



GIEIA

Automatización de un proceso de clasificación, procesado y paletizado de materias primas mediante autómatas Schneider M241 y software de simulación Factory I/O

Nombre del alumno: Joaquín Talavera Llamas

Nombre del tutor: Raúl Simarro Fernández

Fecha: Julio de 2020

Grado en Ingeniería Electrónica Industrial y Automática



Índice de documentos

- 1. MEMORIA**
- 2. PROGRAMACIÓN DEL AUTOMATISMO**
- 3. ANEXO LISTADO VARIABLES**
- 4. PRESUPUESTO**

Agradecimientos:

Quería dedicar este pequeño espacio al principio del trabajo para agradecer a todas las personas que han estado a mi lado ya sea durante todo el grado o más allá de él.

Dar las gracias a mis padres por darme siempre todo lo que he necesitado. Por siempre anteponerme a ellos y desearme lo mejor cada día.

Dar las gracias a mi hermana por crecer conmigo y compartir momentos de alegría y cariño cuando los he necesitado.

Dar las gracias a mi pareja por demostrar el afecto y el apoyo que se da a una persona incluso hasta cuando no se da cuenta.

Y, por último y no por ello menos importante, dar las gracias a mis amigos por todos los momentos, viajes y experiencias que hemos pasado juntos.



GIEIA

DOCUMENTO 1: MEMORIA

Automatización de un proceso de clasificación,
procesado y paletizado de materias primas
mediante autómatas Schneider M241 y software
de simulación Factory I/O

Nombre del alumno: Joaquín Talavera Llamas

Nombre del tutor: Raúl Simarro Fernández

Fecha: Julio de 2020

Grado en Ingeniería Electrónica Industrial y Automática

ÍNDICE

1. INTRODUCCION	13
1.1. OBJETIVO.....	13
1.2. ANTECEDENTES	13
1.3. NECESIDADES DEL CLIENTE	13
1.4. LEGISLACIÓN.....	14
2. ESTUDIO DE ALTERNATIVAS	14
2.1. AUTÓMATA	14
2.1.1. LENGUAJE DE PROGRAMACIÓN	16
2.2. SOFTWARE.....	19
3. DESARROLLO DE LA SOLUCIÓN	22
3.1. RECREACIÓN DE LA FÁBRICA.....	22
3.1.1. EMERGENCIA/ALARMA	34
3.1.2. PICK AND PLACE	35
3.1.3. ORGANIZADOR PESO.....	35
3.1.4. MÁQUINAS DE CONTROL NUMÉRICO (CNC)	36
3.1.5. PALLETIZADOR	37
3.1.6. PIEZA DEFECTUOSA	38
3.1.7. SIEMPRE ENCENDIDO	38
3.1.8. CONTADORES.....	39
3.1.9. IMPLEMENTACIÓN DEL AUTOMATISMO SOBRE PLC M241 DE SCHNEIDER	39

3.1.10. CONEXIÓN ENTRE FACTORY I/O Y AUTÓMATA	44
3.2. CONSULTA EN LA WEB.....	48
3.2.1. INTERFAZ DE USUARIO	48
4. CONCLUSIONES.....	51
5. BIBLIOGRAFÍA	52

ÍNDICE DE IMÁGENES

Imagen 1: Autómata Omron CP1-L.....	16
Imagen 2: Autómata Schneider M241.....	16
Imagen 3: Diagrama de contactos (LD).....	17
Imagen 4: Diagrama de bloques de funciones (FBD).....	18
Imagen 5: SFC.....	19
Imagen 6: Texto estructurado (ST).....	19
Imagen 7: Lista de instrucciones (IL).....	20
Imagen 8: Software SoMachine de Schneider.....	20
Imagen 9: Software PICS simulation tool.....	21
Imagen 10: Software Factory I/O.....	22
Imagen 11: Software Factory I/O layout conexiones sensores/actuadores.....	22
Imagen 12: Vista general de la fábrica.....	23
Imagen 13: Vista en planta de la fábrica.....	24
Imagen 14: Puntos relevantes de la factoría.....	24
Imagen 15: Inicio de recorrido en fábrica y generador de elementos.....	25
Imagen 16: Sensor de pesaje y rodillos giratorios.....	25
Imagen 17: Sistema de sujeción y sensor.....	26
Imagen 18: Cajas agrupadas de tres en tres.....	27
Imagen 19: Pallet cargado con seis cajas.....	27
Imagen 20: Pallet cargado con seis cajas hacia destino.....	28
Imagen 21: Sensor de piezas defectuosas con barrera.....	28
Imagen 22: Brazo activado con pieza defectuosa.....	29
Imagen 23: Pieza azul continuando trayectoria normal.....	29
Imagen 24: Pieza azul en sensor de color piezas buenas.....	30
Imagen 25: Pick and place.....	30
Imagen 26: Pick and place recogiendo pieza azul.....	31
Imagen 27: Pieza azul llegando a centro de procesado.....	31
Imagen 28: Pieza azul siendo estampada.....	32
Imagen 29: Pieza azul como producto final.....	32

Imagen 30: Pick and place recogiendo pieza verde.....	33
Imagen 31: Pieza verde llegando a centro de procesado.....	33
Imagen 32: Pieza verde siendo estampada.....	34
Imagen 33: Pieza verde como producto final.....	34
Imagen 34: Ejemplo SFC emergencia con “memoria”	35
Imagen 35: Crear proyecto es SoMachine.....	40
Imagen 36: Abrir configuración.....	41
Imagen 37: POUs de proyecto completo.....	41
Imagen 38: Descargar programa autómeta paso primero.....	42
Imagen 39: Descargar programa autómeta paso segundo.....	42
Imagen 40: Descargar programa autómeta paso tercero.....	43
Imagen 41: Descargar programa autómeta paso cuarto.....	43
Imagen 42: Aviso de Schneider al descargar proyecto autómeta.....	44
Imagen 43: Aviso de proyecto diferente en autómeta.....	44
Imagen 44: Aviso iniciar sesión con descarga.....	44
Imagen 45: Botón de inicio.....	45
Imagen 46: Factory I/O opción drivers.....	45
Imagen 47: Factory I/O layout bobinas.....	46
Imagen 48: Configuración de Factory I/O completa.....	47
Imagen 49: Variables globales en SoMachine.....	48
Imagen 50: Conexión correcta en Factory I/O.....	48
Imagen 51: Interfaz de usuario en SoMachine.....	50
Imagen 52: Propiedades de objeto interfaz de usuario.....	50
Imagen 53: Configuración de visualizador web.....	51
Imagen 54: SCADA del proyecto visto en Firefox.....	52

1. INTRODUCCION

1.1. OBJETIVO

El objetivo del proyecto es el diseño y la implementación de la automatización de un sistema de clasificación, paletizado y composición de PCBs (placas de circuito impreso) a través de un proceso de maquinado, utilizando para ello un autómatas y un software de simulación para crear un entorno controlado. Además, se incluye la posibilidad de comprobar en cualquier momento información relevante del proceso a través una aplicación de visualización e interacción con el usuario.

1.2. ANTECEDENTES

A lo largo de cursar el Grado de Ingeniería Electrónica Industrial y Automática se han tratado asignaturas relacionadas con automatismos (p. ej. Automática Básica y Automatización Industrial), sin embargo, los ensayos consistían en automatismos o procesos sencillos. Dicha cosa provocó la curiosidad de realizar un proyecto a mayor escala, tratándose de automatizar una fábrica entera con varios procesos, distintos materiales y productos finales.

Al contrario que diseñar en otro tipo de entornos, el hecho de realizarse una simulación muy similar a lo que sería el mismo proceso en funcionamiento resulta muy llamativo tanto para el ingeniero que realiza la programación del automatismo como para el cliente/espectador, ya permite validar la implementación del automatismo sobre el autómatas real con un sistema dinámico con los mismos elementos de entradas/salidas que la planta real.

1.3. NECESIDADES DEL CLIENTE

Como se trata de una factoría que existe en la realidad, el diseño de dicho automatismo debe realizarse tanto para que funcione en la simulación previa como en la puesta en marcha en dicha factoría. El cliente, por su parte, precisa de unos requisitos mínimos que deben de ser de obligado cumplimiento:

- ❖ Almacenamiento de datos en tiempo real y posibilidad de consulta a través de ordenador o smartphone.
- ❖ Interfaz de usuario simple e intuitiva.
- ❖ Comunicación entre autómatas y software de simulación para hacer pruebas y mantenimiento en entorno seguro.
- ❖ Conseguir los objetivos con un coste bajo.
- ❖ Sistema de seguridad que pare todos los procesos de ser necesario.
- ❖ Posibilidad de realizar uno de los procesos paso a paso de manera manual pulsando un botón.

- ❖ Discriminar entre un proceso u otro con ayuda del peso del material.
- ❖ Eliminar piezas defectuosas antes de comenzar el proceso deseado.
- ❖ Mantener producción actual constante o mejorarla.

1.4. LEGISLACIÓN

A continuación se indica la normativa a cumplir para la realización de proyectos en el ámbito de la automatización, la seguridad y las comunicaciones industriales:

- **UNE-EN IEC 61131:2019:** Estándar Internacional para controladores lógicos programables (PLC). [1]
- **UNE-EN 60848:2013:** Lenguaje de especificación GRAFCET para diagramas funcionales secuenciales. [1]
- **UNE-EN 61508:2011:** Seguridad funcional de los sistemas eléctricos/electrónicos/electrónicos programables relacionados con la seguridad. [1]
- **UNE-EN 60870-5-101:2003:** Norma para la monitorización de sistemas de energía, control y las comunicaciones asociadas a los mismos. [1]
- **UNE-EN ISO 13850:2016:** Seguridad de las máquinas. Función de parada de emergencia. Principios para el diseño. [1]
- **UNE-EN 62439:2018:** Redes de comunicación industrial. Redes de automatización de alta disponibilidad. [1]

2. ESTUDIO DE ALTERNATIVAS

2.1. AUTÓMATA

¿Por qué usar un autómata? Los autómatas programables son capaces de realizar tareas tales como el cómputo, cálculo y comparación de valores analógicos. Al mismo tiempo son capaces de controlar sistemas más complejos a la vez que mejoran la estabilidad, y por tanto fiabilidad, del sistema de control. [5]

Sin embargo, la parte que más destaca en el momento de justificar su uso es la flexibilidad que ofrecen a la hora de realizar modificaciones, añadir nuevos procesos e incluso retirar procesos antiguos o que ya no es necesaria su utilización en el sistema, a la vez que reducen el coste de mano de obra al convertir al operario en un elemento prescindible. [3]

Su instalación es bastante sencilla, lo cual contrasta bastante con poder utilizarlo, ya que se necesita a un trabajador cualificado para la programación y mantenimiento de dicho autómata a través de un ordenador. [8]

Los autómatas tenidos en cuenta para la realización del proyecto son los siguientes:

❖ Omron CJ2M



Imagen 1: Autómata Omron CJ2M

Se trata del autómata programable con el que el alumno estaba familiarizado ya que es el que se ha empleado para la parte práctica de la asignatura de automatización cursada durante el grado. Posee una cantidad considerable de E/S digitales, unidades de expansión y se comunica con el ordenador mediante el protocolo Ethernet TCP/IP. El programa que se utiliza para su uso y programación es CX-Programmer y es propio de Omron, el cual utilizan para todas sus series de PLC. [6]

❖ Schneider M241



Imagen 2: Autómata Schneider M241

Este autómata programable también se encuentra como material disponible en las aulas de laboratorio del departamento de Ingeniería de Sistemas y Automática (DISA), por lo tanto, el tutor propuso al alumno que considerase la posibilidad de utilizar dicho autómata. El autómata posee también puertos de comunicación como

Ethernet TCP/IP, una amplia gama de módulos de extensión de E/S analógicas y digitales y la posibilidad de utilizar un servidor web para alojar una aplicación de visualización con la que poder consultar información referente al programa en ejecución de manera remota. Cabe destacar la posibilidad de utilizar diferentes secciones de código o unidades de programación (POU's), lo que favorece la versatilidad del autómatas para muchas aplicaciones diferentes. El alumno ha elegido este autómatas para realizar el trabajo. [7]

2.1.1. LENGUAJE DE PROGRAMACIÓN

Una vez se ha elegido el autómatas se debe decidir el lenguaje de programación que se va a emplear. Existen dos tipos de lenguaje gráficos y textuales.

- **Lenguajes gráficos**

Dentro de los lenguajes de alto nivel se encuentran los lenguajes gráficos, los cuales utilizan símbolos para realizar la programación y son los más fáciles de comprender y usar por novatos o gente inexperta con automatismos. Los lenguajes gráficos son formados por:

- ❖ **Ladder (LD), diagrama de bloques de relés:**

Se trata de un lenguaje gráfico basado en el desarrollo de los anteriores PLC, el cual se realizaba con relés eléctricos. Permite ver el programa en ejecución, forzar variables a un valor deseado y es práctico para simular ciertos sistemas y gente inexperta en programación de autómatas. Cada escalón o rama, generalmente, termina con una bobina, la cual al cerrar todos los contactos que la preceden, cerrará su propio contacto en otra parte del circuito (si es que se encuentra en uso).

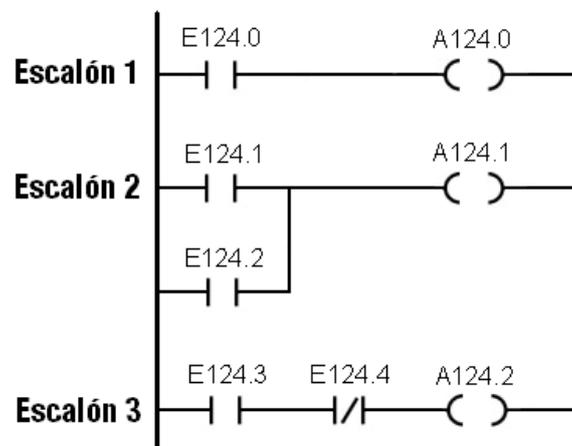


Imagen 3: Diagrama de contactos (LD)

❖ Diagrama de bloques de funciones (FBD):

Se trata de un lenguaje gráfico basado en funciones tales como AND, OR, menor que, etcétera. Su similitud con las puertas lógicas se puede apreciar a simple vista, por lo que es un lenguaje intuitivo apropiado para electrónicos. Como en la programación Ladder, existe la posibilidad de forzar variables a un valor deseado.

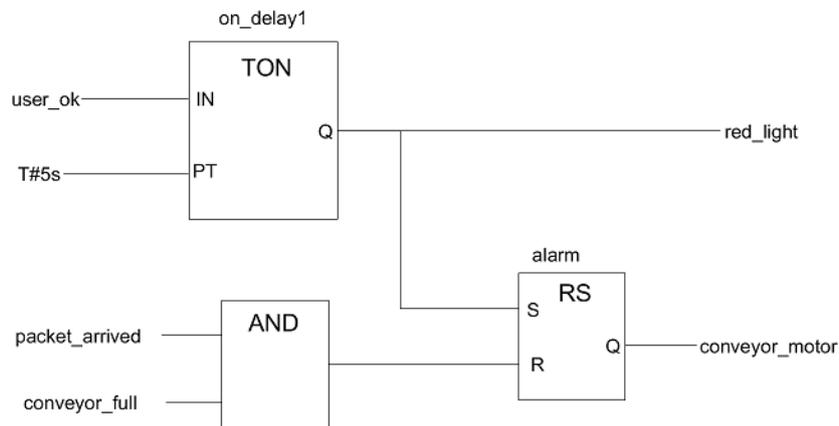


Imagen 4: Diagrama de bloques de funciones (FBD)

❖ Gráfico de funciones secuenciales (SFC):

Se trata de un diagrama funcional el cual está formado por transiciones, etapas y acciones asociadas. Las transiciones se definen como un guión que resulta ser la condición que debe de cumplirse para avanzar de una etapa a la siguiente. Este avance puede ser a la etapa posterior o a una anterior si el programa así lo requiere. Existe la posibilidad de tener una acción asociada a una etapa, la cual será la acción que realizará un actuador controlado por el autómatas mientras el proceso se encuentre dentro de dicha etapa, una vez la transición para avanzar a la siguiente etapa se cumpla, la acción dejará de ejecutarse.

Si el programador lo desea puede realizar un SFC del sistema completo todo compacto en un programa, como podría ser un sistema empotrado, o puede dividirlo en subprogramas para favorecer así su modularidad y adaptación a las nuevas necesidades del cliente. El alumno ha elegido este lenguaje para programar el proyecto. [4]

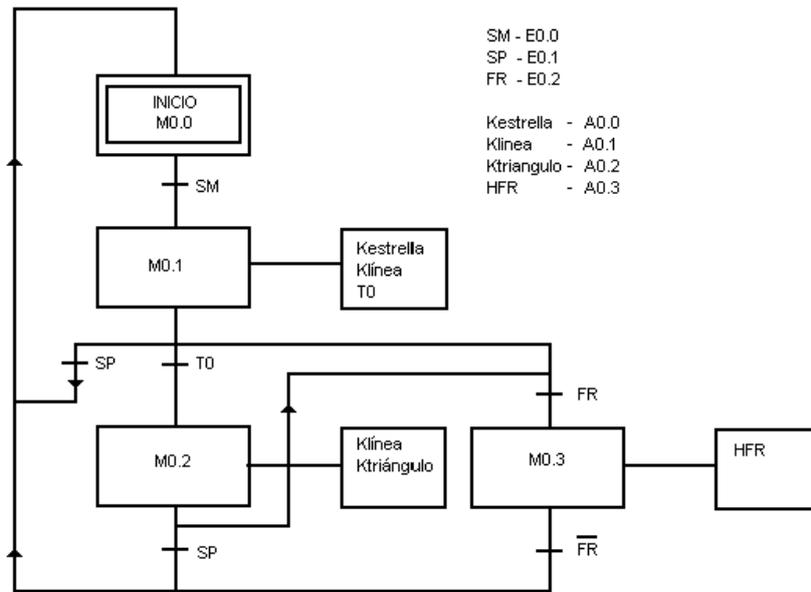


Imagen 5: Automatismo en SFC

▪ Lenguajes textuales

Dentro de los lenguajes de bajo nivel se encuentran los lenguajes textuales, los cuales utilizan cadenas de texto para realizar la programación y son más complejos de comprender debido a la necesidad de estar familiarizado con su estructura y significado de terminología y reglas de escritura. Los lenguajes textuales son formados por:

❖ Texto estructurado (ST):

Se trata de un lenguaje de programación de PLC o autómatas que permite realizar algoritmos complejos y mayor control que los lenguajes de programación gráficos a la vez que se asemeja bastante al lenguaje de programación "C". Gracias a esto, la gente relacionada con el mundo de la informática o electrónica tiene facilidad para leer y comprender rápidamente este lenguaje.

```

1 IF #start = 1 THEN
2     //comment
3     "Max_nr" := #Array[0];
4     FOR #i := 1 TO 10 DO
5         // Statement section FOR
6         IF #Array[#i] > "Max_nr" THEN
7             "Max_nr" := #Array[#i];
8         END_IF;
9     END_FOR;
10 END_IF;
11

```

Imagen 6: Texto estructurado (ST)

❖ Lista de instrucciones (IL):

Se trata de un lenguaje de programación de bajo nivel y se asemeja al lenguaje ensamblador. El programa utiliza saltos y llamadas a funciones para realizar el control del programa.

```
LD %I1.1
R   %C8
LD %I1.2
AND %M0
CU  %C8
LD %C8.D
ST  %Q2.0
```

Imagen 7: Lista de instrucciones (IL)

2.2. SOFTWARE

A la hora de programar el autómatas y construir una simulación de la fábrica real donde va a funcionar el programa, se han considerado diferentes opciones. Sin embargo, para programar el autómatas se ha empleado el software propio de Schneider llamado SoMachine para usar las funcionalidades completas y evitar problemas de compatibilidad.

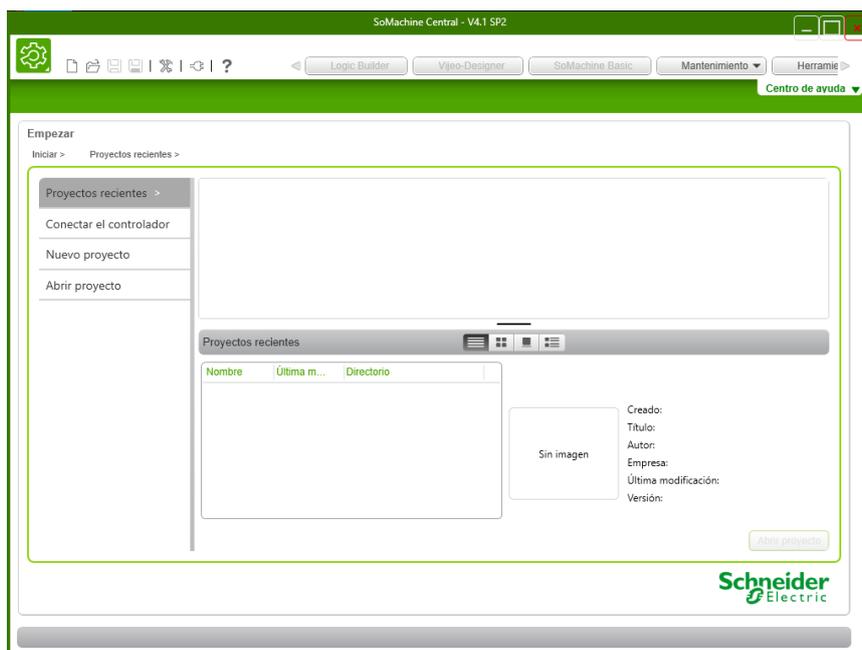


Imagen 8: Software SoMachine de Schneider

Dicho software también posee la posibilidad de realizar un servidor web nombrado anteriormente, el cual se desarrollará en el punto 3.3 de esta memoria (ver abajo).

Por otra parte, para simular la fábrica se ha barajado la posibilidad de utilizar varios softwares de simulación:

❖ PICS simulation tool

Es un software de simulación con años a la espalda, sin embargo, hace tiempo que no recibe actualizaciones de contenido, solamente de compatibilidad (como por ejemplo permitir su uso en Windows 7). PICS consiste en simular sistemas en tiempo real mediante la ayuda de autómatas o PLC, permitiendo al mismo tiempo el uso de SCADAs y uso de sistemas I/Os. Algunos de los usos más populares de este software es simulaciones para la industria de la automoción, alimentación o petroquímica. [9]

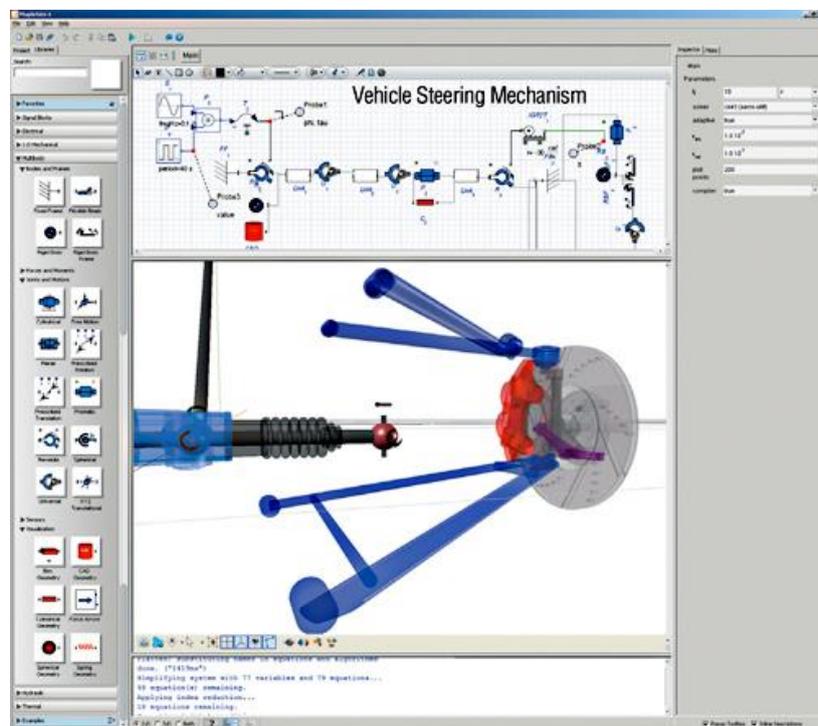


Imagen 9: Software PICS simulation tool

❖ Factory I/O

Se trata de un software de simulación novedoso en el mercado el cual se especializa en la simulación en tres dimensiones de factorías e industrias de manufactura como estampado, transporte y almacenaje de productos. Se utiliza comúnmente como plataforma de aprendizaje a usar autómatas ya que su entorno visual y sencillez hace que sea una gran herramienta para novatos e incluso avanzados.

