



UNIVERSIDAD DE JAÉN  
*Escuela Politécnica Superior de Jaén*

Trabajo Fin de Grado

# APRENDIZAJE DE PROGRAMACIÓN DE PLCs CON SIMULACIONES EN FACTORY IO

**Alumno: Tomás Contreras Jodra**

Tutor: Alejandro Sánchez García.

Cotutor: Elisabet Estevez Estevez.

Dpto: Ingeniería Electrónica y Automática.

**Junio, 2021**



Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Ing. Electrónica y automática

Don **Alejandro Sánchez García** y **Elisabet Estevez Estevez** tutores del Proyecto Fin de Carrera titulado: **Aprendizaje de programación de PLCs con simulaciones en Factory IO**, que presenta **Tomás Contreras Jodra**, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2021

El alumno:

**Tomás Contreras Jodra**

Los tutores:

**Alejandro Sánchez García**

**Elisabet Estevez Estevez**

En primer lugar antes de comenzar con el TFG agradecer a mi familia, y en especial a mis padres, mis abuelos, mis hermanas y Gaia por el apoyo incondicional recibido durante esta etapa universitaria. ¡Por fin hemos conseguido el objetivo!

A Juani que desde el primer día que empecé la carrera ha estado a mi lado haciendo que sacase lo mejor de mi, mostrando una confianza plena en todo lo que hacia y siendo el pilar más fuerte que he tenido durante la carrera.

Gracias a Javi, María, Jesús y Fran. Empezamos la carrera con dudas, miedos e inseguridades, ¿seríamos capaces? ¿terminaríamos? ¿estaríamos solos? Pero estas dudas se disiparon rápido porque desde el primer día hemos sido un equipo, si uno fallaba o flaqueaba estaba el otro para ayudarlo y motivarlo a seguir adelante. Y esto ha sido clave para que en el día de hoy esté aquí exponiendo mi TFG. Porque sí, por fin hemos llegado al final de esta etapa y estoy seguro que lo que la universidad ha unido nadie lo va a separar.

Un especial agradecimiento al grupo Mottus por darme el apoyo y los conocimientos necesarios para poder realizar este proyecto. Ramón, Pablo, Andrés, muchas gracias por ayudarme a crecer como estudiante y futuro ingeniero.

Finalmente me gustaria dar las gracias a los excelentes profesores que hemos tenido durante estos 4 años de carrera, porque sin su ayuda, su trabajo y su dedicación esto no hubiese sido posible.

# Índice

Índice de ilustraciones y tablas .....	5
1. Introducción y objetivos.....	9
1.1. Justificación del TFG .....	9
1.2. Objetivos .....	10
2. Cuadernillo de prácticas alumno.....	11
2.1. Práctica 1 – Entorno TIA Portal .....	11
2.1.1. Introducción .....	12
2.1.2. Objetivos.....	13
2.1.3. Funcionamiento .....	13
Ejemplo 1 – Cinta y sensor. Digital (Factory IO) .....	21
Ejemplo 2 – Cinta y sensor. Analógico (Factory IO) .....	22
Ejemplo 3 – Cinta y sensor. Digital (TIA Portal + Factory IO) .....	23
Ejercicio 1 – Cinta y sensor. Analógico (Factory IO).....	27
2.1.4. Trabajo previo.....	30
2.1.5. En el laboratorio.....	30
2.2. Práctica 2 – Cuadro de mandos. Modos de trabajo.....	31
2.2.1. Introducción .....	32
2.2.2. Objetivos.....	32
2.2.3. Funcionamiento .....	32
2.2.4. Trabajo previo.....	34
2.2.5. En el laboratorio.....	34
2.3. Práctica 3 – Automatización distribuidor de cajas .....	35
2.3.1. Introducción .....	36
2.3.2. Objetivos.....	36
2.3.3. Funcionamiento .....	36
2.3.4. Trabajo previo.....	39
2.3.5. En el laboratorio.....	39
2.4. Práctica 4 – Módulo de trabajo Pick and Place .....	40
2.4.1. Introducción .....	41
2.4.2. Objetivos.....	41
2.4.3. Funcionamiento .....	41
2.4.4. Trabajo previo.....	44
2.4.5. En el laboratorio.....	44
2.5. Práctica 5 – Programación de un ascensor .....	45

Justificación del TFG

2.5.1.	Introducción .....	46
2.5.2.	Objetivos .....	48
2.5.3.	Funcionamiento .....	48
2.5.4.	Trabajo previo .....	52
2.5.5.	En el laboratorio.....	52
2.6.	Práctica 6 – Automatización clasificadora de piezas .....	53
2.6.1.	Introducción .....	54
2.6.2.	Objetivos.....	54
2.6.3.	Funcionamiento .....	54
2.6.4.	Trabajo previo.....	59
2.6.5.	En el laboratorio.....	59
2.7.	Práctica 7 – Llenado de un tanque. Control PID.....	60
2.7.1.	Introducción .....	61
2.7.2.	Objetivos.....	62
2.7.3.	Funcionamiento .....	62
2.7.4.	Trabajo previo.....	64
2.7.5.	En el laboratorio.....	65
3.	Cuadernillo de prácticas docente.....	66
3.1.	Práctica 1 – Entorno TIA Portal .....	66
3.1.1.	Introducción .....	67
3.1.2.	Objetivos.....	68
3.1.3.	Funcionamiento .....	68
	Ejemplo 1 – Cinta y sensor. Digital (Factory IO).....	76
	Ejemplo 2 – Cinta y sensor. Analógico (Factory IO) .....	77
	Ejemplo 3 – Cinta y sensor. Analógico (TIA Portal + Factory IO) .....	78
	Ejercicio 1 – Cinta y sensor. Analógico (Factory IO).....	82
3.1.4.	Trabajo previo.....	85
3.1.5.	En el laboratorio.....	85
3.2.	Práctica 2 – Cuadro de mandos. Modos de trabajo.....	86
3.2.1.	Introducción .....	87
3.2.2.	Objetivos.....	87
3.2.3.	Funcionamiento .....	87
3.2.4.	Trabajo previo.....	89
3.2.5.	En el laboratorio.....	89
3.2.6.	Solución .....	89
3.3.	Práctica 3 – Automatización distribuidor de cajas .....	92

Justificación del TFG

3.3.1.	Introducción .....	93
3.3.2.	Objetivos .....	93
3.3.3.	Funcionamiento .....	93
3.3.4.	Trabajo previo .....	96
3.3.5.	En el laboratorio.....	96
3.3.6.	Solución .....	96
3.4.	Práctica 4 – Módulo de trabajo Pick and Place .....	99
3.4.1.	Introducción .....	100
3.4.2.	Objetivos .....	100
3.4.3.	Funcionamiento .....	100
3.4.4.	Trabajo previo .....	103
3.4.5.	En el laboratorio.....	103
3.4.6.	Solución .....	103
3.5.	Práctica 5 – Programación de un ascensor .....	106
3.5.1.	Introducción .....	107
3.5.2.	Objetivos .....	109
3.5.3.	Funcionamiento .....	109
3.5.4.	Trabajo previo .....	113
3.5.5.	En el laboratorio.....	113
3.5.6.	Solución .....	113
3.6.	Práctica 6 – Automatización clasificadora de piezas .....	119
3.6.1.	Introducción .....	120
3.6.2.	Objetivos .....	120
3.6.3.	Funcionamiento .....	120
3.6.4.	Trabajo previo .....	125
3.6.5.	En el laboratorio.....	125
3.6.6.	Solución .....	125
3.7.	Práctica 7 – Llenado de un tanque. Control PID.....	132
3.7.1.	Introducción .....	133
3.7.2.	Objetivos .....	134
3.7.3.	Funcionamiento .....	134
3.7.4.	Trabajo previo .....	137
3.7.5.	En el laboratorio.....	137
3.7.6.	Solución .....	137
4.	Bibliografía .....	139

# Índice de ilustraciones y tablas.

Figura 2.1. Logo de TIA Portal y Factory IO.....	12
Figura 2.2. Abrir proyecto TIA Portal,.....	13
Figura 2.3. Vista de proyecto TIA Portal. ....	14
Figura 2.4. Dispositivos TIA Portal. ....	14
Figura 2.5. Árbol del proyecto. ....	15
Figura 2.6. Zona de programación. ....	15
Figura 2.7. Ejemplo 1 KOP.....	16
Figura 2.8. Ejemplo 2 KOP.....	16
Figura 2.9 Ejemplo 3 KOP.....	16
Figura 2.11. Inicio Factory IO. ....	17
Figura 2.12. Factory IO entorno de trabajo. ....	17
Figura 2.13 . Barra de tareas Factory IO.....	17
Figura 2.14. Menús Factory IO.....	18
Figura 2.15. Resto de botones Factory IO.....	18
Figura 2.16. Drivers Factory IO. ....	19
Figura 2.17 . Función de Factory IO en TIA Portal.....	20
Figura 2.18. Conexiones Factory IO. ....	20
Figura 2.19. Estación ejemplo 1.....	21
Figura 2.20. Cinta 1 y sensor. ....	21
Figura 2.21. Configuración analógica /digital Factory IO. ....	22
Figura 2.22. Ejemplo 3, cuadro de mandos. ....	23
Figura 2.23. Estación ejemplo 3.....	23
Figura 2.24. Exportar variables Factory IO. ....	23
Figura 2.25. Exportar variables Factory IO 2. ....	24
Figura 2.26. Importar variables en TIA Portal. ....	24
Figura 2.27. GRAFCET ejemplo 3. ....	25
Figura 2.28. Programación en KOP del ejemplo 3.....	26
Figura 2.29. Vídeo funcionamiento ejercicio 1. del ejercicio 1. ....	27
Figura 2.30. Estación en Factory IO	
Figura 2.31. Estación ejercicio 1.....	28
Figura 2.32. Conexión Factory IO ejercicio 1. ....	28
Figura 2.33. Programación variables analógicas. ....	28
Figura 2.34. Programación temporizadores. ....	29
Figura 2.35. GRAFCET ejercicio 1.....	29
Figura 2.36. Estación modos de trabajo.....	32
Figura 2.37. Vídeo del funcionamiento de la práctica 2. ....	33
Figura 2.38. Estación modos de trabajo en Factory IO.....	33
Figura 2.39. Estación distribuidora de cajas. ....	37
Figura 2.40. Cuadro de mandos de la estación distribuidora de cajas. ....	37
Figura 2.41. Conexiones distribuidora de cajas. ....	38
Figura 2.42. Vídeo del funcionamiento de la estación distribuidora de cajas. ....	38
Figura 2.43. Estación distribuidora de cajas en Factory IO.....	38
Figura 2.44. Robot cartesiano. ....	41
Figura 2.45. Estación pick and place. ....	42

Figura 2.46. Vídeo del funcionamiento de la estación Pick and place en Factory IO.	42	Figura 2.47. Estación Pick and place en Factory IO.	42
Figura 2.48. Nivel medio, apilado de cajas	43	Figura 2.49. Tipos de bloque TIA Portal.	46
Figura 2.50. Vídeo del funcionamiento de la estación del ascensor.	48	Figura 2.51. Estación del ascensor en Factory IO.	48
Figura 2.52. Descripción de la estación del ascensor.	49	Figura 2.53. Sensor visión.	50
Figura 2.54. Tipo de piezas.	50	Figura 2.55. Piezas utilizadas en el proceso.	51
Figura 2.56. Función cinta de las piezas.	51	Figura 2.57. Vídeo del funcionamiento de la estación clasificadora.	54
Figura 2.58. Estación clasificadora en Factory IO.	54	Figura 2.59. Transportador 1	55
Figura 2.60. Ascensor.	55	Figura 2.61. Cinta 1.	56
Figura 2.62. Cinta 2-3.	56	Figura 2.63. Robot.	57
Figura 2.64. Estación clasificadora completa.	57	Figura 2.65. Conexiones Factory IO.	58
Figura 2.66. PID.	61	Figura 2.67. Estación de llenado de un tanque.	62
Figura 2.68. Configuración de un PID en TIA Portal.	63	Figura 2.69. Nivel avanzado.	64
Figura 2.70. Vídeo funcionamiento estación llenado de un tanque.	64	Figura 2.71. Estación llenada de un tanque en Factory IO.	64
Figura 3.1. Logo de TIA Portal y Factory IO.	67		
Figura 3.2. Abrir proyecto TIA Portal.	68		
Figura 3.3. Vista de proyecto TIA Portal.	69		
Figura 3.4. Dispositivos TIA Portal.	69		
Figura 3.5. Ejemplo 1 KOP.	71		
Figura 3.6. Ejemplo 2 KOP.	71		
Figura 3.7. Ejemplo 3 KOP.	71		
Figura 3.8. Inicio Factory IO.	72		
Figura 3.9. Factory IO entorno de trabajo.	72		
Figura 3.10. Barra de tareas Factory IO.	72		
Figura 3.11. Menús Factory IO.	73		
Figura 3.12. Resto de botones Factory IO.	73		
Figura 3.13. Drivers Factory IO.	74		
Figura 3.14. Función de Factory IO en TIA Portal.	75		
Figura 3.15. Conexiones Factory IO.	75		
Figura 3.16. Estación ejemplo 1.	76		
Figura 3.17. Cinta 1 y sensor.	76		
Figura 3.18. Configuración analógica /digital Factory IO.	77		



Figura 3.19. Ejemplo 3, cuadro de mandos. ....	78
Figura 3.20. Estación ejemplo 3.....	78
Figura 3.21. Exportar variables Factory IO. ....	78
Figura 3.22. Exportar variables Factory IO 2. ....	79
Figura 3.23. Importar variables en TIA Portal. ....	79
Figura 3.24. GRAFCET ejemplo 3. ....	80
Figura 3.25. Programación en KOP del ejemplo 3.....	81
Figura 3.26. Vídeo funcionamiento ejercicio 1. ....	82
Figura 3.27. Estación en Factory IO del ejercicio 1 .....	82
Figura 3.28. Estación ejercicio 1. ....	83
Figura 3.29. Conexión Factory IO ejercicio 1.....	83
Figura 3.30. Programación variables analógicas. ....	83
Figura 3.31. Programación temporizadores. ....	84
Figura 3.32. GRAFCET ejercicio 1.....	84
Figura 3.33. Estación modos de trabajo.....	87
Figura 3.34. Programación función modos de trabajo. ....	89
Figura 3.35. Programación modo de emergencia. ....	90
Figura 3.36. Programación modo stop. ....	90
Figura 3.37. Programación modo manual. ....	90
Figura 3.38. Programación modo automático. ....	91
Figura 3.39. Programación modo homing. ....	91
Figura 3.40. Estación distribuidora de cajas. ....	94
Figura 3.41. Cuadro de mandos de la estación distribuidora de cajas. ....	94
Figura 3.42. Conexiones distribuidora de cajas. ....	95
Figura 3.43. GRAFCET distribuidora de cajas. ....	97
Figura 3.44. Robot cartesiano. ....	100
Figura 3.45. Estación pick and place. ....	101
Figura 3.46. Nivel medio, apilado de cajas. ....	102
Figura 3.47. GRAFCET transportador de palets. ....	103
Figura 3.48. GRAFCET cinta y robot. ....	104
Figura 3.49. Conexiones Factory IO. ....	105
Figura 3.50. Tipos de bloque TIA Portal.....	107
Figura 3.51. Descripción de la estación del ascensor.....	110
Figura 3.52. Sensor visión.....	111
Ilustración 3.53. Tipo de piezas.....	111
Figura 3.54. Piezas utilizadas en el proceso.....	112
Figura 3.55. Función cinta de las piezas.....	112
Figura 3.56. GRAFCET transportador de cajas. ....	114
Figura 3.57. GRAFCET cinta piezas. ....	114
Figura 3.58. GRAFCET ascensor. ....	115
Figura 3.59. Bloques de trabajo. ....	117
Figura 3.60. Función ascensor.....	118
Figura 3.61. Función cintas. ....	118
Figura 3.62. Transportador 1.....	121
Figura 3.63. Ascensor. ....	121
Figura 3.64. Cinta 1.....	122
Figura 3.65. Cinta 2 y 3.....	122
Figura 3.66. Robot.....	123

Figura 3.67. Estación clasificadora completa.....	123
Figura 3.68. Conexiones Factory IO. ....	124
Figura 3.69. GRAFCET Transportador 1. ....	126
Figura 3.70. GRAFCET Ascensor. ....	126
Figura 3.71. GRAFCET cinta 1. ....	127
Figura 3.72. GRAFCET cinta 2 y 3.....	127
Figura 3.73. GRAFCET Robot 2 ejes.....	128
Figura 3.74. GRAFCET cinta piezas.....	129
Figura 3.75. Estructura general de bloques. ....	131
Figura 3.76. Función cinta piezas. ....	131
Figura 3.77. PID. ....	133
Figura 3.78. Estación de llenado de un tanque.....	134
Figura 3.79. Configuración de un PID en TIA Portal.....	135
Figura 3.80. Nivel avanzado.....	136
Figura 3.81. Bloques del programa.....	138

# 1. Introducción y objetivos

## 1.1. Justificación del TFG

A día de hoy la automatización industrial es un sector con un gran crecimiento y por ello es importante que los alumnos obtengan competencias básicas y avanzadas a la hora de poder enfrentarse a cualquier proyecto que se le ponga por delante sin tener en cuenta su dificultad. Al tratarse de un sector muy potente hay mucha variedad de autómatas programables en el mercado, entre las más destacadas aparecen nombres como Allen Bradley, Siemens, ABB, Omron o Mitsubishi entre muchos otros.

Al encontrar una gran variedad de fabricantes por lo que resulta imposible condensar todos ellos en unas prácticas, ante esta situación se ha creído oportuno que el alumno adquiriera una base sólida a la hora de automatizar procesos para que pueda trabajar con cualquier tipo de PLC.

Para el desarrollo de estas prácticas se centrará en el fabricante Siemens y para ser más exactos en el modelo 1200. Este es el modelo elegido para estas prácticas ya que es un PLC muy completo y con un precio económico lo cual lo hace perfecto para una finalidad educativa.

Cada marca de PLCs tiene su software específico para la programación autómatas, en nuestro caso el software que se utilizará es el conocido TIA Portal. Este software es muy completo y sirve para programar la gran gama de productos que oferta Siemens en el mercado como HMI o variadores de velocidad entre otros. Con él también se podrá configurar las comunicaciones entre los distintos dispositivos, consiguiendo mayor comodidad a la hora de automatizar un proceso.

La mayoría de fabricantes adjuntan al software principal de programación del PLC otro software que permite la simulación de los programas sin necesidad de tener el PLC físico. En este caso Siemens además del software TIA Portal cuenta con otro llamado PLCSIM en que permite visualizar la evolución del proceso. Esta opción es menos completa que si trabajase con el PLC físico pero su principal punto a favor es que tanto el docente como el alumno pueden trabajar desde casa simulando y una vez funcione poder probarlo en el PLC físico.

Como se observa los fabricantes de autómatas han apostado por simuladores que faciliten el trabajo del programador a la hora de trabajar desde casa, pero esto no ha quedado solo ahí, empresas como RealGames han desarrollado softwares de simulación con los cuales se puede diseñar nuestra propia estación de trabajo virtual y con ellos poderlos conectar a softwares como se describió anteriormente.

El software en cuestión se llama FACTORY I/O y está enfocado a la educación, este será el que se utilizará como apoyo en estas prácticas.

Con este trabajo final de grado se pretende crear un programa educativo para el docente que facilite el aprendizaje de programación de PLCs ayudándose de herramientas de simulación como Factory IO y que estas sirvan de base para impartir en asignaturas de la rama de automatización. Ya que con ellas el alumno será capaz de adquirir los conocimientos necesarios para la programación de PLCs, además al contar con un apoyo gráfico como es el simulador.

## 1.2. Objetivos

En este TFG se pretende que el alumno aplique los conocimientos adquiridos en clase sobre programación de autómatas trabajando en los conceptos claves en pequeñas prácticas. Las prácticas constarán de estaciones de trabajo virtual las cuales tienen un comportamiento similar a los que presentan muchos procesos en la industria.

Conociendo el método de trabajo estos son los objetivos que se intentarán alcanzar:

- El objetivo general de este trabajo es que el alumno adquiera competencias básicas y avanzadas en la programación de autómatas programables.

Para alcanzar el objetivo general nos marcaremos los siguientes objetivos específicos:

### Para el alumno:

- Afianzar y adquirir los conocimientos en cuanto a la programación de autómatas se refiere.
- Desarrollo de las prácticas según la planificación descrita en cada una de ellas.
- Fomentar la participación activa tanto en el laboratorio como a distancia.
- El alumno debe adquirir los conocimientos básicos de los softwares TIA Portal y el software de simulación Factory IO.

### Para el docente:

- Transmitir los conocimientos en cuanto a la programación de autómatas se refiere.
- Adaptar los contenidos de la asignatura a diferentes escenarios de enseñanza/aprendizaje. Presencial, semipresencial o a distancia.
- Facilitar al docente su labor educativa.

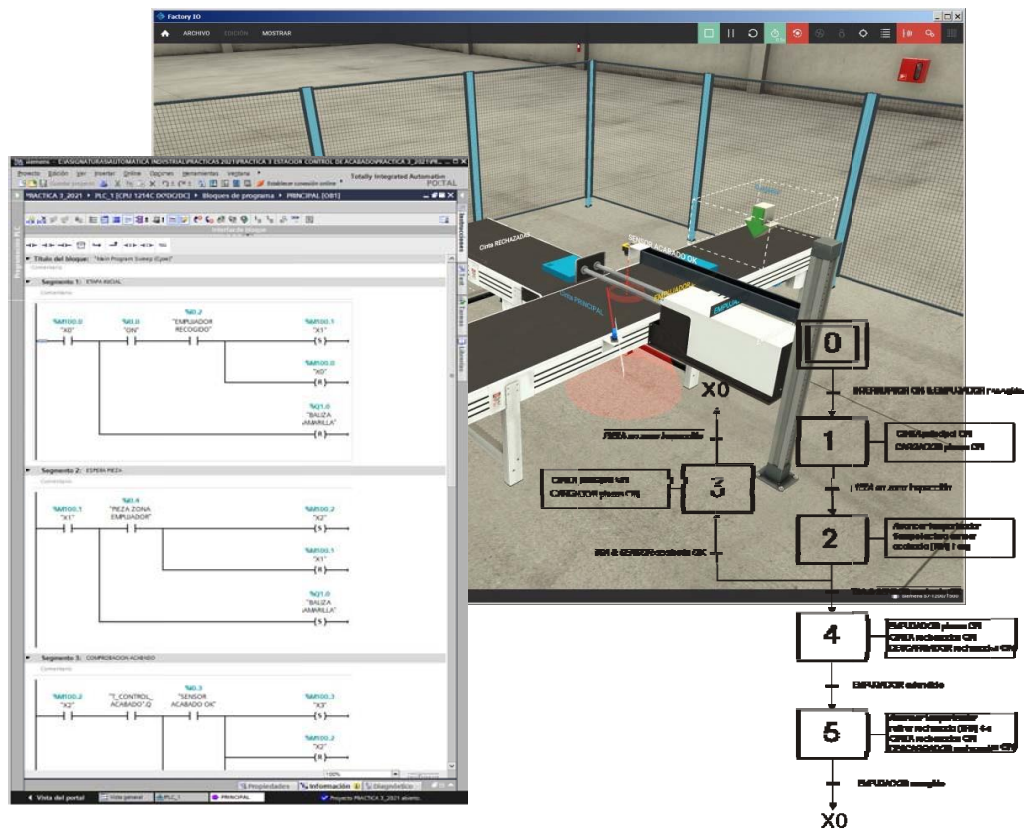
Una vez marcados nuestros objetivos del TFG la estructura que hemos definido para su aplicación, hemos dividido en dos partes los contenidos a desarrollar. Uno centrado en el alumno "**Cuadernillo de prácticas del alumno**" y otro material con los contenidos del docente que incluyen las soluciones de las prácticas a desarrollar "**Cuadernillo de prácticas del docente**".

## 2. Cuadernillo de prácticas alumno

### 2.1. Práctica 1 – Entorno TIA Portal



## AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 1. ENTORNO TÍA PORTAL

ALUMNO:

GRUPO:

### 2.1.1. Introducción

En la automatización de cualquier tipo de proceso siempre es necesario introducir un elemento capaz de controlar las operaciones que se llevan a cabo en él, los Controladores Lógicos Programables o más conocidos como PLC son los encargados de realizar dicha función. En esta práctica se aprenderá sobre que es un PLC, ejemplos de cómo utilizarlos y como programarlos. Para ello se usarán principalmente dos softwares diferentes (*Figura 2.1*):

- **Totally Integrated Automation Portal (TIA Portal)**
- **Factory IO**



**Figura 2.1. Logo de TIA Portal y Factory IO.**

El alumno se iniciará en este software realizando pequeños ejemplos guiados y de esta forma conocer más a fondo todas las herramientas que este contiene.

Como herramienta de ayuda para realizar estos ejemplos se hará uso del software de simulación FACTORY IO.

Factory I/O es un software para automatización en tiempo real donde se puede construir y simular sistemas industriales y utilizarlos con las tecnologías de automatización más comunes. Esta simulación es totalmente interactiva e incluye gráficos de alta calidad y sonido, proporcionando un entorno realista industrial.

Factory I/O utiliza una tecnología innovadora que permite una creación fácil y rápida de los sistemas industriales en 3D con solo arrastrar y soltar. Cualquiera de los sistemas construidos se puede controlar en tiempo real mediante la conexión de Factory I/O y equipos externos como PLCs, microcontroladores, FPGA, etc.

Factory I/O es una valiosa herramienta de enseñanza para la formación de futuros técnicos e ingenieros en varios programas y cursos tales como automatización industrial, Mecatrónica, Ingeniería Eléctrica, Ingeniería Mecánica, Instrumentación y muchos más.

## 2.1.2. Objetivos

En esta práctica el objetivo fundamental es que el alumno se inicie en la programación de PLCs utilizando el entorno TIA Portal de Siemens.

El alumno será capaz de generar GRAFCETS para la solución de problemas de automatización y posteriormente implementarlos en lenguaje KOP dentro de la aplicación TIA Portal.

Para todo ello será imprescindible iniciarse en el software de simulación Factory IO ya que será en él donde se comprobará que lo que se ha programado funciona correctamente.

Objetivos principales:

- Saber configurar los parámetros de TIA Portal.
- Ser capaces de crear un proyecto.
- Configurar la conexión entre Factory IO y TIA Portal.
- Introducirse en pequeños proyectos.
- Saber aplicar los conocimientos aprendidos en clase para programar diferentes proyectos.

## 2.1.3. Funcionamiento

### 2.1.3.1. Entorno TIA Portal

Se empezará creando un primer proyecto en TIA Portal para ello en primer lugar se inicia la aplicación y se hace clic sobre “Crear Proyecto”. (Figura 2.2) En este se debe elegir el nombre, la ruta y el autor. Una vez hecho se pulsa sobre crear proyecto.

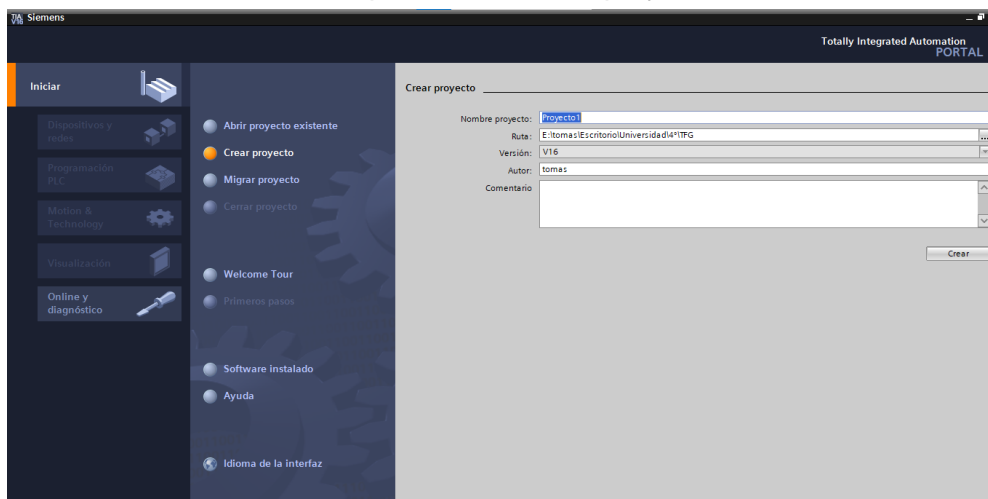


Figura 2.2. Abrir proyecto TIA Portal,

## Práctica 1 – Entorno TIA Portal

Después de esto hay que pulsar en el botón que hay en la esquina izquierda inferior “Vista de proyecto”. Esto abrirá la vista general del proyecto. (Figura 2.4)

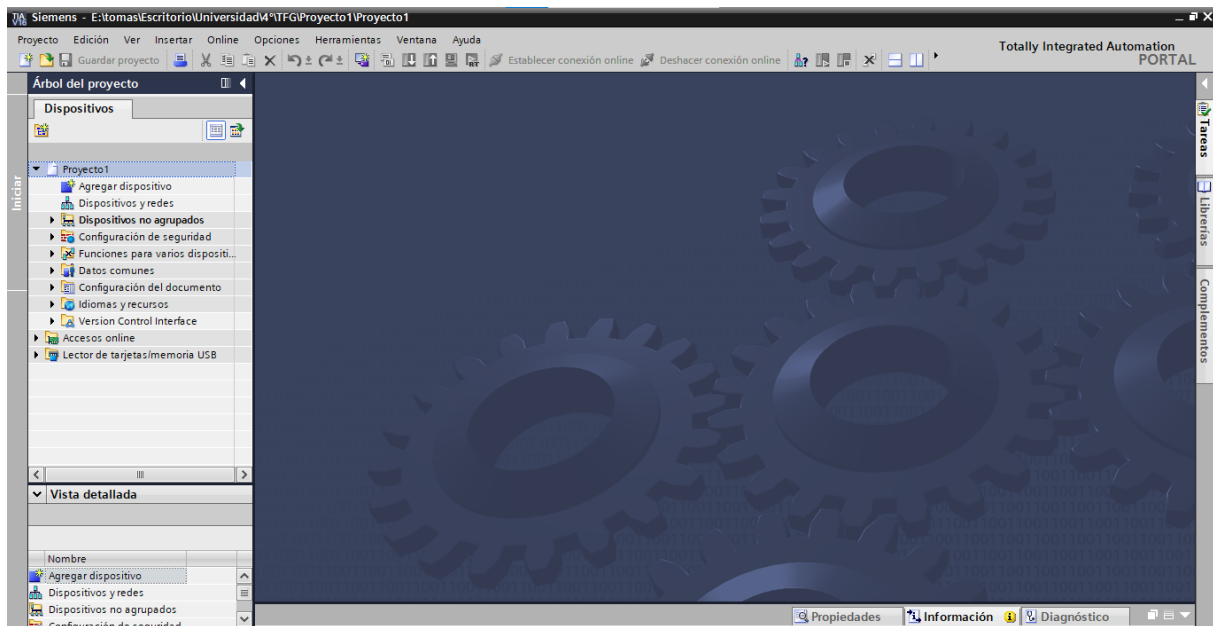


Figura 2.3. Vista de proyecto TIA Portal.

Para añadir un PLC se debe pulsar en “Agregar dispositivo” el cual se encuentra a la izquierda en el árbol del proyecto.

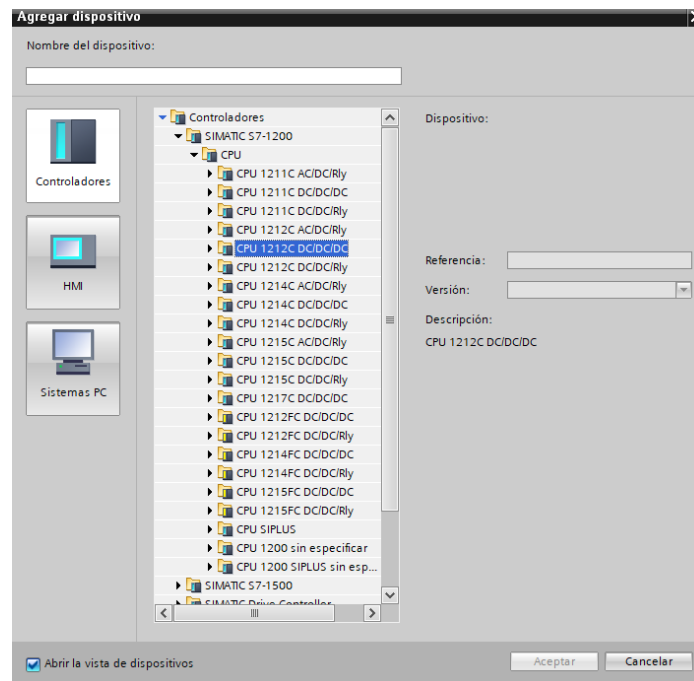


Figura 2.4. Dispositivos TIA Portal.



Práctica 1 – Entorno TIA Portal

Desde aquí (Figura 2.5) habrá que buscar el PLC que tengamos. Una vez añadido se debe configurar el protocolo IP del PLC al cual se accede desde las propiedades del PLC.

Pasemos ahora a explicar los diferentes bloques que se pueden encontrar en TIA Portal una vez configurado el PLC.

A la izquierda el Árbol de trabajo (Figura 2.6), en él se localiza todo el contenido de cada uno de los dispositivos que contiene el proyecto.

**Bloques de programa:** En él se localizan los diferentes bloques de programación que ofrece TIA Portal, para acceder a ellos se pulsa en Agregar nuevo bloque y se abrirá un desplegable donde seleccionar el dispositivo a añadir.

**Variables PLC:** En esta parte es donde se organizan las variables del programa. Para ello hay que añadir una nueva tabla de variables, y una vez dentro de ella poner el nombre de la variable, el tipo de dato y la dirección de la misma. Aparte de añadirla manualmente se pueden importar.

Pasando ahora a la parte de la programación esta se lleva a cabo en los OB que son un tipo de bloques de datos.

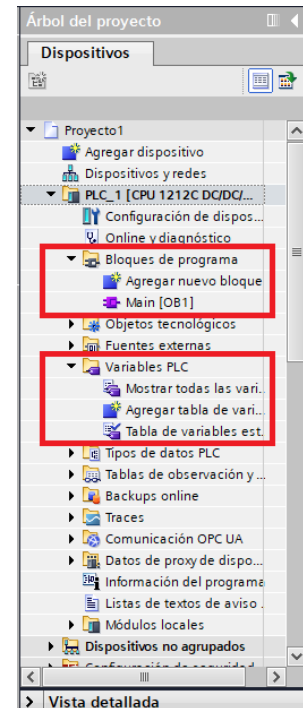


Figura 2.5. Árbol del proyecto.

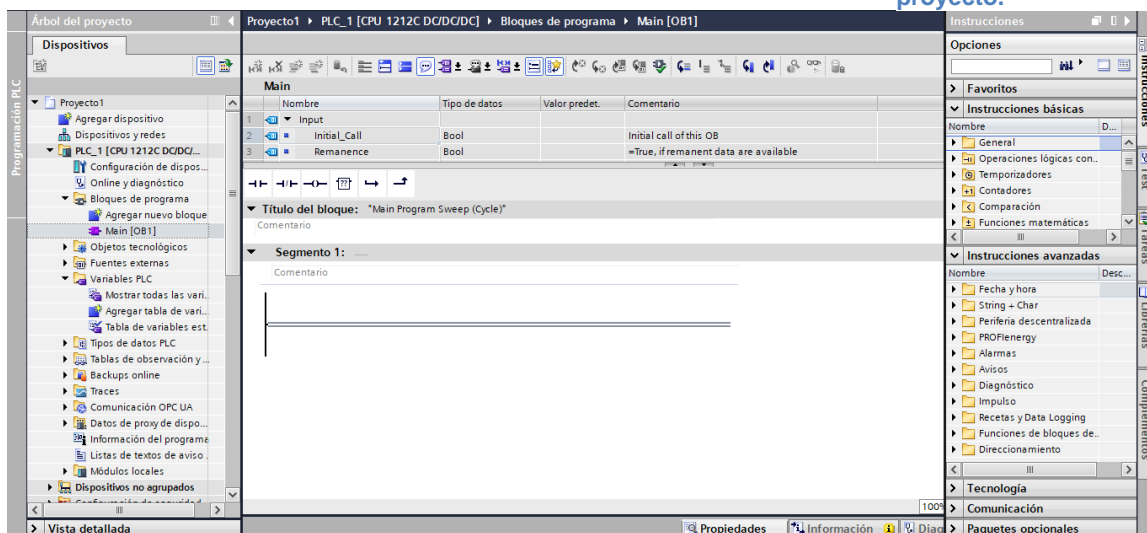


Figura 2.6. Zona de programación.

En la parte central (Figura 2.7) es donde se escribe el código en lenguaje KOP y en la parte derecha en el apartado “Instrucciones” se localizan todos los objetos que disponibles para la programación.

Estos son pequeños ejemplos de programación en KOP:

Ejemplo 1: En este caso si el sensor se activa el motor estará funcionando mientras el sensor esté activo. (Figura 2.8)

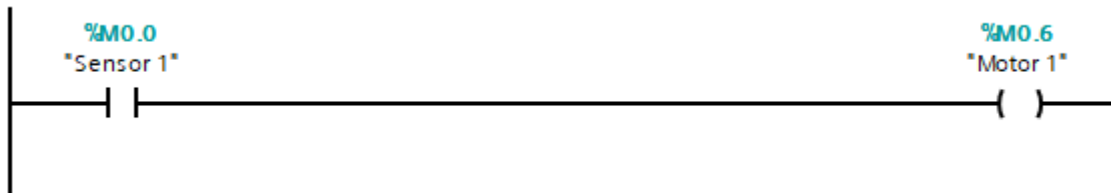


Figura 2.7. Ejemplo 1 KOP.

Ejemplo 2: Para este caso deben darse dos condiciones que el sensor 1 esté activo y el sensor 2 esté desactivado, solo en ese caso se activará el motor. (Figura 2.9)

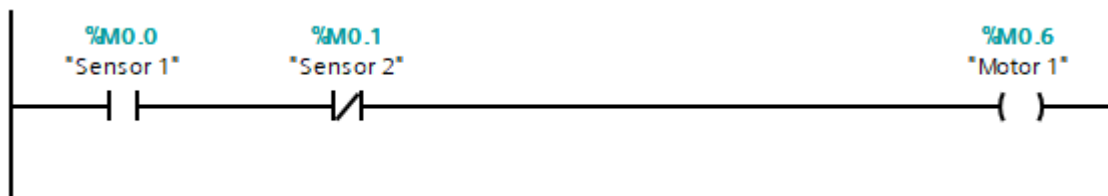


Figura 2.8. Ejemplo 2 KOP

Ejemplo 3: En esta ocasión hay 3 posibilidades de que el motor se active, estas serían que el sensor 2 se active y el 3 no, que se active el sensor 3 y el sensor 2 no o finalmente que ambos se activen. (Figura 2.10)



Figura 2.9 Ejemplo 3 KOP.

### 2.1.3.2. Entorno Factory IO

Se comienza abriendo el software de simulación Factory IO. En la pantalla principal (*Figura 2.10*) se encuentra a la derecha de la misma todo tipo de documentación y manuales sobre Factory IO, que pueden servir de gran ayuda a la hora de resolver diversas dudas.

En la parte izquierda de la pantalla se encuentra el menú que contiene diversas opciones como crear una nueva escena, abrir una nueva escena o abrir una escena prediseñada.

