



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

GRADO EN INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

**Diseño y Programación de Varias Estaciones de
Trabajo Mediante el Uso de PLC y Softwares
Específicos**

Autor:

del Barrio Lagándara, Víctor

Tutor:

**Moya de la Torre, Eduardo Julio
Ingeniería de Sistemas y
Automática**

Valladolid, Marzo 2017.

RESUMEN Y PALABRAS CLAVE

El trabajo consiste en simular de forma virtual mediante el uso de software específico, el comportamiento de varias estaciones de trabajo automatizadas, cuyo control es llevado a cabo mediante la programación de un PLC.

La aparición en el mercado en estos últimos años de nuevos software de simulación y la mejora de los existentes hasta el momento, nos permiten hoy en día estudiar el comportamiento de ciertos sistemas sin la necesidad de disponer de ninguno de los elementos físicos que los componen.

Dentro de estos programas de simulación se encuentra FACTORY I/O enfocado especialmente al ámbito educativo. Este software nos permite disponer de distintas células de fabricación en aulas y centros, para el aprendizaje y formación de los alumnos que de otro modo serían prácticamente imposibles de adquirir debido a su elevado coste económico.

En primer lugar se ha realizado una presentación de los softwares utilizados tanto para la programación del PLC (TIA PORTAL) como para la simulación de las estaciones (FACTORY I/O), para conocer sus características principales y las ventajas que nos aportan.

Posteriormente se exponen distintos procedimientos de conexión y comunicación entre estos dos tipos de softwares que nos garantice la correcta simulación de las estaciones.

En último lugar se van explicando una a una las diferentes estaciones que se han diseñado, en las que al inicio, se presenta una descripción de los objetivos de funcionamiento de cada estación, con los elementos de los cuales están formadas y su configuración. También se desarrolla el comportamiento secuencial de las mismas mediante el uso de los diferentes graficets de funcionamiento y por último se incluye cómo se ha realizado la programación del autómata utilizando las diferentes direcciones de entrada, salida y marcas internas para que el comportamiento sea acorde a los objetivos de comportamiento de cada estación.

Palabras clave: PLC, simulación, TIA PORTAL, FACTORY I/O, programación.

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN Y OBJETIVOS	11
1.1. JUSTIFICACIÓN TFG	11
1.2. OBJETIVOS	12
1.3. ESTRUCTURA DE LA MEMORIA.....	12
2. PRINCIPIOS BÁSICOS DE LOS SOFTWARES DE PROGRAMACIÓN Y SIMULACIÓN	15
2.1. TIA PORTAL V13.....	15
2.1.1. CARACTERÍSTICAS PRINCIPALES	15
2.1.2. CONFIGURACIÓN DE REDES Y DISPOSITIVOS	15
2.1.3. LENGUAJES DE PROGRAMACIÓN IEC	16
2.1.4. SÍMBOLOS.....	17
2.1.5. FUNCIONES ONLINE	17
2.1.6. SEGURIDAD INTEGRADA.....	18
2.2. FACTORY I/O	19
2.2.1. CARATERÍSTICAS PRINCIPALES.....	19
2.2.2. NAVEGACIÓN	19
2.2.3. CONTROL MANUAL	19
2.2.4. CONTROL CON PLC	20
2.2.5. PARTES.....	21
2.2.6. ESCENAS.....	21
3. PROCESO DE INSTALACIÓN DE FACTORY I/O	23
4. CONFIGURACIÓN DE LA COMUNICACIÓN ENTRE FACTORY I/O Y TIA PORTAL CON PLC FÍSICO	27
4.1. CONFIGURACIÓN DE LAS CARACTERÍSTICAS DEL PLC.....	27
4.2. CONFIGURACIÓN DE LOS DRIVERS DE FACTORY I/O	28
4.3. CONFIGURACIÓN DE LA RED LOCAL ENTRE PC Y AUTOMATA 1200	31
5. CONFIGURACIÓN DE LA COMUNICACIÓN ENTRE FACTORY I/O Y TIA PORTAL CON PLCSIM V13/14.....	37
5.1. CONFIGURACION DE LAS CARACTERÍSTICAS DEL PLC.....	37
5.2. CONFIGURACION DE LOS DRIVERS DE FACTORY I/O	39
6. GRAFCET	41
6.1. PRINCIPIOS GENERALES DEL GRAFCET	41
6.2. REGLAS DE EVOLUCIÓN DEL GRAFCET	42

6.3. ELEMENTOS DEL GRAFCET	42
6.4. ESTRUCTURAS BÁSICAS DE PROGRAMACIÓN EN GRAFCET	43
7. ESTACIONES DE SIMULACIÓN	47
7.1. ESTACIÓN PALETIZADORA	47
7.1.1. OBJETIVO DE LA ESTACIÓN.....	47
7.1.2. DISEÑO DE LA ESTACIÓN EN FACTORY I/O	49
7.1.3. PROGRAMACIÓN SECUENCIAL GRAFCET	53
7.1.4. PROGRAMACIÓN DEL PLC	55
7.2. ESTACIÓN DISTRIBUIDORA.....	60
7.2.1. OBJETIVO DE LA ESTACIÓN.....	60
7.2.2. DISEÑO DE LA ESTACIÓN EN FACTORY I/O	61
7.2.3. PROGRAMACIÓN SECUENCIAL GRAFCET	66
7.2.4. PROGRAMACIÓN DEL PLC	68
7.3. DEPÓSITO DE LÍQUIDO.....	76
7.3.1. OBJETIVO DE LA ESTACIÓN.....	77
7.3.2. DISEÑO EN FACTORY I/O	78
7.3.3. CONFIGURACIÓN DE CONTROLADORES PID	82
7.3.4. PROGRAMACIÓN DEL PLC	86
7.4. GESTIÓN DE STOCK DE UN ALMACÉN	89
7.4.1. Objetivo de la estación.....	89
7.4.2. DISEÑO EN FACTORY I/O	90
7.4.3. PROGRAMACIÓN SECUENCIAL GRAFCET	93
7.4.4. PROGRAMACIÓN DEL PLC	95
7.4.5. CONFIGURACIÓN DEL HMI.....	100
7.4.6. PROGRAMACIÓN DEL HMI	103
7.4.7. SIMULACION MEDIANTE PLCSIM	107
8. ESTUDIO ECONOMICO.....	111
9. CONCLUSIONES.....	113
10. BIBLIOGRAFÍA.....	115

ÍNDICE DE FIGURAS

Figura 2.1. Estructura TIA PORTAL	15
Figura 2.2. Vista topológica	16
Figura 2.3. Editor KOP	17
Figura 2.4. Conjunto de características que ofrece TIA PORTAL.....	18
Figura 2.5. Forzado de etiquetas	20
Figura 2.6. Esquema de comunicación entre FACTORY I/O y TIA PORTAL.....	21
Figura 2.7. Elementos para la construcción de estaciones	21
Figura 3.1. Enlace de descarga proporcionado por RealGames en la web	23
Figura 3.2. Pantalla de aviso del instalador	23
Figura 3.3. Pantalla de diferentes complementos a instalar	24
Figura 3.4. Menú FACTORY I/O.....	25
Figura 3.5. Pantalla de activación de licencia.....	25
Figura 4.1. Configuración de las entradas del PLC.....	27
Figura 4.2. Configuración de acceso a la comunicación PUT/GET.....	28
Figura 4.3. Menú de FACTORY I/O	28
Figura 4.4. Selección de driver	29
Figura 4.5. Configuración de dirección y adaptador de red	29
Figura 4.6. Configuración de las direcciones de entradas y salidas	30
Figura 4.7. Distribución final de entradas y salidas.....	30
Figura 4.8. Esquema de dirección de red.....	31
Figura 4.9. Esquema dirección de subred.....	32
Figura 4.10. Ventana de configuración de propiedades de conexión	33
Figura 4.11. Ventana de configuración de dirección IP.....	33
Figura 4.12. Pantalla de creación de nuevo proyecto	34
Figura 4.13. Selección de modelo de CPU	34
Figura 4.14. Visualización de la CPU en TIA PORTAL.....	35
Figura 4.15. Configuración de la dirección de red de la CPU del autómatas	35
Figura 5.1. Plantilla de programación.....	37
Figura 5.2. Inicio de simulación.....	38
Figura 5.3. Configuración opciones de comunicación.....	38
Figura 5.4. Pantalla de ejecución de PLCSIM	39
Figura 5.5. Selección de driver PLCSIM	39
Figura 5.6. Selección de modelo de CPU.....	40
Figura 6.1. Esquema grafcet etapa/acción asociada.....	41
Figura 6.2. Esquema grafcet transición.....	41
Figura 6.3. Etapa grafcet	42
Figura 6.4. Etapa inicial grafcet.....	43
Figura 6.5. Transición grafcet.....	43
Figura 6.6. Inicio secuencia condicionada	43
Figura 6.7. Final secuencia condicionada	44
Figura 6.8. Secuencia simultánea.....	45
Figura 6.9. Salto condicional	45
Figura 7.1. Elemento generador de piezas.....	51
Figura 7.2. Elemento destructor de piezas.....	51
Figura 7.3. Vista lateral estación paletizadora.....	51

Figura 7.4. Vista frontal estación paletizadora	52
Figura 7.5. Cuadro de mando estación paletizadora.....	52
Figura 7.6. Distribución entradas/salidas estación paletizadora.....	53
Figura 7.7. Grafcet control cinta transportadora	53
Figura 7.8. Grafcet control transportador rodillos.....	54
Figura 7.9. Grafcet control cartesiano	54
Figura 7.10. Distribución general de bloques. Estación paletizadora	56
Figura 7.11. Bloque FB130 cartesiano.....	57
Figura 7.12. Bloque FC10 gestión marcha.....	58
Figura 7.13. Bloque FC130 posición cartesiano.....	58
Figura 7.14. Bloque FC131 posición pieza	58
Figura 7.15. Bloque FC110 transportador cinta	59
Figura 7.16. Bloque FC120 transportador rodillos	59
Figura 7.17. Vista frontal estación distribuidora.....	64
Figura 7.18. Vista lateral derecha estación distribuidora	64
Figura 7.19. Vista lateral izquierda estación distribuidora.....	65
Figura 7.20. Cuadro de mando estación distribuidora	65
Figura 7.21. Distribución entradas/salidas estación distribuidora	66
Figura 7.22. Grafcet distribuidor piezas	66
Figura 7.23. Grafcet cinta recoge piezas.....	67
Figura 7.24. Grafcet cinta cajas	67
Figura 7.25. Grafcet robot automático	68
Figura 7.26. Distribución general de bloques. Estación distribuidora	70
Figura 7.27. Bloque FB140. Robot automático	71
Figura 7.28. Bloque FB130. Cinta recoge piezas	72
Figura 7.29. Bloque FB150. Cinta cajas.....	72
Figura 7.30. Bloque FB120. Distribuidor pieza.....	73
Figura 7.31. Bloque FC10. Gestión marcha	74
Figura 7.32. Bloque FC110. Cinta generadora de piezas	74
Figura 7.33. Bloque FC160. Contador de piezas	74
Figura 7.34. Bloque FC141. Piezas robot.....	75
Figura 7.35. Bloque FC142. Posición robot	75
Figura 7.36. Bloque FC130. Gestión cinta piezas	75
Figura 7.38. Bloque FC140. Gestión robot.....	75
Figura 7.39. Bloque FC150. Gestión cinta cajas	75
Figura 7.41. Efecto sobreoscilación de señal	77
Figura 7.42. Esquema control PID válvula llenado depósito.....	78
Figura 7.43. Esquema control PID válvula vaciado depósito	78
Figura 7.44. Vista frontal del depósito.....	80
Figura 7.45. Vista lateral depósito	80
Figura 7.46. Cuadro de mando depósito	81
Figura 7.47. Distribución entradas/salidas. Estación depósito.....	81
Figura 7.48. Selección de bloque interrupción cíclica en TIA PORTAL	82
Figura 7.49. Selección bloque PID_COMPACT	83
Figura 7.50. Asignación de variables en PID_COMPACT	83
Figura 7.51. Configuración PID_COMPACT.....	84
Figura 7.52. Puesta en servicio PID_COMPACT	85
Figura 7.53. Configuración de los parámetros del PID_COMPACT	85
Figura 7.54. Visualización comportamiento PID_COMPACT	86
Figura 7.55. Distribución general de bloques. Depósito líquido.....	87

Figura 7.56. Ciclo de ejecución OB30.....	87
Figura 7.57. BloqueFB1. Depósito	88
Figura 7.58. Vista frontal estación de gestión de stock	92
Figura 7.59. Vista detallada estación de gestión de stock	92
Figura 7.60. Distribución entradas/salidas. Estación gestión de stock.....	93
Figura 7.61. Grafcet cinta de carga	93
Figura 7.62. Grafcet cinta de descarga	94
Figura 7.63. Grafcet transportador	94
Figura 7.64. Distribución general de bloques. Estación de gestión de stock.....	96
Figura 7.65. Bloque FB140. Posiciones	97
Figura 7.66. Bloque FB130. Transportador	98
Figura 7.67. Bloque FB150. Contador piezas	99
Figura 7.68. Bloque FC110. Cinta de carga	99
Figura 7.69. Bloque FC120. Cinta de descarga	100
Figura 7.70. Bloque FC9000	100
Figura 7.71. Estructura en árbol de proyecto.....	101
Figura 7.72. Selección de tipo de pantalla HMI	101
Figura 7.73. Menú de configuración de dispositivo.....	102
Figura 7.74. Menú de asignación de dirección de red a la pantalla HMI	102
Figura 7.75. Menú de acceso a vista de redes	102
Figura 7.76. Vista de dispositivos y redes	103
Figura 7.77. Menú de variables HMI.....	103
Figura 7.78. Lista de variable de HMI.....	104
Figura 7.79. Estructura de imágenes que componen el HMI	104
Figura 7.80. Imagen raíz HMI	105
Figura 7.81. Imagen productos HMI	106
Figura 7.82. Imagen pieza prima azul del HMI	107
Figura 7.83. Selección del PLC del proyecto	108
Figura 7.84. Menú de inicio de simulación	108
Figura 7.85. Ejecución del simulador	109
Figura 7.86. Selección de la pantalla HMI del proyecto.....	109
Figura 7.87. Botón de la barra de herramientas de inicio de RUNTIME	109

1. INTRODUCCIÓN Y OBJETIVOS

1.1. JUSTIFICACIÓN TFG

En este trabajo se pretende programar y simular el comportamiento de varias células de fabricación que sirvan de base para la enseñanza de asignaturas del grado en electrónica y automática industrial, del grado en organización e incluso de asignaturas propias del master en informática industrial o del master en electrónica y automática impartidos por la escuela de ingenieros industriales de Valladolid, permitiendo a los alumnos adquirir y asimilar ciertos conceptos de programación de PLCs que de otra manera pueden resultar un tanto complejos.

Hoy en día existen en el mercado una gran variedad de fabricantes destinados al mundo de la automatización industrial que ofertan entre sus productos diferentes tipos de autómatas programables. A nivel internacional destacan marcas como Schneider Electric, Rockwell Automation o Siemens entre muchas otras. Cada una de estas marcas a su vez fabrica diferentes tipos de PLC de mayor o menor capacidad dependiendo de la aplicación a la que sean destinados.

Para la realización de la automatización de las estaciones del presente trabajo se han utilizado autómatas de la marca Siemens de modelo 1200, ya que son los modelos de PLC de los que dispone la escuela. Además este tipo de autómatas aunque no sean de los modelos ofertados por el fabricante que más potencial tienen, nos aportan una serie de recursos que a nivel educativo son más que suficientes.

La programación de estos autómatas se realiza mediante el uso de un software específico del fabricante denominado TIA PORTAL. Este software es de gran tamaño y no solo nos permite programar autómatas de la serie 1200 sino que nos permite programar la mayor parte de los productos ofertados por el fabricante en el mercado como HMI o variadores de velocidad entre otros muchos, además también nos permite configurar la comunicación entre los mismos aportándonos una gran comodidad a la hora de automatizar una estación de trabajo compuesta por diferentes productos del mismo fabricante.

La mayor parte de los fabricantes mencionados anteriormente proporcionan junto al software de programación del autómata otro software que permite al programador visualizar la simulación del programa. Para el caso de Siemens por ejemplo, existe el software PLCSIM en el que se puede ir visualizando cómo evolucionan el estado de las entradas, salidas o marcas del autómata a medida que se van forzando unas u otras. Sin embargo esta forma de visualización es un poco abstracta cuando se está empezando a adquirir conocimientos en este ámbito o cuando no se dispone del modelo físico para probarlo.

Por esa razón ciertos fabricantes están empezando a desarrollar softwares de simulación que nos permitan construir nuestra propia célula de trabajo a nivel virtual y que esta se comporte acorde al programa diseñado en los softwares de los fabricantes mencionados anteriormente.

Entre los fabricantes de este tipo de softwares de simulación se encuentra RealGames que se centra principalmente en el ámbito educativo mediante el desarrollo del software FACTORY I/O utilizado en las estaciones del presente trabajo.

FACTORY I/O nos proporciona una librería con diferentes tipos de objetos y elementos presentes actualmente en las fábricas a nivel industrial, de tal forma que podemos construir nuestra propia estación de trabajo configurándolos y conectándolos de la manera deseada.

1.2. OBJETIVOS

En este trabajo se presenta una nueva forma de aplicar los conocimientos de programación de autómatas, simulando el comportamiento de dichos programas mediante el funcionamiento de células de trabajo virtuales cuyo modelo y comportamiento es muy similar al que presentan la mayoría de las industrias.

Teniendo en cuenta esto, los objetivos de este trabajo son:

- Afianzar gran parte de los conocimientos adquiridos en cuanto a la programación de autómatas se refiere.
- Adquirir conocimientos nuevos en el ámbito de la programación de HMI.
- Proporcionar unas prácticas para los estudiantes de asignaturas impartidas por el departamento de ingeniería de sistemas y automática de la UVA.
- Realizar la documentación necesaria para su manejo.
- Tener un conocimiento práctico en la conexión y en el manejo del PLC y las pantallas HMI con el exterior.
- Aumentar las habilidades en el manejo de TIA PORTAL y adquirir otras nuevas mediante el uso de FACTORY I/O.

1.3. ESTRUCTURA DE LA MEMORIA

La estructura de la memoria es la siguiente:

- **Introducción y objetivos:** se realiza una introducción al contexto del trabajo, justificando el mismo y los objetivos.

- **Principios básicos de los softwares de programación y simulación:** se presentan los softwares que se van a utilizar para realizar las diferentes estaciones del presente trabajo.
- **Proceso de instalación de FACTORY I/O:** se describe de forma detallada los pasos a seguir para la instalación de este software en el ordenador.
- **Configuración de la comunicación entre FACTORY I/O y TIA PORTAL con PLC físico:** se explica cómo se realiza la configuración de la comunicación entre dichos softwares y el autómata.
- **Configuración de la comunicación entre FACTORY I/O y TIA PORTAL con PLCSIM V13/14:** se explica cómo se realiza la configuración de la comunicación entre dichos elementos.
- **Grafcet:** se detallan los principios teóricos básicos para el uso de dicho lenguaje gráfico de programación de procesos secuenciales.
- **Estaciones de simulación:** se van detallando una a una las diferentes estaciones planteadas, describiendo detalladamente todas las etapas de las que consta su proceso de programación.
- **Estudio económico:** se estudian los costes implicados en la elaboración del trabajo.
- **Conclusiones:** se exponen las conclusiones del trabajo, mostrando los resultados obtenidos.
- **Bibliografía:** se muestra la bibliografía que se ha utilizado para la elaboración del trabajo

2. PRINCIPIOS BÁSICOS DE LOS SOFTWARES DE PROGRAMACIÓN Y SIMULACIÓN

2.1. TIA PORTAL V13

Con Totally Integrated Automation Portal (TIA Portal) Siemens integra diferentes productos SIMATIC en un aplicación de software que permite aumentar la productividad y eficiencia de la automatización en todos los sectores, cubriendo gran parte de las etapas de las que consta el proceso de automatización como la etapa de diseño, puesta en marcha y mantenimiento.

2.1.1. CARACTERÍSTICAS PRINCIPALES

Con TIA Portal se configura tanto el control como la visualización en un sistema de ingeniería unitario. Todos los datos se guardan en un proyecto común. Los componentes de programación (STEP 7) y visualización (WinCC) no son programas independientes, sino editores de un sistema que accede a una base de datos común.

Para todas las tareas se utiliza una interfaz de usuario común desde la que se accede en todo momento a todas las funciones de programación y visualización. Esto hace a Siemens ser uno de los líderes en el sector de la automatización industrial.

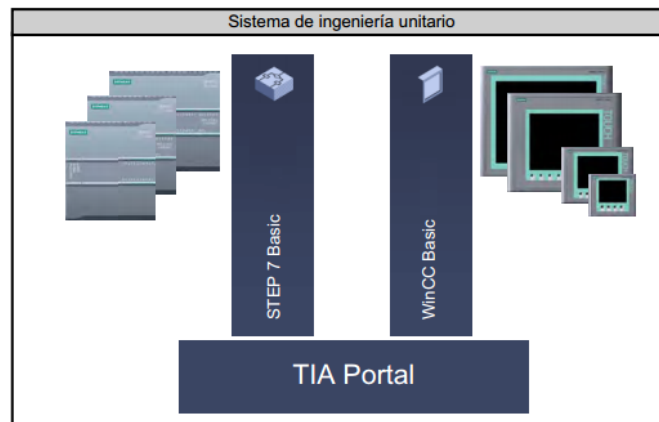


Figura 2.1. Estructura TIA PORTAL

2.1.2. CONFIGURACIÓN DE REDES Y DISPOSITIVOS

Con un solo editor gráfico se puede configurar fácilmente una instalación completa. Para ello el editor ofrece tres vistas:

- **Vista de redes:** la vista de redes permite configurar la comunicación dentro de la planta, se configuran de una forma gráfica y muy clara los enlaces de comunicación entre las diferentes estaciones.

- **Vista de dispositivos:** esta vista nos permite realizar una lectura autónoma del hardware existente y de visualizar de forma jerárquica todos los parámetros y datos de configuración sensibles al contexto.
- **Vista topológica:** en este caso se visualizan de modo gráfico los controladores y la periferia descentralizada asignada a ellos.

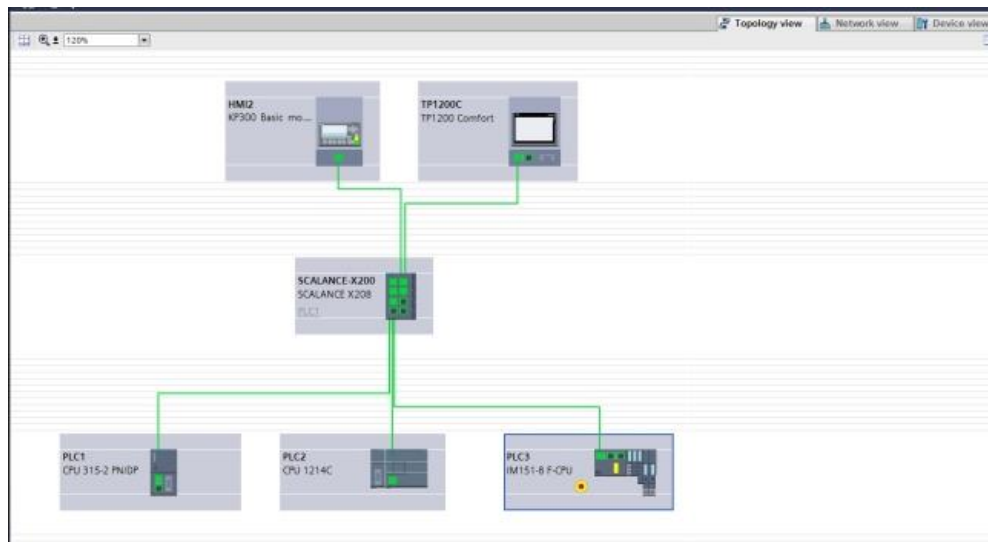


Figura 2.2. Vista topológica

2.1.3. LENGUAJES DE PROGRAMACIÓN IEC

TIA PORTAL nos ofrece de forma intuitiva herramientas para realizar las tareas como arrastras y soltar, lista de referencias cruzadas a nivel de todo el proyecto etc. Además la integración de los distintos editores en un único entorno de trabajo nos permite disponer de todos los datos actualizados de forma coherente.

Por otro lado gracias a la programación simbólica homogénea se reduce la probabilidad de cometer errores.

Entre los lenguajes de programación proporcionados se encuentran:

- **KOP y FUP** (Lenguajes gráficos de programación): debido a las potentes herramientas y a las funcionalidades integradas, la velocidad en la creación de programas mediante el uso de estos lenguajes se ha incrementado considerablemente llegando a estar a la altura de los lenguajes textuales.
- **AWL** (Lista de instrucciones): este tipo de programación en texto permite crear programas de usuario propios de hardware con optimización de tiempo de ejecución y espacio de almacenamiento.
- **GRAPH** (Programa de cadenas secuenciales): se utiliza para describir procesos secuenciales con sucesiones de etapas alternativas o paralelas. Los procesos se

configuran de una forma clara y rápida en un tipo de representación estandarizado.

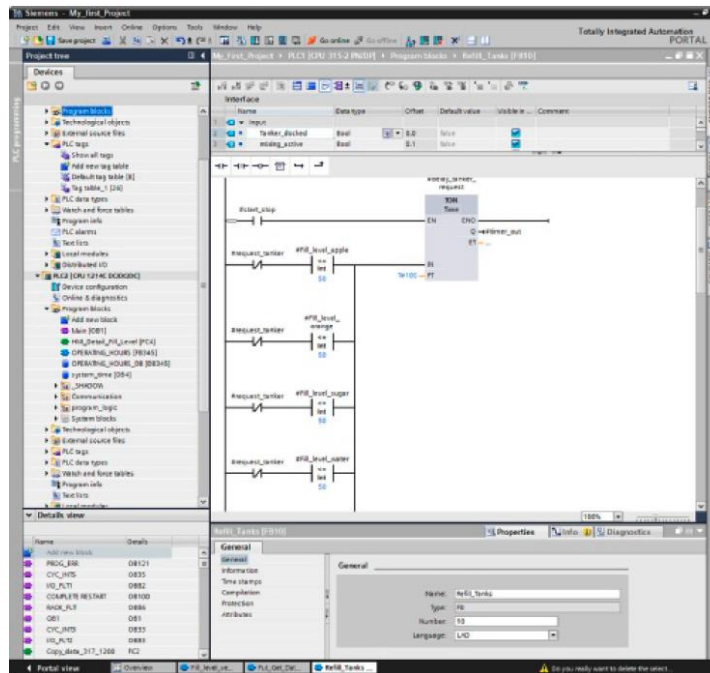


Figura 2.3. Editor KOP

2.1.4. SÍMBOLOS

Como editor común para variables y constantes globales se utiliza la tabla de variables, esto permite generar grandes cantidades de variables utilizando los mismos mecanismos de Microsoft Excel.

Se puede realizar la asignación de hardware a símbolos mediante el sencillo método de arrastrar y colocar entre el programa de usuario y la vista de dispositivo. La coherencia automática de datos garantiza que todas las ubicaciones de uso de un símbolo se actualicen inmediatamente.

2.1.5. FUNCIONES ONLINE

Los datos online/offline de un proyecto se comparan automáticamente, y las diferencias se visualizan con toda claridad. La comparación online/offline a nivel de bloque se realiza directamente en el árbol del proyecto. Tras la conexión al sistema se ofrece en el árbol de proyecto la visualización del estado operativo y una comparación online/offline a nivel de bloque.

2.1.6. SEGURIDAD INTEGRADA

Se ofrece un método de protección de know-how que impide la lectura, copia y modificación inadvertida de los contenidos. Una configuración de copia configurable impide la reproducción no autorizada de bloques de programa.

Por lo tanto cabe destacar que trabajar con TIA PORTAL significa trabajar con un software que integra una gran cantidad de características y capacidades acordes a la actualidad.

