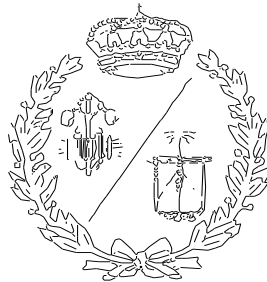


**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Grado

**SIMULACIÓN DE UN SISTEMA INDUSTRIAL DE
CLASIFICACIÓN, DISTRIBUCIÓN Y PALETIZADO
DE PIEZAS
(SIMULATION OF AN INDUSTRIAL SYSTEM FOR
SORTING, DISTRIBUTION AND PALLETISATION
OF PARTS)**

Para acceder al Título de

**GRADUADA EN INGENIERÍA ELECTRÓNICA
INDUSTRIAL Y AUTOMÁTICA**

Autor: Noelia Conde Celis

Septiembre – 2022

Agradecimiento,

A mi familia ,

por el apoyo durante la realización del trabajo,

Y por empujarme a confiar siempre en mí misma.

RESUMEN

En el presente proyecto se realiza la simulación de un proceso industrial de clasificación, distribución y paletizado de piezas. El proceso se puede dividir en tres zonas para facilitar su explicación.

En la primera zona se dispondrá de una cinta central donde se emitirán las tres clases de piezas que serán trasladadas por cintas transportadoras para su selección y clasificación en función de su color. Su desplazamiento hacia la segunda sección será llevado a cabo por los dispositivos de salida hacia las tres ramificaciones de almacenamiento y posterior paletizado.

Esta segunda sección, es la correspondiente al área de almacenamiento de las piezas en las diferentes cajas en función de su tamaño. Esta zona de almacenamiento está formada por una máquina secuencial que meterá las piezas en su caja correspondiente hasta que esta quede completa.

Finalmente, cuando se ha introducido un número determinado de productos en las cajas, y estas están completas, salen hacia la última zona del sistema, el área de paletizado, donde por medio de tres células robóticas se paletizan las cajas en función de su tamaño siguiendo un patrón de dos alturas en un pallet para su posterior retirada por el operario.

ABSTRACT

This project simulates an industrial process for the classification, distribution and palletisation of parts. The process can be divided into three zones to facilitate its explanation.

In the first zone there will be a central belt where the three types of parts will be issued and moved by conveyor belts for their selection and classification according to their colour. Their movement towards the second section will be carried out by the exit devices towards the three storage and subsequent palletising branches.

This second section corresponds to the area for storing the pieces in the different boxes according to their size. This storage area is made up of a sequential machine that places the parts in their corresponding box until the box is full.

Finally, when a certain number of products have been placed in the boxes, and these are complete, they leave for the last area of the system, the palletising area, where, by means of three robotic cells, the boxes are palletised according to their size, following a pattern of two heights on a pallet for subsequent removal by the operator.

ÍNDICE GENERAL:

DOCUMENTO 1: MEMORIA

DOCUMENTO 2: PLIEGO DE CONDICIONE

DOCUMENTO 3: PRESUPUESTO

DOCUMENTO 4: ANEXOS

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Tipos de Robots industriales.....	24
Ilustración 2: Interfaz Gráfica de TIA Portal	26
Ilustración 3: Primeros Pasos en TIA Portal	26
Ilustración 4: Declaración de Variables en TIA Portal	27
Ilustración 5: Tipos de Bloques en TIA Portal	30
Ilustración 6: Sistema Modo Online en TIA Portal	30
Ilustración 7: Interfaz Gráfica de Factory I/O	31
Ilustración 8: Opciones de Factory I/O	33
Ilustración 9: Navegación en Factory I/O	34
Ilustración 10: Items de Factory I/O	35
Ilustración 11: Elementos de carga pesada de Factory I/O	35
Ilustración 12: Elementos de carga Ligera de Factory I/O	36
Ilustración 13: Sensores de Factory I/O	36
Ilustración 14: Operadores de Factory I/O	37
Ilustración 15: Estaciones de Factory I/O	37
Ilustración 16: Elementos de alarma de Factory I/O	38
Ilustración 17: Pasarelas de Factory I/O	38
Ilustración 18: Introducción y Extracción en Factory I/O	39
Ilustración 19: Posibles escenas de Factory I/O	40
Ilustración 20: Drivers de Factory I/O	40
Ilustración 21: Interfaz Gráfica Robot Studio	41
Ilustración 22: Ejemplo de programa RAPID	42
Ilustración 23: Pantalla de Posición Inicial de Robot Studio	43
Ilustración 24: Pantalla de Modelado de Robot Studio	43
Ilustración 25: Pantalla de Simulación de Robot Studio	44

Ilustración 26: Pantalla de controlador en Robot Studio	44
Ilustración 27: Pantalla RAPID en Robot Studio	45
Ilustración 28: Pantalla Complementos en Robot Studio	46
Ilustración 29: Modelo CPU 1214C DC/DC/DC	49
Ilustración 30: Vista general de la fábrica en Factory I/O.....	50
Ilustración 31: Zona de emisión y detección del color.....	51
Ilustración 32: Ramificaciones según el color.....	51
Ilustración 33: Ramificación gris	52
Ilustración 34: Ramificación verde	53
Ilustración 35: Ramificación azul	53
Ilustración 36: Pick and place Cajas Pequeñas.....	54
Ilustración 37: Pick and place Cajas Medianas	55
Ilustración 38: Pick and place Cajas Grandes	55
Ilustración 39: Salida zona Factory I/O	56
Ilustración 40: Salida zona RobotStudio	57
Ilustración 41: Función MHJ-PLC-lab-Function-S71200	58
Ilustración 42: Conexión de Factory I/O con PLCSIM	59
Ilustración 43: Pantalla principal del proyecto en TIA Portal	67
Ilustración 44: Carpetas para la estructuración del proyecto	68
Ilustración 45: Bloque principal OB1	69
Ilustración 46: Bloques del sistema.....	70
Ilustración 47: Sensor de visión	71
Ilustración 48: Conveyor _emisor (1)	72
Ilustración 49: Conveyor _emisor (2)	72
Ilustración 50: Conveyor _emisor (3)	73
Ilustración 51: Conveyor _derivador (1)	74

Ilustración 52: Conveyor_derivador (2)	75
Ilustración 53: Conveyor_derivador (3)	75
Ilustración 54: Conveyor_derivador (4)	76
Ilustración 55: Conveyor_derivador (5)	77
Ilustración 56: Conveyor_derivador (6 y 7)	77
Ilustración 57: Conveyor_derivador (8)	78
Ilustración 58: Estructura del DB inicio de cadena	79
Ilustración 59: Ramificación azul conveyor 7	80
Ilustración 60: Ramificación azul conveyor 8	80
Ilustración 61: Ramificación azul conveyor 9	81
Ilustración 62: Ramificación azul conveyor Final 10	81
Ilustración 63: Ramificación gris conveyor 11	82
Ilustración 64: Ramificación gris conveyor 12	83
Ilustración 65: Ramificación gris conveyor 13	83
Ilustración 66: Ramificación gris conveyor Final 14	84
Ilustración 67: Ramificación verde conveyor 3	85
Ilustración 68: Ramificación verde conveyor 4	85
Ilustración 69: Ramificación verde conveyor 5	86
Ilustración 70: Ramificación verde conveyor Final 6	86
Ilustración 71: Cajas Grandes _FC (1)	87
Ilustración 72: Cajas Grandes _FC (2)	88
Ilustración 73: Cajas Grandes _FC (3)	88
Ilustración 74: Cajas Grandes _FC (4)	89
Ilustración 75: Cajas Grandes _FC (5)	89
Ilustración 76: Cajas Grandes _FC (6)	90
Ilustración 77: Cajas Grandes _FC (7)	90
Ilustración 78: Cajas Grandes _FC (8)	91

Ilustración 79: Cajas Grandes _FC (9)	91
Ilustración 80: Interfaz de Bloque FB del Robot.....	92
Ilustración 81: Cajas Grandes_FB (1)	93
Ilustración 82: Cajas Grandes_FB (2)	93
Ilustración 83: Cajas Grandes_FB (3)	94
Ilustración 84: Cajas Grandes_FB (4)	94
Ilustración 85: Cajas Grandes_FB (5)	95
Ilustración 86: Cajas Grandes_FB (6)	95
Ilustración 87: Cajas Grandes_FB (7)	96
Ilustración 88: Cajas Grandes_FB (8)	96
Ilustración 89: Cajas Grandes_FB (9)	97
Ilustración 90: Cajas Grandes_FB (10)	97
Ilustración 91: Segmentos de la Baliza	98
Ilustración 92: Segmentos de los Pulsadores	98
Ilustración 93: Detección de fallos HMI	99
Ilustración 94: Traspaso de fallos HMI.....	100
Ilustración 95: Borrado de fallos HMI.....	100
Ilustración 96: Lectura del sensor en el HMI	101
Ilustración 97: Emergencias (1)	103
Ilustración 98: Emergencias (2)	103
Ilustración 99: Emergencias (3)	104
Ilustración 100: MHJ-PLC-lab-Function_S71200.....	105
Ilustración 101: Interfaz del bloque FB1	106
Ilustración 102: Conveyor FB1(1)	106
Ilustración 103: Conveyor FB1(2)	107
Ilustración 104: Conveyor FB1(3)	107

Ilustración 105: Interfaz de bloque FB2	108
Ilustración 106: Conveyor FB2 (1)	108
Ilustración 107: Conveyor FB2 (2)	109
Ilustración 108: Conveyor FB2 (3)	109
Ilustración 109: Conveyor FB2 (4)	110
Ilustración 110: Dispositivo KTP700 Basic PN	111
Ilustración 111: Conexión del PLC con el HMI	111
Ilustración 112: Plantilla 1.....	113
Ilustración 113: Plantilla 2	113
Ilustración 114: Pantalla de inicio.....	114
Ilustración 115: Pantalla de inicio Línea	115
Ilustración 116: Pantalla Rama Gris	116
Ilustración 117: Pantalla Rama Verde	117
Ilustración 118: Pantalla Rama Azul	118
Ilustración 119: IRB-1600 de ABB	120
Ilustración 120: IRC5 de ABB	122
Ilustración 121: IRC5 Flex pendant	123
Ilustración 122: Ejemplo de conexionado de estación	123
Ilustración 123: Herramienta Cajas Pequeñas.....	125
Ilustración 124: Herramienta Cajas Medianas	125
Ilustración 125: Herramienta Cajas Grandes	125
Ilustración 126: Menú diseño RobotStudio.....	127
Ilustración 127: Componente Inteligente 1	128
Ilustración 128: Componente Inteligente 2	128
Ilustración 129: Componente Inteligente 3	129
Ilustración 130: Conjunto de colisiones	129
Ilustración 131: Lógica de estación	130

Ilustración 132: Estructura RAPID	130
Ilustración 133: Variables y Constantes RAPID	131
Ilustración 134: MAIN	131
Ilustración 135: Programa cajas grandes (1)	132
Ilustración 136: Programa cajas grandes (2)	132
Ilustración 137: Recoger	133
Ilustración 138: Dejar	133
Ilustración 139: Salida pallet (1)	134
Ilustración 140: Salida pallet (2)	134

DOCUMENTO 1. MEMORIA

ÍNDICE DOCUMENTO 1:

1. INTRODUCCIÓN	17
1.1 MOTIVACIÓN.....	17
1.2 OBJETIVOS.....	17
1.3 ALCANCE.....	18
2. DESCRIPCIÓN DEL SISTEMA	18
3. CONOCIMIENTOS PREVIOS	20
3.1 AUTOMATIZACIÓN.....	20
3.2 AUTÓMATAS PROGRAMABLES.....	21
3.2.1 Definición.....	21
3.2.2 Tipos de autómatas.....	21
3.2.3 Programación de un autómata.....	22
3.3 TIPOS DE ROBOTS INDUSTRIALES.....	23
3.4 TIA PORTAL.....	24
3.4.1 Historia de TIA Portal.....	24
3.4.2 Software en TIA Portal.....	25
3.4.3 Características Básicas del TIA Portal.....	25
3.4.3.1 Gestión de datos.....	27
3.4.3.2 Lenguaje de programación.....	28
3.4.3.3 Funciones de programación en TIA Portal.....	28
3.4.3.4 Funcionamiento Online.....	30
3.4.3.5 Librerías en TIA Portal.....	31
3.5 FACTORY I/O.....	31
3.5.1 Historia de Factory I/O.....	31
3.5.2 Elementos del Factory I/O.....	32
3.5.2.1 Barra de herramientas.....	32
3.5.2.2 Opciones	33

3.5.2.3 Navegación	34
3.5.2.4 Elementos	35
3.5.2.5 Etiquetas	39
3.5.2.6 Escenas	40
3.5.2.7 Controles de entrada/salida	40
3.6 ROBOTSTUDIO.....	41
3.6.1 Historia de RobotStudio	41
3.6.2 Lenguaje Rapid	41
3.6.3 Partes Básicas de RobotStudio	42
4.SISTEMA VIRTUAL DE DISTRUCIÓN Y CLASIFICACIÓN	46
4.1 EQUIPOS DE LA INSTALACIÓN.....	46
4.2 ELEMENTOS DE ENTRADA.....	46
4.3 ELEMENTOS DE SALIDA.....	47
4.4 OTROS.....	48
4.5 ELECCION DEL PLC.....	49
4.6 EMISIÓN DE PIEZAS Y DIVISIÓN DEL COLOR.....	50
4.7 ZONA TRANSPORTE	52
4.8 ZONA DE PICK AND PLACE	54
4.9 ZONA DE SALIDA.....	56
4.10 CONEXIÓN DE SOFTWARE.....	57
4.11 CONFIGURACIÓN FACTORY I/O PARA LA CONEXIÓN...	58
5. PROGRAMACIÓN DEL PLC	59
5.1 CICLO DE FUNCIONAMIENTO.....	59
5.2 SECUENCIAS DE PROGRAMACIÓN.....	60
5.3 VARIABLES DEL SISTEMA.....	61
5.4 BLOQUES Y FUNCIONES DEL PROGRAMA.....	67

5.4.1 Sistema	69
5.4.2 Sensor de visión.....	70
5.4.3 Inicio cadena de producción	71
5.4.4 Ramificaciones	79
5.4.4.1 Pieza azul	79
5.4.4.2 Pieza gris	82
5.4.4.3 Pieza verde	84
5.4.5 Cintas cajas	87
5.4.5.1 Cajas grandes.....	87
5.4.6 Señalización.....	98
5.4.7 Supervisión pantalla HMI.....	99
5.4.8 Emergencias	101
5.4.9 Bloques auxiliares.....	104
6. SISTEMA DE SUPERVISIÓN	110
6.1 PANTALLA HMI	110
6.2 PLANTILLAS	112
6.3 PANTALLA INICIO	114
6.4 PANTALLA INICIO DE LÍNEA	115
6.5 SUPERVISIÓN RAMIFICACIONES	116
6.5.1 Rama gris	116
6.5.2 Rama verde	117
6.5.3 Rama azul	117
7. CÉLULA DE PALETIZADO	119
7.1 ELECCIÓN DEL ROBOT	120
7.2 UNIDAD DE CONTROL	121
7.3 HERRAMIENTA DEL ROBOT	124
7.4 PROGRAMACIÓN DEL ROBOT	126

8. LÍNEAS FUTURAS	135
9. CONCLUSIONES.....	136
10. REFERENCIAS	138

1.INTRODUCCIÓN

1.1 MOTIVACIÓN

El sector de la automatización industrial se encuentra en constante crecimiento y evolución gracias al avance de las tecnologías digitales y de la robótica. Su uso en procesos de producción está muy dilatado en el mercado ya que su accesibilidad y especialización son cada vez más adecuadas al medio.

Del mismo modo se ha producido una evolución en el apartado de las comunicaciones permitiendo cada vez una mayor interacción y extracción de información de cada proceso, además de una mayor velocidad de transmisión y variedad de vías de comunicación.

Por todo esto, el proyecto a desarrollar plantea la ejecución y desarrollo de una fábrica para la clasificación, distribución y paletizado de piezas en un simulador, y empleando tres softwares con gran importancia a nivel industrial, pasando por su diseño, programación, control y comunicación.

TIA Portal y Factory I/O serán los softwares encargados de la programación y simulación para la primera parte del proyecto, enlazándose entre sí mediante una función de comunicaciones, mientras que RobotStudio será el encargado de desarrollar la parte de paletizado del sistema.

1.2 OBJETIVOS

Los objetivos para desarrollar en esta memoria son los siguientes:

- Poner en práctica los conocimientos de programación en el lenguaje de contactos, así como el uso del software TIA Portal y elevar el conocimiento a nivel de uso industrial.
- Introducción a las comunicaciones industriales vía HMI o internet, estudiando su correcta implementación
- Exploración y aprendizaje del uso del software Factory I/O y su implementación para la simulación de procesos reales.
- Elevar el conocimiento en Lenguaje RAPID y avanzar en la implementación con el software RobotStudio a nivel industrial.

-El acercamiento al ámbito industrial a través del diseño de secciones de una fábrica.

1.3 ALCANCE

El alcance de este proyecto comienza con el estudio de los elementos que ofrece el software Factory I/O, y la introducción de estos en este trabajo.

Para ello se realiza una revisión de los diferentes elementos de un sistema de clasificación y distribución de piezas. A continuación, se diseñará la escena requerida con los elementos escogidos en el proyecto y proporcionados por Factory I/O. Además, se hará elección del PLC a usar en función de las necesidades requeridas.

Una vez escogido el PLC, se lleva a cabo el desarrollo de la programación en TIA Portal. De forma adicional y para una mejor supervisión, se añadirá al sistema el control de este mediante una pantalla HMI, en la cual, permitirá interactuar con el sistema a través de cualquiera de los botones del PLC, así como ver cada una de las partes de la estructura que se ha diseñado y desarrollado para el proyecto.

Finalmente, se estudiará una célula de paletizado. Las células de paletizado son sistemas robóticos que permiten automatizar este proceso y ayudan a que este se haga de forma más rápida y productiva. Se realizará con un diseño lo más similar posible al previamente desarrollado en Factory I/O, para que ambas partes tengan una correlación, ya que, se realizaran dos simulaciones debido a la incompatibilidad de enlace entre Factory I/O Y RobotStudio. Se programa el controlador elegido en RobotStudio con el lenguaje RAPID necesario para llegar a obtener el perfecto funcionamiento del sistema.

2. DESCRIPCIÓN DEL SISTEMA

El sistema de clasificación, distribución y paletizado se encuentra diseñado de la siguiente manera:

El ciclo de funcionamiento del sistema comienza con un dispositivo emisor, que simula la llegada de piezas cuadradas de tres colores distintos desde otro lugar. Una vez salen las piezas de la cinta emisora, se activará el sensor de visión, que se encuentra situado encima de una estructura fija a media distancia de la zona emisora de piezas. El sensor de visión detectará el color de la pieza en función de los 3 tipos que puede sacar el emisor y junto a los cilindros neumáticos, que se encuentran situados a ambos lados de la cinta, se

procederá a su clasificación y desplazamiento hacia la ramificación correspondiente en función del color de la pieza detectada.

Si la pieza emitida es azul, esta continuará su trayecto a través de las cintas transportadoras, hasta llegar a la cinta transportadora de la zona de derivación. Una vez allí se activará el segundo cilindro neumático trasladando la pieza hasta otra cinta, donde, continuará su trayecto hasta llegar a la zona final. Una vez situada la pieza en la zona final, un "pick and place", que se trata de una maquina secuencial que permitirá la introducción de la pieza en la caja, y un sensor colocado al final de esta, detectará su presencia y esperará hasta que haya una caja en la zona de almacenamiento. Cuando el emisor de cajas grandes emita una, y esta esté situada en su correcta posición, el "pick and place" introducirá la pieza azul dentro de esta. Este proceso se repetirá un número determinado de veces hasta que la caja esté completa. Una vez suceda esto, la caja saldrá en dirección de la célula de paletizado, donde un robot la cogerá y colocará mediante un patrón en el pallet.

Si por el contrario la pieza detectada por el sensor es verde, continuará hasta llegar a la zona del "pick and place" pasando por las cintas transportadoras que serán las encargadas de trasladarla hasta la zona final y siendo detectada por los diferentes sensores de posición, como en el caso de la azul. Una vez llega a la zona de "pick and place", la pieza esperará hasta que el emisor de cajas medianas emita una. Cuando esta esté en posición, el "pick and place" repetirá el mismo proceso que en las azules, e introducirá un número determinado de piezas verdes en la caja mediana hasta que quede completa. Cuando la caja mediana esté completa será transportada por cintas hasta el robot donde será paletizada del mismo modo que la azul siguiendo un patrón establecido.

Finalmente, si la pieza detectada es gris, esta continuará su transporte a través de las cintas transportadoras hasta llegar a la cinta de la zona de derivación. Una vez allí se activará el primer cilindro neumático, trasladando la pieza hasta otra cinta transportadora donde continuará su trayecto hasta llegar a la zona final de la ramificación. A continuación, encontrándose la pieza en el final de la cinta, un "pick and place" y un sensor, situado en esta, detectarán su presencia y esperará hasta que haya una caja en la zona de almacenamiento.

Cuando el emisor de cajas pequeñas emita una y esta esté situada en la posición correcta, el pick and place introducirá la pieza gris en la misma. Este proceso se repetirá un número determinado de veces hasta que la caja este completa. Una vez suceda esto, la caja saldrá

en dirección de la célula de paletizado, donde un robot la cogerá y colocará mediante un patrón en el pallet.

Cabe destacar que las piezas serán emitidas de forma aleatoria y de forma continua por lo que las tres ramificaciones trabajarán de forma simultánea. Teniendo un buffer de operaciones en caso de fallo o cualquier otra emergencia en la línea.

3. CONOCIMIENTOS PREVIOS

3.1 AUTOMATIZACIÓN

El concepto de automatización está enfocado en una disciplina de control que se basa en el uso de sistemas electromecánicos para controlar de forma automatizada diversos procesos industriales. Abarca control, sistemas digitales, supervisión, gestión de datos, accionamientos, instrumentación, comunicaciones, producción, interacciones y muchos otros.

La automatización incorpora elementos y dispositivos tecnológicos que asegurar tener un control específico sobre los procesos y sus evidentes comportamientos.

Esta automatización, debe ser capaz de poder controlar todo el conjunto de posibles eventos previstos frente a posibles ocurrencias; buscando siempre lograr la situación más favorable según la determinación de los recursos asignados.

La automatización es muy usada en diferentes áreas de trabajo. Los elementos y características de la automatización han tenido un importante impacto en el área industrial, mecánica, informática y máquinas programables.

La alta competitividad empresarial y la necesidad de aumentar eficazmente los procesos de producción mediante la incorporación de la robótica, los robots y automatización de los procesos, han implicado un mayor nivel de integración entre los sistemas productivos y la decisión política empresarial en las áreas de fabricación, gestión de procesos, servicios y la gestión de la información. [1]

3.2 AUTÓMATAS PROGRAMABLES

3.2.1 Definición

Un autómata programable, o PLC, es un ordenador industrial que se utiliza para procesos de automatización industrial. Estos autómatas pueden automatizar un proceso específico, una función de la máquina o incluso una línea de producción completa.

El Autómata Programable recibe información de los sensores o dispositivos de entrada conectados, procesa los datos y activa las salidas según los parámetros preprogramados.

Dependiendo de las entradas y salidas, un PLC puede monitorear y registrar datos de tiempo de ejecución como la productividad de la máquina o la temperatura de operación, iniciar y detener procesos automáticamente, generar alarmas si una máquina funciona mal y más. Los controladores lógicos programables son una solución de control flexible y robusta, adaptable a casi cualquier aplicación. [2]

3.2.2 Tipos de Autómatas

Debido a la gran variedad de tipos distintos de PLC, tanto en sus funciones, en su capacidad, en su aspecto físico y otros, es posible clasificar los distintos tipos en varias categorías.

PLC tipo Nano:

Generalmente PLC de tipo compacto que puede manejar un conjunto pequeño de I/O, generalmente en un número inferior a 100. Así mismo permiten manejar entradas y salidas digitales para alguno de los módulos especiales. [3]

PLC tipo Compacto:

Estos PLC tienen incorporados la fuente de alimentación, su CPU y módulos de I/O en un solo módulo principal. Permiten manejar desde pocas hasta varios cientos de I/O. Su tamaño es superior a los Nano y pueden soportar una gran variedad de módulos especiales como los mencionados a continuación. [3]

- Entradas y salidas analógicas

- Módulos de comunicaciones

- Expansiones de I/O
- Módulos de contadores rápidos

PLC tipo Modular:

Estos PLC están constituidos por un conjunto de elementos que conforman un controlador final. [3]

Estos elementos son los siguientes:

- Rank: perfil soporte.
- Fuente de alimentación: Convierte la tensión de red en tensión de servidor.
- CPU: ejecuta el programa de usuario, alimenta el bus de fondo y se comunica con otras estaciones.
- Módulos de señales (I /O): ajustan diferentes niveles de señal de proceso del autómatas.
- Módulos de función: realizan tareas temporales y de almacenamiento para el procesamiento de las señales en el proceso.
- Procesador de comunicaciones: descarga a la CPU de tareas de comunicación.

3.2.3 Programación de un autómatas

Un PLC (controlador lógico programable) es un dispositivo que se encuentra completamente implantado en cualquier tipo de industria hoy en día. Su operatividad se basa en la realización de tareas de forma periódica y sucesiva.

La secuencia de funcionamiento de este tipo de controladores es la siguiente. [3]

La secuencia comienza con la ejecución del autodiagnóstico. Una vez realizada la misma, se pasa a la lectura de las entradas del dispositivo y la grabación de una imagen de esta en la memoria interna del autómatas.

A continuación, se pasa a ejecución del programa implementado por el programador, teniendo en cuenta la imagen de las entradas mencionadas anteriormente.

Lo siguiente que se realiza es la actualización del estado de todas las salidas de forma simultánea y dando así comienzo a un nuevo ciclo de trabajo.

Si se tiene en cuenta este modo de funcionamiento, podemos decir que los PLC's trabajan en 'tiempo real'.

Dentro de cada ciclo de trabajo, llevado a cabo este en un tiempo relativo de milisegundos (ms), el PLC analiza el estado de las entradas, ejecuta el programa y analiza simultáneamente las salidas; con ello se garantiza una sincronización de todas las acciones llevadas a cabo por el control del sistema.

Hay varios tipos de fabricantes de PLC's como pueden ser:

Allen Bradley, Siemens, ABB, Omron, Schneider, Delta....

A pesar de tener diferentes diseños o entornos gráficos para desarrollar la programación de los PLC's, estos nos permiten realizar la programación en algunos de los lenguajes que se mencionaran en la memoria.

Cabe destacar que en este proyecto se implementará la programación en lenguaje KOP. La programación de lenguaje grafico KOP (esquema de contactos) es similar a la de los esquemas eléctricos, el cual nos permitirá mediante símbolos codificar la secuencia de acciones que deberá realizar el autómatas. [3]

3.3 TIPOS DE ROBOTS INDUSTRIALES

Un robot es un manipulador multifuncional programable, capaz de mover materiales, piezas, herramientas o dispositivos especiales. todo ello a partir de trayectorias variables para ejecutar diversas tareas cíclicas y adaptables.[7]

Los robots se pueden clasificar según los siguientes parámetros:

- Niveles de autonomía
- Accesibilidad
- Grado de posicionamiento
- Carga soportada
- Velocidad

Basándonos en esto, pasamos a mostrar los diferentes tipos de robots industriales:

1.Robot cartesiano: Se caracterizan por posicionarse mediante 3 articulaciones lineales. Hacen movimientos perpendiculares a partir de los 3 ejes cartesianos x, y, z.

2.Robot industrial Scara: Al igual que los cartesianos, también se mueven en los planos cartesianos de x, y, z. Su diferencia radica en la incorporación de un eje final del plano z. Con este eje es posible hacer girar la herramienta en la parte final del brazo robótico.

3.6 grados de libertad o Antropomórfico:

Estos robots son llamados así por sus articulaciones, que pueden colocar su pinza con una posición de 3 movimientos. Esto permite tener más flexibilidad.

4.Máquinas de doble brazo: Trabajan simétricamente sobre un solo objeto.



Ilustración 1: Tipos de Robots Industriales

3.4 TIA PORTAL

3.4.1 Historia de TIA Portal

En 1996, Siemens lanzó el concepto Totally Integrated Automation (TIA) y desde ese momento la plataforma se convirtió en un elemento importante de ingeniería para la aplicación de redes y plantas. Más tarde, en 1998 el administrador SIMATIC integró el

lenguaje de programación para autómatas programables STEP7. Con el que ya se podía resolver todo el proyecto de automatización, con otras herramientas de supervisión y visualización como Simatic HMI, SCADA... [4]

En 2009, el concepto de TIA se hizo realidad con TIA portal, en él se introdujo una plataforma fácil de reconocer para la programación completa de proyectos y puesta en marcha de controles, redes y accionamientos. En este año se presentó una generación de controladores, Simatic S7-1200, en el que se tiene nuevos estándares de flexibilidad con amplias opciones de expansión y en la que se puede adaptar diferentes tareas de automatización. [4]

En esa década, esta plataforma, TIA Porta, presentaba una gran cantidad de funciones y herramientas de desarrollo de aplicaciones de ingeniería bajo un mismo marco y con una interfaz de usuario uniforme [4].

3.4.2 Software de TIA Portal

TIA portal es la evolución de Step 7 y contiene todas las herramientas de este, más otras adicionales, que se han ido integrando durante los años. Es un software que integra diferentes productos SIMATIC para una única aplicación que permite aumentar la productividad y mejorar la eficacia del proceso.

Las diferentes partes que constituyen TIA Portal interactúan entre ellas, ofreciendo así una estabilidad y soporte en la solución automatizada.

TIA Portal dispone de un controlador que ejecuta el programa de control y supervisión del proceso, así como un panel de mando que será utilizado y manejado por el operador. Esta aplicación nos permite programar la CPU en todos los métodos y dispone de elementos propios para estos dispositivos.

En este proyecto se utilizará un PLC SIMATIC S7 (PLC S7-1200), al cual se le podrán añadir diferentes módulos en función de las necesidades requeridas.

3.4.3 Características Básicas de TIA Portal

La primera pantalla que se encontrarán al abrir el software TIA Portal da la opción de abrir un proyecto existente, crear un nuevo proyecto o la de migrar un proyecto.

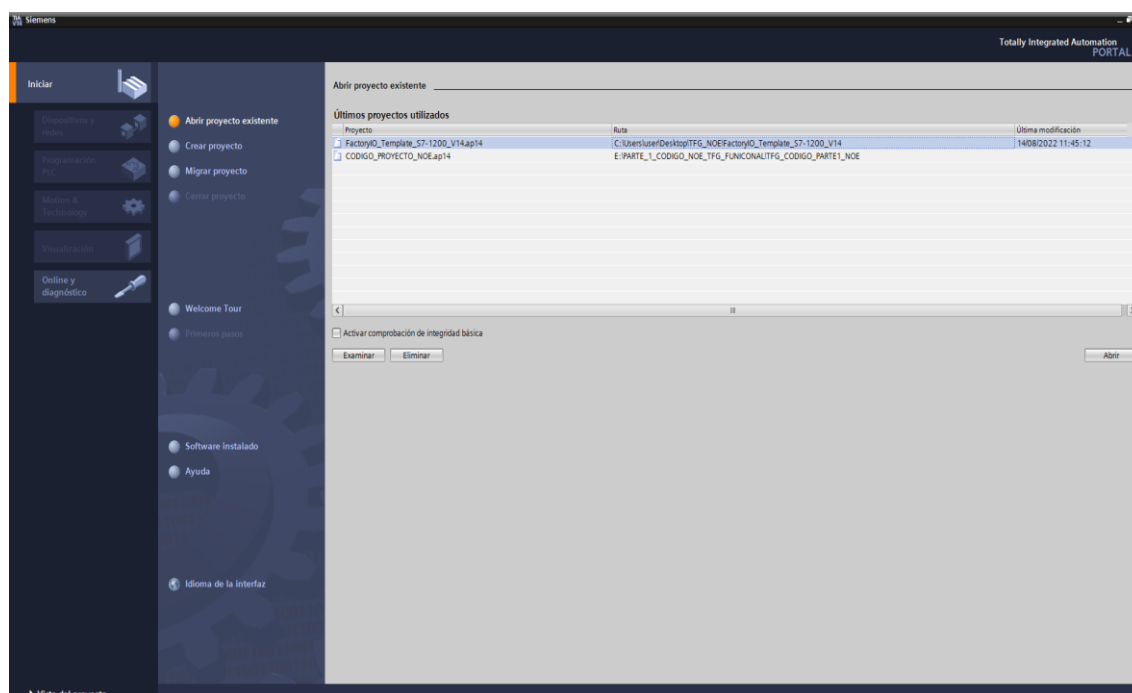


Ilustración 2: interfaz gráfica de TIA Portal

Una vez abierto el proyecto se mostrará la pantalla siguiente. En esta se puede configurar un dispositivo, escribir un programa PLC, configurar los diferentes objetos tecnológicos, configurar una imagen HMI o abrir la vista de proyecto.

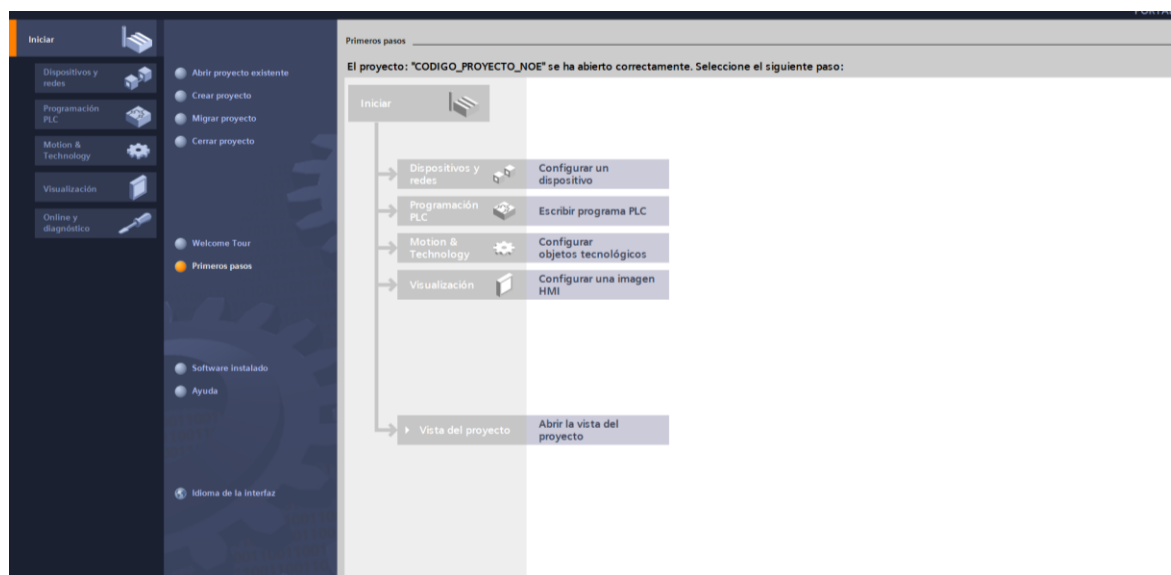


Ilustración 3: Primeros Pasos en TIA Portal

3.4.3.1 Gestión de datos

En TIA portal se programa haciendo acopio de variables y constantes globales, que se pueden generar y reasignar, así como las diferentes variables de entradas y salidas. Para poder editar y declarar el tipo de variables, se engloban todas en lo que se conoce como Tabla de Variables Estándar. La ventaja de esto es que una vez están declaradas las diferentes variables empleadas en la programación, la misma se puede realizar de forma indistinta, teniendo simplemente que invocar la correspondiente dirección asignada a dicha variable o el nombre asignado a la misma.

Para la declaración de una variable se debe incluir lo siguiente:

- Nombre o nemónico: debe ser correcto para una correcta comprensión, ya que aparecerá más de una vez en el programa.
- Tipo de dato: se define la representación de los valores y el rango admisible de estos.
- Dirección de memoria: define el área de memoria en la que la variable lee o escribe un valor.
- Remanencia: que indicará si el valor se queda grabado o no en la memoria una vez desconectamos la CPU de la corriente.

Así mismo, cabe destacar que existen varios tipos de variables. Por un lado, se podrá encontrar las variables globales, que pueden ser remanentes o no, y las variables locales que se declaran dentro de un bloque de programación (OB, FB, FC o DB).

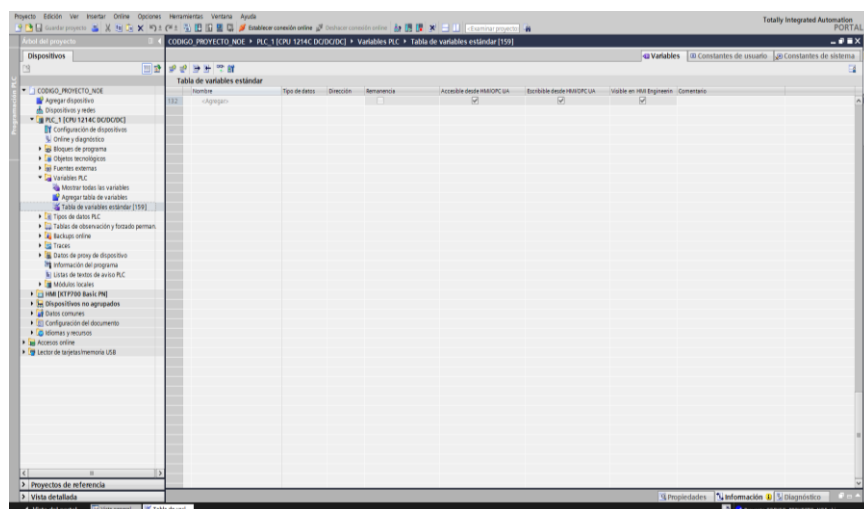


Ilustración 4: Declaración de variables en TIA Portal

3.4.3.2 Lenguajes de programación

En lo referente a la programación en TIA portal, este software ofrece 4 tipos de lenguajes de programación. Todos estos son totalmente equivalentes, queda a la elección y preferencia del programador su uso.

Se explicará cada uno de ellos a continuación.

-Lenguaje FUP: Es un lenguaje gráfico que consiste en un diagrama de funciones que permite visualizar las operaciones en forma de cuadros lógicos similares a los de las puertas lógicas.

-Lenguaje KOP: Lenguaje gráfico, su representación es similar a la de los esquemas eléctricos. los elementos de un esquema de circuitos, tales como los contactos NA o NC se agrupan en segmentos.

-Lenguaje AWL: Es un lenguaje por lista de instrucciones, también conocido como ensamblador, que mediante un conjunto de nemónicos y operandos se componen instrucciones traducibles a código máquina.

-Lenguaje SLC: Es un lenguaje de alto nivel, en el que se realizan códigos de mayor complejidad.

3.4.3.3 Funciones de programación en TIA portal

TIA portal ofrece diferentes bloques de función, pequeñas partes de código a las que es posible llamar en diferentes ocasiones durante la programación.

Los tipos de bloques que facilita TIA portal son:

Bloques de Organización (OB):

El OB1 es el bloque principal, en el que se crea la interfaz de usuario entre el sistema operativo y el programa. Así mismo este bloque se encarga de que se produzca la ejecución de forma reiterativa del programa.

Existen más OB, es decir, TIA portal incorpora más OB en función de la tarea que se va a realizar (ciclos, interrupciones, ...), estos OB dependen de cada CPU, y no se pueden inventar ya que ya vienen establecidos por TIA portal.[5]

Funciones (FC):

Son funciones que no tiene asignado un área de memoria, por lo que al terminar la ejecución de dicha función los datos se pierden. Si se deseara guardar dichos datos, se realizará un bloque de datos globales (DB).

Así mismo, las funciones nos permiten estructurar el programa. Constan de un interfaz de bloque que es el punto de unión entre la función que se programa y el bloque donde se realizará la llamada de esta.

Con el interfaz de bloque se parametrizará la función y se asignarán unos parámetros formales, asegurándose así poder usar dicha función independientemente del área de memoria asignada a las I/O físicas del autómatas. [5]

Bloques de Función (FB):

Es lo mismo que una FC, pero la peculiaridad de este bloque es que tiene asociado un DB de instancia. Con esto, lo que se asegurará la posibilidad de llamar a la FB tantas veces como se requiera, pero sabiendo que cada vez que esta sea llamada, deberá asociarse un DB distinto para el almacenamiento de la memoria.

Esto lo que permite es realizar programaciones más complejas y colocar las peculiaridades de cada una de ellas en las áreas de memoria de los DB de instancia.[5]

Bloques de datos globales (DB):

Como se ha indicado en líneas superiores, los bloques de datos pueden ser de dos tipos, de instancia empleados en los bloques de función o globales. Este bloque es primordial, ya que guarda datos con los que el usuario trabajará, siendo accesibles desde cualquier función o bloque.[5]

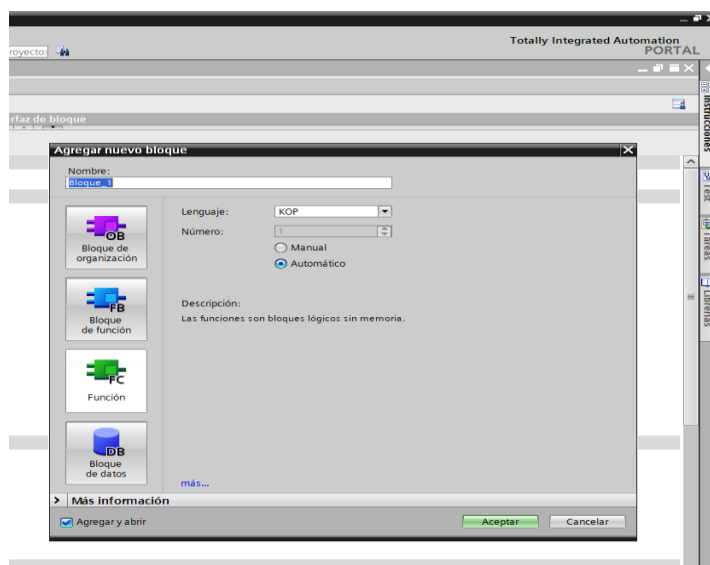


Ilustración 5: Tipos de Bloques en TIA Portal

3.4.3.4 Funcionamiento online

El sistema ofrece una visualización del estado operativo, una sinopsis del correcto funcionamiento y una comparación online/offline.

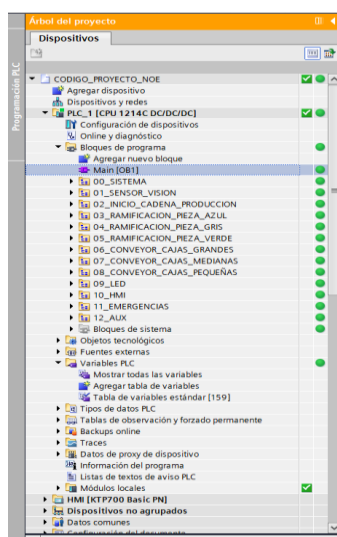


Ilustración 6: Sistema modo Online en TIA Portal

3.4.3.5 Librerías en TIA Portal

Las librerías de TIA Portal permiten utilizar diferentes funciones dentro de un mismo proyecto, utilizar elementos de otros proyectos o incluso exportar y compartir estos con otros usuarios.

3.5 FACTORY I/O

3.5.1 Historia de Factory I/O

Factory I/O es un software de simulación 3D, fue creado en 2016 por Real Games, el cual nos permite construir y controlar procesos industriales en tiempo real.

Factory I/O está diseñado para ser fácil de usar, permite construir rápidamente una fábrica virtual utilizando una selección de piezas industriales comunes. Se programarán escenas completas y se podrán controlar con tecnologías externas para las que dispone las interfaces necesarias como por ejemplo TIA Portal. [6]

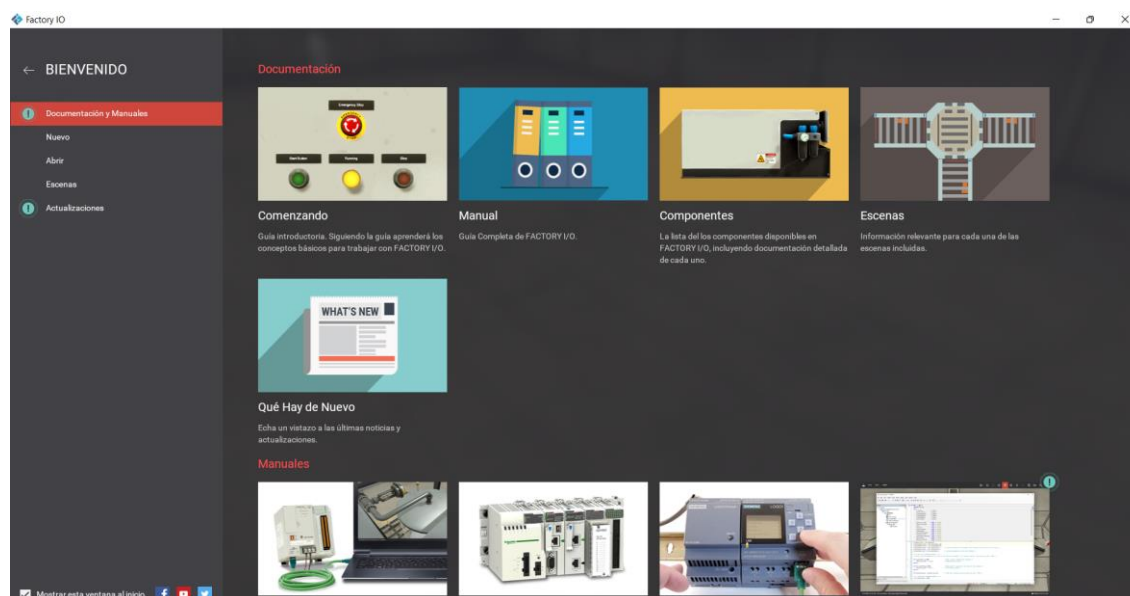
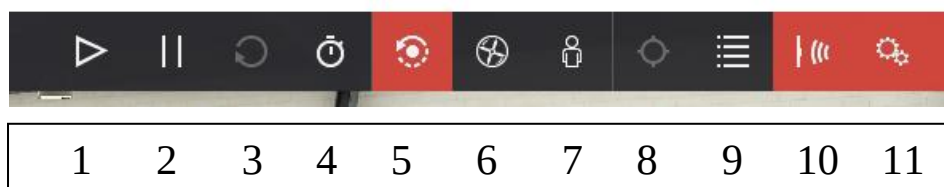


Ilustración 7: Interfaz Gráfica Factory I/O

3.5.2 Elementos de Factory I/O

3.5.2.1 Barra de Herramientas

Para este apartado se mencionarán los elementos empleados en la realización del proyecto, para ello se seguirá el siguiente esquema en la explicación de estos.



1.Ejecutar / editar: Este control cambia el modo de simulación, es imprescindible tener el modo RUN activo en la simulación de TIA Portal y PLCSIM para que funcione correctamente.

2.Pausa: Pone en stop la simulación.

3. Restablecer: Restablece la simulación.

4.Escala de tiempo: En este apartado se puede modificar la velocidad con la que se ejecutará la simulación.

5.Cámara: Selección de la cámara orbital.

6.Fly cámara: Selección de la cámara Fly.

7.Cámara en primera persona: Se puede visualizar la planta desde la perspectiva de una persona física con las limitaciones que esto implica.

10.Etiqueta de sensor: Nos muestra todos los sensores que están en la escena y la situación en la que se encuentran.

11.Etiquetas de actuador: Muestran todos los actuadores que se encuentran en la escena y la situación de estos.

3.5.2.2 Opciones

Del mismo modo que en el apartado anterior se mostrará a continuación las diferentes opciones de las que consta el software de Factory I/O, acompañadas con una pequeña explicación de estas.

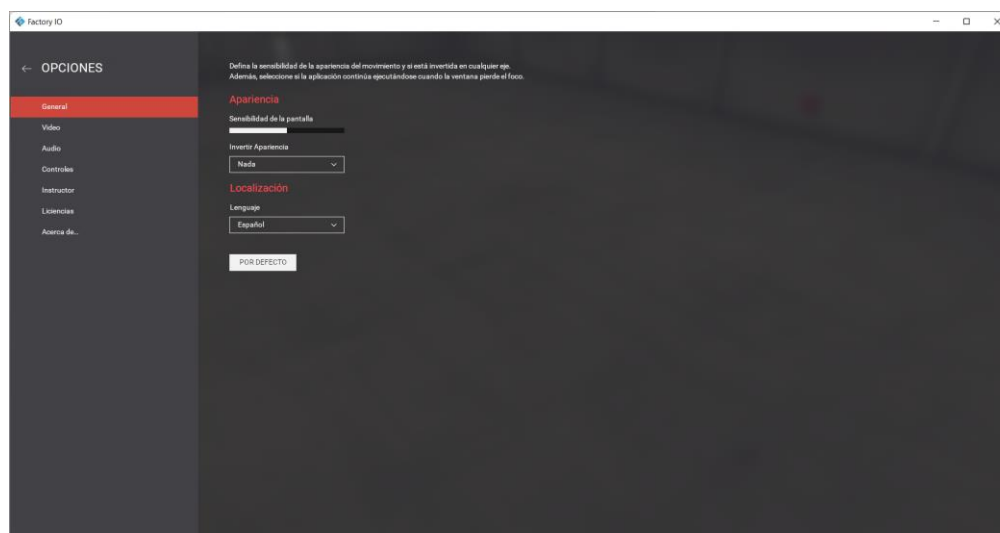


Ilustración 8: Opciones de Factory I/O

1.General: En este apartado se pueden ajustar los siguientes parámetros:

- Sensibilidad de vista (sensibilidad de la cámara y giros de esta).
- Inversión de rotación: invertir la vista seleccionando un eje.
- Idioma: establecer el idioma para la interfaz del programa.

2.Video: Determina la resolución de la pantalla cambiando el modo, pudiendo seleccionar la calidad visual.

3. Audio: Activación y desactivación del sonido en la escena creada.

4. Control S: Visualización y modificación de las letras a pulsar para realizar los movimientos por la escena.

5. Instructor: Permite bloquear la escena en modo ejecución y ocultar advertencias y fallos en las etiquetas.

6.Licencia: Se debe introducir la verificación de esta y se podrá observar detalles de la actual.

3.5.2.3 Navegación

Una de las habilidades más importantes para aprender en Factory I/O es cómo usar las cámaras. Las cámaras se utilizan para navegar en el espacio 3D y son la clave para interactuar con las piezas o construir nuevas escenas. Puedes utilizar tres tipos de cámaras: **Orbit (1)**, **Fly (2)** y **First Person (3)**.

Como se ha mencionado con anterioridad, los botones situados en la barra de herramientas corresponden a estos tres tipos de cámaras. A continuación, se pasará a dar una descripción más detallada de los 3 tipos de cámaras de los que consta Factory I/O.

1. Cámara Orbital: La cámara orbital es la cámara predeterminada y debe usarse para la creación de una escena. Antes esta era la única cámara que nos permitía movernos a través de las piezas sin colisionar con estas. Su funcionamiento es sencillo, simplemente se tiene que girar alrededor del punto de interés que se define con un doble clic izquierdo en la pieza.

2. Cámara Fly: La cámara Fly se utiliza para moverse literalmente en el espacio 3D. esta cámara colisiona con las partes de la escena, y no es detectada por los sensores.

3. En primera persona: La cámara en primera persona representa a un operario de 1.80m aproximadamente de altura. Colisiona con las partes de la escena y por defecto no es detectado por los sensores.

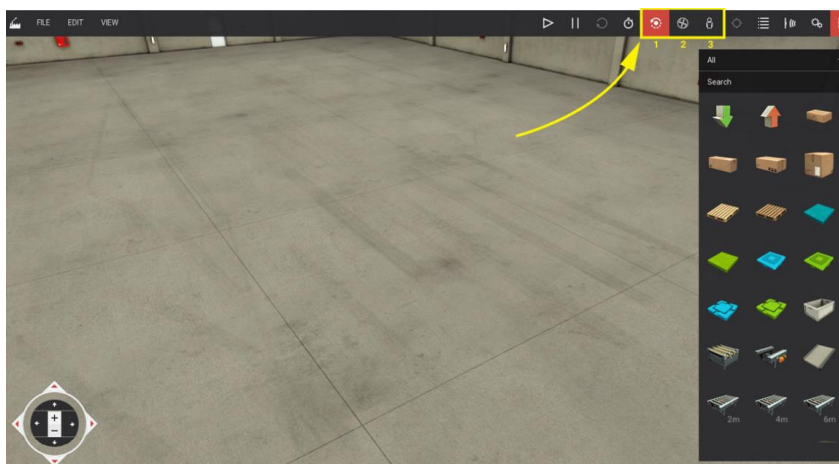


Ilustración 9: Navegación en Factory I/O

3.5.2.4 Elementos

Factory I/O dispone de una amplia gama de elementos divididos en 9 categorías:

1.Items: Está compuesto por todas las cajas y piezas que circularan por el proceso industrial, pudiendo trabajar con las cajas de diferentes tamaños, piezas de tres colores y tres formas distintas, así como pallets de apoyo o necesarios para la circulación de estas en ciertas estaciones de trabajo.



Ilustración10: Items de Factory I/O

2. Carga pesada: Este grupo lo forman todos los elementos de circulación de objetos con mayor peso, por lo que son robustos, anchos y poseen poca altura y velocidad de operación. Están relacionados directamente con los ítems de cajas o pallets.

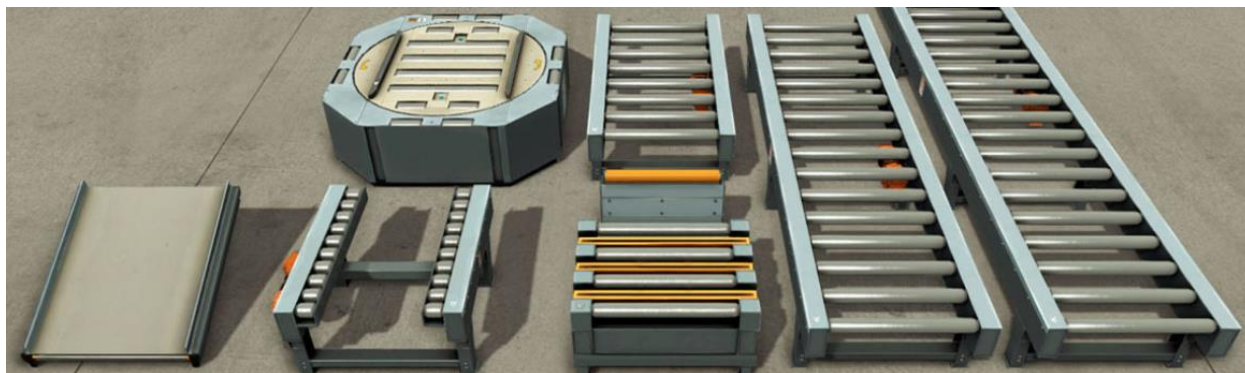


Ilustración 11: Elementos de Carga Pesada de Factory I/O

3.Carga ligera: Se componen de elementos de circulación dedicados al transporte único de cajas pequeñas o piezas, el cual, permite el uso de maquinaria asociada como: su clasificación en forma de brazo, ruedas giratorias o cilindros neumáticos (pushers), que permiten el desplazamiento a distintas líneas de producción o maquinaria de alineación.



Ilustración 12: Elementos de Carga Ligera de Factory I/O

4.Sensores: Está constituido por una pequeña variedad de sensores dedicados a procesos muy comunes como la detección de materiales, elementos conductores o códigos de barras. Además, se permite la clasificación de los objetos según su altura, peso, color y tipo, como en el caso de las piezas.

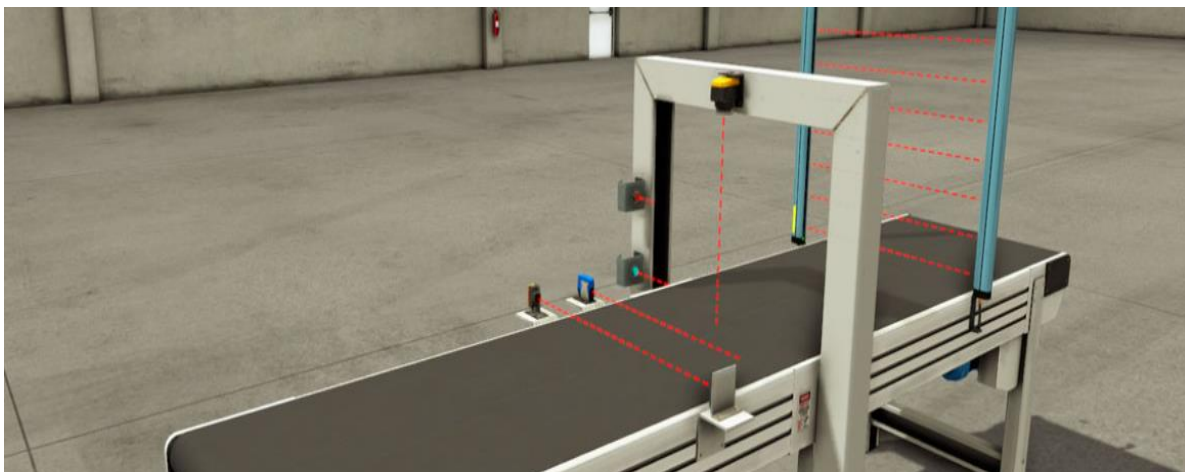


Ilustración 13: Sensores de Factory I/O

5. Operadores: Abarca todos los elementos de actuación sobre la planta, ya sean botones, potenciómetros, selectores, contadores o indicadores.



Ilustración 14: Operadores de Factory I/O

6.Estaciones: Estas poseen una mayor complejidad que el resto de los elementos, ya que están dedicadas a una utilidad directa sobre los ítems. Están compuestas por un conjunto de elementos cerrado, que dispone de su propio sistema de entrada, salida y una botonera sensórica.

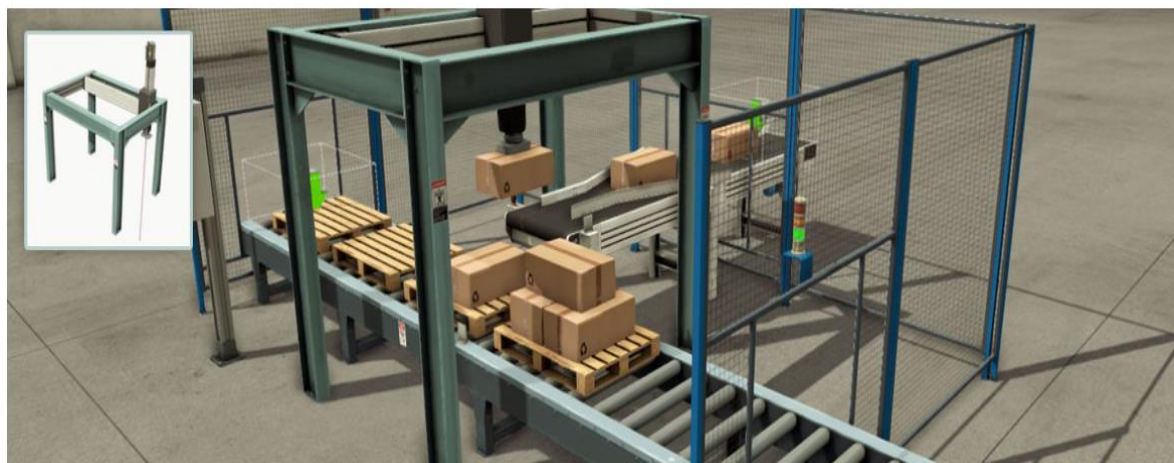


Ilustración 15: Estaciones de Factory I/O

7. Elementos de alarma: Una de las grandes ventajas que nos proporciona Factory I/O es la posibilidad de la introducción de elementos detectores de errores o fallos, los cuales se comunican mediante alarmas visuales o auditivas.

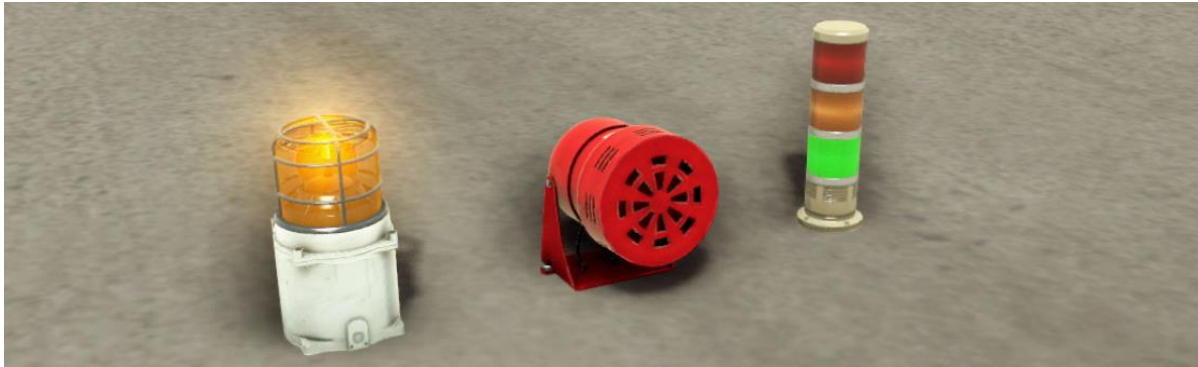


Ilustración 16: Elementos de alarma de Factory I/O

8. Pasarelas: El software de Factory I/O añade estructuras de protección o de accesibilidad a ciertas zonas de la fábrica como medida de realismo y acercamiento a lo que sería una planta de producción en la vida real. Estos elementos no disponen de ningún tipo de control desde el PLC.



Ilustración 17: Pasarelas de Factory I/O

9. introducción y extracción de ítems: Para introducir o extraer elementos, se ha de hacer uso de estos sistemas, que simulan la entrada y salida de la fábrica de los materiales.



Ilustración 18: Introducción y Extracción en Factory I/O

El control de todos estos elementos se dividirá entre los dedicados a la detección (son los sensores introducidos manualmente o pertenecientes a las estaciones) o los actuadores (que son aquellos que producen algún efecto directo en la planta). Se podrán configurar de manera digital, analógica u híbrida entre los anteriores casos.

Cabe añadir que las librerías de elementos son cerradas y no permiten la introducción ni modificación de ninguno de estos.

3.5.2.5 Etiquetas

En Factory I/O cada actuador o sensor está asociado a una etiqueta, si sitúas el ratón encima de estas se mostrará el estado y la variable asociada a la misma.



En la pestaña que aparece al pinchar en una etiqueta se pueden modificar tanto el nombre de esta como su estado, es decir se puede activar o desactivar en función de cómo se requiera forzándola. Con esto se conseguirá modificar las variables que se requieran para poder ver el funcionamiento e interacción de estas en la escena.

3.5.2.6 Escenas

Factory I/O ofrece 21 escenas diferentes para que el usuario se familiarice con las capacidades y funciones que ofrece este software.

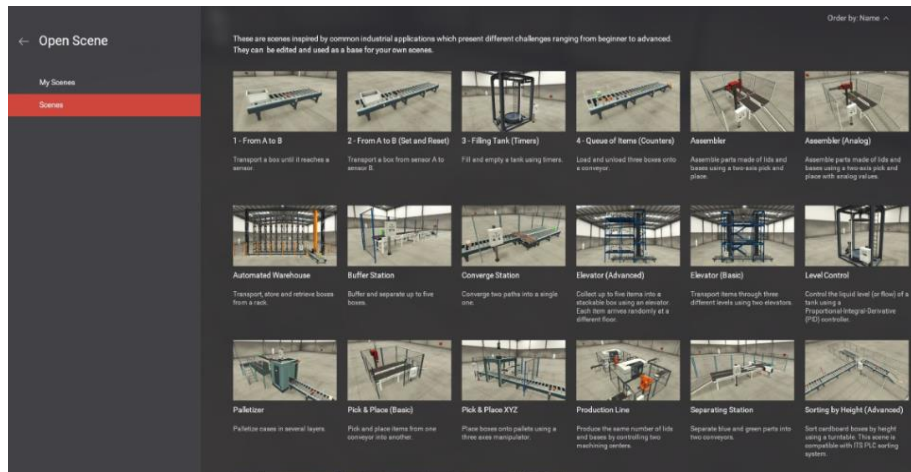


Ilustración 19: Posibles escenas de Factory I/O

3.5.2.7 Controles de Entrada / Salida

Para este apartado, en la barra de herramientas al pulsar 'drivers' aparece la imagen que se mostrará a continuación. En esta imagen se podrán observar todos los sensores y actuadores que se encuentran en el proyecto situados en el controlador. Para que el controlador sea el que necesite hay que seleccionar en la ventana superior el tipo e ir arrastrando cada una de las etiquetas a la dirección de memoria de la CPU.

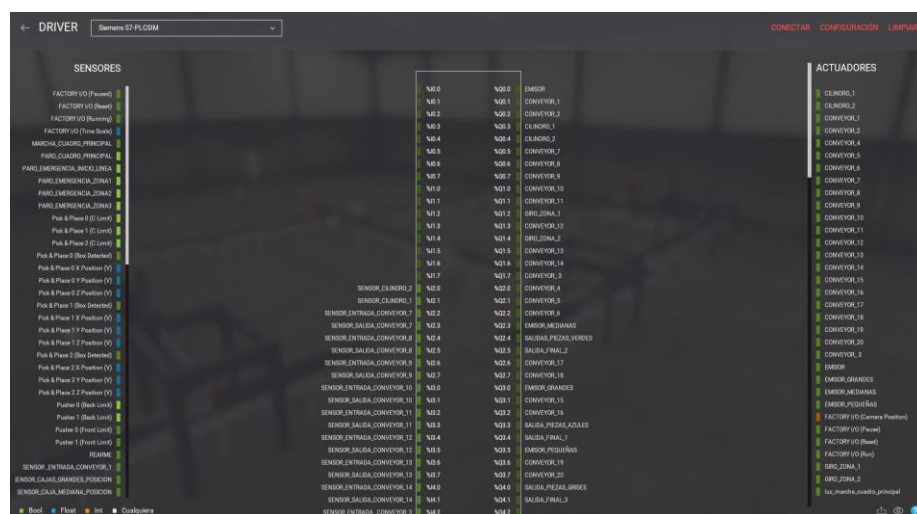


Ilustración 20: Drivers de Factory I/O

3.6. ROBOTSTUDIO

3.6.1 Historia de Robot Studio

Robot Studio ABB permite crear, programar y simular células y estaciones de robots industriales ABB. Es un simulador comercial muy potente, con diversas características y capacidades:

- Creación automática de cualquier tipo de estación.
- Importación de geometrías y modelos 3D de cualquier formato.
- Programación y simulación cinemática de las estaciones.

Con su facilidad de diseño y creación de células robóticas (robots y dispositivos) permite exportar los resultados obtenidos en la simulación a la estación real.

El entorno realista permite simular de manera muy precisa tu estación o proceso real.[8]

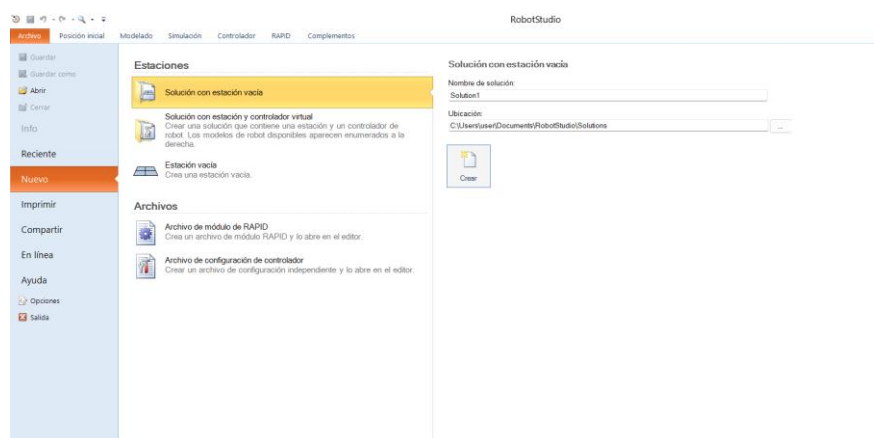


Ilustración 21: Interfaz gráfica de Robot Studio

3.6.2 Lenguaje RAPID

El simulador de RobotStudio funciona sobre RobotWare, que es un conjunto de archivos necesarios para implementar todas las funciones virtual-real y las configuraciones, tanto de datos como de los programas necesarios para el control del sistema robótico. [8]

El lenguaje empleado en Robot Studio es el denominado Lenguaje Rapid. Es un lenguaje de programación textual de alto nivel desarrollado por la empresa ABB.

Una aplicación RAPID consta de un programa y una serie de módulos del sistema.[9]

Este programa es una secuencia de instrucciones que controlan el robot y se dividen en tres partes:

- Main o rutina principal: Donde se inicia la ejecución
- Un conjunto de subrutinas: Sirven para dividir el programa en partes más pequeñas y formar así un programa modular.
- Datos del programa: Definen las posiciones, valores numéricos y sistemas de coordenadas.

● Programa RAPID

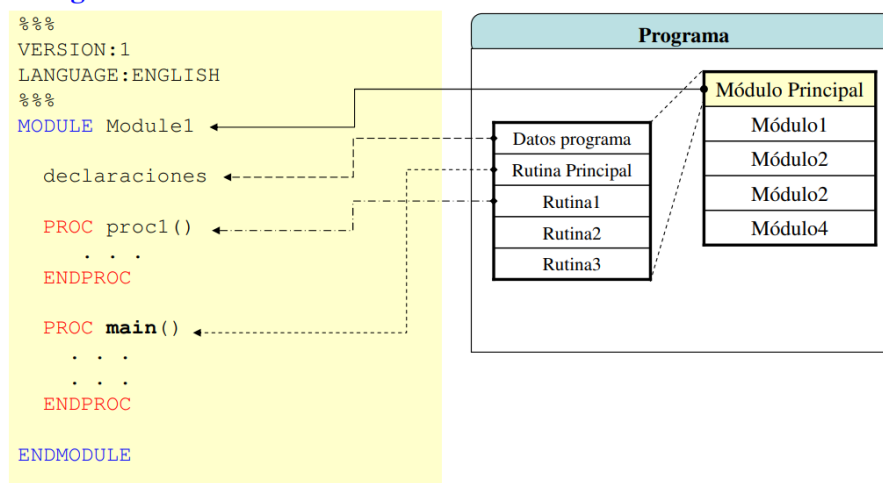


Ilustración 22: Ejemplo Programa RAPID

3.6.3 Partes Básicas de RobotStudio

En este apartado se pasarán a explicar cada una de las partes que constituyen el software. Después de abrir el programa y generar una nueva estación, se encontrará la ventana de posición inicial.

Es una de las más importantes del software ya que no solo permite añadir el robot que se necesite, sino también el controlador y la herramienta que se va a emplear. De forma adicional podemos añadir nuevas bases de coordenadas, posiciones o rutas. Al mismo

tiempo se podrá visualizar el Word-Object en el que se trabajará con el robot, el sistema de coordenadas y la herramienta que está conectada a este, así como sincronizar los cambios realizados al controlador.

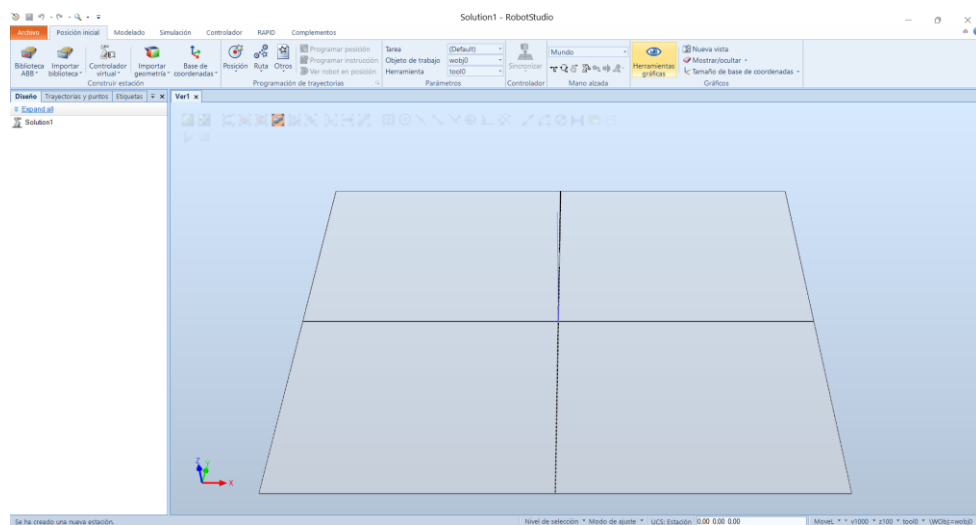


Ilustración 23: pantalla de posición inicial Robot Studio

Continuando con el orden de pantallas, se pasará a explicar la pantalla de modelado. En esta ventana se pueden generar modelos con los que interactuara el robot o bien exportarlos de paquetes adicionales que nos ofrece RobotStudio. También es importante destacar que en esta pantalla se pueden generar los grupos de componentes que se han empleado en el proyecto para cada estación, donde se le incluyen barreras de protección, sistemas de seguridad, ...

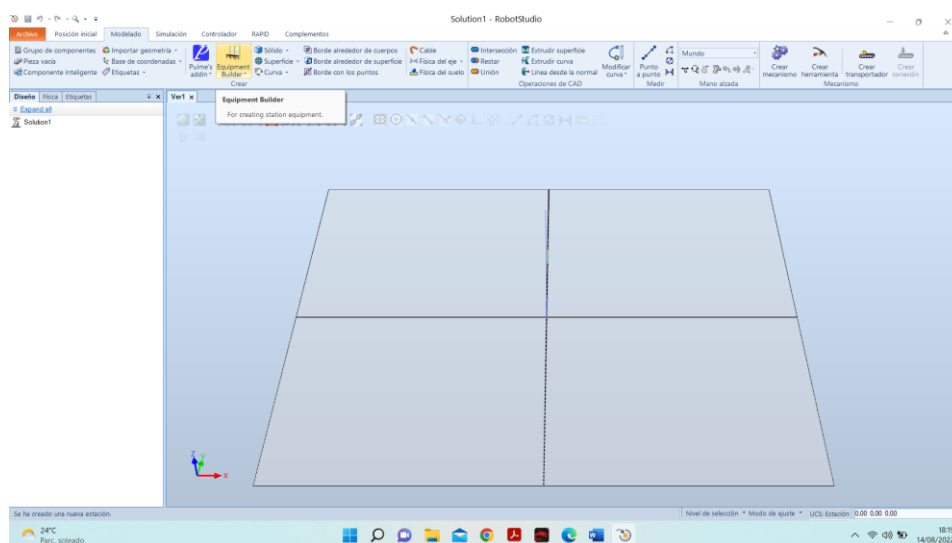


Ilustración 24: pantalla de Modelado Robot Studio

En lo referente a la simulación, RobotStudio es un software muy intuitivo donde se encuentran los botones más comunes para poder realizar la simulación de las estaciones. Lo más importante a destacar es la pestaña lógica de estación, donde es necesario conectar las señales que hemos generado al controlador y en los distintos bloques para que el programa funcione.

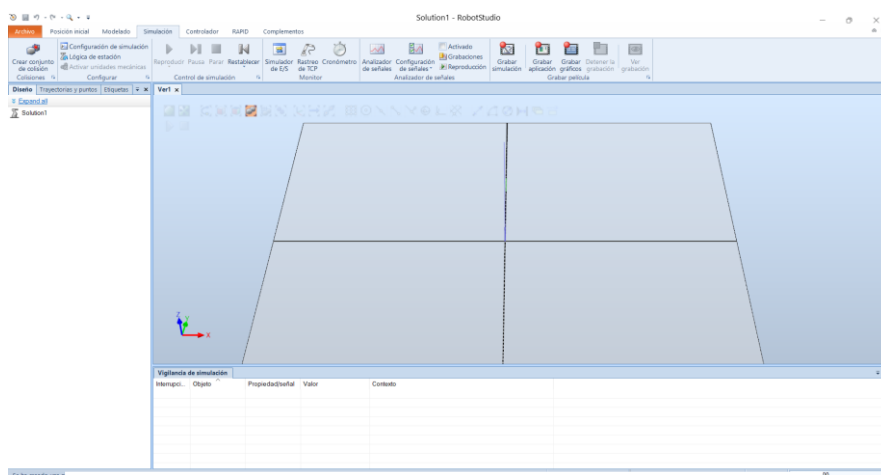


Ilustración 25: pantalla de Simulación Robot Studio

Para el controlador, lo que se debe tener en cuenta es la sincronización de este con la estación o el modo inverso. Al igual que las señales declaradas en la pestaña configuración ya que son las que usaremos más tarde en el programa RAPID.

Es importante también que, al realizar copias de seguridad del controlador o controladores para asegurarse de que, al cerrar el programa y con su posterior apertura, no se pierdan los datos guardados en el mismo.

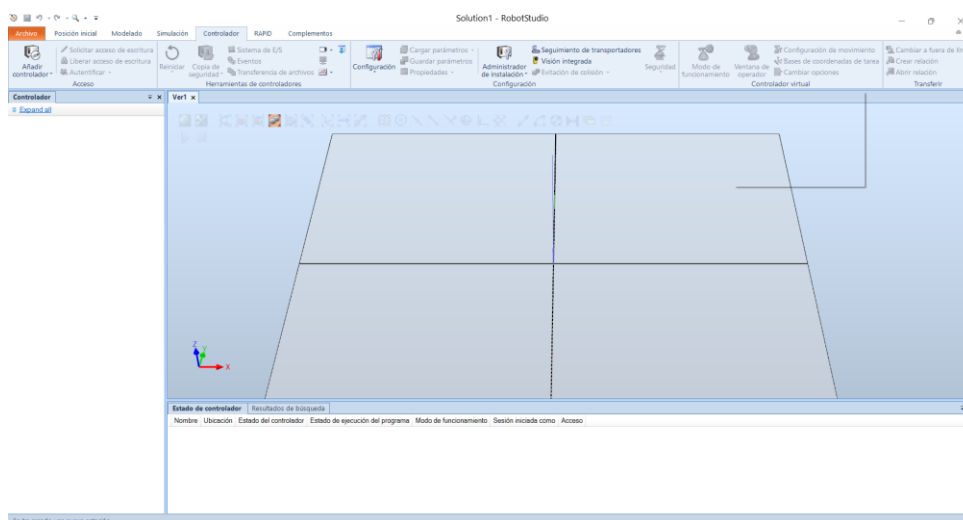


Ilustración 26: pantalla de Controlador en Robot Studio

En la pantalla de RAPID nos aparecerán las diferentes partes del controlador junto al programa principal y las subrutinas que se hayan generado. Es muy importante que al realizar cualquier cambio se aplique ya que, sino no quedará guardado en el programa, así como sincronizar el programa con la estación. También cabe la posibilidad de poder comparar programas, seguir el puntero en la ejecución de este o insertar fragmentos de código ya existente.

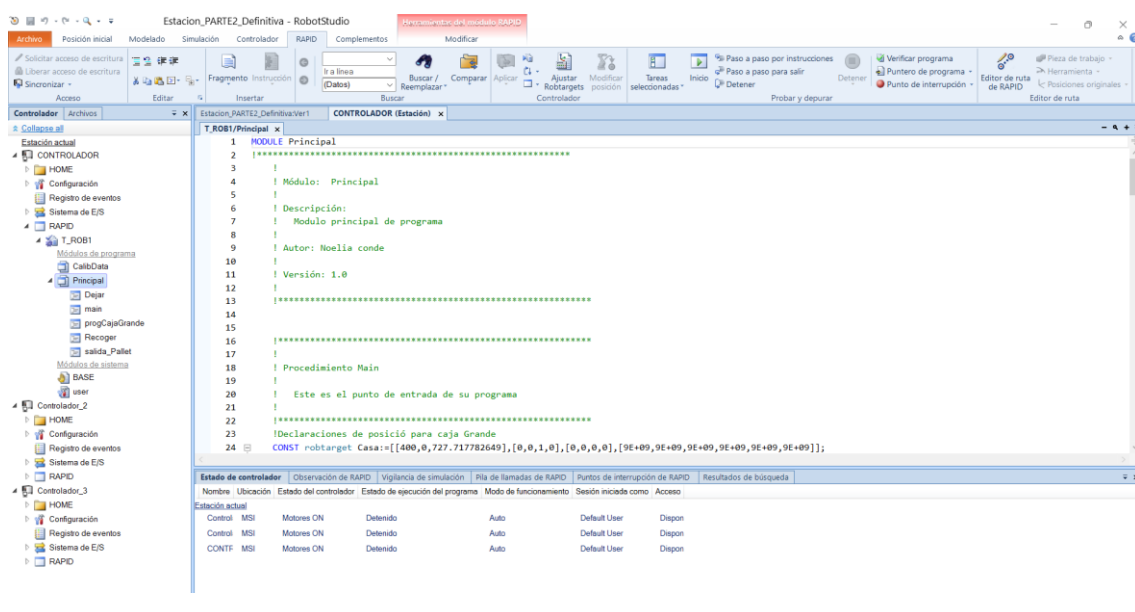


Ilustración 27: pantalla RAPID en Robot Studio

Para finalizar con los elementos básicos que ofrece el software de RobotStudio, se pasará a una breve explicación de la ventana complementos.