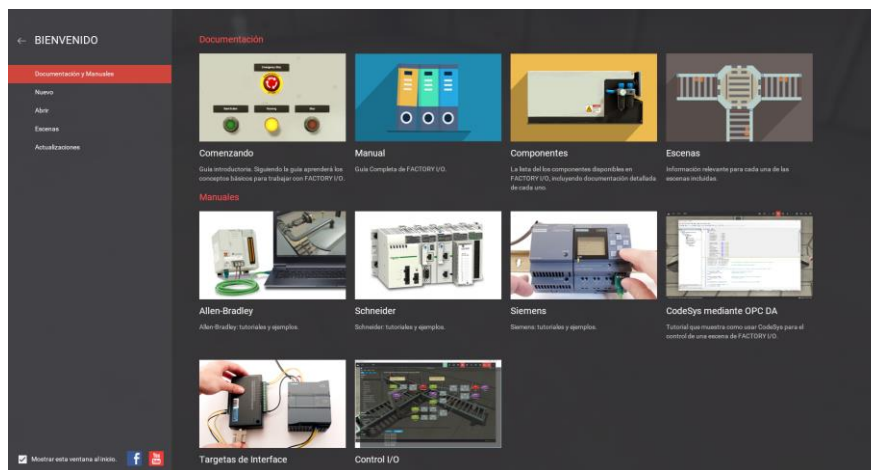


Figura 2.11. Inicio Factory IO.

El espacio virtual (*Figura 2.12*) del que consta Factory IO es de una nave industrial vacía donde se colocarán los componentes con los que se trabajaran, la interface que presenta es muy simple e intuitiva y eso facilita mucho la fluidez a la hora de trabajar en él.

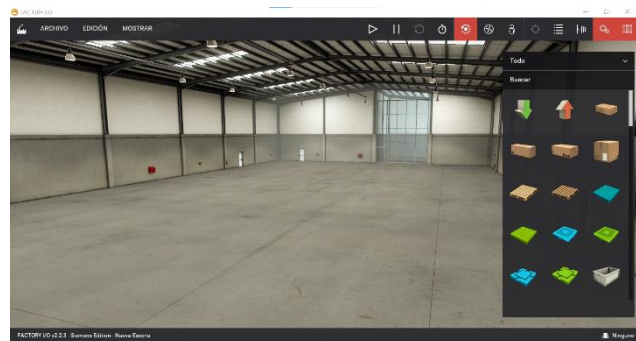


Figura 2.12. Factory IO entorno de trabajo.

#### Barra de tareas

En la parte superior se encuentra la barra de tareas (*Figura 2.13*) la cual consta de diversos comandos que se explicarán a continuación:



Figura 2.13 . Barra de tareas Factory IO.

En los primeros cuatro botones (*Figura 2.14*) de la parte izquierda se encuentran los clásicos botones de control del programa.

- El primero redirigirá a la pantalla de principal descrita anteriormente.

## Práctica 1 – Entorno TIA Portal

- “Archivo”, este permitirá guardar, crear, abrir o cerrar un programa, en las opciones dirigirá a una pantalla donde configurar aspectos relacionados con el audio, video, controles, licencias...etc. El apartado de los drivers es uno de los más importantes de esta sección ya que dentro de este se configurarán las conexiones con los diferentes controladores que se utilizan en el proyecto.

- “Edición”, en este desplegable se encuentran opciones como copiar, pegar o deshacer entre otras, tareas que son habituales en cualquier programa.

- “Mostrar”, desde esta pestaña se pueden marcar que menús se quieren mostrar o no.



Figura 2.14. Menús Factory IO.

El resto de botones de la parte derecha (Figura 2.15) se dividen en tres grupos:

- Proceso, con estos cuatro botones se puede controlar la simulación haciendo que esta comience, pare, vuelva a su estado inicial o que el tiempo de la simulación transcurra más lento.

-Cámaras, desde estos botones se controla la forma de visualizar el proceso, Factory IO cuenta con 3 tipos de cámaras, la primera una cámara orbital, la segunda una cámara “mosca” que sería una vista similar a la que se tiene desde un dron, y la última sería una cámara de primera persona en la que se vería lo que un operario de lesa industria vería. Finalmente el último botón de esta sección es una cámara de seguimiento en la que al pulsar en un objeto esta cámara se dedicaría a seguir este objeto a lo largo del proceso.

-Mostrar, estos botones sirven para mostrar u ocultar diferentes opciones, el primero mostrará un listado de cámaras guardadas por el usuario anteriormente, los dos siguientes permiten mostrar las etiquetas de los sensores y de los actuadores y el último botón sirve para desplegar la paleta con todos los componentes disponibles.



Figura 2.15. Resto de botones Factory IO

### 2.1.3.3. Conexión TIA Portal y Factory IO

#### Conexión con PLC físico

1. Se conecta el PLC a la red.
2. Se abre un nuevo proyecto en TIA PORTAL.
3. Se agrega un CPU para estas prácticas se utilizará un S7-1200 1212 DC/DC/DC.
4. Se clica en detectar dispositivo, se selecciona PN/IE como tipo de interfaz y se elige el adaptador de red. Una vez hecho esto se pulsa sobre buscar, aparecerá el PLC y se clica sobre Cargar.
5. Una vez enlazado el PLC a TIA Portal se procederá a configurarlo. Para ello se debe ir a configuración de dispositivo, pulsar sobre el CPU y propiedades.
6. Las primeras 10 entradas están reservadas internamente en el programa Factory IO así que se debe configurar un offset para empezar a escribir entradas a partir de esa. Para ello dentro de propiedades hay que pulsar en el desplegable DI 6/ DQ4 > Direcciones E/S > Direcciones de entrada y en Dirección inicial e introducir un 10.

*(De lo contrario, los valores escritos por Factory I / O se sobrescribirán cuando el estado de las entradas físicas se copie en la memoria I)*

7. Para terminar la configuración cambiar algunos aspectos de Protección y Seguridad.

Hay que entrar en Protección & Seguridad y realizar los siguientes ajustes:

- Nivel de acceso > Acceso completo.
- Mecanismos de conexión > Permitir acceso vía comunicación PUT/GET.

8. Una vez hecho esto cargar el software que se ha programado en el PLC y abrir Factory IO.
9. En Factory IO se abrirá el proyecto y se debe clicar en Archivo > Drivers (o bien F4), aparecerá un desplegable (*Figura 2.16*) y en él se seleccionará el CPU utilizado.
10. Dentro de Configuración, se escribirá el IP del PLC utilizado en el apartado "Host" y habrá que configurar tanto el adaptador de Red como el número de entradas y salida de cada tipo.

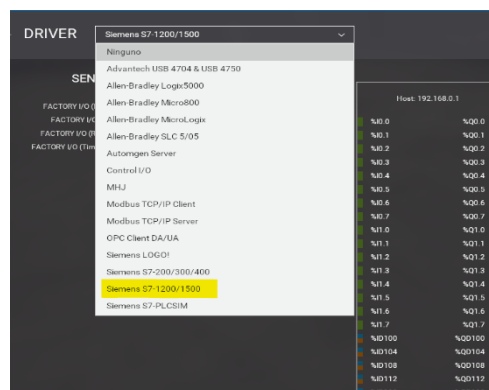


Figura 2.16. Drivers Factory IO.

11. Una vez realizado volver a la pantalla de Drivers y presionar sobre el botón de Conectar. Y el PLC quedará conectado a Factory IO.

### Conexión con S7-PLCSIM

1. Lo primero que se debe hacer es entrar en la página de Factory IO y descargar la plantilla de proyecto compatible para TIA Portal, sin esto no se podrán comunicar ambos programas.

### [Plantilla TIA Portal](#)

2. Una vez descargada dentro de la plantilla se puede encontrar una función (Figura 2.17) la cual hay que escribir en el OB1. El programa comenzaría a partir del segmento 2.

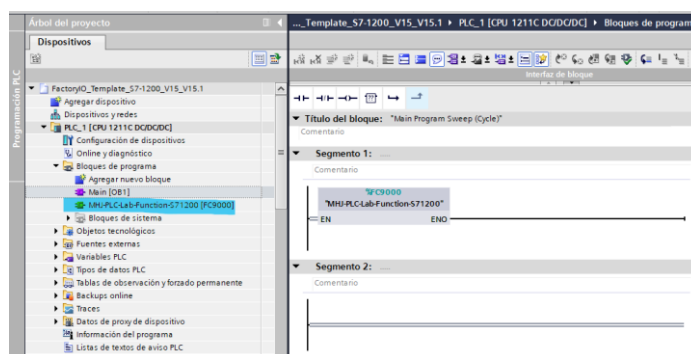


Figura 2.17 . Función de Factory IO en TIA Portal.

3. Tras esto, seleccionar el dispositivo e iniciar la simulación.
4. Se elegirá PN / IE como el tipo de interfaz PG / PC y en la interfaz PG / PC seleccione PLCSIM S7-1200 / S7-1500. Presionar Iniciar búsqueda. Cuando finalice el escaneo, seleccione el dispositivo y presione Cargar.
5. Cuando aparezca la ventana del PLCSIM es importante iniciar el módulo, RUN.
6. Abrir Factory IO, entrar en el proyecto y pulsar en Archivo > Drivers (o bien F4). Seleccionar “Siemens S7-PLCSIM”.
7. Entrar en el panel de Configuración, dentro de configuración seleccionar el modelo de CPU y configurar las entradas y salidas.
8. Presionar el botón de Conectar y el PLCSIM y Factory IO quedarían conectados (Figura 2.18).



Figura 2.18. Conexiones Factory IO.

### Ejemplo 1 – Cinta y sensor. Digital (Factory IO)

Una vez se sabe cómo conectar las diferentes entradas y salidas del proyecto se realizarán pequeños ejemplos. Para este ejemplo no es necesario conectar el PLC. Se usará una cinta (da igual el tamaño), un objeto que colocar sobre ella (en este caso se usará una “Stackable box”) y un sensor retro reflectivo. (Figura 2.19)

El objetivo de este ejemplo es ver cómo funcionan los actuadores y sensores en Factory IO. Para ello pulsar sobre el nombre de cada objeto. Arriba a la derecha se abrirá una pestaña (Figura 2.20) donde ver el estado y desde la cual se podrá controlar la variable asociada.

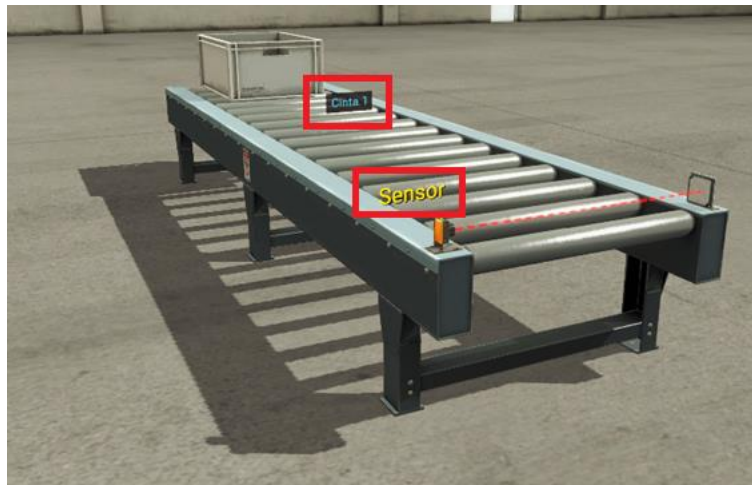


Figura 2.19. Estación ejemplo 1



Figura 2.20. Cinta 1 y sensor.

Si se pulsa en el círculo verde que aparece al lado de la cinta se observará como este pasa a azul y la caja empieza a moverse, se está forzando la variable a True. Cuando la caja pase por el sensor se verá como el sensor se ilumina. Esta es la forma de indicar de que el sensor está activo.

Esta es la configuración digital de la cinta, se ha usado la cinta ya que es un elemento simple y en la que ver de forma visual que es lo que sucede. Pero cualquier elemento que tenga este tipo de configuración se controla de la misma forma.

## Ejemplo 2 – Cinta y sensor. Analógico (Factory IO)

En el ejemplo anterior se aprendió como controlar y visualizar los controladores analógicos en Factory IO, en este ejemplo se verá cómo cambiar la configuración de los actuadores y como trabajar con ellos cuando estos sean analógicos.

Para ello se utilizará la escena creada en el ejemplo anterior. Hay que pulsar en el botón derecho sobre la cinta y seleccionar la opción de configuración (*Figura 2.21*). Desde ahí se podrá cambiar la configuración de casi cualquier objeto de Factory IO.

Una vez dentro quitar la configuración y como se hizo en el ejemplo 1 pulsar sobre el nombre de la cinta, como se observa ahora la ventana ha cambiado y se encuentra un Slider desde el cual controlar el voltaje que se aplica al motor haciendo que este haga girar la cinta a mayor o menor velocidad.

Cuando se marca la configuración analógica se verá en el apartado de las conexiones (apartado de los Drivers) como la variable cambia de Bool a DWord. Esto se debe a que ahora para controlar esta variable se usa un dato de valor int.

Como se comentó anteriormente este tipo de configuración se da en casi todos los actuadores. Aunque no todos son completamente iguales son muy intuitivos por ello antes de comenzar a programar una máquina que no se conozca se recomienda leer el manual para entender las variables que se manejarán y las diferentes configuraciones que esta tenga.

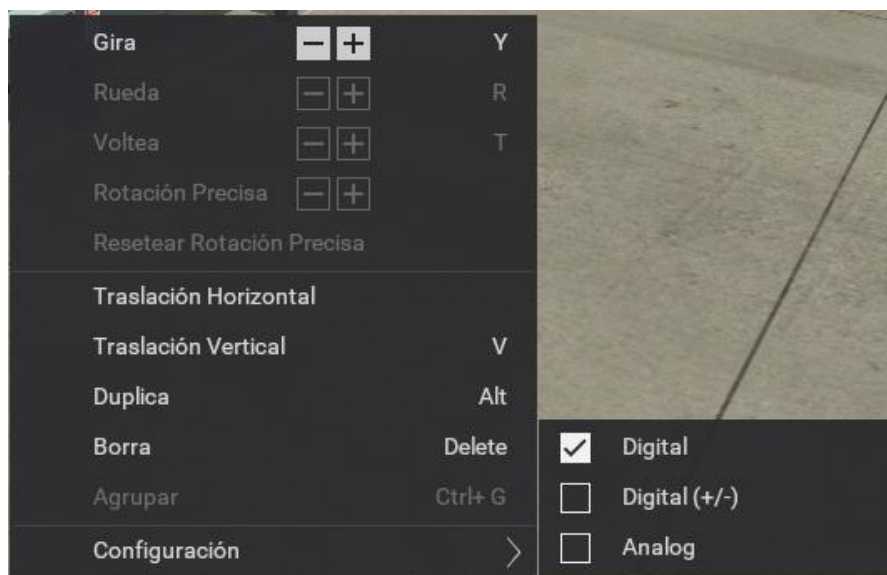


Figura 2.21. Configuración analógica /digital Factory IO.



### Ejemplo 3 – Cinta y sensor. Digital (TIA Portal + Factory IO)

Una vez se conoce el funcionamiento de los elementos con los que se va a trabajar en Factory IO ya se puede pasar a su programación. Es importante que antes de programar se conozca que es lo que se va a programar, sus entradas, sus salidas y su forma de actuar ya que conociendo a la perfección la maquina se podrá programar de una forma más eficiente.

En este ejemplo (Figura 2.23) constará de los siguientes elementos:

- Un emisor
- Un transportador (Digital)
- Un sensor
- Una salida
- Un cuadro de mandos compuesto por un botón de manual, otro de paro y uno de automático. (Figura 2.22)



Figura 2.22. Ejemplo 3, cuadro de mandos.



Figura 2.23. Estación ejemplo 3.

Lo primero que se hará será configurar las variables del PLC en Factory IO y exportarlas a TIA Portal para así trabajar con ellas desde el software de Siemens.

Recordar que para acceder a esto se debe pulsar sobre “ARCHIVO” (Figura 2.24) y después pulsar en drivers. Se abrirá la pantalla de configuración, colocar nuestros sensores y actuadores y seleccionar o bien el PLCSIM o el PLC físico.

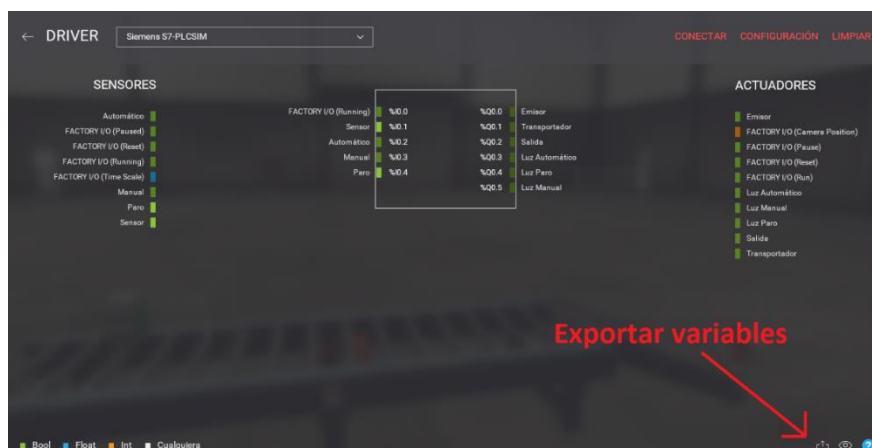


Figura 2.24. Exportar variables Factory IO.

Práctica 1 – Entorno TIA Portal

En la esquina inferior derecha se encuentra el botón para exportar las variables, se pulsa y se clicla en “DE ACUERDO” o si no sabes la ubicación donde este archivo se guardará pulsar en “ABRIR CARPETA” (Figura 2.25).

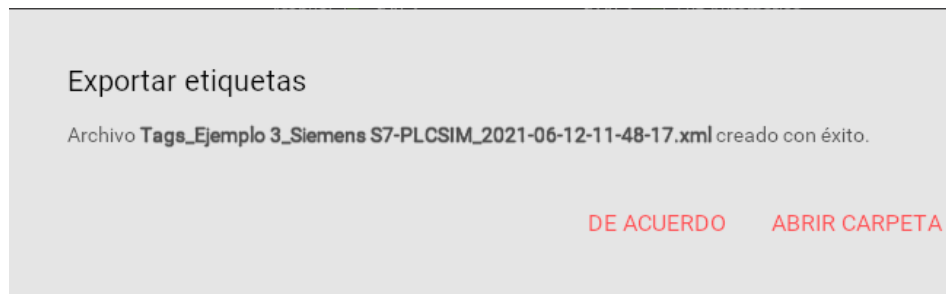


Figura 2.25. Exportar variables Factory IO 2.

Una vez hecho esto abrir ahora a TIA Portal, crear un proyecto nuevo como se ha visto anteriormente y cuando esté creado ir a las tablas de variables e importar las variables generadas anteriormente. (Figura 2.26)

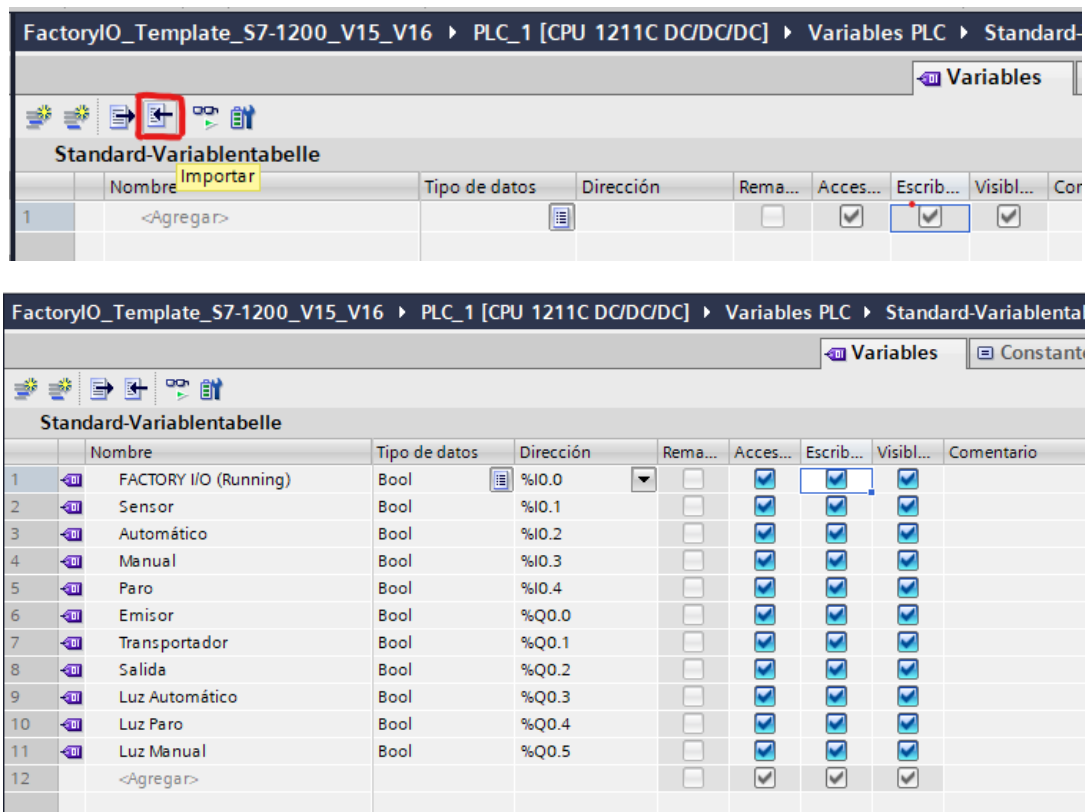


Figura 2.26. Importar variables en TIA Portal.

Pasadas las variables al TIA Portal es turno de diseñar el GRAFCET para el ejercicio propuesto, esta es una de las opciones que se ha diseñado (Figura 2.27).

Para ello se usará la escena creada en el ejemplo anterior. Pulsar en el botón derecho sobre la cinta y pulsar sobre la opción de configuración. Desde ahí se podrá cambiar la configuración de casi cualquier objeto de Factory IO.

Práctica 1 – Entorno TIA Portal

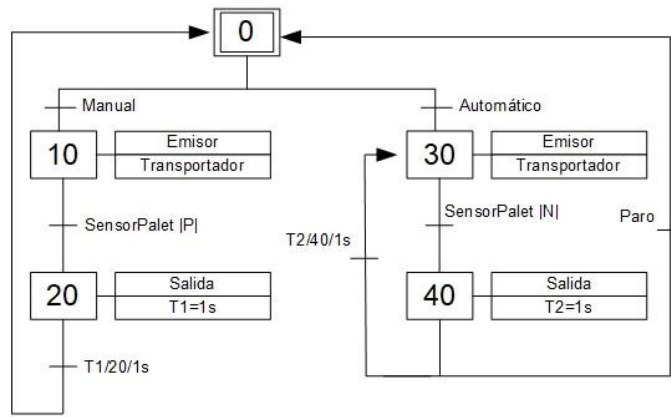


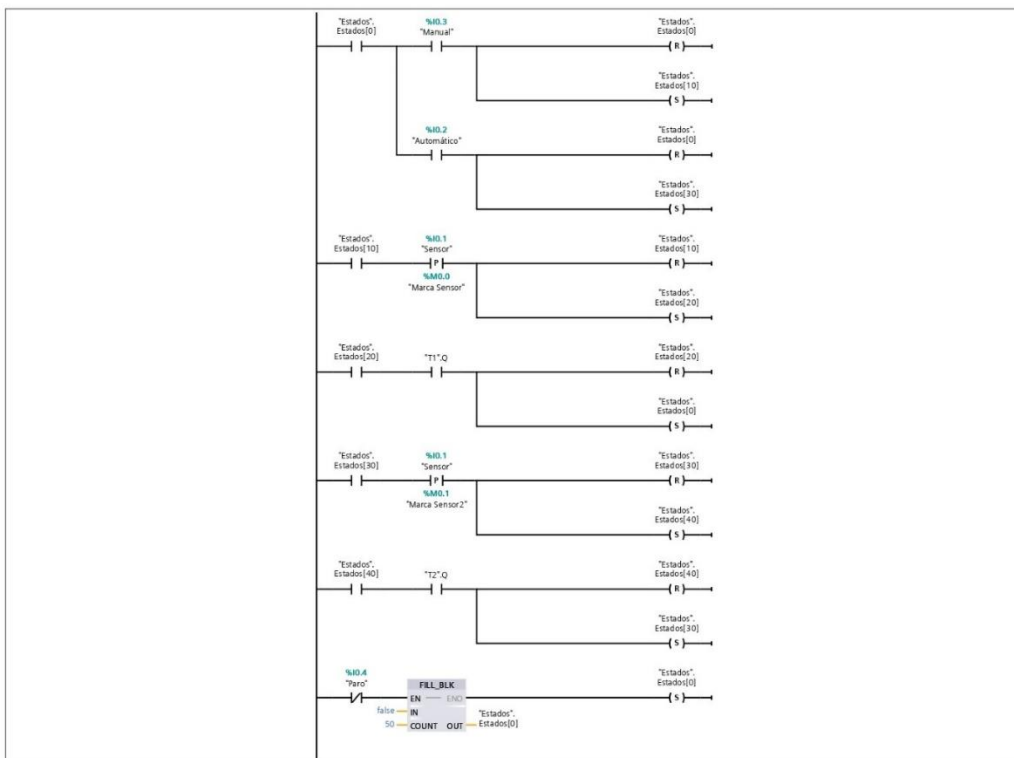
Figura 2.27. GRAFCET ejemplo 3.

Ahora toca programar lo diseñado en el GRAFCET en KOP en TIA Portal, quedaría de la siguiente forma:

Segmento 1:



Segmento 2:



Segmento 3: Transportador

Práctica 1 – Entorno TIA Portal



Figura 2.28. Programación en KOP del ejemplo 3.

Terminada la programación solo quedaría ejecutar el programa en TIA Portal y comprobar su funcionamiento con Factory IO.

## Ejercicio 1 – Cinta y sensor. Analógico (Factory IO)

En este ejercicio se intentará aplicar de forma autónoma lo que se ha aprendido a lo largo de esta práctica.

Este ejemplo constará de los siguientes elementos (*Figura 2.30*):

- Un transportador principal digital de dos sentidos (Tendrá dos salidas).
- Un transportador secundario (el del fondo) analógico el cuál trabaja desde los -10V hasta los 10 V.
- Cuatro transportadores en curva analógicos que trabajan desde los -10V hasta los 10 V.
- Dos sensores ubicados en cada una de las terminaciones del transportador principal.
- Un cuadro de mandos compuesto por un botón de marcha y un potenciómetro desde el que configuraremos la velocidad de las cintas.



Figura 2.29. Vídeo funcionamiento ejercicio 1.



Figura 2.30. Estación en Factory IO del ejercicio 1.

[Vídeo](#)

[Factory IO](#)

El funcionamiento es el siguiente (*Figura 2.29*), se tendrá una caja parada en el centro del transportador principal (*Figura 2.31*). Se ajustará la velocidad y dirección del resto de cintas con el potenciómetro ubicado en cuadro de mandos y se activará el botón de marcha.

En ese momento si el potenciómetro marca un valor mayor de 1V o menor que -1V la cinta principal se moverá hacia la dirección marcada. Cuando el sensor de ese sentido detecte la caja activará el resto de transportadores con la velocidad marcada en el potenciómetro.

Una vez la caja ha dado una vuelta y es detectada por el sensor de entrada al transportador principal contaremos 3 segundos y pararemos todas las cintas como al comienzo.

La idea es poder repetir este proceso todas las veces que se quiera.

Práctica 1 – Entorno TIA Portal



Figura 2.31. Estación ejercicio 1

Estas son las conexiones del Factory IO (Figura 2.32) y por tanto las variables que se tendrán en TIA Portal.

Botón Marcha	%I0.0	%Q0.0	Luz Botón Marcha
Sensor derecha	%I0.1	%Q0.1	Principal Positiva
Sensor izquierda	%I0.2	%Q0.2	Principal Negativa
Potenciómetro	%ID30 (REAL)	(REAL) %QD30	Curva 1
		(REAL) %QD34	Curva 2
		(REAL) %QD38	Curva 3
		(REAL) %QD42	Curva 4
		(REAL) %QD46	Secundaria

Figura 2.32. Conexión Factory IO ejercicio 1.

En la programación de los actuadores analógicos se necesitará conocer la forma de programarlos en TIA Portal esto se hace con la función MOVE. También es importante conocer como programar un temporizador, ambas cosas se muestran a continuación:

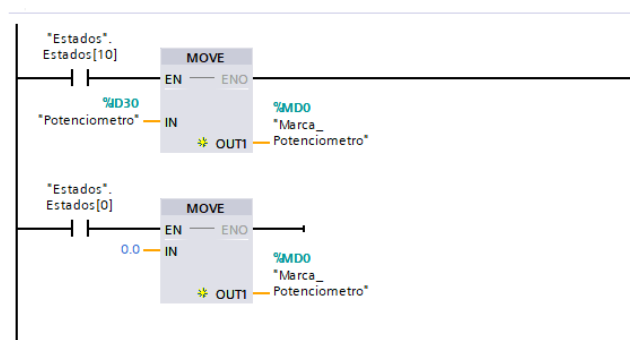


Figura 2.33. Programación variables analógicas.

Práctica 1 – Entorno TIA Portal

En el primer caso (Figura 2.33) en el estado 10 la “Marca\_Potenciometro” tendrá el mismo valor que tenga el potenciómetro. Y en el segundo caso en el estado 0 la “Marca\_Potenciometro” tendrá un valor de 0.0.

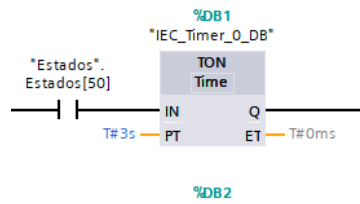


Figura 2.34. Programación temporizadores.

Para el caso del temporizador (Figura 2.34) se ha usado un temporizador tipo TON que es un temporizador con retardo al conectar esto quiere decir que en este ejemplo cuando se active el estado 50 pasado el tiempo de 3 segundos se activará la salida Q.

Una vez adquiridos los conocimientos necesarios podemos empezar a diseñar el GRAFCET este es GRAFCET (Figura 2.35) que se propone para este ejercicio.

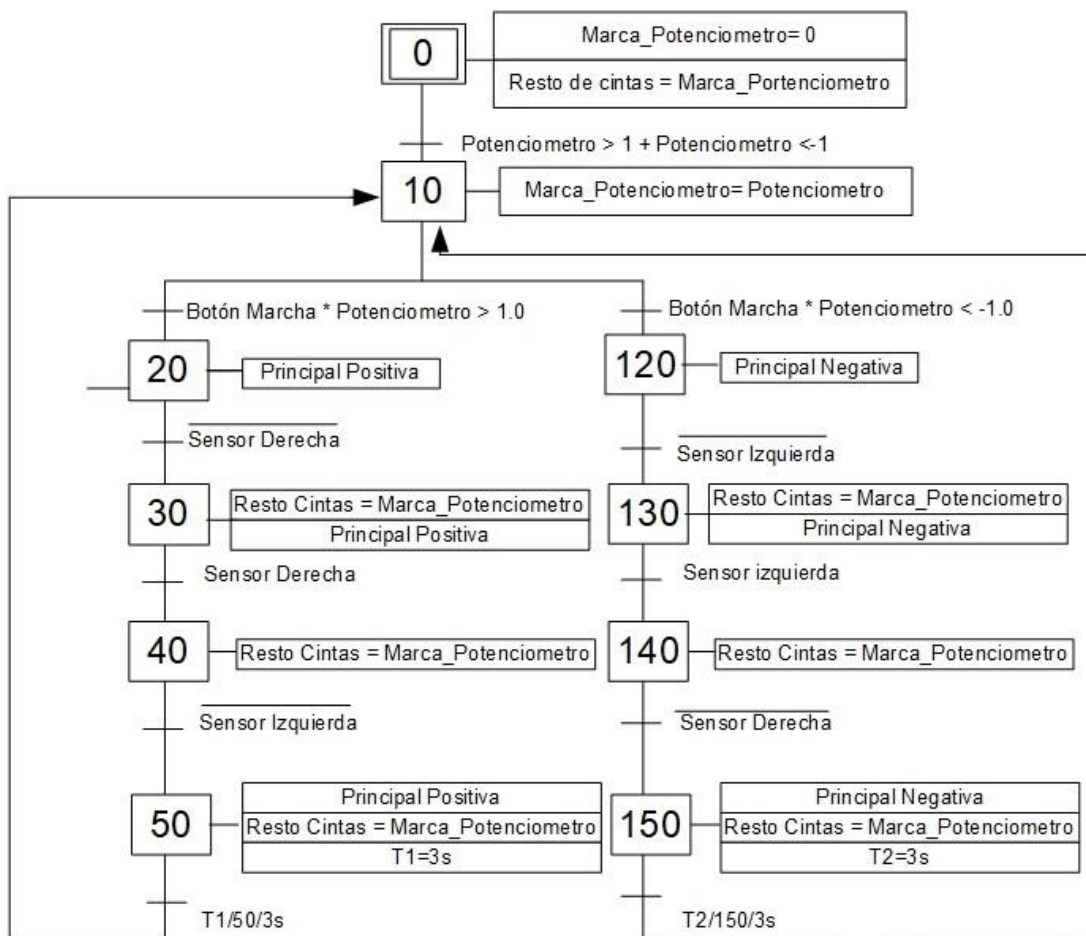


Figura 2.35. GRAFCET ejercicio 1.

#### **2.1.4. Trabajo previo**

Para esta primera práctica es importante que el alumno tenga instalado en su ordenador los dos programas que se utilizarán, tanto TIA Portal como Factory IO.

También es necesario para el alumno actualizar los conocimientos sobre como generar GRAFCETS ya que son una de las claves para automatizar correctamente un proceso.

#### **2.1.5. En el laboratorio**

El alumno creará proyectos en TIA Portal y Factory IO ayudándose de la práctica guiada, una vez sepa crearlos y tenga cierto grado de soltura con ambos softwares realizará los ejemplos.

Para el último ejemplo el alumno deberá generar su propio GRACET.



2.2. Práctica 2 – Cuadro de mandos. Modos de trabajo



AUTOMÀTICA INDUSTRIAL

The image displays a Factory IO simulation environment. On the left, a ladder logic diagram is shown, divided into three segments:

- Segment 1: EMERGENCIA**: Includes logic for emergency stop (X0) and a reset button (X1).
- Segment 2: ESPERA PEZA**: Includes logic for a zone sensor (X2) and a weight sensor (X3).
- Segment 3: COMPLECCION ACABADO**: Includes logic for a control sensor (X4) and a finished sensor (X5).

On the right, a control panel diagram shows a sequence of buttons: 0 (EMERGENCIA), 1 (CARRERA), 2 (ARRANQUE), 4 (REPLUMBO), and 5 (ARRANQUE). A stop button X0 is also present. The buttons are connected to PLC outputs and inputs.

PRÁCTICA 2. CUADRO DE MANDOS. MODOS DE TRABAJO

ALUMNO:

GRUPO:

### 2.2.1. Introducción

Una cosa que tienen en común todas las automatizaciones son los modos de trabajo. No siempre van a ser los mismos ni se tendrá el mismo número de modos, pero normalmente siempre se contará con alguno como automático o paro. Esto dependerá de la complejidad de tu automatización.

#### Los modos de trabajo en TIA Portal

Para la práctica de hoy se hablará sobre 4 modos importantes:

- Modo Paro
- Modo Manual
- Modo Homing
- Modo Automático

### 2.2.2. Objetivos

Para esta práctica se pide diseñar la lógica de un cuadro desde el que controlarlos modos de trabajo descritos anteriormente, para ello se utilizará el entorno de simulación Factory IO y TIA Portal.

Se recomienda hacer el código principal del cuadro dentro de una FB ya que se usará este cuadro en futuras prácticas.

### 2.2.3. Funcionamiento



Figura 2.36. Estación modos de trabajo.

### **Modo Paro**

Este modo puede plantearse de diversas formas, se trata de un paro controlado así se puede enfocarlo desde la perspectiva de terminar el ciclo en el que está trabajando y después parar o desde la perspectiva de terminar el movimiento y parar. Dependiendo de en qué proyecto se esté trabajando se elegirá uno u otro.

Por ejemplo, si se trabaja con una instalación que contiene movimientos sencillos y es fácil recuperar las condiciones iniciales de trabajo, cualquiera de las opciones podría valer, pero si es una máquina muy compleja tal vez sea más adecuado que termine el ciclo completamente y lo único que haga es no iniciar un nuevo ciclo.

### **Modo Manual**

Es un modo completamente necesario, cualquier accionamiento debe poder accionarse de forma individual, ya sea para tareas de mantenimiento como para tener un control más detallado de los actuadores.

### **Modo Homing**

Es un modo optativo. Este sirve para inicializar el proceso desde una posición de arranque controlada (siempre que se pueda).

Esto facilita bastante la función del usuario ya que el posicionado inicial de la máquina puede ser complicado, y con este modo esto sucedería de forma automática.

### **Modo automático**

Es el modo normal de trabajo, cualquier automatización perdería el sentido sin este modo ya que es el encargado de que la instalación trabaje de forma cíclica y autónoma.



Figura 2.37. Vídeo del funcionamiento de la práctica 2.

[Vídeo](#)



Figura 2.38. Estación modos de trabajo en Factory IO.

[Factory IO](#)

#### **2.2.4. Trabajo previo**

Antes de la sesión de prácticas el alumno ha de visualizar el video donde se muestra como de los enlaces enviados para entender el funcionamiento del Portal TIA y de la estación virtual.

#### **2.2.5. En el laboratorio**

El alumno deberá crear un programa capaz de controlar el cuadro de mandos propuesto y que contenga los modos de trabajo expuestos anteriormente.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

### 2.3. Práctica 3 – Automatización distribuidor de cajas



## AUTOMÁTICA INDUSTRIAL

The image displays the Factory IO simulation environment. On the left, a window shows the PLC ladder logic program with three segments: Segment 1 (initialization), Segment 2 (piece zone), and Segment 3 (completion/finished). The 3D model on the right shows a box distributor with a sequence of steps: 0 (initial state), 1 (box placed), 2 (box pushed), 4 (box moved), and 5 (box removed). The sequence is controlled by a PLC and a motor.

### PRÁCTICA 3. AUTOMATIZACIÓN DISTRIBUIDOR DE CAJAS

ALUMNO:

GRUPO:

### 2.3.1. Introducción

En multitud de procesos industriales es necesario clasificar y distribuir las diferentes piezas u objetos que se ven envueltos en el mismo. Para ello se dispone de todo tipo de sensores que son capaces de distinguir las características de esos objetos, características como, por ejemplo, color, tamaño, peso, material...etc.

En esta práctica se desarrollará un distribuidor automático que permita clasificar las diferentes cajas de un proceso basándose en el tamaño de las mismas. Para el desarrollo de esta práctica se contará con una estación virtual creado en el software Factory IO.

### 2.3.2. Objetivos

El objetivo principal de esta práctica es que el alumno se inicie en la programación de PLCs utilizando el software de Siemens TIA Portal. Se deberá pensar en la posible solución al problema de la automatización y generar un GRAFCET con ella.

### 2.3.3. Funcionamiento

El funcionamiento de esta estación es simple (*Figura 2.39*), se emitirán cajas de diferentes tamaños desde la cinta principal, estas pasarán por un sensor el cual determinará la altura de las mismas.

