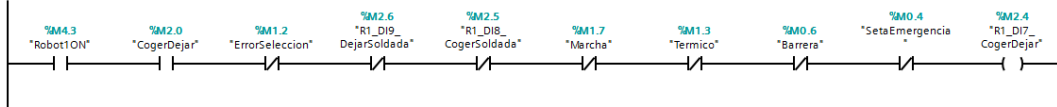
No se podrá presionar dos o más botones simultáneamente en ejecución continua.



Individual (FC3)

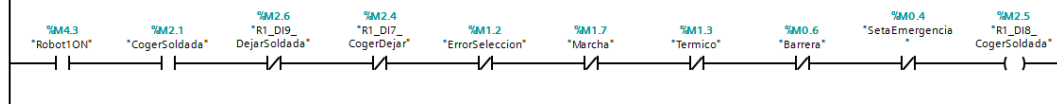
Segmento 1: 4.- Individual: Robot 1 - Cogery Dejar Placa sin Soldar

Si se selecciona el robot 1. Se selecciona coger placa y dejarla en el soporte para ser tratada, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.



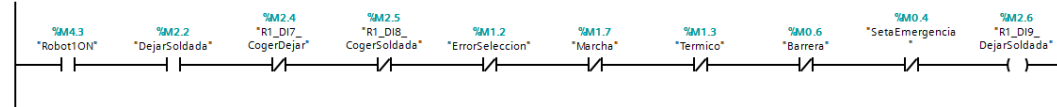
Segmento 2: 4.- Individual: Robot 1 - Coger Placa Soldada

Si se selecciona el robot 1. Se selecciona coger placa ya tratada, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.



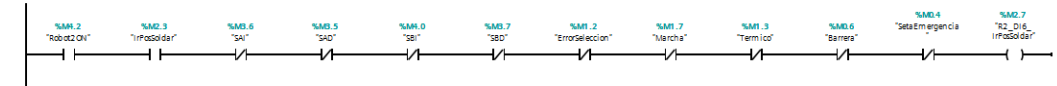
Segmento 3: 4.- Individual: Robot 1 - Dejar Placa Soldada

Si se selecciona el robot 1. Se selecciona dejar placa en el soporte para salir de la estación, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.



Segmento 4: 4.- Individual: Robot 2 - Ira Pos Soldar

Si se selecciona el robot 2. Se selecciona ir a pos_soldar, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.



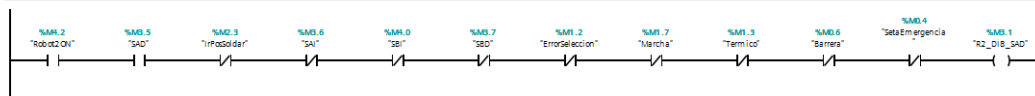
Segmento 5: 4.- Individual: Robot 2 - Soldar Arriba Izquierda

Si se selecciona el robot 2. Se selecciona soldar arriba izquierda, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.

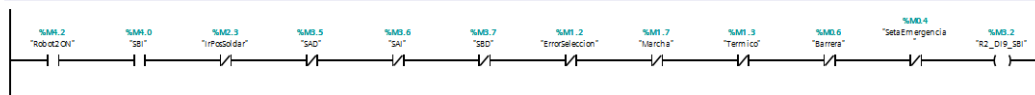




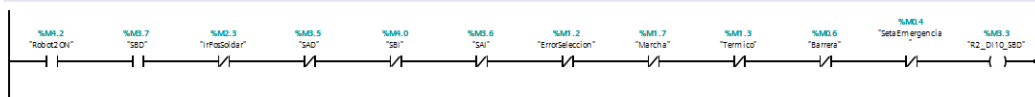
Segmento 6: 4.- Individual: Robot 2 - Soldar Arriba Derecha
 Si se selecciona el robot 2. Se selecciona soldar arriba derecha, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.