En esta ventana se añadirá el controlador de RobotWare, compatible para la versión de Robot Studio que se este utilizando, así como diferentes complementos que ofrece este software en función de las necesidades de las células.

En este proyecto se han añadido dos paquetes muy importantes que han permitido la generación de las cajas de un modo más realista, la creación de unas herramientas más industriales y sobre todo y no menos importante las barreras de seguridad añadidas a las células robóticas.

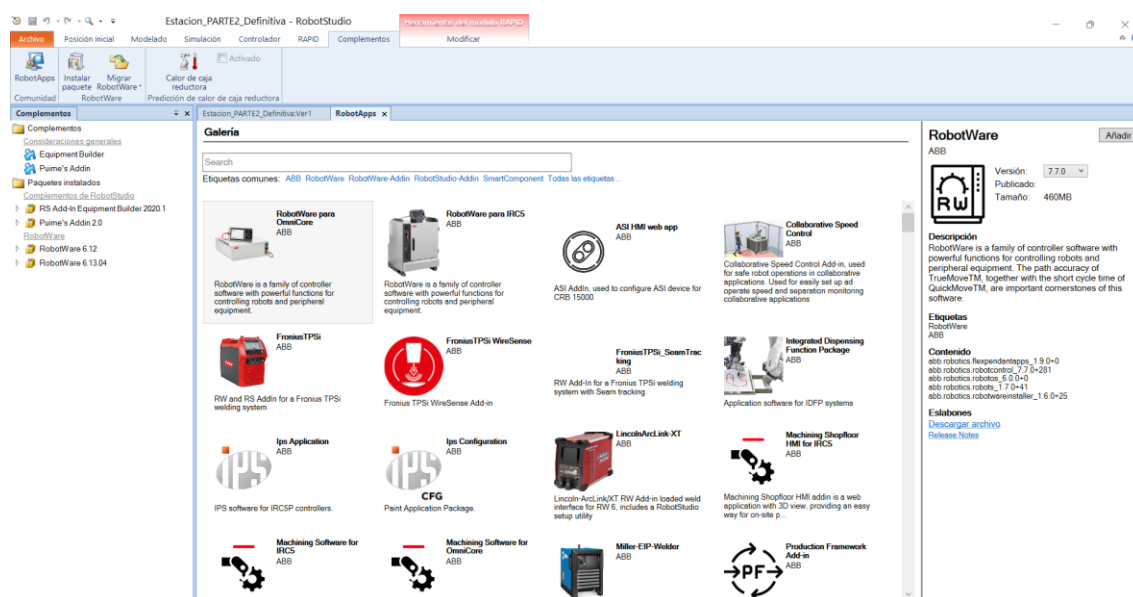


Ilustración 28: pantalla Complementos en Robot Studio

4. DISPOSITIVOS DEL SISTEMA

4.1 EQUIPOS DE LA INSTALACIÓN

Para este apartado se ha introducido una lista de los elementos que se han empleado en la simulación en el programa Factory I/O como se ha explicado con anterioridad. Se realizará también una breve explicación de cada uno de ellos.

4.2 ELEMENTOS DE ENTRADA

Un sensor es un dispositivo que tiene la facultad de detectar movimientos, ruido, presión, luces y cualquier otro tipo de elemento extremo para convertirlo en una señal eléctrica.

-Sensor de visión, sirve para determinar los colores de las piezas emitidas. Es un dispositivo que detecta los colores mediante un haz de luz rojo, cuyo rango oscila entre 0.3 - 2m. Están formados por materiales detectables.

Este sensor se puede configurar para detectar varios tipos de piezas seleccionando la configuración adecuada, es decir:

1. Todo digital: El sensor devuelve cuatro entradas digitales que indican el elemento que se ha detectado.
2. Todo numérico: El sensor nos devuelve un valor que codifica el elemento que se ha detectado.
3. All ID: En este caso el sensor devuelve un único valor, este es aleatorio, e identifica el elemento que ha sido detectado. Funciona de manera similar a los lectores de códigos de barras. [10]

-Sensores difusos de posición, son los encargados de detectar el posicionamiento y movimiento de los elementos. Esta detección se lleva a cabo por un haz de luz rojo, que oscila en un rango de 0 -1.6m para detectar el objeto. Están compuestos por materiales sólidos. En cuanto al ámbito de programación son señales digitales 0-1. [10]

-Otros sensores incorporados en los elementos como los sensores de las células pick and place, que son finales de carrera digitales.[10]

- Un botón verde de inicio del sistema (cuadro principal). Su configuración es digital.[11]

- Un botón amarillo de rearme del sistema (cuadro principal). Su configuración es digital [11]

- Un botón rojo de paro del sistema (cuadro principal). Su configuración es digital [11]

4.3 ELEMENTOS DE SALIDA

En este apartado se enumeran los elementos actuadores de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.

-Cilindros neumáticos. Tienen dos sensores de lengüeta que indican los límites delantero y trasero, incluyen además una servoválvula que se puede utilizar para ajustar y medir la posición de la varilla. El control de los cilindros se puede llevar a cabo mediante valores digitales o analógicos según la configuración que se seleccione.[11]

-Cintas transportadoras de 2 m, 4m y 6m indistintamente empleadas. Los transportadores de banda se emplean para transportar elementos ligeros. Al igual que los cilindros pueden ser controlados por valores digitales o analógicos y como se ha indicado pueden ser de varios tamaños e incluso les hay con forma de giro para facilitar el transporte de los productos.[11]

-Cintas transportadoras de rodillos. Estas se pueden controlar de manera digital y analógica de acuerdo con la configuración seleccionada. [11]

-Robots pick and place. Estos constan de tres ejes que están controlados por servomotores. A menudo se emplean para mover cargas ligeras. El pick and place consta de 4 grados de libertad, tres correspondientes al movimiento lineal de los ejes y el cuarto referido a la rotación de la pinza. La pinza está habilitada por ventosas e incluye un sensor de proximidad. Se puede controlar por valores digitales y analógicos dependiendo de la configuración elegida. Cuando se controla con E/S digitales, el movimiento realizado por el eje es de forma incremental (paso a paso) en cada borde ascendente del valor de la etiqueta que lo controla.[11]

-Cuadros eléctricos. Los cuadros eléctricos están formados por una columna fija donde se sitúa una centralita eléctrica. El cuadro principal que incluye los botones principales del sistema (Marcha, Paro de Emergencia, Paro y Rearme), así como una baliza de colores para indicar de forma visual el estado del sistema en cada instante. Los otros tres cuadros restantes, situados en las ramas del sistema, están constituidos por paros de emergencia para que en caso de fallo el operario los tenga accesibles.[11]

-Setas de emergencia. Las setas de emergencia son de tipo de contacto NC (normalmente cerrado) y constan de dos posiciones. se emplean en eventos de emergencia [11]

-Baliza indicadora de colores (cuadro principal). Las balizas de luz de pila se utilizan comúnmente como indicadores visuales de estados y procesos de las maquinas, están formadas por tres colores diferentes, rojo, amarillo y verde. La configuración de estas es de estado digital.[11]

4.4 OTROS

-Barandillas de protección. Son estructuras de tubos metálicos empleadas para proporcionar seguridad a los operarios. Pueden ser de varios tamaños y formas.[11]

-Alineadores de posición en piezas y cajas. Son estructuras metalizas delgadas, acoplables a un transportador para evitar que las piezas se caigan cuando se transportan a altas velocidades. Hay varios tipos de alineadores.[11]

4.5 ELECCIÓN DEL PLC

La elección del equipo, tanto a nivel de CPU como el número de entradas y salidas, posibilidades de ampliación de módulos, módulos de comunicaciones, redes de supervisión, PLCSIM, etc. Son algunos de los aspectos más complicados de determinar y los más importantes a la hora de elegir el PLC a utilizar. Algunos de los aspectos menos relevantes pueden ser, la seguridad, el costo de los equipos o la fiabilidad.

En cuanto a las comunicaciones, es importante la posible supervisión del sistema por medio de un SCADA.

El PLC elegido para este proyecto pertenece a la serie S7-1200. Se ha escogido una CPU S7 1214 DC/DC/DC.

Esta CPU consta de las siguientes especificaciones:

- Memoria de programas/datos integrada de 100 kbytes, memoria de carga de 2 Mbytes.
- Alimentación de 24 V DC.
- Tiempo de ejecución booleano 0,1 μ s por operación.
- 14 entradas digitales, 10 salidas digitales, 2 entradas analógicas.
- Ampliable con hasta 3 Communication Modules, 8 Signal Modules y 1 Signal Board/Communication Board.
- Entradas digitales utilizables como HSC con 100 kHz.
- Salidas digitales de 24 V DC utilizables como salidas de impulsos (PTO) o salidas con modulación de ancho de impulsos (PWM) con 100 kHz.



Ilustración 29: Modelo CPU 1214 DC/DC /DC

Como ya se ha explicado con anterioridad, el objetivo de la fábrica es la distribución, clasificación y paletizado de piezas cuadradas de tres colores distintos. Para ello se ha realizado una sección principal donde se produce la emisión y clasificación de las piezas y tres bifurcaciones para su posterior transporte, almacenamiento y paletización.

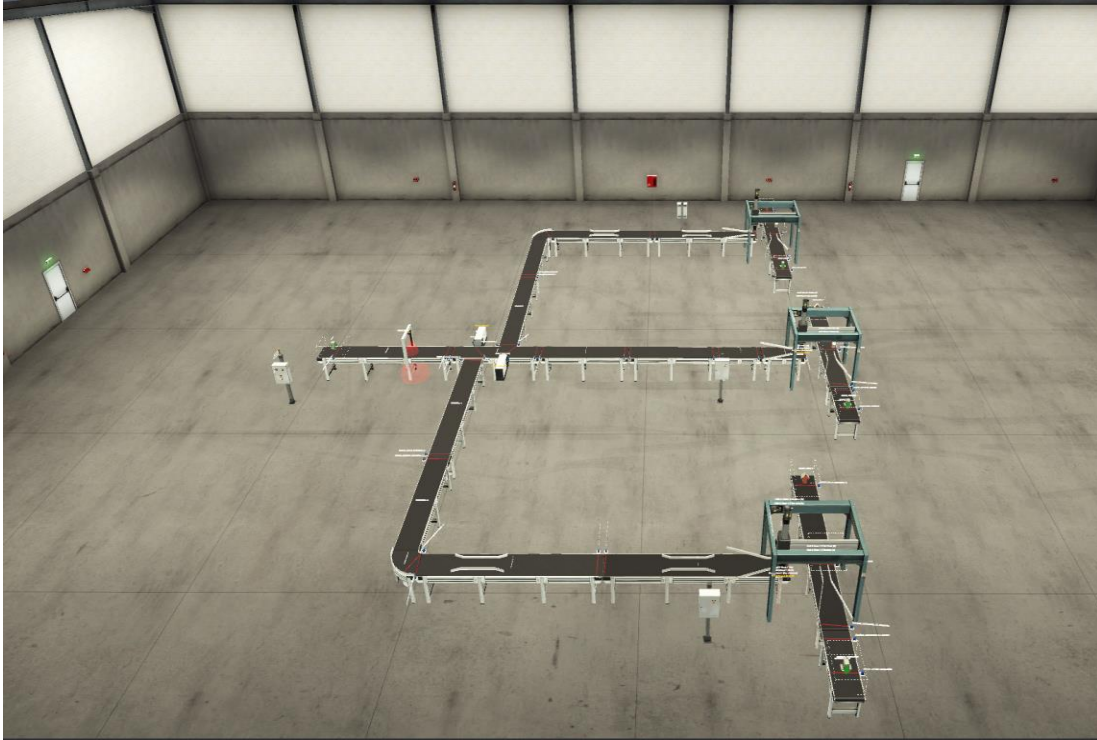


Ilustración 30: Vista General de la Fabrica en Factory I/O

Una vez enumerados los componentes del sistema simulado, se procederá a explicar dicho sistema de manera más extensa.

4.6 EMISIÓN DE PIEZAS Y DIVISIÓN DE COLOR

El proceso comienza con la emisión de las piezas a través del 'Emisor'. Este está configurado para que las piezas cuadradas que aparezcan sean cualquiera de los 3 colores posibles (Gris, Verde, Azul).

Una vez se emita la primera pieza, la siguiente que se emitirá será controlada a través del sensor de salida de la cinta 1.

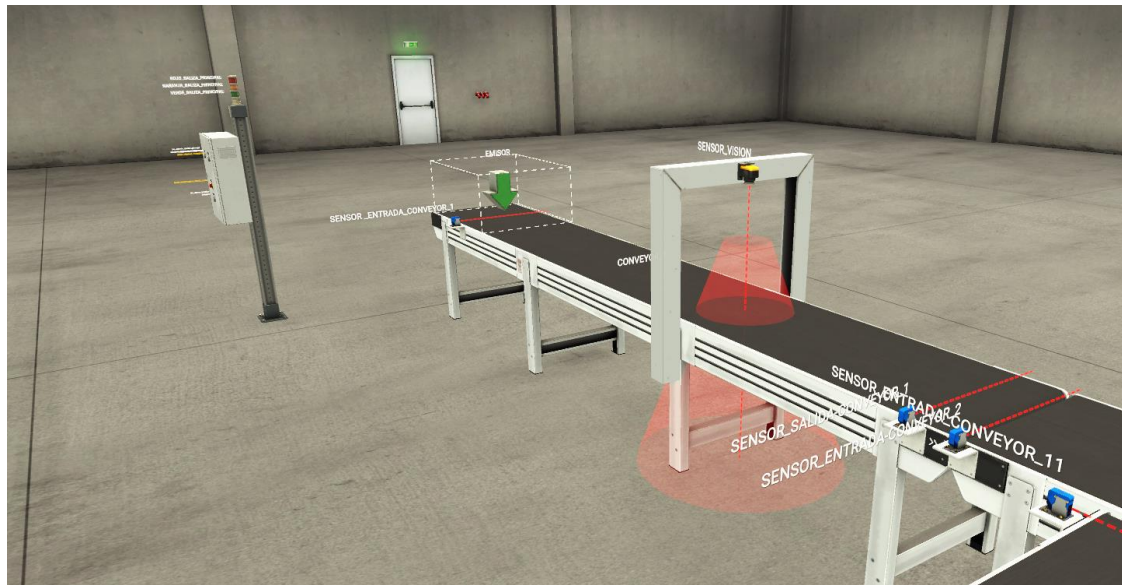


Ilustración 31: Zona de emisión y detección del color

En esta primera etapa se encuentra también la zona de división de color. El propósito de esta línea es la salida a tienda de cajas con tres tipos de piezas diferentes.

Es por ello por lo que se necesita hacer una división de color de las piezas emitidas. El valor integral del sensor ya nos permite hacer dicha clasificación por lo que se diseñan 3 caminos diferentes y junto con los cilindros neumáticos nos permiten derivar las piezas a cualquiera de los 3 caminos existentes para su posterior traslado a las zonas de pick and place.

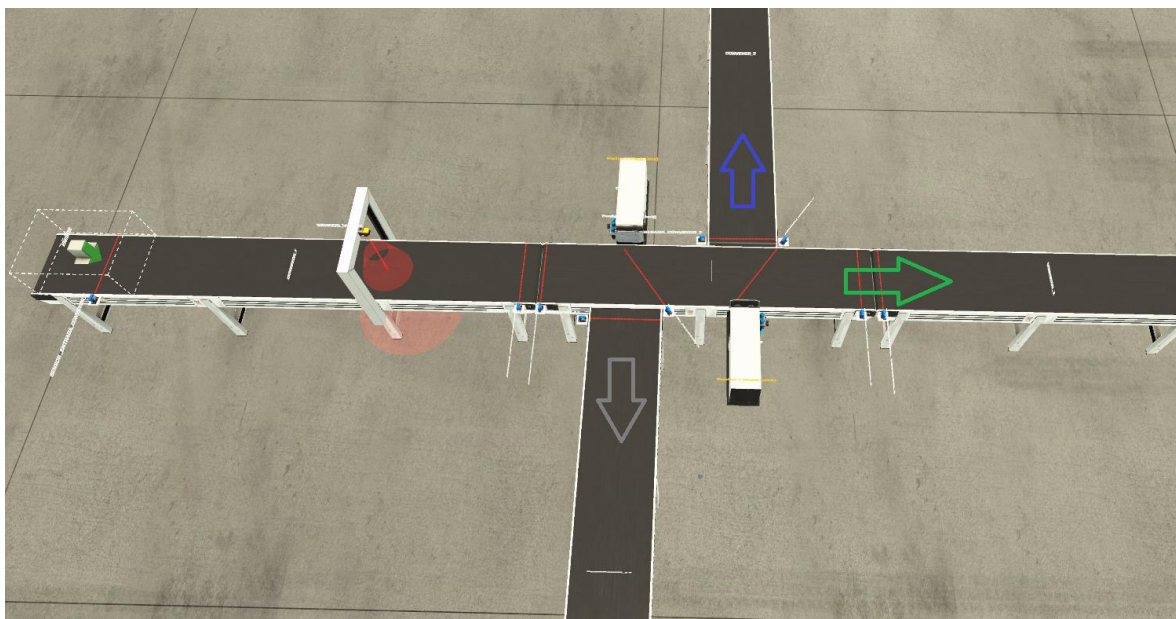


Ilustración 32: ramificaciones según el color

4.7 ZONA DE TRANSPORTE

Una vez han sido clasificadas por el color las piezas y han sido clasificadas en la zona derivadora, las piezas salen por las diferentes ramas según su color. En estas ramas se encuentran las zonas de transporte, donde las piezas son trasladadas por cintas transportadoras hasta la entrada al pick and place.

Pasan por una serie de sensores de posición que marcarán donde se encuentra la pieza en cada instante y en caso de fallo se pasará a activar el buffer de seguridad. Así mismo, se ha añadido como medida de seguridad, en cada línea un cuadro con un paro de emergencia para posibles fallos del sistema.

Cabe destacar que tanto para la rama gris como para la rama azul se han incorporado giros, así como alineadores en las 3 líneas para que la pieza quede correctamente posicionada.

A continuación, se pasan a mostrar las 3 ramas en función de los colores.

1.Rama Gris:

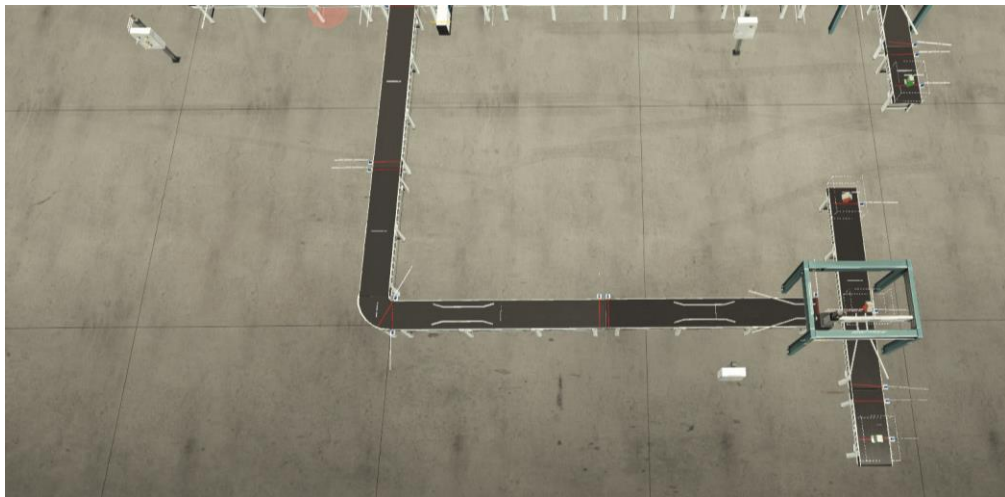


Ilustración 33: Ramificación Gris

Por esta rama circularan las piezas grises, las piezas pasan de la zona derivadora a la rama de grises mediante el accionamiento del cilindro neumático 1, que es el encargado de empujarlas, desplazarlas y posicionarlas en este ramal.

2.Rama Verde:



Ilustración 34: Ramificación Verde

Por esta ramificación circulan las piezas verdes, en este caso al ser clasificadas siguen su curso por el conjunto de cintas transportadoras que las sitúan en la entrada al pick and place.

3.Rama Azul:

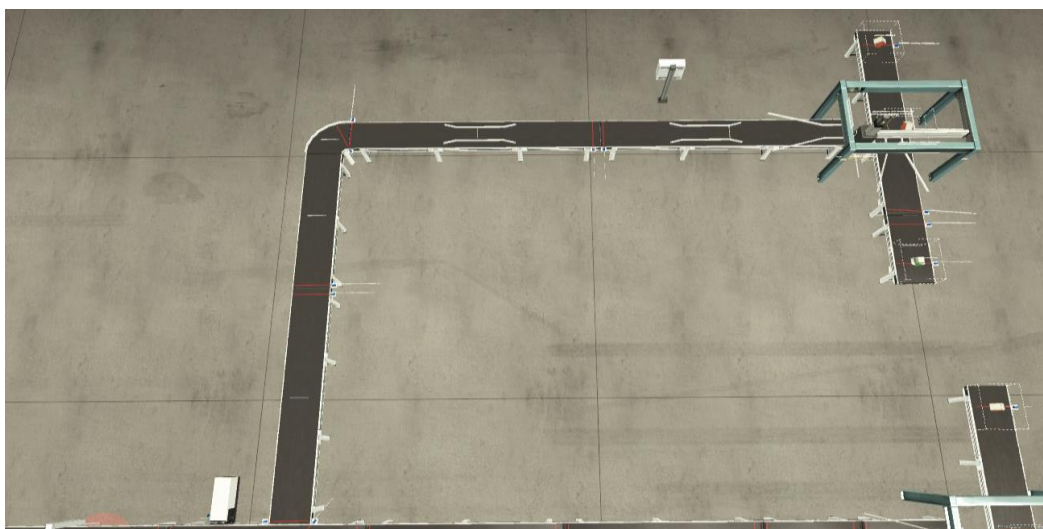


Ilustración 35: Ramificación Azul

Por último, en esta rama, encontramos similitudes a la de la rama gris. La diferencia es que una vez han sido clasificadas las piezas azules, se activa el cilindro neumático 2 el cual es el encargado de desplazarlas y posicionarlas en la ramificación azul para su posterior traslado hasta el inicio del pick and place

4.8 ZONA DE PICK AND PLACE

En esta sección, se realiza la carga de las piezas en los 3 tipos de cajas. Para ello una vez la pieza se encuentra en posición y hay caja del tamaño indicado en cada una de las ramificaciones, el pick and place realiza la introducción del producto en la caja, este proceso es repetido un número determinado de veces. El funcionamiento de cada uno de los pick and place es análogo.

Se pasarán a indicar las 3 zonas de pick and place según el orden del apartado anterior.

1.Cajas pequeñas y piezas grises:



Ilustración 36: Pick and Place Cajas Pequeñas

2.Cajas medianas y piezas verdes:

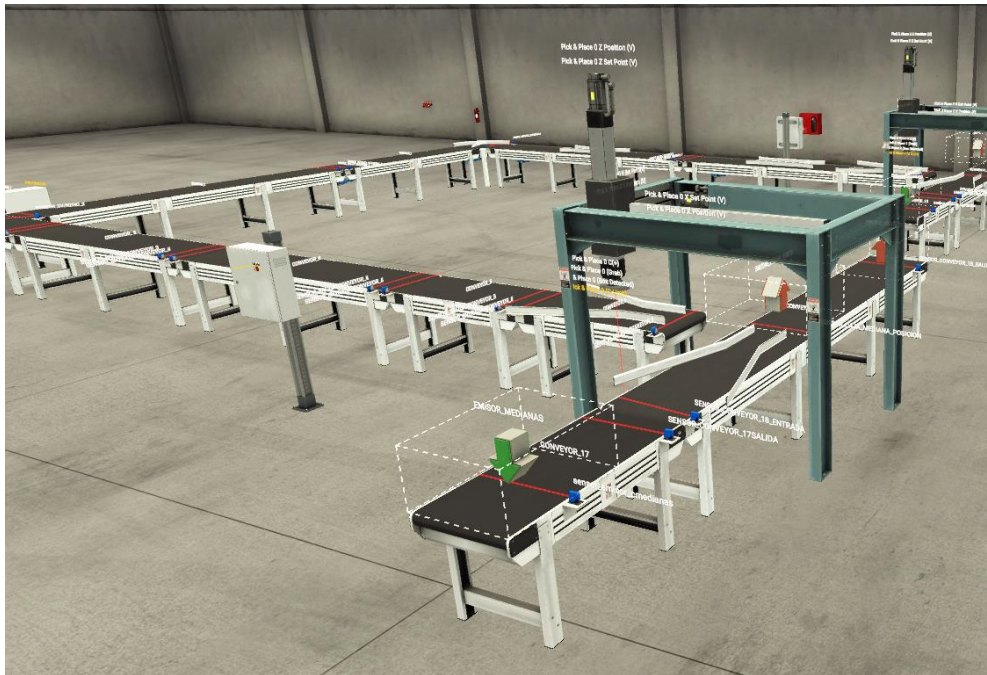


Ilustración 37: Pick and Place Cajas Medianas

3.Cajas grandes y piezas azules:



Ilustración 38: Pick and Place Cajas Grandes

4.9 ZONA DE SALIDA

Esta zona esta adjunta a las células de paletizado desarrolladas en RobotStudio y la cinta que sale de los pick and place. Con esta salida lo que se realiza es la transición del almacenamiento a la paletización.

Como ambos softwares no se pueden conectar de manera virtual, se realizan ambas salidas de forma separada como se mostrará en las ilustraciones bajo estas líneas.

Se encuentran controladas por sensores digitales que nos indican el estado de la caja en cada cinta, así como el estado de las cajas en el pallet que una vez completo será retirado por un operario, concluyendo así el ciclo del sistema.



Ilustración 39: Salida zona Factory I/O

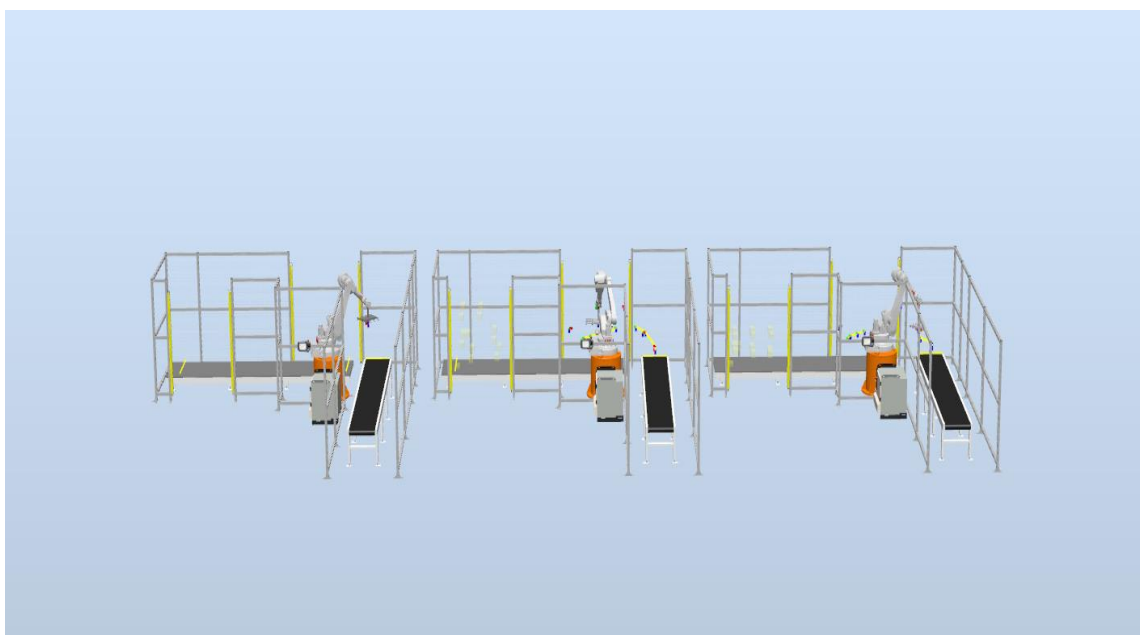


Ilustración 40: Salida zona RobotStudio

4.10 CONEXIÓN DE SOFTWARE

Este proyecto se ha realizado utilizando tres programas diferentes.

En primer lugar, se ha empleado TIA Portal para la realización del programa del control del sistema con una pantalla de interfaz para el operario.

Por otro lado, la visualización y simulación de esta primera parte controlada por TIA Portal se ha realizado con el software Factory I/O.

Finalmente, para la parte de paletización se ha realizado por el software de ABB RobotStudio.

En este apartado se explicará cómo se han conectado los softwares de TIA Portal y Factory I/O.

Tener la visualización que proporciona Factory I/O proporciona al usuario trabajar con un programa que le acercará más a la realidad de la puesta en marcha de la línea; ya que en la programación del proyecto no solo se tendrá en cuenta la realización de las acciones en una secuencia sino también hay que tener en cuenta el tiempo que tarda en desplazarse la

pieza por las cintas, los sensores, el emisor, el correcto posicionamiento de las piezas, el tiempo del pick and place en realizar los movimientos ... todos estos puntos no se tienen en cuenta en una primera programación con TIA Portal, pero en Factory I/O si se pueden visualizar ,lo que nos proporciona una ayuda para corregir esos pequeños fallos que hacen que el proyecto no funcione correctamente.

Para el conexionado, lo primero que se debe tener es la función MHJ-PLC-lab-Function-S71200, mencionada en la explicación de los bloques. Esta función lo que nos proporciona es todo lo necesario para que Factory I/O se comunique de un modo correcto con TIA Portal, para obtenerla simplemente hay que acceder a la página de Factory I/O, y en el apartado de Siemens S7-PLCSIM podremos descargar la función necesaria para emplearla en nuestro proyecto, teniendo en cuenta la versión de TIA Portal empleado y la CPU.

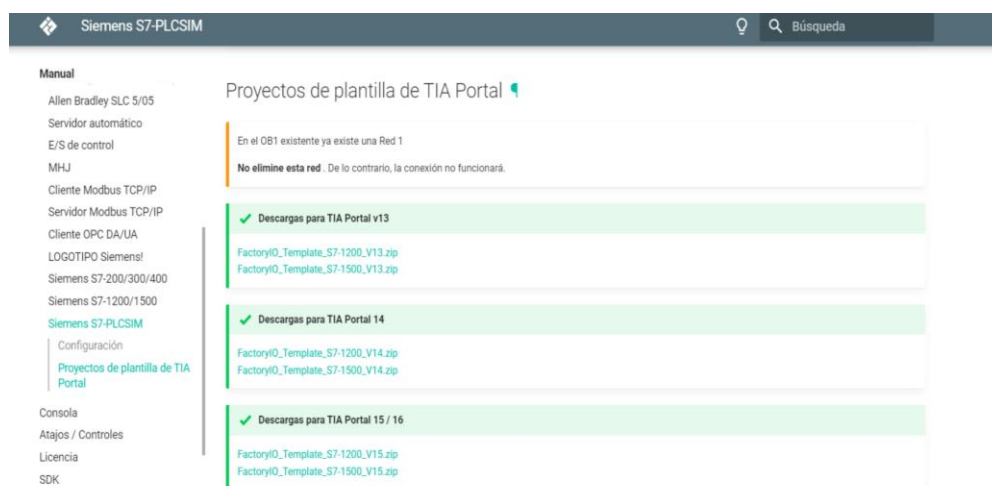


Ilustración 41: Función MHJ-PLC-lab-Function-S71200

4.11 CONFIGURACIÓN FACTORY I/O PARA LA CONEXIÓN

Una vez se ha generado la escena en Factory I/O y se ha creado el proyecto que se desea realizar en TIA Portal, además de haber comprobado el correcto funcionamiento del programa de control en función de las especificaciones y requisitos del sistema, se procede a conectar ambas aplicaciones PLCSIM y Factory I/O para poder observar la simulación del sistema.

Para conectarlos es necesario realizar las siguientes pautas:

En primer lugar, ambas aplicaciones deben estar en funcionamiento, una vez dentro de Factory I/O accederemos a la pestaña drivers (explicada con anterioridad) y aparecerá una pantalla que nos permite asignar las direcciones de memoria para cada sensor o actuador de la escena diseñada, esta parte es muy importante ya que las direcciones de memoria tienen que ser las mismas que se han utilizado en TIA Portal para el programa de control del PLC.

Por último, para la configuración del PLC, debe comprobarse que la simulación del PLCSIM este en estado RUN, sino no se enlazara con Factory I/O.

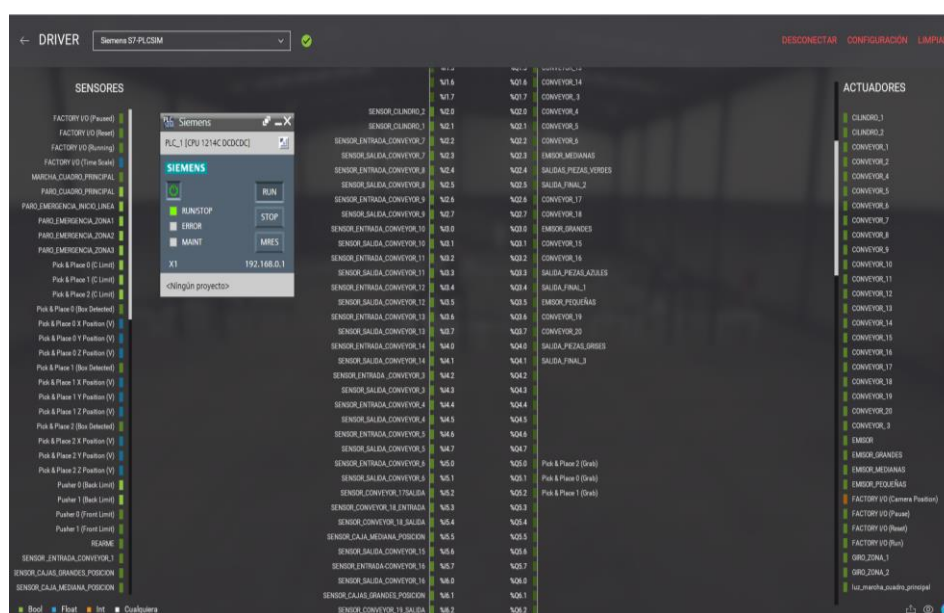


Ilustración 42: Conexión de Factory I/O con PLCSIM

5. PROGRAMACIÓN DEL PLC

5.1 CICLO DE FUNCIONAMIENTO

En lo referente al funcionamiento del proyecto, se realizará de forma cíclica y secuencial, exceptuando cualquier emergencia o fallo en cualquiera de las ramas o celdas robóticas, en este caso se establece un buffer de emergencia para el depósito de piezas hasta su posterior arreglo. Su secuencia de funcionamiento se podrá contemplar en los flujogramas adjuntados en el anexo del proyecto.

5.2 SECUENCIAS DE PROGRAMACIÓN

Se comenzará recordando que el propósito de este proyecto es la clasificación, distribución y paletizado de tres tipos de productos cuadrados en función de su color.

La secuencia comenzará con la emisión de un producto, esto lo realiza el software Factory I/O, será de forma cíclica, controlado mediante los sensores de entrada y salida de la cinta principal con TIA Portal.

Una vez sea emitido el producto, el sensor de visión será el encargado de chequear el producto y determinar si es de color Azul, Verde o Gris, se podría indicar que es esta parte (zona derivadora) donde el sistema puede coger tres caminos diferentes.

Si el producto es gris, se activará el primer cilindro neumático situado en la zona derivadora, siendo el producto desplazado a la primera rama de cintas transportadoras, las cuales, lo transportaran hasta la zona robótica del pick and place. Una vez aquí será un sensor quien lo detendrá a la espera de que la caja pequeña se encuentre situada en su correcta posición para que el pick and place pueda proceder a su correcta introducción y almacenamiento en la misma. Este proceso se repetirá un determinado número de veces de forma cíclica siempre que el sensor detecte que hay pieza para recoger.

Encontrándose la caja completa, pasará por otra sección de cintas transportadoras hasta su llegada a la célula de paletizado. Todo este proceso de transporte esta implementado en TIA Portal.

Para la célula de paletizado llevada a cabo en RobotStudio se han incorporado sensores para la detección de la entrada de la caja, que será desplazada por el robot a un Euro-pallet, donde la situará y colocará de una forma preestablecida en la programación del código RAPID.

En lo referente al producto Azul, la secuencia de pasos es idéntica a la empleada en el producto gris. Se diferencian simplemente en que se activara el cilindro neumático dos en la zona derivadora para desplazar en este caso la pieza azul al conjunto de cintas transportadoras que lo llevarán hasta la célula de pick and place donde, posteriormente, serán introducidas en cajas grandes y paletizadas.

Para finalizar, cuando el producto es verde, este, no activa ningún cilindro, sino que sigue hacia delante por la zona derivadora hasta llegar a las cintas transportadoras, las cuales lo empujaran a la zona de pick and place para su posterior almacenamiento en cajas

medianas y paletizado. Todas las ramas ejecutan la misma secuencia de forma cíclica, ya que la salida de piezas es de forma aleatoria, así mismo son las pequeñas variaciones incorporadas en las ramas de producto azul o gris las encargadas de diferenciar cada parte del sistema.

5.3 VARIABLES DEL SISTEMA

Como ya se indicó anteriormente, el PLC empleado para la realización del proyecto es uno de gama Media/Alta.

La CPU seleccionada proporciona dentro del propio modulo 12 entradas digitales y 9 salidas digitales.

Para esta CPU 1214C DC/DC/DC, podemos destacar el significado de los tres nemónicos que acompañan al modelo de esta.

- El primer nemónico (DC): Es el relativo a la alimentación del PLC, esto lo llevamos a cabo con 24v de continua.
- El segundo nemónico (Dc): Indica las entradas digitales, este siempre será así, al menos en la gama 1200.
- El tercer nemónico (Dc): Nos marca la alimentación de las salidas, que en este caso son de 24v de continua.

Al ser una simulación online y no un proyecto real, no se han incorporado módulos de E/S en los slots colindantes a la CPU, ya que había incompatibilidad de estas con la simulación en Factory I/O. Así mismo cabe destacar que este PLC acepta hasta 8 módulos de expansión de E/S, que se añadirían a los slots colindantes a la CPU un total de módulos de E/S digitales para la realización real del proyecto.

Por lo tanto, se han declarado todas las variables necesarias, pero no se han incorporado los módulos de E/S necesarios por el anterior problema mencionado.

A continuación, se pasarán a mostrar las tablas de las variables empleadas en el desarrollo del sistema, junto a una breve explicación de estas.

ENTRADAS	TAG	TIPO DE DATO	COMENTARIO
I2.0	ENTRADA_AZUL	Bool	Sensor activación cilindro 1
I2.1	ENTRADA_GRIS	Bool	Sensor activación cilindro 2
I2.2	C7_S_ENTRADA	Bool	Sensor entrada rama azul
I2.3	C7_S_SALIDA	Bool	Sensor salida cinta 7
I2.4	C8_S_ENTRADA	Bool	Sensor entrada cinta 8
I2.5	C8_S_SALIDA	Bool	Sensor salida cinta 8
I2.6	C9_S_ENTRADA	Bool	Sensor entrada cinta 9
I2.7	C9_S_SALIDA	Bool	Sensor salida cinta 9
I3.0	C10_S_ENTRADA	Bool	Sensor entrada cinta final rama azul
I3.1	C10_S_SALIDA	Bool	Sensor salida cinta final rama azul
I3.2	C11_S_ENTRADA	Bool	Sensor entrada rama gris
I3.3	C11_S_SALIDA	Bool	Sensor salida cinta 11
I3.4	C12_S_ENTRADA	Bool	Sensor entrada cinta 12
I3.5	C12_S_SALIDA	Bool	Sensor salida cinta 12
I3.6	C13_S_ENTRADA	Bool	Sensor entrada cinta 13
I3.7	C13_S_SALIDA	Bool	Sensor salida cinta 13
I4.0	C14_S_ENTRADA	Bool	Sensor entrada cinta final rama gris
I4.1	C14_S_SALIDA	Bool	Sensor salida cinta final rama gris
I4.2	C3_S_ENTRADA	Bool	Sensor entrada rama verde
I4.3	C3_S_SALIDA	Bool	Sensor salida cinta 3
I4.4	C4_S_ENTRADA	Bool	Sensor entrada cinta 4
I4.5	C4_S_SALIDA	Bool	Sensor salida cinta 4
I4.6	C5_S_ENTRADA	Bool	Sensor entrada cinta 5
I4.7	C5_S_SALIDA	Bool	Sensor salida cinta 5
I5.0	C6_S_ENTRADA	Bool	Sensor entrada cinta final rama verde

15.1	C6_S_SALIDA	Bool	Sensor salida cinta final rama verde
15.2	C17_EMITOR_CAJA_SALIDA	Bool	Sensor control emisión cajas medianas
15.3	C18_S_ENTRADA	Bool	Sensor entrada zona cajas medianas
15.4	C18_S_SALIDA	Bool	Sensor salida zona cajas medianas
15.5	CAJA_EN_POSICION_DEJADA_2	Bool	Sensor detector de posición caja para llenar
15.6	C15_EMITOR_CAJA_SALIDA	Bool	Sensor control emisión cajas grandes
15.7	C16_S_ENTRADA	Bool	Sensor entrada zona cajas grandes
16.0	C16_S_SALIDA	Bool	Sensor salida zona cajas grandes
16.1	CAJA_EN_POSICION_DEJADA_1	Bool	Sensor detector de posición caja para llenar
16.2	C19_EMITOR_CAJA_SALIDA	Bool	Sensor control emisión cajas pequeñas
16.3	C20_S_ENTRADA	Bool	Sensor entrada zona cajas pequeñas
16.4	C20_S_SALIDA	Bool	Sensor salida zona cajas pequeñas
16.5	CAJA_EN_POSICION_DEJADA_3	Bool	Sensor detector de posición caja para llenar
16.6	S_EMITOR_CG	Bool	Sensor emisor cajas grandes
16.7	S_EMITOR_CM	Bool	Sensor emisor cajas medianas
17.0	S_EMITOR_CP	Bool	Sensor emisor cajas pequeñas
17.1	SENSOR_2_PICK_AND_PLACE	Bool	Sensor detector pieza robot 2
17.2	SENSOR_1_PICK_AND_PLACE	Bool	Sensor detector pieza robot 1
17.3	SENSOR_3_PICK_AND_PLACE	Bool	Sensor detector pieza robot 3
17.4	ABSORCION_ROBOT_2	Bool	Absorber pieza robot 2
17.5	ABSORCION_ROBOT_3	Bool	Absorber pieza robot 3
17.6	ABSORCION_ROBOT_1	Bool	Absorber pieza robot 1

I8.0	PARO_CUADRO_PRINCIPAL	Bool	Paro del sistema cuadro principal
I8.1	MARCHA_CUADRO_PRINCIPAL	Bool	Marcha sistema cuadro principal
I8.2	PARO_EMERG_INICIO_LINEA	Bool	Paro emergencia inicio línea cuadro principal
I8.3	REARME	Bool	Rearme del sistema cuadro principal
I8.4	SETA_ZONA1	Bool	Seta emergencia zona 1
I8.5	SETA_ZONA2	Bool	Seta emergencia zona 2
I8.6	SETA_ZONA3	Bool	Seta emergencia zona 3
I9.0	C1_S_ENTRADA	Bool	Sensor entrada cinta 1 zona emisora
I9.1	C1_S_SALIDA	Bool	Sensor salida cinta 1 zona emisora
I9.2	C2_S_ENTRADA	Bool	Sensor entrada cinta 2 zona derivadora
I9.3	C2_S_SALIDA	Bool	Sensor salida cinta 2 zona derivadora
I9.4	CILINDRO_1_BACK	Bool	Retorno cilindro 1(sensor)
I9.5	CILINDRO_2_BACK	Bool	Retorno cilindro 2 (sensor)
I9.6	CILINDRO_1_OUT	Bool	Activación cilindro 1(sensor)
I9.7	CILINDRO_2_OUT	Bool	Activación cilindro 2(sensor)
ID48	PIEZA_ACTUAL_SENSOR	DInt	Lectura del sensor de visión (pieza detectada)

SALIDAS	TAG	TIPO DE DATO	COMENTARIO
Q0.0	EMISOR	Bool	Emisor de piezas (inicio línea)
Q0.1	MARCHA_CONVEYOR_1	Bool	Activación cinta 1
Q0.2	MARCHA_CONVEYOR_2	Bool	Activación cinta 2
Q0.3	MARCHA_CILINDRO_1	Bool	Activación cilindro neumático 1
Q0.4	MARCHA_CILINDRO_2	Bool	Activación cilindro neumático 2
Q0.5	MARCHA_CONVEYOR_7	Bool	Activación cinta 7
Q0.6	MARCHA_CONVEYOR_8	Bool	Activación cinta 8
Q0.7	MARCHA_CONVEYOR_9	Bool	Activación cinta 9
Q1.0	MARCHA_CONVEYOR_10	Bool	Activación cinta 10 final zona azul
Q1.1	MARCHA_CONVEYOR_11	Bool	Activación cinta 11
Q1.2	GIRO_ZONA_1	Bool	Activación giro zona gris
Q1.3	MARCHA_CONVEYOR_12	Bool	Activación cinta 12
Q1.4	GIRO_ZONA_2	Bool	Activación giro zona azul
Q1.5	MARCHA_CONVEYOR_13	Bool	Activación cinta 13
Q1.6	MARCHA_CONVEYOR_14	Bool	Activación cinta 14 final zona gris
Q1.7	MARCHA_CONVEYOR_3	Bool	Activación cinta 3
Q2.0	MARCHA_CONVEYOR_4	Bool	Activación cinta 4

Q2.1	MARCHA_CONVEYOR_5	Bool	Activación cinta 5
Q2.2	MARCHA_CONVEYOR_6	Bool	Activación cinta 6 final zona verde
Q2.3	EMISOR_CAJAS_VERDES	Bool	Emisor cajas medianas
Q2.4	SALIDA_PIEZAS_EN_CAJA_MEDIANA	Bool	Eliminación pieza en la caja
Q2.5	SALIDA_PRODUCTO_2	Bool	Salida caja mediana completa
Q2.6	MARCHA_C17_EMITOR_CAJAS_VERDES	Bool	Activación cinta 17
Q2.7	MARCHA_C18_CONVEYOR_ROBOT_VERDE	Bool	Activación cinta 18
Q3.0	EMISOR_CAJAS_AZULES	Bool	Emisor cajas grandes
Q3.1	MARCHA_C15_EMITOR_CAJAS_AZULES	Bool	Activación cinta 15
Q3.2	MARCHA_C16_CONVEYOR_ROBOT_1	Bool	Activación cinta 16
Q3.3	SALIDA_PIEZAS_EN_CAJA_GRANDE	Bool	Elimina pieza en la caja
Q3.4	SALIDA_PRODUCTO_1	Bool	Salida caja grande completa
Q3.5	EMISOR_CAJAS_GRISES	Bool	Emisor cajas pequeñas
Q3.6	MARCHA_C19_EMITOR_CAJAS_PEQUEÑAS	Bool	Activación cinta 19
Q3.7	MARCHA_C20_CONVEYOR_ROBOT_3	Bool	Activación cinta 20
Q4.0	SALIDA_PIEZAS_EN_CAJA_PEQUEÑA	Bool	Elimina pieza en la caja
Q4.1	SALIDA_PRODUCTO_3	Bool	Salida caja pequeña completa
Q5.0	ROBOT1_GRAB	Bool	Robot azul
Q5.1	ROBOT2_GRAB	Bool	Robot verde
Q5.2	ROBOT3_GRAB	Bool	Robot gris
Q8.0	LUZ_BOTON_MARCHA_CUADRO_PRINCIPAL	Bool	Luz del botón de marcha del cuadro principal
Q8.1	LUZ_BOTON_PARO_CUADRO_PRINCIPAL	Bool	Luz del botón de paro del cuadro principal
Q8.2	LUZ_BOTON_REARME_CUADRO_PRINCIPAL	Bool	Luz del botón de rearme del cuadro principal
Q8.3	ROJO_BALIZA_PRINCIPAL	Bool	Color rojo baliza
Q8.4	NARANJA_BALIZA_PRINCIPAL	Bool	Color naranja baliza

Q8.5	VERDE_BALIZA_PRINCIPAL	Bool	Color verde baliza
QD12	ROBOT2_Y	Real	Coordenada y Robot 2
QD16	ROBOT2_X	Real	Coordenada x Robot 2
QD20	ROBOT2_Z	Real	Coordenada z Robot 2
QD24	ROBOT1_Y	Real	Coordenada y Robot 1
QD28	ROBOT1_X	Real	Coordenada x Robot 1
QD32	ROBOT1_Z	Real	Coordenada z Robot 1
QD36	ROBOT3_Y	Real	Coordenada y Robot 3
QD40	ROBOT3_X	Real	Coordenada x Robot 3
QD44	ROBOT3_Z	Real	Coordenada z Robot 3

5.4 BLOQUES Y FUNCIONES DEL PROGRAMA

En esta parte, se pasará a mostrar cada una de las carpetas, bloques y funciones empleadas en el desarrollo del proyecto. Para ello es necesario acceder al desplegable PLC_1, bloques de programa y se comenzará a observar la estructuración de este.

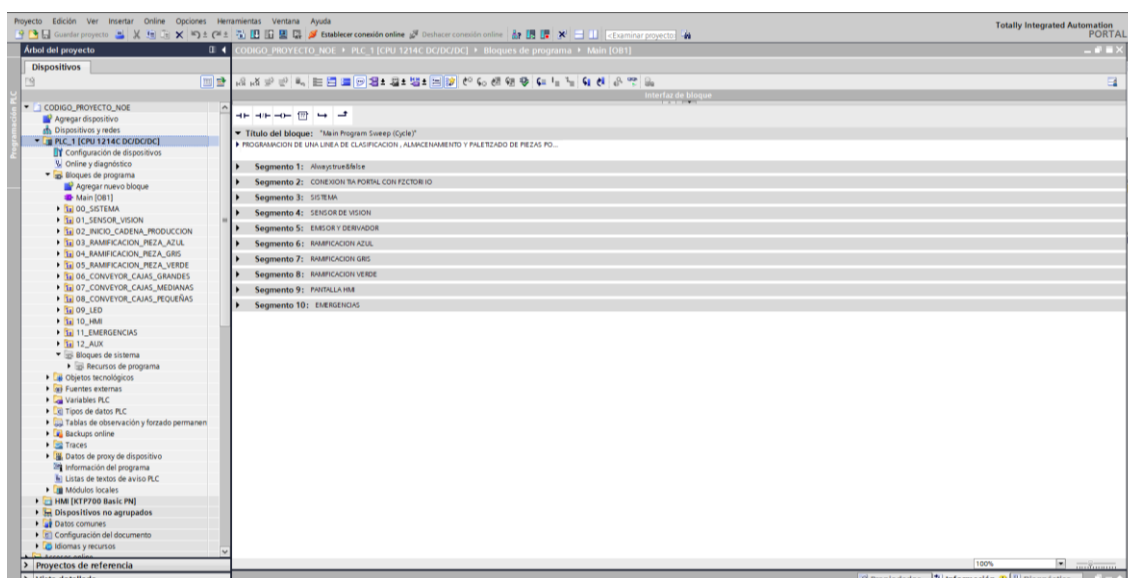


Ilustración 43: Pantalla principal del proyecto en TIA Portal

Como se puede observar en la ilustración 31, la programación de este proyecto se ha llevado a cabo mediante carpetas, las cuales contienen partes fundamentales del mismo. Es una forma de realizar una programación estructurada y de fácil comprensión, ya que todas sus partes guardan un orden y una correlación entre sí.

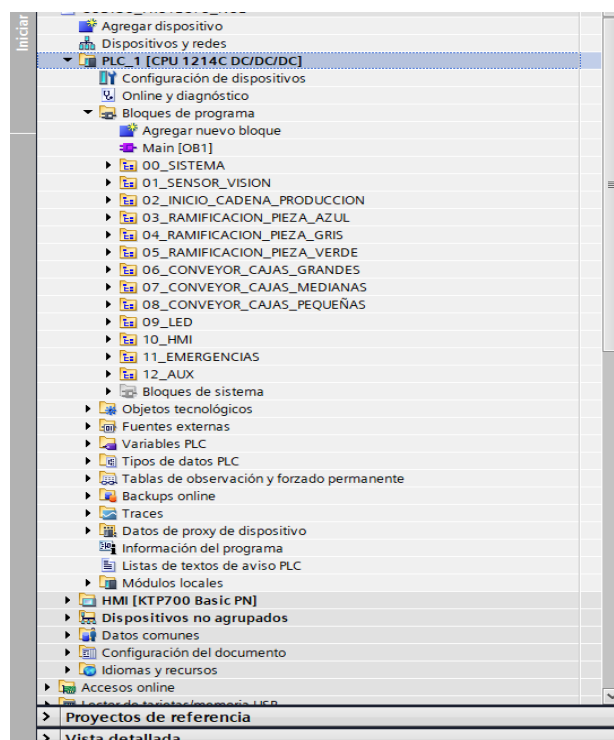


Ilustración 44: Carpetas para la estructuración del proyecto

Se pasarán a explicar cada una de las carpetas, así como cada uno de sus componentes.

Antes de comenzar con la explicación de las carpetas, se observa la existencia del bloque principal OB1. En este bloque se han realizado las llamadas de una de las funciones que se han ido utilizando en las carpetas para el desarrollo del proyecto.

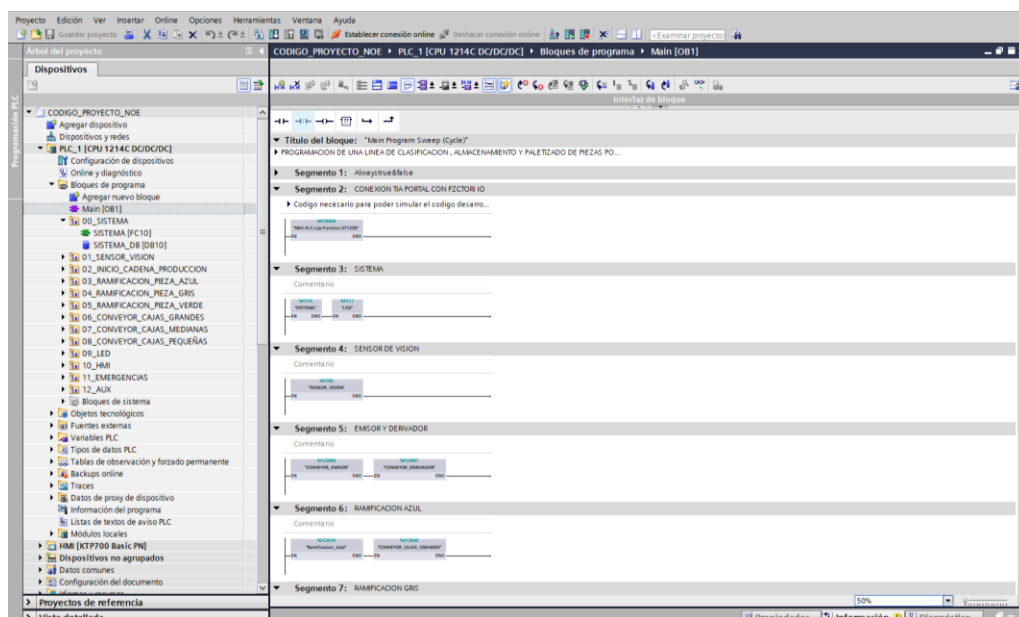


Ilustración 45: Bloque Principal OB1

5.4.1 Sistema

Si se abre la carpeta sistema, se observa un patrón, es un FC y un DB. De forma genérica, por cada carpeta se ha creado un FC donde generaremos todo el código relacionado con la carpeta, y un DB. Se ha creado un DB para estructurar y guardar todas las memorias que necesitemos para esa sección. Cabe señalar que cada sección como mínimo va a tener un FC y un DB.

Dentro del FC de sistema, se pueden observar dos segmentos, inicio de sistema y sistema parado.

En el primer segmento tiene un bloque Set/Reset, con prioridad al Reset, en el que se pondrá a uno el bit de memoria del sistema en marcha. Este bit se encuentra generado en la zona de memoria de su Sistema_DB.

El sistema se pondrá en marcha o bien accionando el botón de la entrada digital del cuadro o por el botón de la pantalla HMI. Ambos botones darían marcha al sistema, siempre y cuando no este accionado el botón de paro desde el HMI o el cuadro principal o no haya ninguna emergencia. En caso de que haya alguna emergencia, el sistema se parará.

En lo referente al segmento dos, sistema parado, es un bit de apoyo, para definir que el sistema no está en marcha.

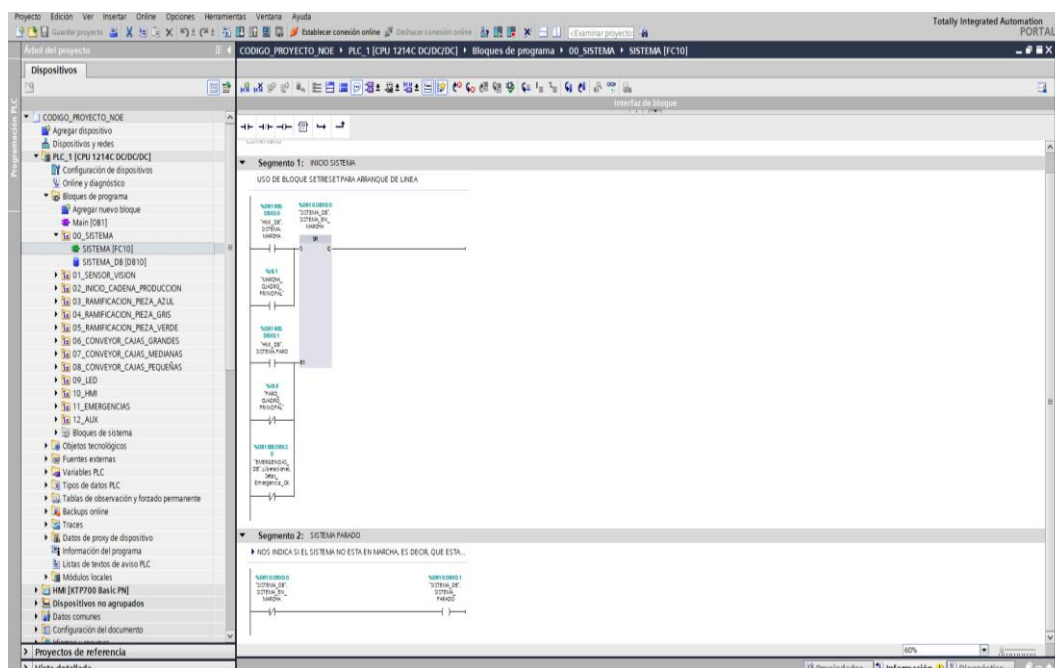


Ilustración 46: Bloque del Sistema

5.4.2 Sensor de visión

Una vez se ha explicado el funcionamiento del arranque de la línea, se pasará a explicar el funcionamiento del sensor de visión.

El sensor de visión es otra carpeta más, en el que internamente se ha creado un FC y un DB, siguiendo así el mismo patrón explicado en líneas superiores. Dentro del FC tendremos programado todo lo relativo al sensor de visión.

En este bloque, lo que se tendrá será que, tras recibir la señal del sensor de visión, que lo vuelca en una doble palabra, se transferirá esta para memorizarla, es decir, se memorizará lo que lee el sensor en pieza memorizada del emisor, que es la primera cinta, donde está situado el sensor.

Por lo tanto, al inicio de cadena, primero llegaría la pieza y luego la leería el sensor, en cuanto este lee la pieza, lo que se hace es traspasar directamente esa lectura a esa zona de memoria. Con esto lo que se conseguirá es poder almacenar y recordar que memoria acaba de pasar por el sensor.

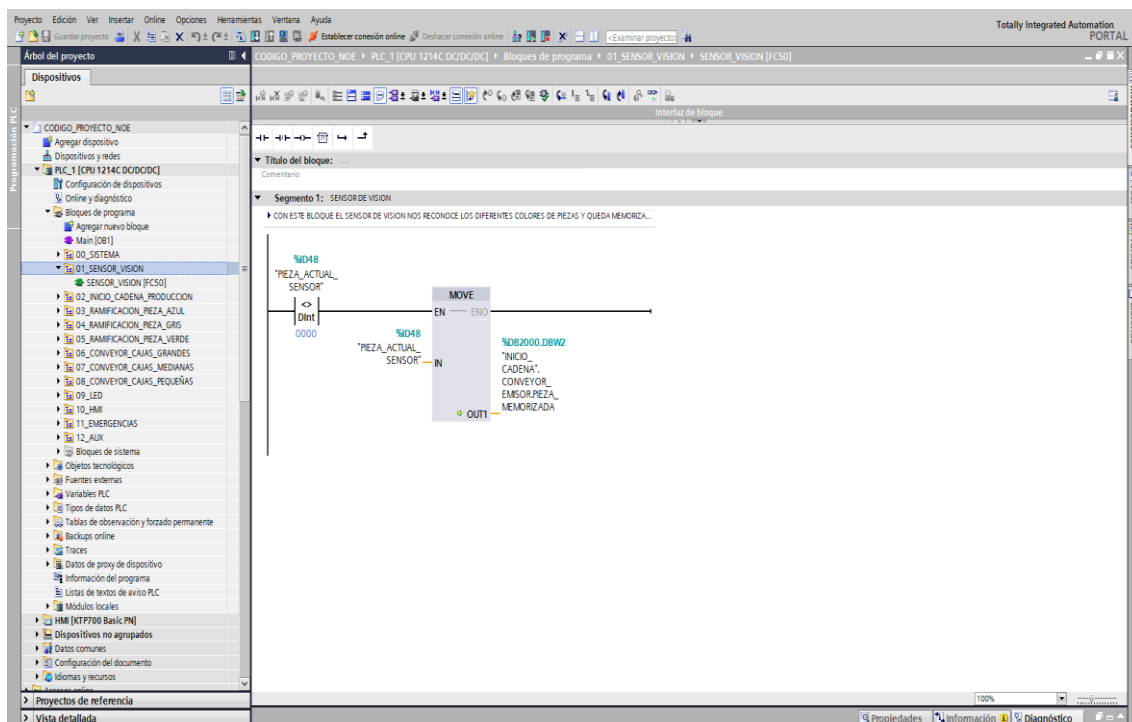


Ilustración 47: Sensor de visión

5.4.3 Inicio cadena de producción

Dentro de inicio de cadena de producción, compuesto por dos cintas transportadoras, se encontrará una cinta emisora, donde llegará la pieza, en esta misma cinta no solo está el emisor de la pieza, sino que también se encuentra situado el sensor de visión, y otra cinta, que será la derivadora. Esta es la encargada de una vez determinado el color de la pieza, derivarla a cada una de las ramificaciones. Del mismo modo que en casos anteriores, esta carpeta de inicio de cadena tiene su zona de memoria, donde estarán almacenados todas las zonas de memoria que van directamente relacionadas con el emisor y con el derivador.

Se comenzará explicando la cinta emisora, sería la primera cinta, y su programación se divide en tres segmentos.

En el primer segmento, lo que se ha programado es la emisión de pieza, es decir, la simulación de que nos llegase una pieza a la cinta. Para poder emitir una pieza, se le da la condición de que el sistema debe estar en marcha.

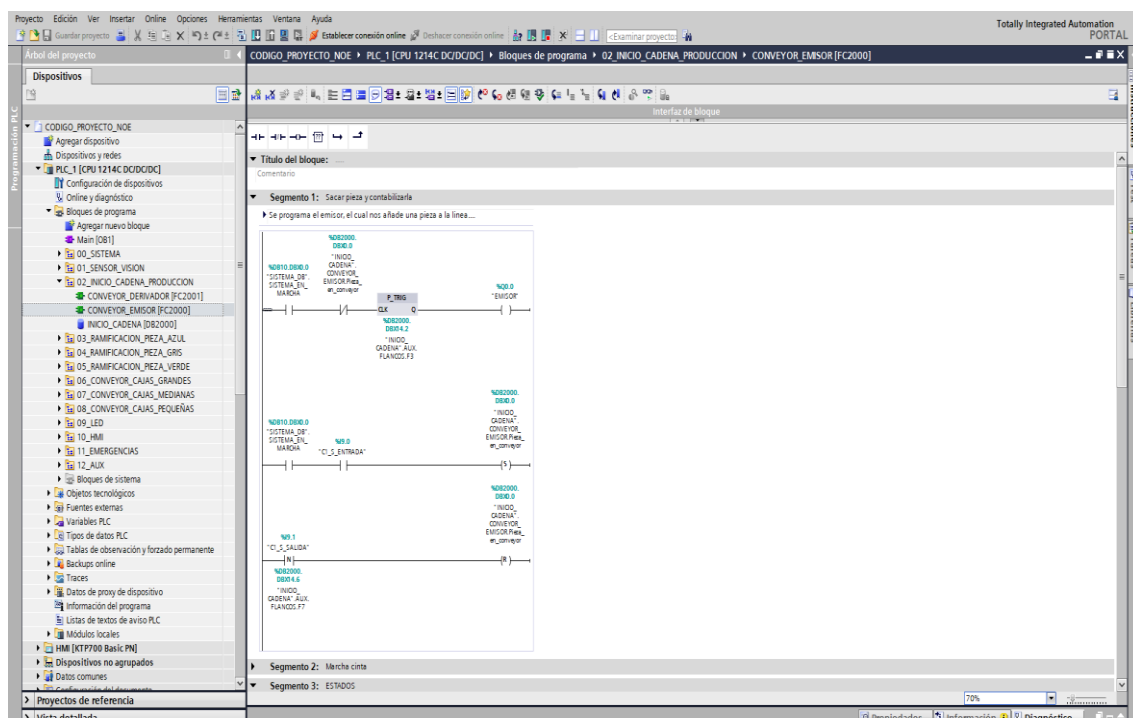


Ilustración 48: Conveyor _emisor (1)

En el siguiente segmento definimos la marcha de la cinta emisora.

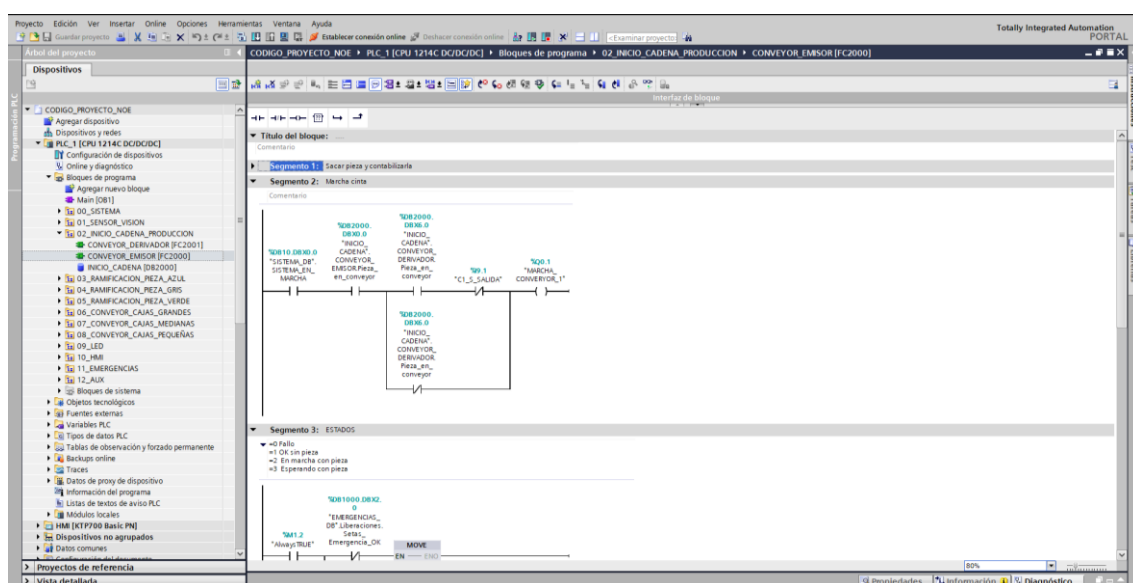


Ilustración 49: Conveyor_emsor (2)

Finalmente, el ultimo segmento de esta cinta emisora, es el referente al estado de este, es decir, se debe conocer lo que ocurre en cada una de las cintas en cada momento, tanto la CPU como mediante la pantalla HMI del operario, por eso se ha generado dentro del DB asignado a esta parte de programa una sección para esta memoria asignada como un entero (doble byte). El numero entero estará directamente relacionado con el estado de la cinta.

Los estados por los que puede pasar una cinta son los siguientes:

0= Fallo

1= Todo Okey, pero no hay pieza

2=En marcha con pieza

3=Esperando con pieza

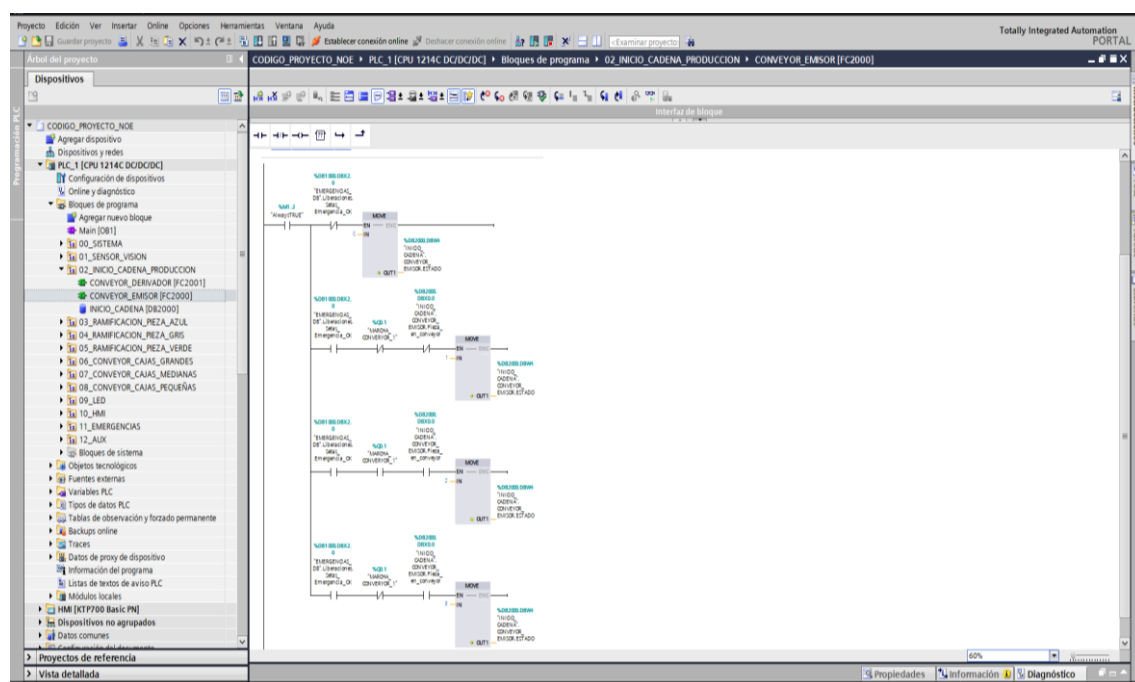


Ilustración 50: Conveyor_emisor (3)

Una vez terminada la explicación de la cinta emisora, se pasará a explicar las partes que componen la segunda cinta derivadora.

El derivador es una cinta, que, una vez identificada la pieza, la derivará a la ramificación correspondiente, no solo traspasara la pieza, sino también la información de la pieza que es.

Está constituido por una programación algo más compleja. Los elementos necesarios para la realización de su función también son algo más complejos.

Al igual que en la cinta anterior, esta está constituido por ocho segmentos que se explicaran a continuación.

En el primer segmento, si el sistema esta en marcha y el sensor de entrada del derivador ha leído el paso de pieza, lo que primero se define es que hay pieza en la cinta, y lo segundo, se ejecutará la transmisión de información de la pieza memorizada, es decir, el color, a la zona de memoria del derivador que hemos creado previamente, para saber el color que se tendrá en el derivador.

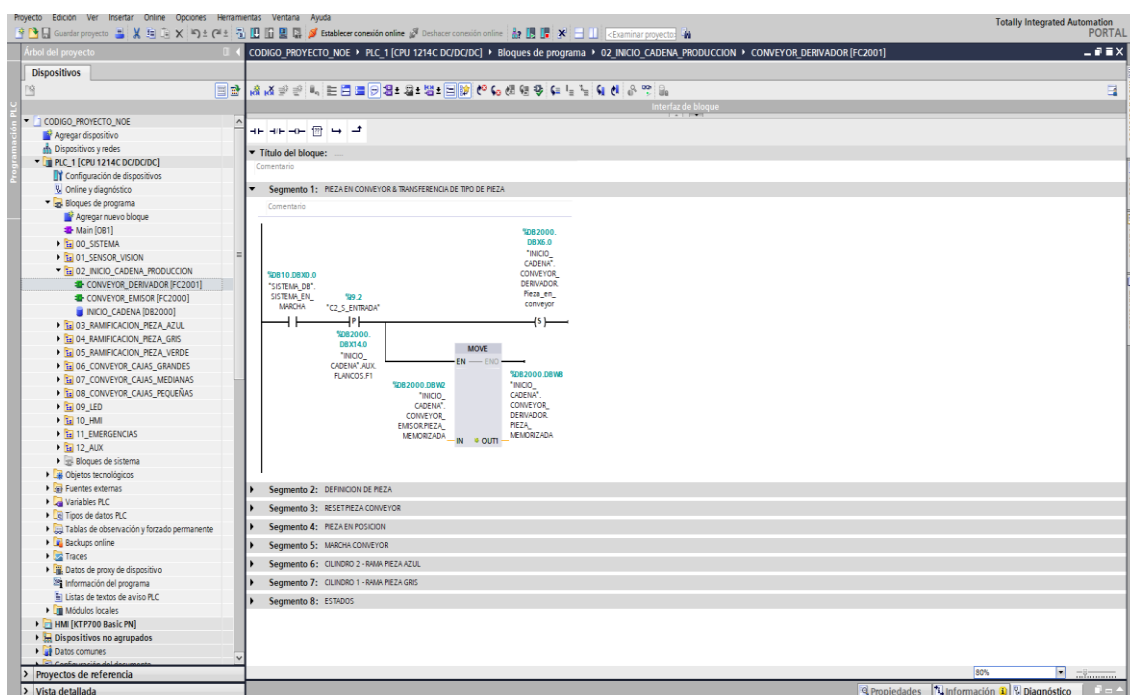


Ilustración 51: Conveyor_derivador (1)

En el segmento dos, lo que se hará es definir si la pieza memorizada es igual a los valores de los colores establecidos previamente, es decir, 1 = Azul, 4= verde, 7 = gris. Con esto lo que se conseguirá es definir el color de la pieza que se encuentra en la cinta, para con posterioridad poder actuar y ramificar o derivar dichas piezas.

En caso de no haber pieza en la cinta, el valor de la memoria en el derivador será 0.

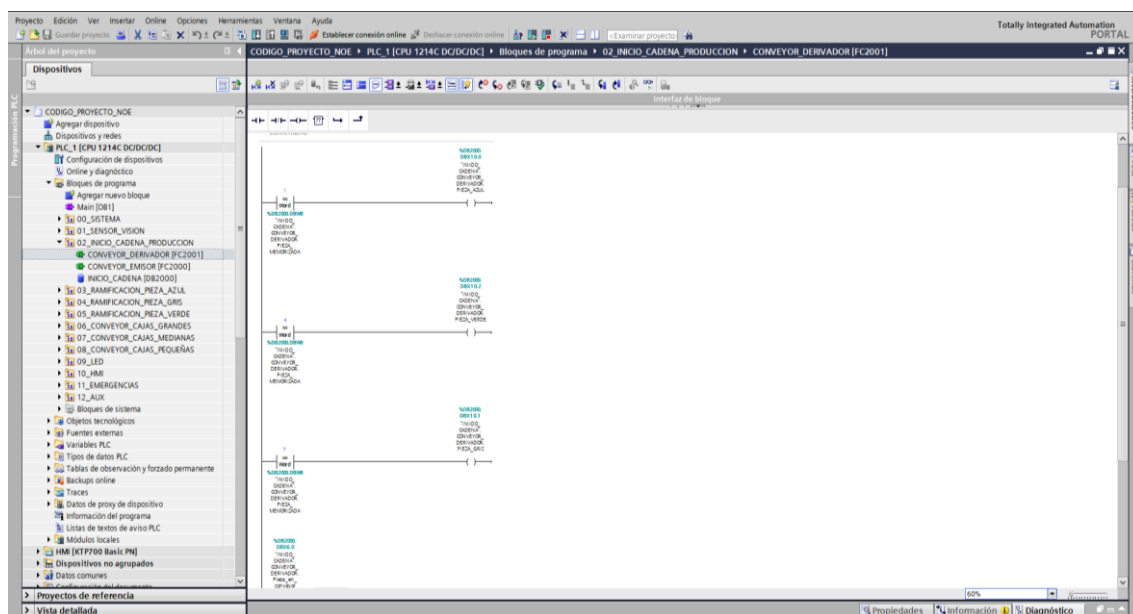


Ilustración 52: Conveyor_derivador (2)

Para el segmento tres, de la misma forma que se definirá que hay pieza en el derivador, también se tiene que poner a cero el bit. Por lo tanto, se pondrá a uno el bit de la pieza en la cinta en los casos donde se detecte la caída de los flancos de los sensores de entrada a las ramificaciones, es decir, cuando el sensor de entrada a cualquiera de los tres ramales deje de detectar la pieza, se pondrá a cero el bit de la pieza en la cinta del derivador.

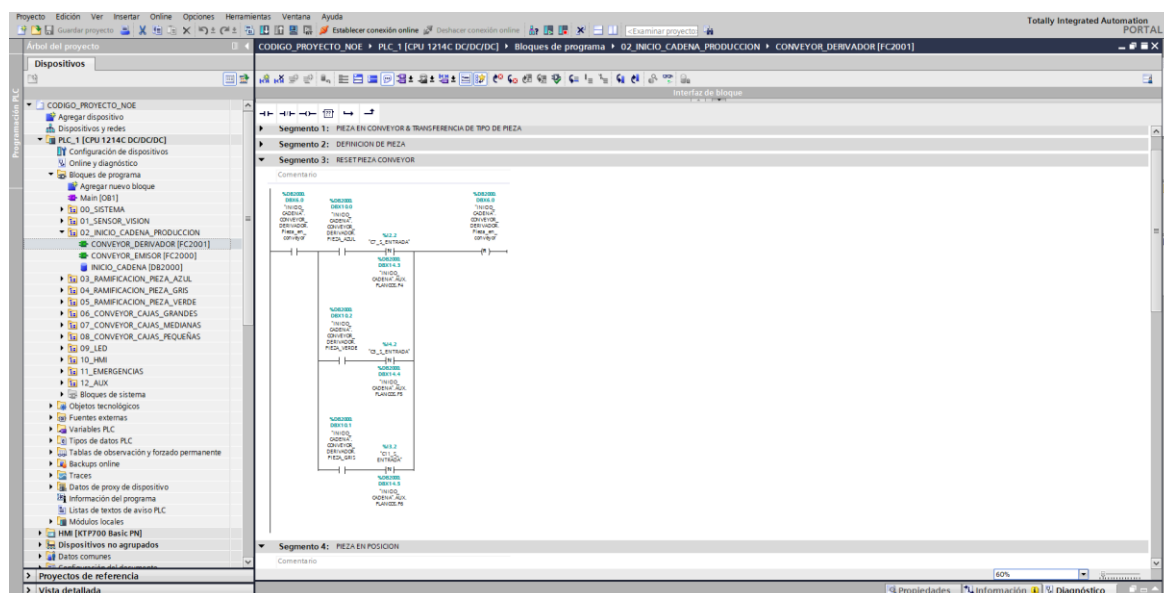


Ilustración 53: Conveyor_derivador (3)

Como se puede observar, el segmento cuatro, es el encargado de determinar que la pieza gris y azul están en posición para que los cilindros neumáticos actúen. En el caso de la pieza verde no es necesario ya que no tiene cilindro, esta continuará hasta llegar a la zona del pick and place.

