



**COMILLAS**  
UNIVERSIDAD PONTIFICIA

**ICAI**

# GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

## TRABAJO FIN DE GRADO AUTOMATIZACIÓN FLEXIBLE DE LA MINIFÁBRICA ICAI

Autor: Rodrigo Marty Plaza

Director: José Antonio Rodríguez Mondéjar

Madrid

julio de 2021

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título Automatización Flexible de la Minifábrica ICAI en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 4º es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo: 

Fecha: 19/ 07/ 2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.:

Fecha: 21/ 07/2021



**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

## TRABAJO FIN DE GRADO AUTOMATIZACIÓN FLEXIBLE DE LA MINIFÁBRICA ICAI

Autor: Rodrigo Marty Plaza

Director: José Antonio Rodríguez Mondéjar

Madrid

julio de 2021

# AUTOMATIZACIÓN FLEXIBLE DE LA MINIFÁBRICA ICAI

Autor: **Marty Plaza, Rodrigo.**

Director: Rodríguez Mondéjar, José Antonio.

Entidad Colaboradora: ICAI-Universidad Pontificia Comillas.

## RESUMEN DEL PROYECTO

### Introducción

#### *Planteamiento del problema*

En medio de una cuarta revolución industrial, en la que la automatización juega un papel esencial, es de vital importancia que las empresas españolas se sumen a esta revolución y empiecen a adoptar cambios en su sistema de producción.

Las grandes empresas, como las de la automoción (España produce 3 millones de automóviles al año [ECO21]) ya están tomando medidas, pero a las pymes (gran parte del tejido empresarial español) les está costando más trabajo [DEDI18].

Por ello, todas las técnicas que permitan una mayor facilidad para estas empresas para automatizar sus procesos y flexibilizar sus sistemas de producción pueden ser de mucha utilidad.

Una en particular es la que se investiga en este proyecto: Pack ML.

#### *Estado de la técnica*

Pack ML es una plantilla para automatizar elaborada por la OMAC (Organization for Machine Automation and Control), que mejora las comunicaciones entre el operador y la máquina, proponiendo una estructura común para el código en el que se programa la máquina, independientemente de su origen o de su fabricante. [OMAC21].

Conociendo esta plantilla, el operador podrá entender mejor el funcionamiento de cualquier máquina, haciendo más rápida la introducción de nuevos equipos en las fábricas.

La estructura común que propone Pack ML se basa en organizar el código de programación en bloques. Primero, unos bloques para los modos de producción o funcionamiento que puede tener la máquina (producción normal o automática, producción manual, mantenimiento, limpieza, etc.).

En segundo lugar, dentro de cada modo de producción, podrá haber hasta 17 bloques que definen el estado de funcionamiento de la máquina: un estado de funcionamiento normal, un estado que indica que la máquina se ha detenido a causa de una situación

peligrosa, otro que indica que la máquina se ha detenido porque la producción ha terminado, etc.

La plantilla cuenta con variables que indican en qué estado se encuentra el automatismo, de manera que con una simple interfaz hombre máquina, el operador puede conocer a tiempo real en qué estado se encuentra la máquina. También puede conocer fácilmente otros parámetros, como cuánto tiempo lleva la máquina en un estado, velocidad...

Además de la mejora en las comunicaciones, la perfecta organización del código que ofrece Pack ML facilita mucho la tarea del programador, lo que se traduce en una mayor rapidez a la hora de programar, menor número de fallos y fácil detección de estos.

### *Objeto del proyecto*

En este proyecto se quiere demostrar las ventajas que ofrece la plantilla Pack ML en la automatización de una cadena de montaje.

Para ello, se ha elaborado un código con esta plantilla para programar el funcionamiento automático de la cadena de montaje Minifábrica-ICAI, situada en el laboratorio de sistemas digitales de la Escuela Técnica Superior de Ingeniería ICAI.

Posteriormente, también se elaboró una interfaz para comprobar el funcionamiento del código y demostrar las ventajas que ofrece Pack ML.

### Metodología

#### *Herramientas*

Para la elaboración del código y su simulación, se ha utilizado la aplicación para programación de PLC TIA Portal (Siemens).

Gracias a una herramienta del grupo Siemens denominada PLCSim, con la que se puede cargar y simular el código en un PLC virtual, la totalidad del proyecto se ha realizado desde casa, utilizando como estación de programación un ordenador portátil.

Todo el código se ha programado en lenguaje KOP (diagrama de contactos) y SFC (Sequential Function Chart).

SFC es un lenguaje de programación de tipo gráfico para describir el funcionamiento de sistemas secuenciales [MOND20]. Está basado en la metodología y lenguaje GRAFCET, por lo que también se ha investigado mucho acerca de esta metodología con el objetivo de explotar sus ventajas, ya que es una manera de escribir código muy clara y visual.

### Descripción de la minifábrica ICAI

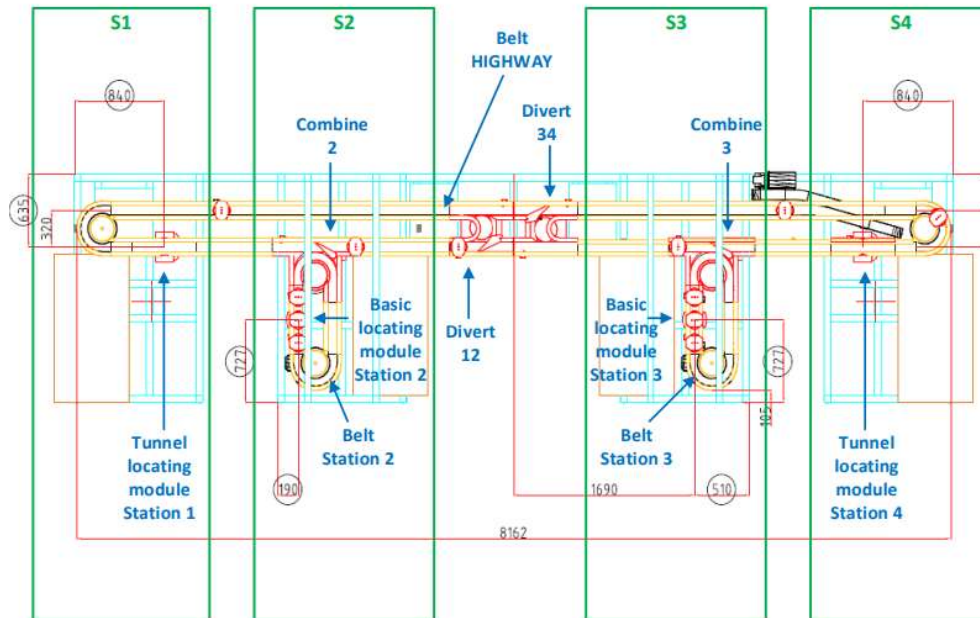


Figura 1: plano de la minifábrica [MOND20]

Los elementos de la minifábrica que se automatizan en este proyecto son:

**Cintas transportadoras.** La fábrica cuenta con una cinta principal y dos cintas satélites. Su funcionamiento es sencillo y en este proyecto se controlan con una variable binaria que activa o desactiva el motor eléctrico que permite su movimiento.

**Posicionadores de tipo básico.** La función de los posicionadores es retener el palé que transporta el producto a elaborar en la fábrica mientras que el robot realiza una acción sobre ese producto. Concretamente, los posicionadores de tipo básico retienen el palé mediante dispositivos inmovilizadores.

**Posicionadores de tipo túnel.** La función es la misma que la de los posicionadores de tipo básico, pero la manera de retener el palé es elevándolo de la cinta.

**Módulos de desviación/combinación interespacial.** Son unos dispositivos en forma de aguja que dividen la cinta principal en dos, permitiendo, si es necesario, que ésta se divida en dos módulos independientes.

**Módulos combinados de desviación/combinación.** Dispositivos también en forma de aguja que permiten a los palés moverse de la cinta principal a las satélites o viceversa.

Cada uno de estos dispositivos se controla mediante un bloque de código programado en SFC (también llamado GRAFCET).

Lejos de funcionar de manera independiente, estos dispositivos se comunican entre sí. Para la programación de esta comunicación se ha investigado acerca de la guía GEMMA (plantilla de automatización semejante a Pack ML, que establece instrucciones para la comunicación entre GRAFCETs).

El funcionamiento de estos elementos es distinto según si la fábrica se encuentra en un estado de la plantilla Pack ML u otro. En este proyecto, se han programado 3 funcionamientos de la fábrica: normal (producción automática), parada de fin de ciclo y parada de emergencia.

### Resultados

El resultado de este proyecto es un código, listo para ser introducido en los controladores de la minifábrica, habiéndose comprobado su funcionamiento mediante su simulación y forzado de variables.

Mediante la simulación se ha podido comprobar que los elementos tienen un funcionamiento correcto, acorde con las especificaciones marcadas durante su diseño (especificaciones en las que la seguridad de personas y equipos ha tenido un papel muy importante).

Se comprueba también que la comunicación entre dispositivos es correcta y que se sincronizan para permitir el movimiento de palés a lo largo de la fábrica perfectamente.

El cambio de estados de la plantilla Pack ML (producción normal a parada de emergencia, parada de fin de ciclo a parada de emergencia, etc.) es inmediata y en el momento exacto en el que se tiene que producir.

Tras una parada de emergencia, el sistema permanece en un estado de parada segura, con los dispositivos apagados, y no vuelve a funcionar incluso si se deja de accionar la seta de emergencia, cumpliendo con la normativa ISO 13850. Tan solo se podrá reiniciar el sistema, cuando se haya rearmado dicha seta.

La parada de fin de ciclo se encarga de recoger los productos terminados y deja al sistema preparado para una nueva puesta en marcha.

Y el estado de producción automática funciona correctamente y es lo suficientemente flexible como para poderse adaptar a la fabricación de numerosos productos, siempre que esta fabricación se produzca en cuatro etapas.

### Conclusiones

Se ha comprobado con este proyecto que la comunicación hombre-máquina mejora mucho con Pack ML. Con la ayuda de una simple interfaz como la siguiente:



*Figura 2: Posible HMI para controlar el sistema*

el operador puede saber en todo momento en qué estado se encuentra la máquina y accionar los comandos necesarios para pasar de un estado a otro. Con unos mínimos conocimientos de esta plantilla, podría controlar cualquier máquina del mercado que funcionase con Pack ML (como es un estándar para automatizar, se espera que cada vez más máquinas utilicen esta plantilla).

Por otra parte, se han comprobado que las ventajas que ofrece Pack ML a la hora de programar automatismos son muchas.

En primer lugar, permite programar de manera muy sencilla el cambio de un estado de funcionamiento a otro, no teniéndose que preocupar el programador por habilitar o inhabilitar ningún GRAFCET.

En segundo lugar, la perfecta organización con la que se puede estructurar el código con esta plantilla permite dividirlo en bloques, favoreciendo el reparto de trabajo con otros programadores y la posibilidad de ir probando el funcionamiento de cada bloque por separado, reduciendo enormemente la posibilidad de que pueda haber errores en el código. Esto reduce considerablemente los tiempos de programación.

Por lo tanto, la utilización de esta plantilla, al mejorar la comunicación hombre-máquina y facilitar la tarea de programación, va a facilitar a las empresas la incorporación de máquinas de funcionamiento automático o semiautomático a sus cadenas de producción.

Esto puede contribuir positivamente al desarrollo económico español, al mismo tiempo que los sistemas de producción se vuelven más eficientes, lo que puede tener un impacto positivo no solamente en la economía y la sociedad, pero también en el medio ambiente.

Por ello, queda demostrado en este proyecto que Pack ML contribuye a hacer más flexible la automatización de la minifábrica ICAI, recomendándose su aplicación en la automatización de equipos.

### Referencias

- [DEDI18] Dedios-Pleite, P., director de la Factoría Digital de Siemens y presidente de Siemens Industry Software, en una entrevista sobre la Industria 4.0 elaborada por KPMG Tendencias. Julio de 2018.  
<https://www.tendencias.kpmg.es/2018/07/entrevista-pascual-dedios-siemens-industria-4-0/>  
Artículo periodístico.  
Último acceso: 18/07/2021.
- [ECO21] Areba, E., periódico elEconomista, artículo: “¿Arrancamos el sector del automóvil en España?” Marzo de 2021.  
<https://www.economista.es/ecomotor/motor/noticias/11094777/03/21/Arrancamos-el-sector-del-automovil-en-Espana.html>  
Artículo periodístico.  
Último acceso: 18/07/2021.
- [MOND20] a. Rodríguez Mondéjar, J.A., “¿En qué lenguaje se programan los PLCs?”, Escuela Técnica Superior de Ingeniería ICAI, Apuntes de Automatización Industrial, Tecnologías-PLC. Enero 2020. Acceso restringido.  
b. Rodríguez Mondéjar, J.A., “Figura 1 Minifábrica ICAI”, COMILLAS-ICAI Práctica de laboratorio de Automatización Avanzada, Madrid. 2020. Acceso restringido.

# FLEXIBLE AUTOMATION OF THE ICAI MINI-FACTORY

Author: **Marty Plaza, Rodrigo.**

Director: Rodríguez Mondéjar, José Antonio.

collaborating entity: ICAI-Universidad Pontificia Comillas.

## PROJECT SUMMARY

### Introduction

#### *Problem statement*

In the middle of a fourth industrial revolution, in which automation has an important role, it is crucial that Spanish companies join this revolution and begin to adopt changes in their production systems.

Large companies, such as the automotive sector (Spain produces 3 million cars per year [ECO21]) are already solving this problem, but SMEs (a large part of the Spanish business fabric) are finding it harder [DEDI18].

Therefore, all the techniques that make it easier for those companies to automate their processes and make their production systems more flexible can be very useful.

One of them is studied in this project: Pack ML.

#### *State of the art*

Pack ML is an automation standard developed by OMAC (Organization for Machine Automation and Control), which improves communications between the operator and the machine, proposing a common structure for the code in which the machine is programmed, regardless of its origin or its manufacturer [OMAC21].

By understanding this template, the operator will be able to better understand the operation of any machine, allowing factories to introduce new machinery in a faster way.

The common structure proposed by Pack ML is based on organizing the programming code in blocks. Some blocks for the unit modes, such as automatic, manual, maintenance, cleaning, etc.

Then, in each unit mode, there may be up to 17 blocks that define the machine's state. For example, one of these states indicates normal operation, other indicates that the machine has stopped due to a dangerous situation, another that the machine has stopped because production have ended, etc.

Pack ML has variables that indicate in which state the machine is, so that with the use of a simple human-machine interface, the operator can know in exact time this information, as well as others like how long the machine has been in a state, speed...

As well as improving communications, the perfect organization of the code offered by Pack ML makes the programmer's task much easier, which results into faster programming, fewer errors, and easier detection of these errors.

### *Objective of the project*

The aim of this project is to demonstrate the advantages offered by the Pack ML standard in the automation of an assembly line.

To do this, a code with Pack ML has been developed to program the automatic operation of the ICAI Mini-factory assembly line, located in the digital systems laboratory of the ICAI School of Engineering.

Subsequently, an HMI was also developed to verify that the code is correct and demonstrate the advantages of Pack ML.

### Methodology

#### *Tools*

For the elaboration of the code and its simulation, the application for PLC programming TIA Portal was used.

Thanks to a tool from the Siemens group called PLCSim, with which the code can be loaded and simulated in a virtual PLC, the entire project has been carried out from home, using a laptop as programming station.

All code has been programmed in KOP (ladder diagram) and SFC (Sequential Function Chart).

SFC is a graphical programming language to describe the operation of sequential systems [MOND20]. It is based on the GRAFCET methodology and language. Plenty of research has been done about this methodology to make advantage of a very clear and visual way of writing code.

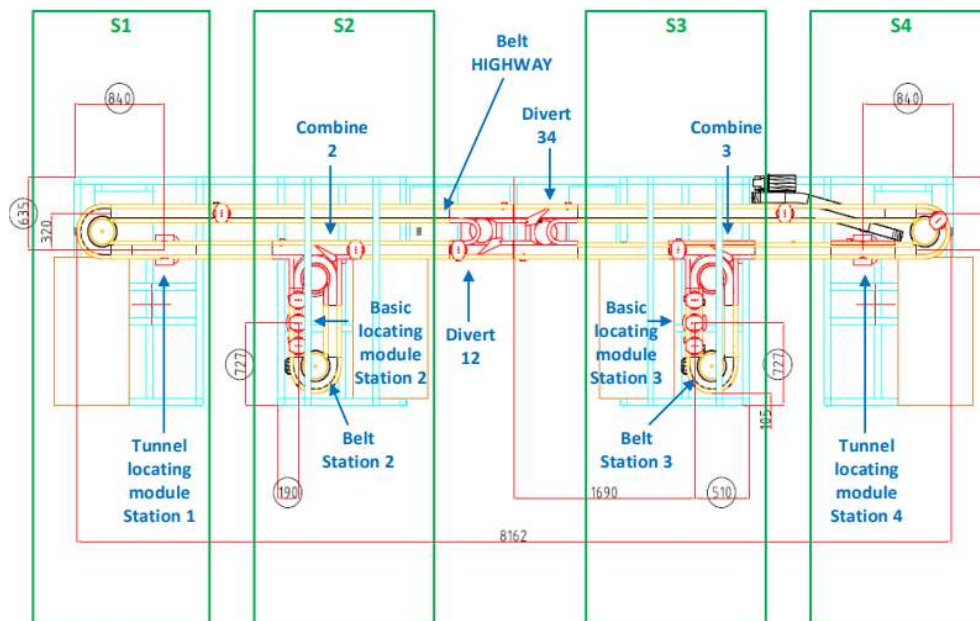
*Mini-Factory ICAI description*

Figure 1: mini factory plan [MOND20]

The elements of the mini factory that are automated in this project are:

**Conveyor belts.** The factory has a main conveyor and two satellite conveyors. Its operation is simple, and, in this project, they are controlled with a binary variable that activates or deactivates the electric engine that allows its movement.

**Basic locating modules.** The function of the locating modules is to retain the pallet that transports the product to be manufactured in the factory while the robot performs an action on that product. Specifically, the basic locating module retains the pallet by means of immobilizing devices.

**Tunnel locating modules.** They have the same function as the basic locating modules, but the way to retain the pallet is by lifting it off the belt.

**Interspatial deviation/merge modules.** They are needle-shaped devices that divide the main conveyor in two, allowing, if necessary, to transform this conveyor into two independent modules.

**Combine diversion/combination modules.** Devices (also in the shape of a needle) that allows the pallets to move from the main conveyor to satellites or vice versa.

Each of these devices is controlled by a block of code programmed in SFC, also called GRAFCET.

Instead of working independently, these devices communicate with each other. For the programming of this communication, research has been done on the GEMMA guide (automation template like Pack ML, which establishes instructions for communication between GRAFCETs).

The operation of these elements is different depending on which state of Pack ML is working the factory.

In this project, 3 different operations have been programmed: normal (automatic production), end-of-cycle stop and emergency stop.

## Results

The result of this project is a code, ready to be introduced into the mini factory controllers, having verified its operation by simulation and forcing input variables.

Trough simulation it has been possible to verify that the elements have a correct operation, in accordance with the specifications set during their design (specifications in which the safety of people and equipment has played a very important role).

It is also verified that the communication between devices is correct and that they are synchronized to allow the movement of pallets throughout the factory perfectly.

The change of states of the Pack ML template (normal production to emergency stop, end-of-cycle stop to emergency stop, etc.) is immediate and at the exact moment in which it must occur.

After an emergency stop, the system remains in a safe stop state, with the devices turned off, and does not work again even if the emergency mushroom is not pressed, complying with the ISO 13850 standard. The system can only be restarted when the emergency mushroom has been reset.

During the end-of-cycle stop, the finished products are collected and leaves the system ready for new commissioning.

And the automatic production state works well and is flexible enough to adapt to the manufacture of multiple products, provided that this manufacturing occurs in four stages.

## Conclusions

This project has proven that man-machine communication is greatly improved with Pack ML. With the help of a simple HMI like the following:



*Figure 2: Possible HMI to control the system*

The operator can know at any time what state the machine is in and actuate the necessary commands to go from one state to another. With a minimal knowledge of this template, you could control any machine on the market that work with Pack ML (as it is a standard to automate, it is expected that more and more machines will use this template).

On the other hand, it has been proven that the advantages offered by this template when programming automation are plenty.

In the first place, it allows to program in a very simple way the change from one state to another, and the programmer does not have to worry about enabling or disabling any GRAFCET.

Second, the perfect organization of this template allows the code to be divided into blocks, stimulating the division of work with other programmers and the possibility of testing each block separately, greatly reducing the chance of finding errors in the code. This reduces considerably programming time.

Therefore, the use of Pack ML, by improving human-machine communication and facilitating the programming task, will make it easier for companies to incorporate automatic or semi-automatic machines into their production lines.

This can positively contribute to Spanish economic development, while production systems become more efficient, which can have a positive impact not only to the economy and society, but also on the environment.

For this reason, it is proved in this project that Pack ML contributes to making ICAI mini-factory automation more flexible, recommending its application in the automation of equipment.

## References

- [DEDI18] Dedios-Pleite, P., director of the digital factory of Siemens and president of Siemens Industry Software, for an interview about Industry 4.0 conducted by KPMG Tendencias. July 2018.  
<https://www.tendencias.kpmg.es/2018/07/entrevista-pascual-dedios-siemens-industria-4-0/>  
Last access: 18/07/2021.
- [ECO21] Areba, E., newspaper elEconomista, name of the article: “¿Arrancamos el sector del automóvil en España?”. March 2021.  
<https://www.eleconomista.es/ecomotor/motor/noticias/11094777/03/21/Arrancamos-el-sector-del-automovil-en-Espana.html>  
Last access: 18/07/2021.
- [MOND20] a. Rodríguez Mondéjar, J.A., “¿En qué lenguaje se programan los PLCs?”, Technical School of Engineering ICAI, Industrial Automation Notes, PLC-technology. January 2020. Restricted access
- b. Rodríguez Mondéjar, J.A., “Figura 1 Minifábrica ICAI”, COMILLAS-ICAI Advanced Automation Laboratory Practices, Madrid. 2020. Restricted access

## Índice

1. Introducción y planteamiento del proyecto .....	19
2. Descripción de las tecnologías .....	21
2.1. Herramientas.....	21
2.1.1. Hardware.....	21
2.1.2. Software .....	23
2.1.3. Lenguajes de programación .....	27
2.2. Técnicas de automatización utilizadas.....	29
2.2.1. GRAFCET.....	29
2.2.2. Guía GEMMA.....	36
2.2.3. PackML .....	36
2.3 Descripción de la minifábrica ICAI.....	44
2.3.1. Palé.....	45
2.3.2. Cintas.....	46
2.3.3. Módulos de desviación/combinación interespacial.....	47
2.3.4. Módulos combinados de desviación/combinación.....	49
2.3.5. Posicionadores de tipo básico.....	49
2.3.6. Posicionadores de tipo túnel.....	51
3. Descripción del modelo desarrollado .....	55
3.1. Programación del estado Execute (Fabricación de productos).....	55
3.1.1. Arranque de las cintas.....	56
3.1.2. Posicionador básico S2.....	57
3.1.3. Posicionador básico S3.....	62
3.1.4. Posicionador túnel S4.....	63
3.1.5. Posicionador túnel S1:.....	64
3.1.6. Aguja Combine C2: .....	65
3.1.7. Aguja Combine C3: .....	69
3.1.8. Conclusiones.....	73
3.2. Simulación del estado Execute (HMI) .....	74
3.2.1. Pulsadores .....	75
3.2.2. Pilotos.....	76
3.2.3. Contador de Palés .....	78
3.2.4. Barreras .....	79

3.2.5. Movimiento de las agujas .....	80
3.2.6. Simulación del movimiento del palé .....	81
3.2.7. Conclusiones.....	87
3.3. Parada de Fin de ciclo.....	89
3.3.1. Introducción .....	89
3.3.2. Control de los módulos de dispersión/combinación .....	89
3.3.3. Programación del posicionador básico S2 .....	94
3.3.4. Fin del estado Completing.....	94
3.3.5. Estado Complete .....	95
3.3.6. Estado Resetting.....	96
3.4. Parada de Emergencia.....	98
3.4.1. Introducción .....	98
3.4.2. Estado Aborting.....	99
3.4.3. Estado Aborted.....	100
3.4.4. Estado Clearing.....	101
3.4.5. Estado Stopped .....	102
4. Alineamiento con los Objetivos de Desarrollo Sostenible (ODS).....	102
5. Conclusiones.....	104
Anexo 1: Comunicación del PLC con el resto de los dispositivos de la minifábrica ICAI.....	107
Referencias.....	117



## 1.Introducción y planteamiento del proyecto

Una empresa se define como una organización de personas cuya función es la de generar un determinado bien o servicio, y cuya finalidad es la de obtener a partir de esta actividad una rentabilidad económica [THOM06].

Con el fin de obtener este beneficio, el conjunto de etapas en las que se genera el producto o servicio, llamado proceso de producción debe ser lo más eficiente posible.

Una manera de hacer más eficiente este proceso de producción es automatizarlo. Automatizar un proceso es lograr que funcione con poca o ninguna intervención humana, de una manera óptima y segura [MOND20]. Con ello se busca reducir los costes y los tiempos de producción, disminuir las situaciones de riesgo o mejorar la calidad de los productos [FORT13].

Todas estas ventajas junto con el rápido desarrollo de la tecnología están haciendo que los sistemas automatizados se hagan cada vez más complejos, haciendo difícil su programación.

Ante esta creciente dificultad, la organización para la automatización y control de máquinas (OMAC por sus siglas en inglés) desarrolla una plantilla para automatizar denominada Pack ML.

El objetivo principal de Pack ML es facilitar el intercambio de información entre los operarios y las máquinas [OMAC21], proponiendo una estructura común en el funcionamiento de toda máquina que esté programada con esta plantilla.

Esta estructura común se basa en organizar el funcionamiento del automatismo en diferentes modos (automático, manual, mantenimiento, etc) y que cada modo está formado por distintos estados. Estos estados los define la plantilla y no el usuario con el objetivo de que todas las máquinas que utilicen Pack ML tengan la misma estructura. Además, la plantilla define para qué se utiliza cada estado, así como la manera de pasar de un estado a otro.

Pack ML tiene una serie de variables que informan de en qué estado se encuentra la máquina en cada momento. De esa manera el operador, a través de una HMI (interfaz hombre máquina) puede detectar que la máquina se encuentra en modo Suspended (suspendida), y debido a que el significado de cada estado es conocido, puede saber que la causa por la que se encuentra en ese estado es la avería de otra máquina en la misma línea de producción. Y al tratarse de una forma estándar de automatizar, esta comunicación entre el hombre y la máquina puede ser igual de cómoda incluso si la máquina procede de un fabricante distinto al que el operador está habituado.

Esta ventaja en la comunicación permitirá a las empresas de montaje integrar más fácilmente máquinas a sus procesos, haciéndolos más flexibles.

También permite detectar más fácilmente fallos y reutilizar código, lo que se puede traducir en un aumento de los beneficios para el cliente o dueño de la fábrica. Cuanto menos tiempo se tarde en detectar el fallo que ha ocasionado la avería de una máquina, también será menor el tiempo que la fábrica tenga que estar parada y sin producir.

Por si fuera poco, como se ha podido comprobar en este proyecto, la perfecta organización de Pack ML facilita enormemente la tarea del programador, quien puede beneficiarse de todas las

ventajas que te da una plantilla de programación: rapidez para desarrollar el código, código perfectamente estructurado, organización modular, menor cantidad de errores, etc.

Para comprobar las ventajas que Pack ML ofrece a las cadenas de montaje y a los procesos industriales en general, este proyecto consiste en la realización de un código para la automatización de la Minifábrica ICAI, cadena de montaje destinada fundamentalmente a la docencia situada en el laboratorio de sistemas digitales de la Escuela Técnica Superior de Ingeniería ICAI.

En este código se programa el comportamiento que van a tener los elementos de la minifábrica en diferentes situaciones de la producción: procesado de productos, parada de fin de ciclo y parada de emergencia. Dado que es un código para programar plc, en estos elementos no se incluyen los robots IRB 120, quienes no se controlan con autómatas programables, si no con un controlador IRC5.

Por otra parte, Pack ML no es la única técnica de automatización empleada en la elaboración del código. En este proyecto se ha podido investigar acerca de la metodología GRAFCET y sobre la guía GEMMA.

GRAFCET recoge una serie de normas y un lenguaje para representar sistemas secuenciales mediante diagramas de etapas y transiciones [MOND18]. Consiste en una manera sencilla y visual de programar código y en este proyecto se ha querido aprovechar esta ventaja diseñando el funcionamiento de cada dispositivo de la fábrica con este lenguaje.

La guía GEMMA es una plantilla que recoge los posibles modos de funcionamiento que puede tener un sistema automatizado y la forma de pasar de uno a otro [MOND18]. Como para este propósito ya se utiliza Pack ML, no se emplea de forma directa en este proyecto, aunque la investigación acerca de esta plantilla ha sido de vital importancia a la hora de establecer las comunicaciones entre los dispositivos de la fábrica.

Finalmente, además del código, se ha diseñado una interfaz hombre máquina para comprobar su funcionamiento y mostrar los resultados obtenidos.

## 2. Descripción de las tecnologías

### 2.1. Herramientas

En esta sección se describen las herramientas utilizadas para la realización de este proyecto. No se incluyen los elementos de la fábrica a automatizar, ya que no son herramientas propiamente dichas, si no el objeto de la automatización. Si bien, se hará una descripción del conjunto de la fábrica en la sección 2.3 de esta memoria.

Las herramientas que se han utilizado se pueden dividir en dos grupos:

- 1) Hardware.
- 2) Software.

#### 2.1.1. Hardware

##### 2.1.1.1. Estación de programación.

Como estación de programación se utiliza un ordenador portátil. En él se va a descargar el software necesario para programar el sistema automatizado. La empresa desarrolladora del Software (Siemens) recomienda que la estación de programación tenga las siguientes características:

Propiedad	Recomendación
Procesador	Intel Core i5-6440EQ, 3.4 GHz
RAM	16 GB
HDD	SSD con al menos 50 GB de memoria libre en el disco duro
Red	1 Gbit
Resolución de pantalla	1920x1080 px
Sistemas operativos soportados	Windows 7 (64 Bits)
	Windows 10 (64 Bits)
	Windows Server (64 Bits)

*Tabla 1: Requisitos para la estación de programación*

*Necesarios para trabajar en TIA Portal [SIEM21] [1]*

Si bien es cierto que el programa TIA Portal admite la versión 7 de Windows, el software donde se va a diseñar la interfaz hombre máquina no admite versiones inferiores a la 10 [SIEM20].

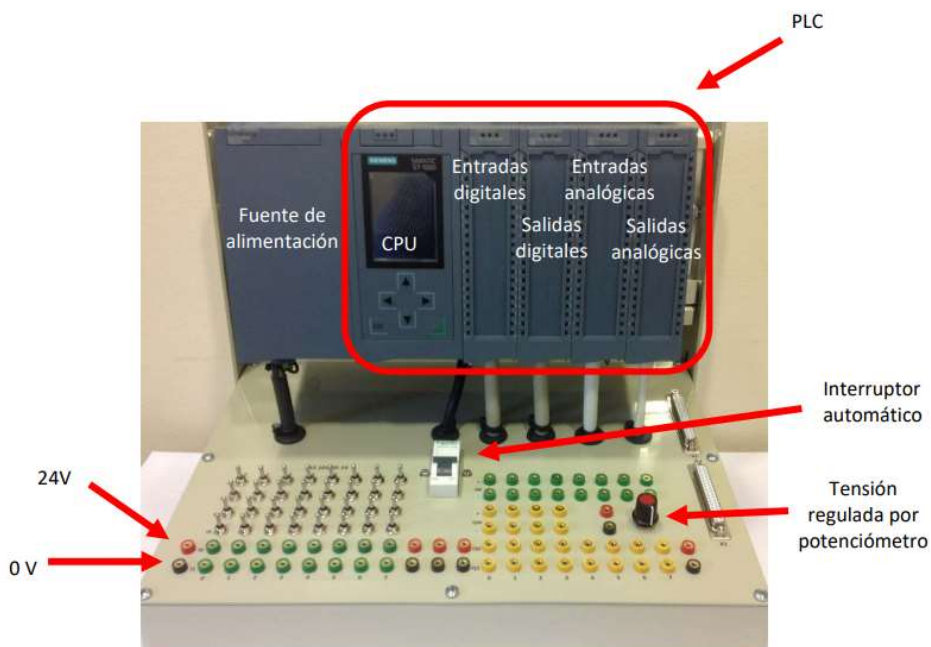
El ordenador portátil es un HP Pavilion con las siguientes características:

HP Pavilion x360 Convertible 14-dh1xxx	
Procesador	intel®Core™i7-10510U, 1.8-2,3 GHz
RAM instalada	8 GB
Disco duro	476 GB
Resolución de pantalla	1920x1080
Sistema operativo	Windows 10 Home (64 bits)

*Tabla 2: Características del ordenador portátil donde se ha realizado este proyecto*

### 2.1.1.2. PLC

Los PLC que se han utilizado son los del laboratorio de sistemas digitales de ICAI. Son PLC S7-1500 de la marca Siemens, con la siguiente hoja de características:



*Figura 1: PLC del laboratorio [MOND20]*

PLC S7-1500 SIEMENS	
CPU	1516-3 PN/DP 6ES7 516- 3AN01-0AB0 V.2.8
Módulo de entradas digitales	DI 32x24VDC HF ->6ES7 521-1BL00-0AB0 V.2.1
Módulo de salidas digitales	DQ 32x24VDC/0.5A HF -> 6ES7 522-1BL01-0AB0 V.1.0
Módulo de entradas analógicas	AI 8xU/I/RTD/TC ST -> 6ES7 531-7KF00-0AB0 V.2.0
Módulo de salidas analógicas	AQ 4xU/I ST -> 6ES7 532-5HD00-0AB0 V.2.0
Fuente de alimentación	24 V/8A DC

	Limitación de intensidad: 9A Consumo: 1,7A (230V)
Memoria de trabajo para programas	1MB
Memoria de trabajo para datos	5MB
Tiempo de ejecución para operaciones de bits	10ns
Funciones especiales	Regulador PID

Tabla 3: Características del PLC S7-1500 [MOND20],[SIEM16]

### 2.1.1.3. Panel físico.

Como interfaz hombre máquina se ha utilizado el panel táctil TP700 Comfort de la marca Siemens. Algunas de sus características son:

Panel Táctil TP700 Comfort SIEMENS	
Pantalla	7'', táctil, 16 millones de colores, resolución 800x480
Sistema Operativo	Windows CE
Comunicaciones	PROFINET (switch de 2 puertos) y PROFIBUS
Número de referencia	6AV2124-0GC01-0AX0

Tabla 4: Características del panel físico. [MOND20]

### 2.1.2. Software

Como software se utilizó la aplicación TIA Portal (Totally Integrated Automation Portal). TIA Portal es una aplicación para la estación de programación en la que se efectuó la escritura del código, así como su edición, depuración y simulación. Es una interfaz de programación de PLCs de Siemens.

Ventajas:

TIA Portal es una interfaz con muchos programas incorporados. En este proyecto se usarán el software para la programación de PLCs SIMATIC STEP 7, el programa para diseñar la interfaz hombre-máquina SIMATIC WinCC Professional y el simulador de PLCs PLCSIM. A continuación, se describirán con más detalle:

#### 2.1.2.1. SIMATIC STEP7 (S7) professional:

SIMATIC S7 es un software de programación de PLCs de la marca Siemens. La primera versión es del año 1995, y para este proyecto se utiliza la versión 16<sup>i</sup>. Permite programar el autómatas programable mediante lenguajes distintos.

Existen dos tipos de S7: el professional y el basic. El basic es de menor coste. Dado que el proceso que se quiere automatizar no es demasiado complejo, valdría para este proyecto, de no ser porque solo permite programar en PLCs de la serie 1200 [SIEM21]. Como los PLCs del laboratorio son S7-1500, se debe escoger el professional, pese a que el coste es mayor.

### 2.1.2.2. SIMATIC WinCC professional:

Esta aplicación sirve para programar un SCADA, acrónimo para Supervisory Control And Data Acquisition. Un SCADA es una plataforma donde el operador puede comprobar en tiempo real el funcionamiento del sistema automático y actuar sobre él. Este software se ha utilizado en este proyecto por dos motivos fundamentales:

- Comprobar el funcionamiento del programa diseñado.
- Mostrar el programa al cliente (en este caso, al profesorado de la universidad).