El programa permite mover la cámara 360º y desplazarse libremente para poner atención o comprobar elementos concretos del sistema. También permite colocar, rotar y dar color a distintos elementos para una mejor organización del espacio. Es posible mostrar u ocultar los nombres de actuadores tales como un brazo Pick and Place, barreras o palancas, y nombres de sensores como sensores de presencia y color. [2]

Dado que el software permite la comunicación con el autómatas mediante comunicación industrial Modbus TCP/IP, el alumno ha elegido utilizarlo con el autómatas M241 de Schneider.

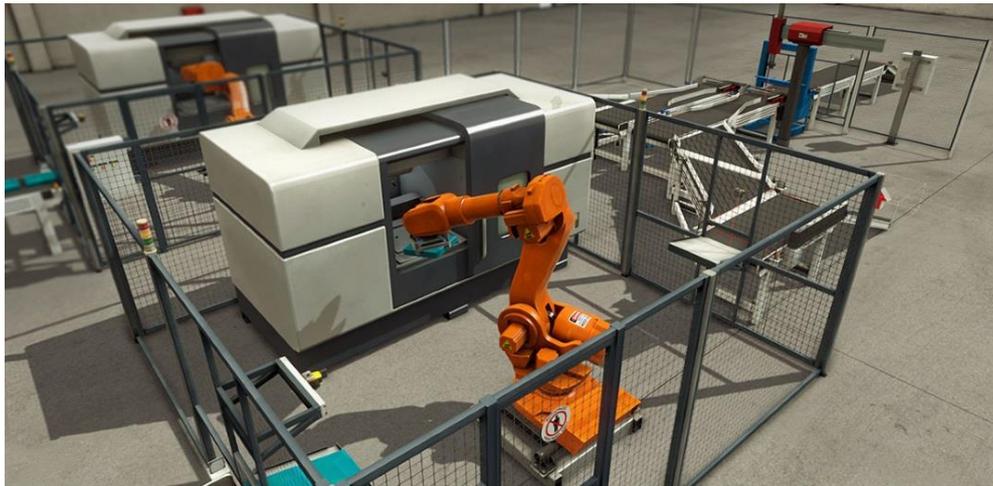


Imagen 10: Software Factory I/O

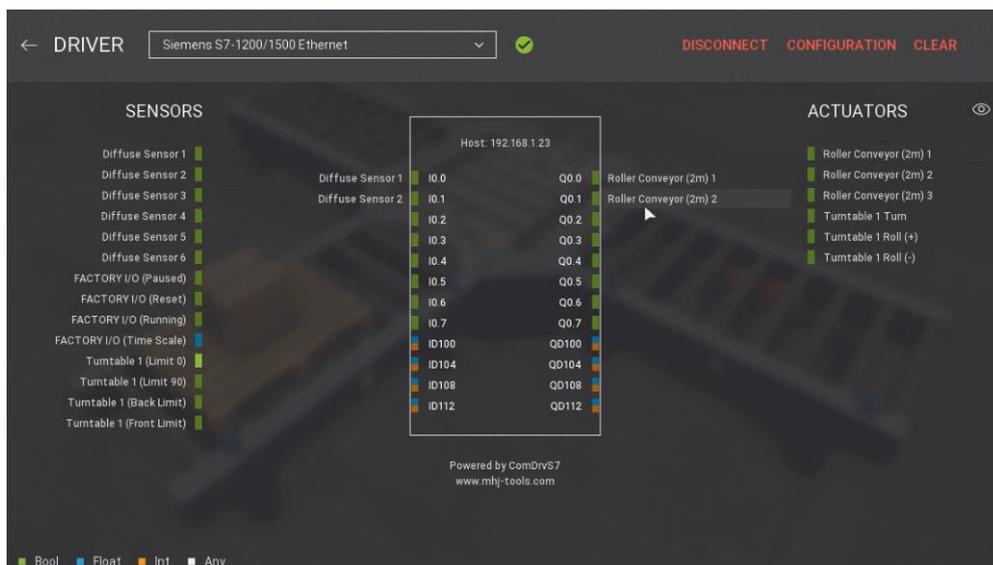


Imagen 11: Software Factory I/O, layout conexiones sensores/actuadores

3. DESARROLLO DE LA SOLUCIÓN

3.1. Recreación de la fábrica

La fábrica ha sido recreada en el software Factory I/O de acuerdo a lo que podría ser una nave real con la maquinaria incluida para realizar las tareas de organización, discriminación según tipo de materia prima, transporte y, por último, refinado y estampado del producto final. Un contador cuenta el número de piezas que llegan.

Se muestra una vista completa de la fábrica tanto de manera general como vista en planta y los puntos de la fábrica más relevantes en cuanto al diseño del automatismo:

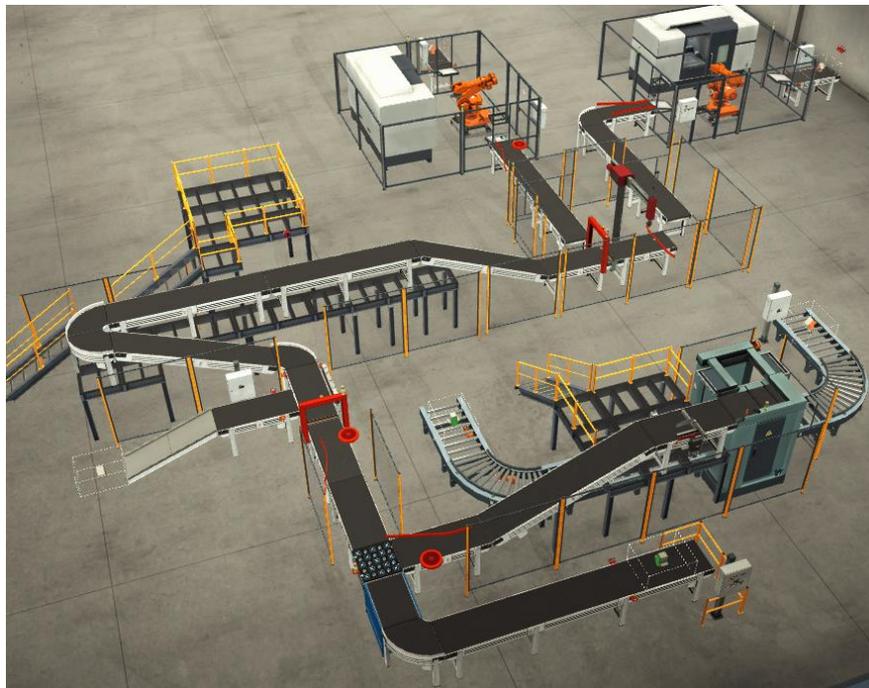


Imagen 12: Vista general de la fábrica



Imagen 13: Vista en planta de la fábrica



Imagen 14: Puntos relevantes de la factoría

La fábrica consta de un generador de PCBs y cajas de cartón y de un sistema de cintas transportadoras para llevar las materias primas que entran en el recorrido a un sensor de pesaje, el cual se encarga de separar en dos caminos con ayuda de unos rodillos giratorios.



Imagen 15: Inicio de recorrido en fábrica y generador de elementos

- ❖ Camino recto: piezas de colores que representan placas de circuito impreso (PCB) primitivas que serán llevadas posteriormente a estampación y procesado.
- ❖ Camino derecho: cajas de cartón las cuales representan materias primas que no van a ser tratadas en esta fábrica pero que la empresa se encarga de juntar en palés de 6 cajas para comodidad del destinatario.



Imagen 16: Sensor de pesaje y rodillos giratorios

Se procede a seguir en primer lugar a las cajas de cartón por el recorrido derecho, el cual es más breve que el camino recto:

Las cajas sufren un breve enderezamiento con ayuda de la barra fija y del rodillo rojos situados justo a la salida de los rodillos giratorios, ya que estos afectan a la orientación original de la caja.

A continuación, las cajas suben una cinta transformadora en forma de rampa ascendente y llegan a un sistema de sujeción o ajuste (*clamped*) el cual se encarga de poner la caja totalmente recta, con ayuda de un sensor que le avisa de que una caja se acerca. En cuanto esto ocurre el sistema de sujeción se levanta para que la caja de cartón pueda continuar avanzando por la cinta transportadora.

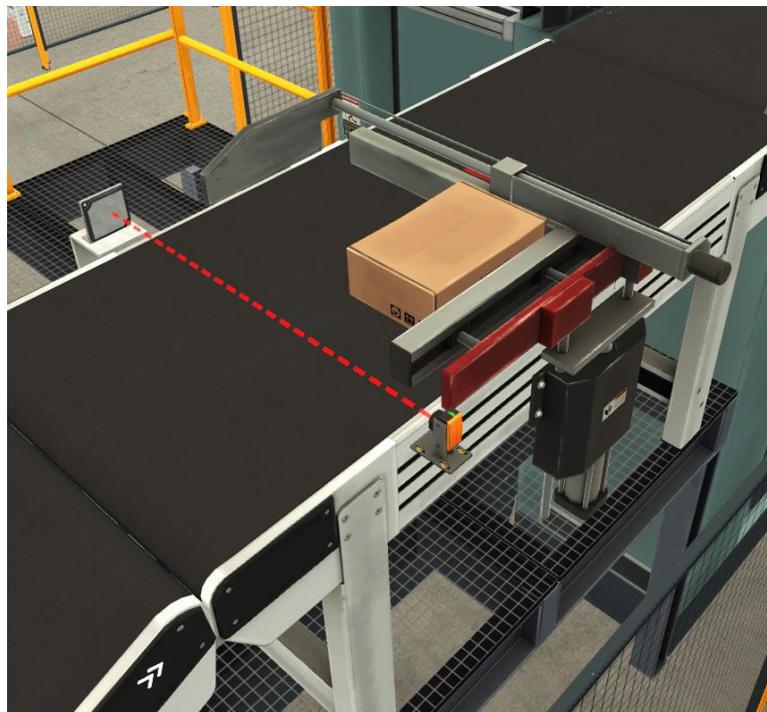


Imagen 17: Sistema de sujeción y sensor

Una vez la caja ya está orientada pasa a la máquina encargada de juntarlas en grupos de seis cajas para posteriormente ser cargadas encima de un pallet a través de una trampilla. Primero se ordenan juntas tres cajas y posteriormente tres más.



Imagen 18: Cajas agrupadas de tres en tres

Una vez el pallet está cargado con las seis cajas correspondientes, es llevado mediante un ascensor vertical y una cinta de rodillos hacia su destino final.

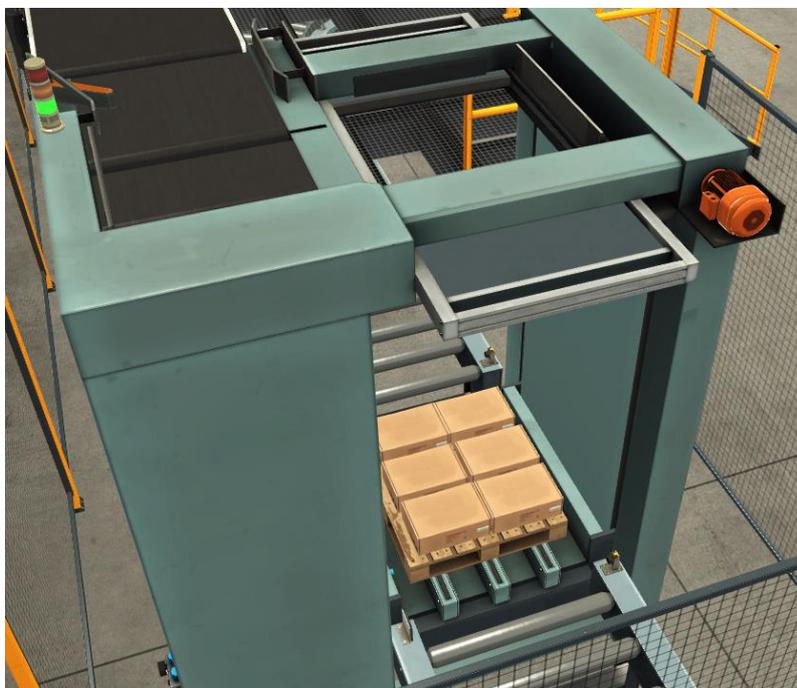


Imagen 19: Pallet cargado con seis cajas

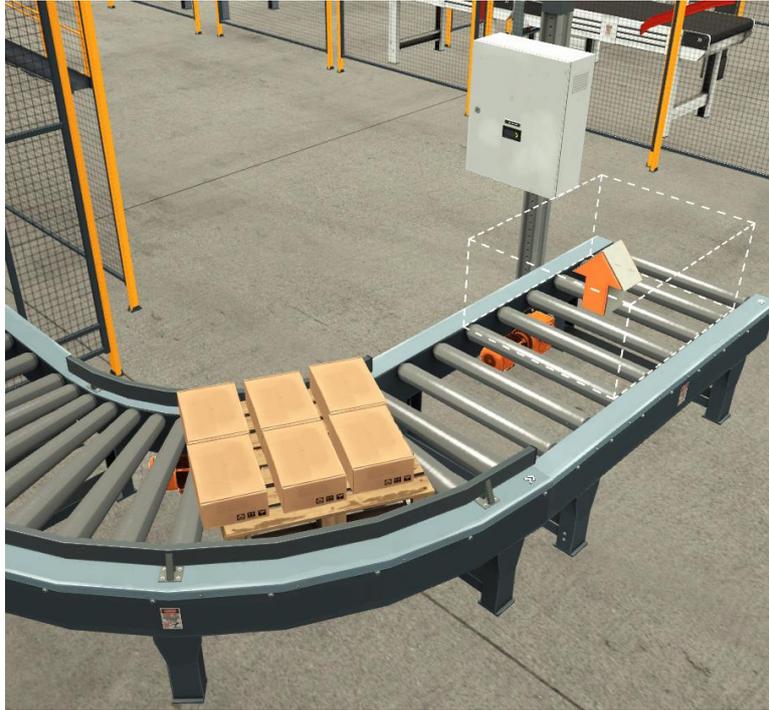


Imagen 20: Pallet cargado con seis cajas hacia destino

Se procede ahora a seguir las piezas de colores por el camino recto. Lo primero que encuentran es otro orientador simple para enderezar las piezas un poco al haber perdido su orientación original. Después, continúan hasta llegar a un sensor de color, el cual detecta si la pieza es azul, verde o gris.

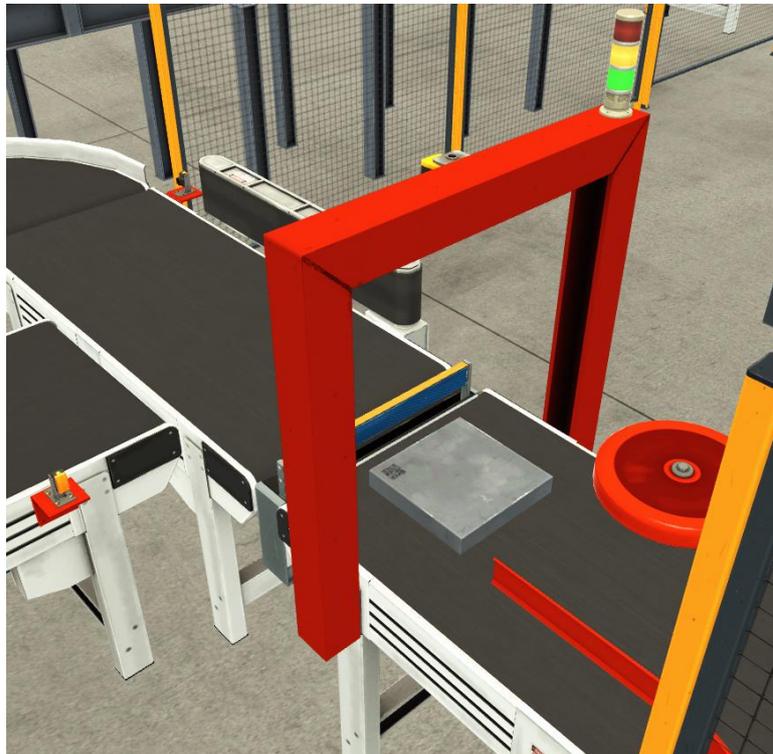


Imagen 21: Sensor de piezas defectuosas con barrera

Si la pieza es gris se interpreta como defectuosa y se desecha para su posterior reciclado (no contemplado en este proyecto). Esta acción se realiza mediante el giro de un brazo con cinta incorporada la cual hace que la pieza la acompañe y cambie de trayectoria. Un contador se encarga de contar el número de piezas defectuosas total para su posterior análisis por parte del propietario de la factoría.



Imagen 22: Brazo activado con pieza defectuosa

Por otra parte, si la pieza es verde o azul se le permite continuar recto.

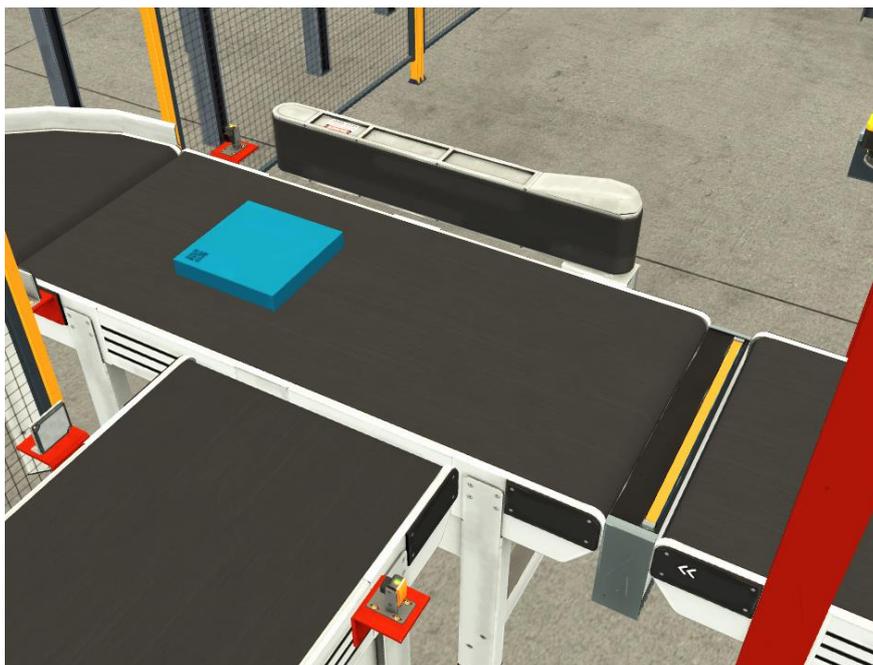


Imagen 23: Pieza azul continuando trayectoria normal

La pieza continúa avanzando por el conjunto de cintas transportadoras hasta llegar de nuevo a un sensor de color, esta vez para separar de recorrido a las piezas buenas.



Imagen 24: Pieza azul en sensor de color piezas buenas

Ya sea la pieza de color azul o verde, es recogida por una máquina llamada *pick and place*, la cual se encarga de dejar la pieza, dependiendo de su color, en una de las dos cintas transportadoras separadas por dicha máquina. Si la pieza es de color azul, la deposita en la cinta de la izquierda. Por el contrario, si la pieza es de color verde, la deposita en la cinta de la derecha.

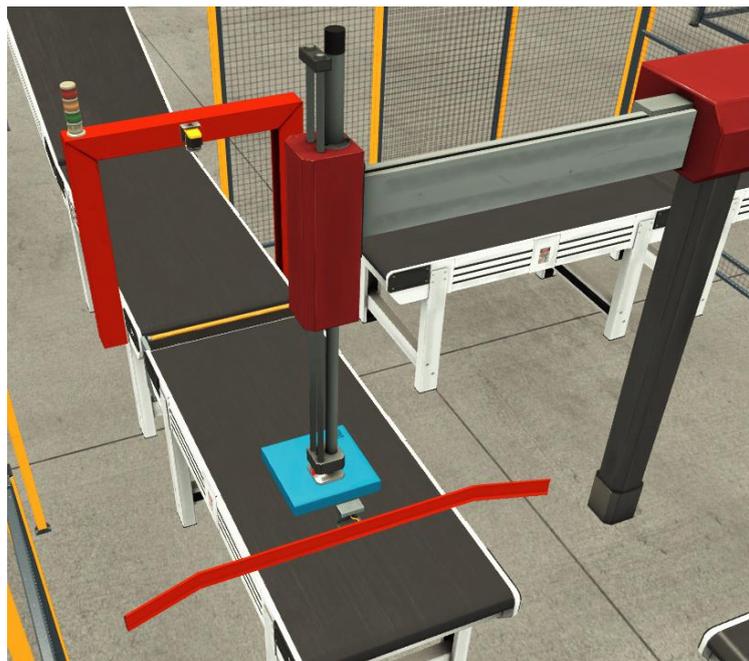


Imagen 25: *Pick and place* recogiendo pieza azul

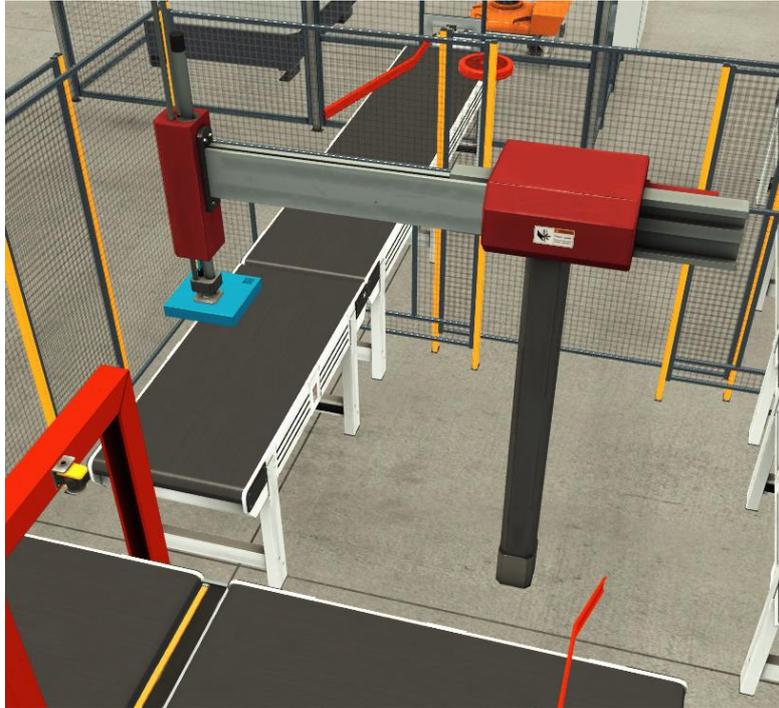


Imagen 26: *Pick and place* recogiendo pieza azul

Una vez que el *pick and place* deja la pieza azul en su cinta, ésta avanza hasta entrar en un centro de procesamiento/estampado el cual se encarga de modificar la pieza original hasta lo que se considera el producto final para su posterior venta.