Se pondrá a cero el bit que marca que está en posición cuando se detecta que han pasado ya a las ramificaciones.

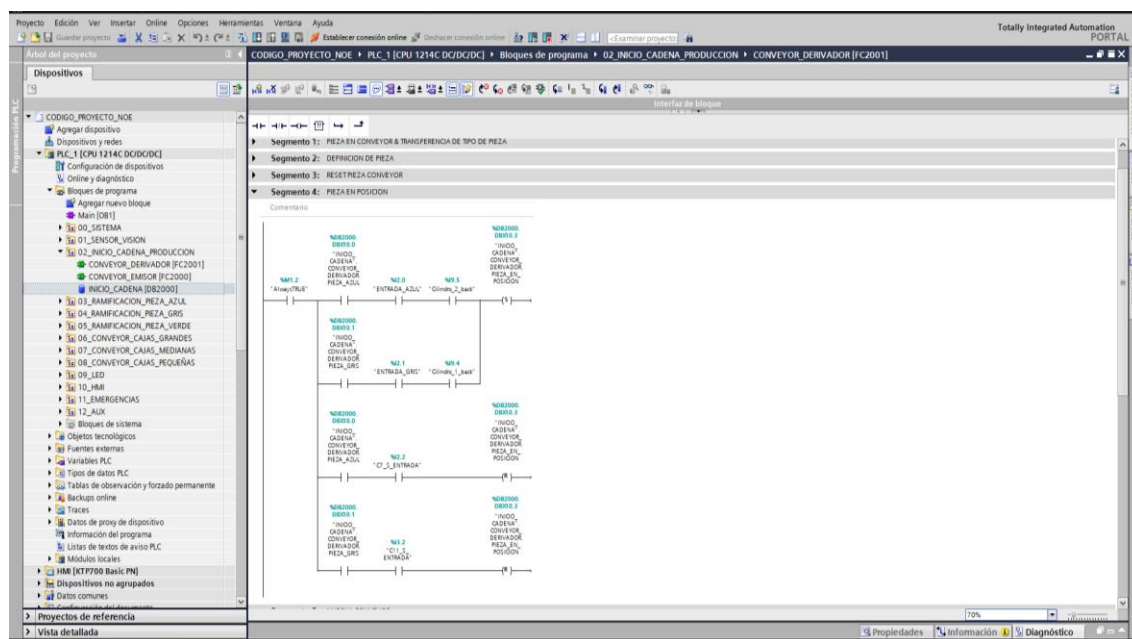


Ilustración 54: Conveyor_derivador (4)

Una vez las piezas ya han sido clasificadas por el sensor y derivadas por los cilindros, llegará al segmento cinco, que es el encargado de definir cuando se dará marcha al derivador.

Si hay pieza azul o gris, parará cuando se detecte la presencia de esta en frente del cilindro, mientras que si se tiene pieza verde se pondrá en marcha la cinta todo el rato siempre y cuando no haya pieza en la siguiente cinta, en caso de haberla, se pasará justo en el sensor de salida del derivador, esperando a que quede libre la siguiente.

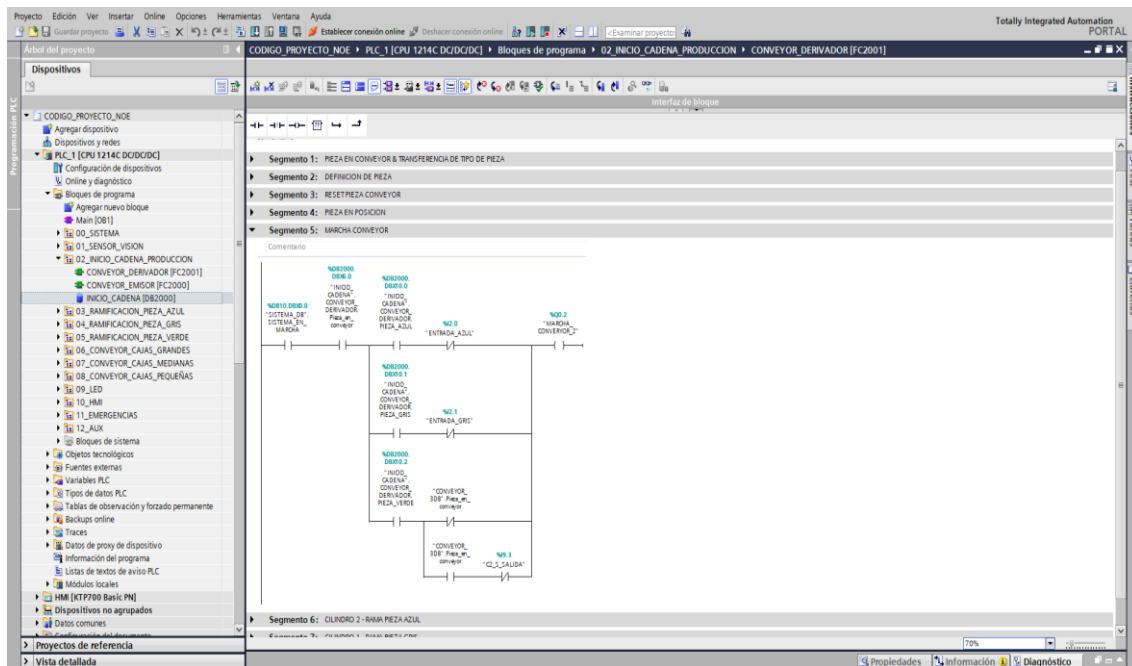


Ilustración 55: Conveyor_derivador (5)

Los segmentos seis y siete, son los encargados de activar los cilindros para que desplacen la pieza a la ramificación determinada cumpliendo una serie de condiciones.

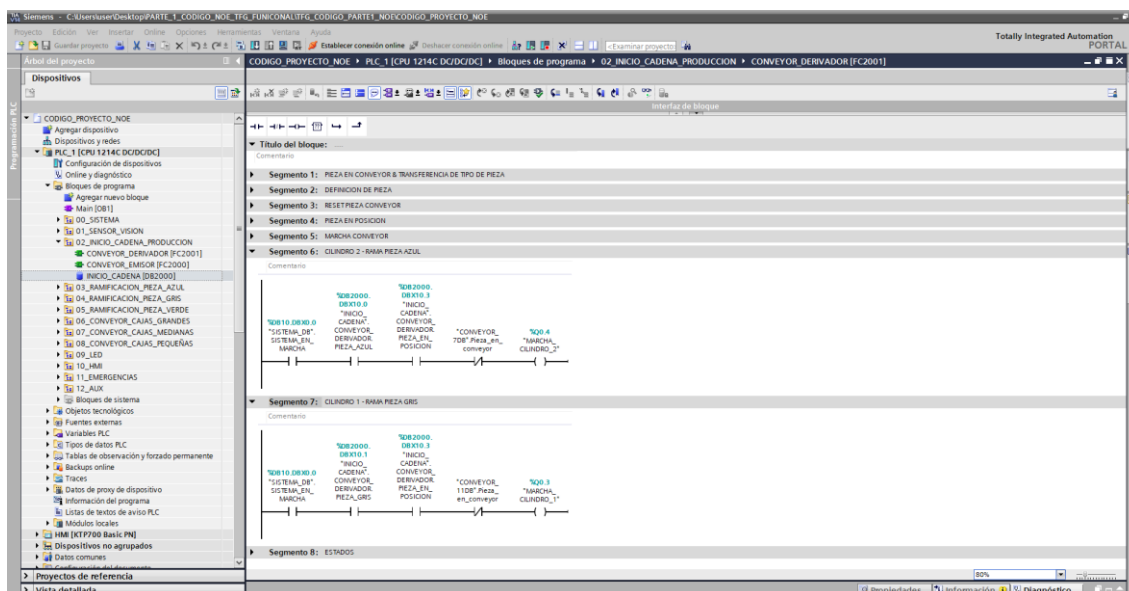


Ilustración 56: Conveyor_derivador (6 y 7)

Finalmente, al igual que ocurre con la cinta emisora, el segmento ocho de la cinta derivadora es el referente a los estados de la cinta.

Siendo estos:

0= Fallo

1= Todo Okey, pero no hay pieza

2=En marcha con pieza

3=Esperando con pieza

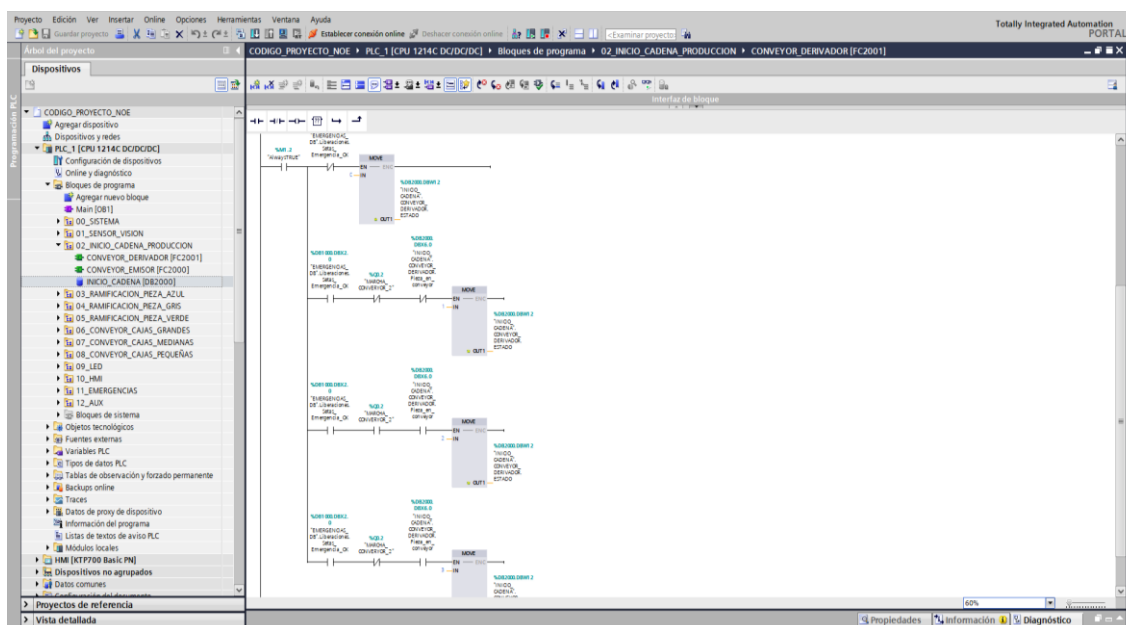


Ilustración 57: Conveyer_derivador (8)

Para finalizar, se remarcará, que dentro del DB de inicio de cadena se puede observar la misma estructura que en casos anteriores. Esta está constituida por una estructura para la cinta emisora, otra para la derivadora y finalmente otra para los auxiliares, ya sean flancos o Timers.

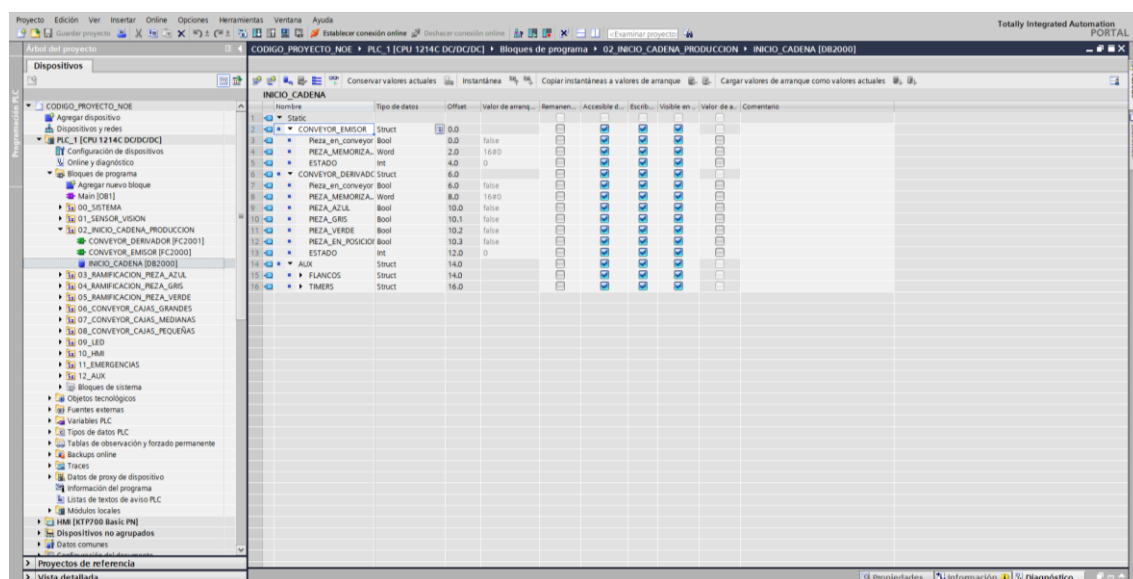


Ilustración 58: Estructura del DB inicio de cadena

Al final, la instalación se compone por una serie de cintas, una detrás de otra. Que lo que hacen es transportar las piezas desde el derivador hasta el robot. Como hay una serie de cintas puestas en serie unas detrás de otra, y dicha programación iba a ser muy repetitiva, se ha decidido crear un FB, que se encuentra en la carpeta AUX.

En primer lugar, se ha creado un FB para las cintas transportadoras y otro FB para las cintas finales que sirven de unión entre la ramificación y el Robot. A ambos FB se les llamará en cada ramificación generando un DB propio para cada cinta denominado DB de instancia.

5.4.4 Ramificaciones

5.4.4.1 Pieza azul

En las ramificaciones se programa todo el paso de la pieza desde el derivador hasta ser cogida por el robot para su introducción en la caja. Esta ramificación está formada por cuatro segmentos.

En primer lugar, hay que señalar que la ramificación azul esta compuesta por las cintas genéricas 7, 8 y 9 así como por una cinta final 10.

El primer segmento corresponde a la cinta 7, que es el paso del cilindro a la cinta.

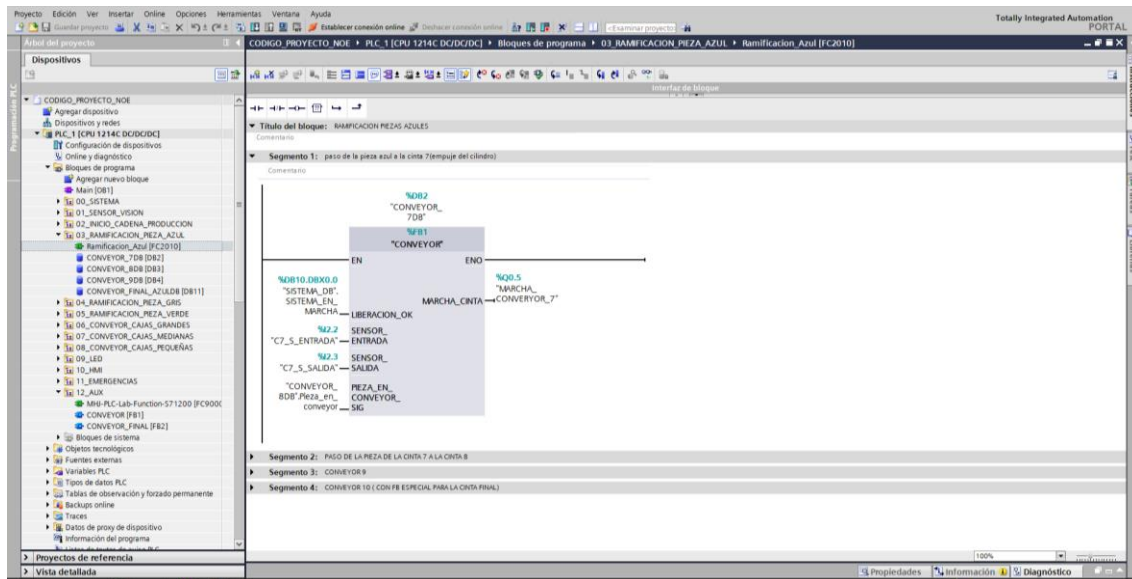


Ilustración 59: Ramificación Azul Conveyor 7

Del mismo modo se han programado los segmentos dos y tres, correspondientes a las cintas 8 y 9. Invocando el FB1 y personalizando los DB de instancia para cada cinta.

Cabe señalar que el segmento dos correspondiente a la cinta 8 tiene la particularidad de la incorporación de una cinta llamada giro.

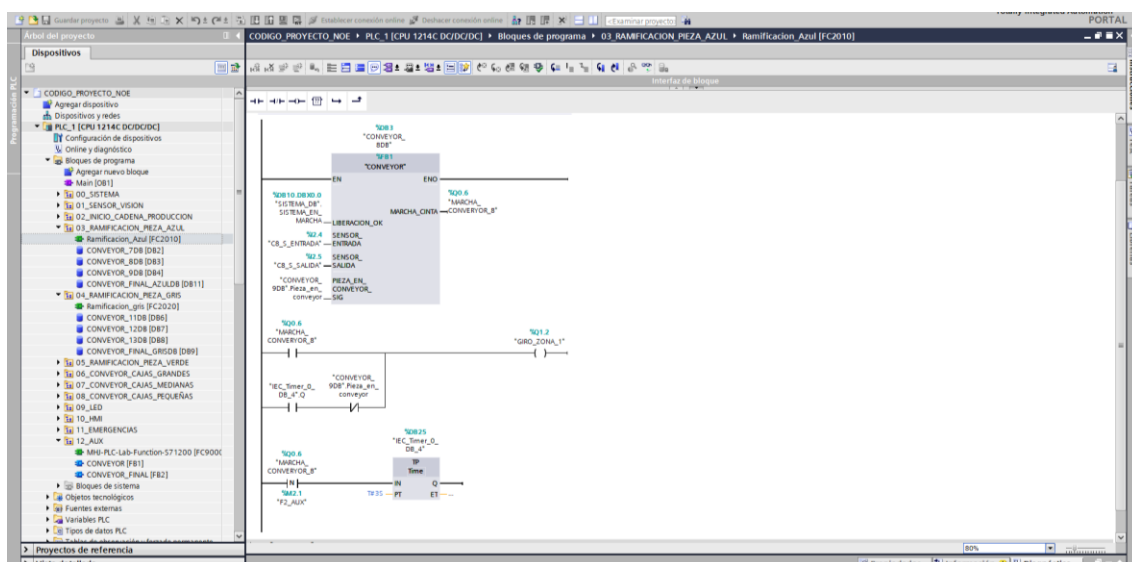


Ilustración 60: Ramificación Azul Conveyor 8

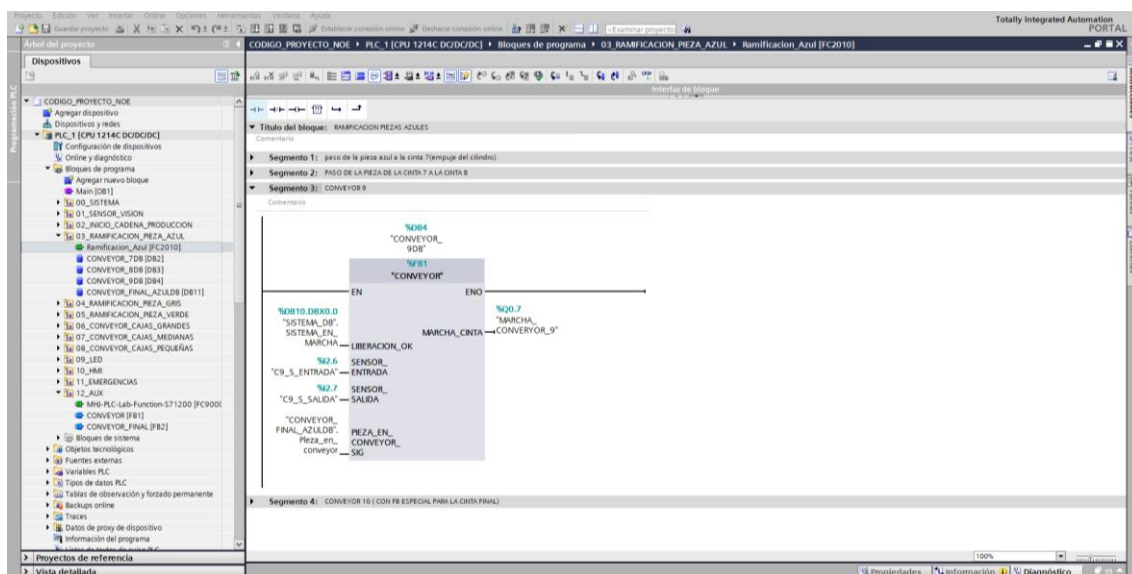


Ilustración 61: Ramificación Azul Conveyor 9

Finalmente, se encuentra el segmento cuatro, propio de la cinta final 10. Hay que indicar que la entrada al robot indica que el robot ha cogido ya la pieza. Por lo tanto, le da a entender a la cinta que puede pasar una pieza nueva a esta.

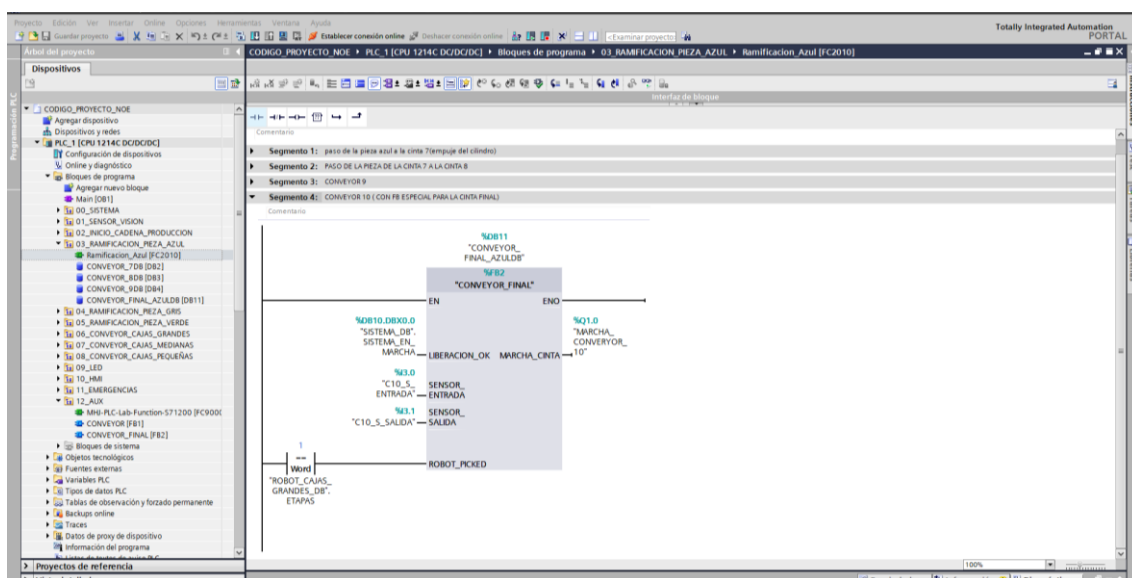


Ilustración 62: Ramificación Azul Conveyor Final 10

5.4.4.2 Pieza gris

Todo lo explicado en la ramificación azul es aplicable para la gris. Ya que esta constituida por la misma estructura en sus cintas.

En primer lugar, hay que señalar que la ramificación gris está compuesta por las cintas genéricas 11,12 y 13, así como por una cinta final 14.

El primer segmento corresponde a la cinta 11, que es el paso del cilindro a la cinta.

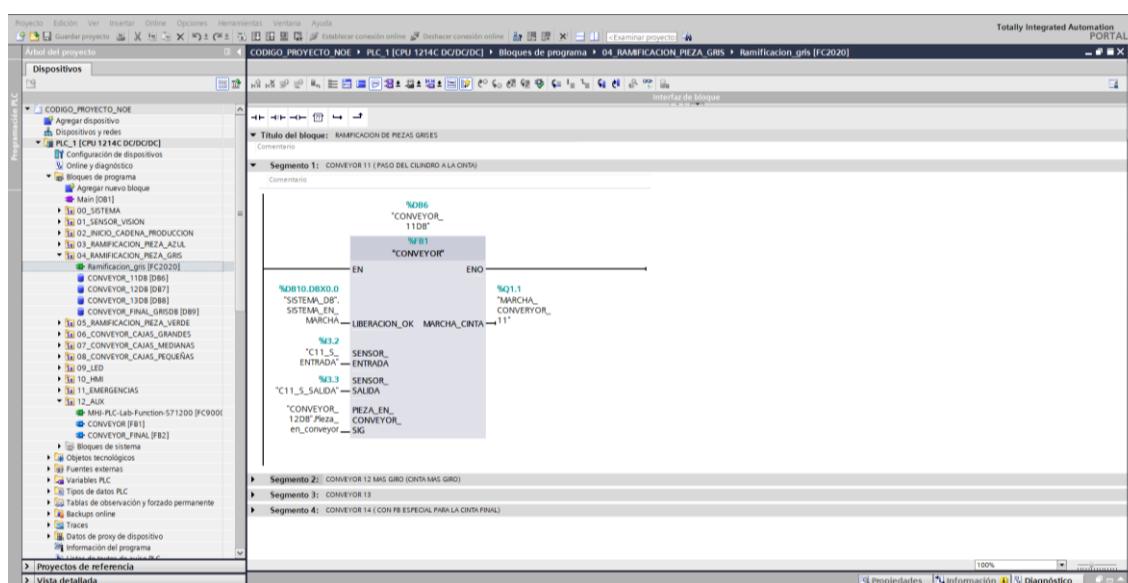


Ilustración 63: Ramificación Gris Conveyor 11

Del mismo modo se han programado los segmentos dos y tres, correspondientes a las cintas 12 y 13. Invocando el FB1 y personalizando los DB de instancia para cada cinta.

Cabe señalar que el segmento dos correspondiente a la cinta 12 tiene la particularidad de la incorporación de una cinta llamada giro.

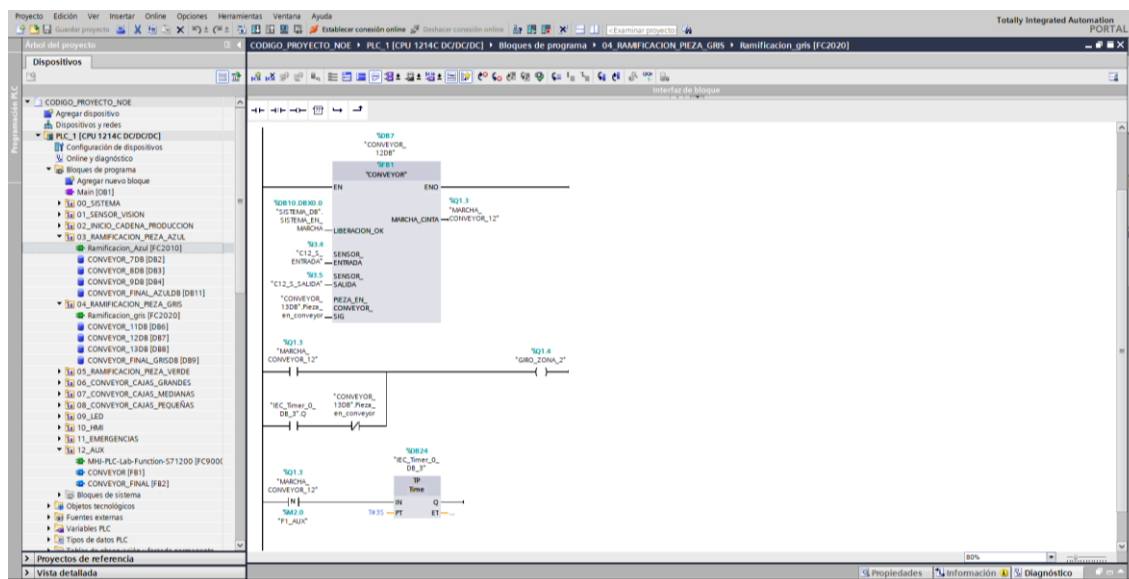


Ilustración 64: Ramificación Gris Conveyor 12

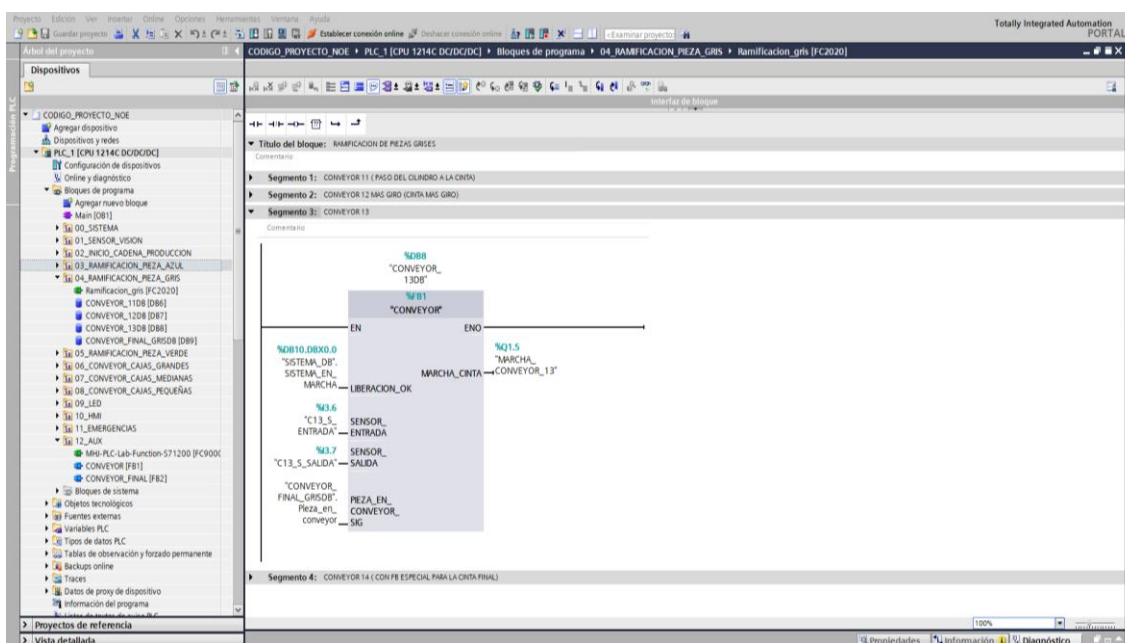


Ilustración 65: Ramificación Gris Conveyor 13

Finalmente, se encuentra el segmento cuatro, propio de la cinta final 14.

Hay que indicar que la entrada al robot indica que el robot ha cogido ya la pieza. Por lo tanto, le da a entender a la cinta que puede pasar una nueva a esta.

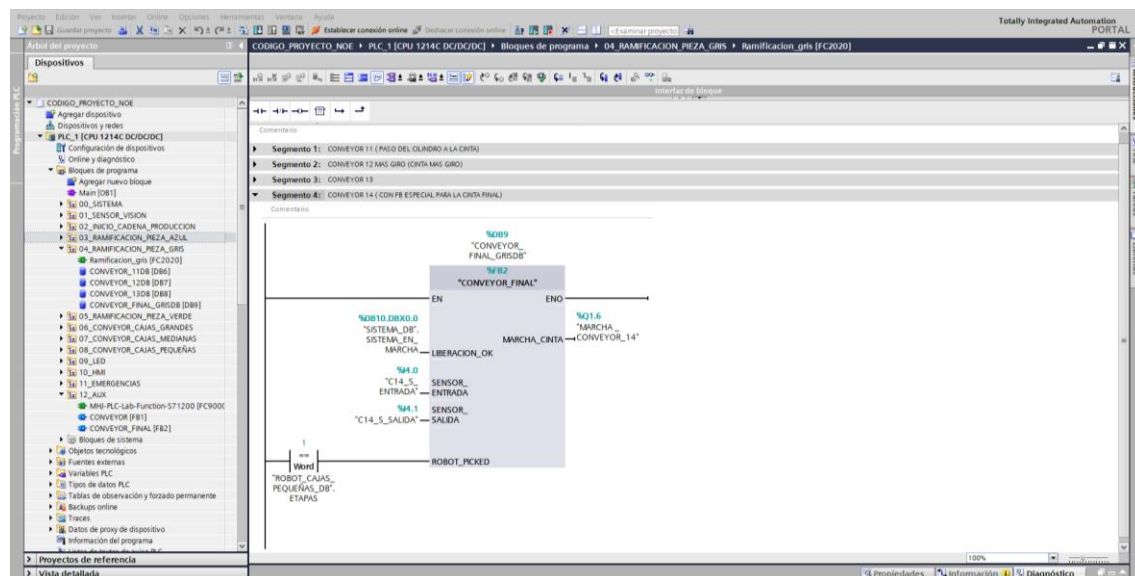


Ilustración 66: Ramificación Gris Conveyor Final 14

5.4.4.3 Pieza verde

Para esta última ramificación, se usará el mismo método que con las anteriores.

En primer lugar, cabe señalar que la ramificación verde está compuesta por las cintas genéricas 3, 4 y 5, así como por una cinta final 6.

A diferencia de las ramificaciones gris y azul, esta no tiene giro, las piezas continúan su trayecto en línea recta. El primer segmento corresponde a la cinta 3, al no emplear cilindros, este segmento solo pondrá en marcha la cinta.

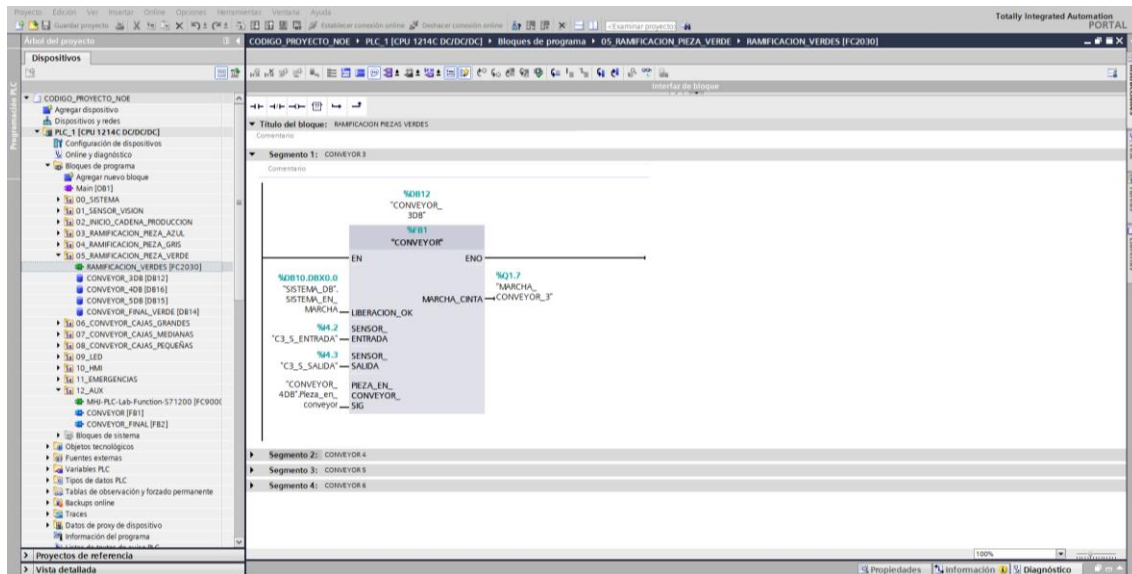


Ilustración 67: Ramificación Verde Conveyor 3

Del mismo modo, se han programado los segmentos dos y tres relativos a las cintas genéricas 4 y 5. Como se ha destacado en esta ramificación no hay giro.

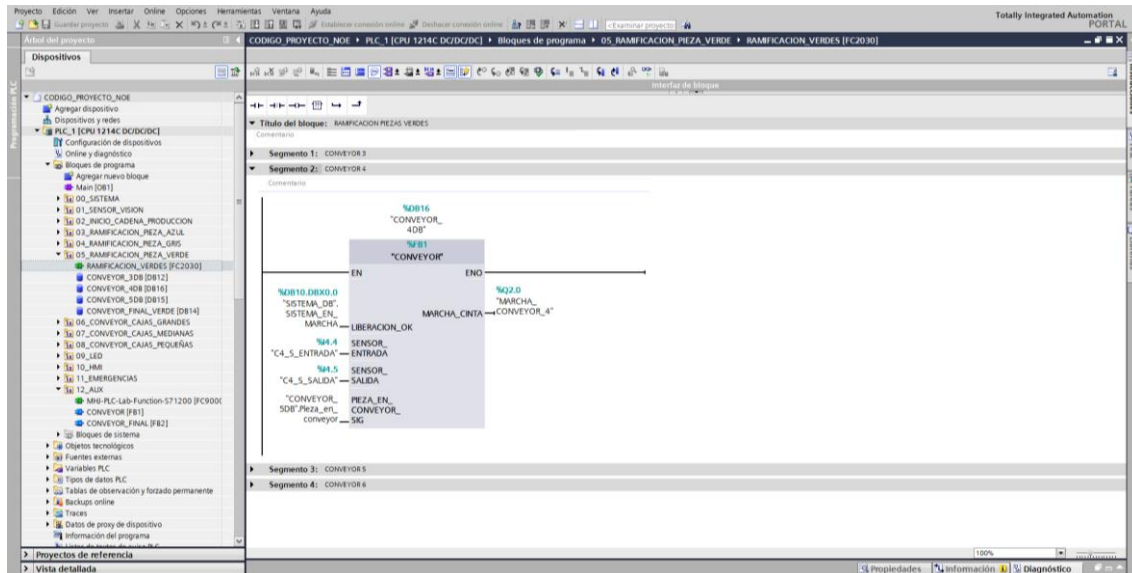


Ilustración 68: Ramificación Verde Conveyor 4

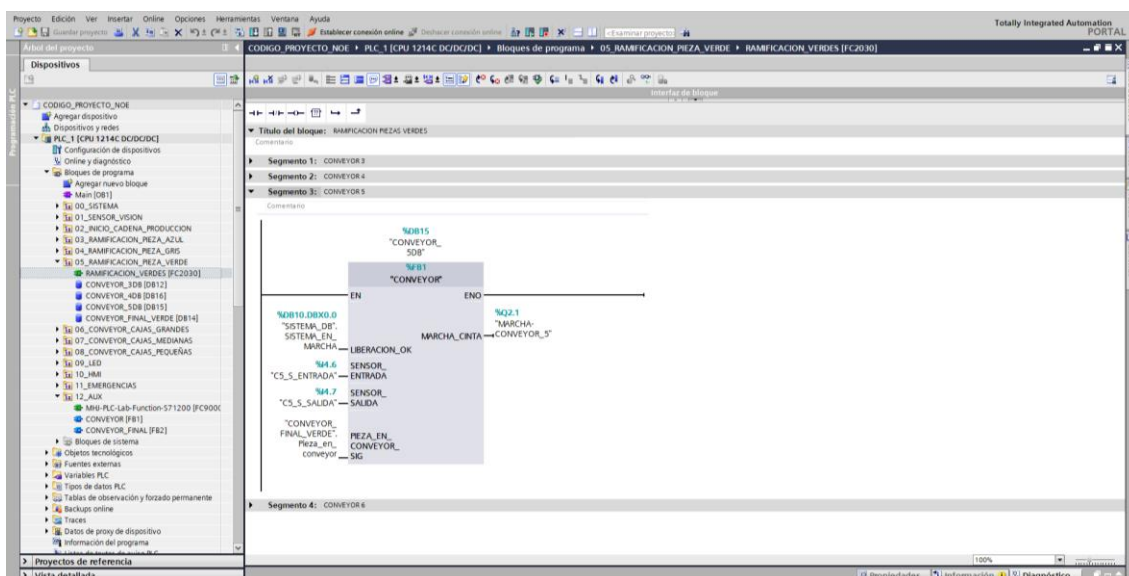


Ilustración 69: Ramificación Verde Conveyor 5

Finalmente, se encuentra el segmento cuatro, propio a la cinta final 6.

Hay que remarcar que la entrada al robot indica que el robot ha cogido ya la pieza. por lo tanto, le da a entender a la cinta que puede pasar una pieza nueva esta.

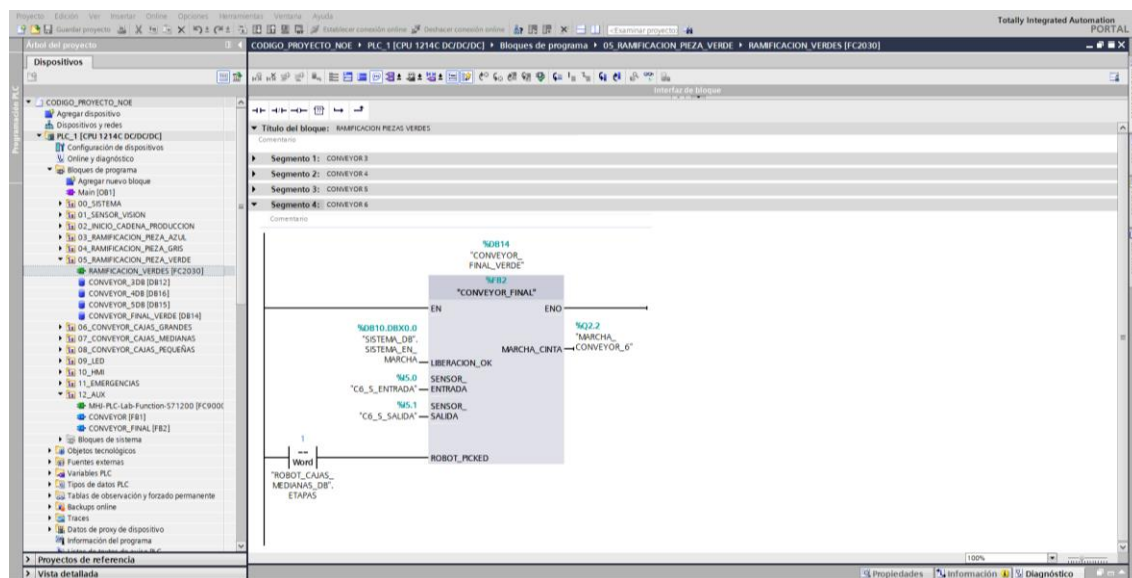


Ilustración 70: Ramificación Verde Conveyor Final 6

Antes de explicar las cintas de las cajas, cabe indicar que se explicará el de las cajas grandes, ya que las otras dos cintas referentes a las cajas medianas (carpeta 7) y pequeñas

(carpeta 8) tienen el mismo código y por lo tanto se ejecutan de la misma manera, por este hecho se considera innecesario repetir la explicación del mismo código para los 3 casos ya que como se ha indicado es el mismo en todos.

5.4.5 Cintas cajas

5.4.5.1 Cajas grandes

Si abrimos la carpeta referente a cinta cajas grandes podemos ver dos añadidos más, en este caso podemos ver el FC de cinta cajas grandes donde tendremos programado como el emisor como el derivador como el FB del robot.

Cada Robot tiene un FB diferente debido a que sus coordenadas varían, por lo tanto, se podrá hacer un FB repetitivo e insertarlo.

La estructuración viene dada por un FC donde está la programación de esta sección, un FB donde estará programado el robot y un par de DBs.

Se comenzará explicando la programación del FC. Remarcar que se ha tenido que volver a programar un emisor para simular la llegada de la caja.

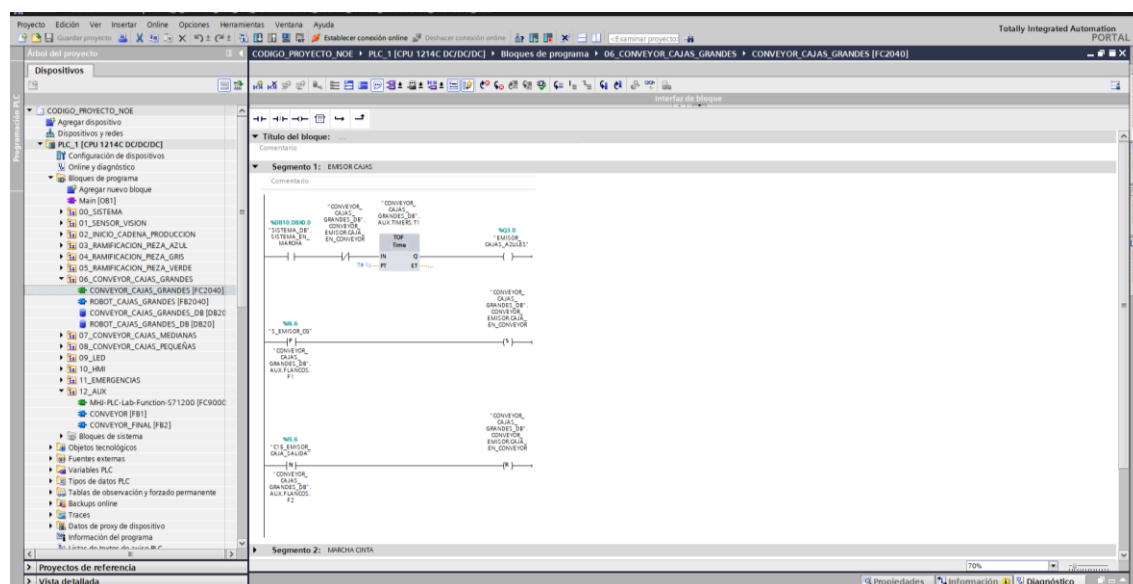


Ilustración 71: Cajas Grandes _FC (1)

En el segmento dos, se activará marcha a la cinta del emisor de cajas.

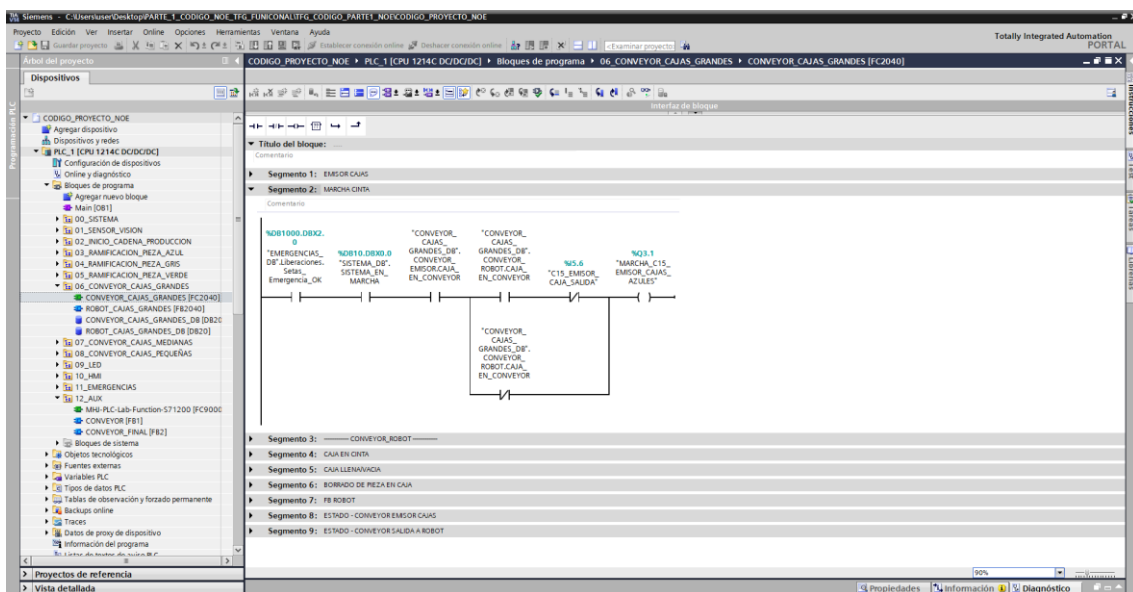


Ilustración 72: Cajas Grandes _FC (2)

La cinta del robot es algo más complejo que el resto, ya que, una vez recibida la caja, deberá posicionarla donde el robot va a dejar la pieza, y una vez introducidas las piezas, se accionará de nuevo la cinta para que la caja llena se dirija a la estación de paletizado.

Como podemos observar en el segmento tres, se pondrá en marcha la cinta que trasportará la caja hasta la posición indicada.

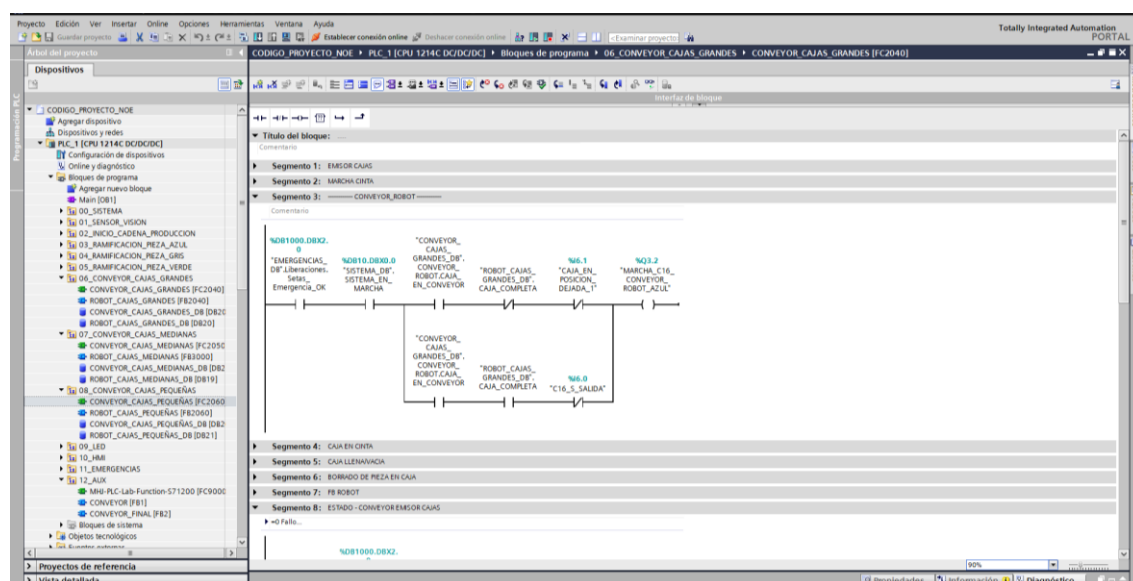


Ilustración 73: Cajas Grandes _FC (3)

Del mismo modo, se podrá ver como en el segmento cuatro se determina que la caja está en la cinta, poner a uno el bit mediante el sensor de entrada de la cita, y poner a cero con el sensor de salida de este.

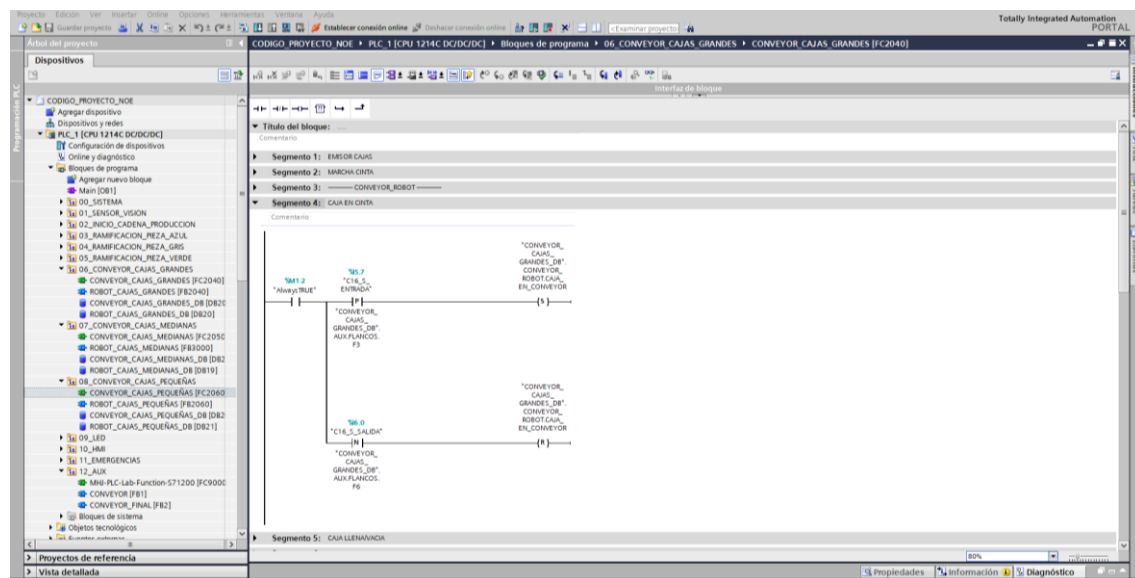


Ilustración 74: Cajas Grandes _FC (4)

A continuación, una vez el robot esta metiendo las piezas en la caja, se chequeará si esta esta completa.

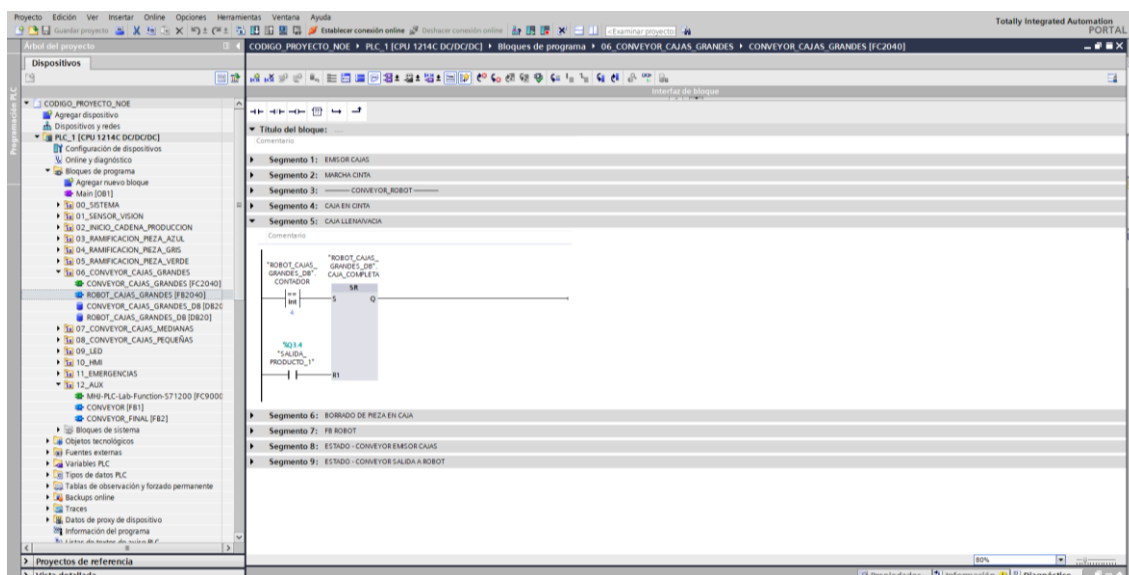


Ilustración 75: Cajas Grandes _FC (5)

Una vez la caja esta completa, se reactiva la cinta y esta sale para la zona de paletización, al ser una simulación y estar compuesta por dos softwares que no se pueden conectar entre si como son Factory I/O y RobotStudio, se introducirá el elemento de salida de producto que es el que simularía la transición a la célula de paletización.

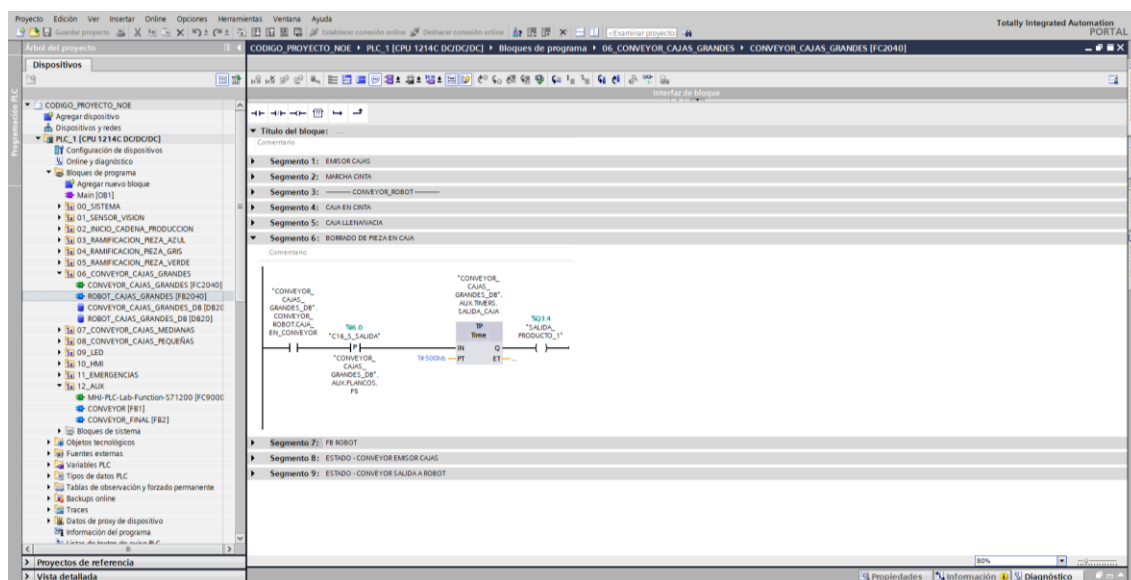


Ilustración 76: Cajas Grandes _FC (6)

Para el segmento siete, lo que se lleva a cabo es efectuar la llamada al FB del robot, donde se encontrarán las etapas que sigue este en la recogida y dejada de la pieza en la caja. Ese código se comentará más tarde.

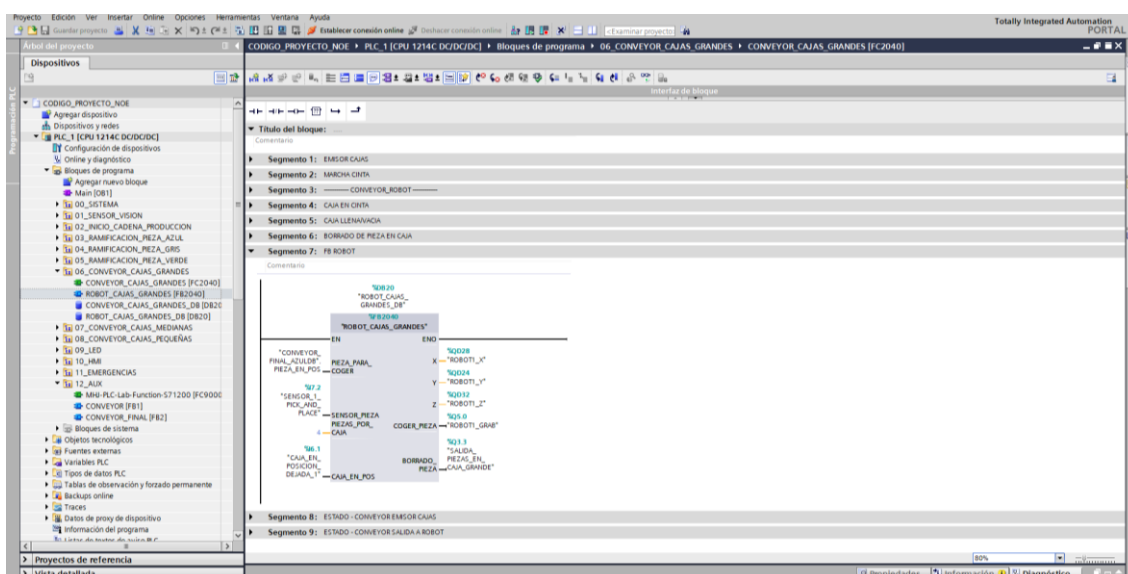


Ilustración 77: Cajas Grandes _FC (7)

Para los dos últimos segmentos, que son los referentes a los estados, remarcar que son exactamente idénticos a los segmentos que se han empleado tanto para la cinta emisora como para la derivadora o en el caso de las ramificaciones.

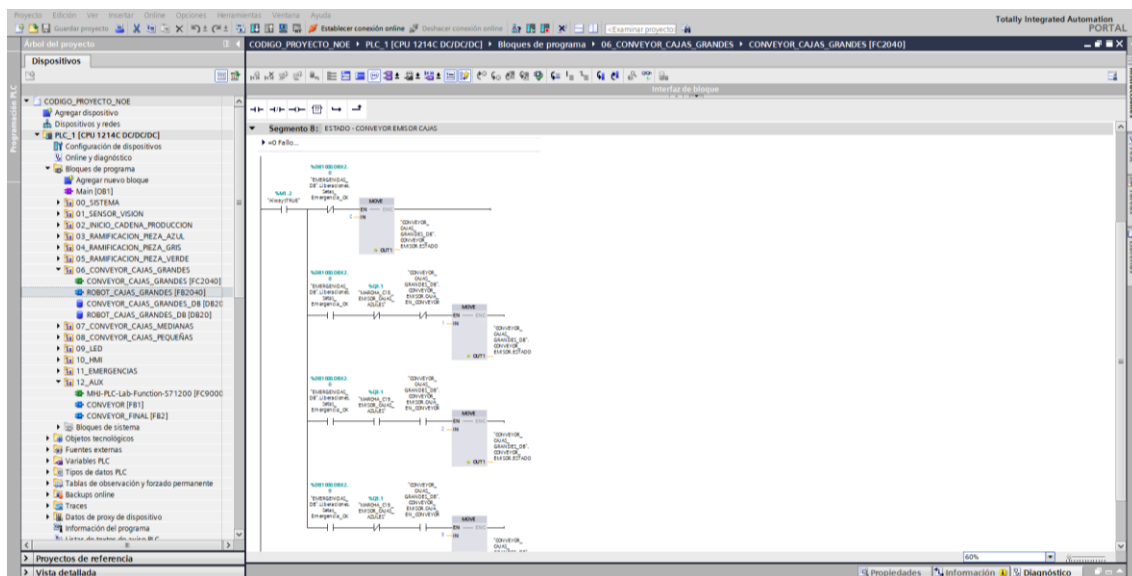


Ilustración 78: Cajas Grandes _FC (8)

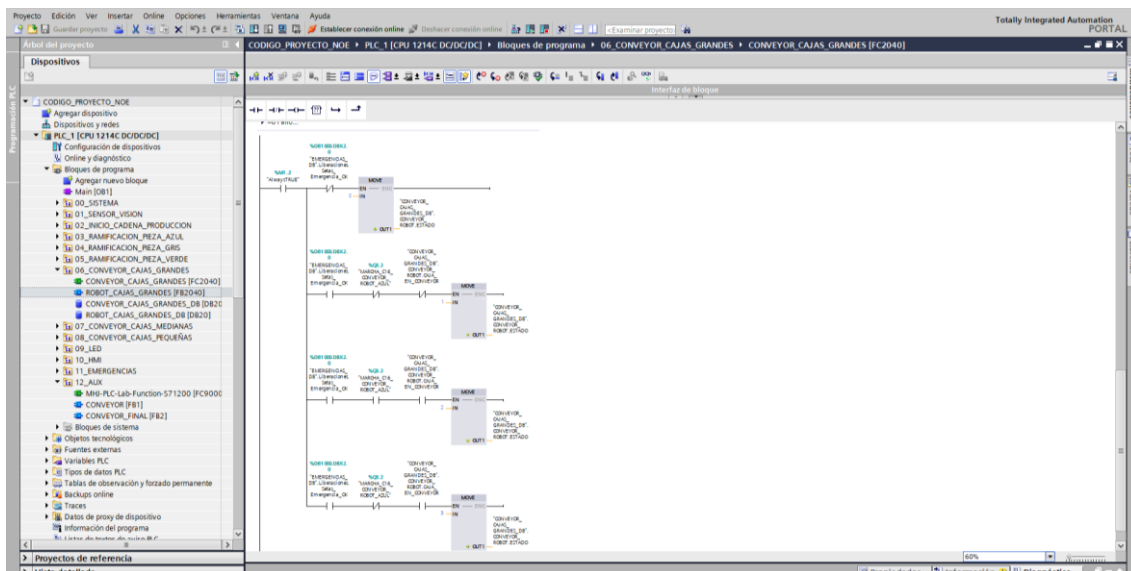


Ilustración 79: Cajas Grandes _FC (9)

Una vez explicada la programación del FC se pasa a explicar la programación propia desarrollada para el FB del Robot.

Del mismo modo que ocurre con el FC de las cajas, la programación del FB del robot es la misma para los 3, lo único que varía son sus coordenadas, por lo tanto, se pasará a explicar el FB del robot correspondiente a la cinta cajas grandes.

Como ya ha ocurrido con otros FB, en este, se han programado variables que se emplearán a lo largo del programa en el interfaz de bloque.

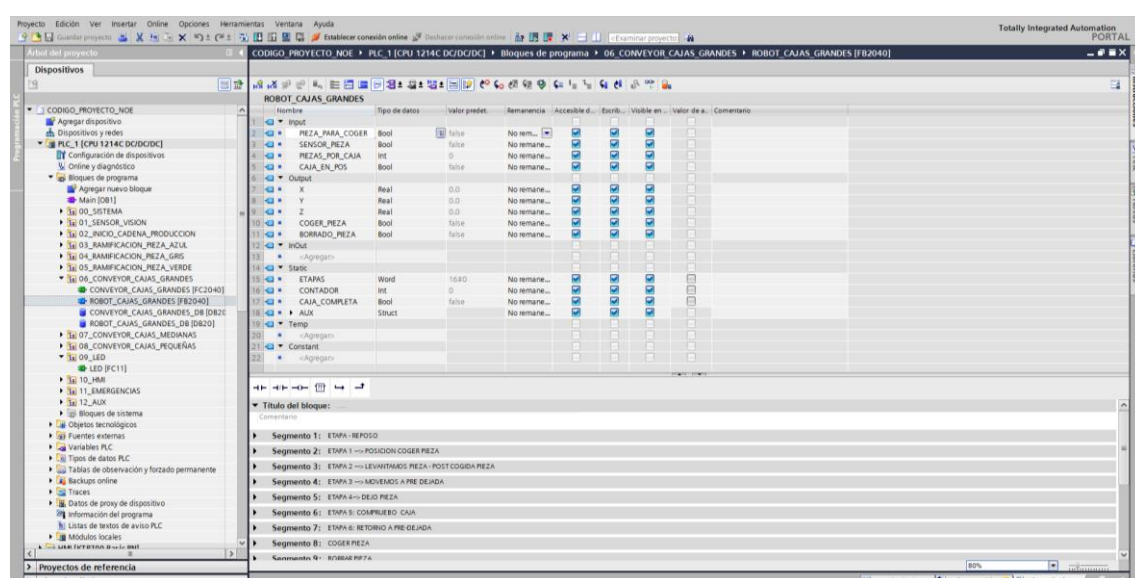


Ilustración 80: Interfaz de Bloque FB del Robot

Este FB se divide en diez segmentos que se pasarán a explicar a continuación. Se comenzará programando la etapa del reposo, en donde el robot está esperando a recibir las señales oportunas que le indiquen que tanto la pieza como la caja están en posición.

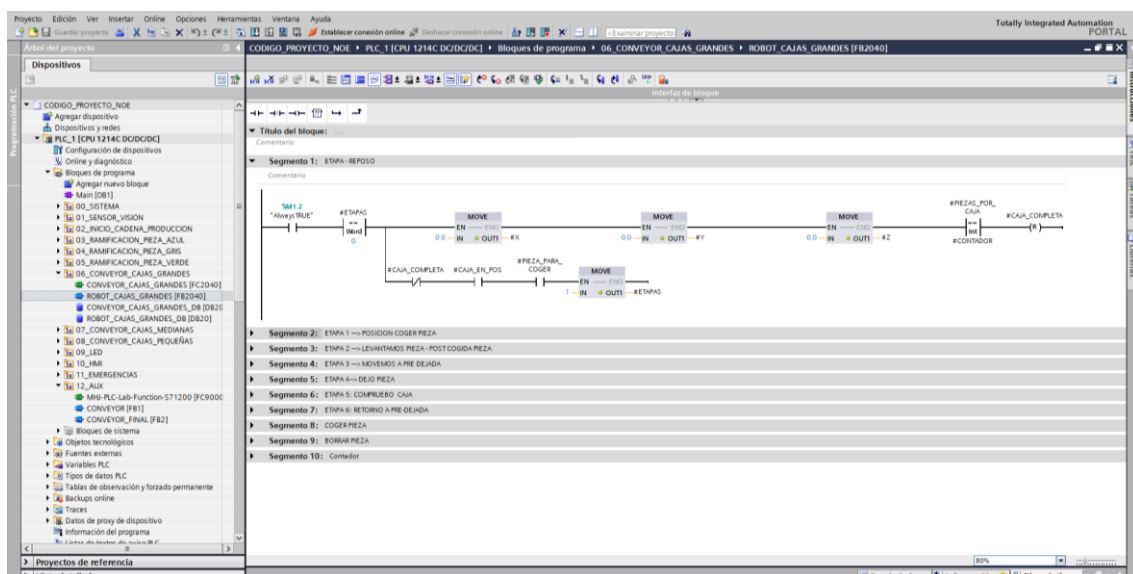


Ilustración 81: Cajas Grandes_FB (1)

A continuación, en la etapa 1, el robot se encuentra en la posición de coger pieza.

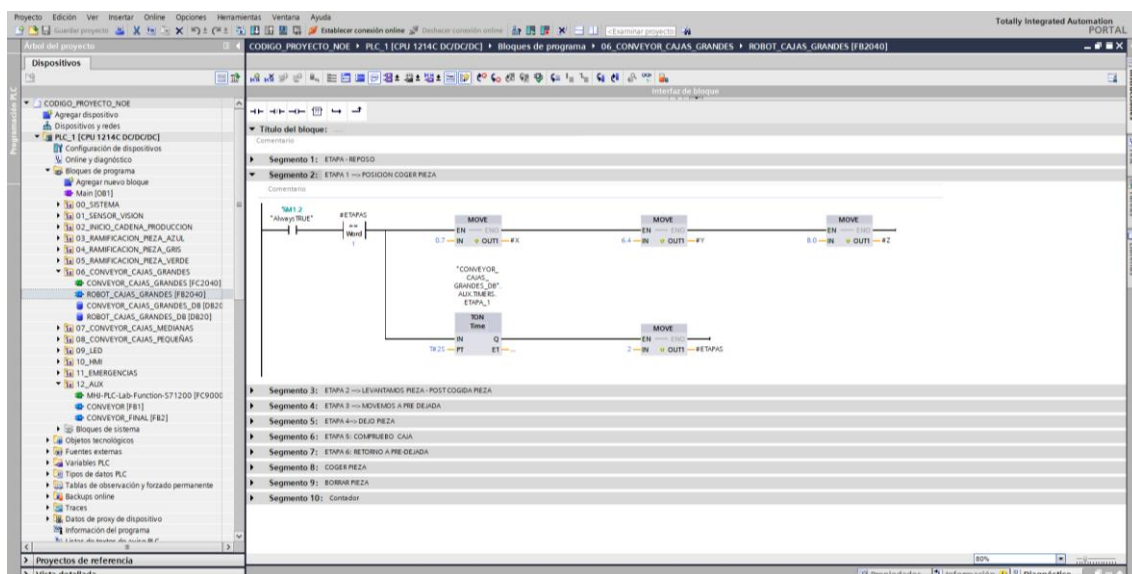


Ilustración 82: Cajas Grandes_FB (2)

Una vez el robot coge la pieza, pasa a la etapa de post cogida, donde elevará la pieza para su posterior desplazamiento.

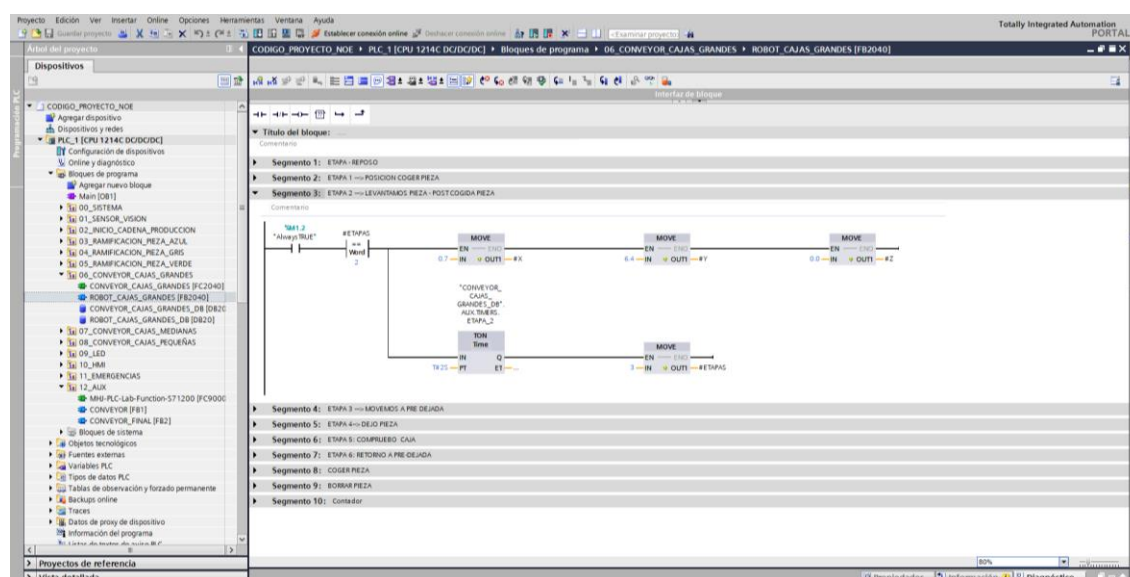


Ilustración 83: Cajas Grandes _FB (3)

Para el segmento cuatro, se ha programado la etapa de moverse a pre dejada, este movimiento será el anteriormente realizado a la introducción de la pieza en su caja correspondiente.

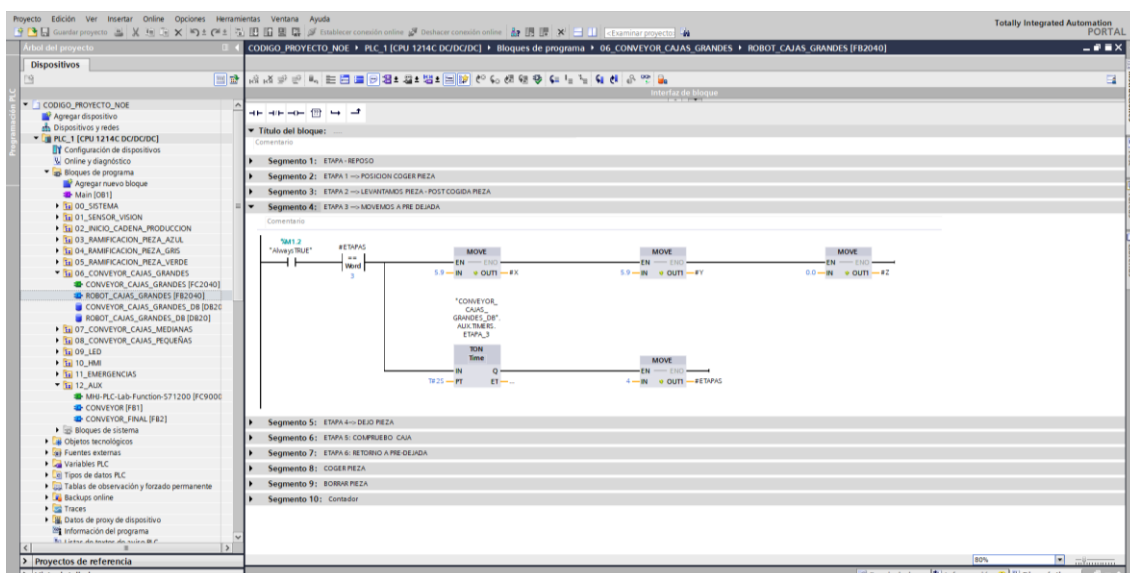


Ilustración 84: Cajas Grandes _FB (4)

Y una vez este situado en pre- dejada, se ejecuta la etapa de dejado de pieza.

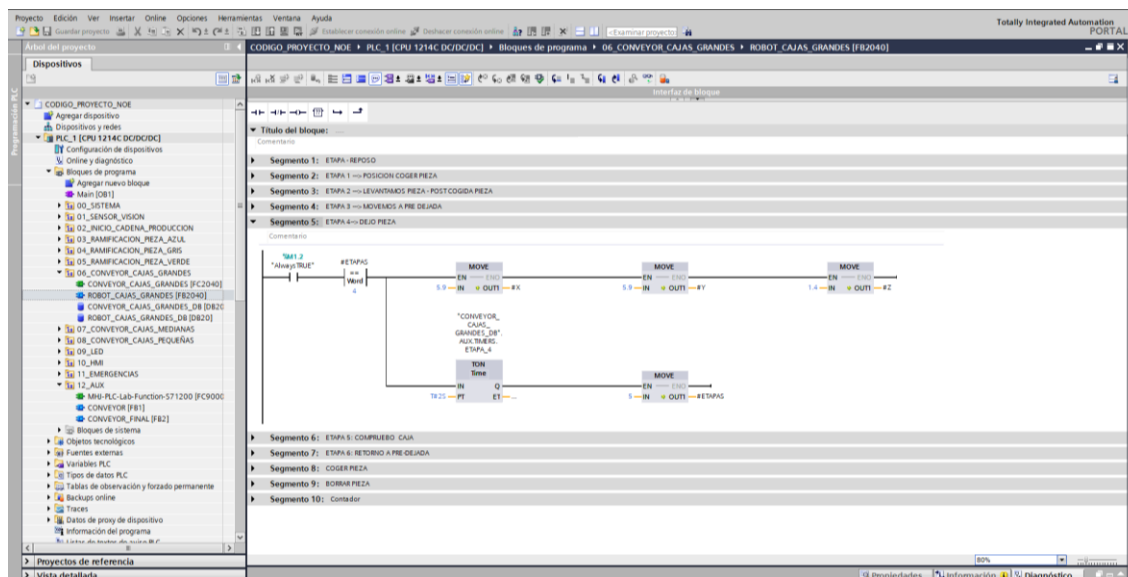


Ilustración 85: Cajas Grandes _FB (5)

Como la dejada de piezas se ejecuta un número de veces determinado para que la caja quede completa, se ha programado la etapa de comprobación de caja, donde se chequea si la caja está llena o no. En caso de estar la caja completa, pasa a la etapa 0 donde se repite el proceso. Si no está completa el robot seguirá metiendo piezas en esta.

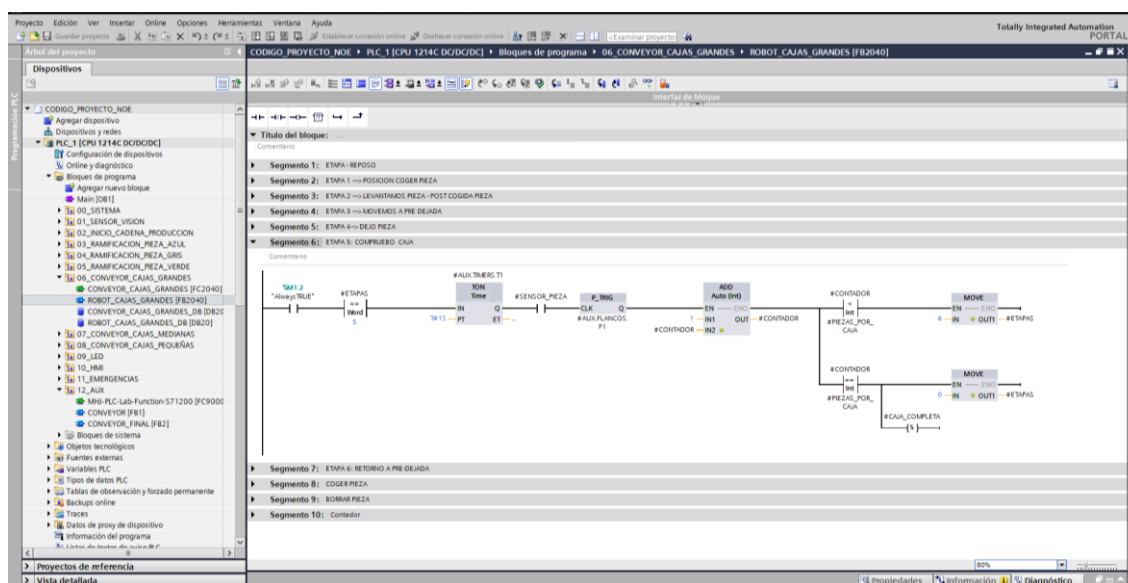


Ilustración 86: Cajas Grandes _FB (6)

Como se ha indicado, tanto si la caja esta completa como si no, el robot retorna a pre dejada esperando a ejecutar una acción.