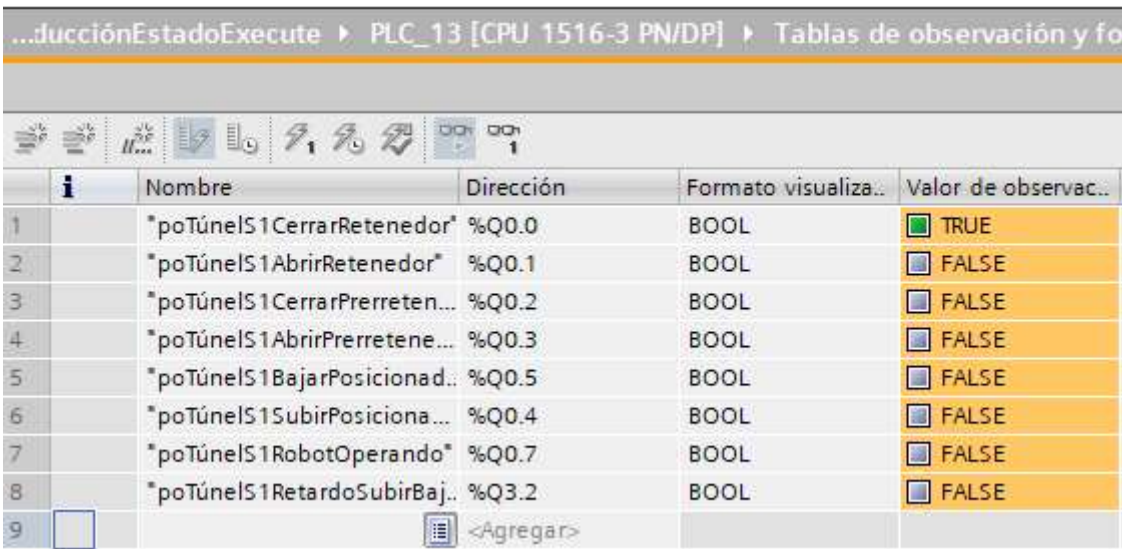
#### A)Comprobación del programa

Hay varias maneras de comprobar si el programa que se ha diseñado en el PLC funciona, antes de aplicarlo sobre la planta. Una manera de hacerlo es forzando valores a las entradas y comprobando los de las salidas. El programa TIA Portal permite hacerlo con las pestañas “tablas de observación” y “tablas de forzado permanente”:



	Nombre	Dirección	Formato visualiza..	Valor de forzado ...	F
1		%I3.4:P	BOOL	FALSE	<input checked="" type="checkbox"/>
2		%I1.0:P	BOOL	TRUE	<input checked="" type="checkbox"/>
3	*RESET*:P	%I0.0:P	BOOL	TRUE	<input checked="" type="checkbox"/>
4	*piTúnelS1PaléEntrada*:P	%I0.1:P	BOOL	TRUE	<input checked="" type="checkbox"/>
5	*piTúnelS1PaléPosicionador*:P	%I0.2:P	BOOL	TRUE	<input checked="" type="checkbox"/>
6	*piTúnelS1PosiciónArriba*:P	%I0.3:P	BOOL	FALSE	<input checked="" type="checkbox"/>
7	*piTúnelS1PosiciónAbajo*:P	%I0.4:P	BOOL	TRUE	<input checked="" type="checkbox"/>
8		%I3.5:P	BOOL	TRUE	<input checked="" type="checkbox"/>
9		%I5.5:P	BOOL		<input type="checkbox"/>
10		<Agregar>			<input type="checkbox"/>

Figura 2: Tabla de forzado permanente



	Nombre	Dirección	Formato visualiza..	Valor de observac..
1	*poTúnelS1CerrarRetenedor*	%Q0.0	BOOL	<input checked="" type="checkbox"/> TRUE
2	*poTúnelS1AbrirRetenedor*	%Q0.1	BOOL	<input type="checkbox"/> FALSE
3	*poTúnelS1CerrarPrerreten...	%Q0.2	BOOL	<input type="checkbox"/> FALSE
4	*poTúnelS1AbrirPrerretene...	%Q0.3	BOOL	<input type="checkbox"/> FALSE
5	*poTúnelS1BajarPosicionad..	%Q0.5	BOOL	<input type="checkbox"/> FALSE
6	*poTúnelS1SubirPosiciona...	%Q0.4	BOOL	<input type="checkbox"/> FALSE
7	*poTúnelS1RobotOperando*	%Q0.7	BOOL	<input type="checkbox"/> FALSE
8	*poTúnelS1RetardoSubirBaj..	%Q3.2	BOOL	<input type="checkbox"/> FALSE
9		<Agregar>		

Figura 3: Tabla de observación

En las figuras 4 y 5 se muestra un ejemplo de cómo funciona el forzado permanente. En este caso se está comprobando el programa cargado a uno de los instrumentos de la minifábrica: el posicionador túnel. La variable lógica piTúnelS1PaléEntrada representa la salida de un sensor que detecta la presencia de un palé a la entrada del posicionador. La primera acción es correcta ya que, tal y como se ha programado, al detectarse el palé, el retenedor del posicionador se va a cerrar para evitar que el palé continúe su recorrido. Lo hace cuando la señal poTúnelS1CerrarRetenedor, que recibe del PLC, vale 1.

#### B) Mostrar al cliente el funcionamiento del programa

Este sistema es práctico solamente cuando se está probando sistemas sencillos. Cuando se quiere comprobar el funcionamiento de sistemas más complejos, puede ser incómodo. Además, es poco visual. Si el cliente quiere hacer una primera comprobación de cómo funciona el programa, puede quedar insatisfecho si se le muestran dos tablas de variables que en nada se parecen a una cadena de montaje.

Por ello, SIMATIC WinCC professional es una herramienta muy útil. Un inconveniente, sin embargo, es que las acciones que se pueden realizar en ella están muy limitadas. Tiene pulsadores, deslizadores con cursor para introducir variables de tipo numérico, o pilotos que se encienden o apagan. Pero a la hora de representar movimiento, las posibilidades que ofrece son bastante escasas en la opinión del autor de esta memoria. Para simular el movimiento de las agujas, las barreras o de la cinta, se ha tenido que invertir un tiempo considerable, además de la necesidad de declarar variables estáticas que ocupan mucha memoria.



Figura 4: Ejemplo de pulsadores y pilotos

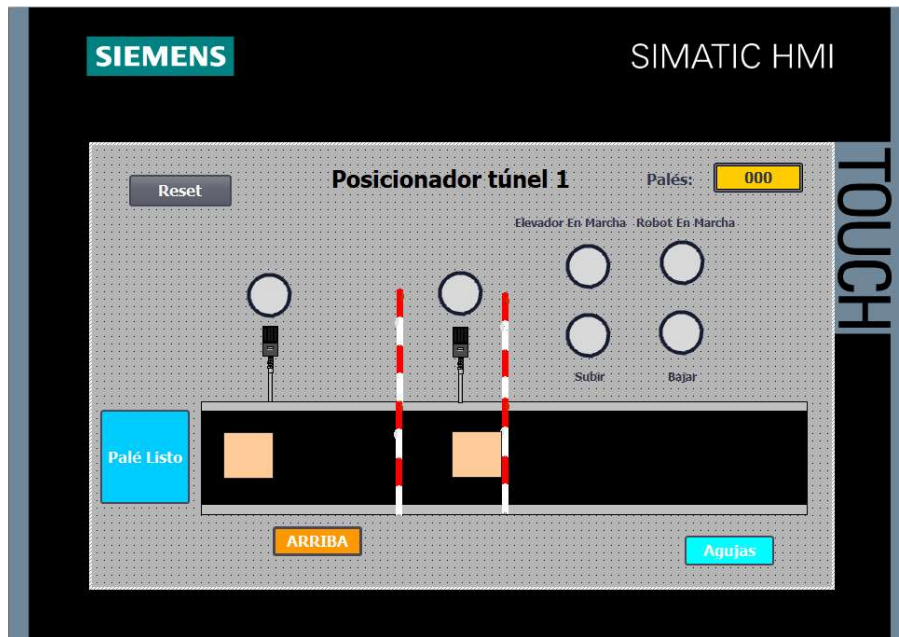


Figura 5: Ejemplo de HMI para el posicionador túnel.

### 2.1.2.3. SIMATIC S7 PLCSIM ADVANCED

Para comprobar el funcionamiento de un programa, la estación de programación debe estar conectada a un PLC. Sin embargo, no siempre se tiene la posibilidad de disponer de un PLC físico. Por ello, TIA Portal dispone de un simulador de PLC virtual al que poder cargar el programa.

Este software ha sido una importante ayuda para la consecución de este trabajo. Gracias a él, se ha podido comprobar el funcionamiento de la totalidad del diseño desde casa, sin necesidad de trasladarse al laboratorio.

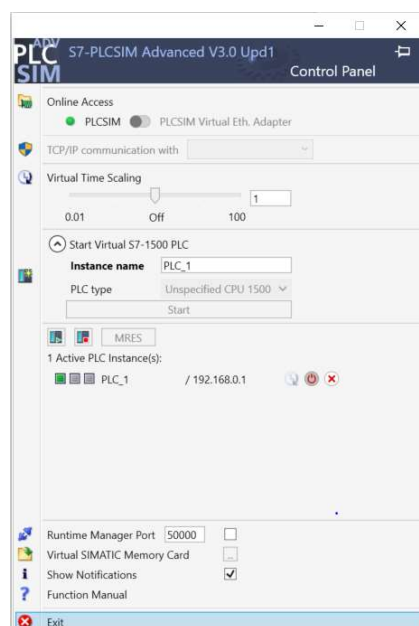


Figura 6: PLC simulado.

### 2.1.3. Lenguajes de programación [MOND20]

**-KOP<sup>ii</sup>:** Diagrama de contactos. Lenguaje de programación de tipo gráfico, utiliza circuitos eléctricos simulados para representar la lógica que debe seguir el PLC. En él, las entradas, se representan mediante interruptores, y las salidas mediante bobinas. Este lenguaje, junto con S7-Graph, es el que se va a utilizar mayoritariamente para la programación de la fábrica. Con el siguiente ejemplo, se explica en líneas generales su funcionamiento:

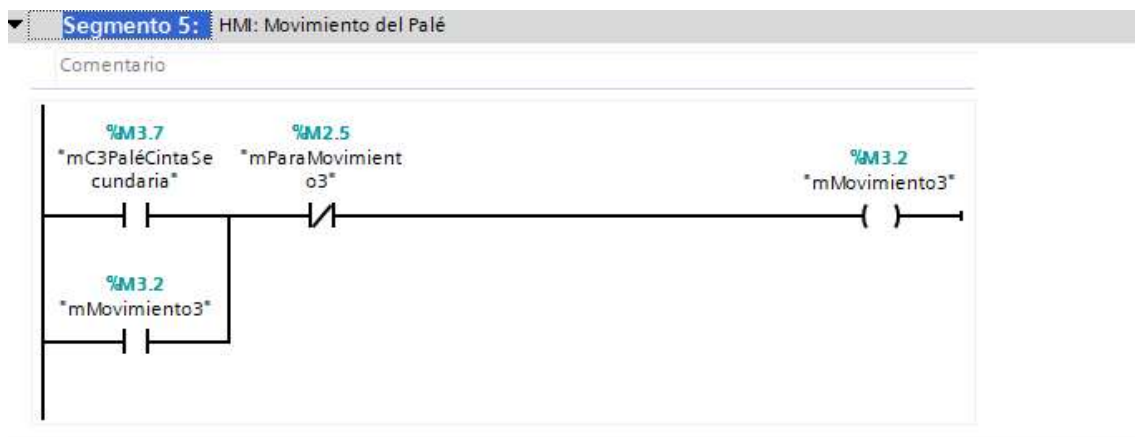


Figura 7: Ejemplo de Diagrama de contactos

La barra de la izquierda es la alimentación, y por lo tanto el 1 lógico. La barra de la derecha es la masa. Cuando hay tensión en la bobina, es decir, cuando llega un 1 por su izquierda, la variable asociada a la bobina (en este caso mMovimiento3) se pone a 1. Las variables de entrada se representan mediante interruptores. Cuando su variable asociada vale 1, el interruptor se cierra y lo que está inmediatamente a su derecha pasa a valer 1. Si su variable asociada vale 0, permanecerá abierto. El interruptor asociado a la variable “mParaMovimiento3” es de tipo normalmente cerrado y funcionará al revés que los otros: cuando su variable asociada vale 0 se cierra y cuando su variable asociada vale 1, se abre. Con este diagrama se implementa en KOP la función lógica:

$$mMovimiento3 = (mC3PaléCintaSecundaria + mMovimiento3) \cdot mParaMovimiento3'$$

Este código en particular se usa en el programa para simular el movimiento del palé en la interfaz hombre-máquina.

**S7-Graph<sup>iii</sup>:** Diagrama de bloques de funciones secuenciales (GRAFCET). Este es otro lenguaje muy utilizado en este proyecto. También es de tipo gráfico y se basa en la metodología GRAFCET para diseñar sistemas secuenciales ya de cierta complejidad.

**SCL<sup>iv</sup> :** Texto estructurado. Emula un lenguaje de alto nivel [MOND20]. No se suele programar todo el PLC con lenguaje SCL, si no que normalmente se utiliza para programar funciones específicas [HEGA20], por ejemplo, para hacer un conversor de milisegundos a segundos. No se utiliza en este proyecto.

**FUP<sup>v</sup>**: Diagrama de bloques de funciones. FUP representa las funciones lógicas del algoritmo de control mediante un conjunto de puertas lógicas, tal y como hacen los sistemas digitales. No se utiliza en este proyecto.

**AWL<sup>vi</sup>**: Es un tipo de lenguaje de programación de texto que emula a ensamblador. Tampoco se utiliza en este proyecto.

## 2.2. Técnicas de automatización utilizadas

Automatizar un sistema, puede ser un proceso muy complejo. Afortunadamente, existen 3 técnicas que permiten facilitar este trabajo. Dichas técnicas para automatizar son:

- 1) Lenguaje y metodología GRAFCET.
- 2) Guía GEMMA.
- 3) Pack ML.

### 2.2.1. GRAFCET

La manera más intuitiva de representar el funcionamiento de un sistema automático es mediante un diagrama de etapas y transiciones. Las etapas son cada uno de los estados en los que se puede encontrar el sistema, y las transiciones definen la condición para pasar de una etapa a otra. Resulta tan intuitivo, que en 1977 se decidió en Francia elaborar una serie de normas, así como un lenguaje, para representar los sistemas automáticos de esta manera. Este lenguaje es GRAFCET (Grphe Fonctionnel de Commande des Étapes et Transitions<sup>vii</sup>). Posteriormente, en 1988, la IEC lo estandariza como lenguaje de especificación [MOND18].

El lenguaje GRAFCET es, por lo tanto, una de las técnicas utilizadas en este proyecto y ha sido una herramienta esencial para hacer la primera parte de la automatización, que es el diseño del funcionamiento de cada uno de los elementos, por separado.

Dichos elementos son: las cintas transportadoras, cuatro posicionadores, que sujetan el palé y lo colocan para que el robot actúe sobre él, dos módulos de combinación/dispersión, también conocidos como agujas Combine, para mover los palés de la cinta principal a la secundaria y de la secundaria a la principal, y un módulo de dispersión interespacial o agujas Divert que divide en dos la cinta principal si así fuese necesario.

La automatización de los robots no se ha realizado. La razón principal es que el código diseñado sirve para programar plc. Sin embargo, los robots no se controlan con este dispositivo. Se ha decidido, por lo tanto, simular su funcionamiento mediante temporizadores.

El diseño del resto de elementos se llevó a cabo en varias etapas:

- 1) Estudio del funcionamiento
- 2) Comunicación de los elementos con el PLC.
- 3) Medidas de seguridad para personas e instrumentos.
- 4) Elaboración de un GRAFCET tecnológico.

#### 2.2.1.1. Estudio del funcionamiento

Esta etapa consiste en investigar cómo deberían de funcionar los elementos cuya automatización se quiere mejorar. Permite adquirir una primera noción de cómo realizar esta automatización. Por ejemplo, en el caso de las agujas Combine, se comprobó qué posiciones podían tener. Estas posiciones son dos, la de mover el palé a la cinta principal y la de moverlo a la secundaria. El movimiento se realiza gracias a que la agujas llevan incorporadas dos válvulas neumáticas de doble efecto, pudiéndose controlar cada una de ellas con una variable binaria.

En el código, se pondrá a 1 una de las variables cuando se quiera mover la aguja a una posición que permita el paso del palé de la cinta principal a la secundaria, y la otra , para una posición que lo permita mover de una cinta secundaria a la principal.

También se estudiaron los sensores con los que cuenta la minifábrica. Durante este proyecto se ha comprobado que los sensores son elementos muy importantes ya que facilitan mucho la tarea de automatización.

La mayoría de los sensores son inductivos y su salida es digital. Los dispositivos elevadores de los posicionadores de tipo túnel tienen también finales de carrera.

### 2.2.1.2. Comunicación de los elementos con el PLC.

La comunicación entre el PLC y los elementos de la minifábrica se realiza exclusivamente mediante variables binarias. Las variables de salida del plc son las entradas de los actuadores (válvulas). Las salidas de los sensores son las variables de entrada del PLC.

A continuación, se describirá con más detalle cómo es esta comunicación, mediante el esquema de control de uno de los dispositivos, el posicionador túnel S1. Para este esquema de control se adopta la convención de mínima protección para los circuitos:

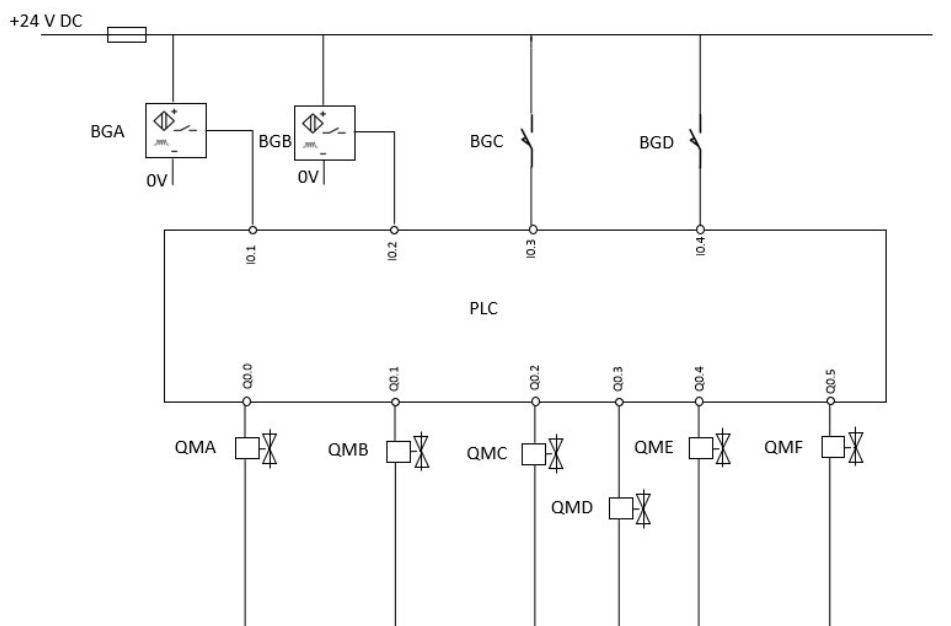


Figura 1: Esquema de control del posicionador túnel S1

A continuación, se añade una tabla con el significado de cada variable:

<b>ENTRADAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Entrada PLC</b>	<b>Significado</b>	<b>Entrada física PLC</b>
Sensor inductivo BGA	piTúnelS1PaléEntrada	1: Palé detectado en la entrada	%I0.1
Sensor inductivo BGB	piTúnelS1PaléPosicionador	1: Palé detectado dentro del posicionador	%I0.2
Final de Carrera BGC	piTúnelS1PosiciónArriba	1: Elevador arriba	%I0.3
Final de Carrera BGD	piTúnelS1PosiciónAbajo	1: Elevador abajo	%I0.4
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QMA	poTúnelS1CerrarRetenedor	1: Cerrar 0: No cerrar	%Q0.0
Válvula QMB	poTúnelS1AbrirRetenedor	1: Abrir 0: No abrir	%Q0.1
Válvula QMC	poTúnelS1CerrarPrerretenedor	1: Cerrar 0: No cerrar	%Q0.2
Válvula QMD	poTúnelS1AbrirPrerretenedor	1: Abrir 0: No abrir	%Q0.3
Válvula QME	poTúnelS1SubirPosicionador	1: Subir 0: No subir	%Q0.4
Válvula QMF	poTúnelS1BajarPosicionador	1: Bajar 0: No bajar	%Q0.5

*Tabla 1: Entradas y salidas del plc para el posicionador túnel S1*

Aclaraciones: En el nombre del dispositivo se pone un nombre genérico y el código de dos letras de la IEC 81346-2. Como hay más de un elemento del mismo tipo, se añade una tercera letra para diferenciarlos.

En cuanto al nombre de la variable del PLC, la primera letra “p” indica que es una salida o entrada física. Las segundas letras “i” y “o” indican entrada y salida. Después se añade la palabra “Túnel” para que quede identificado el tipo de dispositivo y “S1” porque existen dos posicionadores de tipo túnel en la fábrica. Las últimas palabras hacen referencia a la función de la variable.

En el significado, en aquellos casos donde un valor es el contrario a otro, solo se ha añadido un caso y el valor de la variable asociada a ese caso.

Para más información sobre la comunicación del plc con cada dispositivo de la minifábrica, consulte el Anexo 1 de esta memoria.

### 2.2.1.3. Medidas de seguridad para personas e instrumentos

Lo más importante para automatizar es garantizar la seguridad, tanto de personas como de equipos. Los daños causados a personas son los más importantes para tener en cuenta, pero también la rotura de un equipo puede producir una parada en la producción, con los perjuicios económicos que eso supone.

Las medidas de seguridad se clasifican en dos tipos:

- 1) Medidas pasivas.
- 2) Medidas activas.

Las medidas pasivas, que, ante un accidente, se encargan de minimizar los daños, son las protecciones eléctricas (protecciones diferenciales, interruptores de sobrecorriente, etc.). También son medidas pasivas las barreras metálicas que hay alrededor de la fábrica.

Pero la razón por la que la seguridad se menciona en este capítulo es la seguridad activa, que es la que forma parte de la automatización del proceso [MOND20].

En cuanto a la seguridad activa para prevenir accidentes hacia personas, se han implementado en este diseño dos medidas: la señalización y la seta de emergencia.

La señalización está presente en la HMI. Cada vez que un equipo se ha considerado peligroso, su funcionamiento se ha señalado con un piloto rojo en señal de peligro. Es el caso del posicionador túnel. Cuando el posicionador túnel está ascendiendo o descendiendo puede resultar peligroso, y se deben alejar manos y brazos de él. Por ello, una señal roja de advertencia o peligro puede contribuir a prevenir un accidente. Como medida de seguridad adicional, se añadirán en su programación unos tiempos de espera tanto para la subida como para la bajada. Esto es para dar tiempo al operador a retirarse antes de que el equipo se ponga en funcionamiento.

La seta de emergencia es una medida de seguridad activa que se debe de implementar siempre que se hace una automatización. Es un pulsador que, al accionarse, lleva el sistema a un estado de parada segura.

Después de accionarse la seta de emergencia, el sistema no podrá volver a su funcionamiento normal hasta que no se produzca un rearme (norma ISO 13850), señal que necesita el sistema para saber que se ha eliminado o aislado el riesgo, y el proceso puede volver a funcionar de manera segura.

Habiendo introducido medidas para prevenir el riesgo hacia personas, el diseño pasó a centrarse en cómo prevenir el daño a equipos y también al producto que la fábrica está procesando.

Una posible causa de daño al producto podría ser que chocaran dos palés. Para prevenir este choque, en las agujas, se ha realizado el siguiente protocolo:

Siempre que se vaya a efectuar un desplazamiento de un palé de una cinta a otra, la cinta de donde no viene el palé permanecerá cerrada hasta que se haya producido ese desplazamiento.

Otra posible causa de fallo, también en las agujas, se produce si un palé se acerca a la aguja cuando está efectuando un cambio de posición. Como medida de seguridad, por lo tanto, se diseñó el automatismo para cerrar tanto la cinta principal como la secundaria cuando la aguja va a cambiar de posición.

#### 2.2.1.4. Elaboración de un GRAFCET tecnológico

Hay que resaltar que GRAFCET es un lenguaje que se utiliza para describir el funcionamiento de un automatismo, pero no está pensado para ser utilizado directamente como lenguaje de programación [MOND18]. Por ello, la forma correcta de trabajar es diseñar el GRAFCET “a lápiz y papel”, para posteriormente buscar un lenguaje de programación para añadir ese diseño en el PLC.

A este primer boceto se le denomina GRAFCET descriptivo. En él, las transiciones pueden ser expresiones lógicas del tipo “Cerrada cinta principal”.

Al GRAFCET listo para ser programado en el PLC se le denomina GRAFCET tecnológico.

Afortunadamente, existe un lenguaje de programación, estandarizado en la IEC, que es muy similar al que utiliza la metodología GRAFCET: SFC<sup>viii</sup>. Por ello, pasar del GRAFCET descriptivo al tecnológico resulta muy mecánico.

En particular, cuando se trata de un GRAFCET relativamente sencillo, se puede hacer en tecnológico directamente. Es el caso de este proyecto, donde cada uno de los GRAFCETS se ha realizado directamente sobre TIA Portal.

A continuación, se explicará, en líneas generales, las partes más importantes que tiene un GRAFCET. Al igual que como ya se ha hecho en el capítulo de herramientas con los diagramas de contactos, esta breve explicación ayudará a entender mejor los códigos cuando se expliquen.

En un GRAFCET se pueden diferenciar cuatro partes básicas:

- 1)Etapas.
- 2)Transiciones.
- 3)Uniones
- 4)Acciones.

##### 1) GRAFCET tecnológico: Etapas

Representa una situación en la secuencia que debe seguir un sistema [MOND18]. En ella se realizan una o varias acciones.

##### 2) GRAFCET tecnológico: Transiciones

Representan la condición que se tiene que cumplir para pasar de una etapa a otra [MOND18]. En el momento en el que se cumple la transición, la etapa anterior se desactiva y se activa la etapa posterior.

Como ya se ha dicho, en los GRAFCETS descriptivos esta condición puede ser una proposición lógica, pero en el caso de los GRAFCETS tecnológicos de TIA Portal, la condición debe expresarse en LD (Ladder Diagram) o diagrama de contactos.

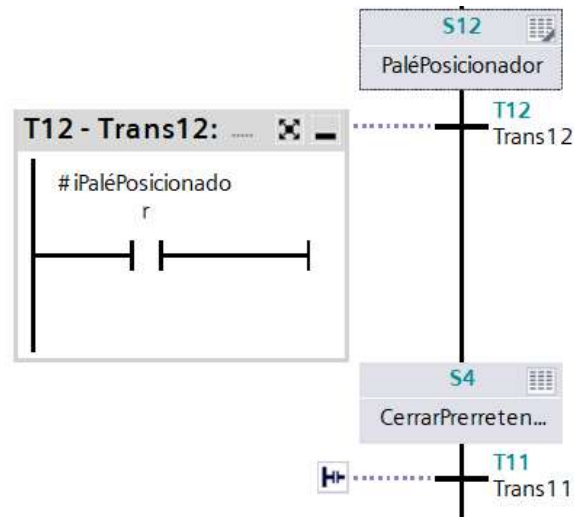


Figura 2: Ejemplo de transición en TIA Portal

En la figura 2 no se pasa de la etapa S12 a la S4 hasta que no se detecte un palé en el posicionador (variable iPaléPosicionador a 1).

### 3) GRAFCET tecnológico: Uniones

Son las líneas que unen las etapas. En este proyecto las hay de dos tipos: uniones simples y uniones de selección. Las uniones simples conectan una etapa con una única transición. La figura 2 muestra una unión simple. Las uniones de selección permiten formar caminos alternativos: si se cumple una transición se evoluciona a una etapa, si se cumple otra se evoluciona a otra distinta. Las transiciones tienen que ser mutuamente excluyentes, pues solamente se puede evolucionar a una de ellas. A continuación, se muestra un ejemplo para comprender mejor el funcionamiento:

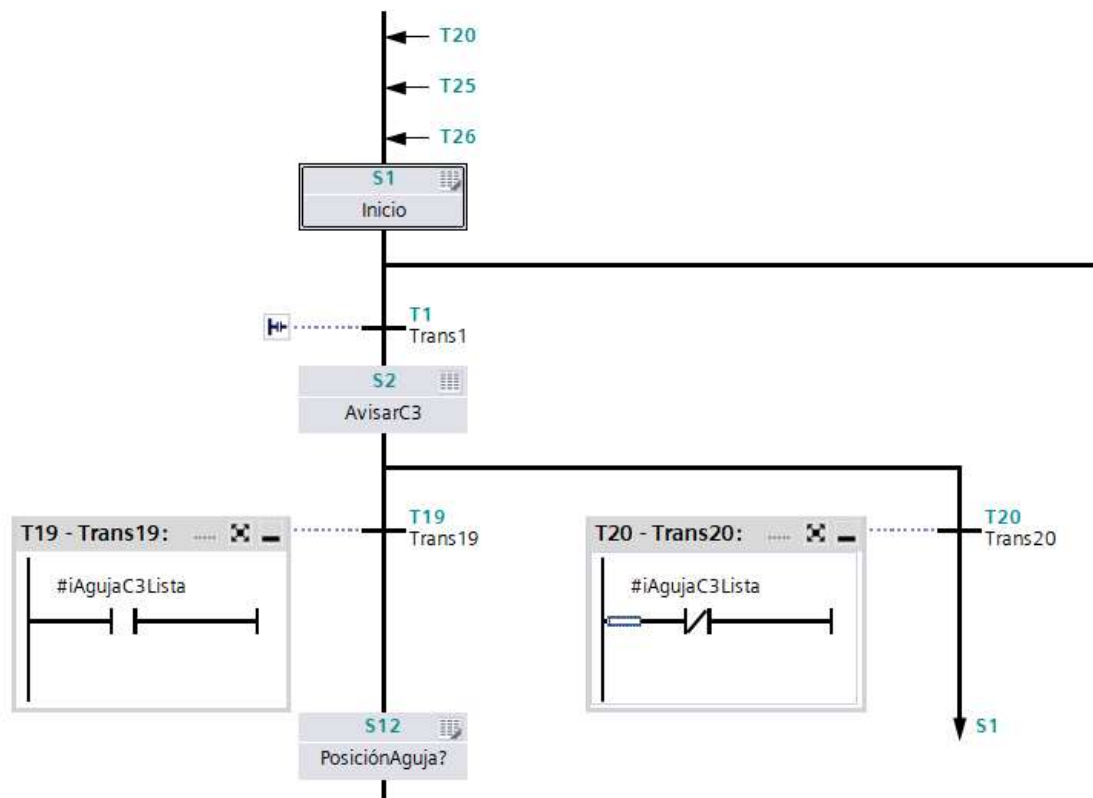


Figura 3: Ejemplo unión de selección

Este ejemplo sirve también para introducir cómo los dispositivos no solamente funcionan de forma individual, sino que también están programados para que haya comunicación entre ellos. En este caso, la aguja C2 espera a que la aguja C3 se coloque. Mientras que no recibe respuesta (agujaC3Lista=0), volverá a Inicio (S1) para después preguntarle otra vez si está lista. En cuanto reciba una señal afirmativa (agujaC3Lista=1), pasará a la etapa “PosiciónAguja?”, en la que se comprobará en que posición está la aguja para mantener esa posición o cambiarla.

4) GRAFCET tecnológico: Acciones

Las acciones representan las variables de salida que se activan en cada etapa. Se representan en GRAFCET dentro de rectángulos a la derecha de la etapa, y en TIA Portal mediante una lista como la siguiente:

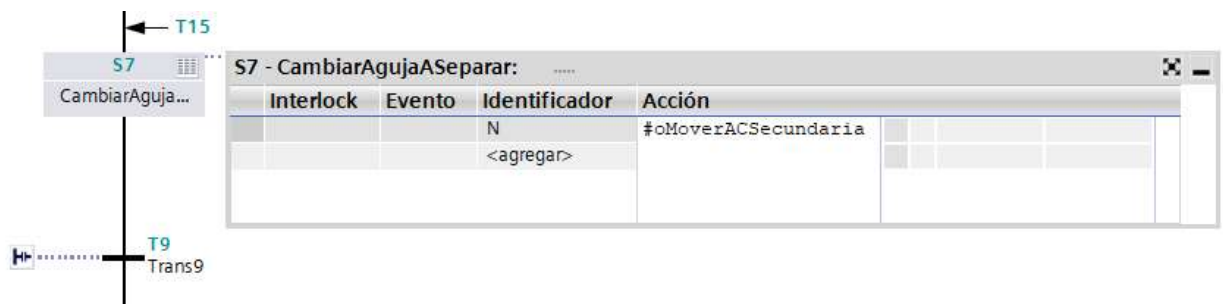


Figura 4: Ejemplo de acción

Se observa en la imagen como, cuando el GRAFCET se encuentre en la etapa CambiarAgujaASeparar, se pondrá a 1 la variable OMoverACSecundaria, que activará la válvula encargada de mover la aguja a esa posición. El identificador N indica que dicha variable permanecerá a 1 mientras que la etapa esté activa. Una vez que se verifique la transición 9, pasará a ejecutarse la etapa siguiente y OMoverACSecundaria se pondrá a 0.

### 2.2.2. Guía GEMMA.

La guía GEMMA es una plantilla creada por la ADEPA<sup>ix</sup> en 1984 que recoge los distintos modos de funcionamiento que puede tener un sistema automatizado, así como la manera de pasar de uno a otro [MOND18].

Aunque no se emplee directamente en este proyecto, ya que para pasar de un estado de funcionamiento a otro se utilizará otra plantilla más moderna (Pack-ML), se ha investigado mucho acerca de esta guía para poder diseñar las comunicaciones entre los GRAFCETs.

En la guía GEMMA, cada GRAFCET es un modo de funcionamiento de la fábrica: automático, manual, parada de fin de ciclo, parada de emergencia, etc. Para las comunicaciones entre estos estados, GEMMA establece unas instrucciones. Aunque en este proyecto, los GRAFCETs no representan estados sino dispositivos de la fábrica, también éstos necesitan comunicarse. Para ello, se han utilizado las instrucciones que establece esta guía, así como la información de como adaptar estas instrucciones a las posibilidades que permite TIA Portal.

La guía GEMMA establece dos tipos de dependencias entre GRAFCETs: jerárquica y de cooperación.

En la relación jerárquica, existe un GRAFCET superior o maestro que activa o desactiva otro GRAFCET, denominado esclavo o inferior. Esto se realiza mediante instrucciones de forzado emitidas por el GRAFCET superior [MOND18]. En este proyecto, para las instrucciones de forzado se han utilizado las variables INIT\_SQ, OFF\_SQ y EN.

Un ejemplo de este forzado dentro del proyecto es la parada de fin de ciclo. El paso del estado de producción normal a esta parada se produce gracias a un contador de palés que, al terminar su cuenta, fuerza a los GRAFCETs a que dejen de ejecutarse, poniendo a 1 las variables OFF\_SQ y a 0 la variable EN.

La guía GEMMA también permite que los GRAFCETs cooperen entre sí. Un GRAFCET pide un servicio a otro GRAFCET, y este le avisa cuando lo ha realizado.

Tanto en la guía GEMMA como en TIA Portal, esta comunicación síncrona se realiza a través de variables.

Como no se puede conectar la variable de salida de un GRAFCET con la variable de entrada de otro, se utilizan variables estáticas o temporales. Estas últimas se emplean cuando el GRAFCET receptor se encuentra después que el emisor, ya que se borran después de un ciclo de ejecución del programa principal.

### 2.2.3. PackML

PackML es una plantilla para automatizar sistemas industriales complejos. Durante este proyecto, se ha hecho un trabajo de investigación de esta plantilla para poder introducir sus ventajas en la minifábrica.

La principal ventaja de esta plantilla es que propone una forma común para automatizar, reduciendo las variantes en la programación entre un sistema y otro. Así, si un ingeniero tiene que reparar una máquina automatizada con esta plantilla, puede comprender fácil y rápidamente su funcionamiento, incluso si nunca ha trabajado con ella, ya que todos los sistemas automatizados con PackML tienen comportamientos comunes.

Este comportamiento común se consigue definiendo la misma estructura de programación. El código de programación quedará dividido por modos de funcionamiento y dentro de cada modo de funcionamiento, por estados. Las transiciones entre modos de funcionamiento y entre estados vendrán ya definidas por la plantilla y el programador solamente tendrá que preocuparse por diseñar los estados, pudiendo repartirse el trabajo fácilmente con otros programadores.

PackML también define una estructura común para los mensajes transmitidos por las máquinas para informar sobre sus condiciones de funcionamiento, tales como en qué estado se encuentra la máquina, el tiempo que llevan en ese estado, velocidad, etc. Estos mensajes se denominan packtags [OMAC21].

#### 2.2.3.1. Descripción

Descripción de Pack ML: Modos de funcionamiento

En la plantilla Pack ML vienen definidos ya 3 modos de funcionamiento, que son: producción, manual y mantenimiento. Considerando la enorme diferencia que puede haber entre procesos de producción, deja libertad al programador para añadir, a parte de estos 3 modos, hasta 27 modos de funcionamiento distintos, que son los definidos por el usuario (“User-defined unit modes”).

Descripción de Pack ML: Estados

La organización que PackML hace de cada modo de funcionamiento es parecida a un GRAFCET, en el sentido de que en cada modo hay también etapas y transiciones.

Las etapas se denominan estados. Hay 17 estados, y cada uno define un comportamiento específico del proceso. Por ejemplo, en el estado Execute (en el modo de producción), la fábrica produce objetos, o en el estado Completing (Completando) se realizan las actividades necesarias para finalizar el proceso.

Los nombres de los estados, la función que debe realizar el sistema en ellos y las transiciones entre estados no se pueden cambiar. Tampoco se pueden definir más estados. La razón principal de esto es reducir la cantidad de variantes entre un código para automatizar y otro.

Cumpliendo estas restricciones, el programador es libre de diseñar cada estado como crea conveniente.

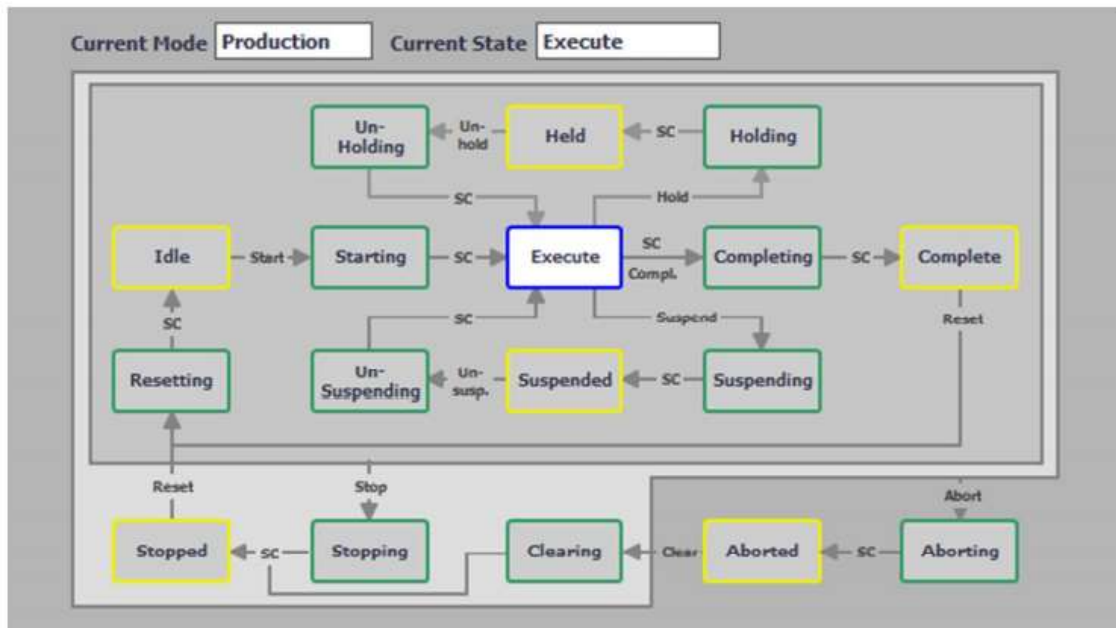


Figura 5: Ejemplo de plantilla Pack ML [SIEM15]

En la figura 5 se muestran los diferentes estados que puede haber. Los que aparecen en verde son estados de tipo Acción. En ellos se produce una actividad, y cuando se termina, se genera un flanco positivo en la variable SC para pasar a un estado de espera (amarillo). En el estado de espera, el sistema aguarda a que se produzca una orden para iniciar la siguiente acción. El color azul en el estado Execute indica que es el estado que se está ejecutando actualmente.