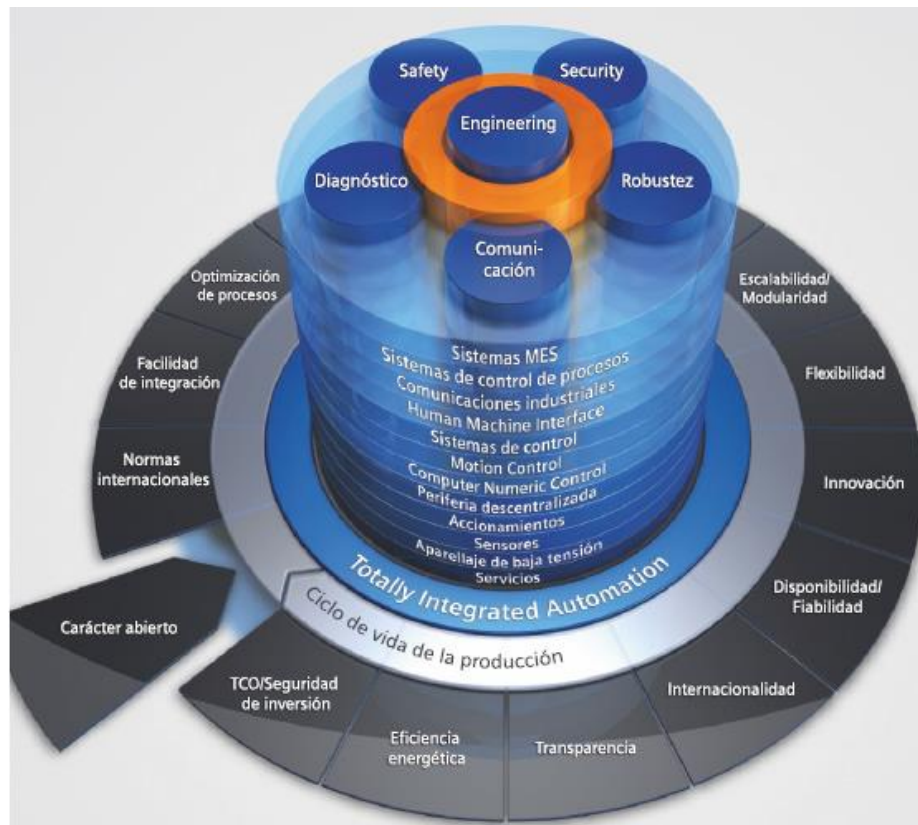


Figura 2.4. Conjunto de características que ofrece TIA PORTAL

2.2. FACTORY I/O

2.2.1. CARACTERÍSTICAS PRINCIPALES

El uso de FACTORY I/O nos permite realizar de forma sencilla simulaciones de fábricas en 3D para el aprendizaje de tecnologías de automatización. Permite construir rápidamente una fábrica virtual utilizando una selección de piezas industriales comunes.

Por otro lado, incluye muchas escenas inspiradas en aplicaciones industriales típicas de diferente nivel de dificultad.

El escenario más común es usar el software como una plataforma de entrenamiento de PLC ya que son los controladores más comunes que se encuentran en aplicaciones industriales. Sin embargo, este programa también ofrece la posibilidad de utilizar microcontroladores, SoftPLC, Modbus, entre otras muchas tecnologías.

2.2.2. NAVEGACIÓN

Se dispone de varias cámaras que nos permiten navegar en el espacio 3D y son clave para interactuar con las diferentes partes. Las cámaras son de tres tipos diferentes y cada una fue diseñada para un propósito funcional específico. Estos tres tipos de cámaras son:

- **Cámara de órbita:** es la cámara por defecto y debe utilizarse al construir una escena. Esta es la única cámara que le permite moverse a través de partes sin colisionar con ellos.
- **Cámara Fly:** se utiliza para moverse libremente en el espacio 3D. Esta cámara choca con partes de la escena pero no es detectada por los sensores.
- **Cámara de primera persona:** esta cámara representa a una persona de 1,8m de altura. Se debe utilizar cuando se simula a una persona en una fábrica virtual. Colisiona con partes de la escena.

2.2.3. CONTROL MANUAL

Para verificar el correcto diseño de la escena construida, FACTORY I/O dispone de una serie de etiquetas que nos permiten probar manualmente el funcionamiento de cada uno de los elementos que constituyen nuestra estación.

Las etiquetas están hechas de un nombre y un valor y pueden ser de dos tipos:

- Etiquetas de sensores
- Etiquetas de actuadores

Por otro lado las etiquetas pueden contener tres tipos de datos diferentes:

- Boolean: para valores on/off
- Float: para valores analógicos
- Integer: para datos específicos (enteros)

Los valores de etiqueta pueden ser forzados en cualquier momento, lo que permite desempeñar el papel de controlador.



Figura 2.5. Forzado de etiquetas

2.2.4. CONTROL CON PLC

El control de nuestra escena mediante el uso de un PLC se realiza a través de los controladores de Entrada/Salida que incorpora FACTORY I/O.

El controlador de E/S es el responsable de “hablar” con un controlador externo. FACTORY I/O incluye muchos controladores de E/S, cada uno para una tecnología específica. Para el caso de este trabajo se han adquirido la licencia de controladores de la marca SIEMENS.

Configurando adecuadamente dicho controlador, este sabrá leer y escribir Entradas/Salidas desde el mismo.

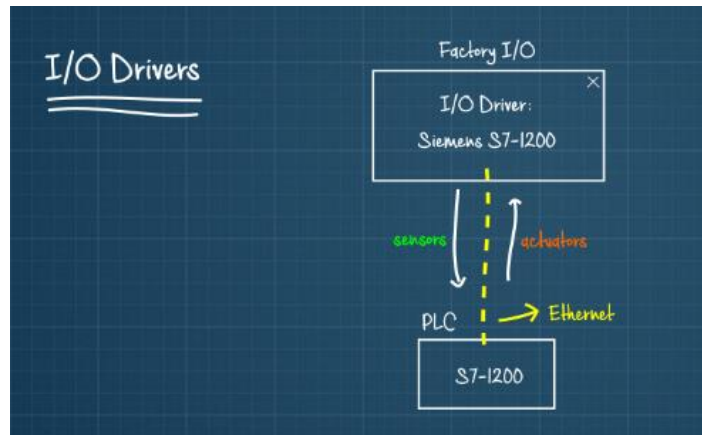


Figura 2.6. Esquema de comunicación entre FACTORY I/O y TIA PORTAL

2.2.5. PARTES

Se incluye una colección de piezas basadas en el equipo industrial más común. Hay más de 80 partes, organizadas en ocho categorías: artículos, piezas de carga pesada, partes de carga ligera, sensores, operadores, estaciones, dispositivos de aviso y pasarelas.



Figura 2.7. Elementos para la construcción de estaciones

2.2.6. ESCENAS

Como ya se comentó anteriormente FACTORY I/O proporciona una lista de escenas para usar inspiradas en aplicaciones industriales típicas de diferente nivel de dificultad.

3. PROCESO DE INSTALACIÓN DE FACTORY I/O

Debido a que FACTORY I/O es un software relativamente nuevo en el mercado se explica a continuación su proceso de instalación para poder trabajar con él de una forma óptima y desarrollar las estaciones descritas más adelante en este trabajo.

En primer lugar es necesario descargar el paquete de instalación del programa. Este paquete se puede descargar del enlace que RealGames te proporciona cuando has comprado la licencia del software o bien puedes descargarlo desde su página web si accedes a la versión trial del programa.



Figura 3.8. Enlace de descarga proporcionado por RealGames en la web

Una vez descargado el paquete ejecutamos el instalador y si el control de cuentas de usuario de nuestro ordenador lo solicita, otorgamos los permisos para la ejecución del instalador.

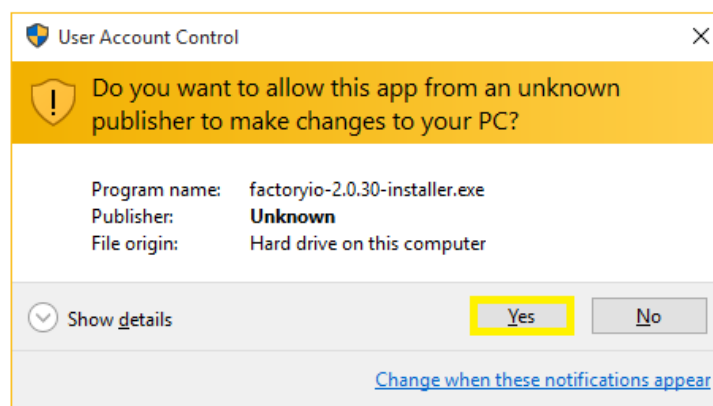


Figura 3.9. Pantalla de aviso del instalador

La siguiente pantalla que nos mostrará el programa nos permite seleccionar aquellos elementos que deseamos instalar en el ordenador en función del uso que vayamos a dar al software.

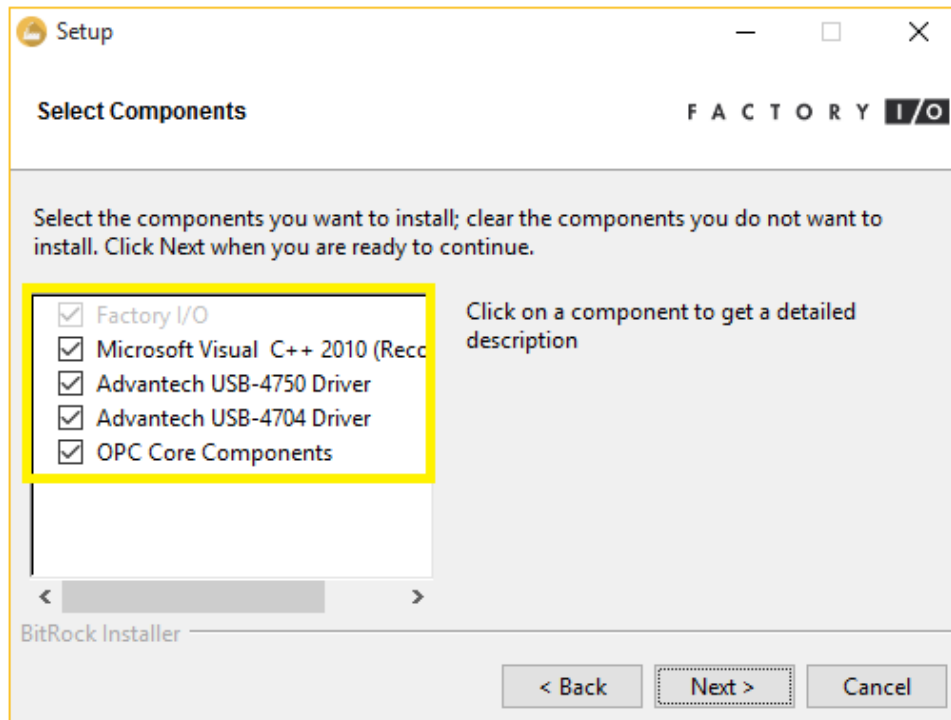


Figura 3.10. Pantalla de diferentes complementos a instalar

- El complemento **Microsoft Visual C++** es necesario para que FACTORY I/O se ejecute con total normalidad.
- **Advantech USB-4750 Driver** y **Advantech USB-4704 Driver** son necesarias instalarlas si se va a utilizar FACTORY I/O con las tarjetas Advantech USB 4750 o 4704 DAQ.
- **OPC Core Components** se deben seleccionar si se va a utilizar FACTORY I/O con el controlador de acceso de cliente OPC.

A partir de este punto se nos mostrará la barra instaladora del programa, indicándonos el estado en el que se encuentra el proceso de instalación.

Una vez ha finalizado la instalación del programa, es el momento de ejecutar FACTORY I/O haciendo doble click mediante su acceso directo.

A continuación nos dirigimos a **Opciones** en el menú **ARCHIVO**.



Figura 3.11. Menú FACTORY I/O

En el apartado Licencias ingresamos la clave de activación que nos ha proporcionado RealGames en el correo a la hora de comprar el software.

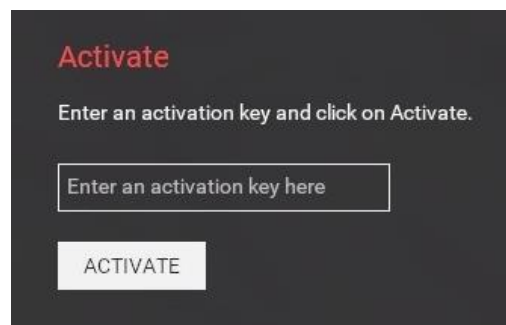


Figura 3.12. Pantalla de activación de licencia

Si no se encuentra licencia, FACTORY I/O inicia automáticamente una versión de prueba de 30 días.

4. CONFIGURACIÓN DE LA COMUNICACIÓN ENTRE FACTORY I/O Y TIA PORTAL CON PLC FÍSICO

Para que la simulación de las diferentes estaciones sea acorde al programa cargado en el PLC, es necesario configurar adecuadamente tanto el software FACTORY I/O como las características del PLC con el que vamos a trabajar.

4.1. CONFIGURACIÓN DE LAS CARACTERÍSTICAS DEL PLC

Las entradas físicas del PLC usan por defecto las primeras direcciones de memoria. Sin embargo para que FACTORY I/O pueda escribir los valores de los sensores, debe compensar las direcciones de entrada, por eso se recomienda un offset de 10.

Este valor se puede asignar en la ficha **General** en el apartado **Direcciones E/S**.

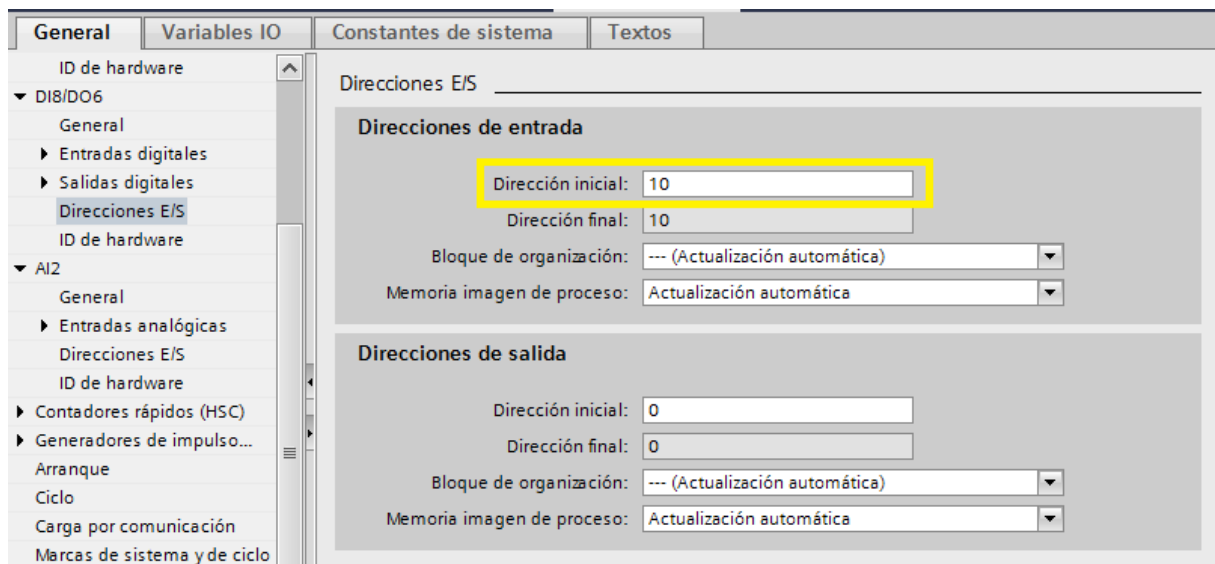


Figura 4.13. Configuración de las entradas del PLC

FACTORY I/O debe utilizar direcciones de entrada que no estén asignadas a entradas físicas. De lo contrario, los valores escritos por FACTORY I/O se sobrescribirán a medida que el estado de las entradas físicas se copie en la memoria.

Otra característica que es necesaria configurar es la del acceso completo a la memoria del PLC. De la misma forma que se ha comentado anteriormente, en la ficha **General**, en el apartado **Protección** seleccionamos **Acceso completo (sin protección)**. Además en **Mecanismos de conexión**, es necesario seleccionar **Permitir acceso con la comunicación PUT/GET desde el interlocutor remoto**.

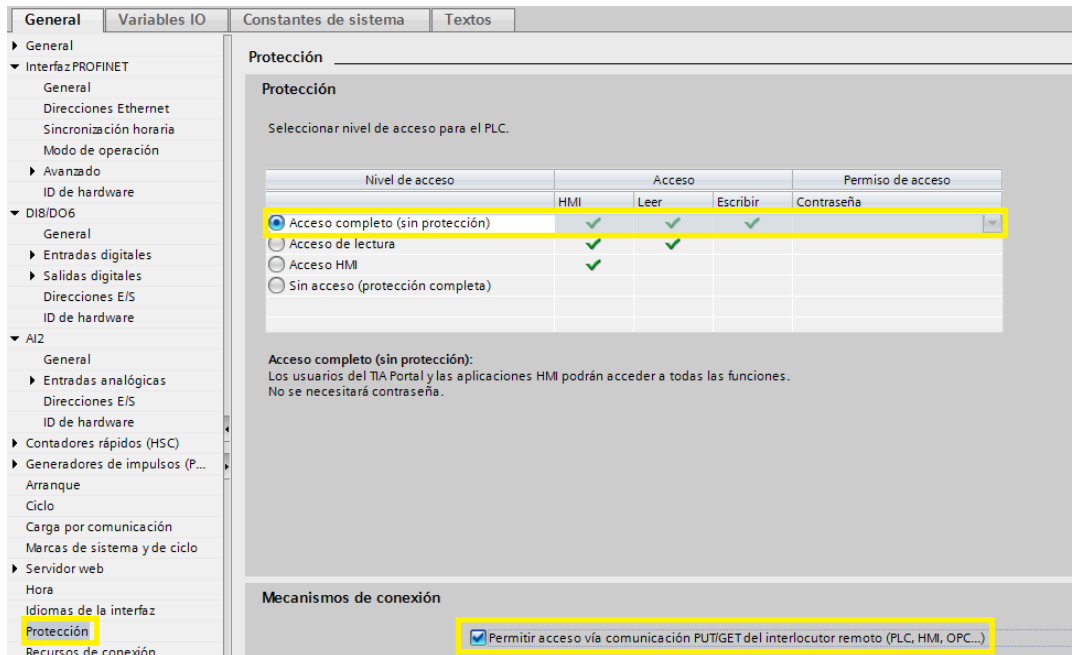


Figura 4.14. Configuración de acceso a la comunicación PUT/GET

4.2. CONFIGURACIÓN DE LOS DRIVERS DE FACTORY I/O

Para que FACTORY I/O pueda trabajar con el PLC que vamos a programar en TIA PORTAL, es necesario asignarle los drivers correspondientes que permitan la correcta comunicación. Para ello en el menú **ARCHIVO** seleccionamos el apartado **Drivers**.



Figura 4.15. Menú de FACTORY I/O

En la ventana que se nos despliega aparece una lista con los diferentes tipos de drivers que dispone FACTORY I/O para comunicarse con distintos tipos de autómatas.

Como en nuestro caso la programación se va a realizar con un PLC 1200 de la marca Siemens, escogemos el driver **Siemens S7-1200/1500**.

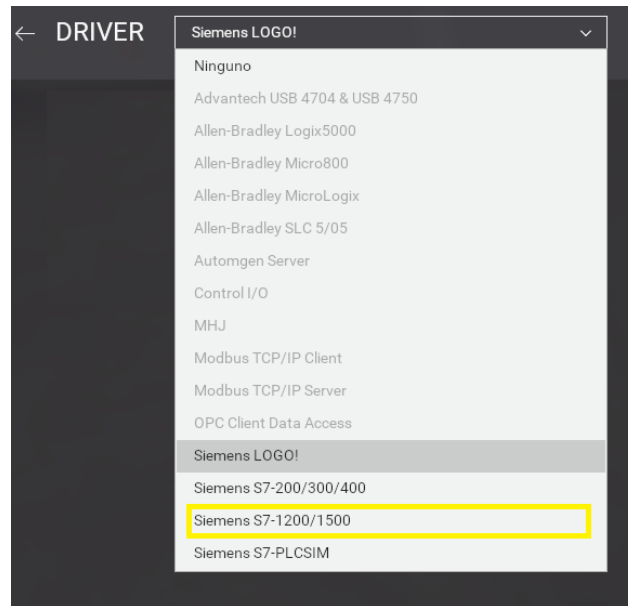


Figura 4.16. Selección de driver

Por otro lado es necesario que FACTORY I/O conozca cómo ha sido configurada la CPU del autómatas. Para ello es necesario abrir el panel de configuración del controlador haciendo click en **CONFIGURACIÓN**, donde se nos mostrará tres apartados diferenciados.

En la parte del Autómatas seleccionamos el **Modelo**, introducimos su dirección IP (**Host**) y seleccionamos el **Adaptador de Red** a través del cual se va a realizar la comunicación.



Figura 4.17. Configuración de dirección y adaptador de red

En **I/O CONFIGURACIÓN** para el **Tipo de dato numérico** escogemos el formato **WORD** si nuestra estación posee algún elemento que utilice valores de entrada o salida de tipo entero y **DWORD** si nuestra estación posee algún elemento que utilice valores de tipo real.

En **I/O TERMINALES** debemos elegir los bytes de entradas y de salidas acordes a los que hemos utilizado en la programación del PLC.

FACTORY I/O nos permite disponer de un máximo de 16 bits de entradas y salidas digitales y 8 bits de entradas y salidas analógicas.

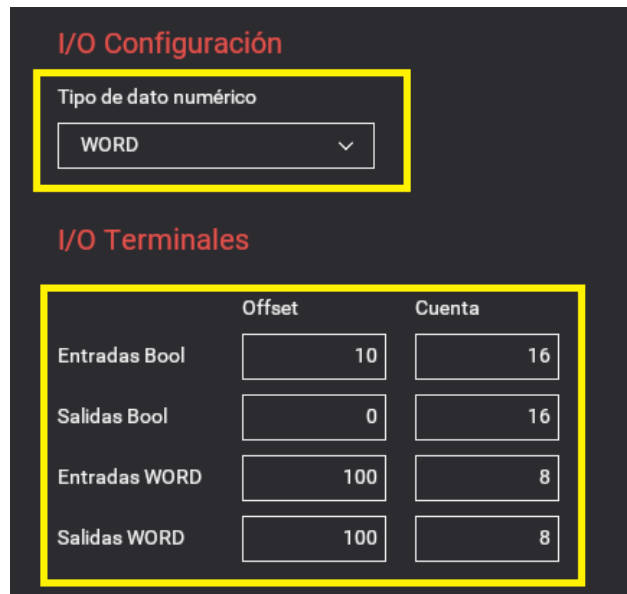


Figura 4.18. Configuración de las direcciones de entradas y salidas

Volviendo a la ventana principal del controlador, se puede apreciar la disposición final de entradas y salidas que tendrá nuestro autómatas. Finalmente para establecer la conexión con el PLC pulsamos **CONECTAR**.



Figura 4.19. Distribución final de entradas y salidas

Arrastrando y soltando se traza cada etiqueta al puerto deseado. Para quitar una etiqueta de un puerto se vuelve a arrastrar de nuevo a la lista

4.3. CONFIGURACIÓN DE LA RED LOCAL ENTRE PC Y AUTOMATA 1200

Para configurar adecuadamente la red local entre el PLC y el ordenador, es necesario conocer previamente cómo funciona el direccionamiento y encaminamiento en redes IP.

El protocolo IP usa direcciones IP para identificar cada una de las interfaces de comunicación de cualquier equipo participante de una red IP. En la mayoría de los casos basta con una única dirección para cada equipo y esta dirección es necesaria para encaminar los mensajes hacia el mismo.

Una dirección IP es un número binario de 32 bits o 4 octetos. Este número suele representarse en la notación punto, más fácil de leer y de escribir, en la que cada octeto de una dirección se convierte a su número decimal correspondiente.

El número máximo representado para cada octeto es 255, correspondiente al número binario 11111111.

El formato general de una dirección IP en el que se distinguen la dirección de red y la dirección local es el que se muestra a continuación:

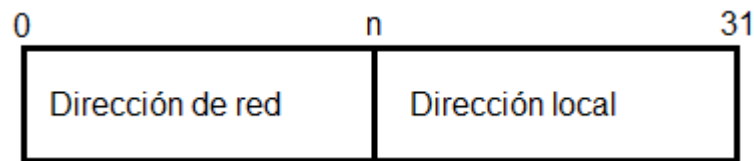


Figura 4.20. Esquema de dirección de red

La dirección de red identifica la red a la que está conectado el equipo y la dirección local identifica un nodo particular y específico dentro de la red. En esa red todos los equipos deben tener una dirección local única en el rango de la red para poder establecer comunicación con éxito.

Existen tres formatos diferentes de direcciones de red cuyas características son las siguientes:

Clase	Nº octetos parte de red	Primeros bits	Primer octeto	Número de direcciones locales
A	1	0	0-127	16.777.216
B	2	10	128-191	65.536
C	3	110	192-223	256

Además se han reservado varios bloques de direcciones para poder ser usadas en redes que no se van a poder conectar a Internet y que no necesitarán conectividad con otra organización. Estas direcciones son:

10.0.0.0-10.255.255.255

172.16.0.0-172.31.255.255

192.168.0.0-192.168.255.255

Estas direcciones no son encaminables en la red de Internet.

A menudo se utilizan las subredes o redes dentro de la propia red, que no es más que un segundo nivel lógico. Para ello, la parte local de la dirección IP se divide en una parte de subred y una parte de equipo.

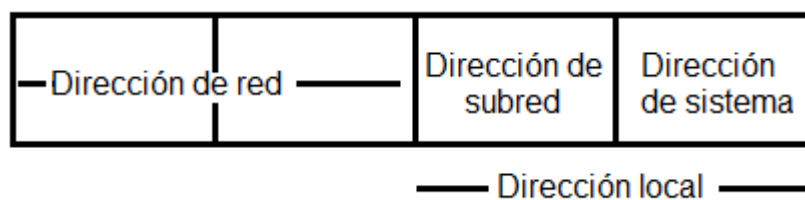


Figura 4.21. Esquema dirección de subred

En este caso el cuarto octeto es el que se usa para identificar los equipos particulares de una subred.

Para poder decir que un equipo está correctamente preparado desde el punto de vista de su comunicación en una red IP, debe haber sido configurado con:

- La dirección IP
- La máscara de red y subred, que es otro número de 32 bits que tiene tantos bits a 1, desde el inicio del número, como bits hay dedicados en la dirección IP a la parte de red.

Como en nuestro caso la red que vamos a crear, está compuesta únicamente por el ordenador y el autómata 1200, no es necesario que exista una conexión a Internet por lo que se utilizará una de las direcciones reservadas comentadas anteriormente. En este caso se ha elegido la dirección 192.160.0.0.

Para asignar la dirección IP que esté dentro de esa red a nuestro ordenador, debemos de acceder a las propiedades de conexión de área local, desde donde tenemos acceso a las propiedades del Protocolo de Internet (TCP/IPv4).

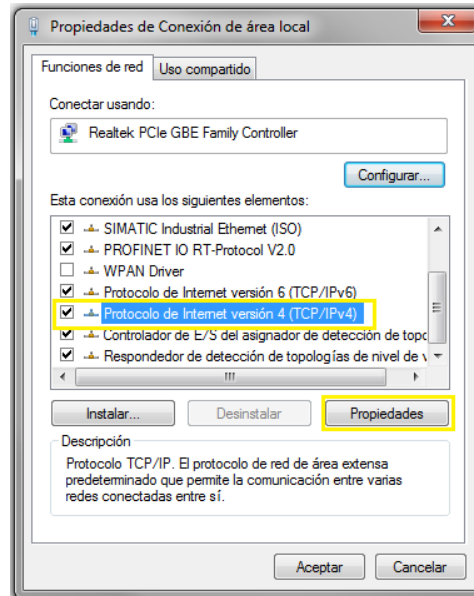


Figura 4.22. Ventana de configuración de propiedades de conexión

A continuación se nos abre una nueva ventana en la que ya podemos asignar la dirección de red deseada para nuestro ordenador con su dirección de subred asociada.

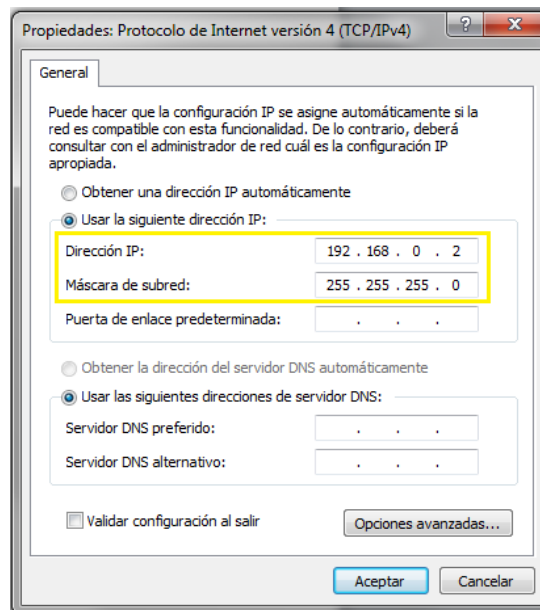


Figura 4.23. Ventana de configuración de dirección IP

Una vez que nuestro ordenador ya se encuentra en red, el siguiente paso es meter el PLC en la misma. Para ello empezamos creando un nuevo proyecto en TIA POTAL.