Una vez medidas y detectada el tipo de caja que tenemos el distribuidor central se encargará de enviar la caja a la cinta correspondiente. La distribución de las cajas y la altura es la siguiente:

Tipo de caja	Altura	Dirección
Pequeña	128	Derecha
Mediana	192	Izquierda
Grande	224	Centro

Tabla 1. Tamaños de las cajas

Práctica 3 – Automatización distribuidor de cajas

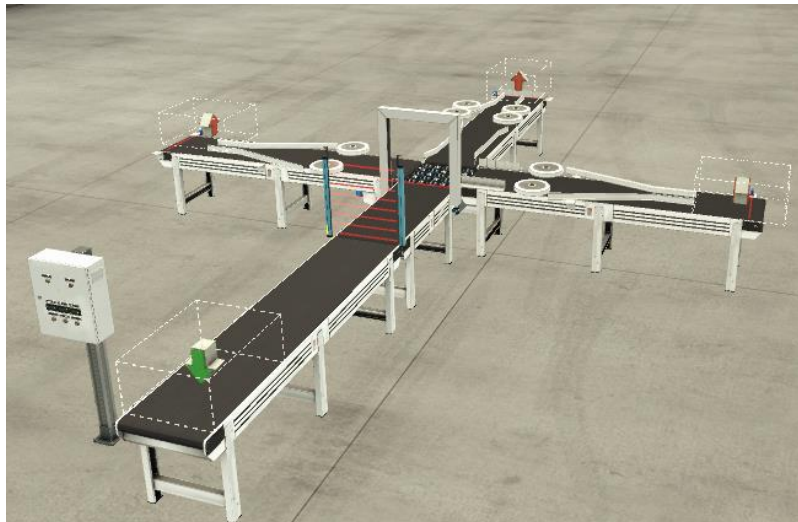


Figura 2.39. Estación distribuidora de cajas.

En el cuadro de mandos (Figura 2.40) se compone de 5 botones y tres displays. En los display se mostrará la cantidad de cajas de cada tamaño que ya han sido clasificadas y los botones serán uno de marcha, otro de paro y los tres de la parte inferior serán los encargados de resetear cada display.



Figura 2.40. Cuadro de mandos de la estación distribuidora de cajas.

Práctica 3 – Automatización distribuidor de cajas

Estás serán conexiones a llevar a cabo entre TIA Portal y Factory IO (Figura 2.41).

Botón Marcha	%I5.0	%Q5.0	Emisor
Botón Paro	%I5.1	%Q5.1	Cinta principal 1
Reset Grandes	%I5.2	%Q5.2	Cinta principal 2
Reset Reset Medianas	%I5.3	%Q5.3	Distribuidor
Reset Pequeñas	%I5.4	%Q5.4	Distribuidor Izquierda
Sensor Cinta Principal	%I5.5	%Q5.5	Distribuidor Derecha
Sensor Izquierda	%I5.6	%Q5.6	Cinta Izquierda
Sensor Centro	%I5.7	%Q5.7	Cinta Centro
Sensor Derecha	%I6.0	%Q6.0	Cinta Derecha
	%I6.1	%Q6.1	Salida Izquierda
	%I6.2	%Q6.2	Salida Centro
	%I6.3	%Q6.3	Salida Derecha
Sensor Altura	%ID30 (DINT)	%Q6.4	
	%ID34	%Q6.5	
	%ID38	%Q6.6	
	%ID42	(DINT) %QD30	Contador Grandes
		(DINT) %QD34	Contador Medianas
		(DINT) %QD38	Contador Pequeñas
		%QD42	

Figura 2.41. Conexiones distribuidora de cajas.



Figura 2.42. Vídeo del funcionamiento de la estación distribuidora de cajas.

[Vídeo](#)



Figura 2.43. Estación distribuidora de cajas en Factory IO.

[Factory IO](#)



### **2.3.4. Trabajo previo**

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar lo visto en la práctica 1
3. Graficet con la solución propuesta.

### **2.3.5. En el laboratorio**

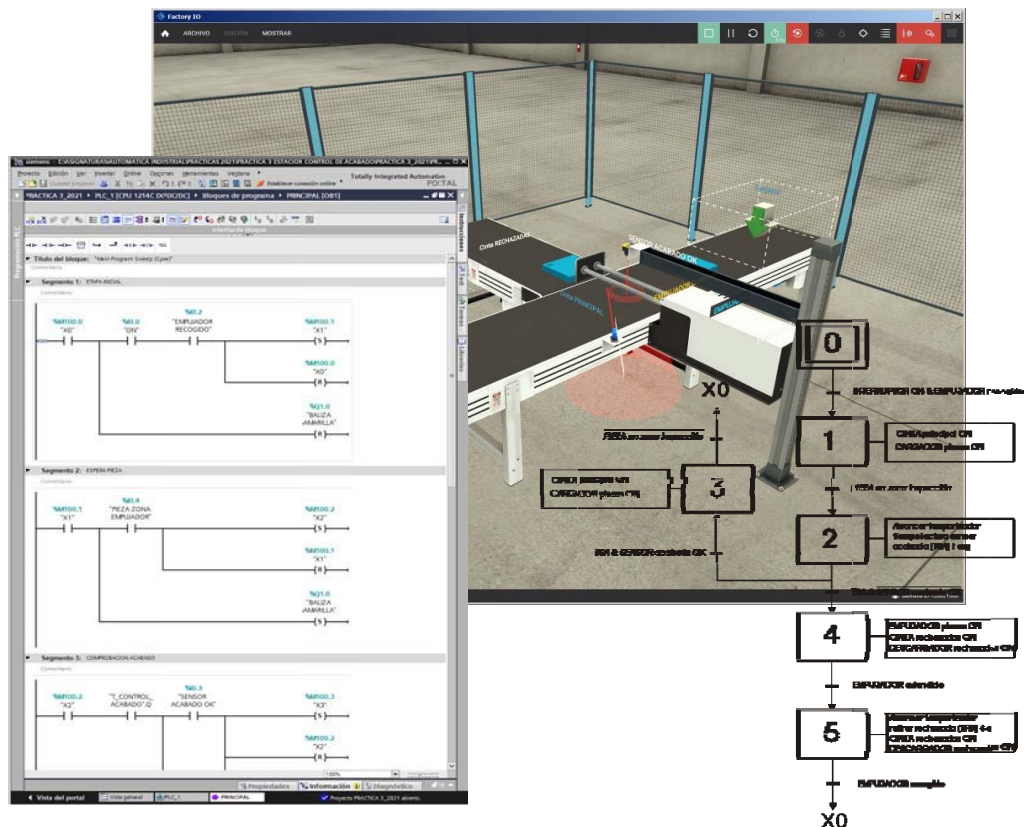
El alumno deberá crear un programa en KOP capaz de controlar el distribuidor de cajas propuesto en Factory IO.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

## 2.4. Práctica 4 – Módulo de trabajo Pick and Place



# AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 4. MÓDULO DE TRABAJO PICK AND PLACE

ALUMNO:

GRUPO:

### 2.4.1. Introducción

La estación Pick and Place (*Figura 2.44*) con tres ejes controlados por servomotores. A menudo se utiliza para trasladar cargas ligeras (por ejemplo, cajas de cartón) a otros transportadores o palés.

El Pick and Place tiene cuatro grados de libertad, tres corresponden al movimiento lineal de los ejes y otro a la rotación de la pinza. La pinza está habilitada por ventosas e incluye un sensor de proximidad. Puede ser controlado por valores digitales y analógicos, según la configuración seleccionada. Cuando se controla con E / S digital, el movimiento del eje se realiza de forma incremental (paso a paso) en cada flanco ascendente del valor de la etiqueta de control.



Figura 2.44. Robot cartesiano.

### 2.4.2. Objetivos

El objetivo de esta práctica es continuar avanzando en la programación de PLCs enfrentándose a diferentes situaciones. En el caso de esta práctica es la primera vez que el alumno se enfrenta a una máquina con diversas entradas y salidas como puede ser el robot cartesiano.

Además de ello con la propuesta de varios ejercicios con diferentes niveles de dificultad se pretende que una vez dominado el aspecto básico el alumno siga creciendo y pueda dar distintas soluciones para un mismo problema.

### 2.4.3. Funcionamiento

Esta práctica estará compuesta por tres módulos (*Figura 2.45*):

- Cinta transportadora de cajas: Esta parte será la encargada de llevar las cajas a la máquina Pick and Place.
- Cinta transportadora de palets: Este módulo llevará los palets al centro de la máquina Pick and Place donde quedarán parados hasta que estos se carguen y estén listos para continuar.
- Pick and Place: Es el nexo de unión entre los dos módulos anteriores ya que su función es coger las cajas que habrá en la cinta transportadora de cajas y apilarlas en el palet.

Práctica 4 – Módulo de trabajo Pick and Place

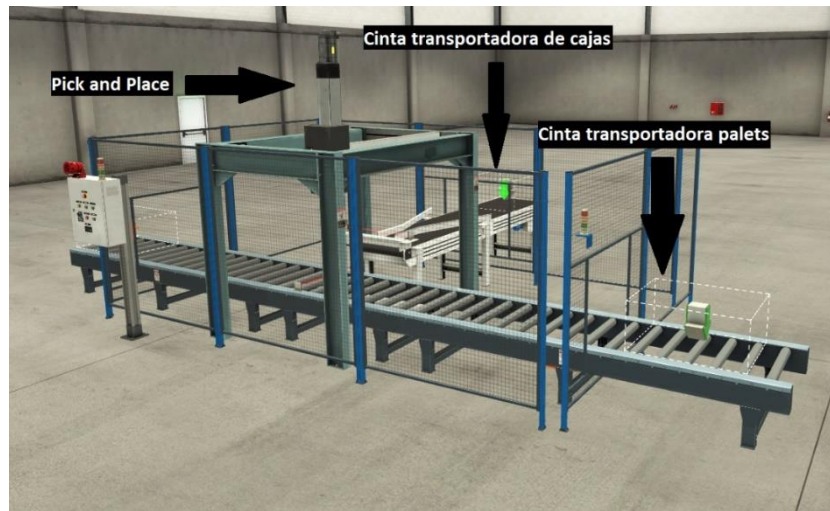


Figura 2.45. Estación pick and place.



Figura 2.46. Vídeo del funcionamiento de la estación Pick and place.

[Vídeo](#)



Figura 2.47. Estación Pick and place en Factory IO.

[Factory IO](#)

**Descripción de las variables en configuración analógica del Pick and Place**

En esta tabla se muestran todas las entradas y salidas pertenecientes al robot cartesiano:

Nombre de la variable	Controlador I/O	Tipo	Descripción
Pick & Place # X Set Point (V)	Salida	Float	[0, 10] V: establece la posición de destino a lo largo del eje X.
Pick & Place # Y Set Point (V)	Salida	Float	[0, 10] V: establece la posición de destino a lo largo del eje Y.
Pick & Place # Z Set Point (V)	Salida	Float	[0, 10] V: establece la posición de destino a lo largo del eje Z.
Pick & Place # C(+)	Salida	Bool	Gire la pinza.
Pick & Place # (Grab)	Salida	Bool	Activar ventosas.

Práctica 4 – Módulo de trabajo Pick and Place

Pick & Place # X Position (V)	Entrada	Float	[0, 10] V: posición actual a lo largo del eje X.
Pick & Place # Y Position (V)	Entrada	Float	[0, 10] V: posición actual a lo largo del eje Y.
Pick & Place # Z Position (V)	Entrada	Float	[0, 10] V: posición actual a lo largo del eje Z.
Pick & Place # (Box Detected)	Entrada	Bool	Detectando un artículo.
Pick & Place # (C Limit)	Entrada	Bool	Pinza en límite angular.

Tabla 2. Configuración analógica del robot cartesiano.

Para esta práctica se proponen tres niveles de dificultad:

- **Nivel básico:** Programar la estación con el objetivo de cargar los palets, se apilarán las cajas de la forma que el alumno desee. (Se recomienda de la forma que se hace en el vídeo)
- **Nivel medio:** Programar la estación y en este caso el operario podrá elegir en que formación apilar las cajas (*Figura 2.48*).

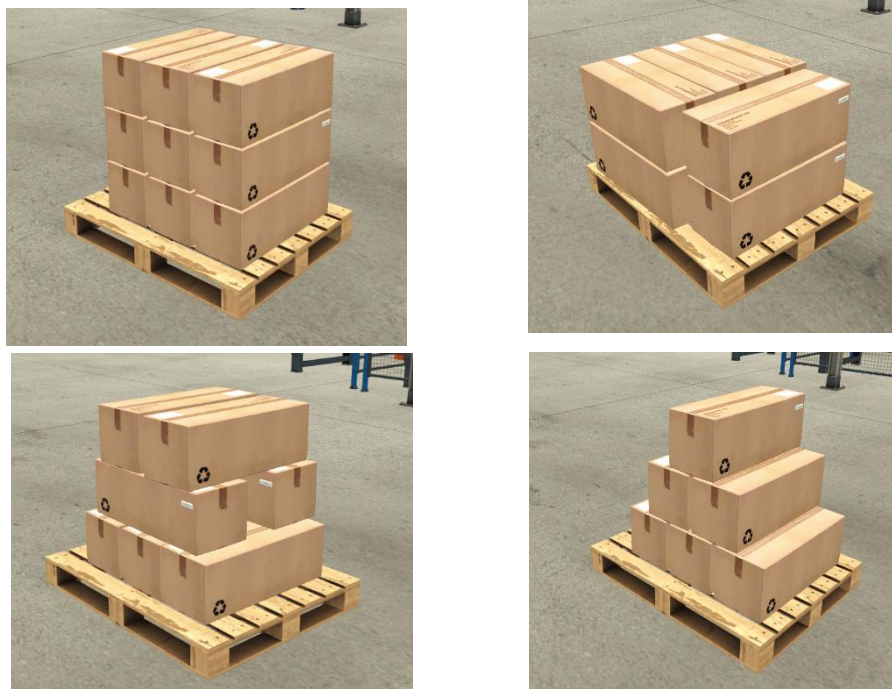


Figura 2.48. Nivel medio, apilado de cajas

- **Nivel avanzado:** Realizar un HMI desde el cual el operario pueda controlar la estación (incluir lo programado en los niveles anteriores).

#### **2.4.4. Trabajo previo**

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar lo visto en clase y en las prácticas anteriores.
3. Graficet con la solución propuesta.

#### **2.4.5. En el laboratorio**

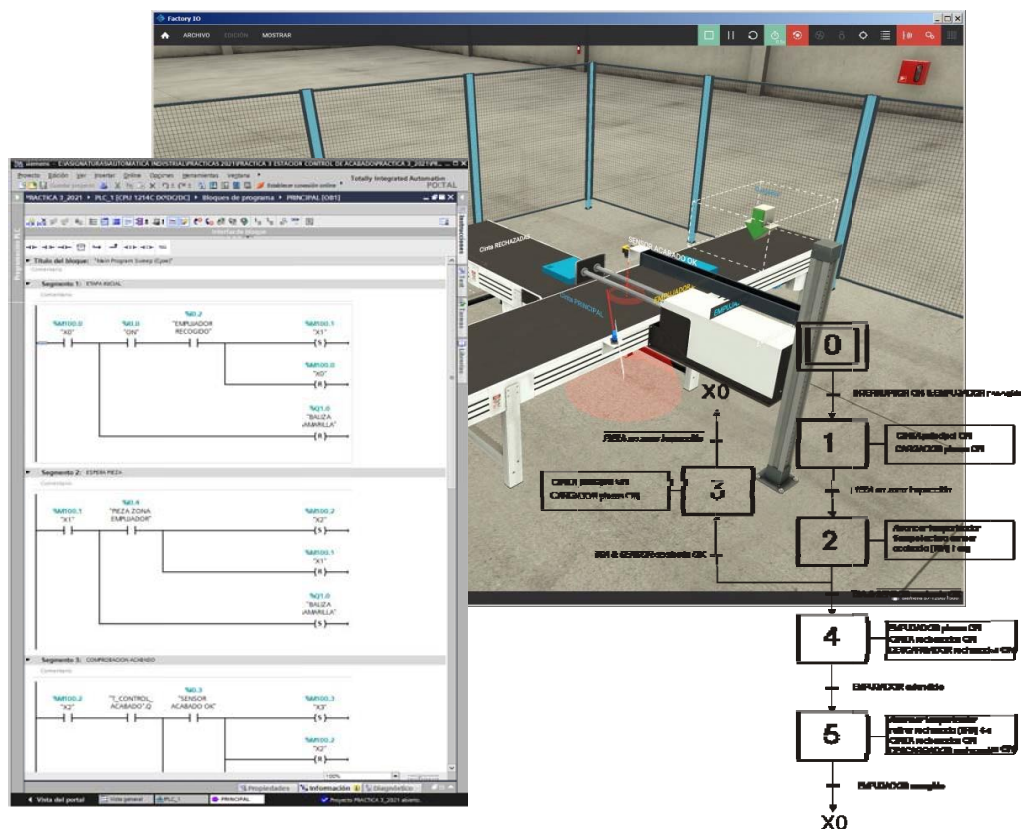
El alumno deberá crear un programa en KOP capaz de controlar el módulo pick and place propuesto en Factory IO.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

## 2.5. Práctica 5 – Programación de un ascensor



# AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 5. PROGRAMACIÓN DE UN ASCENSOR

ALUMNO:

GRUPO:

## 2.5.1. Introducción

### Las funciones

A la hora de programar resulta muy poco eficiente repetir las mismas secuencias dentro de un mismo programa ya que el programa será volverá muy extenso. Otro factor importante a la hora de programar es la memoria del PLC, debemos intentar que esta sea la mínima posible y esto se logra con la implementación de subrutinas.

### *¿Qué es una subrutina?*

Una subrutina es un proceso que se ejecuta sólo cuando es invocada. Esto hace que mientras no esté activa no consumirá recursos. Permanecerá sin consumir recursos del PLC mientras no sea invocada.

Por ejemplo, el bloque OB1 es leído y ejecutado en cada ciclo SCAN del PLC, a diferencia de una subrutina que permanecerá “dormida” sin consumir recursos, sólo hasta que esta se active.

### *Tipos de subrutinas en TIA Portal*

En TIA Portal se pueden encontrar dos tipos de subrutinas para los diferentes tipos de PLCs de Siemens, ambas subrutinas son imprescindibles en la programación estructurada de un proyecto. Las dos subrutinas son las denominadas FC y FB (*Figura 2.49*).

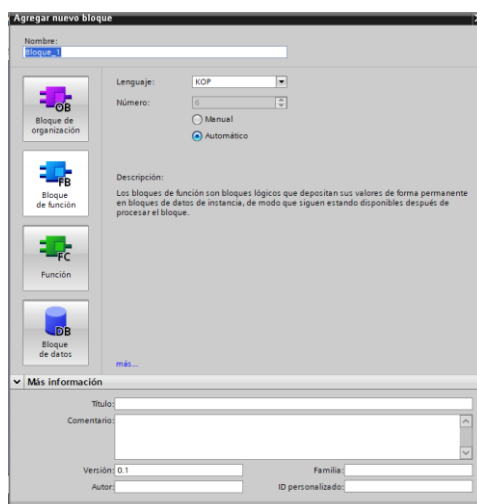


Figura 2.49. Tipos de bloque TIA Portal.

Antes de comenzar a explicar que es un FB y un FC hay un concepto que es importante conocer y es los bloques DB.

### *¿Qué es un bloque DB en un PLC?*



Un DB es un **Bloque de Datos** dentro del cual solo se puede almacenar datos. Los datos que se almacenan en su interior son de tipo Lectura/Escritura y se puede acceder a estos desde cualquier parte del programa, ya sea por algún bloque o por alguna operación.

Dentro de los DB existen 2 tipos:

- **DB Instancia:** Este tipo de DB está relacionado con un bloque FB y solo desde este bloque se pueden modificar los datos.
- **DB Global:** Los datos que contiene pueden ser tantos escritos como leídos desde cualquier bloque.

### ***¿Qué son los bloques FC y FB?***

Los bloques **FC (Función)** son subrutinas que contienen secuencias específicas. Este bloque se puede llamar desde otro bloque OB, FC o FB. La particularidad del bloque FC es que no contiene almacenamiento de datos cíclicos, es decir, si se realiza una operación dentro del bloque, el valor de resultado solo es útil en la operación, y solo es útil durante la ejecución, y este valor se perderá posteriormente. Cuando se llame al bloque FC la próxima vez, el resultado de la operación anterior ya no existirá.

Un bloque **FB (bloque de funciones)** es una subrutina que contiene una secuencia u operación. Se puede llamar desde otro bloque OB, FC o FB. A diferencia de FC, este bloque tiene asociado un bloque de datos de instancia por defecto, que almacena los valores en el FB, estos datos se denominan estáticos y solo el FB los tiene.

Cuando se llama al bloque FB, ejecutará su rutina y almacenará sus datos en el bloque de datos de instancia relacionado con este FB. Esto significa que, en la siguiente llamada del bloque, dado que los datos anteriores permanecen contenidos estáticamente en el DB asociado, se volverán a cargar los datos anteriores.

### ***¿Cuál es la diferencia entre un bloque FB y un bloque FC?***

La diferencia es que el bloque FC no tiene capacidad para almacenar datos estáticos, es decir, los datos de su funcionamiento son volátiles y se perderán una vez finalizada la ejecución del bloque. Por lo contrario, el bloque FB almacena los datos de su operación en el DB asociado, sus datos no son volátiles y permanecen almacenados incluso después de que se completa la ejecución del bloque.

### 2.5.2. Objetivos

El objetivo de esta práctica es introducir al alumno los diferentes bloques de trabajos que ofrece TIA Portal. Para ello se propone una práctica sencilla y con acciones repetitivas para que se muestre con claridad cómo funcionan estos bloques.

Además de esto también se pretende que el alumno siga progresando con la programación de HMIs.

### 2.5.3. Funcionamiento

La estación a simular en esta práctica se trata de una estación de carga de piezas a diferentes alturas.



Figura 2.50. Vídeo del funcionamiento de la estación del ascensor.

[Vídeo](#)



Figura 2.51. Estación del ascensor en Factory IO.

[Factory IO](#)

La estación (*Figura 2.52*) consta de tres partes:

- **Cinta transportadora de palets:** En este caso la cinta transportará una caja contenedora que la llevará al ascensor y ahí se detendrá hasta que la caja esté llena de piezas.
- **Ascensor:** Con él ascenderá o descenderá la caja contenedora para llenarla o vaciarla.
- **Cinta piezas:** Desde cada una de ellas vendrán las piezas que debe cargar.

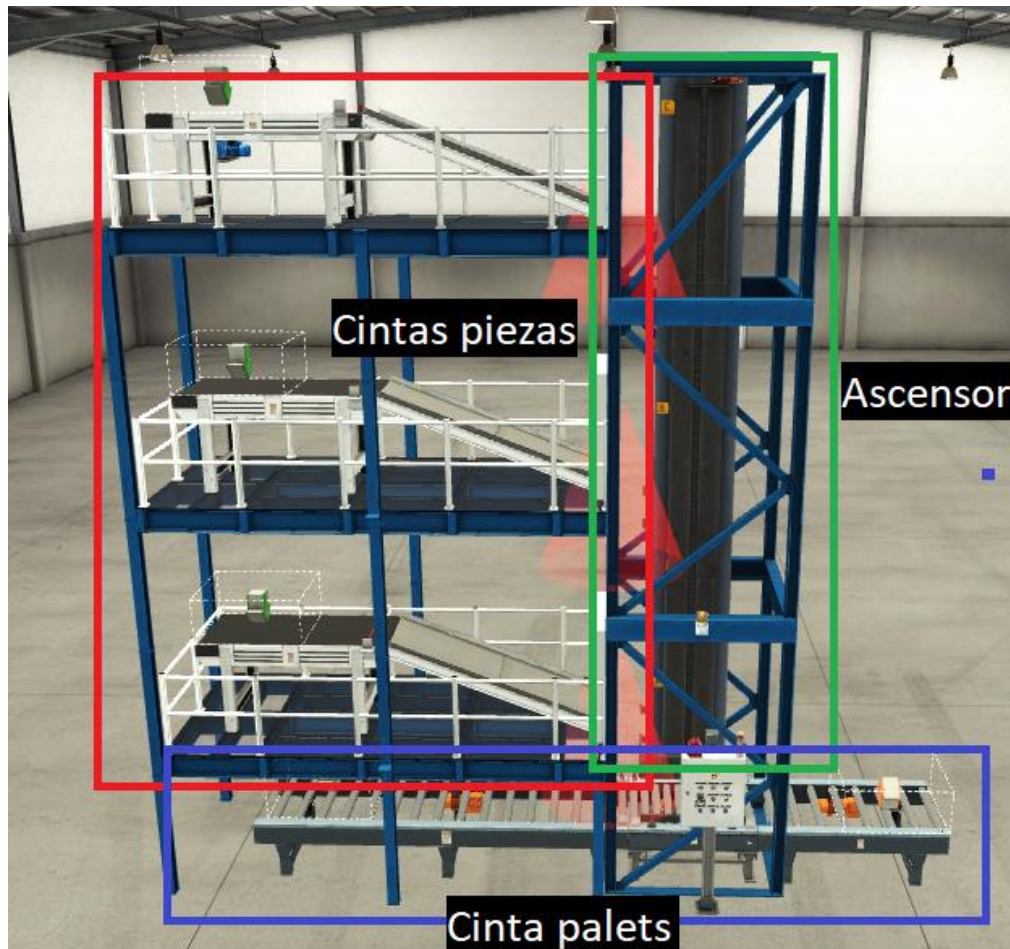


Figura 2.52. Descripción de la estación del ascensor.

### Módulo Ascensor

Aquí está la descripción de las entradas y salidas del ascensor, para esta práctica la cual se ha configurado el módulo en digital.

Nombre de la variable	Controlador I/O	Tipo	Descripción
Elevator # (Up)	Salida	Bool	Mueva la plataforma hacia arriba.
Elevator # (Down)	Salida	Bool	Mueva la plataforma hacia abajo.
Elevator # (Slow)	Salida	Bool	Movimiento lento (20% de la velocidad predeterminada).
Elevator # (+)	Salida	Bool	Ruede (dirección de la flecha).
Elevator # (-)	Salida	Bool	Rodar.
Elevator # (Left Limit)	Entrada	Bool	Elemento detectado.
Elevator # (Right Limit)	Entrada	Bool	Elemento detectado.

Tabla 3. Entradas y salidas del módulo del ascensor.

## Práctica 5 – Programación de un ascensor

### Lectura de las piezas mediante un sensor de visión

En esta sesión de prácticas al final de cada cinta transportadora de piezas se ha decidido colocar un sensor de visión para detectar si la pieza ha llegado correctamente a la caja contenedora.

#### Sensor visión

El sensor de visión (*Figura 2.53*) reconoce las materias primas, las tapas y las bases de los productos y sus respectivos colores.

- LED: rojo (detectando)
- Materiales detectables: materias primas, bases de productos y tapas de producto.
- Rango de detección: 0,3 - 2 m

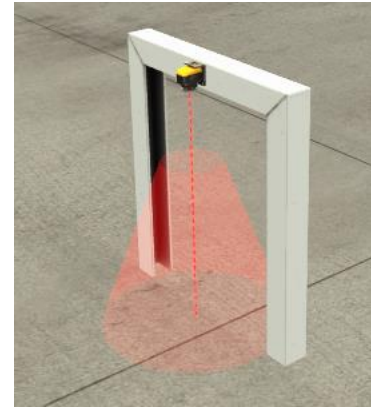


Figura 2.53. Sensor visión

Este sensor se puede configurar para detectar más de un tipo de pieza (*Figura 2.54*) seleccionando la configuración adecuada:

- Todo digital: devuelve cuatro entradas digitales que indican qué elemento se detectó
- Todo numérico: devuelve un valor que codifica el elemento detectado
- Todo ID: devuelve un valor único (aleatorio) que identifica el artículo detectado. Se puede utilizar de forma similar a los lectores de códigos de barras o RFID.
- Bases verdes/azul/metal: se debe configurar el color que se quiere y en el caso de encontrar una base del color elegido nos devolverá un booleano con valor positivo.
- Tapas verdes/azul/metal: se debe configurar el color que se quiera y en el caso de encontrar una tapa del color elegido devolverá un booleano con valor positivo.
- Materia prima verde/azul/metal: se debe configurar el color que se quiera y en el caso de encontrar una materia prima del color elegido devolverá un booleano con valor positivo.

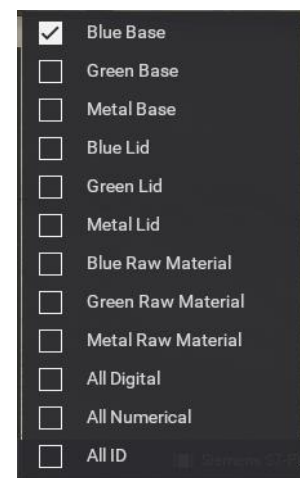


Figura 2.54. Tipo de piezas.

Para la estación las piezas con las que se trabajará son bases (*Figura 2.55*), se dispone de tres cintas así que las piezas azules irán a una altura, las verdes a otra y las metálicas a otras. Esto hará que la configuración del sensor sea mucho más fácil ya que se puede configurar en este modo “Bases verde/azul/metal” de esta forma devolverá un booleano si encuentra el color programado.

Práctica 5 – Programación de un ascensor



Figura 2.55. Piezas utilizadas en el proceso.

**Función para las cintas**

Como se observa se dispone de tres cintas transportadoras de piezas cuyo funcionamiento es exactamente el mismo, lo único que varía es el color de la pieza y la altura a la que se sitúa cada una de ellas.

Si se programase cada una de ellas en el “main” se observaría que se está repitiendo el código tres veces así que este es un buen caso para usar lo aprendido en esta práctica.

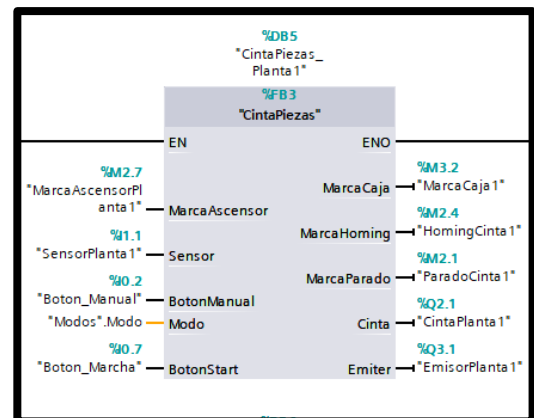


Figura 2.56. Función cinta de las piezas.

Aquí se tiene un ejemplo de como se vería la función una vez programada (Figura 2.56)

Se proponen para esta práctica tres tipos de ejercicios:

- **Nivel básico:** El ascensor recogerá una pieza de cada planta, bajará y avanzará la caja llena. (Lo visto en el vídeo)
- **Nivel medio:** Programación de un HMI donde se vea en tiempo real cómo evoluciona el proceso.
- **Nivel avanzado:** Desde el HMI que se ha programado el operario elegirá la configuración de llenado de la caja contenedora. (La capacidad de la caja es de tres piezas).

#### **2.5.4. Trabajo previo**

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar los conceptos de subrutina, funciones y bloques de datos
3. Graficet con la solución propuesta.

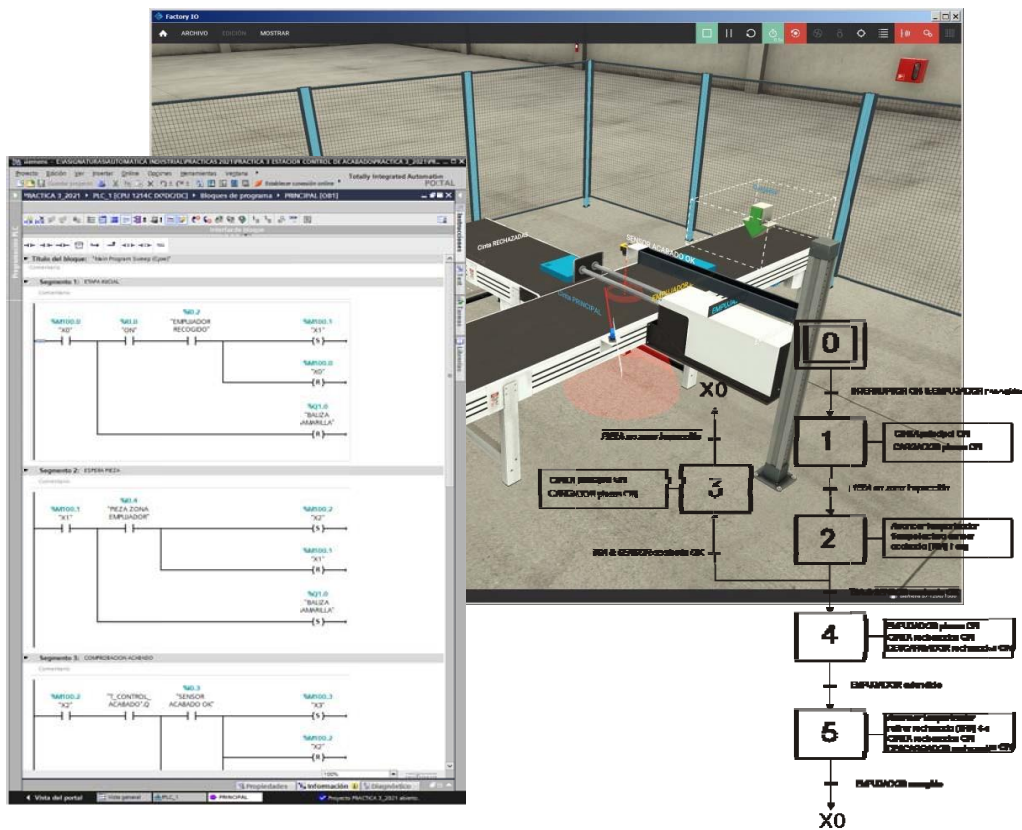
#### **2.5.5. En el laboratorio**

El alumno debe realizar los ejercicios propuestos anteriormente en KOP y cargarlo tanto en el PLC físico como en el PLCSIM y de esta forma verificar el correcto funcionamiento mediante la estación diseñada en Factory IO.

## 2.6. Práctica 6 – Automatización clasificadora de piezas



# AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 6. AUTOMATIZACIÓN CLASIFICADORA DE PIEZAS

ALUMNO:

GRUPO:

### 2.6.1. Introducción

La mayoría de proyectos a nivel industrial están compuestos por pequeños subprocesos que al juntarlos permiten desarrollar un proceso de mayor tamaño por ello es importante saber ligar esos subprocesos de forma correcta para un correcto funcionamiento de la estación.

En las estaciones realizadas anteriormente el proceso giraba sobre un elemento central como por ejemplo el ascensor, el robot cartesiano o el distribuidor. En esta práctica se intentará englobar todas ellas para así trabajar un escenario más cercano a un proceso industria real.

### 2.6.2. Objetivos

En esta estación se va a realizar la automatización de un proceso de clasificación de piezas. El objetivo principal de esta estación es que al tratarse una de las prácticas finales del guion el alumno pueda desarrollar todo lo aprendido hasta ahora con estas prácticas y aplicarlo de forma efectiva.

### 2.6.3. Funcionamiento

La estación que se programará en esta práctica se trata de una estación clasificadora de piezas.



Figura 2.57. Vídeo del funcionamiento de la estación clasificadora.

[Vídeo](#)



Figura 2.58. Estación clasificadora en Factory IO.

[Factory IO](#)

Como se puede observar se trata de una estación con varios procesos diferenciados. Esta estación se ha clasificado de la siguiente manera para que sea más fácil su programación:



Práctica 6 – Automatización clasificadora de piezas

- **Transportador 1:** En él se encuentra el emisor de piezas, las cuales aparecerán encima de un palet y pueden ser de color verde o azul. Las llevará hasta la entrada del ascensor y allí esperará a que este esté listo (*Figura 2.59*).



Figura 2.59. Transportador 1

- **Ascensor:** Una vez el palet con las piezas está colocado este coge el palet en el piso de arriba y lo baja al piso inferior (*Figura 2.60*).

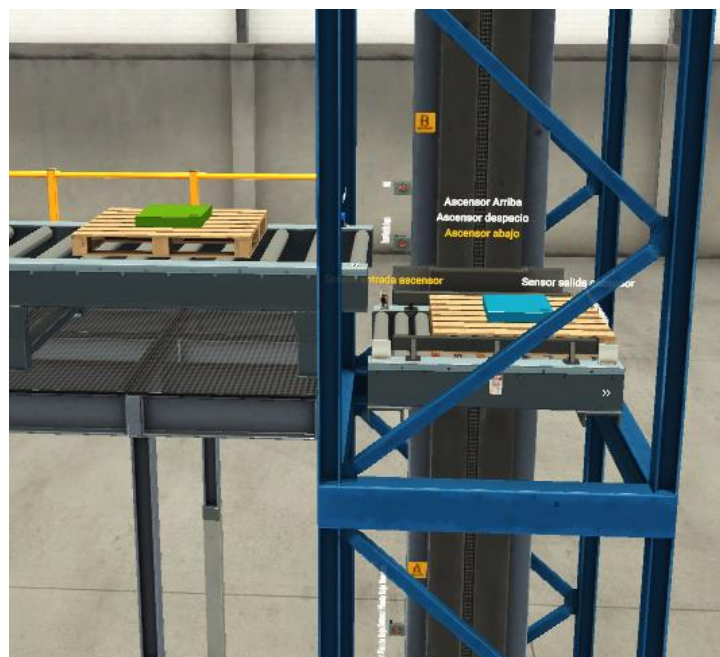


Figura 2.60. Ascensor.

## Práctica 6 – Automatización clasificadora de piezas

- **Cinta 1:** Cuando el ascensor lleve el palet a la planta baja este pasará a la cinta 1, sobre la cual está el sensor de visión. Este nos dará un valor de 1 si la pieza es azul y un 4 si la pieza es verde (Figura 2.61).

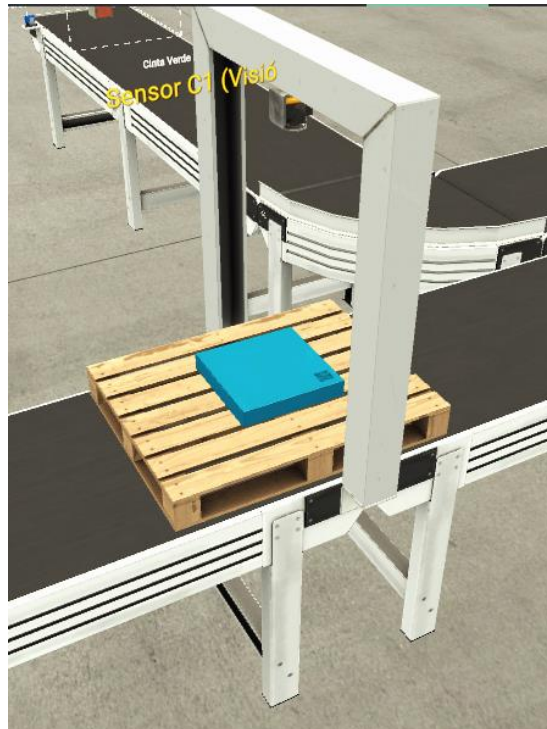


Figura 2.61. Cinta 1.

- **Cinta 2-3:** Una vez la pieza pasa por el sensor de visión esta avanza hasta la zona de agarre del robot, en la cual espera hasta que la pieza es cogida. Para finalizar el palet vacío avanza a la cinta 3 hasta la rampa de salida del palet (Figura 2.62).



Figura 2.62. Cinta 2-3

## Práctica 6 – Automatización clasificadora de piezas

- **Robot:** El robot coge la pieza y dependiendo del color de la misma la coloca en una cinta u en otra (*Figura 2.63*).