A continuación, se adjunta una tabla con información de cada uno de los estados:

Número	Estado	Descripción
0	Undefined	Estado no válido
1	Clearing	<b>Tipo de estado: acción</b> Tras haber pasado por el estado de Abort, en este estado se corrigen los fallos que hayan ocasionado la parada brusca del sistema.
2	Stopped	<b>Tipo de estado: espera</b> Sistema parado tras haberse completado el estado Stopping, a la espera de que se active la señal reset.
3	Starting	<b>Tipo de estado: acción</b> La máquina/sistema realiza las acciones necesarias para su puesta en marcha. Ejemplo: horno, precalentado.
4	Idle	<b>Tipo de estado: espera</b> Es el estado al que se llega una vez que se ha completado el reinicio del sistema (Resetting).
5	Suspended	<b>Tipo de estado: espera</b> En suspended el sistema no debe producir, aunque puede continuar funcionando si así fuera necesario. Permanecerá en este estado hasta que las

		condiciones externas vuelvan a la normalidad, momento en el que entrará en el estado Unsuspending.
6	Execute	<b>Tipo de estado: acción</b> Es el estado principal del modo. En el modo de producción, en este estado se hacen los productos. En el modo de limpieza, se limpia la máquina, etc.
7	Stopping	<b>Tipo de estado: acción</b> Se entra en este estado tras una orden de stop. En él se efectúa la lógica necesaria para llevar el sistema a un estado de parada controlada.
8	Aborting	<b>Tipo de estado: acción</b> Se entra en este estado tras una orden de abortar y en él se toman las acciones necesarias para que se produzca una parada rápida y segura.
9	Aborted	<b>Tipo de estado: espera</b> Se mantiene información relevante de la condición de abort. Únicamente con la orden de Clearing para solucionar manualmente lo que ha causado el problema se puede salir de este estado.
10	Holding	<b>Tipo de estado: acción</b> Se evoluciona a este estado cuando condiciones internas de la máquina no permiten continuar con la producción. Estas condiciones de la máquina no son graves y ocurren con bastante frecuencia y requieren una intervención mínima por parte del operador, como puede ser llenar el depósito a un dispensador de pegamento. Terminada esta intervención, el sistema debe continuar la acción que había dejado a medias, por lo que toda la información de la condición de la máquina en el momento en el que se activa el comando Hold debe quedar guardada en el PLC.
11	Held	<b>Tipo de estado: espera</b> El sistema espera en este estado a que sus condiciones internas sean adecuadas para reanudarse la producción. Cuando esto ocurra, se activará el comando Unhold.
12	Unholding	<b>Tipo de estado: acción</b> Típicamente, la máquina o sistema entrará en este estado automáticamente cuando sus condiciones internas sean aptas para la producción. También el comando Unhold puede ser introducido por un operador.
13	Suspending	<b>Tipo de estado: acción</b> La máquina o sistema entran en este estado cuando condiciones externas no permiten continuar con la producción. Por ejemplo, un robot que introduce botellas en una caja puede entrar en este estado cuando la caja se ha llenado y está siendo sustituida por una nueva. La máquina evoluciona a una situación de parada controlada y toda la información sobre su condición antes de activarse el comando suspend se deben guardar en el PLC.

14	Unsuspending	<b>Tipo de estado: acción</b> Lleva a cabo las acciones para que el estado pase de Suspended a Execute.
15	Resetting	<b>Tipo de estado: acción</b> Resultado de activarse el comando Reset en el estado Stopped o Complete. Se corrigen los fallos y las causas de parada, y los mecanismos de seguridad se energizan. Se pondrá la máquina en el estado Idle donde se esperará el comando de Start para comenzar otra vez la producción.
16	Completing	<b>Tipo de estado: acción</b> Si la fábrica no es de producción continua, tras concluir el proceso de producción se evolucionará a este estado.
17	Complete	<b>Tipo de estado: espera</b> Una vez finalizadas las acciones relacionadas con la parada a final de ciclo, el sistema permanecerá en este estado se activa el comando Reset para reiniciar el sistema.

*Tabla 2: Estados de la plantilla Pack ML [SIEM15]*

Por falta de tiempo, en este proyecto solo se automatiza el modo de funcionamiento de producción.

Dentro del modo de producción, no se van a diseñar todos los estados, ya que muchos de ellos, debido a la naturaleza del sistema a automatizar, no son muy relevantes.

Por ejemplo, la minifábrica no tiene la necesidad de interrumpir su producción debido a condiciones internas (no hay material que se agote). Tampoco intervienen más máquinas en el proceso de producción, por lo que no se debería de interrumpir su funcionamiento normal por condiciones externas.

Por ello, en este proyecto, se han automatizado únicamente los estados que se ha considerado que tienen más relevancia: el estado Execute o de producción normal, la parada de emergencia, la parada a final de ciclo y un reset para que el sistema se restablezca después de una parada y vuelva a funcionar con normalidad.

### 2.2.3.2. Librería LPMLV30 de Siemens

Para implantar la plantilla Pack ML, se ha descargado un proyecto en TIA Portal desde la página web de Siemens. En este proyecto hay una plantilla formada por segmentos vacíos que el usuario debe ir rellenando, y una librería de bloques de funciones.

#### 2.2.3.2.1. Segmentos

En cada estado vienen 3 segmentos:

Uno al inicio. Junto a este segmento hay una descripción de lo que establece la OMAC que debe hacer la máquina en este estado.

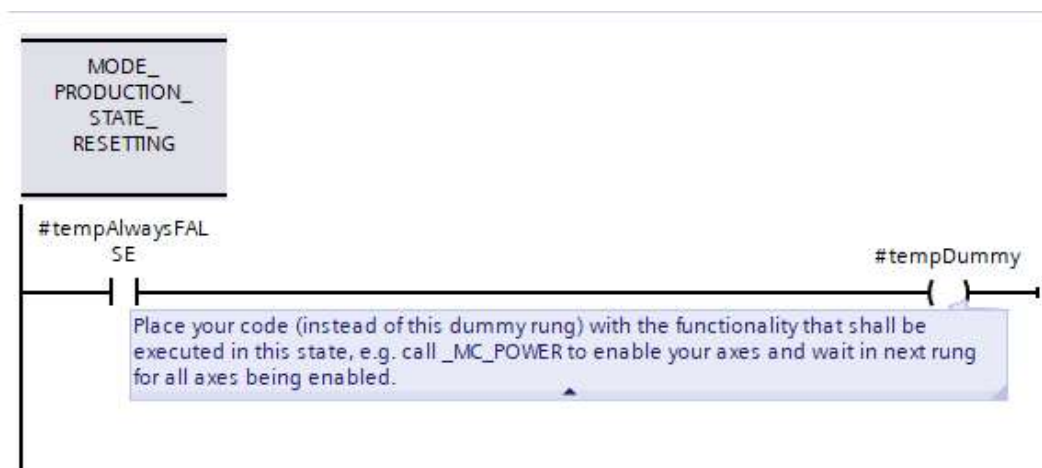


Figura 6: Segmento de inicio. El programador puede sustituir la bobina *tempDummy* por su código o ponerlo en un segmento debajo.

Después, hay otro segmento, para habilitar las transiciones de un estado a otro. El programador borra el contacto #tempAlwaysFalse (siempre 0) que viene por defecto en la plantilla. A continuación, lo sustituye por una variable (o una combinación de variables) utilizada para el cambio de estado. Por ejemplo, una variable que se pone a 0 al pulsarse la seta de emergencia o la variable de salida de un contador de palés terminados para la parada de fin de ciclo.

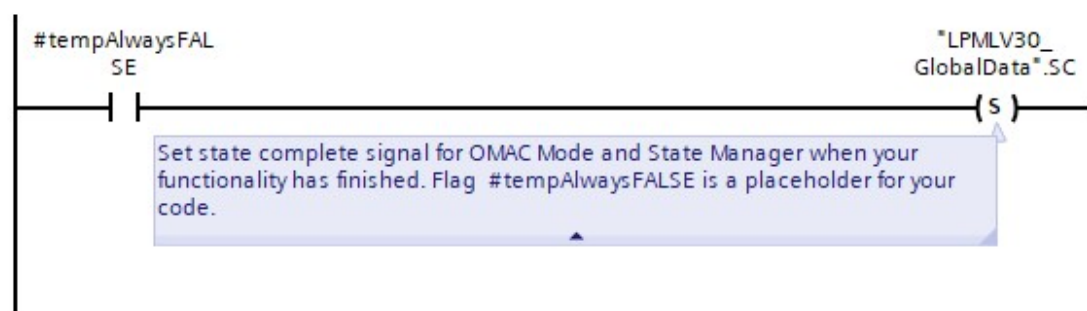


Figura 7: Segmento que habilita el cambio de estado

Por último, un segmento final que no se debe modificar

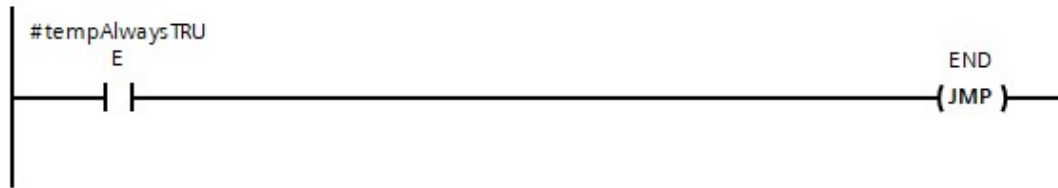


Figura 8: Segmento al final del estado

Este segmento permite saltar directamente al segmento final de la plantilla sin ejecutar el resto de los estados.

#### 2.2.3.2.1. Bloques de Funciones

Además, el proyecto de Siemens cuenta con una librería de bloques de funciones. Dentro de estos bloques, uno muy relevante es el UnitModeStateManager. En él, va programada la lógica necesaria para pasar de un estado de funcionamiento a otro, o de un modo de funcionamiento a otro, para lo cual, el usuario tiene que activar el comando adecuado.

Este comando es un flanco positivo en la variable SC, en el caso de que el sistema se encuentre en un estado de tipo acción. En cualquier otro estado, se hará caso omiso a lo que ocurra en la variable SC y solo se cambiará de estado si el usuario activa una variable de cambio de estado válida. Por ejemplo, desde Execute no se puede pasar a Reset, pero sí a Complete, activando la variable CompleteCmdRequest.

Estas variables son entradas del bloque UnitModeStateManager. Todas ellas son variables binarias. Si se quiere utilizar variables de tipo entero para cambiar de estado o de modo, se debe desactivar la interfaz booleana, poniendo a 0 la entrada del bloque “enableBooleanInterface”.

En función del modo de funcionamiento solicitado, UnitModeStateManager variará el valor de una variable de tipo entero denominada “UnitModeCurrent”, donde cada valor de esta variable es un modo de funcionamiento.

A continuación, después de convertirse en un entero sin signo, dicha variable es la entrada de una lista de saltos como la siguiente:

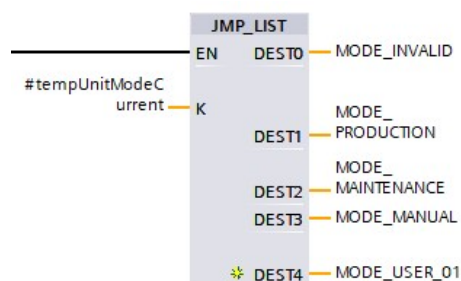


Figura 9: Lista de saltos en la librería LMPLV30

En esta lista de saltos, el valor de K determinará qué segmentos se deben ejecutar, en función del modo de funcionamiento que haya elegido el usuario. Por ejemplo, si el usuario quiere que la máquina funcione en modo producción, el programa saltará al siguiente segmento:

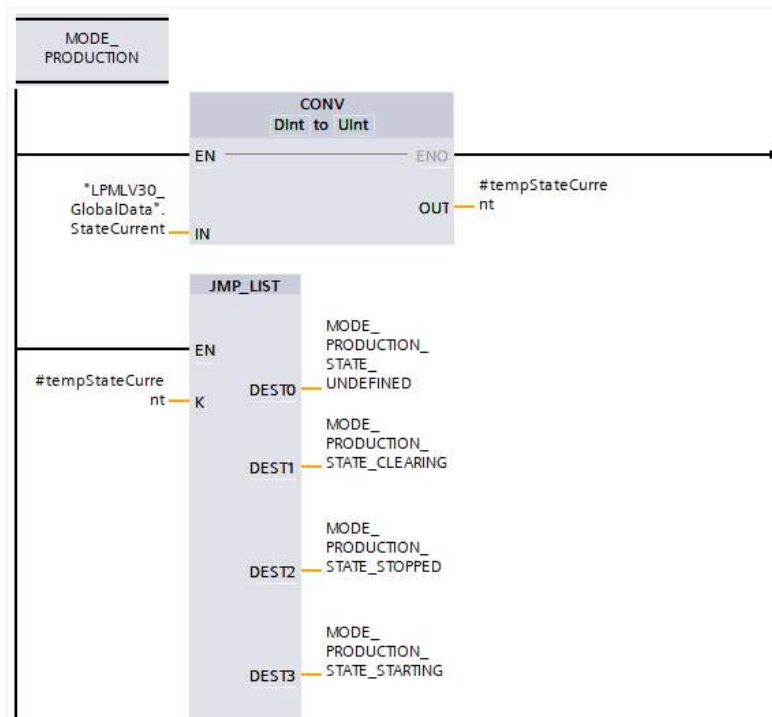


Figura 10: Lista de saltos dentro del modo Producción

Desde donde se elegirá el estado en función de una variable de tipo entero que también es una salida de UnitModeStateManager.

Además, la librería incluye otros bloques, que permiten controlar el tiempo que el sistema pasa en un estado, elegir el número de estados que puede haber dentro de un modo de funcionamiento, modos de funcionamiento que tiene el automatismo, etc.

### 2.3 Descripción de la minifábrica ICAI

En este capítulo se describirá la fábrica a automatizar. Esta fábrica es la planta situada en el laboratorio de sistemas digitales de la escuela técnica superior de ingeniería ICAI. Su apariencia a primera vista es la de una cadena de montaje de pequeño tamaño, por lo que se la denomina Minifábrica.

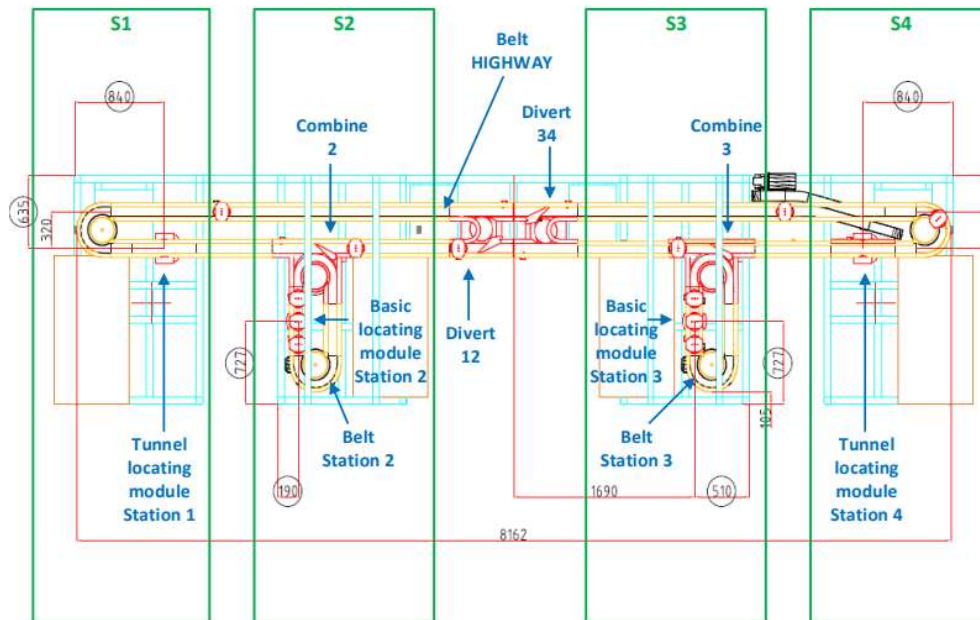


Figura 1: plano de la minifábrica [MOND20]

Como se puede observar en la figura 1, la fábrica se divide en 4 estaciones, cada una de ellas habilitada con un robot IRB120, y una cámara.

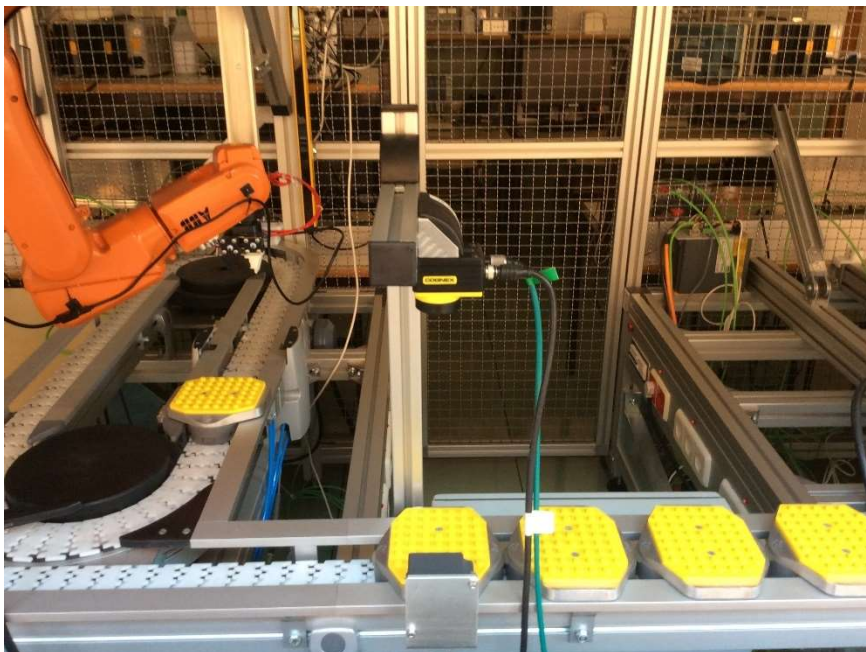
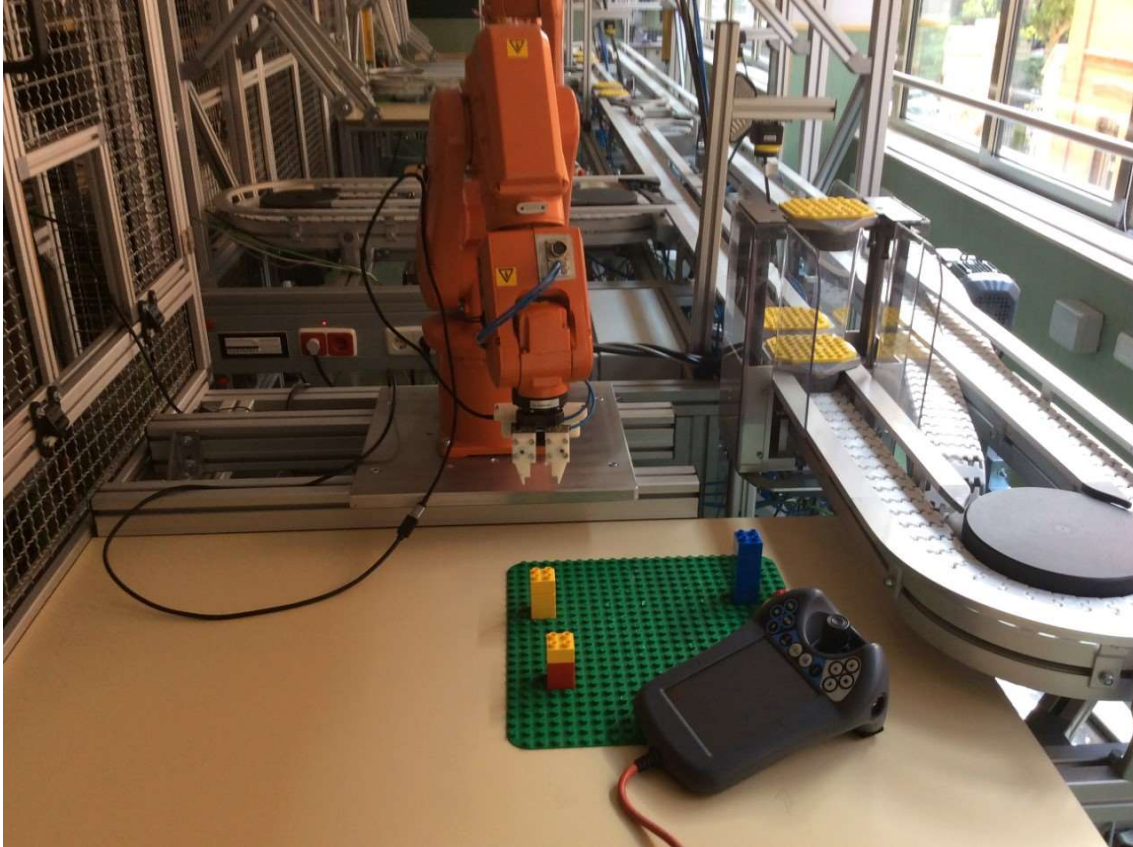


Figura 2: Estación S2

El IRB120 es un robot de la marca ABB. Pesa 25 kg y su capacidad de carga máxima son 4 kg. En cuanto a movimientos, tiene hasta 6 grados de libertad. Su control se realiza desde un dispositivo IRC5 y su programación se realiza desde la unidad de programación Flex Pendant [SIGM20].



*Figura 3: Robot IRB 120*

En la figura 3 se observa el manipulador del robot, junto con el FlexPendant. Como se puede comprobar también, la función del robot va a ser la de colocar piezas de lego sobre el palé, cuando este se encuentre inmobilizado por alguno de los posicionadores (en la figura 3, el posicionador de tipo túnel de la estación 4).

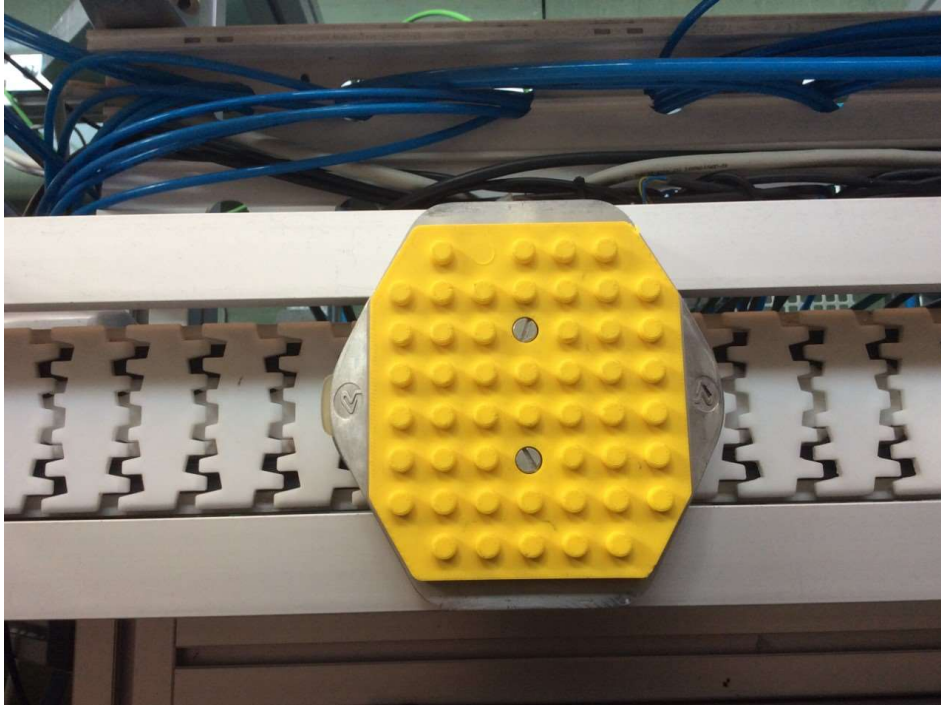
No se profundizará en detalles sobre las características del robot al no realizarse su automatización en este proyecto. Se simulará su actuación por medio de temporizadores, de manera que el posicionador túnel de la imagen, esperará a que pase el tiempo del temporizador, y a continuación bajará el palé y lo libera para que este pueda avanzar a la siguiente etapa.

### 2.3.1. Palé [FLEX16]

El palé tiene una base de aluminio fundido sobre la que se ha atornillado una base plana de lego hecha de plástico, donde el robot va a colocar las piezas.

Los palés son arrastrados por la cinta gracias a dos discos de guiado, situados por debajo de la base de aluminio. Estos discos están cubiertos por una placa de material plástico.

Los palés son de la marca FlexLink, modelo x85. Su carga máxima son 10 kg y están equipados con una etiqueta para ser detectados por sensores RFID (usados por la cámara) a velocidades de hasta 30 m/min.



*Figura 4: Palé*

### 2.3.2. Cintas

En cuanto a las cintas, existe una cinta principal que comunica las estaciones, junto con dos cintas secundarias (satélites) de menor tamaño, situadas en las estaciones 2 y 3. Las cintas secundarias albergan los posicionadores básicos mientras que en la cinta principal se encuentran los posicionadores de tipo túnel. La cinta principal tiene la posibilidad de dividirse en dos cintas por la mitad gracias a un módulo de desviación/combinación en forma de agujas.

Para el desplazamiento de palés entre cinta principal y satélites, la cinta está equipada con dos módulos de desviación/combinación (también en forma de agujas). Estas agujas girarán 45 grados, para permitir al palé desplazarse del satélite a la cinta principal o viceversa.

La cinta transportadora se mueve solo en un sentido y su automatización es muy sencilla. Tan solo se controlará con una variable, para su puesta en marcha.

Esas son las características específicas de la cinta transportadora en la minifábrica [MOND20].

En cuanto a sus características generales, la cinta transportadora es una x-85 de la marca FlexLink (modelo y marca se corresponden con los del palé). Como material para la cinta se usa el poliacetal o polioximetileno, que es un material plástico que proporciona mucha rigidez y resistencia al desgaste. La forma es la de cadena plana:

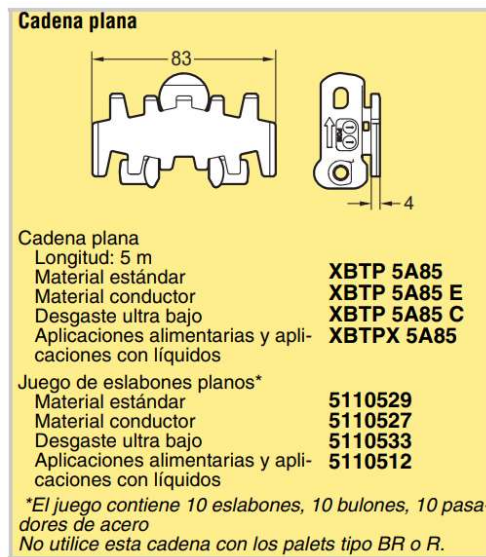


Figura 5: Características generales de la cinta [FLEX21][1]



Figura 6: Motor de la cinta

El motor es de la marca SEW-EURODRIVE. Es un motor trifásico. Funciona a 50 Hz y su velocidad nominal son 1435,27 rpm, cos phi nominal 0,77 y se puede alimentar tanto en triángulo como en estrella, a 230V y 400 V respectivamente.

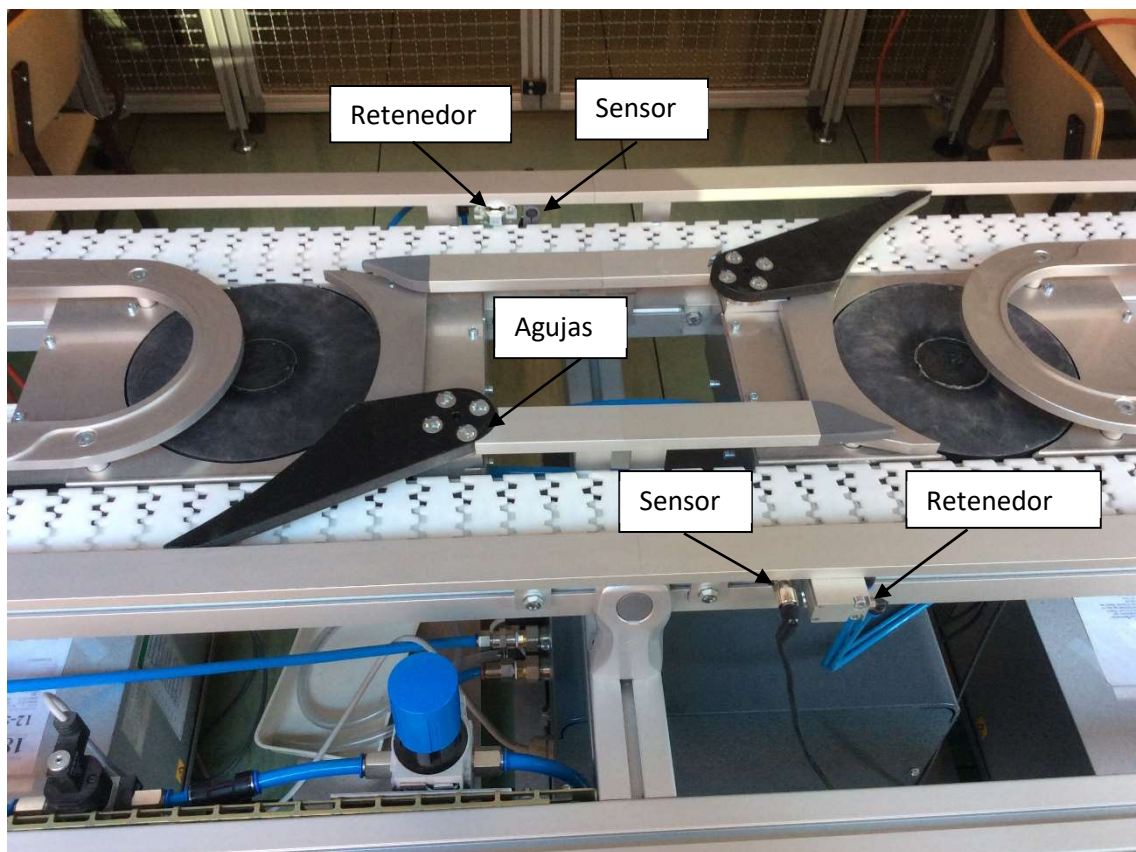
### 2.3.3. Módulos de desviación/combinación interespacial

El módulo de desviación/combinación interespacial está situado en medio de la cinta principal y la divide en dos si así fuera necesario. También se le conoce como agujas Divert. Este

dispositivo permite tener dos sistemas de producción funcionando a la vez, e independientes entre sí.

En este proyecto, en condiciones de trabajo normales, las agujas Divert se colocarán en una posición que permita al palé recorrer toda la cinta principal. Solo cambiarán de posición, dividiendo en dos la cinta principal, tras dar por concluida la producción.

Las agujas Divert están provistas de sensores inductivos que la advierten de la proximidad de un palé. También tienen retenedores o barreras que evitan que un palé se cuele durante el movimiento de las agujas. Hay dos agujas y cada una tiene dos barreras retenedoras y dos sensores: un grupo sensor-barrera está situado en la cinta de ida y otro en la cinta de regreso. También tienen sensores que informan de la posición en la que se encuentra cada aguja.



*Figura 7: Agujas Divert*

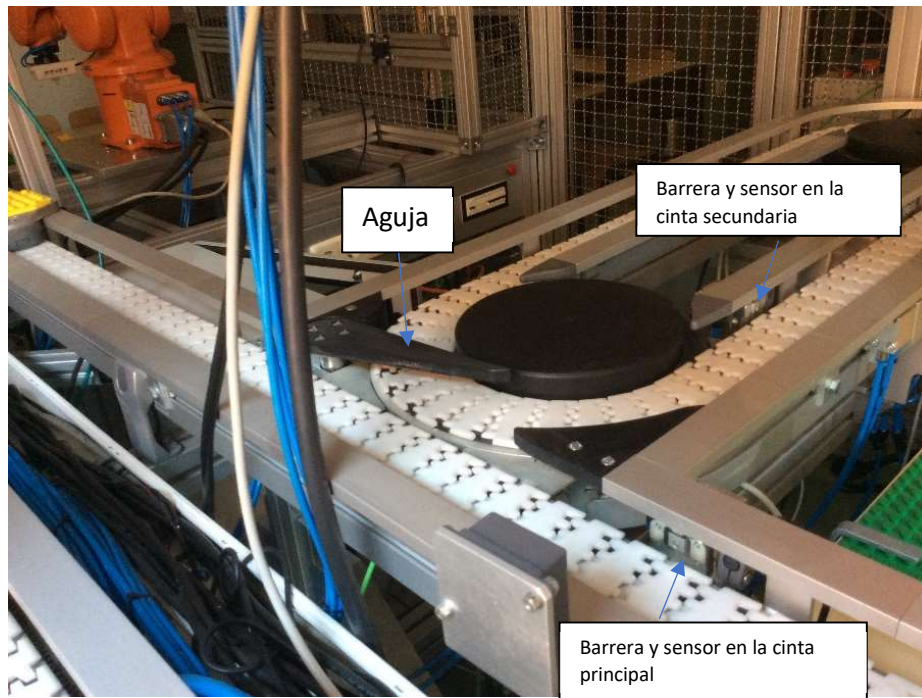
En la imagen, las agujas están colocadas en la posición ShortCut o recorrido corto. Se puede observar también que hay dos cables de aire (color azul) saliendo del retenedor, lo que indica que se controla con una válvula neumática de doble efecto.

Efectivamente, el movimiento de las agujas y de las barreras se produce gracias a electroválvulas neumáticas de doble efecto. Habrá dos válvulas en cada barrera, una para abrir y otra para cerrar. En las agujas, también hay dos, que se corresponde con las dos posiciones que tiene la aguja. La actuación de las válvulas se controla desde un PLC.

### 2.3.4. Módulos combinados de desviación/combinación

Tal y como se ha explicado en la introducción, estos módulos (también conocidos como agujas Combine o C) permiten el desplazamiento de un palé desde la cinta principal a la secundaria, y viceversa.

Cuentan con sensores inductivos en ambas cintas, acompañados de barreras.

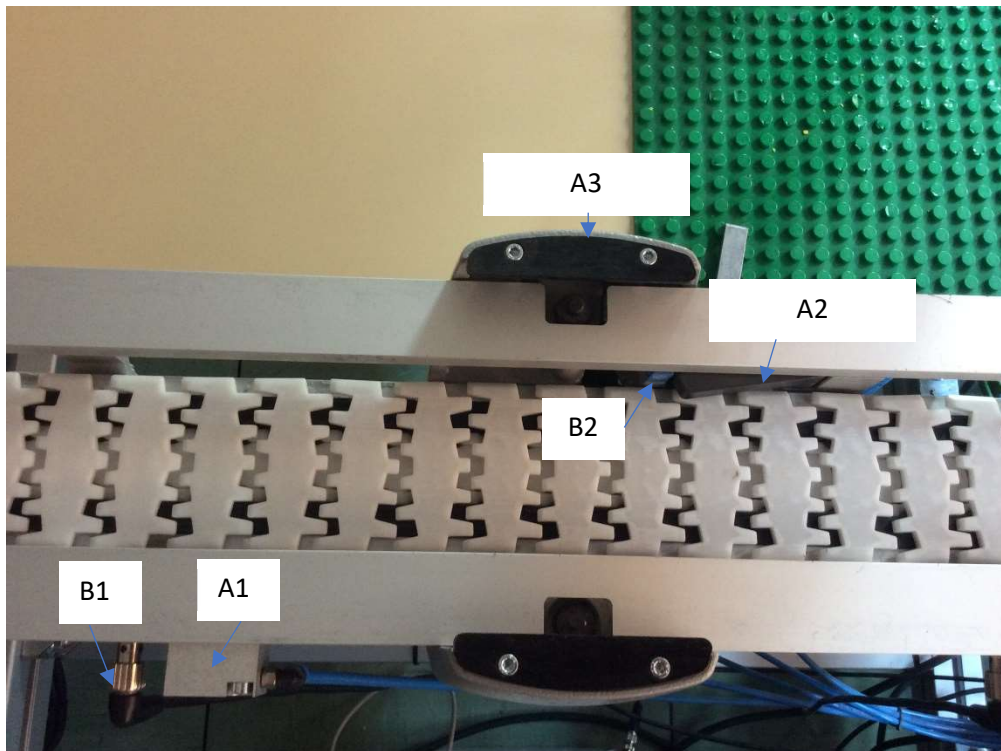


*Figura 8: Agujas Combine*

### 2.3.5. Posicionadores de tipo básico

Hay dos posicionadores básicos colocados cada uno en una cinta secundaria. Su función es la de retener el palé para que se pueda realizar una o varias fases del procesado de un producto.

Elementos [FLEX21]



*Figura 9: Posicionador Básico*

**Sensores:**

B1: Sensor de la barrera A1.

B2: Sensor de la barrera A2.

**Actuadores:**

A1: Barrera A1 "pre-stop".

A2: Barrera A2 "stop".

A3: Posicionador.

**Posicionadores de tipo básico: Barrera A1**

La barrera A1 retiene el flujo de palés mientras que haya uno procesándose en el posicionador. Funciona con una válvula neumática de doble efecto.

Esta barrera está equipada con un sensor inductivo que detecta la presencia de un palé, colocado inmediatamente antes del cilindro en la dirección de flujo de los palés.