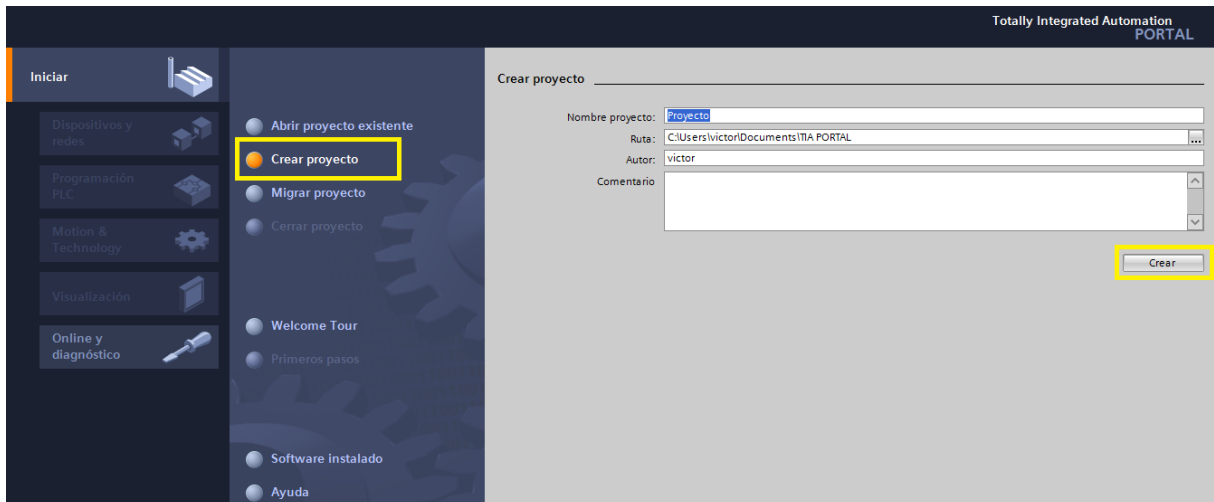


Figura 4.24. Pantalla de creación de nuevo proyecto

A continuación agregamos la CPU que vamos a programar y con la que realizaremos la automatización de las diferentes estaciones.

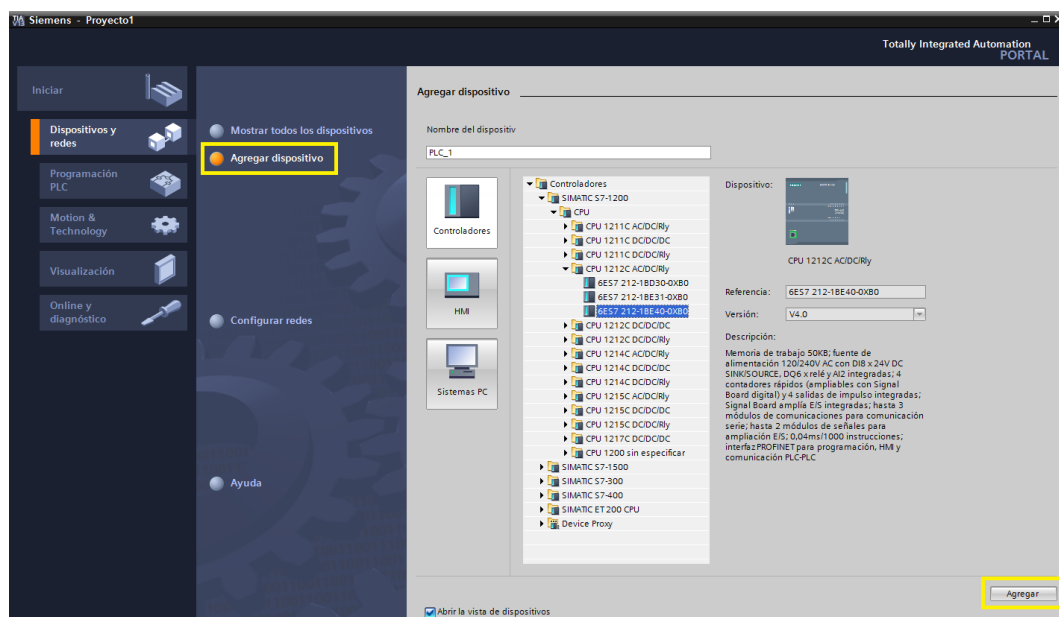


Figura 4.25. Selección de modelo de CPU

Ahora el PLC aparece en la vista de dispositivos, desde donde se puede acceder a sus propiedades haciendo doble click con el botón izquierdo.

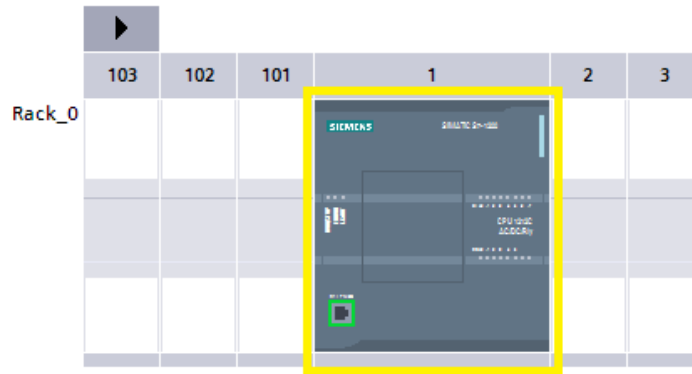


Figura 4.26. Visualización de la CPU en TIA PORTAL

Se muestra una ventana en la parte inferior desde donde en la ficha General en el apartado Direcciones Ethernet, se asigna la dirección IP y la máscara de subred apropiada.

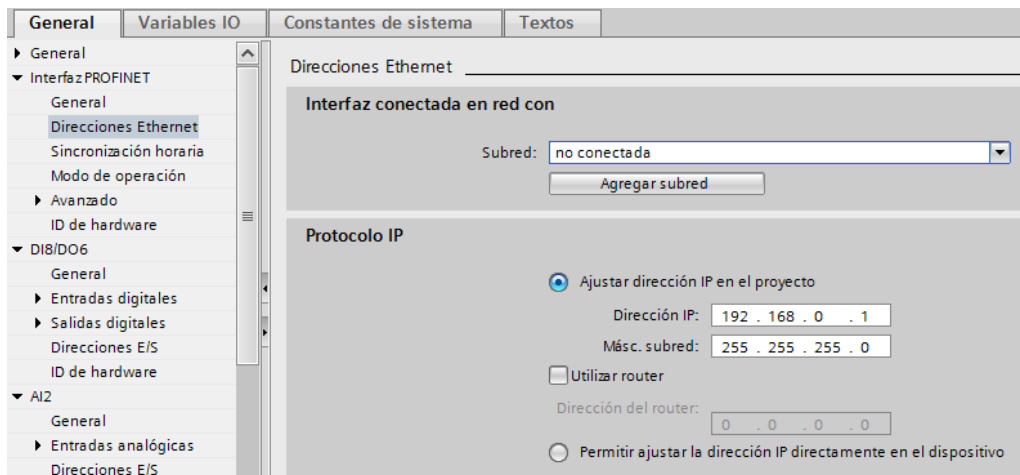


Figura 4.27. Configuración de la dirección de red de la CPU del automatista

5. CONFIGURACIÓN DE LA COMUNICACIÓN ENTRE FACTORY I/O Y TIA PORTAL CON PLCSIM V13/14

Factory I/O nos permite también simular el comportamiento de las estaciones sin necesidad de poseer el autómatas físico. Para ello se utiliza un software que sustituirá al PLC, denominado PLCSIM, que incorporan los fabricantes de Siemens en el propio paquete de TIA PORTAL, en sus versiones V13 o V14.

Sin embargo, debido a que Siemens todavía no ha liberado la API de estas nuevas versiones del PLCSIM, es necesario partir de una plantilla de programación de TIA PORTAL para crear nuestros propios programas, proporcionada por RealGames en su web.

Esta plantilla lo que contiene es un FC con una subrutina programada en SCL en su interior, que se encuentra situada en el OB1 y que permitirá la comunicación entre el PLCSIM y FACTORY I/O.

5.1. CONFIGURACION DE LAS CARACTERÍSTICAS DEL PLC

Los pasos para su configuración son muy similares a los descritos anteriormente para el caso del PLC físico, tal y como se muestra a continuación.

Lo primero es descargar la plantilla de programación de la página de RealGames, fabricante del software FACTORY I/O. Dependiendo de la versión que tengamos de TIA PORTAL el archivo a descargar será diferente.

A continuación, a partir de la plantilla descargada, podemos escribir nuestro código y guardar el proyecto con un nombre adecuado.

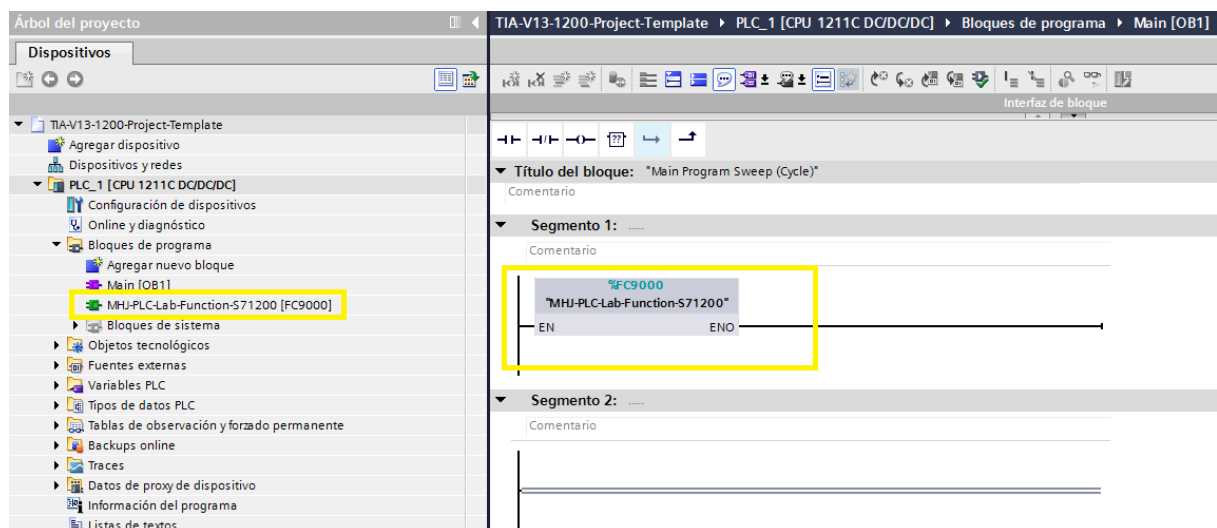


Figura 5.28. Plantilla de programación

Una vez concluida la programación del código, el siguiente paso es ejecutar la simulación. Para ello se selecciona el PLC que se quiere simular y se pulsa el botón de comenzar simulación.

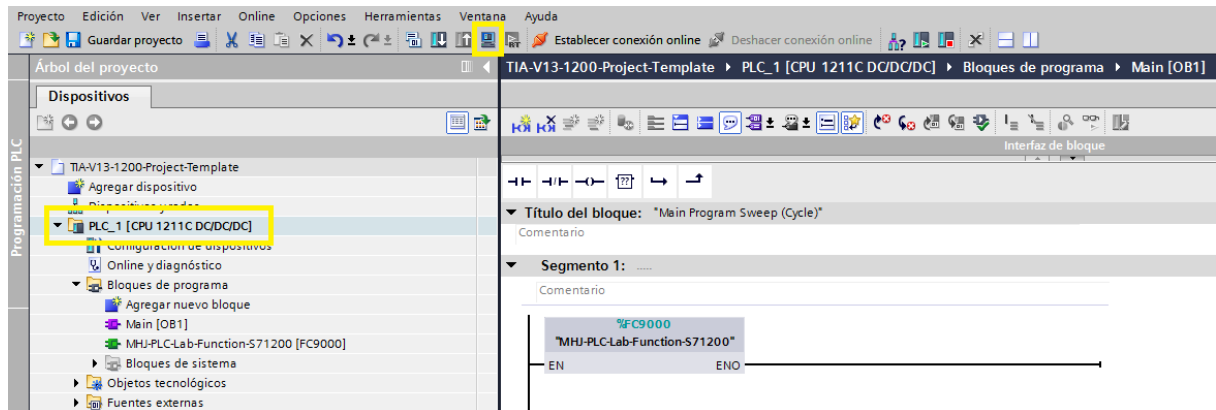


Figura 5.29. Inicio de simulación

Se abre una nueva ventana para configurar las opciones de comunicación, en la que debemos elegir las características que aparecen en la siguiente imagen si queremos que la comunicación se realice con éxito.

Una vez ha sido detectada la CPU del PLC cargamos todo el programa.

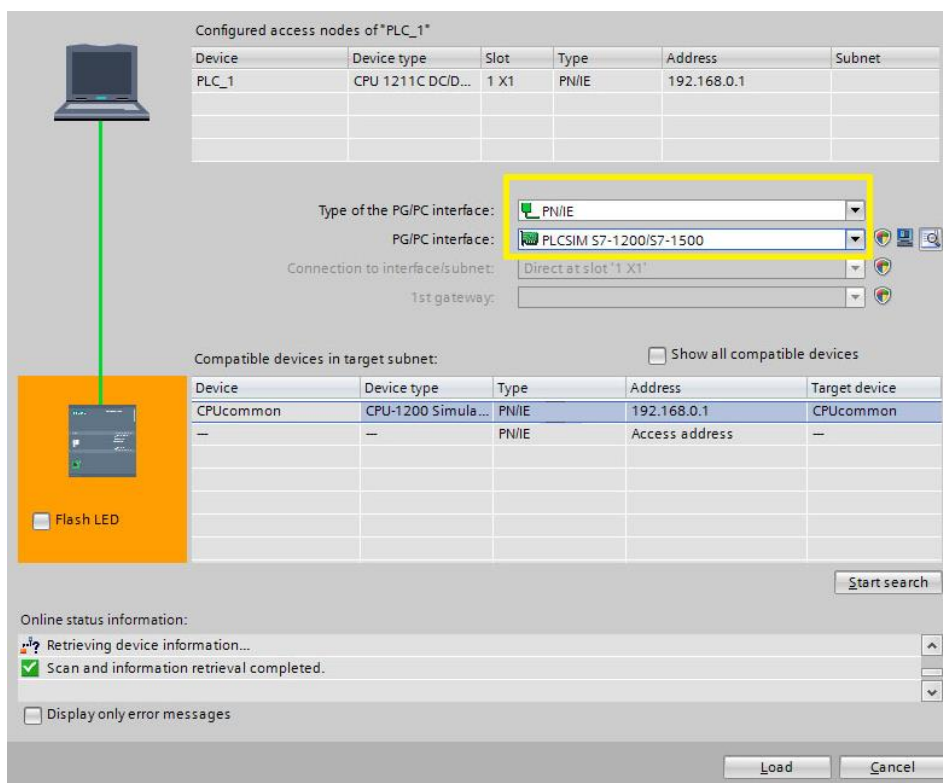


Figura 5.30. Configuración opciones de comunicación

Finalmente aparece la pantalla de ejecución del PLCSIM V13/14, lo que indica que la carga del programa se ha realizado satisfactoriamente y el programa se está ejecutando.

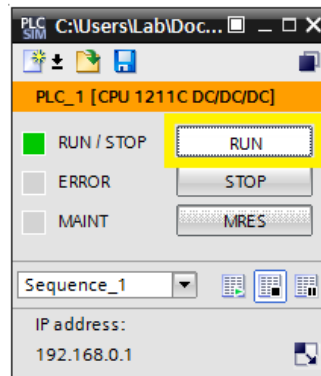


Figura 5.31. Pantalla de ejecución de PLCSIM

5.2. CONFIGURACION DE LOS DRIVERS DE FACTORY I/O

Por último falta configurar los drivers de FACTORY I/O que permitan la comunicación con el PLCSIM.

Por lo tanto el driver que debemos elegir es **Siemens S7-PLCSIM** ya que usaremos el simulador en vez del dispositivo físico, como ocurría en el caso anterior.

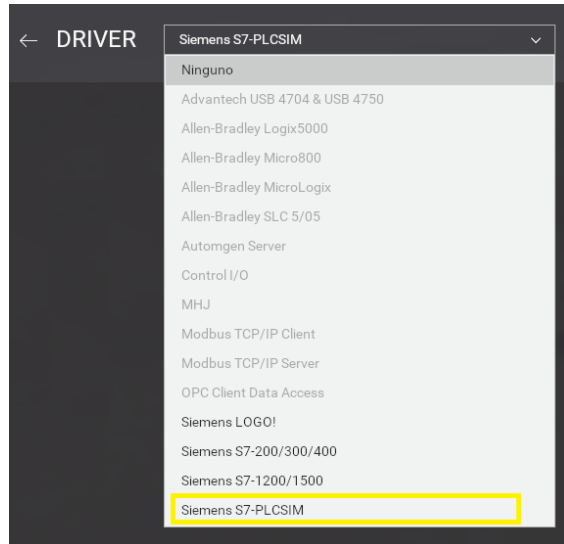


Figura 5.32. Selección de driver PLCSIM

Por otro lado es necesario que FACTORY I/O conozca el modelo de la CPU con la que se va a trabajar. Por lo tanto en el apartado de **CONFIGURACIÓN** seleccionamos el modelo que hayamos escogido en TIA PORTAL.

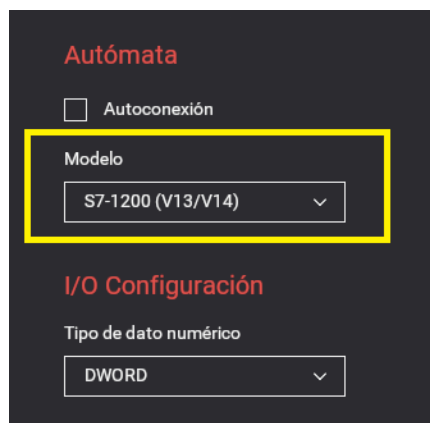


Figura 5.33. Selección de modelo de CPU

6. GRAFCET

Para realizar la mayor parte de la automatización de las diferentes estaciones de trabajo, se recurrió a la metodología del Grafcet, siendo este un método gráfico que permite representar el comportamiento de los sistemas secuenciales.

6.1. PRINCIPIOS GENERALES DEL GRAFCET

Un Grafcet es una sucesión de etapas que tienen acciones asociadas de forma que cuando una etapa está activa se realizan las correspondientes acciones. Por otro lado estas acciones no podrán ejecutarse nunca si la etapa no está activa.

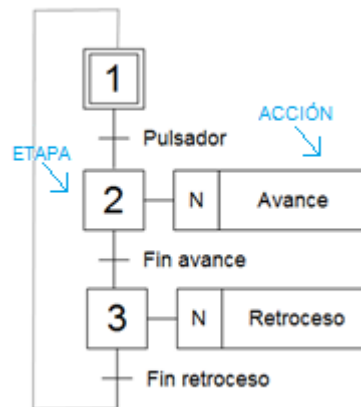


Figura 6.34. Esquema grafcet etapa/acción asociada

Entre dos etapas hay una transición y a cada una le corresponde una receptividad, que se trata de una condición que se ha de cumplir para poder pasar la transición. Una transición es válida cuando la etapa inmediatamente anterior a ella está activa. Si una transición es válida y su receptividad asociada se cumple se dice que la transición es franqueable.

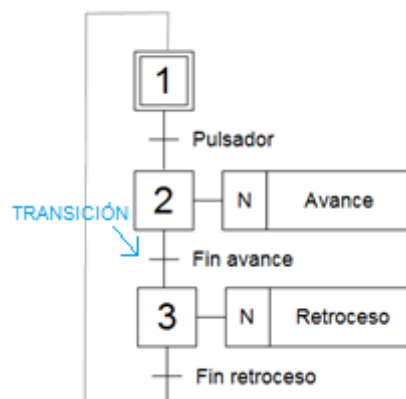


Figura 6.35. Esquema grafcet transición

Al franquear una transición se desactivan sus etapas anteriores y se activan las posteriores.

6.2. REGLAS DE EVOLUCIÓN DEL GRAFCET

Al llevar a la práctica un Grafcet se deben respetar una serie de reglas de evolución, ya que en caso contrario el funcionamiento del automatismo no sería el que cabría esperar a la vista del Grafcet representado.

- Regla 1: Inicialización

En la inicialización del sistema se han de activar todas las etapas iniciales y sólo las iniciales.

- Regla 2: Evolución de las transiciones

Una transición está validada cuando todas las etapas inmediatamente anteriores a ella están activadas. Una transición es franqueable cuando está validada y su receptividad asociada es cierta. Toda transición franqueable debe ser inmediatamente franqueada.

- Regla 3: Evolución de las etapas activas

Al franquear una transición se deben activar todas las etapas inmediatamente posteriores y desactivar simultáneamente todas las inmediatamente anteriores.

- Regla 4: Simultaneidad en el franqueamiento de las transiciones

Las transiciones simultáneamente franqueables han de ser simultáneamente franqueadas.

- Regla 5: Prioridad de la activación

Si al evolucionar un Grafcet, una etapa ha de ser activada y desactivada al mismo tiempo, deberá permanecer activa.

6.3. ELEMENTOS DEL GRAFCET

- Etapas

Una etapa del Grafcet se representa mediante un cuadrado identificado por un número. No puede haber dos etapas con el mismo número, además la entrada a una etapa es siempre por la parte superior y la salida por la parte inferior. Por otro lado las etapas pueden estar en dos estados activas e inactivas.

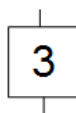


Figura 6.36. Etapa grafcet

Un cuadrado con una línea doble simboliza una etapa inicial del Grafcet. Las etapas iniciales son las que se activan al inicializar el Grafcet. Una vez se ha inicializado el Grafcet, las etapas iniciales actúan como etapas normales. Puede haber tantas etapas iniciales como se desee pero como mínimo una, además pueden estar situadas en cualquier lugar dentro del Grafcet.



Figura 6.37. Etapa inicial grafcet

- Transiciones

Las transiciones representan la posibilidad de evolución de una etapa a la siguiente; esta evolución se produce al franquear la transición. El franqueamiento de una transición implica un cambio en la situación de actividad de las etapas.

Las transiciones se representan con un trazo perpendicular a la línea que une dos etapas consecutivas. Una transición está validada cuando todas las etapas inmediatamente anteriores están activas.



Figura 6.38. Transición grafcet

6.4. ESTRUCTURAS BÁSICAS DE PROGRAMACIÓN EN GRAFCET

- Elección condicionada entre varias secuencias

Suele ocurrir que en un proceso se llegue a un punto del ciclo en el que hay que efectuar una elección entre varias secuencias posible, en función de las variables que intervienen en el proceso.

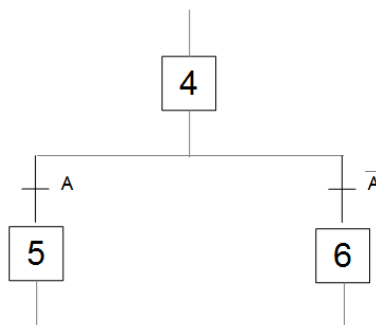


Figura 6.39. Inicio secuencia condicionada

Partiendo de la etapa 4 activada, se pueden realizar solamente una de las dos secuencias.

Si se cumple la transición A se activará la etapa 5.

Si se cumple la transición A negada, se activará la etapa 6.

La transición condicional implica que sólo una de las etapas posteriores se activará. Por lo tanto, la condición de transición asociada a la etapa 4 tiene que ser opuesta a la condición de transición asociada a la etapa 5.

La primera de las condiciones de transición que se cumpla desactivará la etapa 3.

El final de dos secuencias condicionadas se produce cuando una de las dos condiciones de transición asociadas a la etapa siguiente se cumple.

Por ejemplo en la imagen siguiente si la etapa 7 está activada y se cumple la condición de transición A, la etapa 9 se activará.

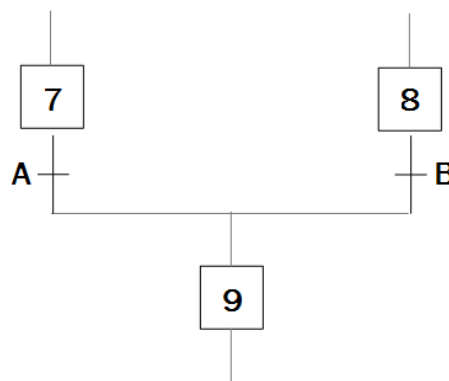


Figura 6.40. Final secuencia condicionada

- Secuencias simultaneas

Puede darse el caso de que sea necesario el desarrollo de más de una secuencia a la vez, cuyas etapas no tengan ninguna interrelación. Para poder representar este funcionamiento simultáneo, se utilizan un par de trazos paralelos que indican el principio y final de esta secuencia.

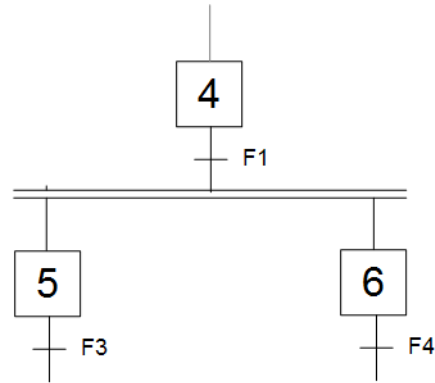


Figura 6.41. Secuencia simultánea

- Salto condicional a otra etapa

El salto condicional a otra etapa permite pasar de una etapa a otra sin activar las etapas intermedias. El salto condicional se puede hacer tanto en el sentido de evolución del graficet como en el sentido inverso. El sentido del salto viene indicado por las flechas.

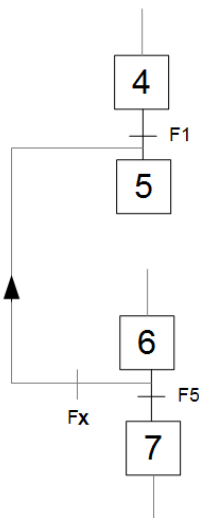


Figura 6.42. Salto condicional

7. ESTACIONES DE SIMULACIÓN

En este apartado vamos a detallar todo el proceso de simulación y programación de cada una de las estaciones que se han desarrollado, incluyendo el objetivo principal de cada estación y el resultado que se ha obtenido en cuanto al diseño y simulación de las mismas.

Se han desarrollado cuatro estaciones de trabajo, independientes unas de otras, cuyos elementos característicos son:

- **Estación paletizadora:** robot cartesiano.
- **Estación distribuidora:** selector de visión artificial.
- **Depósito de líquido:** válvulas de control mediante PID.
- **Estación de gestión de stock:** almacén dinámico.

7.1. ESTACIÓN PALETIZADORA

7.1.1. OBJETIVO DE LA ESTACIÓN

En esta estación se va a realizar la automatización de un proceso de paletizado en el que una serie de cajas serán distribuidas de una determinada forma sobre un palé.

La estación consta de una cinta transportadora por la que llegarán las cajas que deben ser posicionadas sobre el palé.

Al inicio de la cinta se encuentra una célula fotoeléctrica que si detecta la presencia de alguna caja, pone en marcha el motor de la misma, haciendo que la caja se desplace a lo largo de la cinta hasta que llegue al final, donde existe otra fotocélula que al detectar la caja detiene la cinta.

Por otro lado, también al final de la cinta, se encuentra un pistón cuya función es posicionar de forma adecuada la caja para que pueda ser accesible por el robot, de tal modo que al activarse la célula fotoeléctrica del final de la cinta se activa el retroceso del pistón que provoca el correcto posicionamiento de la pieza. Esto se consigue gracias a que el pistón posee un sensor de límite de retroceso que se activa cuando no puede retroceder más, ya sea porque ha llegado al final del recorrido de su vástago o porque se encuentra un objeto en medio de su recorrido. Cuando este sensor de límite de recorrido se activa, el pistón vuelve a su posición inicial.