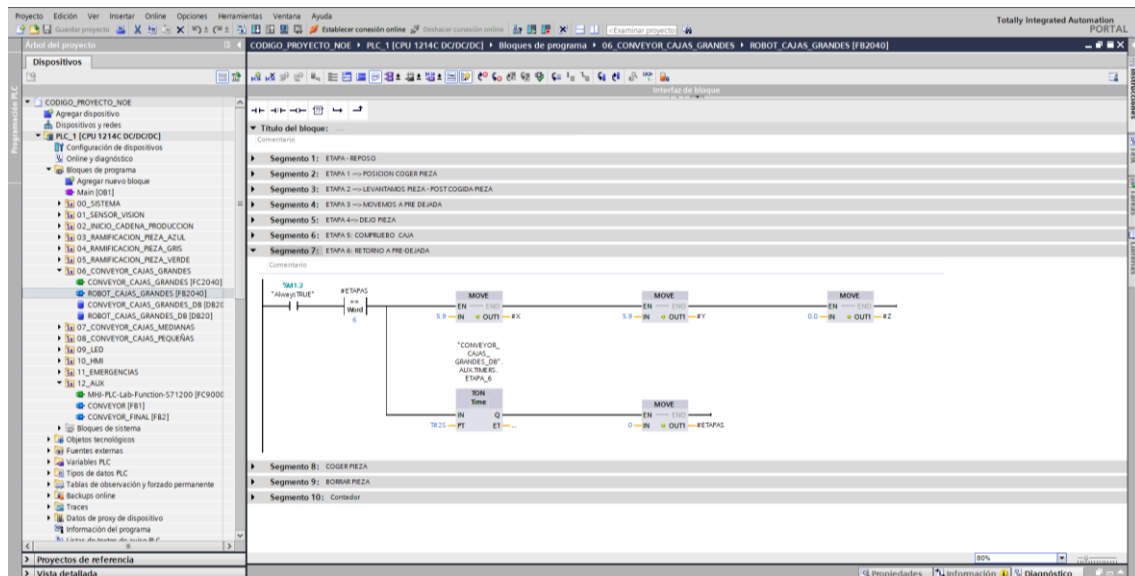


Ilustración 87: Cajas Grandes _FB (7)

Se establece en un segmento a parte las acciones que debe ejecutar el robot en cada etapa. Como la de coger pieza, que la llevara a cabo entre las etapas 1 y 4.

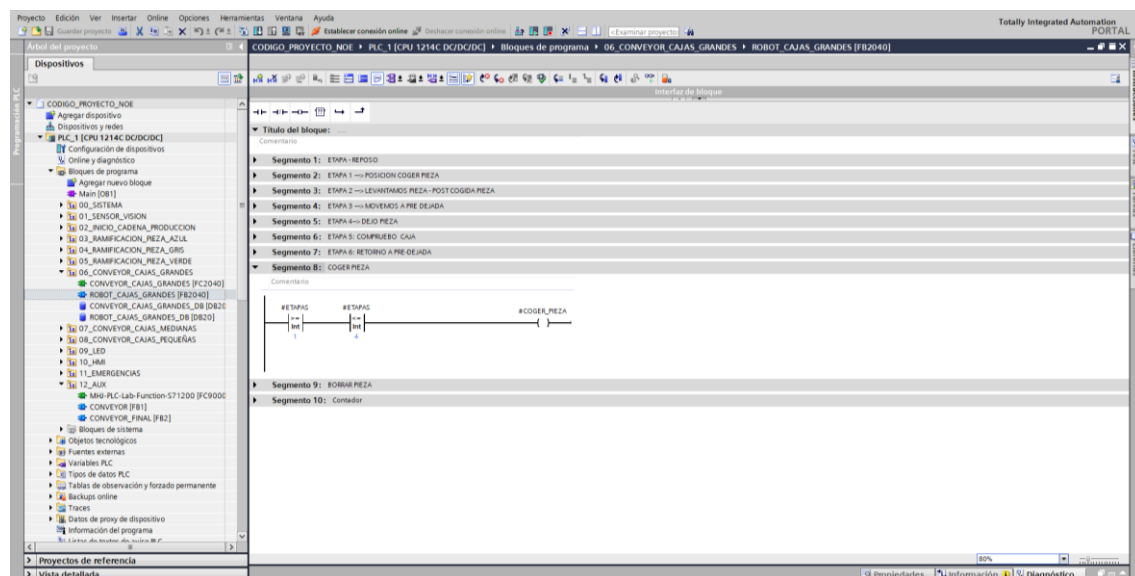


Ilustración 88: Cajas Grandes _FB (8)

Para que dé la impresión de que la pieza se mete en la caja, lo que se ha introducido es un elemento de salida que borra la pieza. Esto se ha tenido que realizar así debido a las limitaciones que presenta Factory I/O.

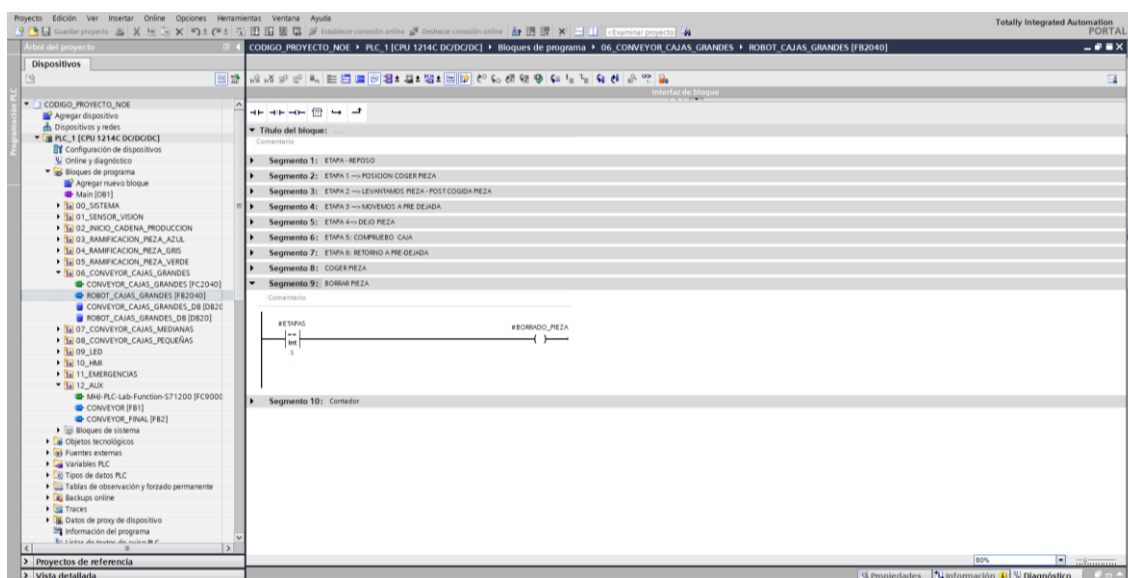


Ilustración 89: Cajas Grandes _FB (9)

Finalmente, para contabilizar las piezas que van siendo introducidas en la caja, se ha programado un contador.

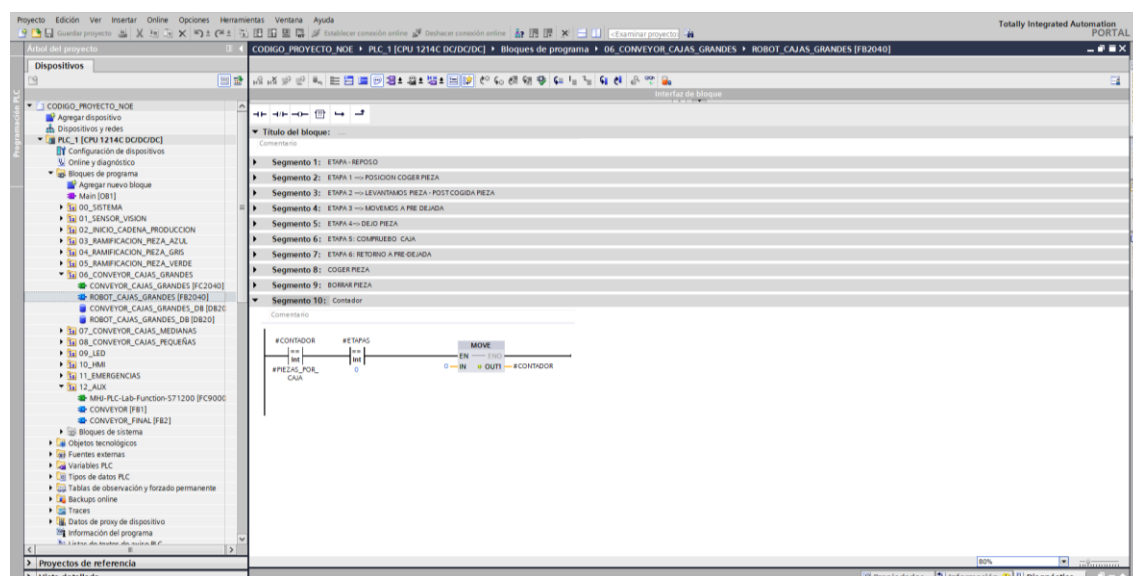


Ilustración 90: Cajas Grandes _FB (10)

5.4.6 Señalización

En esta carpeta, se programarán las señales luminosas del sistema, ya sean las señales de la baliza, así como las señales de los pulsadores.

El programa este compuesto por segmentos muy sencillos e intuitivos.

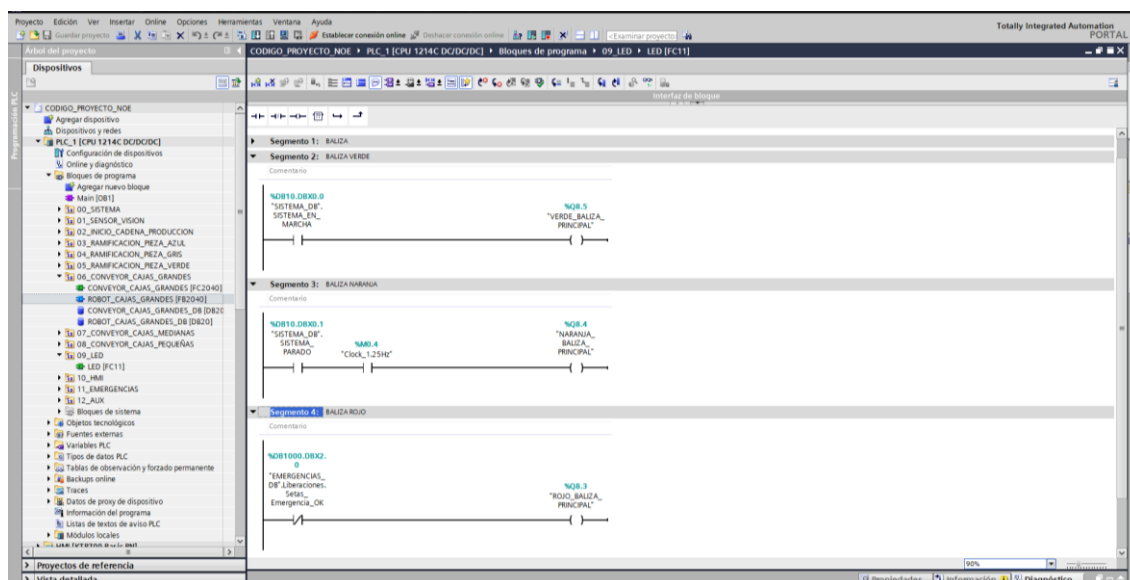


Ilustración 91: Segmentos de la Baliza

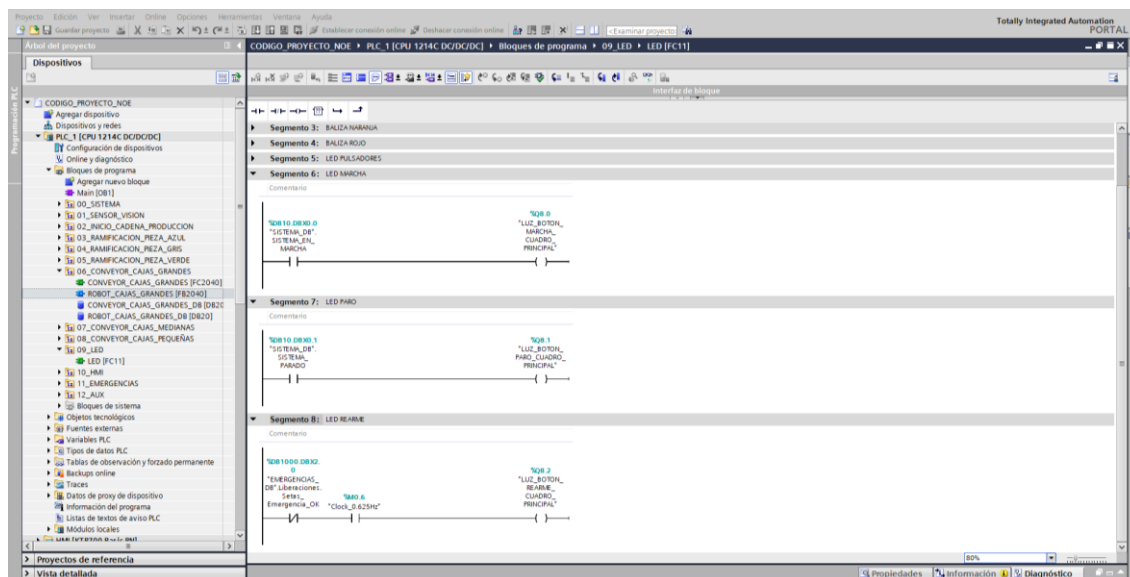


Ilustración 92: Segmentos de los Pulsadores

5.4.7: Supervisión pantalla HMI

En esta carpeta se ha programado básicamente todo lo relacionado con la pantalla HMI, tanto la muestra de fallos como las lecturas del sensor. Lo primero que se lleva a cabo, mediante dos Sting, que son dos cadenas de caracteres donde se indicara el fallo. Los fallos se irán mostrando en función de el orden en el que vayan entrando al sistema.

A continuación, se pasarán a mostrar los segmentos para esta programación.

Lo primero que se realiza es contemplar todos los fallos que tenemos en la instalación.

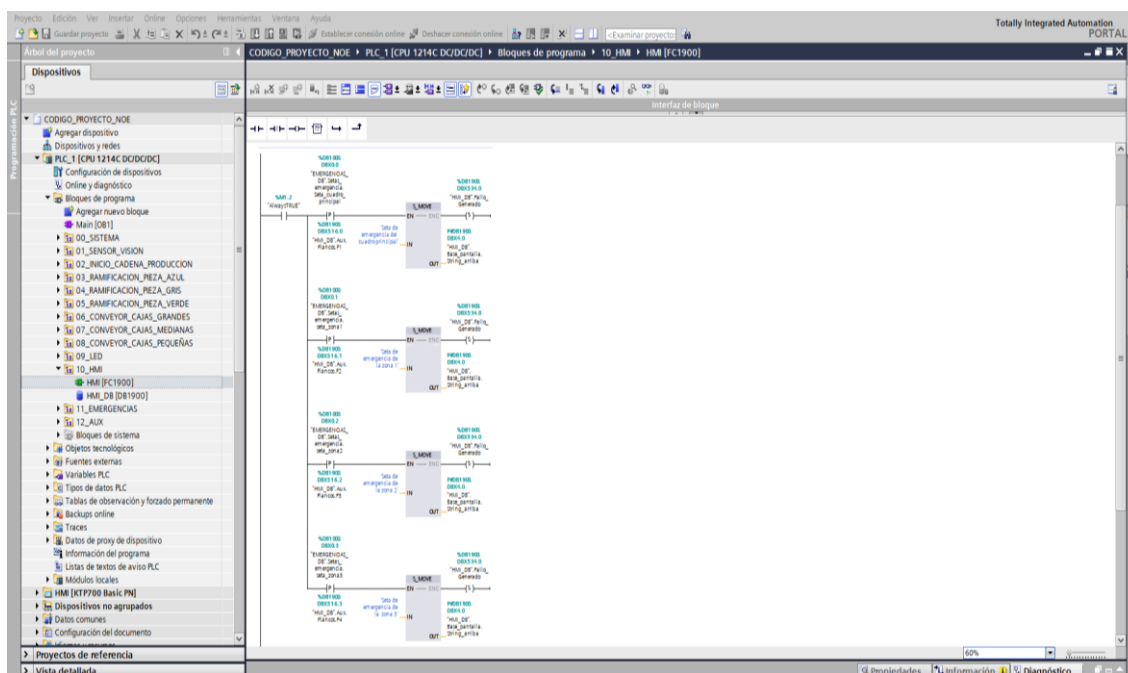


Ilustración 93: Detección de fallos HMI

A continuación, lo que se realiza es el traspaso del fallo de Sting arriba abajo.

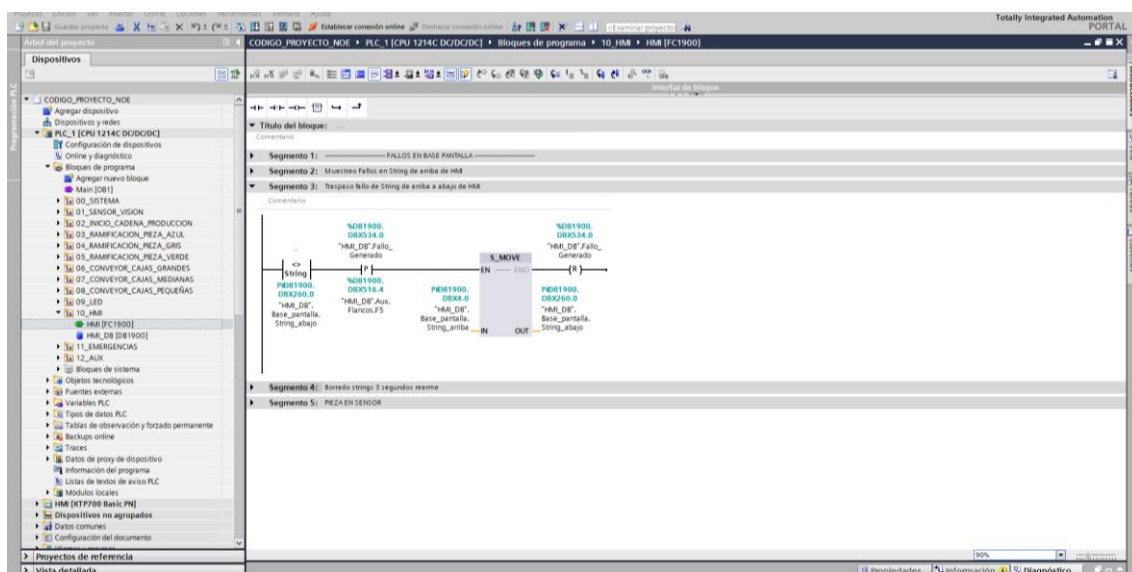


Ilustración 94: Traspaso de fallos HMI

Una vez se traspasa el fallo, y este ha sido solventado por el operario, se ha programado el borrado de estos pulsando durante tres segundos el botón de rearme.

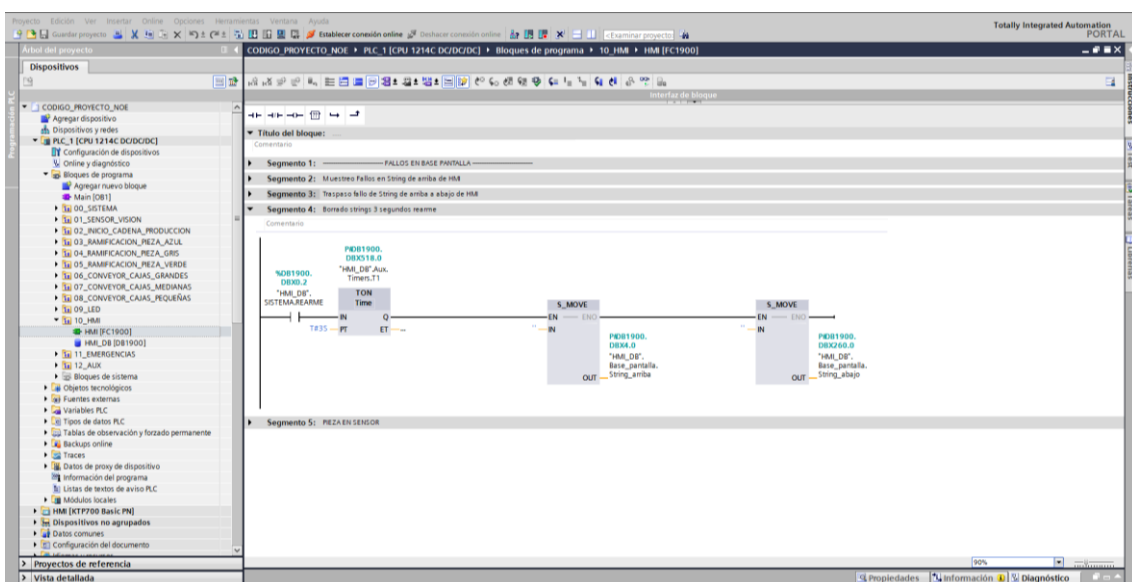


Ilustración 95: Borrado de fallos HMI

Finalmente, se ha programado también, que la pantalla HMI muestre la pieza que esta leyendo el sensor en ese instante de tiempo.

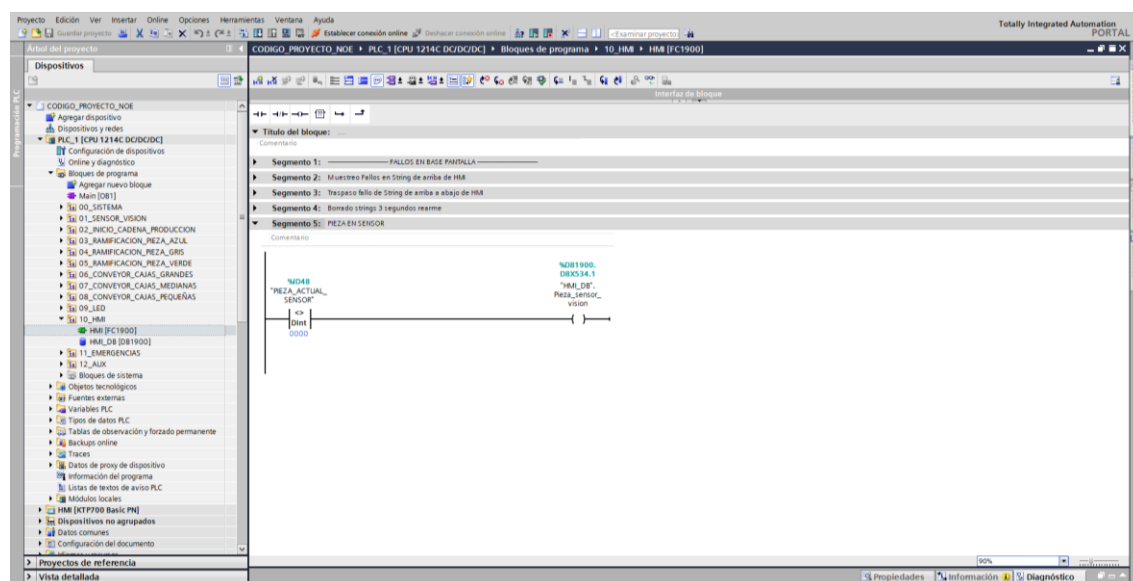


Ilustración 96: Lectura del sensor en el HMI

5.4.8 Emergencias

Este apartado es uno de los más importantes en el código, ya que se trata del tema de la seguridad.

Como se puede observar, sigue el mismo patrón que se explicó al inicio de este apartado. Esta constituido por un FC donde se encuentra programado todo el código para las setas de emergencia y emergencias que se puedan tener en la instalación, así como un DB para poder almacenar todas las memorias necesarias para la realización de este código. Si entramos en el FC, se pueden observar varios apartados, por eso se han dividido en dos secciones, setas de emergencia y liberaciones.

Con relación a las setas de emergencias se pueden definir varios segmentos:

-Rearme Emergencias, que mediante una entrada digital asociada a un botón del cuadro principal o bien mediante la pantalla HMI se puedan rearmar todas las emergencias del sistema, activando la memoria asociada a esta.

Esta memoria es la que se utilizará para poner a uno los bits de cualquier fallo u otras emergencias que haya en la línea. Cuando se de a rearme se debe volver a pulsar marcha para que el sistema vuelva a funcionar.

-Seta de Emergencia Cuadro Principal, para la programación de las setas de emergencia se ha utilizado un bloque Reset/Set, con prioridad al Set, para darle prioridad al fallo. El bit de seta de cuadro principal quedará a uno cuando perdamos el paro de emergencia del cuadro principal.

-Seta de Emergencia Zona 1, Zona 2 y Zona 3: Es el mismo sistema que para el de la Seta de Emergencia del Cuadro Principal. Estas zonas son seteadas por las setas de emergencia de la zona correspondiente a cada una de ellas. Todas pueden ser rearmadas desde el rearme de emergencia.

Finalmente, en lo referente a las liberaciones, será el relativo al código que proporciona la liberación para que una línea funcione. La liberación que se ha generado para que esta línea funcione es que no haya ningún fallo de seta, todos ellos, la serie de que no se tenga ningún fallo de seta genera un bit de liberaciones.

Cabe señalar, que en este DB aparecen las variables ordenadas en estructuras, esto es algo similar a las carpetas, dejando así las variables relacionadas con un tema,

unificadas en una misma estructura.

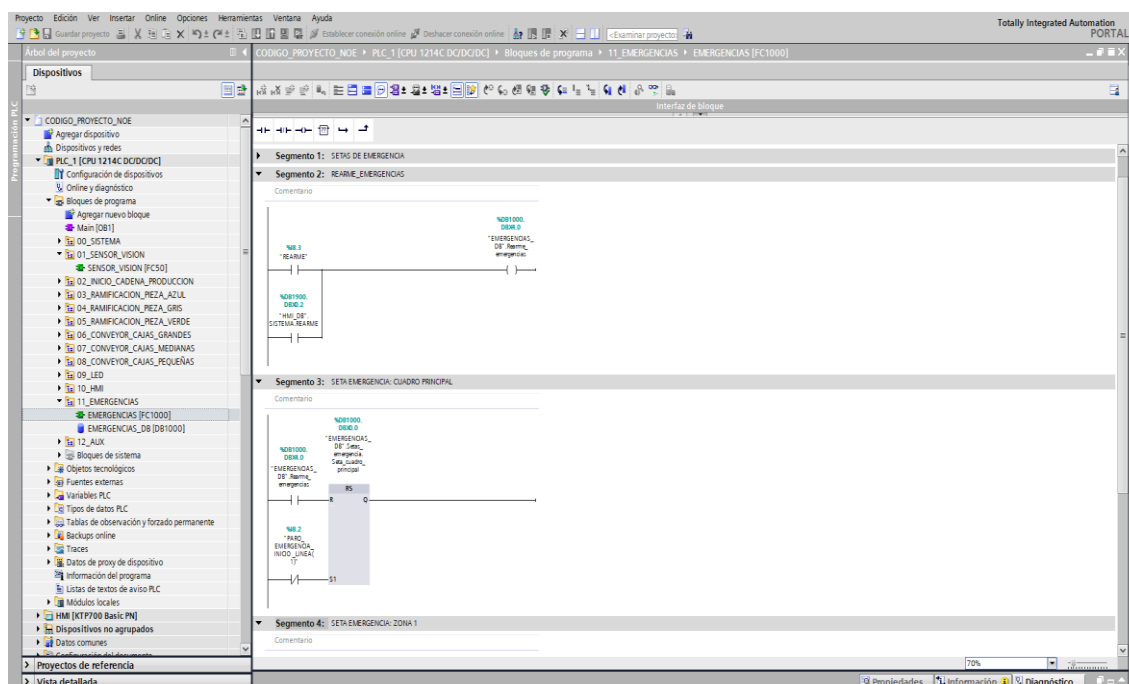


Ilustración 97: Emergencias (1)

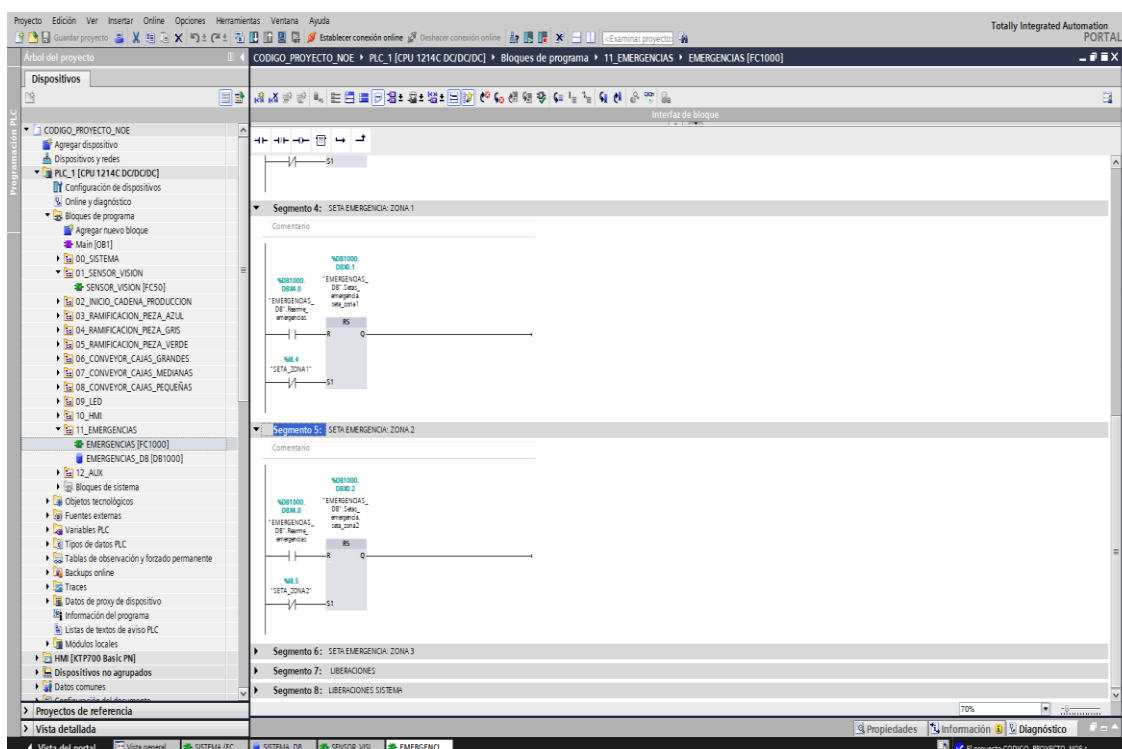


Ilustración 98: Emergencias (2)

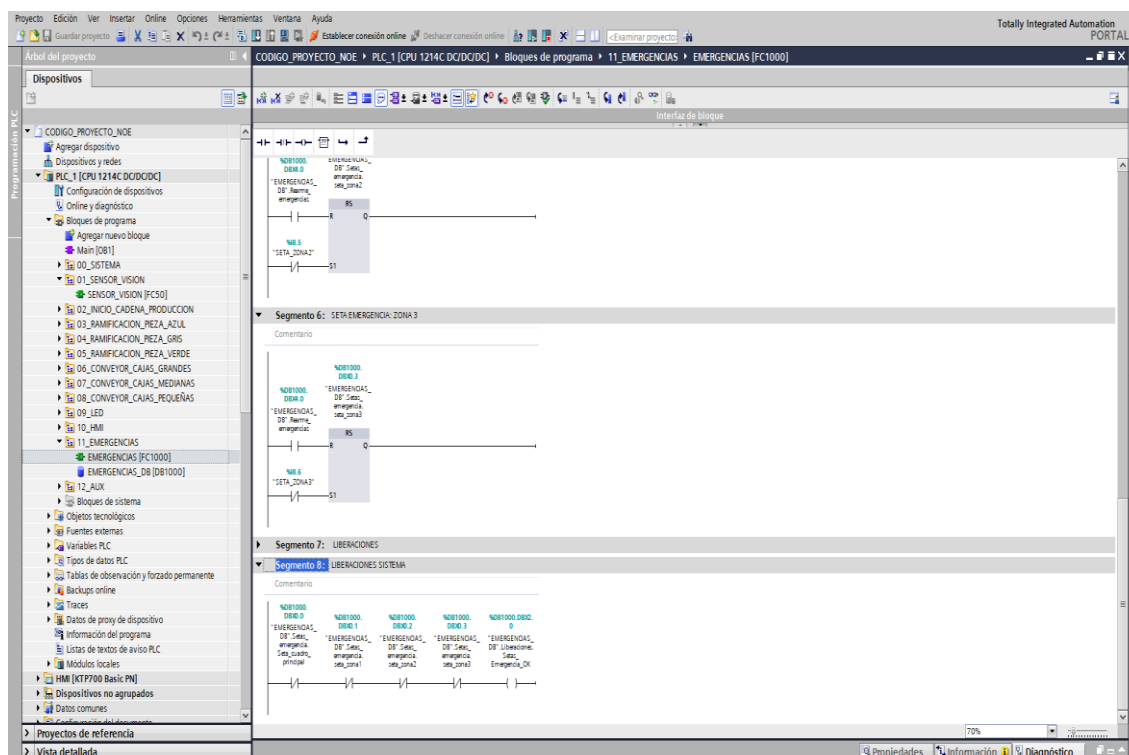


Ilustración 99: Emergencias (3)

5.4.9 Bloques auxiliares

Como ya se ha remarcado en líneas superiores, esta carpeta contiene un FC, que será el que use en la conexión de TIA Portal con Factory I/O, y dos FB, estos últimos usados para las cintas transportadoras de la línea.

Por cada cinta que se tenga en la línea, se realizará una llamada a este FB1. Y por cada cinta final, es decir, cada cinta que actúa de punto de unión entre la ramificación y el Robot, tendremos un FB2.

Se pasará a ver internamente cada uno de ellos.

En primer lugar, se mostrará la función empleada para la conexión de TIA Portal con Factory I/O, es una función que se descarga programada ya, es decir, no se tendrá que integrar ninguna línea más de código, y debe ser a la primera que se llame en el OB1.

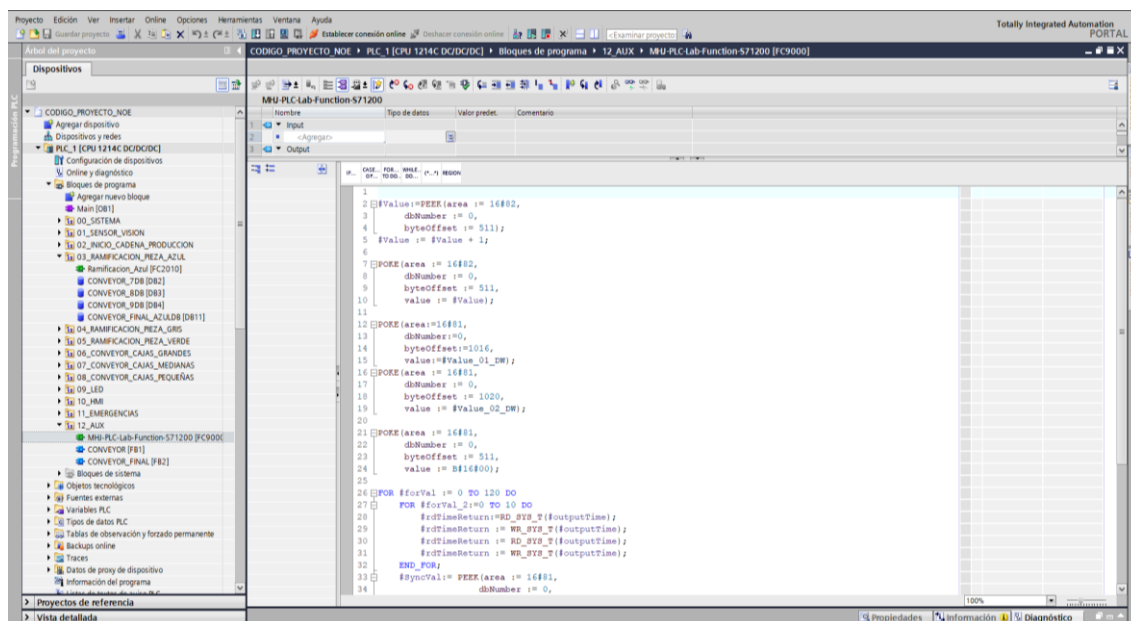


Ilustración 100: MHJ-PLC-Lab-Function-S71200

Una vez se ha explicado el FC. Se pasará a explicar los dos FB empleados para las cintas.

FB1: Lo primero que debemos saber, es que en el interfaz de bloque tiene una serie de valores (inputs, outputs, static...), que son con los que se programará en los segmentos y a los que asociaremos diferentes entradas o salidas en las llamadas de las diferentes cintas.

Así mismo, cabe destacar que este FB1 consta con un DB propio llamado DB de instancia donde se podrán almacenar memorias.

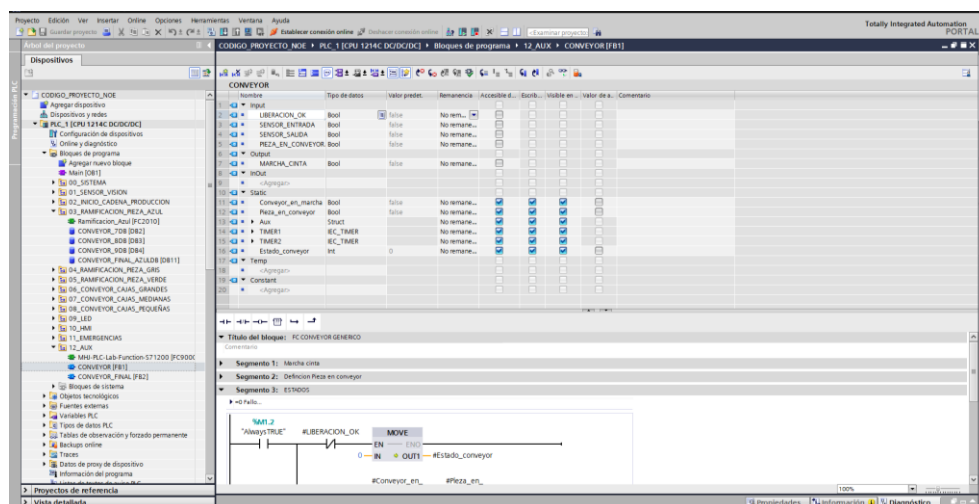


Ilustración 101: Interfaz de bloque FB1

En lo referente a la programación de este FB1 se pueden distinguir tres segmentos base.

El primer segmento, marcha cinta, se podrá ver que para darle a la salida de la marcha cinta, será necesario tener las liberaciones en estado OK y cumplir una serie de condiciones en la cinta.

Se ha programado en paralelo con un temporizador para un pequeño empujón a la pieza antes de que la siguiente cinta sea consciente de que le esta entrando la pieza. Esto se realiza por problemas con la conexión de Factory I/O, a veces las piezas se quedaban ligeramente atascadas entre cinta y cinta.

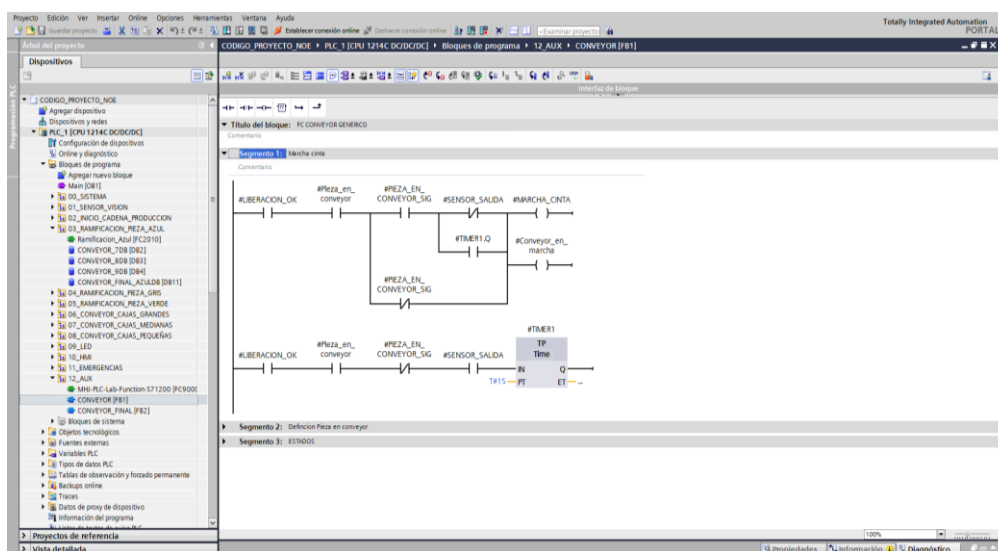


Ilustración 102: Conveyor FB1(1)

En el segmento dos, es el encargado de definir cuando ha entrado pieza y cuando la ha perdido. En este caso con el flanco positivo del sensor de entrada de la cinta, pondrá a uno el bit de la pieza en esta, y con el flanco negativo del sensor de salida se pondrá a cero el bit de la pieza en cinta.

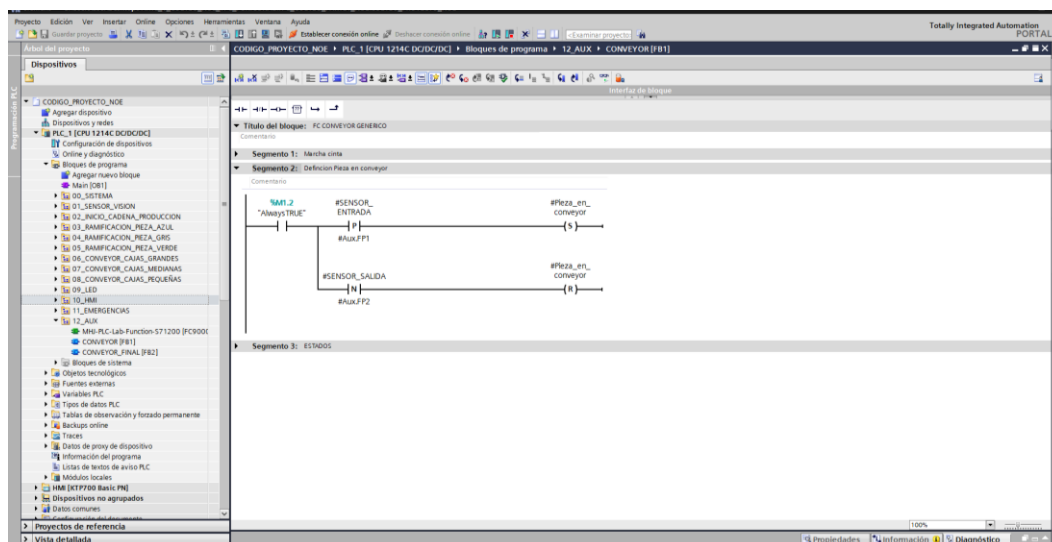


Ilustración 103: Conveyor FB1(2)

Finalmente, en el segmento tres, que es el referente a los estados, ocurre lo mismo que en el caso de la cinta emisora y derivadora explicados con anterioridad.

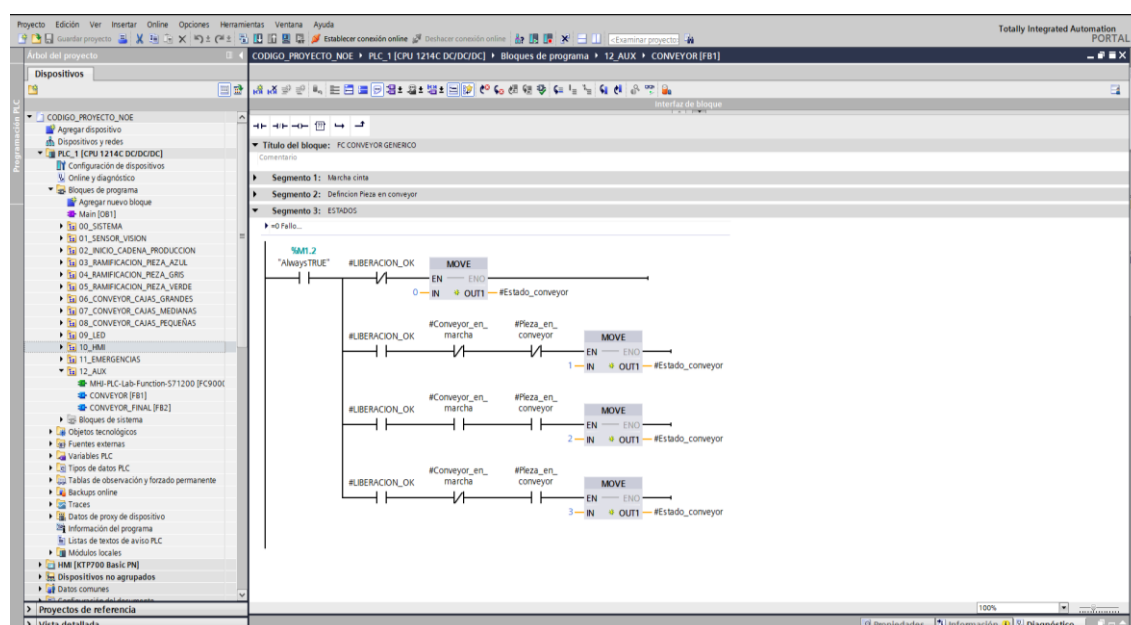


Ilustración 104: Conveyor FB1(3)

A continuación, se pasará a explicar la programación del FB2, que es el utilizado en las cintas finales. Está compuesto por cuatro segmentos, que pasaremos a explicar de forma detallada.

En primer lugar, al igual que en el FB1, este FB2 también tiene unas variables predefinidas en la interfaz de bloque que se muestran a continuación. Son las que se emplearán para la programación de este FB.

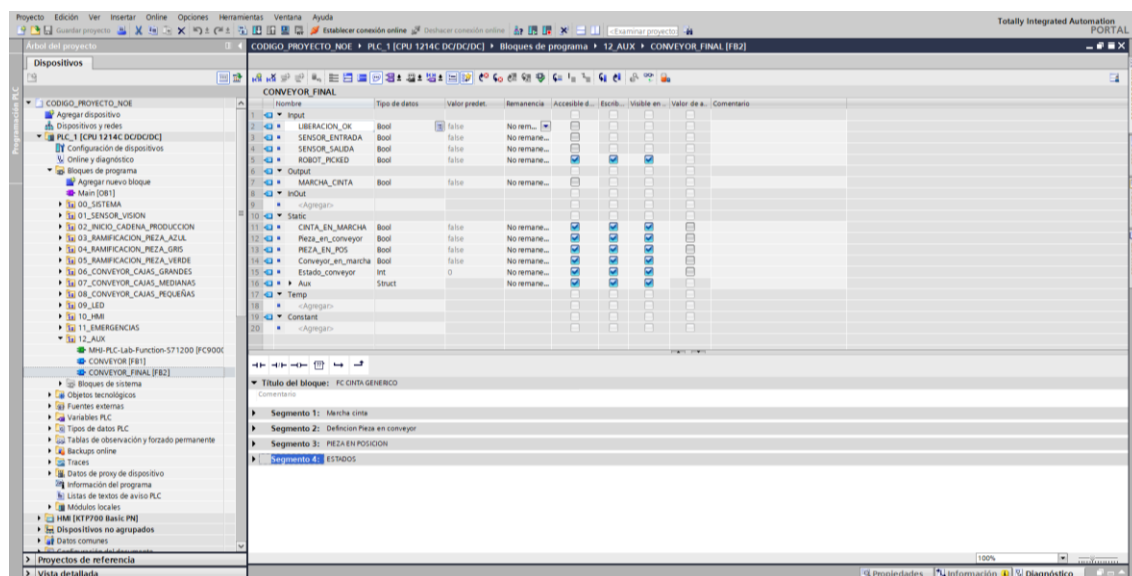


Ilustración 105: Interfaz de bloque FB2

El segmento uno, marcha cinta, es el encargado de poner en marcha esta cinta final cumpliendo una serie de condiciones.

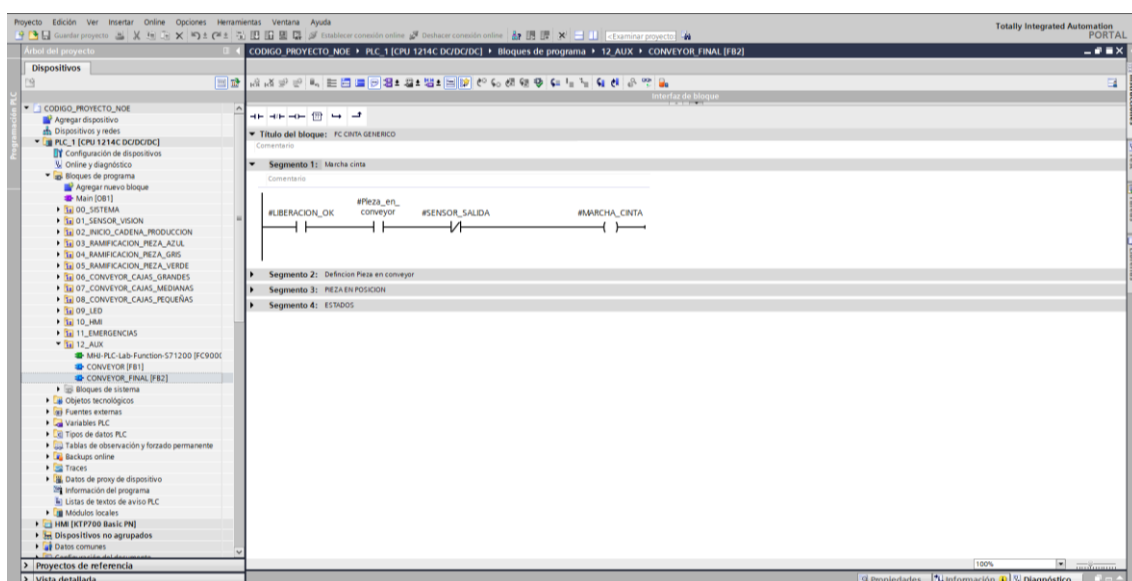


Ilustración 106: Conveyor FB2 (1)

En lo referente al segmento dos, siendo este es el encargado de definir si la pieza está en la cinta, pondrá a uno el bit de la pieza en cinta cuando se tenga un flanco positivo en el sensor de entrada. Para poner a cero el bit de la pieza en cinta es necesario un flanco positivo de que el robot está para coger la pieza, y un segundo después se pondrá a cero este bit de pieza en esta.

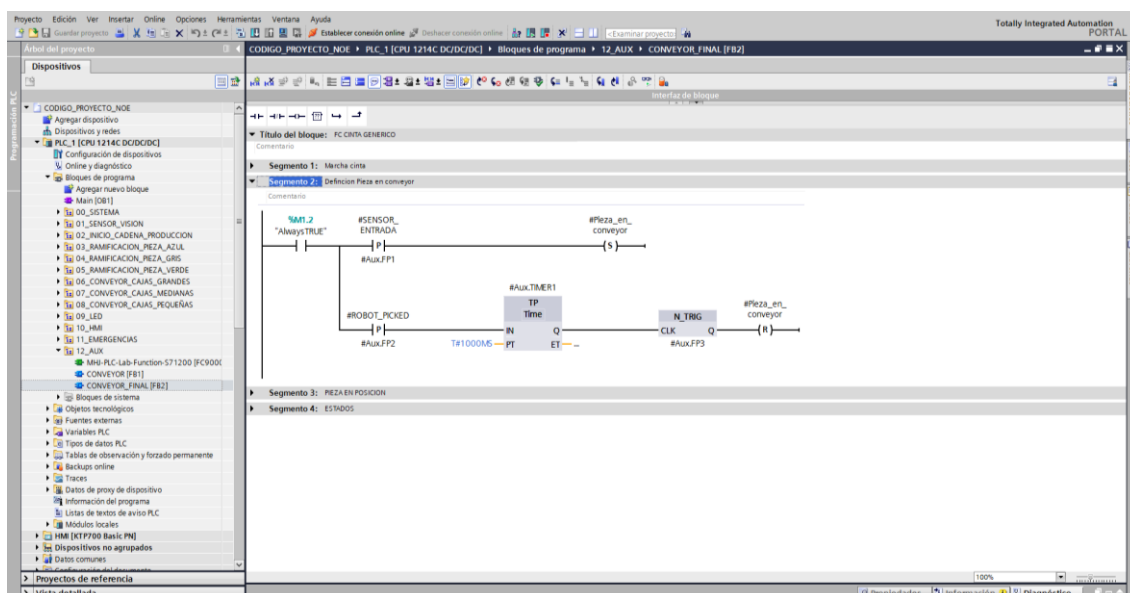


Ilustración 107: Conveyor FB2 (2)

En el segmento de pieza en posición, segmento tres, se definirá mediante el sensor de salida.

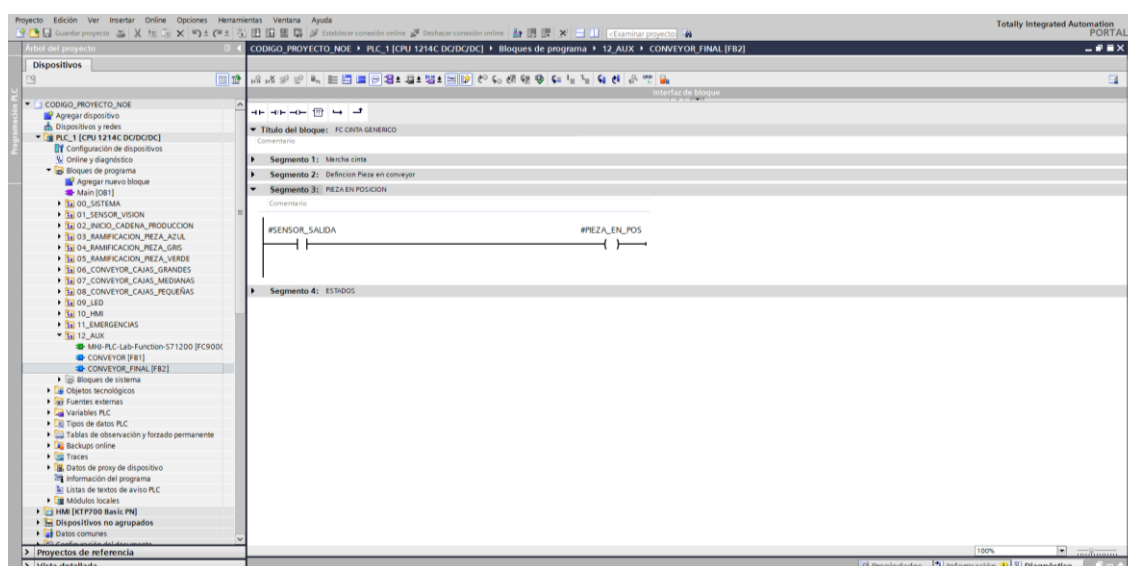


Ilustración 108: Conveyor FB2 (3)

Para finalizar, el ultimo segmento, es el ya explicado segmento de estados. Es el encargado de determinar cómo se encuentra la cinta en cada momento.

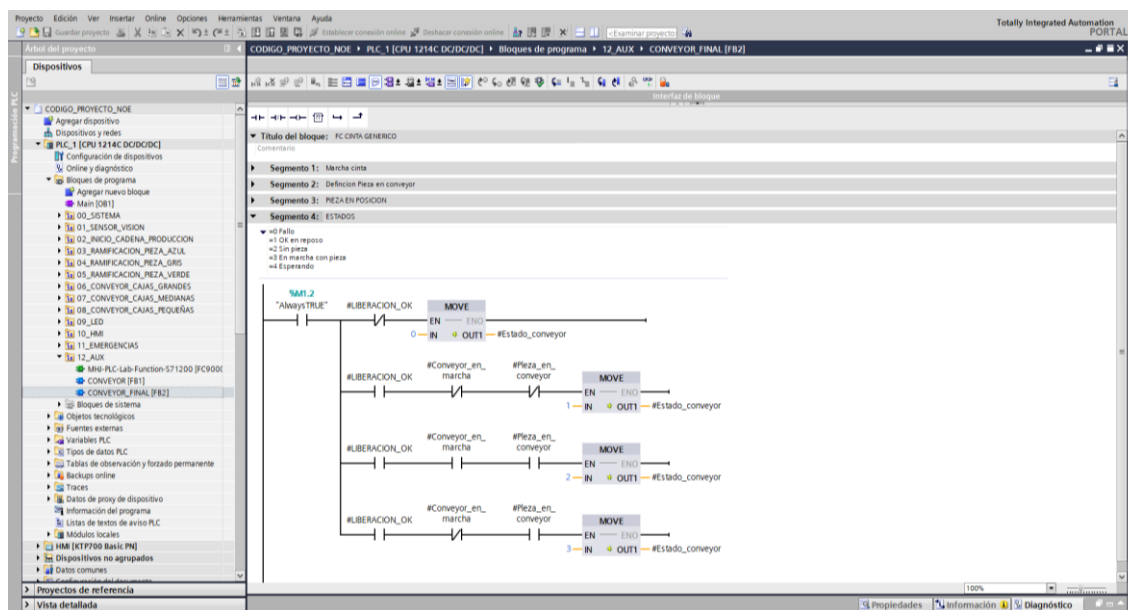


Ilustración 109: Conveyor FB2 (4)

6. SISTEMA DE SUPERVISIÓN

6.1 PANTALLA HMI

Una pantalla HMI (Human Machine Interface) es un interfaz hombre – máquina que permite a este comunicarse con una maquina software o sistema.

Este sistema es el empleado por los operadores y supervisores de las maquinas, con el objetivo de controlar y monitorear los procesos industriales.

En este proyecto se ha introducido una pantalla HMI para supervisar tanto el correcto funcionamiento del sistema y cada una de sus ramificaciones, como para la detección rápida de posibles errores en la línea.

Del mismo modo que se ha realizado con el controlador, se añade un dispositivo HMI que será el KTP700 Basic PN con referencia 6AV2 123-2GB03-0AX0.

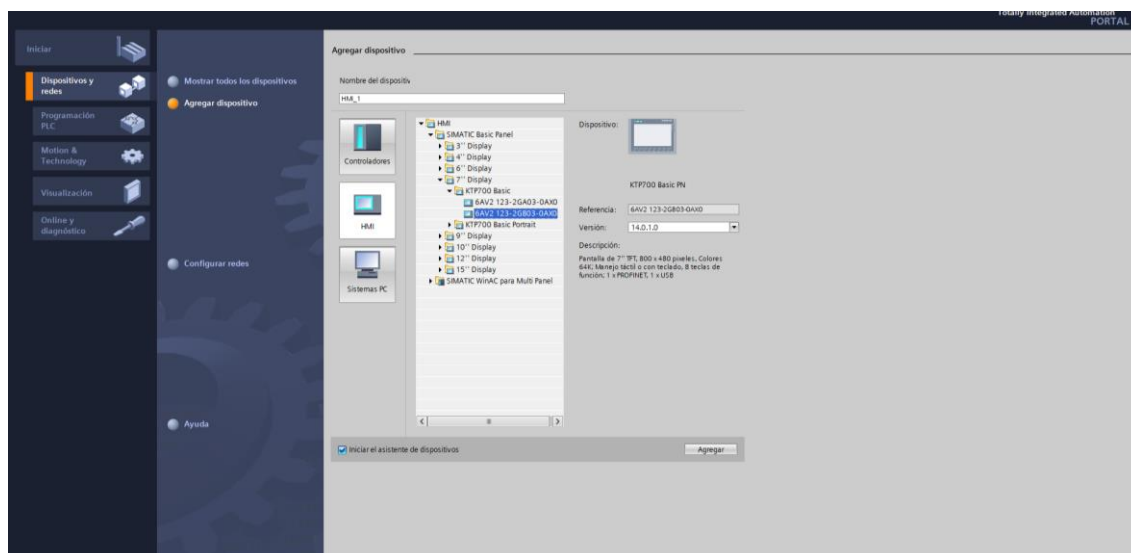


Ilustración 110: Dispositivo KTP700 Basic PN

Mediante el asistente de dispositivos se configurará el panel de operador. Seleccionando el PLC y el driver empleado para la comunicación con este.

La conexión con el PLC se realizará mediante PN/IE de Siemens (RJ45 dentro del cuadro).

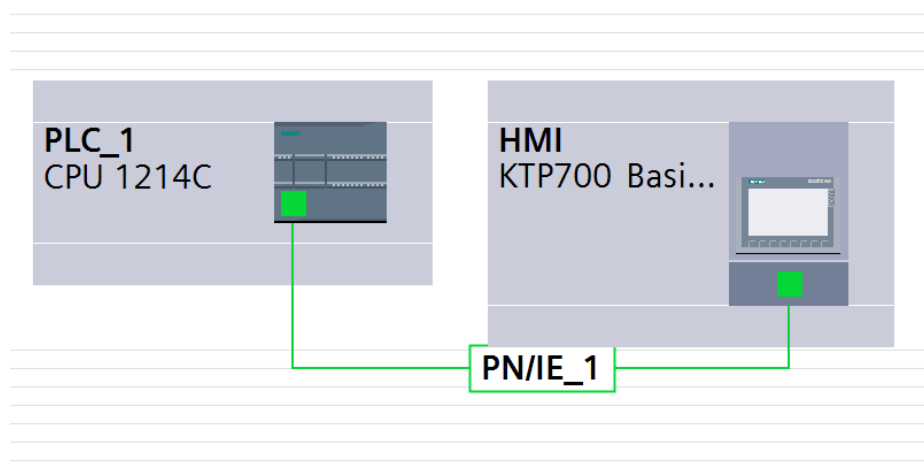


Ilustración 111: Conexión del PLC con el HMI

A través de las imágenes que se diseñan, el usuario interactuara con el PLC programado pudiendo de este modo obtener más información de forma rápida y visual. Para ello se ha

diseñado un conjunto de 4 pantallas y una pantalla común para todas haciendo así un total de 5 pantallas.

Las imágenes desarrolladas son:

1. Pantalla de inicio: En esta pantalla se muestran los controles principales para el arranque, paro o rearme del sistema, así como las diferentes partes que componen la línea.
2. Pantalla de Inicio de Línea: Nos muestra la zona emisora de producto, así como la zona derivadora.
3. Pantalla Rama Gris: En esta pantalla podemos observar la zona derivadora, donde se dirigen las piezas grises, así como su pick and place.
4. Pantalla Rama Verde: De igual modo que ocurre en la gris, podemos ver el proceso que sigue la pieza verde hasta llegar a su pick and place, así como el estado en el que se encuentran las cintas transportadoras en cada instante.
5. Pantalla Rama Azul: En esta pantalla observamos lo mismo que en la zona verde o gris, pero para la pieza azul. Es el proceso que sigue la pieza desde que es derivada por el cilindro hasta llegar a su pick and place correspondiente.

6.2 PLANTILLAS

Una vez se han establecido las conexiones con el PLC, se diseñan las plantillas empleadas en el HMI. En este caso habrá dos tipos de plantillas.

La primera plantilla es para la pantalla de inicio principal, exclusiva para la misma, sin embargo, la segunda plantilla se usará tanto en la pantalla de inicio de línea como en las pantallas de las tres ramificaciones.

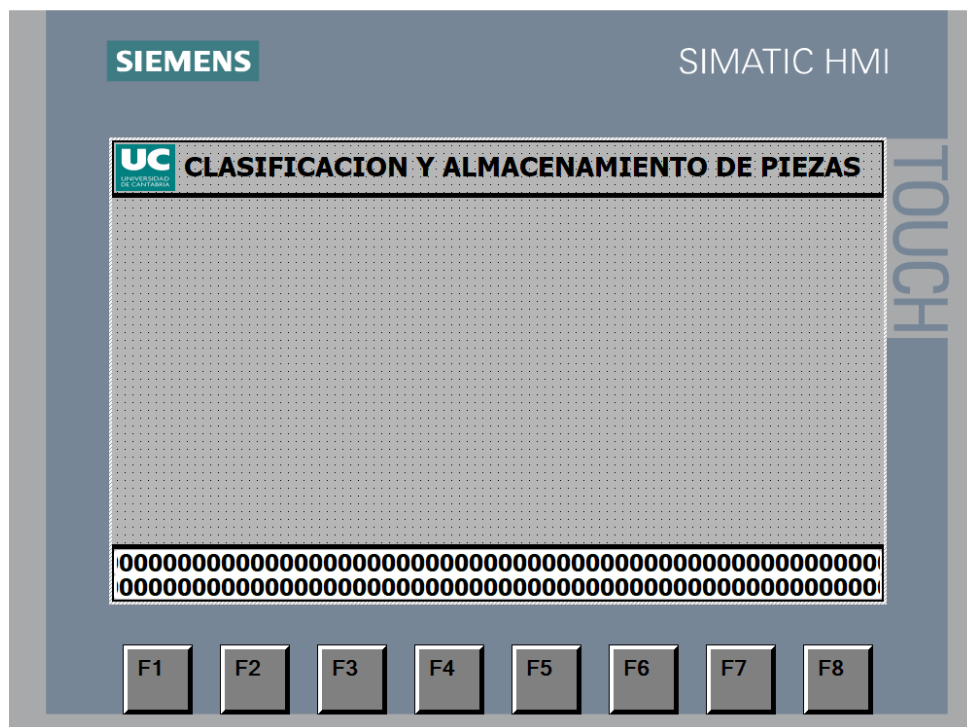


Ilustración 112: Plantilla 1

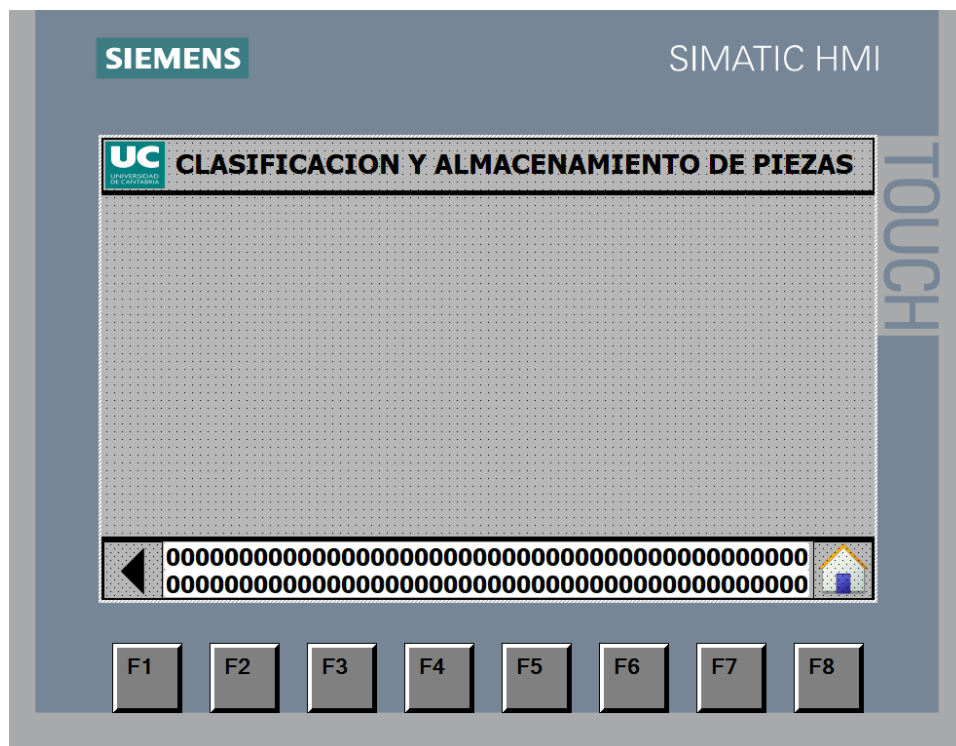


Ilustración113: Plantilla 2

La principal diferencia entre la plantilla 1 y la plantilla 2, es la incorporación a esta última de dos botones a la parte inferior derecha e izquierda de esta. Estos son empleados para regresar a la pantalla inicial (botón derecho inferior) o regresar a la pantalla anterior (botón izquierdo inferior).

Así mismo, en la parte inferior de ambas pantallas aparecen dos cadenas de caracteres que servirán para indicar al operario el error que se ha producido, así como la zona donde está el mismo.

6.3 PANTALLA DE INICIO

La pantalla principal será lo que el usuario vea nada más entre al sistema. Es, por lo tanto, por lo que es imprescindible que tenga información importante como los fallos del sistema, los indicadores del funcionamiento de la línea, el rearme o las diferentes partes de esta.

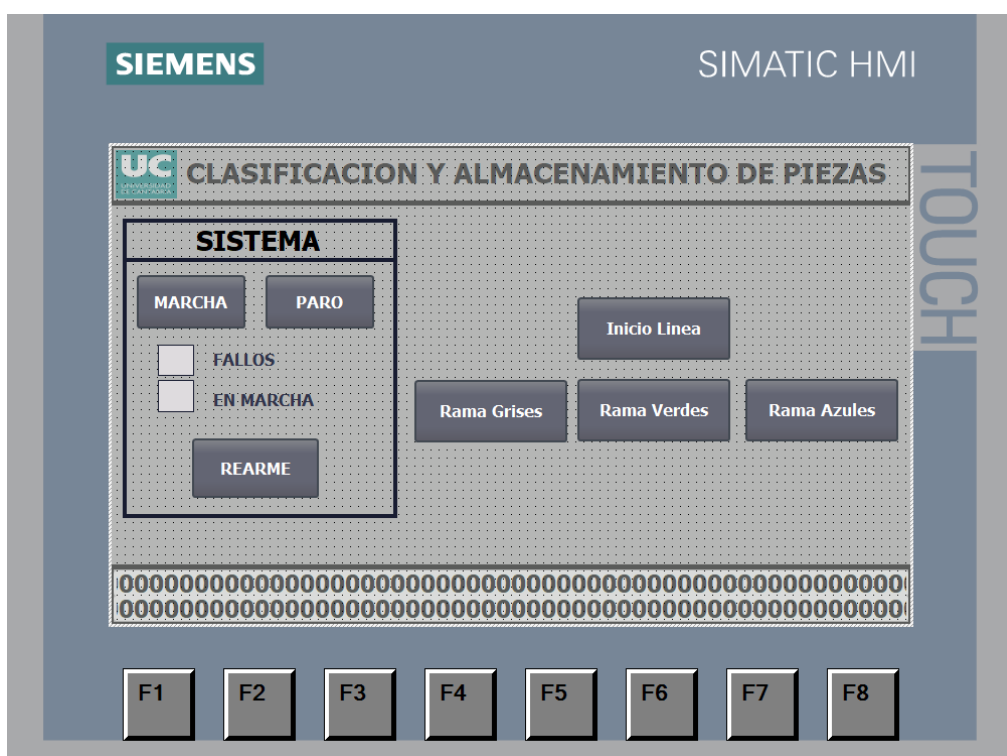


Ilustración 114: Pantalla de inicio

Como se puede observar, en la parte izquierda de esta pantalla se encontrarán los botones principales del control del sistema junto a unos leds indicadores. Estos indican si el sistema está en marcha o si por el contrario se ha producido un error y se tendrá un fallo en la línea.

En la parte derecha de la misma, se encuentran situados los botones que derivaran a cada una de las partes del sistema, inicio de línea, rama gris, rama verde y rama azul, se han dividido en 4 pantallas distintas para así obtener más información visual y un control más detallado de estas partes.

6.4 PANTALLA INICIO DE LÍNEA

El primer botón que se encuentra en la parte derecha de la pantalla inicial es el de inicio línea. En esta pantalla el usuario podrá observar tanto la zona de emisor de pieza, la lectura del sensor de visión y posterior clasificación de la pieza.

Del mismo modo, se marcarán los estados en los que se van encontrando las cintas, como está el sensor de visión o los dos cilindros neumáticos. De forma auxiliar se han introducido unas leyendas para servir de guía al operario, marcando estas los detalles de las principales zonas en esta etapa.

El operario podrá de forma adicional observar la transición de las piezas en función de su color.

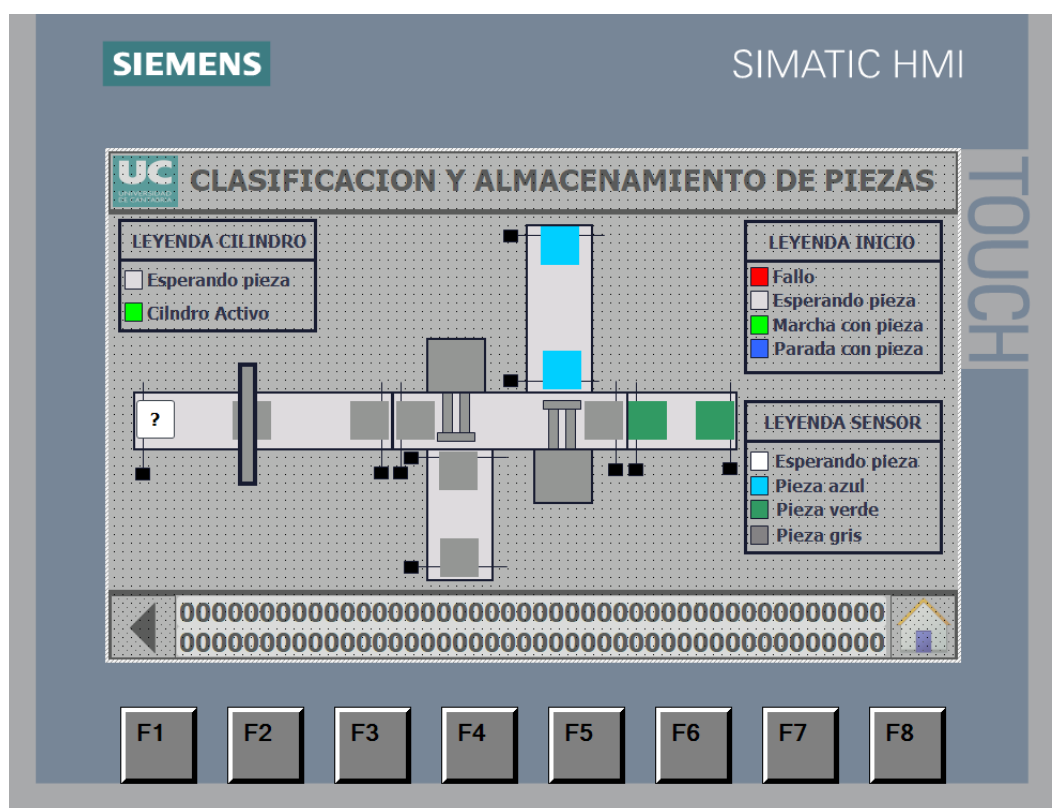


Ilustración 115: Pantalla de inicio Línea

6.5 SUPERVISIÓN RAMIFICACIONES

6.5.1 Rama Gris

En esta pantalla se muestra la zona de derivación para las piezas grises.

Una vez el sensor detecta que la pieza es gris, el cilindro neumático 1 de la pantalla de inicio de línea se activa y desplaza la pieza hacia la zona gris. De igual modo que en el inicio de línea, en esta pantalla se puede observar la transición de posiciones que sigue la pieza y el estado de las cintas transportadoras en cada instante.

Al final de las cintas transportadoras se encuentra la zona de pick and place, donde se podrá observar las diferentes etapas que realiza este para la introducción de la pieza en la caja, así como la zona de cajas pequeñas y el estado de sus cintas.

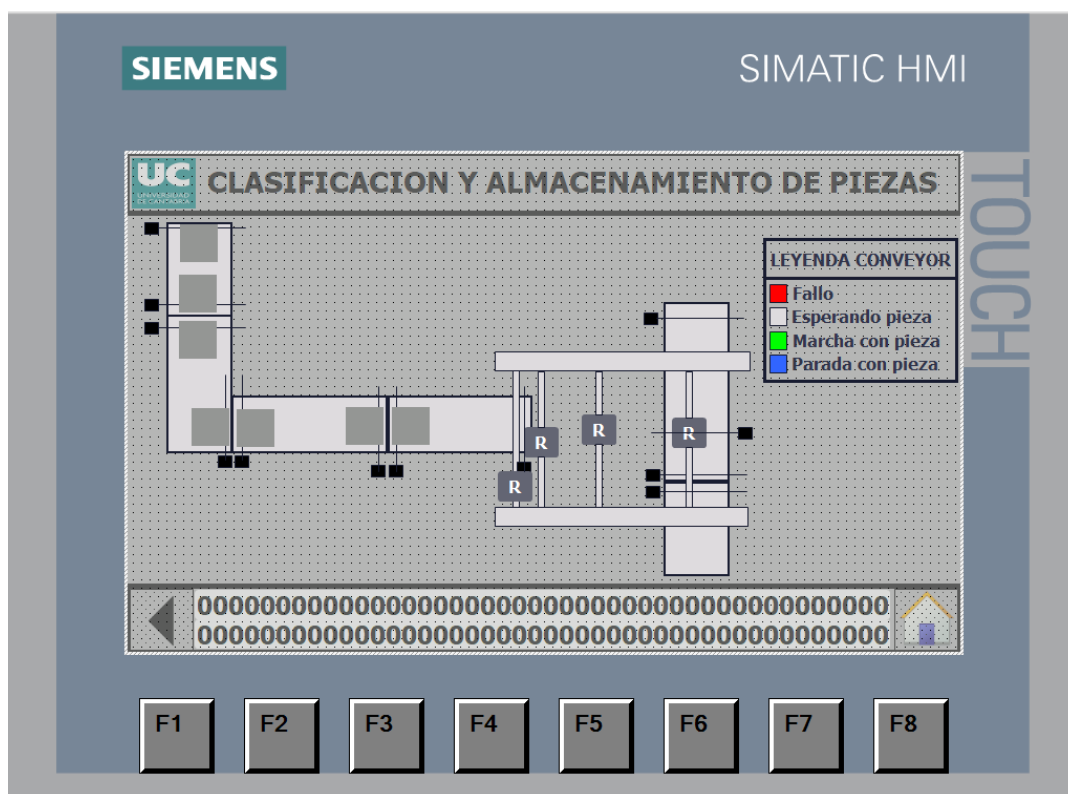


Ilustración 116: Pantalla Rama Gris

6.5.2 Rama verde

Para esta pantalla, se ha seguido el mismo ejemplo que en la rama gris. Se puede observar como la pieza verde una vez sale de la zona derivadora (pantalla de inicio de línea), y no activa ningún cilindro neumático, la pieza sigue su curso por las cintas transportadoras cambiando estas de estado según la presencia de la pieza llegando así al pick and place, donde se podrán seguir de forma muy sencilla los diferentes movimientos realizados en las etapas para la cogida y posterior dejada de la pieza en las cajas medianas.

Estas pantallas son muy útiles ya que facilitan al operario la supervisión de la pieza hasta la llegada al pick and place, así como si este realiza de forma correcta los movimientos.

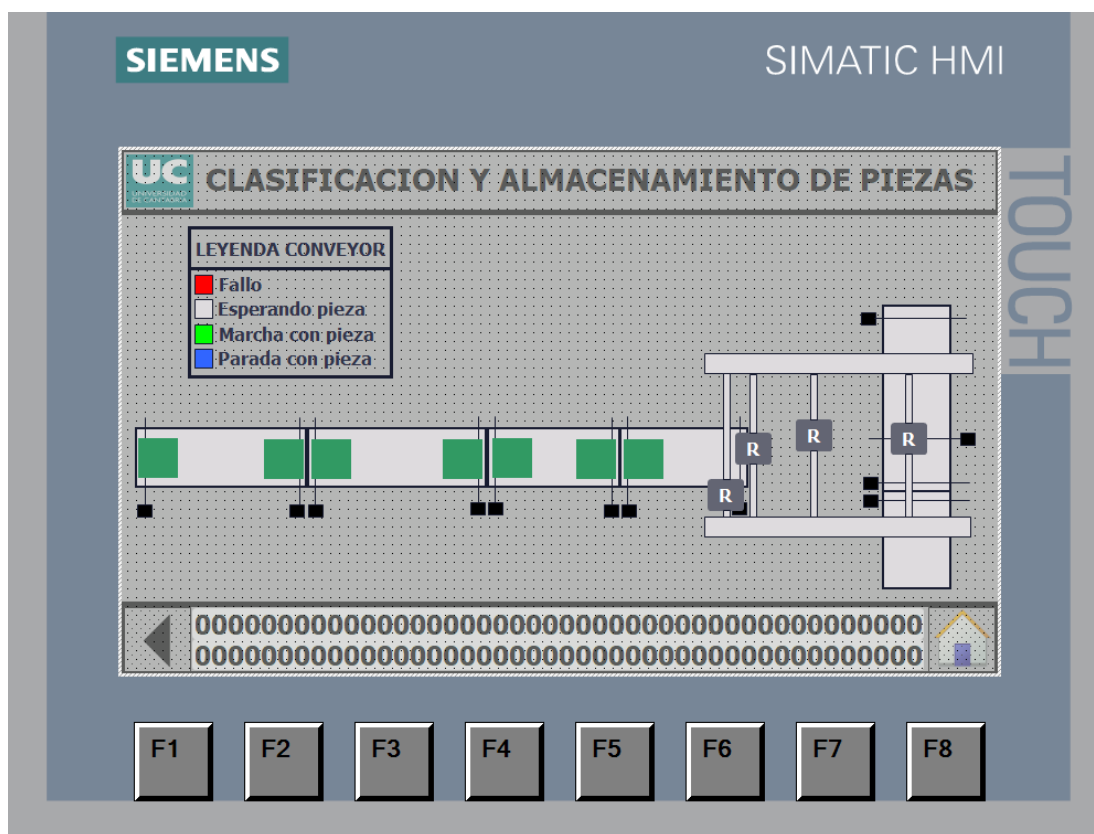


Ilustración 117: Pantalla Rama Verde

6.5.3 Rama azul

Finalmente, para la pantalla de rama azul, se ha seguido el mismo patrón que en las dos anteriores.