Para aumentar la seguridad, en este proyecto, la barrera A1 permanecerá cerrada mientras no se detecte un palé en la entrada. Cuando el sensor detecte un palé, se comprobará si no hay un proceso en marcha, es decir, si no hay un palé en el posicionador. Si no lo hay, esperará a que se cierre la barrera situada delante del posicionador, y a continuación se abrirá. Una vez que se haya detectado un palé dentro del posicionador, volverá a cerrarse.

**Posicionadores de tipo básico: Barrera A2**

Una vez liberado por la barrera A1, el palé que necesite ser procesado es retenido por la barrera A2, situada delante del posicionador. En ese momento el proceso puede comenzar.

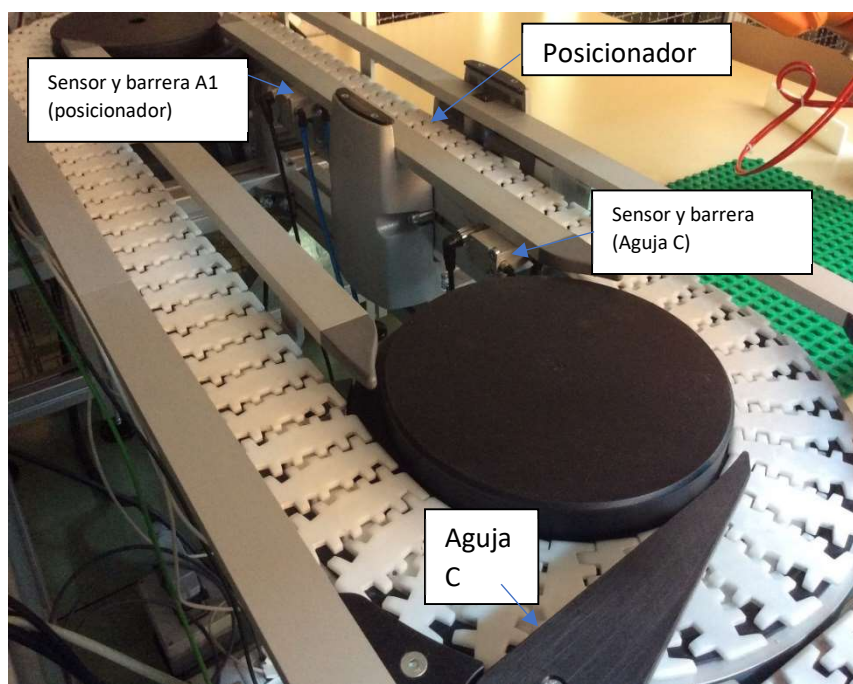
Al igual que la A1, también funciona con una válvula y lleva un sensor incorporado.

En este proyecto, normalmente la barrera A2 está abierta, y solo se cierra durante el procesado del producto.

Posicionadores de tipo básico: Posicionadores

Los posicionadores se encargan de inmovilizar el palé para que el robot pueda realizar sobre él trabajos de precisión. Si no, estaría continuamente moviéndose por la fricción con la cinta. También se controlan con válvulas.

En la imagen siguiente, se observa una de las cintas secundarias o satélites. Para que la automatización se haga correctamente, es preciso que agujas y posicionadores interactúen, desplazando palés a través de las cintas en función de qué posicionador ha terminado su tarea:



*Figura 9: Interacción entre aguja y posicionador*

### 2.3.6. Posicionadores de tipo túnel

Los posicionadores de tipo túnel están situados en cada extremo de la cinta principal, en las estaciones 1 y 4.

Su función es la de inmovilizar el palé, pero lo hacen levantándolo de la cinta:



*Figura 8: Imagen del posicionador túnel*

En la imagen anterior, el palé situado en la parte superior está listo para ser procesado por el robot. Por otra parte, este posicionador permite que los palés que no necesiten ser procesados en esta etapa continúen su movimiento, reduciendo los tiempos de producción.

#### Elementos [FLEX21]

##### Sensores:

B1: Sensor de la barrera A1.

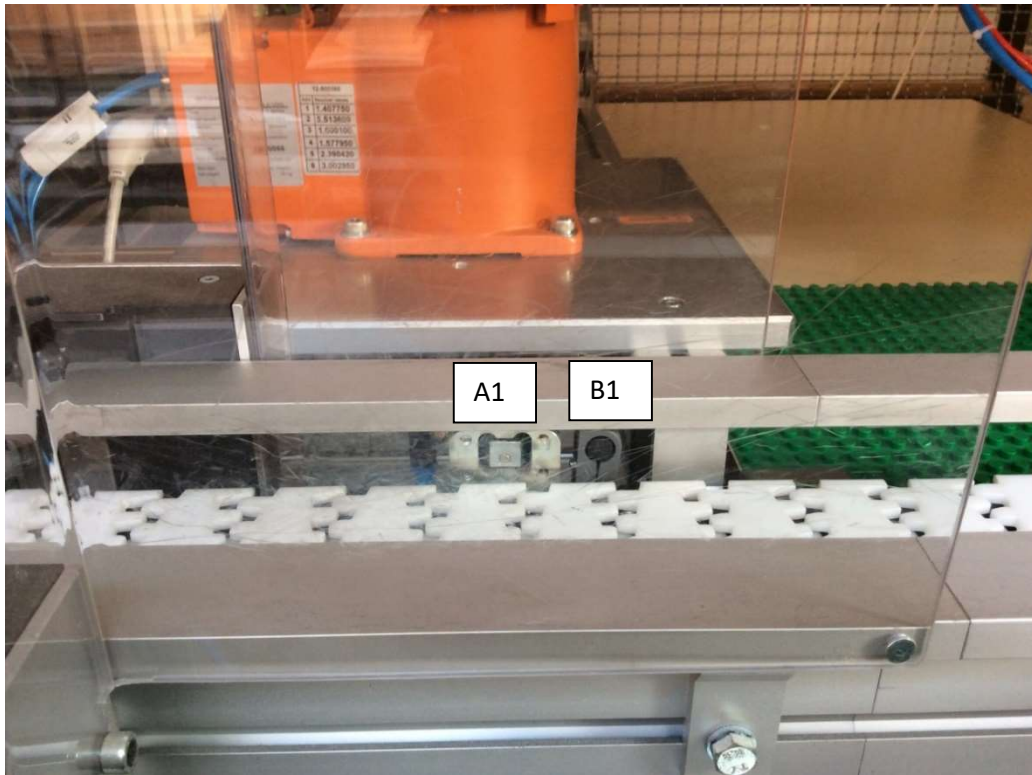
B2: Sensor de la barrera A2.

##### Actuadores:

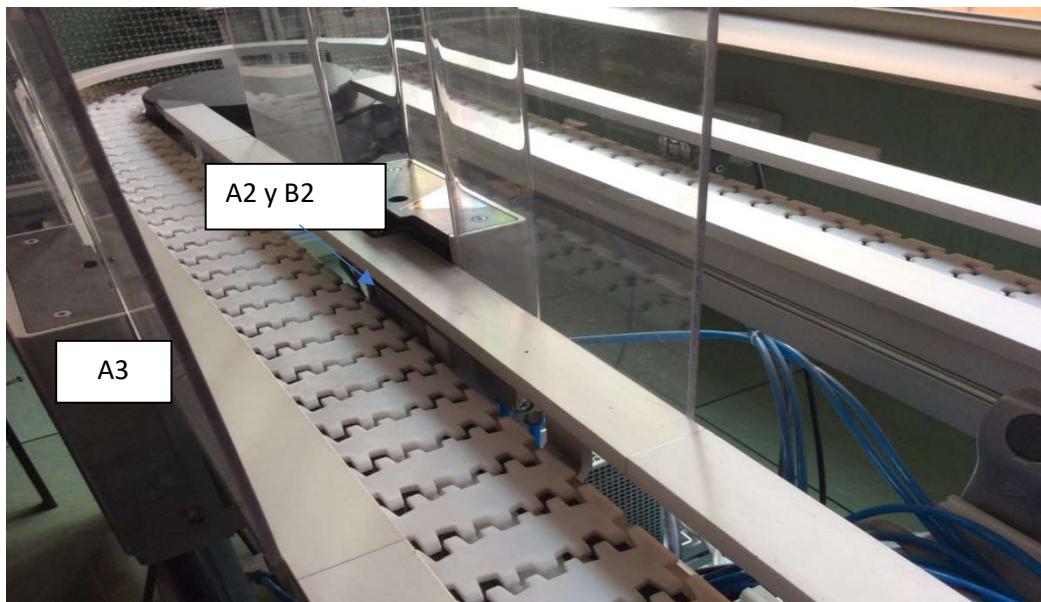
A1: Barrera A1 “pre-stop”.

A2: Barrera A2 “stop”.

A3: Posicionador.



*Figura 9: Detalle de sensor y barrera*



*Figura 9: Resto de elementos del posicionador túnel*

El funcionamiento del posicionador túnel es muy similar al del posicionador básico, solo que la acción de inmovilizar se produce mediante un dispositivo elevador, que sube el palé que es necesario procesar, dejando que aquellos palés donde no se tenga que aplicar ese proceso pasen por debajo.

Para informar al dispositivo elevador de que ha llegado a la parte más alta o a la parte más baja, el posicionador tiene dos finales de carrera.

### 3. Descripción del modelo desarrollado

#### 3.1. Programación del estado Execute (Fabricación de productos)

El estado Execute de la plantilla Pack ML describe el modo de operación habitual de la fábrica. Dentro del modo producción, en el estado Execute se procesan los productos.

En el caso de la minifábrica, este procesado consistirá en hacer pasar a los palés por todos sus posicionadores. En cada uno de ellos se retendrá el palé para que el robot pueda llevar a cabo una acción sobre él, en este caso, la de colocar una pieza de lego.

En este proceso también intervendrán los módulos de desviación/combinación o agujas C. Estas se encargarán de que el palé pueda pasar de una cinta satélite a la cinta principal, o viceversa.

Los módulos de desviación/combinación interespacial (agujas D) no intervienen en este estado. Dado que se van a utilizar todos los elementos de la minifábrica, permanecerán en una posición que permita que el palé pueda moverse por toda la cinta principal.

El recorrido de cada palé por la minifábrica será el siguiente<sup>x</sup>:

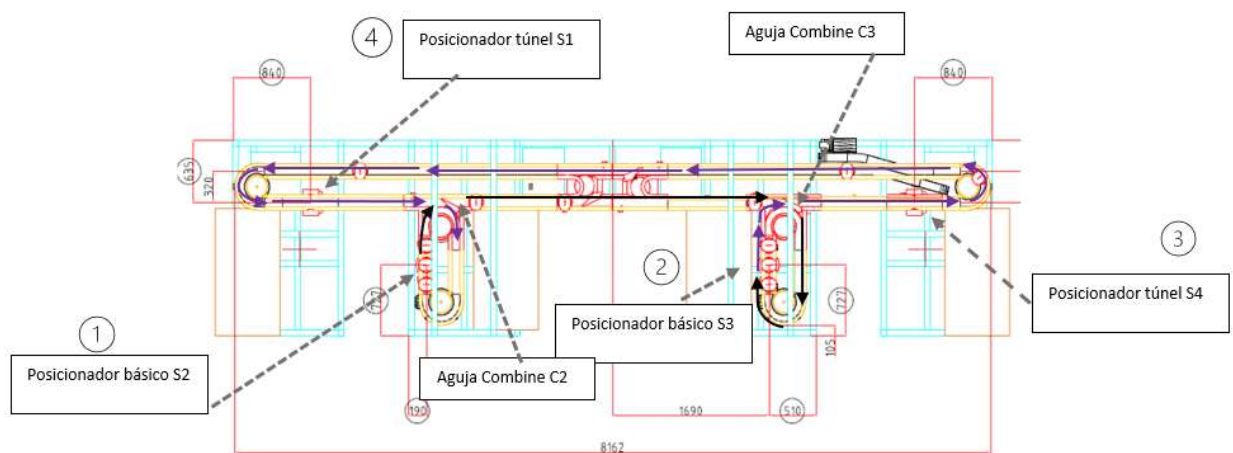


Figura 1: Plano de la minifábrica [MOND20]

El funcionamiento de cada elemento se programa mediante un GRAFCET. Los GRAFCETs se comunican entre sí mediante variables. Gracias a esta comunicación, por ejemplo, cuando hay un palé terminado en el posicionador básico S2, éste puede avisar a la aguja C2 para que se cambie de posición y el palé pueda desplazarse a la cinta principal.

También se utiliza el lenguaje de diagrama de contactos para representar funciones lógicas del programa.

### 3.1.1. Arranque de las cintas

La programación de las cintas es muy sencilla, ya que solo se controlan con una variable. Esta variable es “poArrancarCintas” cuya activación se produce con el siguiente diagrama de contactos:

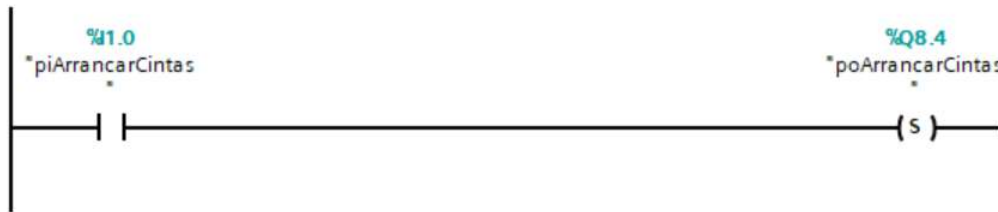


Figura 2: Automatización de las cintas

El lenguaje KOP o diagrama de contactos simula un circuito eléctrico, pero no deja de ser un lenguaje de programación. Como en todo lenguaje de programación, habrá unas variables de entrada (en la figura 2, piArrancarCintas) y unas variables de salida (poArrancarCintas). Las variables de entrada se representan mediante interruptores. Los normalmente abiertos como el de la figura se cerrarán cuando su variable asociada valga 1. Estos interruptores forman parte del programa, pero no son interruptores reales. El interruptor real o físico se encuentra fuera del plc y es el que, tras ser accionado por el operador, pondrá a 1 la variable piArrancarCintas.

Las variables de salida en KOP se representan mediante bobinas. Cuando a la bobina llegue un 1 por su izquierda (al cerrarse el interruptor) la variable poArrancarCintas se pondrá a 1. Además, la de la figura 2 es un caso especial de bobina, en el que una vez que se ha puesto a 1 la variable, no se vuelve a poner a 0 si deja de llegar un 1 a la izquierda de la bobina. De esa manera el operador puede soltar el botón sin que se dejen de mover las cintas.

PoArrancarCintas, que controla el encendido del motor, solo se pondrá a 0 cuando se desactive mediante una bobina reset, que pone a 0 su variable asociada cuando llega un 1 por su izquierda:

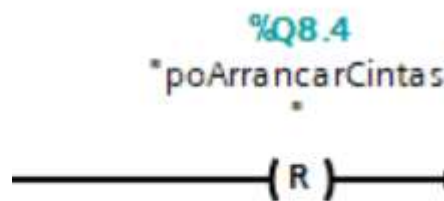


Figura 3: Ejemplo de una bobina Reset.

### 3.1.2. Posicionador básico S2

El funcionamiento del posicionador básico S2 comienza cuando el sensor de su entrada detecta un palé y no hay uno procesándose ya dentro del posicionador. Además, dado que este dispositivo es tanto la primera como la última etapa en el recorrido del palé, en su entrada puede haber productos no procesados todavía y productos ya terminados que no deben entrar en el posicionador. La manera de hacer esta distinción es mediante la variable *iUltimoPalé*, que vale 0 en el primer caso.

Si la variable *iUltimoPalé* vale 0, hay un palé a la entrada del posicionador y no hay un palé procesándose ya dentro de él, el posicionador cerrará su barrera de salida, y a continuación, abrirá la de su entrada.

Una vez el palé haya entrado dentro del posicionador, la barrera de entrada se cerrará, y comenzará a inmovilizarse el palé.

Inmovilizado el palé, el robot realizará su acción. Cuando termine, se dejará de inmovilizar el palé y se comunicará a las agujas que se coloquen en la posición adecuada para que el palé pueda pasar a la siguiente etapa en su recorrido, el posicionador básico S3.

Todo esto se realiza mediante un GRAFCET, descrito a continuación:

#### 3.1.2.1. Transiciones

En el lenguaje GRAPH de Tia Portal, las condiciones para la transición entre etapas se deben expresar en diagrama de contactos. Por ejemplo, tal y como se explicaba antes, para comenzar el funcionamiento del posicionador, se debe cumplir la siguiente condición:

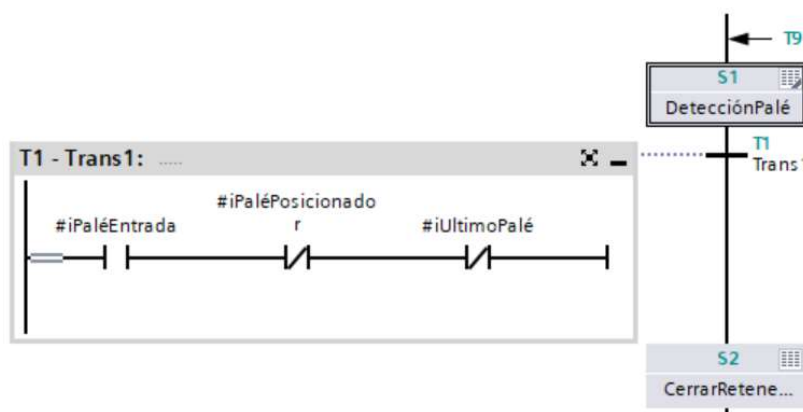


Figura 4: Programación del cierre de la barrera de salida

La mayoría de las acciones del posicionador se controlan con tiempo. Por ello, la condición de transición a muchas etapas del GRAFCET es una variable situada a la salida de un temporizador.

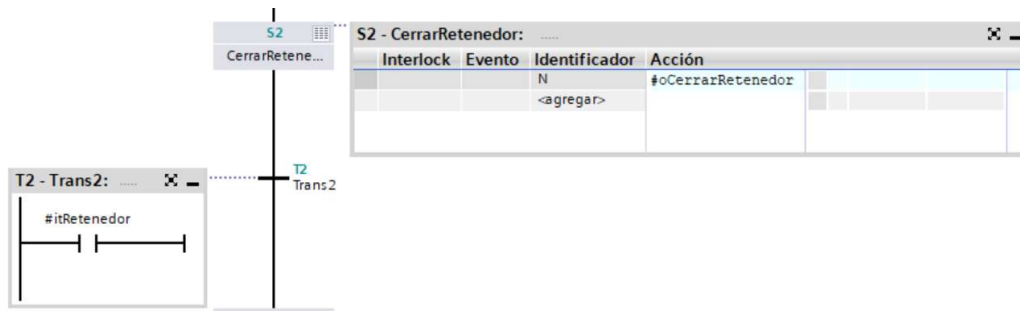


Figura 5: etapa CerrarRetenedor

En el caso de la figura 5, el control por tiempo del cierre de la barrera evita sobrecalentar la válvula encargada de dicho cierre cerrándola más tiempo del necesario.

Por otra parte, el temporizador encargado de activar dicha variable se pone a 1 en cuanto empieza a cerrarse la barrera (también cuando comienza a abrirse porque es el mismo para las dos acciones):

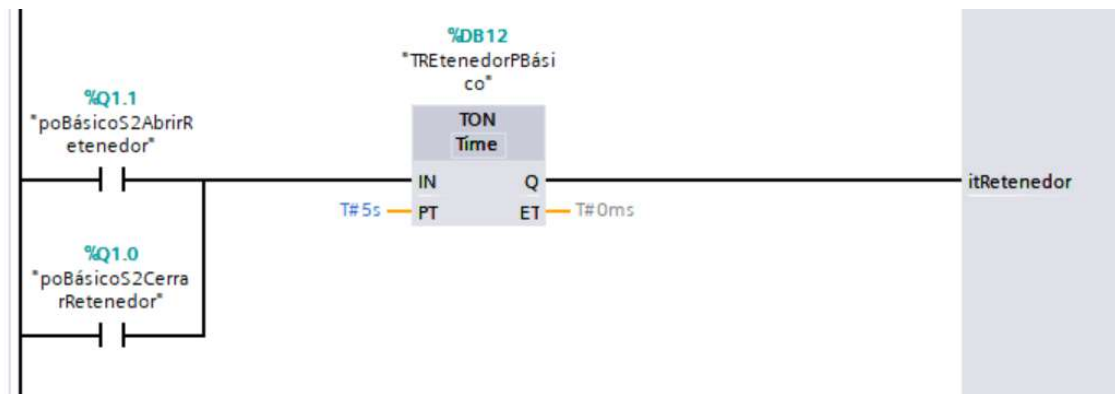


Figura 6: Temporizador que controla el tiempo de cierre de la barrera de salida.

Aunque algunas acciones se controlan con tiempo, otras se controlan con sensores. Por ejemplo, el cierre de la barrera de entrada no ocurre hasta que no se detecta un palé dentro del posicionador:

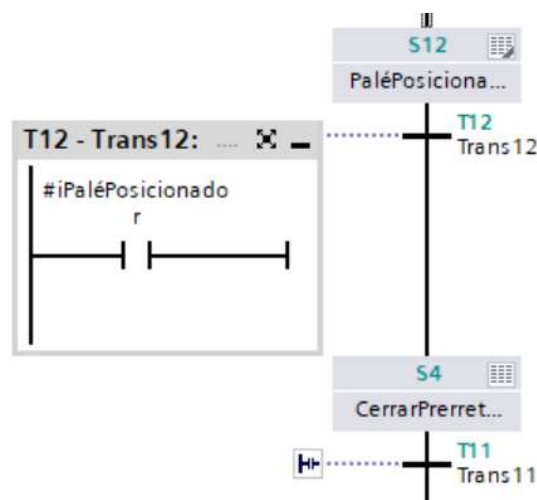


Figura 7: Etapa PaléPosicionador.

### 3.1.2.2. Acciones

Las acciones de este GRAFCET son acciones asociadas a etapas, y todas ponen una variable a 1 cuando el posicionador realiza una acción, como puede ser inmovilizar el palé, cerrar la barrera de salida, avisar a las agujas...

Esta variable vuelve a ponerse a 0 al salir de la etapa. En la figura 8 se muestra un ejemplo:

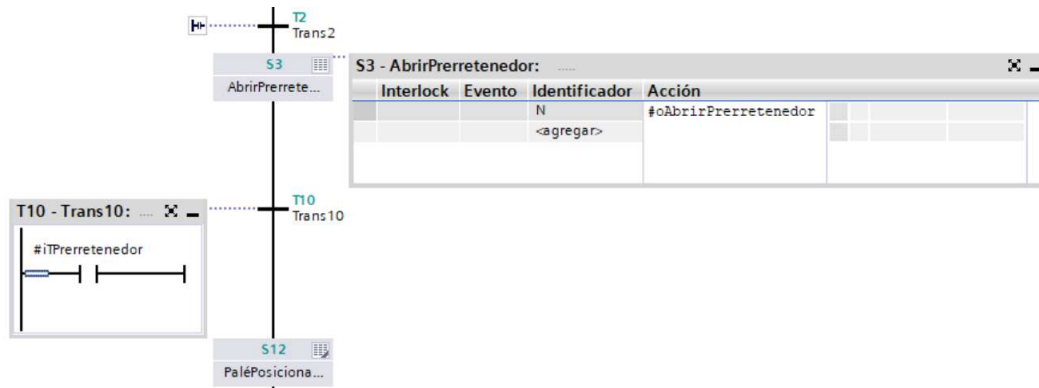


Figura 8: Programación de la apertura de la entrada del posicionador.

### 3.1.2.3. Comunicación con las agujas

Una vez terminada la acción del robot, y antes de abrir la barrera de salida, el posicionador manda una señal a la aguja C2 para que se coloque en la posición MoverACintaPrincipal. Cuando reciba esta señal, la aguja C2 mandará otra a la aguja C3 para que se coloque en la posición MoverACintaSecundaria. Solo de esta manera el palé podrá hacer el recorrido mostrado en la figura 9:

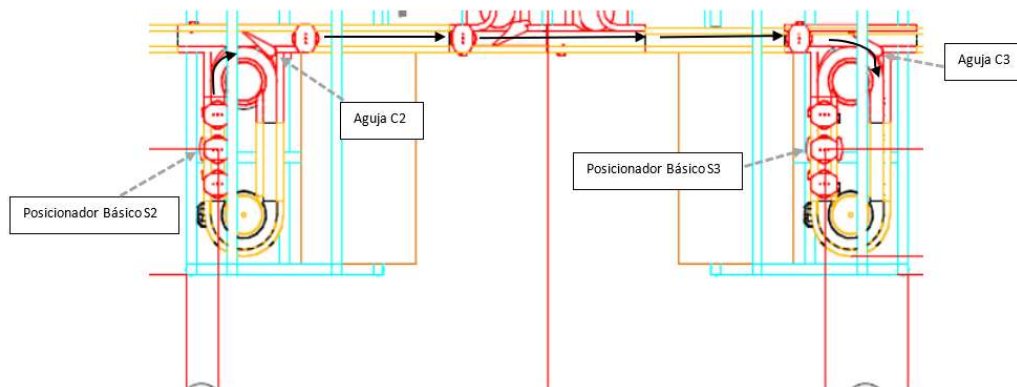


Figura 9: Recorrido del palé desde el posicionador básico S2 al S3

En el GRAFCET del posicionador básico, el mensaje a la aguja C2 es enviado en la etapa EsperarSeñal. La salida de esta etapa se producirá cuando C2 se coloque en posición y avise al posicionador mediante la puesta a 1 de la variable iAgujaEnPosiciónMoverAPrincipal.

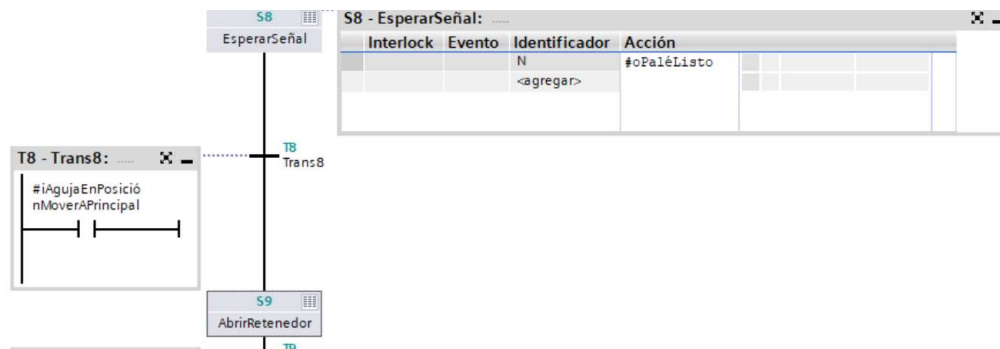


Figura 10: Comunicación entre el posicionador y las agujas

Recibido el mensaje, el posicionador abre la barrera de salida para liberar el palé.

Tras abrir la barrera, el GRAFCET vuelve a su etapa inicial donde esperará la llegada de un nuevo palé.

#### 3.1.2.4. Fin del estado Execute en el posicionador básico S2

En el modo de producción que se ha diseñado en este proyecto, una vez que se han procesado un determinado número de palés, se produce una parada de fin de ciclo. Este número puede ser modificado por el operador en cualquier momento.

Como habrá podido observar en la figura 1, el posicionador básico S2 es el principio y el fin del recorrido del palé, de manera que se necesita programar un código para que el posicionador sepa si el producto que transporta el palé de su entrada necesita ser procesado o si está terminado ya. Para ello se ha implementado un contador que lleva la cuenta de los palés que salen del posicionador y compara este número con el introducido por el operador para la parada de fin de ciclo. Cuando se llegue a este número se pondrá a 1 la variable iÚltimoPalé. Desde ese momento, el posicionador, aunque esté libre, no dejará pasar al palé que hay a su entrada porque sabe que es un palé que ya se ha procesado.

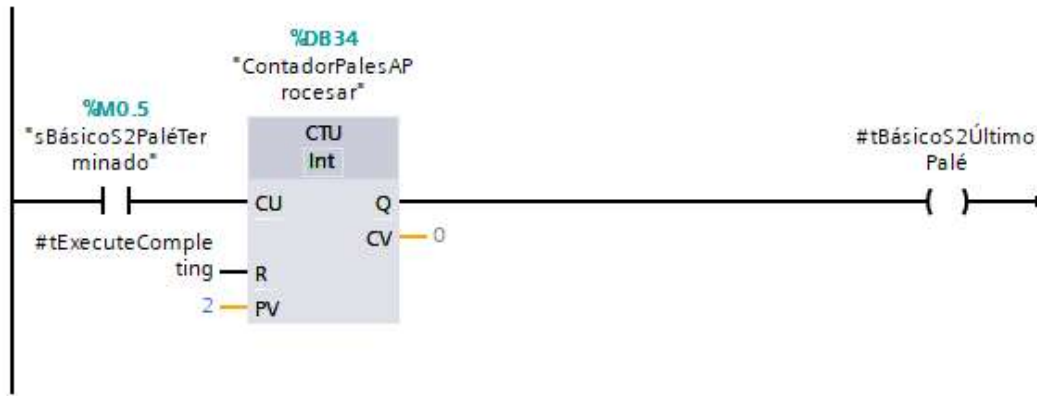


Figura 11: Contador de palés procesados.

El contador de la figura 11 incrementará su cuenta cada vez que se produzca un flanco positivo en su entrada CU. A esta entrada está conectada la variable sBásicoS2PaléTerminado, que guarda el valor de una variable de salida del posicionador básico, tal y como se muestra:



Figura 12: La señal se guarda en una variable temporal

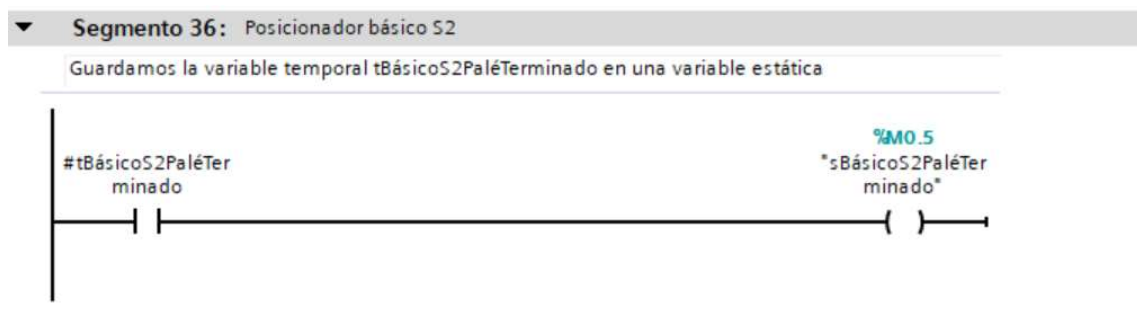


Figura 13: La variable temporal se guarda en una variable estática

Dentro del GRAFCET, la salida oPaléTerminado se activa al salir de su última etapa:

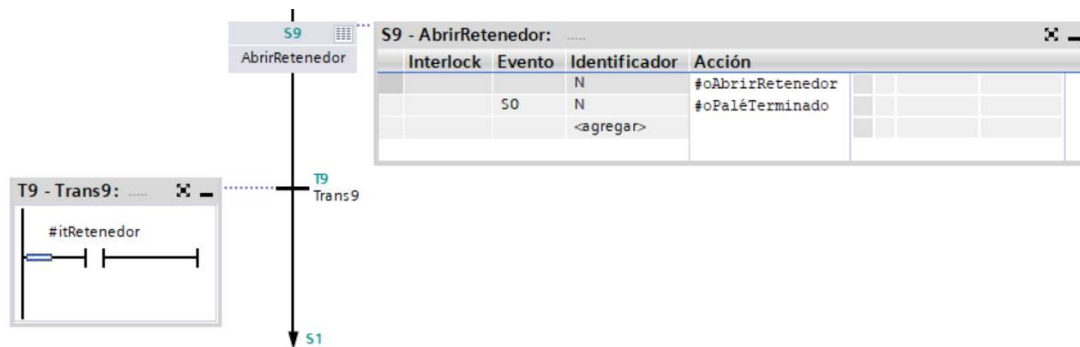


Figura 14: Etapa AbrirRetenedor

Cuando la cuenta se haya terminado, se pondrá a 1 la variable temporal tBásicoS2ÚltimoPalé, esta vez conectada a la variable de entrada iUltimoPalé del posicionador.

Al pasar a la parada de fin de ciclo, se reiniciará el contador.

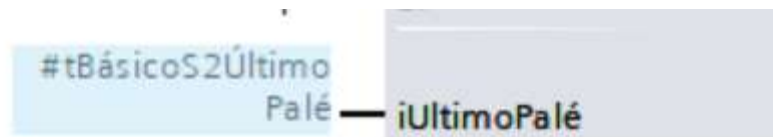


Figura 15: Variable iUltimoPalé

En este caso no se guarda la variable temporal en una variable estática porque la comunicación se puede hacer en un solo ciclo de ejecución del programa principal.

### 3.1.3. Posicionador básico S3

El funcionamiento del posicionador básico S3 es idéntico al del S2, con la excepción de que no se necesita distinguir entre palés a la entrada y la comunicación es con la aguja C3. Esta se debe colocar en la posición MoverACintaPrincipal para que el palé pueda realizar el recorrido mostrado en la figura 16:

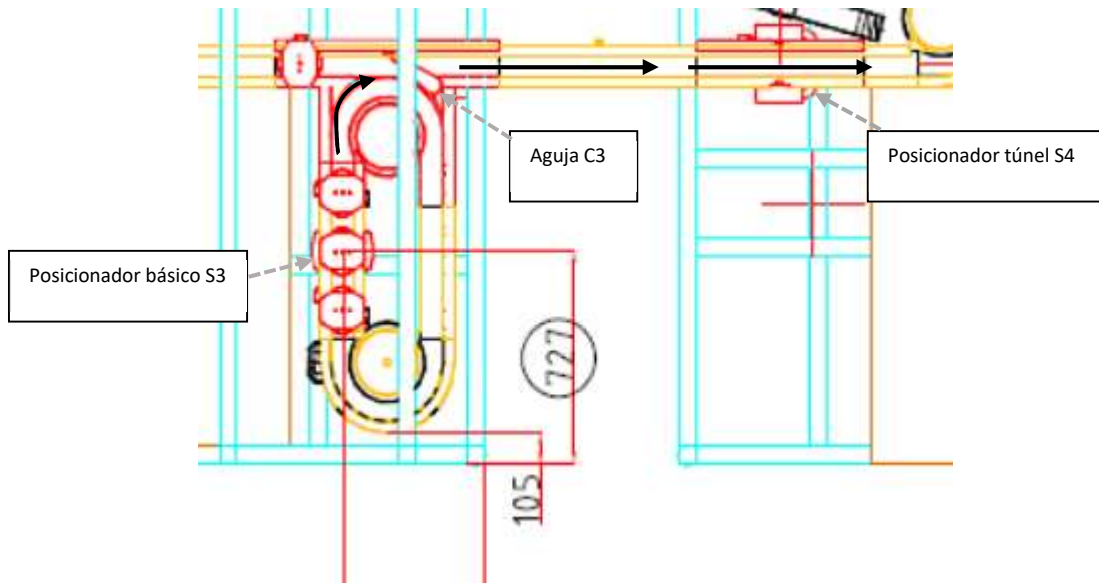


Figura 16: Recorrido del palé desde el posicionador básico S3 al posicionador túnel S4

#### 3.1.4. Posicionador túnel S4

El posicionador túnel S4 es la siguiente etapa en el recorrido del palé. Se avanzará rápidamente a través de la explicación de las etapas puesto que su funcionamiento es similar al del posicionador básico.

Si se detecta un palé a su entrada y no hay un palé procesándose en su interior, el posicionador cerrará la barrera de salida. Una vez que se haya cerrado, se abrirá la barrera de entrada. Como en el básico, la apertura y cierre de las barreras está controlada mediante temporizadores para que no se sobrecalienten las válvulas.

A continuación, se espera a que llegue el palé al posicionador, momento en el que es detectado mediante el sensor situado en su interior. Cuando se detecte, se cierra la barrera de entrada.

A partir de aquí, comienza las diferencias con respecto al posicionador básico. La manera de retener el palé y que no se mueva mientras está actuando el robot, consiste en elevarlo de la cinta. Este elevador puede ser peligroso para el operador si tiene el cuerpo apoyado sobre la cinta en el momento en el que sube o baja. Por ello, en el programa, una vez que se ha cerrado la barrera de entrada, se deja un tiempo antes de subir el elevador para que el operador pueda retirarse un poco de la cinta. Durante este tiempo se enciende un piloto para indicarle que el elevador está en marcha.

Transcurrido este tiempo, el elevador empieza a subir. En este caso, el posicionador lleva incorporados finales de carrera. En cuanto el final de carrera lo indique, el elevador se parará y en ese momento comienza a operar el robot:

La acción del robot se simula con un temporizador. En cuanto ha terminado, y tras esperar un tiempo, el elevador comienza a bajar y lo hará hasta que un final de carrera le indique que ya ha llegado a la posición más baja.

Por último, se abre la barrera de la salida para liberar al palé y el GRAFCET vuelve a ejecutarse desde el principio. De esa manera, si hay un palé a la entrada, el proceso comienza otra vez.

### 3.1.5. Posicionador túnel S1:

Del posicionador túnel S4, el palé recorre toda la minifábrica hasta llegar al posicionador túnel S1. El funcionamiento del S1 es similar al del S4, y solo se diferencia en dos aspectos:

- 1) Cuando el posicionador túnel S1 termina de procesar un palé, debe avisar a la aguja C2 para que ésta se coloque en posición de MoverACintaSecundaria.
- 2) El posicionador túnel S1 es la última etapa en el recorrido del palé. Se van a contabilizar los palés a su salida para, llegado el número indicado por el operador, preparar el sistema para la parada de fin de ciclo.