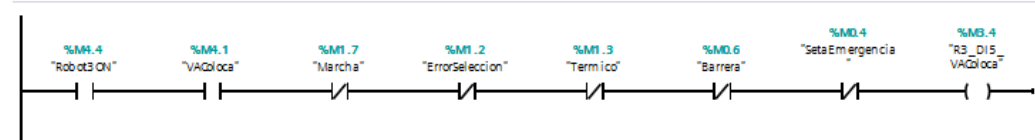
Segmento 7: 4.- Individual: Robot 2 - Soldar Abajo Izquierda
 Si se selecciona el robot 2. Se selecciona soldar abajo izquierda, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.



Segmento 8: 4.- Individual: Robot 2 - Soldar Abajo Derecha
 Si se selecciona el robot 2. Se selecciona soldar abajo derecha, no hay fallos y no se ha seleccionado otra acción entonces realizará la acción seleccionada.

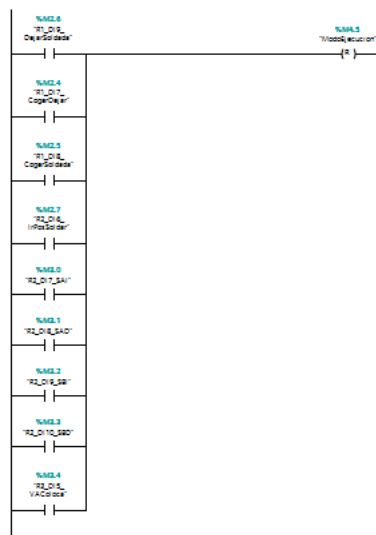


Segmento 9: 4.- Individual: Robot 3 - Visión Artificial y Colocar Piezas
 Si se selecciona el robot 3. Se selecciona la única opción disponible y no hay fallos, entonces se realiza dicha acción.



Segmento 10: Reseteo del bit de Modo de Ejecución

Si estamos en cualquier situación de la ejecución individual, el bit ModoEjecución debe valer 0. En caso de parada de emergencia u e...





ANEXO B. CÓDIGOS DE PROGRAMACIÓN EN TEACH PENDANT.

Robot1.- MAIN

Programa que ejecuta el Robot1 en bucle infinito. Según el valor de la DI10, entrará en modo manual o automático.

```
1: LBL[1]
2: IF DI[10:MODO IND*/AUT]=OFF,CALL INDIVIDUAL
3: IF DI[10:MODO IND*/AUT]=ON,CALL RUTINAAUT
4: JMP LBL[1]
```

Robot1.- INDIVIDUAL

Si DI10 = 0, entrará en modo manual. El robot queda a la espera de ejecutar cualquier movimiento – espera la activación de la DI7, DI8 o DI9.

```
1: LINE[1] ON
2: LINE[2] ON
3:
4: J PR[4:HomeInd] 100% CNT25
5:
6: IF (DI[7:COGER Y DEJAR PLACA]=ON) THEN
7: JMP LBL[2]
8: ENDIF
9:
10: IF (DI[8:COGER PLACA SOLDADA]=ON) THEN
11: JMP LBL[3]
12: ENDIF
13:
14: IF (DI[9:DEJAR PLACA SOLDADA]=ON) THEN
15: JMP LBL[4]
16: ELSE
17: JMP LBL[5]
18: ENDIF
19:
20: LBL[2]
21: LBL[30]
22: WAIT DI[2:LINE TRACKING ENTRADA]=ON
23: LINECOUNT[2] R[3]
24: LINERATE[2] R[4]
25: IF R[4]<1,JMP LBL[30]
26:
27: SETTRIG LNSCH[2] R[3]
28: SELBOUND LNSCH[2] BOUND[2]
29:
30: CALL PICKUP
31: J PR[4:HomeInd] 100% CNT25
32: J P[1] 100% CNT25
```



```
33: CALL DROP
34: J P[1] 100% CNT25
35: JMP LBL[5]
36:
37: LBL[3]
38: J P[1] 100% CNT25
39: CALL PICKUPSOLDADO
40: J P[1] 100% CNT25
41: J PR[4:HomeInd] 100% CNT25
42: JMP LBL[5]
43:
44: LBL[4]
45: LBL[20]
46: WAIT DI[1:LINE TRACKING SALIDA]=ON
47: LINECOUNT[1] R[1]
48: LINERATE[1] R[2]
49: IF R[2]<1, JMP LBL[20]
50:
51: SETTRIG LNSCH[1] R[1]
52: SELBOUND LNSCH[1] BOUND[1]
53:
54: CALL DROPSOLDADO
55: LBL[5]
```