Se observa como en la zona derivadora se activa el cilindro si es detectada la pieza y pasa a esta ramificación, donde será transportada por las cintas, indicando estas sus estados en cada instante y pudiendo ver la transición de la pieza, hasta llegar al pick and place donde será introducida en la caja grande situada en su correcta posición.

Como ya se ha indicado, se ha elegido este diseño de pantalla ya que para la supervisión y detección de fallos en el sistema es la forma más simplificada e intuitiva para el usuario.

Cabe destacar que en la zona de cajas también se indica cuando la caja esta completa y sale de la cinta. Estas funcionan del mismo modo que las cintas transportadoras de producto y constan de los mismos estados que estas con la diferencia de que serán las encargadas de transportar las cajas con las piezas.

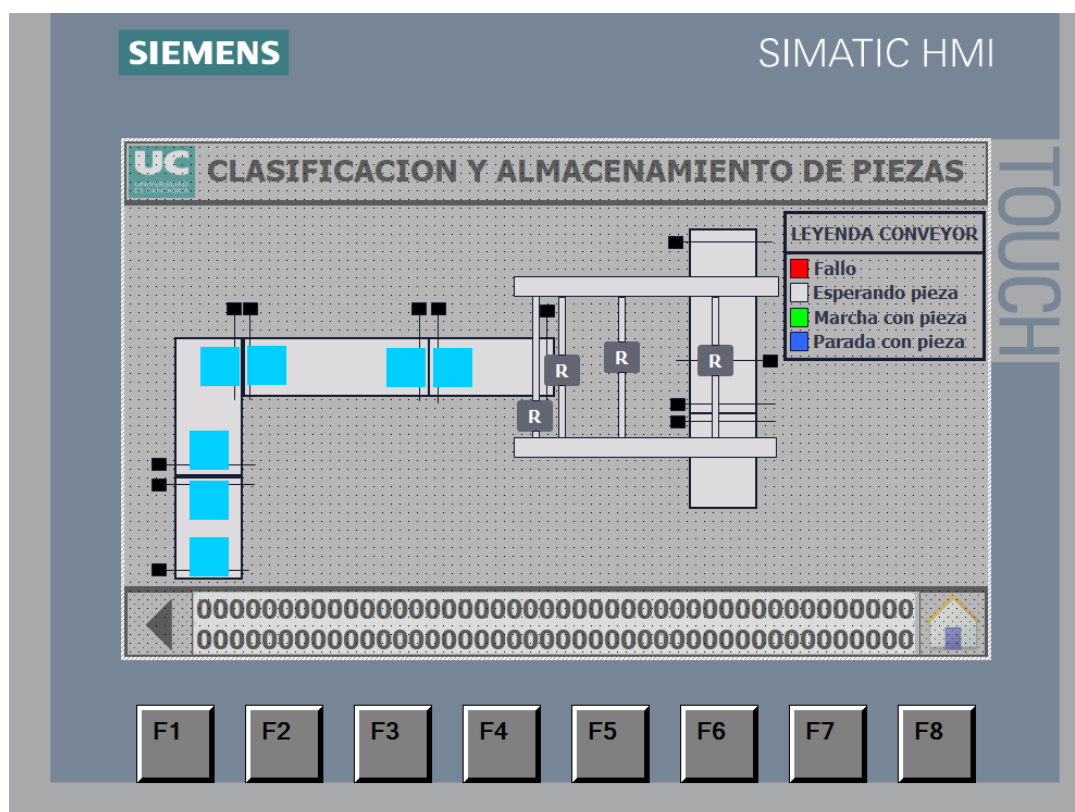


Ilustración 118: Pantalla Rama Azul

7. CÉLULA DE PALETIZADO

Las células robotizadas de paletizado son sistemas robóticos que ayudan a que este proceso se haga de manera más rápida y productiva.[12]

Una célula robotizada o robot paletizador permite automatizar este proceso, de manera que sea más rápido y los ciclos de fabricación del producto sean más cortos.

Este robot paletizador cuenta con un brazo robótico que permite operar con diferentes piezas, así son capaces de:

- Coger una pieza y colocarla de manera automática en ubicaciones programadas.[12]
- Hacer de forma más rápida lo que harían los operarios ya que es un proceso mecánico y repetitivo.[12]
- Liberar de trabajo a los operarios de la fabrica para poder realizar otro tipo de tareas.[12]
- Estos robots de paletización también pueden trabajar a diferentes temperaturas extremas, por lo que son mucho mas adecuados que el uso de otro tipo de maquinaria o procedimiento manual.[12]

Algunas de las ventajas del uso de células de paletizado son las siguientes:

- Mayor productividad.[12]
- Reducción del ciclo de paletizado de productos. [12]
- Ayudan a reducir las bajas laborales.[12]
- Pueden trabajar de forma ininterrumpida.[12]
- Se integran fácilmente en las cadenas de producción.[12]
- Ayudan a aumentar tu competitividad.[12]

7.1 ELECCIÓN DEL ROBOT

Del mismo modo que se ha elegido un PLC determinado, que cumple unas condiciones, para su programación en apartados anteriores, es de suma importancia elegir un Robot adecuado, tanto a nivel de un correcto funcionamiento del sistema, como para que satisfaga las necesidades a desarrollar en el mismo.

Para este proyecto se ha elegido un Robot de las librerías internas de ABB. En este caso se ha elegido el robot IRB-1600, es un robot antropomórfico de 6 ejes.

Un robot antropomórfico de 6 ejes se asemeja al ser humano, principalmente a la parte del hombro, brazo y muñeca. Nos permite realizar movimientos más complejos para la carga y descarga de cajas, ya sea por la complejidad de la caja o la colocación de estas en el pallet.[13]

Este tipo de robots es actualmente el más común en la industrial, debido a su versatilidad en gran cantidad de aplicaciones distintas. [13]

El robot de 6 ejes o 6 grados de libertad puede moverse a una posición en las coordenadas cartesianas y hacerlo con diferentes orientaciones.



Ilustración 119: IRB-1600 de ABB

Algunas de las características que presenta el IRB 1600 de ABB son las siguientes:

- Es el robot de 10kg de mayor rendimiento.[14]
- Los ciclos del robot son más cortos, a veces siendo la mitad que con otros robots y permite incrementar la producción. Esta constituido por ciclos un 50% más rápidos que otros competidores.[14]
- El diseño rígido y pesado, combinado con engranajes rectos, hacen que el robot sea extremadamente robusto. El software inteligente de detección de colisiones aumenta aún más la excelente confiabilidad del robot.[14]
- A alta velocidad, la mayoría de los robots toman atajos, con el IRB 1600 el camino será el mismo independientemente de la velocidad, esto se produce gracias a una única combinación de cerebro y fuerza del robot.[14]
- Lo mismo ocurre en términos de calidad, a alta velocidad, la mayoría de los robots ofrecen menos calidad. Con el IRB 1600, la trayectoria será la misma sin importar la velocidad gracias a su combinación de fuerza y cálculo.[14]
- Sostenible y saludable, los engranajes rectos de baja fricción, y sin movimientos innecesarios gracias al QuickMove y TrueMove, reducen el consumo de energía hasta en 0.58kW a máxima velocidad.[14]
- Finalmente, cabe destacar que su montaje es totalmente flexible.[14]

7.2 UNIDAD DE CONTROL

En lo referente a la unidad de control, este robot IRB 1600 tiene un controlador llamado IRC5. El IRC5 opera con muchas versiones de RobotWare, pero para este proyecto la única posible debido a la versión proporcionada de RobotStudio es la 6.12.00.00.

El IRC5 tiene un control del movimiento exclusivo de ABB, ofrece también flexibilidad, seguridad, modularidad, interfaces de aplicación, un control para múltiples robots y compatibilidad con herramientas del PC.[15]

El IRC5 puede aparecer en diferentes variantes para proporcionar soluciones rentables y optimizadas en función de las necesidades.[15]

En lo referente al IRC5 se puede decir que:

- Está diseñado para una alta protección IP y una capacidad de expansión total.[15]
- Proporciona un entorno protegido para el equipo auxiliar del robot.[15]
- Capaz de controlar hasta cuatro robots en una configuración MultiMove.[15]
- El IRC brinda a los robots la capacidad de realizar sus tareas de manera totalmente eficiente. Esta basado en un modelo dinámico avanzado. Optimiza automáticamente el rendimiento del robot reduciendo los tiempos de ciclo.[15]
- Es compatible con la mayoría de las redes industriales.[15]



Ilustración 120: IRC5 de ABB

Así mismo, el IRC5 trae consigo un teach pendant. Un teach pendant es un tipo de interfaz HMI diseñada para la programación y verificación de los programas a ejecutar por parte del robot industrial.[16]



Ilustración 121: IRC5 Flex pendant

A continuación, se muestra una imagen de como irán conectados los elementos en la estación en la realidad.



Ilustración 122: Ejemplo conexionado de estación

7.3 HERRAMIENTA DEL ROBOT

El diseño de la herramienta del robot se desarrolló en función del elemento que se debía trasladar. En este caso como las celdas de paletizado de cajas son diferentes, se desarrollaron tres herramientas diferentes en función de los tamaños de las cajas.

Estas herramientas están constituidas por una lámina de aluminio con las dimensiones apropiadas para las cajas, así como, cinco ventosas que mediante el efecto Venturi, succionaran la caja para su posicionamiento en el pallet.

Tamaño caja	Tamaño herramienta
Caja grande (400x400x400)	Herramienta grande (350x350)
Caja mediana (600x300x200)	Herramienta mediana (550x250)
Caja pequeña (350x250x200)	Herramienta pequeña (350x200)

No se ha puesto el tamaño exacto de la caja en la lamina de aluminio de la herramienta del robot ya que no es necesario. Con una medida algo mas pequeña al tamaño de la caja se ajusta de una manera perfecta para que junto con las ventosas se produzca su recogida y dejada de una forma óptima.

Se han programado las posiciones adecuadas para que esta herramienta no colisione ni con el robot, ni con su base, ni mucho menos produzca colisión entre las cajas cuando se disponga a colocarlas en el pallet. Permitiendo así que la herramienta efectúe los giros necesarios (ya sean de 180 grados, 90 grados, 45 grados...) para no colisionar con nada.

A continuación, se pasarán a mostrar los tres modelos de herramientas diseñados en RobotStudio para las tres células de paletizado.

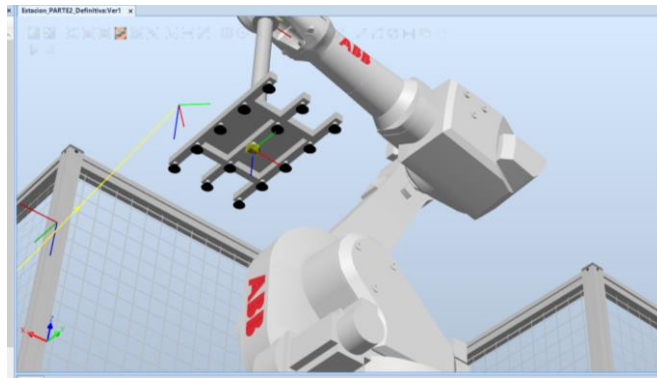


Ilustración 123: Herramienta Cajas Pequeñas

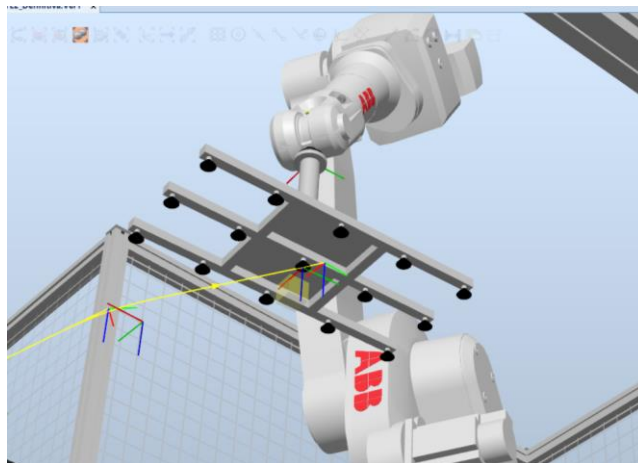


Ilustración 124: Herramienta Cajas Medianas



Ilustración 125: Herramienta Cajas Grandes

7.4 PROGRAMACIÓN DEL ROBOT

Antes de comenzar explicando la programación del robot, cabe destacar dos cosas.

En primer lugar, al tratarse de una simulación de dos partes, no se han podido conectar las células de paletizado con el resto de la fábrica para la ejecución de una única simulación, por lo tanto, se ha seguido el mismo modelo que se aplicó para el diseño en Factory I/O en estas células de paletizado, siendo las mismas, el final de esta línea. Si la simulación fuese real, se usarían unas tarjetas llamadas DeviceNet, con las cuales conectaríamos nuestra célula robótica al PLC. Esta tarjeta, enviaría las señales del PLC al robot indicándole los valores de estas para que ejecute las tareas requeridas en función de estas.

En segundo lugar, cabría destacar que las tres células son idénticas, es decir, se procedió al diseño y programación de la célula de cajas grandes, y mediante pequeñas modificaciones del código RAPID, la herramienta o las posiciones de los elementos se efectuaron dos copias más para las cajas medianas y las cajas pequeñas. Haciendo así una cadena de paletizado que cumple los requisitos y presenta una programación sencilla.

Dicho esto, se pasará a explicar la programación y el desarrollo de una de las células, ya que como se ha indicado con anterioridad las otras dos son idénticas.

En lo referente a la programación, se distinguen tres secciones, la primera es la del diseño de la estación, en ella podremos ver la definición de las trayectorias y puntos tanto de recogida de la caja como la dejada en el pallet. En esta parte también veremos los elementos introducidos como el robot, las cintas, barreras ...

Por otro lado, la programación de las estaciones se lleva a cabo mediante componentes inteligentes, que serán los encargados de dar órdenes y relacionar elementos para que el robot las efectúe.

Y finalmente, se encuentra el lenguaje RAPID, donde programaremos las secuencias y las rutinas que deberá hacer el robot, es una programación algo más compleja si se compara con la anteriormente vista, pero de fácil comprensión.

Una vez comentado esto, nada más se accede a el proyecto realizado, en la esquina izquierda, sección de diseño, se puede observar cómo está constituido por las tres estaciones, así como muchos de elementos.

Bajo los mecanismos, se encuentran los elementos relacionados con las mismas, en la zona de componentes.

Estos componentes son los elementos empleados para el diseño de la estación, es decir, las barreras, las cintas transportadoras, pedestal del robot, flex pendant...

Se ha dividido en tres carpetas, por las tres estaciones, introduciendo en cada una los elementos que la constituyen.

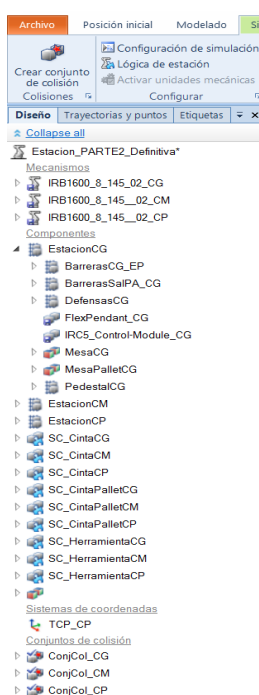


Ilustración 126: Menú diseño RobotStudio

En segundo lugar, se observa algo muy importante en la programación de la primera parte que se ha comentado, estos son los componentes inteligentes. Como ya se remarcó, las estaciones no solo están programadas por el lenguaje RAPID, sino que también se han empleado una serie de componentes inteligentes que darán ordenes, tanto a los sensores como al robot para ejecutar la secuencia.

El primer componente inteligente que se encuentra es el de cinta. Este componente es el encargado del comienzo de la emisión de cajas, así como el inicio de la simulación y del control del sensor que detecta la caja al final de la cinta, donde el robot recibirá la señal de

su llegada y se posicionará a la trayectoria de recogida para efectuarla, así como de la propia cinta que transporta las cajas.

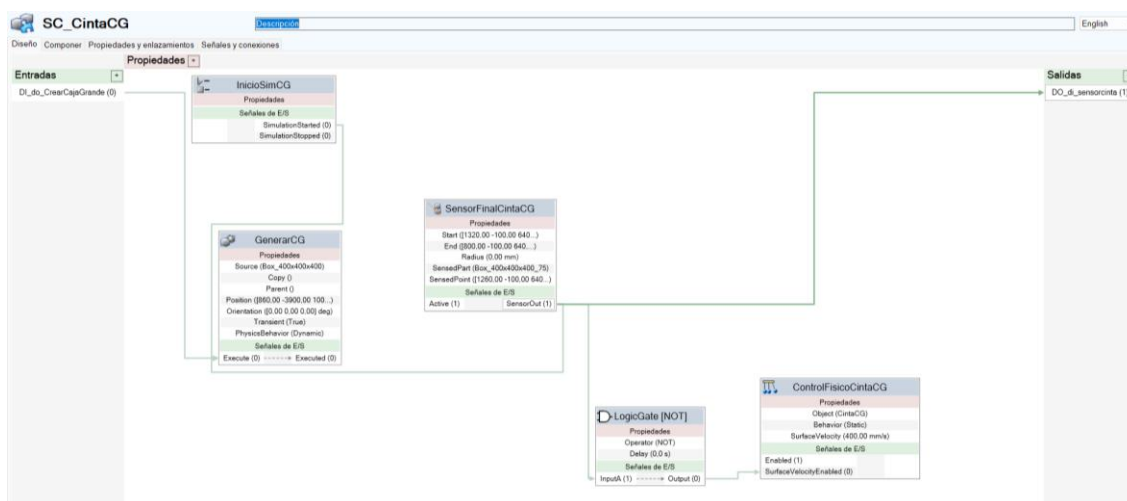


Ilustración 127: Componente Inteligente 1

El segundo componente inteligente es el relacionado con el control físico del pallet. Este componente es el encargado tanto de la generación del pallet, la retirada de este y el control del sensor para la entrada y la salida del pallet.

Como se puede observar, se han declarado entradas que serán las que le den los avisos al robot para la ejecución de las secuencias, estas son las que irán ligadas al controlador en la lógica de la estación.

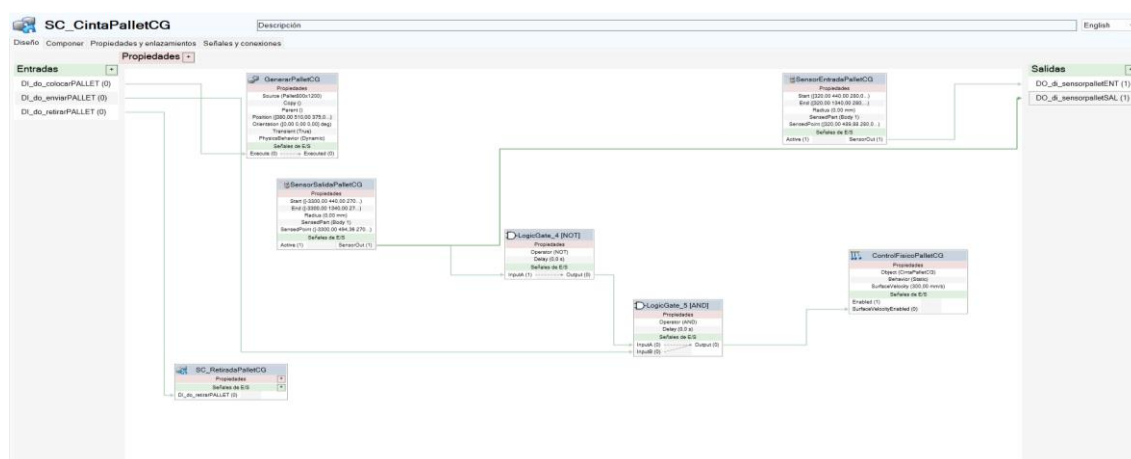


Ilustración 128: Componente Inteligente 2

Finalmente, el ultimo componente inteligente es el relativo a la herramienta del robot. En este componente lo que se realizara es la activación y desactivación de las ventosas de la herramienta, así como el control del sensor que este situado en la herramienta para la recogida y dejada de las cajas.

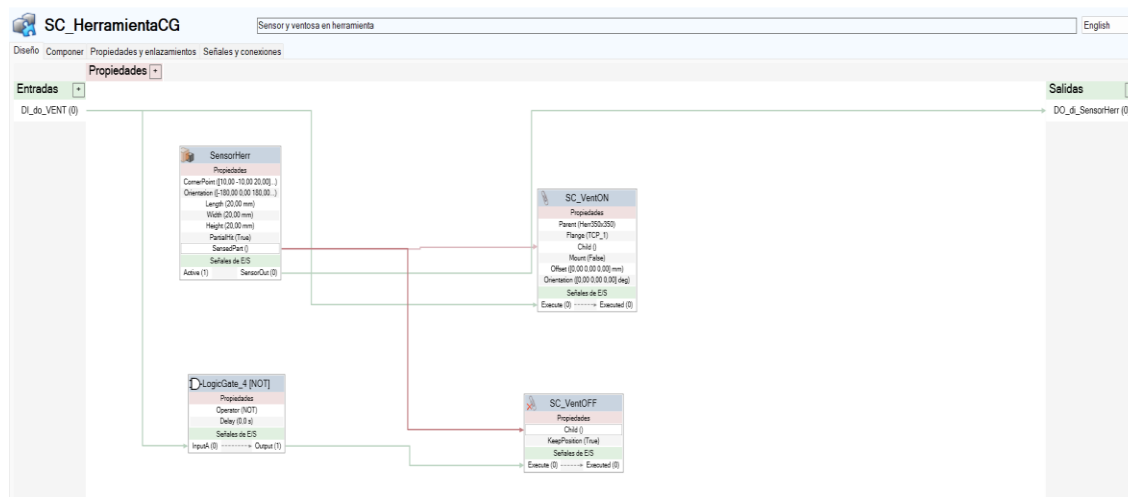


Ilustración 129: Componente Inteligente 3

También se puede observar que se han añadido conjuntos de colisiones. Los conjuntos de colisiones que se han establecido para estas estaciones son los siguientes:

El conjunto A: es el relativo al robot con la herramienta.

El conjunto B: es el relativo al robot con las barreras de seguridad.

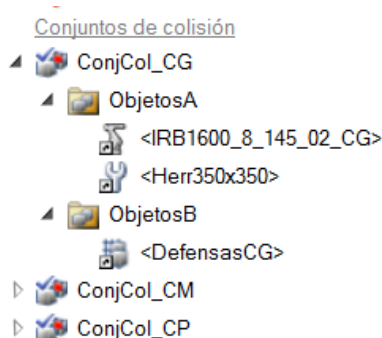


Ilustración 130: Conjunto de colisiones

Una vez se han explicado los componentes inteligentes y se ha enseñado el diseño de la estación, se pasará a mostrar la lógica de la estación. Esto consiste en la conexión entre todos los componentes inteligentes anteriormente explicados y el controlador, mediante las señales declaradas.

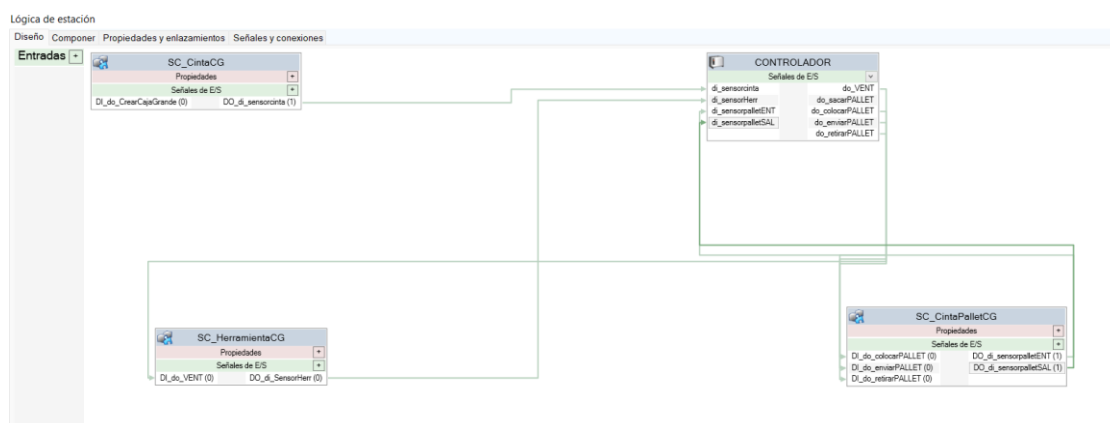


Ilustración 131: Lógica de estación

Por último, se explicará la programación llevada a cabo en el lenguaje RAPID.

El lenguaje RAPID es el relativo al controlador, es decir, si no se sincroniza ese código con la estación, los robots no tendrán las ordenes programadas en este. Lo mismo ocurre si no se sincronizan los datos de la estación al controlador.

El programa RAPID esta constituido por varios sectores.

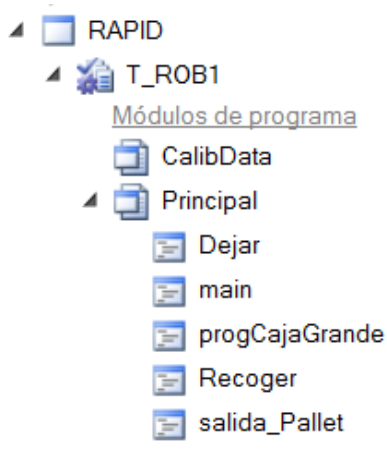


Ilustración 132: Estructuración RAPID

En primer lugar, se encuentra un programa llamado Principal, que será donde se encuentre todo el código que hemos programado.

A continuación, las primeras líneas de código son las constantes y variables de posicionamiento declaradas para usar en el programa.

```

-----
!Declaraciones de posición para caja Grande
CONST robtarget Casa:=[400,0,727.717782649],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_Rec2:=[19.998,187.309,-112.062],[0,0,0,1],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_Rec1:=[259.998,287.309,-62.062],[0,0,0,1],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_Rec3:=[259.998,287.309,-12.062],[0,0,0,1],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_RecSal:=[39.998,-412.691,-162.062],[0.382683432,0,0,0.923879533],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_dej2:=[391.465778884,291.477069422,-162.646834995],[0.707106781,0,-0.024677671,0.706676031],[1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_dej1:=[241.465778884,241.477069422,-112.646834995],[0.707106781,0,-0.024677671,0.706676031],[1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
CONST robtarget p_dej3:=[190,190,-62],[0.707106781,0,0,0.707106781],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
!Variables de desplazamiento persistentes para que no se pierda la colocación con un reinicio
PERS pose DespCaja:=[0,400,0],[1,0,0,0];
PERS num Despx;
PERS num Despy;
PERS num Despz;
CONST robtarget p_dej3:=[541.465746741,341.477091992,-212.6469008],[0.707106832,0.000000004,-0.024677683,0.706675979],[1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
VAR btnres br_ColocaPallet;
VAR btnres br_RetiraPallet;

```

Ilustración 133: Variables y Constantes en RAPID

Pasaremos a explicar el MAIN. Este es el programa principal, en donde se programan las llamadas a las acciones de colocación de pallet al inicio de la cinta de rodillos, el enviar pallet al final de la cinta y retirarlo, activación de la ventosa y finalmente la posición de inicio del robot (casa), posición la cual adoptará una vez ha terminado de llenar el pallet y el operario aún no ha indicado que se coloque otro en el inicio de la cinta.

```

PROC main()
  PDispOff;
  !Nos aseguramos que todas las señales están a OFF
  SetDO do_colocarPALLET,0;
  SetDO do_enviarPALLET,0;
  SetDO do_retirarPALLET,0;
  SetDO do_VENT,0;
  !Movimiento a posición segura
  MoveJ Casa,v1000,fine,TCP_1\WObj:=wobj0; !Posición segura del robot
  progCajaGrande; !Proceso colocación de cajas
  !Fin Proceso
ENDPROC

```

Ilustración 134: MAIN

Para el programa de caja grande, lo que se ha programado es la colocación de las cajas en el pallet mediante una matriz. Así mismo este programa es el encargado de sacar un aviso a la consola de operario indicándole si desea colocar un pallet en la entrada para llenarlo de cajas o retirarlo una vez está completo.

```
PROC progCajaGrande()
!Programa que ejecuta la matriz de colocacion
!mediante desplazamientos y llama a los programas
!Prog: Recoger y Dejar
DespCaja:=[[0,0,0],[1,0,0,0]];
Despx:=0;
Despy:=0;
Despz:=0;
!Si no detecta pallet en entrada advierte
!Para la simulación ejecuta la aparición del pallet
IF di_sensorpalletENT = 0 THEN
    repito_preg_entrada:
        UIMsgBox\Header:="No Hay Pallet en Entrada",":Desea Colocar?"\Buttons:=btnYesNo\Icon:=iconInfo\Result:=br_ColocaPallet;
        !Coloca el pallet mediante el SC,
        !en la vida real solo esperaría la confirmación de que el pallet esta colocado
        !y lo confirmaría con la señal del sensor del pallet de entrada
        IF br_ColocaPallet=6 THEN
            SetDO do_colocarPALLET,1;
            WaitTime 2;
            WaitDI di_sensorpalletENT,1;
        ELSE
            GOTO repito_preg_entrada;
        ENDF
ENDIF
```

Ilustración 135: programa cajas grandes (1)

```
ENDIF
SetDO do_colocarPALLET,0;
ENDIF

3 FOR k FROM 0 TO 1 DO
    Despx:=0;
    Despy:=0;
    DespCaja.trans.z:=- (Despz*400);
    Despz:=Despz+1;
3 FOR j FROM 0 TO 2 DO
    Despy:=0;
    DespCaja.trans.x:=Despx*400;
    Despx:=Despx+1;
3 FOR i FROM 0 TO 1 DO
    DespCaja.trans.y:=Despy*400;
    Despy:=Despy+1;
    Recoger;
    Dejar;
    ENDFOR
    ENDFOR
    ENDFOR
    salida_Pallet;
!Fin Proceso
ENDPROC
```

Ilustración 136: Programa cajas grandes (2)

Continuando con el código, aparece el programa recoger. En este se han definido las coordenadas previamente con el robot generando así una trayectoria de recogida de la caja. Una vez está la trayectoria programada, se sincronizaron los puntos de la estación al controlador y se generó el código RAPID.

```
PROC Recoger()
    !Programa de recogida de pieza, activa ventosas venturi y comprueba
    !que tiene la caja
    MoveJ p_Rec2,v1500,z200,TCP_1\WObj:=Wobj_cinta;
    MoveL p_Rec1,v400,fine,TCP_1\WObj:=Wobj_cinta;
    WaitDI di_sensorcinta,1;
    MoveL p_Rec,v50,fine,TCP_1\WObj:=Wobj_cinta;
    !! ejecuto el coger la caja
    WaitDI di_sensorHerr,1;
    SetDO\Sync,DO_VENT,1;
    MoveL p_Rec1,v50,z50,TCP_1\WObj:=Wobj_cinta;
    MoveL p_Rec2,v300,z100,TCP_1\WObj:=Wobj_cinta;
    MoveL p_RecSal,v300,z100,TCP_1\WObj:=Wobj_cinta;
    !Fin Proceso
ENDPROC
```

Ilustración 137: Recoger

De igual forma que el programa recoger, se ha generado el código y los puntos para el programa dejar.

```
PROC Dejar()
    PDispSet DespCaja;
    MoveL p_dej3,v400,z200,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej2,v400,z100,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej1,v200,z50,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej,v50,fine,TCP_1\WObj:=Wobj_Pallet;
    setdo\Sync,DO_VENT,0; !! ejecuto el coger la caja
    MoveL p_dej1,v200,z50,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej2,v1000,z100,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej3,v1500,z200,TCP_1\WObj:=Wobj_Pallet;
    !Fin Proceso
    PDispOff;
ENDPROC
```

Ilustración 138: Dejar

Finalmente, se ha programado la salida del pallet una vez está completo. Se ha generado al igual un mensaje para el operario que le pregunta si desea que el pallet sea retirado eliminándolo así del final de la cinta.

```
PROC salida_Pallet()
  MoveJ Casa,v1000,fine,TCP_1\WObj:=wobj0;
  WaitTime 2;
  !Si hay un Pallet en salida hay que retirarlo
  !Esto lo hacemos con un mensaje en pantalla
  !Para la simulacion ejecuta la retirada del pallet
  IF di_sensorpalletSAL = 1 THEN
    repito_preg_salida:
    UIMsgBox\Header:="Pallet en Salida","¿Desea Retirarlo?"\Buttons:=btnYesNo\Icon:=iconInfo\Result:=br_RetiraPallet;
    !Retira el pallet mediante el SC,
    !en la vida real solo esperaria la confirmacion de que el pallet esta retirado
    !y lo confirmaria con la señal del sensor del pallet de salida
    IF br_RetiraPallet=6 THEN
      SetDO do_retirarPALLET,1;
      WaitTime 2;
      WaitDI di_sensorpalletSAL,0;
    ELSE
      GOTO repito_preg_salida;
    ENDIF
    SetDO do_retirarPALLET,0;
  ENDIF
  !Enviamos el pallet a la salida
  SetDO do_enviarPALLET,1;
```

Ilustración 139: Salida pallet (1)

```
IF br_RetiraPallet=6 THEN
  SetDO do_retirarPALLET,1;
  WaitTime 2;
  WaitDI di_sensorpalletSAL,0;
ELSE
  GOTO repito_preg_salida;
ENDIF
SetDO do_retirarPALLET,0;
ENDIF
!Enviamos el pallet a la salida
SetDO do_enviarPALLET,1;
WaitDI di_sensorpalletSAL,1;
SetDO do_enviarPALLET,0;
progCajaGrande;
!Fin de Proceso
ENDPROC

ENDMODULE
```

Ilustración 140: Salida pallet (2)

8. LÍNEAS FUTURAS

Respecto a las posibles mejoras que podrían llevarse a cabo, destacaría la conexión del PLC con los softwares a simular, tanto con Factory I/O como con Robot Studio.

En lo referente al software de Robot Studio, cabría mejorar la sincronización de las estaciones con el controlador, así como el programa RAPID, ya que muchas veces falla o no transfiere toda la información implementada en cualquiera de las dos partes, ya sea por la estación o por el controlador. Este fallo ralentiza mucho la realización de las estaciones o sus copias.

Por otro lado, otras de las mejoras que no se han podido introducir por la disponibilidad que proporciona Factory I/O son las siguientes:

- Mejorar la velocidad de las cintas para que el proyecto tenga una mayor productividad, ya que las cintas de las que dispone Factory I/O son fijas a una velocidad.

- Mejorar la velocidad y precisión de los cilindros, ya que muchas veces estos actuadores tardan en clasificar las piezas o fallan al colocar la pieza en el ramal, ocurre lo mismo con las pletinas clasificadoras.

- Mejoras en los movimientos de las células de Pick and Place, así como mejoras al establecer las coordenadas para la recogida y dejada de elementos.

- La Introducción de plataformas levadizas para una mejor clasificación y un ahorro en cintas, ya que con esta plataforma dispondríamos de 3 posiciones (arriba, en medio, abajo) para clasificar evitando elementos como cilindros, pletinas... así como incorporar posibles ramificaciones para que en caso de fallo en el sistema clasificador no sea necesario parar la producción.

- Introducción de elementos óptimos para el almacenamiento de este tipo de piezas (cajas con un aspecto y forma más adecuados), ya que las cajas que nos proporciona el software no se ajustan de un modo real a los tipos de piezas que se producen.

- Por último, la introducción un doble chequeo de sensor de visión para poder asegurar una clasificación optima en caso de fallo en el primer sensor.

9. CONCLUSIONES

El objetivo de este proyecto era la realización de un sistema para la clasificación, distribución y paletizado de piezas, el cual, mediante un sensor de visión, pudiese clasificar las piezas en tres tipos diferentes, para su posterior almacenamiento y paletización en cajas de diferentes tamaños.

La motivación principal de este proyecto fue la utilización de un software nuevo llamado Factory I/O, así como una ampliación de conocimientos en los softwares TIA Portal y Robot Studio estudiados años atrás. Poder visualizar la programación desarrollada en TIA Portal en Factory I/O era un reto nuevo que no se tuvo al estudiar la programación de PLC's y automatización.

Se comenzó estudiando el funcionamiento de Factory I/O, sus elementos, y su rango de posibilidades a la hora de desarrollar la escena, al igual que su correcto funcionamiento. Posteriormente se investigó la conexión de este programa con TIA Portal y se verificó lo aprendido con ejemplos sencillos usando cintas o diferentes sensores, que más adelante se fue utilizando para el proyecto final.

Cuando ya se visualizó el alcance de ambos softwares, se fueron implementando más elementos a la vez que se iban programando en el PLC. Al ser Factory un Software algo limitado en cuanto a los elementos a emplear, se divisaron dos opciones de desarrollo en la escena, y tras estudiar ambas estructuras, se decidió que la mejor para la implementación de esta era la ya mostrada, ya que era lo más parecido a un proceso real.

Una vez se terminó el desarrollo de la primera etapa con Factory I/O y TIA Portal, se procedió a determinar cómo estructurar las células de paletizado.

De igual modo que sucedió en el caso de Factory I/O que fue necesario estudiar los elementos del programa, aunque ya se había trabajado con RobotStudio, también fue necesario revisar los paquetes que ofrecía el software y cuáles serán los más adecuados para la realización de esta parte.

Tras desarrollar una de las estaciones de la forma más similar a las empleadas en industria, se estableció que las otras dos serían idénticas a la primera generada, pero con modificaciones tanto en las cajas, como en la herramienta empleada por el robot.

Por último, hay que destacar que este proyecto es una solución general al problema que se nos planteó, aunque se pueden incorporar algunas mejoras y evoluciones como las que se han descrito en el apartado de líneas futuras.

10. REFERENCIAS

[1] Automatizacion

La Automatización ¿Qué es? | Definición, Importancia y Objetivos. (ripipsacobots.com)

[2] Autómata Programable. Definición

Qué es un Autómata Programable o PLC y cómo se programa (academiaintegral.com.es)

[3] clasificación del PLC

PLC: Controlador Lógico Programable: Clasificación de PLC
(controladoreslogicosprogramables.blogspot.com)

[4] Manual de SIEMENS correspondiente al TIA portal y por lo tanto a la automatización del proyecto:

SIMATIC TIA Portal STEP 7 Basic V10.5 (siemens.com)

[5] Informacion sobre el uso de los bloques de funcion:

https://youtu.be/8bcv5iupH_I

Guia de Inicio en Siemens TIA Portal - Bloques de Función (En) - infoPLC

[6] Factory I/O

Acerca de - FACTORY I/O (factoryio.com)

[7] Robots Industriales. Tipos

Tipos de robots industriales: clasificación y características | Esneca

[8] Robot Studio. Historia y Software

[Introducción a la utilización de RobotStudio ABB \(monografias.com\)](http://monografias.com)

[9] Lenguaje RAPID

[Microsoft PowerPoint - RAPID.ppt \(biada.org\)](http://biada.org)

[10] Elementos de Entrada de Factory I/O

[Sensores - FACTORY I/O \(factoryio.com\)](http://factoryio.com)

[11] Elementos de Salida de Factory I/O

[Piezas - FACTORY I/O \(factoryio.com\)](http://factoryio.com)

[12] Célula de paletizado

<https://gruposim.eu/blog/para-que-sirven-las-celulas-robotizadas-de-paletizado/>

[13] Robot antropomórfico

[Robots Antropomorficos \[34m7xx5xop46\] \(idoc.pub\)](http://idoc.pub)

[14] IRB 1600 ABB

[IRB 1600 - Robots industriales | ABB](http://ABB)

[15] IRC 5

[IRC5 - IRC5 overview \(Controladores\) | ABB](#)

[16] Teach Pendant ABB

[Teach pendant para robot industrial | Industrias y Empresas](#)

DOCUMENTO 2. PLIEGO DE CONDICIONES

ÍNDICE DOCUMENTO 2:

1.CONDICIONES GENERALES DEL CONTENIDO DEL PROYECTO.....	143
2.CONDICIONES FACULTATIVAS.....	143
3.CONDICIONES ECONÓMICAS.....	144
4.CONDICIONES TÉCNICAS	145
5.NORMATIVAS.....	147
5.1. NORMATIVA GENERAL.....	147
5.2. NORMATIVA ESPECÍFICA.....	149
6.REFERENCIAS.....	151

PLIEGO DE CONDICIONES

El objetivo de este documento es fijar las condiciones mínimas que se deben cumplir para la automatización del sistema, especificando los requisitos de durabilidad, fiabilidad y seguridad.

El dominio de la aplicación de este documento engloba todos los sistemas electrónicos, eléctricos, mecánicos e hidráulicos que forman el mismo.

1.CONDICIONES GENERALES DEL CONTENIDO DEL PROYECTO.

En este apartado deberíamos especificar los datos tanto del promotor como del contratista. En este caso el promotor sería el que ha realizado el proyecto, y el contratista el que pidió realizar una solución a este problema y por tanto la ejecución del proyecto.

En este documento se recogerán todas las especificaciones técnicas estipuladas, en planos, y en la memoria del proyecto, exigiendo pues los aspectos técnicos de los materiales y medios a emplear durante el transcurso de este, al igual que los equipos necesarios para el correcto cumplimiento del proyecto.

El pliego de condiciones tiene como objetivo, transmitir los aspectos técnicos de calidad que se deben cumplir en el proyecto, demandando las intervenciones necesarias para el cumplimiento del contrato y de la legislación aplicable, al contratista a la propiedad, ...

El objetivo del proyecto es el desarrollo de un sistema para la clasificación, distribución y paletizado de productos terminados, en este caso se ha realizado y detallado el funcionamiento de un sistema de clasificación distribución y paletizado de piezas en función de su color.

2.CONDICIONES FACULTATIVAS

En lo referente a las condiciones facultativas, se podrán destacar principalmente, las obligaciones del director de montaje, las obligaciones del director de programación y las facultados de la dirección técnica.

En cuanto a las obligaciones del director de montaje, las especificaciones técnicas que se declaran deben estar aceptadas por el mismo y serán de obligado cumplimiento.

Por otro lado, el director de montaje tiene la responsabilidad de encargarse de la completa dirección de la obra desde la fase de anteproyecto hasta la ejecución de esta. Será el encargado de la coordinación de las diferentes áreas de la empresa, además de asumir y redactar todas las modificaciones de la instalación, teniendo en cuenta la normativa de prevención de riesgos laborales y medioambientales.

Todos los trabajos del proyecto serán llevados a cabo por personas capacitadas, tanto en lo referente al conocimiento como en la ejecución de este.

Igualmente, el director de montaje será el encargado del cumplimiento de todas las normativas de trabajo, ya que, será el responsable ante los tribunales de los posibles accidentes que por desconocimiento o mala ejecución se cometan a lo largo del proyecto.

En segundo lugar, están las obligaciones del director de programación, que será el encargado de aceptar las especificaciones técnicas que se declaran y que serán de obligado cumplimiento.

La instalación del autómeta y la programación en el proyecto deberá haber sido realizada por personas preparadas y capacitadas con el conocimiento teórico- práctico sobre el autómeta de la instalación.

Se hará cargo también, del cumplimiento de las especificaciones indicadas en la memoria del proyecto, así como de las dificultades y complicaciones que se tengan a lo largo de la programación del autómeta.

Finalmente, se encuentran las facultades de la dirección técnica. Queda indicado que todas las dificultades o problemas que tenga el director de montaje serán resueltas por la dirección facultativa que están en concordancia con el pliego de condiciones técnicas.

Los materiales que utilizar para el proyecto deberán ser analizados y examinados antes del montaje, sin la aprobación de estos, no podrán ser empleados.

Se podrá prescindir de los artículos que no tengan las condiciones o características necesarias para su uso.

3. CONDICIONES ECONÓMICAS

Para el desarrollo de esto, se realiza un estudio de precios del mercado para componente del proyecto. Se han escogido los más adecuados para el mismo, teniendo en cuenta el

precio. Una vez se han elegido los componentes necesarios, se multiplicará el precio de la unidad por el número necesario de estos y obtendremos así un presupuesto.

Los precios no se encuentran sujetos a nada fijo, hay variables externas a nosotros que hacen que fluctúen a lo largo del tiempo. Se puede determinar que, a largo plazo, los elementos mejorarán por lo que los elegidos en este momento podrán quedar obsoletos y por consiguiente tendrán precios más reducidos. Cabe señalar que otro de los factores influyentes en el precio de los productos es la demanda de estos.

4. CONDICIONES TÉCNICAS

Se destacarán cuatro partes importantes para el proyecto.

1. Normas de mantenimiento del autómata

El mantenimiento del autómata programable es fundamental para garantizar el correcto funcionamiento en el proceso de fabricación. Para garantizar el buen funcionamiento, se llevará a cabo un mantenimiento preventivo del PLC, para esto, hay que tener en cuenta que al PLC le afectará el entorno en el que se encuentre y es importante que para que se lleve a cabo de forma correcta este mantenimiento deberán tenerse en cuenta los siguientes aspectos:

- Comprobación de la temperatura del entorno.
- Desconexión del PLC.
- Limpieza del panel y la placa electrónica.
- Comprobaciones de conexiones E/S.
- Revisar el estado de la batería.
- Calibración de las tarjetas del circuito y los sensores.
- Guardado de copias de seguridad del software.

2. mantenimiento preventivo de cintas transportadoras, sensores y actuadores neumáticos

Para el mantenimiento diario de los equipos debemos comprobar el correcto funcionamiento de las cintas transportadoras, la limpieza y comprobación de los sensores y actuadores neumáticos, así como la realización de una última comprobación de los componentes del

cuadro eléctrico principal, ya que en caso de que no funcionaran de forma óptima deberán ser remplazados por otro.

Por otra parte, el mantenimiento semanal consistirá en la comprobación de las cintas transportadoras, así como de los accionadores neumáticos.

El mantenimiento mensual, constara por la comprobación de la dureza de los rodamientos y el chequeo de la calibración de la sensórica.

Finalmente, el mantenimiento anual será para el nivelado de las cintas y se comprobará tanto el estado funcional de los sensores como la batería del PLC.

3.Mantenimiento preventivo del Robot

Para el mantenimiento diario se deberá chequear el correcto funcionamiento de los brazos, herramientas y sensores que constituyen la célula robótica.

Por otro lado, para el mantenimiento semanal, será necesario una inspección de cableado y limpieza general, sin embargo, para el chequeo mensual, nos bastará con la inspección de topes mecánicos.

Finalmente, para el mantenimiento anual, se llevará a cabo una comprobación de las baterías, que deberán ser sustituidas cada 3 años aproximadamente.

4. Mantenimiento general

De forma general, se deberá garantizar el correcto estado de cada puesto de trabajo, tanto en lo referente a la limpieza como a las inspecciones diarias del funcionamiento de la instalación y equipos.

Cada operario será encargado de revisar su puesto de trabajo y de mantenerlo en las condiciones óptimas para su trabajo en el mismo. Así mismo está obligado a avisar de cualquier fallo o incidencia que se encuentre en el mismo.

5. NORMATIVAS

En este proyecto se cumplen todas las normativas que se van a indicar a continuación. Se explicarán de forma breve la normativa de industria para controladores lógicos programables (IEC 61131-1), así como la aplicada a celdas robotizadas.

5.1 NORMATIVA GENERAL:

-Material eléctrico:

En relación a la normativa europea con relaciona el material eléctrico aplicación de la directiva 2006/95/CE del parlamento Europeo, relativo a la legislación y aprobación de esta de los miembros sobre el material eléctrico, destinado a utilizarse con diferentes límites de tensión.

-Maquinas:

R.D. 1644/2008, el Parlamento Europeo aprueba la directiva de 2006/42/CE, en relación a las máquinas y por la que fue modificada la directiva 95/16/CE. Este real decreto tiene como objetivo instaurar las normas relativas a la comercialización y puesta en servicio de las maquinas

-Instalación y empleo de Robots:

A continuación, se van a redactar las normativas más relevantes a nivel mundial:

- Normativa internacional ISO/TC 299: estandarización en el campo de los robots y dispositivos robóticos controlados automáticamente, reprogramables y manipulables, programables en más de un eje y fijos o móviles. [1]
- Normativa internacional ISO 10218-1 2012: basada en la norma ISO 10218-1 1992. En ella se especifican requisitos y pautas para el diseño seguro inherente, medidas de protección e información para el uso de robots industriales. Detalla los peligros básicos asociados con los robots y proporciona requisitos para eliminar o reducir adecuadamente los riesgos asociados a estos peligros.
- Normativa americana ANSI/RIA R15.06-1992: el Instituto Nacional de Normalización de Estados Unidos (ANSI) describe de manera superflua una guía y sus respectivos requisitos para poder llevar a cabo un diseño seguro con las medidas necesarias para la

protección de los robots. Así como los posibles riesgos básicos relacionados con los robots o pautas necesarias para eliminar o reducir el porcentaje de riesgos asociados a dichos peligros. [2]

Existen normativas que definen aspectos más específicos como:

- Normativa europea UNE-EN 60204-1 2019: se aplica a los equipos y sistemas eléctricos, electrónicos programables de las máquinas fijas o amovibles en funcionamiento, incluyendo un grupo de máquinas que trabajan conjuntamente de forma coordinada entre hombre-máquina. [3]
- Normativa europea UNE-EN 418 1993: explica los aspectos fundamentales de los equipos de parada de emergencia, y los principios de diseño. Esta norma presupone la conformidad con los requisitos básicos de la directiva sobre máquinas. La función principal del paro de emergencia es como elemento preventivo ya que su función es evitar posibles riesgos entre robot y cualquier objeto próximo a él. En el momento de un posible riesgo, se debe presionar dicho botón y así poder parar el proceso.[3]
- EN 294: distancias de seguridad para impedir que las extremidades superiores alcancen zonas peligrosas.
- EN 1037: prevención de una puesta en marcha accidental.
- EN 1050: principios para la evaluación del riesgo.
- EN 954-1: partes de los sistemas de control relativas a la seguridad.
- EN 953: protectores.
- EN 999: posicionamiento de los equipos de protección en función de la velocidad de aproximación de las partes del cuerpo.
- EN 1088: dispositivos de enclavamiento asociados a resguardos.

-Energías renovables y eficiencia energética:

La directiva UE 2018/2001 en relación al uso de energía procedente de fuentes renovables, modifica y renueva los cambios de la directiva 2009/28/CE relativa al fomento de estas energías y por lo que se modifica las directivas 2001/77/CE y 2003/30/CE

-Seguridad e Higiene:

Las normas básicas de seguridad, salud e higiene tratan de unificar las condiciones generales de trabajo necesarias y las medidas para asegurar la salud de los trabajadores, así como de prevenir posibles accidentes laborales y promover el cuidado de la maquinaria, herramientas y materiales con los que se trabaja, y el uso de las prendas adecuadas para cada actividad y manejo del utillaje.

Existe una norma internacional, la OHSAS 18001 para la gestión de la seguridad y salud laboral en las organizaciones. Esta norma deriva de la BS 8800 desarrollada por BSI.

-Prevención de riesgos laborales

En el R.D. se establecen las medidas mínimas de seguridad y salud que se deben de disponer en los lugares de trabajo, en concordancia con las obligaciones establecidas.

En relación a la seguridad frente al riesgo eléctrico el R.D. 614/2001.

Legislación que se aplica a las instalaciones eléctricas en los puestos de trabajo.

5.2 NORMATIVA ESPECÍFICA:

En el PNE 157001, se definen los aspectos generales que deben de incluirse en un proyecto, se deberá de aplicar para la realización de los proyectos.

-Representación de planos y esquemas electricos

Aunque en este proyecto no se han incluido ni realizado planos para la realización del proyecto de forma virtual, si se desea llevar a cabo de forma física, se deberían incluir planos de la instalación, así como de la misma en sí. Además de cumplir las siguientes normativas en los mismos.

- IEC 61082: Simbología y representación de esquemas eléctricos.
- UNE-EN ISO 5455:1996: Escalas normalizadas en los planos y esquemas.
- UNE 102795: Plegado y normalización de planos y esquemas

-Seguridad en Máquinas

Para este apartado , se adjuntan las especificaciones y aspectos generales de la seguridad en maquinas [4]

-Programación de HMI:

En lo referente a la programación de las pantallas HMI, se han adjuntado algunas de las normativas mas importantes para la misma.[5]

- ISA 101.
- IEC 60447 (actuadores principales).
- IEC 61310-1 (Seguridad en máquinas).
- ISO 3864 (Colores y símbolos de seguridad).
- ISO 8201 (audios para señales de emergencia y evacuación).

6.REFERENCIAS

Documentación de apoyo en la normativa:

[1] Normas Técnicas en Seguridad Robótica.

https://issga.xunta.es/export/sites/default/recursos/descargas/documentacion/material-formativo/relatorios/2017_05_CO_Industria_4.0_Arribas.pdf

[2] Normatividad en cuanto a Robótica.

[innovaciones tecnologicas: NORMATIVIDAD EN CUANTO A ROBOTICA \(novatecnologiajorge.blogspot.com\)](http://innovaciones-tecnologicas.blogspot.com/2014/06/NORMATIVIDAD-EN-CUANTO-A-ROBOTICA.html)

[3] Seguridad de las máquinas. Equipo eléctrico de las máquinas. Parte 1: Requisitos generales” UNE-EN 60204-1.

<https://www.aenor.com/normas-y-libros/buscador-de-normas/UNE?c=N0061674>

[4] Información sobre la normativa del uso de maquinaria.

<http://verificacionmaquinaria.lineaprevencion.com/consideraciones-generales/marco-normativo/real-decreto-1644-2008>

[5] Directiva del parlamento europeo y del consejo.

<https://boe.es/doue/2014/096/L00357-00374.pdf>

DOCUMENTO 3. PRESUPUESTO

ÍNDICE DOCUMENTO 3:

1.PRESUPUESTO	154
----------------------------	------------

PRESUPUESTO

Para el desarrollo de este proyecto se tendrá en cuenta todo el trabajo realizado tanto en el aspecto de control y comunicación de los softwares, mano de obra, así como las licencias para el desarrollo de este. Se incluirá el autómatas, los módulos de entradas y salidas que se añadirían en caso de que el proyecto se llevase a cabo de forma física, así como el cableado empleado y los softwares utilizados en el desarrollo de este. Finalmente, también se tendrá en cuenta la documentación del proyecto.

Descripción	Cantidad	Precio/Unidad	Precio Total
Análisis y decisión de la solución del proyecto	20h	25 €/hora	500,00 €
Programación de la solución obtenida	90h	25 €/hora	2250,00 €
Documentación del proyecto	80h	25 €/hora	2000,00 €
Licencia TIA Portal (SIMATIC STEP 7 V14)	1	140,00 €	140,00 €
Licencia SIMATIC WinCC	1	391,50 €	391,50 €
Licencia Factory I/O anual	1	144,00 €	18 €
Licencia RobotStudio ABB anual (paquetes a parte)	1	1500, 00 €	1500, 00 €

Controlador S7-1200 CPU 1214C DC/DC/DC 6ES7214-1AG40- 0XB0	1	382,50 €+ 21% IVA	462,82 €
Módulo entradas /Salidas digitales DI 16/ DQ 16 X 24V 6ES7 223-1BL32- 0XB0	5	277,20 €+ 21% IVA	1677,05 €
Módulo salidas analógicas AQ4X14 6ES7 232-4HD32- 0XH0	3	425,00 €+ 21% IVA	1542,75 €

Presupuesto Total: 10.482,12€

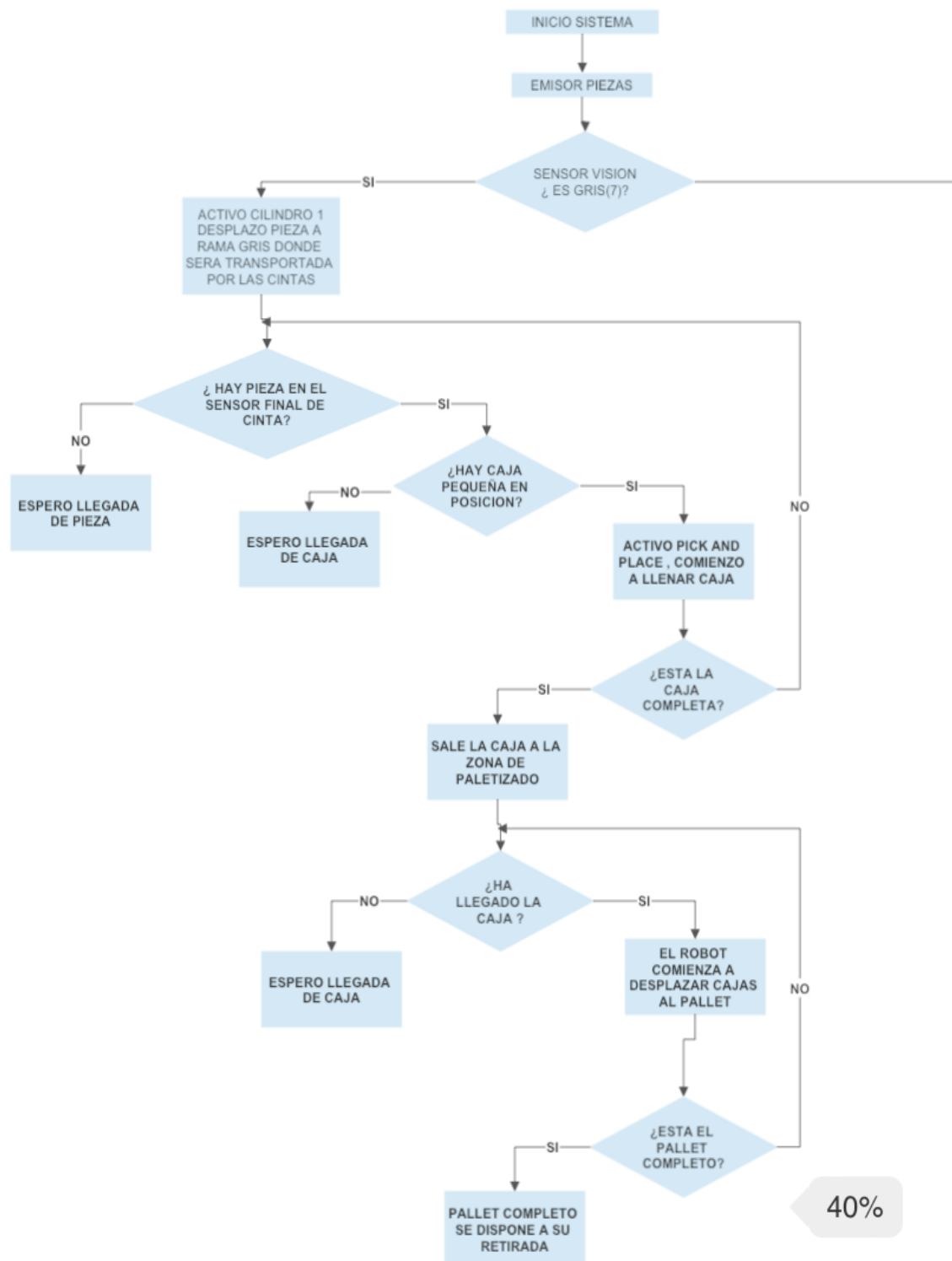
Este presupuesto queda sujeto a modificaciones en función a las características y requisitos de la solución llevada a terreno físico. Ya que los precios que han sido tomados de empresas con stock y estos pueden ser variables.

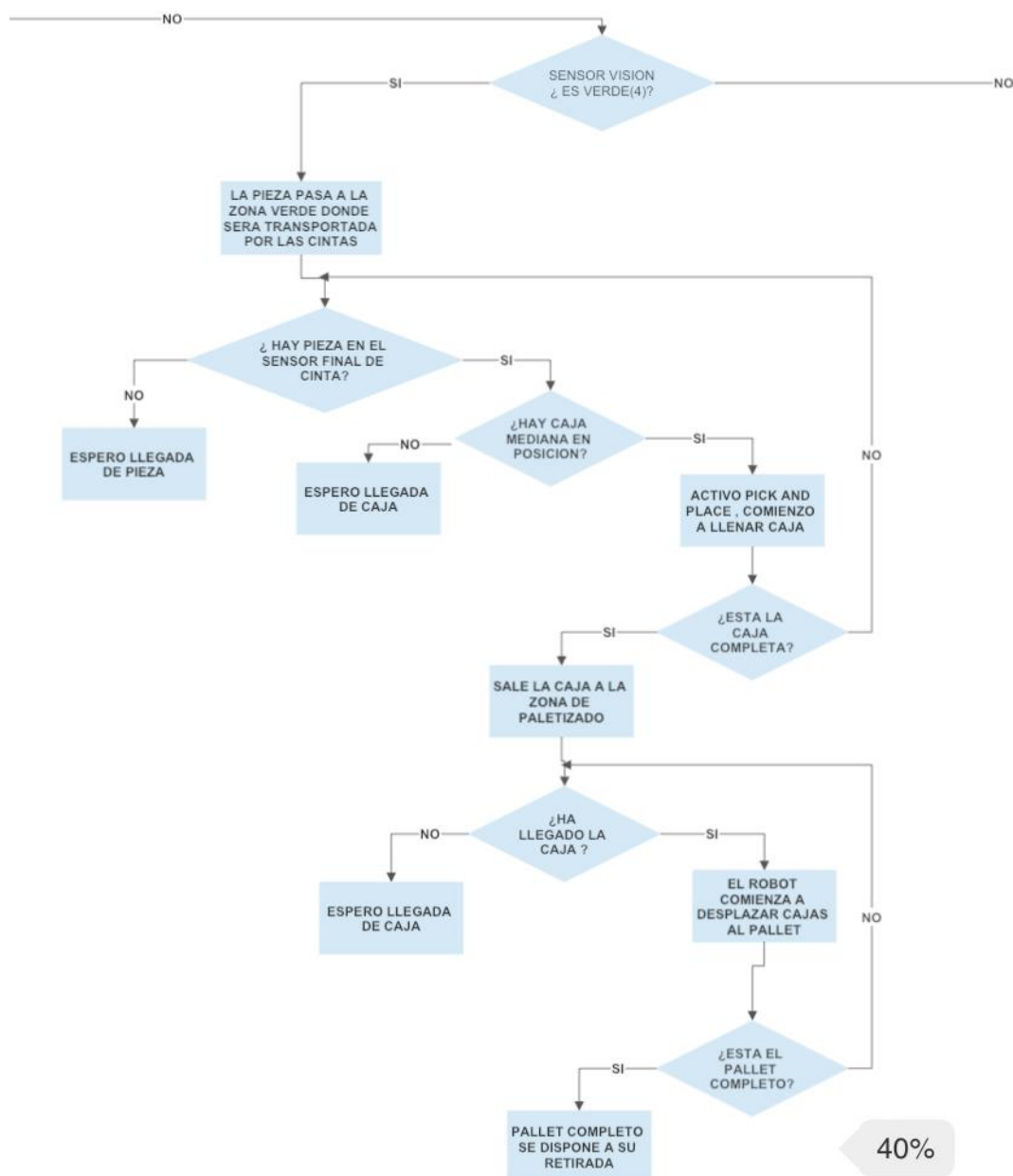
DOCUMENTO 4. ANEXOS

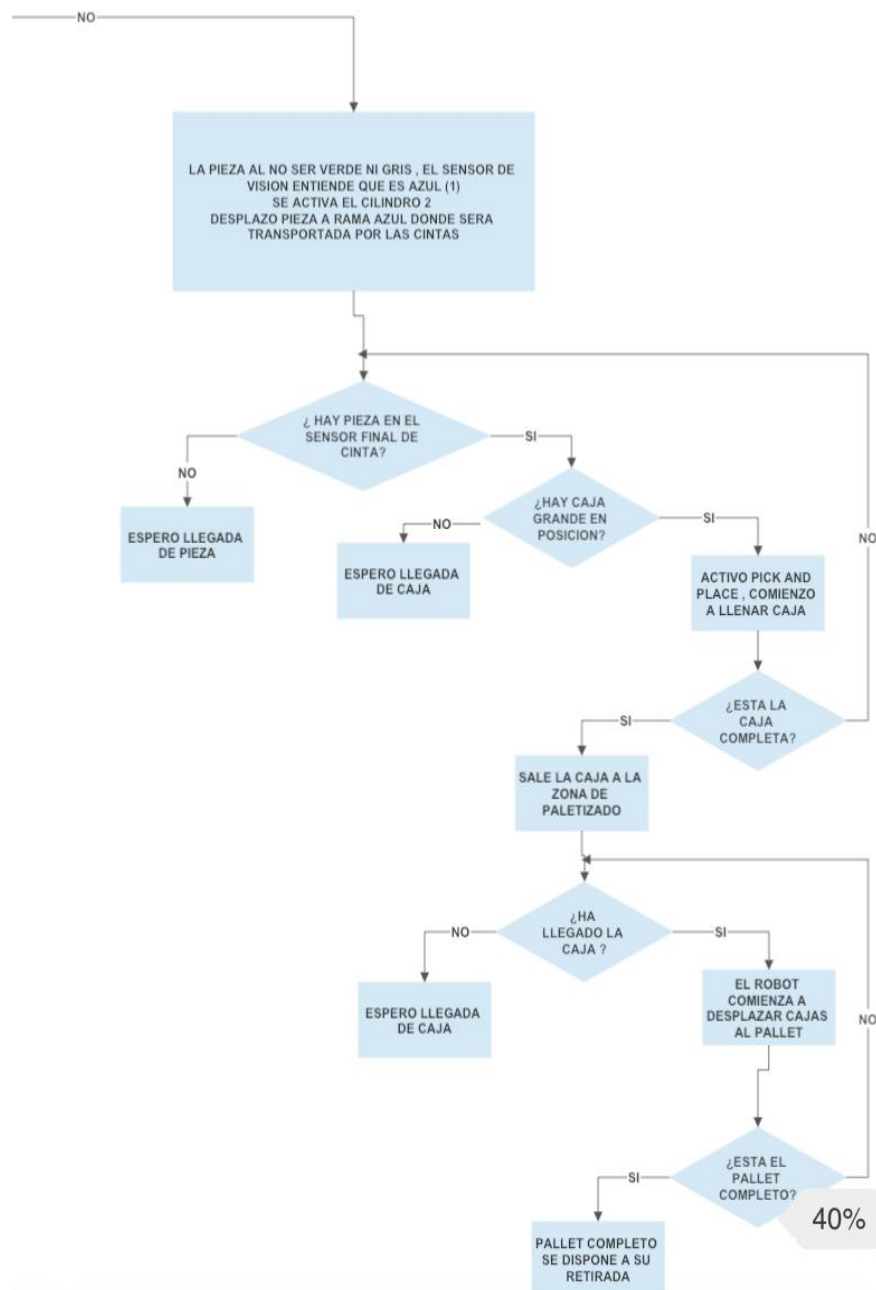
ÍNDICE DOCUMENTO 4:

4.1 FLUJOGRAMAS.....	158
4.2 PROGRAMACIÓN RAPID	161
4.3 TIA PORTAL.....	164

4.1 FLUJOGRAMAS







4.2 PROGRAMACIÓN RAPID

```

MODULE Principal
|*****
|
| ! Módulo: Principal
| !
| ! Descripción:
| !   Modulo principal de programa
| !
| ! Autor: Noelia conde
| !
| ! Versión: 1.0
| !
|*****

|*****
|
| ! Procedimiento Main
| !
| !   Este es el punto de entrada de su programa
| !
|*****
!Declaraciones de posición para caja Grande
CONST robtarget Casa:=[[400,0,727.717782649],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_Rec2:=[[19.998,187.309,-112.062],[0,0,0,1],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_Rec1:=[[259.998,287.309,-62.062],[0,0,0,1],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_Rec:=[[259.998,287.309,-12.062],[0,0,0,1],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_RecSal:=[[39.998,-412.691,-162.062],[0.382683432,0,0,0.923879533],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_dej2:=[[391.465778884,291.477069422,-162.646834995],
[0.707106781,0,-0.024677671,0.706676031],[1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_dej1:=[[241.465778884,241.477069422,-112.646834995],
[0.707106781,0,-0.024677671,0.706676031],[1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_dej:=[[190,190,-62],[0.707106781,0,0,0.707106781],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
!Variables de desplazamiento persistentes para que no se pierda la colocacion con un reinicio
PERS pose DespCaja:=[[0,0,0],[1,0,0,0]];
PERS num Despx;
PERS num Despy;
PERS num Despz;
CONST robtarget p_dej3:=[[541.465746741,341.477091992,-212.6469008],
[0.707106832,0.000000004,-0.024677683,0.706675979],[1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
VAR btnres br_ColocaPallet;
VAR btnres br_RetiraPallet;

PROC main()
  PDispOff;

```

```

!Nos aseguramos que todas las señales estan a OFF
SetDO do_colocarPALLET,0;
SetDO do_enviarPALLET,0;
SetDO do_retirarPALLET,0;
SetDO do_VENT,0;
!Movimiento a posicion segura
MoveJ Casa,v1000,fine,TCP_1\WObj:=wobj0; !Posición segura del robot
progCajaGrande; !Proceso colocacion de cajas
!Fin Proceso
ENDPROC

PROC progCajaGrande()
!Programa que ejecuta la matriz de colocacion
!mediante desplazamientos y llama a los programas
!Prog: Recoger y Dejar
DespCaja:=[[0,0,0],[1,0,0,0]];
Despx:=0;
Despy:=0;
Despz:=0;
!Si no detecta pallet en entrada advierte
!Para la simulación ejecuta la aparicion del pallet
IF di_sensorpalletENT = 0 THEN
  repito_preg_entrada:
    UIMsgBox\Header:="No Hay Pallet en Entrada","¿Desea ColocarIo?"\Buttons:=btnYesNo
    \Icon:=iconInfo\Result:=br_ColocaPallet;
    !Coloca el pallet mediante el SC,
    !en la vida real solo esperaria la confirmacion de que el pallet esta colocado
    !y lo confirmaria con la señal del sensor del pallet de entrada
    IF br_ColocaPallet=6 THEN
      SetDO do_colocarPALLET,1;
      WaitTime 2;
      WaitDI di_sensorpalletENT,1;
    ELSE
      GOTO repito_preg_entrada;
    ENDIF
    SetDO do_colocarPALLET,0;
  ENDIF

  FOR k FROM 0 TO 1 DO
    Despx:=0;
    Despy:=0;
    DespCaja.trans.z:=- (Despz*400);
    Despz:=Despz+1;
    FOR j FROM 0 TO 2 DO
      Despy:=0;
      DespCaja.trans.x:=Despx*400;
      Despx:=Despx+1;
      FOR i FROM 0 TO 1 DO
        DespCaja.trans.y:=Despy*400;
        Despy:=Despy+1;
        Recoger;
      
```

```

        Dejar;
    ENDFOR
ENDFOR
ENDFOR
salida_Pallet;
!Fin Proceso
ENDPROC

PROC Recoger()
    !Programa de recogida de pieza, activa ventosas venturi y comprueba
    !que tiene la caja
    MoveJ p_Rec2,v1500,z200,TCP_1\WObj:=Wobj_cinta;
    MoveL p_Rec1,v400,fine,TCP_1\WObj:=Wobj_cinta;
    WaitDI di_sensorcinta,1;
    MoveL p_Rec,v50,fine,TCP_1\WObj:=Wobj_cinta;
    !! ejecuto el coger la caja
    WaitDI di_sensorHerr,1;
    SetDO\Sync,DO_VENT,1;
    MoveL p_Rec1,v50,z50,TCP_1\WObj:=Wobj_cinta;
    MoveL p_Rec2,v300,z100,TCP_1\WObj:=Wobj_cinta;
    MoveL p_RecSal,v300,z100,TCP_1\WObj:=Wobj_cinta;
    !Fin Proceso
ENDPROC

PROC Dejar()
    PDispSet DespCaja;
    MoveL p_dej3,v400,z200,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej2,v400,z100,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej1,v200,z50,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej,v50,fine,TCP_1\WObj:=Wobj_Pallet;
    setdo\Sync,DO_VENT,0; !! ejecuto el coger la caja
    MoveL p_dej1,v200,z50,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej2,v1000,z100,TCP_1\WObj:=Wobj_Pallet;
    MoveL p_dej3,v1500,z200,TCP_1\WObj:=Wobj_Pallet;
    !Fin Proceso
    PDispOff;
ENDPROC

PROC salida_Pallet()
    MoveJ Casa,v1000,fine,TCP_1\WObj:=wobj0;
    WaitTime 2;
    !Si hay un Pallet en salida hay que retirarlo
    !Esto lo hacemos con un mensaje en pantalla
    !Para la simulacion ejecuta la retirada del pallet
    IF di_sensorpallet5AL = 1 THEN
        repito_preg_salida:
        UIMsgBox\Header:="Pallet en Salida", "{Desea Retirarlo?}"\Buttons:=btnYesNo\Icon:=iconInfo
        \Result:=br_RetiraPallet;
        !Retira el pallet mediante el SC,
        !en la vida real solo esperaria la confirmacion de que el pallet esta retirado
        !y lo confirmaria con la señal del sensor del pallet de salida
    
```

```
IF br_RetiraPallet=6 THEN
    SetDO do_retirarPALLET,1;
    WaitTime 2;
    WaitDI di_sensorpalletSAL,0;
ELSE
    GOTO repito_preg_salida;
ENDIF
SetDO do_retirarPALLET,0;
ENDIF
!Enviamos el pallet a la salida
SetDO do_enviarPALLET,1;
WaitDI di_sensorpalletSAL,1;
SetDO do_enviarPALLET,0;
progCajaGrande;
!Fin de Proceso
ENDPROC

ENDMODULE
```

4.3 TIA PORTAL

Totally Integrated Automation Portal

Bloques de programa

Main [OB1]

Main Propiedades								
General								
Nombre	Main	Número	1	Tipo	OB	Idioma	KOP	
Numeración	Automático							
Información								
Título	"Main Program Sweep (Cycle)"	Autor		Comentario	PROGRAMACION DE UNA LINEA DE CLASIFICACION , ALMACENAMIENTO Y PALETIZADO DE PIEZAS POR COLOR.		Familia	
Versión	0.1	ID personalizada						

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
▼ Input				
Initial_Call	Bool			Initial call of this OB
Remanence	Bool			=True, if remanent data are available
Temp				
Constant				

Segmento 1: Alwaystrue&>false

Segmento 2: CONEXION TIA PORTAL CON FZCTORI IO

Codigo necesario para poder simular el codigo desarrollado en Tia Portal con Factory IO

Segmento 3: SISTEMA

Segmento 4: SENSOR DE VISION

Segmento 5: EMISOR Y DERIVADOR

Segmento 6: RAMIFICACION AZUL

Segmento 7: RAMIFICACION GRIS



Segmento 9: PANTALLA HMI

%FC1900

"HMI"

EN

ENO

Segmento 10: EMERGENCIAS

%FC1000

"EMERGENCIAS"

EN

ENO

Totally Integrated Automation Portal

Bloques de programa / Bloques de sistema / Recursos de programa

IEC_Timer_0_DB_3 [DB24]

IEC_Timer_0_DB_3 Propiedades

General

Nombre	IEC_Timer_0_DB_3	Número	24	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor	Simatic	Comentario		Familia	IEC
Versión	1.0	ID personalizada	IEC_TMR				

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Totally Integrated Automation Portal

Bloques de programa / Bloques de sistema / Recursos de programa

IEC_Timer_0_DB_4 [DB25]

IEC_Timer_0_DB_4 Propiedades

General

Nombre	IEC_Timer_0_DB_4	Número	25	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor	Simatic	Comentario		Familia	IEC
Versión	1.0	ID personalizada	IEC_TMR				

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Bloques de programa / 00_SISTEMA

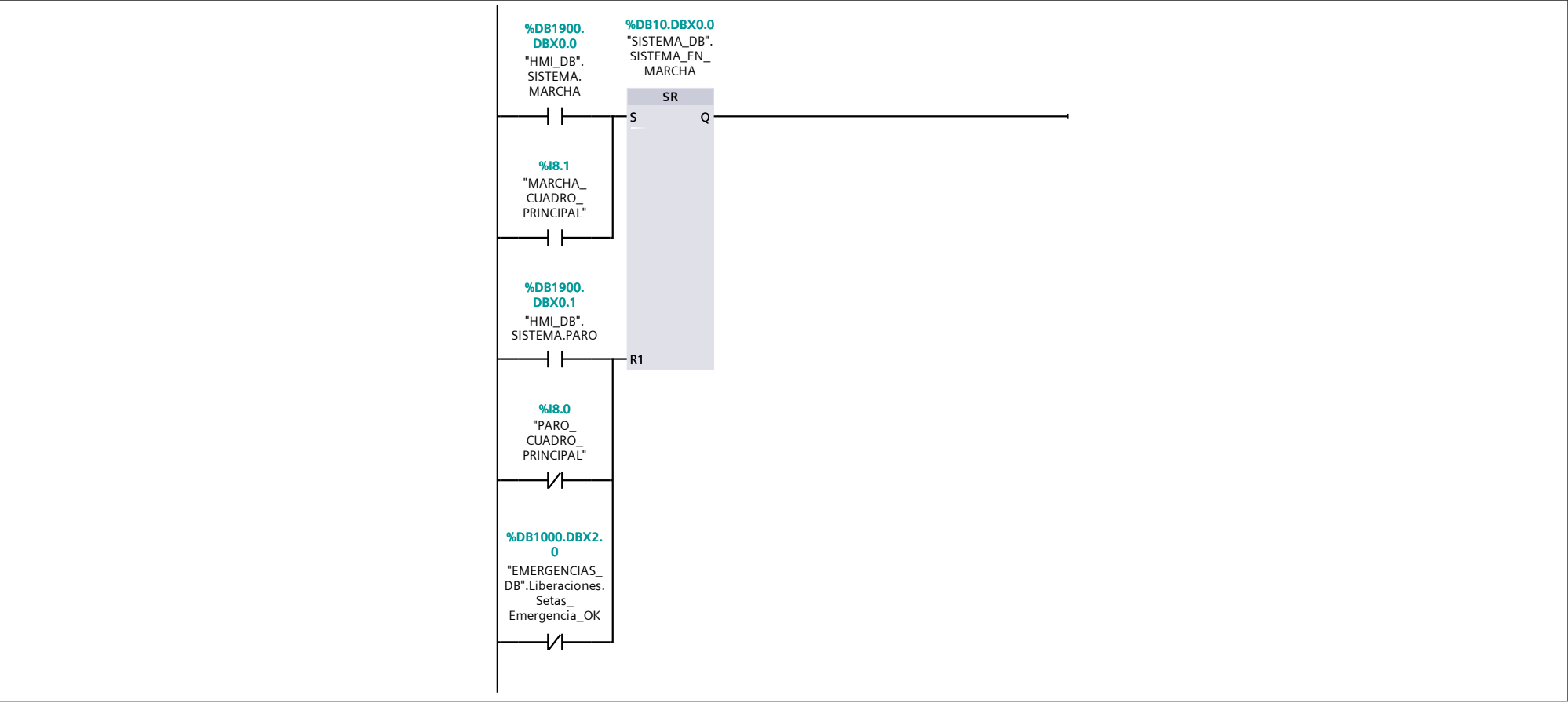
SISTEMA [FC10]

SISTEMA Propiedades							
General							
Nombre	SISTEMA	Número	10	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
SISTEMA	Void			

Segmento 1: INICIO SISTEMA

USO DE BLOQUE SET/RESET PARA ARRANQUE DE LINEA



Segmento 2: SISTEMA PARADO

NOS INDICA SI EL SISTEMA NO ESTA EN MARCHA, ES DECIR, QUE ESTA PARADO



Totally Integrated Automation Portal

Bloques de programa / 00_SISTEMA

SISTEMA_DB [DB10]

SISTEMA_DB Propiedades

General

Nombre	SISTEMA_DB	Número	10	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
SISTEMA_EN_MARCHA	Bool	0.0	false	False	True	True	True	False		
SISTEMA_PARADO	Bool	0.1	false	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 11_EMERGENCIAS

EMERGENCIAS_DB [DB1000]