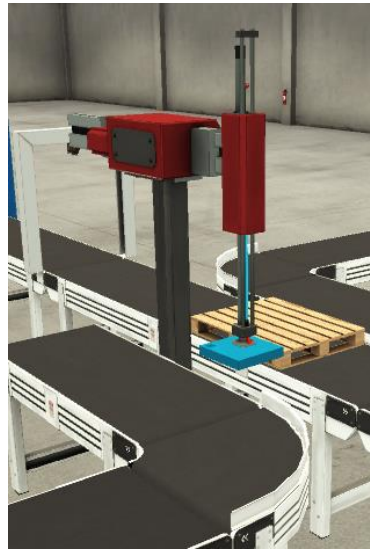


Figura 2.63. Robot.

- **Cintas de colores:** Una vez el robot coge debe colocarla o en la cinta verde o en la cinta de las azules. Estas cintas trasladarán la pieza hasta que desaparezca en la salida.

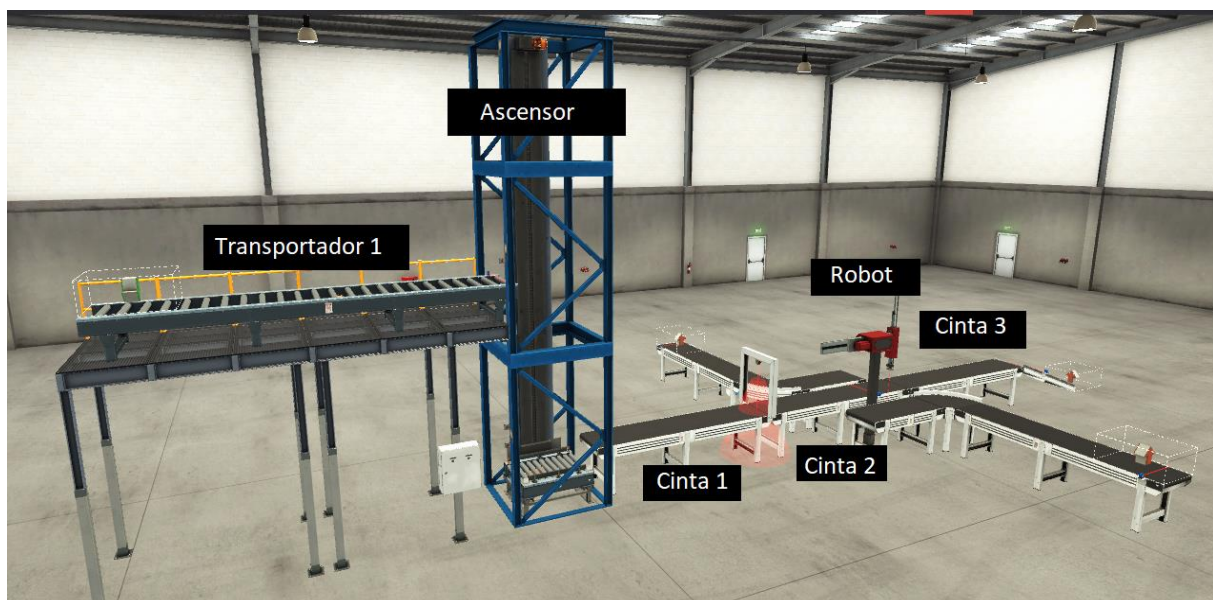


Figura 2.64. Estación clasificadora completa.

Práctica 6 – Automatización clasificadora de piezas

**Robot de dos ejes**

En este módulo se encuentra un robot de dos ejes pick and place, se ha configurado de forma analógica para una mayor precisión de los movimientos, estas son las entradas y salidas que se usarán.

Nombre de la variable	Controlador I/O	Tipo	Descripción
Robot Eje X	Salida	Real	Es el encargado de mover el robot en el eje X.
Robot Eje Z	Salida	Real	Es el encargado de mover el robot en el eje Z.
Robot Giro Antihorario	Salida	Bool	Al activarlo el robot hará un giro antihorario.
Robot Giro Horario	Salida	Bool	Al activarlo el robot hará un giro horario.
Robot Agarre	Salida	Bool	Al activarlo el robot agarrará.
Sensor Robot Eje X	Entrada	Real	Sensor que nos indica en que posición se encuentra el robot en el eje X.
Sensor Robot Eje Z	Entrada	Real	Sensor que nos indica en que posición se encuentra el robot en el eje Z.
Robot Girando	Entrada	Bool	Sensor que nos indica si el robot se encuentra girando.
Robot Agarre	Entrada	Bool	Sensor que nos indica si el robot se encuentra agarrado un objeto.

Tabla 4. Entradas y salidas del robot de dos ejes.

Finalmente, para un correcto funcionamiento de la estación se mostrará el esquema de configuración de los sensores y actuadores que intervienen en la estación (Figura 2.65).



Figura 2.65. Conexiones Factory IO.

#### **2.6.4. Trabajo previo**

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar los conceptos de subrutina, funciones y bloques de datos
3. Graficet con la solución propuesta.

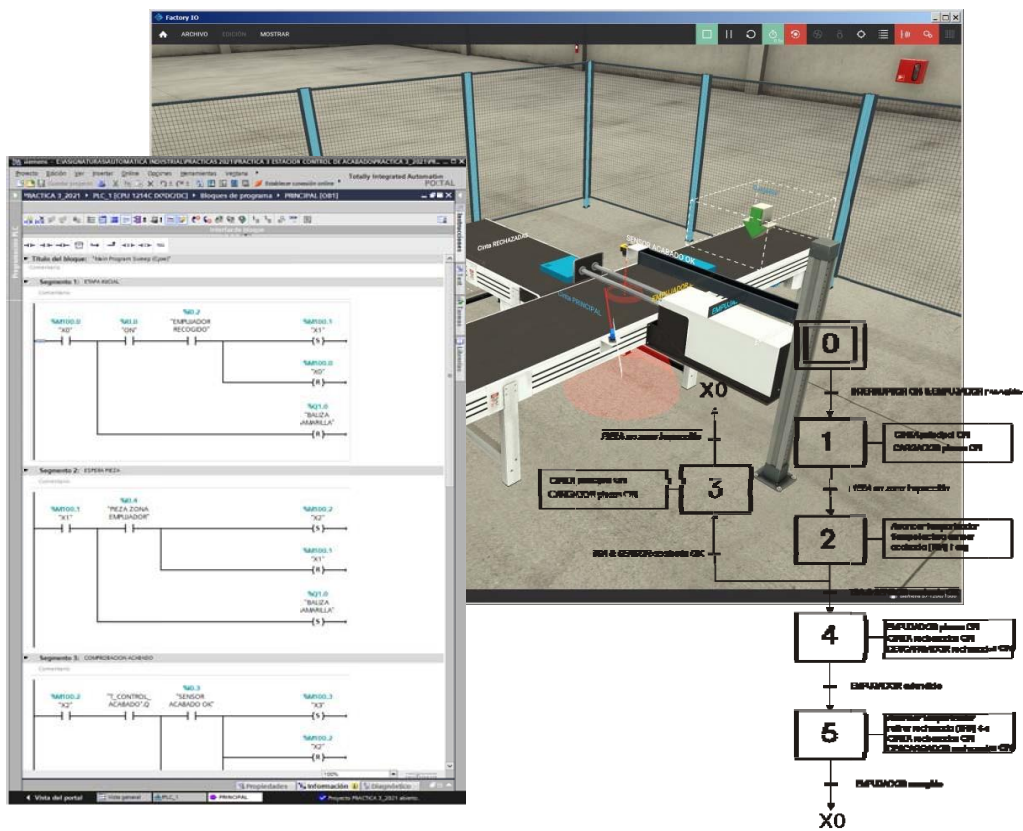
#### **2.6.5. En el laboratorio**

El alumno debe realizar el ejercicio propuesto anteriormente en KOP y cargarlo tanto en el PLC físico como en el PLCSIM y de esta forma verificar el correcto funcionamiento mediante la estación diseñada en Factory IO.

## 2.7. Práctica 7 – Llenado de un tanque. Control PID



# AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 7. LLENADO DE UN TANQUE. CONTROL PID

ALUMNO:

GRUPO:

### 2.7.1. Introducción

Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. Estas tres acciones son las que dan nombre al controlador PID.

Este tipo de controladores es muy usual utilizarlo en sistemas de bombeo ya que en estos regularmente interesa mantener la presión o flujo constante, por lo tanto, el control PID (*Figura 2.66*) mide la diferencia entre la presión en la tubería y la presión requerida y actúa variando la velocidad del motor para que pueda tener nuestra presión o flujo constante.

El **parámetro Proporcional (P)** mide la diferencia entre el valor actual y el set-point (en porcentaje) y aplica el cambio.

El **parámetro Integral (I)** se refiere al tiempo que se toma para llevar a cabo acción correctiva. Mientras el valor sea más pequeño, el ajuste es más rápido, pero puede causar inestabilidad en el sistema, oscilaciones, vibración de motor y de la bomba.

El **parámetro Derivativo (D)** emite una acción predictiva, es decir, prevé el error e inicia una acción oportuna. Responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande.

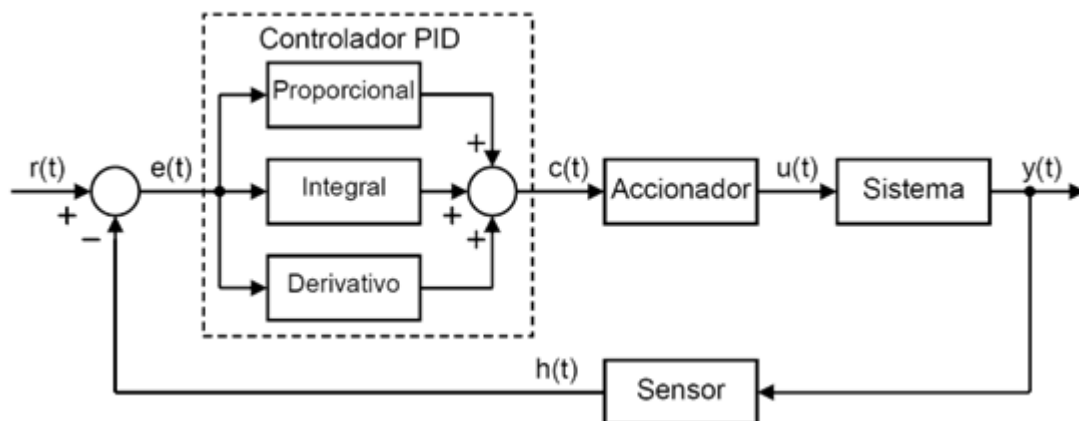


Figura 2.66. PID.

La correcta sintonización o programación de estos parámetros ayuda a controlar de manera efectiva la presión o flujo deseado. Si no se programa adecuadamente estos parámetros, el sistema puede quedar inestable y el motor y la bomba pueden comenzar a vibrar y dañarse.

### 2.7.2. Objetivos

El objetivo de esta práctica es que el alumno aprenda como programar un controlador PID en TIA Portal y que una vez programado pueda verlo actuar de forma práctica gracias al entorno de simulación Factory IO.

### 2.7.3. Funcionamiento

Para esta práctica se utilizará un tanque de agua (*Figura 2.67*) el cual cuenta dos válvulas una de entrada y otra de salida y un sensor de llenado. El objetivo será marcar un SetPoint de llenado (Punto de referencia) del sistema y que de forma automática el nivel del tanque se mantenga a ese SetPoint.



Figura 2.67. Estación de llenado de un tanque.



Práctica 7 – Llenado de un tanque. Control PID

Esto se realizará a través de un controlador PID, para añadir el bloque PID al programa se debe ir a la parte derecha de TIA Portal y pulsar en instrucciones, tecnología y ahí aparecerá el bloque PID\_Compact (Figura 2.68).

Este bloque PID se añade en un bloque de organización “Cyclic interrupt” para que de esta forma el PID esté funcionando de forma constante. Una vez añadido el bloque debe configurarse.

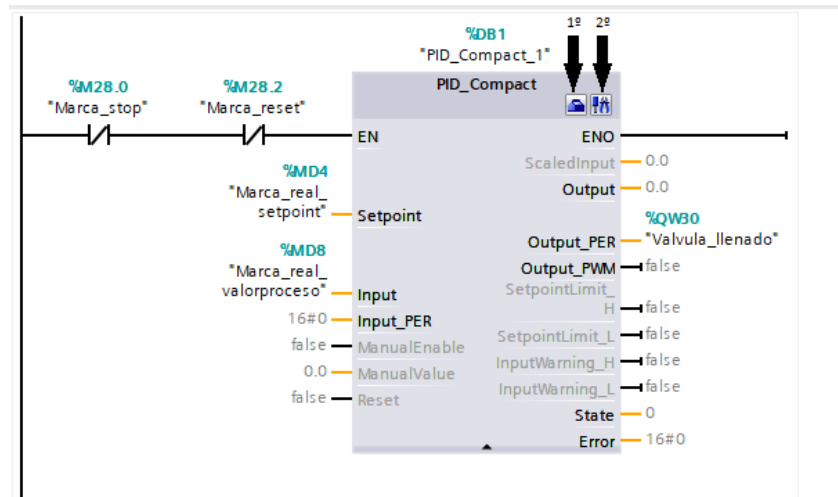


Figura 2.68. Configuración de un PID en TIA Portal.

En la esquina derecha del bloque se observan dos botones. El primero se trata de un botón que lleva a la ventana de configuración del PID a la cual se debe ir antes de ponerlo en marcha y ajustarlo al proceso. El segundo se trata de una ventana para la puesta en servicio, aquí se debe ir la primera vez que se active la estación y ejecutar la puesta en servicio para que el PID se adapte de forma automática al proceso que este controla.

Para esta práctica se presentan tres niveles de dificultad:

- **Nivel básico:** Tanque + Cuadro de mandos (Start – Stop – Apertura válvula salida).
- **Nivel intermedio:** Tanque + Cuadro de mandos (Start – Stop – Apertura válvula salida) + HMI (Visualización y control del proceso).
- **Nivel avanzado:** Tanque + Cuadro de mandos (Con modos de trabajo) + HMI (Visualización y control del proceso) (Figura 2.69).



Figura 2.69. Nivel avanzado.



Figura 2.70. Vídeo funcionamiento estación llenado de un tanque.

[Vídeo](#)



Figura 2.71. Estación llenada de un tanque en Factory IO.

[Factory IO](#)

#### 2.7.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar el modelado de sistemas y el diseño de control PID.

### **2.7.5. En el laboratorio**

El alumno deberá crear un programa en KOP capaz de controlar el llenado del tanque propuesto en Factory IO.

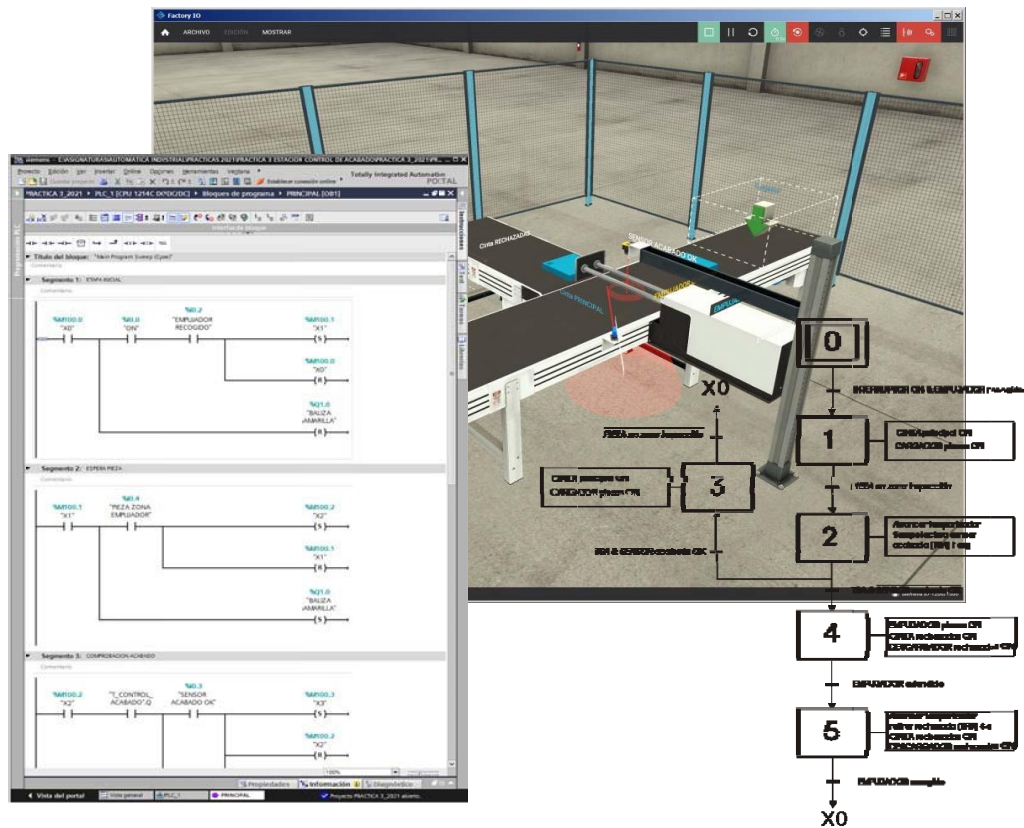
Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

### 3. Cuadernillo de prácticas docente

#### 3.1. Práctica 1 – Entorno TIA Portal



## AUTOMÁTICA INDUSTRIAL



### PRÁCTICA 1. ENTORNO TÍA PORTAL

ALUMNO:

GRUPO:

### 3.1.1. Introducción

En la automatización de cualquier tipo de proceso siempre es necesario introducir un elemento capaz de controlar las operaciones que se llevan a cabo en él, los Controladores Lógicos Programables o más conocidos como PLC son los encargados de realizar dicha función. En esta práctica se aprenderá sobre que es un PLC, ejemplos de cómo utilizarlos y como programarlos. Para ello se usarán principalmente dos softwares diferentes (*Figura 3.1*):

- **Totally Integrated Automation Portal (TIA Portal)**
- **Factory IO**



**Figura 3.1.** Logo de TIA Portal y Factory IO.

El alumno se iniciará en este software realizando pequeños ejemplos guiados y de esta forma conocer más a fondo todas las herramientas que este contiene.

Como herramienta de ayuda para realizar estos ejemplos se hará uso del software de simulación FACTORY IO.

Factory I/O es un software para automatización en tiempo real donde se puede construir y simular sistemas industriales y utilizarlos con las tecnologías de automatización más comunes. Esta simulación es totalmente interactiva e incluye gráficos de alta calidad y sonido, proporcionando un entorno realista industrial.

Factory I/O utiliza una tecnología innovadora que permite una creación fácil y rápida de los sistemas industriales en 3D con solo arrastrar y soltar. Cualquiera de los sistemas construidos se puede controlar en tiempo real mediante la conexión de Factory I/O y equipos externos como PLCs, microcontroladores, FPGA, etc.

Factory I/O es una valiosa herramienta de enseñanza para la formación de futuros técnicos e ingenieros en varios programas y cursos tales como automatización industrial, Mecatrónica, Ingeniería Eléctrica, Ingeniería Mecánica, Instrumentación y muchos más.

### 3.1.2. Objetivos

En esta práctica el objetivo fundamental es que el alumno se inicie en la programación de PLCs utilizando el entorno TIA Portal de Siemens.

El alumno será capaz de generar GRAFCETS para la solución de problemas de automatización y posteriormente implementarlos en lenguaje KOP dentro de la aplicación TIA Portal.

Para todo ello será imprescindible iniciarse en el software de simulación Factory IO ya que será en él donde se comprobará que lo que se ha programado funciona correctamente.

Objetivos principales:

- Saber configurar los parámetros de TIA Portal.
- Ser capaces de crear un proyecto.
- Configurar la conexión entre Factory IO y TIA Portal.
- Introducirse en pequeños proyectos.
- Saber aplicar los conocimientos aprendidos en clase para programar diferentes proyectos.

### 3.1.3. Funcionamiento

#### 3.1.3.1. Entorno TIA Portal

Se empezará creando un primer proyecto en TIA Portal para ello en primer lugar se inicia la aplicación y se hace clic sobre “Crear Proyecto” (*Figura 3.2*). En este se debe elegir el nombre, la ruta y el autor. Una vez hecho se pulsa sobre crear proyecto

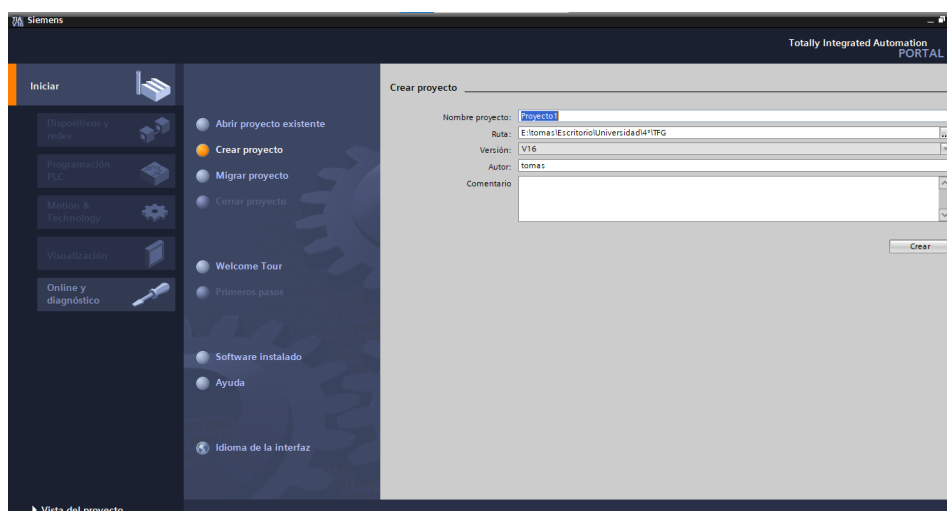


Figura 3.2. Abrir proyecto TIA Portal.

Práctica 1 – Entorno TIA Portal

Después de esto hay que pulsar en el botón que hay en la esquina izquierda inferior “Vista de proyecto”. Esto abrirá la vista general del proyecto. (Figura 3.4).

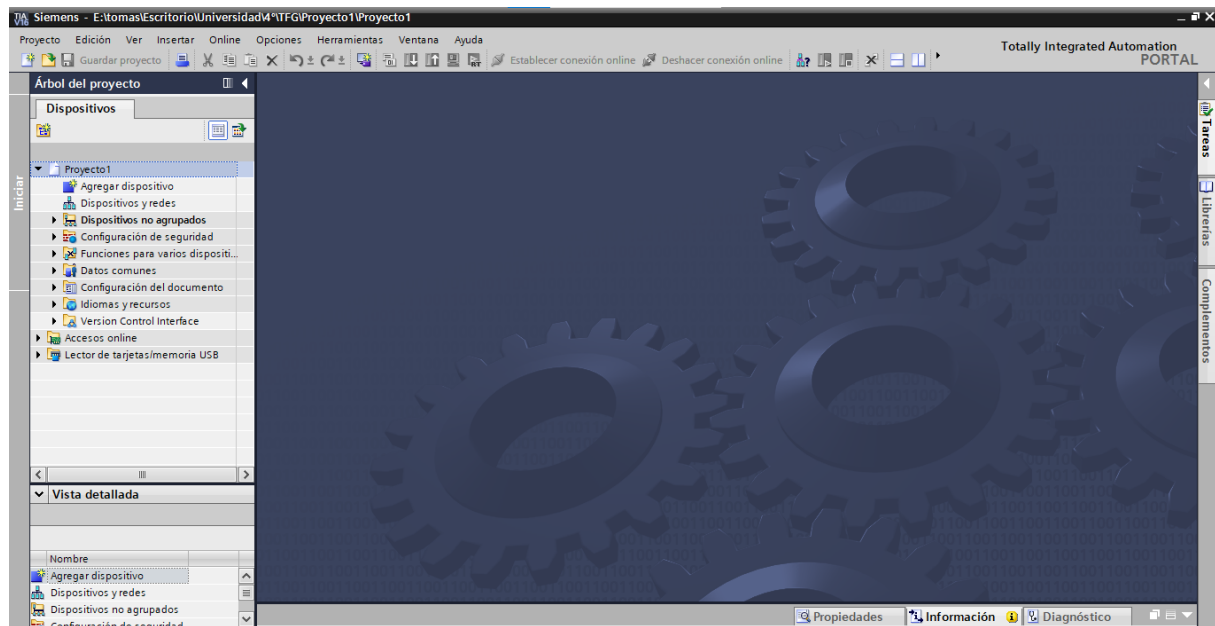


Figura 3.3. Vista de proyecto TIA Portal.

Para añadir un PLC se debe pulsar en “Agregar dispositivo” el cual se encuentra a la izquierda en el árbol del proyecto.

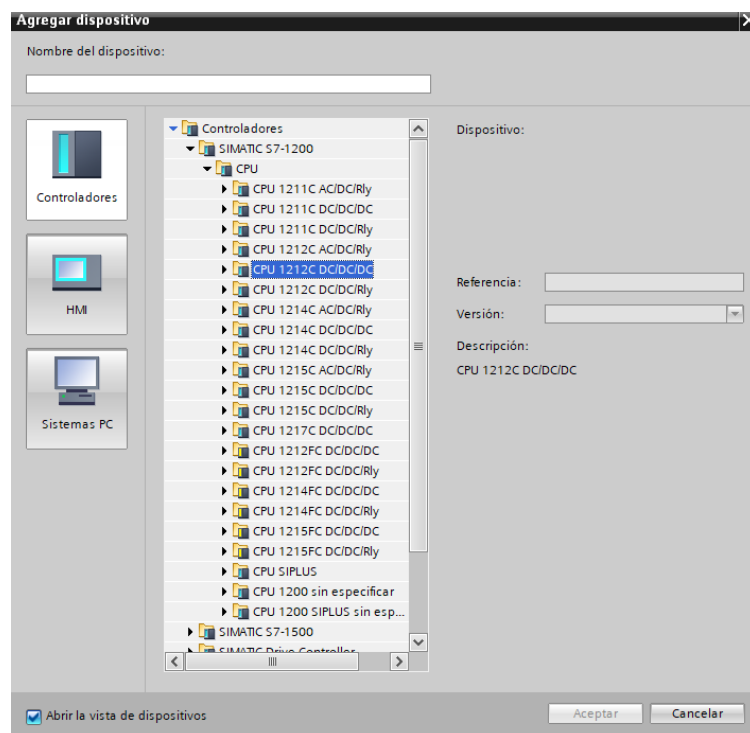


Figura 3.4. Dispositivos TIA Portal.

Práctica 1 – Entorno TIA Portal

Desde aquí (Figura 3.4) habrá que buscar el PLC que tengamos. Una vez añadido se debe configurar el protocolo IP del PLC al cual se accede desde las propiedades del PLC.

Pasemos ahora a explicar los diferentes bloques que se pueden encontrar en TIA Portal una vez configurado el PLC.

A la izquierda el Árbol de trabajo (Figura 3.5), en él se localiza todo el contenido de cada uno de los dispositivos que contiene el proyecto.

**Bloques de programa:** En él se localizan los diferentes bloques de programación que ofrece TIA Portal, para acceder a ellos se pulsa en Agregar nuevo bloque y se abrirá un desplegable donde seleccionar el dispositivo a añadir.

**Variables PLC:** En esta parte es donde se organizan las variables del programa. Para ello hay que añadir una nueva tabla de variables, y una vez dentro de ella poner el nombre de la variable, el tipo de dato y la dirección de la misma. Aparte de añadirla manualmente se pueden importar.

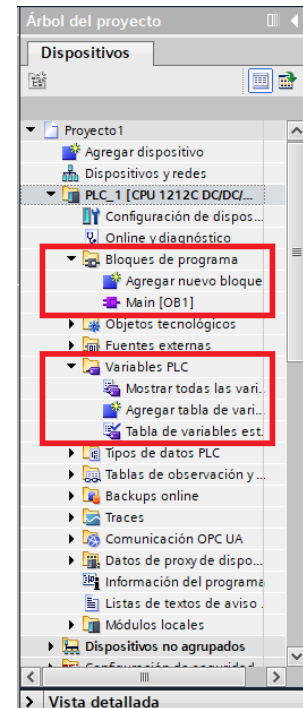


Figura 3.5. Árbol del proyecto.

Pasando ahora a la parte de la programación esta se lleva a cabo en los OB que son un tipo de bloques de datos.

Pasando ahora a la parte de la programación esta se lleva a cabo en los OB que son un tipo de bloques de datos.

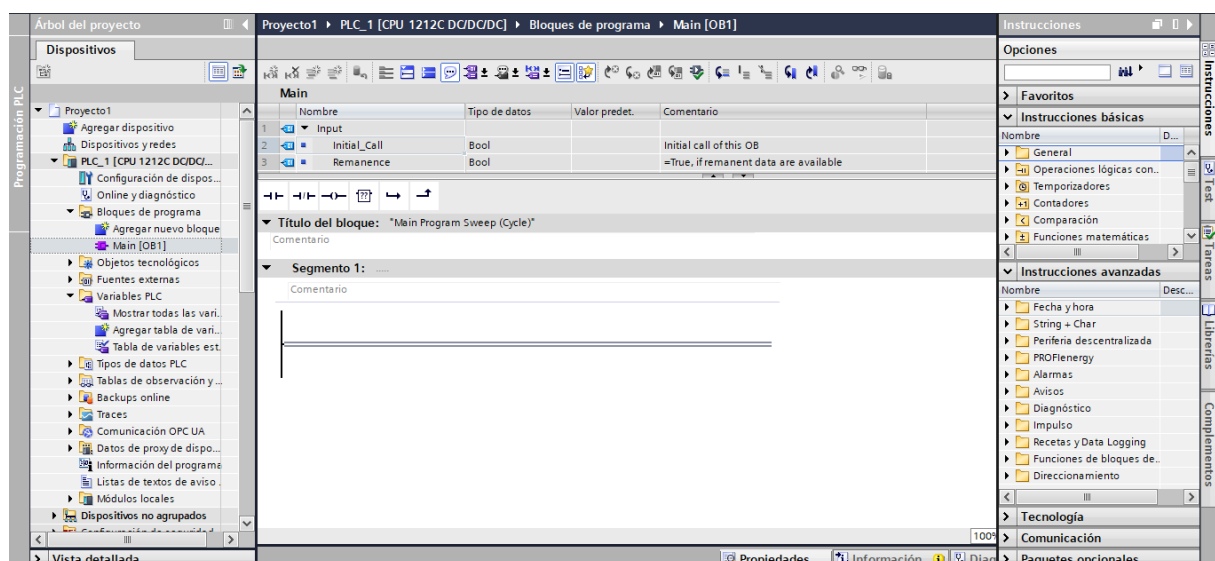


Figura 3.6. Zona de programación.



En la parte central (*Figura 3.6*) es donde se escribe el código en lenguaje KOP y en la parte derecha en el apartado “Instrucciones” se localizan todos los objetos que disponibles para la programación.

Estos son pequeños ejemplos de programación en KOP:

Ejemplo 1: En este caso si el sensor se activa el motor estará funcionando mientras el sensor esté activo. (*Figura 3.5*)

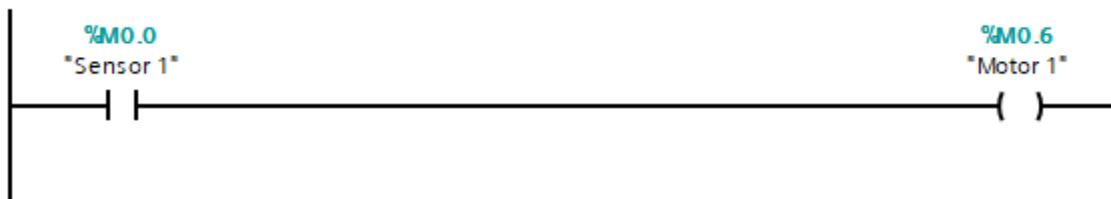


Figura 3.5. Ejemplo 1 KOP.

Ejemplo 2: Para este caso deben darse dos condiciones que el sensor 1 esté activo y el sensor 2 esté desactivado, solo en ese caso se activará el motor. (*Figura 3.6*)

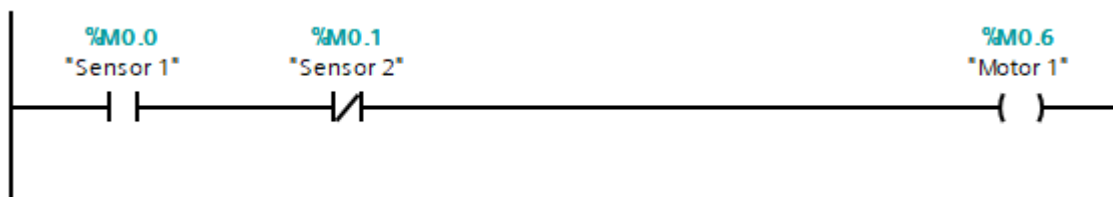


Figura 3.6. Ejemplo 2 KOP.

Ejemplo 3: En esta ocasión hay 3 posibilidades de que el motor se active, estas serían que el sensor 2 se active y el 3 no, que se active el sensor 3 y el sensor 2 no o finalmente que ambos se activen. (*Figura 3.7*)

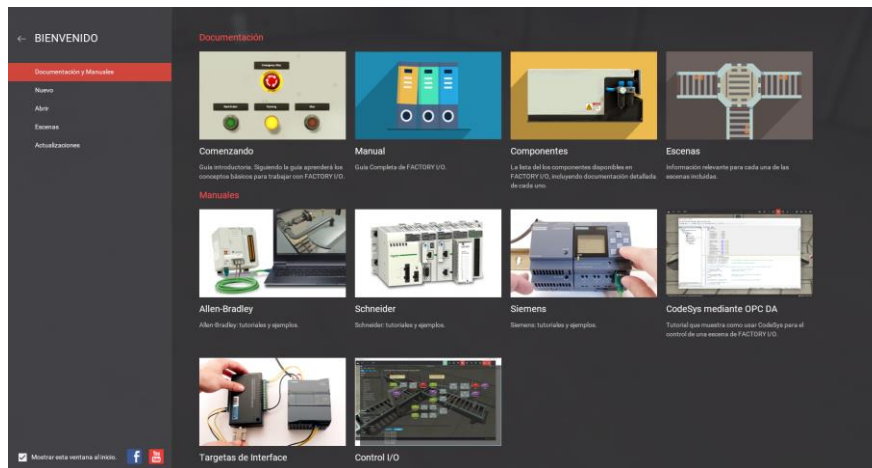


Figura 3.7. Ejemplo 3 KOP.

### 3.1.3.2. Entorno Factory IO

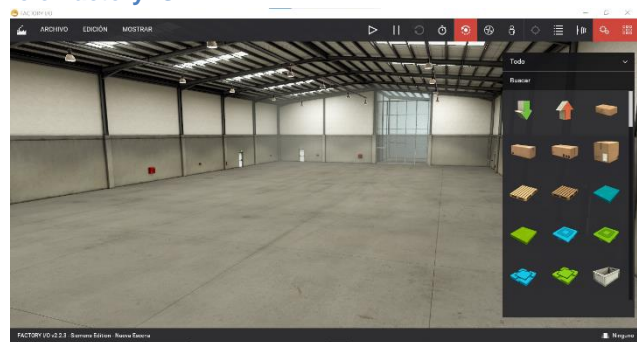
Se comienza abriendo el software de simulación Factory IO. En la pantalla principal (*Figura 3.8*) se encuentra a la derecha de la misma todo tipo de documentación y manuales sobre Factory IO, que pueden servir de gran ayuda a la hora de resolver diversas dudas.

En la parte izquierda de la pantalla se encuentra el menú que contiene diversas opciones como crear una nueva escena, abrir una nueva escena o abrir una escena prediseñada



**Figura 3.8. Inicio Factory IO.**

El espacio virtual (*Figura 3.9*) del que consta Factory IO es de una nave industrial vacía donde se colocarán los componentes con los que se trabajaran, la interface que presenta es muy simple e intuitiva y eso facilita mucho la fluidez a la hora de trabajar en él.



**Figura 3.9. Factory IO entorno de trabajo.**

#### **Barra de tareas**

En la parte superior se encuentra la barra de tareas (*Figura 3.10*) la cual consta de diversos comandos que se explicarán a continuación:



**Figura 3.10. Barra de tareas Factory IO.**

En los primeros cuatro botones (*Figura 3.11*) de la parte izquierda se encuentran los clásicos botones de control del programa.

- El primero redirigirá a la pantalla de principal descrita anteriormente.
- “Archivo”, este permitirá guardar, crear, abrir o cerrar un programa, en las opciones dirigirá a una pantalla donde configurar aspectos relacionados con el audio, video, controles, licencias...etc. El apartado de los drivers es uno de los más importantes de

esta sección ya que dentro de este se configurarán las conexiones con los diferentes controladores que se utilizan en el proyecto.

- “Edición”, en este desplegable se encuentran opciones como copiar, pegar o deshacer entre otras, tareas que son habituales en cualquier programa.

- “Mostrar”, desde esta pestaña se pueden marcar que menús se quieren mostrar o no.



Figura 3.11. Menús Factory IO.

El resto de botones de la parte derecha (Figura 3.12) se dividen en tres grupos:

- Proceso, con estos cuatro botones se puede controlar la simulación haciendo que esta comience, pare, vuelva a su estado inicial o que el tiempo de la simulación transcurra más lento.

-Cámaras, desde estos botones se controla la forma de visualizar el proceso, Factory IO cuenta con 3 tipos de cámaras, la primera una cámara orbital, la segunda una cámara “mosca” que sería una vista similar a la que se tiene desde un dron, y la última sería una cámara de primera persona en la que se vería lo que un operario de lesa industria vería. Finalmente, el último botón de esta sección es una cámara de seguimiento en la que al pulsar en un objeto esta cámara se dedicaría a seguir este objeto a lo largo del proceso.

-Mostrar, estos botones sirven para mostrar u ocultar diferentes opciones, el primero mostrará un listado de cámaras guardadas por el usuario anteriormente, los dos siguientes permiten mostrar las etiquetas de los sensores y de los actuadores y el último botón sirve para desplegar la paleta con todos los componentes disponibles.

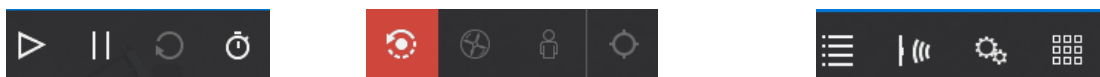


Figura 3.12. Resto de botones Factory IO.

### 3.1.3.3. Conexión TIA Portal y Factory IO

#### Conexión con PLC físico

12. Se conecta el PLC a la red.
13. Se abre un nuevo proyecto en TIA PORTAL.
14. Se agrega un CPU para estas prácticas se utilizará un S7-1200 1212 DC/DC/DC.
15. Se clic en detectar dispositivo, se selecciona PN/IE como tipo de interfaz y se elige el adaptador de red. Una vez hecho esto se pulsa sobre buscar, aparecerá el PLC y se clic sobre Cargar.
16. Una vez enlazado el PLC a TIA Portal se procederá a configurarlo. Para ello se debe ir a configuración de dispositivo, pulsar sobre el CPU y propiedades.
17. Las primeras 10 entradas están reservadas internamente en el programa Factory IO así que se debe configurar un offset para empezar a escribir entradas a partir de esa. Para ello dentro de propiedades hay que pulsar en el desplegable DI 6/ DQ4 > Direcciones E/S > Direcciones de entrada y en Dirección inicial e introducir un 10.

*(De lo contrario, los valores escritos por Factory I / O se sobrescribirán cuando el estado de las entradas físicas se copie en la memoria I)*

18. Para terminar la configuración cambiar algunos aspectos de Protección y Seguridad.

Hay que entrar en Protección & Seguridad y realizar los siguientes ajustes:

- Nivel de acceso > Acceso completo.
- Mecanismos de conexión > Permitir acceso vía comunicación PUT/GET.