Robot1.- RUTINA AUT

Si DI10 = 1, entrará en modo automático.

```
1: DO[5]=OFF
2: LINE[1] ON
3: LINE[2] ON
4: J P[3] 100% FINE
5: LBL[10]
6: LBL[30]
7: WAIT DI[2:LINE TRACKING ENTRADA]=ON
8: LINECOUNT[2] R[3]
9: LINERATE[2] R[4]
10: IF R[4]<1, JMP LBL[30]
11:
12: SETTRIG LNSCH[2] R[3]
13: SELBOUND LNSCH[2] BOUND[2]
14: CALL PICKUP
15:
16: J P[3] 100% FINE
17: J P[2] 100% CNT25
18: CALL DROP
19: DO[5]=ON
20: J P[2] 100% FINE
21: J P[3] 100% CNT25
22:
23: DO[5]=OFF
24: WAIT DI[6:FIN SOLDADURA]=ON
25: LBL[60]
```



```
26: J P[2] 100% CNT100
27: CALL PICKUPSOLDADO
28: J P[2] 100% CNT100
29: J P[3] 100% FINE
30:
31: LBL[20]
32: WAIT DI[1:LINE TRACKING SALIDA]=ON
33: LINECOUNT[1] R[1]
34: LINERATE[1] R[2]
35: IF R[2]<1, JMP LBL[20]
36:
37: SETTRIG LNSCH[1] R[1]
38: SELBOUND LNSCH[1] BOUND[1]
39:
40: J P[3] 100% CNT25
41: CALL DROPSOLDADO
42:
43: J P[3] 100% CNT25
44:
45: JMP LBL[10]
```

Robot1.- PICKUP

Subrutina de tomar placa activada tras el 'Line Tracking'.

```
1: L P[1] 3000mm/sec FINE Tool_Offset,PR[2:OffsetZ]
2: L P[1] 100mm/sec FINE
3: WAIT .50(sec)
4: CALL PICKUP_SIM
5: L P[1] 100mm/sec FINE Tool_Offset,PR[2:OffsetZ]
```

Robot1.- DROP

Subrutina de dejar placa en el soporte de la mesa de trabajo.

```
1: J P[1] 100% FINE Tool_Offset,PR[2:OffsetZ]
2: L P[1] 3000mm/sec FINE
3: CALL DROP_SIM
4: L P[1] 3000mm/sec FINE Tool_Offset,PR[2:OffsetZ]
```

Robot1.- PICKUP_SOLDADO

Subrutina de tomar placa ya tratada del soporte de la mesa de trabajo.

```
1: J P[1] 100% FINE Tool_Offset,PR[2:OffsetZ]
2: L P[1] 2000mm/sec FINE
3: CALL PICKUPSOLDADO_SIM
4: CALL PICKUPPIEZAS1_SIM
5: CALL PICKUPPIEZAS2_SIM
6: L P[1] 2000mm/sec FINE Tool_Offset,PR[2:OffsetZ]
```



Robot1.- DROP_SOLDADO

Subrutina de dejar placa en el soporte de la línea de salida tras el 'Line Tracking'.

```
1: L P[3] 3000mm/sec FINE Tool_Offset,PR[2:OffsetZ]
2: L P[3] 3000mm/sec FINE Tool_Offset,PR[3:OffsetZ2]
3: WAIT .25(sec)
4: CALL DROPSOLDADO_SIM
5: L P[3] 3000mm/sec FINE Tool_Offset,PR[2:OffsetZ]
```