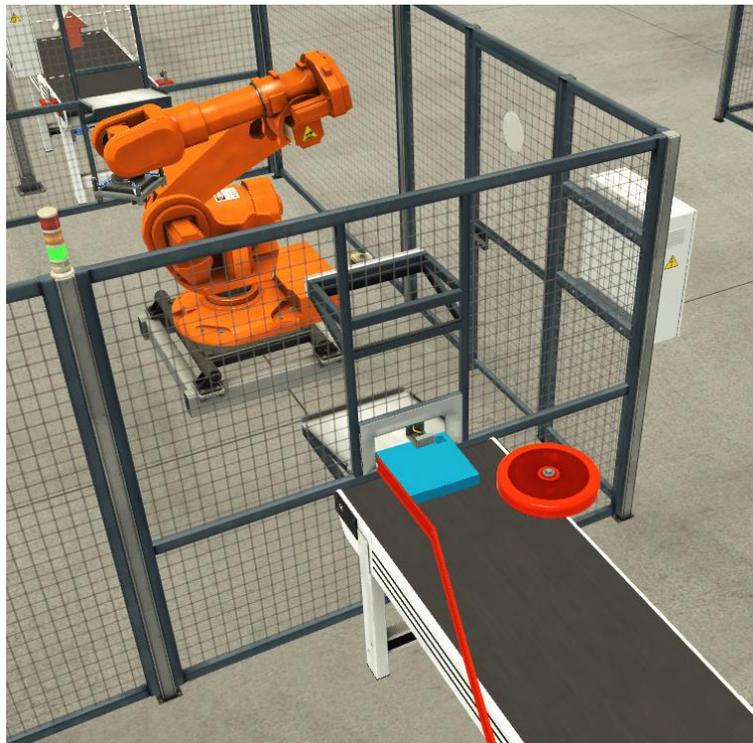


Imagen 27: Pieza azul llegando a centro de procesamiento



Imagen 28: Pieza azul siendo estampada

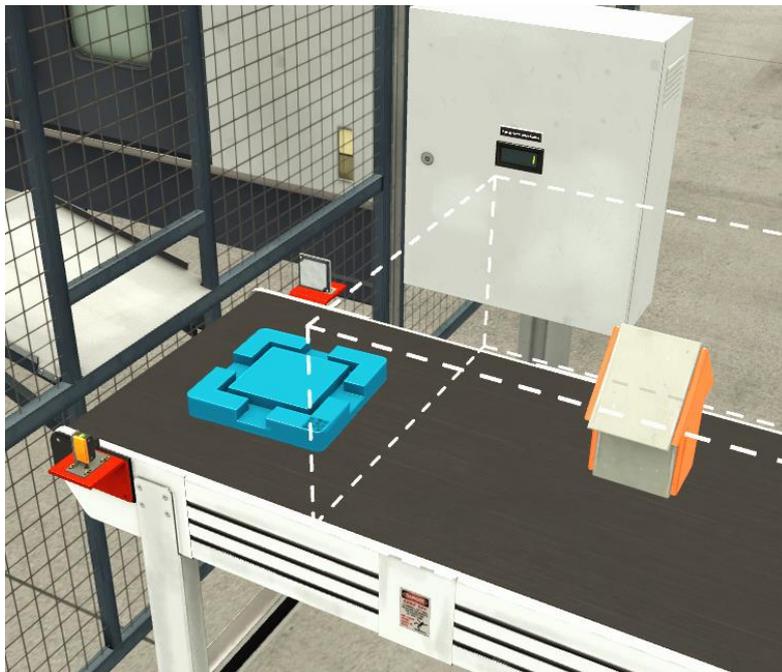


Imagen 29: Pieza azul como producto final

Las piezas verdes reciben el mismo tratamiento solo que partiendo de que el *pick and place* las coloca en la cinta transportadora de la derecha.

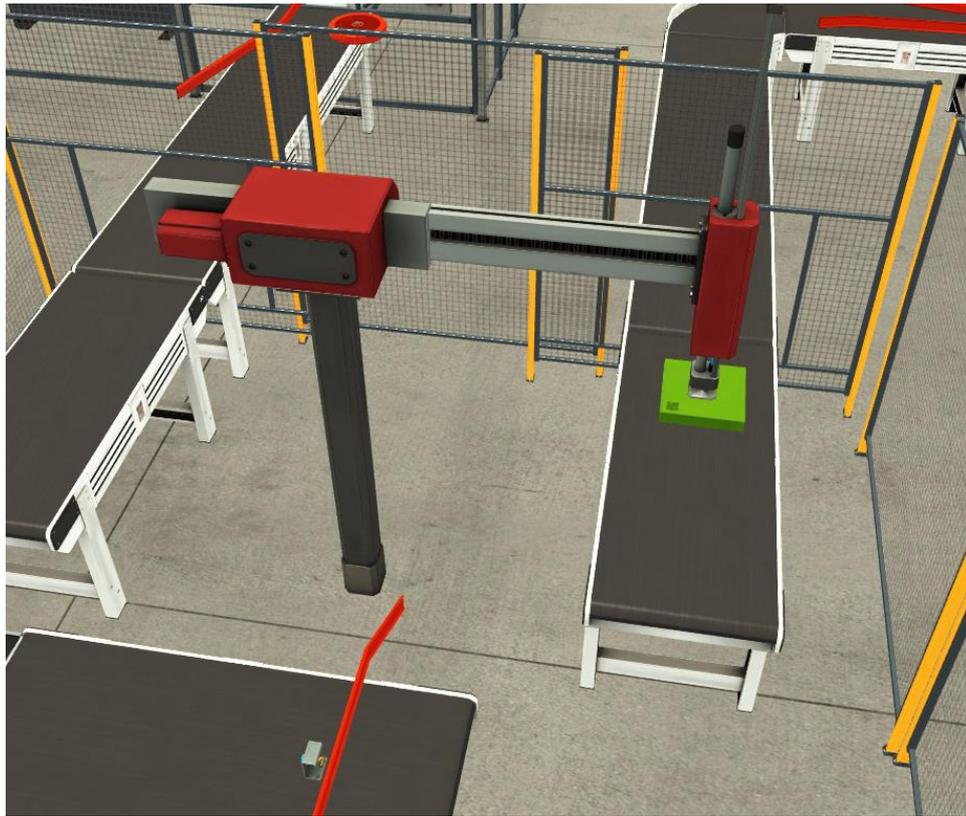


Imagen 30: *Pick and place* recogiendo pieza verde

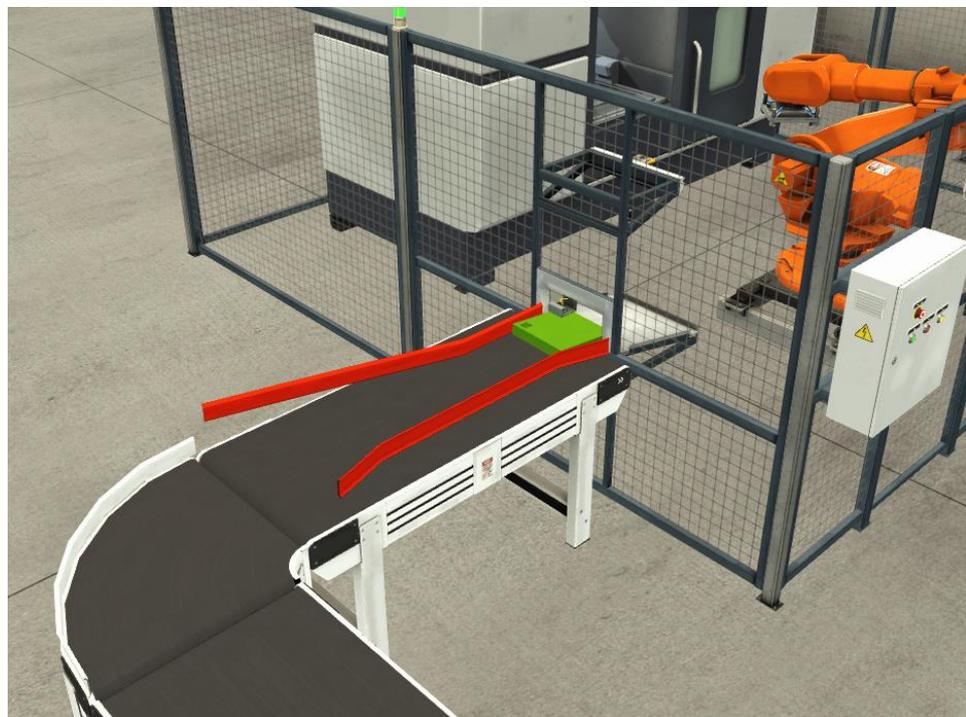


Imagen 31: Pieza verde llegando a centro de procesado

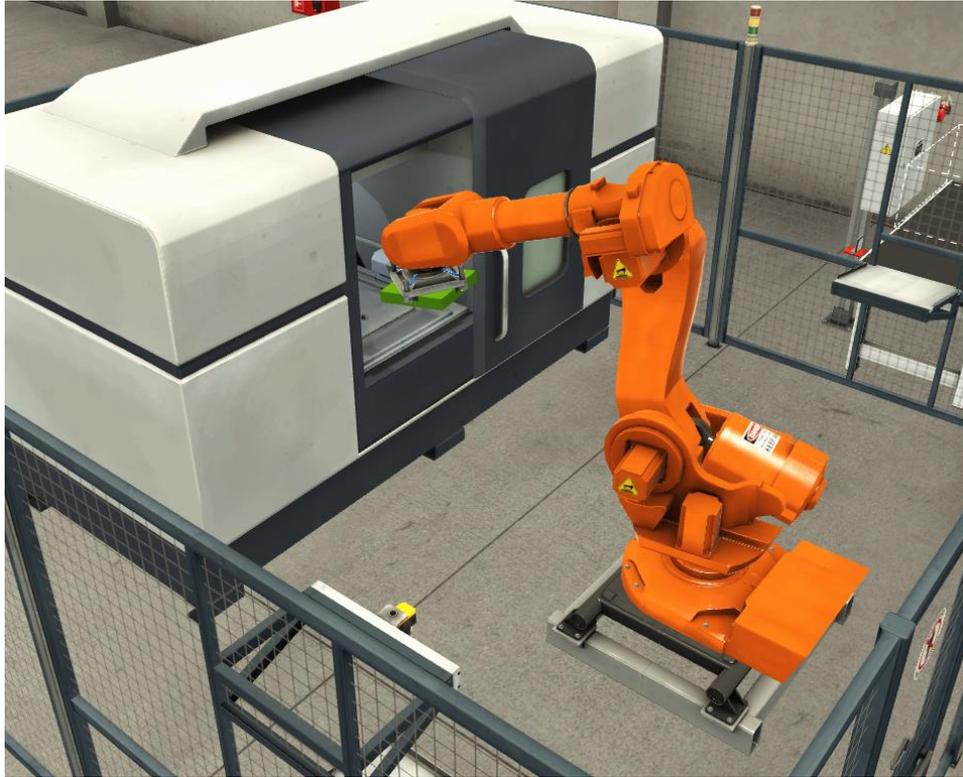


Imagen 32: Pieza verde siendo estampada

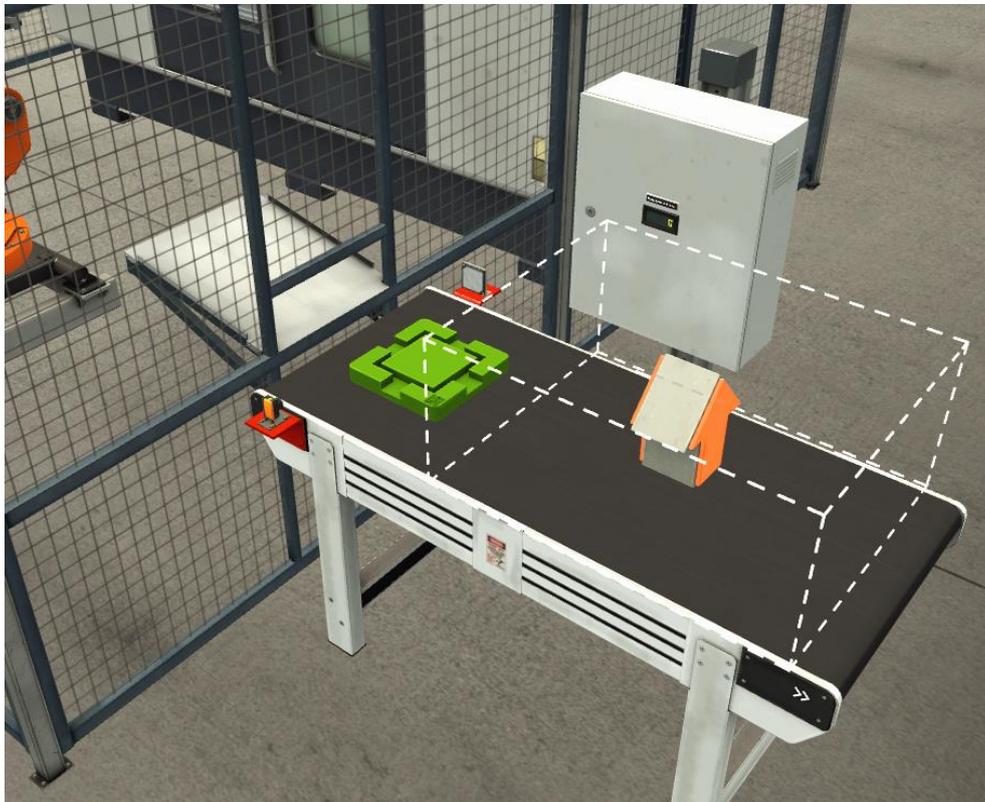


Imagen 33: Pieza verde como producto final

De apretarse la seta de emergencia situada en el panel de control principal de la fábrica, todas las máquinas se paran al momento y una alarma y luces rojas indican la emergencia. Por último, Descripción del funcionamiento

Se pasa ahora a describir en detalle el funcionamiento de cada SFC principal de la factoría. Para poder aprovechar al máximo el potencial del autómata programable M241 de Schneider se ha decidido separar el funcionamiento de las máquinas en subprogramas o POU separados. Esto permite tener una mayor flexibilidad a la hora de incorporar nuevos programas si se decide ampliar la fábrica o añadir nuevas funcionalidades, hacer modificaciones a programas ya existentes, o eliminar programas defectuosos, antiguos, o con falta de perfeccionamiento.

Como excepción hay que destacar el programa de contadores que se ha hecho en Diagrama de contactos o LD, a la vez que no existe un SFC de emergencia como tal, sino que se ha decidido incluir un propio apartado en cada SFC de subprograma.

3.1.1. Emergencia/Alarma

Como se ha descrito anteriormente, el SFC de parada de emergencia se encuentra dentro de cada POU de manera que el sistema de programas no queda dependiente de la estructura de este.

Lo que se traduce en que se puede aprovechar de tener un sistema de parada de emergencia “con memoria”. Esto quiere decir que al presionar la seta de emergencia la fábrica pasa a un estado de parada total pero cada máquina recuerda el estado donde se encuentra. Esto se ha conseguido incluyendo una etapa llamada igual que la etapa normal, pero con una “E” de emergencia para distinguirla de la original. Las luces de funcionamiento de las máquinas pasan a color rojo, y una alarma sonora comienza a sonar para avisar a todo el personal de que algo ocurre y debe subsanarse.

Una vez la emergencia se ha solucionado, una nueva presión de la seta de emergencia hace que las máquinas vuelvan a continuar justo en la etapa en el que se encontraban, lo que permite continuar el proceso sin tener que empezar otra vez desde el principio, como les ocurre a otros proyectos.

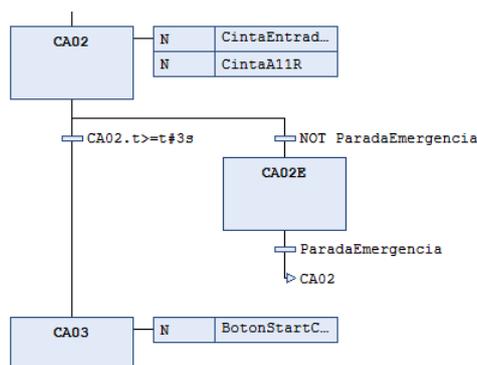


Imagen 34: Ejemplo SFC emergencia con “memoria”

3.1.2. Pick and place

La máquina descrita anteriormente como *pick and place* funciona con ayuda de un sensor que se encuentra en la cinta transportadora, la cual detecta si una pieza ha llegado a la posición adecuada. El uso de este sensor auxiliar es necesario ya que el propio *pick and place* carece de un sensor lo suficientemente potente como para detectar si una pieza se encuentra a su rango sin tener que bajar el brazo completamente.

Una vez una pieza se encuentra en el rango de la máquina, la cinta transportadora se detiene y el brazo del *pick and place* comienza a bajar a la vez que se activa la succión para poder coger la pieza.

Para ser capaz de saber si la pieza que lleva es azul o verde se ha instalado antes un sensor de color el cual almacena como una variable numérica dicho color. Si la pieza es de color azul, se guarda un "1" como valor de la variable, si la pieza es verde, el valor numérico de la variable es "4".

Con esta información el *pick and place* es capaz de hacer la toma de decisión si girar 90° en el sentido o al contrario de las agujas del reloj. Si la pieza es de color azul, gira 90°. Si la pieza es de color verde, gira -90°.

Una vez hecho este giro, el brazo del *pick and place* se extiende y baja para depositar la pieza en su cinta transportadora y después recoge y sube el brazo de nuevo. Aprovechando la información de qué color es la pieza, sabe en qué dirección ha girado al principio, por lo tanto, también sabe en qué dirección girar ahora para retornar a su estado de reposo. Si la pieza es de color azul, gira ahora -90°. Si la pieza es de color verde, gira ahora 90°.

Como se ha destacado anteriormente, si ocurre una parada de emergencia en cualquier momento, el *pick and place* permanecerá en estado de pausa en la misma etapa en la que se encontraba, ya que no posee elementos que puedan dañar a ningún personal al encontrarse encerrado tras una jaula de protección. Se ha decidido dar prioridad a este SFC para no sufrir embotellamientos en el resto del recorrido.

3.1.3. Organizador peso

El organizador de peso está en control de las cintas transportadoras que le preceden. Dichas cintas se encuentran en funcionamiento siempre que ninguna materia prima se encuentre en la báscula de pesado ni en los rodillos giratorios.

Cuando una materia prima llega a la báscula del organizador, ésta detecta el cambio de peso (mayor de medio kilo). Para realizar una medida con la mayor precisión posible se detiene también la propia cinta de la báscula, ya que realizar medidas sobre una cinta transportadora en funcionamiento provoca medidas erróneas.

Una vez la cinta se detiene, el organizador espera un segundo y medio para que el valor de peso se estabilice y toma la medida, guardándola en una variable numérica a la que se le aplica un factor de multiplicación (necesario en la comunicación Factory I/O <→ SoMachine).

A continuación, con la información de esta variable de peso se ha programado los rodillos giratorios del organizador para que modifiquen o no su posición de acuerdo al tipo de materia prima:

- ❖ Si se trata de una caja de cartón, su peso es igual o menor a 3'5Kg, por lo tanto, se ha programado los rodillos para que roten 45º a la derecha y girando envíen la caja al circuito descrito en primer lugar de la fábrica. Una vez la caja abandona el organizador, los rodillos giratorios vuelven a su posición original.
- ❖ Si se trata de una pieza azul, verde o gris, su peso es superior a 3'5Kg, por lo tanto, se ha programado los rodillos para que no roten en ninguna dirección y girando envíen la pieza de color hacia el segundo circuito que posee la fábrica.

Cabe destacar que, si la siguiente máquina en el circuito de las piezas de color se encuentra haciendo algún proceso, los rodillos esperarán para enviar la pieza en dirección recta. Esto no afecta al circuito de las cajas de cartón.

Si en algún momento ocurre una parada de emergencia, el organizador permanecerá en estado de pausa en la etapa en la que se encontraba, pero deteniendo los rodillos ya que la máquina no se encuentra bajo una jaula de protección y puede afectar a cualquier operario.

3.1.4. Máquinas de control numérico (CNC)

Las máquinas de control numérico (la fábrica posee uno para cada color de pieza, tanto azul como verde), son unas máquinas cuya programación interna no se puede afectar ya que se encuentran dentro del propio software Factory I/O. Sin embargo, dichas máquinas de control numérico poseen un panel de control básico con botones de encendido, pausa y stop, a la vez que sensores de funcionamiento en marcha o no.

Aprovechando las herramientas que se proporcionan y añadiendo un sensor más extra a la entrada de cada centro de maquinado para saber si una pieza está a punto de entrar, se ha realizado la programación de su comportamiento:

Cuando una pieza se aproxima a la entrada, el sensor emite una señal que es detectada por el autómatas y espera tres segundos a que la pieza se deslice hacia la posición donde, a continuación, será cogida por el brazo del centro de maquinado y llevado a estampación.

Una vez se han esperado estos segundos, se simula que un operario ha presionado el botón de encendido y el centro de maquinado mueve el brazo, coge la pieza, la lleva a estampación, la pieza es estampada y llevada finalmente a la salida donde se realizará el conteo de piezas terminadas con ayuda de otro sensor y finalizará el recorrido.

Si en algún momento ocurre una parada de emergencia, el centro de maquinado permanecerá en estado de pausa en la etapa en la que se encontraba, ya que está situado dentro de una jaula de protección y no puede afectar a ningún operario.

3.1.5. Palletizador

El funcionamiento del palletizador consta de un SFC para si mismo, un SFC para un sistema de sujeción y otro para las cintas de rodillos que proporcionan los pallets donde se alojarán las cajas de cartón en grupos de seis.

Comencemos primero con las cintas de rodillos. Estas se encuentran siempre activas hasta que un pallet se sitúa entre ambos sensores de posición que posee el ascensor del palletizador, una vez esto ocurre las cintas se detienen y el resto de los pallets se quedan a la espera de que el primero abandone el ascensor con la carga.

A continuación de colocarse el pallet entre los sensores, el ascensor se mueve hacia arriba hasta colocarse justo debajo de una trampilla que se abrirá cuando haya seis cajas de cartón, y dicho pallet permanecerá a la espera hasta que esto ocurra, ya que el factor limitante en este proceso y el que puede provocar embotellamiento es las cajas de cartón.

Al mismo tiempo que esto ocurre, comienzan a llegar cajas de cartón al *clamper*, el cual se encarga de terminar de dejar las cajas en posición recta para después orientarlas a la manera deseada y poder ser cargadas en la trampilla mediante un empuje.