19. Una vez hecho esto cargar el software que se ha programado en el PLC y abrir Factory IO.

20. En Factory IO se abrirá el proyecto y se debe clic en Archivo > Drivers (o bien F4), aparecerá un desplegable (*Figura 3.13*) y en él se seleccionará el CPU utilizado.

21. Dentro de Configuración, se escribirá el IP del PLC utilizado en el apartado "Host" y habrá que configurar tanto el adaptador de Red como el número de entradas y salida de cada tipo

22. Una vez realizado volver a la pantalla de Drivers y presionar sobre el botón de Conectar. Y el PLC quedará conectado a Factory IO.

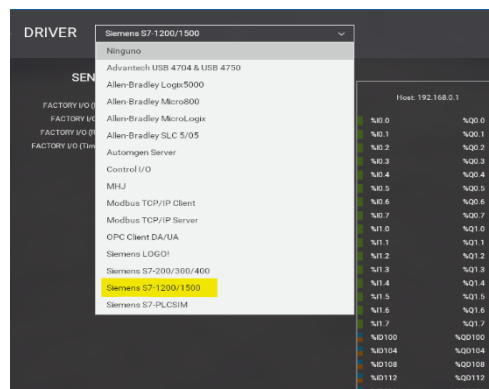


Figura 3.13. Drivers Factory IO.

### Conexión con S7-PLCSIM

- Lo primero que se debe hacer es entrar en la página de Factory IO y descargar la plantilla de proyecto compatible para TIA Portal, sin esto no se podrán comunicar ambos programas.

[Plantilla TIA Portal](#)

- Una vez descargada dentro de la plantilla se puede encontrar una función (Figura 3.14) la cual hay que escribir en el OB1. El programa comenzaría a partir del segmento 2.

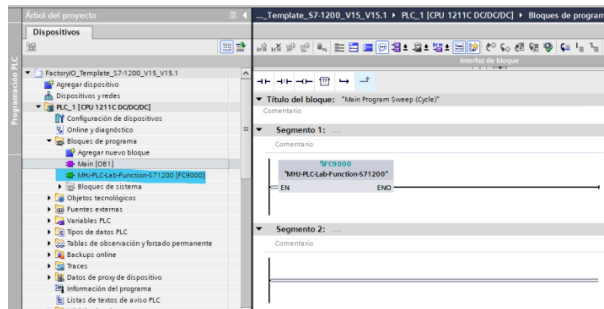


Figura 3.14. Función de Factory IO en TIA Portal.

- Tras esto, seleccionar nuestro dispositivo e iniciar la simulación.
- Se elegirá PN / IE como el tipo de interfaz PG / PC y en la interfaz PG / PC seleccione PLCSIM S7-1200 / S7-1500. Presionar Iniciar búsqueda. Cuando finalice el escaneo, seleccione el dispositivo y presione Cargar.
- Cuando aparezca la ventana del PLCSIM es importante iniciar el módulo, RUN.
- Abrir Factory IO, entrar en el proyecto y pulsar en Archivo > Drivers (o bien F4). Seleccionar “Siemens S7-PLCSIM”.
- Entrar en el panel de Configuración, dentro de configuración seleccionar el modelo de CPU y configurar las entradas y salidas.
- Presionar el botón de Conectar y el PLCSIM y Factory IO quedarían conectados (Figura 3.15).

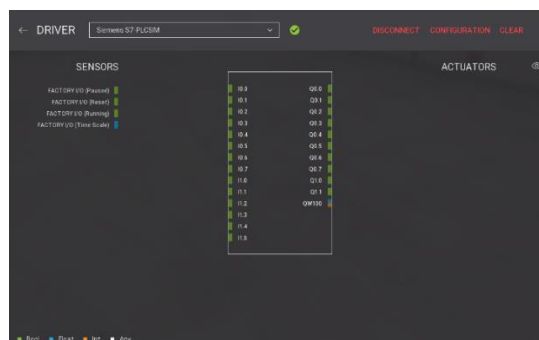


Figura 3.15. Conexiones Factory IO.

### Ejemplo 1 – Cinta y sensor. Digital (Factory IO)

Una vez se sabe cómo conectar las diferentes entradas y salidas del proyecto se realizarán pequeños ejemplos. Para este ejemplo no es necesario conectar el PLC. Se usará una cinta (da igual el tamaño), un objeto que colocar sobre ella (en este caso se usará una “Stackable box”) y un sensor retro reflectivo. (Figura 3.16)

El objetivo de este ejemplo es ver cómo funcionan los actuadores y sensores en Factory IO. Para ello pulsar sobre el nombre de cada objeto. Arriba a la derecha se abrirá una pestaña (Figura 3.17) donde ver el estado y desde la cual se podrá controlar la variable asociada.

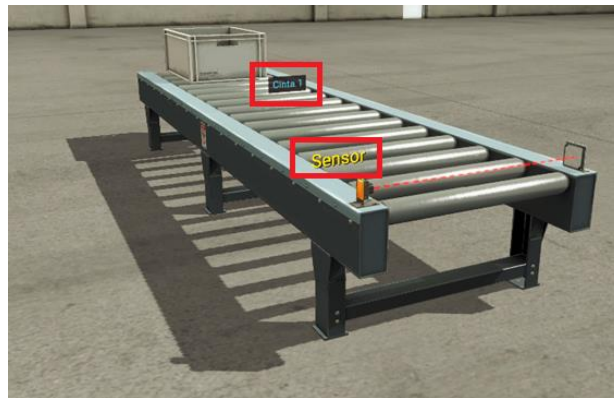


Figura 3.16. Estación ejemplo 1.



Figura 3.17. Cinta 1 y sensor.

Si se pulsa en el círculo verde que aparece al lado de la cinta se observará como este pasa a azul y la caja empieza a moverse, se está forzando la variable a True. Cuando la caja pase por el sensor se verá como el sensor se ilumina. Esta es la forma de indicar de que el sensor está activo.

Esta es la configuración digital de la cinta, se ha usado la cinta ya que es un elemento simple y en la que ver de forma visual que es lo que sucede. Pero cualquier elemento que tenga este tipo de configuración se controla de la misma forma.

## Ejemplo 2 – Cinta y sensor. Analógico (Factory IO)

En el ejemplo anterior se aprendió como controlar y visualizar los controladores analógicos en Factory IO, en este ejemplo se verá cómo cambiar la configuración de los actuadores y como trabajar con ellos cuando estos sean analógicos.

Para ello se utilizará la escena creada en el ejemplo anterior. Hay que pulsar en el botón derecho sobre la cinta y seleccionar la opción de configuración (*Figura 3.18*). Desde ahí se podrá cambiar la configuración de casi cualquier objeto de Factory IO.

Una vez dentro quitar la configuración y como se hizo en el ejemplo 1 pulsar sobre el nombre de la cinta, como se observa ahora la ventana ha cambiado y se encuentra un Slider desde el cual controlar el voltaje que se aplica al motor haciendo que este haga girar la cinta a mayor o menor velocidad.

Cuando se marca la configuración analógica se verá en el apartado de las conexiones (apartado de los Drivers) como la variable cambia de Bool a DWord. Esto se debe a que ahora para controlar esta variable se usa un dato de valor int.

Como se comentó anteriormente este tipo de configuración se da en casi todos los actuadores. Aunque no todos son completamente iguales son muy intuitivos por ello antes de comenzar a programar una máquina que no se conozca se recomienda leer el manual para entender las variables que se manejarán y las diferentes configuraciones que esta tenga.

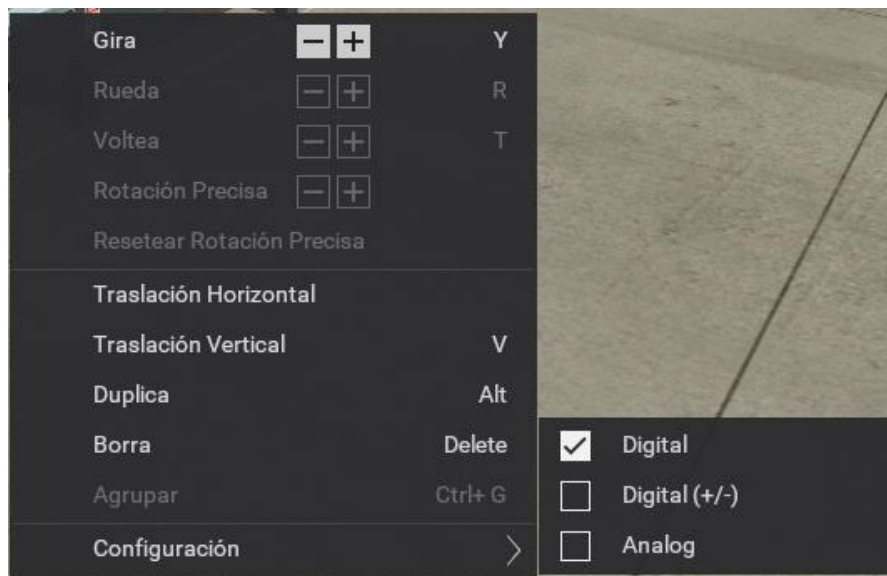


Figura 3.18. Configuración analógica /digital Factory IO.

### Ejemplo 3 – Cinta y sensor. Analógico (TIA Portal + Factory IO)

Una vez se conoce el funcionamiento de los elementos con los que se va a trabajar en Factory IO ya se puede pasar a su programación. Es importante que antes de programar se conozca que es lo que se va a programar, sus entradas, sus salidas y su forma de actuar ya que conociendo a la perfección la maquina se podrá programar de una forma más eficiente.

En este ejemplo (Figura 2.23) constará de los siguientes elementos:

- Un emisor
- Un transportador
- Un sensor
- Una salida
- Un cuadro de mandos compuesto por un botón de manual, otro de paro y uno de automático. (Figura 3.19)



Figura 3.19. Ejemplo 3, cuadro de mandos.

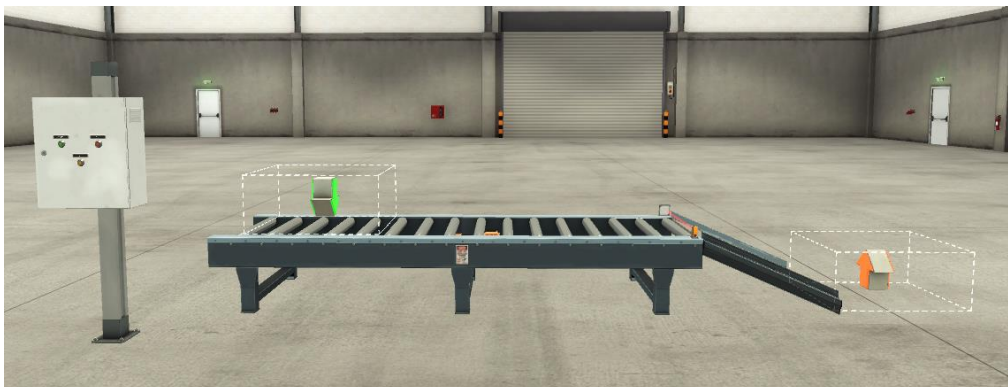


Figura 3.20. Estación ejemplo 3.

Lo primero que se hará será configurar las variables del PLC en Factory IO y exportarlas a TIA Portal para así trabajar con ellas desde el software de Siemens.

Recordar que para acceder a esto se debe pulsar sobre “ARCHIVO” (Figura 3.21) y después pulsar en drivers. Se abrirá la pantalla de configuración, colocar nuestros sensores y actuadores y seleccionar o bien el PLCSIM o el PLC físico.

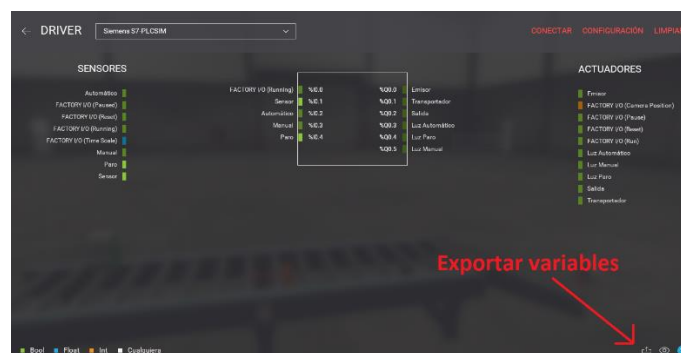


Figura 3.21. Exportar variables Factory IO.



Práctica 1 – Entorno TIA Portal

En la esquina inferior derecha se encuentra el botón para exportar las variables, se pulsa y se clicla en “DE ACUERDO” o si no se sabe la ubicación donde este archivo se guardará pulsar en “ABRIR CARPETA” (Figura 3.22).

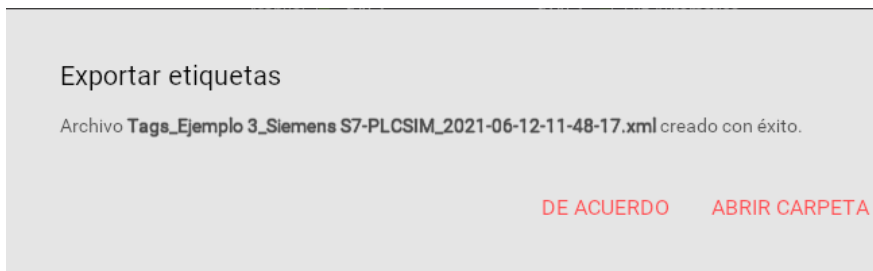


Figura 3.22. Exportar variables Factory IO 2.

Una vez hecho esto abrir ahora a TIA Portal, crear un proyecto nuevo como se ha visto anteriormente y cuando esté creado ir a las tablas de variables e importar las variables generadas anteriormente. (Figura .23).

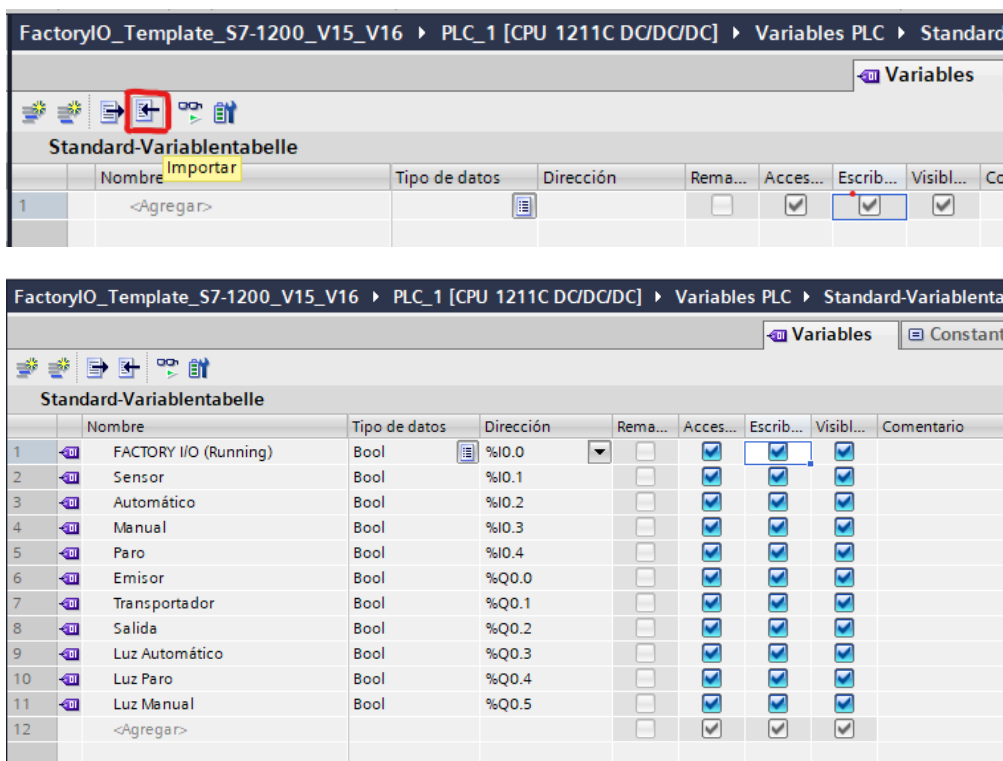


Figura 3.23. Importar variables en TIA Portal.

Pasadas las variables al TIA Portal es turno de diseñar el GRAFCET para el ejercicio propuesto, esta es una de las opciones que se ha diseñado (Figura 3.24).

Para ello se usará la escena creada en el ejemplo anterior. Pulsar en el botón derecho sobre la cinta y pulsar sobre la opción de configuración. Desde ahí se podrá cambiar la configuración de casi cualquier objeto de Factory IO.

Práctica 1 – Entorno TIA Portal

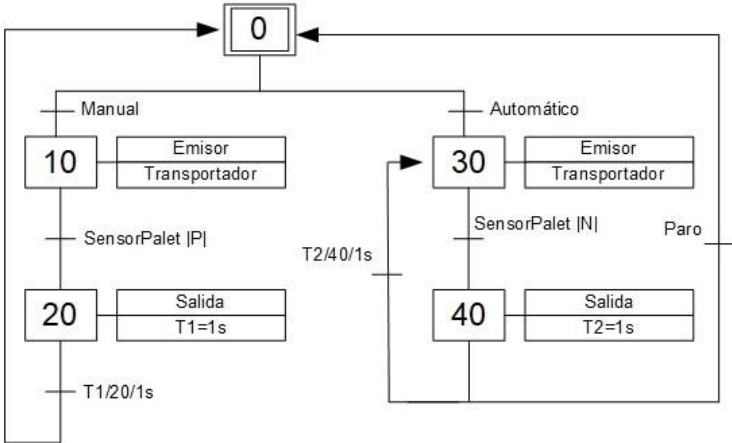


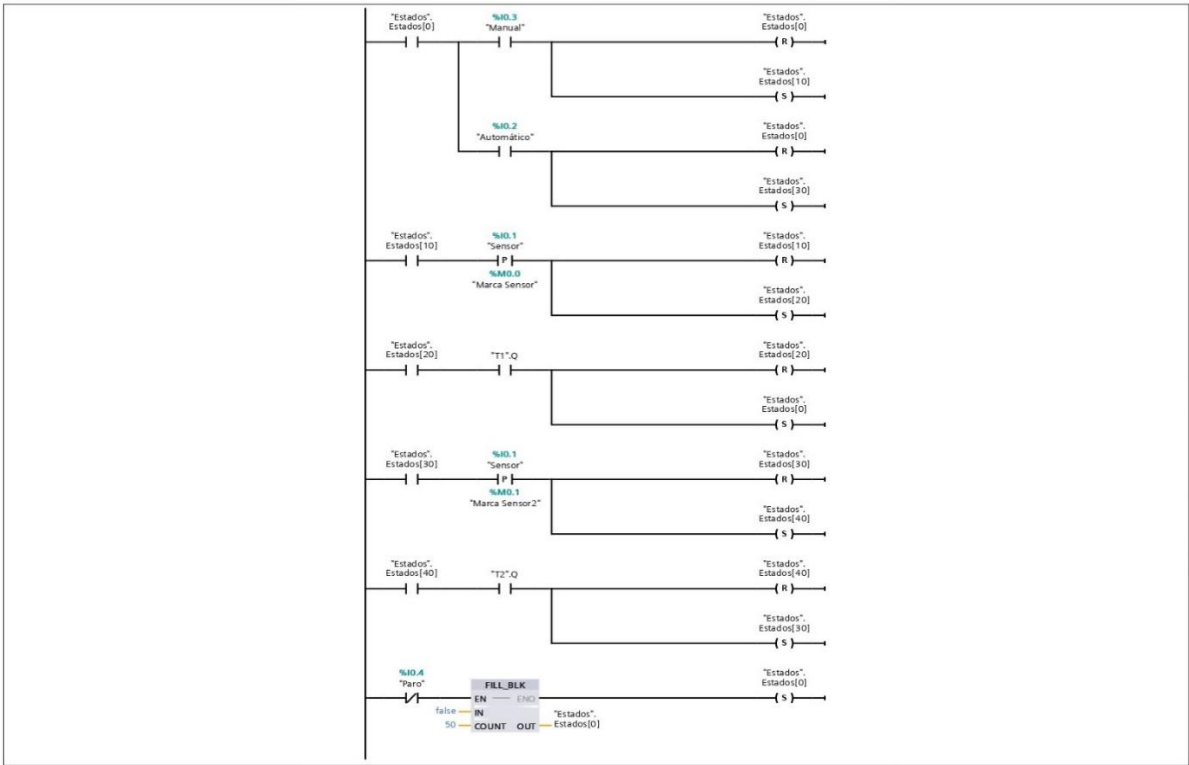
Figura 3.24. GRAFCET ejemplo 3.

Ahora toca programar lo diseñado en el GRAFCET en KOP en TIA Portal, quedaría de la siguiente forma:

Segmento 1:



Segmento 2:



Segmento 3: Transportador

Práctica 1 – Entorno TIA Portal

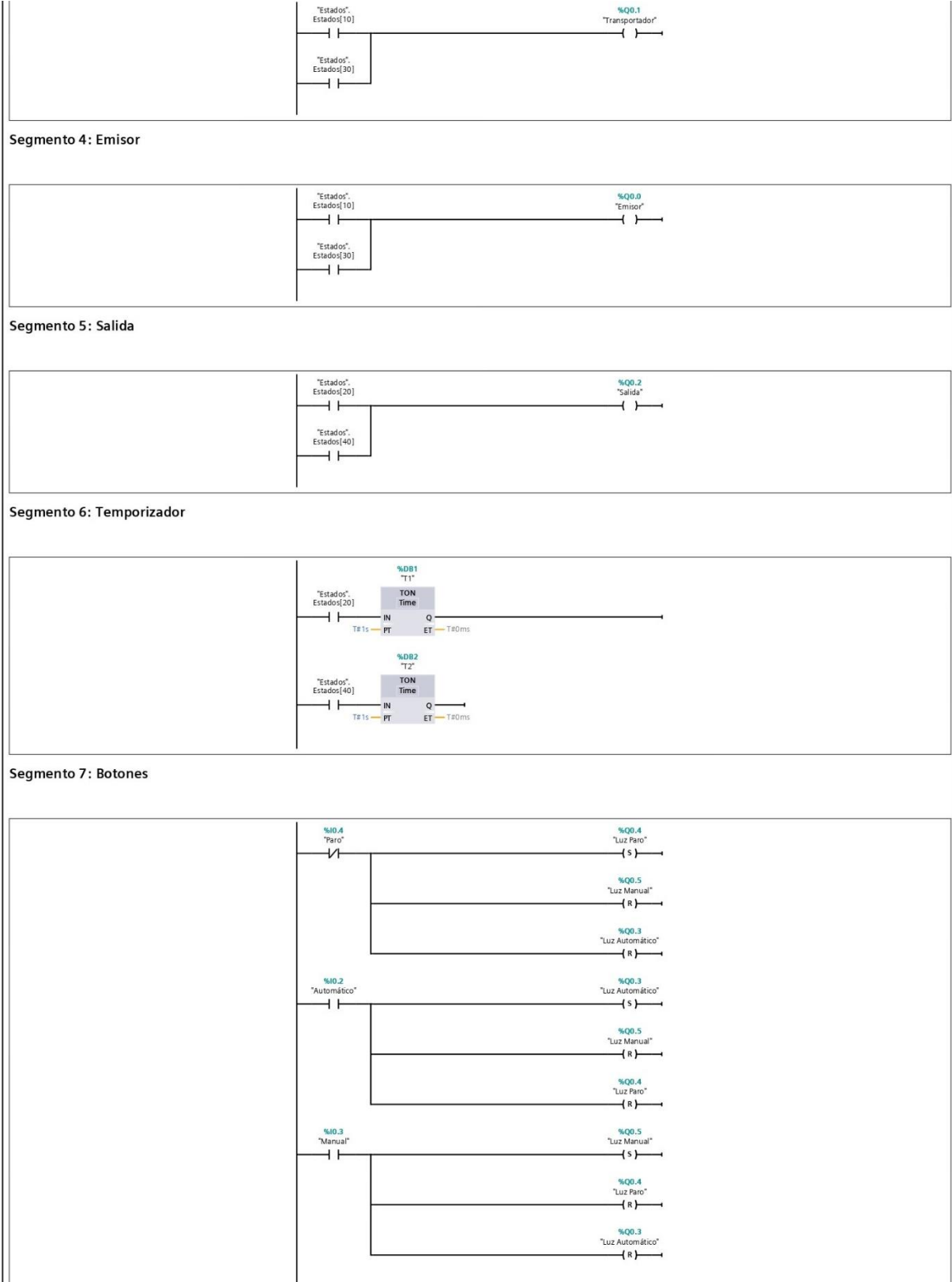


Figura 3.25. Programación en KOP del ejemplo 3.

Terminada la programación solo quedaría ejecutar el programa en TIA Portal y comprobar su funcionamiento con Factory IO.

## Ejercicio 1 – Cinta y sensor. Analógico (Factory IO)

En este ejercicio se intentará aplicar de forma autónoma lo que se ha aprendido a lo largo de esta práctica.

Este ejemplo constará de los siguientes elementos (*Figura 3.27*):

- Un transportador principal digital de dos sentidos (Tendrá dos salidas).
- Un transportador secundario (el del fondo) analógico el cuál trabaja desde los -10V hasta los 10 V.
- Cuatro transportadores en curva analógicos que trabajan desde los -10V hasta los 10 V.
- Dos sensores ubicados en cada una de las terminaciones del transportador principal.
- Un cuadro de mandos compuesto por un botón de marcha y un potenciómetro desde el que configuraremos la velocidad de las cintas.



Figura 3.26. Vídeo funcionamiento ejercicio 1.



Figura 3.27. Estación en Factory IO del ejercicio 1

[Vídeo](#)

[Factory IO](#)

El funcionamiento es el siguiente (*Figura 3.26*), se tendrá una caja parada en el centro del transportador principal (*Figura 3.28*). Se ajustará la velocidad y dirección del resto de cintas con el potenciómetro ubicado en cuadro de mandos y se activará el botón de marcha.

En ese momento si el potenciómetro marca un valor mayor de 1V o menor que -1V la cinta principal se moverá hacia la dirección marcada. Cuando el sensor de ese sentido detecte la caja activará el resto de transportadores con la velocidad marcada en el potenciómetro.

Una vez la caja ha dado una vuelta y es detectada por el sensor de entrada al transportador principal contaremos 3 segundos y pararemos todas las cintas como al comienzo.

La idea es poder repetir este proceso todas las veces que se quiera.

Práctica 1 – Entorno TIA Portal



Figura 3.28. Estación ejercicio 1.

Estas son las conexiones del Factory IO (Figura 3.29) y por tanto las variables que se tendrán en TIA Portal.

Botón Marcha	%I0.0	%Q0.0	Luz Botón Marcha
Sensor derecha	%I0.1	%Q0.1	Principal Positiva
Sensor izquierda	%I0.2	%Q0.2	Principal Negativa
Potenciómetro	%ID30 (REAL)	(REAL) %QD30	Curva 1
		(REAL) %QD34	Curva 2
		(REAL) %QD38	Curva 3
		(REAL) %QD42	Curva 4
		(REAL) %QD46	Secundaria

Figura 3.29. Conexión Factory IO ejercicio 1.

En la programación de los actuadores analógicos se necesitará conocer la forma de programarlos en TIA Portal esto se hace con la función MOVE. También es importante conocer como programar un temporizador, ambas cosas se muestran a continuación:

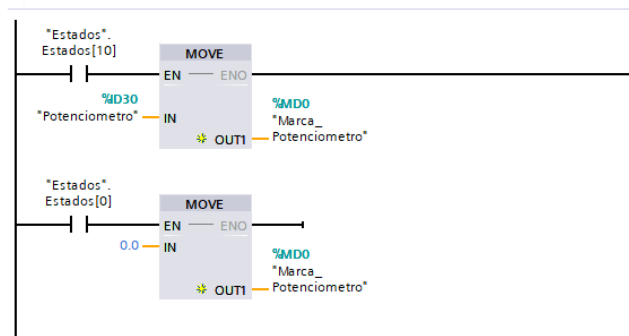


Figura 3.30. Programación variables analógicas.

Práctica 1 – Entorno TIA Portal

En el primer caso (Figura 3.30) en el estado 10 la “Marca\_Potenciometro” tendrá el mismo valor que tenga el potenciómetro. Y en el segundo caso en el estado 0 la “Marca\_Potenciometro” tendrá un valor de 0.0.

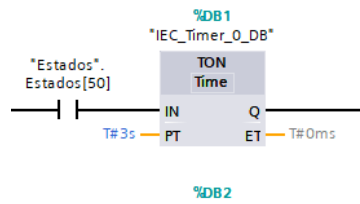


Figura 3.31. Programación temporizadores.

Para el caso del temporizador (Figura 3.31) se ha usado un temporizador tipo TON que es un temporizador con retardo al conectar esto quiere decir que en este ejemplo cuando se active el estado 50 pasado el tiempo de 3 segundos se activará la salida Q.

Una vez adquiridos los conocimientos necesarios podemos empezar a diseñar el GRAFCET este es GRAFCET (Figura 2.35) que se propone para este ejercicio.

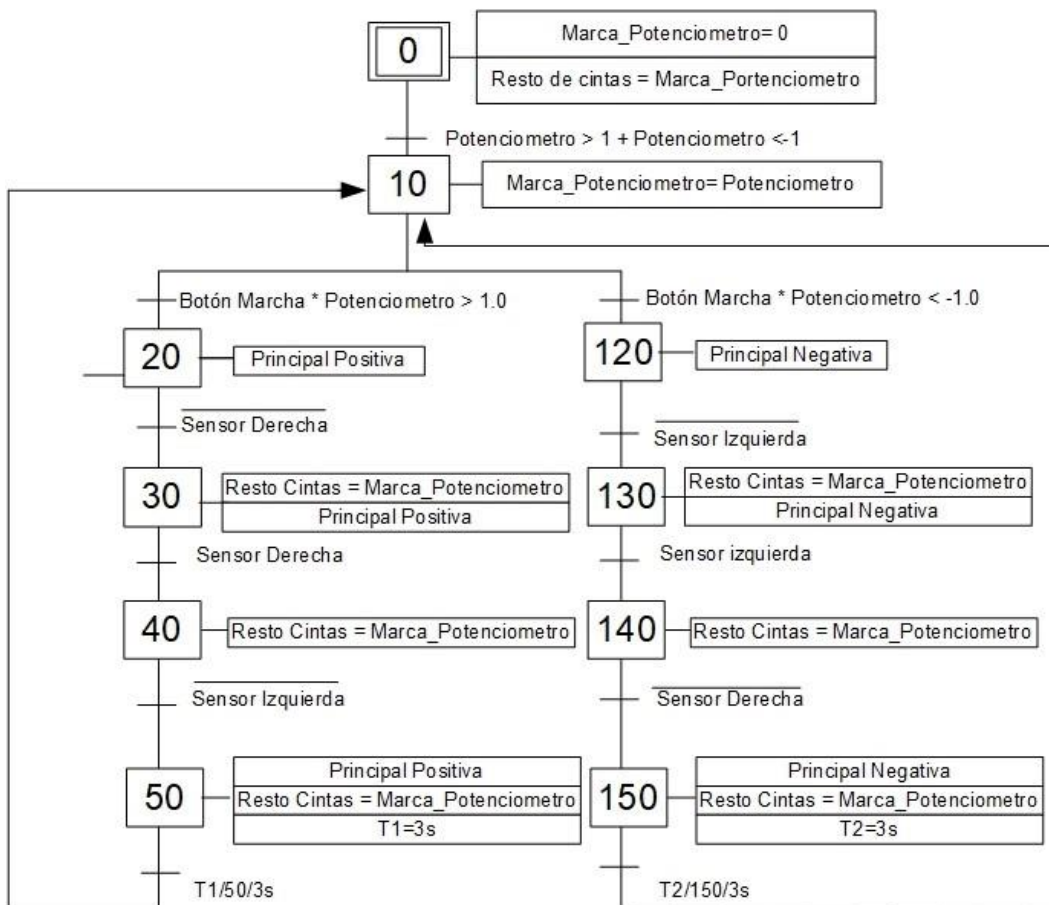


Figura 3.32. GRAFCET ejercicio 1.

### **3.1.4. Trabajo previo**

Para esta primera práctica es importante que el alumno tenga instalado en su ordenador los dos programas que vamos a utilizar, tanto TIA Portal como Factory IO.

También será muy útil para el alumno tener claros los conocimientos sobre como generar los GRAFCETS ya que son una de las claves para automatizar correctamente un proceso.

### **3.1.5. En el laboratorio**

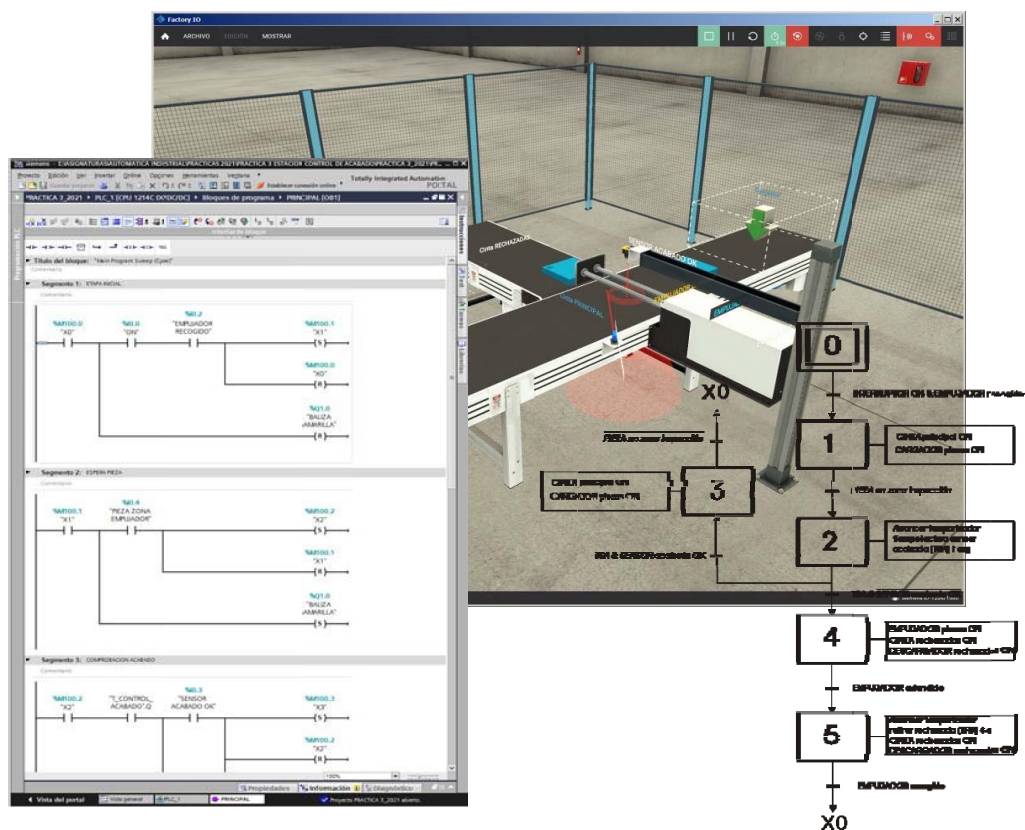
El alumno creará proyectos en TIA Portal y Factory IO ayudándose de la práctica guiada, una vez sepa crearlos y tenga cierto grado de soltura con ambos softwares realizará los ejemplos.

Para el último ejemplo el alumno deberá generar su propio GRACET.

### 3.2. Práctica 2 – Cuadro de mandos. Modos de trabajo



## AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 2. CUADRO DE MANDOS. MODOS DE TRABAJO

ALUMNO:

GRUPO:



### 3.2.1. Introducción

Una cosa que tienen en común todas las automatizaciones son los modos de trabajo. No siempre van a ser los mismos ni se tendrá el mismo número de modos, pero normalmente siempre se contará con alguno como automático o paro. Esto dependerá de la complejidad de tu automatización.

#### Los modos de trabajo en TIA Portal

Para la práctica de hoy se hablará sobre 4 modos importantes:

- Modo Paro
- Modo Manual
- Modo Homing
- Modo Automático

### 3.2.2. Objetivos

Para esta práctica se pide diseñar la lógica de un cuadro desde el que controlarlos modos de trabajo descritos anteriormente, para ello se utilizará el entorno de simulación Factory IO y TIA Portal.

Se recomienda hacer el código principal del cuadro dentro de una FB ya que se usará este cuadro en futuras prácticas.

### 3.2.3. Funcionamiento



Figura 3.33. Estación modos de trabajo.

### **Modo Paro**

Este modo puede plantearse de diversas formas, se trata de un paro controlado así se puede enfocarlo desde la perspectiva de terminar el ciclo en el que está trabajando y después parar o desde la perspectiva de terminar el movimiento y parar. Dependiendo de en qué proyecto se esté trabajando se elegirá uno u otro.

Por ejemplo, si se trabaja con una instalación que contiene movimientos sencillos y es fácil recuperar las condiciones iniciales de trabajo, cualquiera de las opciones podría valer, pero si es una máquina muy compleja tal vez sea más adecuado que termine el ciclo completamente y lo único que haga es no iniciar un nuevo ciclo.

### **Modo Manual**

Es un modo completamente necesario, cualquier accionamiento debe poder accionarse de forma individual, ya sea para tareas de mantenimiento como para tener un control más detallado de los actuadores.

### **Modo Homing**

Es un modo optativo. Este sirve para inicializar el proceso desde una posición de arranque controlada (siempre que se pueda).

Esto facilita bastante la función del usuario ya que el posicionado inicial de la máquina puede ser complicado, y con este modo esto sucedería de forma automática.

### **Modo automático**

Es el modo normal de trabajo, cualquier automatización perdería el sentido sin este modo ya que es el encargado de que la instalación trabaje de forma cíclica y autónoma.



Figura 3.34. Vídeo del funcionamiento de la práctica 2.

[Vídeo](#)



Figura 3.35. Estación modos de trabajo en Factory IO.

[Factory IO](#)

### 3.2.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de visualizar el video donde se muestra como de los enlaces enviados para entender el funcionamiento del Portal TIA y de la estación virtual.

### 3.2.5. En el laboratorio

El alumno deberá crear un programa capaz de controlar el cuadro de mandos propuesto y que contenga los modos de trabajo expuestos anteriormente.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

### 3.2.6. Solución

#### Programación TIA Portal:

Para la resolución de esta práctica es necesario saber cómo funciona cada modo de trabajo a la perfección al igual que la lógica que usan los botones e interruptores que vamos a utilizar.

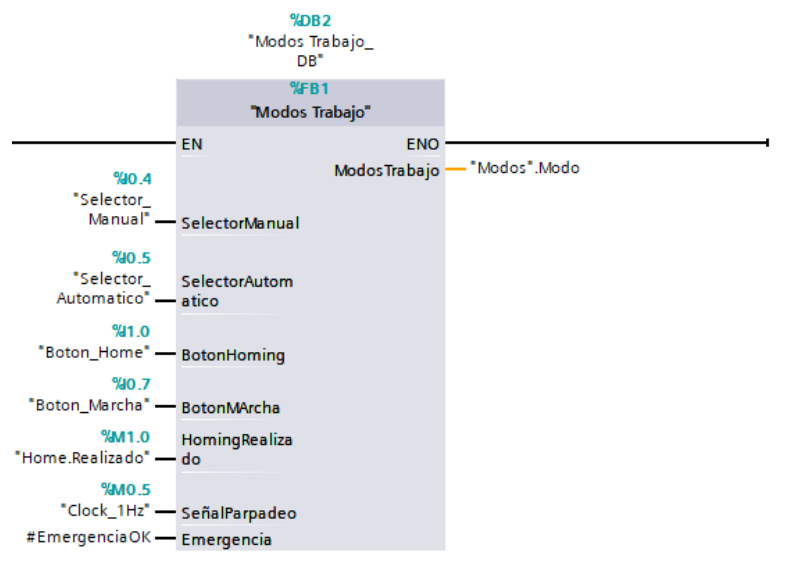


Figura 3.34. Programación función modos de trabajo.