ROBOT2.- MAIN

Programa que ejecuta el Robot2 en bucle infinito. Según el valor de la DI11, entrará en modo manual o automático.

```
1: LBL[1]
2: IF DI[11:MODO IND*/AUT]=OFF,CALL INDIVIDUAL
3: IF DI[11:MODO IND*/AUT]=ON,CALL RUTINAAUT
4: JMP LBL[1]
```

ROBOT2.- INDIVIDUAL

Si DI11 = 0, entrará en modo manual. El robot queda a la espera de ejecutar cualquier movimiento – espera la activación de la DI6, DI7, DI8, DI9 o DI10.

```
1: LBL[7]
2: R[1:contador]=0
3: R[2:uframe]=0
4: UFRAME_NUM=R[2:uframe]
5:
6: IF DI[6:IR A POSSOLDAR]=ON,JMP LBL[2]
7:
8: IF DI[7:SAI IND]=ON,JMP LBL[3]
9:
10: IF DI[8:SAD IND]=ON,JMP LBL[4]
11:
12: IF DI[9:SBI IND]=ON,JMP LBL[5]
13:
14: IF DI[10:SBD IND ]=ON,JMP LBL[6]
15:
16: IF R[1:contador]=4,JMP LBL[7]
17:
18: JMP LBL[1]
19:
20: LBL[2]
21: J PR[2:pos_soldar] 100% FINE
22: JMP LBL[1]
23:
```




```
24: LBL[3]
25: R[2:uframe]=1
26: CALL SOLDARARRIBA
27: R[1:contador]=R[1:contador]+1
28: JMP LBL[1]
29:
30: LBL[4]
31: R[2:uframe]=2
32: CALL SOLDARARRIBA
33: R[1:contador]=R[1:contador]+1
34: JMP LBL[1]
35:
36: LBL[5]
37: R[2:uframe]=3
38: CALL SOLDARABAJO
39: R[1:contador]=R[1:contador]+1
40: JMP LBL[1]
41:
42: LBL[6]
43: R[2:uframe]=4
44: CALL SOLDARABAJO
45: R[1:contador]=R[1:contador]+1
46: JMP LBL[1]
47:
48: LBL[1]
49: R[2:uframe]=0
50: UFRAME_NUM=R[2:uframe]
51: J PR[1:homepos] 100% CNT25
```

ROBOT2.- RUTINA AUT

Si DI11 = 1, entrará en modo automático.

```
1: R[1:contador]=0
2: R[2:uframe]=0
3: UFRAME_NUM=0
4: J PR[1:homepos] 100% FINE
5:
6: LBL[10]
7: DO[3]=OFF
8: R[2:uframe]=0
9: J PR[11:pos_intermedia] 100% CNT100
10:
11: WAIT DI[1:IR POSSOLDAR AUT]=ON
12: J PR[2:pos_soldar] 100% FINE
13:
14: IF (DI[2:SAI AUT]=ON) THEN
15: R[2:uframe]=1
16: CALL SOLDARARRIBA
17: R[1:contador]=R[1:contador]+1
18: ENDIF
19:
20: IF (DI[3:SAD AUT]=ON) THEN
```



```
21: R[2:uframe]=2
22: CALL SOLDARARRIBA
23: R[1:contador]=R[1:contador]+1
24: ENDIF
25:
26: IF (DI[4:SBI AUT]=ON) THEN
27: R[2:uframe]=3
28: CALL SOLDARABAJO
29: R[1:contador]=R[1:contador]+1
30: ENDIF
31:
32: IF (DI[5:SBD AUT]=ON) THEN
33: R[2:uframe]=4
34: CALL SOLDARABAJO
35: R[1:contador]=R[1:contador]+1
36: ENDIF
37:
38: UFRAME_NUM=0
39: IF R[1:contador]=4, JMP LBL[80]
40: JMP LBL[10]
41:
42: LBL[80]
43: DO[3]=ON
44: DO[10]=ON
45: R[1:contador]=0
46: J PR[1:homepos] 100% FINE
47: WAIT 2.00(sec)
48: JMP LBL[10]
```