EMERGENCIAS_DB Propiedades

General

Nombre	EMERGENCIAS_DB	Número	1000	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
▼ Setas_emergencia	Struct	0.0		False	True	True	True	False		
Seta_cuadro_principal	Bool	0.0	false	False	True	True	True	False		=1 FALLO EN SETA PRINCIPAL // =0 SIN FALLO
seta_zona1	Bool	0.1	false	False	True	True	True	False		=1 FALLO EN SETA PRINCIPAL // =0 SIN FALLO
seta_zona2	Bool	0.2	false	False	True	True	True	False		=1 FALLO EN SETA PRINCIPAL // =0 SIN FALLO
seta_zona3	Bool	0.3	false	False	True	True	True	False		=1 FALLO EN SETA PRINCIPAL // =0 SIN FALLO
▼ Liberaciones	Struct	2.0		False	True	True	True	False		
Setas_Emergencia_OK	Bool	2.0	false	False	True	True	True	False		resumen OK de todos los fallos
Rearme_emergencias	Bool	4.0	false	False	True	True	True	False		

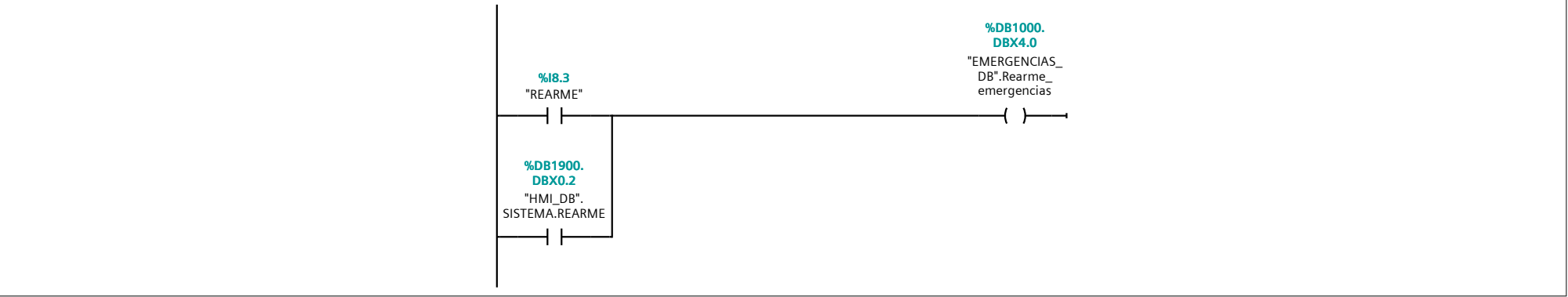
EMERGENCIAS [FC1000]

EMERGENCIAS Propiedades							
General							
Nombre	EMERGENCIAS	Número	1000	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					
Nombre		Tipo de datos	Valor predet.	Supervisión	Comentario		
Input							
Output							
InOut							
Temp							
Constant							
▼ Return							
EMERGENCIAS		Void					

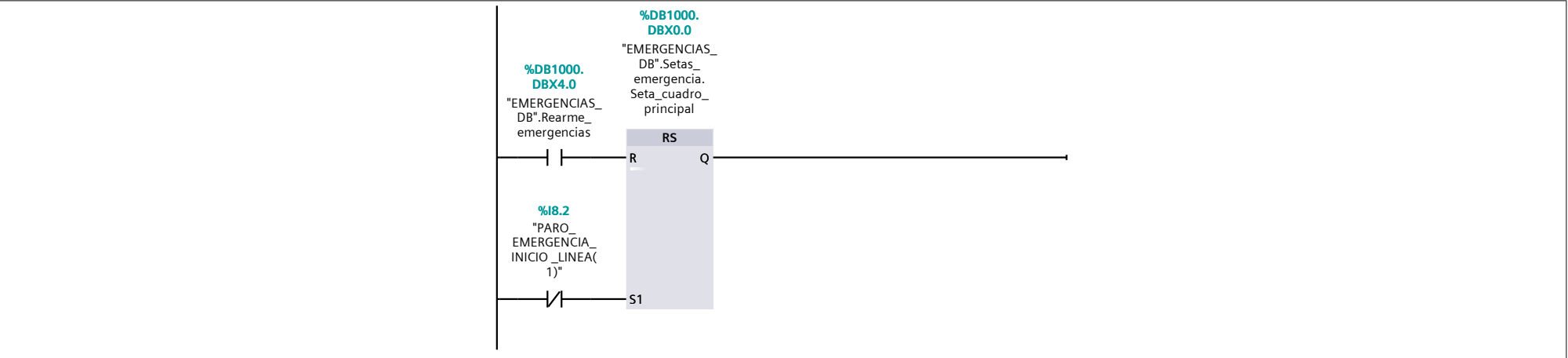
Segmento 1: SETAS DE EMERGENCIA

0001

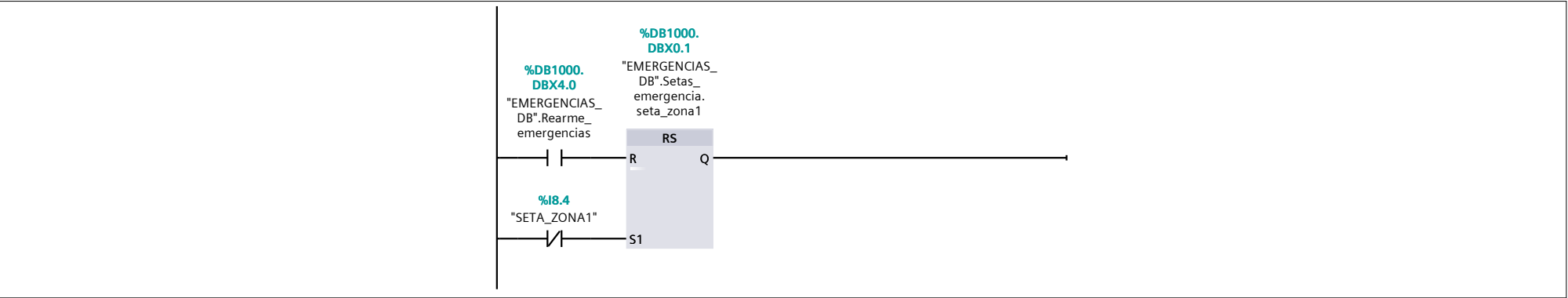
Segmento 2: REARME_EMERGENCIAS



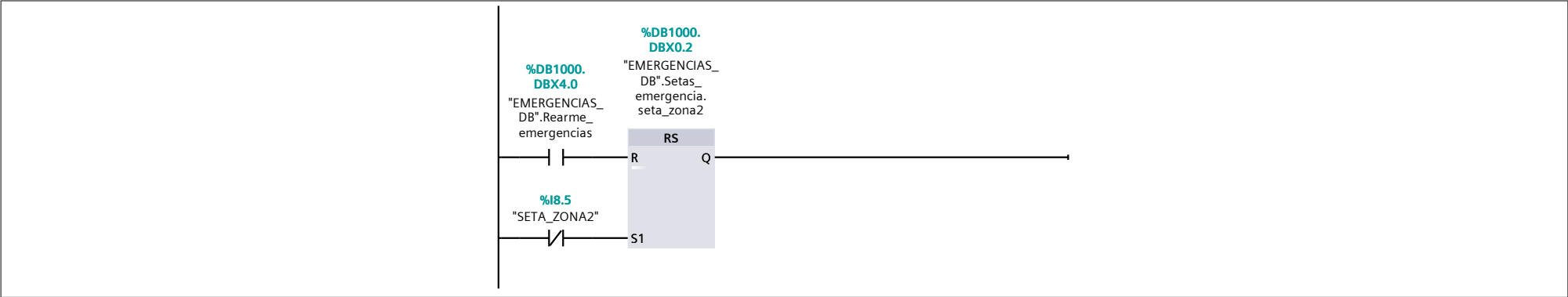
Segmento 3: SETA EMERGENCIA: CUADRO PRINCIPAL



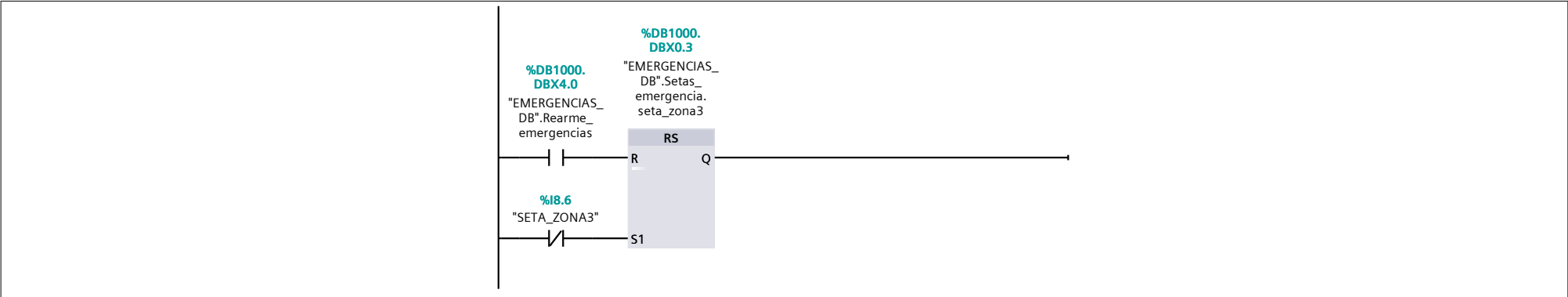
Segmento 4: SETA EMERGENCIA: ZONA 1



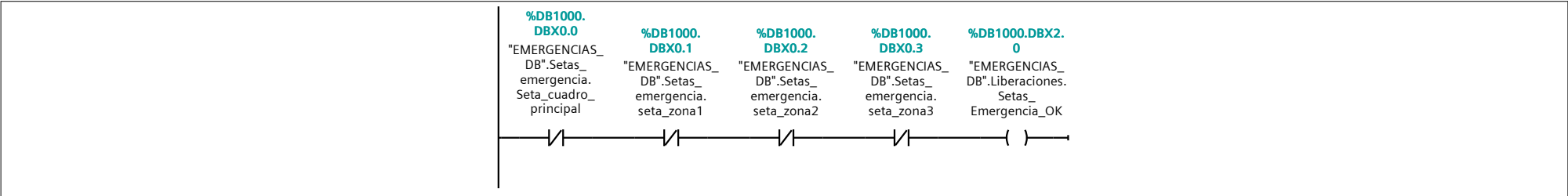
Segmento 5: SETA EMERGENCIA: ZONA 2



Segmento 6: Seta Emergencia: ZONA 3



Segmento 8: LIBERACIONES SISTEMA



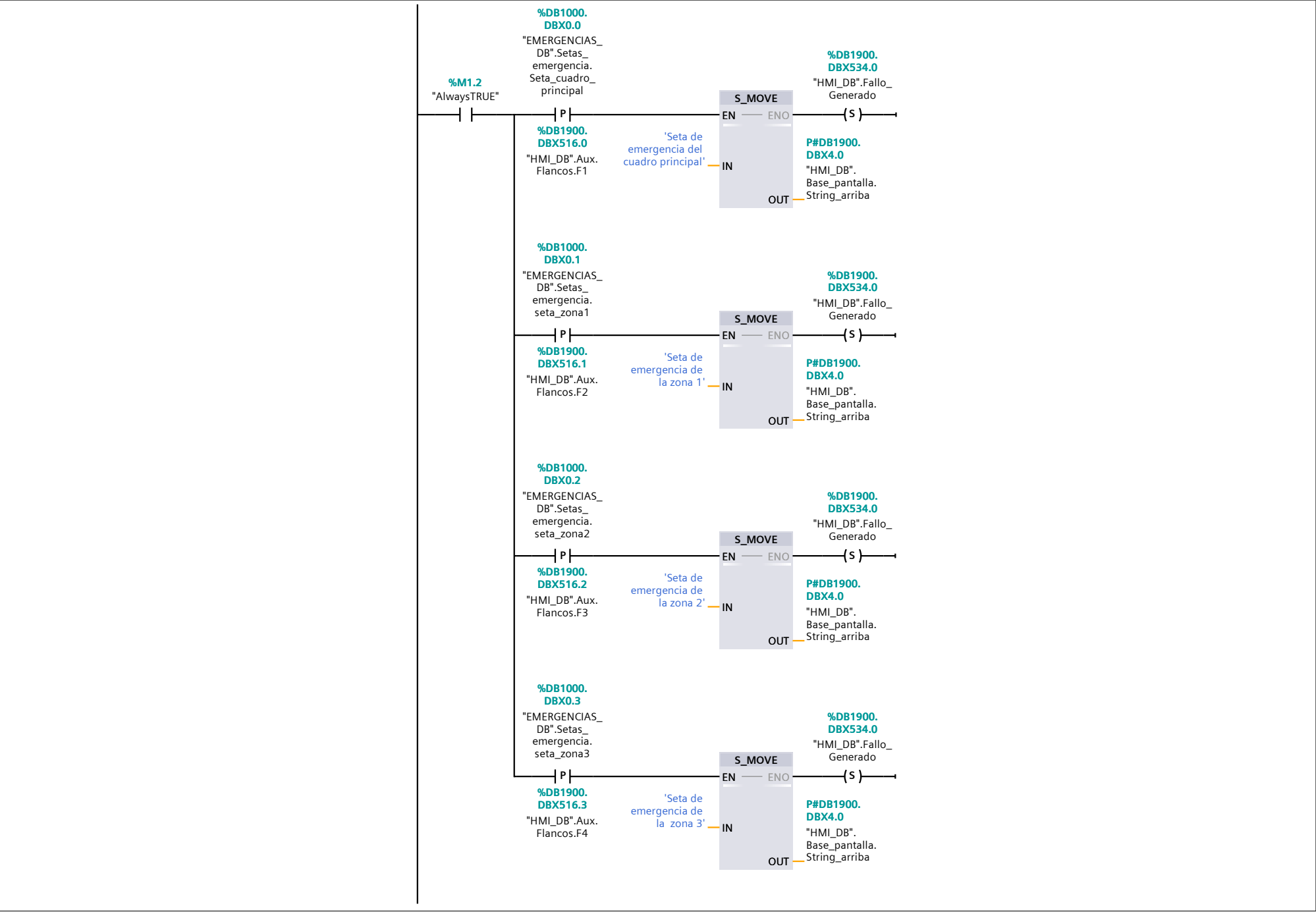
Bloques de programa / 10_HMI

HMI [FC1900]

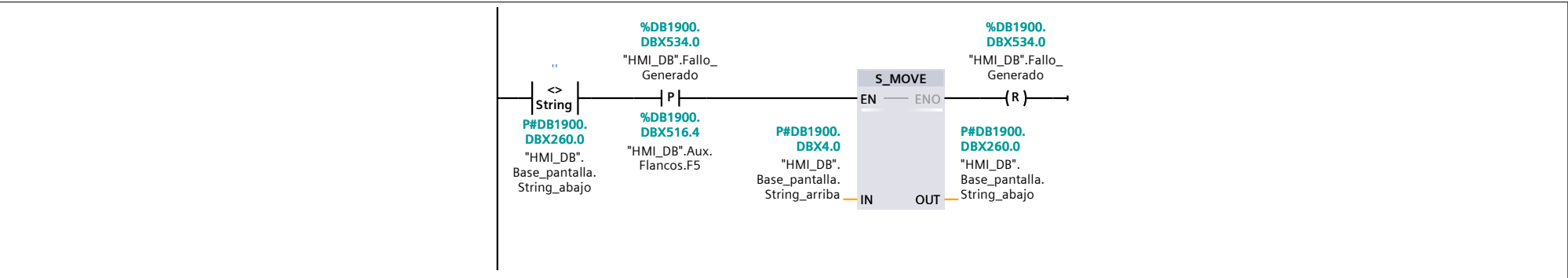
HMI Propiedades							
General							
Nombre	HMI	Número	1900	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
HMI	Void			

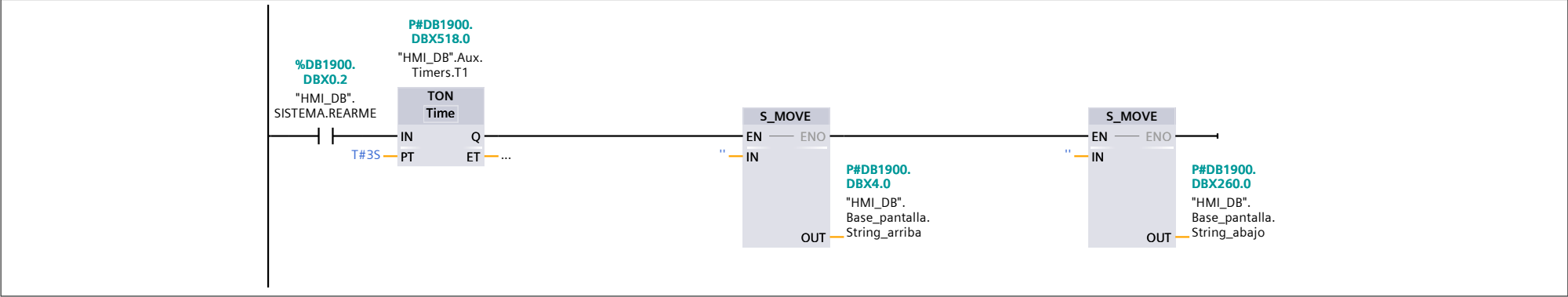
Segmento 2: M uestreo Fallos en String de arriba de HMI



Segmento 3: Traspaso fallo de String de arriba a abajo de HMI



Segmento 4: Borrado strings 3 segundos rearme



Segmento 5: PIEZA EN SENSOR



Totally Integrated Automation Portal

Bloques de programa / 10_HMI

HMI_DB [DB1900]

HMI_DB Propiedades

General

Nombre	HMI_DB	Número	1900	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
▼ SISTEMA	Struct	0.0		False	True	True	True	False		
MARCHA	Bool	0.0	false	False	True	True	True	False		
PARO	Bool	0.1	false	False	True	True	True	False		
REARME	Bool	0.2	false	False	True	True	True	False		
▼ Fallos	Struct	2.0		False	True	True	True	False		
Seta_Emergencia_Principal	Bool	2.0	false	False	True	True	True	False		
Seta_emergencia_zona_1	Bool	2.1	false	False	True	True	True	False		
Seta_emergencia_zona_2	Bool	2.2	false	False	True	True	True	False		
Seta_emergencia_zona_3	Bool	2.3	false	False	True	True	True	False		
▼ Base_pantalla	Struct	4.0		False	True	True	True	False		
String_arriba	String	4.0	"	False	True	True	True	False		
String_abajo	String	260.0	"	False	True	True	True	False		
▼ Aux	Struct	516.0		False	True	True	True	False		
▼ Flancos	Struct	516.0		False	True	True	True	False		
F1	Bool	516.0	false	False	True	True	True	False		
F2	Bool	516.1	false	False	True	True	True	False		
F3	Bool	516.2	false	False	True	True	True	False		
F4	Bool	516.3	false	False	True	True	True	False		
F5	Bool	516.4	false	False	True	True	True	False		
F6	Bool	516.5	false	False	True	True	True	False		
▼ Timers	Struct	518.0		False	True	True	True	False		
▼ T1	IEC_TIMER	518.0		False	True	True	True	False		
PT	Time	522.0	T#0ms	False	True	True	True	False		
ET	Time	526.0	T#0ms	False	True	False	True	False		
IN	Bool	530.1	false	False	True	True	True	False		
Q	Bool	530.2	false	False	True	False	True	False		
Fallo_Generado	Bool	534.0	false	False	True	True	True	False		
Pieza_sensor_vision	Bool	534.1	false	False	True	True	True	False		

Bloques de programa / 01_SENSOR_VISION

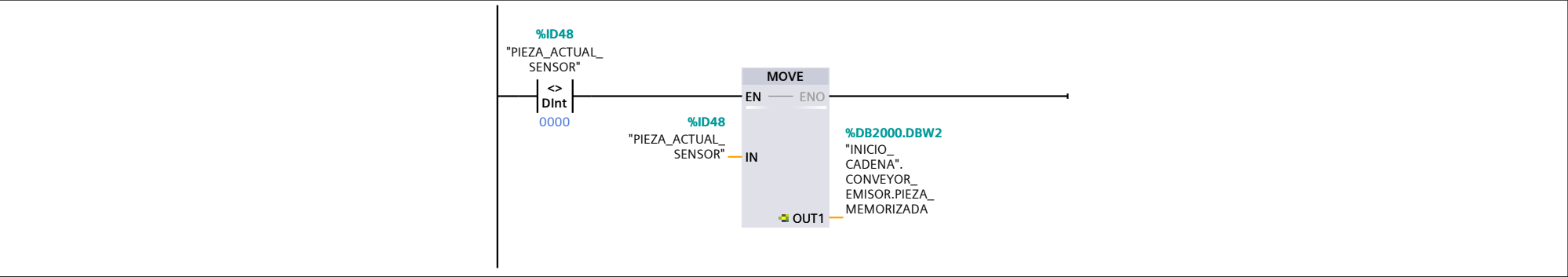
SENSOR_VISION [FC50]

SENSOR_VISION Propiedades							
General							
Nombre	SENSOR_VISION	Número	50	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
SENSOR_VISION	Void			

Segmento 1: SENSOR DE VISION

CON ESTE BLOQUE EL SENSOR DE VISION NOS RECONOCE LOS DIFERENTES COLORES DE PIEZAS Y QUEDA MEMORIZADO PARA SU POSTERIOR USO

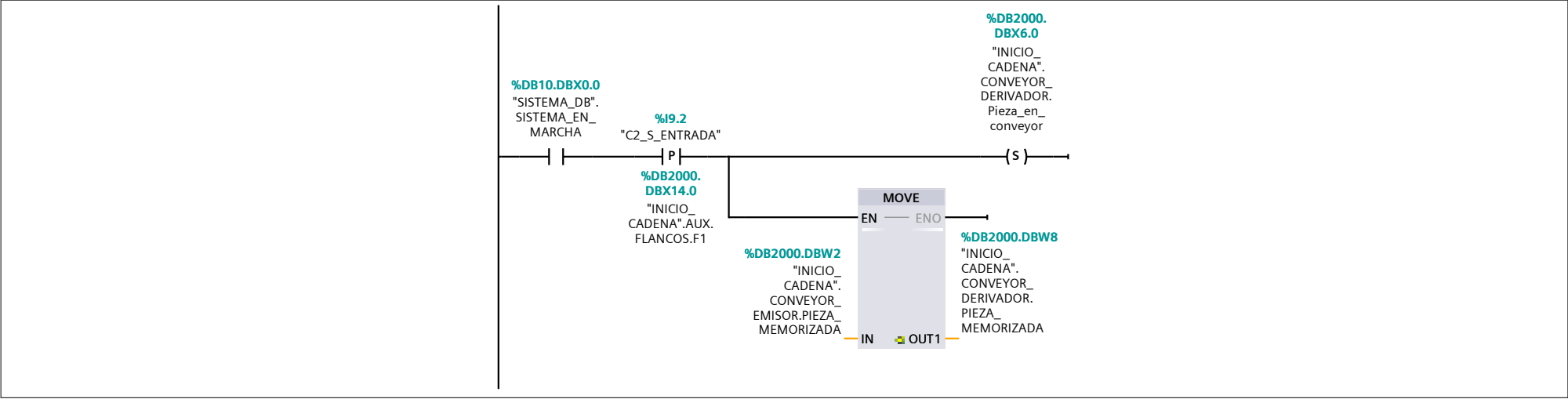


Bloques de programa / 02_INICIO_CADENA_PRODUCION

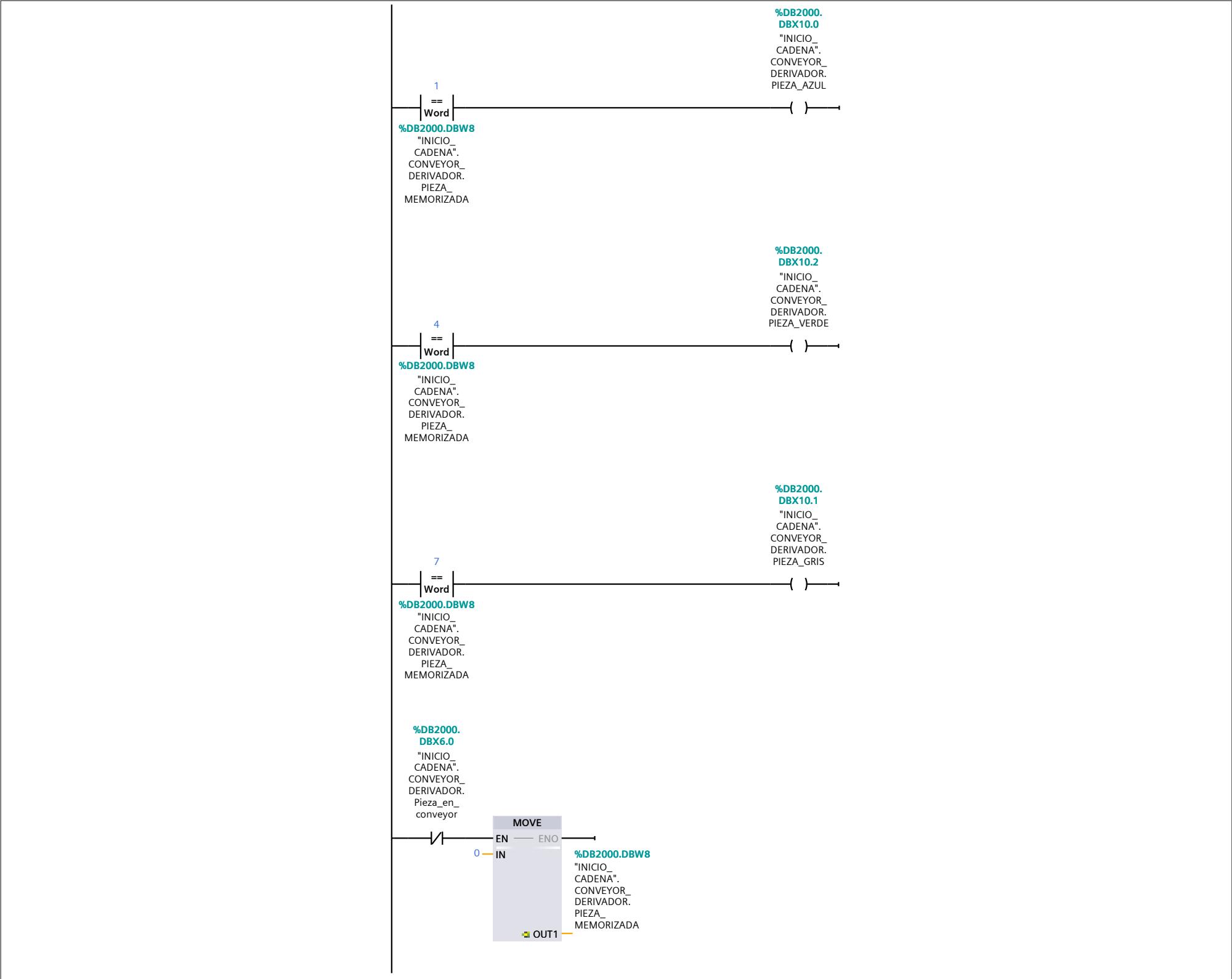
CONVEYOR_DERIVADOR [FC2001]

CONVEYOR_DERIVADOR Propiedades							
General							
Nombre	CONVEYOR_DERIVADOR	Número	2001	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					
Nombre		Tipo de datos	Valor predet.	Supervisión	Comentario		
Input							
Output							
InOut							
Temp							
Constant							
▼ Return							
CONVEYOR_DERIVADOR		Void					

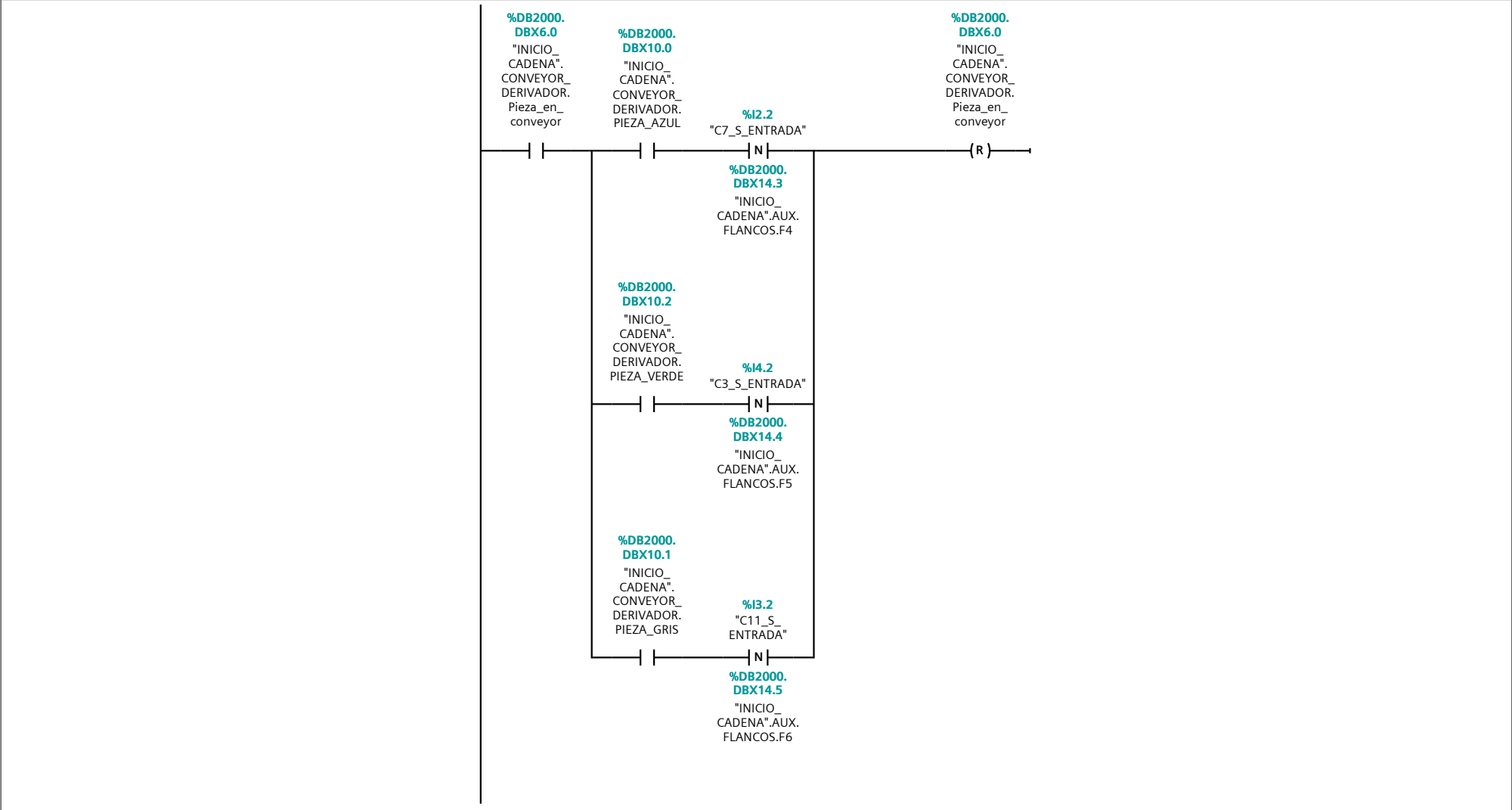
Segmento 1: PIEZA EN CONVEYOR & TRANSFERENCIA DE TIPO DE PIEZA



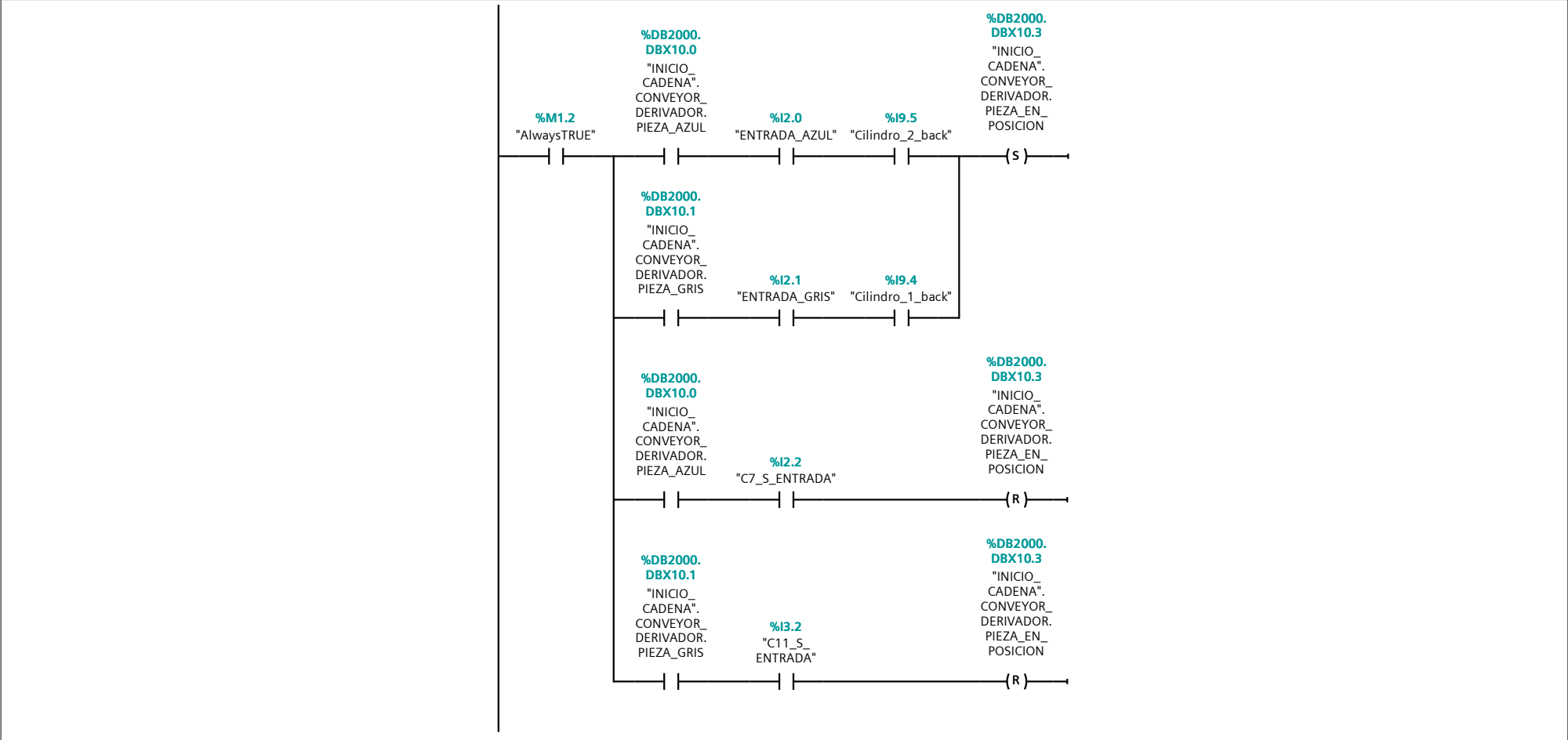
Segmento 2: DEFINICION DE PIEZA



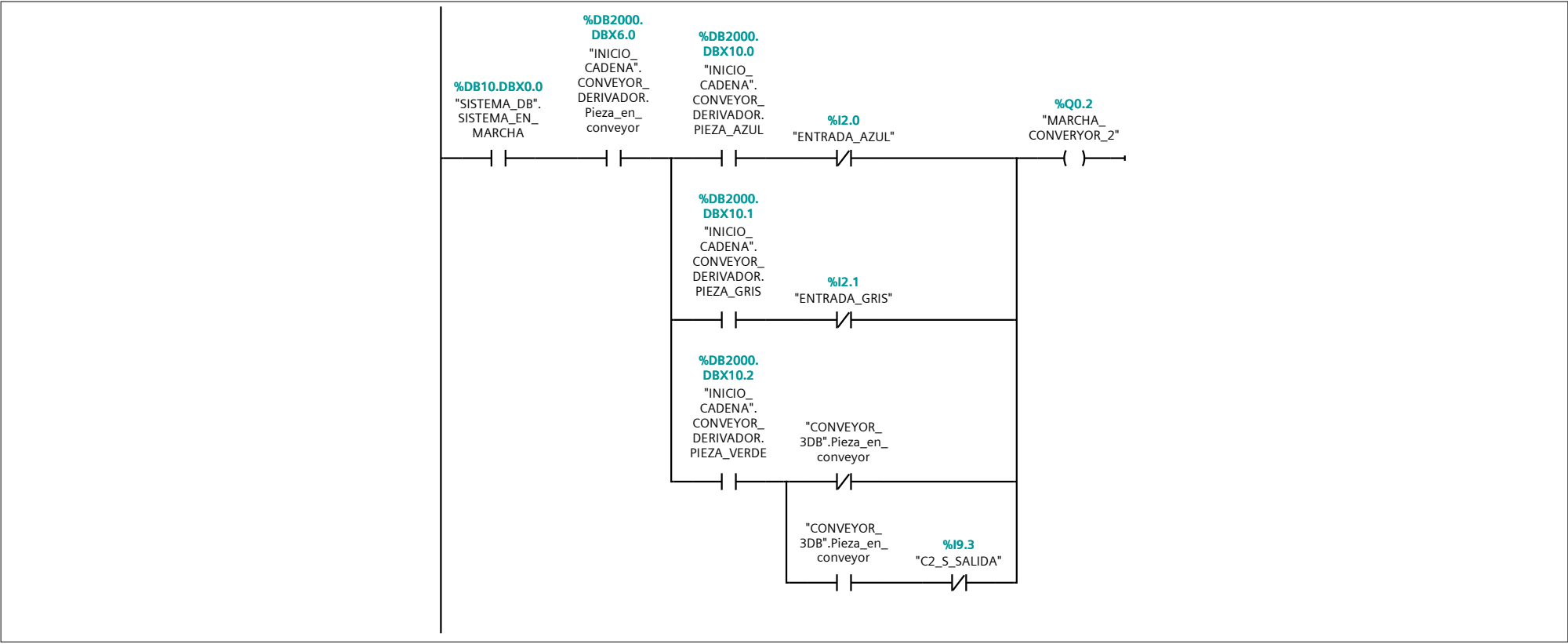
Segmento 3: RESET PIEZA CONVEYOR



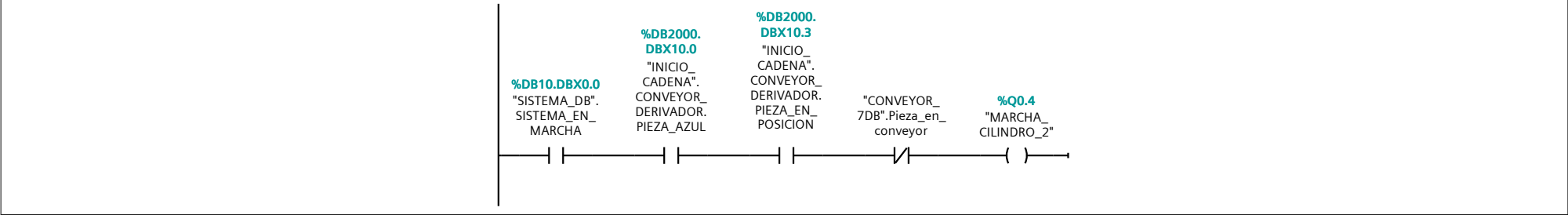
Segmento 4: PIEZA EN POSICION



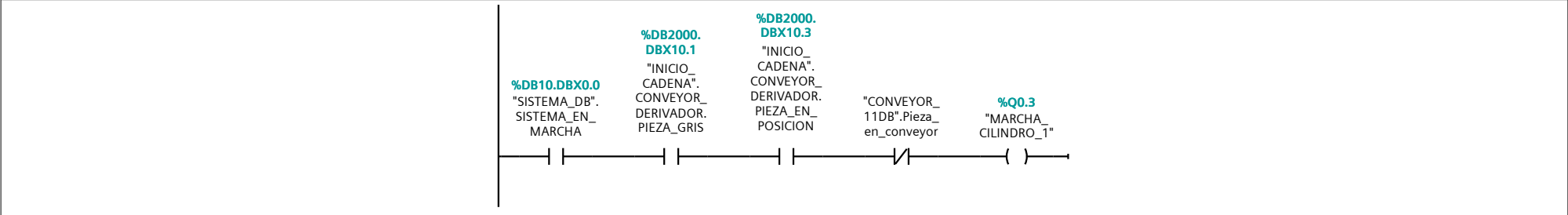
Segmento 5: MARCHA CONVEYOR



Segmento 6: CILINDRO 2 - RAMA PIEZA AZUL

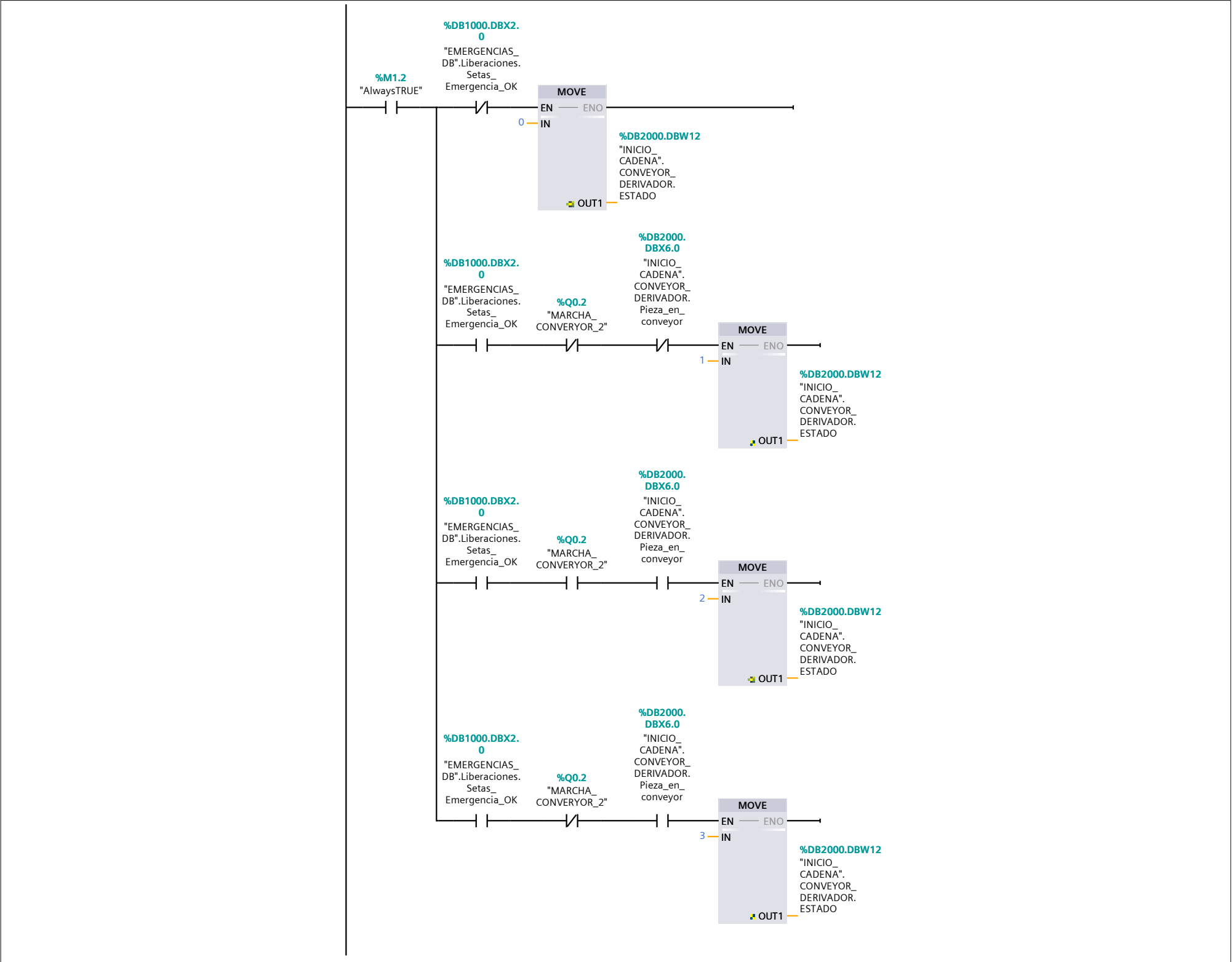


Segmento 7: CILINDRO 1 - RAMA PIEZA GRIS



Segmento 8: ESTADOS

- =0 Fallo
- =1 OK sin pieza
- =2 En marcha con pieza
- =3 Esperando con pieza



Totally Integrated Automation Portal

Bloques de programa / 02_INICIO_CADENA_PRODUCCION

INICIO_CADENA [DB2000]

INICIO_CADENA Propiedades

General

Nombre	INICIO_CADENA	Número	2000	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Es-cribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
▼ CONVEYOR_EMITOR	Struct	0.0		False	True	True	True	False		
Pieza_en_conveyor	Bool	0.0	false	False	True	True	True	False		
PIEZA_MEMORIZADA	Word	2.0	16#0	False	True	True	True	False		
ESTADO	Int	4.0	0	False	True	True	True	False		
▼ CONVEYOR_DERIVADOR	Struct	6.0		False	True	True	True	False		
Pieza_en_conveyor	Bool	6.0	false	False	True	True	True	False		
PIEZA_MEMORIZADA	Word	8.0	16#0	False	True	True	True	False		
PIEZA_AZUL	Bool	10.0	false	False	True	True	True	False		
PIEZA_GRIS	Bool	10.1	false	False	True	True	True	False		
PIEZA_VERDE	Bool	10.2	false	False	True	True	True	False		
PIEZA_EN_POSICION	Bool	10.3	false	False	True	True	True	False		
ESTADO	Int	12.0	0	False	True	True	True	False		
▼ AUX	Struct	14.0		False	True	True	True	False		
▼ FLANCOS	Struct	14.0		False	True	True	True	False		
F1	Bool	14.0	false	False	True	True	True	False		
F2	Bool	14.1	false	False	True	True	True	False		
F3	Bool	14.2	false	False	True	True	True	False		
F4	Bool	14.3	false	False	True	True	True	False		
F5	Bool	14.4	false	False	True	True	True	False		
F6	Bool	14.5	false	False	True	True	True	False		
F7	Bool	14.6	false	False	True	True	True	False		
▼ TIMERS	Struct	16.0		False	True	True	True	False		
▼ TIMER1	IEC_TIMER	16.0		False	True	True	True	False		
PT	Time	20.0	T#0ms	False	True	True	True	False		
ET	Time	24.0	T#0ms	False	True	False	True	False		
IN	Bool	28.1	false	False	True	True	True	False		
Q	Bool	28.2	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER	32.0		False	True	True	True	False		
PT	Time	36.0	T#0ms	False	True	True	True	False		
ET	Time	40.0	T#0ms	False	True	False	True	False		
IN	Bool	44.1	false	False	True	True	True	False		
Q	Bool	44.2	false	False	True	False	True	False		

Bloques de programa / 02_INICIO_CADENA_PRODUCCION

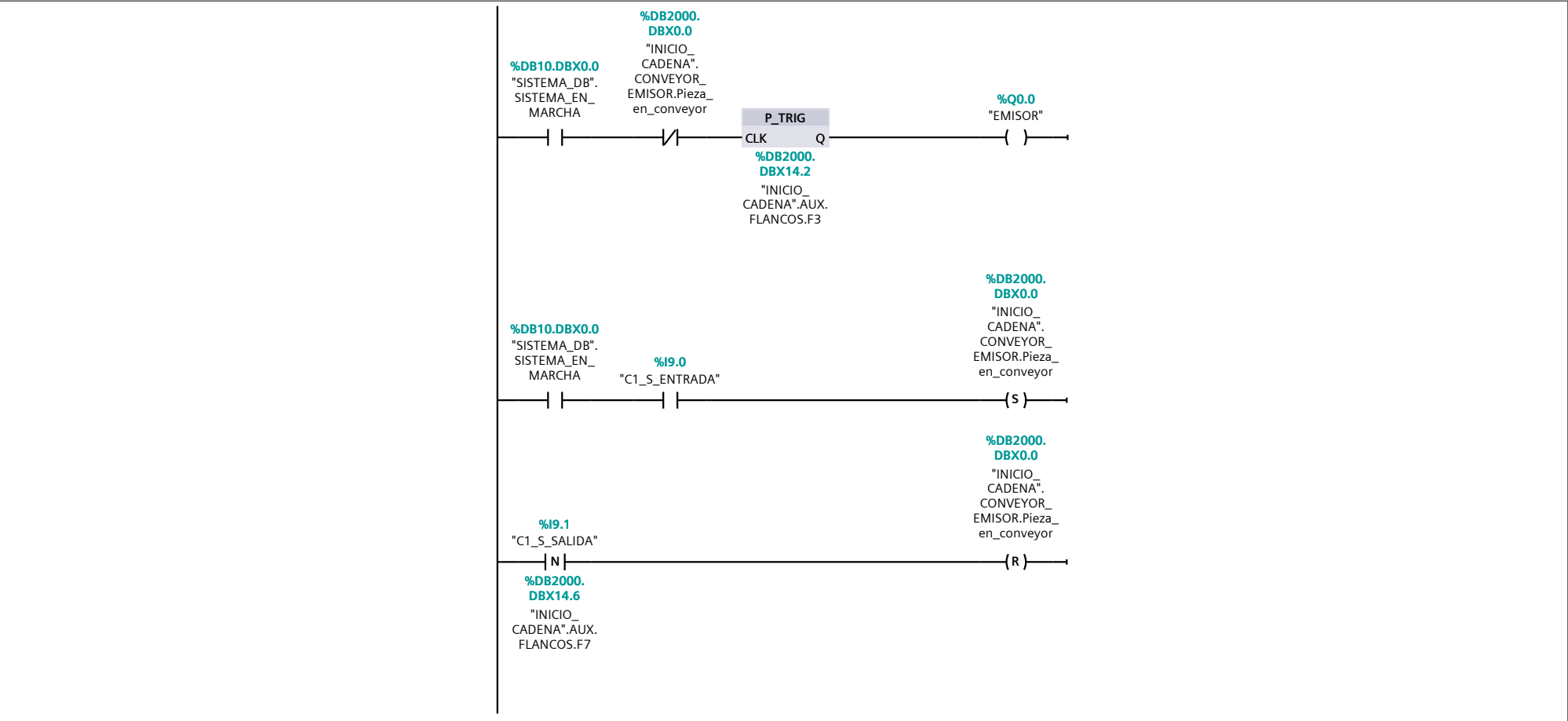
CONVEYOR_EMITOR [FC2000]

CONVEYOR_EMITOR Propiedades							
General							
Nombre	CONVEYOR_EMITOR	Número	2000	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

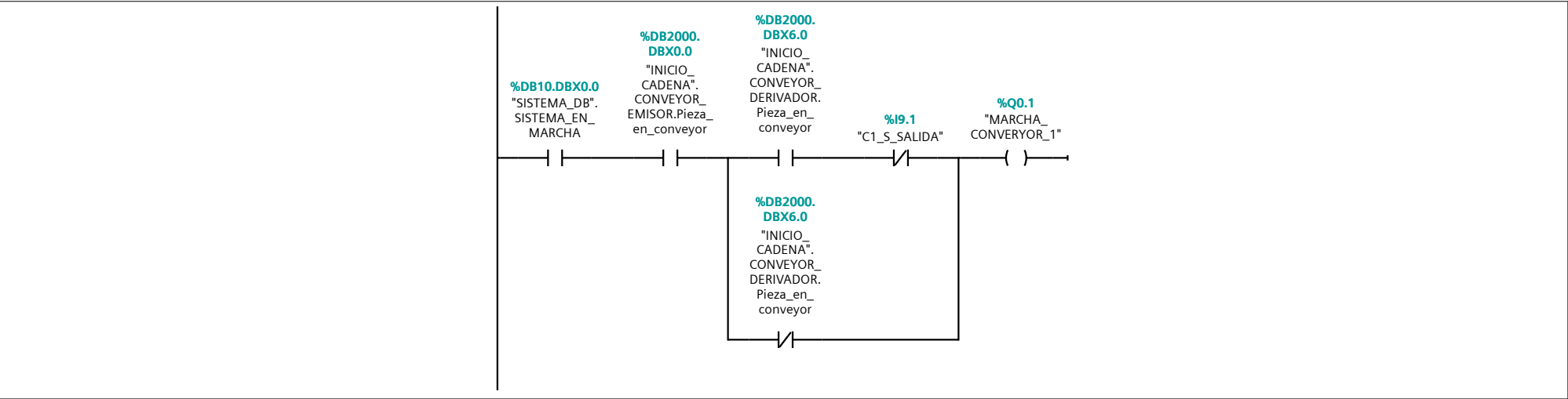
Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
CONVEYOR_EMITOR	Void			

Segmento 1: Sacar pieza y contabilizarla

Se programa el emisor, el cual nos añade una pieza a la linea.
Defininos pieza en conveyor

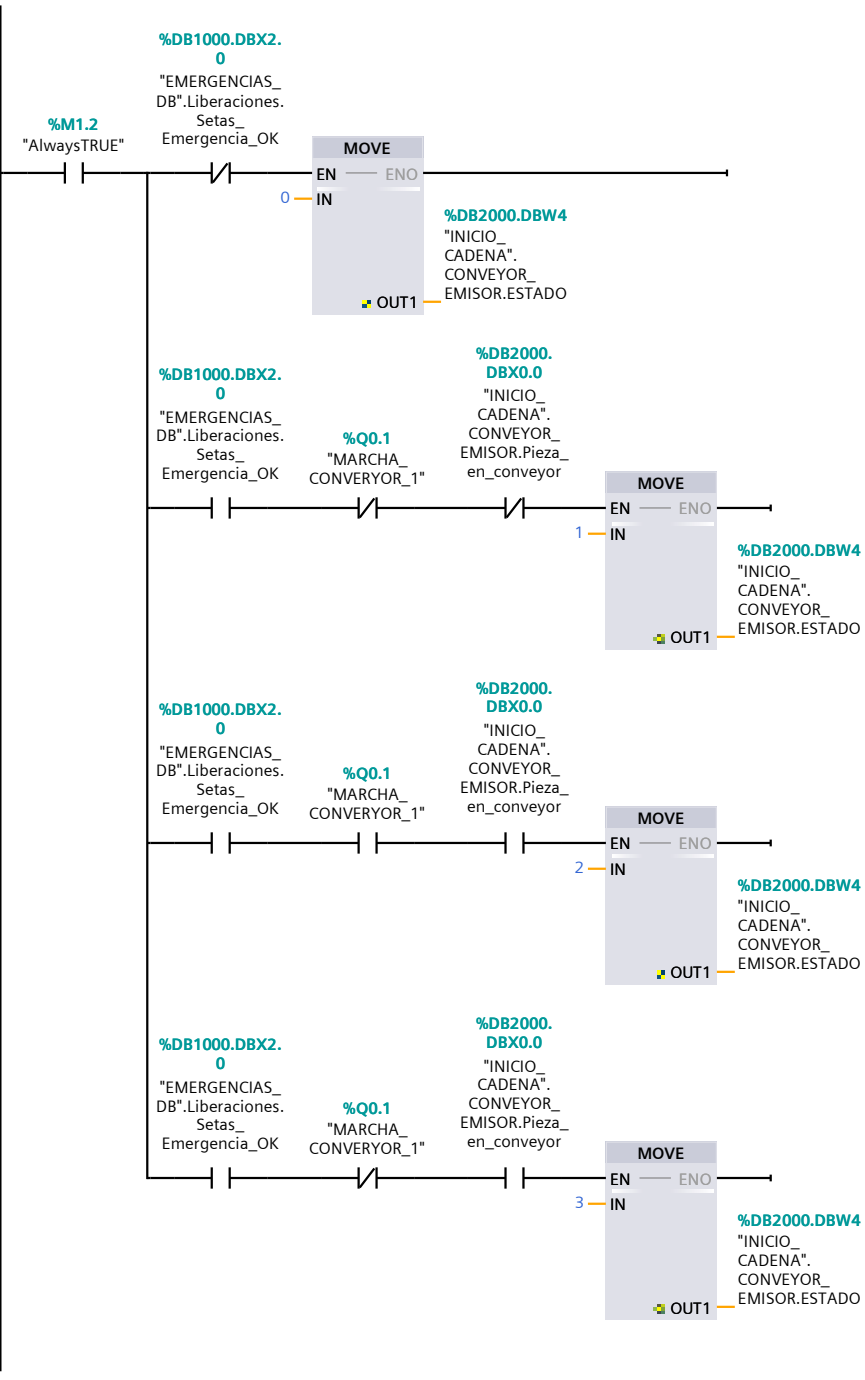


Segmento 2: Marcha cinta



Segmento 3: ESTADOS

- =0 Fallo
- =1 OK sin pieza
- =2 En marcha con pieza
- =3 Esperando con pieza



Totally Integrated Automation Portal

Bloques de programa / 12_AUX

MHJ-PLC-Lab-Function-S71200 [FC9000]

MHJ-PLC-Lab-Function-S71200 Propiedades

General

Nombre	MHJ-PLC-Lab-Function-S71200	Número	9000	Tipo	FC	Idioma	SCL
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
▼ Temp				
rdTimeReturn	Int			
▼ outputTime	DTL			
YEAR	UInt			
MONTH	USInt			
DAY	USInt			
WEEKDAY	USInt			
HOURL	USInt			
MINUTE	USInt			
SECOND	USInt			
NANOSECOND	UDInt			
SyncVal	Byte			
forVal	Int			
forVal_2	Int			
Value	Byte			
▼ Constant				
CompVal	Byte	16#34		
Value_01	Byte	16#11		
Value_01_DW	DWord	16#A165_D992		
Value_02_DW	DWord	16#58BE_4401		
▼ Return				
MHJ-PLC-Lab-Function-S71200	Void			

```
0001
0002 #Value:=PEEK(area := 16#82,
0003     dbNumber := 0,
0004     byteOffset := 511);
0005 #Value := #Value + 1;
0006
0007 POKE(area := 16#82,
0008     dbNumber := 0,
0009     byteOffset := 511,
0010     value := #Value);
0011
0012 POKE(area:=16#81,
0013     dbNumber:=0,
0014     byteOffset:=1016,
0015     value:=#Value_01_DW);
0016 POKE(area := 16#81,
0017     dbNumber := 0,
0018     byteOffset := 1020,
0019     value := #Value_02_DW);
0020
0021 POKE(area := 16#81,
0022     dbNumber := 0,
0023     byteOffset := 511,
0024     value := B#16#00);
0025
0026 FOR #forVal := 0 TO 120 DO
0027     FOR #forVal_2:=0 TO 10 DO
0028         #rdTimeReturn:=RD_SYS_T(#outputTime);
0029         #rdTimeReturn := WR_SYS_T(#outputTime);
0030         #rdTimeReturn := RD_SYS_T(#outputTime);
0031         #rdTimeReturn := WR_SYS_T(#outputTime);
0032     END_FOR;
0033     #SyncVal:= PEEK(area := 16#81,
0034         dbNumber := 0,
0035         byteOffset := 511);
0036     IF #SyncVal = #CompVal THEN
0037         GOTO M_1;
0038     END_IF;
0039 END_FOR;
0040 RETURN;
0041
0042 M_1:
0043 POKE(area := 16#81,
```

Totally Integrated Automation Portal																																										
0044 dbNumber := 0, 0045 byteOffset := 511, 0046 value := B#16#0); 0047 0048 0049																																										
<table><tr><th>Símbolo</th><th>Dirección</th><th>Tipo</th><th>Comentario</th></tr><tr><td>#CompVal</td><td>16#34</td><td>Byte</td><td></td></tr><tr><td>#forVal</td><td></td><td>Int</td><td></td></tr><tr><td>#forVal_2</td><td></td><td>Int</td><td></td></tr><tr><td>#outputTime</td><td></td><td>DTL</td><td></td></tr><tr><td>#rdTimeReturn</td><td></td><td>Int</td><td></td></tr><tr><td>#SyncVal</td><td></td><td>Byte</td><td></td></tr><tr><td>#Value</td><td></td><td>Byte</td><td></td></tr><tr><td>#Value_01_DW</td><td>16#A165_D992</td><td>DWord</td><td></td></tr><tr><td>#Value_02_DW</td><td>16#58BE_4401</td><td>DWord</td><td></td></tr></table>			Símbolo	Dirección	Tipo	Comentario	#CompVal	16#34	Byte		#forVal		Int		#forVal_2		Int		#outputTime		DTL		#rdTimeReturn		Int		#SyncVal		Byte		#Value		Byte		#Value_01_DW	16#A165_D992	DWord		#Value_02_DW	16#58BE_4401	DWord	
Símbolo	Dirección	Tipo	Comentario																																							
#CompVal	16#34	Byte																																								
#forVal		Int																																								
#forVal_2		Int																																								
#outputTime		DTL																																								
#rdTimeReturn		Int																																								
#SyncVal		Byte																																								
#Value		Byte																																								
#Value_01_DW	16#A165_D992	DWord																																								
#Value_02_DW	16#58BE_4401	DWord																																								

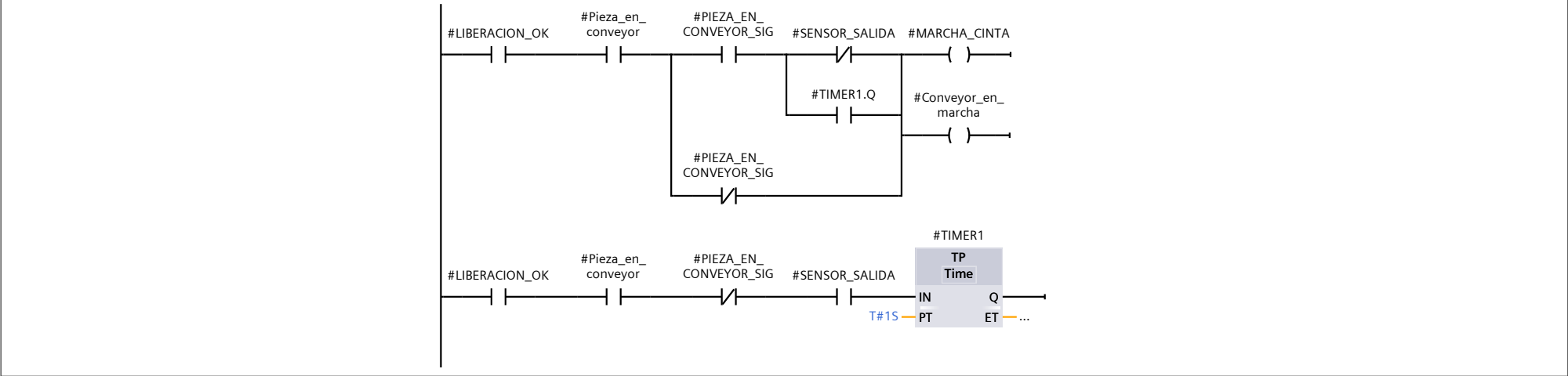
Bloques de programa / 12_AUX

CONVEYOR [FB1]

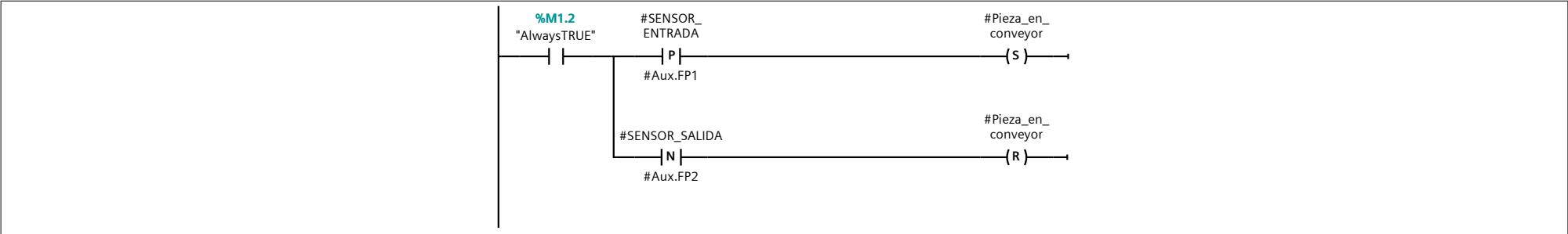
CONVEYOR Propiedades							
General							
Nombre	CONVEYOR	Número	1	Tipo	FB	Idioma	KOP
Numeración	Automático						
Información							
Título	FC CONVEYOR GENERICO	Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible desde HMI/OPC UA	Es-cribi-ble desd e HMI/ OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	No remanente	False	False	False	False		
SENSOR_ENTRADA	Bool	false	No remanente	False	False	False	False		
SENSOR_SALIDA	Bool	false	No remanente	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	No remanente	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	No remanente	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	No remanente	True	True	True	False		
Pieza_en_conveyor	Bool	false	No remanente	True	True	True	False		
▼ Aux	Struct		No remanente	True	True	True	False		
FP1	Bool	false	No remanente	True	True	True	False		
FP2	Bool	false	No remanente	True	True	True	False		
FP3	Bool	false	No remanente	True	True	True	False		
▼ TIMER1	IEC_TIMER		No remanente	True	True	True	False		
PT	Time	T#0ms	No remanente	True	True	True	False		
ET	Time	T#0ms	No remanente	True	False	True	False		
IN	Bool	false	No remanente	True	True	True	False		
Q	Bool	false	No remanente	True	False	True	False		
▼ TIMER2	IEC_TIMER		No remanente	True	True	True	False		
PT	Time	T#0ms	No remanente	True	True	True	False		
ET	Time	T#0ms	No remanente	True	False	True	False		
IN	Bool	false	No remanente	True	True	True	False		
Q	Bool	false	No remanente	True	False	True	False		
Estado_conveyor	Int	0	No remanente	True	True	True	False		
Temp									
Constant									

Segmento 1: Marcha cinta

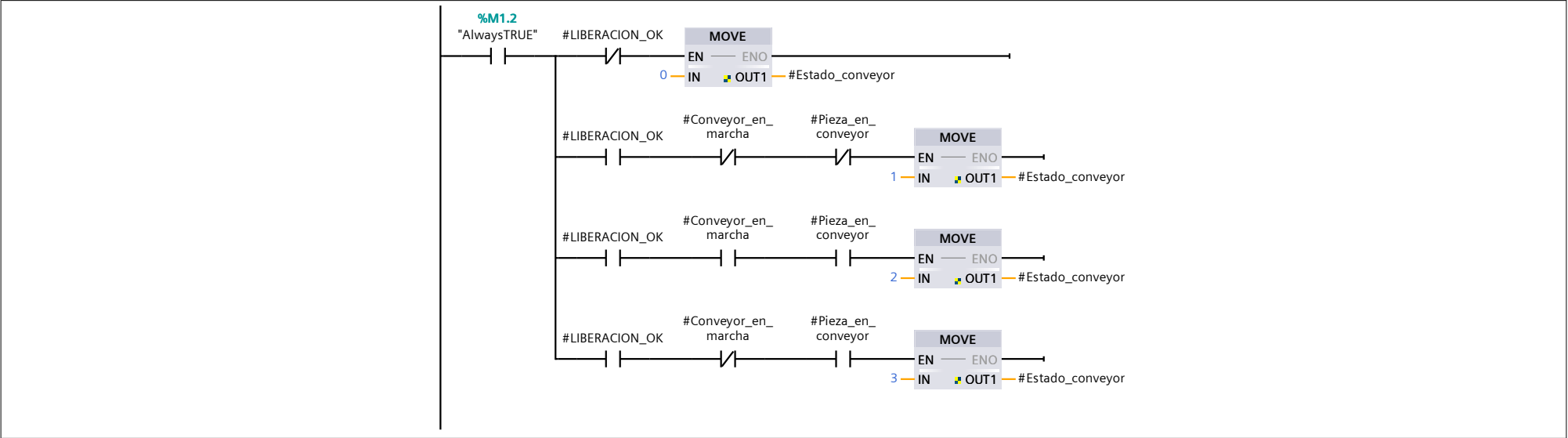


Segmento 2: Defincion Pieza en conveyor



Segmento 3: ESTADOS

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza
- =3 En marcha con pieza
- =4 Esperando



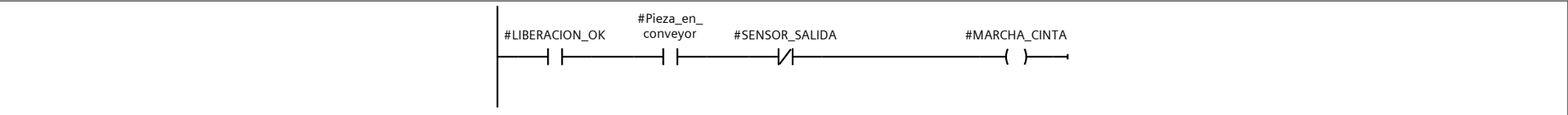
Bloques de programa / 12_AUX

CONVEYOR_FINAL [FB2]

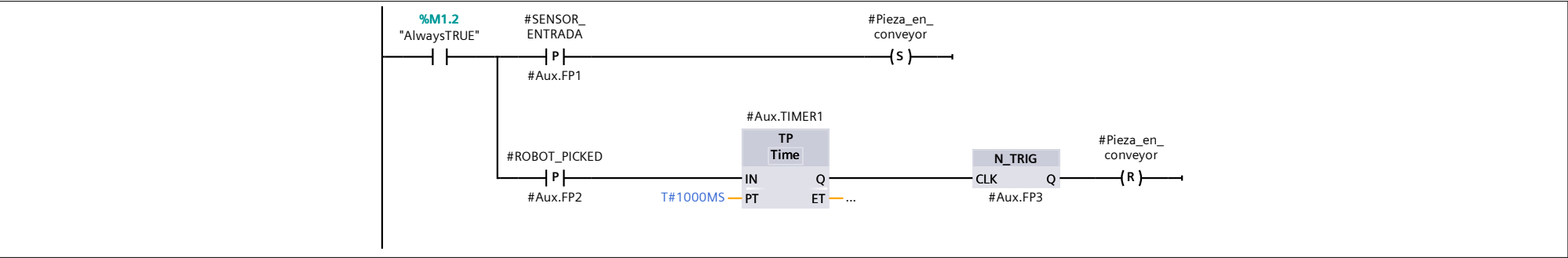
CONVEYOR_FINAL Propiedades							
General							
Nombre	CONVEYOR_FINAL	Número	2	Tipo	FB	Idioma	KOP
Numeración	Manual						
Información							
Título	FC CINTA GENERICO	Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible desde HMI/OPC UA	Es-cribi-ble desd e HMI/ OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	No remanente	False	False	False	False		
SENSOR_ENTRADA	Bool	false	No remanente	False	False	False	False		
SENSOR_SALIDA	Bool	false	No remanente	False	False	False	False		
ROBOT_PICKED	Bool	false	No remanente	True	True	True	False		
▼ Output									
MARCHA_CINTA	Bool	false	No remanente	False	False	False	False		
InOut									
▼ Static									
CINTA_EN_MARCHA	Bool	false	No remanente	True	True	True	False		
Pieza_en_conveyor	Bool	false	No remanente	True	True	True	False		
PIEZA_EN_POS	Bool	false	No remanente	True	True	True	False		
Conveyor_en_marcha	Bool	false	No remanente	True	True	True	False		
Estado_conveyor	Int	0	No remanente	True	True	True	False		
▼ Aux	Struct		No remanente	True	True	True	False		
FP1	Bool	false	No remanente	True	True	True	False		
FP2	Bool	false	No remanente	True	True	True	False		
FP3	Bool	false	No remanente	True	True	True	False		
▼ TIMER1	IEC_TIMER		No remanente	True	True	True	False		
PT	Time	T#0ms	No remanente	True	True	True	False		
ET	Time	T#0ms	No remanente	True	False	True	False		
IN	Bool	false	No remanente	True	True	True	False		
Q	Bool	false	No remanente	True	False	True	False		
Temp									
Constant									

Segmento 1: Marcha cinta



Segmento 2: Defincion Pieza en conveyor



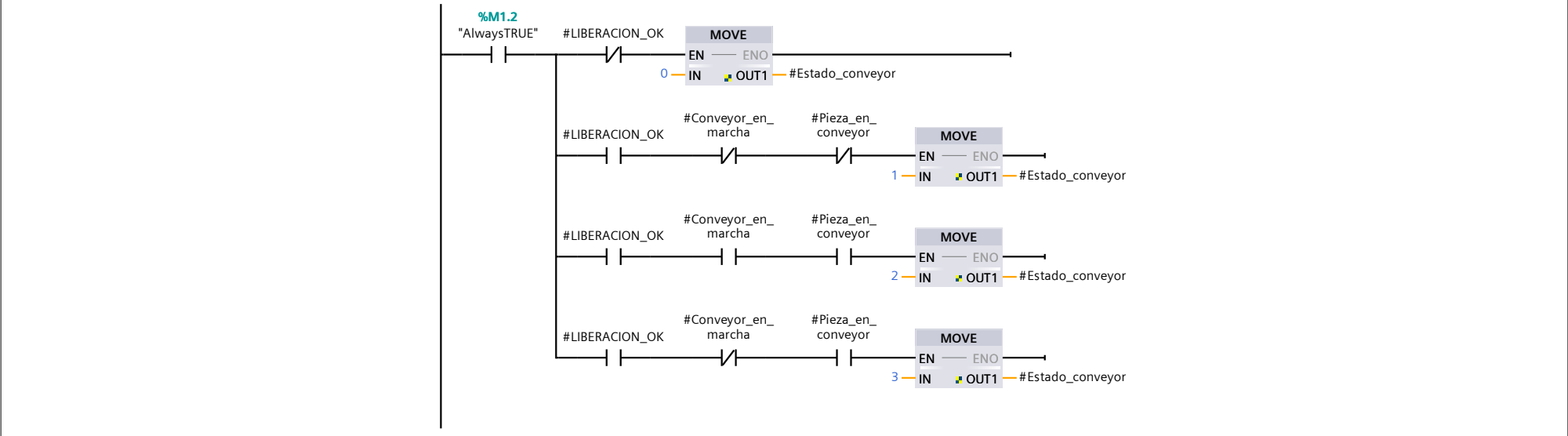
Segmento 3: PIEZA EN POSICION



Segmento 4: ESTADOS

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza

=3 En marcha con pieza
=4 Esperando



Totally Integrated Automation Portal

Bloques de programa / 03_RAMIFICACION_PIEZA_AZUL

CONVEYOR_7DB [DB2]

CONVEYOR_7DB Propiedades

General

Nombre	CONVEYOR_7DB	Número	2	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 03_RAMIFICACION_PIEZA_AZUL

CONVEYOR_8DB [DB3]

CONVEYOR_8DB Propiedades

General

Nombre	CONVEYOR_8DB	Número	3	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanen-cia	Accesible desde HMI/OPC UA	Es-cribi-ble desd e HMI/ OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 03_RAMIFICACION_PIEZA_AZUL

CONVEYOR_9DB [DB4]

CONVEYOR_9DB Propiedades

General

Nombre	CONVEYOR_9DB	Número	4	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

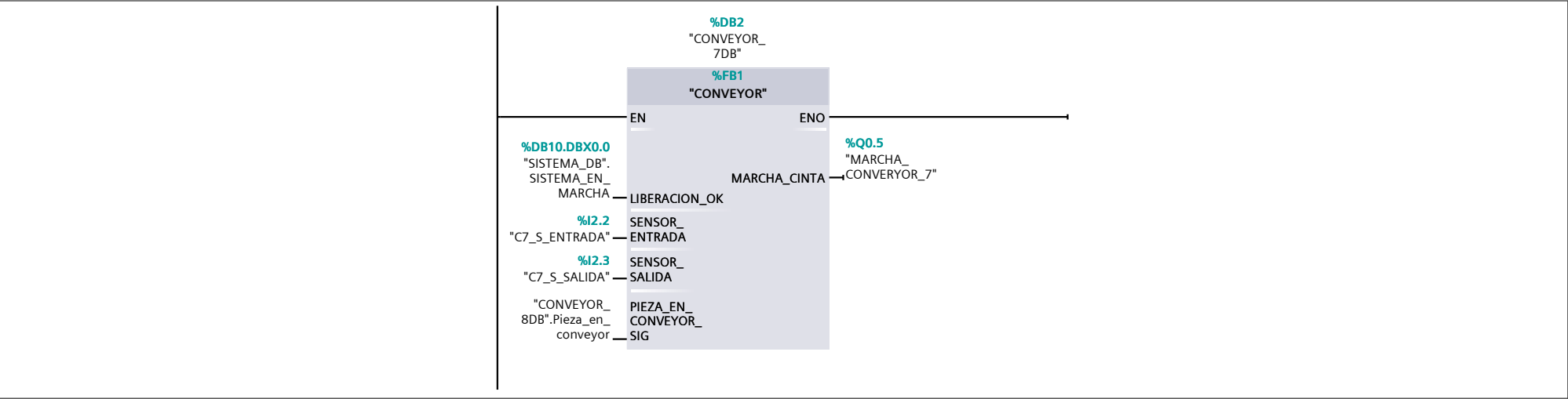
Bloques de programa / 03_RAMIFICACION_PIEZA_AZUL

Ramificacion_Azul [FC2010]

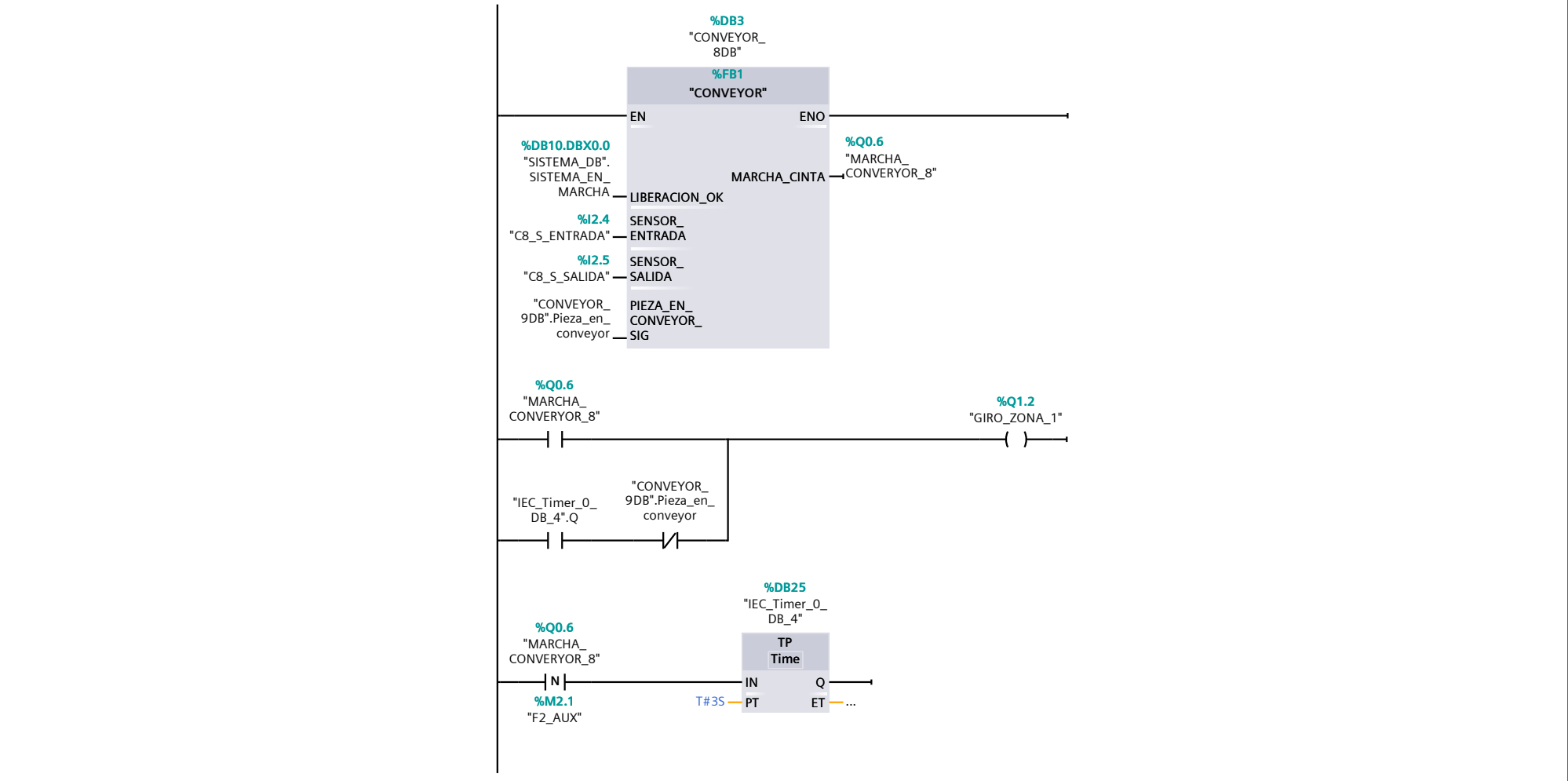
Ramificacion_Azul Propiedades							
General							
Nombre	Ramificacion_Azul	Número	2010	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título	RAMIFICACION PIEZAS AZULES	Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
Ramificacion_Azul	Void			