Se ha creado una función (Figura 3.36) que dependiendo del estado de los botones del cuadro y del proceso determinará en qué modo se está trabajando y después este modo de trabajo se enviará al sistema para que el proceso trabaje de esta forma. Se empieza mostrando la programación de cada uno de los modos:

- **Modo emergencia:** Solo se entrará en este módulo si se pulsa el botón de emergencia (Figura 3.37).

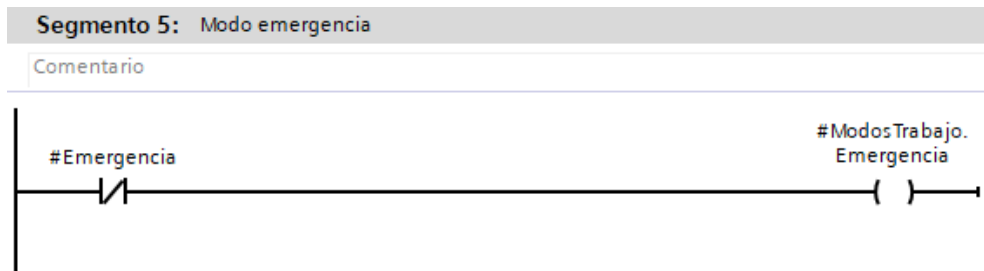


Figura 3.35. Programación modo de emergencia.

- **Modo stop:** Para entrar en este módulo existen dos posibilidades, una que se pulse la seta de emergencia y por lo tanto se detendrá el proceso y la otra opción es que el selector no se encuentre ni en automático ni en manual y por lo tanto estará parado el proceso (Figura 3.38).

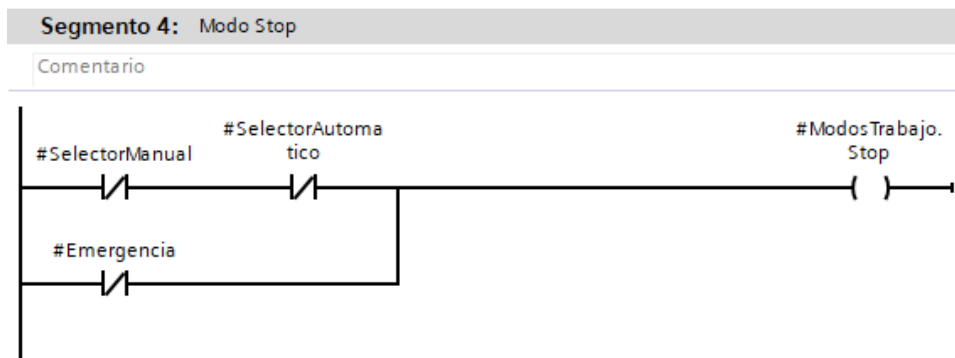


Figura 3.36. Programación modo stop.

- **Modo Manual:** Para activar este modo se debe de seleccionar en el selector la opción manual y además deben darse una serie de factores para que esto se cumpla. No debe haber ninguna emergencia, no se puede activar el modo homing, ni tampoco se puede activar el modo automático. Si sucediese alguna de estas cosas se desactivaría (Figura 3.39).

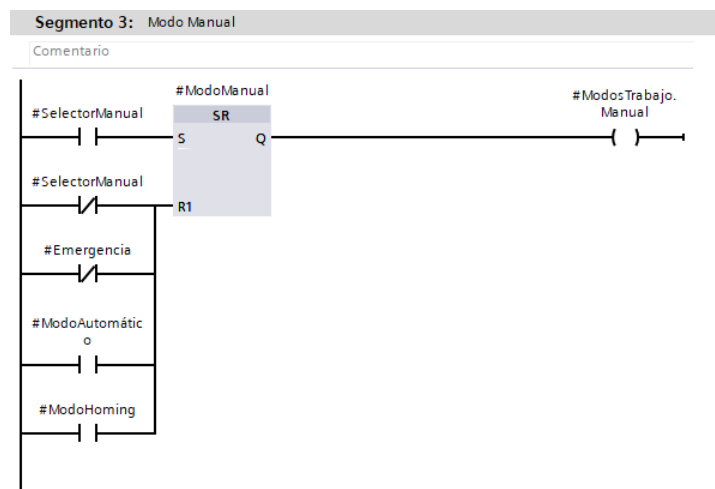


Figura 3.37. Programación modo manual.

Práctica 2 – Cuadro de mandos. Modos de trabajo

- **Modo automático:** En este modo necesario es necesario que el selector esté en automático, se haya realizado previamente un homing y finalmente pulsar el botón de marcha. Si en algún momento se cambiase el selector de posición o se entrara en modo paro o emergencias el modo automático se desactivaría (Figura 3.40).

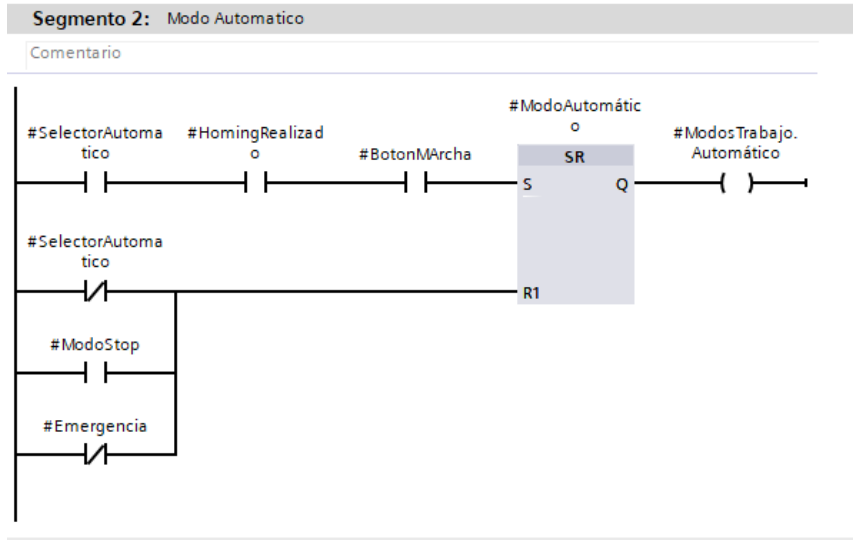


Figura 3.38. Programación modo automático.

- **Modo homing:** Para realizar un homing es necesario que el selector esté en automático y una vez lo esté se debe pulsar el botón de homing. No se podrá realizar un homing si el proceso se encuentra ya trabajando en automático, si estamos en manual o si se está en modo emergencia (Figura 3.41).

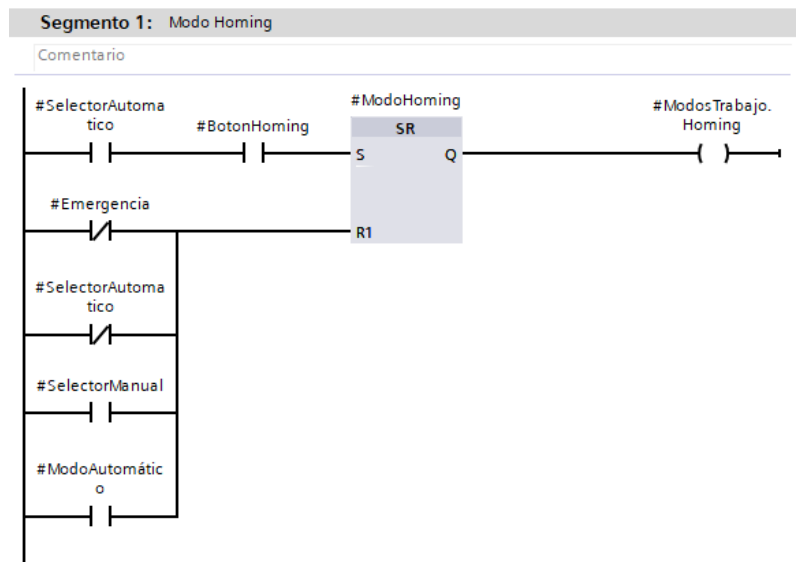


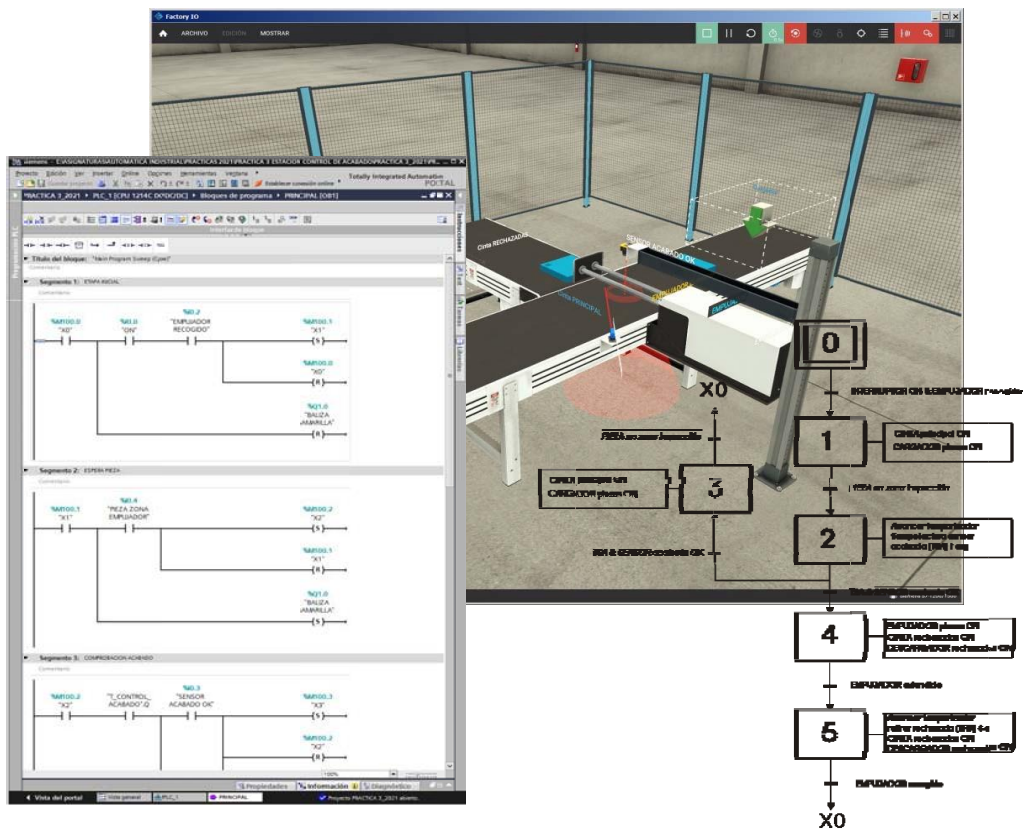
Figura 3.39. Programación modo homing.

Para finalizar una vez que se tiene programada nuestra función con los diferentes modos de trabajo ya solo que utilizarlos para programar los botones, semáforos o alarmas.

### 3.3. Práctica 3 – Automatización distribuidor de cajas



## AUTOMÁTICA INDUSTRIAL



### PRÁCTICA 3. AUTOMATIZACIÓN DISTRIBUIDOR DE CAJAS

ALUMNO:

GRUPO:

### 3.3.1. Introducción

En multitud de procesos industriales es necesario clasificar y distribuir las diferentes piezas u objetos que se ven envueltos en el mismo. Para ello se dispone de todo tipo de sensores que son capaces de distinguir las características de esos objetos, características como, por ejemplo, color, tamaño, peso, material...etc.

En esta práctica se desarrollará un distribuidor automático que permita clasificar las diferentes cajas de un proceso basándose en el tamaño de las mismas. Para el desarrollo de esta práctica se contará con una estación virtual creado en el software Factory IO.

### 3.3.2. Objetivos

El objetivo principal de esta práctica es que el alumno se inicie en la programación de PLCs utilizando el software de Siemens TIA Portal. Se deberá pensar en la posible solución al problema de la automatización y generar un GRAFCET con ella.

### 3.3.3. Funcionamiento

El funcionamiento de esta estación es simple, se emitirán cajas de diferentes tamaños desde la cinta principal, estas pasarán por un sensor el cual determinará la altura de las mismas.

Una vez medidas y detectada el tipo de caja que tenemos el distribuidor central se encargará de enviar la caja a la cinta correspondiente. La distribución de las cajas y la altura es la siguiente:

Tipo de caja	Altura	Dirección
Pequeña	128	Derecha
Mediana	192	Izquierda
Grande	224	Centro

Tabla 5. Tamaños de las cajas.

Práctica 3 – Automatización distribuidor de cajas

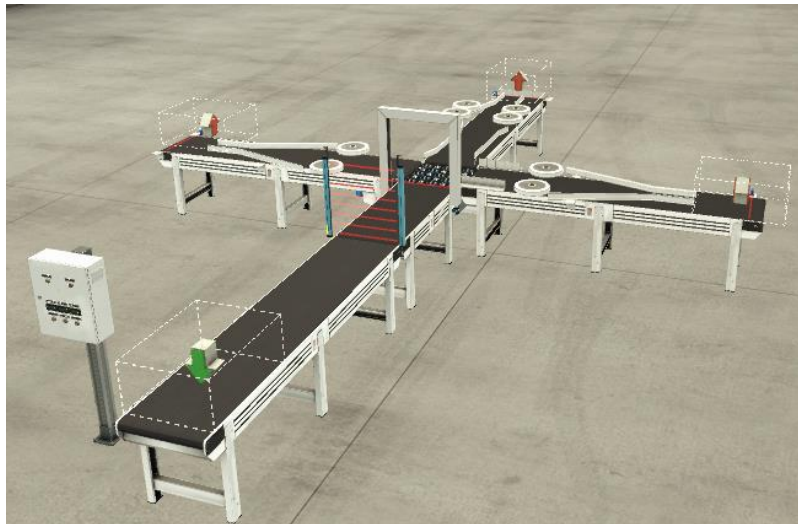


Figura 3.40. Estación distribuidora de cajas.

En el cuadro de mandos (Figura 3.43) se compone de 5 botones y tres displays. En los display se mostrará la cantidad de cajas de cada tamaño que ya han sido clasificadas y los botones serán uno de marcha, otro de paro y los tres de la parte inferior serán los encargados de resetear cada display.



Figura 3.41. Cuadro de mandos de la estación distribuidora de cajas.



Práctica 3 – Automatización distribuidor de cajas

Estás serán conexiones a llevar a cabo entre TIA Portal y Factory IO (Figura 3.44).

Botón Marcha	%I5.0	%Q5.0	Emisor
Botón Paro	%I5.1	%Q5.1	Cinta principal 1
Reset Grandes	%I5.2	%Q5.2	Cinta principal 2
Reset Medianas	%I5.3	%Q5.3	Distribuidor
Reset Pequeñas	%I5.4	%Q5.4	Distribuidor Izquierda
Sensor Cinta Principal	%I5.5	%Q5.5	Distribuidor Derecha
Sensor Izquierda	%I5.6	%Q5.6	Cinta Izquierda
Sensor Centro	%I5.7	%Q5.7	Cinta Centro
Sensor Derecha	%I6.0	%Q6.0	Cinta Derecha
	%I6.1	%Q6.1	Salida Izquierda
	%I6.2	%Q6.2	Salida Centro
	%I6.3	%Q6.3	Salida Derecha
Sensor Altura	%ID30 (DINT)	%Q6.4	
	%ID34	%Q6.5	
	%ID38	%Q6.6	
	%ID42	(DINT) %QD30	Contador Grandes
		(DINT) %QD34	Contador Medianas
		(DINT) %QD38	Contador Pequeñas
		%QD42	

Figura 3.42. Conexiones distribuidora de cajas.



Figura3.45. Vídeo del funcionamiento de la estación distribuidora de cajas.



Figura 3.46. Estación distribuidora de cajas en Factory IO.

[Vídeo](#)

[Factory IO](#)

### 3.3.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar lo visto en la práctica 1
3. Graficet con la solución propuesta.

### 3.3.5. En el laboratorio

El alumno deberá crear un programa en KOP capaz de controlar el distribuidor de cajas propuesto en Factory IO.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

### 3.3.6. Solución

#### Parte secuencial:

Para el control de esta estación de trabajo se ha diseñado un GRAFCET que permite englobar el funcionamiento del proceso de una forma clara y concisa.

- **Distribuidor de cajas:** Este GRAFCET (*Figura 3.47*) refleja el trabajo completo de toda la estación. Como se observa el GRAFCET comienza con una parte común la cual es la responsable de controlar la cinta principal de abastecimiento y después se bifurca en tres ramas, cada una de ellas corresponde con los distintos tipos de cajas que tenemos en el proceso.

Práctica 3 – Automatización distribuidor de cajas

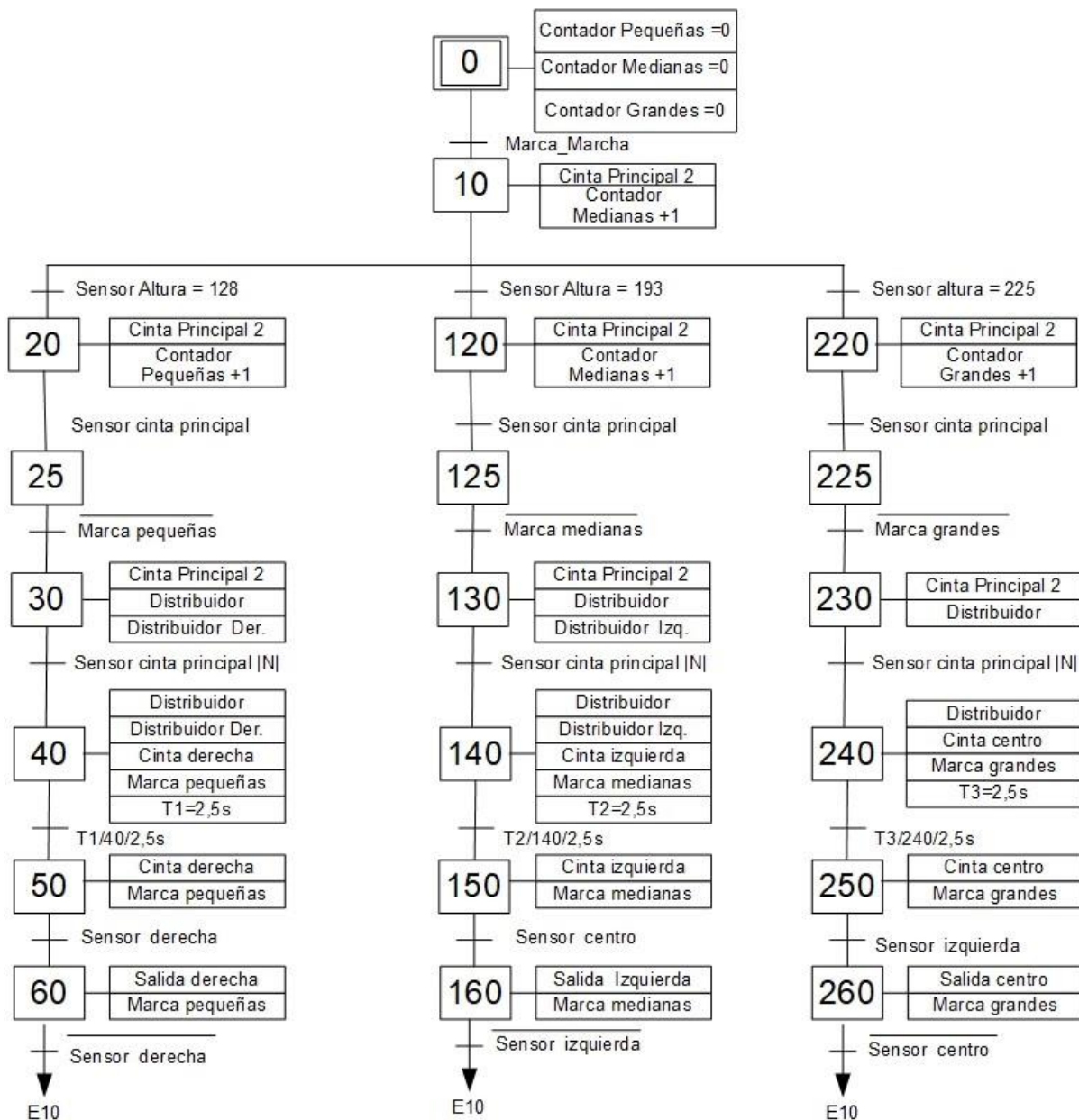


Figura 3.43. GRAFCET distribuidora de cajas.

**Programación PLC:**

La programación de la estación se realizará en lenguaje de contactos también conocido como KOP.

La configuración de entradas y salidas utilizadas en esta práctica es igual a la que se puede visualizar en Factory IO por lo tanto esas variables sumadas a marcas del sistema utilizadas, temporizadores y otros elementos componen todas las variables el proyecto.

Práctica 3 – Automatización distribuidor de cajas

Nombre de la variable	Tipo de dato	Dirección
Botón Marcha	Bool	%I5.0
Botón Paro	Bool	%I5.1
Reset Grandes	Bool	%I5.2
Reset Medianas	Bool	%I5.3
Reset Pequeñas	Bool	%I5.4
Sensor Cinta Principal	Bool	%I5.5
Sensor Izquierda	Bool	%I5.6
Sensor Centro	Bool	%I5.7
Sensor Derecha	Bool	%I6.0
Emisor	Bool	%Q5.0
Cinta principal 1	Bool	%Q5.1
Cinta principal 2	Bool	%Q5.2
Distribuidor	Bool	%Q5.3
Distribuidor Izquierda	Bool	%Q5.4
Distribuidor Derecha	Bool	%Q5.5
Cinta Izquierda	Bool	%Q5.6
Cinta Centro	Bool	%Q5.7
Cinta Derecha	Bool	%Q6.0
Salida Izquierda	Bool	%Q6.1
Salida Centro	Bool	%Q6.2
Salida Derecha	Bool	%Q6.3
Sensor Altura	DInt	%ID30
Contador Grandes	DInt	%QD30
Contador Medianas	DInt	%QD34
Contador Pequeñas	DInt	%QD38

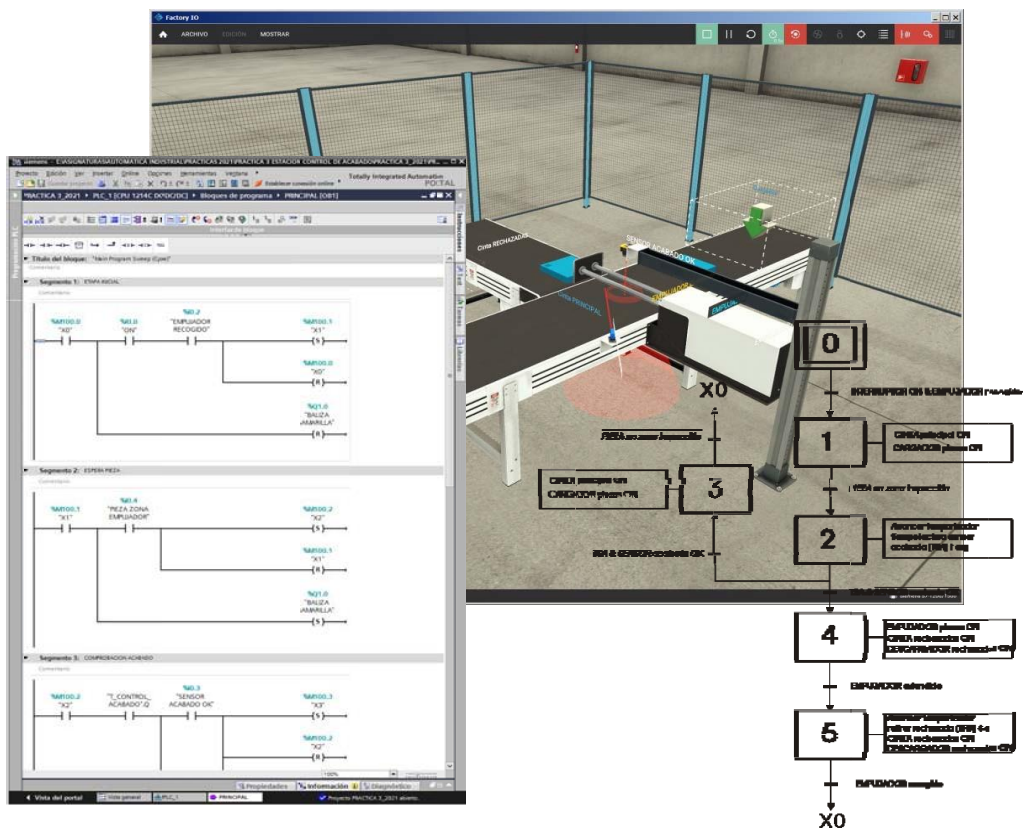
Tabla 6. Variables de la distribuidora de cajas.

Al programar todo un GRAFCET y quedar claro el método de programación se ha decidido realizar toda la programación en el bloque de trabajo OB1.

### 3.4. Práctica 4 – Módulo de trabajo Pick and Place



## AUTOMÁTICA INDUSTRIAL



### PRÁCTICA 4. MÓDULO DE TRABAJO PICK AND PLACE

ALUMNO:

GRUPO:

### 3.4.1. Introducción

La estación Pick and Place (*Figura 3.48*) con tres ejes controlados por servomotores. A menudo se utiliza para trasladar cargas ligeras (por ejemplo, cajas de cartón) a otros transportadores o palets.

El Pick and Place tiene cuatro grados de libertad, tres corresponden al movimiento lineal de los ejes y otro a la rotación de la pinza. La pinza está habilitada por ventosas e incluye un sensor de proximidad. Puede ser controlado por valores digitales y analógicos, según la configuración seleccionada. Cuando se controla con E / S digital, el movimiento del eje se realiza de forma incremental (paso a paso) en cada flanco ascendente del valor de la etiqueta de control.



Figura 3.44. Robot cartesiano.

### 3.4.2. Objetivos

El objetivo de esta práctica es continuar avanzando en la programación de PLCs enfrentándose a diferentes situaciones. En el caso de esta práctica es la primera vez que el alumno se enfrenta a una máquina con diversas entradas y salidas como puede ser el robot cartesiano.

Además de ello con la propuesta de varios ejercicios con diferentes niveles de dificultad se pretende que una vez dominado el aspecto básico el alumno siga creciendo y pueda dar distintas soluciones para un mismo problema.

### 3.4.3. Funcionamiento

Esta práctica estará compuesta por tres módulos (*Figura 3.49*):

- Cinta transportadora de cajas: Esta parte será la encargada de llevar las cajas a la máquina Pick and Place.
- Cinta transportadora de palets: Este módulo llevará los palets al centro de la máquina Pick and Place donde quedarán parados hasta que estos se carguen y estén listos para continuar.
- Pick and Place: Es el nexo de unión entre los dos módulos anteriores ya que su función es coger las cajas que habrá en la cinta transportadora de cajas y apilarlas en el palet.

Práctica 4 – Módulo de trabajo Pick and Place

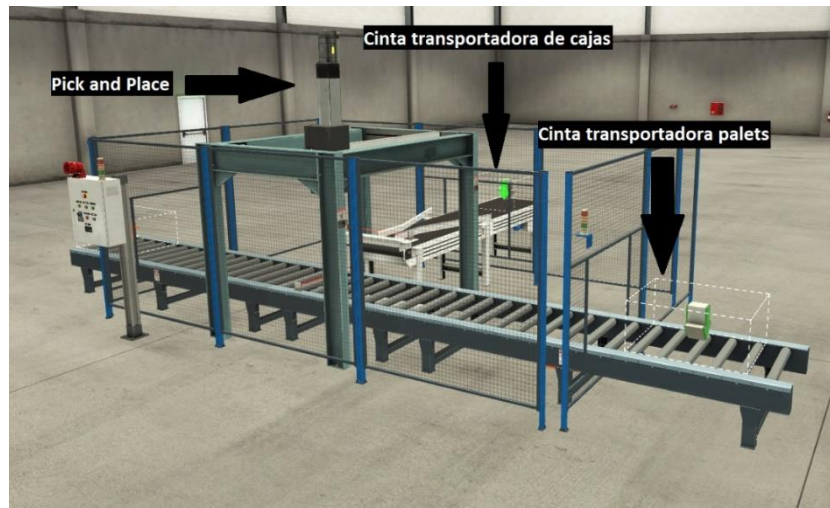


Figura 3.45. Estación pick and place.



Figura 3.50. Vídeo del funcionamiento de la estación Pick and place. IO.

[Vídeo](#)



Figura 3.51. Estación Pick and place en Factory IO.

[Factory IO](#)

**Descripción de las variables en configuración analógica del Pick and Place**

En esta tabla se muestran todas las entradas y salidas pertenecientes al robot cartesiano:

Nombre de la variable	Controlador I/O	Tipo	Descripción
Pick & Place # X Set Point (V)	Salida	Float	[0, 10] V: establece la posición de destino a lo largo del eje X.
Pick & Place # Y Set Point (V)	Salida	Float	[0, 10] V: establece la posición de destino a lo largo del eje Y.
Pick & Place # Z Set Point (V)	Salida	Float	[0, 10] V: establece la posición de destino a lo largo del eje Z.
Pick & Place # C(+)	Salida	Bool	Gire la pinza.
Pick & Place # (Grab)	Salida	Bool	Activar ventosas.

Práctica 4 – Módulo de trabajo Pick and Place

Pick & Place # X Position (V)	Entrada	Float	[0, 10] V: posición actual a lo largo del eje X.
Pick & Place # Y Position (V)	Entrada	Float	[0, 10] V: posición actual a lo largo del eje Y.
Pick & Place # Z Position (V)	Entrada	Float	[0, 10] V: posición actual a lo largo del eje Z.
Pick & Place # (Box Detected)	Entrada	Bool	Detectando un artículo.
Pick & Place # (C Limit)	Entrada	Bool	Pinza en límite angular.

Tabla 7. Configuración analógica del robot cartesiano.

Para esta práctica se proponen tres niveles de dificultad:

- **Nivel básico:** Programar la estación con el objetivo de cargar los palets, se apilarán las cajas de la forma que el alumno deseé. (Se recomienda de la forma que se hace en el vídeo)
- **Nivel medio:** Programar la estación y en este caso el operario podrá elegir en que formación apilar las cajas (*Figura 3.52*).

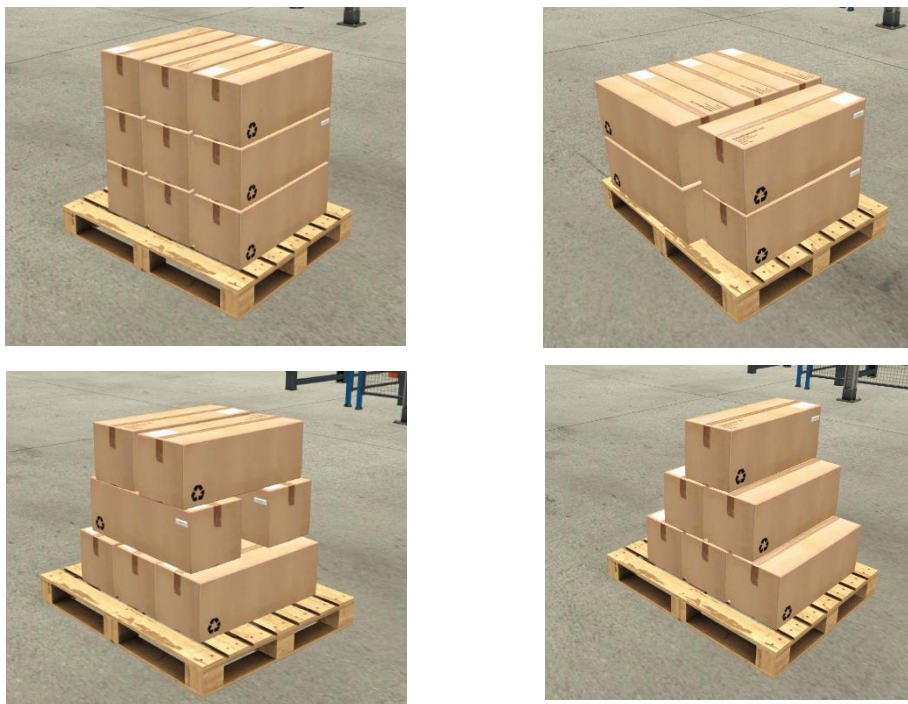


Figura 3.46. Nivel medio, apilado de cajas.

- **Nivel avanzado:** Realizar un HMI desde el cual el operario pueda controlar la estación (incluir lo programado en los niveles anteriores).



### 3.4.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar lo visto en las prácticas anteriores.
3. Graficet con la solución propuesta.

### 3.4.5. En el laboratorio

El alumno deberá crear un programa en KOP capaz de controlar el módulo pick and place propuesto en Factory IO.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

### 3.4.6. Solución

#### Parte secuencial:

Para el control de esta célula de trabajo se han diseñado dos GRAFCETS que permiten entender de una forma más rápida el funcionamiento de los diferentes procesos.

Estos son los dos GRAFCET en los que se han dividido el proceso:

- **Transportador de palets:** Este GRAFCET (*Figura 3.53*) es el encargado de nutrir al robot de bases de palets sobre los que el robot colocará las cajas que le lleguen de la cinta.

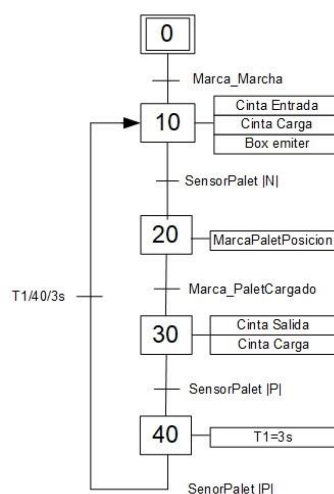


Figura 3.47. GRAFCET transportador de palets.

Práctica 4 – Módulo de trabajo Pick and Place

- **Cinta y Robot:** Este GRAFCET (Figura 3.54) engloba dos procesos, el control de la cinta que lleva las cajas hasta el robot y el encargado del robot cartesiano. Es por ello la complejidad que este puede llegar a tener.

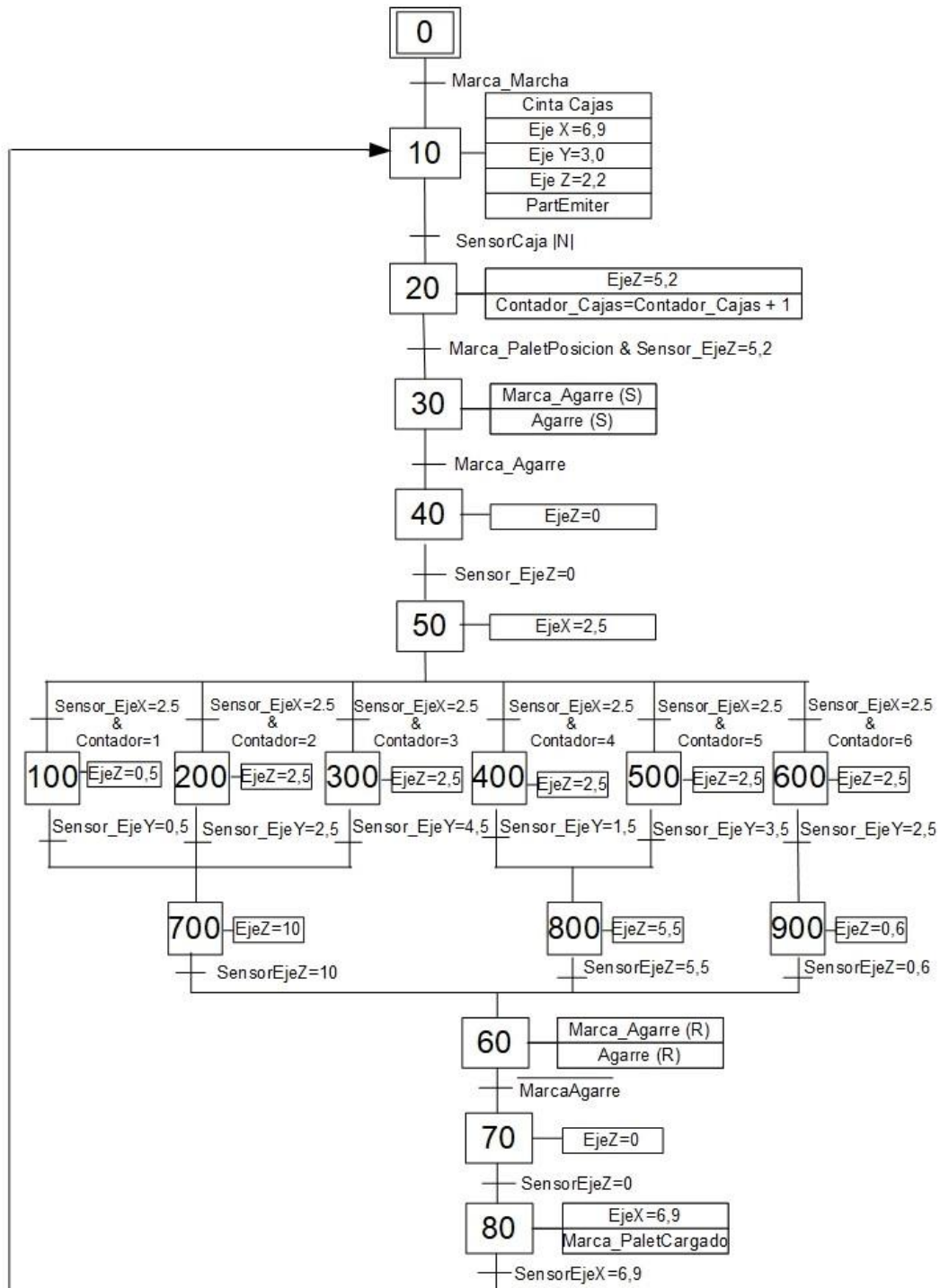


Figura 3.48. GRAFCET cinta y robot.

Práctica 4 – Módulo de trabajo Pick and Place

**Programación TIA Portal**

La programación de la estación se realizará en lenguaje de contactos también conocido como KOP.

La configuración de entradas y salidas (*Figura 3.55*) utilizadas en esta práctica es igual a la que se puede visualizar en Factory IO por lo tanto esas variables sumadas a marcas del sistema utilizadas, temporizadores y otros elementos componen todas las variables el proyecto.

SensorPalet	%I0.0	%Q0.0	CintaCarga
SensorCaja	%I0.1	%Q0.1	CintaCaja
Objeto_detectado	%I0.2	%Q0.2	CitaEntrada
C limit	%I0.3	%Q0.3	CintaSalida
Start	%I0.4	%Q0.4	Agarre
Reset	%I0.5	%Q0.5	Exit green
Stop	%I0.6	%Q0.6	Exit yellow
Emergency stop	%I0.7	%Q0.7	Exit red
Auto	%I1.0	%Q1.0	Start light
FACTORY I/O (Running)	%I1.1	%Q1.1	Reset light
	%I1.2	%Q1.2	Stop light
	%I1.3	%Q1.3	Exit green
	%I1.4	%Q1.4	Exit red
	%I1.5	%Q1.5	Box emitter
	%I1.6	%Q1.6	Part emitter
	%I1.7	%Q1.7	Exit yellow
	%I2.0	%Q2.0	
	%I2.1	%Q2.1	
	%I2.2	%Q2.2	
	%I2.3	%Q2.3	
Sensor_EjeX	%ID30 (REAL)	(REAL) %QD30	EjeX
Sensor_EjeY	%ID34 (REAL)	(REAL) %QD34	EjeY
Sensor_EjeZ	%ID38 (REAL)	(REAL) %QD38	EjeZ
		(DINT) %QD42	Counter

Figura 3.49. Conexiones Factory IO.