El *clamper* se contrae para enderezar la caja de cartón y después se eleva para dejar que ésta pase. La caja avanza y con ayuda de un saliente del palletizador se coloca en posición para ser empujada. Un sensor colocado junto al *clamper* ayuda a realizar el conteo de cajas actuales. Cuando tres cajas de cartón se encuentran listas para ser empujadas, una barra los lleva encima de la trampilla y otras dos se encargan de presionarlas para optimizar el espacio y asegurar que no se caigan.

Una vez llegan tres cajas más, son empujadas y presionadas y la trampilla se abre, haciendo que se depositen las cajas de cartón encima del pallet. Después el ascensor baja con el pallet cargado, la trampilla se cierra y el pallet abandona el ascensor para llevar las cajas a su destino final mediante las cintas de rodillo. Finalmente, otro pallet asciende por el ascensor y el proceso está listo para volver a repetirse.

Si en algún momento se activa la emergencia, las cintas transportadoras se detienen y el *clamper* libera aquello que estaba presionando para evitar así cualquier tipo de daño ocasionado a un posible operario por la zona.

3.1.6. Pieza defectuosa

Este SFC maneja tanto el sensor de color de las piezas, como la barrera para que no pasen, como el brazo con cinta que se activa cuando se detecta una pieza defectuosa. Cuando la pieza de color abandona el organizador de peso, llega mediante cintas transportadoras al sensor de color, que detecta este cambio y detiene la cinta.

Una vez ha ocurrido esto, el sensor de color identifica qué pieza es y dicho color se almacena en una variable de tipo numérica tal que: "1" pertenece a pieza azul, "4" pertenece a pieza verde y "7" pertenece a pieza gris (que en este proyecto se trata como pieza defectuosa).

A continuación, la barrera que impide a la pieza avanzar baja y la cinta transportadora se vuelve a poner en marcha. Dependiendo del color de la pieza pueden ocurrir dos cosas:

- ❖ Si se trata de una pieza de color gris, un brazo con cinta transportadora se activa, desviando así la pieza a través de una rampa de metal para ser desechada y posteriormente reciclada (no contemplado en este proyecto). El brazo y la barrera vuelven después a su posición original para repetir así el proceso de nuevo.
- ❖ Si se trata de una pieza azul o verde, el brazo permanece en su posición original dejando así pasar la pieza para que continúe el recorrido con normalidad. La barrera vuelve después a su posición original para repetir así el proceso de nuevo.

Si en algún momento se activa la emergencia, las cintas transportadoras se detienen, la barrera sube para liberar aquello que podría haberse quedado atrapado en la junta y el brazo accionado detiene su cinta también.

3.1.7. Siempre encendido

Este SFC es el más simple de todos ya que contiene únicamente todos aquellos actuadores que deben permanecer siempre encendidos o que no precisan de un control selectivo dependiendo de la etapa en la que se encuentre el proceso.

Como, por ejemplo: generador de materia prima, eliminadores de piezas acabadas y defectuosas y cintas que se encuentran al final de los distintos recorridos.

Si en algún momento se activa la emergencia, todas estas cintas y elementos detienen su actividad completamente hasta que dicha emergencia cese.

3.1.8. Contadores

Este POU se ha decidido hacer en diagrama de contactos ya que existen bloques de funciones que realizan la función de contadores ascendentes, descendentes o ascendentes/descendentes.

Existe contador de los siguientes elementos: Materia prima generada, pallets de cajas de cartón acabados, piezas defectuosas desechadas, piezas azul estampadas, piezas verdes estampadas.

Activar la parada de emergencia no resetea el valor de los contadores.

3.1.9. Implementación del automatismo sobre PLC M241 de Schneider

Para poder crear y cargar el automatismo al PLC M241 de Schneider es necesario utilizar el software SoMachine.

El primer paso es crear un nuevo proyecto y nombrarlo con el nombre que se desee.

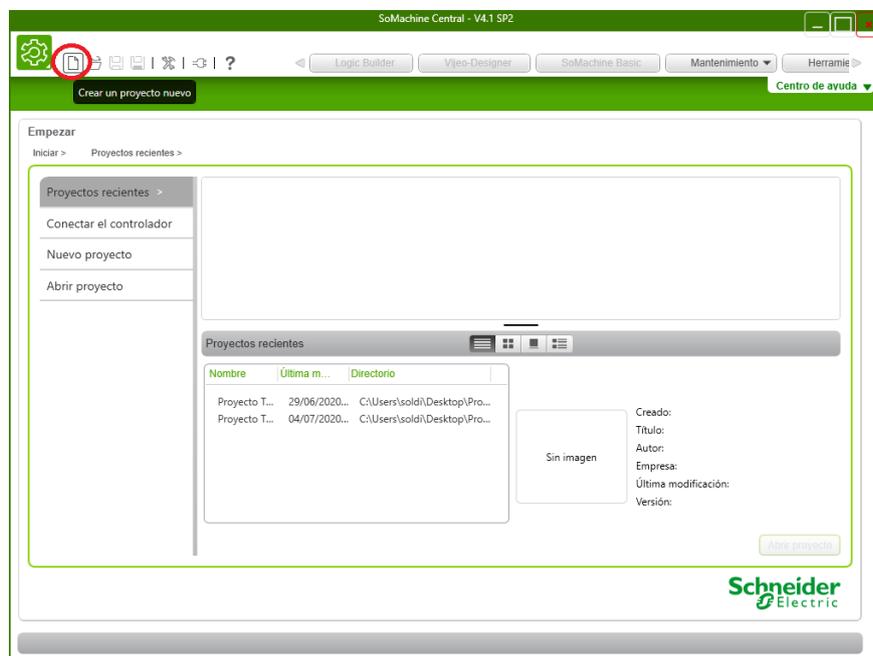


Imagen 35: Crear proyecto en SoMachine

A continuación, se hace clic sobre Abrir configuración, el cual lleva a lo que sería el menú central donde se tiene el control absoluto de la programación. Una vez allí el programador o encargado del desarrollo del programa será capaz de crear y gestionar el programa a cargar a voluntad.

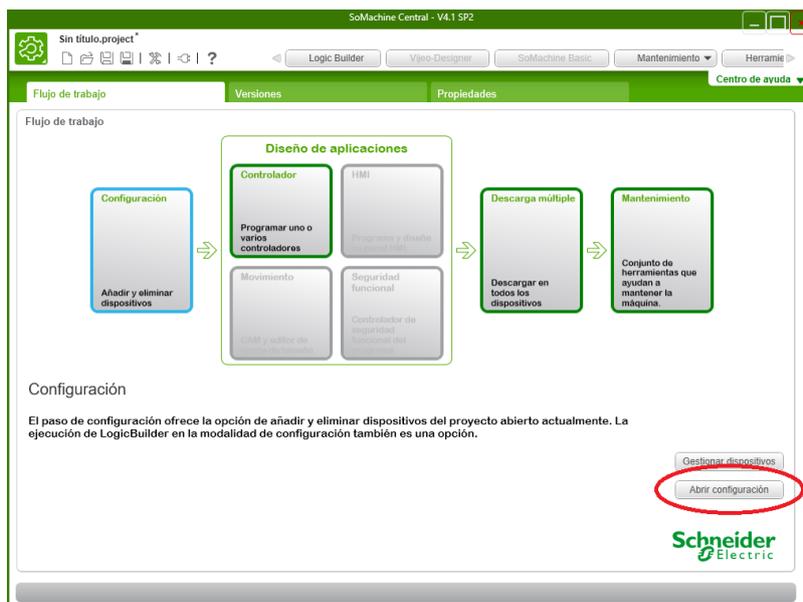


Imagen 36: Abrir configuración

Como en este proyecto ya tiene creados los distintos POUs y tiene la lista de variables globales (GVL) por el alumno, ahora se procederá a descargar el proyecto al propio autómeta.

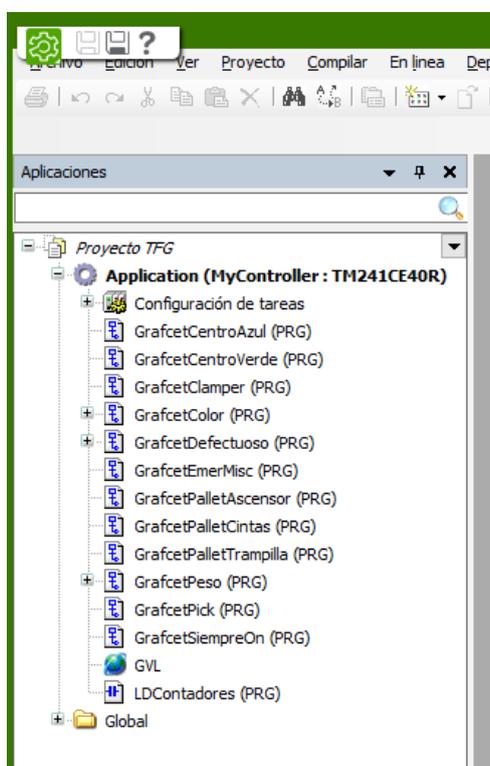


Imagen 37: POUs de proyecto completo

Para descargar el proyecto al autómeta primero deben de hacerse unos ajustes previos acerca de la conexión Ethernet y la IP que posee el autómeta que va a recibir el proyecto.

Para esto primero se hace clic en la pestaña de Dispositivos y después hacemos doble clic en la opción de MyController (TM241CE40R).

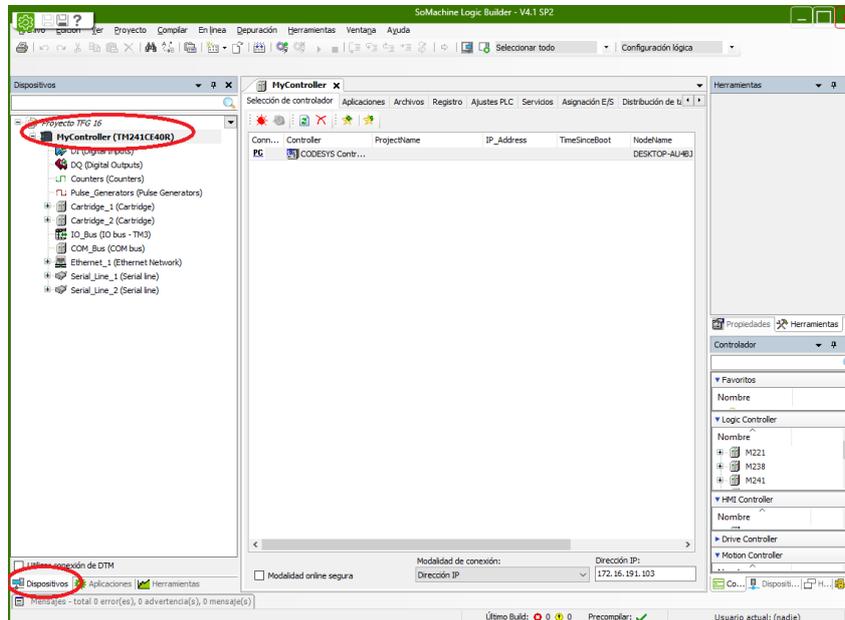


Imagen 38: Descargar programa autómatas paso primero

Una vez hecho esto, se elige la opción de Dirección IP como modalidad de conexión y se escribe la dirección IP que posee el autómatas. La IP varía para cada autómatas en la red.

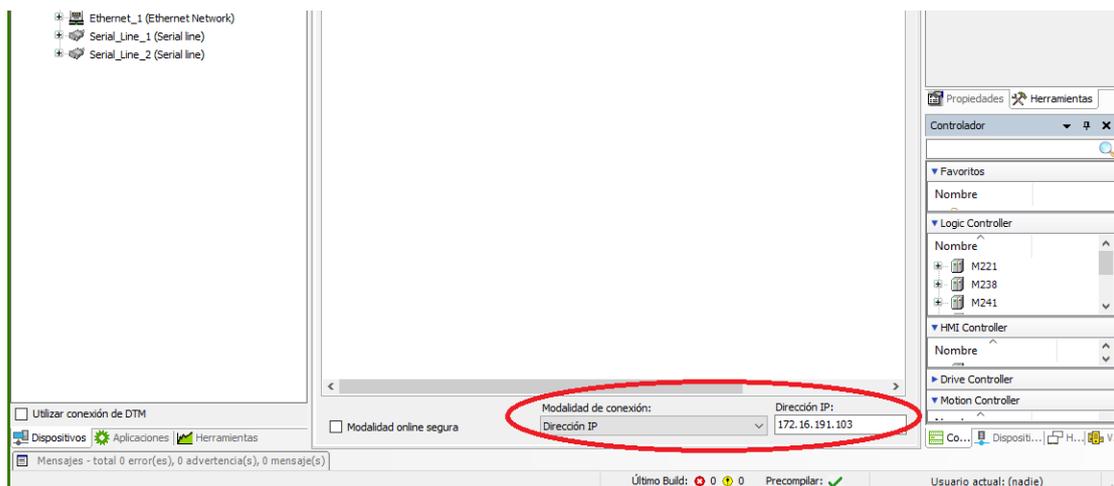


Imagen 39: Descargar programa autómatas paso segundo

A continuación, se hace doble clic en la opción "Ethernet1 (Ethernet Network)" y se selecciona "Dirección IP de DHCP" dejando el resto de los valores por defecto.

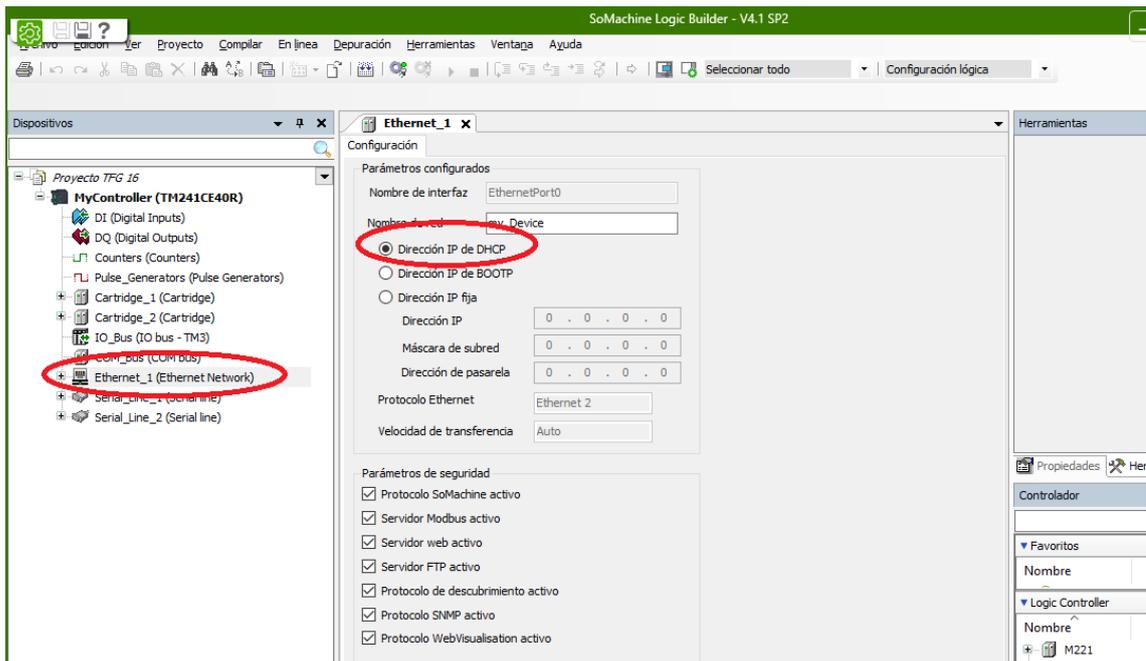


Imagen 40: Descargar programa automático paso tercero

Una vez hecho todo esto ya está todo configurado para descargar el programa en el automático. El siguiente paso es seleccionar el botón de “Iniciar la sesión”.



Imagen 41: Descargar programa automático paso cuarto

Aparecerá entonces un mensaje emergente preguntando si estamos de acuerdo en seguir las instrucciones de Schneider sobre direccionamiento simbólico y los peligros que pueden ocasionar un mal funcionamiento del equipo controlado por el automático. Para continuar se debe presionar ALT+F.

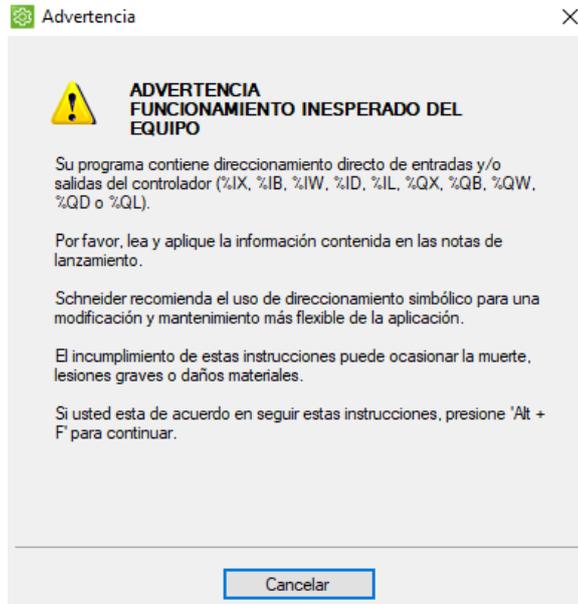


Imagen 42: Aviso de Schneider al descargar proyecto a autómeta

En el caso de aparecer un mensaje más después de este avisando de que en el autómeta se encuentra un proyecto diferente, debemos de seleccionar la opción iniciar sesión con descarga, ya que una modificación en línea puede ocasionar fallos.

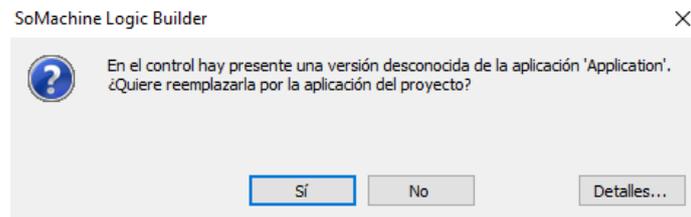


Imagen 43: Aviso de proyecto diferente en autómeta

Una vez todo este proceso finalice ya solo quedará poner el autómeta en marcha haciendo clic en el botón de inicio o presionar F5.

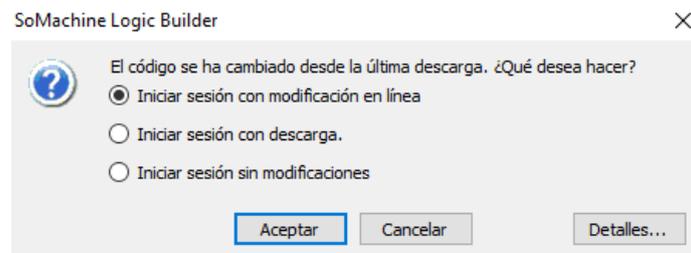


Imagen 44: Aviso iniciar sesión con descarga

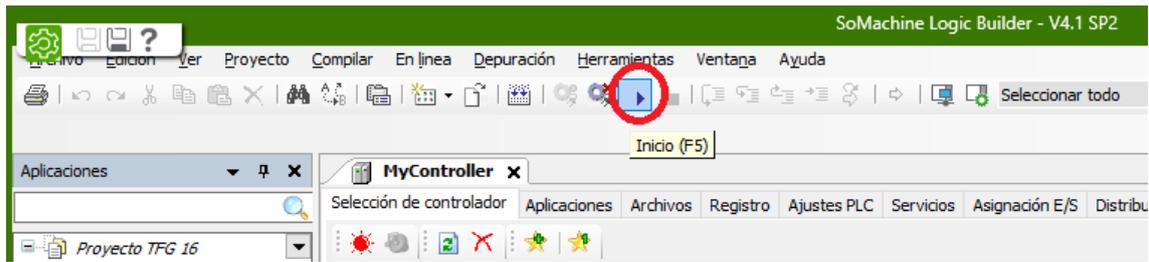


Imagen 45: Botón de inicio

3.1.10. Conexión entre Factory I/O y autómeta

En cuanto el autómeta ya tiene el programa deseado cargado en su memoria, lo único que falta por hacer es configurar Factory I/O para que sea capaz de comunicarse con él.

Dentro de Factory I/O hacemos clic en la pestaña de File y seleccionamos Drivers, lo cual nos llevará a una ventana diferente.

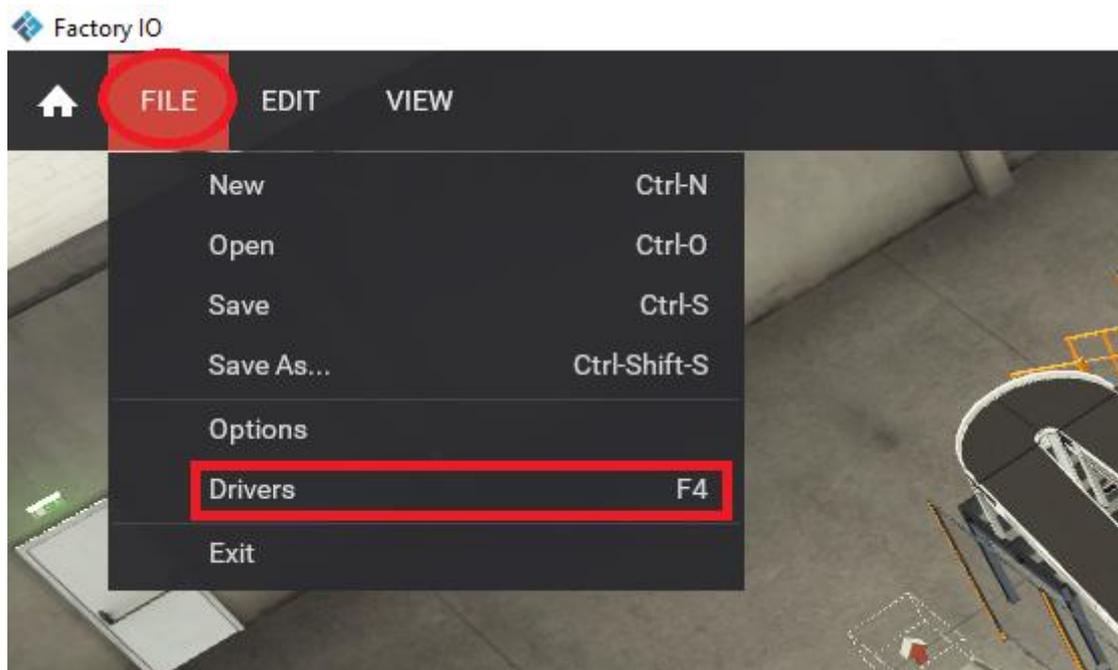


Imagen 46: Factory I/O opción drivers

La ventana que se presenta a continuación es la que permite organizar y personalizar el *layout* de las bobinas o *coils* que se encuentran dentro de Factory I/O, las cuales están formadas por actuadores y sensores. Y que debemos de asignar a una dirección de memoria que es la que se comunicará posteriormente con el autómeta. Es necesario, en primer lugar, seleccionar en el menú desplegable de drivers Modbus TCP/IP Client.