#### 3.1.5.1. Programación de la parada de fin de ciclo

El contador de palés terminados en el posicionador túnel S1 es el siguiente:

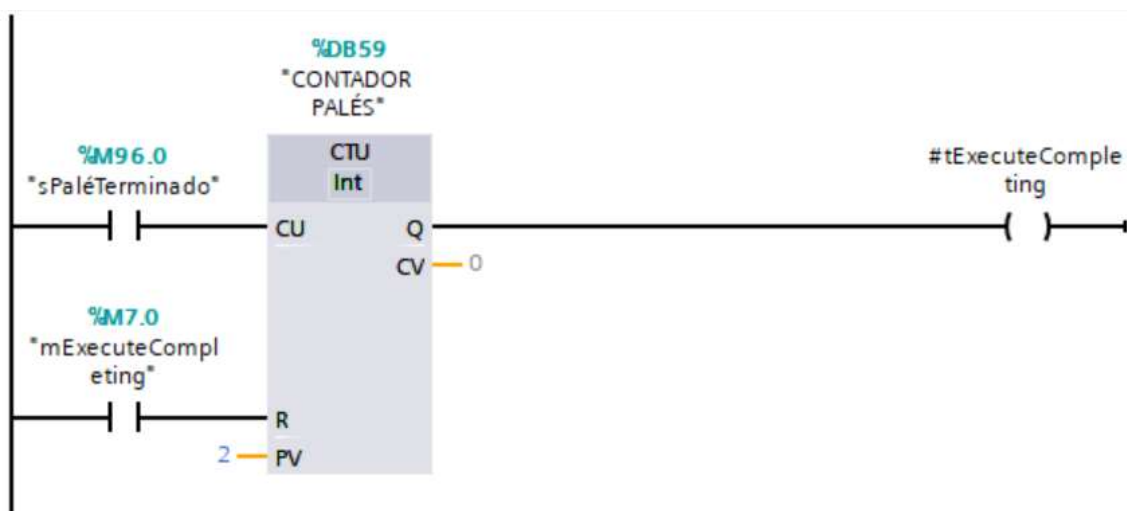


Figura 17: Contador de palés terminados en el posicionador túnel S1.

El contador se incrementa con un flanco positivo en la variable sPaléTerminado. Esto se produce cada vez que el posicionador S1 abre la barrera de salida.

Una vez que el contador ha terminado, se pondrá a 1 la variable temporal tExecuteCompleting.

Esta variable se encarga de deshabilitar las etapas de todos los GRAFCETs del estado Execute (poniendo a 1 la variable OFF\_SQ) para posteriormente guardar su valor en una variable estática: mExecuteCompleting. Esta variable reinicia el contador y activa el comando SC para pasar al estado Completing (Completando).



Figura 18: Se guarda el valor de tExecuteCompleting

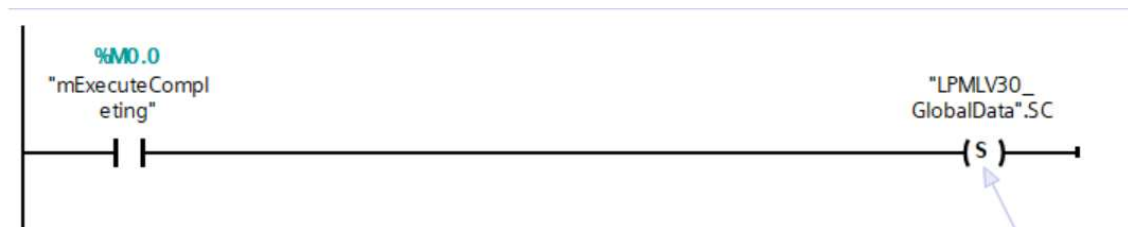


Figura 19: Activación de la señal SC

### 3.1.6. Aguja Combine C2:

El funcionamiento de las agujas Combine consiste en colocarse en una posición que permita al palé desplazarse de la cinta principal a la secundaria, o viceversa. En el caso de la aguja C2, la posición en la que se va a colocar depende de si hay un palé listo en el posicionador túnel S1 o en el básico S2.

No se puede mover la aguja si ya está colocada en la posición correcta, o se sobrecalentaría la válvula. Por ello, se han programado unas variables estáticas para conocer su posición en cada instante y solo moverlas si es necesario.

Las agujas tienen dos barreras y dos sensores, unos en la cinta principal y otros en la satélite. En cuanto uno de los palés esté preparado y la aguja esté colocada, se abrirá una de las cintas

(principal o secundaria dependiendo del posicionador) para dejar pasar el palé, permaneciendo cerrada la otra barrera por seguridad. En cuanto el sensor de la barrera detecte un palé, se esperará un tiempo y después se vuelve a cerrar la barrera.

Después, se espera un tiempo a que el palé haya cruzado de la cinta principal a la secundaria (o al revés) antes de volver a la etapa inicial. Esto se hace para que la aguja no se cambie de posición antes de tiempo, cortando el paso al palé.

Por último, este dispositivo debe comunicarse con los dos posicionadores y con la aguja C3.

Todo ello se programa mediante un GRAFCET, descrito a continuación:

### 3.1.6.1. Uniones

Las uniones entre etapas en el GRAFCET de las agujas pueden ser simples o de selección. Las de selección se utilizan, entre otras cosas, para comprobar qué posicionador tiene un palé listo, o para ver en qué posición se encuentra la aguja.

Para el primer caso, la unión de selección es la siguiente:

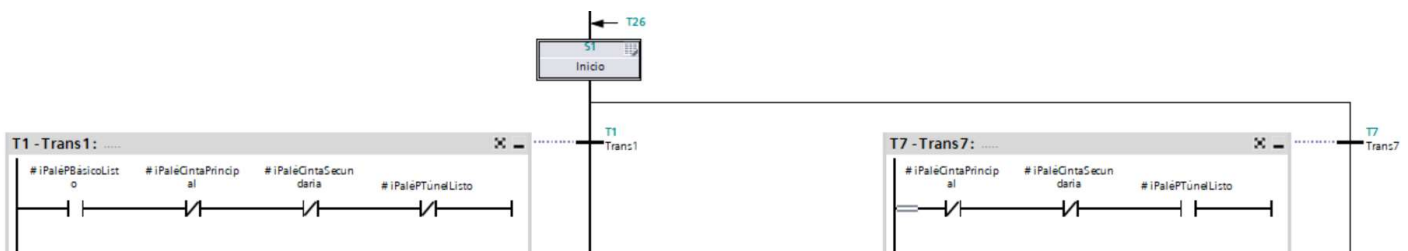


Figura 20: primera unión de selección en la programación de la aguja C2

Como se puede comprobar en la imagen, en el caso de que haya palés terminados en ambos posicionadores, tendrá preferencia el túnel. Además, no se pasará a la siguiente etapa si alguno de los sensores de la aguja detecta un palé.

Las otras dos uniones de selección se utilizan para comprobar la posición de la aguja y en las comunicaciones con la aguja C3. Se explican con más detalle en los siguientes apartados.

### 3.1.6.2. Control de posición de la aguja

Como se ha explicado en la introducción, la posición de las agujas Combine se controla mediante variables estáticas. Dichas variables son sAgujaMoverACPrincipal y sAgujaMoverACSecundaria, cuyos valores son:

Aguja en MoverACintaPrincipal:

- sAgujaMoverACPrincipal =1
- sAgujaMoverACSecundaria =0

Aguja en MoverACintaSecundaria:

- sAgujaMoverACPrincipal =0
- sAgujaMoverACSecundaria =1

Se les asignan estos valores mediante diagramas de contactos con bobinas Set y Reset:

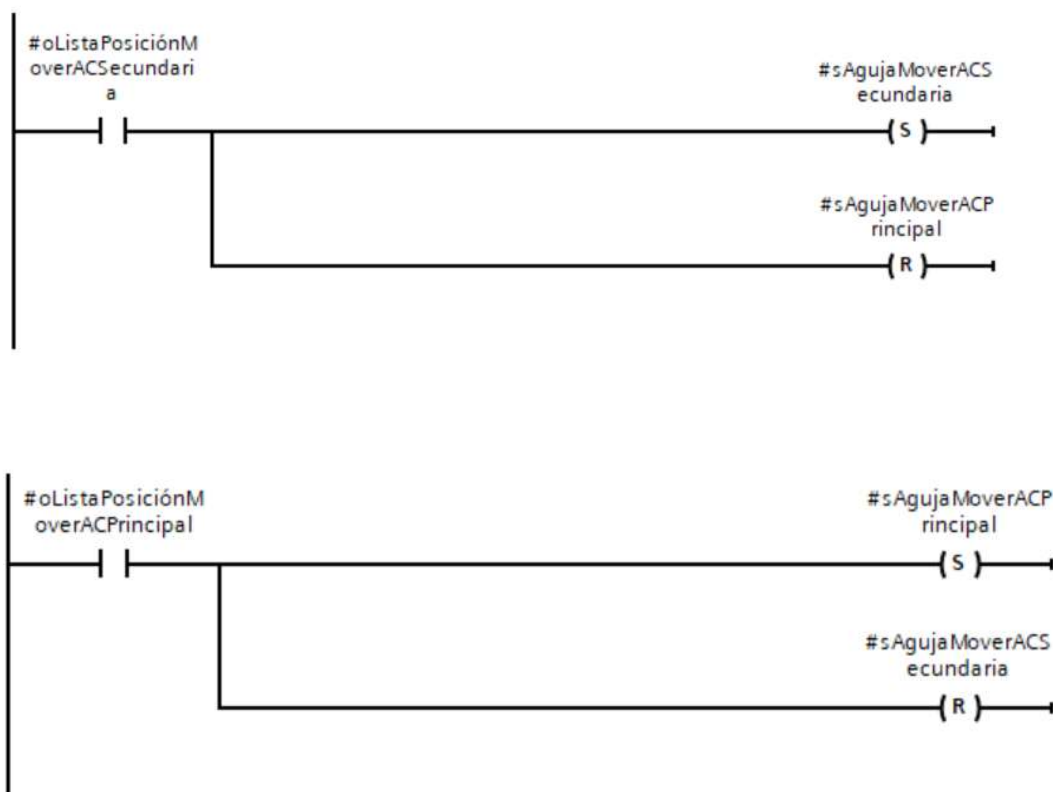


Figura 21: Registro de la posición de la aguja C2

Las variables `oListaPosición...` permanecen activas solo durante una etapa del GRAFCET, pero las estáticas mantienen su valor hasta que la aguja vuelva a cambiar de posición.

Y en el GRAFCET, si la aguja no está colocada en la posición correcta, la aguja se moverá, y en el caso en el que lo esté, se abrirá la barrera directamente:

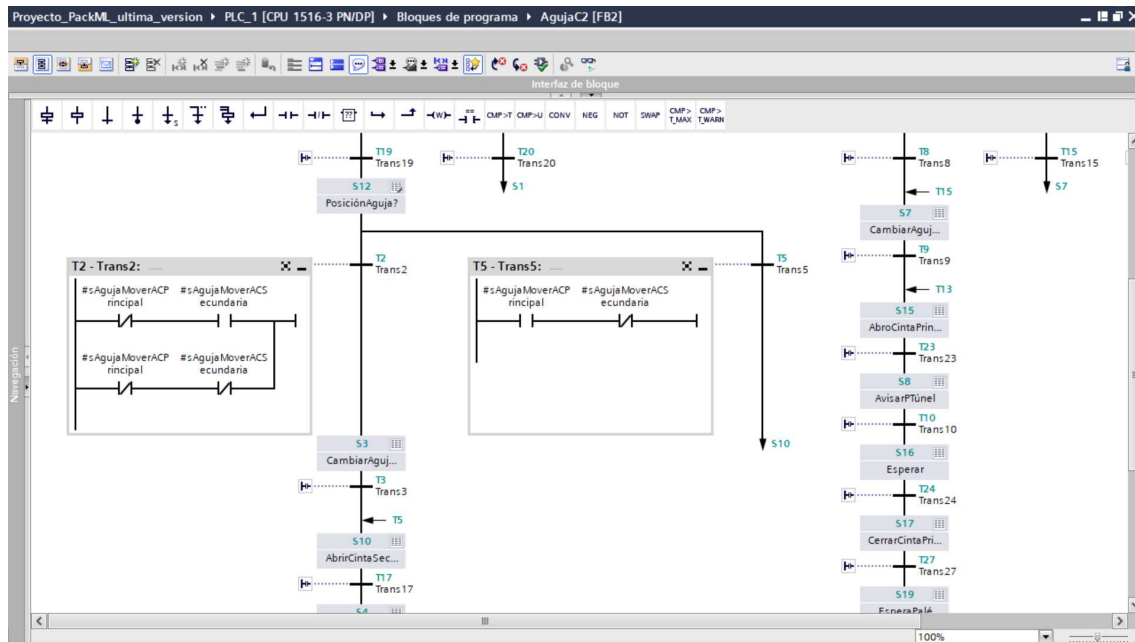


Figura 22: Unión de selección tras la etapa PosiciónAguja?

La situación en la que las dos variables estáticas valen 0 ocurre cuando se enciende el plc por primera vez.

### 3.1.6.3. Comunicaciones con otros dispositivos

La aguja se comunica con los posicionadores. Esta comunicación es bidireccional ya que éstos la informan de si tienen un palé preparado y la aguja, una vez colocada, les indican que pueden abrir su barrera de salida para dejar salir al palé.

Por otro lado, en el caso de que haya un palé listo en el posicionador básico S2, es necesario también la intervención de la aguja C3 para que el palé pueda pasar de una cinta satélite de la estación 2 a la de la estación 3. Por ello, se avisa a la aguja C3 de que se tiene que colocar en la posición MoverACintaSecundaria.

En caso de que la aguja C3 esté lista, el GRAFCET pasará a la siguiente etapa. Si no lo está todavía, el GRAFCET volverá a la etapa S1 o etapa inicial. El motivo de esto es agilizar la producción, ya que en ese momento puede haber un palé en el posicionador túnel preparado para desplazarse y para lo cual no se necesita mandar ninguna señal a C3.

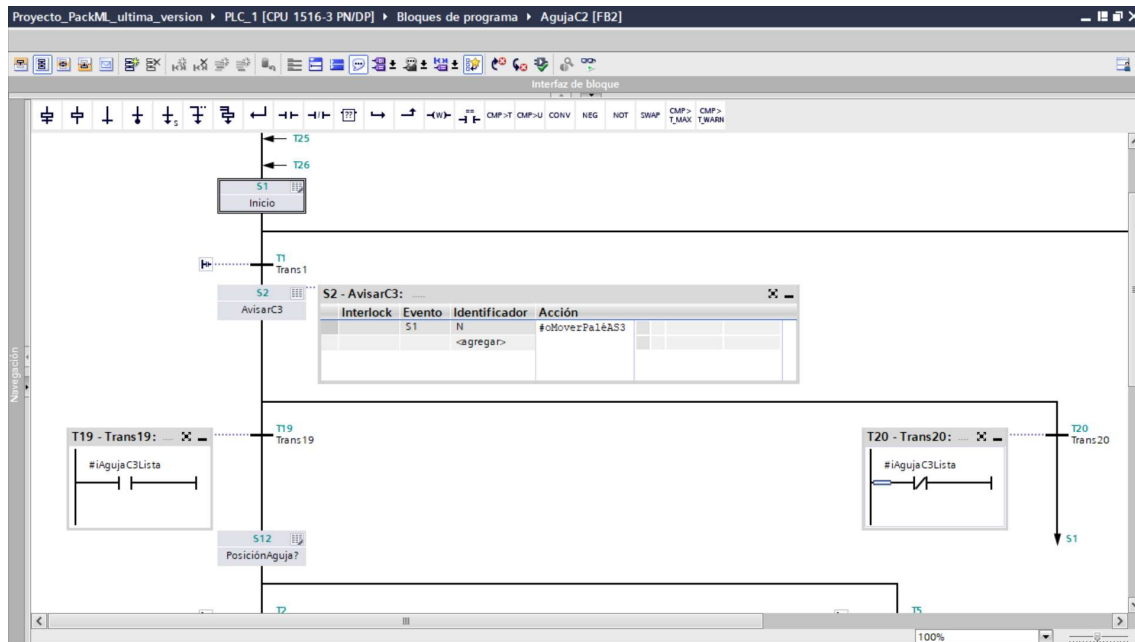


Figura 23: Unión de selección después de la etapa AvisarC3

### 3.1.7. Aguja Combine C3:

Como con la aguja C2, el GRAFCET evolucionará a etapas distintas en función de si hay un palé listo en un posicionador o en otro. En este caso, pueden ser el posicionador básico S2 o el S3. En el caso del S2, la aguja se debe colocar en una posición que permita desplazar un palé de la cinta principal a la satélite. En el caso del S3, de la satélite a la principal.

Como en la aguja anterior, independientemente de si hay un palé listo en uno de los posicionadores, no se saldrá de la etapa inicial si alguno de los sensores de la aguja detecta un palé.

Para el control de la posición de la aguja se utilizan variables estáticas, y esta aguja también cuenta con dos barreras y dos sensores, unos en la cinta principal y otros en la satélite. Cuando hay un palé listo en uno de los posicionadores, la aguja abrirá la barrera correspondiente y esperará a que pase el palé para volver a cerrarla.

Aunque el funcionamiento es similar a la aguja Combine de la estación 2, hay ciertas diferencias en su programación que se explicarán a continuación.

#### 3.1.7.1. Cambios de posición de la aguja

En el caso de que haya un palé listo en el posicionador básico S2, la aguja C3 ordena que se vaya enviando el palé antes de haberse cambiado de posición. El motivo de esto es agilizar la producción, aprovechando que la distancia entre las estaciones 2 y la 3 es considerable. Cuanto antes pueda liberar el posicionador básico su palé, antes podrá empezar a procesar otro.

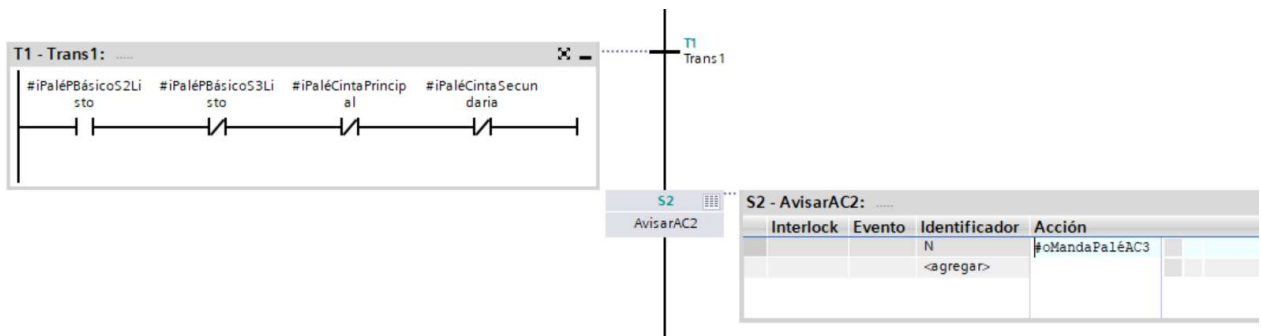


Figura 24: etapa avisar a C2

Por otra parte, si hay un palé listo en el posicionador básico S3 y no lo hay en el S2, la aguja C3 se cambia de posición primero, y después avisa al posicionador. Se ha programado así porque la aguja se encuentra más cerca del posicionador S3 que del S2.

Por último, en el caso de que haya palés preparados a la vez en ambos posicionadores, se siguen los siguientes pasos:

- 1) Se avisa a la aguja C2 y al posicionador S2 de que pueden ir mandando el palé.

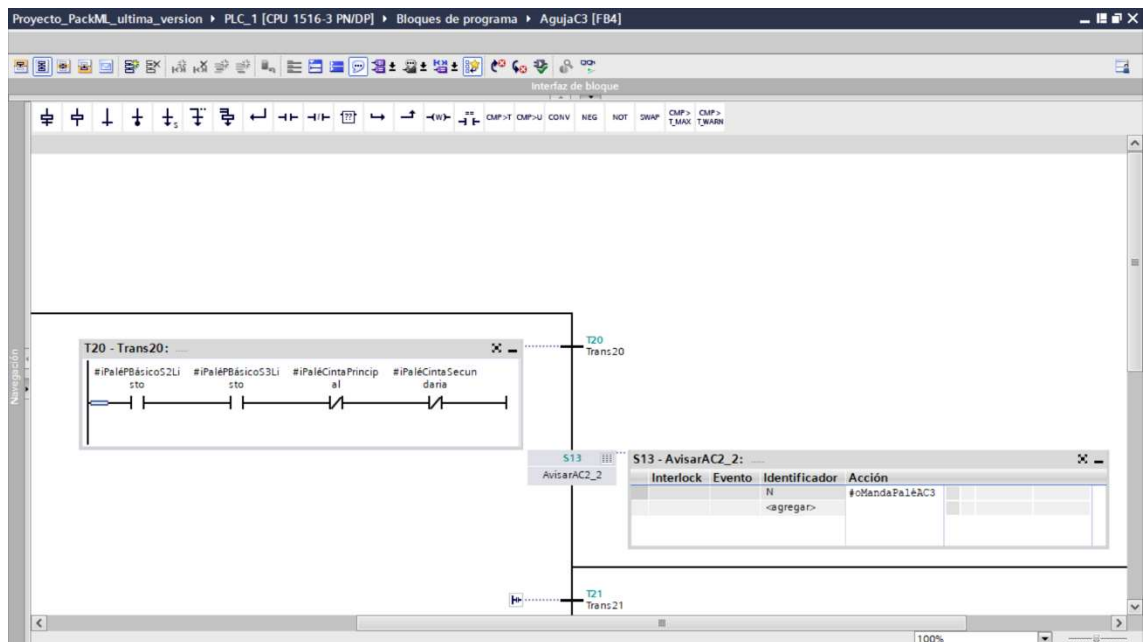


Figura 25: Etapa AvisarAC2\_2

- 2) Se comprueba si la aguja está colocada en la posición MoverACintaPrincipal y, si no, se pone en esa posición.

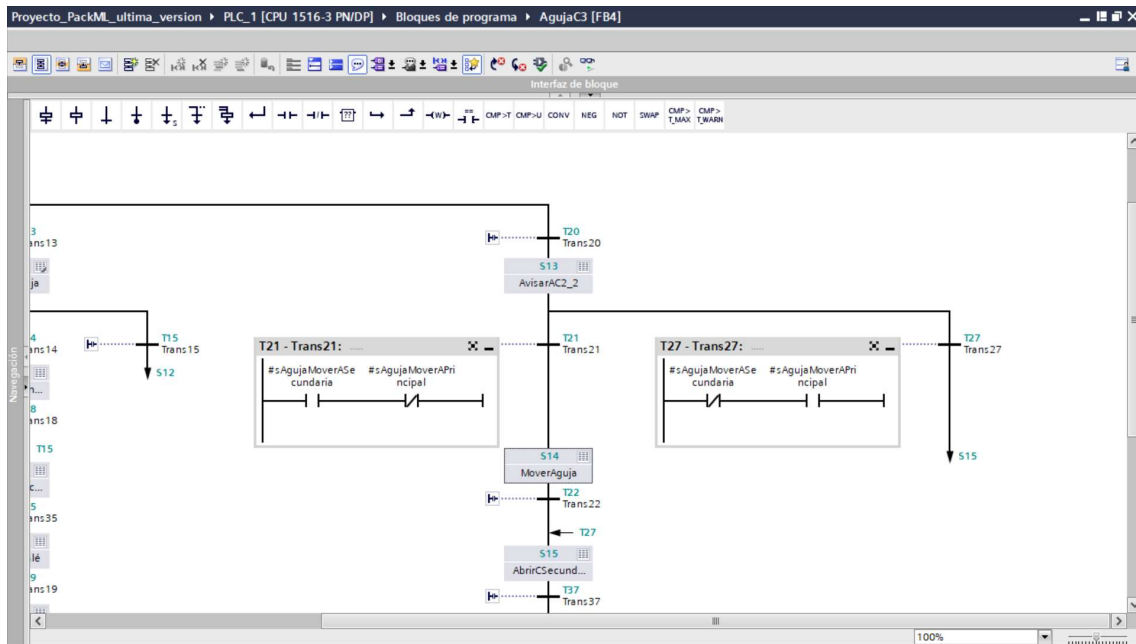


Figura 26: Unión de selección tras la etapa AvisarAC2\_2

- 3) Con la barrera de la cinta principal cerrada por seguridad, se abre la cinta secundaria y se avisa al posicionador básico S3 de que puede ir mandando el palé.

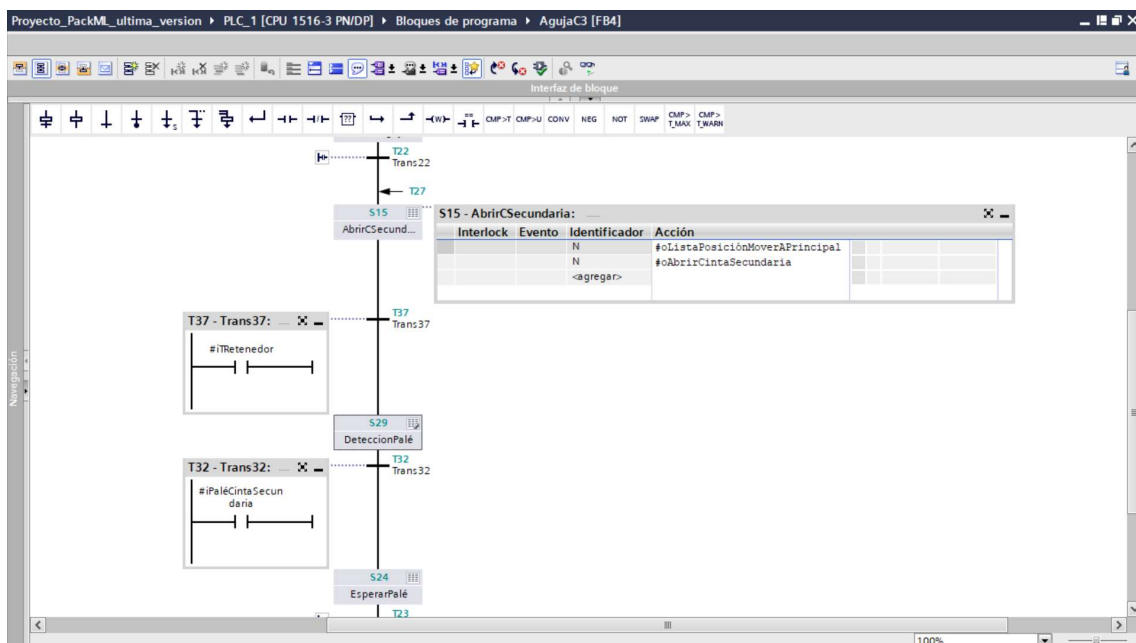


Figura 27: Etapas AbrirCSecundaria y DeteccionPalé

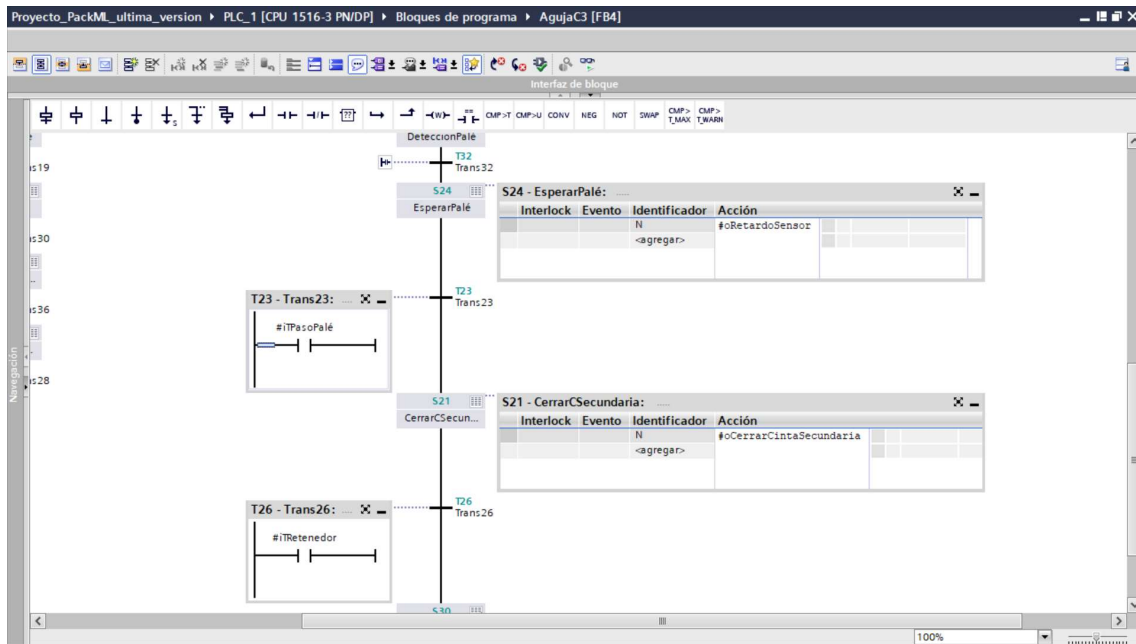


Figura 28: Etapas EsperarPalé y CerrarCSecundaria

- Una vez que el palé haya pasado por el sensor y se le haya dado tiempo para cruzar la barrera y pasar a la cinta principal, se cierra la barrera de la cinta secundaria, se cambia la posición de la aguja y por último se abre la barrera de la cinta principal para dejar pasar al palé procedente del posicionador básico S2.

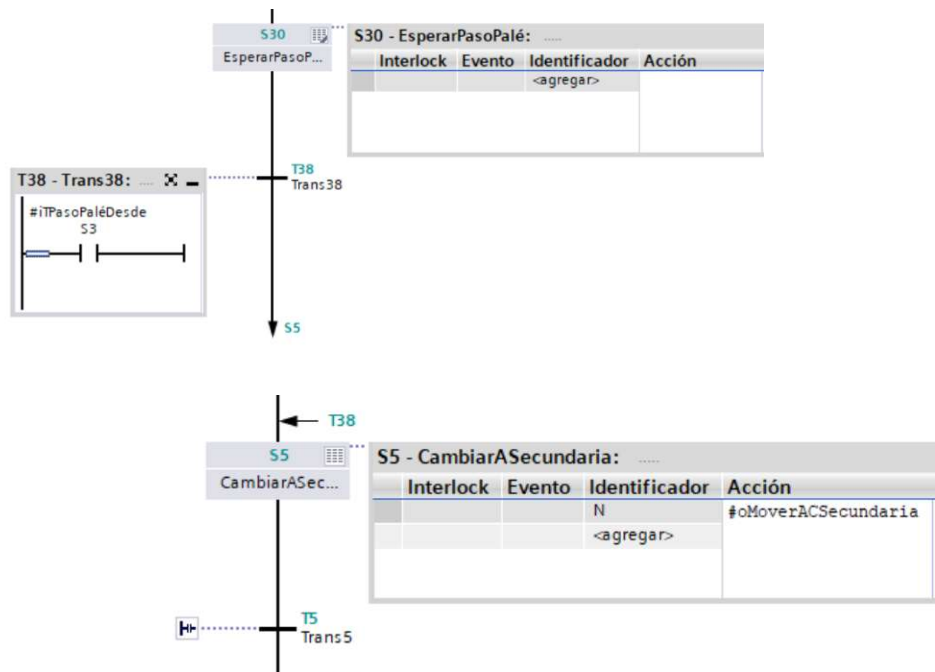


Figura 29: Salto de la etapa EsperarPasoPalé a CambiarASecundaria

- 5) Una vez que el palé ha cruzado la barrera, se cierra y una vez que ha pasado a la cinta satélite el GRAFCET de la aguja vuelve a su etapa inicial.

En resumen, se aprovecha que la estación S2 está algo alejada de la estación S3, para acelerar el proceso de producción. En lugar de que un posicionador tenga que estar esperando con el palé listo a que pase el otro, se envían los dos al mismo tiempo. Sabiendo que siempre va a haber una de las dos barreras cerrada, no hay riesgo de que choquen dos palés. Además, los posicionadores van a poder procesar antes un nuevo palé.

### 3.1.8. Conclusiones

El estado Execute es el estado en el que se va a encontrar la fábrica la mayor parte del tiempo. En su programación, se ha tenido muy presente la seguridad de personas y equipos. Independientemente de ello, se pueden producir fallos no previstos y por ello es necesario y obligatorio por normativa implementar una parada de emergencia, que lleve al proceso de producción a un estado de parada segura en el menor tiempo posible. Esta parada de emergencia se implementa en la plantilla Pack ML dentro del estado Aborting, y se explicará en capítulos posteriores como se ha diseñado este estado.

Por otra parte, en los procesos de producción que no funcionan las 24 horas, es necesario también implementar una parada de fin de ciclo. En el caso de este proyecto, a esta parada se llega cuando se han producido un determinado número de palés. No se ha fijado un número concreto para hacer flexible el código y que pueda adaptarse a diferentes situaciones de producción. La parada de fin de ciclo se implementa en el estado Completing de la plantilla Pack ML. También se explicará en capítulos posteriores cómo se ha programado este estado.

Al hacer el estado Execute, se ha llegado a una serie de conclusiones muy importantes. La primera es que por muy sencillo que un dispositivo parezca, su programación es compleja. Por ello un lenguaje sencillo, visual y organizado en etapas como GRAFCET, permite reducir esta complejidad. En otros lenguajes, resulta difícil para alguien que no sabe de programación entender cómo funciona el código y por lo tanto el sistema automatizado. Todo lo contrario en el caso de GRAFCET, donde con una simple ojeada se puede llegar a la conclusión de cómo funciona el posicionador túnel, el básico o los módulos de combinación/dispersión.

La segunda es que por mucho que se reduzca la complejidad de programar un sistema automatizado con GRAFCET, todavía puede haber errores y fallos que se pasan por alto durante la programación, y por ello es necesario comprobar el código mediante simulación.

Aunque existen alternativas menos laboriosas a la simulación a la hora de comprobar si un código funciona correctamente, como el forzado permanente de variables, se recomienda porque permite la detección de fallos difíciles de ver con el forzado. Para este estado se ha diseñado un sinóptico que ha ayudado a descubrir fallos importantes en el código, tales como el cambio repentino en la posición de la aguja antes de pasar el palé por ella. Sobre cómo se ha programado este sinóptico se explicará en el capítulo siguiente de esta memoria.

### 3.2. Simulación del estado Execute (HMI)

En este capítulo, se describe la programación de la interfaz hombre máquina (HMI) utilizada para el estado Execute. Esta interfaz tiene dos objetivos:

- 1) Detectar fallos en el código
- 2) Demostrar el correcto funcionamiento de este estado.

La interfaz tiene dos sinópticos, uno para el posicionador túnel S1 y otro en el que se muestran las dos agujas C y los dos posicionadores básicos.

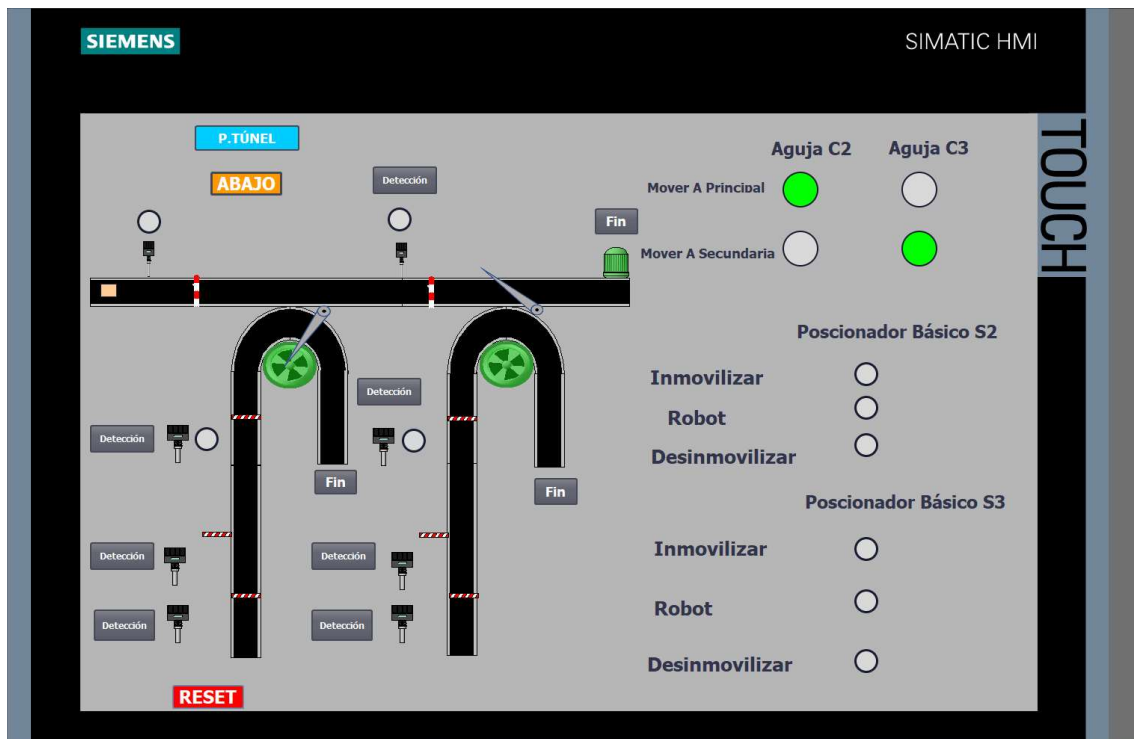


Figura 1: Sinóptico para probar el funcionamiento de agujas y posicionadores básicos

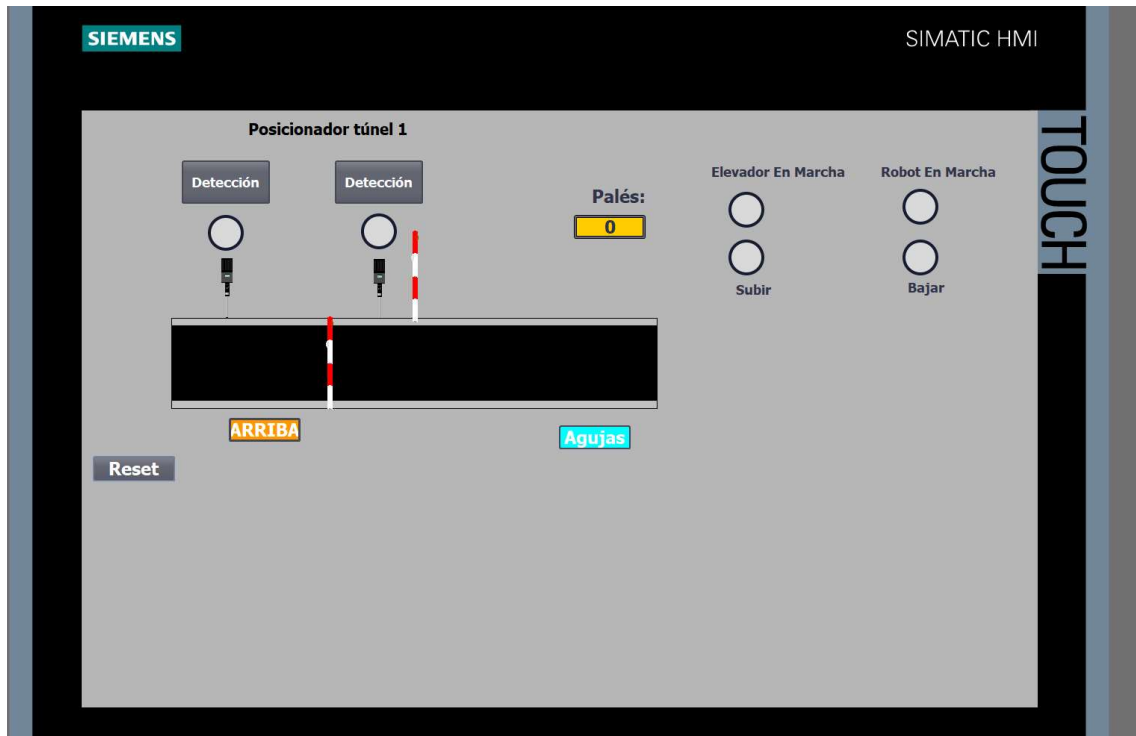


Figura 2: Sinóptico para probar el funcionamiento del posicionador túnel

En ambos se han utilizado elementos gráficos de todo tipo: cintas, motores, agujas, pulsadores, pilotos o luces, barreras, etc. A algunos de estos elementos se les ha puesto animaciones para tratar de recrear el funcionamiento de la fábrica de la manera más exacta y visual posible.