Debido a que de la forma en la que hemos diseñado el GRAFCET no era necesario el uso de funciones los únicos bloques de trabajo utilizados son el **OB100** para la inicialización del programa y de las entradas y salidas y el bloque **OB1** que es donde está el programa completo.

Práctica 5 – Programación de un ascensor

### 3.5. Práctica 5 – Programación de un ascensor



## AUTOMÁTICA INDUSTRIAL

The image displays the Factory IO simulation environment. On the left, the PLC ladder logic is shown in three segments:

- Segmento 1: INICIACIÓN:** Ladder logic involving relays NAR00.0, NAR00.1, NAR00.2, NAR00.3, NAR00.4, NAR00.5, NAR00.6, NAR00.7, NAR00.8, NAR00.9, NAR00.10, NAR00.11, NAR00.12, NAR00.13, NAR00.14, NAR00.15, NAR00.16, NAR00.17, NAR00.18, NAR00.19, NAR00.20, NAR00.21, NAR00.22, NAR00.23, NAR00.24, NAR00.25, NAR00.26, NAR00.27, NAR00.28, NAR00.29, NAR00.30, NAR00.31, NAR00.32, NAR00.33, NAR00.34, NAR00.35, NAR00.36, NAR00.37, NAR00.38, NAR00.39, NAR00.40, NAR00.41, NAR00.42, NAR00.43, NAR00.44, NAR00.45, NAR00.46, NAR00.47, NAR00.48, NAR00.49, NAR00.50, NAR00.51, NAR00.52, NAR00.53, NAR00.54, NAR00.55, NAR00.56, NAR00.57, NAR00.58, NAR00.59, NAR00.60, NAR00.61, NAR00.62, NAR00.63, NAR00.64, NAR00.65, NAR00.66, NAR00.67, NAR00.68, NAR00.69, NAR00.70, NAR00.71, NAR00.72, NAR00.73, NAR00.74, NAR00.75, NAR00.76, NAR00.77, NAR00.78, NAR00.79, NAR00.80, NAR00.81, NAR00.82, NAR00.83, NAR00.84, NAR00.85, NAR00.86, NAR00.87, NAR00.88, NAR00.89, NAR00.90, NAR00.91, NAR00.92, NAR00.93, NAR00.94, NAR00.95, NAR00.96, NAR00.97, NAR00.98, NAR00.99.
- Segmento 2: ESPERA PIEZA:** Ladder logic involving relays NAR00.1, NAR00.2, NAR00.3, NAR00.4, NAR00.5, NAR00.6, NAR00.7, NAR00.8, NAR00.9, NAR00.10, NAR00.11, NAR00.12, NAR00.13, NAR00.14, NAR00.15, NAR00.16, NAR00.17, NAR00.18, NAR00.19, NAR00.20, NAR00.21, NAR00.22, NAR00.23, NAR00.24, NAR00.25, NAR00.26, NAR00.27, NAR00.28, NAR00.29, NAR00.30, NAR00.31, NAR00.32, NAR00.33, NAR00.34, NAR00.35, NAR00.36, NAR00.37, NAR00.38, NAR00.39, NAR00.40, NAR00.41, NAR00.42, NAR00.43, NAR00.44, NAR00.45, NAR00.46, NAR00.47, NAR00.48, NAR00.49, NAR00.50, NAR00.51, NAR00.52, NAR00.53, NAR00.54, NAR00.55, NAR00.56, NAR00.57, NAR00.58, NAR00.59, NAR00.60, NAR00.61, NAR00.62, NAR00.63, NAR00.64, NAR00.65, NAR00.66, NAR00.67, NAR00.68, NAR00.69, NAR00.70, NAR00.71, NAR00.72, NAR00.73, NAR00.74, NAR00.75, NAR00.76, NAR00.77, NAR00.78, NAR00.79, NAR00.80, NAR00.81, NAR00.82, NAR00.83, NAR00.84, NAR00.85, NAR00.86, NAR00.87, NAR00.88, NAR00.89, NAR00.90, NAR00.91, NAR00.92, NAR00.93, NAR00.94, NAR00.95, NAR00.96, NAR00.97, NAR00.98, NAR00.99.
- Segmento 3: COMPROBACION ACABADO:** Ladder logic involving relays NAR00.1, NAR00.2, NAR00.3, NAR00.4, NAR00.5, NAR00.6, NAR00.7, NAR00.8, NAR00.9, NAR00.10, NAR00.11, NAR00.12, NAR00.13, NAR00.14, NAR00.15, NAR00.16, NAR00.17, NAR00.18, NAR00.19, NAR00.20, NAR00.21, NAR00.22, NAR00.23, NAR00.24, NAR00.25, NAR00.26, NAR00.27, NAR00.28, NAR00.29, NAR00.30, NAR00.31, NAR00.32, NAR00.33, NAR00.34, NAR00.35, NAR00.36, NAR00.37, NAR00.38, NAR00.39, NAR00.40, NAR00.41, NAR00.42, NAR00.43, NAR00.44, NAR00.45, NAR00.46, NAR00.47, NAR00.48, NAR00.49, NAR00.50, NAR00.51, NAR00.52, NAR00.53, NAR00.54, NAR00.55, NAR00.56, NAR00.57, NAR00.58, NAR00.59, NAR00.60, NAR00.61, NAR00.62, NAR00.63, NAR00.64, NAR00.65, NAR00.66, NAR00.67, NAR00.68, NAR00.69, NAR00.70, NAR00.71, NAR00.72, NAR00.73, NAR00.74, NAR00.75, NAR00.76, NAR00.77, NAR00.78, NAR00.79, NAR00.80, NAR00.81, NAR00.82, NAR00.83, NAR00.84, NAR00.85, NAR00.86, NAR00.87, NAR00.88, NAR00.89, NAR00.90, NAR00.91, NAR00.92, NAR00.93, NAR00.94, NAR00.95, NAR00.96, NAR00.97, NAR00.98, NAR00.99.

On the right, the 3D model shows the elevator shaft with floors 0, 1, 2, 4, and 5, and a call button X0. The PLC program includes logic for starting the elevator, waiting for a piece, and checking for completion at each floor.

## PRÁCTICA 5. PROGRAMACIÓN DE UN ASCENSOR

ALUMNO:

GRUPO:

### 3.5.1. Introducción

#### Las funciones

A la hora de programar resulta muy poco eficiente repetir las mismas secuencias dentro de un mismo programa ya que el programa será volverá muy extenso. Otro factor importante a la hora de programar es la memoria del PLC, debemos intentar que esta sea la mínima posible y esto se logra con la implementación de subrutinas.

#### *¿Qué es una subrutina?*

Una subrutina es un proceso que se ejecuta sólo cuando es invocada. Esto hace que mientras no esté activa no consumirá recursos. Permanecerá sin consumir recursos del PLC mientras no sea invocada.

Por ejemplo, el bloque OB1 es leído y ejecutado en cada ciclo SCAN del PLC, a diferencia de una subrutina que permanecerá “dormida” sin consumir recursos, sólo hasta que esta se active.

#### **Tipos de subrutinas en TIA Portal**

En TIA Portal se pueden encontrar dos tipos de subrutinas para los diferentes tipos de PLCs de Siemens, ambas subrutinas son imprescindibles en la programación estructurada de un proyecto. Las dos subrutinas son las denominadas FC y FB (*Figura 3.56*).

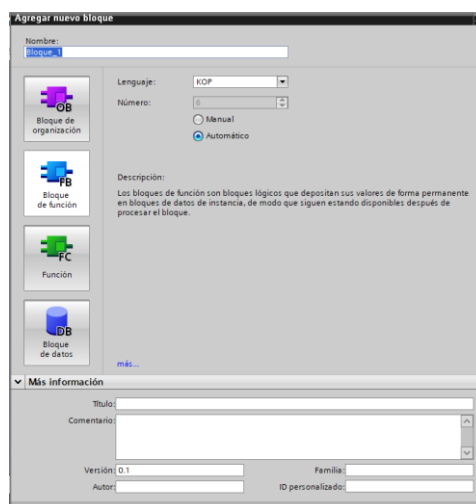


Figura 3.50. Tipos de bloque TIA Portal.

Antes de comenzar a explicar que es un FB y un FC hay un concepto que es importante conocer y es los bloques DB.

### ***¿Qué es un bloque DB en un PLC?***

Un DB es un **Bloque de Datos** dentro del cual solo se puede almacenar datos. Los datos que se almacenan en su interior son de tipo Lectura/Escritura y se puede acceder a estos desde cualquier parte del programa, ya sea por algún bloque o por alguna operación.

Dentro de los DB existen 2 tipos:

- **DB Instancia:** Este tipo de DB está relacionado con un bloque FB y solo desde este bloque se pueden modificar los datos.
- **DB Global:** Los datos que contiene pueden ser tantos escritos como leídos desde cualquier bloque.

### ***¿Qué son los bloques FC y FB?***

Los bloques **FC (Función)** son subrutinas que contienen secuencias específicas. Este bloque se puede llamar desde otro bloque OB, FC o FB. La particularidad del bloque FC es que no contiene almacenamiento de datos cíclicos, es decir, si se realiza una operación dentro del bloque, el valor de resultado solo es útil en la operación, y solo es útil durante la ejecución, y este valor se perderá posteriormente. Cuando se llame al bloque FC la próxima vez, el resultado de la operación anterior ya no existirá.

Un bloque **FB (bloque de funciones)** es una subrutina que contiene una secuencia u operación. Se puede llamar desde otro bloque OB, FC o FB. A diferencia de FC, este bloque tiene asociado un bloque de datos de instancia por defecto, que almacena los valores en el FB, estos datos se denominan estáticos y solo el FB los tiene.

Cuando se llama al bloque FB, ejecutará su rutina y almacenará sus datos en el bloque de datos de instancia relacionado con este FB. Esto significa que, en la siguiente llamada del bloque, dado que los datos anteriores permanecen contenidos estáticamente en el DB asociado, se volverán a cargar los datos anteriores.

### ***¿Cuál es la diferencia entre un bloque FB y un bloque FC?***

La diferencia es que el bloque FC no tiene capacidad para almacenar datos estáticos, es decir, los datos de su funcionamiento son volátiles y se perderán una vez finalizada la ejecución del bloque. Por lo contrario, el bloque FB almacena los datos de su operación en el DB asociado, sus datos no son volátiles y permanecen almacenados incluso después de que se completa la ejecución del bloque.

### 3.5.2. Objetivos

El objetivo de esta práctica es introducir al alumno los diferentes bloques de trabajos que ofrece TIA Portal. Para ello se propone una práctica sencilla y con acciones repetitivas para que se muestre con claridad cómo funcionan estos bloques.

Además de esto también se pretende que el alumno siga progresando con la programación de HMIs.

### 3.5.3. Funcionamiento

La estación a simular en esta práctica se trata de una estación de carga de piezas a diferentes alturas.



Figura 3.57. Vídeo del funcionamiento de la estación del ascensor.

[Vídeo](#)



Figura3.58. Estación del ascensor en Factory IO.

[Factory IO](#)

La estación (*Figura 3.59*) consta de tres partes:

- **Cinta transportadora de palets:** En este caso la cinta transportará una caja contenedora que la llevará al ascensor y ahí se detendrá hasta que la caja esté llena de piezas.
- **Ascensor:** Con él ascenderá o descenderá la caja contenedora para llenarla o vaciarla.
- **Cinta piezas:** Desde cada una de ellas vendrán las piezas que debe cargar.

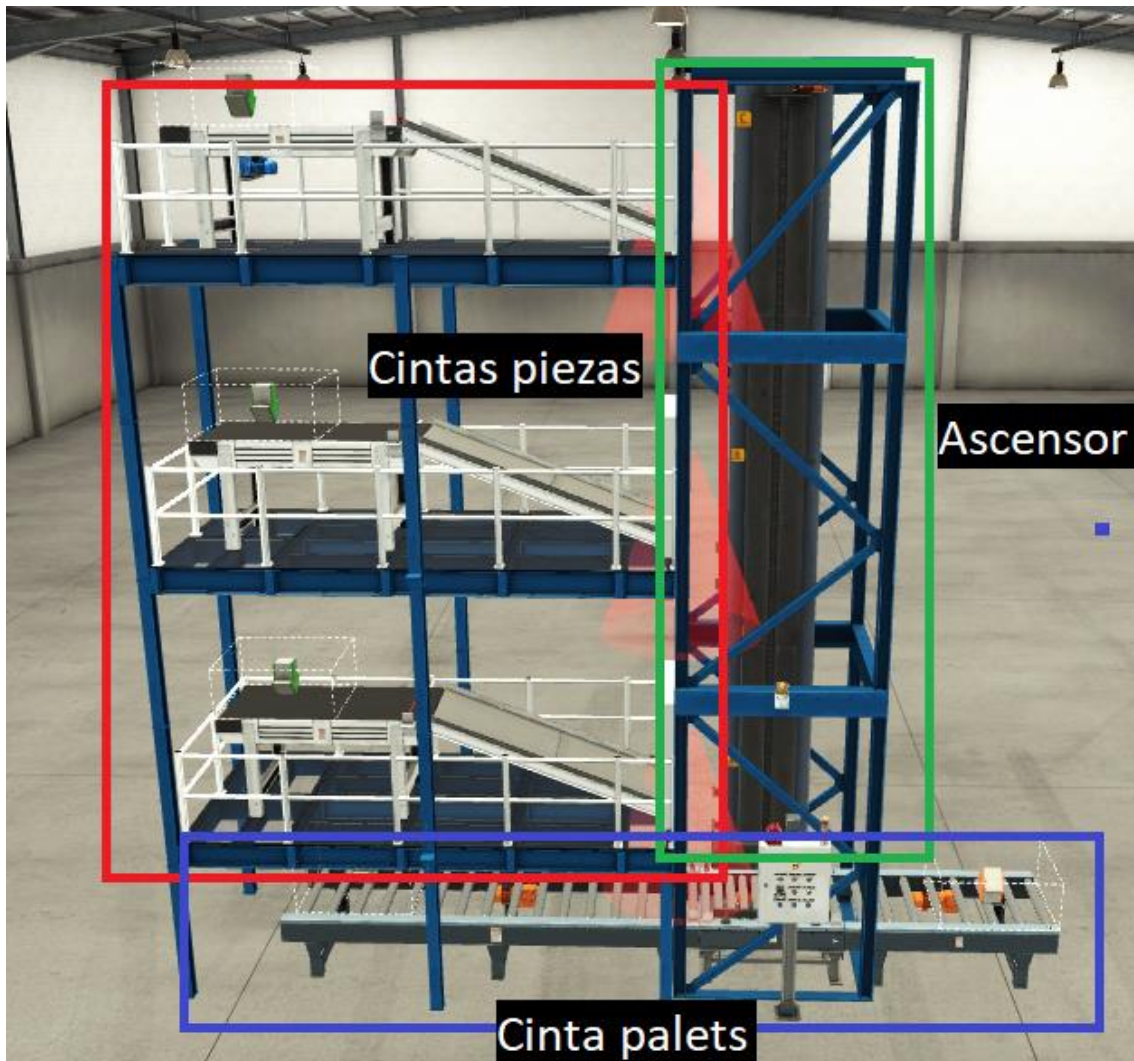


Figura 3.51. Descripción de la estación del ascensor.

### Módulo Ascensor

Aquí está la descripción de las entradas y salidas del ascensor, para esta práctica la cual se ha configurado el módulo en digital.

Nombre de la variable	Controlador I/O	Tipo	Descripción
Elevator # (Up)	Salida	Bool	Mueva la plataforma hacia arriba.
Elevator # (Down)	Salida	Bool	Mueva la plataforma hacia abajo.
Elevator # (Slow)	Salida	Bool	Movimiento lento (20% de la velocidad predeterminada).
Elevator # (+)	Salida	Bool	Ruede (dirección de la flecha).
Elevator # (-)	Salida	Bool	Rodar.
Elevator # (Left Limit)	Entrada	Bool	Elemento detectado.
Elevator # (Right Limit)	Entrada	Bool	Elemento detectado.

Tabla 8. Entradas y salidas del módulo del ascensor.



## Lectura de las piezas mediante un sensor de visión

En esta sesión de prácticas al final de cada cinta transportadora de piezas se ha decidido colocar un sensor de visión para detectar si la pieza ha llegado correctamente a la caja contenedora.

### Sensor visión

El sensor de visión (*Figura 3.60*) reconoce las materias primas, las tapas y las bases de los productos y sus respectivos colores.

- LED: rojo (detectando)
- Materiales detectables: materias primas, bases de productos y tapas de producto.
- Rango de detección: 0,3 - 2 m

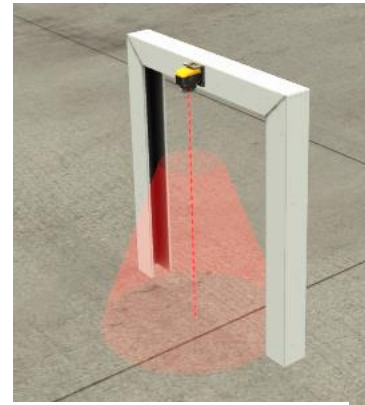


Figura 3.52. Sensor visión.

Este sensor se puede configurar para detectar más de un tipo de pieza seleccionando la configuración adecuada (*Figura 3.61*):

- Todo digital: devuelve cuatro entradas digitales que indican qué elemento se detectó
- Todo numérico: devuelve un valor que codifica el elemento detectado
- Todo ID: devuelve un valor único (aleatorio) que identifica el artículo detectado. Se puede utilizar de forma similar a los lectores de códigos de barras o RFID.
- Bases verdes/azul/metal: se debe configurar el color que se quiere y en el caso de encontrar una base del color elegido nos devolverá un booleano con valor positivo.
- Tapas verdes/azul/metal: se debe configurar el color que se quiera y en el caso de encontrar una tapa del color elegido devolverá un booleano con valor positivo.
- Materia prima verde/azul/metal: se debe configurar el color que se quiera y en el caso de encontrar una materia prima del color elegido devolverá un booleano con valor positivo.

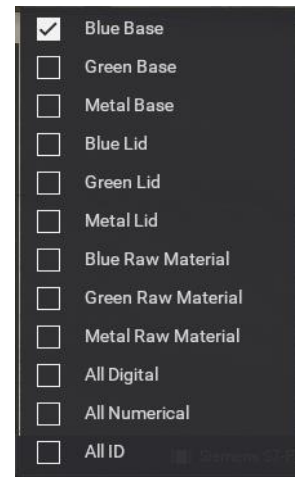


Ilustración 3.53. Tipo de piezas.

Para la estación las piezas con las que se trabajará son bases (*Figura 3.62*), se dispone de tres cintas así que las piezas azules irán a una altura, las verdes a otra y las metálicas a otras. Esto hará que la configuración del sensor sea mucho más fácil ya que se puede configurar en este modo “Bases verde/azul/metal” de esta forma devolverá un booleano si encuentra el color programado.

Práctica 5 – Programación de un ascensor

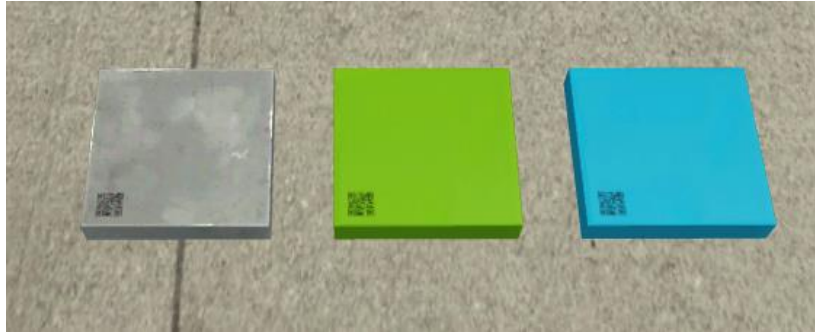


Figura 3.54. Piezas utilizadas en el proceso.

**Función para las cintas**

Como se observa se dispone de tres cintas transportadoras de piezas cuyo funcionamiento es exactamente el mismo, lo único que varía es el color de la pieza y la altura a la que se sitúa cada una de ellas.

Si se programase cada una de ellas en el “main” se observaría que se está repitiendo el código tres veces así que este es un buen caso para usar lo aprendido en esta práctica.

Aquí se tiene un ejemplo de como se vería la función una vez programada (Figura 3.63)

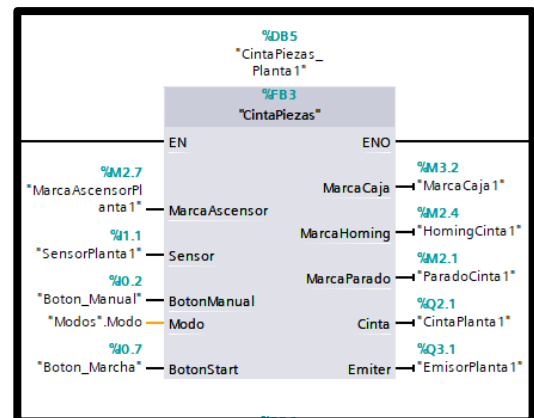


Figura 3.55. Función cinta de las piezas.

Se proponen para esta práctica tres tipos de ejercicios:

- **Nivel básico:** El ascensor recogerá una pieza de cada planta, bajará y avanzará la caja llena. (Lo visto en el vídeo)
- **Nivel medio:** Programación de un HMI donde se vea en tiempo real cómo evoluciona el proceso.
- **Nivel avanzado:** Desde el HMI que se ha programado el operario elegirá la configuración de llenado de la caja contenedora. (La capacidad de la caja es de tres piezas).

### 3.5.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar los conceptos de subrutina, funciones y bloques de datos
3. Graficet con la solución propuesta.

### 3.5.5. En el laboratorio

El alumno debe realizar los ejercicios propuestos anteriormente en KOP y cargarlo tanto en el PLC físico como en el PLCSIM y de esta forma verificar el correcto funcionamiento mediante la estación diseñada en Factory IO.

### 3.5.6. Solución

#### Parte secuencial:

Para el control de esta célula de trabajo se han diseñado tres GRAFCETS que permiten entender de una forma más rápida el funcionamiento de los diferentes procesos.

Estos son los tres GRAFCET en los que hemos dividido el proceso:

- **Transportador de cajas:** Este GRAFCET (Figura 3.64) es el encargado de generar, llevar las cajas hasta el ascensor y finalmente retirarlas una vez están cargadas con las piezas.

Práctica 5 – Programación de un ascensor

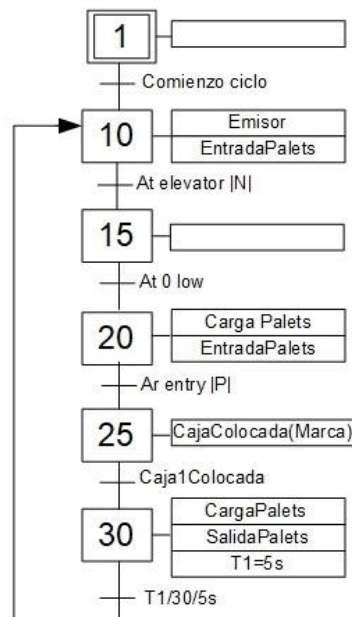


Figura 3.56. GRAFCET transportador de cajas.

- **Cintas piezas:** Este GRAFCET (Figura 3.65) es el encargado de controlar una de las cintas por la que entrarán las diferentes cintas de colores, al ser una acción repetitiva usaremos este GRAFCET en una función.

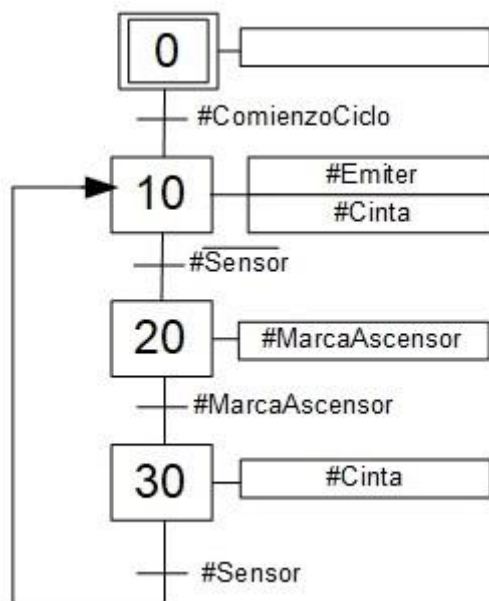


Figura 3.57. GRAFCET cinta piezas.

- **Ascensor:** Finalmente el GRAFCET (Figura 3.66) encargado de controlar el ascensor del proceso, este cogerá la caja vacía en el nivel más bajo e irá subiendo planta a planta recogiendo una pieza de cada color, una vez llena bajará.

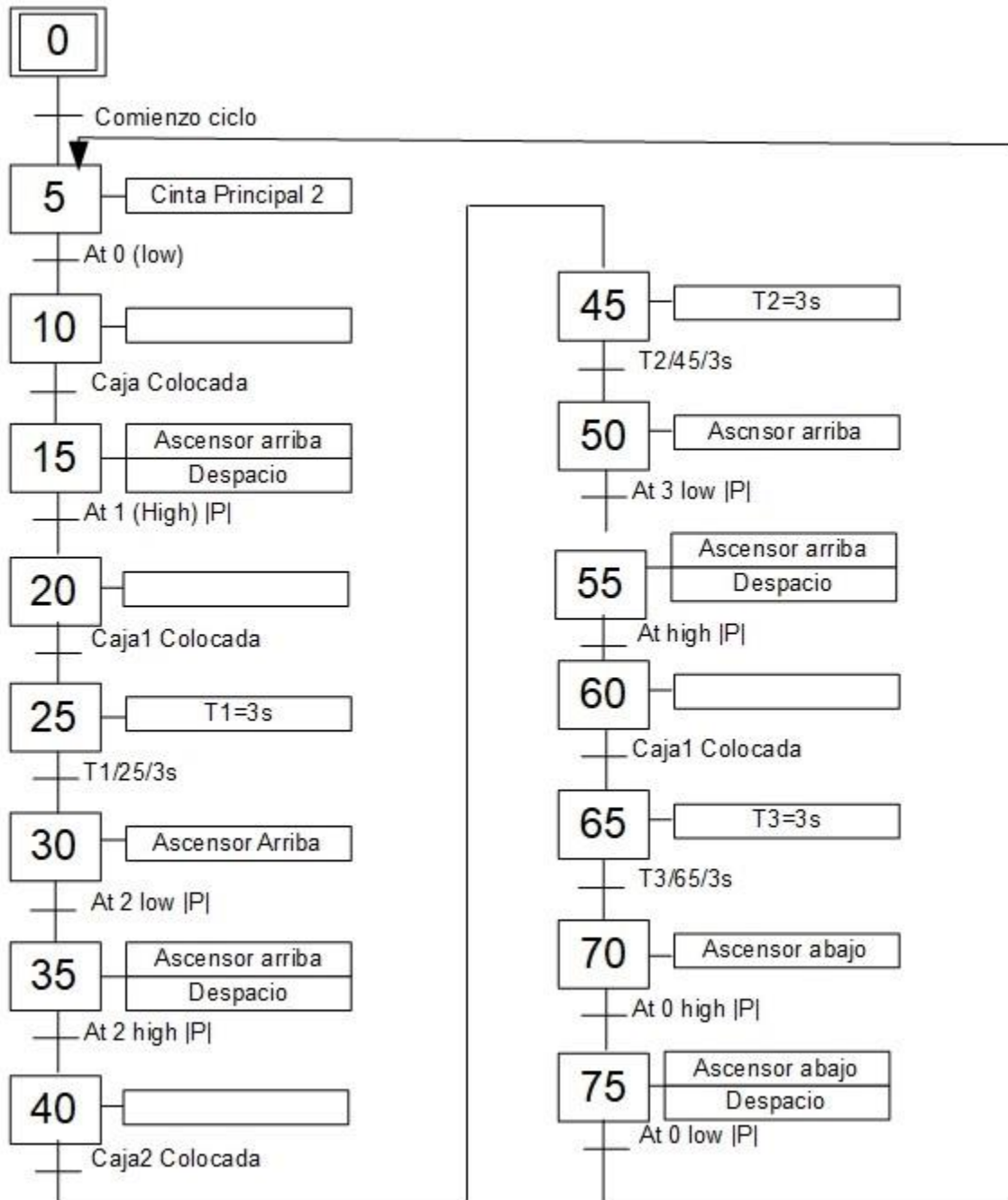


Figura 3.58. GRAFCET ascensor.

Práctica 5 – Programación de un ascensor

**Programación TIA Portal**

La programación de la estación se realizará en lenguaje de contactos también conocido como KOP.

La configuración de entradas y salidas utilizadas en esta práctica es igual a la que se puede visualizar en Factory IO por lo tanto esas variables sumadas a marcas del sistema utilizadas, temporizadores y otros elementos componen todas las variables el proyecto.

Entradas y salidas:

Nombre de la variable	Tipo	Dirección	Nombre de la variable	Tipo	Dirección
<b>Boton_Manual</b>	Bool	%I0.2	Luz_Boton_Home	Bool	%Q0.1
<b>Boton_Confirmar</b>	Bool	%I0.3	Luz_Boton_Manual	Bool	%Q0.2
<b>Selector_Manual</b>	Bool	%I0.4	Luz_Boton_Confirmar	Bool	%Q0.3
<b>Selector_Automatico</b>	Bool	%I0.5	Luz Paro	Bool	%Q0.4
<b>Seta_Emergencia</b>	Bool	%I0.6	Semaforo Verde	Bool	%Q0.5
<b>Boton_Marcha</b>	Bool	%I0.7	Semaforo Rojo	Bool	%Q0.6
<b>Boton_Home</b>	Bool	%I1.0	Semaforo Ámbar	Bool	%Q0.7
<b>SensorPlanta1</b>	Bool	%I1.1	Sirena	Bool	%Q1.0
<b>SensorPlanta2</b>	Bool	%I1.2	Luz_Boton_Marcha	Bool	%Q1.1
<b>SensorPlanta3</b>	Bool	%I1.3	Luz_Primer_Piso	Bool	%Q1.2
<b>At 0 (low)</b>	Bool	%I1.4	Luz_Segundo_Piso	Bool	%Q1.3
<b>At 0 (high)</b>	Bool	%I1.5	Luz_Tercer_Piso	Bool	%Q1.4
<b>At 1 (low)</b>	Bool	%I1.6	Warning light	Bool	%Q1.5
<b>At 1 (high)</b>	Bool	%I1.7	EntradaPalets	Bool	%Q1.6
<b>At 2 (low)</b>	Bool	%I2.0	CargaPalets	Bool	%Q1.7
<b>At 2 (high)</b>	Bool	%I2.1	SalidaPalets	Bool	%Q2.0
<b>At 3 (low)</b>	Bool	%I2.2	CintaPlanta1	Bool	%Q2.1
<b>At 3 (high)</b>	Bool	%I2.3	CintaPlanta2	Bool	%Q2.2
<b>At elevator</b>	Bool	%I2.4	CintaPlanta3	Bool	%Q2.3
<b>At entry</b>	Bool	%I2.5	Ascensor Abajo	Bool	%Q2.4
<b>At exit</b>	Bool	%I2.6	Ascensor arriba	Bool	%Q2.5
<b>Vision Sensor Planta1</b>	Bool	%I2.7	AscensorEspacio	Bool	%Q2.6
<b>Vision Sensor Planta2</b>	Bool	%I3.0	Emisor Palets	Bool	%Q3.0
<b>Vision Sensor Planta3</b>	Bool	%I3.1	EmisorPlanta1	Bool	%Q3.1
<b>FACTORY I/O (Running)</b>	Bool	%I3.3	EmisorPlanta2	Bool	%Q3.2
			EmisorPlanta3	Bool	%Q3.3

Tabla 9. entradas y salidas módulo ascensor.

## Práctica 5 – Programación de un ascensor

Para esta práctica se ha utilizado la siguiente estructura general de bloques (Figura 3.67):

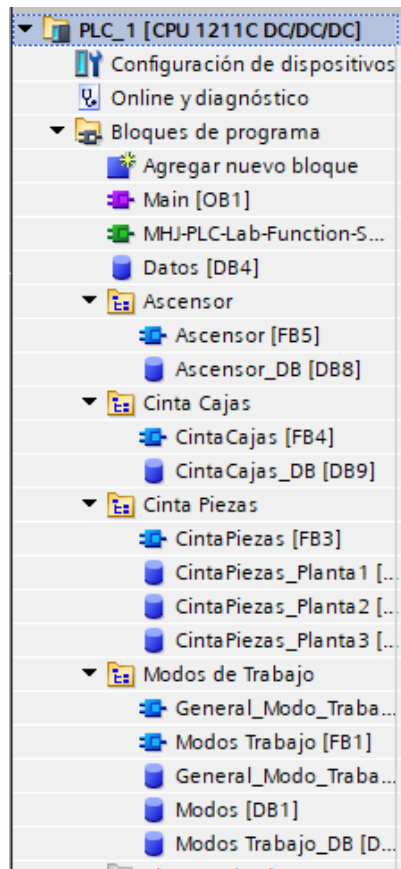


Figura 3.59. Bloques de trabajo.

En esta ocasión se ha optado por un solo bloque de organización el bloque **OB1 (Main)** desde el cual se hará la llamada al resto de bloques que integran el programa.

Cada pequeño subproceso se ha programado utilizando un bloque tipo FB para así tener una vista general y simplificada de como controlar la estación.

- **Modos de trabajo [FB1]:** Este FB es el encargado de gestionar el modo de trabajo de la estación y el cuadro de mandos, incluye modos como manual, automático, homing, emergencia o paro.
- **Ascensor [FB5]:** La función principal de este bloque es la de controlar el movimiento del ascensor y sincronizarlo con el de las cintas y el transportador. Las entradas y salidas que tiene esta función están asociadas a marcas para conocer el estado del proceso o a variables de control de los modos de trabajo. (Figura 3.68)

Práctica 5 – Programación de un ascensor

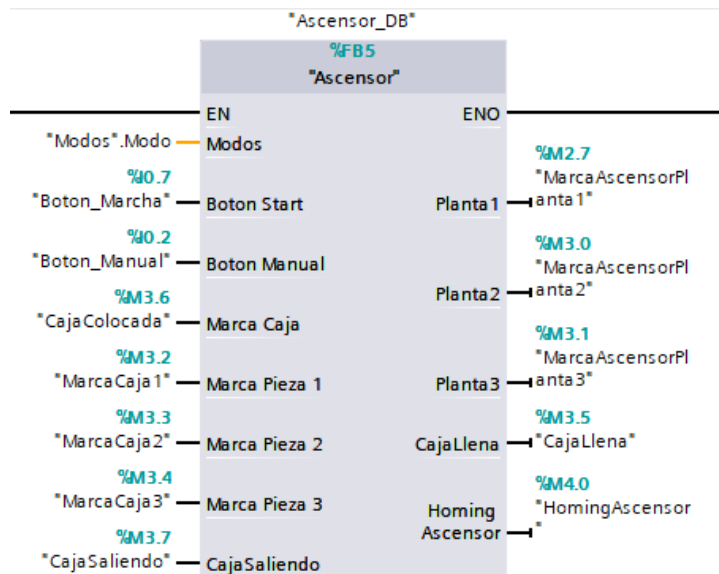


Figura 3.60. Función ascensor.

- **Cintas Cajas [FB4]:** Este bloque es el encargado de llevar las cajas vacías hasta el ascensor y una vez llenas llevarlas a la salida. Al igual que en el anterior FB las entradas y salidas que tiene esta función son marcas del proceso o de los modos de trabajo.
- **Cinta Piezas [FB3]:** Finalmente este bloque se ha programado como una función con la intención de poder usarlo repetido para las tres cintas que componen el proceso ya que todas ellas llevan a cabo una labor repetitiva. (Figura 3.69)

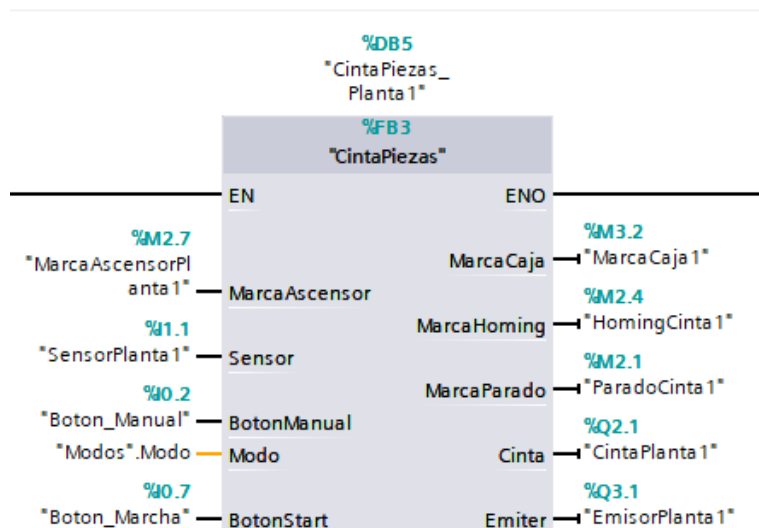


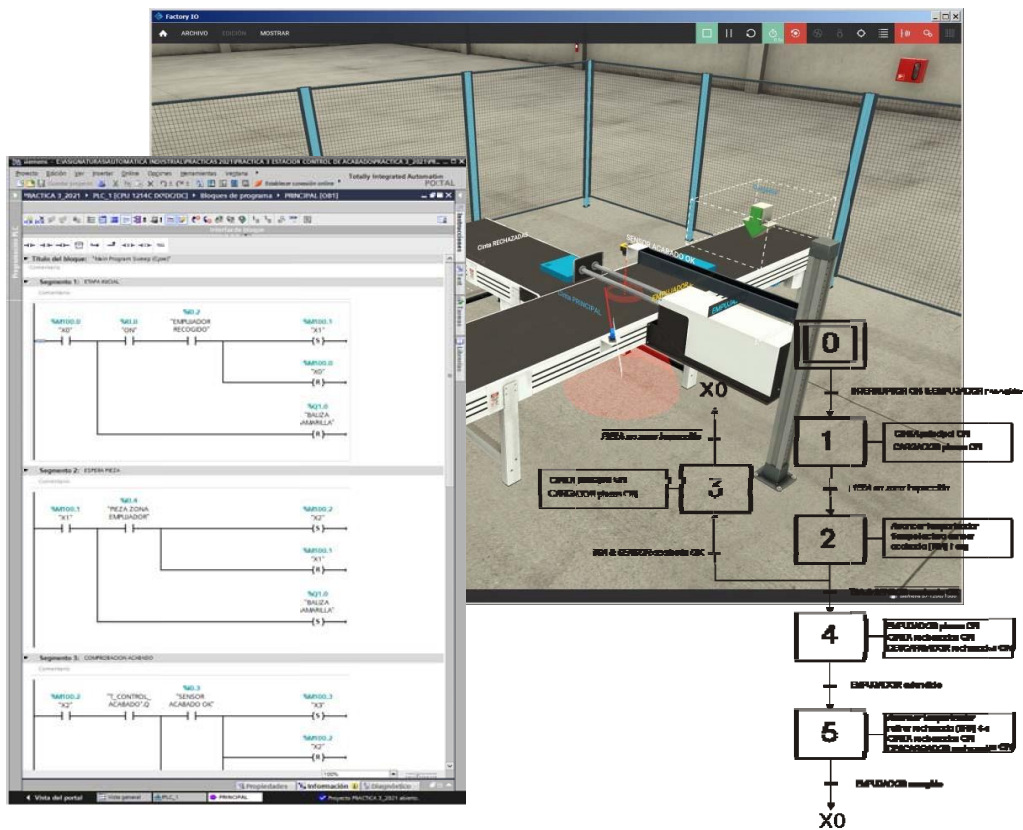
Figura 3.61. Función cintas.