Esta estación también consta de una cinta de rodillos a través de la cual se van desplazando los palés. Similar a lo que ocurría en la otra cinta transportadora, al principio de la misma se encuentra una célula fotoeléctrica que al detectar la presencia de un palé se pone en marcha, desplazándose el palé hasta llegar a la posición adecuada donde se van a distribuir las cajas. También a mitad de cinta se encuentra un rodillo de parada que se activa al ponerse en marcha la misma y que sirve para impedir

el avance del palé. En dicha posición se encuentra otra célula fotoeléctrica que al activarse por la presencia del palé, detiene el funcionamiento del motor de la cinta y desactiva el rodillo de parada.

De esta forma se consigue que el palé se posicione en el lugar adecuado para la realización de los movimientos del robot.

Por último, el elemento que es encargado de llevar a cabo todo el proceso de paletizado de cajas es un robot cartesiano.

Cuando una caja se encuentra posicionada ya sobre la cinta, el robot parte de una posición de reposo en la que se encuentra inicialmente y se desplaza sobre la caja para activar la ventosa de su brazo.

Una vez el robot ya tiene cogida la caja se desplaza sobre la posición del palé correspondiente donde tiene que posicionar la misma.

El robot distribuirá 6 cajas sobre cada palé, de tal forma que dependiendo del número de caja, el robot se situará sobre unas posiciones u otras.

Cuando el robot ha alcanzado la posición correspondiente sobre el palé desactiva su ventosa y la caja queda depositada.

Cuando un palé está completo se vuelve a poner en marcha la cinta de rodillos, desplazando el palé que ya se encuentra ocupado por cajas hacia el final de la cinta, al mismo tiempo que otro nuevo palé se coloca en la posición correspondiente para ser cargado por cajas.

Para gobernar el comportamiento de la estación se dispone de un cuadro de mando con los siguientes elementos:

- **Botón de marcha:** es necesario activarlo para que la estación comience su funcionamiento en el caso de que la misma se encuentre parada.
- **Botón de paro:** detiene el funcionamiento de la estación en cualquier estado en el que se encuentre la misma.
- **Botón de parada de emergencia:** detiene inmediatamente el funcionamiento de la cinta. Debido a su enclavamiento, cualquier acción que se realice sobre otro botón de cuadro de mando queda totalmente anulada mientras este permanezca en la posición de enclavamiento.

7.1.2. DISEÑO DE LA ESTACIÓN EN FACTORY I/O

La célula de trabajo ha sido diseñada para estar compuesta por sensores y actuadores cuya descripción y configuración se muestra a continuación:

ELEMENTO	CONFIGURACIÓN	ETIQUETA	I/O	TIPO	DESCRIPCIÓN
	digital	Motor cinta	output	bool	Desplazamiento de la cinta en la dirección de la flecha
	digital	Motor rodillos	output	bool	Desplazamiento de rodillos en la dirección de la flecha
	analógica	Cartesiano desplazamiento X	output	real	[0,10]V Establecer la posición en el eje X
		Cartesiano desplazamiento Y	output	real	[0,10]V Establecer la posición en el eje Y
		Cartesiano desplazamiento Z	output	real	[0,10]V Establecer la posición en el eje Z
		Giro pinza	output	bool	Giro de la pinza
		Succión	output	bool	Activar ventosa
		Cartesiano posición X	input	real	[0,10]V Posición actual en el eje X
		Cartesiano posición Y	input	real	[0,10]V Posición actual en el eje Y
		Cartesiano posición Z	input	real	[0,10]V Posición actual en el eje Z
		Neumática pistón	output	bool	Desplazamiento pistón
		Sensor pistón	input	bool	Artículo bloqueado o límite alcanzado

		Rodillo parada	output	bool	Elevado
		Sensor inicio cinta	input	bool	Detectando
		Sensor final cinta	input	bool	
		Sensor inicio rodillos	input	bool	
		Sensor final rodillos	input	bool	
		Emergencia	input	bool	Parada de emergencia activada
	Acción momentánea	Marcha	input	bool	Presionado
	Acción momentánea	Paro	input	bool	Presionado

Por otro lado en la estación de trabajo también se han incluidos elementos básicos para la generación y destrucción de las diferentes piezas que atraviesan el proceso.

Las características básicas de estos elementos se describen a continuación:

- **Emisor:** emite un elemento para ser utilizado en una escena. Mientras un elemento todavía está dentro del volumen del emisor, no se emiten más elementos. Se puede configurar la parte o la base a emitir, el tiempo entre emisiones, el número de elementos a emitir y si se debe tener en cuenta o no la posición y orientación.

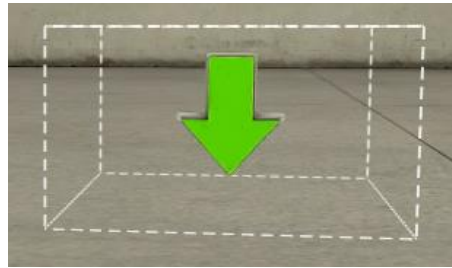


Figura 7.43. Elemento generador de piezas

- **Eliminador:** elimina uno o más artículos de la escena cuando intersectan el volumen del eliminador.

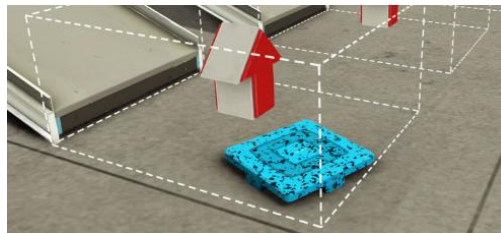


Figura 7.44. Elemento destructor de piezas

Finalmente, posicionando cada uno de los elementos que compone la estación en su lugar correspondiente acorde a los objetivos fijados, el resultado que se obtiene es el que se muestra a continuación:

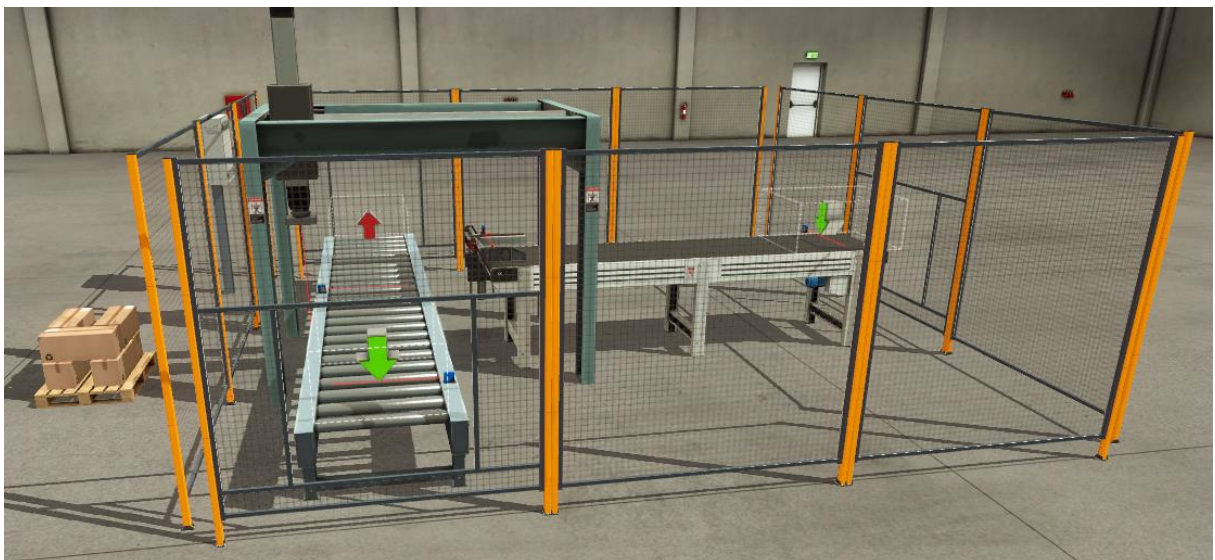


Figura 7.45. Vista lateral estación paletizadora



Figura 7.46. Vista frontal estación paletizadora



Figura 7.47. Cuadro de mando estación paletizadora

Por último para la correcta interpretación de la configuración de la estación, se muestra a continuación la distribución de las señales de todos los sensores y actuadores que intervienen en la estación, siendo la distribución física que tendría en las tarjetas del PLC la siguiente:

SENSORES		Host: 192.168.0.1		ACTUADORES	
EMERGENCIA	Sensor inicio cinta	I10.0	Q0.0	Motor cinta	Cartesiano Desplazamiento X(V)
Cartesiano Posición X(V)	Sensor final cinta	I10.1	Q0.1	Neumática pistón	Cartesiano Desplazamiento Y(V)
Cartesiano Posición Y (V)	Sensor piston	I10.2	Q0.2	succión	Cartesiano Desplazamiento Z(V)
Cartesiano Posición Z (V)	Sensor inicio rodillos	I10.3	Q0.3	Giro pinza	
FACTORY I/O (Paused)	Sensor final rodillos	I10.4	Q0.4	Motor rodillos	
FACTORY I/O (Reset)		I10.5	Q0.5	Rodillo parada	
FACTORY I/O (Running)		I10.6	Q0.6		
FACTORY I/O (Time Scale)		I10.7	Q0.7		
MARCHA	MARCHA	I11.0	Q1.0	Señalización marcha	
PARO	PARO	I11.1	Q1.1	Señalización paro	
Pick & Place 1 (C Limit)		I11.2	Q1.2		
Pick & Place 1 (Box Detected)	EMERGENCIA	I11.3	Q1.3		
Right Positioner 1 (Limit)		I11.4	Q1.4		
Sensor final cinta		I11.5	Q1.5		
Sensor final rodillos		I11.6	Q1.6		
Sensor inicio cinta		I11.7	Q1.7		
Sensor inicio rodillos		ID100 (REAL)	QD100	Cartesiano Desplazamiento X(V)	
Sensor piston		ID104 (REAL)	QD104	Cartesiano Desplazamiento Y(V)	
		ID108 (REAL)	QD108	Cartesiano Desplazamiento Z(V)	
		ID112 (REAL)	QD112		
		ID116 (REAL)	QD116		
		ID120 (REAL)	QD120		
		ID124	QD124		
		ID128	QD128		

Figura 7.48. Distribución entradas/salidas estación paletizadora

7.1.3. PROGRAMACIÓN SECUENCIAL GRAFCET

El control de esta estación se ha diseñado mediante el uso de grafkets parciales de nivel 1, que se centran en una descripción global del automatismo y que permiten rápidamente comprender su funcionamiento.

A continuación se muestran los grafkets de los principales elementos de la estación:

- **Transportador cinta:** con este grafket lo que se pretende es gobernar el funcionamiento de la cinta transportadora por la cual aparecerán de forma continuada las cajas que debemos distribuir sobre nuestro palé.

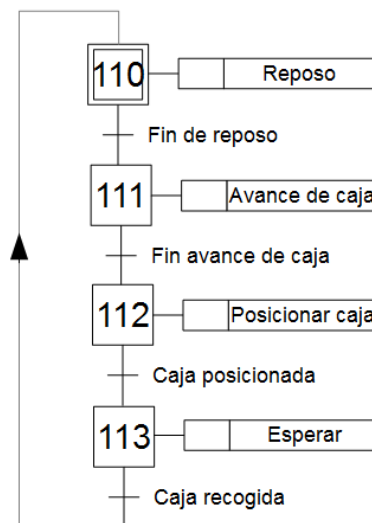


Figura 7.49. Grafket control cinta transportadora

- **Transportador rodillos:** en este caso el grafcet se corresponde a la automatización del transportador por el cual aparecen los palés que serán usados para cargar con las cajas que aparecen por la otra cinta de la estación.

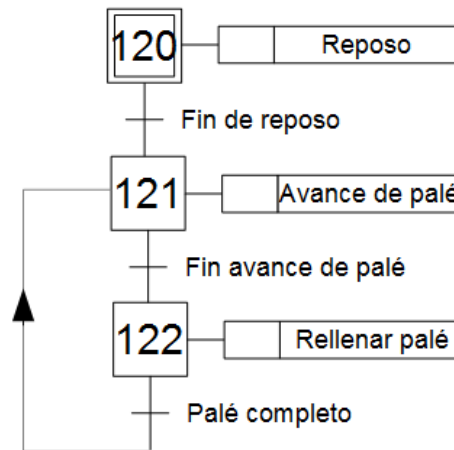


Figura 7.50. Grafcet control transportador rodillos

- **Cartesiano:** por último este grafcet está asociado al funcionamiento del robot cartesiano encargado de recoger las cajas que le llegan de la cinta transportadora y situarlas sobre el pale de una forma preestablecida.

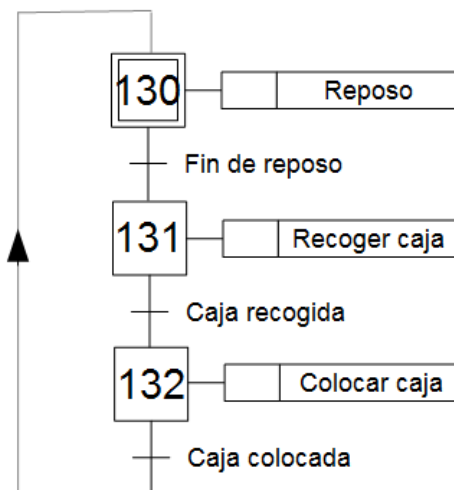


Figura 7.51. Grafcet control cartesiano

Es necesario destacar que todos los grafcets anteriores al tratarse de grafcets de nivel 1, como se comentó anteriormente, no describen de forma detallada el comportamiento de cada parte del sistema, lo que quiere decir que lo que aquí se representa mediante

una sola etapa, en un graficet de nivel 2 en el que se explica con mucho más detalle el comportamiento de cada parte y la tecnología utilizada, puede ocupar muchas más etapas.

7.1.4. PROGRAMACIÓN DEL PLC

La programación del autómatas se va a realizar mediante la utilización del lenguaje de contactos o KOP, siendo un lenguaje sencillo, intuitivo y fácil de interpretar.

Así mismo tal y como se comentó, las entradas y salidas utilizadas a la hora de realizar el programa en TIA PORTAL son las mismas que las utilizadas en la configuración de la estación en FACTORY I/O ya que sino no existiría una correcta comunicación entre ambas partes.

De tal forma que a continuación, se muestra en una tabla las entradas y salidas utilizadas en la elaboración del programa en TIA PORTAL.

Nombre	Tipo de dato	Dirección tarjeta del PLC
Sensor inicio cinta	Bool	%I10.0
Sensor final cinta	Bool	%I10.1
Sensor pistón	Bool	%I10.2
Sensor inicio rodillos	Bool	%I10.3
Sensor final rodillos	Bool	%I10.4
MARCHA	Bool	%I11.0
PARO	Bool	%I11.1
EMERGENCIA	Bool	%I11.2
Cartesiano Posición X	Real	%ID112
Cartesiano Posición Y	Real	%ID116
Cartesiano Posición Z	Real	%ID120

Nombre	Tipo de dato	Dirección tarjeta del PLC
Motor cinta	Bool	%Q0.0
Neumática pistón	Bool	%Q0.1
Succión	Bool	%Q0.2
Giro pinza	Bool	%Q0.3
Motor rodillos	Bool	%Q0.4
Rodillo parada	Bool	%Q0.5
Funcionando	Bool	%Q0.6
Señalización marcha	Bool	%Q1.0
Señalización paro	Bool	%Q1.1
Cartesiano Desplazamiento X	Real	%QD100
Cartesiano Desplazamiento Y	Real	%QD104
Cartesiano Desplazamiento Z	Real	%QD108

La lista de variables que se han usado en el programa es mucho mayor ya que TIA PORTAL necesita asociar una variable a cada elemento que almacene información. Por lo tanto llevarán variables asociadas elementos como temporizadores, contadores, bits que se activan con un flanco de subida o de bajada y que han sido utilizados para realizar la automatización de nuestra línea.

Para comprender un poco más a fondo el funcionamiento del programa de control que será cargado en el autómatas, se describen a continuación aquellos bloques de los que está compuesto.

La estructura general de bloques es la siguiente:

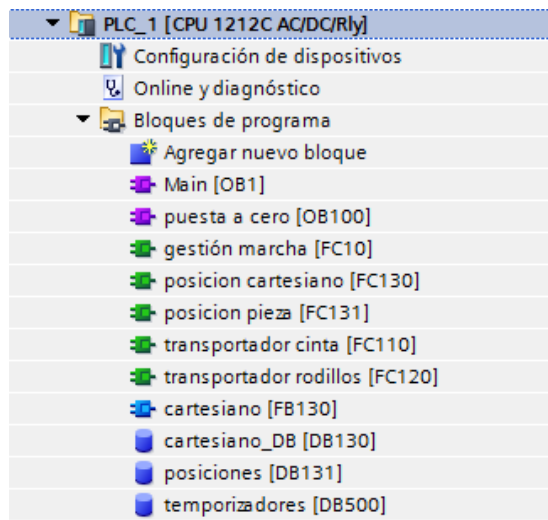


Figura 7.52. Distribución general de bloques. Estación paletizadora

Tal y como se aprecia en la imagen, existen dos bloques de organización:

- **Main [OB1]:** es el encargado de realizar las llamadas al resto de bloques que componen el programa.
- **Puesta a cero [OB100]:** este bloque es de tipo Startup, lo que se pretende con el uso de este bloque es que al cargar por primera vez el programa en el PLC, antes de que este pase a modo RUN se inicialicen todos los valores de variables tanto internas como de salidas.

Por otro lado se ha utilizado también un bloque de tipo FB que es el encargado de gestionar el control del robot cartesiano. Como todo bloque de tipo FB, tiene que tener asociado un DB de instancia donde se almacenen las variables de tipo “estáticas” utilizadas en el mismo. En nuestro caso, este DB es el DB130.

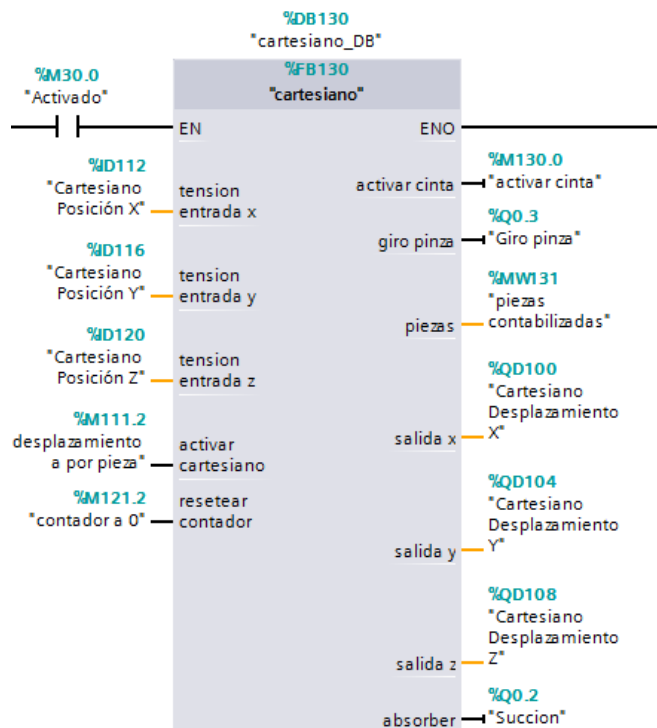


Figura 7.53. Bloque FB130 cartesiano

Este FB al encargarse del control del movimiento del robot tiene asociadas unas entradas y unas salidas correspondientes a las direcciones de entrada y salida del PLC que se encuentran conectadas con los sensores y actuadores del robot, encargados de transmitir y recibir las señales.

También este bloque posee otras entradas y salidas que se corresponden a señales de control entre otros bloques del PLC, ya que es necesario que el robot se comuniquen tanto con la cinta transportadora encargada de generar cajas, como con la cinta de rodillos que gestiona la distribución de palés.

Por último, relacionado con el control del robot cartesiano, se ha utilizado un bloque de datos globales, “posiciones [DB131]” en el que se encuentran almacenadas las posiciones a las cuales debe de desplazarse el robot tanto para coger una caja como para depositarla en el lugar correspondiente sobre el palé.

El valor de estas posiciones se corresponde con la tensión que es necesaria aplicar a cada uno de los motores del robot para que su brazo alcance las coordenadas del lugar deseado.

El otro tipo de bloques que ha sido usado para la elaboración del programa ha sido los bloques FC.

El uso de este tipo de bloques se destina a la ordenación del código del programa o a la realización de cálculos que no requieran el almacenamiento de la información entre ciclo y ciclo de ejecución del PLC.

Entre los bloques de este tipo se encuentran:

- **Gestión marchar [FC10]:** este bloque se encarga del control del cuadro de mando de la estación, gestionando de este modo la marcha, el paro y la parada de emergencia de la misma.

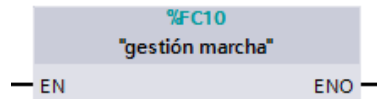


Figura 7.54. Bloque FC10 gestión marcha

- **Posición cartesiano [FC130]:** este bloque transforma los valores de posición leídos por los sensores del robot, en otros cuya precisión sea menor con el fin de comparar dichos valores con los almacenados en el DB de posiciones y que el robot alcance la posición deseada con éxito.

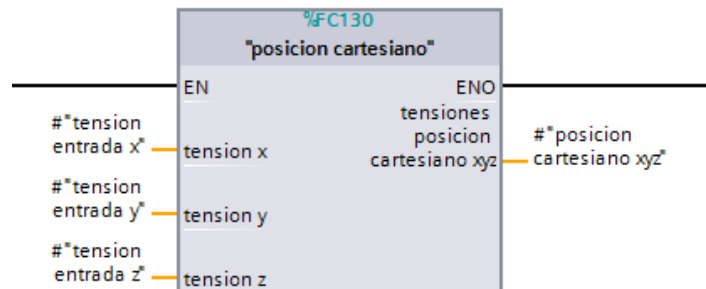


Figura 7.55. Bloque FC130 posición cartesiano

- **Posición pieza [FC131]:** la función principal de este bloque reside en que cada caja que llega por la cinta transportadora tiene una posición específica dentro del palé, por lo que es necesario indicarle al robot hacia que posición tiene que desplazarse en función de la caja que llegue por la cinta.

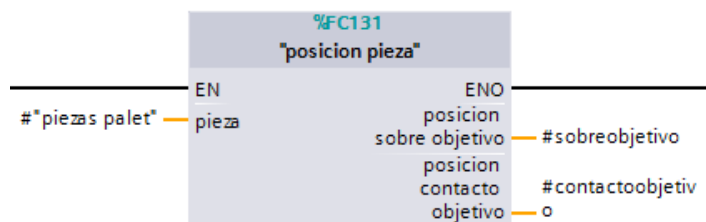


Figura 7.56. Bloque FC131 posición pieza

- **Transportador cinta [FC110]:** esta función gobierna el comportamiento de la cinta transportadora sobre la cual llegan las diferentes cajas y el comportamiento del pistón de final de línea que posicionará las cajas en el lugar adecuado para que sean accesibles por el robot cartesiano.

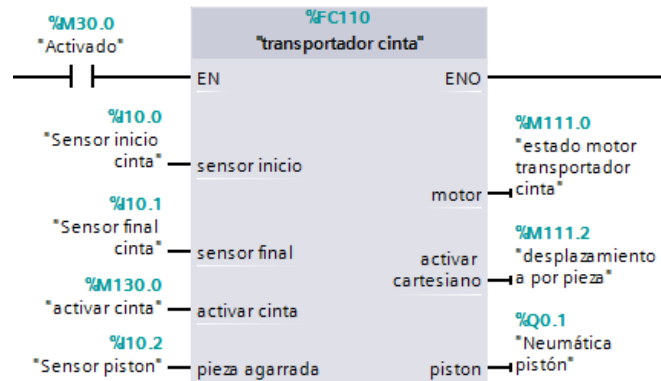


Figura 7.57. Bloque FC110 transportador cinta

Cuenta con una serie de entradas y de salidas, asociadas a los diferentes tipos de sensores encargados de detectar la posición de las cajas a lo largo de la cinta y a los actuadores encargados de poner en movimiento la cinta y de colocar en la posición adecuada la caja.

A su vez también posee entradas y salidas de control que le permiten establecer comunicación con otros bloques, consiguiendo de este modo un comportamiento global controlado.

- **Transportador rodillos [FC120]:** este bloque se encarga de controlar el comportamiento de la cinta de rodillos sobre la cual irán apareciendo los diferentes palés.

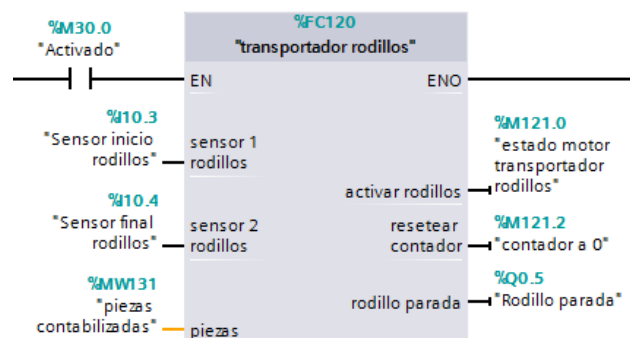


Figura 7.58. Bloque FC120 transportador rodillos

De forma similar a lo que sucedía con bloques anteriores, cuenta con entradas y salidas asociadas a las posiciones de entradas y salidas del PLC, que se encuentran "cableadas" con los sensores y actuadores que me permiten detectar la presencia del palé en diferentes lugares de la cinta y gestionar su desplazamiento.

También el bloque posee entradas y salidas de control ya que se tiene que comunicar con el FB del robot cartesiano para que el comportamiento de la estación sea el deseado.

Por último se ha utilizado un bloque de datos globales, denominado "temporizadores [DB500]" en el que se encuentran todos los temporizadores usados a lo largo del programa.

7.2. ESTACIÓN DISTRIBUIDORA

7.2.1. OBJETIVO DE LA ESTACIÓN

En esta estación se va a realizar la automatización de un proceso en el cual dos tipos diferentes de piezas serán distribuidas y almacenadas en cajas de cuatro unidades cada una para su posterior transporte.

El inicio de la estación está constituido por una cinta transportadora sobre la cual llegarán los diferentes tipos de piezas. Una vez que la estación ha sido puesta en marcha esta cinta estará constantemente funcionando y aparecerá una nueva pieza sobre la misma en un intervalo fijo de tiempo, sin embargo el tipo de pieza es aleatorio, es decir no se puede saber si aparecerán dos piezas del mismo tipo seguidas o si los diferentes tipos de piezas aparecerán de forma alterna.

Al final de esta cinta se encuentra un clasificador de ruedas, al que se le ha incorporado un sensor de visión artificial, de tal forma que si el sensor detecta que la pieza es de un tipo determinado, activa el clasificador desplazando la pieza situada sobre él hacia un lado u otro del mismo.

A ambos lados del clasificador se encuentran dos líneas distribuidoras de producto independientes una de la otra.

Al final de cada línea se encuentra una barrera de parada que detiene el desplazamiento de la pieza, colocándola en la posición adecuada para que posteriormente pueda ser alcanzada por el robot. De tal modo que cuando el sensor de visión identifica que la pieza es de un tipo determinado, pone en marcha la línea distribuidora correspondiente a ese tipo de pieza, activando a su vez la barrera de parada.

Junto a esta barrera, se encuentra una fotocélula que detecta la presencia de la pieza en la posición correcta, haciendo que se detenga la cinta y enviando una señal al robot indicándole que ya puede desplazarse a por la pieza.

Cada robot tiene asociado otra cinta transportadora sobre la cual aparecerán las cajas en las que se van a almacenar hasta cuatro unidades de piezas de cada tipo. Cuando se pone en marcha la estación esta cinta empieza a moverse desplazándose la caja a lo largo de la misma hasta que llega al final, donde se encuentra activada una barrera de parada que detiene el avance de la caja. Al lado de la barrera de parada se encuentra un sensor capacitivo que al detectar la presencia de la caja envía una señal al robot, indicándole que la caja se encuentra en la posición apropiada para que se introduzcan las piezas en ella.



Cuando una caja se ha completado con cuatro piezas del mismo tipo, la barrera de parada que impedía el desplazamiento de la caja se desactiva y se pone en marcha el motor de la cinta sobre la cual estaba la caja, haciendo que esta se desplace y desaparezca al final de la línea, volviendo otra caja vacía a ocupar la posición abandonada por la caja anterior para ser nuevamente ocupada por piezas.