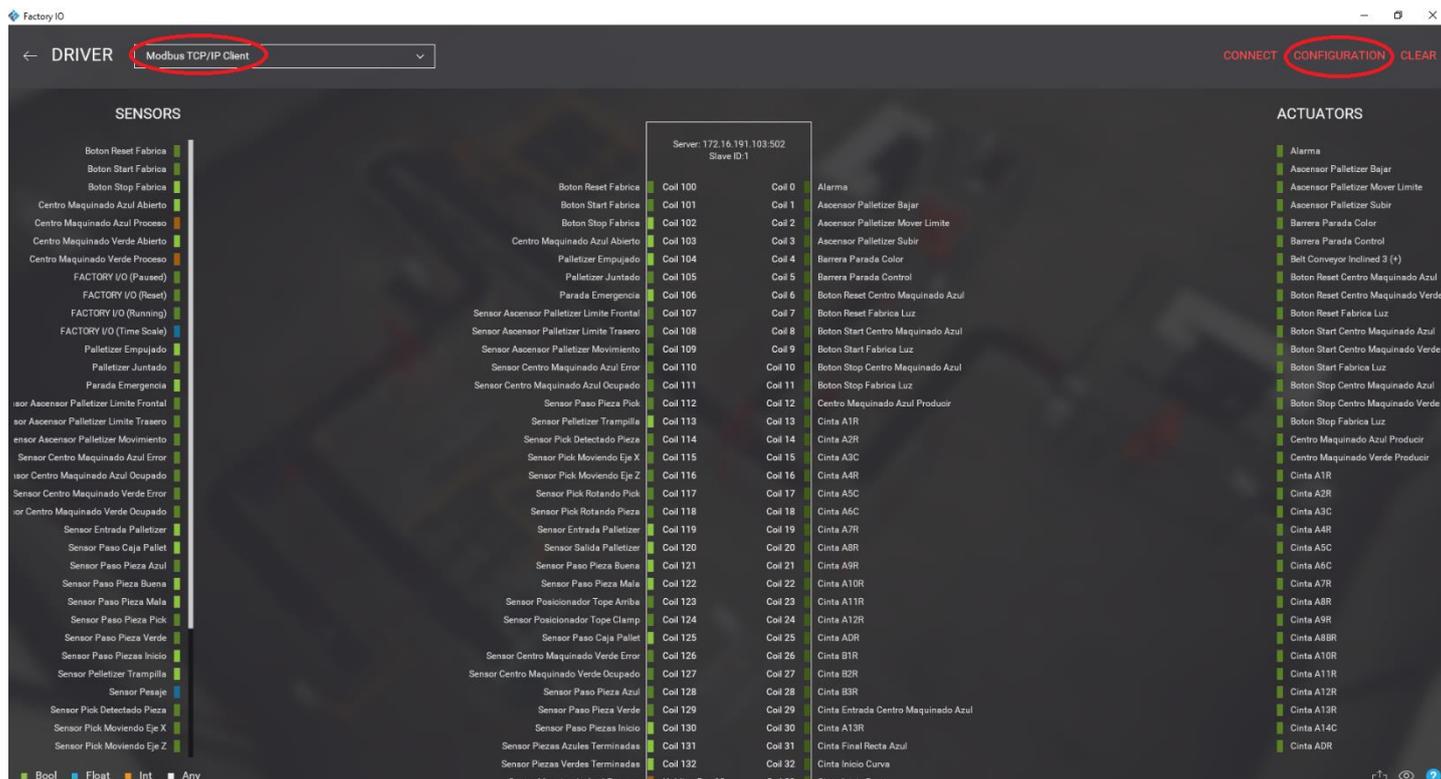


Imagen 47: Factory I/O layout bobinas

Una vez se tiene el *layout* completo de los sensores y actuadores en el orden deseado, seleccionamos la opción de *configuration*. Después debemos seleccionar de nuevo Modbus TCP/IP Client a la izquierda y empezar a introducir los ajustes:

- ❖ Port: Es la IP que debe coincidir con la del autómatas para que se pueda hacer la comunicación.
- ❖ Port: Se deja por defecto
- ❖ Slave ID: Se deja por defecto
- ❖ I/O Config:
 - Read Digital: Se deben seleccionar Coils.
 - Read Register: Se debe seleccionar Holding Registers.
 - Scale: 100 para este proyecto, pero varía en función de como se desee trabajar con los valores que tienen decimales ya que deben eliminarse, de aquí este factor de multiplicación.
- ❖ I/O Points: Aquí se configura de cuantas entradas y salidas tanto digitales como analógicas va a disponer Factory I/O, a mayor número de I/O, más grande o complejo es el proyecto. Es muy importante destacar que estos valores no pueden chafarse entre ellos, sino que hay que hacer uso de *offsets* que desplazan las variables dirección de memoria.

Se adjunta captura a continuación de los valores usados para este proyecto.

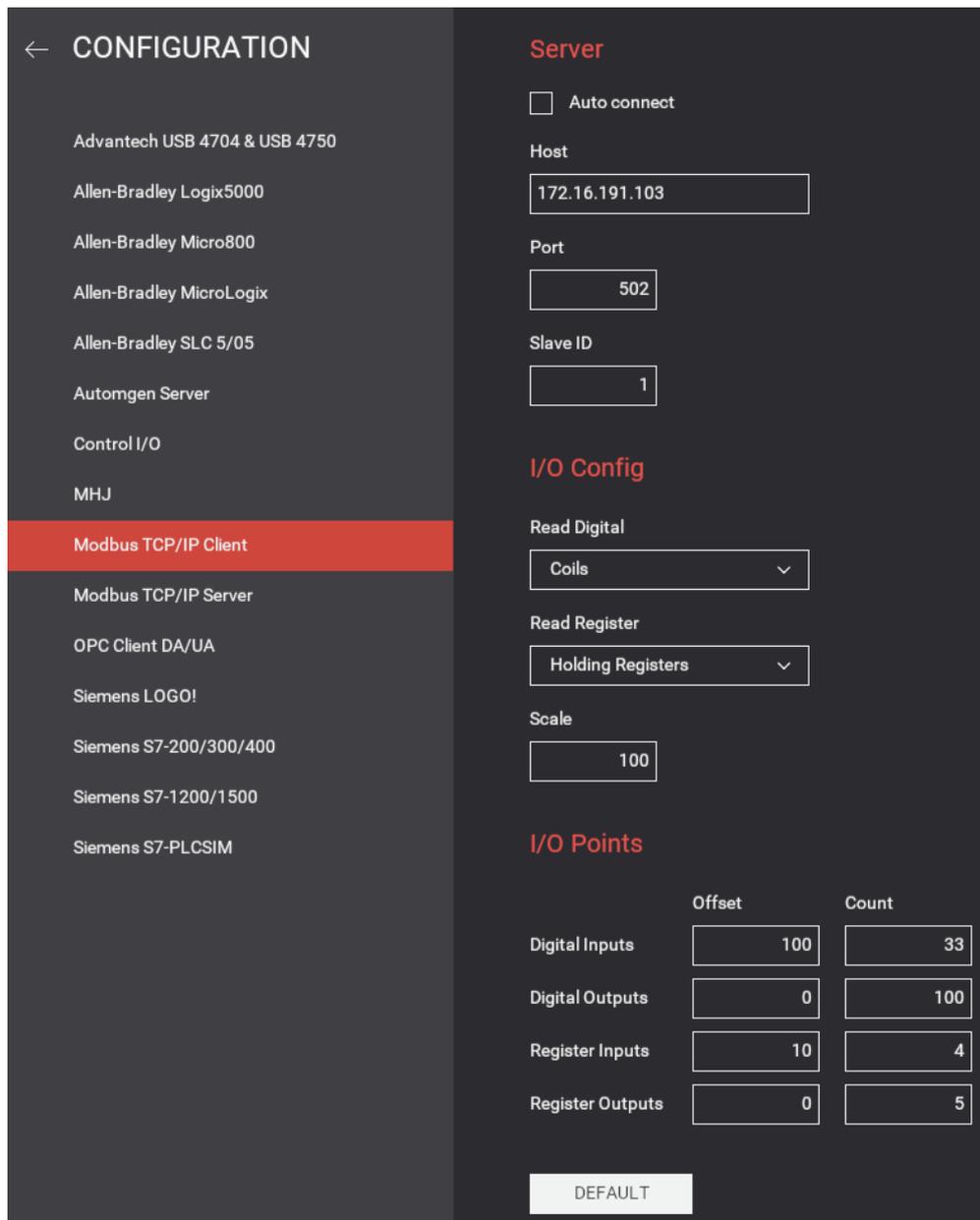


Imagen 48: Configuración de Factory I/O completa

Por otra parte, la lista de variables globales de SoMachine tiene que coincidir con las direcciones de memoria reservada en el apartado I/O Points, respetando los *offsets* nombrados anteriormente. Si esto no se hace, el autómatas accionará valores de memoria incorrectos y, por tanto, el programa no funcionará correctamente. Toda dirección de memoria digital empieza por %QX y las analógicas por %MW. El tipo de variable digital es siempre BOOL mientras que las analógicas pueden ser INT, entre otros.

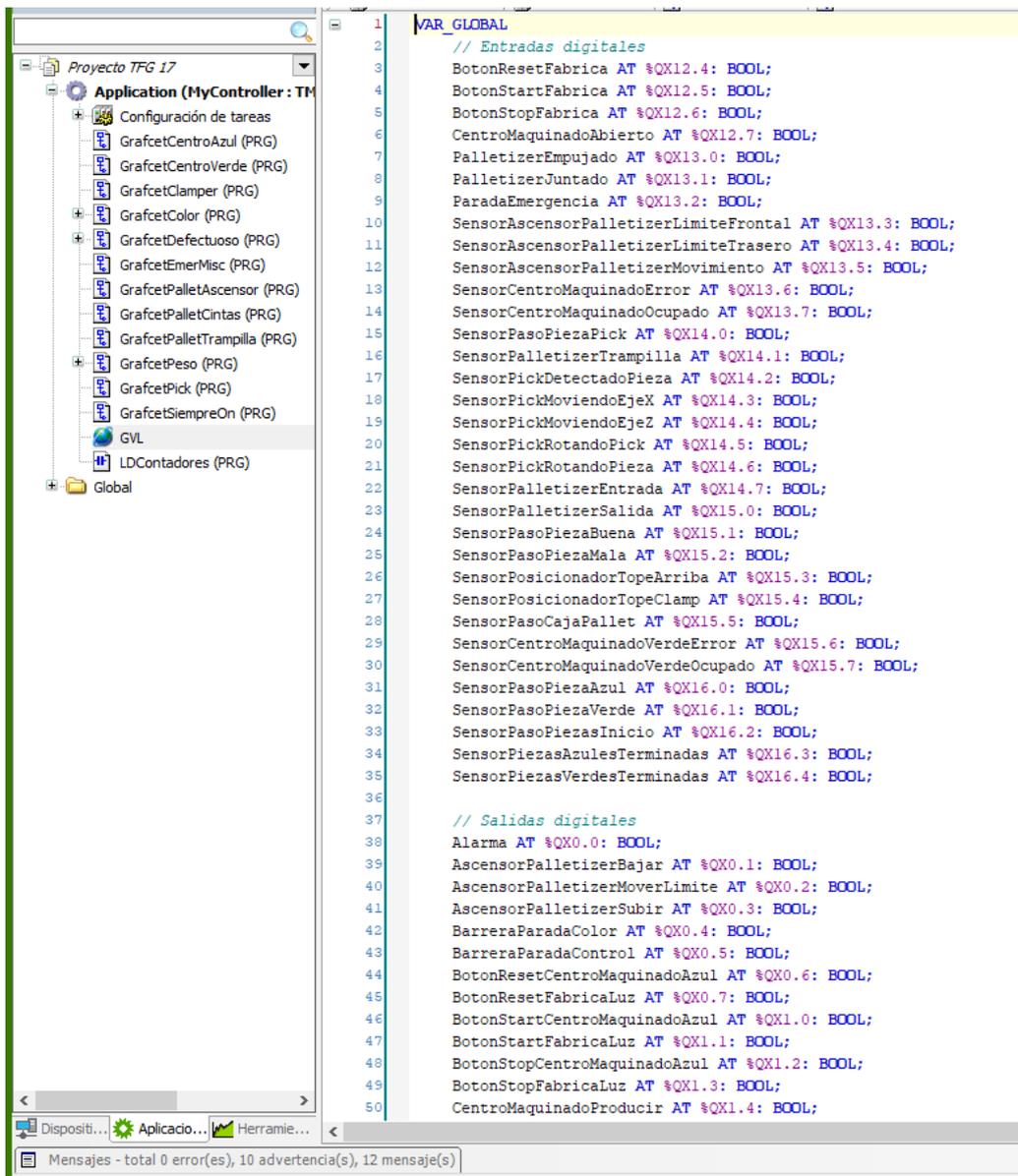


Imagen 49: Variables globales en SoMachine

Una vez realizados todos estos ajustes solo quedaría pendiente seleccionar *connect* y esperar a que aparezca el tic verde que indica que la conexión ha sido correcta.

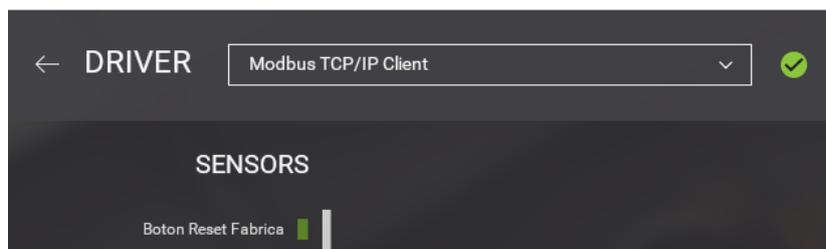


Imagen 50: Conexión correcta en Factory I/O

3.2. Consulta en la web

El software SoMachine permite acceder de manera remota a través de un dispositivo con navegador o incluso desde el mismo ordenador donde se ha cargado el programa al autómatas para simular un SCADA.

Dicho servidor web es totalmente personalizable de acuerdo a las necesidades del cliente, ya que se puede elegir qué mostrar en pantalla en cada momento y qué información es más relevante para tener controlada.

El dirección de la página web dentro de la red de la Universidad Politécnica de Valencia que permite la visualización del autómatas es 172.16.191.103:8080/webvisu.htm (véase imagen 54).

3.2.1. Interfaz de usuario

La interfaz de usuario diseñada para acompañar este proyecto se ha diseñado de manera en que quede claro en todo momento si la fábrica se encuentra en funcionamiento normal o ha ocurrido una parada de emergencia.

Al mismo tiempo hay presente un interruptor que hace la función de conmutador para activar manualmente desde el SCADA la parada de emergencia.

También se ha decidido mostrar los valores de los contadores que están presentes en la factoría y que representan la cantidad total de materia prima que ha llegado a la fábrica, las piezas defectuosas que se han identificado, las piezas azules estampadas que han acabado el recorrido, las piezas verdes estampadas que han acabado el recorrido y el número de pallets cargado con un grupo de seis cajas que ha acabado el recorrido.

Se ha determinado de utilidad el representar el estado de las máquinas principales de la fábrica para, de un vistazo, poder ver si éstas se encuentran en funcionamiento en todo momento.

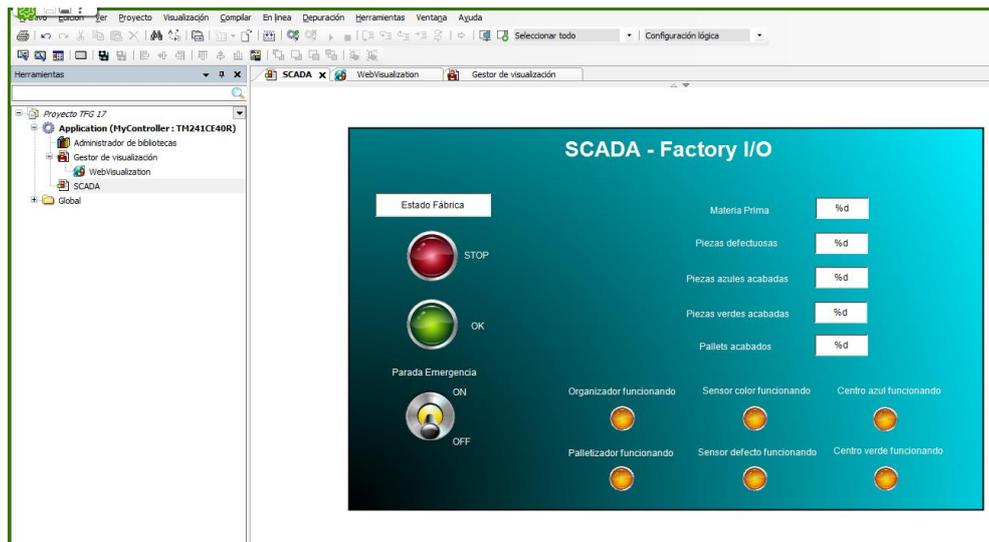


Imagen 51: Interfaz de usuario en SoMachine

Para crear dicha interfaz de usuario se ha de hacer clic derecho en *Application* dentro de la pestaña de Aplicaciones, después en el menú desplegable seleccionar “Visualización”. Una vez hecho esto se deberá nombrar dicho elemento y comenzar a personalizar el entorno visual que se quiera crear para la aplicación.

Cada elemento que se incluye en la visualización tiene unas propiedades propias que permiten su personalización, como por ejemplo posición, tamaño, tipo de fuente y tamaño de fuente (si usa cadena de texto), nombre identificativo, variable asignada (de ser necesaria) o incluso color (de permitirse el cambio).

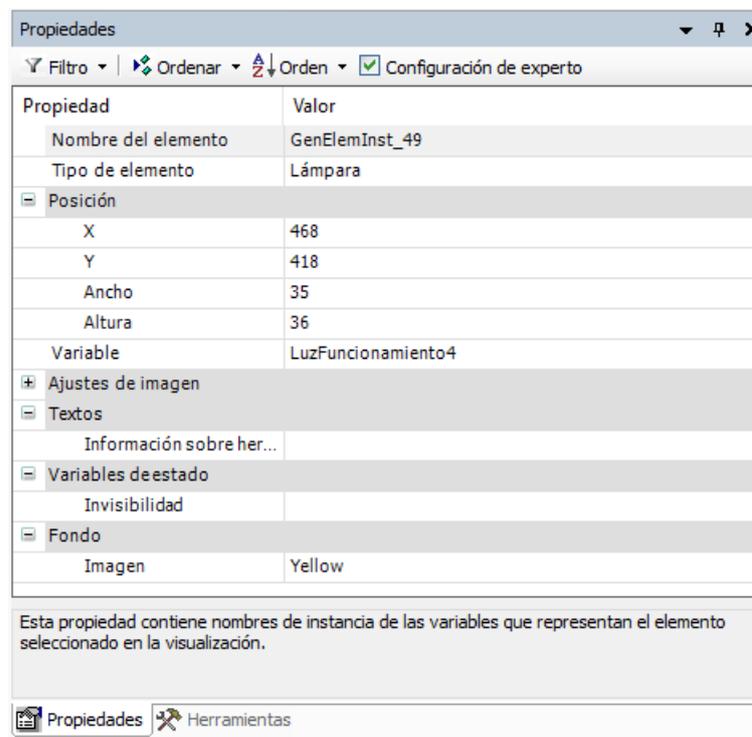


Imagen 52: Propiedades de objeto interfaz de usuario

Como el autómata M241 permite la visualización del programa en web a través de este SCADA, se ha configurado de manera en que cualquier persona dentro de la red de la Universidad Politécnica de Valencia pueda acceder a través de un navegador web a dicho SCADA y poder echar un vistazo a lo que está ocurriendo en tiempo real dentro del autómata.

Sin embargo, por precauciones de seguridad, no se permite el uso del interruptor que activa la parada de emergencia.

Para hacer posible esto, debe incluirse el nombre del Visualizador dentro del campo Visualizador de arranque, introducir "webvisu" en Nombre del archivo .htm, y un tiempo de actualización de información que puede ajustarse a las necesidades del proyecto.

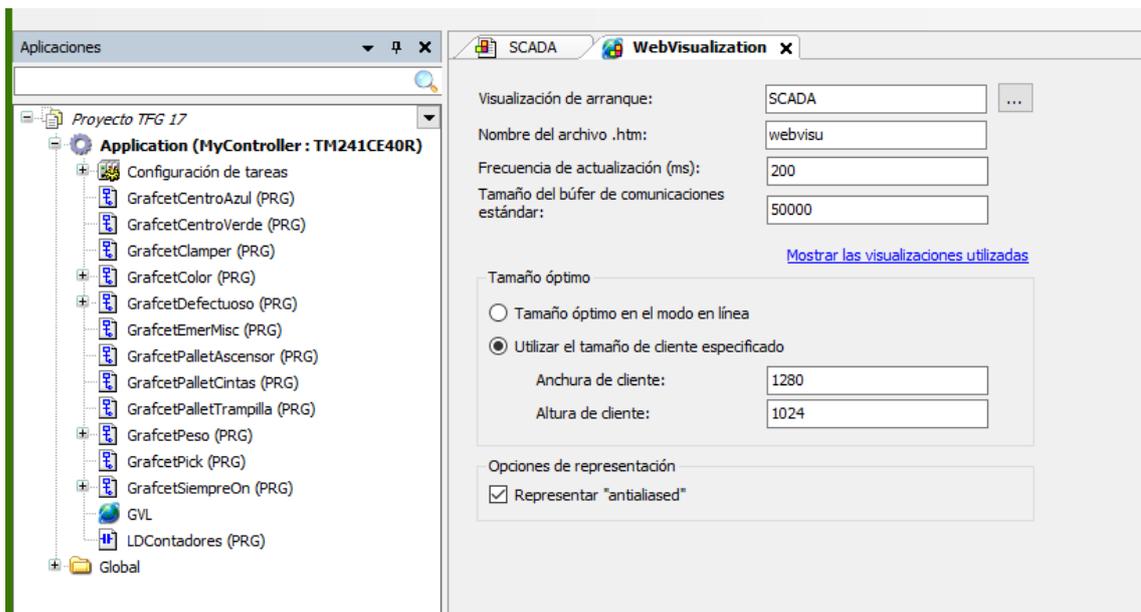


Imagen 53: Configuración de visualizador web

A partir de este momento, cuando el autómata inicia sesión y el programa se pone en marcha, cualquier persona dentro de la red de la universidad es capaz de visualizar el SCADA introduciendo la IP del autómata, seguido del puerto 8080 y después acabando en webvisu.htm que es el nombre que hemos elegido en el anterior párrafo.



Imagen 54: SCADA del proyecto visto en Firefox

4. Conclusiones

A la hora de realizar el proyecto, el alumno ha trabajado gran parte de las competencias transversales adquiridas durante el grado, destacando la de organización del tiempo para cumplir el plazo de entrega determinado.

Se han reforzado también conceptos aprendidos durante las asignaturas de automatización, incluso ampliándolas ya que se ha realizado el proyecto sobre un nuevo autómeta, el M241 de Schneider. Hacer la programación de un proyecto de manera modular ha servido para optimizar bien cada POU, teniendo mayor libertad que si se programa un sistema cerrado.

El proyecto ha terminado siendo muy diferente al concepto original ya que a la hora de poner en práctica todos los planes teóricos aparecen problemas, ideas de mejora y cambios a realizar para mayor calidad de producto final. Como añadido importante del proyecto cabe destacar la implementación del sistema de parada de emergencia con memoria, ya que favorece la modularidad antes destacada y hace que el sistema sea mucho más versátil.

La mayor dificultad del proyecto ha sido conocer el momento en el cual dejar de mejorar el sistema. Porque a cada sesión del alumno programando los distintos POU, al final siempre acababa con alguna idea nueva para mejorar el automatismo. Por lo tanto, el proyecto está totalmente personalizado al gusto de las necesidades del cliente y lo estará gracias a la modularidad antes destacada.