### 3.2.1. Pulsadores

Hay 18 pulsadores en el panel. Los que tienen el rótulo “detección” sirven para simular el funcionamiento de los sensores. Cada pulsador controla una variable conectada a la entrada de un GRAFCET en paralelo con la variable del sensor.

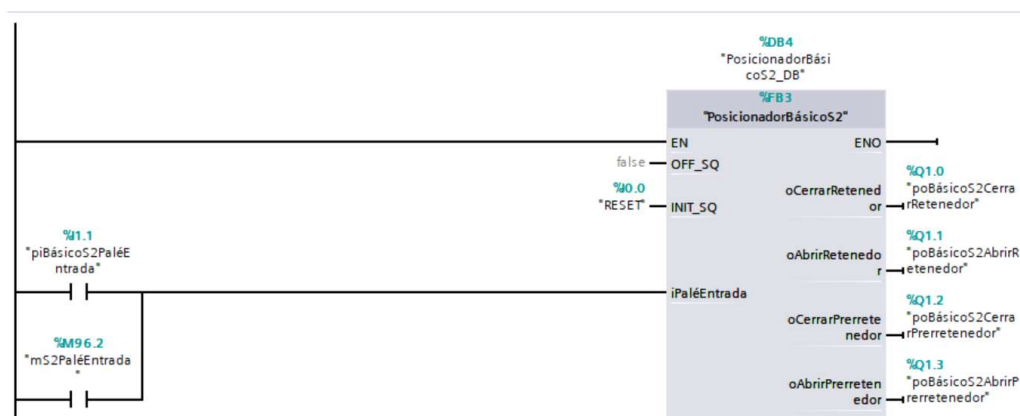


Figura 3: Activación de la variable iPaléEntrada

Tal y como se observa en la figura 3, la variable que informa al posicionador básico S2 de que hay un palé en su entrada puede ser tanto la activada por el sensor piBásicoS2PaléEntrada,

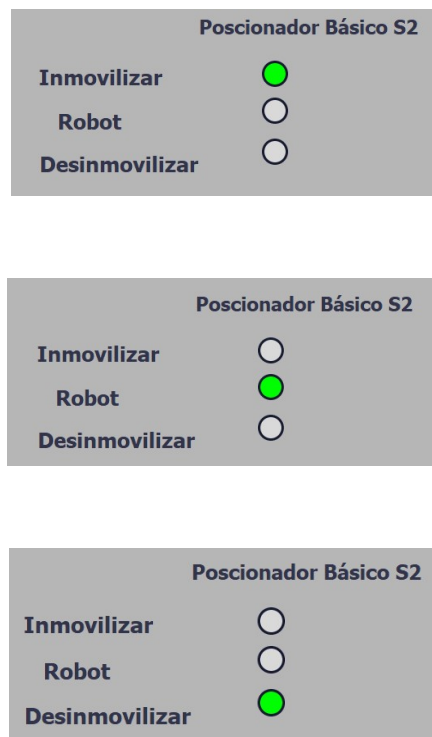
como la activada con el pulsador mS2PaléEntrada. De esa manera se puede simular el funcionamiento del programa sin necesidad de tener un sensor.

También los finales de carrera se simulan mediante pulsadores (son los de color naranja con el rótulo “abajo” en la figura 1 y “arriba” en la figura2).

Además, hay pulsadores para que el usuario del panel pueda pasar de un sinóptico a otro, un reset para reiniciar la simulación y otros que se utilizan en la simulación del movimiento del palé.

### 3.2.2. Pilotos

Los pilotos informan sobre el estado del programa. Cada uno tiene asociada una variable de salida, cambiando de color a verde cuando esta variable vale 1.



*Figura 4: Pilotos que indican la actividad que se está realizando en el posicionador básico S2*





Figura 4: Pilotos en el posicionador túnel.

En la última imagen, el piloto de “elevador en marcha” se ilumina en rojo al ser un indicador de peligro. El resto de los pilotos son verdes porque indican funcionamiento normal [MOND20].

El encendido del piloto de “elevador en marcha” se controla con el siguiente diagrama de contactos:

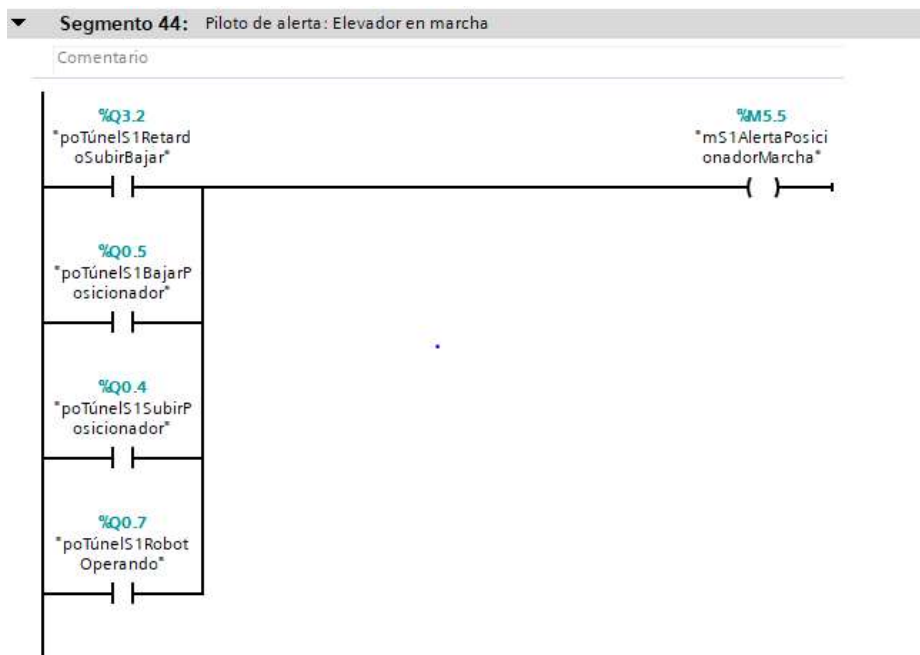


Figura 5: Piloto de elevador en marcha.

### 3.2.3. Contador de Palés

En el sinóptico del posicionador túnel hay un contador que muestra el número de palés que se han procesado hasta el momento:



Figura 6: Contador de Palés

El valor del contador es el de una variable de tipo entero llamada intPalésTerminados conectado a la salida CV de un contador para guardar el valor de su cuenta:

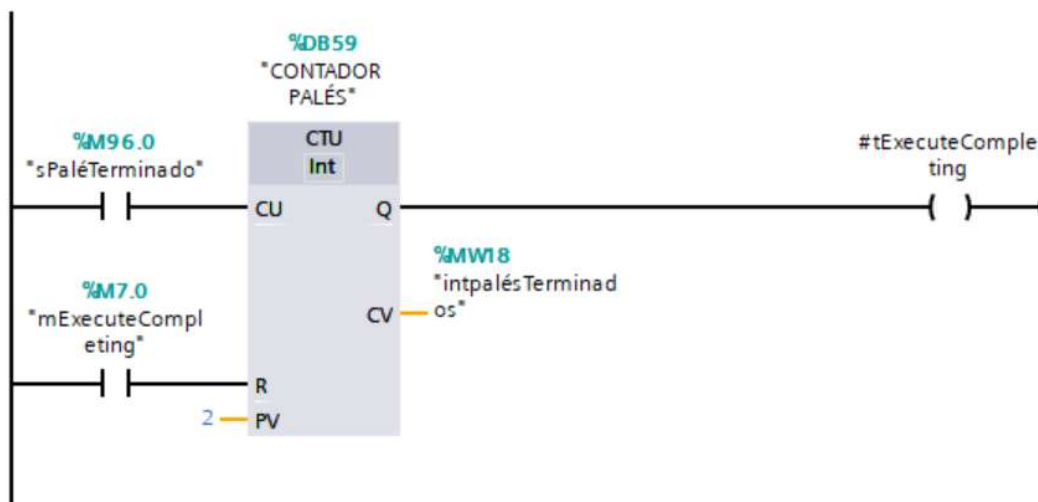


Figura 7: Variable intpalésTerminados

### 3.2.4. Barreras

Para conseguir recrear el movimiento de las barreras, éstas están formadas por dos objetos de la librería de gráficos, colocados en línea el uno del otro.

El funcionamiento se va a explicar utilizando la barrera de entrada del posicionador túnel, pero es idéntico para el resto de las barreras.

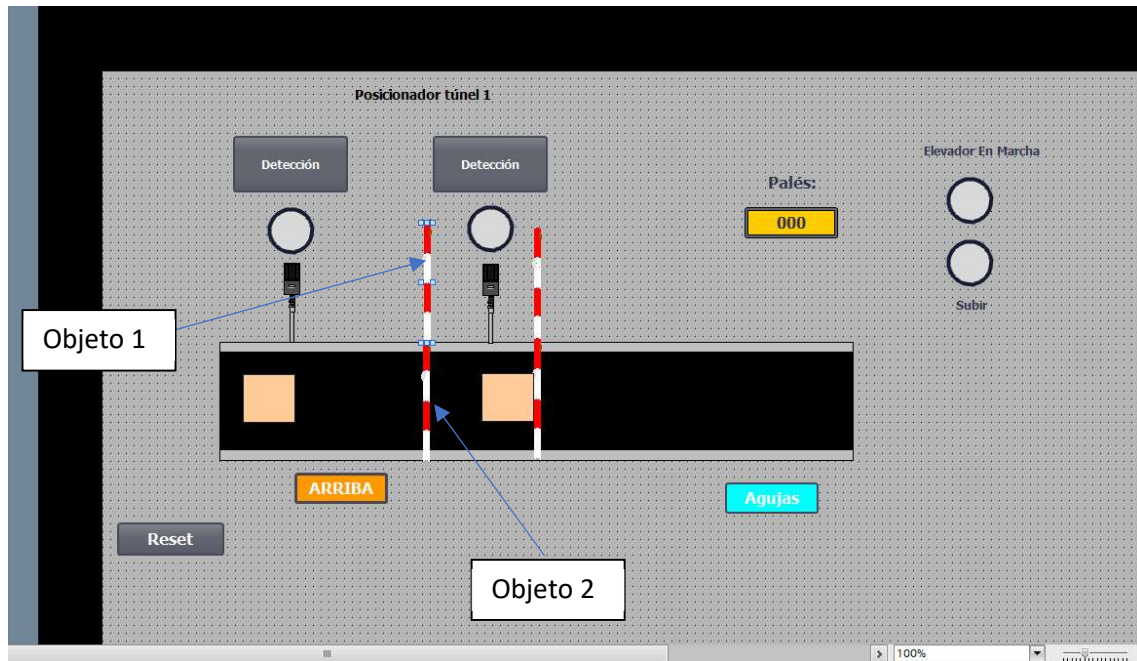


Figura 8: Simulación de las barreras

El objeto 1 es visible durante el cierre de la barrera, momento en el que se mueve verticalmente hacia abajo. El objeto 2 es visible el resto del tiempo y se moverá hacia arriba durante la apertura de la barrera.

Para simular movimiento en un panel de Tia Portal se utiliza una variable de tipo Time. Cuanto mayor sea el rango de valores de esta variable más lento es el movimiento. Así, para simular el cierre de la barrera (movimiento del objeto 1) se utiliza la variable `timeS1CerrarPrerretenedor`, cuyo valor va de 1 a 5000 para que el cierre dure 5 segundos.

`timeS1CerrarPrerretenedor` se conecta a la salida ET de un temporizador tal y como se muestra en la figura 9:

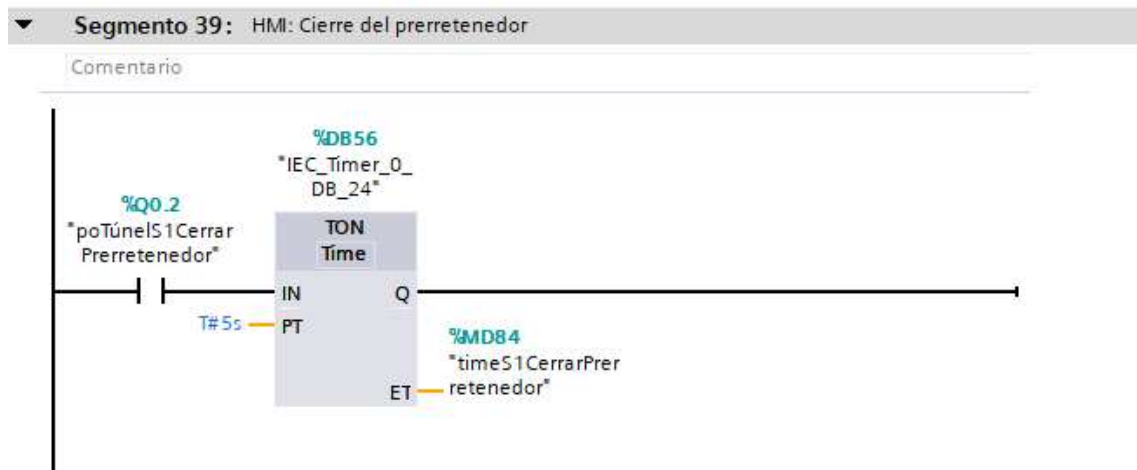


Figura 9: Simulación del cierre de la barrera de entrada

En cuanto se comience a cerrar la barrera, se guardará en la variable la cuenta del temporizador. La apertura se controla con la variable timeS1AperturaPrerretenedor. También dura 5 segundos y su valor lo marca un temporizador.

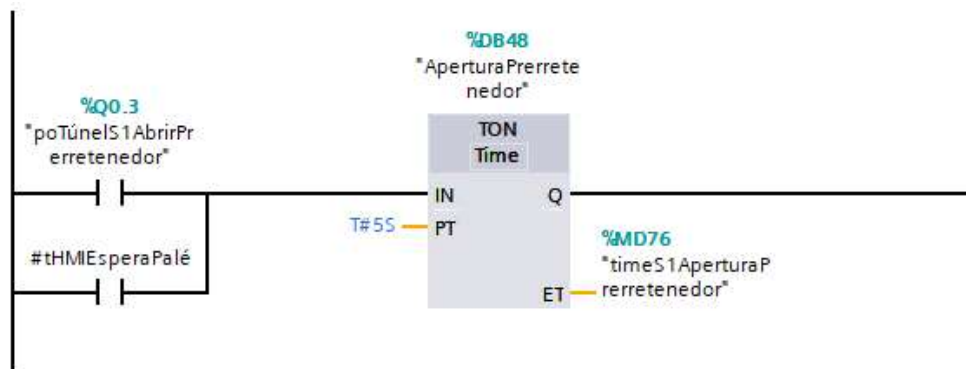
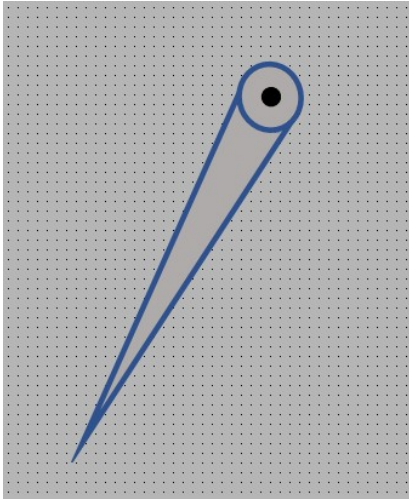


Figura 10: Simulación de la apertura de la barrera de entrada

Un inconveniente es que, si cualquiera de las dos variables se pone a 0, el objeto que se ha desplazado no se mantiene en su posición final, si no que vuelve a la del principio. Por ello, para que el objeto 2 se mantenga en la posición de abierto es necesario mantener la variable timeS1AperturaPrerretenedor a 5000 hasta que la barrera se comience a cerrar y no solo cuando se está abriendo. De ahí que a la entrada se haya colocado también la variable tHMIesperaPalé.

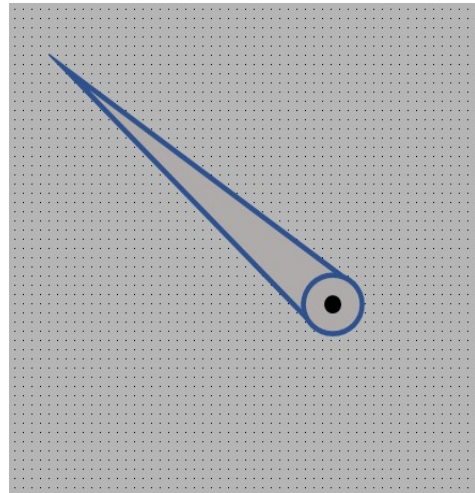
### 3.2.5. Movimiento de las agujas

Para la simulación del movimiento de las agujas, se han utilizado dos objetos realizados con la herramienta PowerPoint:



*Figura 11: Aguja en posición*

*MoverACintaPrincipal*



*Figura 12: Aguja en posición*

*MoverACintaSecundaria*

El objeto de la figura 11 está visible cuando la aguja se encuentre en la posición *MoverACintaPrincipal*. El de la figura 12 cuando esté en *MoverACintaSecundaria*.

### 3.2.6. Simulación del movimiento del palé

Aunque en la simulación se ve un solo palé, para recrear su movimiento es necesario crear varios objetos. Por ejemplo, para representar el recorrido del palé desde la entrada del posicionador básico S2 hasta el S3 se han creado 7 palés en el sinóptico:

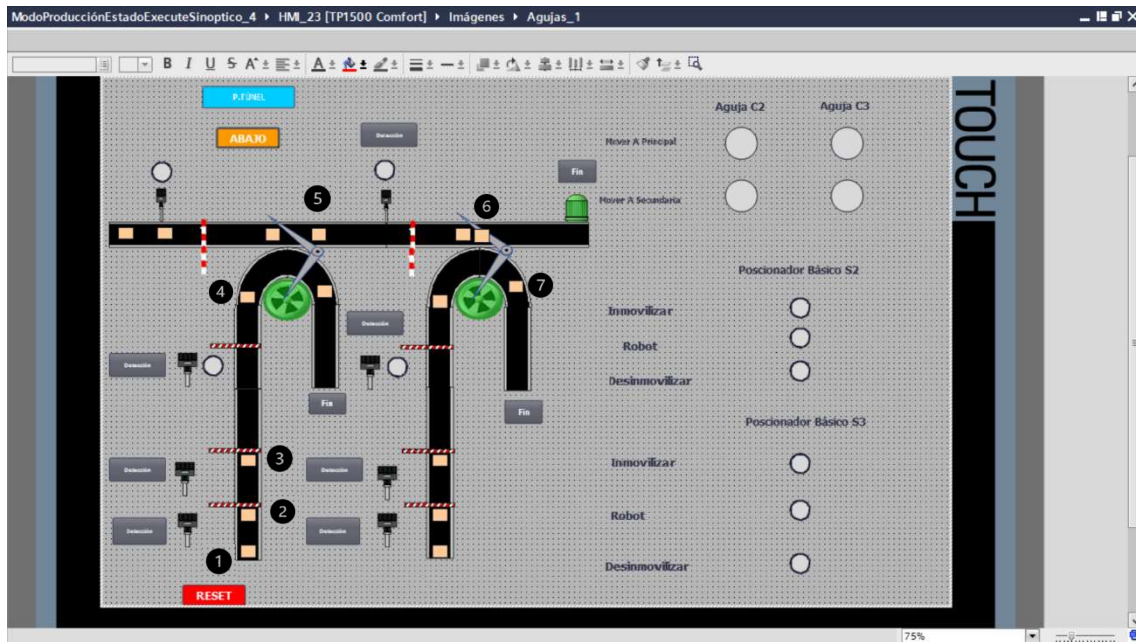


Figura 13: Simulación del movimiento del palé

La visibilidad y el movimiento del palé 1 se controlan con una variable de tipo time llamada timeS2MovimientoPalé1. El valor de esta variable está comprendido entre 1 y 5000. Mientras que sea distinta de 0 el palé va a ser visible en la simulación y de 1 a 3000 se moverá verticalmente hacia arriba.

Esta variable está conectada a la salida ET de un temporizador:

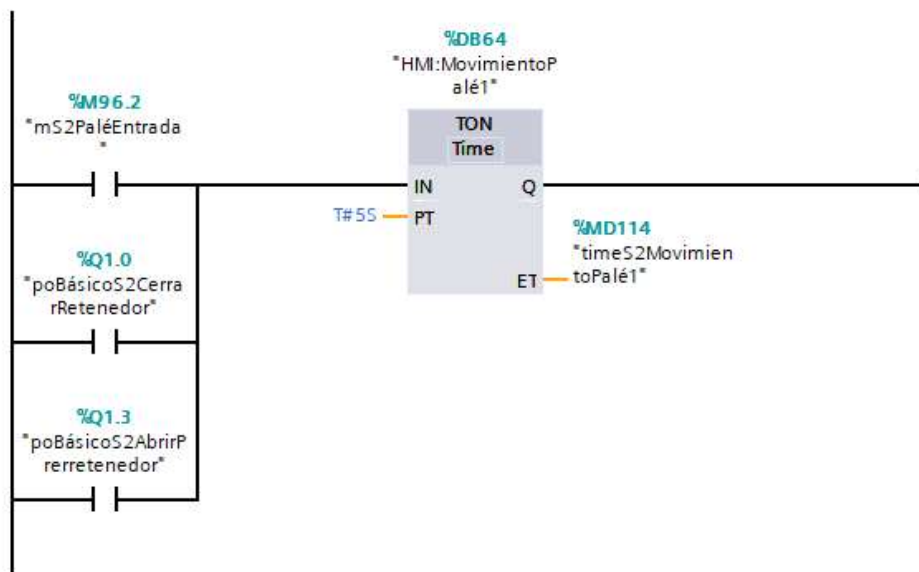


Figura 13: Simulación del primer movimiento del palé

En cuanto se detecte un palé a la entrada del posicionador (detección simulada cuando el usuario pulsa un botón, poniendo a 1 la variable mS2PaléEntrada) el temporizador se activa y aparece un palé en el sinóptico que comienza a moverse hacia la barrera de entrada.

Transcurridos 3 segundos, se detiene delante de la barrera, aunque permanece visible, pues el valor de la variable timeS2MovimientoPalé1 es distinto de 0. Y permanecerá visible hasta que termine de abrirse la barrera de entrada

Después de abrirse la barrera de entrada, el GRAFCET del posicionador entra en una etapa donde se espera la detección de un palé dentro del posicionador. En esta etapa, para la simulación, se pone a 1 una variable estática llamada sHMIesperaPalé. Esta variable es la responsable de iniciar el movimiento y la visibilidad del segundo palé, tal y como se muestra en la siguiente figura:

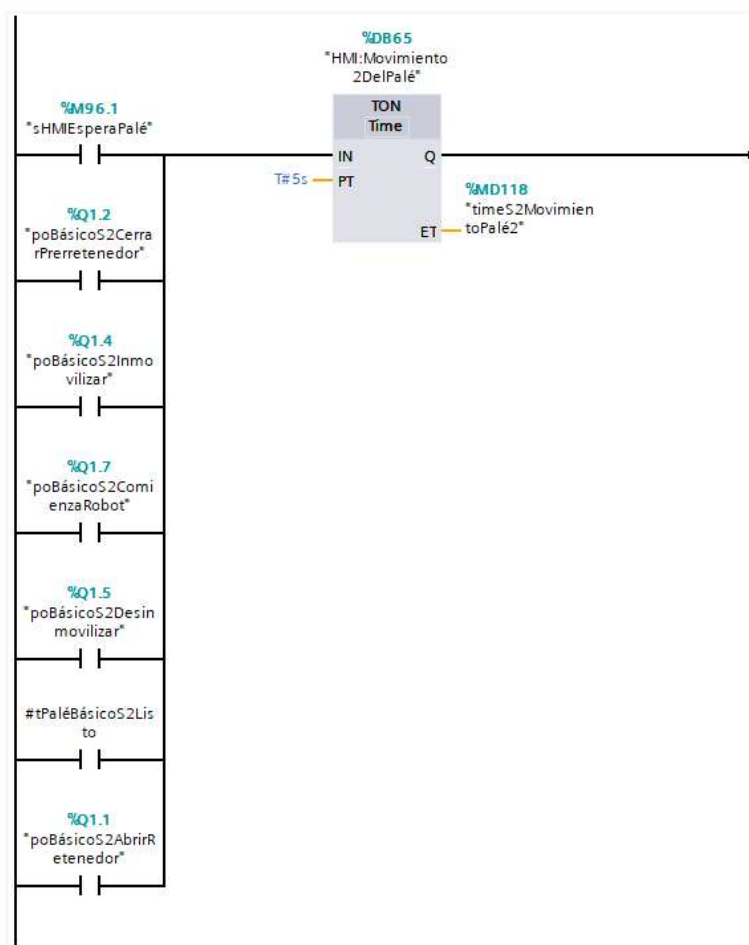


Figura 14: Simulación del segundo movimiento del palé

El segundo palé se moverá hasta chocar con la barrera de salida, y debe permanecer visible hasta la apertura de esta, de ahí que haya tantas entradas en paralelo en el temporizador.

El tercer movimiento del palé se inicia en el instante en el que finaliza su procesado en el posicionador básico. En ese momento, se produce un flanco positivo en la variable temporal tBásicoS2PaléTerminado con el que se pone a 1 la variable mS2MovimientoPalé3:

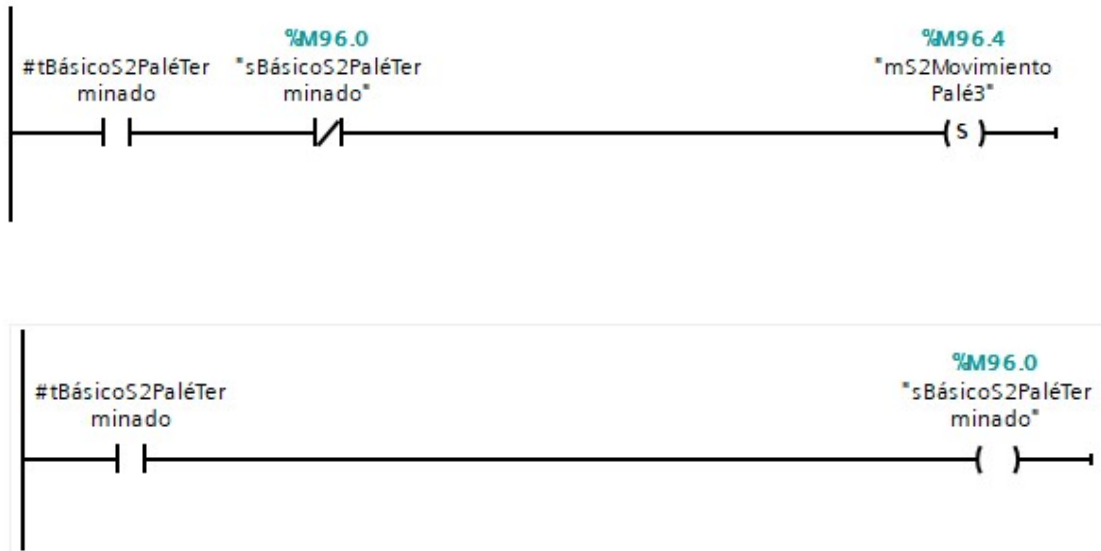


Figura 14: Simulación del tercer movimiento del palé

Como en los dos movimientos anteriores, esta variable activa un temporizador con el que se va a controlar la visibilidad y el movimiento del tercer palé, el cual se va a desplazar hasta llegar a la aguja C2.

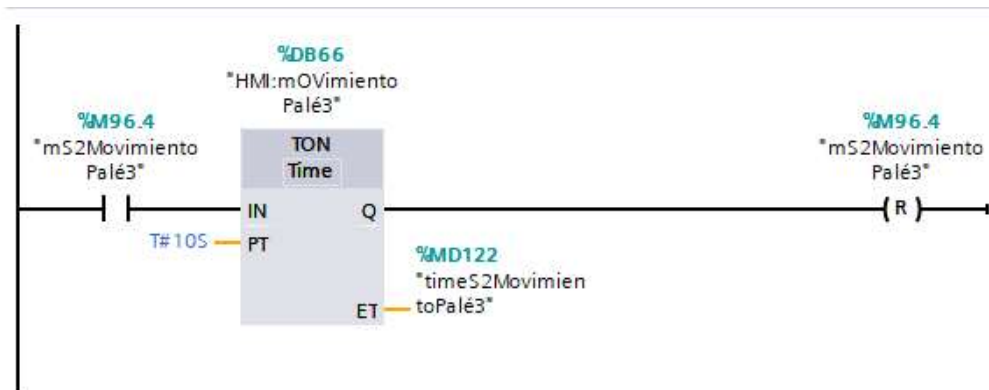


Figura 15: Temporizador

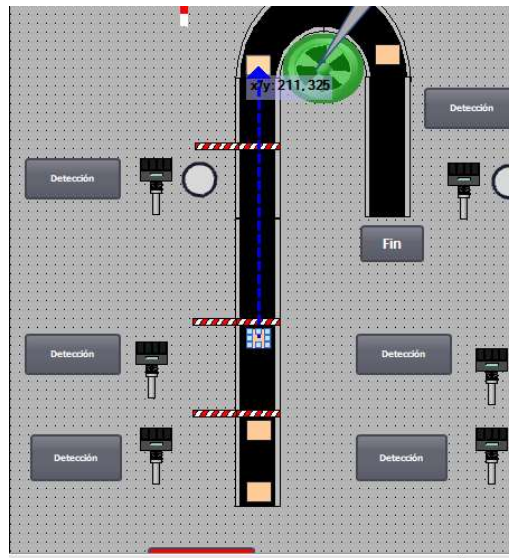


Figura 16: Tercer movimiento del palé

En cuanto el palé pasa por el sensor que tiene la aguja C2 en la cinta secundaria, el usuario pulsa un botón con el que activa la variable mC2PaléCintaSecundaria. En ese momento, se pone a 1 la variable mMovimiento2. Tal y como se ha configurado en el siguiente diagrama de contactos, esta variable no se pondrá a 0 hasta que el palé llegue al posicionador básico S3 (momento en el que se pone a 1 mParaMovimiento2).

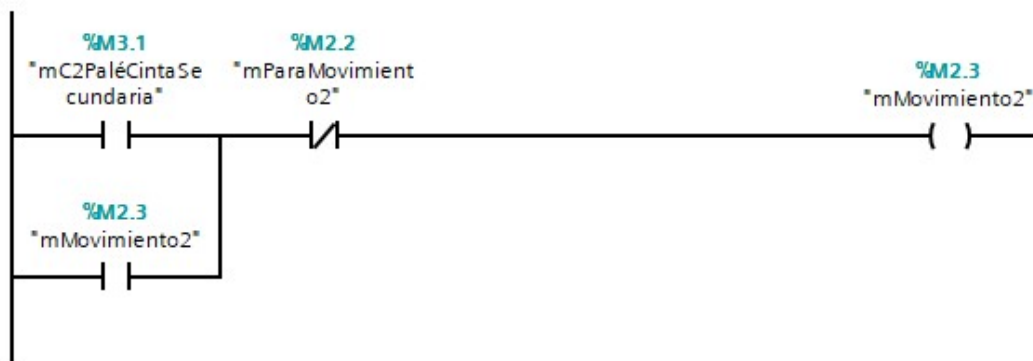


Figura 17: Movimientos de palé siguientes

Con la variable mMovimiento2 a 1 y una vez concluido el desplazamiento anterior del palé (variable mS2MovimientoPalé3 a 0) se arranca un temporizador asociado a la variable timeRecorridoPaléDesdeBásicoS2. Con esta variable de 48 segundos se controlará la visibilidad y el movimiento de los siguientes palets.

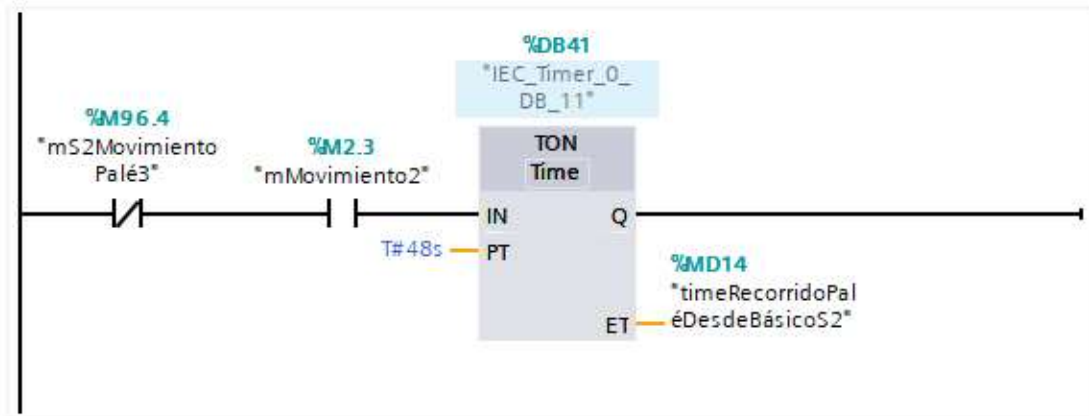


Figura 17: Movimientos de palé siguientes

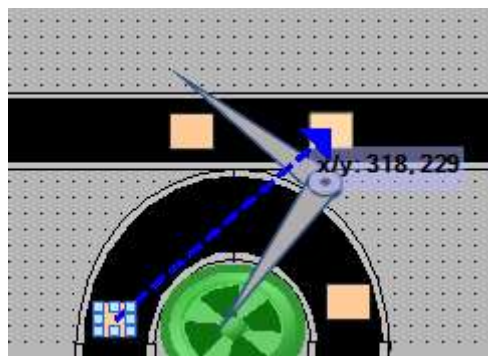


Figura 18: Cuarto movimiento  
(12 segundos)

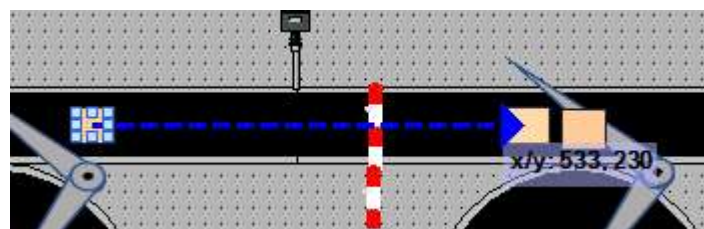
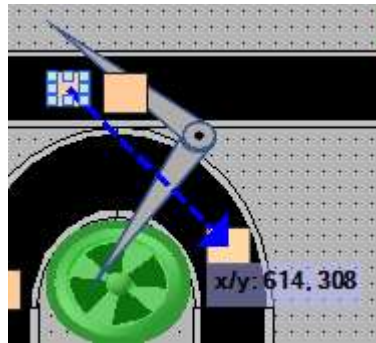
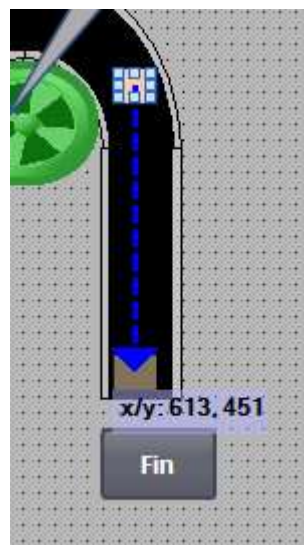


Figura 18: Quinto movimiento  
(12 segundos)



*Figura 19: Sexto movimiento  
(12 segundos)*



*Figura 20: Séptimo movimiento  
(12 segundos)*

Concluido este último, se pulsa el botón de fin para activar la variable mParaMovimiento2.

### 3.2.7. Conclusiones

Como se puede observar, la programación de un sinóptico es compleja y requiere algo de tiempo, pero es recomendable ya que permite la detección de fallos que serían muy difíciles de detectar con otro método.

Por ejemplo, se detectó un fallo importante con la simulación en la programación de las agujas, ya que su GRAFCET, una vez detectado el palé por uno de sus sensores, volvía a la etapa inicial. Al no esperar a que el palé pase por la aguja, ésta podía cambiar de posición demasiado pronto, cerrando el paso al palé.

Se comprobó este problema en la simulación y se añadió una etapa más al GRAFCET.

Por otra parte, la simulación también puede ser útil para mostrar al cliente que el programa funciona correctamente y que está listo para probarse sobre la fábrica real.

### 3.3. Parada de Fin de ciclo

#### 3.3.1. Introducción

A la parada de fin de ciclo se entra cuando ha terminado el proceso de producción en el estado Execute (Ejecución). Se ejecuta entonces un programa para recoger los productos terminados y llevar los palés vacíos a la entrada del posicionador básico S2, dejando al sistema preparado para una nueva puesta en marcha.