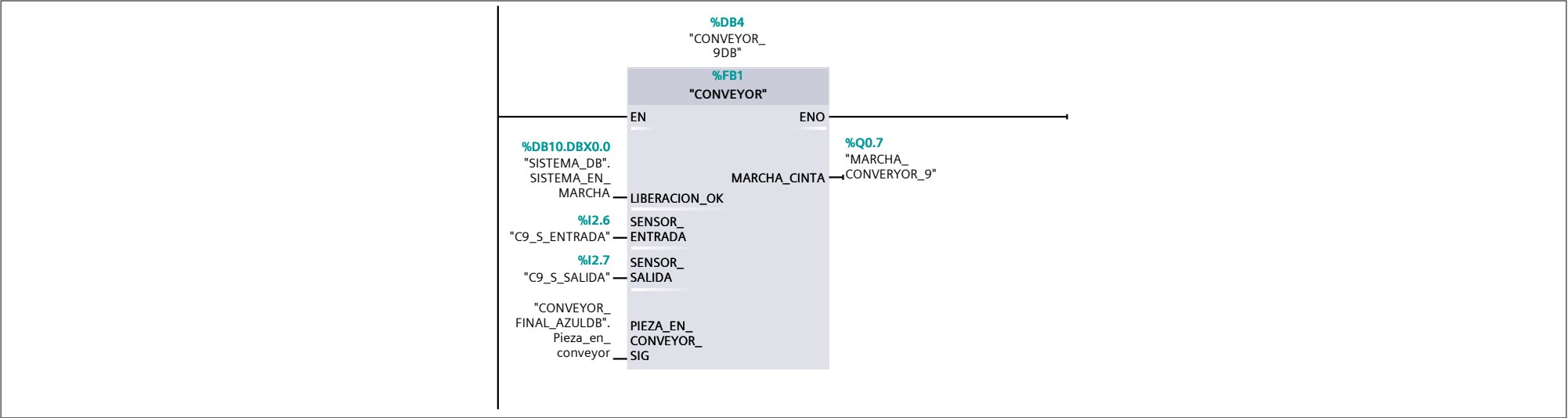
Segmento 1 : paso de la pieza azul a la cinta 7(empuje del cilindro)



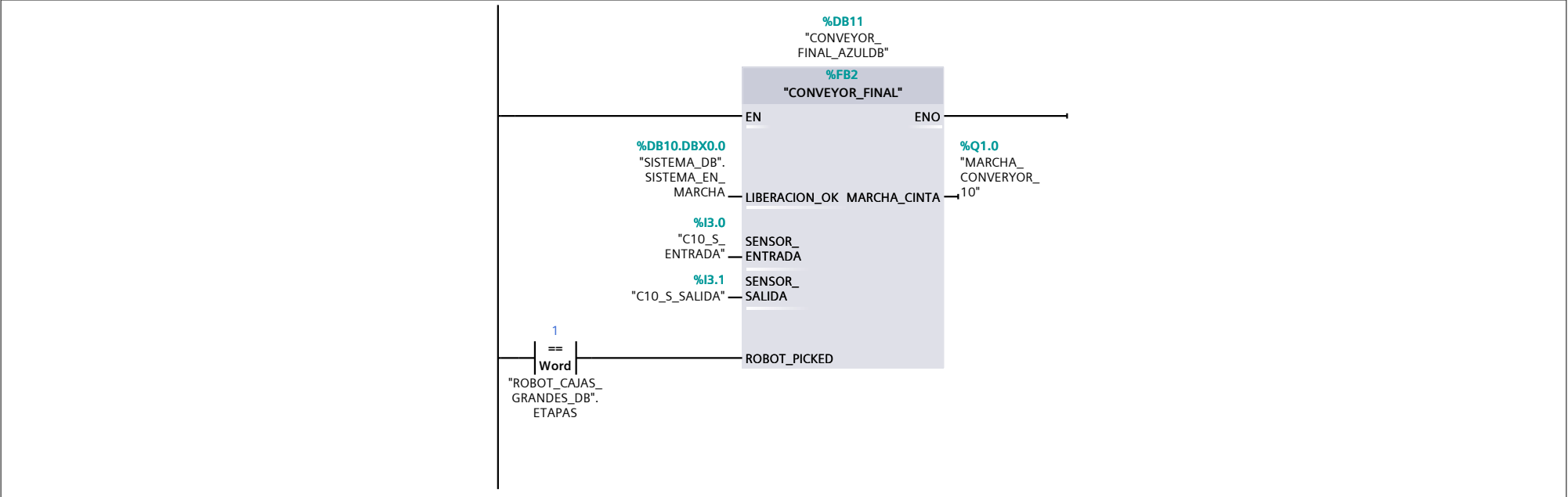
Segmento 2: PASO DE LA PIEZA DE LA CINTA 7 A LA CINTA 8



Segmento 3: CONVEYOR 9



Segmento 4: CONVEYOR 10 (CON FB ESPECIAL PARA LA CINTA FINAL)



Totally Integrated Automation Portal

Bloques de programa / 03_RAMIFICACION_PIEZA_AZUL

CONVEYOR_FINAL_AZULDB [DB11]

CONVEYOR_FINAL_AZULDB Propiedades

General

Nombre	CONVEYOR_FINAL_AZULDB	Número	11	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
ROBOT_PICKED	Bool	false	False	True	True	True	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
CINTA_EN_MARCHA	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
PIEZA_EN_POS	Bool	false	False	True	True	True	False		
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Totally Integrated Automation Portal

Bloques de programa / 04_RAMIFICACION_PIEZA_GRIS

CONVEYOR_11DB [DB6]

CONVEYOR_11DB Propiedades

General

Nombre	CONVEYOR_11DB	Número	6	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 04_RAMIFICACION_PIEZA_GRIS

CONVEYOR_12DB [DB7]

CONVEYOR_12DB Propiedades

General

Nombre	CONVEYOR_12DB	Número	7	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanen-cia	Accesible desde HMI/OPC UA	Es-cribi-ble desd e HMI/ OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 04_RAMIFICACION_PIEZA_GRIS

CONVEYOR_13DB [DB8]

CONVEYOR_13DB Propiedades

General

Nombre	CONVEYOR_13DB	Número	8	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

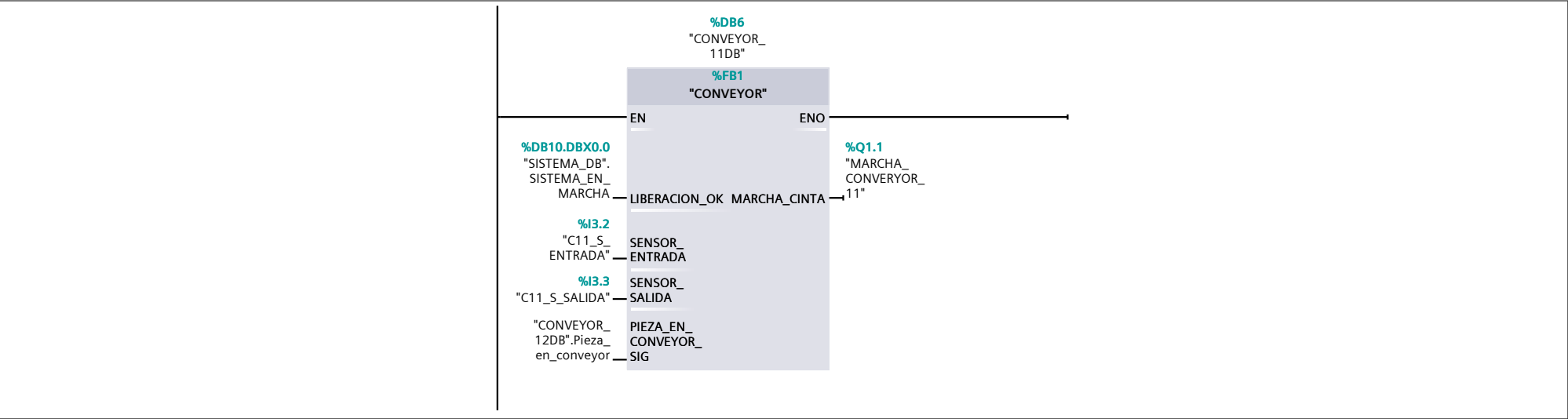
Bloques de programa / 04_RAMIFICACION_PIEZA_GRIS

Ramificacion_gris [FC2020]

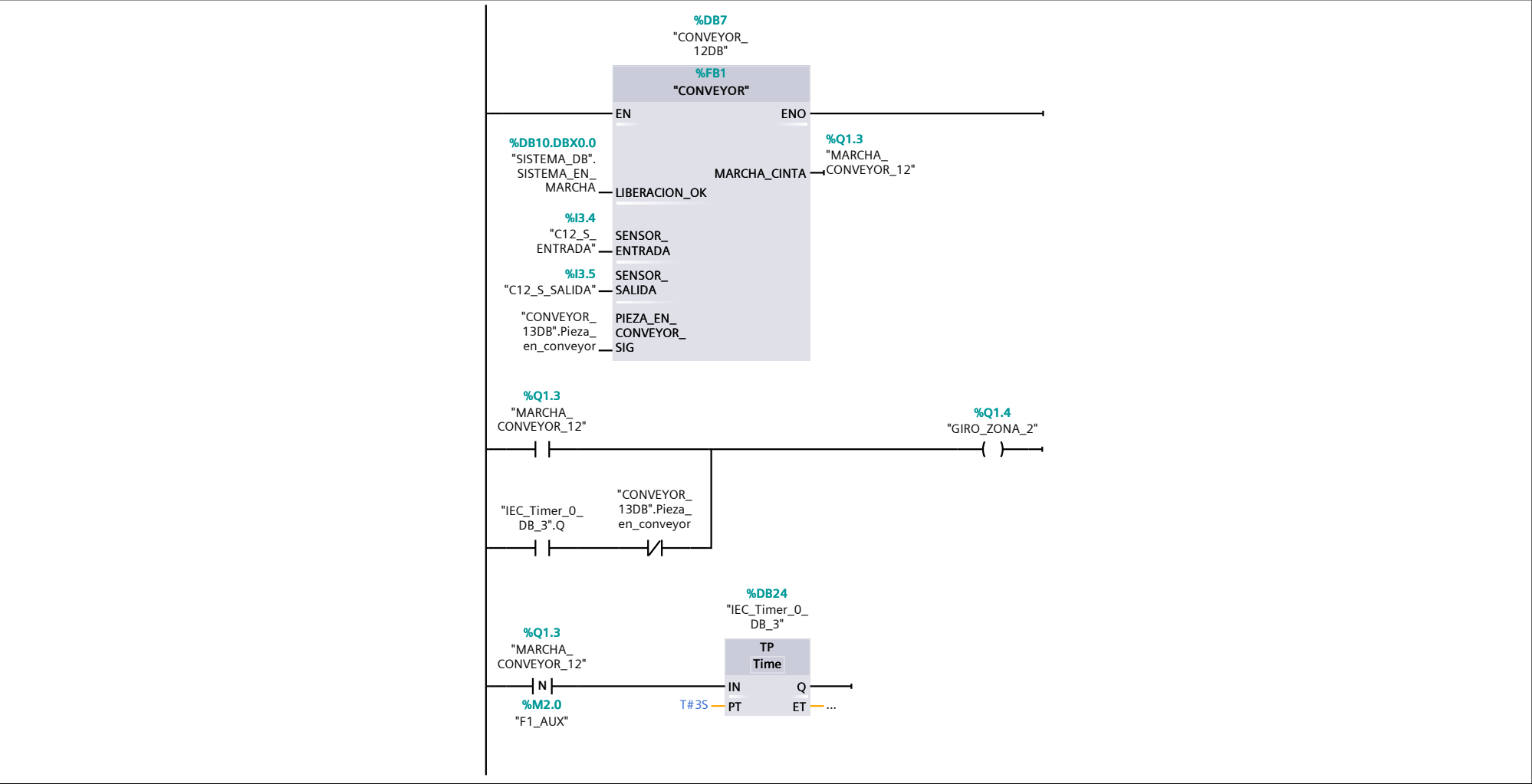
Ramificacion_gris Propiedades							
General							
Nombre	Ramificacion_gris	Número	2020	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título	RAMIFICACION DE PIEZAS GRISES	Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
Ramificacion_gris	Void			

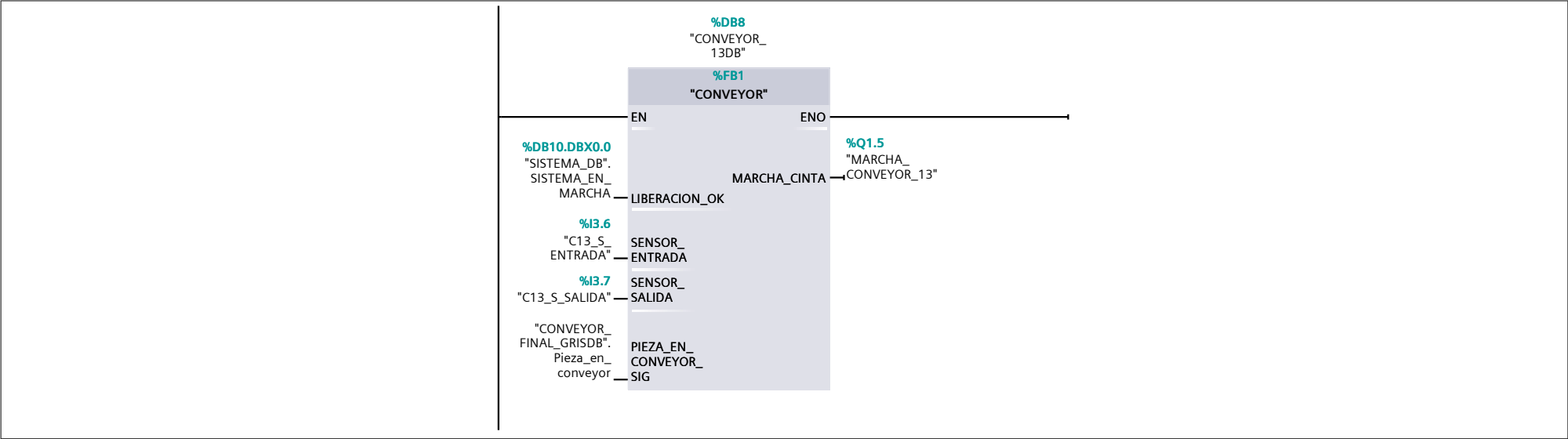
Segmento 1: CONVEYOR 11 (PASO DEL CILINDRO A LA CINTA)



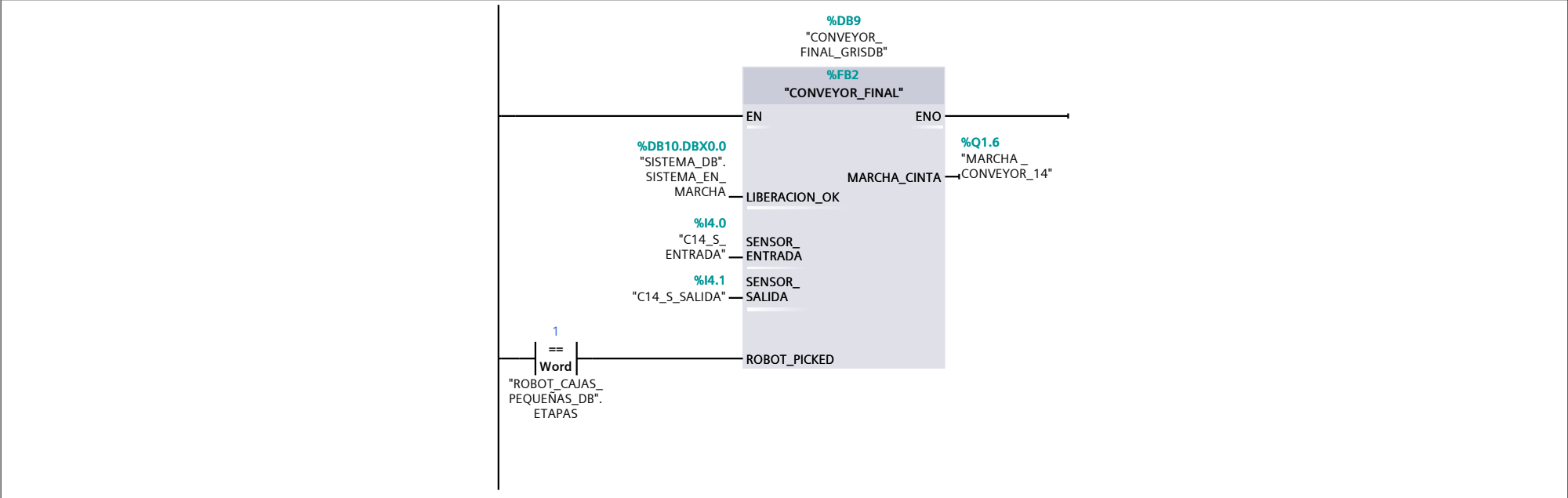
Segmento 2: CONVEYOR 12 MAS GIRO (CINTA MAS GIRO)



Segmento 3: CONVEYOR 13



Segmento 4: CONVEYOR 14 (CON FB ESPECIAL PARA LA CINTA FINAL)



Totally Integrated Automation Portal

Bloques de programa / 04_RAMIFICACION_PIEZA_GRIS

CONVEYOR_FINAL_GRISDB [DB9]

CONVEYOR_FINAL_GRISDB Propiedades

General

Nombre	CONVEYOR_FINAL_GRISDB	Número	9	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanen-cia	Accesible desde HMI/OPC UA	Es-cribi-ble desd e HMI/ OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
ROBOT_PICKED	Bool	false	False	True	True	True	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
CINTA_EN_MARCHA	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
PIEZA_EN_POS	Bool	false	False	True	True	True	False		
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

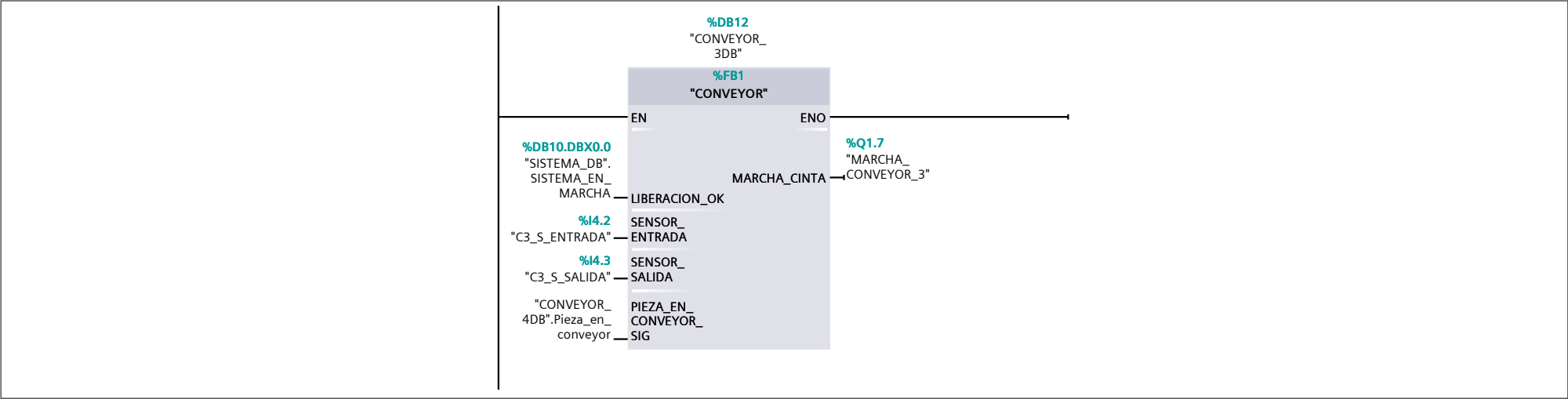
Bloques de programa / 05_RAMIFICACION_PIEZA_VERDE

RAMIFICACION_VERDES [FC2030]

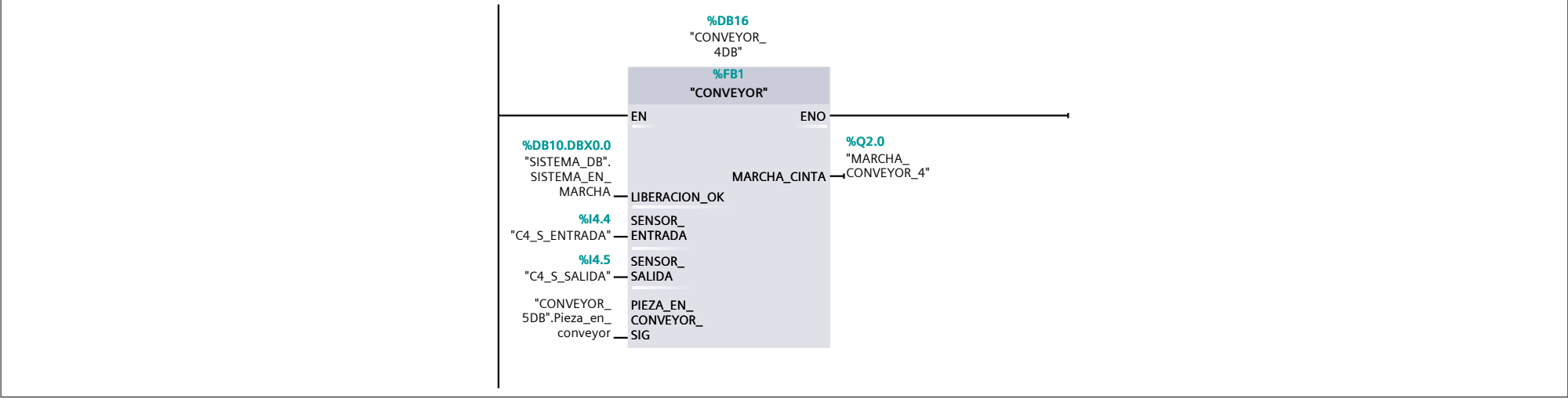
RAMIFICACION_VERDES Propiedades							
General							
Nombre	RAMIFICACION_VERDES	Número	2030	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título	RAMIFICACION PIEZAS VERDES	Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
RAMIFICACION_VERDES	Void			

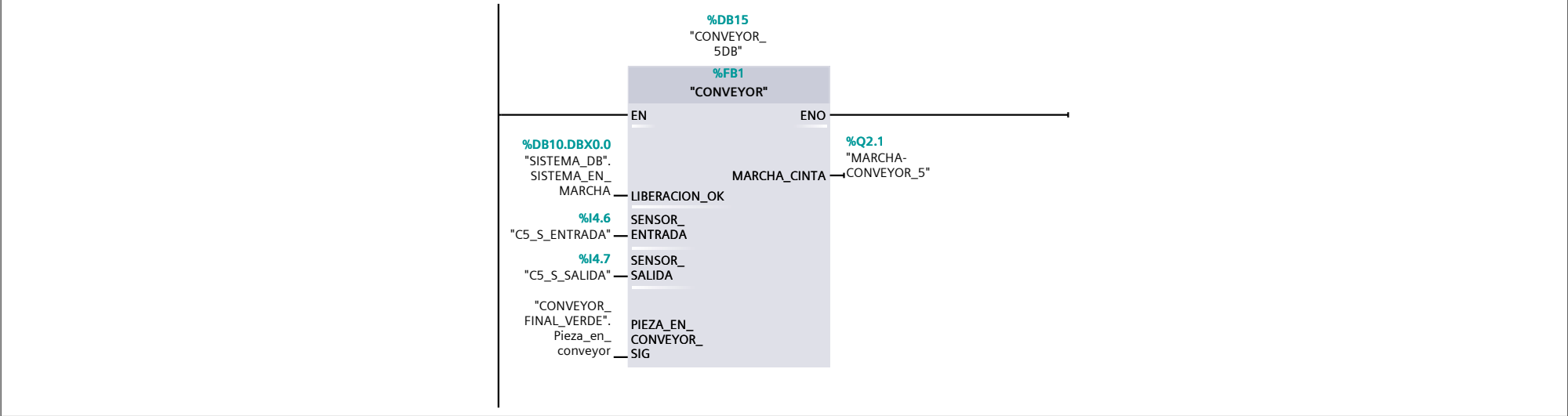
Segmento 1: CONVEYOR 3



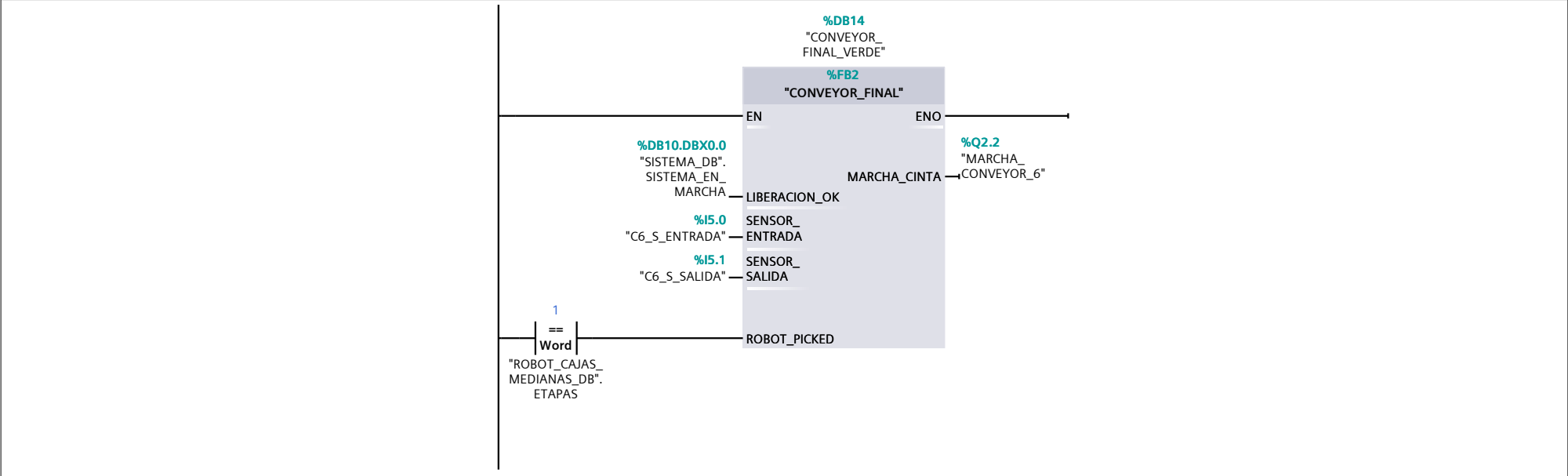
Segmento 2: CONVEYOR 4



Segmento 3: CONVEYOR 5



Segmento 4: CONVEYOR 6



Totally Integrated Automation Portal

Bloques de programa / 05_RAMIFICACION_PIEZA_VERDE

CONVEYOR_3DB [DB12]

CONVEYOR_3DB Propiedades

General

Nombre	CONVEYOR_3DB	Número	12	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 05_RAMIFICACION_PIEZA_VERDE

CONVEYOR_FINAL_VERDE [DB14]

CONVEYOR_FINAL_VERDE Propiedades

General

Nombre	CONVEYOR_FINAL_VERDE	Número	14	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
ROBOT_PICKED	Bool	false	False	True	True	True	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
CINTA_EN_MARCHA	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
PIEZA_EN_POS	Bool	false	False	True	True	True	False		
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Bloques de programa / 05_RAMIFICACION_PIEZA_VERDE

CONVEYOR_5DB [DB15]

CONVEYOR_5DB Propiedades							
General							
Nombre	CONVEYOR_5DB	Número	15	Tipo	DB	Idioma	DB
Numeración	Automático						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personaliza- da					

Nombre	Tipo de datos	Valor de arranque	Remanen- cia	Accesible desde HMI/OPC UA	Es- cribi- ble desd e HMI/ OPC UA	Visible en HMI Engi- neering	Valor de ajuste	Supervi- sión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

Totally Integrated Automation Portal

Bloques de programa / 05_RAMIFICACION_PIEZA_VERDE

CONVEYOR_4DB [DB16]

CONVEYOR_4DB Propiedades

General

Nombre	CONVEYOR_4DB	Número	16	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

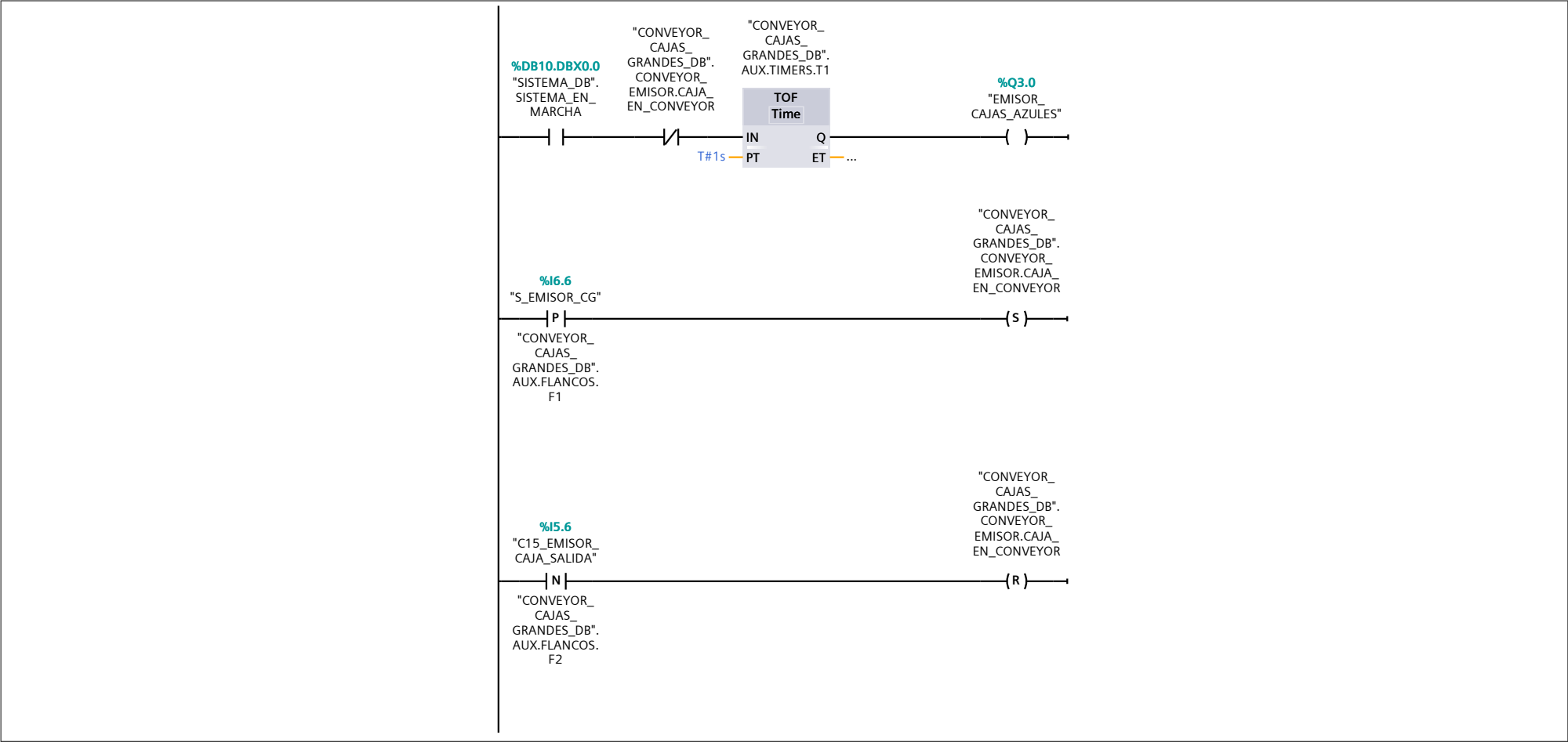
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
LIBERACION_OK	Bool	false	False	False	False	False	False		
SENSOR_ENTRADA	Bool	false	False	False	False	False	False		
SENSOR_SALIDA	Bool	false	False	False	False	False	False		
PIEZA_EN_CONVEYOR_SIG	Bool	false	False	False	False	False	False		
▼ Output									
MARCHA_CINTA	Bool	false	False	False	False	False	False		
InOut									
▼ Static									
Conveyor_en_marcha	Bool	false	False	True	True	True	False		
Pieza_en_conveyor	Bool	false	False	True	True	True	False		
▼ Aux	Struct		False	True	True	True	False		
FP1	Bool	false	False	True	True	True	False		
FP2	Bool	false	False	True	True	True	False		
FP3	Bool	false	False	True	True	True	False		
▼ TIMER1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ TIMER2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
Estado_conveyor	Int	0	False	True	True	True	False		

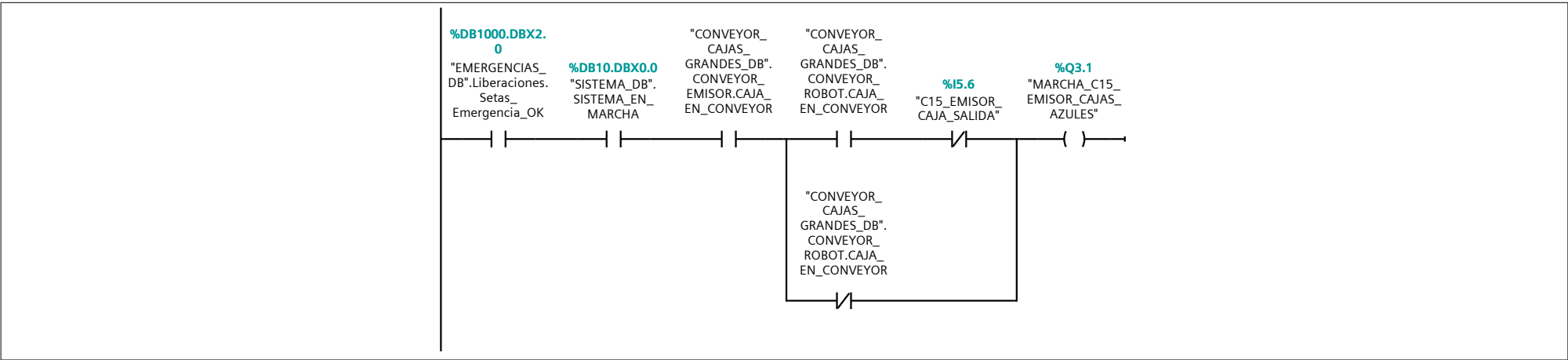
CONVEYOR_CAJAS_GRADES [FC2040]

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
CONVEYOR_CAJAS_GRADES	Void			

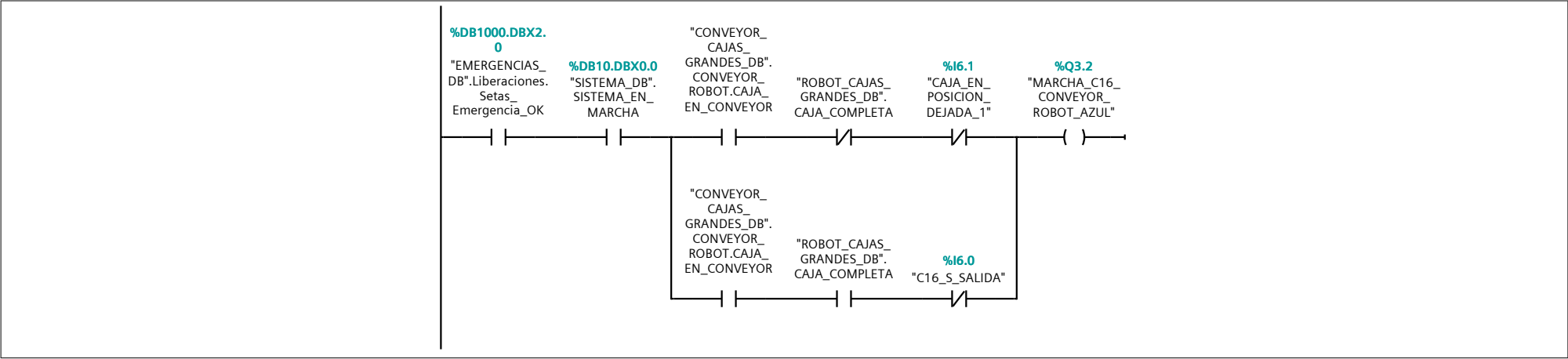
Segmento 1: EMISOR CAJAS



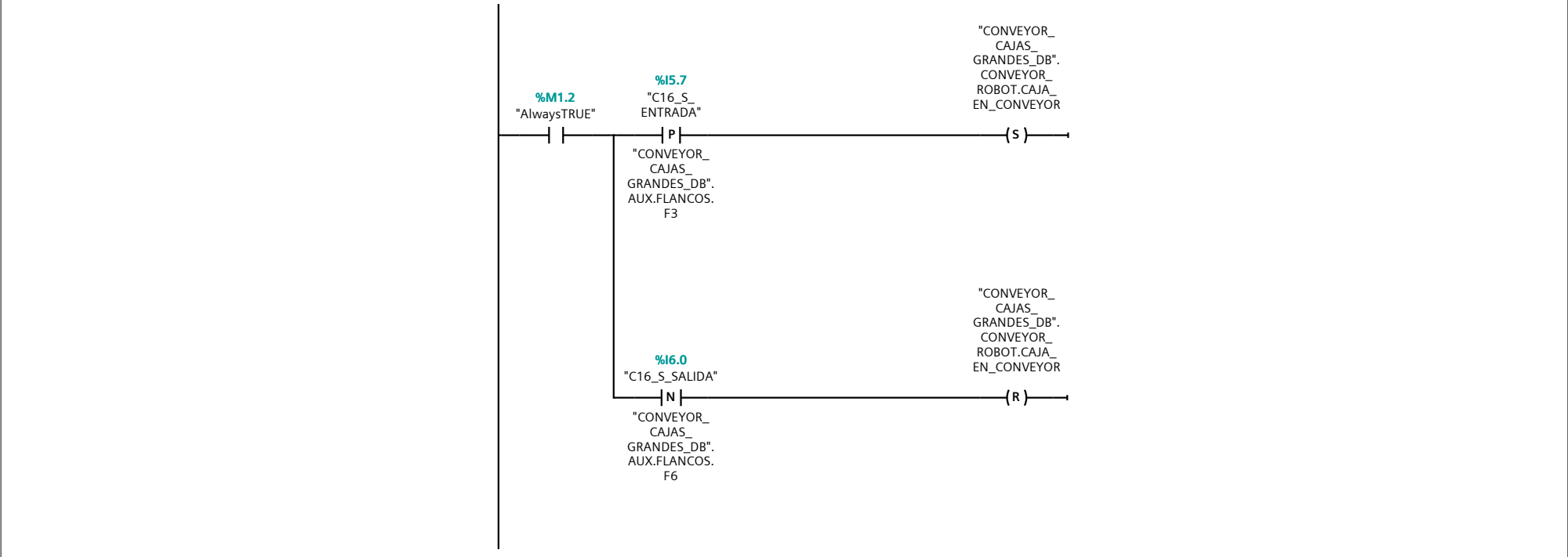
Segmento 2: MARCHA CINTA



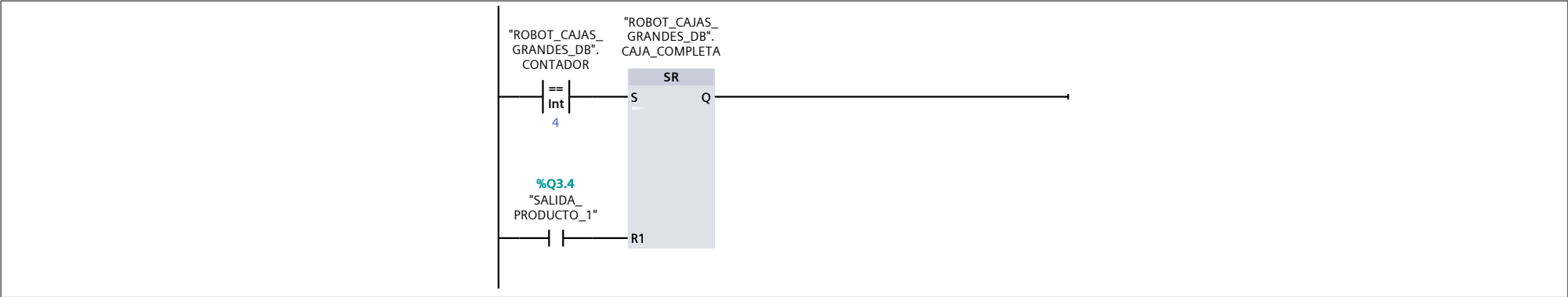
Segmento 3: ----- CONVEYOR_ROBOT -----



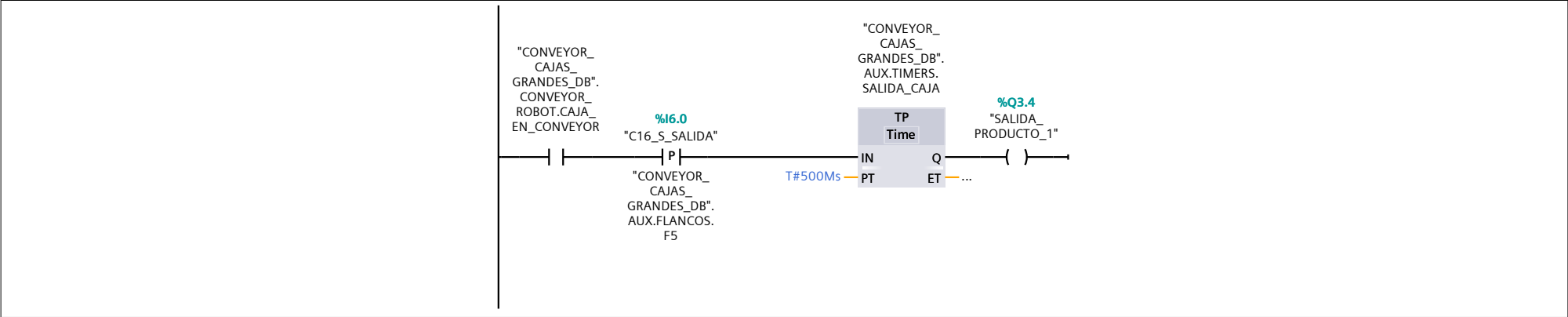
Segmento 4: CAJA EN CINTA



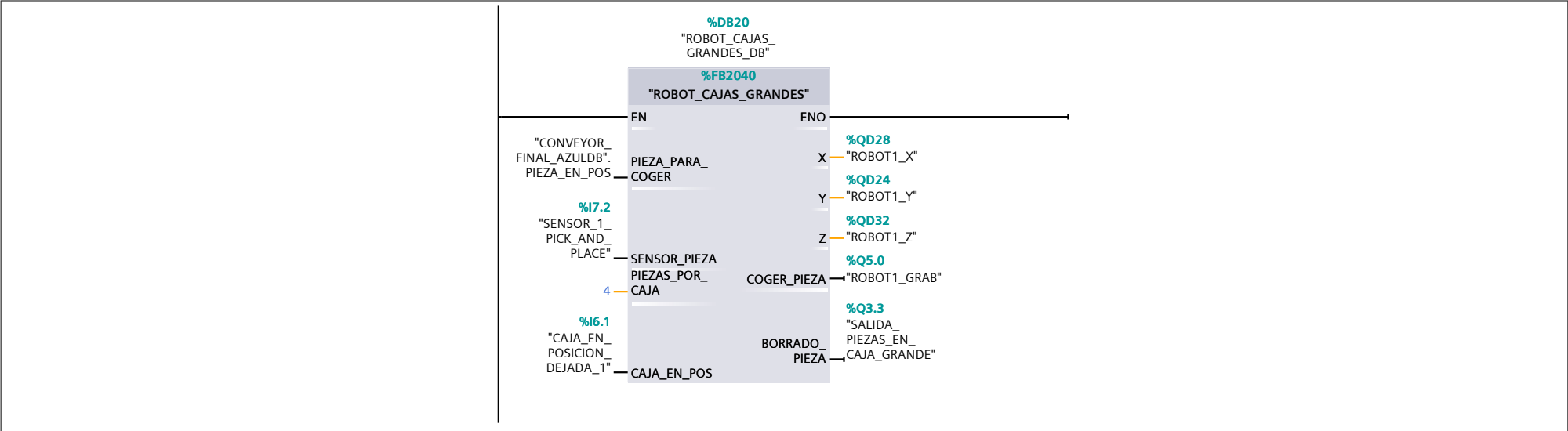
Segmento 5: CAJA LLENA/VACIA



Segmento 6: BORRADO DE PIEZA EN CAJA

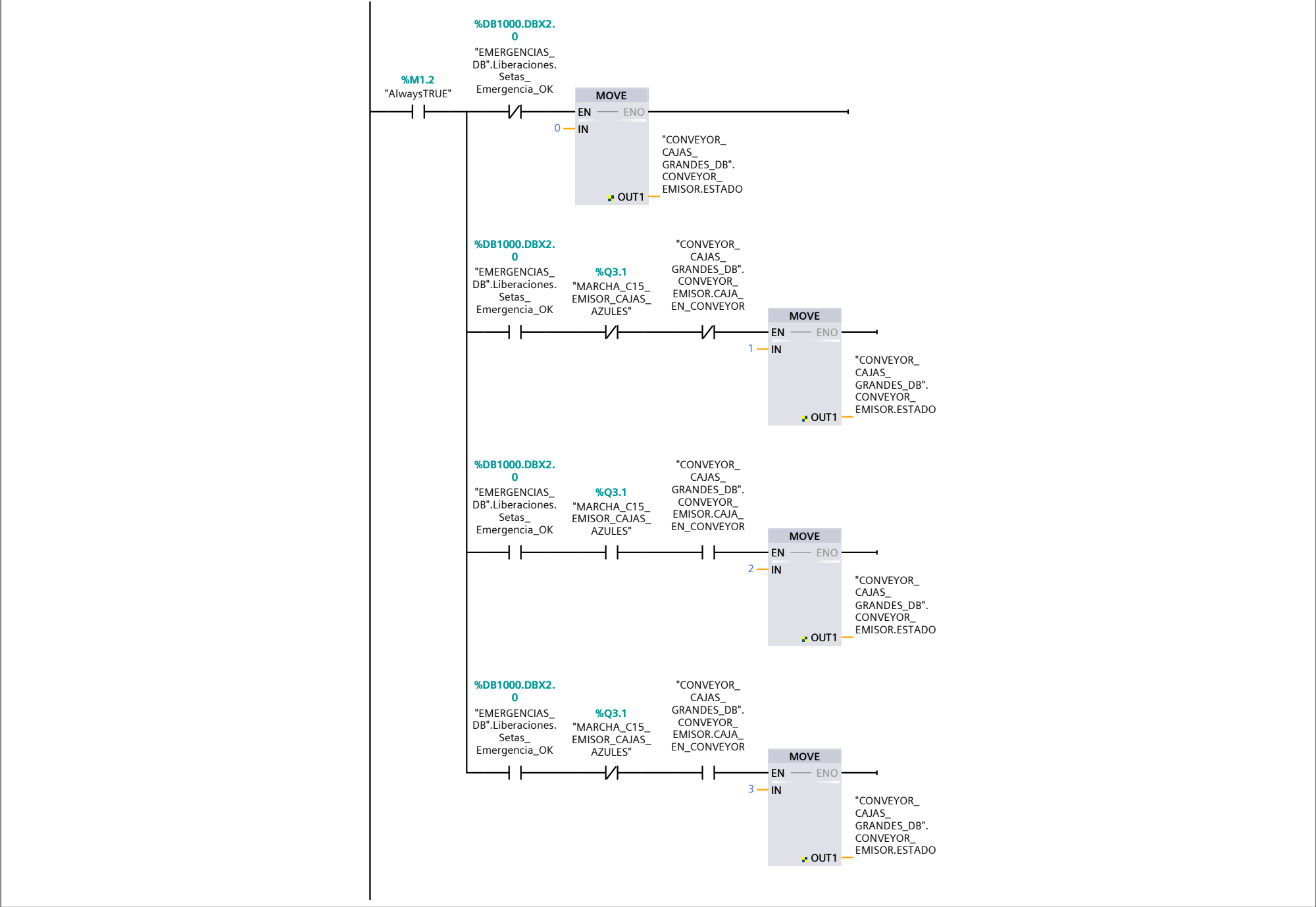


Segmento 7: FB ROBOT



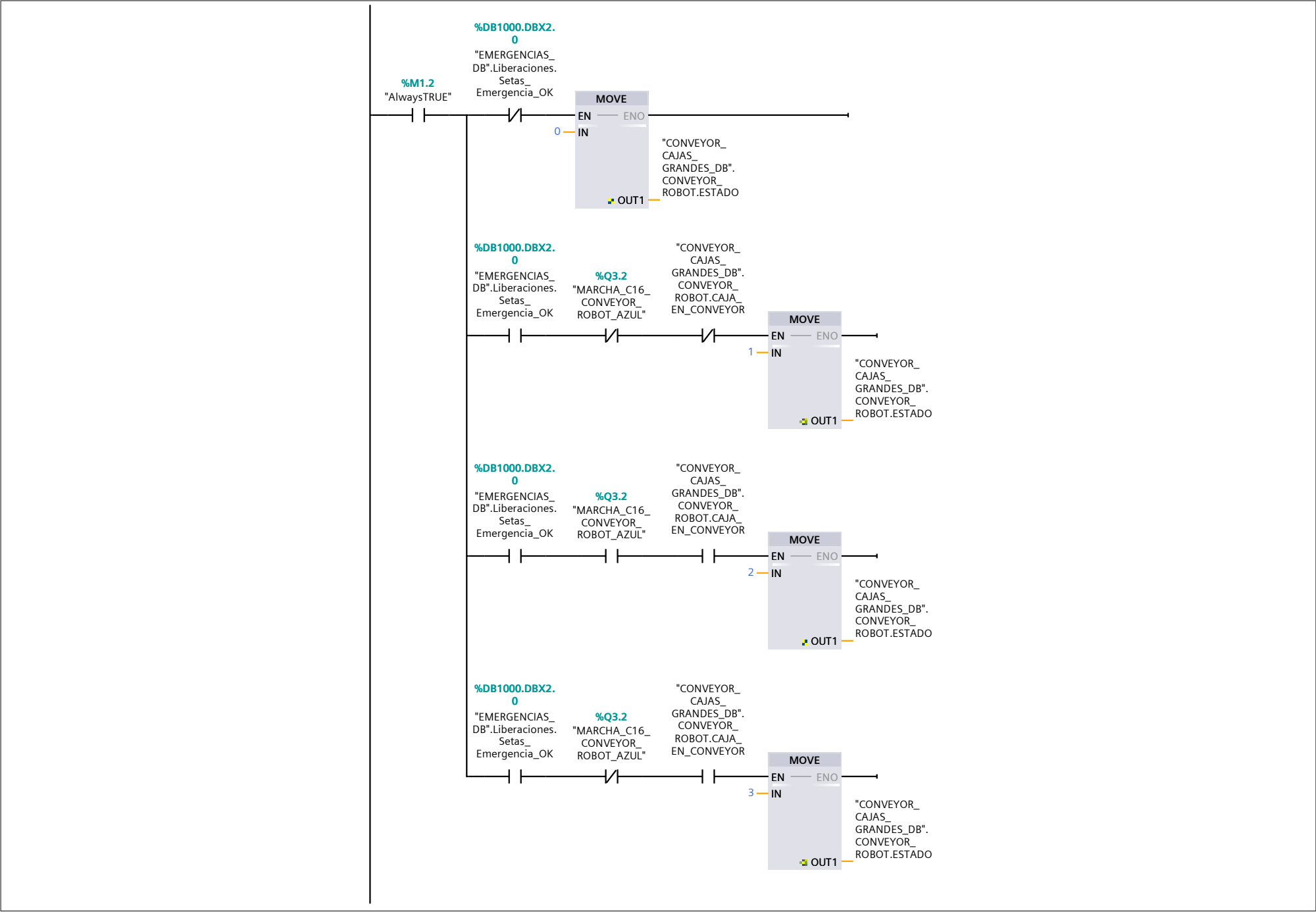
Segmento 8: ESTADO - CONVEYOR EMISOR CAJAS

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza
- =3 En marcha con pieza
- =4 Esperando



Segmento 9: ESTADO - CONVEYOR SALIDA A ROBOT

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza
- =3 En marcha con pieza
- =4 Esperando



Totally Integrated Automation Portal

Bloques de programa / 06_CONVEYOR_CAJAS_GRADES

CONVEYOR_CAJAS_GRADES_DB [DB2040]

CONVEYOR_CAJAS_GRADES_DB Propiedades

General

Nombre	CONVEYOR_CAJAS_GRADES_DB	Número	2040	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanen-cia	Accesible desde HMI/OPC UA	Es-cribi-ble desde HMI/OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Static									
▼ CONVEYOR_EMITOR	Struct		False	True	True	True	False		
CAJA_EN_CONVEYOR	Bool	false	False	True	True	True	False		
ESTADO	Int	0	False	True	True	True	False		
▼ CONVEYOR_ROBOT	Struct		False	True	True	True	False		
CAJA_EN_CONVEYOR	Bool	false	False	True	True	True	False		
PIEZAS_EN_CAJA	Bool	false	False	True	True	True	False		
SET-POINT_PIEZAS	Int	4	False	True	True	True	False		
CAJA_LLENA	Bool	false	False	True	True	True	False		
▼ CONTADOR_PIEZAS	IEC_COUNTER		False	True	True	True	False		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		
PV	Int	0	False	True	True	True	False		
CV	Int	0	False	True	True	True	False		
POS_ROBOT_COGER	Bool	false	False	True	True	True	False		
POS_ROBOT_DEJAR	Bool	false	False	True	True	True	False		
COGER_PIEZA	Bool	false	False	True	True	True	False		
DEJAR_PIEZA	Bool	false	False	True	True	True	False		
ESTADO	Int	0	False	True	True	True	False		
▼ AUX	Struct		False	True	True	True	False		
▼ FLANCOS	Struct		False	True	True	True	False		
F1	Bool	false	False	True	True	True	False		
F2	Bool	false	False	True	True	True	False		
F3	Bool	false	False	True	True	True	False		
F4	Bool	false	False	True	True	True	False		
F5	Bool	false	False	True	True	True	False		
F6	Bool	false	False	True	True	True	False		
▼ TIMERS	Struct		False	True	True	True	False		
▼ T1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_3	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_4	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

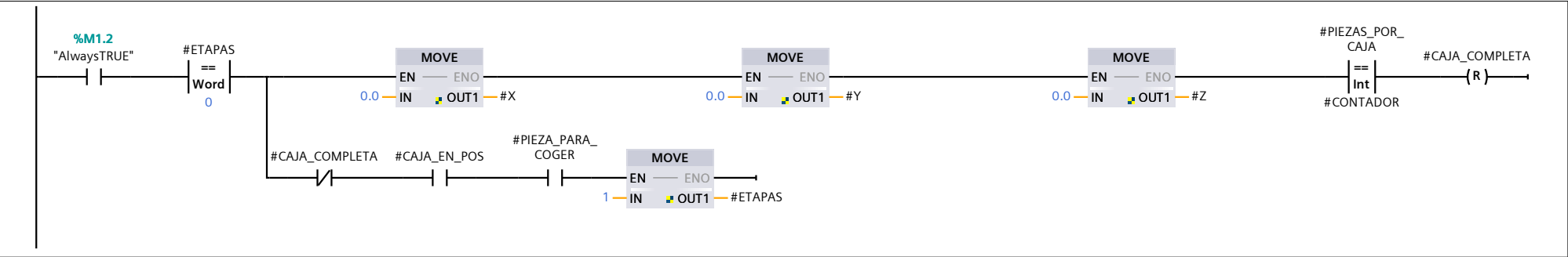
Bloques de programa / 06_CONVEYOR_CAJAS_GRADES

ROBOT_CAJAS_GRADES [FB2040]

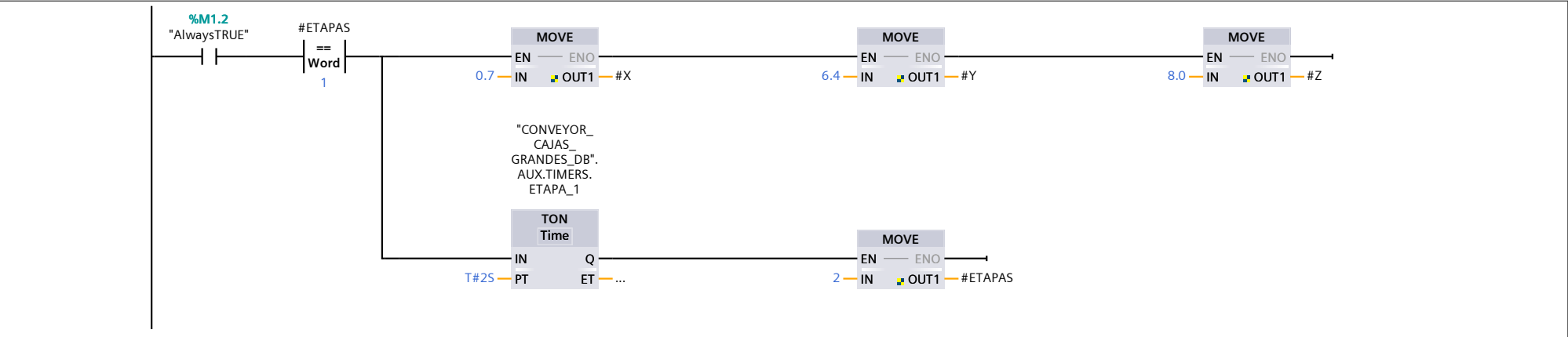
ROBOT_CAJAS_GRADES Propiedades							
General							
Nombre	ROBOT_CAJAS_GRADES	Número	2040	Tipo	FB	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personaliza- da					

Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible desde HMI/OPC UA	Es-cribi-ble desde HMI/OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
PIEZA_PARA_COGER	Bool	false	No remanente	True	True	True	False		
SENSOR_PIEZA	Bool	false	No remanente	True	True	True	False		
PIEZAS_POR_CAJA	Int	0	No remanente	True	True	True	False		
CAJA_EN_POS	Bool	false	No remanente	True	True	True	False		
▼ Output									
X	Real	0.0	No remanente	True	True	True	False		
Y	Real	0.0	No remanente	True	True	True	False		
Z	Real	0.0	No remanente	True	True	True	False		
COGER_PIEZA	Bool	false	No remanente	True	True	True	False		
BORRADO_PIEZA	Bool	false	No remanente	True	True	True	False		
InOut									
▼ Static									
ETAPAS	Word	16#0	No remanente	True	True	True	False		
CONTADOR	Int	0	No remanente	True	True	True	False		
CAJA_COMPLETA	Bool	false	No remanente	True	True	True	False		
▼ AUX	Struct		No remanente	True	True	True	False		
▼ FLANCOS	Struct		No remanente	True	True	True	False		
F1	Bool	false	No remanente	True	True	True	False		
▼ TIMERS	Struct		No remanente	True	True	True	False		
▼ T1	IEC_TIMER		No remanente	True	True	True	False		
PT	Time	T#0ms	No remanente	True	True	True	False		
ET	Time	T#0ms	No remanente	True	False	True	False		
IN	Bool	false	No remanente	True	True	True	False		
Q	Bool	false	No remanente	True	False	True	False		
Temp									
Constant									

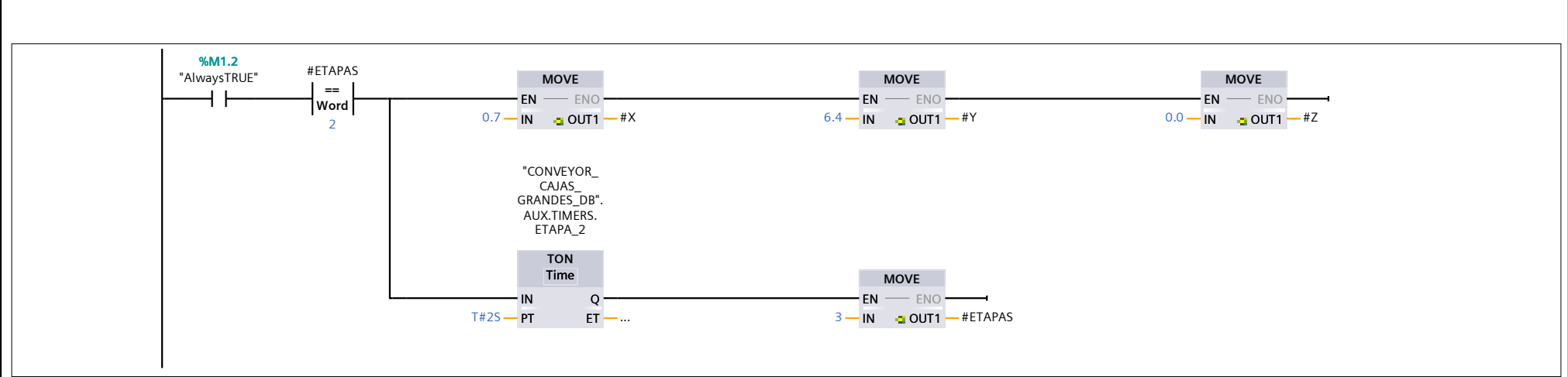
Segmento 1: ETAPA - REPOSO



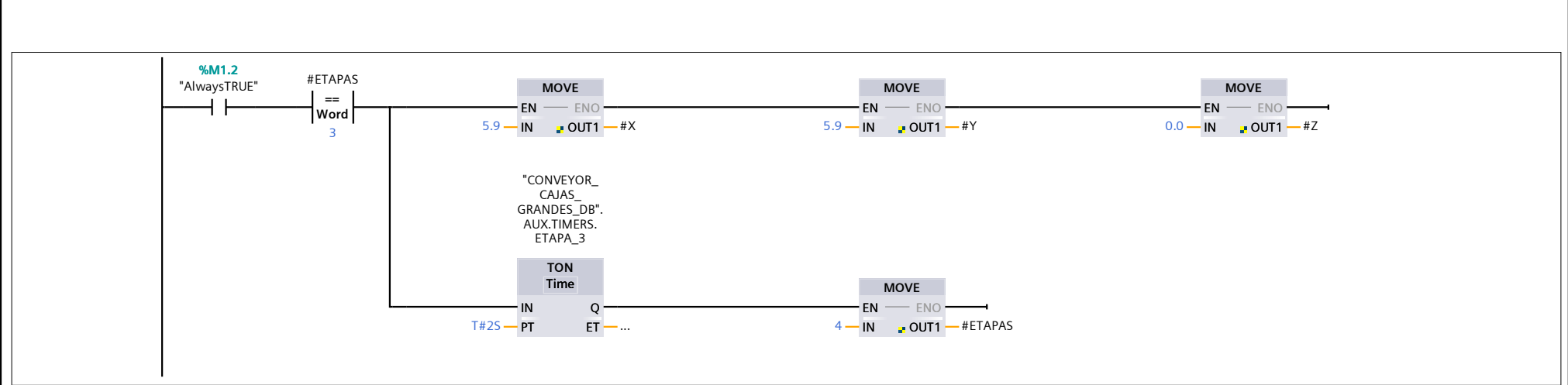
Segmento 2: ETAPA 1 ----> POSICION COGER PIEZA



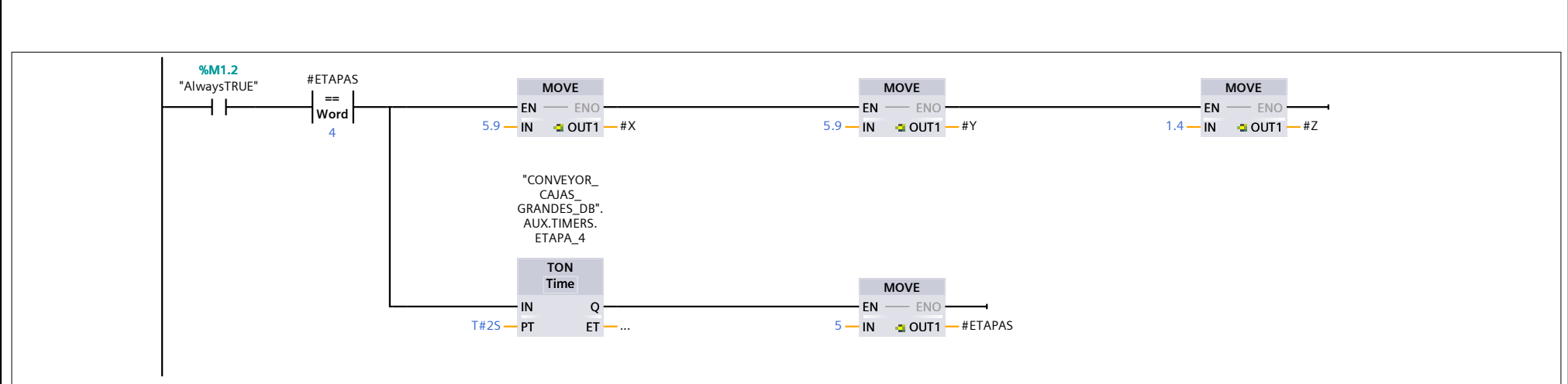
Segmento 3: ETAPA 2 ---> LEVANTAMOS PIEZA - POST COGIDA PIEZA



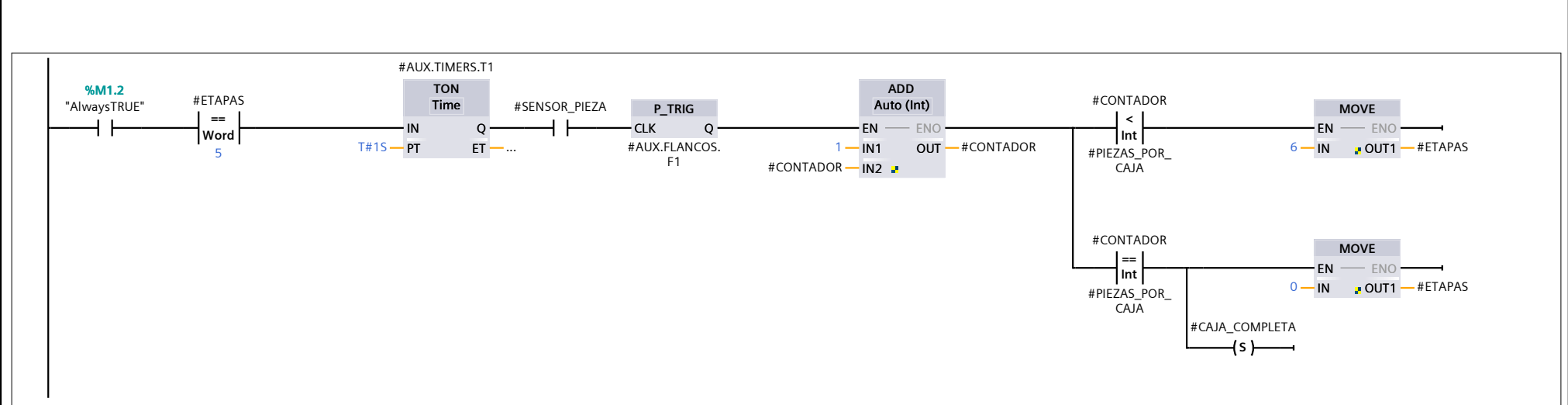
Segmento 4: ETAPA 3 ---> MOVEMOS A PRE DEJADA



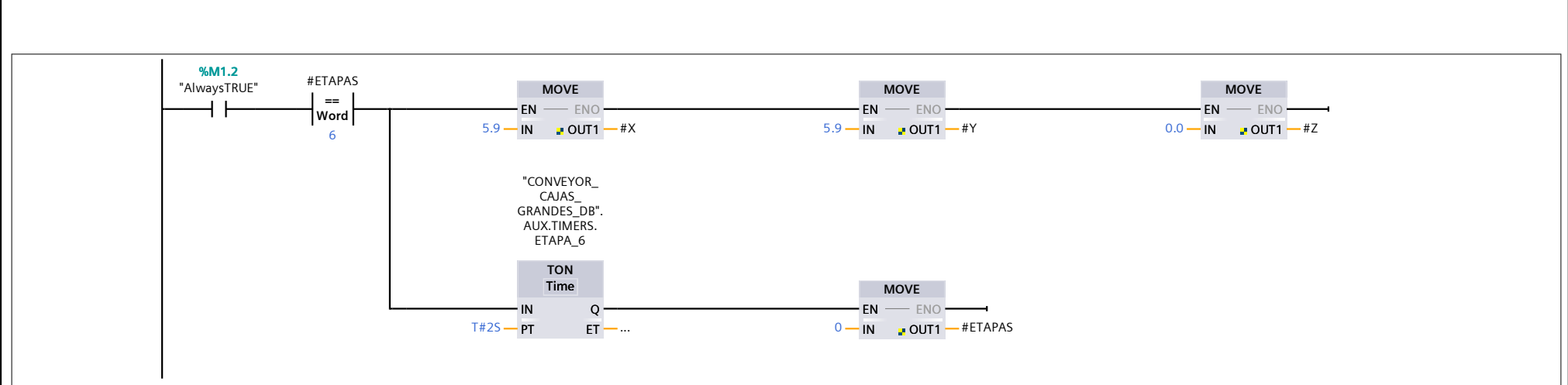
Segmento 5: ETAPA 4--> DEJO PIEZA



Segmento 6: ETAPA 5: COMPRUEBO CAJA



Segmento 7: ETAPA 6: RETORNO A PRE-DEJADA



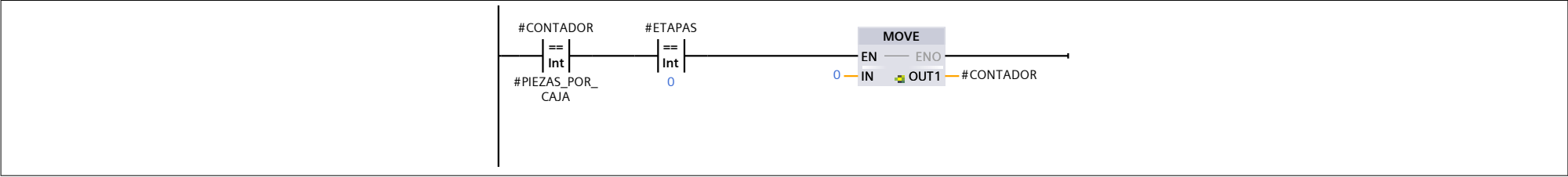
Segmento 8: COGER PIEZA



Segmento 9: BORRAR PIEZA



Segmento 10: Contador



Totally Integrated Automation Portal

Bloques de programa / 06_CONVEYOR_CAJAS_GRADES

ROBOT_CAJAS_GRADES_DB [DB20]

ROBOT_CAJAS_GRADES_DB Propiedades

General

Nombre	ROBOT_CA-JAS_GRADES_DB	Número	20	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanen-cia	Accesible desde HMI/OPC UA	Es-cribi-ble desd e HMI/ OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
PIEZA_PARA_COGER	Bool	false	False	True	True	True	False		
SENSOR_PIEZA	Bool	false	False	True	True	True	False		
PIEZAS_POR_CAJA	Int	0	False	True	True	True	False		
CAJA_EN_POS	Bool	false	False	True	True	True	False		
▼ Output									
X	Real	0.0	False	True	True	True	False		
Y	Real	0.0	False	True	True	True	False		
Z	Real	0.0	False	True	True	True	False		
COGER_PIEZA	Bool	false	False	True	True	True	False		
BORRADO_PIEZA	Bool	false	False	True	True	True	False		
InOut									
▼ Static									
ETAPAS	Word	16#0	False	True	True	True	False		
CONTADOR	Int	0	False	True	True	True	False		
CAJA_COMPLETA	Bool	false	False	True	True	True	False		
▼ AUX	Struct		False	True	True	True	False		
▼ FLANCOS	Struct		False	True	True	True	False		
F1	Bool	false	False	True	True	True	False		
▼ TIMERS	Struct		False	True	True	True	False		
▼ T1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

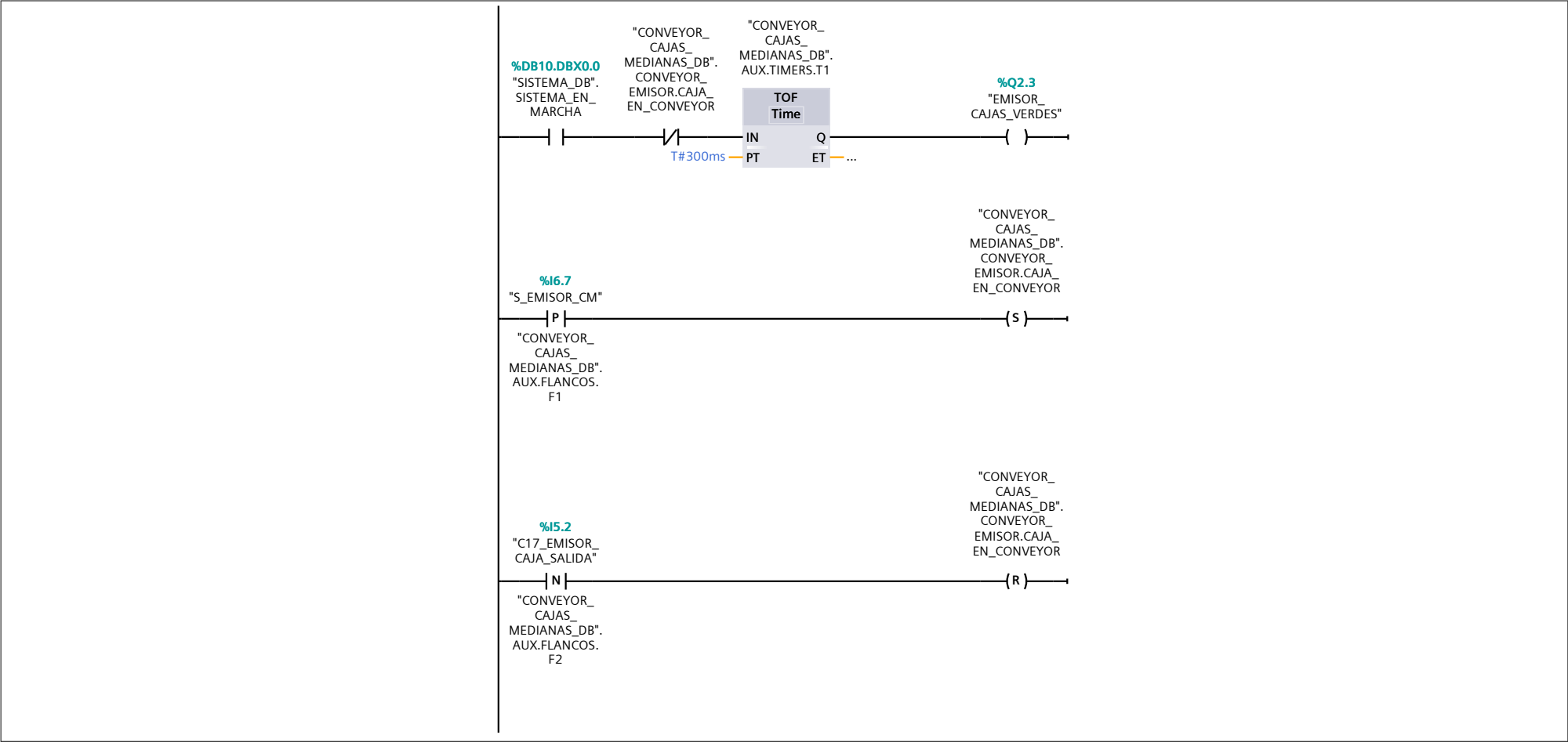
Bloques de programa / 07_CONVEYOR_CAJAS_MEDIANAS

CONVEYOR_CAJAS_MEDIANAS [FC2050]

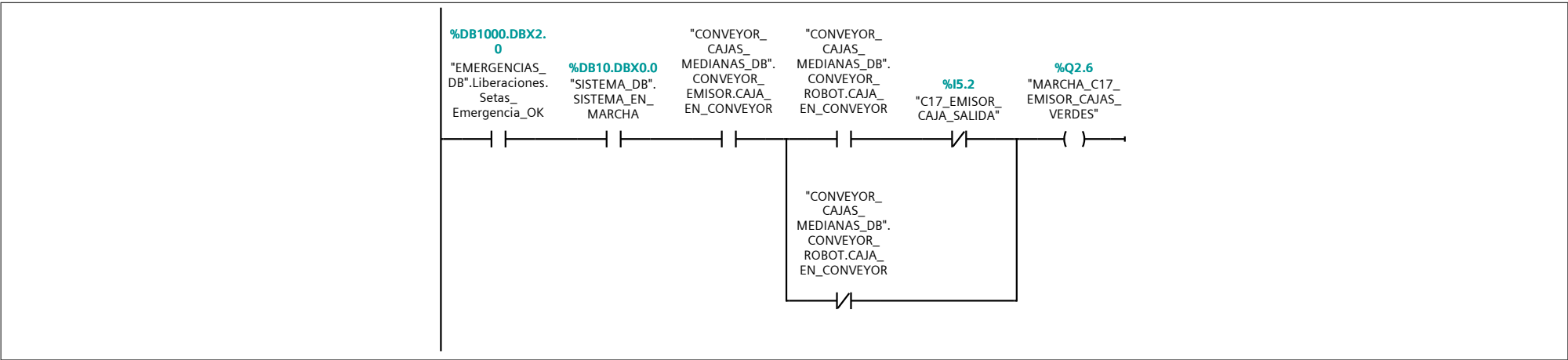
CONVEYOR_CAJAS_MEDIANAS Propiedades							
General							
Nombre	CONVEYOR_CAJAS_MEDIANAS	Número	2050	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
CONVEYOR_CAJAS_MEDIANAS	Void			