Para gobernar el comportamiento de la estación se dispone de un cuadro de mando con los siguientes elementos:

- **Botón de marcha:** es necesario activarlo para que la estación comience su funcionamiento en el caso de que la misma se encuentre parada.
- **Botón de paro:** detiene el funcionamiento de la estación en cualquier estado en el que se encuentre la misma.
- **Botón de parada de emergencia:** detiene inmediatamente el funcionamiento de la cinta. Debido a su enclavamiento, cualquier acción que se realice sobre otro botón de cuadro de mando queda totalmente anulada mientras este permanezca en la posición de enclavamiento.
- Botón de reset de contadores: **pone a cero el valor de los contadores de los diferentes** tipos de piezas.

7.2.2. DISEÑO DE LA ESTACIÓN EN FACTORY I/O

En este caso, la célula de trabajo ha sido diseñada por un conjunto de elementos de los cuales sus características y configuración se muestran a continuación:

ELEMENTO	CONFIGURACIÓN	ETIQUETA	I/O	TIPO	DESCRIPCIÓN
	digital	Cinta piezas	output	bool	Desplazamiento de la cinta en la dirección de la flecha
		Cinta piezas azules			
		Cinta piezas verdes			
		Cinta1 cajas azules			
		Cinta2 cajas azules			
		Cinta1 cajas verdes			
Cinta2 cajas verdes					
	digital	Elevación distribuidor	output	bool	Elevación y rotación
		Desplazamiento derecha distribuidor	output	bool	Giro a la derecha
		Desplazamiento izquierda distribuidor	output	bool	Giro a la izquierda

		Pistón stop azules	output	bool	Elevado
		Pistón stop verdes	output	bool	
		Pistón stop caja azul	output	bool	
		Pistón stop caja verde	output	bool	
	analógica	Motor X robot verde	output	real	[0,10]V Establecer la posición en el eje X
		Motor Z robot verde	output	real	[0,10]V Establecer la posición en el eje Z
		Sensor X robot verde	input	real	[0,10]V Posición actual en el eje X
		Sensor Z robot verde	input	real	[0,10]V Posición actual en el eje Z
		Motor X robot azul	output	real	[0,10]V Establecer la posición en el eje X
		Motor Z robot azul	output	real	[0,10]V Establecer la posición en el eje Z
		Sensor X robot azul	input	real	[0,10]V Posición actual en el eje X
		Sensor Z robot azul	input	real	[0,10]V Posición actual en el eje Z
		Absorción robot verde	output	bool	Activar ventosa
		Absorción robot azul	output	bool	Activar ventosa
				Fotocélula piezas verdes	input
Fotocélula piezas azules	input			bool	
	Todo numérico	Sensor de visión	input	int	Valor que representa al elemento detectado
	Digital	Capacitivo cajas azules	input	bool	Detectando
		Capacitivo cajas verdes	input	bool	

		Emergencia	input	bool	Parada de emergencia activada
	Acción momentánea	Marcha	input	bool	Presionado
	Acción momentánea	Paro	input	bool	Presionado
	Acción momentánea	Reset contadores de cajas	input	bool	Presionando
	Entero	Número de cajas verdes	output	int	Valor del display
		Número de cajas azules	output	int	

Del mismo modo a como se realizó con la estación anterior, en esta ocasión se han vuelto a incluir los elementos proporcionados por FACTORY I/O para la generación y destrucción de las piezas, como son el emisor y el eliminador, que nos permiten simular de una forma continuada el funcionamiento de la estación.

Distribuyendo de manera conveniente todos los elementos citados anteriormente con el fin de simular el objetivo de funcionamiento de la estación el resultado obtenido es el que se muestra a continuación:

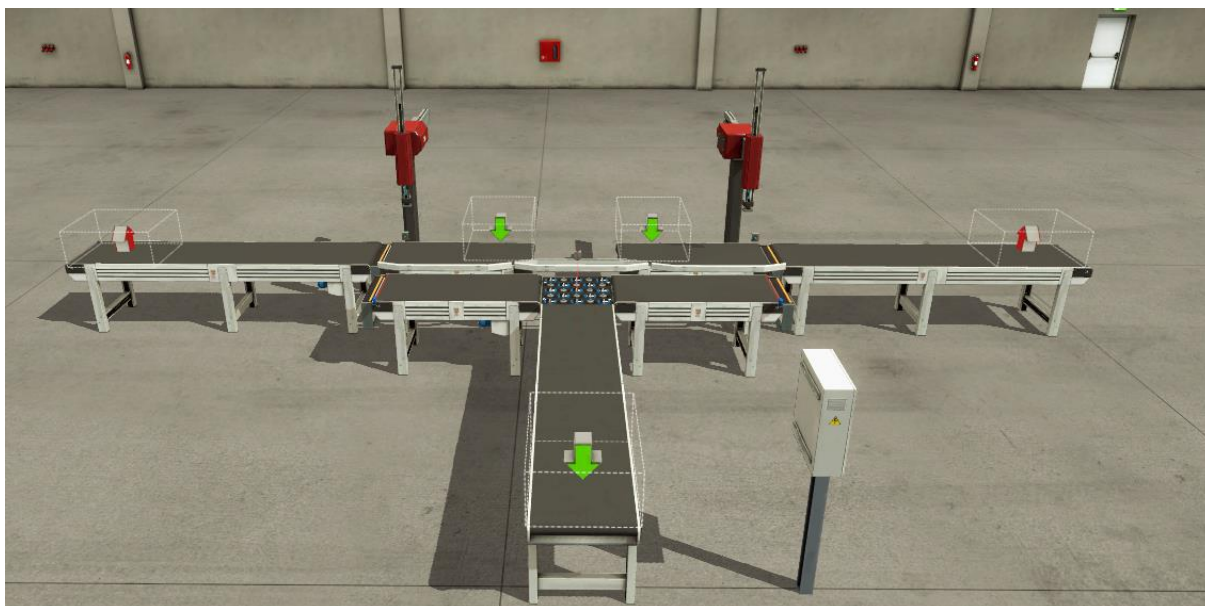


Figura 7.59. Vista frontal estación distribuidora



Figura 7.60. Vista lateral derecha estación distribuidora



Figura 7.61. Vista lateral izquierda estación distribuidora



Figura 7.62. Cuadro de mando estación distribuidora

Por último para el correcto entendimiento de la configuración de la estación, se muestra a continuación la distribución de las señales de todos los sensores y actuadores que intervienen en la estación, siendo la distribución física que tendría en las tarjetas del PLC la siguiente:

SENSORES		Host: 192.168.0.1		ACTUADORES	
capacitivo cajas azules	fotocélula piezas verdes	I10.0	Q0.0	Elevación distribuidor	Absorción robot azul
capacitivo cajas verdes	capacitivo cajas verdes	I10.1	Q0.1	Desplazamiento derecha distribuidor	Absorción robot verde
Emergencia		I10.2	Q0.2	Desplazamiento izquierda distribuidor	Cinta1 cajas azules
FACTORY I/O (Paused)	fotocélula piezas azules	I10.3	Q0.3	Cinta piezas verdes	Cinta1 cajas verdes
FACTORY I/O (Reset)	capacitivo cajas azules	I10.4	Q0.4	Pistón stop verdes	Cinta2 cajas azules
FACTORY I/O (Running)		I10.5	Q0.5	Cinta1 cajas verdes	Cinta2 cajas verdes
FACTORY I/O (Time Scale)	Marcha	I10.6	Q0.6	Cinta2 cajas verdes	Cinta piezas
fotocélula piezas azules	Paro	I10.7	Q0.7	Pistón stop caja verde	Cinta piezas azules
fotocélula piezas verdes	Reset contadores de cajas	I11.0	Q1.0	Cinta piezas azules	Cinta piezas verdes
Marcha	Emergencia	I11.1	Q1.1	Pistón stop azules	Desplazamiento derecha distribuidor
Paro		I11.2	Q1.2	Cinta1 cajas azules	Desplazamiento izquierda distribuidor
Reset contadores de cajas		I11.3	Q1.3	Cinta2 cajas azules	Elevación distribuidor
sensor de visión	sensor de visión	I11.4	Q1.4	Pistón stop caja azul	Emitter 1 (Emit)
sensor X robot azul	sensor X robot azul	I11.5	Q1.5	Cinta piezas	Emitter 2 (Emit)
sensor X robot verde	sensor X robot verde	I11.6	Q1.6	Absorción robot verde	Emitter 3 (Emit)
sensor Z robot azul	sensor Z robot azul	I11.7	Q1.7	Absorción robot azul	Motor X robot azul
sensor Z robot verde	sensor Z robot verde	ID100 (DINT)	(DINT) QD100	Número de cajas azules	Motor X robot verde
vo-Axis Pick & Place 1 (Item Detected)		ID104 (DINT)	(DINT) QD104	Número de cajas verdes	Motor Z robot azul
vo-Axis Pick & Place 2 (Item Detected)		ID108	QD108		Motor Z robot verde
		ID112	QD112		Motor Z robot azul
		ID116 (REAL)	(REAL) QD116	Motor X robot azul	Número de cajas azules
		ID120 (REAL)	(REAL) QD120	Motor Z robot azul	Número de cajas verdes
		ID124 (REAL)	(REAL) QD124	Motor X robot verde	Pistón stop azules
		ID128 (REAL)	(REAL) QD128	Motor Z robot verde	Pistón stop caja azul

Figura 7.63. Distribución entradas/salidas estación distribuidora

7.2.3. PROGRAMACIÓN SECUENCIAL GRAFCET

El diseño del control de la estación se ha realizado mediante el uso de grafkets parciales de nivel 1 que se centran en una descripción global del automatismo y que permiten rápidamente comprender su funcionamiento, sin entrar en detalles sobre qué elementos se van a utilizar para llevar a cabo dicha automatización.

A continuación se muestran los grafkets de los principales elementos de la estación:

- **Distribuidor piezas:** con este grafket lo que se pretende es demostrar la forma de actuación que tendrá el distribuidor sobre el cual irán apareciendo de forma continuada piezas y gracias a la cámara de visión las distribuirá hacia un lado u otro.

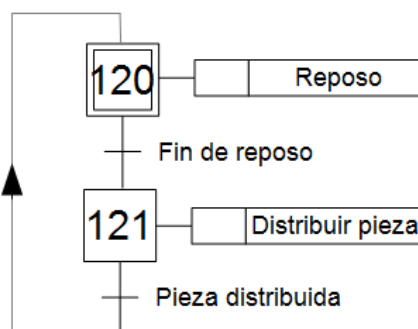


Figura 7.64. Grafket distribuidor piezas

- **Cinta recoger piezas:** en este caso el grafcet se corresponde con la automatización de las cintas por las que únicamente circularán un solo tipo de piezas y que serán recogidas por el robot para su posterior almacenaje en cajas.

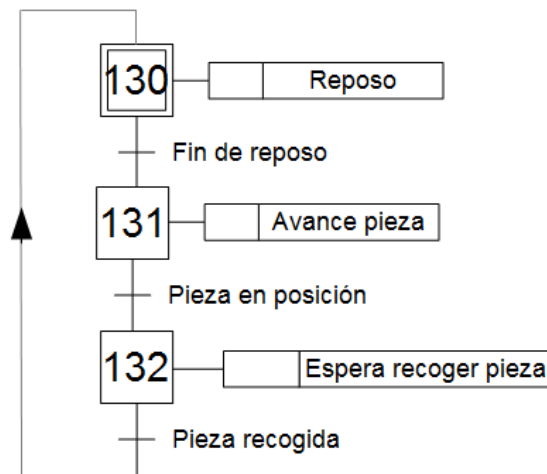


Figura 7.65. Grafcet cinta recoge piezas

- **Cinta cajas:** mediante este grafcet se pretende gobernar el funcionamiento de la cinta por la que circularán las cajas vacías que serán completadas por las piezas que recogen los robots.

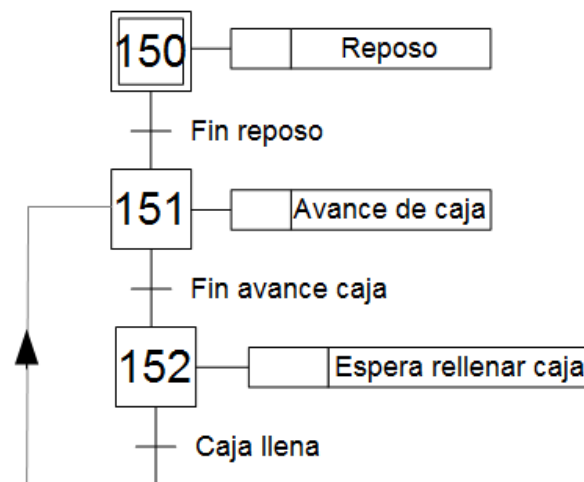


Figura 7.66. Grafcet cinta cajas

- **Robot automático:** por último este grafcet se corresponde con el funcionamiento de los robots encargados de recoger las piezas de un determinado tipo y de depositarlas en su caja correspondiente.

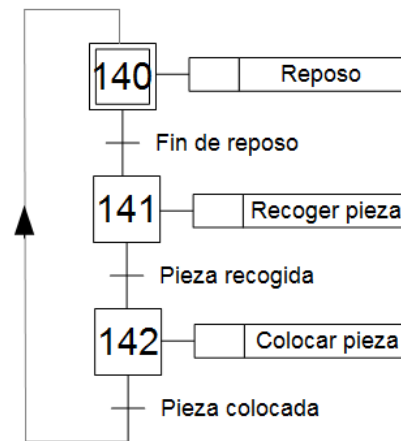


Figura 7.67. Grafcet robot automático

7.2.4. PROGRAMACIÓN DEL PLC

De la misma forma a como se ha realizado en la estación anterior, la programación del autómatas se va a desarrollar mediante la utilización del lenguaje de contactos o KOP.

Así mismo tal y como se comentó, las entradas y salidas utilizadas a la hora de realizar el programa en TIA PORTAL son las mismas que las utilizadas en la configuración de la estación en FACTORY I/O ya que sino no existiría una correcta comunicación entre ambas partes.

De tal forma que a continuación se muestra en una tabla las entradas y salidas utilizadas en la elaboración del programa en TIA PORTAL.

Nombre	Tipo de dato	Dirección tarjeta del PLC
capacitivo piezas verdes	Bool	%I10.0
capacitivo caja verdes	Bool	%I10.1
capacitivo piezas azules	Bool	%I10.4
capacitivo cajas azules	Bool	%I10.5
Marcha	Bool	%I11.0
Paro	Bool	%I11.1
Reset contadores de cajas	Bool	%I11.2
Emergencia	Bool	%I11.3
sensor de visión	DWord	%ID100
sensor X robot azul	Real	%ID116
sensor Z robot azul	Real	%ID120
sensor X robot verde	Real	%ID124
sensor Z robot verde	Real	%ID128

Nombre	Tipo de dato	Dirección tarjeta del PLC
elevación distribuidor	Bool	%Q0.0
desplazamiento derecha distribuidor	Bool	%Q0.1
desplazamiento izquierda distribuidor	Bool	%Q0.2
cinta piezas verdes	Bool	%Q0.3
pistón stop verdes	Bool	%Q0.4
cinta1 cajas verdes	Bool	%Q0.5
cinta2 cajas verdes	Bool	%Q0.6
pistón stop caja verde	Bool	%Q0.7
cinta piezas azules	Bool	%Q1.0
pistón stop azules	Bool	%Q1.1
cinta1 cajas azules	Bool	%Q1.2
cinta2 cajas azules	Bool	%Q1.3
pistón stop caja azul	Bool	%Q1.4
cinta piezas	Bool	%Q1.5
absorción robot verde	Bool	%Q1.6
absorción robot azul	Bool	%Q1.7
Número de cajas azules	DInt	%QD100
Número de cajas verdes	DInt	%QD104
motor X robot azul	Real	%QD116
motor Z robot azul	Real	%QD120
motor X robot verde	Real	%QD124
motor Z robot verde	Real	%QD128

Por otro lado todos aquellos elementos que se han utilizado en la programación del PLC y que necesitan parte de su memoria para almacenar valores entre ciclos de ejecución, hace que la lista de variables usadas sea mayor.

Para obtener una visión general del funcionamiento, a continuación se presenta la estructura de bloques que adopta el programa que realizar el control de la automatización de esta estación.



Figura 7.68. Distribución general de bloques. Estación distribuidora

A través de la estructura de bloques, se puede apreciar que se han utilizado diferentes tipos.

Se ha usado bloques de tipo organización [OB] cuyas características son las siguientes:

- **Main [OB1]:** es el bloque que contiene el programa principal y se encarga de hacer las llamadas al resto de bloques que componen el programa.
- **Puesta a cero [OB100]:** es un bloque de tipo STARTUP que lo que hace es inicializar el valor de variables internas del PLC y poner a cero todas las salidas cuando se carga por primera vez el programa en el PLC, antes de pasar a modo RUN y ejecutarse cíclicamente.

Por otro lado dado, esta estación está compuesta por elementos de iguales características y que realizan las mismas funciones pero en partes distintas de la estación, por lo que se ha recurrido al uso de bloques de tipo FB para programar su control.

Entre estos bloques se encuentran:

- **Robot automático [FB140]:** este bloque se encarga del control del movimiento de los robots que recogen las piezas que llegan por la cinta transportadora correspondiente y la depositan sobre las cajas de almacenaje.

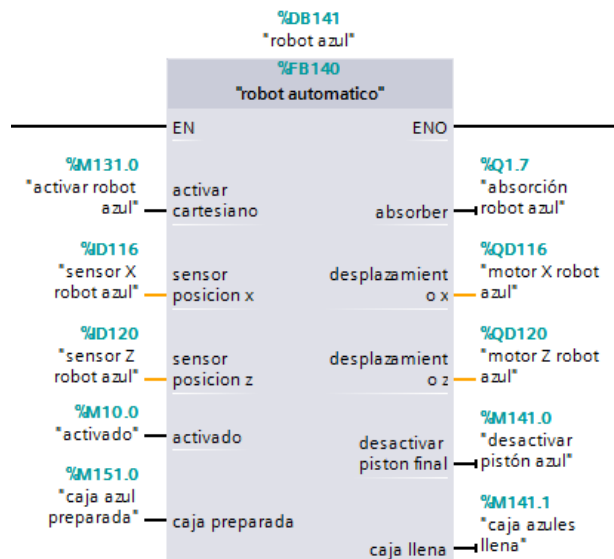


Figura 7.69. Bloque FB140. Robot automático

Este bloque cuenta con una serie de entradas y salidas que se encuentran conectadas directamente con las entradas y salidas del PLC, de las cuales leerán los valores de los sensores del robot y de las cintas transportadoras y accionarán los actuadores del mismo como los motores y la neumática de absorción.

Tal y como se ha planteado la estación, estos robots trabajan de forma coordinada con las cintas transportadoras de piezas y las cintas transportadoras por la que llegan las cajas de almacenaje, por lo que es necesario que este bloque de control también se comunice con otros bloques del programa. Por esa razón, otra parte de las entradas y salidas del bloque son señales de comunicación hacia otras partes del programa.

Esta estación cuenta con dos robots bien diferenciados, uno encargado de la gestión de las piezas de color azul, y otro de las de color verde, pero tal y como se ha comentado anteriormente al realizar ambos las mismas funciones utilizan el mismo FB.

Por otro lado, al ser robots diferentes cada uno debe almacenar su propia información por eso al hacer la multiinstancia a este bloque FB mediante TIA PORTAL debemos asociarles DB diferentes para que cada uno almacene su propia información. Por esa razón disponemos en el esquema general de bloques de un bloque de datos "robot verde [DB140]" y "robot azul [DB141]".

- **Cinta recoge pieza [FB130]:** gestiona el control de las cintas transportadoras sobre las cuales circula un único tipo de pieza, o bien piezas azules o piezas verdes. Son estas cintas transportadoras las que desplazan la pieza a la posición adecuada para que los robots accedan a ellas para su posterior almacenaje.

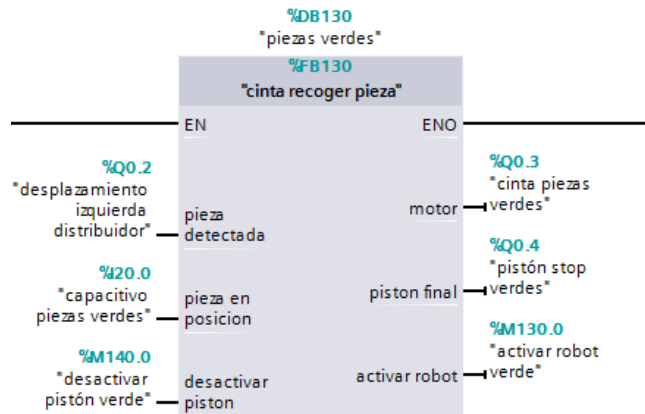


Figura 7.70. Bloque FB130. Cinta recoge piezas

Este bloque FB consta de una serie de entradas y salidas que se comunican directamente con los sensores y actuadores de la cinta transportadora, que hacen que la pieza se desplace hasta su posición de destino.

También posee otras señales de control que le permiten comunicarse tanto con el robot encargado de coger la pieza como con el distribuidor encargado de gestionar la dirección a seguir por las piezas en función del color que estas posean.

Al igual que sucedía en el caso anterior con los robots, también disponemos de dos cintas transportadoras de piezas con las mismas características y funciones pero totalmente independientes una de otra. Por lo tanto para realizar su control se utiliza una multiinstancia a este bloque FB para cada una de las cintas, pero a cada bloque se le asocia un bloque de datos diferente para que cada uno almacene su propia información.

Por esa razón aparecen otros dos bloques de datos en el esquema general de bloques, “piezas verdes [DB130]” y “piezas azules [DB131]”.

- **Cinta cajas [FB150]:** este bloque FB se encarga del control de las cintas transportadoras sobre las cuales se desplazan las cajas que serán rellenas por las piezas recogidas por los robots.

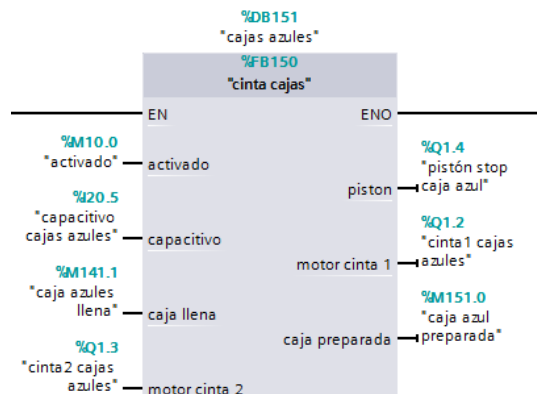


Figura 7.71. Bloque FB150. Cinta cajas

Al igual que sucedía en los casos anteriores, está comunicada directamente con los sensores y actuadores propios de la cinta y que desplazarán las cajas a lo largo de la estación.

Este bloque también utiliza señales de control, ya que las cintas transportadoras de cajas trabajan de forma coordinada con el bloque de control de los robots.

Como ya se ha comentado, al estar parte del proceso de la estación duplicado para la clasificación de ambos tipos de piezas, en este caso también se realiza una multiinstancia a este bloque FB, asociándole un DB de datos diferente para cada llamada, como son “cajas verdes [DB150]” y “cajas azules [DB151]”.

- **Distribuidor pieza [FB120]:** este bloque controla el funcionamiento del distribuidor de piezas de la estación, encargado de dirigir en una dirección u otra las piezas que le van llegando.

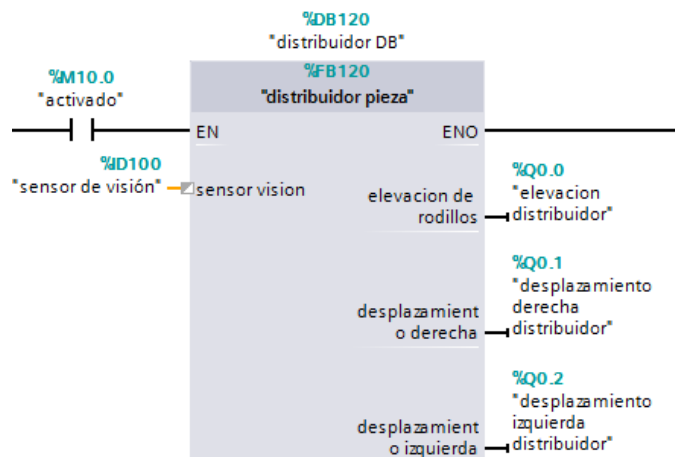


Figura 7.72. Bloque FB120. Distribuidor pieza

Este bloque cuenta en la entrada con la señal que le envía el sensor de visión y que permite identificar el tipo de pieza. En función del tipo de pieza detectada el bloque activa unas salidas u otras que están conectadas directamente con las salidas del PLC que gobiernan los actuadores del distribuidor.

Como todo bloque FB, este necesita tener asociado un bloque de datos DB que le permita almacenar el valor de las variables entre ciclo y ciclo de ejecución, por esta razón este bloque tiene asociado el bloque de datos “distribuidor DB [DB120]”.

Otro tipo de bloque utilizado en el programa son los bloques FC, dichos bloques se utilizan para la ordenación del código del programa y para la realización de operaciones que no requieran que su valor permanezca entre ciclos de ejecución.

Entre este tipo de bloques se encuentran:

- **Gestión marcha [FC10]:** en este bloque se encuentra el control del cuadro de mando de la estación que gobierna la marcha, el paro y la parada de emergencia de la estación.

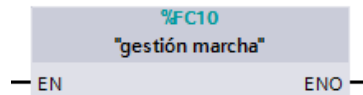


Figura 7.73. Bloque FC10. Gestión marcha

- **Cinta generadora de piezas [FC110]:** se encarga de poner en funcionamiento la cinta sobre la que aparecen de forma periódica las diferentes piezas que se irán procesando a lo largo de la estación.

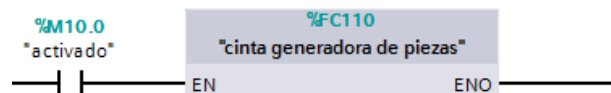


Figura 7.74. Bloque FC110. Cinta generadora de piezas

- **Contador de piezas [FC160]:** en el cuadro de mando de la estación se disponen de dos displays que van contabilizando el número de piezas de diferente tipo que se van procesando en la estación. Por ello este bloque se encarga de realizar la cuenta de los diferentes tipos de piezas.

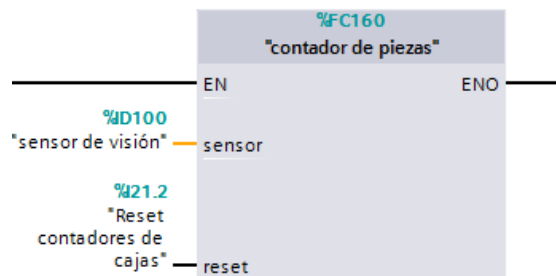


Figura 7.75. Bloque FC160. Contador de piezas

- **Piezas robot [FC141]:** este bloque se encarga de contabilizar el número de piezas que han sido depositadas por el robot sobre cada caja de almacenaje. Cuando se hayan depositado cuatro piezas sobre una misma caja, esta avanza a lo largo de la cinta y llega otra nueva vacía.

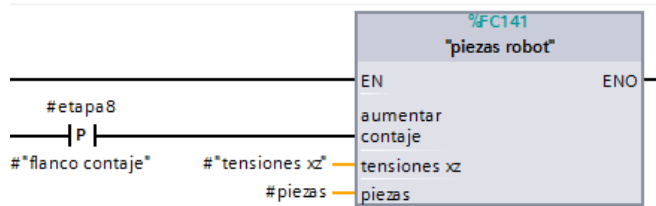


Figura 7.76. Bloque FC141. Piezas robot

- **Posición robot [FC142]:** la función de este bloque es reducir la presión de la posición proporcionada por los sensores del robot, para que cuando este se desplace hacia la posición objetivo lo realice de forma correcta, ya que el valor de la precisión de estas posiciones es menor que la proporcionada por los sensores.

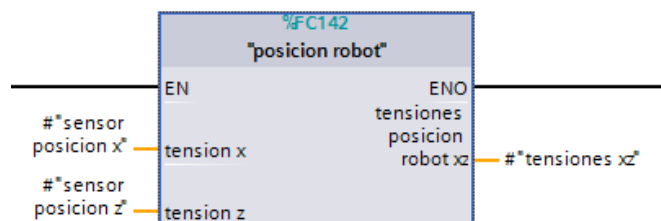


Figura 7.77. Bloque FC142. Posición robot

- **Gestión cinta piezas [FC130]:** esta función se encarga de organizar el código en el bloque principal [OB1], gestionando la llamada a los bloques FB de las cintas transportadoras de piezas.