El alumno considera que el proyecto ha servido también para subir un nivel a la hora de realizar memorias de trabajos realizados, en comparación a los trabajos realizados durante el grado.

Como posible mejora a la hora de revisar al proyecto, el SCADA obtenido sería más complejo y personalizado, el que se ha realizado cumple las funciones básicas y podría ser más llamativo visualmente.

5. Bibliografía

- [1] - AENOR. (s.f.). *Buscador de normas*. [20 de Junio de 2010]. Obtenido de <https://www.aenor.com/normas-y-libros/buscador-de-normas/>
- [2] - Factory I/O. (s.f.). *Factory I/O*. [29 de Junio de 2010]. Obtenido de <https://factoryio.com/features/>
- [3] - Gandhi, M. (s.f.). *Autycom*. [26 de Junio de 2010]. Obtenido de <https://www.autycom.com/automatas-programables-en-la-produccion-industrial/>
- [4] - Gea, J. M. (2006). *Automatas*. [26 de Junio de 2010]. Obtenido de <https://www.automatas.org/redes/grafcet.htm>
- [5] - Mecanica Moderna. (s.f.). *Mecanica Moderna*. [26 de Junio de 2010]. Obtenido de <https://mecmod.com/automatas-programables-y-el-sector-industrial/>
- [6] - OMRON. (s.f.). *OMRON Automatización Industrial*. [27 de Junio de 2010]. Obtenido de <https://industrial.omron.es/es/products/CJ2M-CPU32>
- [7] - Schneider Electric. (s.f.). *Modicon M241*. [27 de Junio de 2010]. Obtenido de <https://www.se.com/es/es/product-range/62129-modicon-m241/>
- [8] - SEAS. (23 de Septiembre de 2019). *Blog SEAS*. [25 de Junio de 2010]. Obtenido de <https://www.seas.es/blog/automatizacion/la-importancia-de-los-automatas-programables-en-la-industria-actual/>
- [9] - Systerra. (Agosto de 2006). *PICS Simulation Software*. [30 de Junio de 2010]. Obtenido de https://www.systerra.de/downloads/WOD-SW-SST_PICS.pdf



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño

GIEIA

DOCUMENTO 2: PROGRAMACIÓN DEL AUTOMATISMO

Automatización de un proceso de clasificación,
procesado y paletizado de materias primas
mediante autómatas Schneider M241 y software
de simulación Factory I/O

Nombre del alumno: Joaquín Talavera Llamas

Nombre del tutor: Raúl Simarro Fernández

Fecha: Julio de 2020

Grado en Ingeniería Electrónica Industrial y Automática

ÍNDICE DE PROGRAMAS

SFC 1: Máquina control numérico piezas azules

SFC 2: Máquina control numérico piezas verdes

SFC 3: Sistema de sujeción

SFC 4: Sensor de color

SFC 5: Sensor pieza defectuosa

SFC 6: Emergencia y misceláneos

SFC 7: Ascensor pallet

SFC 8: Cintas pallet

SFC 9: Trampilla pallet

SFC 10: Organizador de peso

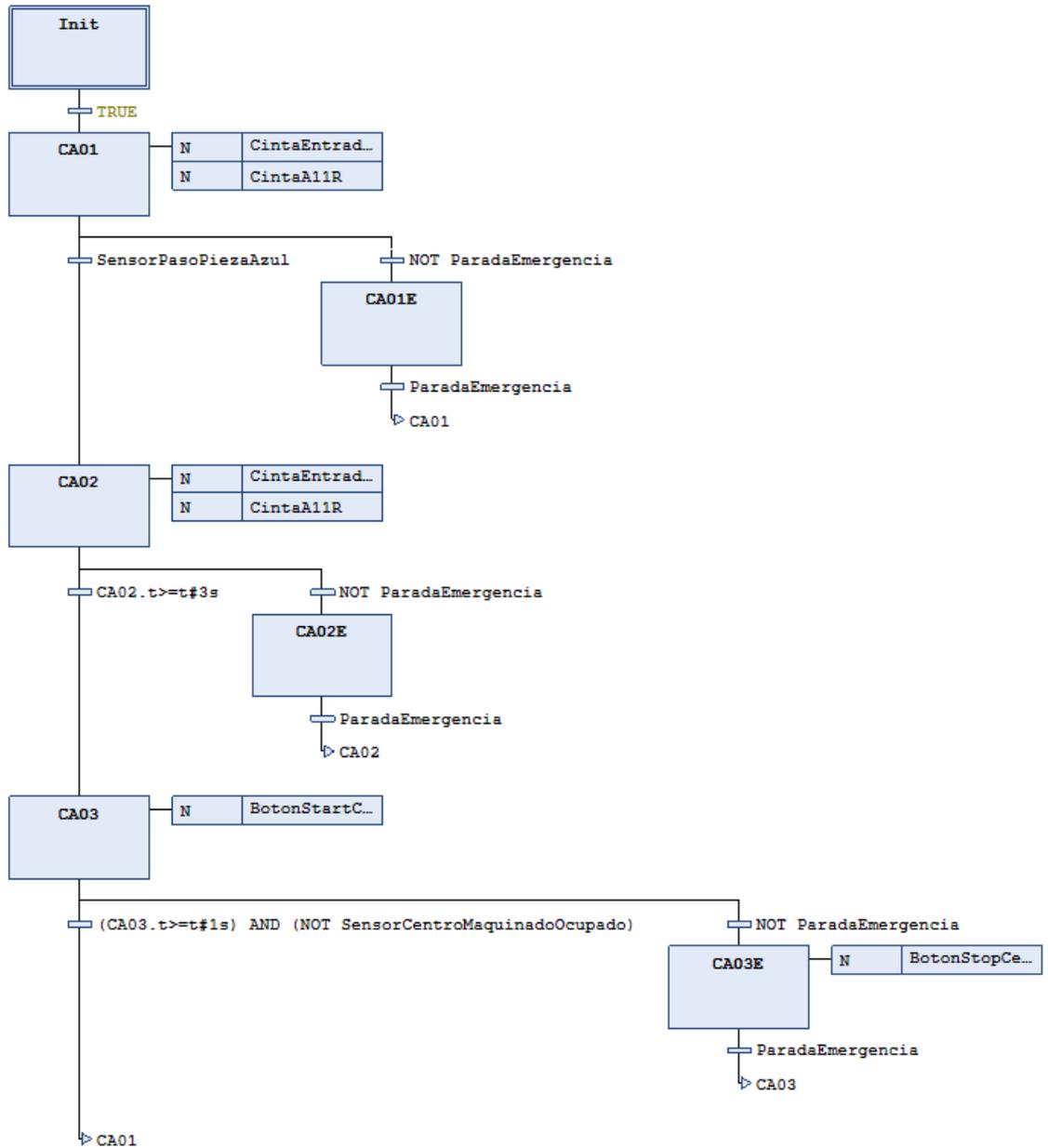
SFC 11: Pick and place

SFC 12: Siempre funcionando

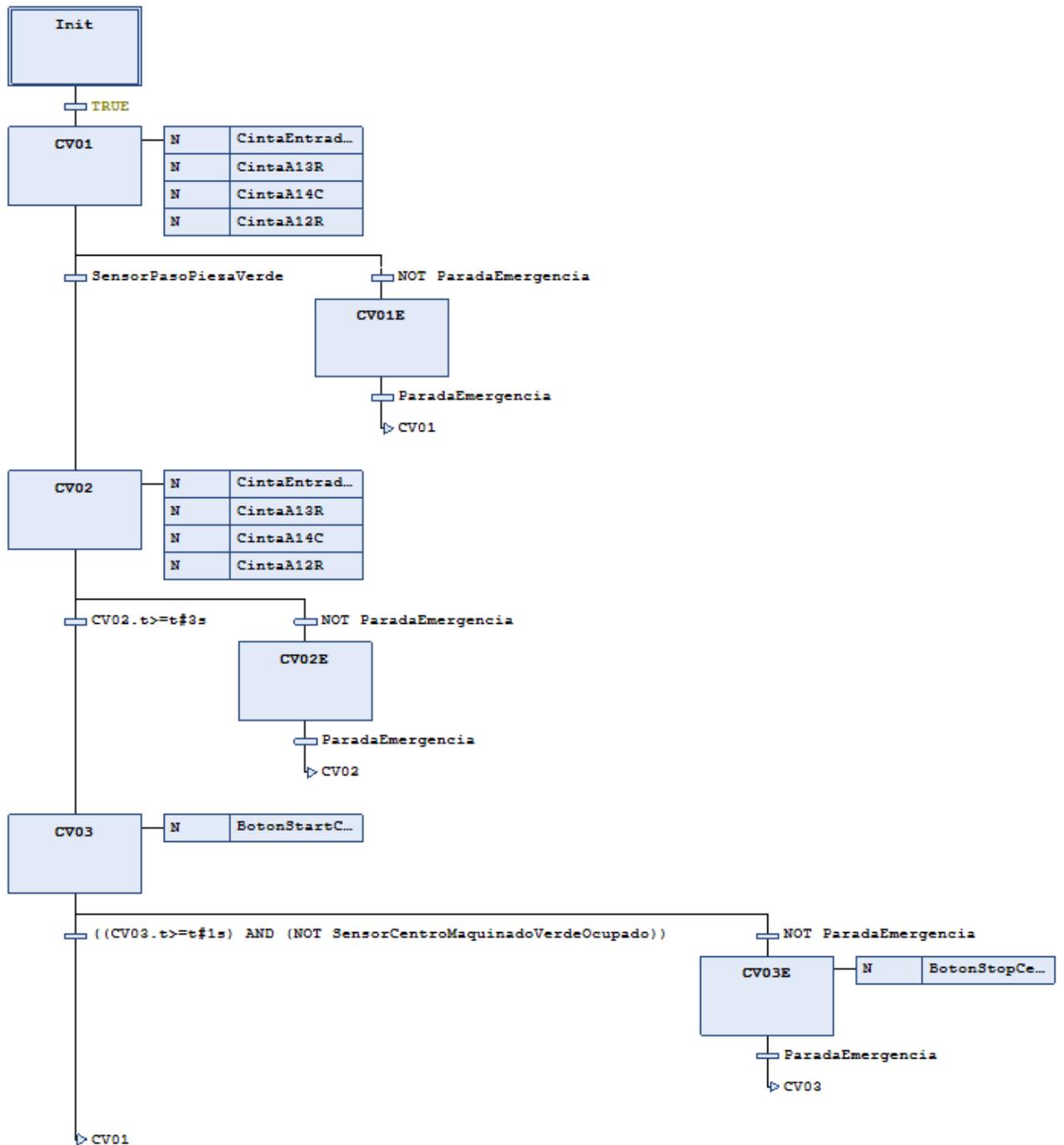
LADDER 1: Contadores

Este anexo contiene todos los POUs pertenecientes al proyecto de automatización y que consta de doce diagramas secuenciales de funciones con la programación de los distintos subprocesos y un *Ladder Diagram* que consta de los contadores utilizados. Todas las variables usadas en ellos pueden consultarse con más detalle en el documento tercero de este proyecto.

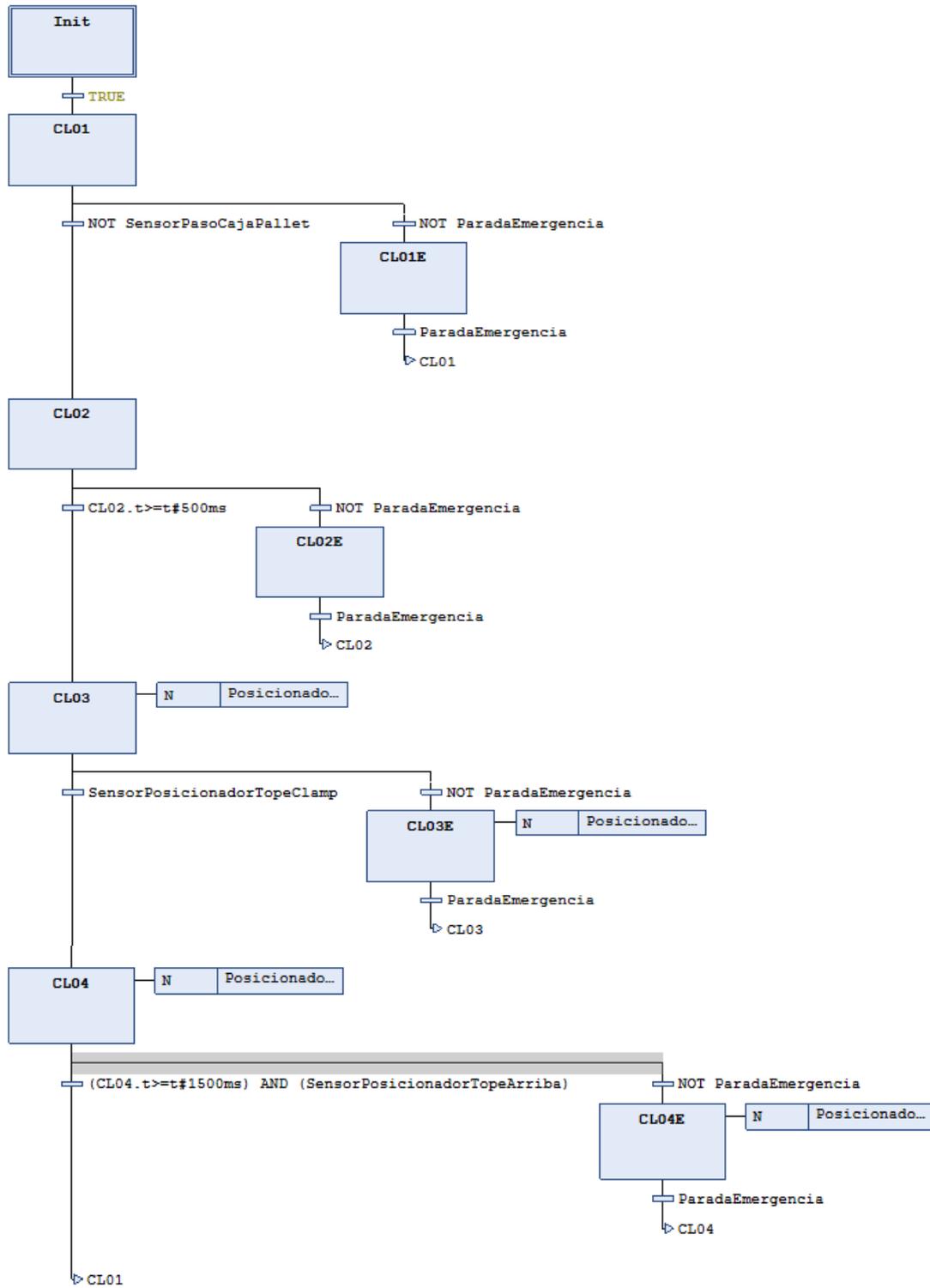
1. Máquina control numérico piezas azules



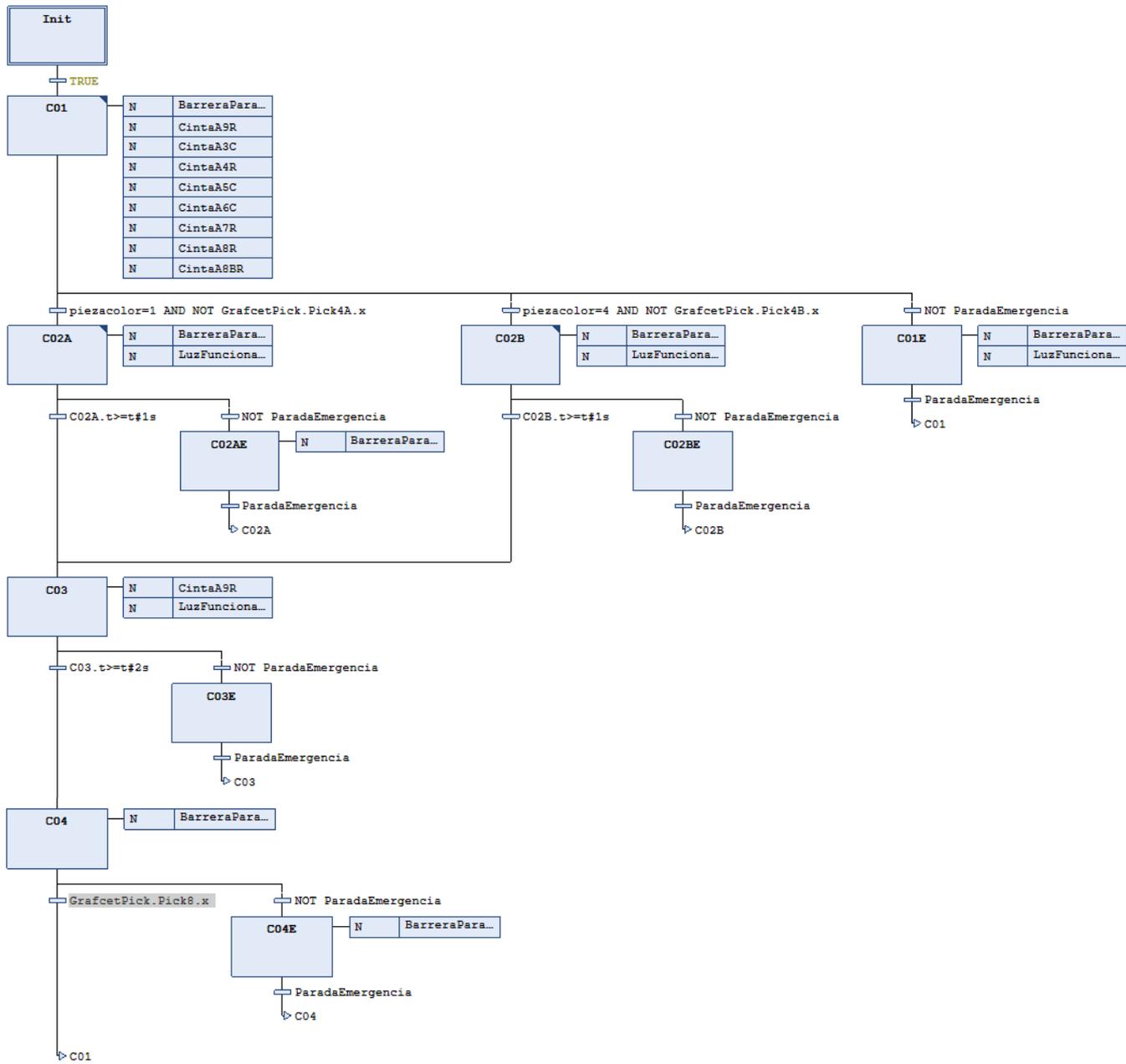
2. Máquina control numérico piezas verdes