Segmento 1: EMISOR CAJAS

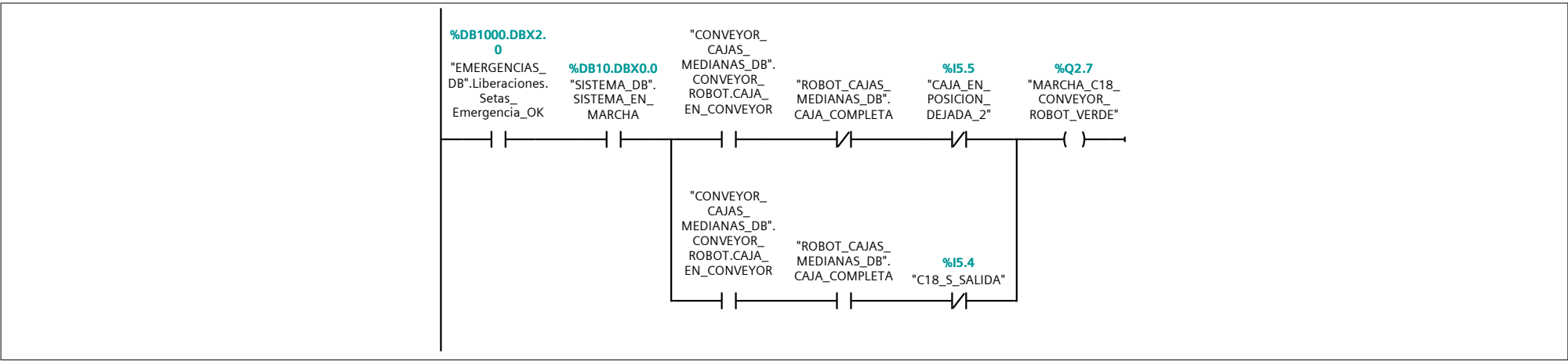


Segmento 2: MARCHA CINTA

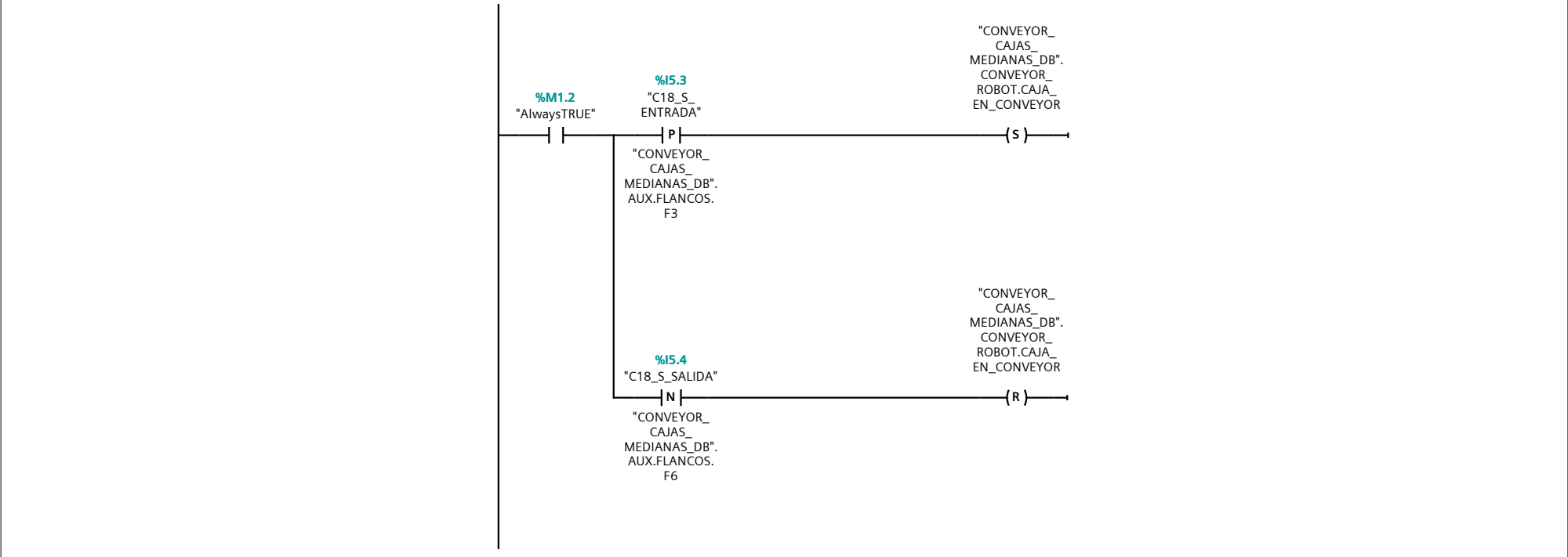


Segmento 3: ----- CONVEYOR_ROBOT -----

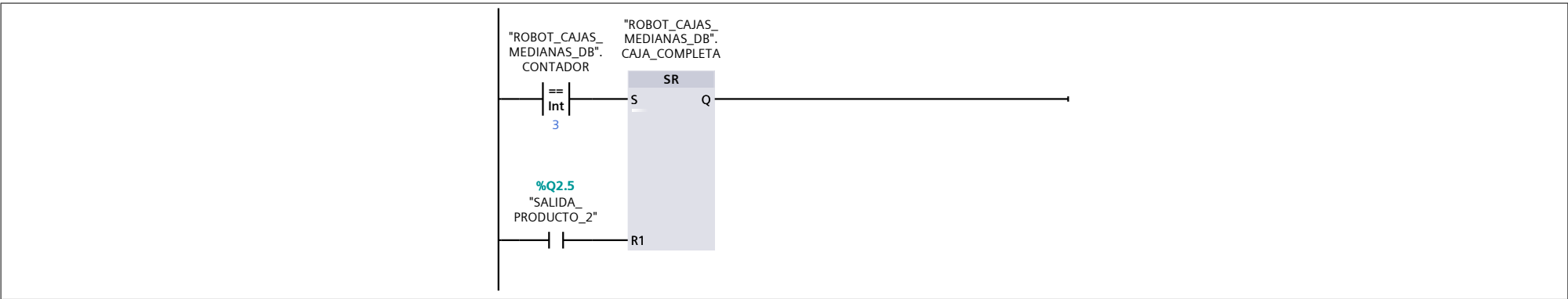
Segmento 4: MARCHA CONVEYOR



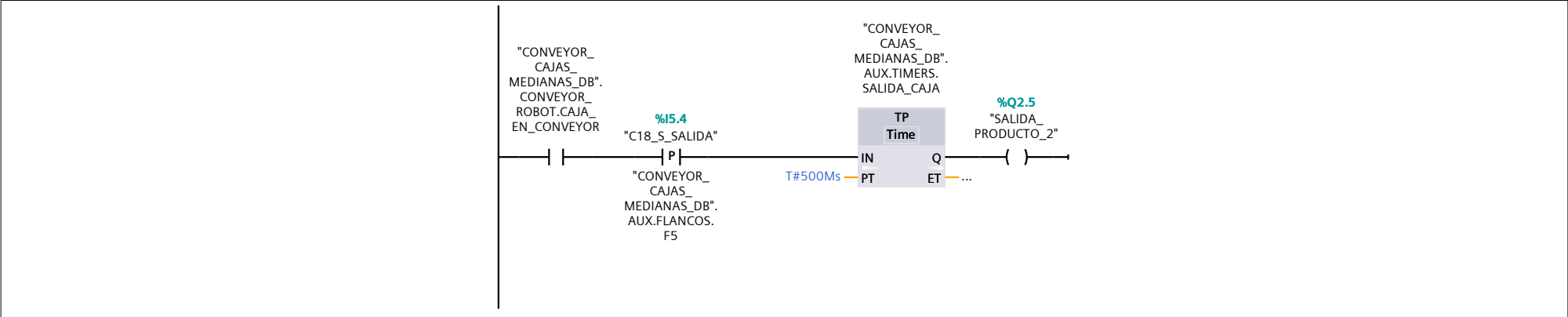
Segmento 5: CAJA EN CINTA



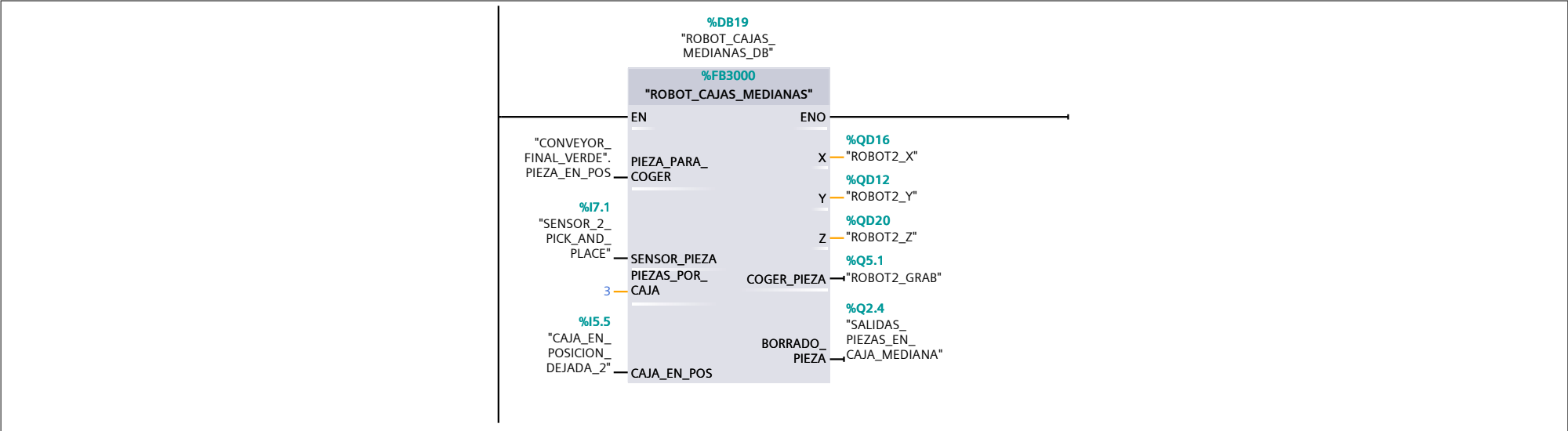
Segmento 6: CAJA LLENA/VACIA



Segmento 7: BORRADO DE PIEZA EN CAJA

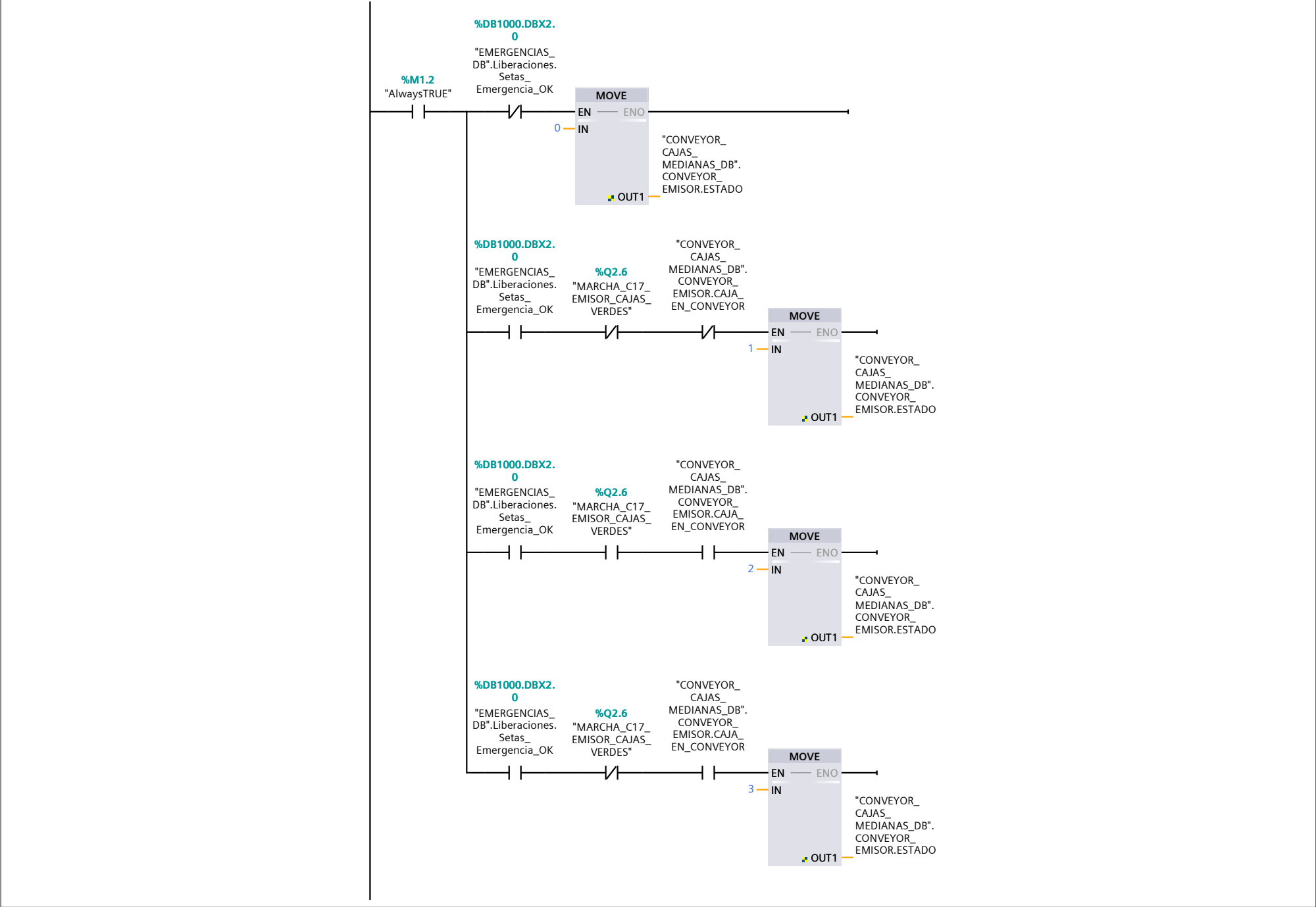


Segmento 8: FB ROBOT



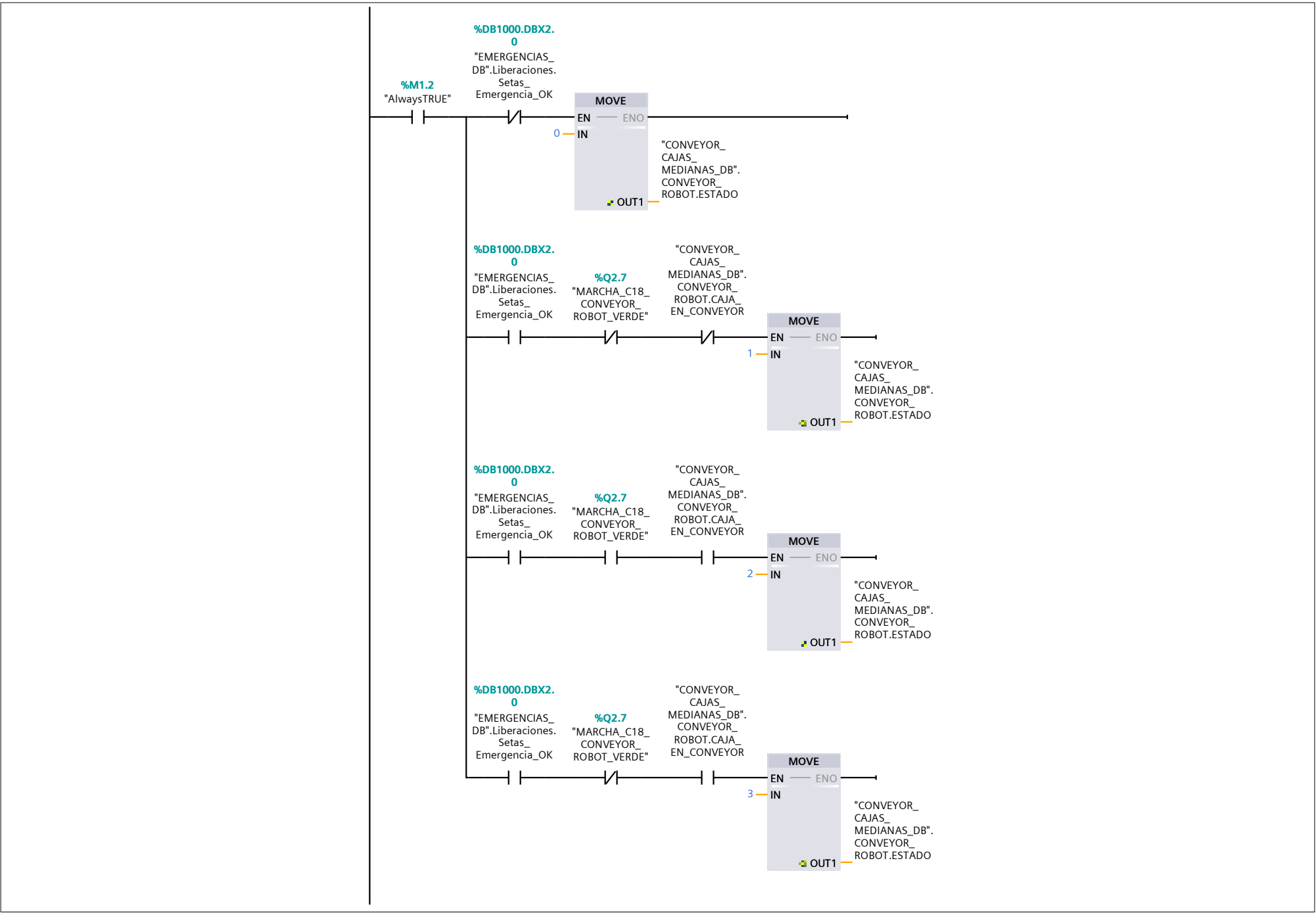
Segmento 9: ESTADO - CONVEYOR EMISOR CAJAS

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza
- =3 En marcha con pieza
- =4 Esperando



Segmento 10: ESTADO - CONVEYOR SALIDA A ROBOT

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza
- =3 En marcha con pieza
- =4 Esperando



Totally Integrated Automation Portal

Bloques de programa / 07_CONVEYOR_CAJAS_MEDIANAS

CONVEYOR_CAJAS_MEDIANAS_DB [DB2050]

CONVEYOR_CAJAS_MEDIANAS_DB Propiedades

General

Nombre	CONVEYOR_CAJAS_MEDIANAS_DB	Número	2050	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
▼ CONVEYOR_EMITOR	Struct		False	True	True	True	False		
CAJA_EN_CONVEYOR	Bool	false	False	True	True	True	False		
ESTADO	Int	0	False	True	True	True	False		
▼ CONVEYOR_ROBOT	Struct		False	True	True	True	False		
CAJA_EN_CONVEYOR	Bool	false	False	True	True	True	False		
PIEZAS_EN_CAJA	Int	0	False	True	True	True	False		
SET-POINT_PIEZAS	Int	3	False	True	True	True	False		
CAJA_LLENA	Bool	false	False	True	True	True	False		
▼ CONTADOR_PIEZAS	IEC_COUNTER		False	True	True	True	False		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		
PV	Int	0	False	True	True	True	False		
CV	Int	0	False	True	True	True	False		
POS_ROBOT_COGER	Bool	false	False	True	True	True	False		
POS_ROBOT_DEJAR	Bool	false	False	True	True	True	False		
COGER_PIEZA	Bool	false	False	True	True	True	False		
DEJAR_PIEZA	Bool	false	False	True	True	True	False		
ESTADO	Int	0	False	True	True	True	False		
▼ AUX	Struct		False	True	True	True	False		
▼ FLANCOS	Struct		False	True	True	True	False		
F1	Bool	false	False	True	True	True	False		
F2	Bool	false	False	True	True	True	False		
F3	Bool	false	False	True	True	True	False		
F4	Bool	false	False	True	True	True	False		
F5	Bool	false	False	True	True	True	False		
F6	Bool	false	False	True	True	True	False		
▼ TIMERS	Struct		False	True	True	True	False		
▼ T1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_3	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_4	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

Totally Integrated Automation Portal									
Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_5	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_6	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ SALIDA_CAJA	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ BORRADO_PIEZA	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

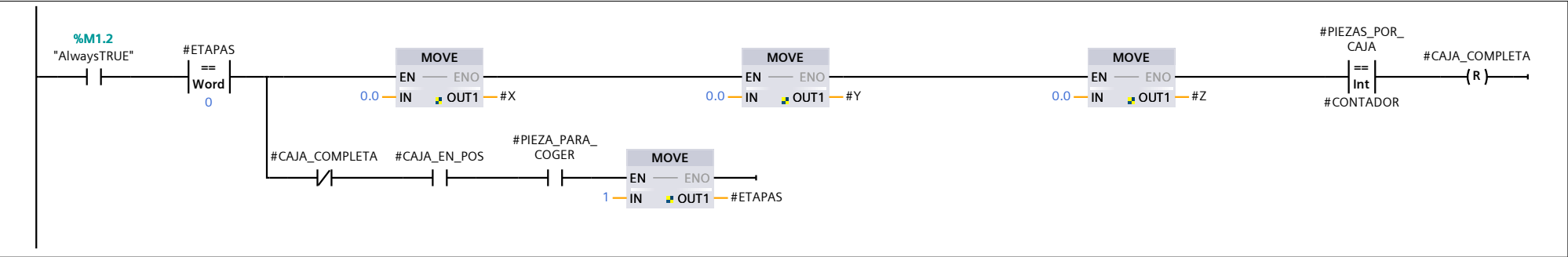
Bloques de programa / 07_CONVEYOR_CAJAS_MEDIANAS

ROBOT_CAJAS_MEDIANAS [FB3000]

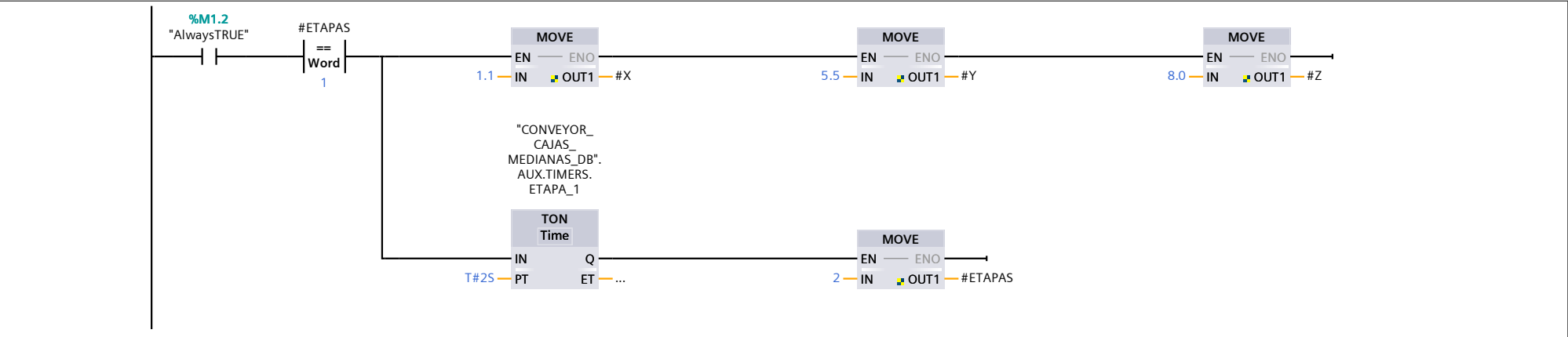
ROBOT_CAJAS_MEDIANAS Propiedades							
General							
Nombre	ROBOT_CAJAS_MEDIANAS	Número	3000	Tipo	FB	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible desde HMI/OPC UA	Es-cribi-ble desde HMI/OPC UA	Visible en HMI Engi-neering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
PIEZA_PARA_COGER	Bool	false	No remanente	True	True	True	False		
SENSOR_PIEZA	Bool	false	No remanente	True	True	True	False		
PIEZAS_POR_CAJA	Int	0	No remanente	True	True	True	False		
CAJA_EN_POS	Bool	false	No remanente	True	True	True	False		
▼ Output									
X	Real	0.0	No remanente	True	True	True	False		
Y	Real	0.0	No remanente	True	True	True	False		
Z	Real	0.0	No remanente	True	True	True	False		
COGER_PIEZA	Bool	false	No remanente	True	True	True	False		
BORRADO_PIEZA	Bool	false	No remanente	True	True	True	False		
InOut									
▼ Static									
ETAPAS	Word	16#0	No remanente	True	True	True	False		
CONTADOR	Int	0	No remanente	True	True	True	False		
CAJA_COMPLETA	Bool	false	No remanente	True	True	True	False		
▼ AUX	Struct		Remanente	True	True	True	False		
▼ FLANCOS	Struct		Remanente	True	True	True	False		
F1	Bool	false	Remanente	True	True	True	False		
▼ TIMERS	Struct		Remanente	True	True	True	False		
▼ T1	IEC_TIMER		Remanente	True	True	True	False		
PT	Time	T#0ms	Remanente	True	True	True	False		
ET	Time	T#0ms	Remanente	True	False	True	False		
IN	Bool	false	Remanente	True	True	True	False		
Q	Bool	false	Remanente	True	False	True	False		
Temp									
Constant									

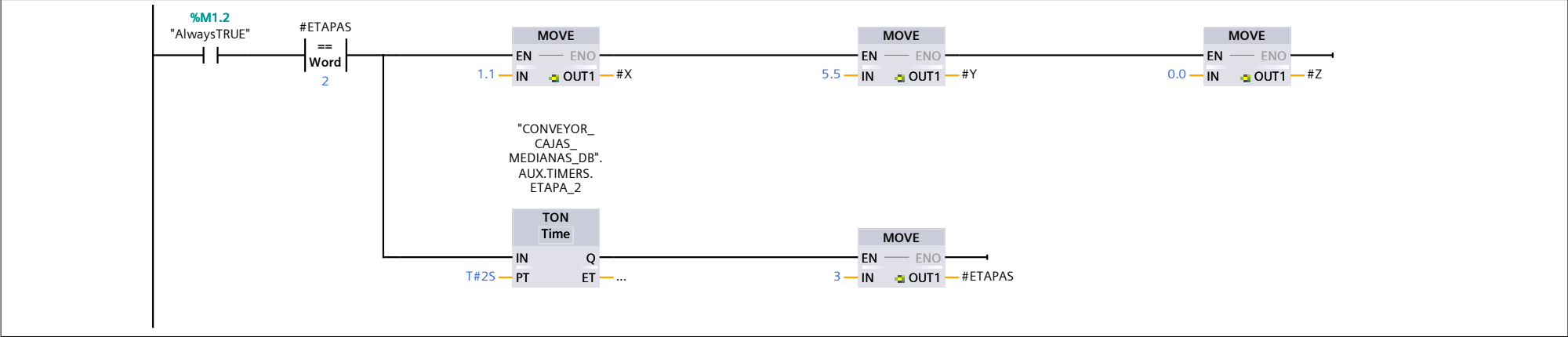
Segmento 1: ETAPA - REPOSO



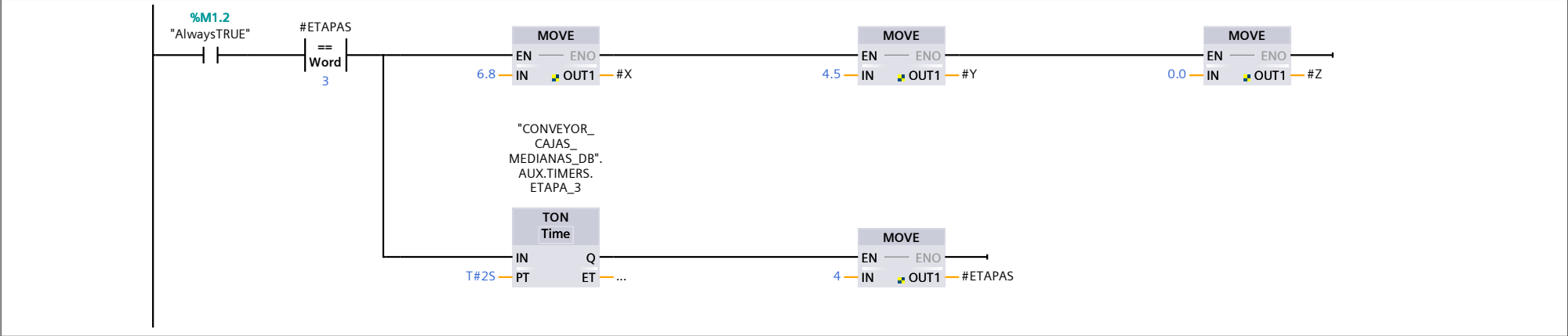
Segmento 2: ETAPA 1 ----> POSICION COGER PIEZA



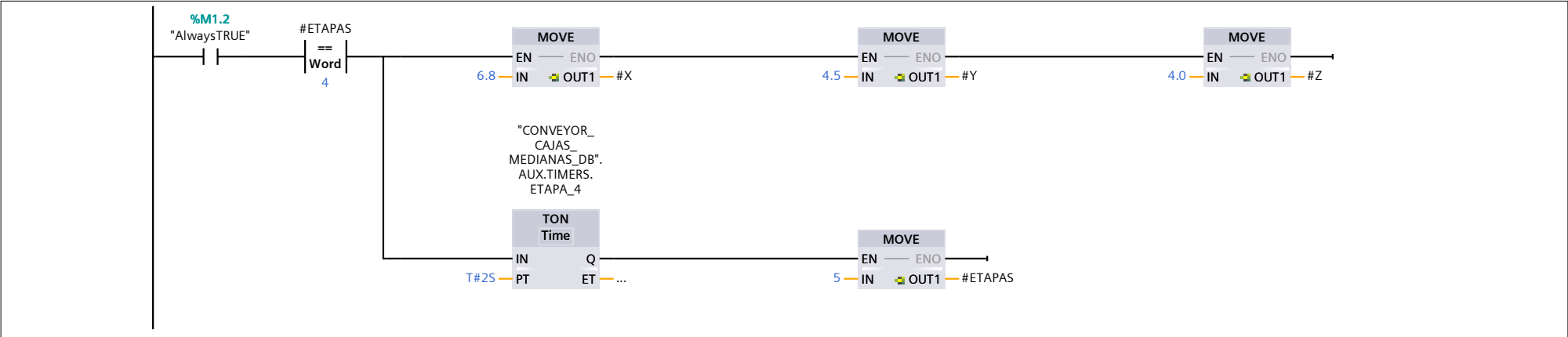
Segmento 3: ETAPA 2 ---> LEVANTAMOS PIEZA



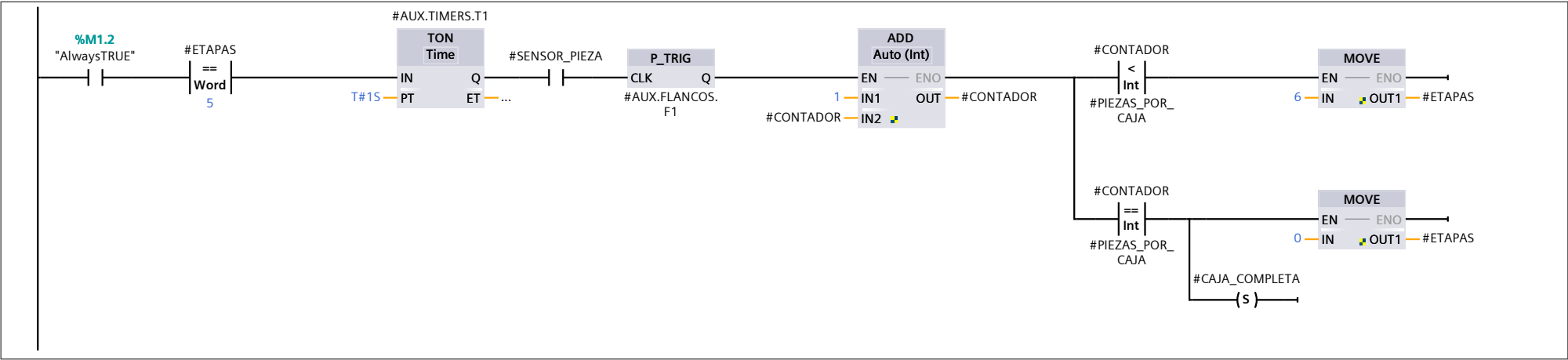
Segmento 4: ETAPA 3 ---> MOVEMOS A PRE DEJADA



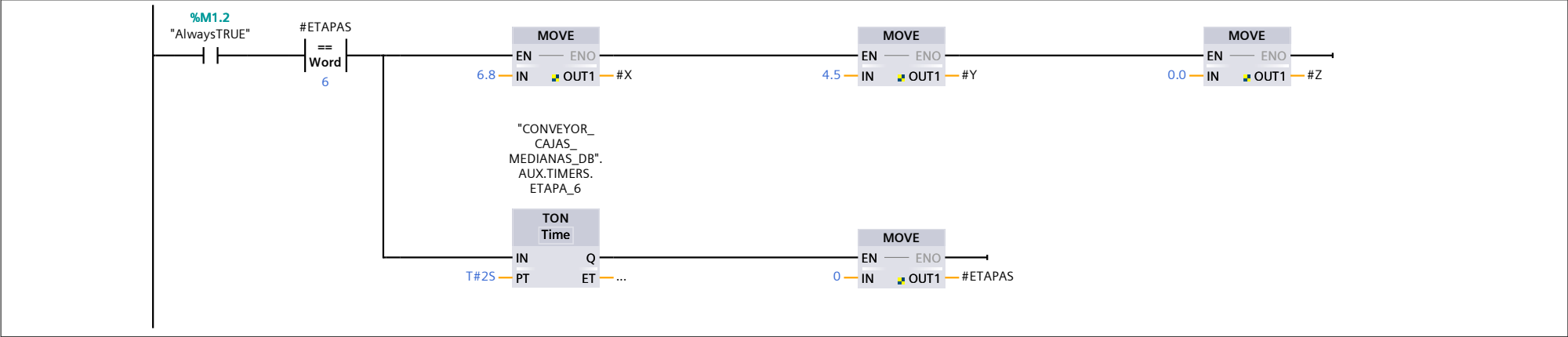
Segmento 5: ETAPA 4-DEJO PIEZA



Segmento 6: ETAPA 5--> COMPROBACION CAJA



Segmento 7: ETAPA 6--> RETORNO A PRE-DEJADA



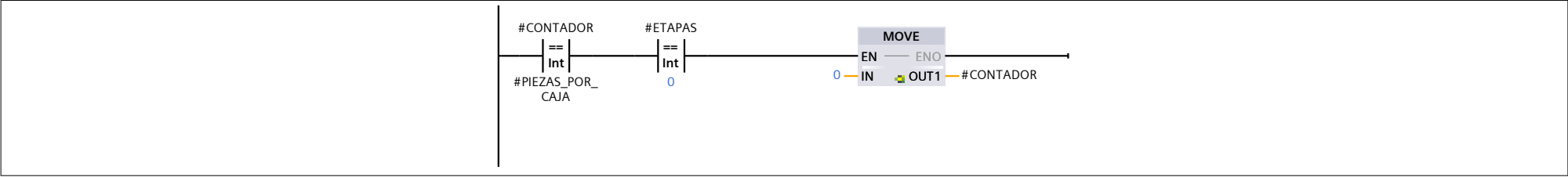
Segmento 8: COGER PIEZA



Segmento 9: BORRAR PIEZA



Segmento 10: Contador



Totally Integrated Automation Portal

Bloques de programa / 07_CONVEYOR_CAJAS_MEDIANAS

ROBOT_CAJAS_MEDIANAS_DB [DB19]

ROBOT_CAJAS_MEDIANAS_DB Propiedades

General

Nombre	ROBOT_CAJAS_MEDIANAS_DB	Número	19	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
PIEZA_PARA_COGER	Bool	false	False	True	True	True	False		
SENSOR_PIEZA	Bool	false	False	True	True	True	False		
PIEZAS_POR_CAJA	Int	0	False	True	True	True	False		
CAJA_EN_POS	Bool	false	False	True	True	True	False		
▼ Output									
X	Real	0.0	False	True	True	True	False		
Y	Real	0.0	False	True	True	True	False		
Z	Real	0.0	False	True	True	True	False		
COGER_PIEZA	Bool	false	False	True	True	True	False		
BORRADO_PIEZA	Bool	false	False	True	True	True	False		
InOut									
▼ Static									
ETAPAS	Word	16#0	False	True	True	True	False		
CONTADOR	Int	0	False	True	True	True	False		
CAJA_COMPLETA	Bool	false	False	True	True	True	False		
▼ AUX	Struct		True	True	True	True	False		
▼ FLANCOS	Struct		True	True	True	True	False		
F1	Bool	false	True	True	True	True	False		
▼ TIMERS	Struct		True	True	True	True	False		
▼ T1	IEC_TIMER		True	True	True	True	False		
PT	Time	T#0ms	True	True	True	True	False		
ET	Time	T#0ms	True	True	False	True	False		
IN	Bool	false	True	True	True	True	False		
Q	Bool	false	True	True	False	True	False		

Bloques de programa / 08_CONVEYOR_CAJAS_PEQUEÑAS

CONVEYOR_CAJAS_PEQUEÑAS [FC2060]

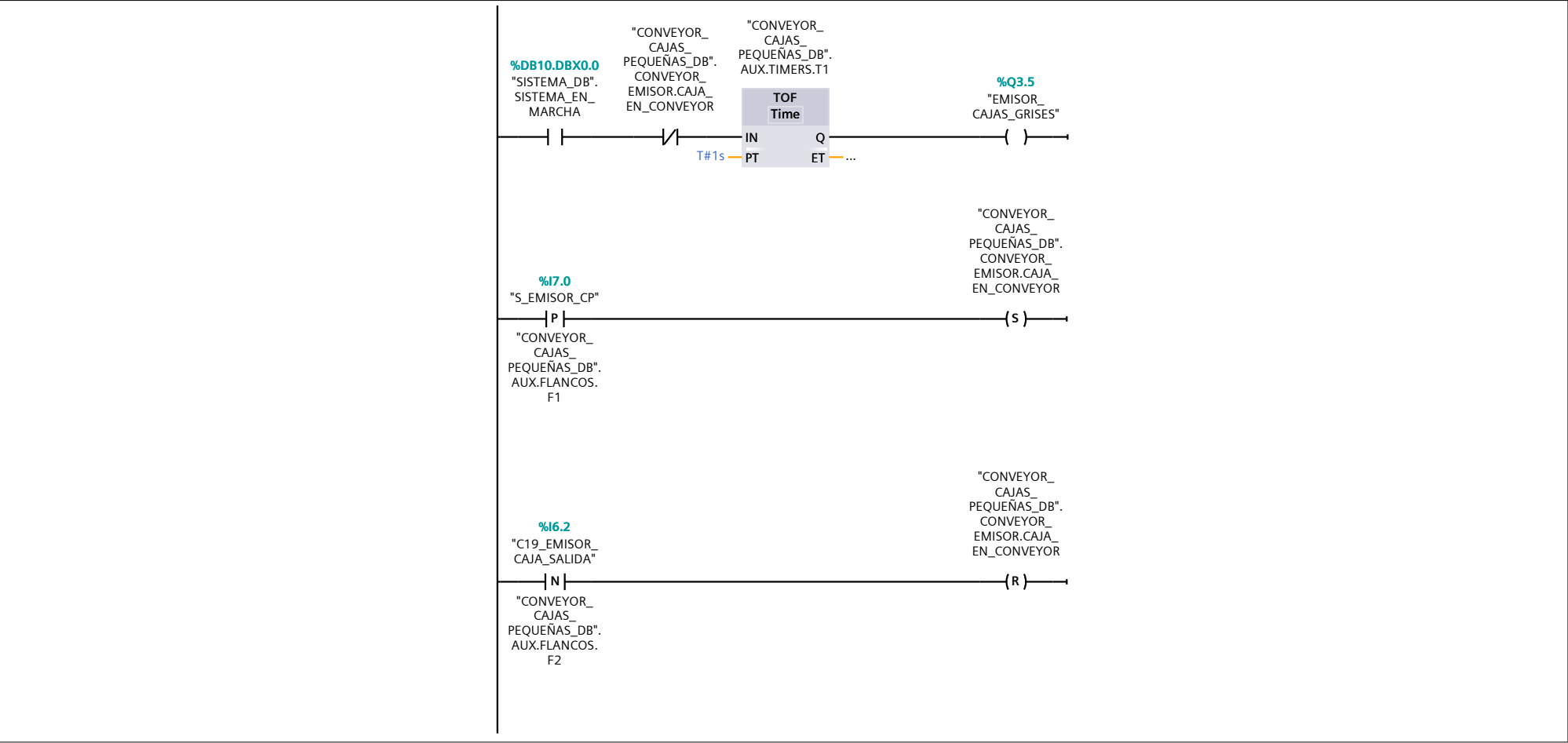
CONVEYOR_CAJAS_PEQUEÑAS Propiedades							
General							
Nombre	CONVEYOR_CAJAS_PEQUEÑAS	Número	2060	Tipo	FC	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
CONVEYOR_CAJAS_PEQUEÑAS	Void			

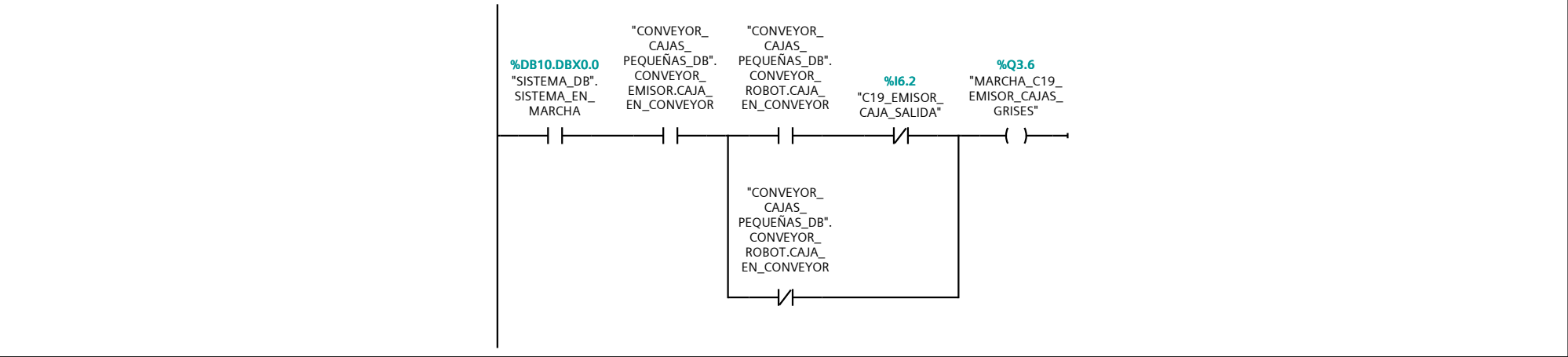
Segmento 1: ----- CONVEYOR_EMITOR CAJAS -----



Segmento 2: EMISOR CAJAS



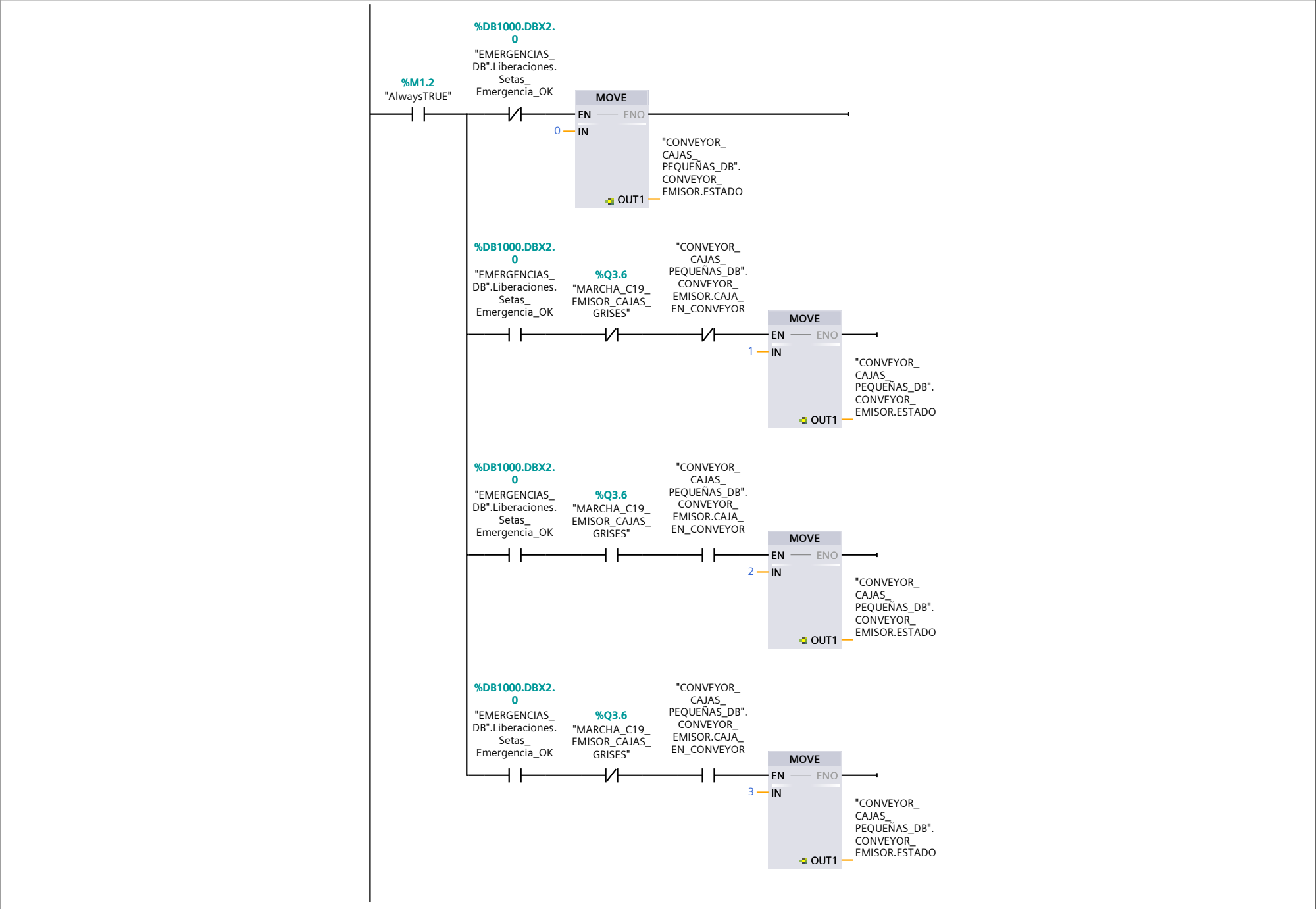
Segmento 3: MARCHA CONVEYOR CAJAS



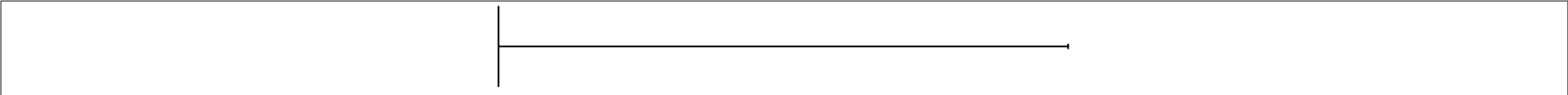
Segmento 4: ESTADO - CONVEYOR EMISOR CAJAS

=0 Fallo

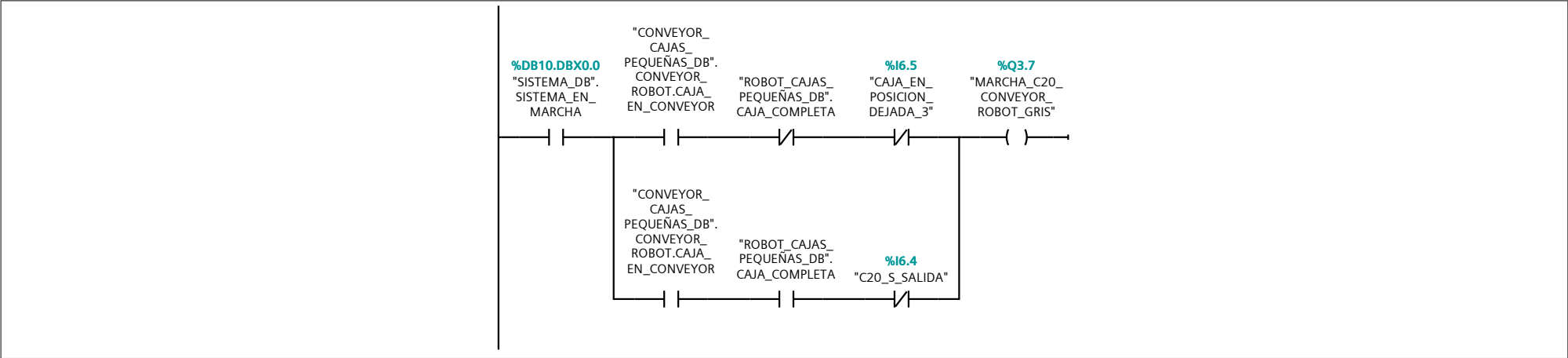
=1 OK en reposo
=2 Sin pieza
=3 En marcha con pieza
=4 Esperando



Segmento 5: ----- CONVEYOR_ROBOT -----

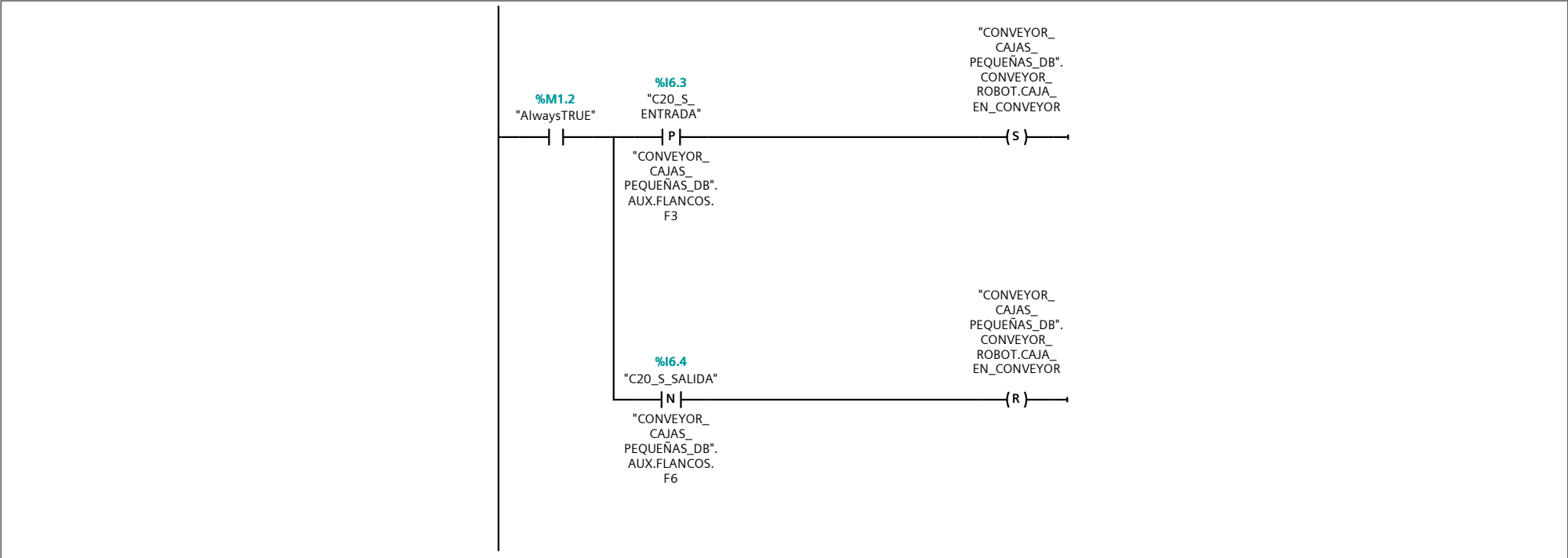


Segmento 6: MARCHA CONVEYOR ROBOT

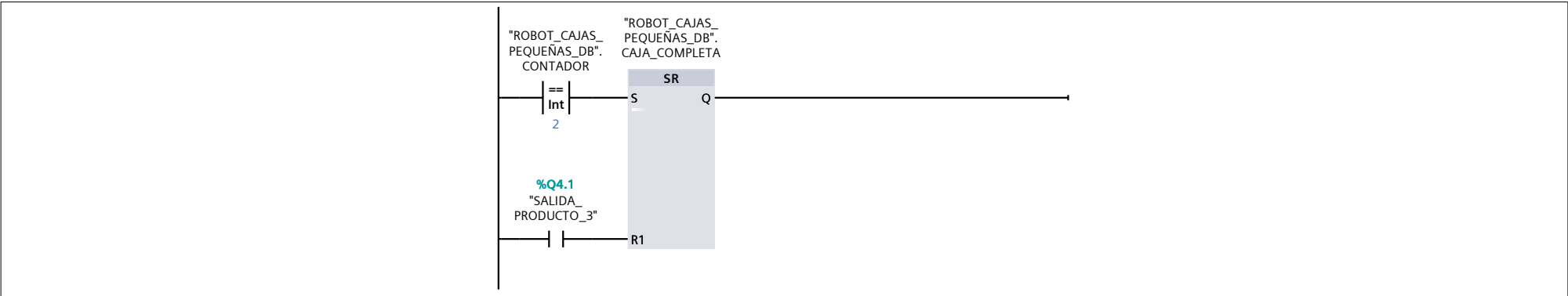


Segmento 7: CAJA EN CINTA

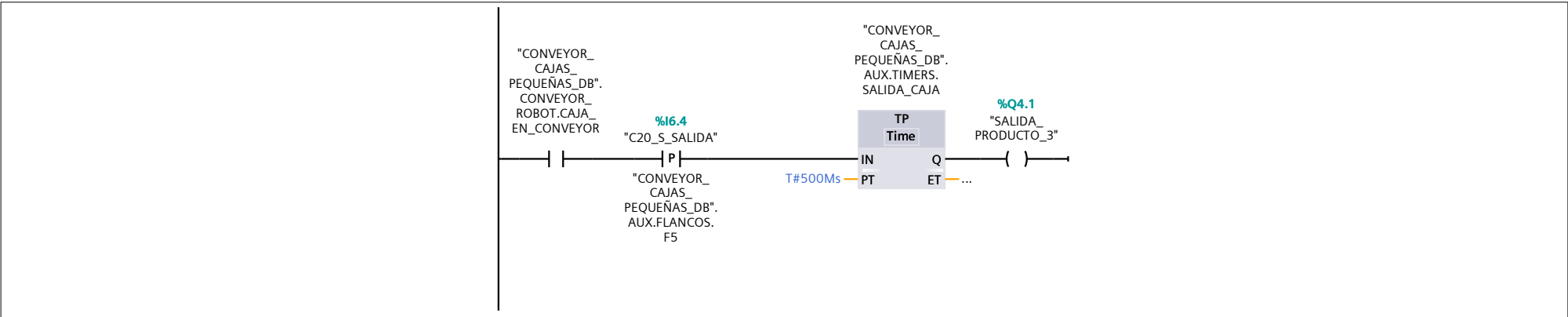




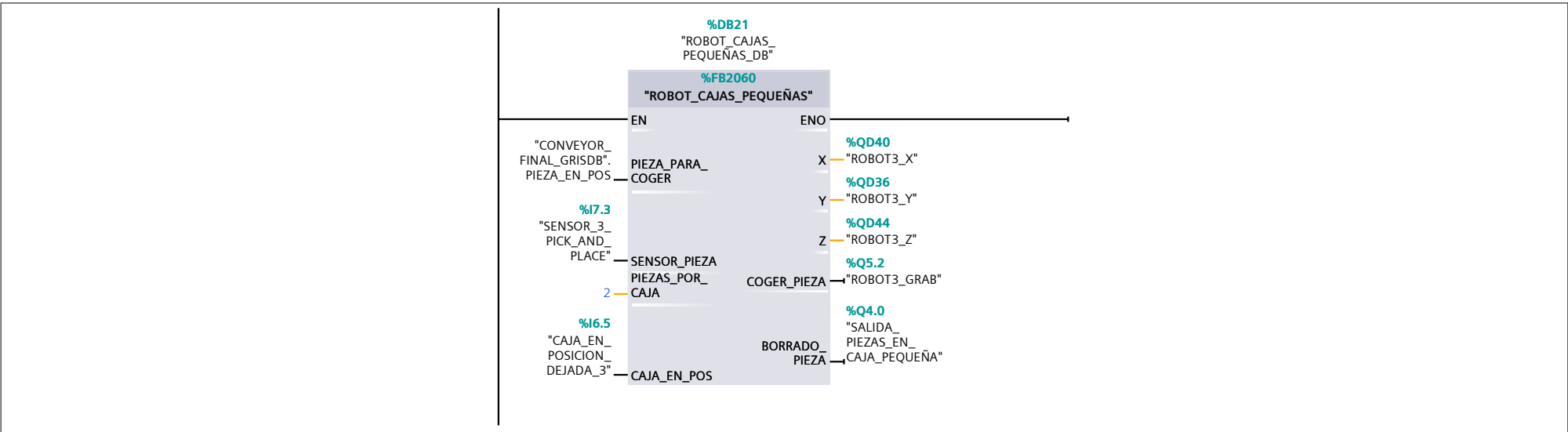
Segmento 8: CAJA LLENA/VACIA



Segmento 9: BORRADO DE CAJA

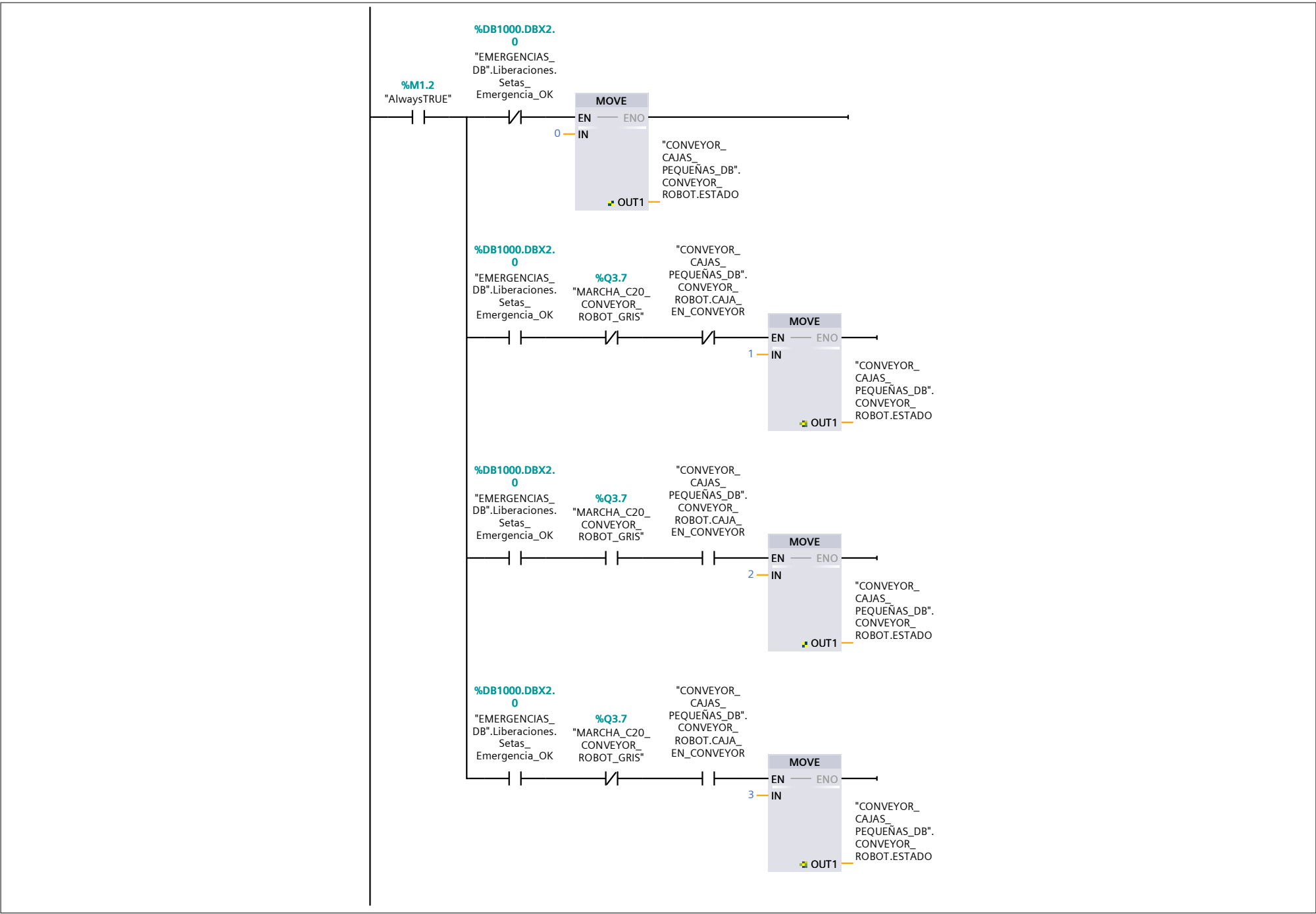


Segmento 10: FB ROBOT



Segmento 11: ESTADO - CONVEYOR SALIDA A ROBOT

- =0 Fallo
- =1 OK en reposo
- =2 Sin pieza
- =3 En marcha con pieza
- =4 Esperando



Totally Integrated Automation Portal

Bloques de programa / 08_CONVEYOR_CAJAS_PEQUEÑAS

CONVEYOR_CAJAS_PEQUEÑAS_DB [DB2060]

CONVEYOR_CAJAS_PEQUEÑAS_DB Propiedades

General

Nombre	CONVEYOR_CAJAS_PEQUEÑAS_DB	Número	2060	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanen- cia	Accesible desde HMI/OPC UA	Es- cribi- ble desd e HMI/ OPC UA	Visible en HMI Engi- neering	Valor de ajuste	Supervi- sión	Comentario
▼ Static									
▼ CONVEYOR_EMITOR	Struct		False	True	True	True	False		
CAJA_EN_CONVEYOR	Bool	false	False	True	True	True	False		
ESTADO	Int	0	False	True	True	True	False		
▼ CONVEYOR_ROBOT	Struct		False	True	True	True	False		
CAJA_EN_CONVEYOR	Bool	false	False	True	True	True	False		
PIEZAS_EN_CAJA	Bool	false	False	True	True	True	False		
SET-POINT_PIEZAS	Int	2	False	True	True	True	False		
CAJA_LLENA	Bool	false	False	True	True	True	False		
▼ CONTADOR_PIEZAS	IEC_COUNTER		False	True	True	True	False		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		
PV	Int	0	False	True	True	True	False		
CV	Int	0	False	True	True	True	False		
POS_ROBOT_COGER	Bool	false	False	True	True	True	False		
POS_ROBOT_DEJAR	Bool	false	False	True	True	True	False		
COGER_PIEZA	Bool	false	False	True	True	True	False		
DEJAR_PIEZA	Bool	false	False	True	True	True	False		
ESTADO	Int	0	False	True	True	True	False		
▼ AUX	Struct		False	True	True	True	False		
▼ FLANCOS	Struct		False	True	True	True	False		
F1	Bool	false	False	True	True	True	False		
F2	Bool	false	False	True	True	True	False		
F3	Bool	false	False	True	True	True	False		
F4	Bool	false	False	True	True	True	False		
F5	Bool	false	False	True	True	True	False		
F6	Bool	false	False	True	True	True	False		
▼ TIMERS	Struct		False	True	True	True	False		
▼ T1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_2	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_3	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_4	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

Nombre	Tipo de datos	Valor de arranque	Remanen- cia	Accesible desde HMI/OPC UA	Es- cribi- ble desd e HMI/ OPC UA	Visible en HMI Engi- neering	Valor de ajuste	Supervi- sión	Comentario
Q	Bool	false	False	True	False	True	False		
▼ ETAPA_6	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ SALIDA_CAJA	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ BORRADO_PIEZA	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

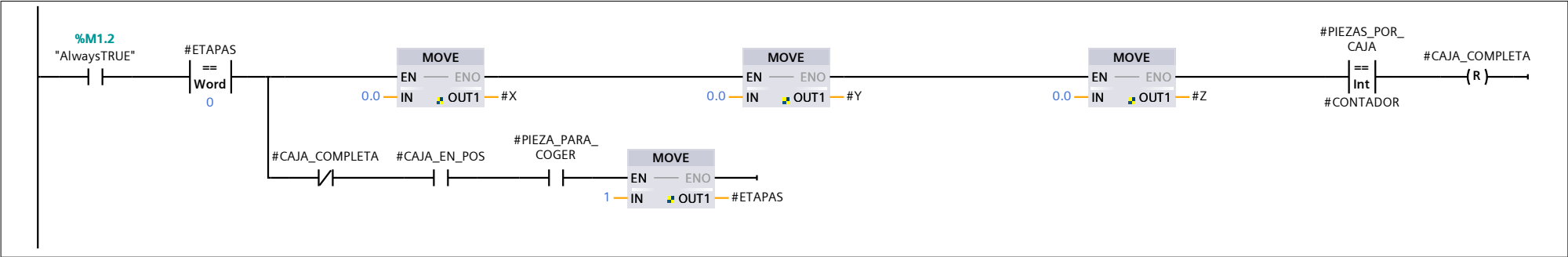
Bloques de programa / 08_CONVEYOR_CAJAS_PEQUEÑAS

ROBOT_CAJAS_PEQUEÑAS [FB2060]

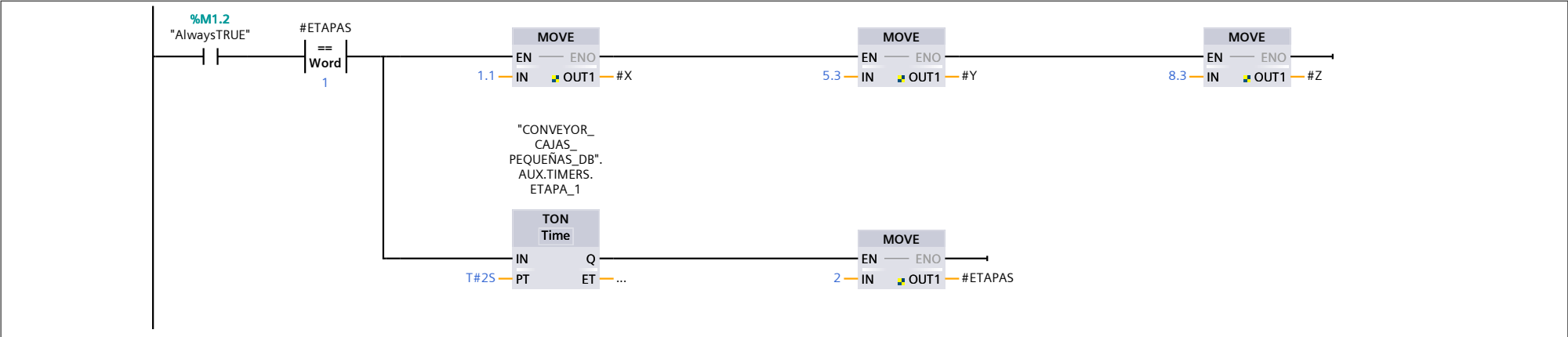
ROBOT_CAJAS_PEQUEÑAS Propiedades							
General							
Nombre	ROBOT_CAJAS_PEQUEÑAS	Número	2060	Tipo	FB	Idioma	KOP
Numeración	Manual						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible desde HMI/OPC UA	Es-cribi-ble desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervi-sión	Comentario
▼ Input									
PIEZA_PARA_COGER	Bool	false	No remanente	True	True	True	False		
SENSOR_PIEZA	Bool	false	No remanente	True	True	True	False		
PIEZAS_POR_CAJA	Int	0	No remanente	True	True	True	False		
CAJA_EN_POS	Bool	false	No remanente	True	True	True	False		
▼ Output									
X	Real	0.0	No remanente	True	True	True	False		
Y	Real	0.0	No remanente	True	True	True	False		
Z	Real	0.0	No remanente	True	True	True	False		
COGER_PIEZA	Bool	false	No remanente	True	True	True	False		
BORRADO_PIEZA	Bool	false	No remanente	True	True	True	False		
InOut									
▼ Static									
ETAPAS	Word	16#0	No remanente	True	True	True	False		
CONTADOR	Int	0	No remanente	True	True	True	False		
CAJA_COMPLETA	Bool	false	No remanente	True	True	True	False		
▼ AUX	Struct		No remanente	True	True	True	False		
▼ FLANCOS	Struct		No remanente	True	True	True	False		
F1	Bool	false	No remanente	True	True	True	False		
▼ TIMERS	Struct		No remanente	True	True	True	False		
▼ T1	IEC_TIMER		No remanente	True	True	True	False		
PT	Time	T#0ms	No remanente	True	True	True	False		
ET	Time	T#0ms	No remanente	True	False	True	False		
IN	Bool	false	No remanente	True	True	True	False		
Q	Bool	false	No remanente	True	False	True	False		
Temp									
Constant									

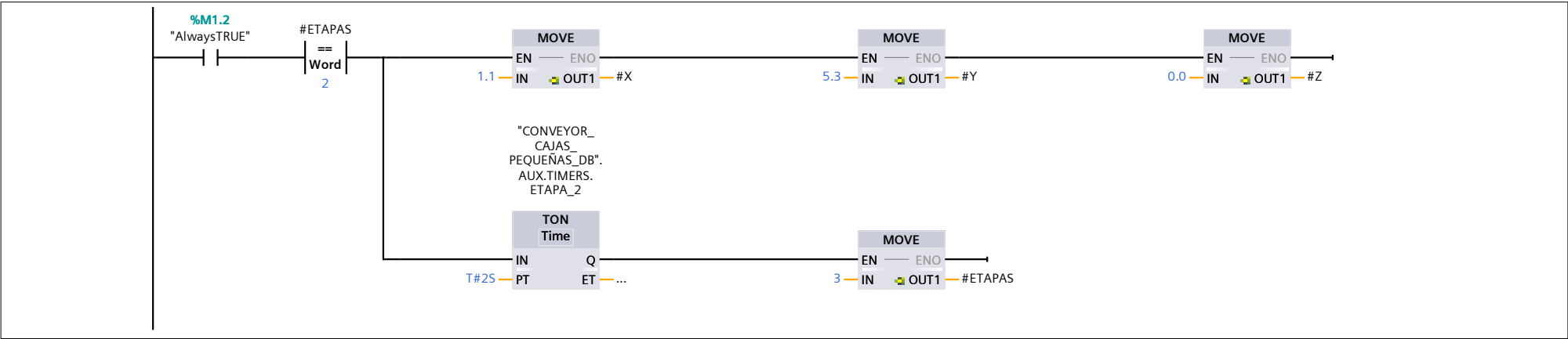
Segmento 1: ETAPA - REPOSO



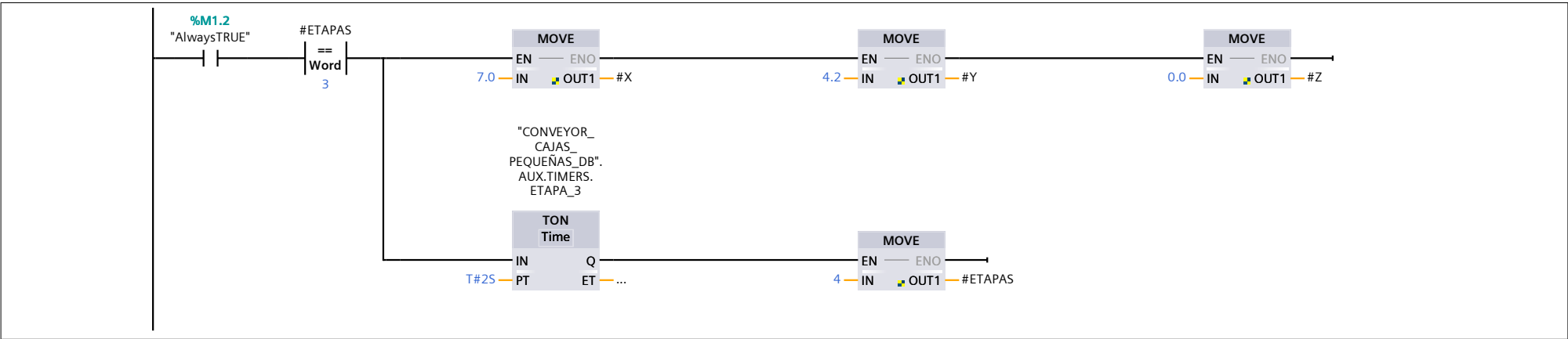
Segmento 2: ETAPA 1 ---> POSICION COGER PIEZA



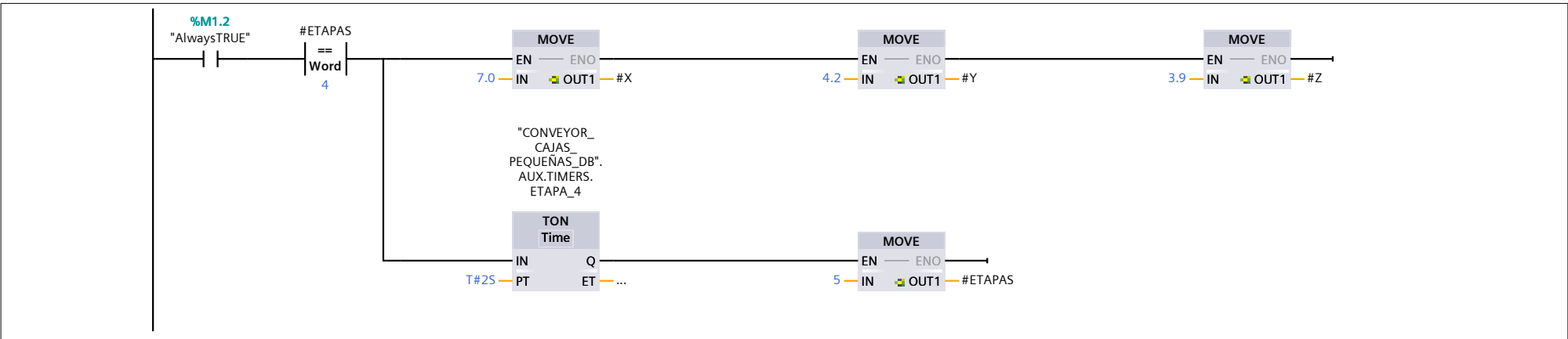
Segmento 3: ETAPA 2 ---> LEVANTAMOS PIEZA - POST COGIDA PIEZA



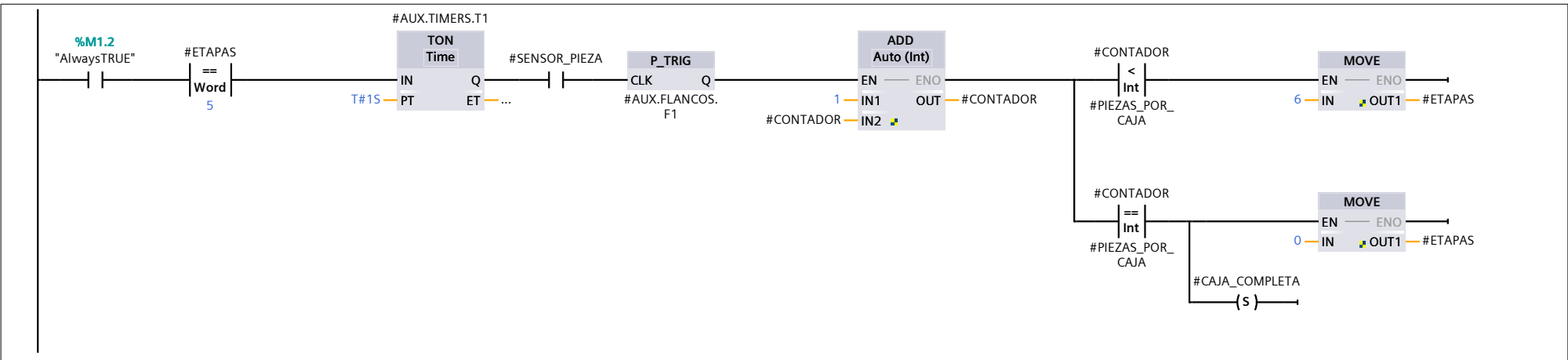
Segmento 4: ETAPA 3 ---> MOVEMOS A PRE DEJADA



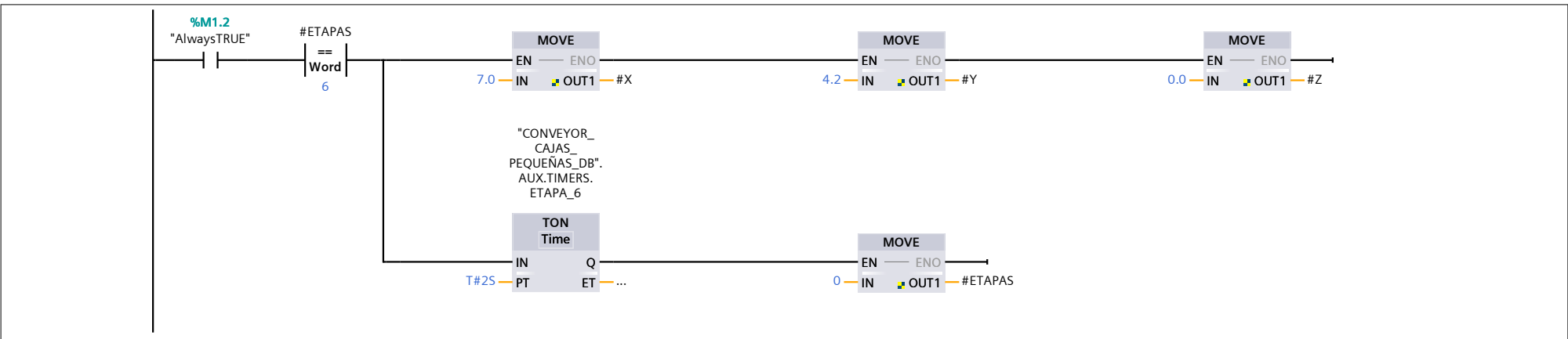
Segmento 5: ETAPA 4 ---> DEJO PIEZA



Segmento 6: ETAPA 5: COMPROBAICON CAJAS



Segmento 7: ETAPA 6: RETORNO A PRE-DEJADA



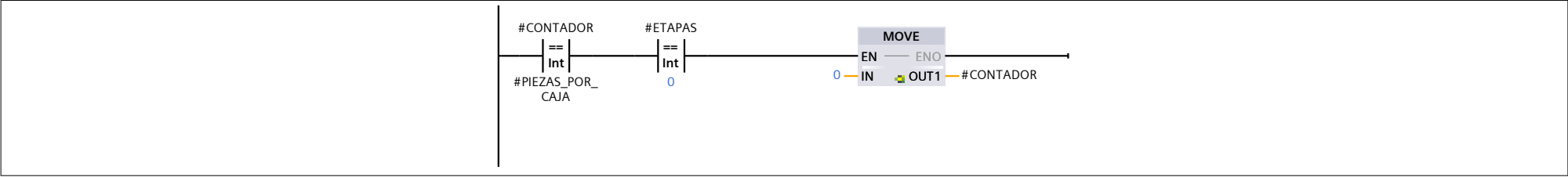
Segmento 8: COGER PIEZA



Segmento 9: BORRAR PIEZA



Segmento 10: Contador



Totally Integrated Automation Portal

Bloques de programa / 08_CONVEYOR_CAJAS_PEQUEÑAS

ROBOT_CAJAS_PEQUEÑAS_DB [DB21]

ROBOT_CAJAS_PEQUEÑAS_DB Propiedades

General

Nombre	ROBOT_CAJAS_PEQUEÑAS_DB	Número	21	Tipo	DB	Idioma	DB
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA	Escribible desde HMI/OPC UA	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Input									
PIEZA_PARA_COGER	Bool	false	False	True	True	True	False		
SENSOR_PIEZA	Bool	false	False	True	True	True	False		
PIEZAS_POR_CAJA	Int	0	False	True	True	True	False		
CAJA_EN_POS	Bool	false	False	True	True	True	False		
▼ Output									
X	Real	0.0	False	True	True	True	False		
Y	Real	0.0	False	True	True	True	False		
Z	Real	0.0	False	True	True	True	False		
COGER_PIEZA	Bool	false	False	True	True	True	False		
BORRADO_PIEZA	Bool	false	False	True	True	True	False		
InOut									
▼ Static									
ETAPAS	Word	16#0	False	True	True	True	False		
CONTADOR	Int	0	False	True	True	True	False		
CAJA_COMPLETA	Bool	false	False	True	True	True	False		
▼ AUX	Struct		False	True	True	True	False		
▼ FLANCOS	Struct		False	True	True	True	False		
F1	Bool	false	False	True	True	True	False		
▼ TIMERS	Struct		False	True	True	True	False		
▼ T1	IEC_TIMER		False	True	True	True	False		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Totally Integrated Automation Portal

Bloques de programa / 09_LED

LED [FC11]

LED Propiedades

General

Nombre	LED	Número	11	Tipo	FC	Idioma	KOP
Numeración	Manual						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizada					

Nombre	Tipo de datos	Valor predet.	Supervisión	Comentario
Input				
Output				
InOut				
Temp				
Constant				
▼ Return				
LED	Void			

Segmento 1: BALIZA

Segmento 2: BALIZA VERDE

%DB10.DBX0.0
"SISTEMA_DB".
SISTEMA_EN_
MARCHA

%Q8.5
"VERDE_BALIZA_
PRINCIPAL"

Segmento 3: BALIZA NARANJA

%DB10.DBX0.1
"SISTEMA_DB".
SISTEMA_
PARADO

%M0.4
"Clock_1.25Hz"

%Q8.4
"NARANJA_
BALIZA_
PRINCIPAL"

Segmento 4: BALIZA ROJO

%DB1000.DBX2.
0
"EMERGENCIAS_
DB".Liberaciones.
Setas_
Emergencia_OK

%Q8.3
"ROJO_BALIZA_
PRINCIPAL"

Segmento 6: LED MARCHA

%DB10.DBX0.0
"SISTEMA_DB".
SISTEMA_EN_
MARCHA

%Q8.0
"LUZ_BOTON_
MARCHA_
CUADRO_
PRINCIPAL"

Segmento 7: LED PARO

%DB10.DBX0.1
"SISTEMA_DB".
SISTEMA_
PARADO

%Q8.1
"LUZ_BOTON_
PARO_CUADRO_
PRINCIPAL"

Segmento 8: LED REARME

```
graph LR; I1["%DB1000.DBX2.0  
\"EMERGENCIAS_DB\"  
Setas_Emergencia_OK"] --- I2["%M0.6  
\"Clock_0.625Hz\""]; I1 --- I3["%Q8.2  
\"LUZ_BOTON_REARME_CUADRO_PRINCIPAL\""]; I2 --- I3; I3 --- O1["%Q8.2  
\"LUZ_BOTON_REARME_CUADRO_PRINCIPAL\""];
```

The diagram illustrates the logic for Segment 8, LED REARME. It features a single network with three inputs and one output. The inputs are:

- %DB1000.DBX2.0**: Labeled "EMERGENCIAS_DB" and "Setas_Emergencia_OK".
- %M0.6**: Labeled "Clock_0.625Hz".
- %Q8.2**: Labeled "LUZ_BOTON_REARME_CUADRO_PRINCIPAL".

The output is **%Q8.2**, labeled "LUZ_BOTON_REARME_CUADRO_PRINCIPAL". The logic is represented by a horizontal line with three normally open contacts in series, leading to the output coil.