ROBOT2.- SOLDARABAJO

Subrutina de soldar las piezas de tipo 1 que van sobre la parte inferior de la placa.
En función de las piezas de tipo 1 ya colocadas, el 'user frame' es uno u otro.

```
1: UFRAME_NUM=R[2:uframe]
2: J PR[2:pos_soldar] 100% FINE
3: L PR[13:origenpiezal] 2000mm/sec FINE
4: L PR[13:origenpiezal] 2000mm/sec FINE Offset,PR[14:x_offset1]
5: L PR[13:origenpiezal] 2000mm/sec FINE Offset,PR[15:xy_offset1]
6: L PR[13:origenpiezal] 2000mm/sec FINE Offset,PR[16:y_offneg1]
7: L PR[13:origenpiezal] 2000mm/sec FINE
8: L PR[13:origenpiezal] 2000mm/sec FINE Offset,PR[3:z_offset]
9: J PR[2:pos_soldar] 100% FINE
```



ROBOT2.- SOLDARARRIBA

Subrutina de soldar las piezas de tipo 2 que van sobre la parte superior de la placa. En función de las piezas de tipo 2 ya colocadas, el 'user frame' es uno u otro.

```
1:  UFRAME_NUM=R[2:uframe]
2:  J PR[2:pos_soldar] 100% FINE
3:  L PR[12:origenpieza2] 2000mm/sec FINE
4:  L PR[12:origenpieza2] 2000mm/sec FINE Offset,PR[4:x_offset2]
5:  L PR[12:origenpieza2] 2000mm/sec FINE Offset,PR[7:y_offneg]
6:  L PR[12:origenpieza2] 2000mm/sec FINE Offset,PR[5:y_offset2]
7:  L PR[12:origenpieza2] 2000mm/sec FINE
8:  L PR[12:origenpieza2] 2000mm/sec FINE Offset,PR[3:z_offset]
9:  J PR[2:pos_soldar] 100% FINE
```

ROBOT3.- MAIN

Programa que ejecuta el Robot3 en bucle infinito. Según el valor de la DI16, entrará en modo manual o automático. Además incluye información para la HMI.

```
1:  LBL[1]
2:  IF (UI[6:Start]=ON),UO[4:Prg paused]=(OFF)
3:  IF DI[6:MODO IND*/AUT]=OFF,CALL INDIVIDUAL
4:  IF DI[6:MODO IND*/AUT]=ON,CALL RUTINAAUT
5:  JMP LBL[1]
```

ROBOT3.- INDIVIDUAL

Si DI6 = 0, entrará en modo manual. El robot queda a la espera de ejecutar el único movimiento disponible – espera la activación de la DI5.

```
1:  J PR[1:HOME] 100% CNT25
2:
3:  IF (DI[5:VA & COLOCAR IND]) THEN
4:  IF R[21:Piezas1Colocadas]>0 OR R[22:Piezas2Colocadas]>0 OR
   R[5:ID]<>0,CALL BORRAR_REGISTROS
5:
6:  CALL VISIONPIEZAS
7:
8:  J PR[1:HOME] 100% CNT25
9:  ENDIF
10: IF (R[20:PiezasQuedan]=0) THEN
11: JMP LBL[400]
12: LBL[400]
13: ENDIF
```



ROBOT3.- RUTINA AUT

Si DI6 = 1, entra en modo automático.

```
1:  LBL[400]
2:  J PR[1:HOME] 100% CNT25
3:
4:  DO[1]=OFF
5:  DO[2]=OFF
6:  WAIT DI[4:VA & COLOCA AUT]=ON
7:  IF R[21:Piezas1Colocadas]>0 OR R[22:Piezas2Colocadas]>0 OR
   R[5:ID]<>0,CALL BORRAR_REGISTROS
8:
9:  CALL VISIONPIEZAS
10:
11: J PR[1:HOME] 100% CNT25
12: DO[1]=OFF
13: IF (R[20:PiezasQuedan]=0) THEN
14: JMP LBL[400]
15: ENDIF
```