Inicialmente, se había pensado que esta tarea la realizara un operario. Pero dada la monotonía y simplicidad de este trabajo, se ha optado por su automatización. Al fin y al cabo, un trabajador es mucho más valioso que una máquina, y por ello se le puede asignar a éste una tarea más productiva.

El encargado de recoger los productos terminados y llevarlos a las cajas o lugares destinados para su transporte va a ser el robot de la estación 2. Al posicionador básico S2 se le va a asignar la misma tarea que en el estado Execute: inmovilizar, esperar a que el robot termine su tarea y dejar de inmovilizar.

Cuando se haya recogido el producto del palé, se colocarán las agujas para llevarlo a la entrada del posicionador túnel S1. Dado que no se utilizan todos los posicionadores, las agujas D se colocarán en la posición “short-cut” (camino corto) para hacer más rápido el proceso.

Finalmente, los palés se colocarán a la entrada del posicionador túnel donde esperarán a que el operador active el comando “reset”. Una vez activado, la aguja C2 cambia de posición, se abren las barreras del posicionador túnel y los palés vacíos se dirigen a la entrada del posicionador básico S2, desde donde esperarán a una nueva puesta en marcha.

La parada de fin de ciclo comprende los estados Completing (Completando), Complete (Completado) y Resetting (Reinicio) de la plantilla Pack ML.

#### 3.3.2. Control de los módulos de dispersión/combinación

Para la correcta colocación de las agujas C y D, se ha programado el GRAFCET CompletingColocarAgujas.

En él se coloca la aguja C2 en una posición que permita mover el palé a la cinta principal, y las agujas D dividiendo en 2 la cinta principal.

##### 3.3.2.1. Posición de la aguja C2

El último movimiento que realiza el palé en el estado Execute es del posicionador túnel S1 al posicionador básico S2. Si todo ha funcionado con normalidad, la aguja C2 debería estar en la posición MoverACintaSecundaria al entrar en el estado Completing. Sin embargo, por seguridad, se van a seguir utilizando en este estado las variables estáticas que informan de la posición de la aguja.

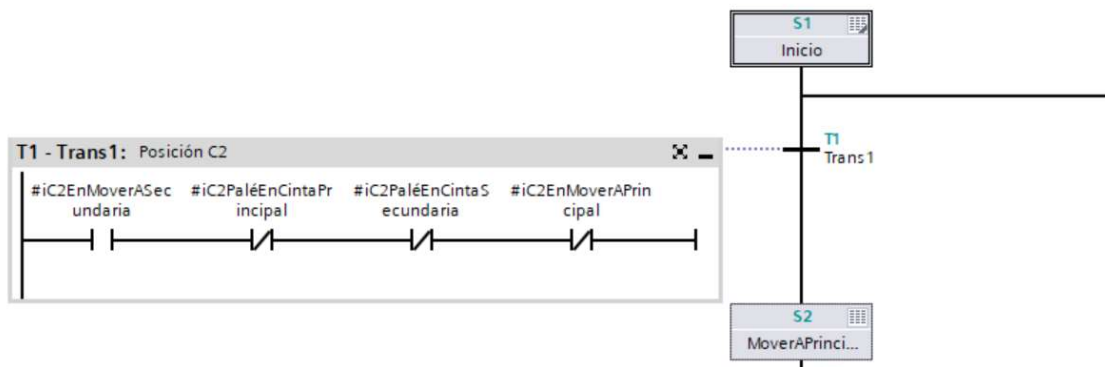


Figura 1: Condición para la transición a la etapa MoverAPrincipal

Si la aguja está en MoverASecundaria, se mueve a la posición MoverAPrincipal. Transcurrido el tiempo necesario para mover la aguja, se abre la barrera de la cinta secundaria para dejar pasar a los palés.

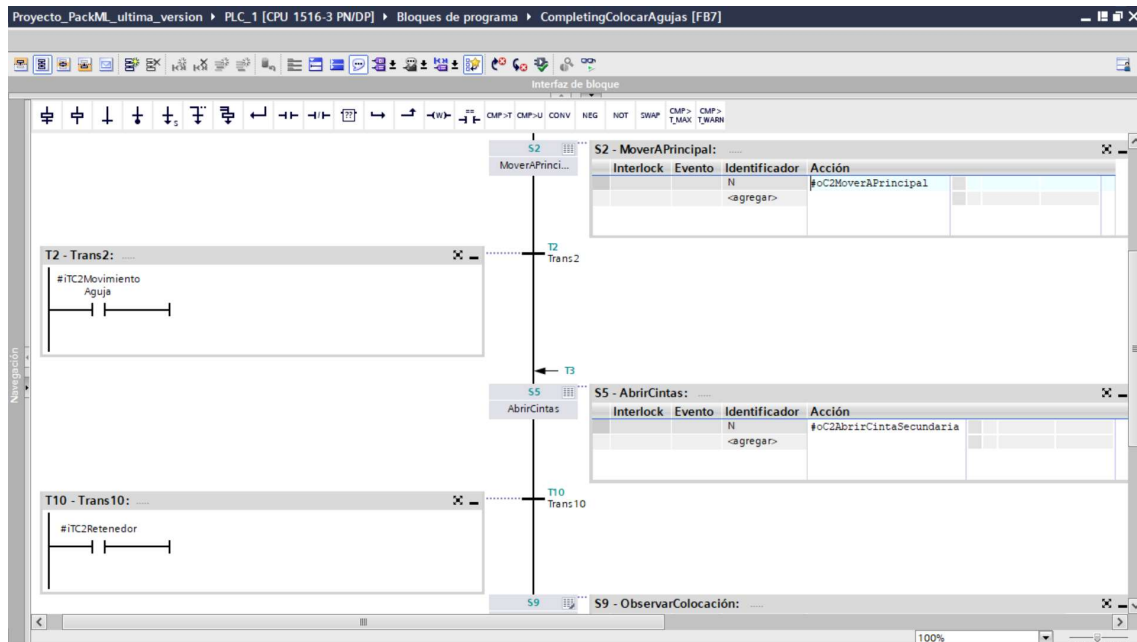


Figura 2: Etapas MoverAPrincipal y AbrirCintas

Si por alguna extraña circunstancia, la aguja se encontrase ya en la posición MoverAPrincipal, pasaría a abrir la cinta secundaria directamente.

El movimiento de las agujas y la apertura y cierre de las barreras se controla mediante temporizadores.

### 3.3.2.2. Colocación de las agujas D

Cuando termine de abrirse la barrera, se pasará a la etapa ObservarColocación, desde la que se comprobará la posición de las agujas D, posición que se controla mediante sensores. Éstos activan las variables iD12EnRecorridoLargo, iD12EnRecorridoCorto, iD34EnRecorridoLargo e iD34EnRecorridoCorto. Lo siguiente que se va a hacer es cambiar la posición de las agujas si

están en recorrido largo y si no, dejarlas como están. De nuevo, lo normal es que estén en recorrido largo las dos, porque se parte del estado Execute, pero no se pierde nada al hacer esta distinción y se gana en la seguridad de los equipos.

Tampoco se moverán las agujas si se detecta algún objeto en sus sensores.

El primer caso es que las dos agujas estén en la posición “Recorrido largo”:

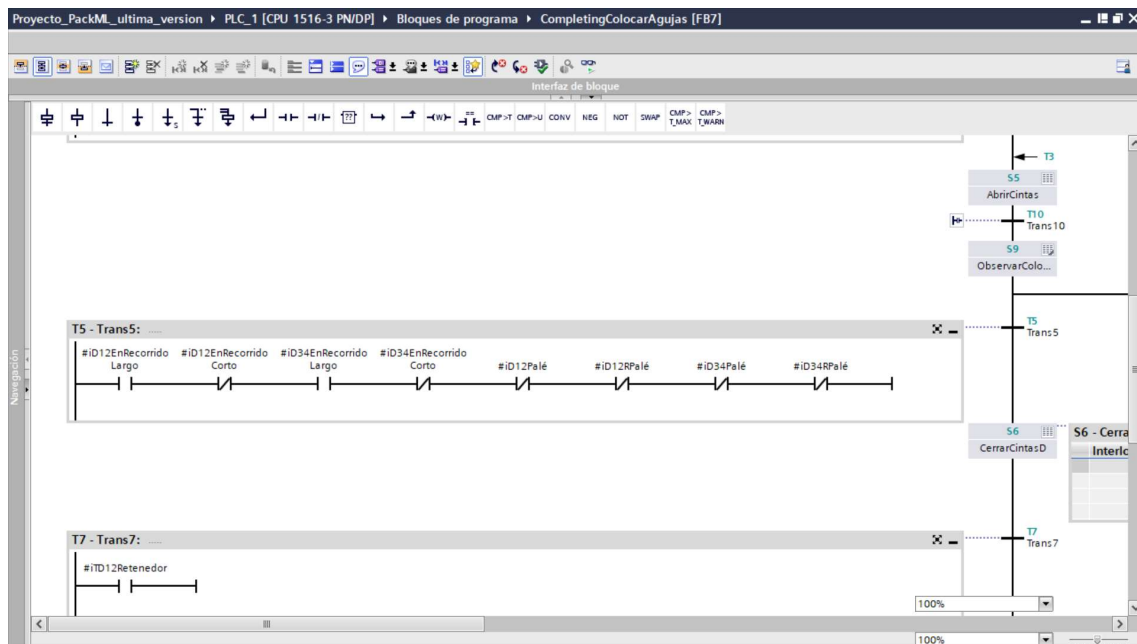


Figura 3: Las dos agujas en “recorrido largo”

En ese caso, después de cerrarse las barreras de las cintas, se mueven las dos agujas a “recorrido corto”.

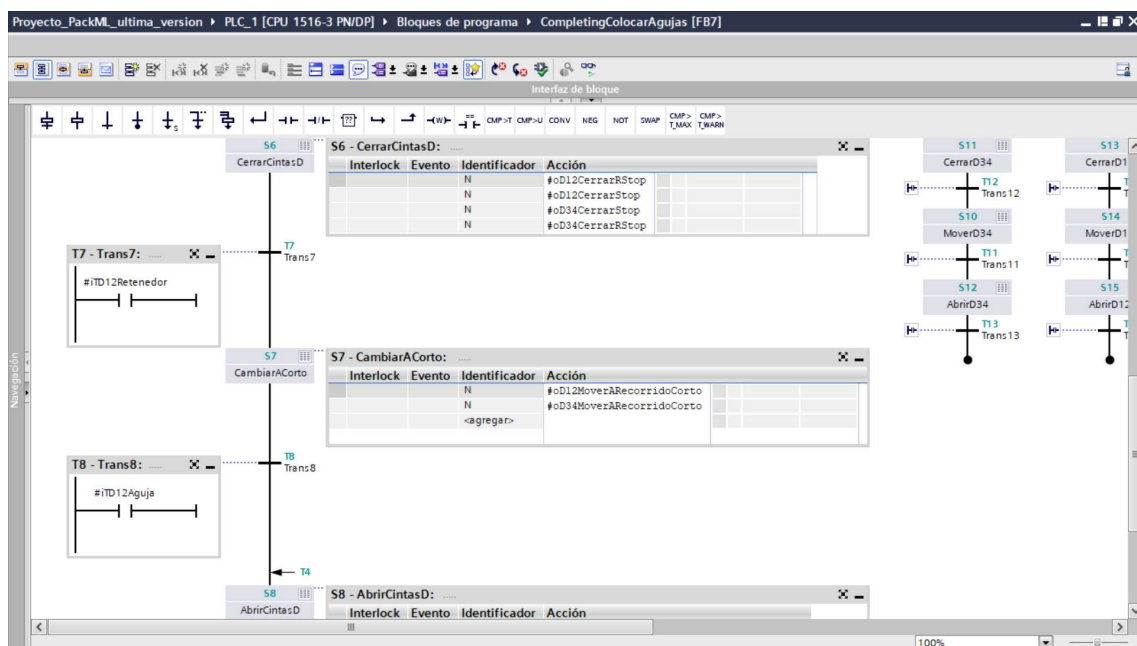


Figura 4: Etapas CerrarCintasD y CambiarACorto

Posteriormente se abren las cintas.

En el caso de que las dos agujas ya estén en recorrido corto, se pasa a la etapa AbrirCinta directamente:

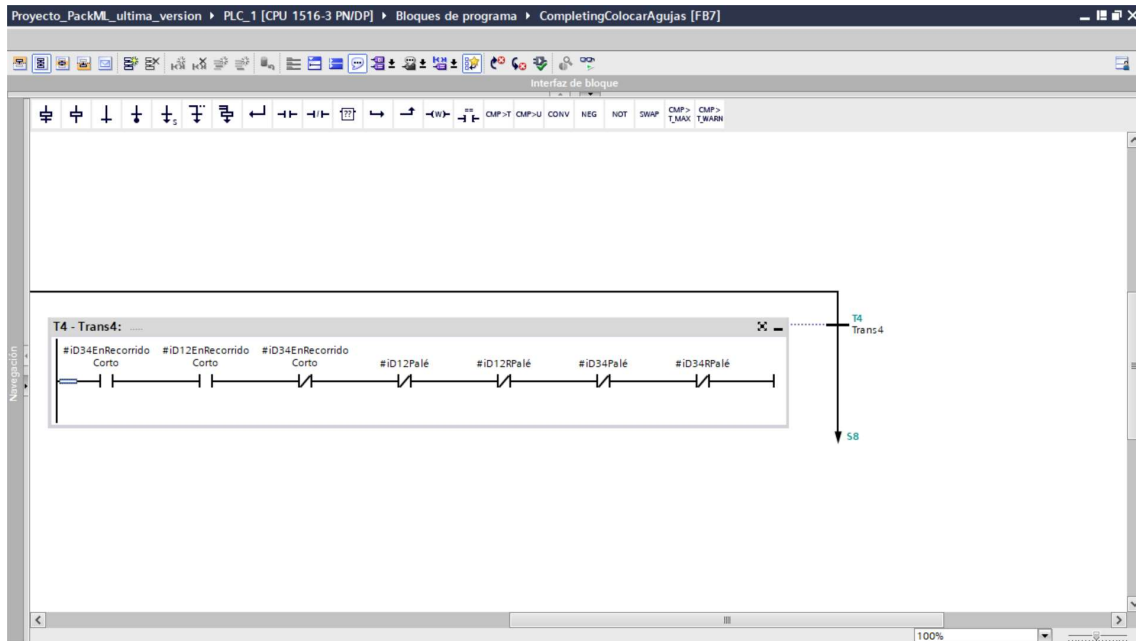


Figura 5: Las dos agujas en “recorrido corto”

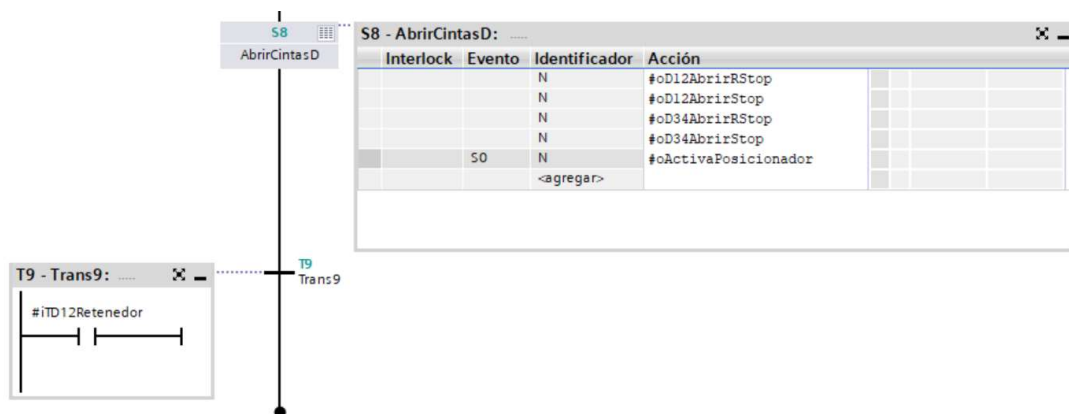


Figura 6: Etapa AbrirCintasD

Por abreviar no se incluyen en esta memoria, pero también se han tenido en cuenta los casos en los que las dos agujas se encuentran en posiciones diferentes.

A la salida de la etapa AbrirCintasD (también en la última etapa de los casos con las agujas en posiciones diferentes), se activa una variable (oActivaPosicionador) para avisar al posicionador básico de que puede ir procesando los palés. Esta variable temporal se guarda en una variable estática al final del estado Completing.

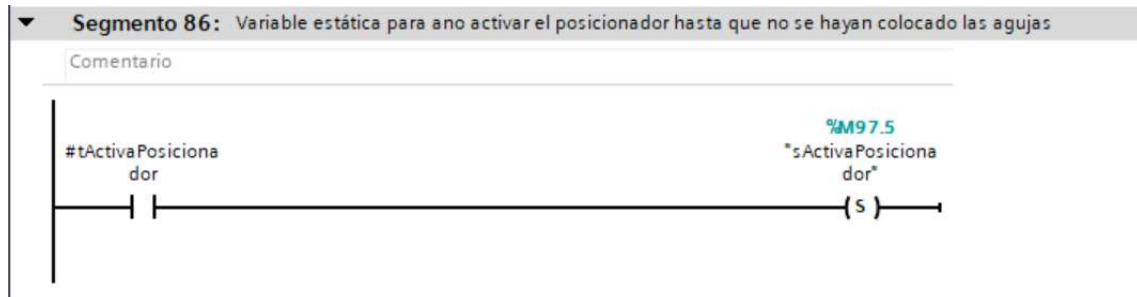


Figura 7: Variable sActivaPosicionador

Posteriormente en el estado Complete, se pone a 0 esta variable.

Una vez abiertas las dos barreras, dado que se necesita que el GRAFCET se ejecute una sola vez, se concluye éste con un fin de cadena.

### 3.3.2.3. Reinicio del GRAFCET en una nueva parada de fin de ciclo

Cuando el sistema vuelva a hacer otra parada de fin de ciclo, es necesario reiniciar el GRAFCET generando un flanco positivo en su entrada INIT\_SQ, para que vuelva a la etapa inicial.

Para ello se va a activar una variable estática a la entrada del estado Completing y otra a la salida, y después se colocarán en INIT\_SQ de esta manera:

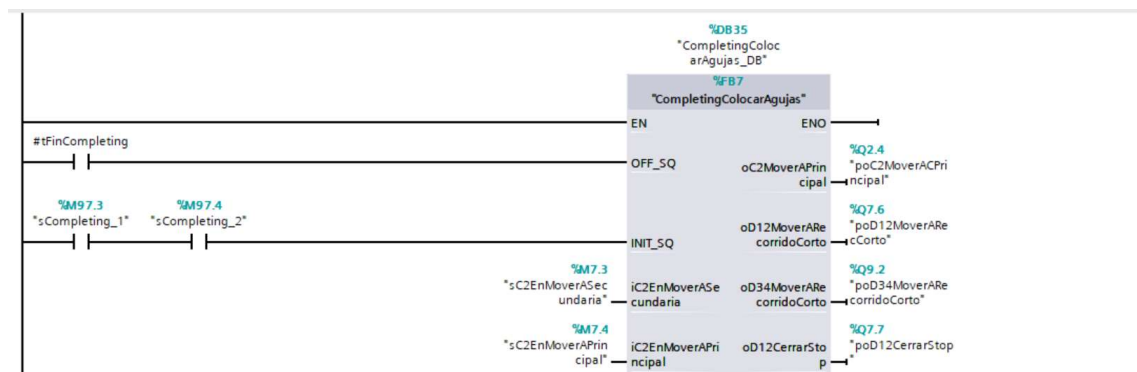


Figura 8: Reiniciar GRAFCET PosiciónAgujas

Con esta colocación, se consigue un flanco positivo en INIT\_SQ, un ciclo de ejecución después de haberse habilitado el GRAFCET.

Al pasar al estado Complete estas variables se ponen a 0.

### 3.3.3. Programación del posicionador básico S2

La actuación del posicionador básico S2 comienza cuando hay un palé a su entrada, no lo hay en el posicionador y las agujas ya están colocadas en posición:

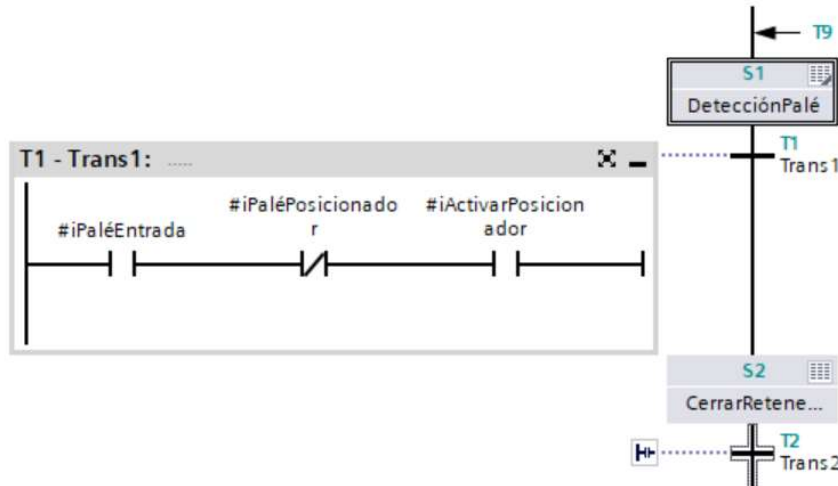


Figura 9: Condición de transición a la etapa CerrarRetenedor

A partir de aquí, el GRAFCET es casi idéntico al del estado Execute: se cierra la barrera de salida, luego se abre la de entrada y se espera a que pase el palé esté dentro del posicionador. Una vez dentro, se inmoviliza y empieza la actividad del robot. Esta actividad en el estado Execute consiste en poner una pieza de lego sobre el palé. En Completing consiste en retirar las piezas de lego.

#### 3.3.4. Fin del estado Completing

La manera que tiene el programa de saber cuándo debe pasar del estado Completing a Complete es mediante un contador de palés. El contador se incrementa cada vez que un palé abandona el posicionador básico. Finalizada su cuenta, activará una variable denominada tFinCompleting.

Esta variable temporal desactivará las etapas de los GRAFCETs para posteriormente guardar su valor en la variable estática sFinCompleting

Cuando sFinCompleting esté a 1, se pasa al estado Complete si la seta de emergencia no está pulsada.

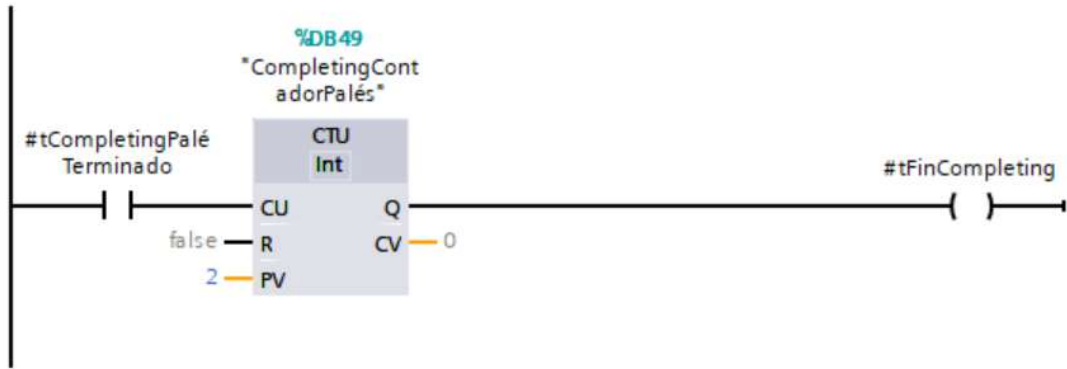


Figura 10: Contador de Palés terminados en Completing

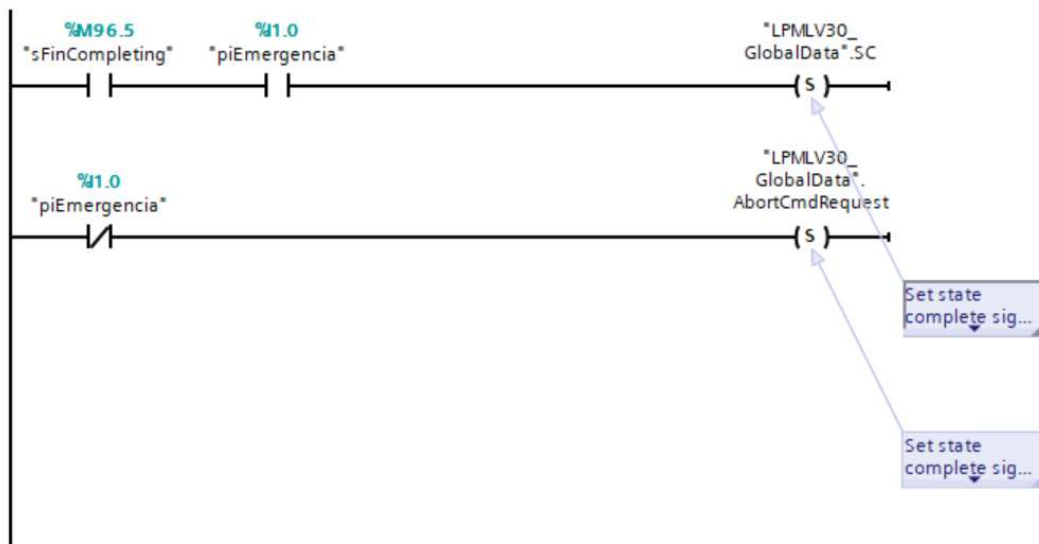


Figura 11: Transición al estado Complete

### 3.3.5. Estado Complete

En el estado Complete se apagan las cintas y se ponen a 0 todas las variables ya mencionadas:

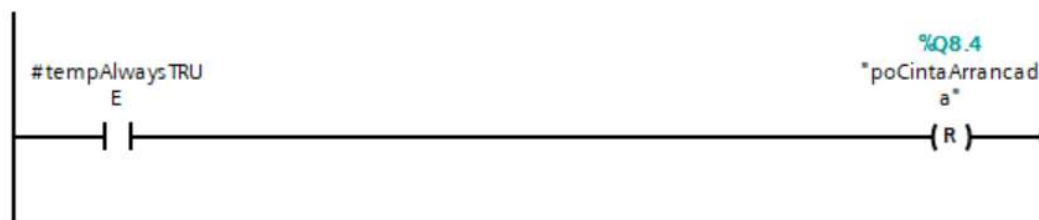


Figura 15: Apagar las cintas

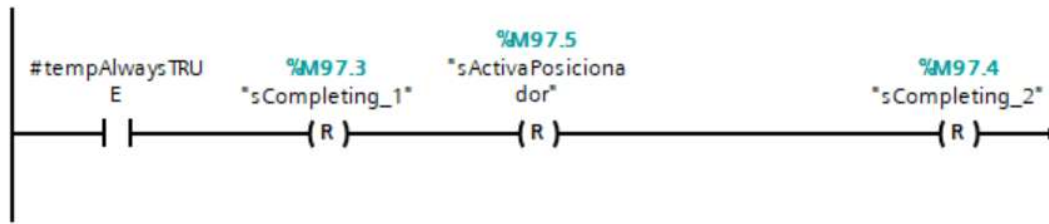


Figura 12: Preparar el sistema para una nueva parada de fin de ciclo

Finalmente, si el operador activa el botón de reiniciar y no está pulsada la seta de emergencia, se evoluciona al estado Resetting de la plantilla Pack ML.

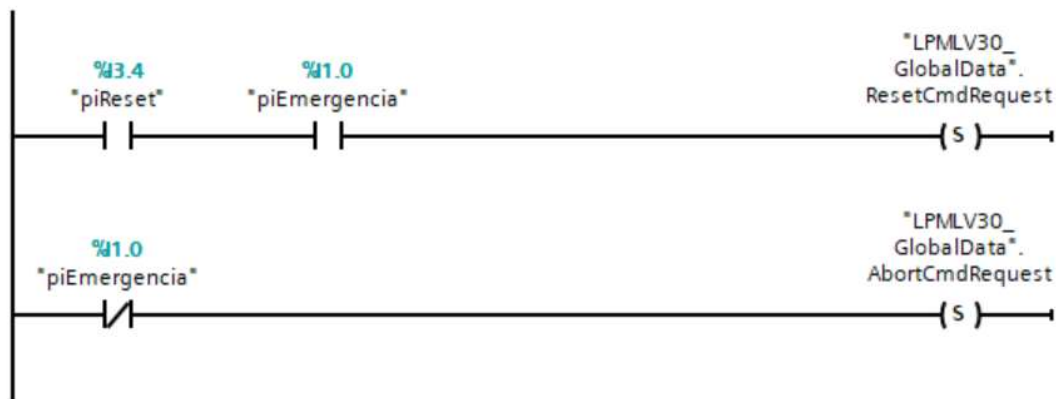


Figura 13: Transición al estado Resetting

### 3.3.6. Estado Resetting

En el estado Resetting se programa un GRAFCET para mover los palés de nuevo al posicionador básico y dejar las agujas y los posicionadores listos para una nueva puesta en marcha de los equipos.

Si el sistema viene de la parada de fin de ciclo y no de la de emergencia, se ejecutan los siguientes pasos:

- Se cambia la aguja C2 a “MoverACintaSecundaria”.
- Se abren las barreras del posicionador túnel S1.
- Se espera un tiempo a que terminen de pasar todos los palés, y después se cierran las barreras.

- Se colocan las agujas D en la posición de “Recorrido Largo”
- Por último, se activa una variable oTerminarResetting para salir de este estado.

Si se ha entrado en Resetting tras una parada de emergencia, se sale de él directamente, sin realizar los pasos anteriores:

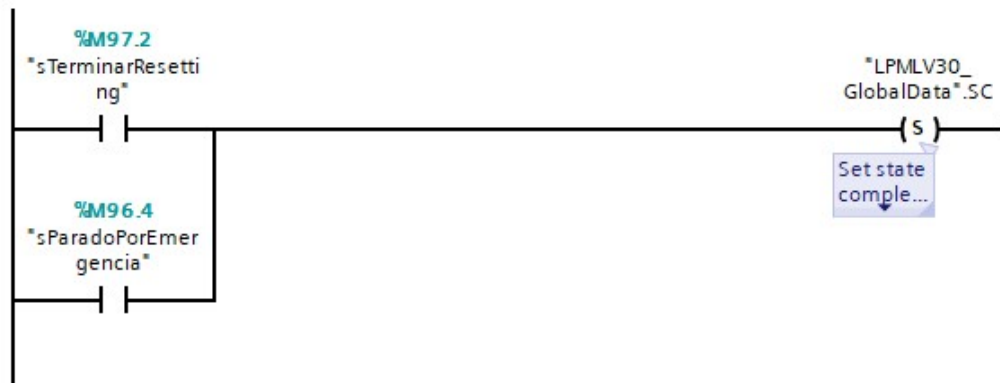


Figura 14: Salida del estado Resetting.

### 3.4. Parada de Emergencia

#### 3.4.1. Introducción

En este proyecto se ha considerado necesario implementar, además de la parada de fin de ciclo, una parada de emergencia.

Se programa por tanto un algoritmo para que cuando se pulse la seta de emergencia, el sistema evolucione hacia un estado de parada segura para las personas, los mecanismos o el medio ambiente [MOND20]. El sistema no abandonará ese estado hasta que la situación de riesgo haya desaparecido, aún incluso si se ha dejado de pulsar la seta de emergencia, cumpliendo con lo establecido por la norma ISO 13850. Una vez corregido el fallo, el automatismo podrá, mediante el rearme de la seta de emergencia, regresar al funcionamiento que tenía antes de que se pulsase dicho dispositivo. Sin embargo, pasará primero por un estado de espera del que no se saldrá a no ser que se active el comando de reinicio.

En la plantilla Pack ML, el estado al que debe evolucionar el sistema en una situación de riesgo es Aborting. Por ello, en todos los demás estados, se pasará a Aborting si se pulsa la seta de emergencia. Además, en el estado Execute también se pasará a Aborting si, durante el funcionamiento de alguno de los posicionadores túneles, el elevador tarda más tiempo del normal en subir o bajar, ya que puede ser síntoma de un fallo en alguno de los finales de carrera.

La manera de controlar el fallo en los finales de carrera es mediante temporizadores:

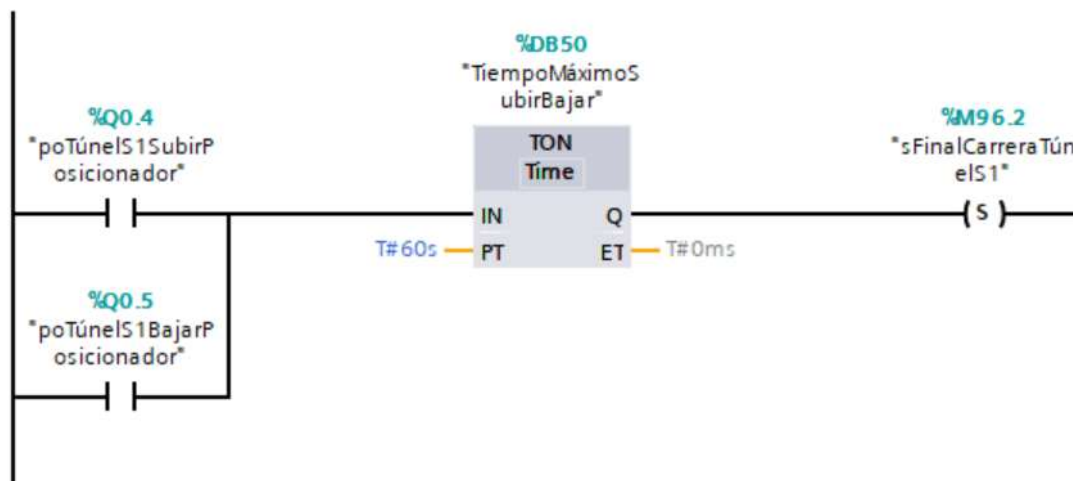


Figura 1: Temporizador para el control de fallo en los finales de carrera

En cuanto haya pasado un minuto<sup>xi</sup> sin que el elevador haya dejado de subir o de bajar, se pondrá a 1 la variable sFinalCarreraTúnelS1.

En la figura siguiente se puede observar como se ha programado la transición de Execute a Aborting.



Figura 2: Paso de Execute a Aborting

El contacto de la seta de emergencia se ha puesto normalmente cerrado. De esa manera, si por algún fallo, se interrumpe un cable en el dispositivo y deja de funcionar, se produce también una parada de emergencia.

### 3.4.2. Estado Aborting

Una vez en el estado Aborting, se ponen a 0 todas las variables de salida del plc que puedan suponer un riesgo para el operador o para el sistema, por ejemplo, los dispositivos elevadores de los posicionadores:

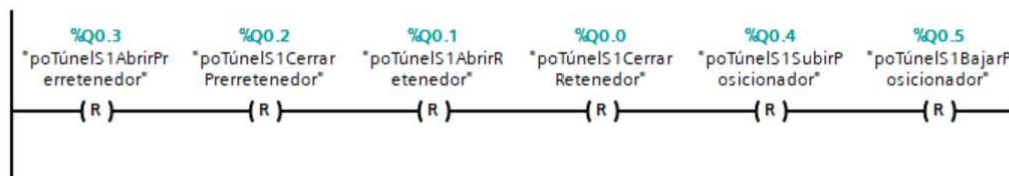


Figura 3: Puesta a 0 de las variables de salida del posicionador túnel S1

Al volver al estado normal después de una parada de emergencia, se quiere que los GRAFCETs continúen su ejecución en la etapa en la que se encontraban antes de dicha parada. Por lo tanto, no se va a usar la variable OFF\_SQ. Y eso implica que en el estado Aborting hay que poner a 0 todas las variables, incluso aquellas asociadas a etapas.

Una vez puestas a 0 todas las variables, el programa pasa a Aborted. Solo permanece en Aborting durante un ciclo de ejecución.

### 3.4.3. Estado Aborted

En el estado Aborted, se enciende un piloto que indica que el sistema se encuentra en situación de parada de emergencia. Este piloto es intermitente con una frecuencia de 0,5 Hz, gracias al segmento siguiente:

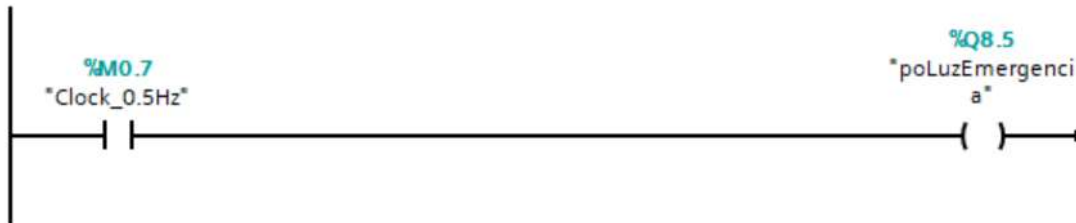


Figura 4: Encendido de la luz de emergencia (1)

Una vez que el operador deje de pulsar la seta de emergencia, se pasa al estado Clearing.