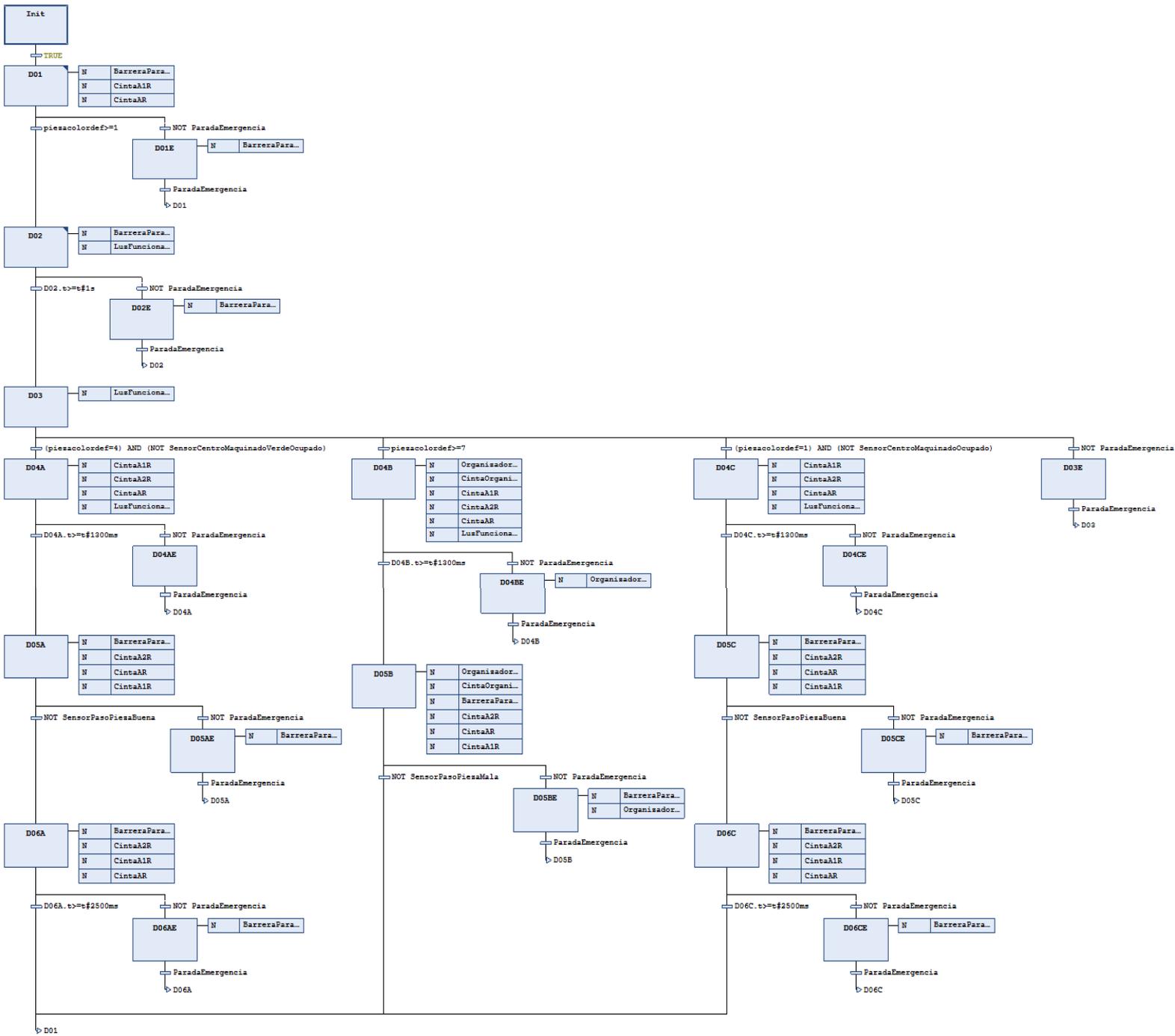
3. Sistema de sujeción



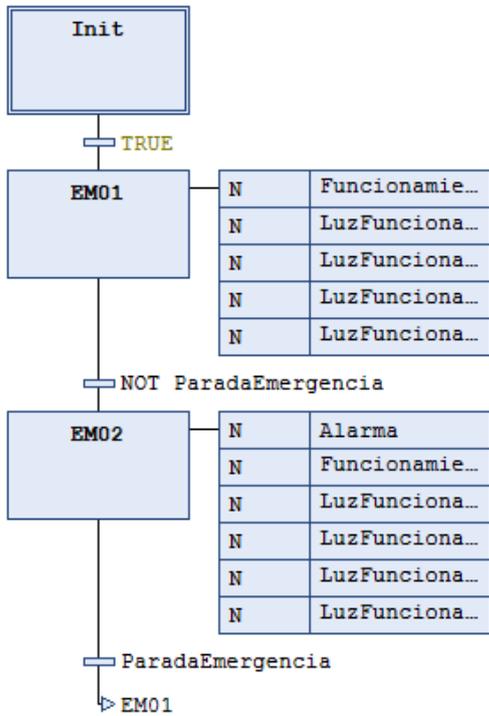
4. Sensor de color



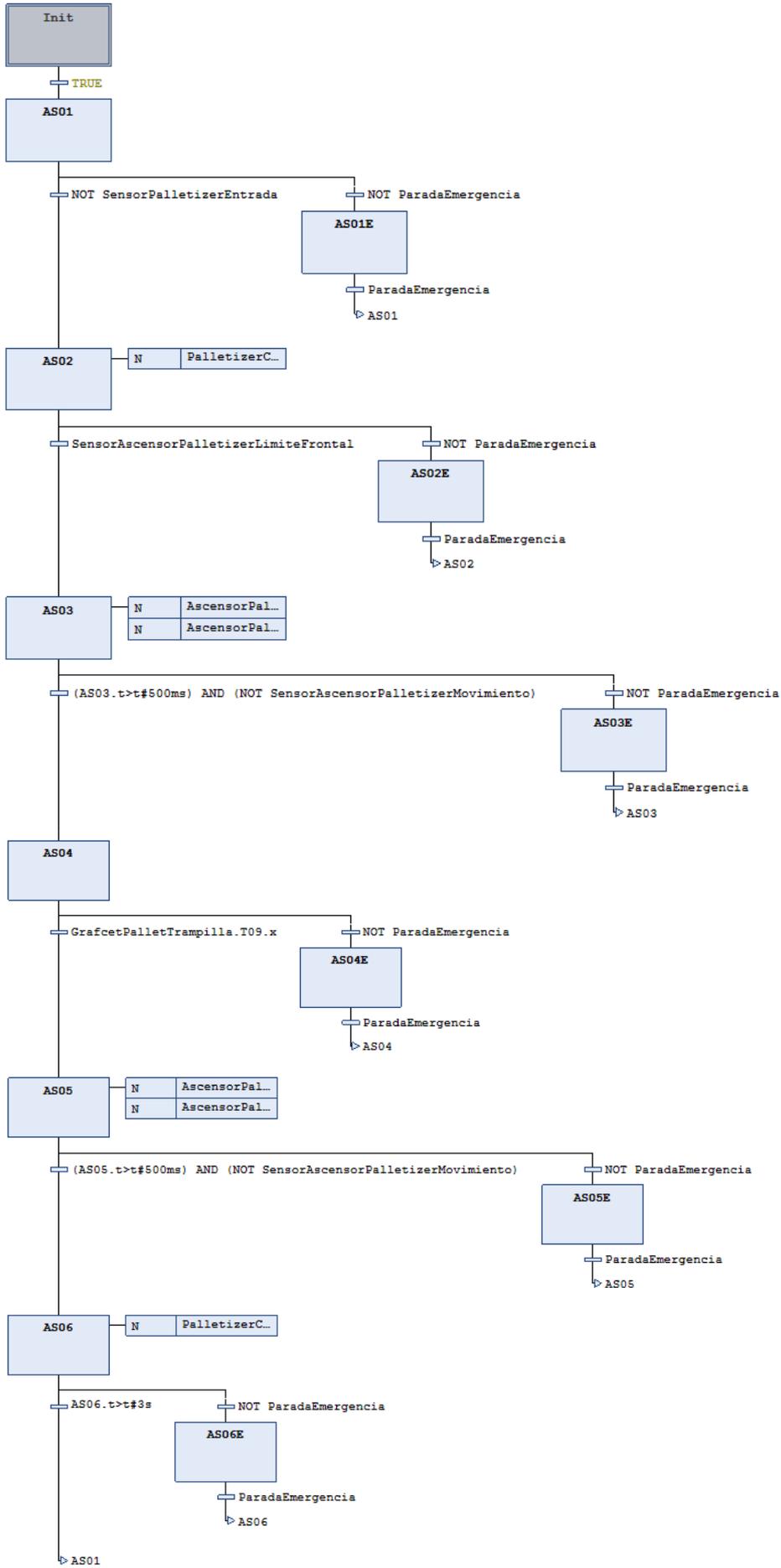
5. Sensor de pieza defectuosa



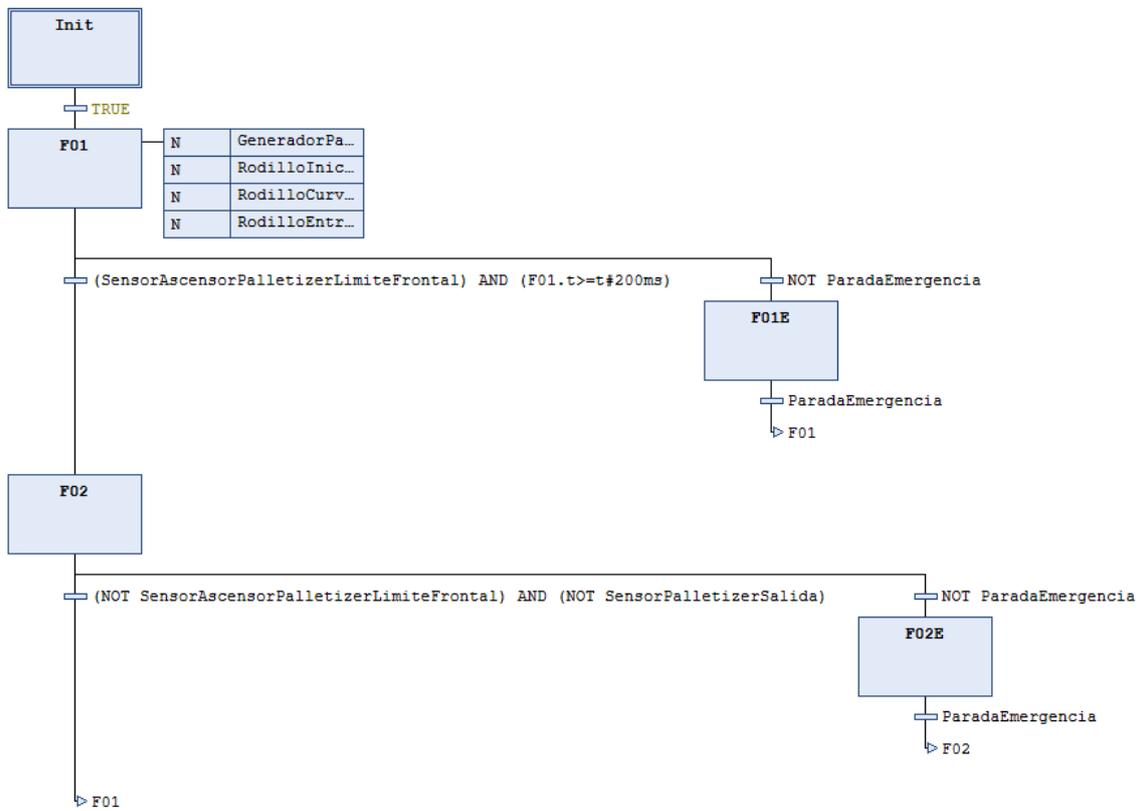
6. Emergencia y misceláneos



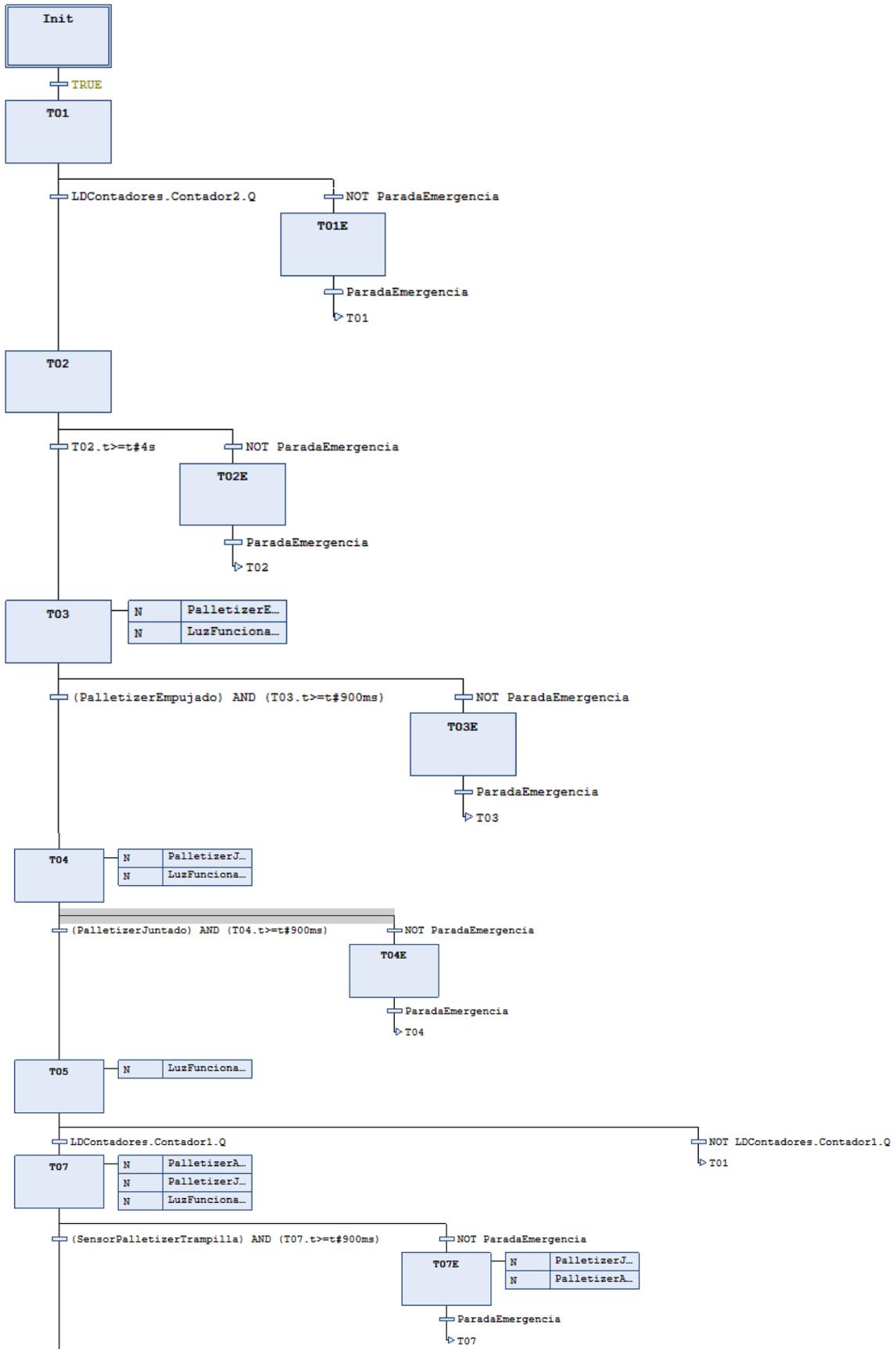
7. Ascensor de pallet

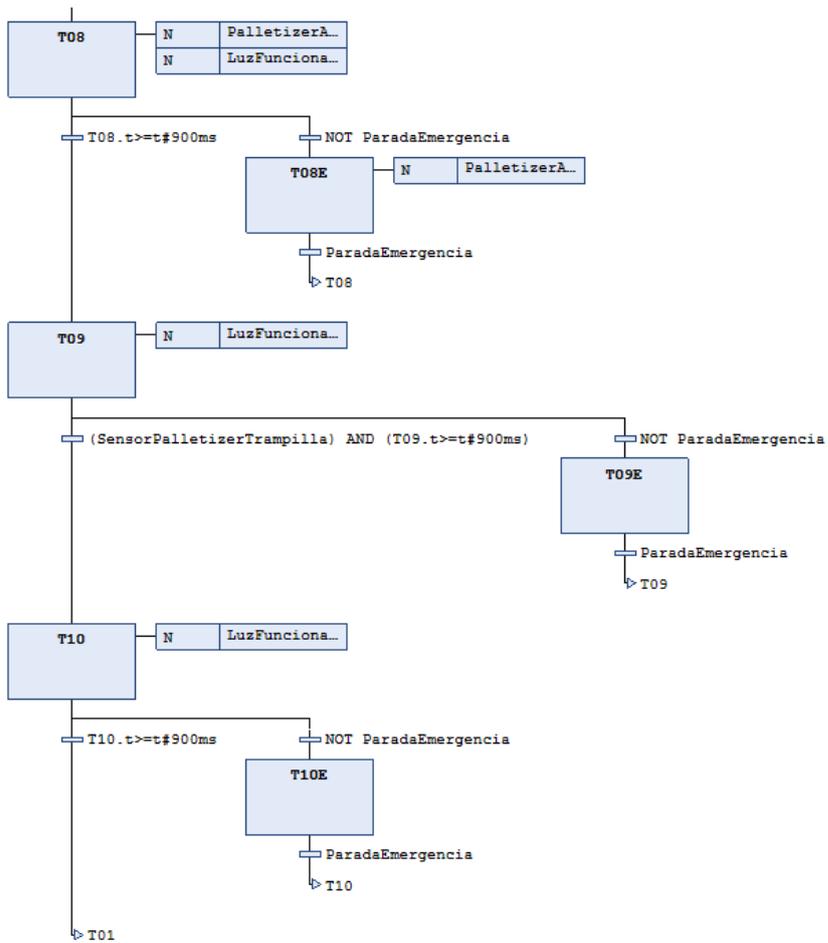


8. Cintas pallet

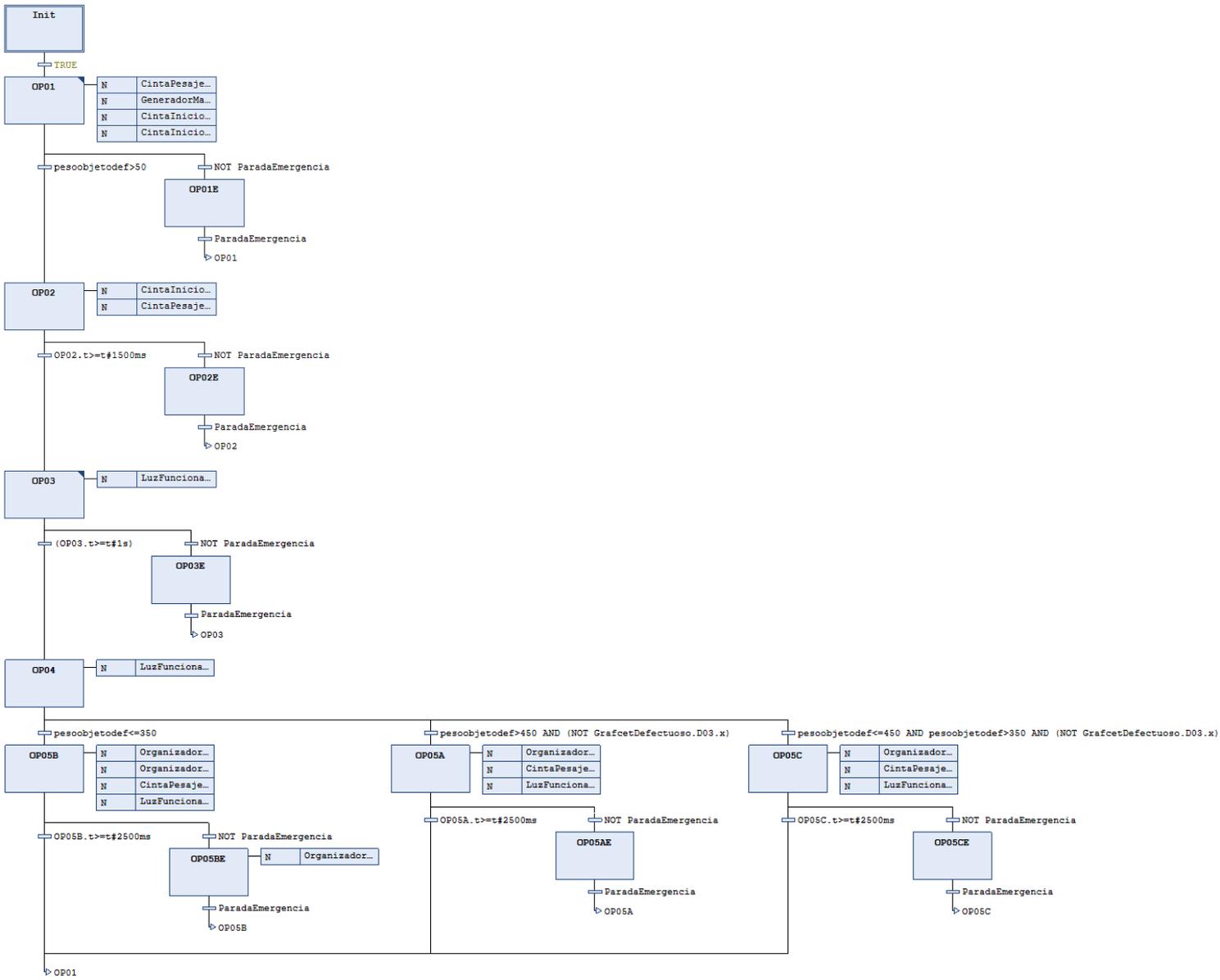


9. Trampilla pallet

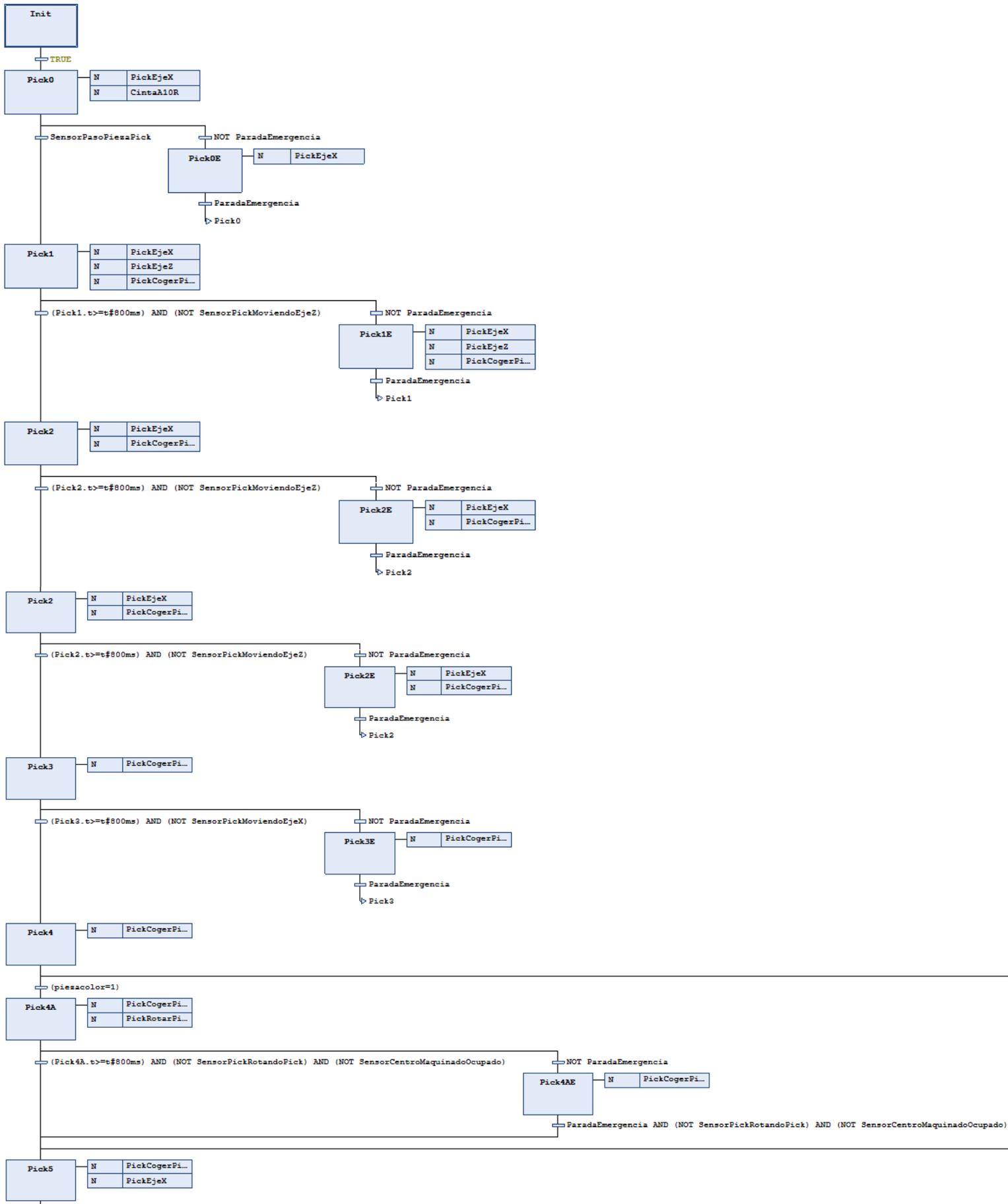


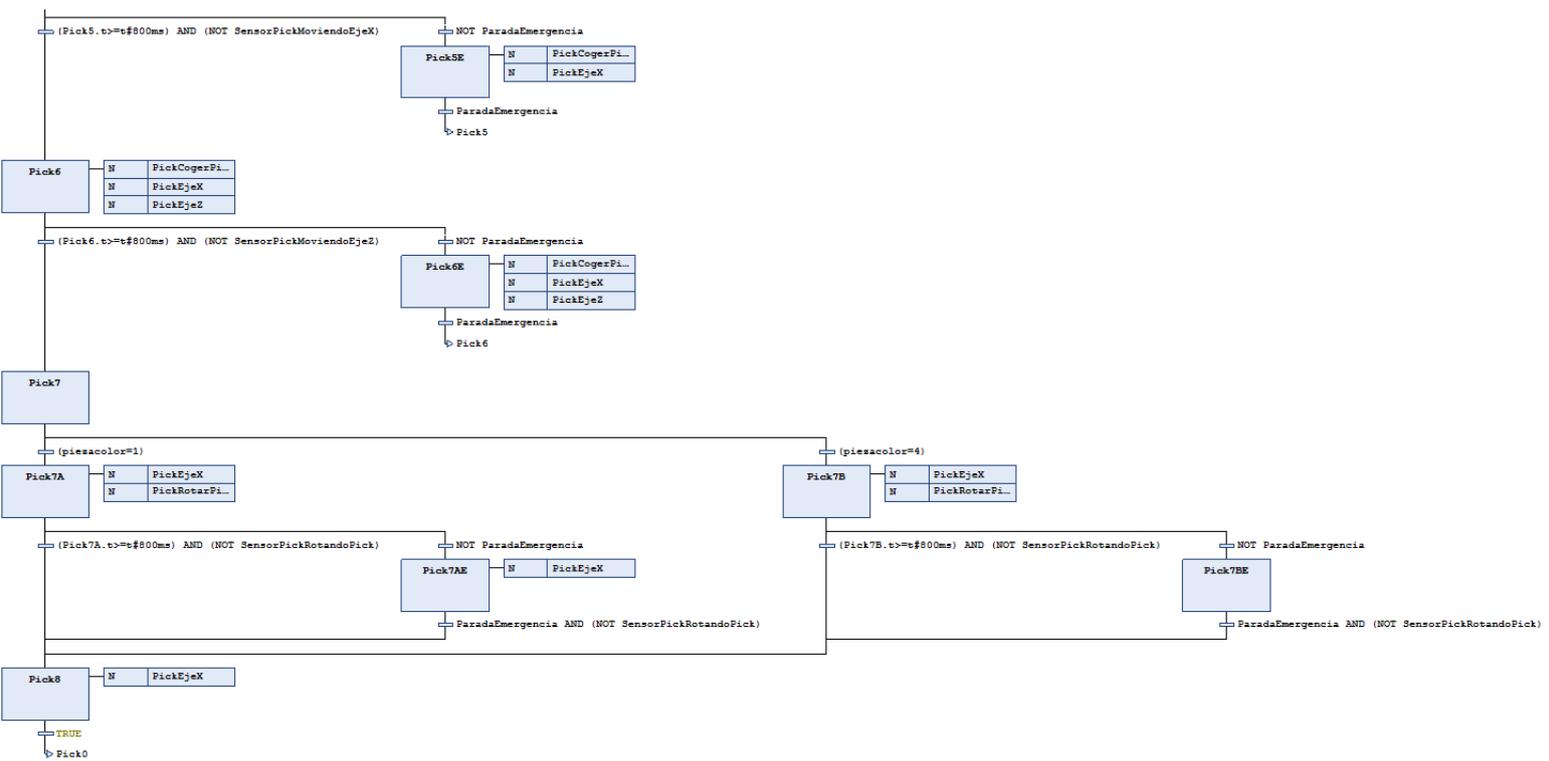


10. Organizador de peso

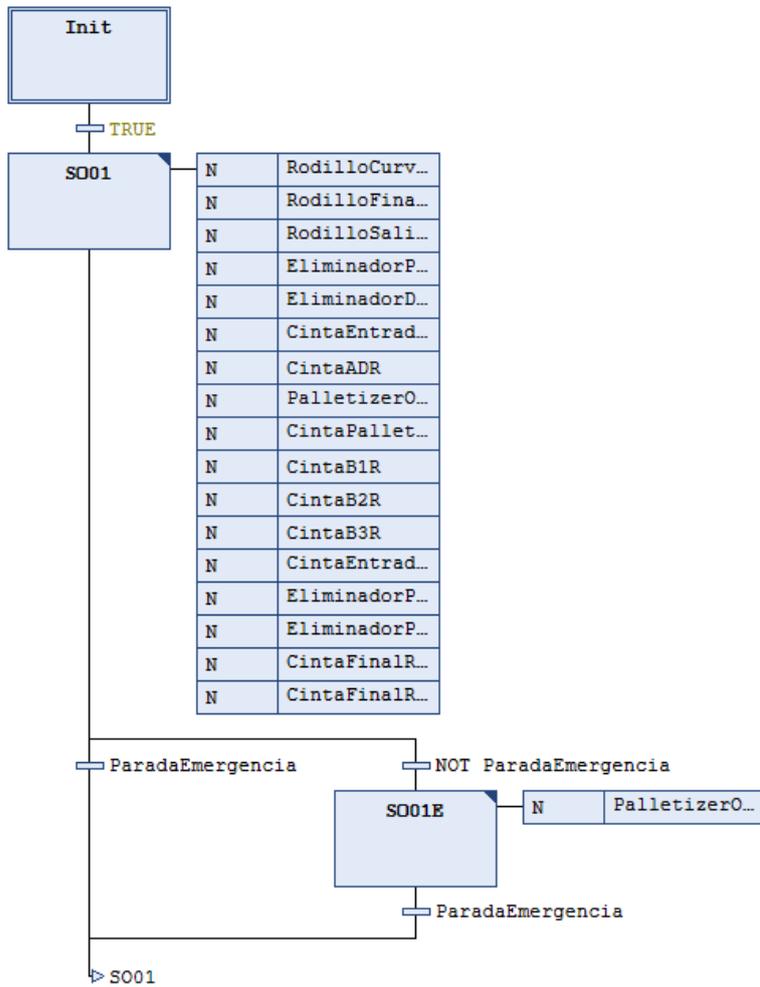


11. Pick and place

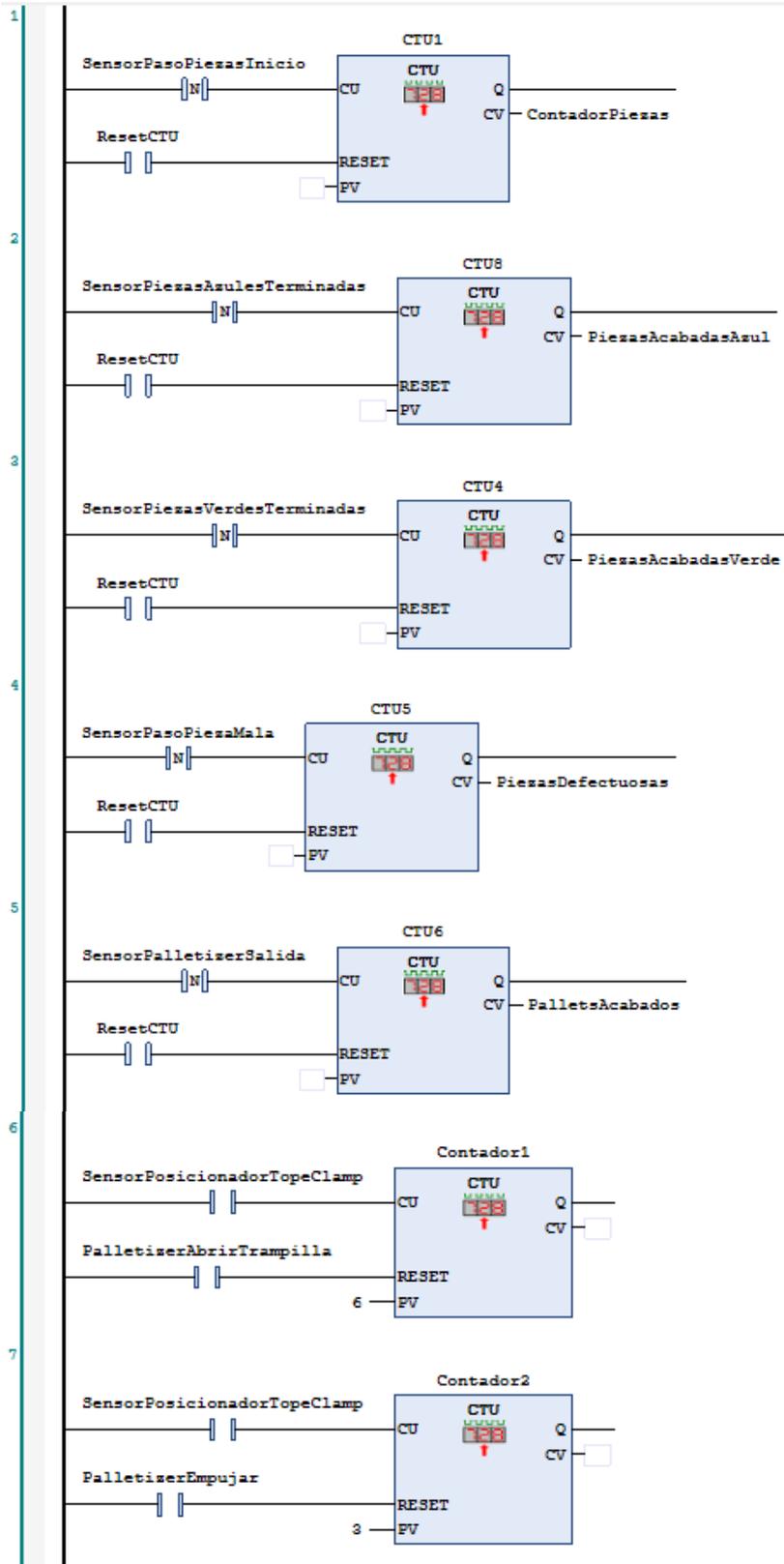




12. Siempre funcionando



13. Contadores





GIEIA

DOCUMENTO 3: ANEXO LISTA DE VARIABLES

Automatización de un proceso de clasificación,
procesado y paletizado de materias primas
mediante autómatas Schneider M241 y software
de simulación Factory I/O

Nombre del alumno: Joaquín Talavera Llamas

Nombre del tutor: Raúl Simarro Fernández

Fecha: Julio de 2020

Grado en Ingeniería Electrónica Industrial y Automática

Entradas digitales

Nombre	Dirección	Tipo	Descripción
BotonResetFabrica	%QX12.4	BOOL	Reseteo de procesos en fábrica
BotonStartFabrica	%QX12.5	BOOL	Comienzo de procesos en fábrica
BotonStopFabrica	%QX12.6	BOOL	Parada de comienzos en fábrica
CentroMaquinadoAbierto	%QX12.7	BOOL	Maquina control numerico azul abierta
PalletizerEmpujado	%QX13.0	BOOL	Sensor activado pestaña empujar cajas
PalletizerJuntado	%QX13.1	BOOL	Sensor activado pestaña juntar cajas
ParadaEmergencia	%QX13.2	BOOL	Seta de emergencia
SensorAscensorPalletizerLimiteFrontal	%QX13.3	BOOL	Cinta frontal ascensor palletizador
SensorAscensorPalletizerLimiteTrasero	%QX13.4	BOOL	Cinta trasera ascensor palletizador
SensorAscensorPalletizerMovimiento	%QX13.5	BOOL	Sensor de movimiento ascensor palletizador
SensorCentroMaquinadoError	%QX13.6	BOOL	Maquina control numerico azul fallo
SensorCentroMaquinadoOcupado	%QX13.7	BOOL	Maquina control numerico azul en proceso
SensorPasoPiezaPick	%QX14.0	BOOL	Sensor de presencia pieza delante pick
SensorPalletizerTrampilla	%QX14.1	BOOL	Sensor trampilla palletizador
SensorPickDetectadoPieza	%QX14.2	BOOL	Pick and place sensor de pieza
SensorPickMoviendoEjeX	%QX14.3	BOOL	Sensor pick and place moviendo en eje x
SensorPickMoviendoEjeZ	%QX14.4	BOOL	Sensor pick and place moviendo en eje z
SensorPickRotandoPick	%QX14.5	BOOL	Sensor pick and place rotando
SensorPickRotandoPieza	%QX14.6	BOOL	Sensor pick and place rotando pieza
SensorPalletizerEntrada	%QX14.7	BOOL	Sensor palletizer entrada cintas
SensorPalletizerSalida	%QX15.0	BOOL	Sensor palletizer salida cintas
SensorPasoPiezaBuena	%QX15.1	BOOL	Sensor paso de pieza no defectuosa
SensorPasoPiezaMala	%QX15.2	BOOL	Sensor paso de pieza no defectuosa
SensorPosicionadorTopeArriba	%QX15.3	BOOL	Sensor de alineador cajas levantado
SensorPosicionadorTopeClamp	%QX15.4	BOOL	Sensor de alineador cajas presionado
SensorPasoCajaPallet	%QX15.5	BOOL	Sensor de paso de pallets
SensorCentroMaquinadoVerdeError	%QX15.6	BOOL	Maquina control numerico verde fallo
SensorCentroMaquinadoVerdeOcupado	%QX15.7	BOOL	Maquina control numerico verde en proceso
SensorPasoPiezaAzul	%QX16.0	BOOL	Sensor paso pieza de color azul
SensorPasoPiezaVerde	%QX16.1	BOOL	Sensor paso pieza de color verde
SensorPasoPiezasInicio	%QX16.2	BOOL	Sensor paso materias primas al inicio
SensorPiezasAzulesTerminadas	%QX16.3	BOOL	Sensor paso de piezas azules terminadas
SensorPiezasVerdesTerminadas	%QX16.4	BOOL	Sensor paso de piezas verdes terminadas

Salidas digitales

Nombre	Dirección	Tipo	Descripción
Alarma	%QX0.0	BOOL	Alarma sonora parada de emergencia
AscensorPalletizerBajar	%QX0.1	BOOL	Ascensor palletizador accion bajar
AscensorPalletizerMoverLimite	%QX0.2	BOOL	Ascensor palletizador habilitar limites
AscensorPalletizerSubir	%QX0.3	BOOL	Ascensor palletizador accion bajar
BarreraParadaColor	%QX0.4	BOOL	Accionar barrera parada en sensor color
BarreraParadaControl	%QX0.5	BOOL	Accionar barrera parada en sensor defectuosos
BotonResetCentroMaquinadoAzul	%QX0.6	BOOL	Boton reset maquina control numerico azul
BotonResetFabricaLuz	%QX0.7	BOOL	Luz de boton de reset de fabrica
BotonStartCentroMaquinadoAzul	%QX1.0	BOOL	Boton start maquina control numerico azul
BotonStartFabricaLuz	%QX1.1	BOOL	Luz de boton de start de fabrica
BotonStopCentroMaquinadoAzul	%QX1.2	BOOL	Boton stop maquina control numerico azul
BotonStopFabricaLuz	%QX1.3	BOOL	Luz de boton de stop de fabrica
CentroMaquinadoProducir	%QX1.4	BOOL	Maquina control numerico azul produciendo
CintaA1R	%QX1.5	BOOL	Cinta 1 camino A recta
CintaA2R	%QX1.6	BOOL	Cinta 2 camino A recta
CintaA3C	%QX1.7	BOOL	Cinta 3 camino A curva
CintaA4R	%QX2.0	BOOL	Cinta 4 camino A recta
CintaA5C	%QX2.1	BOOL	Cinta 5 camino A curva
CintaA6C	%QX2.2	BOOL	Cinta 6 camino A curva
CintaA7R	%QX2.3	BOOL	Cinta 7 camino A recta
CintaA8R	%QX2.4	BOOL	Cinta 8 camino A recta
CintaA9R	%QX2.5	BOOL	Cinta 9 camino A recta
CintaA10R	%QX2.6	BOOL	Cinta 10 camino A recta
CintaA11R	%QX2.7	BOOL	Cinta 11 camino A recta
CintaA12R	%QX3.0	BOOL	Cinta 12 camino A recta
CintaADR	%QX3.1	BOOL	Cinta piezas defectuosas
CintaB1R	%QX3.2	BOOL	Cinta 1 camino B recta
CintaB2R	%QX3.3	BOOL	Cinta 2 camino B recta
CintaB3R	%QX3.4	BOOL	Cinta 3 camino B recta
CintaEntradaCentroMaquinadoAzul	%QX3.5	BOOL	Cinta previa a entrada de maquina control azul
CintaA13R	%QX3.6	BOOL	Cinta 13 camino A recta
CintaFinalRectaAzul	%QX3.7	BOOL	Cinta final recorrido piezas azules
CintaInicioCurva	%QX4.0	BOOL	Cinta inicial fabrica curva
CintaInicioRecta	%QX4.1	BOOL	Cinta inicial fabrica recta
CintaOrganizadorCalidadAdelante	%QX4.2	BOOL	Cinta organizador piezas defectuosas adelante
CintaOrganizadorCalidadAtras	%QX4.3	BOOL	Cinta organizador piezas defectuosas atrás
CintaPalletizerAtras	%QX4.4	BOOL	Cinta previa a palletizador adelante
CintaPalletizerDelante	%QX4.5	BOOL	Cinta previa a palletizador atrás
CintaPesajeAdelante	%QX4.6	BOOL	Cinta maquina de pesaje adelante

CintaPesajeAtras	%QX4.7	BOOL	Cinta maquina de pesaje atrás
EliminadorDefectuosos	%QX5.0	BOOL	Eliminador piezas defectuosas