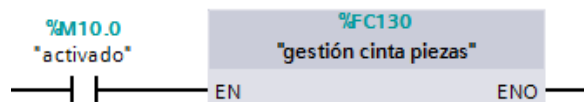


Figura 7.78. Bloque FC130. Gestión cinta piezas

- **Gestión robot [FC140]:** este bloque se encarga de gestionar las llamadas a los bloques FB que controlan el funcionamiento de los dos robots de la estación.

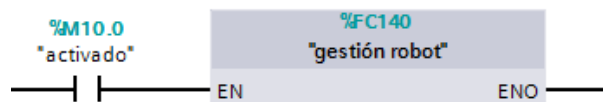


Figura 7.79. Bloque FC140. Gestión robot

- **Gestión cintas cajas [FC150]:** se encarga de gestionar las llamadas a los bloques FB que gobiernan el funcionamiento de las cintas transportadoras de las cajas de almacenaje.

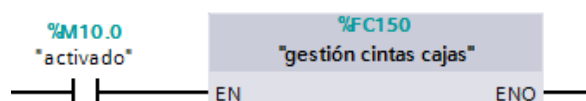


Figura 7.80. Bloque FC150. Gestión cinta cajas

7.3. DEPÓSITO DE LÍQUIDO

Esta estación es muy diferente a las realizadas anteriormente ya que en este caso lo que vamos a controlar es un proceso continuo que consiste en el llenado y vaciado de un depósito mediante una serie de válvulas.

Hasta el momento las estaciones que se habían automatizado eran procesos secuenciales, por lo que se recurría al uso del grafset. Sin embargo, en este caso lo que utilizaremos para controlar el proceso será unos controladores de tipo PID.

En un controlador PID se realizan tres tipos de acciones distintas:

- P: acción proporcional
- I: acción integral
- D: acción derivativa

Cada una de estas acciones tiene sus propias características, que se describen a continuación.

Acción proporcional

Genera una acción de control proporcional al error. Es el efecto más corriente y más intuitivo: si el error es positivo (velocidad excesivamente baja), es necesario aumentar la tensión sobre el motor para aumentar la velocidad; si es negativo (velocidad excesiva), será necesario reducir la tensión de entrada al motor. Existe una constante de proporcionalidad que llamaremos K_p que relaciona el error con la acción de control, de modo que para errores grandes las variaciones de intensidad serán también grandes. Si llamamos $e(t)$ a la señal de error y $u(t)$ a la acción sobre el sistema (tensión a aplicar sobre el motor), quedará:

$$u(t) = K_p \cdot e(t)$$

Acción integral

En algunos casos, la aplicación de un efecto proporcional no consigue que el sistema alcance el valor de referencia indicado, aún después de un prolongado periodo de tiempo. Se dice que el sistema presenta **error en régimen permanente**. En estos casos la mejor opción es recurrir a un efecto integral, que no actúa en función del error sino en función de la **integral del error**. De este modo, mientras la referencia no sea alcanzada la integral del error no parará de crecer, y con ella la acción de control sobre el sistema, hasta que la acción sea suficiente para llevar al sistema al punto deseado. La constante propia del efecto integral se denomina K_I , con lo que la relación entre el error y acción de control será:

$$u(t) = K_I \cdot \int_0^t e(\tau) \cdot d\tau$$

Acción diferencial

Este efecto busca conseguir un comportamiento más suave del sistema de control. Utilizando exclusivamente los efectos anteriores, la forma de alcanzar el valor de referencia puede ser excesivamente brusca, presentando picos de sobreoscilación excesivos.

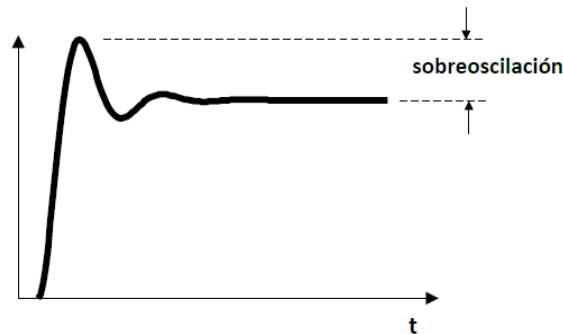


Figura 7.81. Efecto sobreoscilación de señal

Para evitar este problema, se introduce un efecto extra que es el efecto diferencial: la acción de control es proporcional a la derivada de la señal de error. Esto evita que el sistema pase de largo la referencia: si la derivada del error es negativa (nos acercaremos a la referencia), el efecto derivativo 'frena' ligeramente la acción de control. La constante propia de un efecto derivativo se denomina K_D ; y la expresión resultante queda:

$$u(t) = K_D \cdot \frac{d}{dt} e(t)$$

Un controlador PID presenta las tres acciones simultáneamente:

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(t) \cdot dt + K_D \cdot \frac{d}{dt} e(t)$$

El ajuste o la sintonización de un regulador consiste en elegir los valores de K_P, K_I, K_D y que consiguen un funcionamiento adecuado del sistema (en términos de tiempo de respuesta aceptable, pico de sobreoscilación aceptable, etc).

7.3.1. OBJETIVO DE LA ESTACIÓN

La estación consta de un depósito de líquido que posee dos válvulas diferentes, una para realizar el llenado del mismo, y otra para vaciarlo. Junto a este depósito se encuentra un cuadro de mando desde el cual el operario puede seleccionar una cantidad determinada de líquido mediante el uso de un potenciómetro y añadirla al depósito o extraerla.

A su vez se dispone de dos display encargados de informar al operario de la cantidad de litros seleccionados y la cantidad de litros que contiene el depósito en su interior.

De tal forma que el control que se va a realizar, en función de los elementos de los que se dispone en la estación adquiere la siguiente estructura:

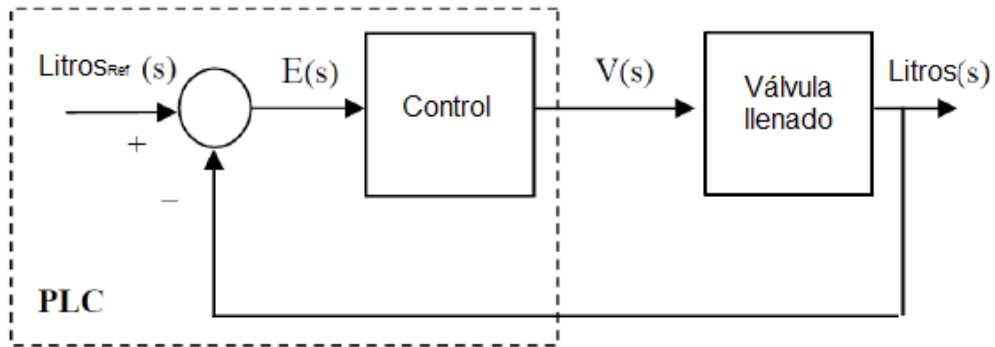


Figura 7.82. Esquema control PID válvula llenado depósito

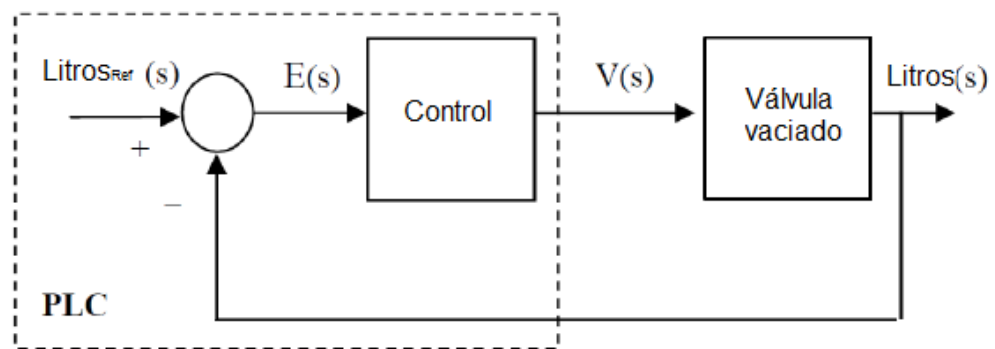







Figura 7.83. Esquema control PID válvula vaciado depósito

Cada válvula del depósito posee su propio controlador PID ya que son elementos independientes uno del otro.

En los esquemas, $E(s)$ representa la señal de error y es la diferencia entre los litros deseados en el depósito (referencia) y los litros realmente contenidos en su interior. La zona marcada con línea discontinua representa el autómata programable (S7-1200), en el que se encuentran los sistemas de control (PID) y desde el que se generan las señales de referencia.

7.3.2. DISEÑO EN FACTORY I/O

En el caso que nos ocupa, la célula de trabajo consta de muy pocos elementos pero es necesario conocer sus características y configuración para que el control se realice de forma óptima.

ELEMENTO	CONFIGURACIÓN	ETIQUETA	I/O	TIPO	DESCRIPCIÓN
	Analógica	Válvula de carga	input	real	[0,10]V Posición válvula llenado
		Válvula de descarga	input	real	[0,10]V Posición válvula vaciado
		Nivel	output	real	[0,10]V Valor de nivel
	Acción momentánea	Llenar	input	bool	Presionando
	Acción momentánea	Vaciar	input	bool	Presionando
	[0,10] V	Regulador	input	int	Valor actual
	Entero	Litros en depósito	output	int	Valor de display
		Litros solicitados	output	int	

Distribuyendo adecuadamente todos los elementos anteriores para cumplir el objetivo de comportamiento de la estación, el resultado obtenido es el siguiente:

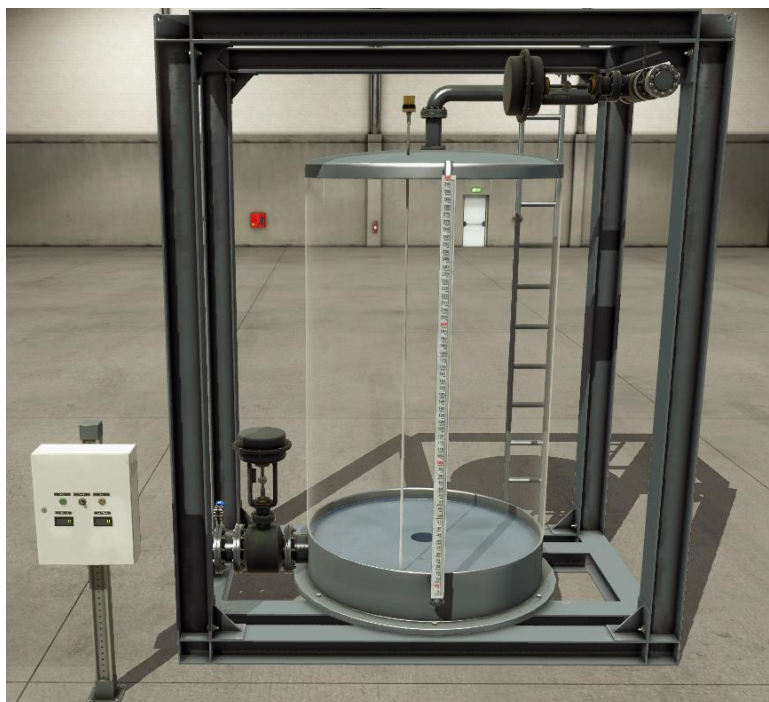


Figura 7.84. Vista frontal del depósito

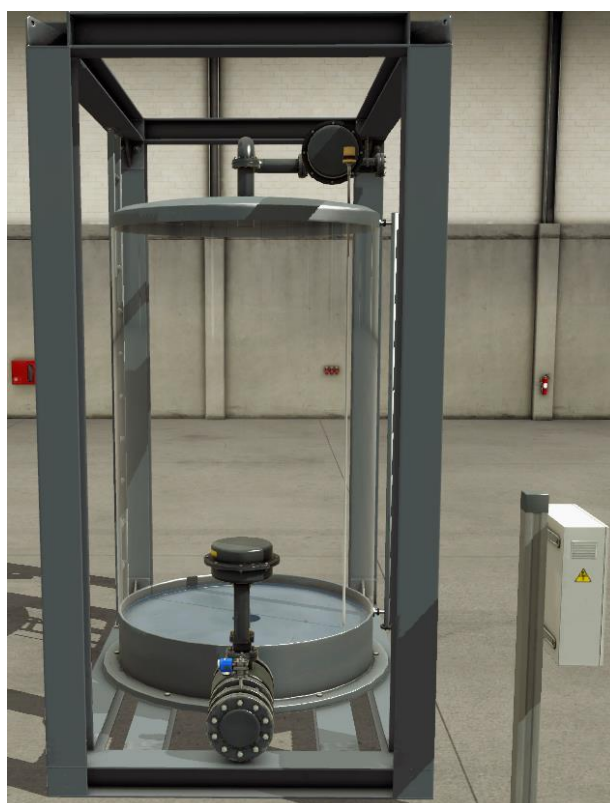


Figura 7.85. Vista lateral depósito



Figura 7.86. Cuadro de mando depósito

La distribución física que tendrían en la tarjeta del PLC las señales de todos los sensores y actuadores que intervienen en la estación sería la siguiente:



Figura 7.87. Distribución entradas/salidas. Estación depósito

7.3.3. CONFIGURACIÓN DE CONTROLADORES PID

Debido a que para el control de esta estación se han usado controladores PID, que son elementos relativamente complejos. Se detalla a continuación cuales van a ser sus funciones y como se configuran en TIA PORTAL.

El programa de control del PID realiza las siguientes operaciones:

- Leer de la entrada analógica el valor de la señal que se desea controlar (en este caso el nivel de líquido en el depósito).
- Comparar esta medida con la referencia (litros solicitados) y obtener el error como resta de los dos valores.
- Aplicar el algoritmo de control PID al error, calculando los efectos proporcional, diferencial e integral.
- Escribir en la salida analógica el resultado calculado.

Todas estas operaciones se deben repetir periódicamente, con una frecuencia ajustable.

Dado que se trata de un programa relativamente complejo, el programa TIA PORTAL ofrece un asistente que permite generar el programa especificando únicamente los ajustes deseados: fundamentalmente el periodo de muestreo y valores de las constantes proporcional, integral y diferencial.

Los pasos a seguir para añadir y configurar un nuevo bloque PID son los que se detallan a continuación.

En primer lugar es necesario añadir un bloque de programa de organización del tipo “interrupción cíclica”, con un determinado tiempo de ciclo, en nuestro caso el valor elegido ha sido 1ms (periodo de muestreo del controlador).

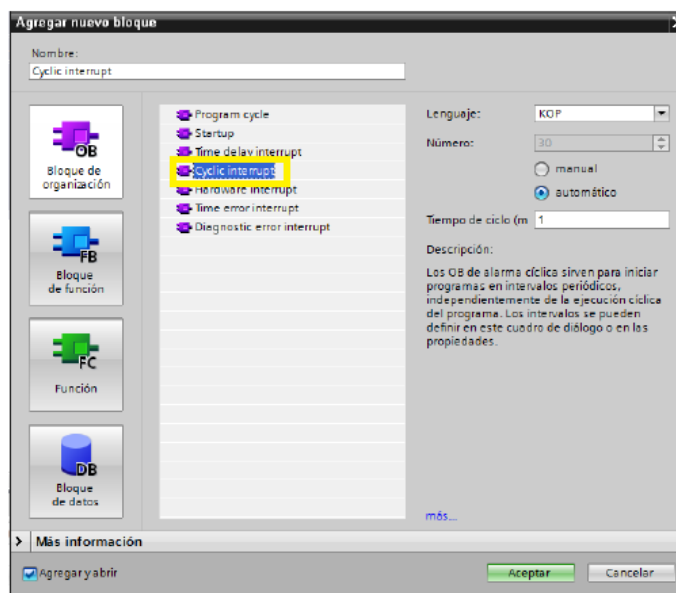


Figura 7.88. Selección de bloque interrupción cíclica en TIA PORTAL

El siguiente paso es insertar en el bloque de interrupción cíclica creado una instrucción de tipo “PID_Compact”, que se encuentra dentro del apartado “Tecnología”.

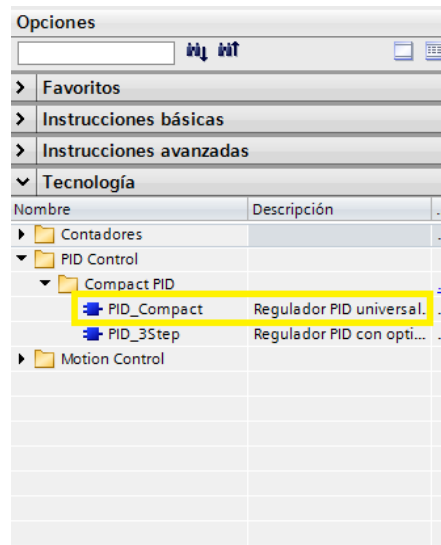


Figura 7.89. Selección bloque PID_COMPACT

Al colocar esta instrucción en el primer segmento de la interrupción cíclica, aparece una ventana para configurar el bloque de datos (DB) del PID. En esta ventana no modificamos ninguna opción ya que lo haremos internamente.

La estructura que se ha obtenido del PID es la siguiente:

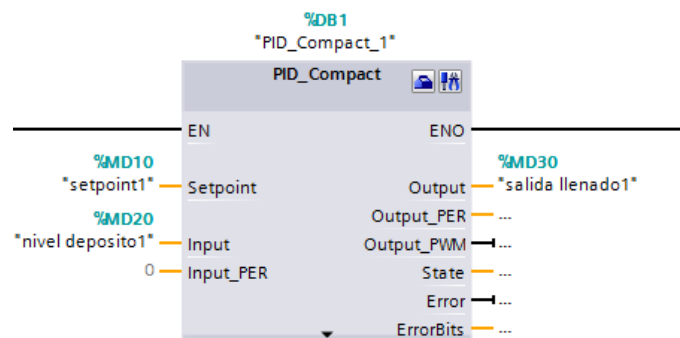


Figura 7.90. Asignación de variables en PID_COMPACT

Para entender de una forma más precisa este bloque, se describen algunos de sus parámetros utilizados.

- **Setpoint (parámetro de entrada):** es de tipo REAL y representa la referencia del controlador PID en modo automático. En nuestro caso ese valor representa la tensión del nivel del depósito asociada a la cantidad de litros deseados.
- **Input (parámetro de entrada):** es de tipo REAL y representa la tensión del nivel del depósito asociada a la cantidad de litros actuales. Este valor de la tensión será comparado (restado) con la referencia (setpoint).

- **Output (parámetro de salida):** es de tipo REAL y representa la salida analógica del controlador PID. En nuestro caso esta salida será la señal de control generada por el PID expresada como el voltaje necesario para abrir la válvula que permita alcanzar los litros deseados (setpoint).

A continuación lo que hay que hacer es configurar el PID, para ello hay que pulsar sobre el icono del maletín de la instrucción del PID.

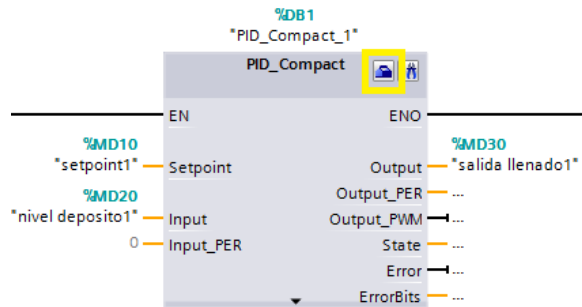


Figura 7.91. Configuración PID_COMPACT

Aparece la ventana de configuración del PID en la que se han introducir los siguientes valores:

- Tipo de regulación (en “Ajustes básicos”): Tensión
- Parámetros de entrada (en “Ajustes básicos”): Input y Output
- Límite superior del valor real (en “Ajustes del valor real → Límites del valor real”): 10.0

Cada uno de estos parámetros tiene su explicación:

- Indicamos tipo de regulación “Tensión” ya que tanto la entrada como la salida del controlador son valores de tensiones.
- El parámetro de entrada Input y Output se debe a que dichos valores provienen de una variable intermedia (MD20 y MD30) y no se coge el valor de las entradas y salidas de periferia.
- El límite superior del valor real tiene valor 10.0 ya que la tensión de la válvula de llenado y vaciado admiten como máximo 10V de entrada.

Una vez se haya finalizado la configuración de los parámetros del PID, se carga en el autómatas y lo ponemos en modo RUN.

El último paso es realizar la puesta en servicio del PID para eso hay que pulsar sobre el icono de las herramientas de la instrucción del PID.

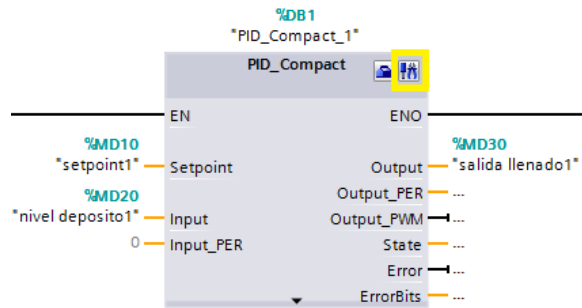


Figura 7.92. Puesta en servicio PID_COMPACT

Aparecerá la ventana de la puesta en servicio del PID. En esta ventana pulsaremos el botón “Start” que aparece en la parte superior izquierda para empezar la medición.

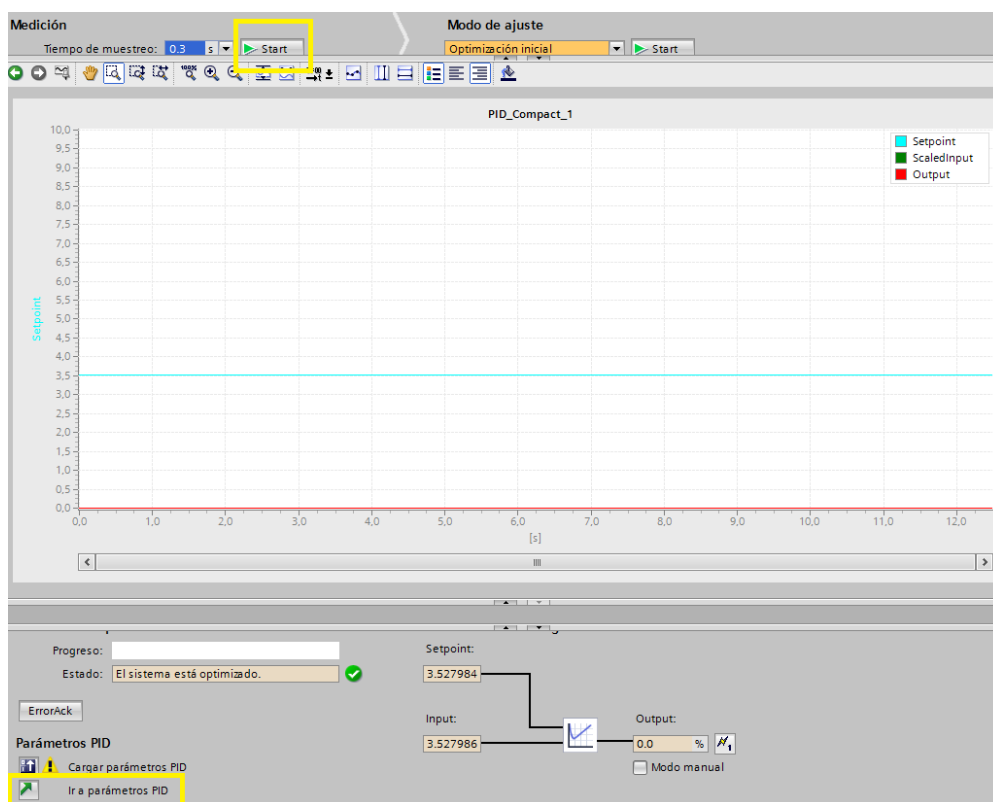


Figura 7.93. Configuración de los parámetros del PID_COMPACT

Mediante la opción de “Ir a parámetros de PID” podemos modificar los valores de la acción proporcional, diferencial y derivativa hasta que el sistema adquiera el comportamiento deseado.

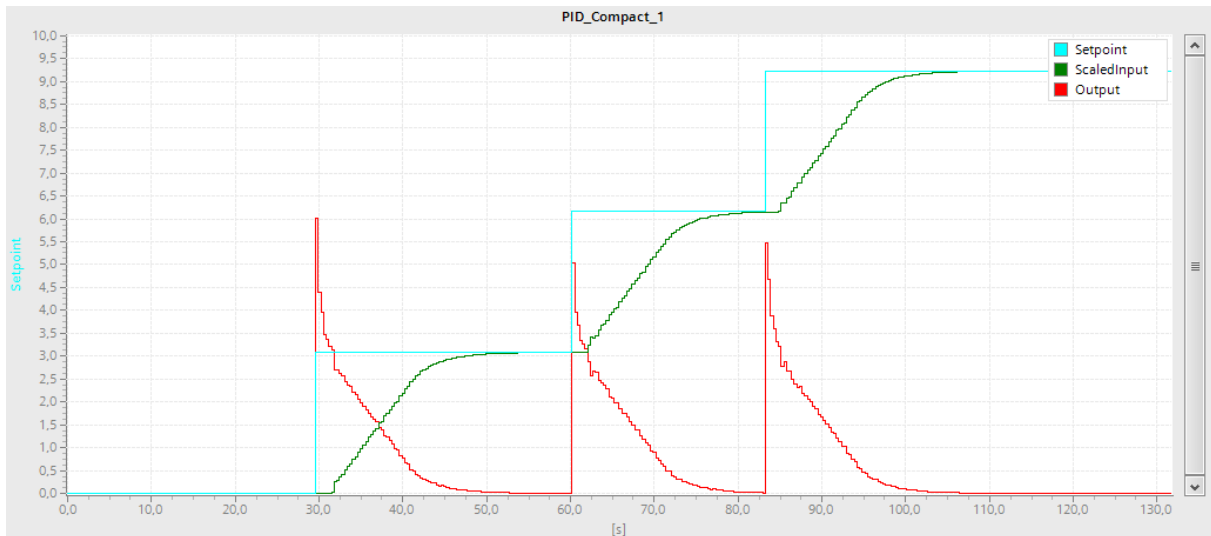


Figura 7.94. Visualización comportamiento PID_COMPACT

7.3.4. PROGRAMACIÓN DEL PLC

El desarrollo del programa se ha realizado mediante la utilización del lenguaje de contactos y el uso de las mismas direcciones de las variables de entrada y salida que las utilizadas en la configuración de la estación en FACTORY I/O.

A continuación se muestra en una tabla las entradas y salidas utilizadas en la elaboración del programa en TIA PORTAL.

Nombre	Tipo de dato	Dirección tarjeta del PLC
Nivel1	Real	%ID116
Llenar	Bool	%I21.0
Vaciar	Bool	%I21.1
Regulador	Real	%ID128
Válvula de carga1	Real	%QD100
Válvula de descarga1	Real	%QD104
Display litros depósito	DInt	%QD124
Display litros solicitados	DInt	%QD128

Para comprender más a fondo el programa de control encargado de gestionar el nivel del depósito, se describe a continuación su estructura general de bloques.

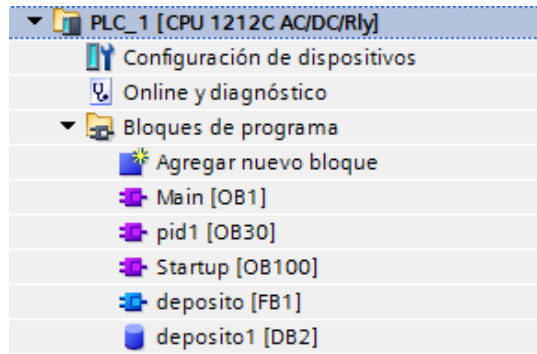


Figura 7.95. Distribución general de bloques. Depósito líquido

En primer lugar se recurre al uso de bloques de tipo organización [OB] entre los que se encuentran:

- **Main [OB1]:** es el bloque principal del programa que se encarga de hacer la llamada al resto de bloques.
- **Startup [OB100]:** es un bloque de tipo STARTUP que su función es inicializar todas las variables internas del PLC y poner a cero todas las salidas la primera vez que se carga el programa en el PLC antes de que pase a modo RUN.
- **PID1 [OB30]:** es un bloque de tipo interrupción cíclica que se ejecuta periódicamente en un tiempo previamente establecido.



Figura 7.96. Ciclo de ejecución OB30

Dentro de este OB será donde se coloque el bloque PID_Compact explicado anteriormente con detenimiento, siendo el periodo de ejecución del OB el periodo de muestreo del controlador del PID_Compact.

Otro tipo de bloques que se han utilizado en la realización del programa son los bloques de tipo FB, en nuestro caso el bloque deposito [FB1].