### 3.6. Práctica 6 – Automatización clasificadora de piezas



## AUTOMÁTICA INDUSTRIAL



## PRÁCTICA 6. AUTOMATIZACIÓN CLASIFICADORA DE PIEZAS

ALUMNO:

GRUPO:

### 3.6.1. Introducción

La mayoría de proyectos a nivel industrial están compuestos por pequeños subprocesos que al juntarlos permiten desarrollar un proceso de mayor tamaño por ello es importante saber ligar esos subprocesos de forma correcta para un correcto funcionamiento de la estación.

En las estaciones realizadas anteriormente el proceso giraba sobre un elemento central como por ejemplo el ascensor, el robot cartesiano o el distribuidor. En esta práctica se intentará englobar todas ellas para así trabajar un escenario más cercano a un proceso industria real.

### 3.6.2. Objetivos

En esta estación se va a realizar la automatización de un proceso de clasificación de piezas. El objetivo principal de esta estación es que al tratarse una de las prácticas finales del guion el alumno pueda desarrollar todo lo aprendido hasta ahora con estas prácticas y aplicarlo de forma efectiva.

### 3.6.3. Funcionamiento

La estación que se programará en esta práctica se trata de una estación clasificadora de piezas.



Figura 3.70. Vídeo del funcionamiento de la estación clasificadora.

[Vídeo](#)



Figura 3.71. Estación clasificadora en Factory IO.

[Factory IO](#)

## Práctica 6 – Automatización clasificadora de piezas

Como se puede observar se trata de una estación con varios procesos diferenciados. Esta estación se ha clasificado de la siguiente manera para que sea más fácil su programación:

- **Transportador 1:** En él se encuentra el emisor de piezas, las cuales aparecerán encima de un palet y pueden ser de color verde o azul. Las llevará hasta la entrada del ascensor y allí esperará a que este esté listo (*Figura 3.72*).



Figura 3.62. Transportador 1.

- **Ascensor:** Una vez el palet con las piezas está colocado este coge el palet en el piso de arriba y lo baja al piso inferior (*Figura 3.73*).

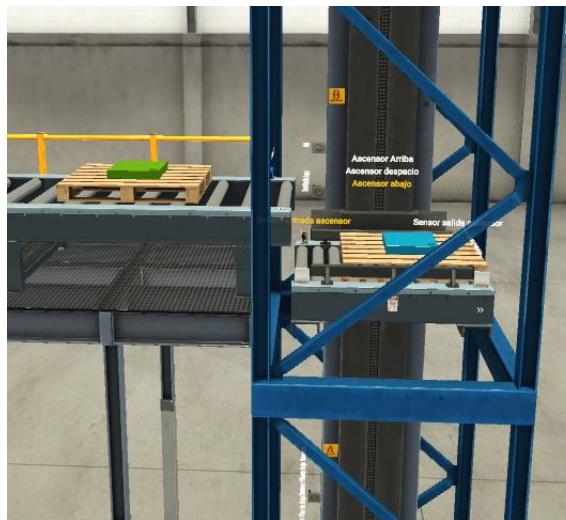


Figura 3.63. Ascensor.

## Práctica 6 – Automatización clasificadora de piezas

- **Cinta 1:** Cuando el ascensor lleve el palet a la planta baja este pasará a la cinta 1, sobre la cual está el sensor de visión. Este nos dará un valor de 1 si la pieza es azul y un 4 si la pieza es verde (*Figura 3.74*).



Figura 3.64. Cinta 1.

- **Cinta 2-3:** Una vez la pieza pasa por el sensor de visión esta avanza hasta la zona de agarre del robot, en la cual espera hasta que la pieza es cogida. Para finalizar el palet vacío avanza a la cinta 3 hasta la rampa de salida del palet (*Figura 3.75*).



Figura 3.65. Cinta 2 y 3.

Práctica 6 – Automatización clasificadora de piezas

- **Robot:** El robot coge la pieza y dependiendo del color de la misma la coloca en una cinta u en otra (Figura 3.76).

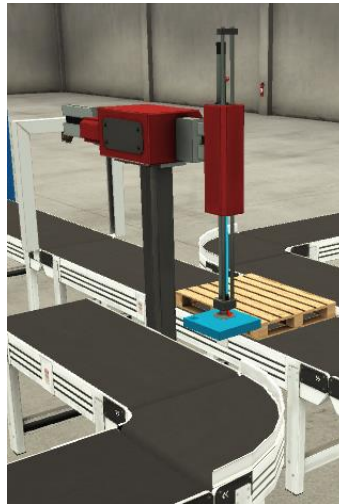


Figura 3.66. Robot.

- **Cintas de colores:** Una vez el robot coge debe colocarla o en la cinta verde o en la cinta de las azules. Estas cintas trasladarán la pieza hasta que desaparezca en la salida.

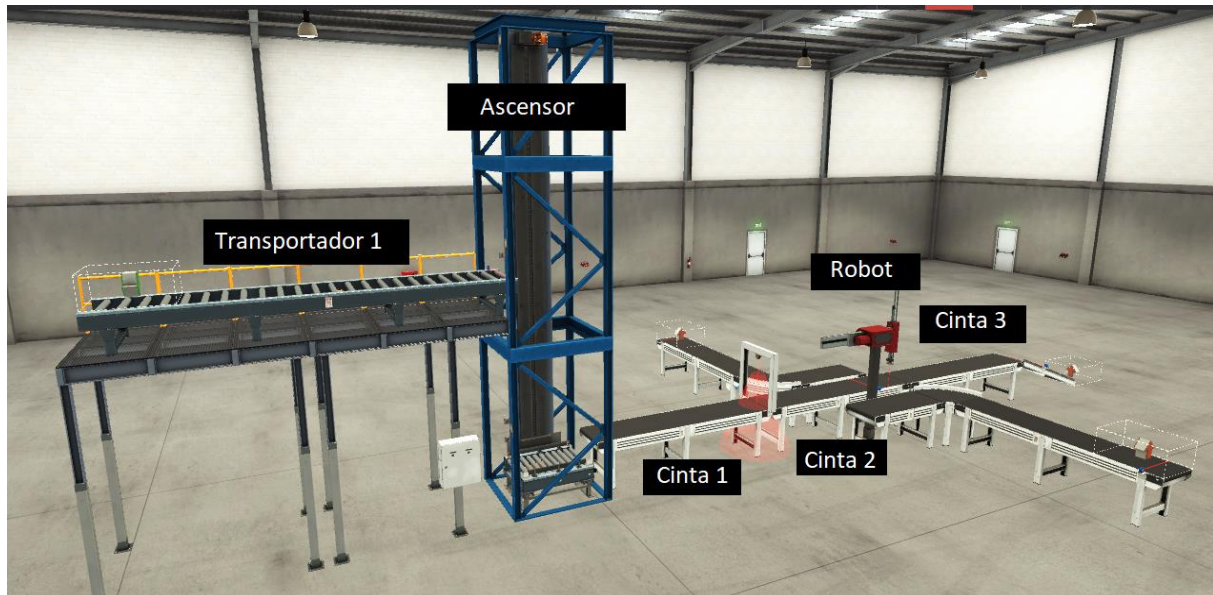


Figura 3.67. Estación clasificadora completa.

Práctica 6 – Automatización clasificadora de piezas

**Robot de dos ejes**

En este módulo se encuentra un robot de dos ejes pick and place, se ha configurado de forma analógica para una mayor precisión de los movimientos, estas son las entradas y salidas que se usarán.

Nombre de la variable	Controlador I/O	Tipo	Descripción
Robot Eje X	Salida	Real	Es el encargado de mover el robot en el eje X.
Robot Eje Z	Salida	Real	Es el encargado de mover el robot en el eje Z.
Robot Giro Antihorario	Salida	Bool	Al activarlo el robot hará un giro antihorario.
Robot Giro Horario	Salida	Bool	Al activarlo el robot hará un giro horario.
Robot Agarre	Salida	Bool	Al activarlo el robot agarrará.
Sensor Robot Eje X	Entrada	Real	Sensor que nos indica en que posición se encuentra el robot en el eje X.
Sensor Robot Eje Z	Entrada	Real	Sensor que nos indica en que posición se encuentra el robot en el eje Z.
Robot Girando	Entrada	Bool	Sensor que nos indica si el robot se encuentra girando.
Robot Agarre	Entrada	Bool	Sensor que nos indica si el robot se encuentra agarrado un objeto.

Tabla 10. Entradas y salidas del robot de dos ejes.

Finalmente, para un correcto funcionamiento de la estación se mostrará el esquema de configuración de los sensores y actuadores que intervienen en la estación (Figura 3.78).

FACTORY I/O (Running)	%I2.0	%Q5.0	Emisor piezas
Sensor T1	%I2.1	%Q5.1	Transportador 1
Sensor Planta Alta	%I2.2	%Q5.2	Ascensor Arriba
Sensor Planta Alta Despacio	%I2.3	%Q5.3	Ascensor abajo
Sensor Planta Baja Despacio	%I2.4	%Q5.4	Ascensor despacio
Sensor Planta Baja	%I2.5	%Q5.5	Transportador ascensor
Sensor entrada ascensor	%I2.6	%Q5.6	Cinta 1
Sensor salida ascensor	%I2.7	%Q5.7	Cinta 2
Sensor C2	%I3.0	%Q6.0	Cinta 3
Sensor C3	%I3.1	%Q6.1	Robot Agarre
Sensor Robot Agarre	%I3.2	%Q6.2	Cinta Azul 1
Sensor Final Verde	%I3.3	%Q6.3	Cinta Azul 2
Sensor final Azul	%I3.4	%Q6.4	Cinta Azul 3
Marcha	%I3.5	%Q6.5	Cinta Verde 1
Paro	%I3.6	%Q6.6	Cinta Verde 2
Robot Girando	%I3.7	%Q6.7	Cinta Verde 3
	%I4.0	%Q7.0	Salida Palets
	%I4.1	%Q7.1	Salida Verde
	%I4.2	%Q7.2	Salida Azul
	%I4.3	%Q7.3	Marcha Luz
Sensor Robot Eje x	%ID30 (REAL)	%Q7.4	Paro Luz
Sensor Robot Eje z	%ID34 (REAL)	%Q7.5	Robot Giro Horario
Sensor C1 (Visión)	%ID38 (DINT)	%Q7.6	Robot Giro Antihorario
	%ID42 (REAL)	%QD30	Robot Eje x
	%ID46 (REAL)	%QD34	Robot Eje z

Figura 3.68. Conexiones Factory IO.

### 3.6.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar los conceptos de subrutina, funciones y bloques de datos
3. Grafcet con la solución propuesta.

### 3.6.5. En el laboratorio

El alumno debe realizar el ejercicio propuesto anteriormente en KOP y cargarlo tanto en el PLC físico como en el PLCSIM y de esta forma verificar el correcto funcionamiento mediante la estación diseñada en Factory IO.

### 3.6.6. Solución

#### Parte secuencial:

Para el control de esta célula de trabajo se han diseñado varios GRAFCETS de control debido a la complejidad que puede tener la estación de esta forma tratamos de simplificar la resolución de la práctica.

Estos son los GRAFCETS en los que se ha dividido el proceso:

- **Transportador 1:** Este GRAFCET es el encargado de generar las piezas y los palets y llevarlos hasta la entrada del ascensor (*Figura 3.79*).

Práctica 6 – Automatización clasificadora de piezas

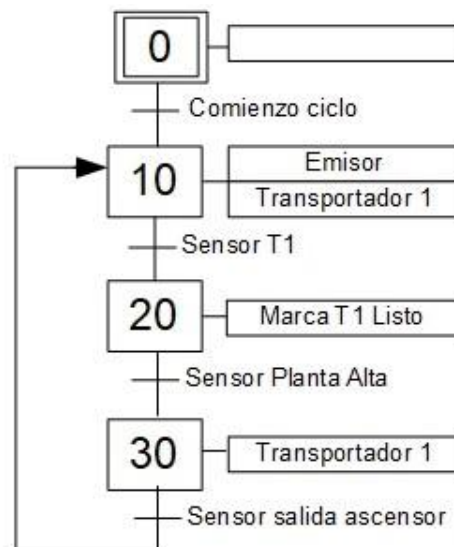


Figura 3.69. GRAFCET Transportador 1.

- **Ascensor:** El control del ascensor se hará siguiendo el GRAFCET que se presenta a continuación (Figura 3.80).

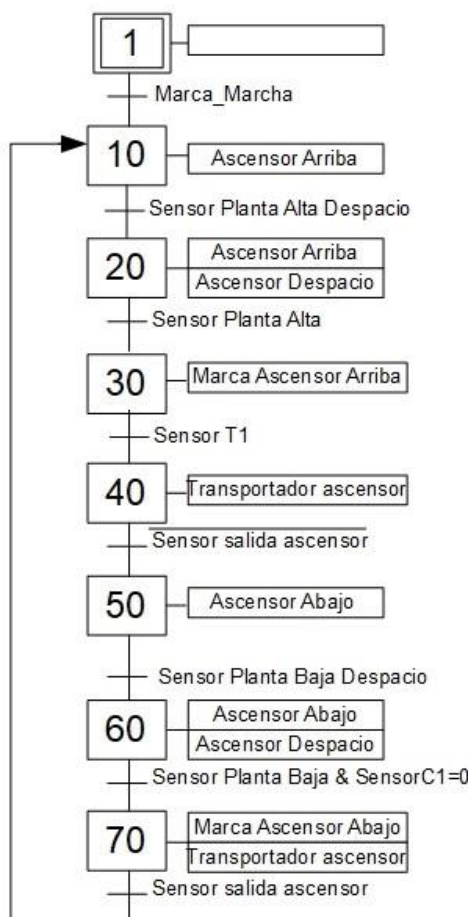


Figura 3.70. GRAFCET Ascensor.



Práctica 6 – Automatización clasificadora de piezas

- **Cinta 1:** Esta cinta es la encargada de llevar los palets con las piezas hasta la zona de carga del robot (*Figura 3.81*).

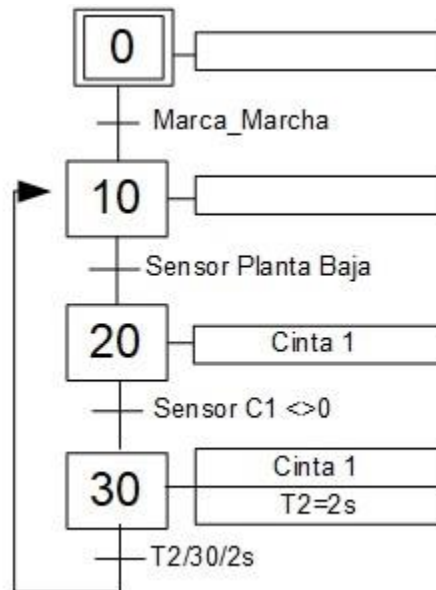


Figura 3.71. GRAFCET cinta 1.

- **Cinta 2-3:** En esta parte del proceso se detectará el tipo de pieza que hay, la llevará hacia el robot y una vez el robot la coge se llevará el palet hasta la salida. (*Figura 3.82*)

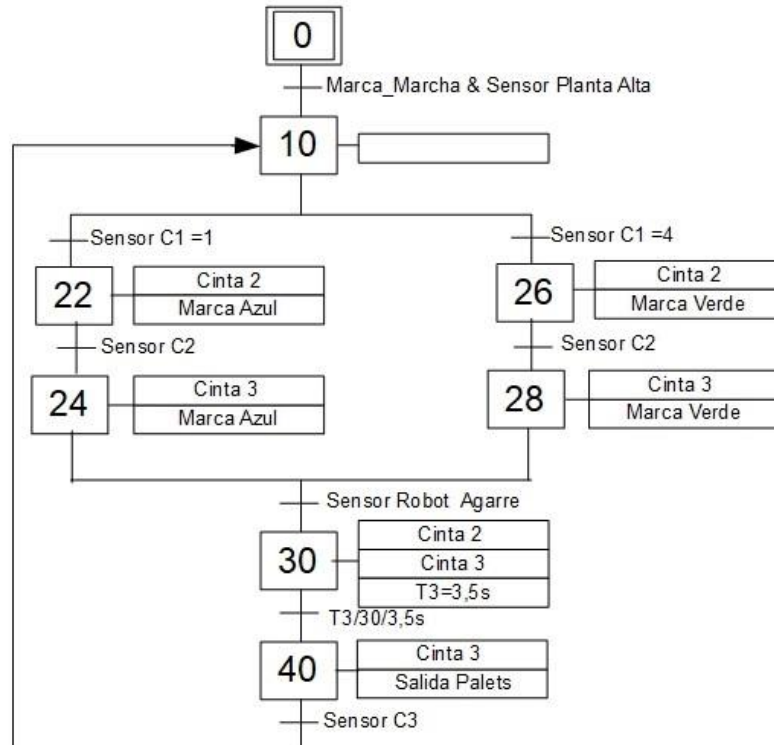


Figura 3.72. GRAFCET cinta 2 y 3.

Práctica 6 – Automatización clasificadora de piezas

- **Robot 2 ejes:** Este GRAFCET es el encargado de controlar el robot al completo cuya función es clasificar las piezas en una cinta o en otra dependiendo del color (Figura 3.83).

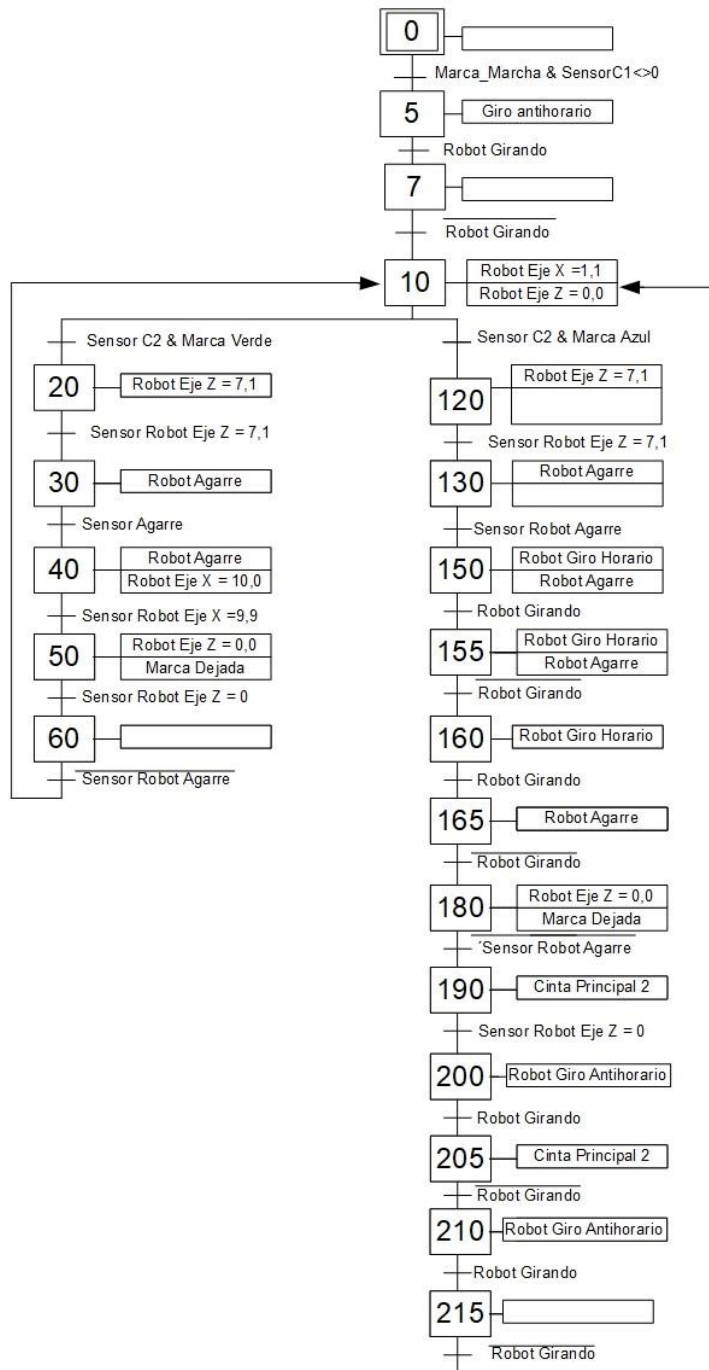


Figura 3.73. GRAFCET Robot 2 ejes.

Práctica 6 – Automatización clasificadora de piezas

- **Cintas Piezas:** Para finalizar el siguiente GRAFCET se usará dos veces en el proceso ya que ambas cintas a los lados del robot trabajan de la misma forma (Figura 3.84).

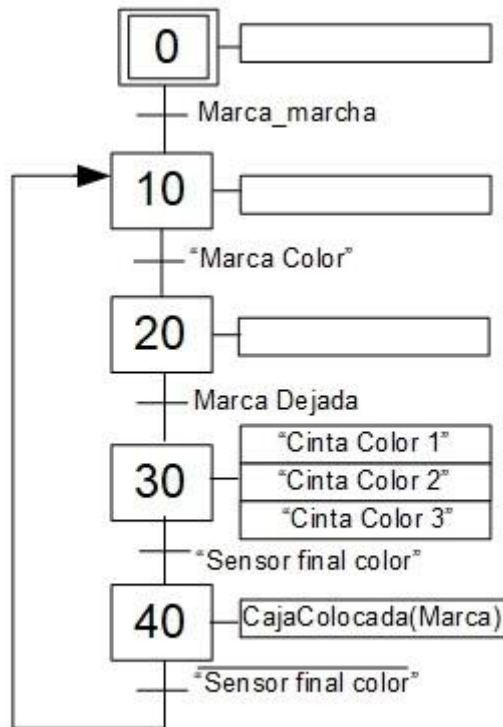


Figura 3.74. GRAFCET cinta piezas.

Nombre de la variable	Tipo	Dirección	Nombre de la variable	Tipo	Dirección
<b>FACTORY I/O (Running)</b>	Bool	%I2.0	Emisor piezas	Bool	%Q5.0
<b>Sensor T1</b>	Bool	%I2.1	Transportador 1	Bool	%Q5.1
<b>Sensor Planta Alta</b>	Bool	%I2.2	Ascensor Arriba	Bool	%Q5.2
<b>Sensor Planta Alta Despacio</b>	Bool	%I2.3	Ascensor abajo	Bool	%Q5.3
<b>Sensor Planta Baja Despacio</b>	Bool	%I2.4	Ascensor despacio	Bool	%Q5.4
<b>Sensor Planta Baja</b>	Bool	%I2.5	Transportador ascensor	Bool	%Q5.5
<b>Sensor entrada ascensor</b>	Bool	%I2.6	Cinta 1	Bool	%Q5.6
<b>Sensor salida ascensor</b>	Bool	%I2.7	Cinta 2	Bool	%Q5.7
<b>Sensor C2</b>	Bool	%I3.0	Cinta 3	Bool	%Q6.0
<b>Sensor C3</b>	Bool	%I3.1	Robot Agarre	Bool	%Q6.1
<b>Sensor Robot Agarre</b>	Bool	%I3.2	Cinta Azul 1	Bool	%Q6.2
<b>Sensor Final Verde</b>	Bool	%I3.3	Cinta Azul 2	Bool	%Q6.3
<b>Sensor final Azul</b>	Bool	%I3.4	Cinta Azul 3	Bool	%Q6.4
<b>Marcha</b>	Bool	%I3.5	Cinta Verde 1	Bool	%Q6.5
<b>Paro</b>	Bool	%I3.6	Cinta Verde 2	Bool	%Q6.6
<b>Robot Girando</b>	Bool	%I3.7	Cinta Verde 3	Bool	%Q6.7
<b>Sensor Robot Eje x</b>	Real	%ID30	Salida Palets	Bool	%Q7.0
<b>Sensor Robot Eje z</b>	Real	%ID34	Salida Verde	Bool	%Q7.1
<b>Sensor C1 (Visión)</b>	DInt	%ID38	Salida Azul	Bool	%Q7.2
			Marcha Luz	Bool	%Q7.3
			Paro Luz	Bool	%Q7.4
			Robot Giro Horario	Bool	%Q7.5
			Robot Giro Antihorario	Bool	%Q7.6
			Robot Eje x	Real	%QD30
			Robot Eje z	Real	%QD34

Tabla 11. Entradas y salidas estación clasificadora.

### Programación TIA Portal

La programación de la estación se realizará en lenguaje de contactos también conocido como KOP.

La configuración de entradas y salidas utilizadas en esta práctica es igual a la que se pueden visualizar en Factory IO por lo tanto esas variables sumadas a marcas del sistema utilizadas, temporizadores y otros elementos componen todas las variables el proyecto.

Entradas y salidas:

Práctica 6 – Automatización clasificadora de piezas

Para esta práctica hemos utilizado la siguiente estructura general de bloques (Figura 3.85):

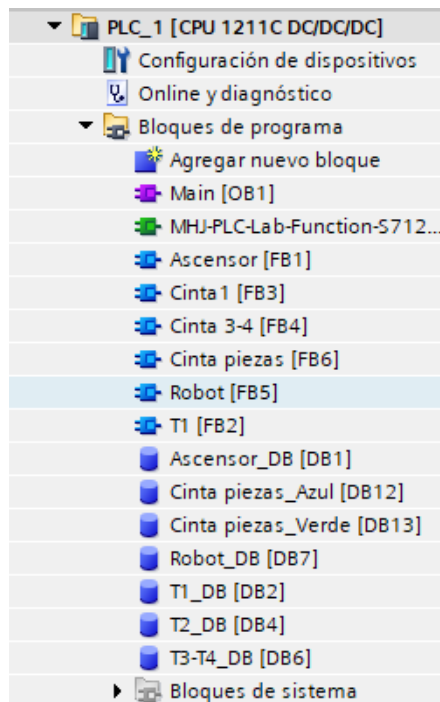


Figura 3.75. Estructura general de bloques.

Se puede observar que se ha programado cada GRAFCET mostrado en la parte secuencial en un FB propio y todos esos FB se han recogido en el bloque de organización **Main [OB1]**.

De estos FB cabe destacar la función de **Cinta piezas [FB6]** (Figura 3.86) la cual se usará para controlar ambas cintas de salida. Así es como quedaría programada con sus entradas y salidas:

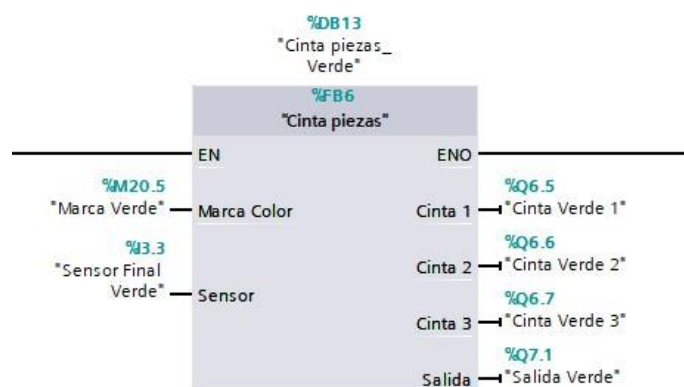


Figura 3.76. Función cinta piezas.

3.7. Práctica 7 – Llenado de un tanque. Control PID



AUTOMÁTICA INDUSTRIAL

The image shows a Factory IO simulation of a tank filling process. On the left, a ladder logic diagram is displayed, divided into three segments: 'PRINCIPAL', 'ZONA PIEZA', and 'COMPLECIÓN ACABADO'. The diagram uses various sensors (NAR100.0 to NAR100.5) and actuators (MOTOR.0, MOTOR.1, MOTOR.2) to control the process. On the right, a 3D rendering of the tank and its components is shown, with a sequence of states (0 to 5) and associated actions like 'REPLLENAR CON EMPUJADOR', 'CERRAR MOTOR', and 'ABRIR MOTOR'.

PRÁCTICA 7. LLENADO DE UN TANQUE. CONTROL PID

ALUMNO:

GRUPO:

### 3.7.1. Introducción

Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. Estas tres acciones son las que dan nombre al controlador PID.

Este tipo de controladores es muy usual utilizarlo en sistemas de bombeo ya que en estos regularmente interesa mantener la presión o flujo constante, por lo tanto, el control PID (*Figura 3.87*) mide la diferencia entre la presión en la tubería y la presión requerida y actúa variando la velocidad del motor para que pueda tener nuestra presión o flujo constante.

El **parámetro Proporcional (P)** mide la diferencia entre el valor actual y el set-point (en porcentaje) y aplica el cambio.

El **parámetro Integral (I)** se refiere al tiempo que se toma para llevar a cabo acción correctiva. Mientras el valor sea más pequeño, el ajuste es más rápido, pero puede causar inestabilidad en el sistema, oscilaciones, vibración de motor y de la bomba.

El **parámetro Derivativo (D)** emite una acción predictiva, es decir, prevé el error e inicia una acción oportuna. Responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande.

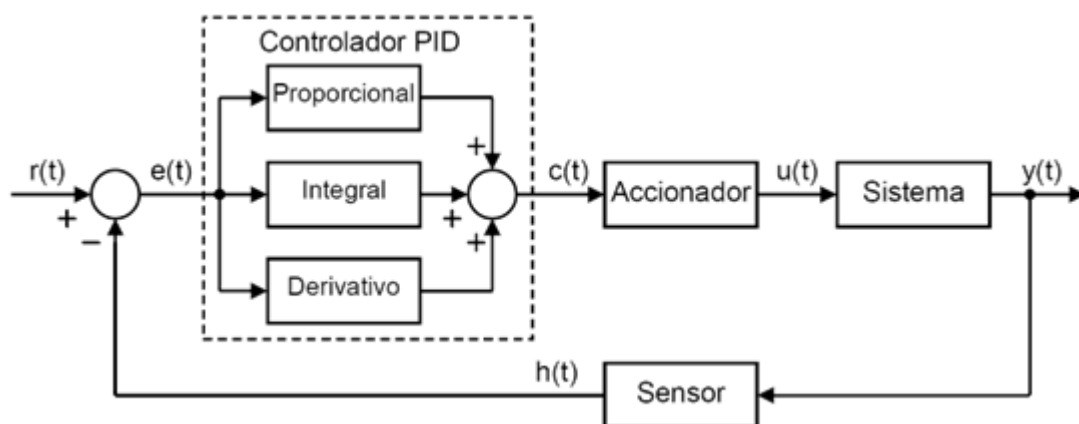


Figura 3.77. PID.

La correcta sintonización o programación de estos parámetros ayuda a controlar de manera efectiva la presión o flujo deseado. Si no se programa adecuadamente estos parámetros, el sistema puede quedar inestable y el motor y la bomba pueden comenzar a vibrar y dañarse.

### 3.7.2. Objetivos

El objetivo de esta práctica es que el alumno aprenda como programar un controlador PID en TIA Portal y que una vez programado pueda verlo actuar de forma práctica gracias al entorno de simulación Factory IO.

### 3.7.3. Funcionamiento

Para esta práctica se utilizará un tanque de agua (*Figura 3.88*) el cual cuenta dos válvulas una de entrada y otra de salida y un sensor de llenado. El objetivo será marcar un SetPoint de llenado (Punto de referencia) del sistema y que de forma automática el nivel del tanque se mantenga a ese SetPoint.



Figura 3.78. Estación de llenado de un tanque.



Práctica 7 – Llenado de un tanque. Control PID

Esto se realizará a través de un controlador PID, para añadir el bloque PID al programa se debe ir a la parte derecha de TIA Portal y pulsar en instrucciones, tecnología y ahí aparecerá el bloque PID\_Compact (Figura 3.89).

Este bloque PID se añade en un bloque de organización “Cyclic interrupt” para que de esta forma el PID esté funcionando de forma constante. Una vez añadido el bloque debe configurarse.

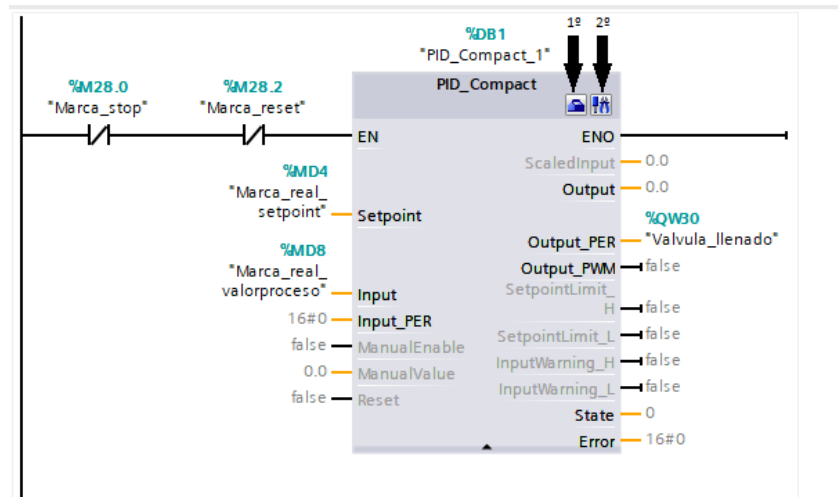
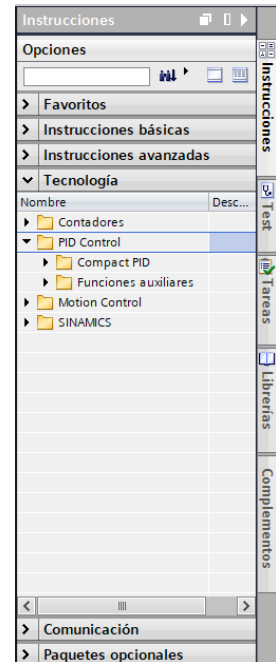


Figura 3.79. Configuración de un PID en TIA Portal.

En la esquina derecha del bloque se observan dos botones. El primero se trata de un botón que lleva a la ventana de configuración del PID a la cual se debe ir antes de ponerlo en marcha y ajustarlo al proceso. El segundo se trata de una ventana para la puesta en servicio, aquí se debe ir la primera vez que se active la estación y ejecutar la puesta en servicio para que el PID se adapte de forma automática al proceso que este controla.

Para esta práctica se presentan tres niveles de dificultad:

- **Nivel básico:** Tanque + Cuadro de mandos (Start – Stop – Apertura válvula salida).
- **Nivel intermedio:** Tanque + Cuadro de mandos (Start – Stop – Apertura válvula salida) + HMI (Visualización y control del proceso).
- **Nivel avanzado:** Tanque + Cuadro de mandos (Con modos de trabajo) + HMI (Visualización y control del proceso) (Figura 3.90).

Práctica 7 – Llenado de un tanque. Control PID



Figura 3.80. Nivel avanzado.



Figura 3.91. Vídeo funcionamiento estación llenado de un tanque.

[Vídeo](#)



Figura 3.92. Estación llenada de un tanque en Factory IO.

[Factory IO](#)

### 3.7.4. Trabajo previo

Antes de la sesión de prácticas el alumno ha de tener preparado una memoria que incluya:

1. Visualizar el vídeo donde se muestra el funcionamiento de la estación.
2. Repasar el modelado de sistemas y el diseño de control PID.

### 3.7.5. En el laboratorio

El alumno deberá crear un programa en KOP capaz de controlar el llenado del tanque propuesto en Factory IO.

Una vez creado cargará el programa al PLC y verificará su correcto funcionamiento.

### 3.7.6. Solución

#### Programación TIA Portal:

La programación de la estación se realizará en lenguaje de contactos también conocido como KOP.

La configuración de entradas y salidas utilizadas en esta práctica es igual a la que se puede visualizar en Factory IO por lo tanto esas variables sumadas a marcas del sistema utilizadas, temporizadores y otros elementos componen todas las variables el proyecto.

Entradas y salidas:

Nombre de las variables	Tipo	Dirección
Boton_start	Bool	%I0.0
Boton_Home	Bool	%I0.1
Boton_Confirmar	Bool	%I0.2
Automatico	Bool	%I0.3
Manual	Bool	%I0.4
Seta_emergencia	Bool	%I0.5
Luz_Boton_start	Bool	%Q0.0
Luz_boton_home	Bool	%Q0.1

Luz_Boton_confirmar	Bool	%Q0.2
Luz_Paro	Bool	%Q0.3
Luz_Modo_Manual	Bool	%Q0.4
Semaforo Verde	Bool	%Q0.6
Semaforo amarillo	Bool	%Q0.7
Semaforo Rojo	Bool	%Q1.0
Sirena	Bool	%Q1.3
Sensor_Llenado	Real	%IW30
Sensor_caudal	Real	%IW32
Regulador	Real	%IW34
Valvula_Llenado	Real	%QW30
Valvula_descarga	Real	%QW32
Display_setpoint	Int	%QW34
Display_valor_actual	Int	%QW36

Tabla 12. Entradas y salidas módulo llenado de un tanque de agua.

Para entender más a fondo la programación de la estación a continuación se muestra la estructura general de bloques (Figura 3.93):

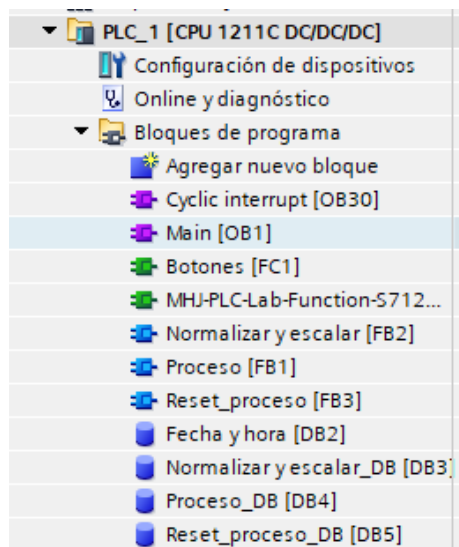


Figura 3.81. Bloques del programa.

En esta estructura general se pueden encontrar dos bloques de tipo organización [OB] son los siguientes:

- **Main [OB1]:** Este bloque es el principal del programa y es el encargado de llamar al resto de bloques,
- **Cyclic interrupt [OB30]:** Se trata de un bloque que se ejecuta periódicamente sin interrupción, Dentro de este bloque es donde se debe colocar la programación de nuestro PID.

## 4. Bibliografía

- Real Games, 2021. **Setting up S7-PLCSIM V13-16 - Factory I/O - Documentation.** Disponible en: <https://docs.factoryio.com/tutorials/siemens/setting-up-s7-plcsim-v13/> [Consultado en febrero de 2021].
- García, D., 2021. **Control PID con autómatas Siemens S7-1200.** Disponible en: <https://www.infoplcn.net/descargas/103-siemens/automatas/s7-1200/2660-siemens-s7-1200-control-pid> [Consultado en marzo 2021].
- Gútiez, I., 2021. **Blog - PROGRAMACIÓN SIEMENS.** Disponible en: <https://programacionsiemens.com/blog/> [Consultado en febrero 2021].
- Industrial online support Siemens. n.d. **Manual SIMATIC STEP 7 Basic/Professional V16.** Disponible en: <https://support.industry.siemens.com/cs/document/109773506/simatic-step-7-basic-professional-v16-y-simatic-wincc-v16> [Consultado en febrero 2021].