EliminadorPallets	%QX5.1	BOOL	Eliminador pallets acabados
EliminadorPiezasVerdeFin	%QX5.2	BOOL	Eliminador piezas verdes acabadas
EliminadorPiezasAzulFin	%QX5.3	BOOL	Eliminador piezas azules acabadas
FuncionamientoCorrecto	%QX5.4	BOOL	Luz funcionamiento correcto
FuncionamientoDetenido	%QX5.5	BOOL	Luz funcionamiento detenido parada emer
CintaA14C	%QX5.6	BOOL	Cinta 14 camino A curva
GeneradorMaterial	%QX5.7	BOOL	Generador materia prima
GeneradorPallets	%QX6.0	BOOL	Generador pallets
LuzFuncionamientoOK1	%QX6.1	BOOL	Luz funcionamiento OK sensor pieza defecto
LuzFuncionamientoOK2	%QX6.2	BOOL	Luz funcionamiento OK sensor pieza color
LuzFuncionamiento1	%QX6.3	BOOL	Luz funcionando sensor pieza defecto
LuzFuncionamiento2	%QX6.4	BOOL	Luz funcionando sensor pieza color
LuzFuncionamientoSTOP1	%QX6.5	BOOL	Luz stop sensor pieza defecto
LuzFuncionamientoSTOP2	%QX6.6	BOOL	Luz stop sensor pieza color
OrganizadorDefectuosoGirar	%QX6.7	BOOL	Organizador piezas defectuosas girar
OrganizadorPesoA	%QX7.0	BOOL	Organizador peso direccion recto
OrganizadorPesoB	%QX7.1	BOOL	Organizador peso direccion derecha
OrganizadorPesoC	%QX7.2	BOOL	Organizador peso direccion izquierda
PalletizerAbrirTrampilla	%QX7.3	BOOL	Abrir trampilla palletizador
PalletizerCadenaAdelante	%QX7.4	BOOL	Palletizador cinta cadenas pallets adelante
PalletizerCadenaAtras	%QX7.5	BOOL	Palletizador cinta cadenas pallets atrás
PalletizerEmpujar	%QX7.6	BOOL	Activar pestaña empujar cajas palletizador
PalletizerJuntar	%QX7.7	BOOL	Activar pestaña juntar cajas palletizador
PalletizerOrientarCaja	%QX8.0	BOOL	Palletizador activar resorte organizador
PickCogerPieza	%QX8.1	BOOL	Pick and place coger pieza succión
PickEjeX	%QX8.2	BOOL	Pick and place mover en eje X
PickEjeZ	%QX8.3	BOOL	Pick and place mover en eje Z
PickRotarPickAntiReloj	%QX8.4	BOOL	Pick and place rotar direccion antireloj
PickRotarPickReloj	%QX8.5	BOOL	Pick and place rotar direccion reloj
PickRotarPiezaAntiReloj	%QX8.6	BOOL	Pick and place rotar pieza direccion antireloj
PickRotarPiezaReloj	%QX8.7	BOOL	Pick and place rotar pieza direccion reloj
RodilloCurvaPalletsIn	%QX9.0	BOOL	Cinta rodillos curva pallets entrada
RodilloCurvaPalletsOut	%QX9.1	BOOL	Cinta rodillos curva pallets salida
RodilloEntradaPalletizer	%QX9.2	BOOL	Cinta rodillos entrada a palletizador
RodilloFinalPallets	%QX9.3	BOOL	Cinta rodillos final recorrido pallets
RodilloInicioPallets	%QX9.4	BOOL	Cinta rodillos inicio recorrido pallets
RodilloSalidaPalletizer	%QX9.5	BOOL	Cinta rodillos entrada a palletizador
CintaAR	%QX9.6	BOOL	Cinta comienzo camino A recta
PosicionadorArriba	%QX9.7	BOOL	Activar posicionador posicion arriba
PosicionadorClamp	%QX10.0	BOOL	Activar posicionador posicion contraer
CintaEntradaCentroMaquinadoVerde	%QX10.1	BOOL	Cinta entrada a máquina control numerico azul
CintaFinalRectaVerde	%QX10.2	BOOL	Cinta final recorrido piezas verdes
BotonStartCentroMaquinadoVerde	%QX10.3	BOOL	Boton start maquina control numerico verde

BotonResetCentroMaquinadoVerde	%QX10.4	BOOL	Boton reset maquina control numerico verde
BotonStopCentroMaquinadoVerde	%QX10.5	BOOL	Boton stop maquina control numerico verde
CintaA8BR	%QX10.6	BOOL	Cinta 8 bajada camino A recto
LuzFuncionamientoOK3	%QX10.7	BOOL	Luz funcionamiento OK palletizador
LuzFuncionamiento3	%QX11.0	BOOL	Luz funcionando palletizador
LuzFuncionamientoSTOP3	%QX11.1	BOOL	Luz emergencia palletizador
LuzFuncionamientoOK4	%QX11.2	BOOL	Luz funcionamiento OK sensor peso
LuzFuncionamiento4	%QX11.3	BOOL	Luz funcionando sensor peso
LuzFuncionamientoSTOP4	%QX11.4	BOOL	Luz stop sensor peso
NoParadaEmergencia	%QX11.5	BOOL	Variable valor contrario parada emergencia

Entradas analógicas

Nombre	Dirección	Tipo	Descripción
CentroMaquinadoProceso	%MW10	INT	Estado maquina control numérico azul
SensorPesaje	%MW11	INT	Sensor de peso
SensorVisionColorValor	%MW12	INT	Sensor de color
SensorVisionDefectuoso	%MW13	INT	Sensor de color pieza defectuosa

Variables manipulables

Nombre
piezacolor
numerocajas
piezacolordef
pesoobjetodef

Tipo	Descripción
INT	Valor pieza color
INT	Valor número de cajas
INT	Valor memoria pieza color
REAL	Valor memoria peso objeto

Salidas Analógicas

Nombre	Dirección	Tipo	Descripción
ContadorPiezas	%MW0	INT	Contador total piezas
PiezasAcabadasAzul	%MW1	INT	Contador piezas azules acabadas
PiezasAcabadasVerde	%MW2	INT	Contador piezas verdes acabadas
PiezasDefectuosas	%MW3	INT	Contador piezas defectuosas
PalletsAcabados	%MW4	INT	Contador pallets acabados



GIEIA

DOCUMENTO 4: PRESUPUESTO

Automatización de un proceso de clasificación,
procesado y paletizado de materias primas
mediante autómatas Schneider M241 y software
de simulación Factory I/O

Nombre del alumno: Joaquín Talavera Llamas

Nombre del tutor: Raúl Simarro Fernández

Fecha: Julio de 2020

Grado en Ingeniería Electrónica Industrial y Automática

A continuación se realiza el cálculo del presupuesto de realización del proyecto usando valores orientativos en el mercado actual:

COMPONENTES Y LICENCIAS	UNIDADES	PRECIO (€)	TOTAL (€)
Autómata Schneider M241	1	412,25	412,25 €
Ordenador gama media	1	450,00	350,00 €
Licencia SoMachine	1	0,00	0,00 €
Licencia Factory I/O (licencia permanente)	1	395,00	395,00 €
TOTAL			1157,25 €

MANO DE OBRA	HORAS	PRECIO (€/h)	TOTAL (€)
Ingeniero técnico investigación previa y reunión con cliente	10	24,07	240,70
Ingeniero técnico diseño del automatismo inicial	165	24,07	3.971,55
Ingeniero técnico corrección de errores	25	24,07	601,75
Ingeniero técnico implementación de mejoras	100	24,07	2407,00
Técnico de instalación de autómata	2	40,00	80,00 €
TOTAL			7.301,00 €

*Nota: se ha hecho el cálculo de las horas necesarias de ingeniero técnico como las horas que ha dedicado el alumno a la realización del proyecto.

RESUMEN	UNIDAD	PRECIO (€)	TOTAL (€)
Componentes y licencias	1	1.157,25	1.157,25 €
Mano de obra	1	7.301,00	7.301,00 €
TOTAL			8.458,25 €

El sumatorio total de coste para realizar este proyecto es de 8.458,25€, ya que el precio de la maquinaria existente en la factoría y la infraestructura de telecomunicaciones no se ha tenido en cuenta al suponerse que ya se encuentra allí.