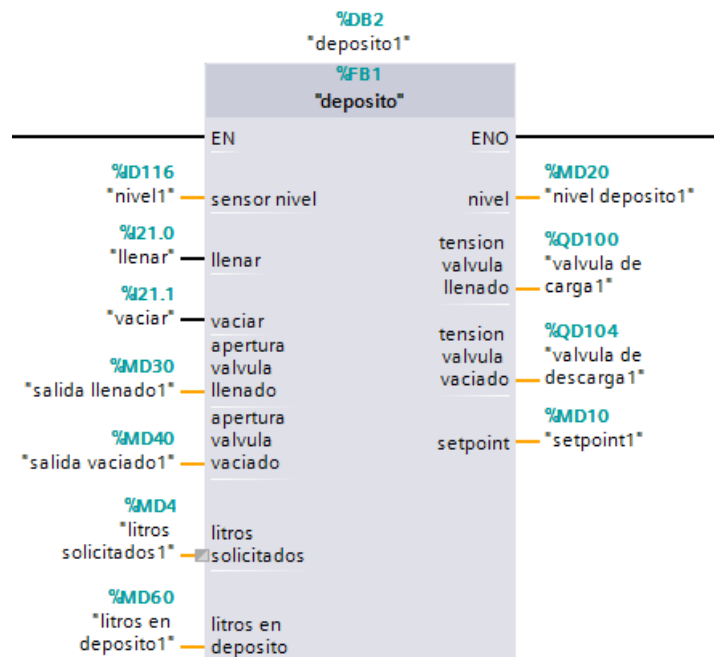


Figura 7.97. BloqueFB1. Depósito

Este bloque cuenta con una serie de entradas y salidas, parte de las cuales están conectadas directamente con los sensores y actuadores del depósito que nos van a permitir controlar el nivel de líquido en el depósito.

Por otro lado, otra parte de estas entradas provienen del cuadro de mando de la estación que nos indica la acción a realizar sobre el depósito, llenar de líquido o vaciar.

También es necesario que este bloque se comuniquen con los bloques PID_Compact que regulan el comportamiento de ambas válvulas del depósito, por lo que este bloque consta de entradas y salidas que permiten comunicarse con estos bloques transmitiendo información y trabajando de una forma coordinada y satisfactoria.

Como los bloques FB tienen una zona de memoria cuya información no se pierde entre ciclo y ciclo de ejecución del PLC, necesitan tener un bloque de tipo DB para almacenar todos esos datos, por esa razón este bloque tiene asociado el DB de datos "deposito1 [DB2]".

7.4. GESTIÓN DE STOCK DE UN ALMACÉN

7.4.1. Objetivo de la estación

En esta estación se pretende automatizar el stock de un almacén y controlar su gestión mediante una pantalla HMI manipulada por parte del operario.

El funcionamiento básico de la estación consiste en que por una cinta transportadora de rodillos llegarán diferentes tipos de piezas en palé. Al final de esta cinta transportadora se encuentra un sensor capacitivo que detecta la presencia del palé y detiene la cinta. En ese mismo lugar se encuentra una cámara de visión artificial que nos permite diferenciar un tipo de piezas u otras en función de su color y forma.

Una vez que la pieza ha sido reconocida y el palé se encuentra en la posición adecuada, el carro repartidor de la estación, desplazará el palé con la pieza al lugar asociado del almacén, en función del tipo de pieza.

Cuando el palé con la pieza es retirado de la cinta transportadora de entrada, otro nuevo palé con otra pieza se desplazará a dicha posición para ser también almacenado.

Cuando el carro repartidor ha depositado el palé con la pieza en la posición correspondiente del almacén, vuelve a la posición de inicio para volver a repetir el proceso con otro nuevo palé.

Por otro lado el operario va a disponer de una pantalla HMI para controlar la gestión del almacén.

En dicha pantalla, en la imagen principal, se disponen de dos botones uno de MARCHA y otro de PARO para controlar el funcionamiento de la estación. Además el operario dispone de la opción de seleccionar el tipo de gestión que quiere realizar con los productos que se van almacenando en el almacén. Si prefiere adoptar una gestión de stock tipo FIFO (First Input First Output) o a su vez prefiere una gestión tipo LIFO (Last Input First Output).

En esa misma pantalla el operario dispone a su disposición de un botón que le permite acceder a otra nueva pantalla donde puede seleccionar el tipo de pieza deseada y ver toda su información asociada.

Entre esa información se encuentra el número de piezas de ese tipo que hay en stock y el tiempo que llevan en el almacén. Es aquí donde el operario decide si desea extraer algunas de las piezas de ese tipo que se encuentran en el stock.


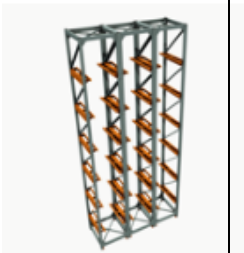
Cuando el operario decide extraer alguna pieza del stock, el carro repartidor de la estación se dirige a la fila del tipo de pieza seleccionada y en función del tipo de gestión (FIFO/LIFO), empezará a extraer piezas en una posición u otra de la fila.

Una vez la pieza ha sido extraída y depositada en el carro repartidor, este se dirige a la cinta transportadora de salida donde será depositado el palé y finalmente abandonará la estación.

Cuando el carro repartidor haya terminado de extraer todas las piezas solicitadas por el operario, volverá a reanudar su trabajo colocando los nuevos palés que le van llegando por la cinta transportadora de entrada.

7.4.2. DISEÑO EN FACTORY I/O

Esta célula de trabajo ha sido diseñada por un conjunto de elementos de los cuales sus características y configuración se muestran a continuación:

ELEMENTO	CONFIGURACIÓN	ETIQUETA	I/O	TIPO	DESCRIPCIÓN
 	numérica	Posición destino	output	int	Desplazamiento a la posición deseada
		Dientes izquierda	output	bool	Mover dientes izquierda
		Dientes derecha	output	bool	Mover dientes derecha
		Desplazamiento vertical	output	bool	Desplazamiento de la plataforma para carga o descarga
		Sensor desplazamiento x	input	bool	Desplazamiento a lo largo del eje x
		Sensor desplazamiento z	input	bool	Desplazamiento a lo largo del eje z
		Sensor límite izquierdo dientes	input	bool	Dientes en el límite izquierdo
		Sensor límite derecho dientes	input	bool	Dientes en el límite derecho
		Sensor central dientes	input	bool	Dientes en posición central

	digital	Rodillo de entrada	output	bool	Desplazamiento de rodillos en dirección de la flecha
		Rodillos salida	output	bool	
	digital	Rodillos de carga	output	bool	Desplazamiento de rodillos en dirección de la flecha
		Rodillos de descarga	output	bool	
	Todo numérico	Sensor de visión	input	int	Valor que representa al elemento detectado
	Digital	Sensor entrada	input	bool	Detectando
		Sensor de carga	input	bool	
		Sensor salida	input	bool	
		Sensor eliminación	input	bool	

Al igual a como se realizó en alguna de las estaciones anteriores se ha vuelto a incluir los elementos proporcionados por FACTORY I/O para la generación y destrucción de las piezas, como son el emisor y el eliminador, que nos permiten simular de una forma continuada el funcionamiento de la estación.

Distribuyendo de manera conveniente todos los elementos citados anteriormente con el fin de simular el objetivo de funcionamiento de la estación, el resultado obtenido es el que se muestra a continuación:



Figura 7.98. Vista frontal estación de gestión de stock



Figura 7.99. Vista detallada estación de gestión de stock

La distribución física que tendrían en la tarjeta del PLC las señales de todos los sensores y actuadores que intervienen en la estación sería la siguiente:



Figura 7.100. Distribución entradas/salidas. Estación gestión de stock

7.4.3. PROGRAMACIÓN SECUENCIAL GRAFCET

El diseño del control de la estación se ha realizado mediante el uso de grafquets parciales de nivel 1 que se centran en una descripción global del automatismo y que permiten rápidamente comprender su funcionamiento, sin entrar en detalles sobre qué elementos se van a utilizar para llevar a cabo dicha automatización.

A continuación se muestran los grafquets de los principales elementos de la estación.

- **Cinta de carga:** este grafcet se encarga de describir el funcionamiento de las cintas transportadoras de rodillos por las cuales llegarán los palés con las piezas que van a ser distribuidas en las posiciones correspondientes del almacén.

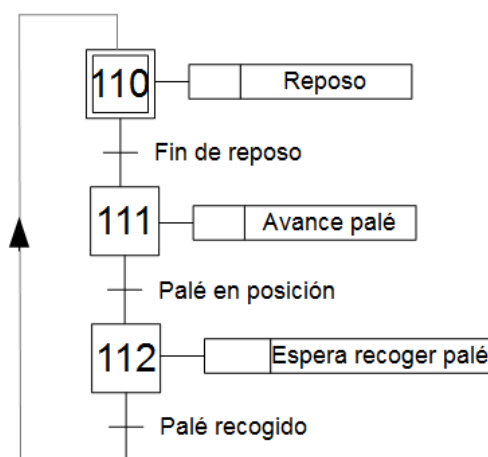


Figura 7.101. Grafcet cinta de carga

- **Cinta de descarga:** en este caso el grafcet se corresponde con la automatización de la cinta transportadora de rodillos sobre la cual se situarán aquellos palés que previamente el operario haya solicitado mediante el HMI su extracción.

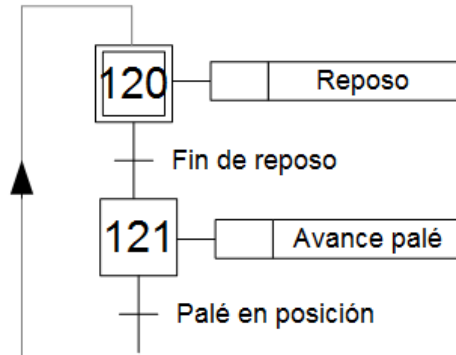


Figura 7.102. Grafcet cinta de descarga

- **Transportador:** mediante este diagrama se pretende describir el comportamiento que va a adoptar el transportador encargado de recoger los palés con piezas que van llegando a la estación por la cinta de carga y colocarlos en la parte correspondiente del almacén. Por otro lado, si el operario solicita mediante la pantalla del HMI la extracción de alguna pieza, será también este transportador el encargado de extraer dicho palé y depositarlo en la cinta transportadora de descarga.

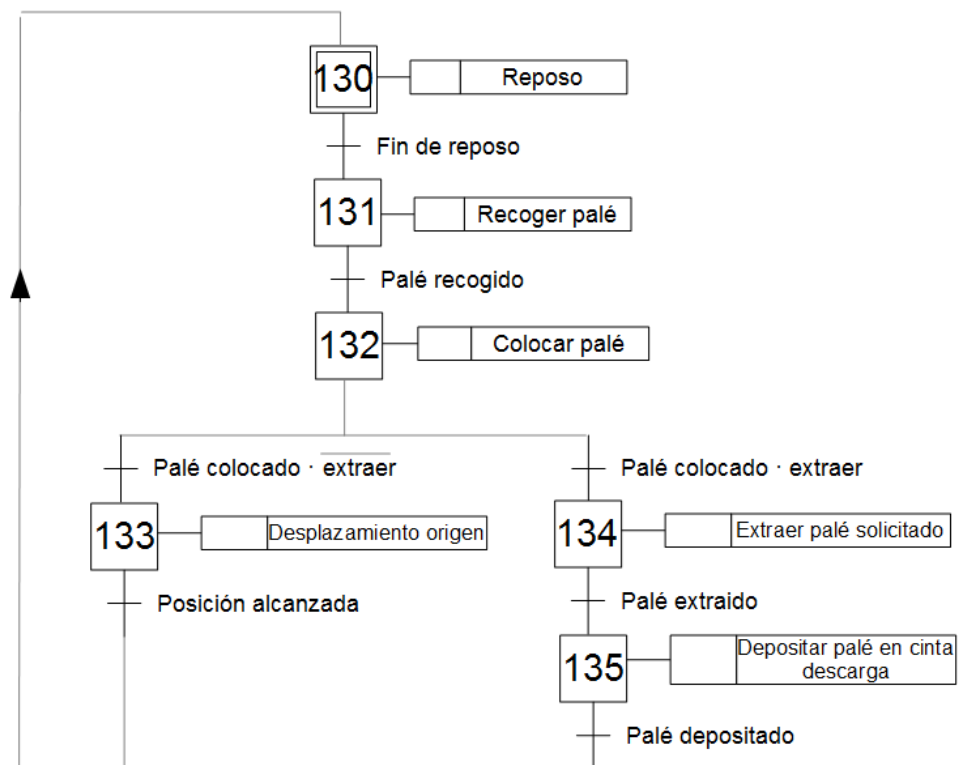


Figura 7.103. Grafcet transportador

7.4.4. PROGRAMACIÓN DEL PLC

La programación del PLC se ha realizado mediante el lenguaje de contactos o KOP, en el que se han utilizado variables de entradas y salidas con las mismas direcciones a las configuradas a la hora de diseñar la estación en FACTORY I/O para que la comunicación con TIA PORTAL se realice de forma satisfactoria.

A continuación se muestra la tabla con las entradas y salidas utilizadas en la elaboración del programa.

Nombre	Tipo de dato	Dirección tarjeta del PLC
sensor entrada	Bool	%I0.0
sensor carga	Bool	%I0.1
sensor limite derecho dientes	Bool	%I0.2
sensor limite izquierdo dientes	Bool	%I0.3
sensor central dientes	Bool	%I0.4
sensor desplazamiento x	Bool	%I0.5
sensor desplazamiento z	Bool	%I0.6
sensor inicio repartidor	Bool	%I0.7
sensor salida	Bool	%I1.0
sensor eliminación	Bool	%I1.1
sensor visión	Int	%IW30
rodillos de entrada	Bool	%Q0.0
rodillos de carga	Bool	%Q0.1
dientes derecha	Bool	%Q0.2
dientes izquierda	Bool	%Q0.3
desplazamiento vertical	Bool	%Q0.4
rodillos de descarga	Bool	%Q0.5
rodillos de salida	Bool	%Q0.6
posición destino	Int	%QW30

A continuación se expone los diferentes bloques en los que se estructura el programa de control que se cargará en el PLC para gobernar esta estación. Esta estructura nos aporta una visión más intuitiva sobre la elaboración del programa.

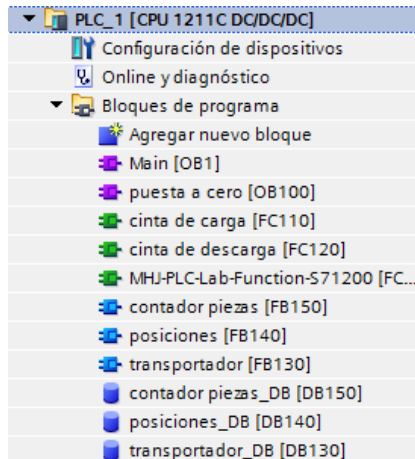


Figura 7.104. Distribución general de bloques. Estación de gestión de stock

En primer lugar se recurre al uso de bloques de tipo organización [OB] entre los que se encuentran:

- **Main [OB1]:** es el bloque principal del programa que se encarga de hacer la llamada al resto de bloques.
- **Puesta a cero [OB100]:** es un bloque de tipo STARTUP que su función es inicializar todas las variables internas del PLC y poner a cero todas las salidas la primera vez que se carga el programa en el PLC antes de que pase a modo RUN.

Por otro lado también se utilizan bloques de tipo FB, que poseen variables estáticas cuyos valores no desaparecen entre ciclo y ciclo de ejecución del PLC. Entre los bloques usados de este tipo se encuentran:

- **Posiciones [FB140]:** la función principal de este bloque es calcular las posiciones a las que debe desplazarse el transportador en función del tipo de pieza que es detectada en la cinta de entrada o de la pieza que haya seleccionado el operario para extraer.

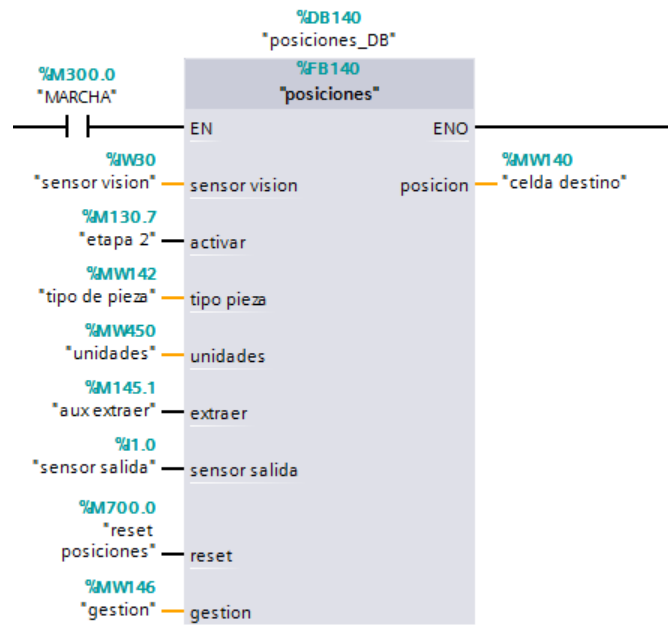


Figura 7.105. Bloque FB140. Posiciones

Este bloque posee una serie de variables de entrada y de salida que le permiten trabajar de forma coordinada con otros bloques, como es el caso del bloque que gobierna el control del transportador para enviarle las posiciones a las cuales se tiene que dirigir. También parte de estas variables se encuentran conectadas con las entradas del PLC ya que necesita la información proporcionada por algunos de los sensores.

Al tratarse de un bloque de tipo FB, es necesario asociarle un bloque de datos DB cuando se le hace la llamada desde el OB1. El bloque asociado en este caso es el “posiciones_DB [DB140]”.

- **Transportador [FB130]:** gestiona el control del carro transportador encargado de recoger los palés que llegan por la cinta de entrada y de colocarlos en las posiciones correspondientes del almacén. También este mismo bloque es el que se encarga de gobernar el funcionamiento del carro cuando el operario solicita la extracción de alguna pieza del almacén.



Figura 7.106. Bloque FB130. Transportador

Este bloque recibe las señales provenientes de todos los sensores del carro transportador para poder controlar su funcionamiento. Además también trabaja con un gran número de señales de control ya que es bloque principal del programa, que debe comunicarse tanto con la cinta de entrada como con la de salida, sin olvidarnos que también recibe parte de las señales de control introducidas por el operario mediante la pantalla HMI.

Al trabajar con tanta información, en este bloque FB se ha hecho uso de su tabla de variables estáticas por lo que es necesario asociarle un DB para que esos valores de desaparezcan. Por esa razón tiene asociado el DB de datos "transportador_DB [DB130]".

- **Contador de piezas [FB150]:** este bloque se encarga de contabilizar el número de piezas de cada tipo que hay en el almacén ya que dicha información será mostrada mediante la pantalla HMI al usuario.

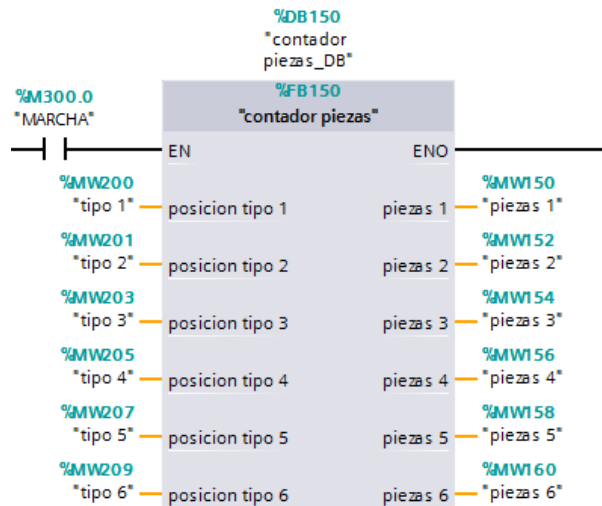


Figura 7.107. Bloque FB150. Contador piezas

El bloque de datos DB que tiene asociado es "contador piezas_DB [DB150]" donde se almacenan los valores que deben conservarse entre ciclos de ejecución.

Otro tipo de bloques que se han utilizado en la elaboración del programa son los bloques FC, entre los que se encuentran:

- **Cinta de carga [FC110]:** se encarga poner en funcionamiento la cinta de entrada de la estación, por la que llegarán los palés con las piezas para ser colocadas en el almacén.

Dependiendo de si ya existe una pieza en la cinta o no, se pondrá en marcha o se detendrá esperando a que el palé que esté presente sea recogido por el carro transportador.

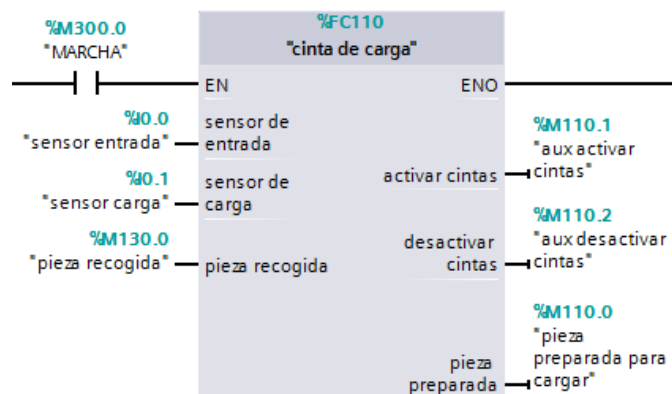


Figura 7.108. Bloque FC110. Cinta de carga

Este bloque tiene entre sus variables de entrada los sensores que se encuentran en la cinta y que transmiten la información sobre la posición en la que se encuentra el palé, además de señales de control provenientes del bloque del transportador.

En sus salidas se encuentran las señales que activarán o desactivarán las cintas en función del estado en el que se encuentren.

- **Cinta de descarga [FC120]:** se encarga de gestionar la cinta de descarga sobre la cual se depositarán aquellos palés que el usuario haya requerido mediante su extracción del almacén.

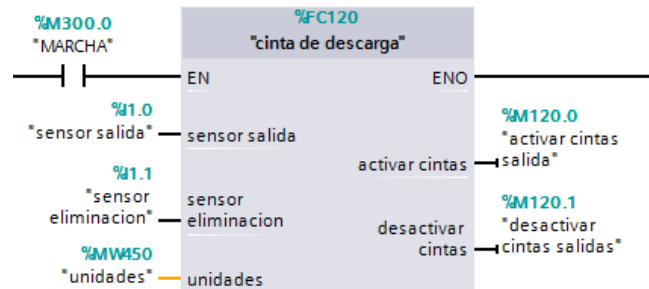


Figura 7.109. Bloque FC120. Cinta de descarga

Cuenta en sus variables de entrada con los sensores que detectan la presencia del palé en las distintas partes de la cinta y con una señal para controlar el número de unidades que quedan en el almacén de un determinado tipo de piezas. Como salidas posee señales para activar y desactivar los motores de las cintas de salida.

- **MHJ-PLC-Lab-Function-S71200 [FC9000]:** este bloque es el que proporciona FACTORY I/O en su plantilla de programación para que se pueda realizar la comunicación de forma satisfactoria con el PLCSIM V13/14.

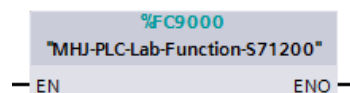


Figura 7.110. Bloque FC9000

7.4.5. CONFIGURACIÓN DEL HMI

Como en esta estación se ha incluido la simulación mediante pantalla HMI por parte del operario, antes de entrar de lleno en cómo se ha realizado la programación se expone de forma detallada como se configura la pantalla para que pueda comunicarse con el PLC.

En primer lugar es necesario incorporar la pantalla a nuestro proyecto, mediante la agregación de nuevos dispositivos y seleccionando el tipo de pantalla que deseamos.

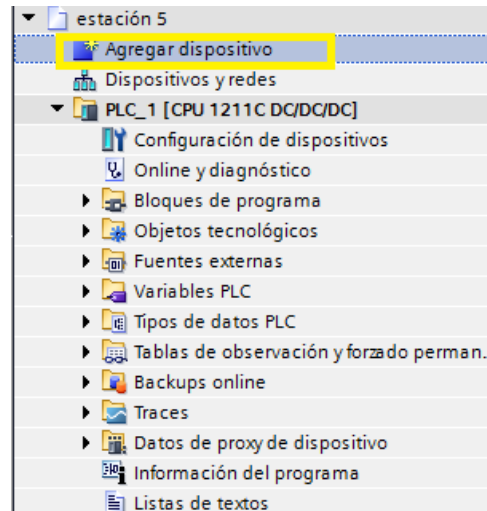


Figura 7.111. Estructura en árbol de proyecto

En el caso de esta estación se eligió una pantalla de tipo BASIC de modelo KTP700.

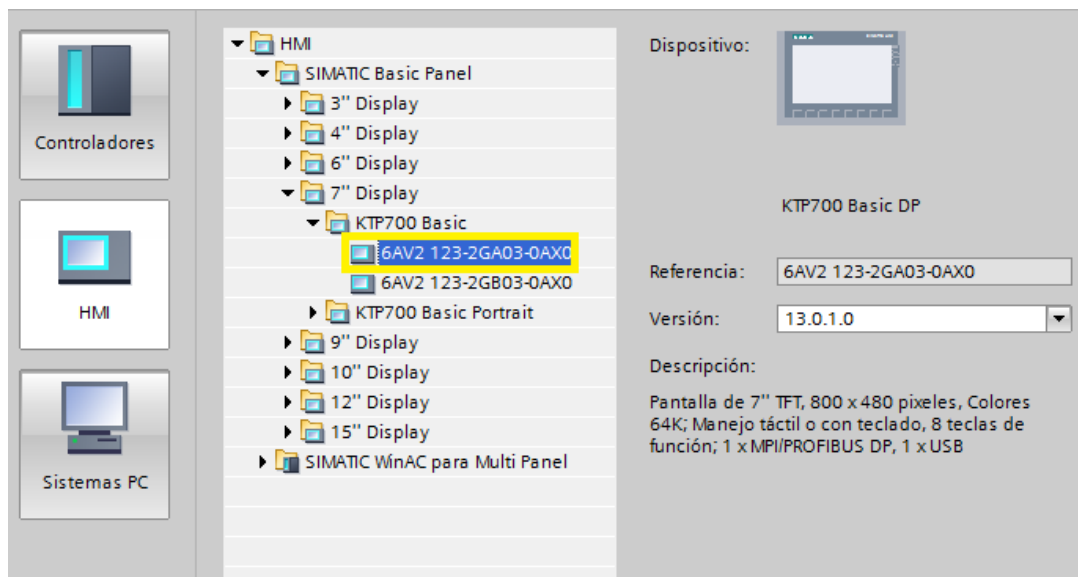


Figura 7.112. Selección de tipo de pantalla HMI

Para que pueda existir una comunicación entre esta pantalla y el PLC es necesario que ambos elementos se reconozcan uno al otro, por lo que es necesario meterlos en red. Para ello a la nueva pantalla incorporada es necesario asignarle una dirección que se encuentre en la misma red que el PLC. En este caso el procedimiento es muy similar al seguido en la asignación de dirección del PLC.

En la opción de configuración de dispositivo de la pantalla HMI, seleccionamos las propiedades de la pantalla y en direcciones Ethernet asignamos una dirección que se encuentre en la misma red que la dirección que posee el PLC.

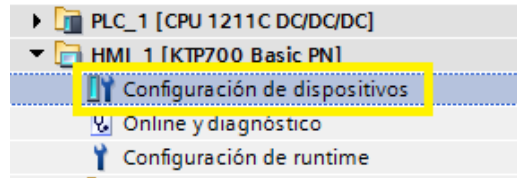


Figura 7.113. Menú de configuración de dispositivo

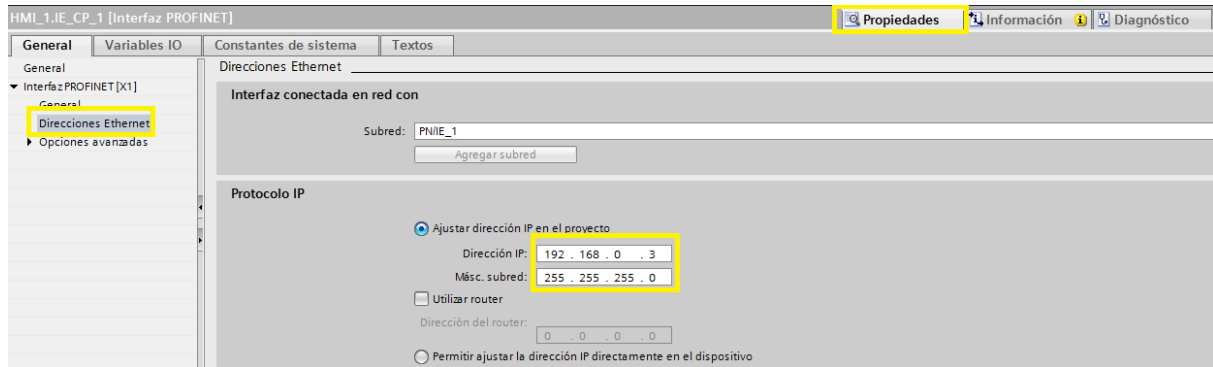


Figura 7.114. Menú de asignación de dirección de red a la pantalla HMI

Una vez que ambos dispositivos están en red es necesario que se reconozcan tanto el uno como el otro. Para ello TIA PORTAL nos proporciona unas herramientas muy sencillas mediante sus opciones de configuración de dispositivos.