ROBOT3.- VISIONPIEZAS

Subrutina de visión de piezas ya explicada en el capítulo 5.1.6.6.

```
1:  UTOOL_NUM=2
2:  UFRAME_NUM=1
3:  OVERRIDE=100%
4:
5:  VISION RUN_FIND 'RECONOCER_PIEZAS'
6:  VISION GET_NFOUND 'RECONOCER_PIEZAS' R[10]
7:
8:
9:  R[20:PiezasQuedan]=R[10]
10: IF R[10]=0,JMP LBL[100]
11: LBL[10]
12:
13:
14: VISION GET_OFFSET 'RECONOCER_PIEZAS' VR[R[20]] JMP LBL[100]
15: R[10]=R[10]-1
16: IF R[10]<>0,JMP LBL[10]
17:
18: LBL[50]
19: DO[1]=OFF
20: R[5]=VR[R[20]].MODELID
21: IF R[5:ID]=2,JMP LBL[20]
22:
23: CALL PICKUP1
24: CALL DROPOFF1
25: IF R[20:PiezasQuedan]<>0,JMP LBL[50]
26: JMP LBL[30]
27:
```



```
28: LBL[20]
29: CALL PICKUP2
30: CALL DROPOFF2
31: IF R[20:PiezasQuedan]<>0,JMP LBL[50]
32: LBL[30]
33: DO[1]=ON
34: LBL[100]
35:
36: !Error num piezas < 4
37: LBL[700]
38: DO[2]=ON
```

ROBOT3.- PICKUP1

Subrutina de tomar piezas del tipo 1 empleando visión artificial.

```
1: UFRAME_NUM=1
2: L PR[3:REF_PIEZA1] 2000mm/sec CNT25
   Tool_Offset,PR[2:offsetZ] VOFFSET,VR[R[20]]
3: L PR[3:REF_PIEZA1] 1000mm/sec FINE VOFFSET,VR[R[20]]
4: CALL PICKUP_SIM1
5: L PR[3:REF_PIEZA1] 1000mm/sec CNT25
   Tool_Offset,PR[2:offsetZ] VOFFSET,VR[R[20]]
6: J PR[1:HOME] 100% FINE
7: R[20:PiezasQuedan]=R[20:PiezasQuedan]-1
8: R[21:Piezas1Colocadas]=R[21:Piezas1Colocadas]+1
```

ROBOT3.- PICKUP2

Subrutina de tomar piezas del tipo 2 empleando visión artificial.

```
1: UFRAME_NUM=1
2: L PR[4:REF_PIEZA2] 2000mm/sec FINE
   Tool_Offset,PR[2:offsetZ] VOFFSET,VR[R[20]]
3: L PR[4:REF_PIEZA2] 1000mm/sec FINE VOFFSET,VR[R[20]]
4: CALL PICKUP_SIM2
5: L PR[4:REF_PIEZA2] 1000mm/sec FINE
   Tool_Offset,PR[2:offsetZ] VOFFSET,VR[R[20]]
6: J PR[1:HOME] 100% FINE
7: R[20:PiezasQuedan]=R[20:PiezasQuedan]-1
8: R[22:Piezas2Colocadas]=R[22:Piezas2Colocadas]+1
```

ROBOT3.- DROPOFF1

Subrutina de dejar piezas del tipo 1 que van sobre la parte inferior de la placa. En función de las piezas de tipo 1 ya colocadas, se activa la D04 o la D05 que comunican al Robot2 el tipo de pieza y donde va colocada en la placa.

```
1: IF R[21:Piezas1Colocadas]=2,JMP LBL[80]
2: DO[4]=ON
3: J P[1] 100% FINE
4: J PR[5:PIEZA1.1_SOP] 100% FINE Tool_Offset,PR[2:offsetZ]
```



```
5: L PR[5:PIEZA1.1_SOP] 2000mm/sec FINE
6: CALL DROPOFF_SIM1
7: DO[1]=ON
8: L PR[5:PIEZA1.1_SOP] 2000mm/sec FINE Tool_Offset,PR[2:offsetZ]
9: J P[2] 100% CNT25
10: J PR[1:HOME] 100% FINE
11: WAIT 1.50(sec)
12: DO[4]=OFF
13: JMP LBL[90]
14:
15: LBL[80]
16: DO[5]=ON
17: J P[1] 100% CNT25
18: J PR[6:PIEZA1.2_SOP] 100% FINE Tool_Offset,PR[2:offsetZ]
19: L PR[6:PIEZA1.2_SOP] 2000mm/sec FINE
20: CALL DROPOFF_SIM1
21: DO[1]=ON
22: L PR[6:PIEZA1.2_SOP] 2000mm/sec FINE Tool_Offset,PR[2:offsetZ]
23: J P[2] 100% CNT25
24: J PR[1:HOME] 100% FINE
25: WAIT 1.50(sec)
26: DO[5]=OFF
27:
28: LBL[90]
29: J PR[1:HOME] 100% FINE
```