Ya sea tanto dentro del PLC como dentro de HMI, accedemos a la opción configuración de dispositivos en su estructura de árbol.

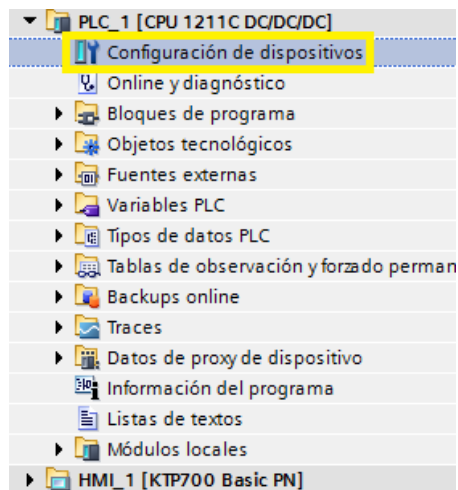


Figura 7.115. Menú de acceso a vista de redes

En la ventana que se nos abre, nos situamos en la pestaña de vista de dispositivos y pinchando en el puerto Ethernet de uno de ellos y arrastrando hasta soltar en el puerto del otro se crea la red de comunicación.

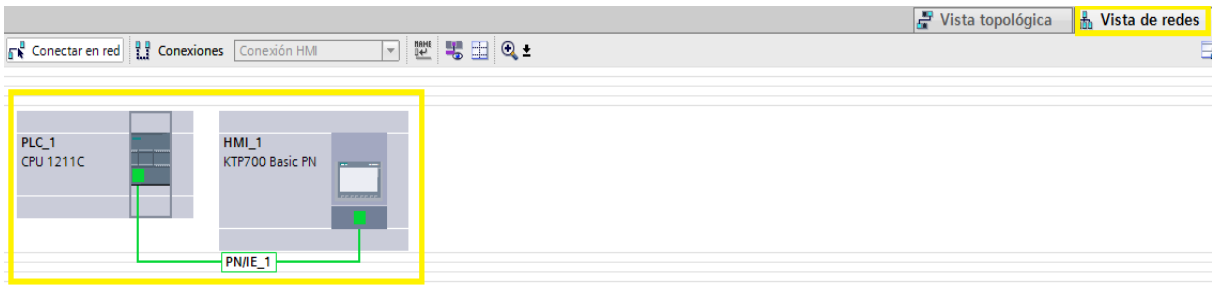


Figura 7.116. Vista de dispositivos y redes

7.4.6. PROGRAMACIÓN DEL HMI

De manera general, la programación del HMI se realiza mediante la utilización de imágenes a las cuales se las van agregando diferentes tipos de objetos con unas propiedades específicas que se adaptan a las necesidades del programador.

Por otro lado, estos objetos e imágenes pueden realizar ciertas acciones sobre las variables del PLC o sobre las propias variables del HMI a las cuales el PLC no tiene acceso.

Para interpretar estos conceptos en el caso de la pantalla HMI configurada para nuestra estación, se exponen a continuación como han sido utilizados dichos elementos.

En la estructura de árbol de la pantalla, observamos que existe un apartado denominado “variables HMI”, es ahí donde se declaran todas las variables que van a ser usadas o manipuladas por la pantalla.

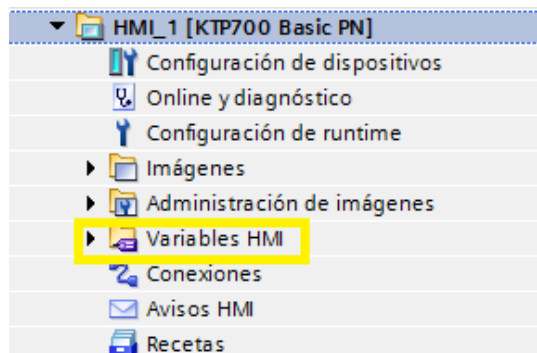


Figura 7.117. Menú de variables HMI

En nuestro caso en particular, esta tabla de variables adopta la siguiente configuración:

Tabla de variables estándar							
Nombre	Tipo de datos	Conexión	Nombre PLC	Variable PLC	Dirección	Modo de acceso	Ciclo de adquisi...
extraer	Bool	HMI_Conexión_1	PLC_1	extraer		<Acceso simbólico>	1 s
gestion	Int	HMI_Conexión_1	PLC_1	gestion		<Acceso simbólico>	1 s
MARCHA	Bool	HMI_Conexión_1	PLC_1	MARCHA		<Acceso simbólico>	1 s
Número_imagen_variable	UInt	<Variable intern...		<No definido>			1 s
piezas 1	Int	HMI_Conexión_1	PLC_1	*piezas 1*		<Acceso simbólico>	1 s
piezas 2	Int	HMI_Conexión_1	PLC_1	*piezas 2*		<Acceso simbólico>	1 s
piezas 3	Int	HMI_Conexión_1	PLC_1	*piezas 3*		<Acceso simbólico>	1 s
piezas 4	Int	HMI_Conexión_1	PLC_1	*piezas 4*		<Acceso simbólico>	1 s
piezas 5	Int	HMI_Conexión_1	PLC_1	*piezas 5*		<Acceso simbólico>	1 s
piezas 6	Int	HMI_Conexión_1	PLC_1	*piezas 6*		<Acceso simbólico>	1 s
tipo de pieza	Int	HMI_Conexión_1	PLC_1	*tipo de pieza*		<Acceso simbólico>	1 s
unidades	Int	HMI_Conexión_1	PLC_1	unidades		<Acceso simbólico>	1 s
unidades base azul	Int	<Variable intern...		<No definido>			1 s
unidades base verde	Int	<Variable intern...		<No definido>			1 s
unidades prima azul	Int	<Variable intern...		<No definido>			1 s
unidades prima verde	Int	<Variable intern...		<No definido>			1 s
unidades tapa azul	Int	<Variable intern...		<No definido>			1 s
unidades tapa verde	Int	<Variable intern...		<No definido>			1 s
<Agregar>							

Figura 7.118. Lista de variable de HMI

Como se puede apreciar por la tabla anterior, en la columna conexión disponemos de dos opciones:

- **HMI_Conexión_1:** significa que dicha variable es compartida por el PLC.
- **Variable interna:** indica que dicha variable solo es accesible para el HMI.

Para conocer cuál va a ser el comportamiento de la pantalla, se expone a continuación la estructura de las diferentes imágenes que la componen.

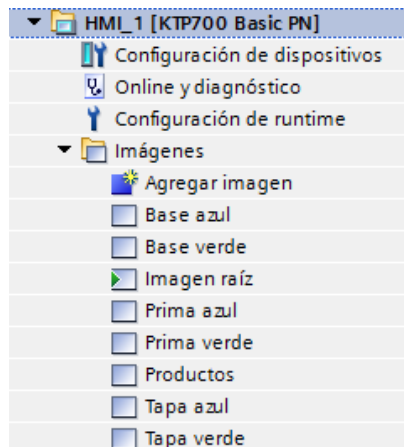


Figura 7.119. Estructura de imágenes que componen el HMI

En primer lugar cabe destacar que todas las pantallas poseen lo que se denomina la “Imagen raíz”, que es aquella imagen con la que se iniciará el HMI cuando se le conecte por primera vez. En nuestro caso la imagen raíz nos muestra los botones de mando para gobernar la MARCHA y PARO de la estación, además de información complementaria.



Figura 7.120. Imagen raíz HMI

En la parte izquierda de la imagen, se muestra información complementaria para el operario sobre la capacidad total que posee el almacén, así como del estado en el que se encuentra, acompañado de un indicador luminoso.

Por otro lado posee la opción de que el operario decida el tipo de gestión que quiere realizar con los productos que se encuentran en stock, proponiéndole realizar un tipo de gestión FIFO o LIFO.

Por último, el botón PRODUCTOS nos lleva a una nueva imagen en la que el operario puede seleccionar el tipo de producto que desea consultar. Esta imagen se corresponde con la denominada “Productos” dentro de la estructura de imágenes del proyecto mostrada con anterioridad y su apariencia es la que se muestra a continuación.

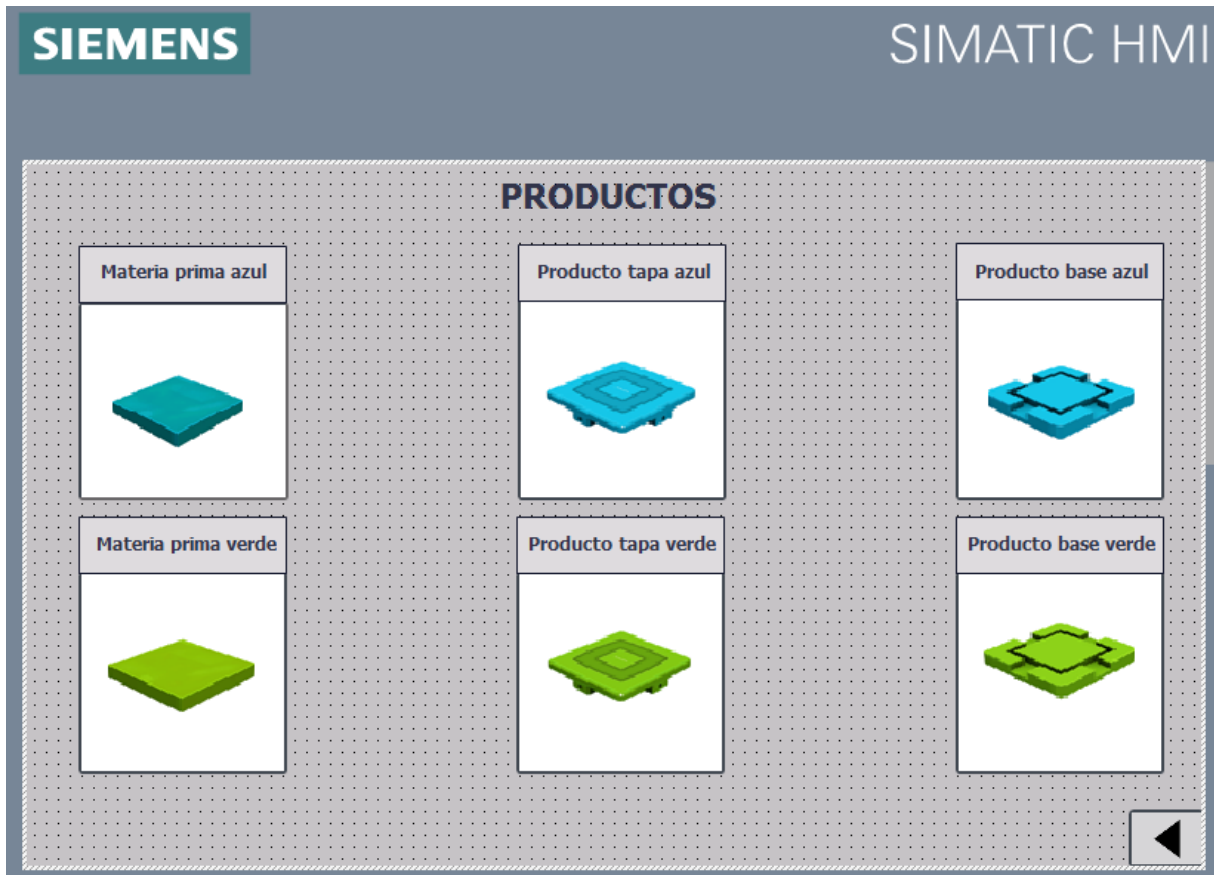


Figura 7.121. Imagen productos HMI

Desde aquí el operario puede seleccionar el producto que desee consultar y le llevará a otra imagen donde puede consultar información asociada al mismo.

Por otro lado, si el operario desea retroceder a la imagen anterior, en la esquina inferior derecha posee un botón de retorno que le permitirá realizar dicha acción.

Una vez que el operario haya escogido el producto deseado, se le abrirá una nueva imagen cuya estructura es idéntica para todas las piezas pero que los valores de las variables representadas en las mismas serán diferentes para cada caso.

Estas imágenes se corresponden con las denominadas en la estructura como “Base azul”, “Base verde”, “Prima azul”, “Prima verde”, “Tapa azul” y “Tapa verde”.

La apariencia que adoptan estas imágenes es la siguiente:

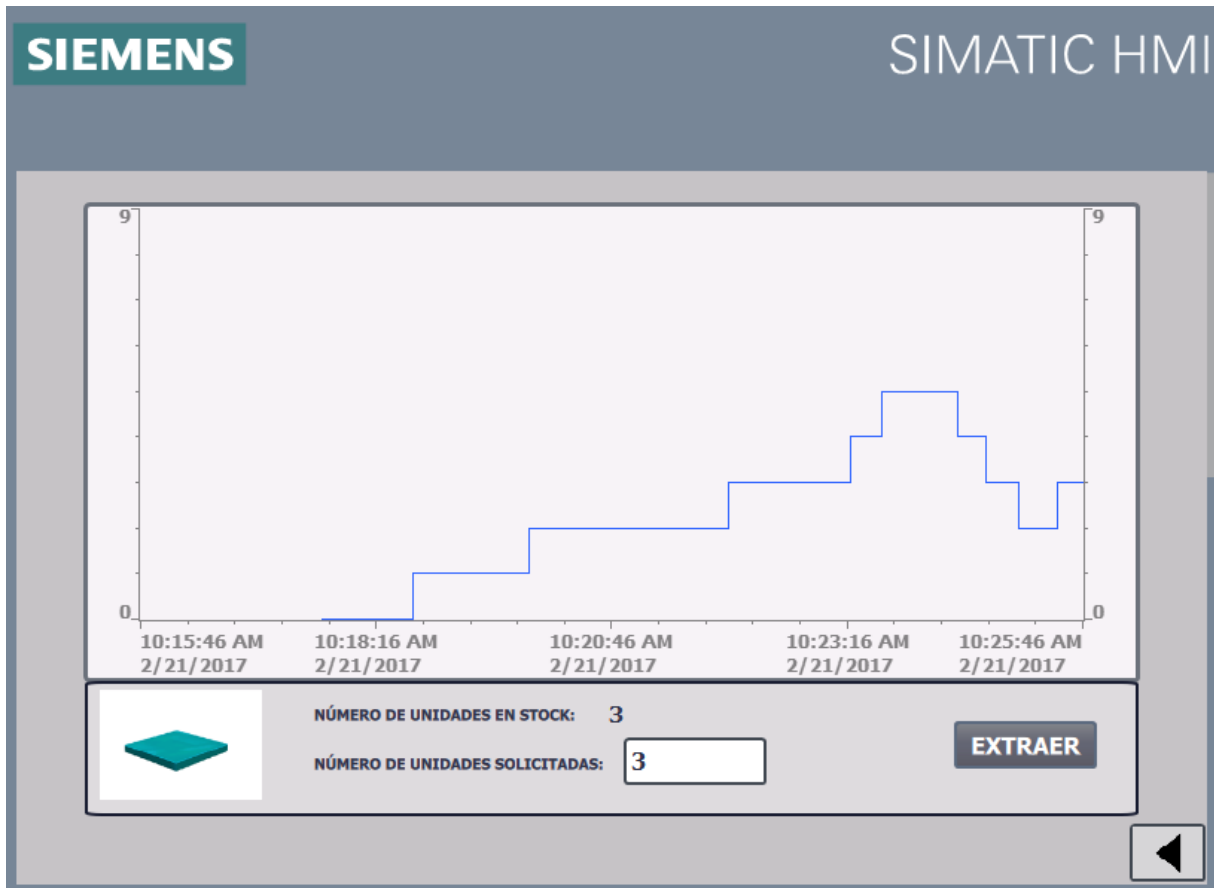


Figura 7.122. Imagen pieza prima azul del HMI

Tal y como se puede apreciar, la imagen consta principalmente de una gráfica en la que se va mostrando el número de piezas de ese tipo que existen en el almacén, además de aportar información sobre el instante en el que se incluye una nueva pieza en el almacén o se extrae.

Por otro lado en la parte inferior de la pantalla, se muestra el número de piezas de ese tipo que hay en ese instante en el almacén. Además desde aquí el operario puede seleccionar el número de unidades que desea extraer del almacén y mediante el botón “EXTRAER” enviar la orden a la estación.

Al igual a como sucedía en la imagen anterior, en la esquina inferior derecha se ha incluido un botón de retorno para regresar a la imagen “Productos”.

7.4.7. SIMULACION MEDIANTE PLCSIM

Como esta estación es la única de todas las anteriores en la que se ha usado para la simulación PLCSIM, ya que en las anteriores se ha utilizado el PLC físico, se explica cómo se accede a la simulación desde TIA PORTAL y como hay que ejecutar el RUN TIME que nos permita simular el comportamiento de la pantalla HMI.

Parte de esta explicación fue descrita de manera superficial al inicio del presente trabajo.

Para simular un proyecto mediante PLCSIM, lo primero es seleccionar el PLC en el cual se encuentra cargado el programa que queremos simular.

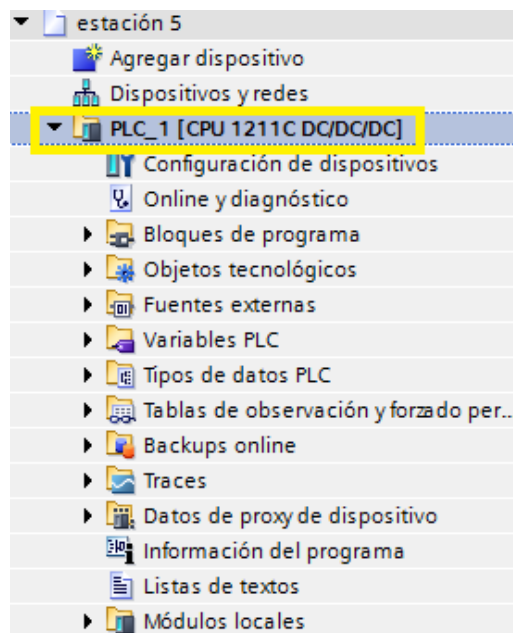


Figura 7.123. Selección del PLC del proyecto

A continuación en la barra de menú, seleccionamos Online y en la lista que se despliega escogemos Simulación e Iniciar.

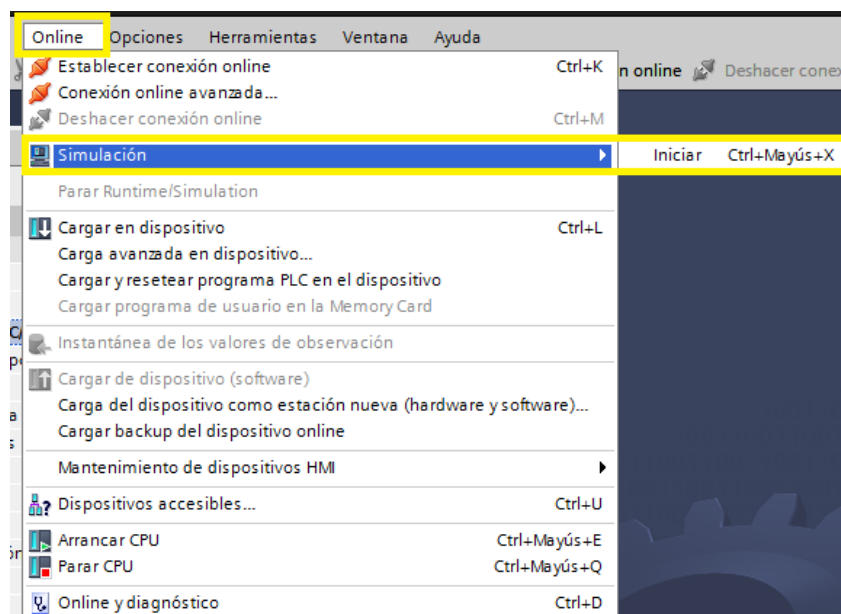


Figura 7.124. Menú de inicio de simulación

Si todo ha ido correctamente, se nos mostrará la pantalla del PLCSIM simulando el funcionamiento del PLC en modo RUN.

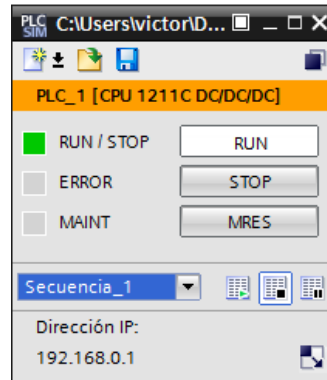


Figura 7.125. Ejecución del simulador

Una vez que ya tenemos ejecutando el PLCSIM es hora de ejecutar el RUNTIME ya que el HMI tiene un PLC con el que conectarse.

Si se hubiera iniciado antes el RUNTIME que el PLCSIM, nos hubiera salido un mensaje de error avisándonos de que es imposible establecer conexión ya que no se dispone de ningún PLC.

Para iniciar el RUNTIME es necesario seleccionar previamente la pantalla HMI que vamos a simular y a continuación escoger en la barra de herramientas la opción de iniciar simulación.

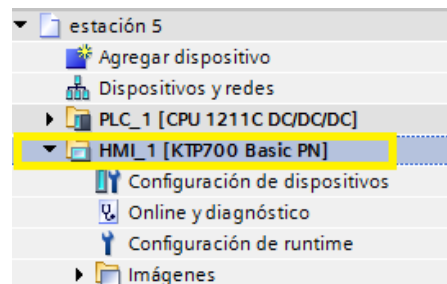


Figura 7.126. Selección de la pantalla HMI del proyecto

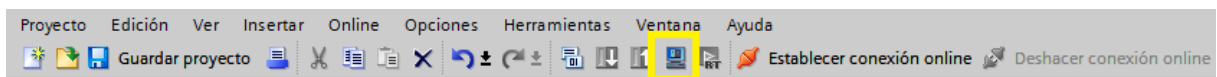


Figura 7.127. Botón de la barra de herramientas de inicio de RUNTIME

8. ESTUDIO ECONOMICO

A continuación se va a realizar el cálculo asociado al coste económico que ha supuesto la realización de este trabajo fin de grado, dividiendo estos costes entre directos e indirectos.

Costes directos

Los costes directos son los que han tenido que ver directamente con el producto final obtenido, la programación y simulación de las diferentes estaciones.

Estos costes directos los dividimos a su vez en costes de personal y de material.

- Costes de personal

En estos, se supone que este trabajo ha sido realizado de una forma profesional, por un ingeniero, es decir, se estudiará el tiempo que ha costado realizar cada actividad, y sabiendo cual es el coste por hora que tiene asignado el ingeniero, obtener el coste de personal.

Por lo tanto las actividades realizadas y el tiempo empleado para las mismas se muestran en la siguiente tabla.

ACTIVIDAD	HORAS
Adquisición de conocimiento de programación de TIA PORTAL	60
Manejo de software FACTORY I/O	10
Configuración de comunicación entre TIA PORTAL y FACTORY I/O	2
Estación paletizadora	
Diseño	10
Programación	30
Simulación	5
Documentación	2
Estación distribuidora	
Diseño	15
Programación	40
Simulación	8
Documentación	5
Depósito de líquido	
Diseño	15
Programación	35
Simulación	15
Documentación	4
Gestión de stock de almacén	
Diseño	20
Programación	60
Simulación	15
Documentación	5
Redacción de la memoria	30
TOTAL	386

Suponiendo un salario de ingeniero recién titulado que cobre 20€/h, obtenemos el siguiente coste de personal:

$$20€/h \cdot 386 h = 7.720 €$$

- Costes materiales

En esta parte se tienen en cuenta el precio de todos aquellos elementos que han sido necesarios para la realización del presente trabajo.

Siendo dichos elementos y su coste asociado los que se muestran en la siguiente tabla:

CONCEPTO	IMPORTE (€)
Ordenador portátil i7	679.50
Pantalla externa	350.95
PLC 1212 AC/DC/RLY de Siemens	277.09
Software de programación TIA PORTAL	750.50
Software de simulación FACTORY I/O	477.95
Panel HMI Basic Siemens	353.43
Cable de comunicación ethernet	2.50
Cable de conexión HMI	5.30
TOTAL	2.896,72

Costes indirectos

En este apartado incluimos los costes que no son físicamente considerados, ya que damos por hecho que están ahí sin problema, pero que por supuesto generan unos gastos a considerar.

Estos gastos se reparten entre el consumo eléctrico por los distintos dispositivos y la conexión a internet utilizada, considerando un periodo de cuatro meses para la realización de este trabajo.

CONCEPTO	IMPORTE (€)
Consumo eléctrico	30.29
Conexión a internet	20.75
TOTAL	51.04

Finalmente considerando tanto los costes directos como los costes indirectos asociados, el coste final de la realización del proyecto es de:

CONCEPTO	IMPORTE (€)
Costes directos	10.616,72
Costes indirectos	51,04
TOTAL	10.667,76

9. CONCLUSIONES

Tras la finalización de la programación y la simulación de las diferentes estaciones del presente trabajo y observando los resultados obtenidos podemos establecer que el uso del software FACTORY I/O nos facilita claramente la adquisición de nuevos conceptos en cuanto a materia de programación de PLC se refiere.

En las estaciones anteriores, a la hora de realizar su programación no se abordaba directamente la programación completa de la misma, sino que se iban dividiendo en secciones y programando poco a poco. Cada vez que se tenía una sección programada, se simulaba para ver su comportamiento, lo que en muchos casos permite darse cuenta de la existencia de errores de programación que de otra forma hubieran pasados desapercibidos y sería a la hora de poner en marcha la estación completa cuando hubieran aparecido y siendo mucho más complicado dar con ellos y tratar de corregirlos.

Por otro lado el uso del PLC físico en todas aquellas estaciones cuya simulación se ha realizado mediante el uso del mismo, permite reforzar los conocimientos en el campo de redes de comunicación ya que es necesaria la creación de una red local para la correcta simulación de la estación.

A sí mismo, en el caso concreto del control del depósito del tanque, el uso del elemento de simulación proporcionado por FACTORY I/O permite observar el efecto que tiene sobre dicho elemento físico las diferentes estrategias de control, de otro modo hubiera sido necesario la existencia de un sistema físico con su respectivo coste y mantenimiento para poder observar los resultados.

También se ha observado que se obtienen ventajas en cuanto al uso y la programación de pantallas HMI ya que por un lado nos permite apreciar el efecto inmediato que tiene sobre la estación la activación de algún botón o elemento de la pantalla y por otro lado nos permite hacer uso de elementos proporcionados para la programación de dichas pantallas como el uso de gráficas para monitorizar el comportamiento de ciertos procesos, el uso de avisos de sistema en caso de que se produzca algún error en la estación o el uso de recetas para cambiar ciertas variables del proceso, que de otro modo hubieran sido muy complicados de simular.

Por lo tanto, el software FACTORY I/O es un elemento más que recomendable tanto para todos aquellos que se estén iniciando en el mundo de la automatización industrial como para todas aquellas instituciones que se dediquen a la enseñanza de dicha materia.

10. BIBLIOGRAFÍA

- Manual SIMATIC STEP 7 en el Totally Integrated Automation Portal. Ingeniería intuitiva y eficiente: desde el microcontrolador hasta el controlador basado en PC. Consultado por última vez en enero de 2017.

https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Tia-Portal/tia_portal/Documents/Folleto%20STEP7%20en%20TIAP%20Nov11.pdf

- Manual PID Control with PID_COMPACT. Consultado por última vez en enero de 2017.

http://www.infopl.net/files/descargas/siemens/infoPLC_net_100746401_S7-1200_PID_Compact_DOKU_v1d0_en.pdf

- Página web de RealGames. Consultada por última vez en febrero de 2017.
<https://realgames.co/documentation/pages/viewpage.action?pageId=491525>

- Programación avanzada: Controlador PID. Universidad Miguel Hernández. Automatización Industrial. Consultado por última vez en enero de 2017.

<http://umh2210.edu.umh.es/wp-content/uploads/sites/800/2013/02/Practica-6.pdf>