ROBOT3.- DROPOFF2

Subrutina de dejar piezas del tipo 1 que van sobre la parte inferior de la placa. En función de las piezas de tipo 1 ya colocadas, se activa la D02 o la D03 que comunican al Robot2 el tipo de pieza y donde va colocada en la placa.

```
1: IF R[22:Piezas2Colocadas]=2,JMP LBL[80]
2: DO[2]=ON
3: J P[1] 100% CNT25
4: J PR[7:PIEZA2.1_SOP] 100% FINE Tool_Offset,PR[2:offsetZ]
5: L PR[7:PIEZA2.1_SOP] 2000mm/sec FINE
6: CALL DROPOFF_SIM2
7: DO[1]=ON
8: L PR[7:PIEZA2.1_SOP] 2000mm/sec FINE Tool_Offset,PR[2:offsetZ]
9: J P[2] 100% CNT25
10: J PR[1:HOME] 100% FINE
11: WAIT 1.50(sec)
12: DO[2]=OFF
13: JMP LBL[90]
14:
15: LBL[80]
16: DO[3]=ON
17: J P[1] 100% CNT25
18: J PR[8:PIEZA2.2_SOP] 100% FINE Tool_Offset,PR[9:offsetpiezas2]
19: L PR[8:PIEZA2.2_SOP] 2000mm/sec FINE
```



```
20: CALL DROPOFF_SIM2
21: DO[1]=ON
22: L PR[8:PIEZA2.2 _SOP] 2000mm/sec FINE
    Tool_Offset,PR[9:offsetpiezas2]
23: J P[2] 100% CNT25
24: J PR[1:HOME] 100% FINE
25: WAIT 1.50(sec)
26: DO[3]=OFF
27:
28: LBL[90]
29: J PR[1:HOME] 100% FINE
```

ROBOT3.- BORRAR_REGISTROS

Borrar registros para que no haya errores en la transición manual-automático.

```
1: R[21:Piezas1Colocadas]=0
2: R[22:Piezas2Colocadas]=0
3: R[5:ID]=0
```



ANEXO C. DIAGRAMAS.

Sinóptico de la maniobra automática de la estación.

https://alumnosuvaes-my.sharepoint.com/personal/jorge_villa_gonzalez_alumnos_uva_es/_layouts/15/Doc.aspx?sourcedoc={cd6d66f3-d5ac-45fb-8cff-fe91ccc7e80f}&action=embedview

Diagramas de flujo del Robot1.

https://alumnosuvaes-my.sharepoint.com/personal/jorge_villa_gonzalez_alumnos_uva_es/_layouts/15/Doc.aspx?sourcedoc={bba21f47-2afc-4c50-b4f8-a930c36f0d93}&action=embedview

Diagramas de flujo del Robot2.

https://alumnosuvaes-my.sharepoint.com/personal/jorge_villa_gonzalez_alumnos_uva_es/_layouts/15/Doc.aspx?sourcedoc={5afa8c2a-b27e-40b9-81a3-49b9cd7bc235}&action=embedview

Diagramas de flujo del Robot3.

https://alumnosuvaes-my.sharepoint.com/personal/jorge_villa_gonzalez_alumnos_uva_es/_layouts/15/Doc.aspx?sourcedoc={4ffa4fd3-3838-4394-801d-5d85f432fa25}&action=embedview