En caso de que la parada de emergencia la hayan provocado los finales de carrera, se pasa a este estado después de 30 segundos. Todo ello se programa mediante el siguiente segmento:

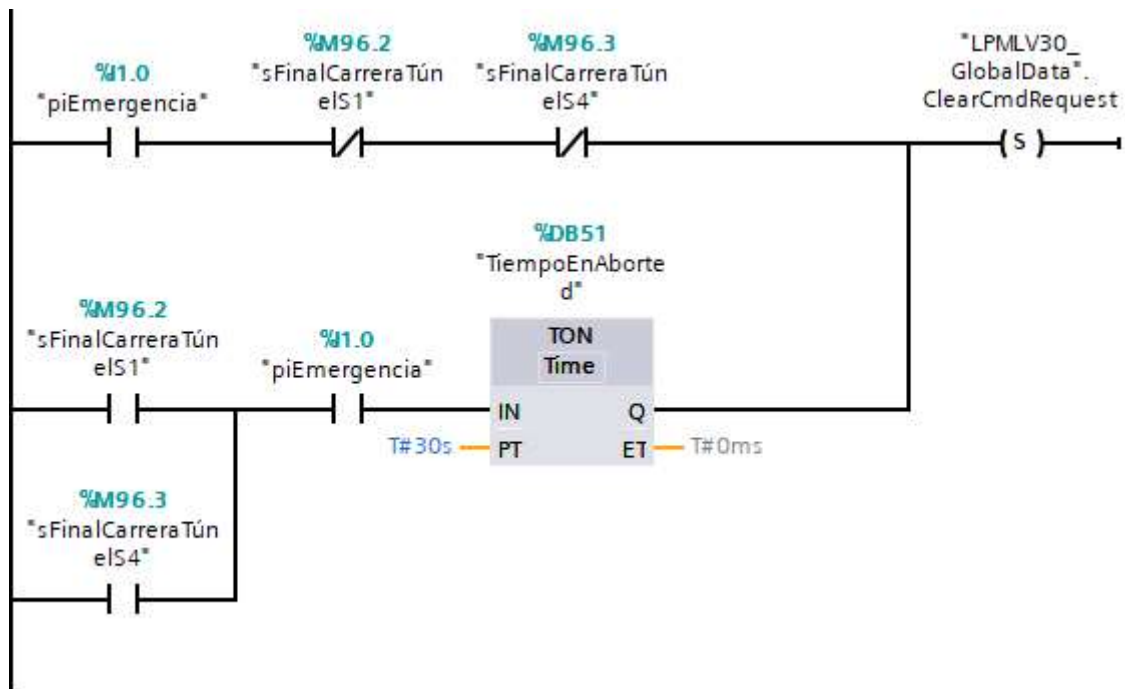


Figura 5: Paso al estado Clearing

### 3.4.4. Estado Clearing

En el estado Clearing, sigue encendida la luz de emergencia, pero ya sin intermitencias:



Figura 6: Encendido de la luz de emergencia (2)

En él se ponen a 0 las variables estáticas de los finales de carrera:

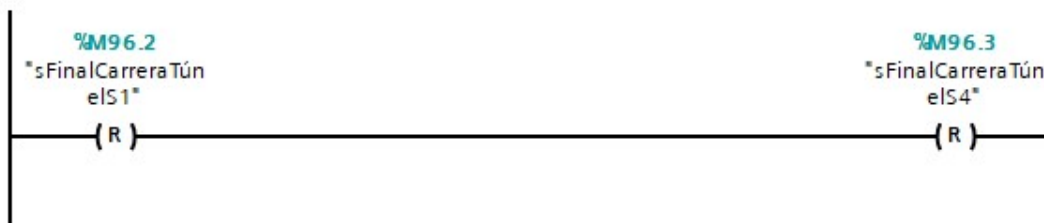


Figura 7: Puesta a 0 de las variables estáticas de los finales de carrera

Del estado Clearing se sale cuando ya se ha corregido el fallo que ha provocado la parada de emergencia. En ese momento, el operador rearma la seta de emergencia para reiniciar el programa. Por otro lado, si durante este estado, se pulsase otra vez la seta, se volvería al estado Aborting, tal y como se muestra en la figura 7:

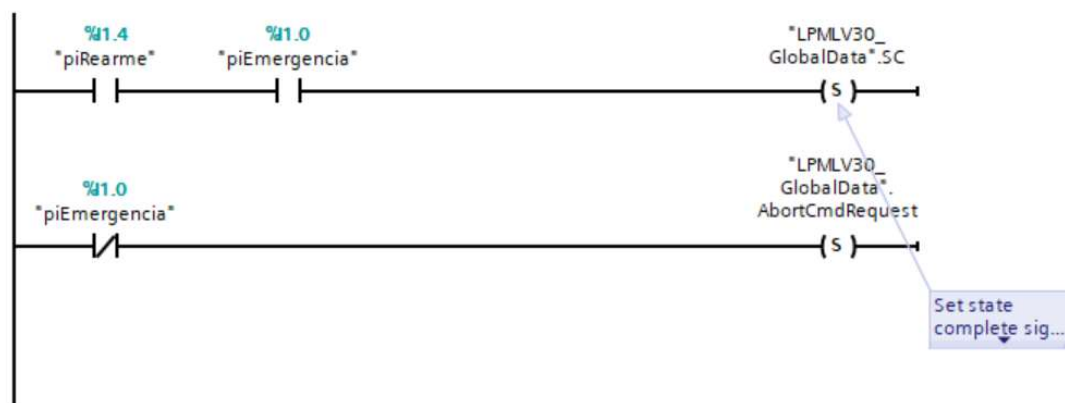


Figura8: Fin del estado Clearing

### 3.4.5. Estado Stopped

El estado Stopped es muy sencillo y en él solo se ejecutan dos acciones. Una acción consiste en activar una variable estática sParadoPorEmergencia para que en Resetting el sistema sepa que se llega de una parada de emergencia y no de la parada de fin de ciclo. La segunda consiste en apagar la luz de emergencia.

## 4. Alineamiento con los Objetivos de Desarrollo Sostenible (ODS)

En este proyecto se investiga Pack ML como una de las técnicas para mejorar los sistemas de fabricación, haciéndolos más flexibles y también más eficientes. Esto no solamente tiene una repercusión positiva en la economía, sino también en la sociedad y en el medio ambiente.

En primer lugar, un desarrollo de la industria impulsa la economía y ayuda a erradicar la pobreza. Es lo que ha ocurrido en China, donde, tras tres décadas de fuerte desarrollo industrial, ha conseguido erradicar su pobreza extrema [BBC21]. El alineamiento con el objetivo 1 (Fin de la pobreza) es claro.

Por otra parte, también puede ayudar a conseguir el objetivo de hambre 0 marcado por la ONU para el año 2030. La industria alimentaria también se puede beneficiar de Pack ML. Se puede potenciar más fácilmente esta industria en los países en vías de desarrollo, como en América latina, y, por otro lado, puede hacer que las industrias alimentarias de los países desarrollados produzcan más eficientemente, no agotando los recursos terrestres y sin contaminar el planeta (los cambios en el clima son una de las principales causas de que millones de personas sufran desnutrición, por ejemplo, en el África subsahariana [ONU21]).

Igual que la industria alimentaria, también la farmacéutica se puede beneficiar de esta plantilla.

Gracias a la facilidad para incorporar máquinas que permite Pack ML, este tipo de industria puede beneficiarse de ello, por ejemplo, produciendo más vacunas.

La cobertura de vacunación contra la covid 19 en los países de ingresos altos es 10 veces mayor que en los de ingresos bajos [CNN21]. La reducción en los costes de fabricación puede provocar un abaratamiento del precio de la vacuna que contribuya a reducir esta diferencia.

Por otro lado, está el objetivo 8 de los ODS que trata sobre trabajo decente y crecimiento económico. La automatización aleja a los empleados de los trabajos más peligrosos, en los que pueden ser sustituidos por robots u otras máquinas. Además, una mayor facilidad para programar como la que proporciona Pack ML puede permitir sistemas de seguridad programados más complejos y con menos índice de fallos.

También está en línea con el objetivo número 9 de industria, innovación e infraestructura, por lo que se ha dicho ya de que esta técnica fomenta el desarrollo industrial (al hacer menos dificultosa para las empresas la adquisición de maquinaria nueva).

Por último, una mejora en la automatización conlleva una utilización más eficiente de los recursos. Pack ML facilita la tarea de programación, lo que permite controlar automatismos más complejos. Este aumento de la complejidad podría traducirse en un mayor número de sensores para ayudar a hacer más eficiente el consumo de electricidad o de agua en las fábricas, o a reducir las emisiones de gases contaminantes. Este proyecto se alinea por lo tanto con el objetivo número 12 marcado por la ONU, que es el de producción y consumo responsable.

## 5. Conclusiones

Si hay algo que se ha aprendido durante la realización de este proyecto es que nada es sencillo. Que incluso la automatización de un dispositivo simple como puede ser una aguja que cambia de una posición a otra tiene mucha dificultad, y mucho trabajo diseñando, programando, simulando, etc. Esto es para una sola aguja. El trabajo se multiplica cuando son varios los dispositivos que se tienen que programar y más si estos deben coordinarse entre ellos. Por si fuera poco, los aparatos no solamente tienen un estado de funcionamiento, si no que pueden funcionar de una manera cuando se están procesando productos, de otra para tareas de mantenimiento, otra en la parada de fin de ciclo, etc.

El proyecto realizado, con solo siete dispositivos, es asumible incluso para un programador novato (como es el caso), pero los sistemas reales (cuando al proceso automatizado se le suman robots, cámaras, etc) son mucho más complejos. Y ya no solo es el programador el perjudicado, pero también los operadores que están controlando el proceso, o el técnico que repara una máquina.

La conclusión fundamental es que la existencia de una plantilla como Pack ML puede ayudar a solucionar este problema de complejidad. Por un lado, facilitando el trabajo a los programadores y por otro, contribuyendo a una mejor comunicación entre el operador y las máquinas.

La mayor ventaja que da Pack ML a la hora de programar es la facilidad con la que se puede pasar de un estado a otro de la plantilla. Por ello, implementar ciertas acciones, como la seta de emergencia, se ha podido realizar en muy poco tiempo, ya que los segmentos que se han tenido que diseñar son relativamente sencillos:



Figura 1: Activación de la parada de emergencia

Una vez puesta a 1 la variable "AbortCmdRequest" el programador no se tiene que ocupar de deshabilitar unos GRAFCETs y habilitar otros, la plantilla Pack ML lo hace automáticamente. Eso es algo que no se puede hacer en la guía GEMMA.

Otra ventaja es que se puede comprobar el funcionamiento de cada bloque por separado, sin necesidad de diseñar el programa completo. Volviendo al ejemplo de la seta de emergencia, con la plantilla pack ML se puede saber si al pulsarse el sistema evoluciona de Execute a Abort, sin haber diseñado si quiera esos dos estados. El programador no se tiene que preocupar si Execute está diseñado de una manera o de otra para diseñar otros estados, y esto es una gran ventaja ya que facilita el reparto de trabajo.

Y permite también ir probando parte por parte el código, lo que reduce el número de errores que se pueden cometer.

Por otra parte, la comunicación entre operador y máquina mejora mucho con Pack ML.

La plantilla tiene incorporadas unas variables con las que se puede saber en qué estado se encuentra la máquina en cada momento, basta con hacer un sinóptico simple como el que se muestra en la figura 2:



*Figura 2: Ejemplo de HMI*

Desde el panel, el operador puede ver que el sistema se encuentra en el estado Execute y que por lo tanto se encuentra procesando productos. Si dejara de iluminarse Execute y se iluminase Abort, entonces sabría que en el sistema se ha ejecutado una parada de emergencia.

Además, como es un estándar para automatizar, da igual que la máquina sea de un fabricante desconocido, basta que el operador esté familiarizado con la plantilla para comprender el funcionamiento de todas las máquinas que hay en el mercado. De esa manera, Pack ML puede contribuir al desarrollo industrial, permitiendo a las fábricas incorporar más maquinaria y adaptar mejor y en menos tiempo su sistema de producción.

Queda demostrado en este proyecto que esta plantilla contribuye a hacer más flexible la automatización de la minifábrica ICAI, recomendándose su aplicación en la automatización de equipos.

## Anexo 1: Comunicación del PLC con el resto de los dispositivos de la minifábrica ICAI

En este anexo se facilitan los esquemas de control y las tablas de variables del resto de dispositivos de la fábrica: posicionador túnel S4, posicionador básico S2, posicionador básico S3, agujas C2 y C3 y agujas Divert.

### 1. Posicionador túnel de la estación 4 (S4).

#### 1.1. Esquema de control:

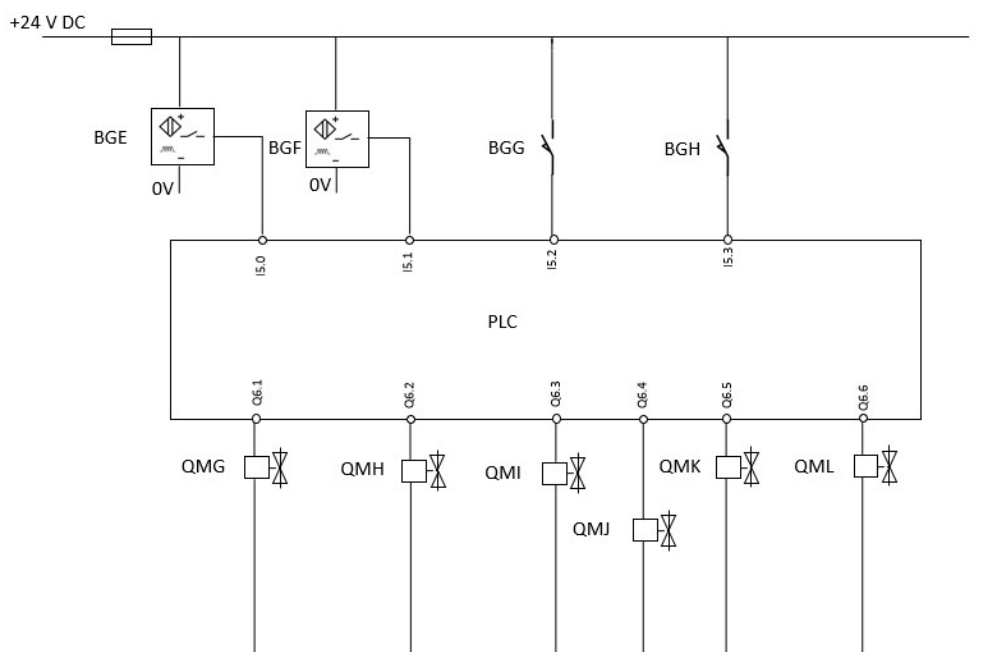


Figura 1: Esquema de control del posicionador túnel S4

#### 1.2. Tabla de Variables

ENTRADAS			
DISPOSITIVO	Nombre Variable Entrada PLC	Significado	Entrada física PLC
Sensor inductivo BGE	piTúnelS4PaléEntrada	1: Palé detectado en la entrada	%I5.0
Sensor inductivo BGF	piTúnelS4PaléPosicionador	1: Palé detectado dentro del posicionador	%I5.1

Final de Carrera BGG	piTúnelS4PosicionadorArriba	1: Elevador arriba	%I5.2
Final de Carrera BGH	piTúnelS4PosicionadorAbajo	1: Elevador abajo	%I5.3
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QMG	poTúnelS4CerrarRetenedor	1: Cerrar 0: No cerrar	%Q6.1
Válvula QMH	poTúnelS4AbrirRetenedor	1: Abrir 0: No abrir	%Q6.2
Válvula QMI	poTúnelS4CerrarPrerretenedor	1: Cerrar 0: No cerrar	%Q6.3
Válvula QMJ	poTúnelS4AbrirPrerretenedor	1: Abrir 0: No abrir	%Q6.4
Válvula QMK	poTúnelS4SubirPosicionador	1: Subir 0: No subir	%Q6.5
Válvula QML	poTúnelS4BajarPosicionador	1: Bajar 0: No bajar	%Q6.6

Tabla 1: Entradas y salidas del plc para el posicionador túnel S4

## 2. Posicionador Básico S2

### 2.1. Esquema de control

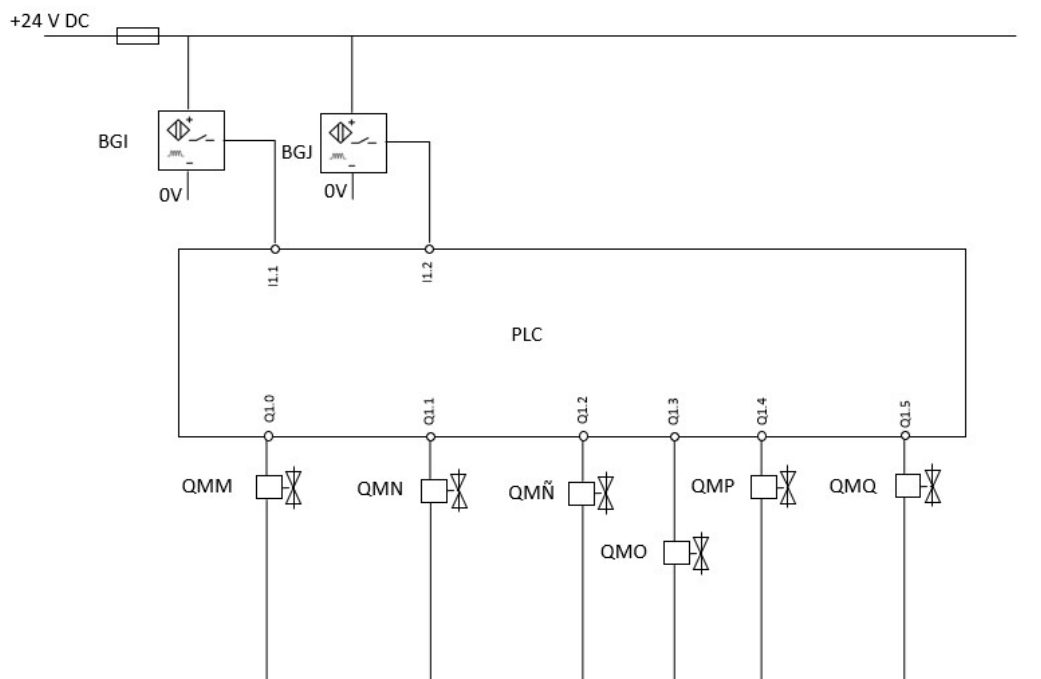


Figura 2: Esquema de control del posicionador básico S2

## 2.2. Tabla de variables

<b>ENTRADAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Entrada PLC</b>	<b>Significado</b>	<b>Entrada física PLC</b>
Sensor inductivo BGI	piBásicoS2PaléEntrada	1: Palé detectado en la entrada	%I1.1
Sensor inductivo BGJ	piBásicoS2PaléPosicionador	1: Palé detectado dentro del posicionador	%I1.2
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QMM	poBásicoS2CerrarRetenedor	1: Cerrar 0: No cerrar	%Q1.0
Válvula QMN	poBásicoS2AbrirRetenedor	1: Abrir 0: No abrir	% Q1.1
Válvula QMÑ	poBásicoS2CerrarPrerretenedor	1: Cerrar 0: No cerrar	%Q1.2
Válvula QMO	poBásicoS2AbrirPrerretenedor	1: Abrir 0: No abrir	%Q1.3
Válvula QMP	poBásicoS2Inmovilizar	1: Inmovilizar 0: No inmovilizar	%Q1.4
Válvula QMQ	poBásicoS2Desinmovilizar	1: Des inmovilizar 0: Mantener inmovilizado	%Q1.5

*Tabla 2: Entradas y salidas del plc para el posicionador básico S2*

### 3. Posicionador Básico S3

#### 3.1. Esquema de control

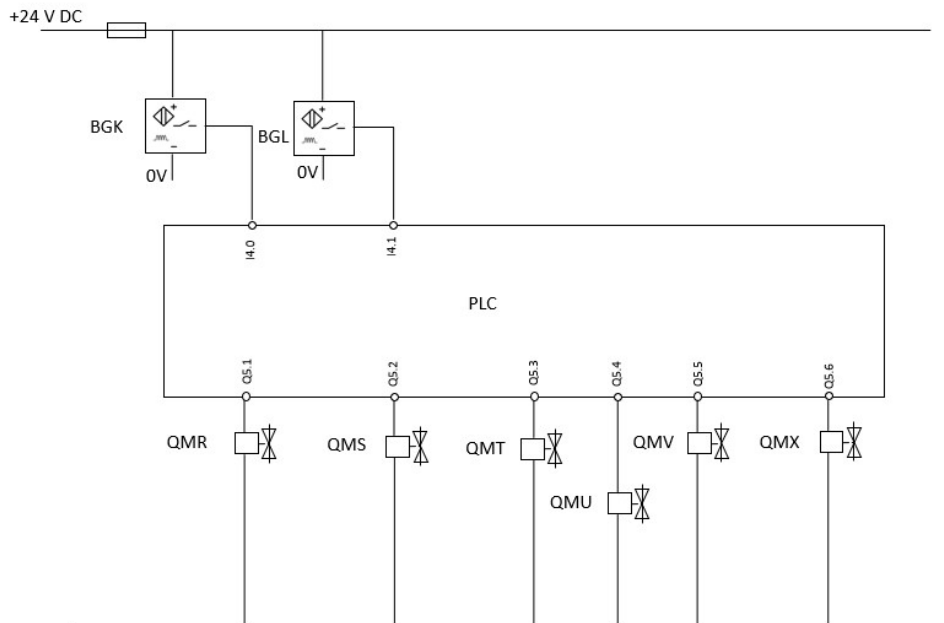


Figura 3: Esquema de control del posicionador básico S3

#### 3.2. Tabla de variables

<b>ENTRADAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Entrada PLC</b>	<b>Significado</b>	<b>Entrada física PLC</b>
Sensor inductivo BGK	piBásicoS3PaléEntrada	1: Palé detectado en la entrada	%I4.0
Sensor inductivo BGL	piBásicoS3PaléPosicionador	1: Palé detectado dentro del posicionador	%I4.1
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QMR	poBásicoS3CerrarRetenedor	1: Cerrar 0: No cerrar	%Q5.1
Válvula QMS	poBásicoS3AbrirRetenedor	1: Abrir 0: No abrir	% Q5.2
Válvula QMT	poBásicoS3CerrarPrerretenedor	1: Cerrar 0: No cerrar	%Q5.3
Válvula	poBásicoS3AbrirPrerretenedor	1: Abrir	%Q5.4

QMU		0: No abrir	
Válvula QMV	poBásicoS3Inmovilizar	1: Inmovilizar 0: No inmovilizar	%Q5.5
Válvula QMX	poBásicoS3Desinmovilizar	1: Des inmovilizar 0: Mantener inmovilizado	%Q5.6

Tabla 3: Entradas y salidas del plc para el posicionador básico S3

#### 4. Aguja C2

##### 4.1. Esquema de control

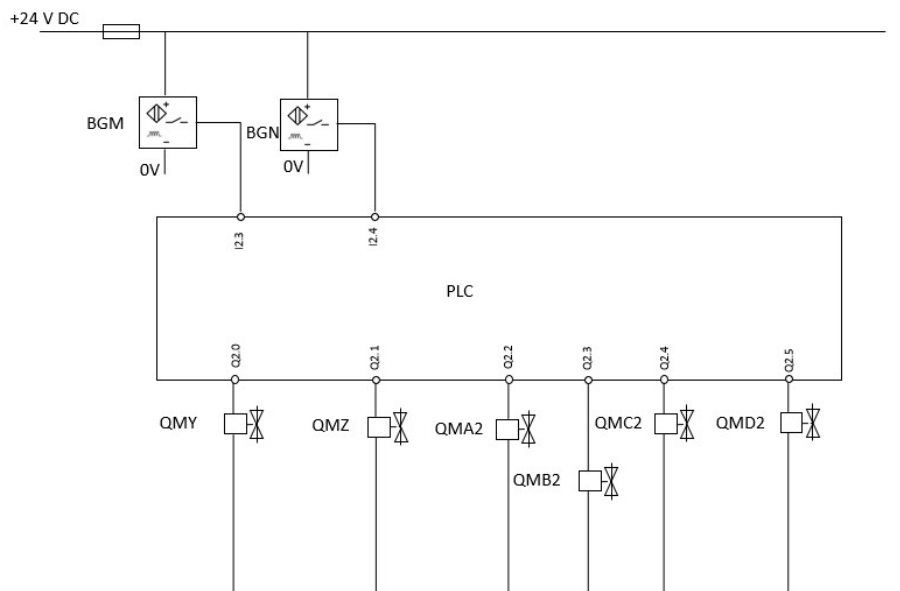


Figura 4: Esquema de control de la aguja C2

##### 4.2. Tabla de variables

<b>ENTRADAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Entrada PLC</b>	<b>Significado</b>	<b>Entrada física PLC</b>
Sensor inductivo BGM	piC2PaléCintaPrincipal	1: Palé detectado en cinta principal	%I2.3
Sensor inductivo BGN	piC2PaléCintaSecundaria	1: Palé detectado en cinta secundaria	%I2.4
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QMY	poC2CerrarCintaPrincipal	1: Cerrar 0: No cerrar	%Q2.0
Válvula QMZ	poC2AbrirCintaPrincipal	1: Abrir 0: No abrir	% Q2.1
Válvula QMA2	poC2CerrarCintaSecundaria	1: Cerrar 0: No cerrar	%Q2.2
Válvula QMB2	poC2AbrirCintaSecundaria	1: Abrir 0: No abrir	%Q2.3
Válvula QMC2	poC2MoverACPrincipal	1: Colocarse en esa posición 0: No colocarse	%Q2.4
Válvula QMD2	poC2MoverACSecundaria	1: Colocarse en esa posición 0: No colocarse	%Q2.5

Tabla 4: Entradas y salidas del plc para la aguja C2

## 5. Aguja C3

### 5.1. Esquema de control

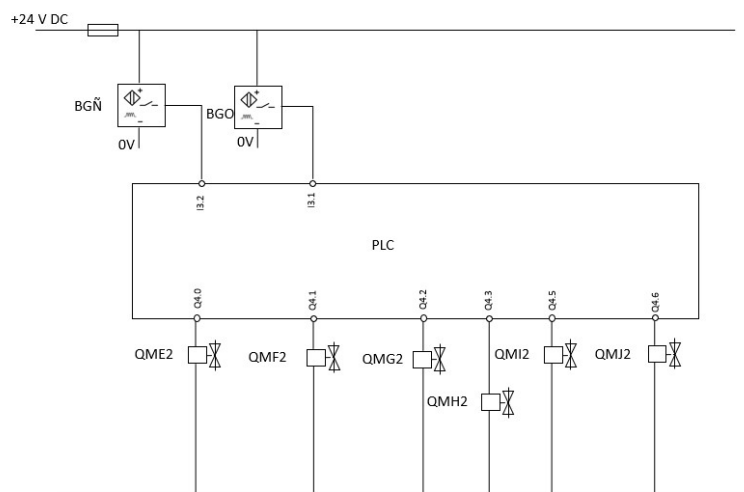


Figura 5: Esquema de control de la aguja C3

## 5.2. Tabla de variables

<b>ENTRADAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Entrada PLC</b>	<b>Significado</b>	<b>Entrada física PLC</b>
Sensor inductivo BGÑ	piC3PaléCintaPrincipal	1: Palé detectado en cinta principal	%I3.2
Sensor inductivo BGO	piC3PaléCintaSecundaria	1: Palé detectado en cinta secundaria	%I3.3
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QME2	poC3CerrarCintaSecundaria	1: Cerrar 0: No cerrar	%Q4.0
Válvula QMF2	poC3CerrarCintaPrincipal	1: Cerrar 0: No cerrar	% Q4.1
Válvula QMG2	poC3AbrirCintaprincipal	1: Abrir 0: No abrir	%Q4.2
Válvula QMH2	poC3AbrirCintaSecundaria	1: Abrir 0: No abrir	%Q4.3
Válvula QMI2	poC3MoverACPrincipal	1: Colocarse en esa posición 0: No colocarse	%Q4.5
Válvula QMJ2	poC3MoverACSecundaria	1: Colocarse en esa posición 0: No colocarse	%Q4.6

*Tabla 5: Entradas y salidas del plc para la aguja C3*

## 6. Aguja D12

## 6.1. Esquema de control

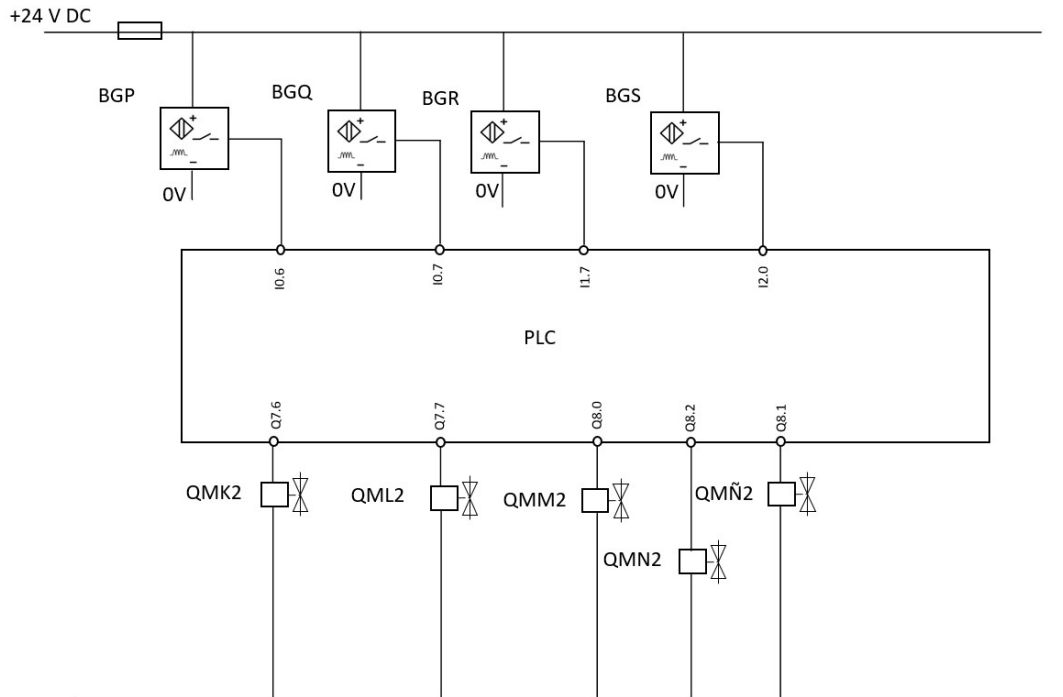


Figura 6: Esquema de control de la aguja D12

## 6.2. Tabla de variables

ENTRADAS			
DISPOSITIVO	Nombre Variable Entrada PLC	Significado	Entrada física PLC
Sensor inductivo BGP	piD12enRecLargo	1: Aguja en posición que permita el recorrido del palé por toda la cinta principal	%I0.6
Sensor inductivo BGQ	piD12enRecCorto	1: Aguja en posición que permita el recorrido del palé solo por una parte de la cinta principal	%I0.7
Sensor inductivo BGR	piD12RPalé	1: palé detectado	%I1.7
Sensor inductivo BGS	piD12Palé	1: palé detectado	%I2.0
SALIDAS			
DISPOSITIVO	Nombre Variable Salida PLC	Significado	Salida física PLC
Válvula QMK2	poD12MoverARecCorto	1: Mover a recorrido corto	%Q7.6

		0: No mover	
Válvula QML2	poD12CerrarStop	1: Cerrar 0: No cerrar	%Q7.7
Válvula QMM2	poD12CerrarRStop	1: Cerrar 0: No cerrar	%Q8.0
Válvula QMN2	poD12AbrirStop	1: Abrir 0: No abrir	%Q8.2
Válvula QMÑ2	poD12AbrirRStop	1: Abrir 0: No abrir	%Q8.1

Tabla 6: Entradas y salidas del plc para la aguja D12

7. Aguja D34

7.1. Esquema de control

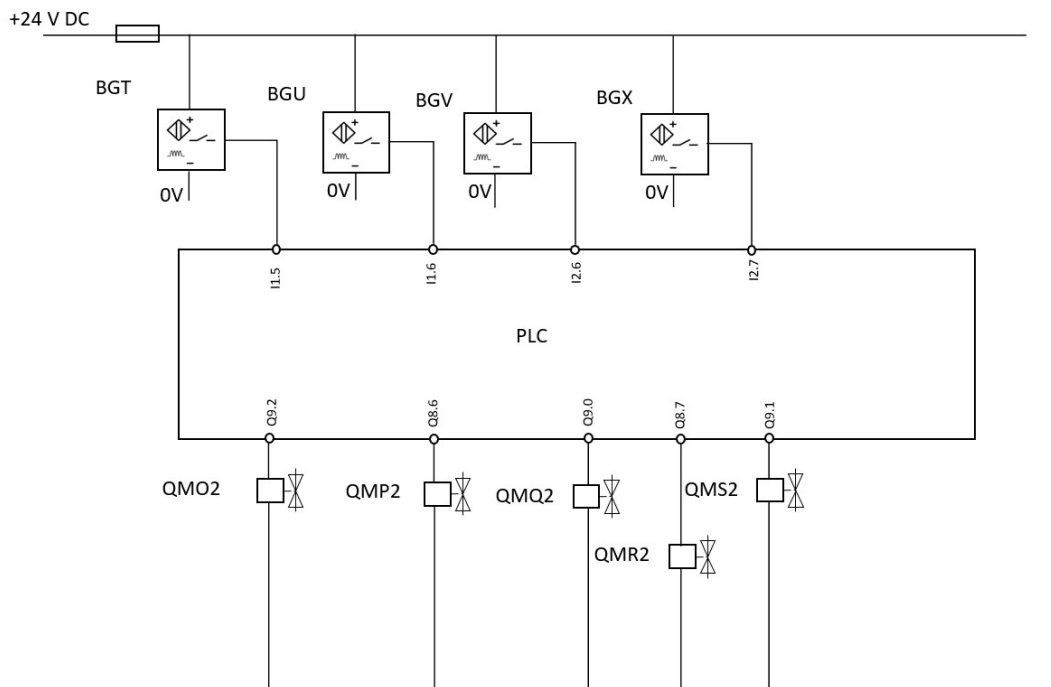


Figura 7: Esquema de control de la aguja D34

7.2. Tabla de variables

ENTRADAS			
DISPOSITIVO	Nombre Variable Entrada PLC	Significado	Entrada física PLC
Sensor inductivo BGT	piD34enRecLargo	1: Aguja en posición que permita el recorrido del palé por toda la cinta principal	%I1.5

Sensor inductivo BGU	piD34enRecCorto	1: Aguja en posición que permita el recorrido del palé solo por una parte de la cinta principal	%I1.6
Sensor inductivo BGV	piD34RPalé	1: palé detectado	%I2.6
Sensor inductivo BGX	piD34Palé	1: palé detectado	%I2.7
<b>SALIDAS</b>			
<b>DISPOSITIVO</b>	<b>Nombre Variable Salida PLC</b>	<b>Significado</b>	<b>Salida física PLC</b>
Válvula QMO2	poD34MoverARecCorto	1: Mover a recorrido corto 0: No mover	%Q9.2
Válvula QMP2	poD34CerrarStop	1: Cerrar 0: No cerrar	%Q8.6
Válvula QMQ2	poD34CerrarRStop	1: Cerrar 0: No cerrar	%Q9.0
Válvula QMR2	poD34AbrirStop	1: Abrir 0: No abrir	%Q8.7
Válvula QMS2	poD34AbrirRStop	1: Abrir 0: No abrir	%Q9.1

*Tabla 7: Entradas y salidas del plc para la aguja D34*

## Referencias

### Introducción y planteamiento del proyecto

- [THOM06] Thomson, I., “Definición de empresa”. Enero 2006.  
[DEFINICIÓN DE EMPRESA - Promonegocios.net](#)  
Artículo divulgativo.  
Último acceso: 18/07/2021
- [MOND20] Rodríguez Mondéjar, J.A., “Introducción”, pág.2. Apuntes de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. Enero 2020.  
Apuntes.  
Acceso restringido.
- [FORT13] García Fortes, J.A., Res López, V., “Introducción a la Automatización Industrial”, pág.7. Asignatura de automatización industrial. Departamento de Ingeniería de Sistemas y Automática. Universidad de Málaga.  
[Tema 08 Introducción a la automatización industrial \(uma.es\)](#)  
Apuntes.  
Último acceso: 18/07/2021
- [OMAC21] Organization for Machine Automation and Control (OMAC), “What is Pack ML?”  
[PackML \(omac.org\)](#)  
Página web.  
Último acceso: 18/07/2021

- [MOND18] a. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 175, párrafo 3. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.
- Apuntes.
- Acceso restringido.
- b. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 326, párrafo 3. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.
- Apuntes.
- Acceso restringido.

## Descripción de las tecnologías

### Herramientas

- [SIEM21] a. Siemens, PLC Programming with SIMATIC STEP7, “Hardware requirements”, 2021.
- [PLC programming with SIMATIC STEP 7 \(TIA Portal\) | Software in TIA Portal | Siemens Global](#)
- Página web.
- Último acceso: 18/07/2021
- b. Siemens, Products & Services, “SIMATIC STEP7 Basic (TIA Portal)”, 2021.
- [SIMATIC STEP 7 Basic | PLC Programming with SIMATIC STEP 7 | Siemens USA](#)
- Página web.
- Último acceso: 18/07/2021
- [SIEM20] Delivery Release SIMATIC WinCC Unified V16, “Operating Systems”. Abril 2020.
- [Delivery Release SIMATIC WinCC Unified V16 - ID: 109771777 - Industry Support Siemens](#)
- Página web.
- Último acceso: 18/07/2021

- [MOND20] a. Rodríguez Mondéjar, J.A., “Figura 2 Bastidor con PLC”, COMILLAS-ICAI Prácticas de Automatización Industrial, Madrid. 2020.
- Práctica de laboratorio.
- Acceso restringido.
- b. Rodríguez Mondéjar, J.A., “Tabla1 Elementos montados en el bastidor, de izquierda a derecha” COMILLAS-ICAI Prácticas de Automatización Industrial, Madrid. 2020.
- Práctica de laboratorio.
- Acceso restringido.
- c. Rodríguez Mondéjar, J.A., “Panel táctil TP700 Comfort” COMILLAS-ICAI Prácticas de Automatización Industrial, Madrid. 2020.
- Práctica de laboratorio.
- Acceso restringido.
- [4]Rodríguez Mondéjar, J.A., “¿En qué lenguaje se programan los PLCs?”, Escuela Técnica Superior de Ingeniería ICAI. Apuntes de Automatización Industrial. Tecnologías-PLC. Enero 2020.
- Apuntes.
- Acceso restringido.
- [SIEM16] SIMATIC, S7-1500, CPU 1516-3 PN/DP(6ES7516-3AN01-0AB0), Manual de Producto, “Datos técnicos”. Septiembre 2016.
- [CPU 1516-3 PN/DP \(6ES7516-3AN01-0AB0\) \(siemens.com\)](https://www.siemens.com/press/en/press-releases/2016/09/09160901.htm)
- Página web.
- Último acceso: 18/07/2021
- [HEGA20] TIA Portal: Basic Mathematics in SCL (Structured Control Language)!.  
YouTube: Hegamurl, [165](https://www.youtube.com/watch?v=165) TIA Portal: Basic Mathematics in SCL (Structured Control Language)! - YouTube
- Contenido audiovisual.

Último acceso: 18/07/2021

Técnicas de automatización utilizadas

[MOND18] a. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 175. Asignatura de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

b. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 177. Asignatura de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

c. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 201. Asignatura de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

d. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 206. Asignatura de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

e. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 326. Asignatura de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

f. Rodríguez Mondéjar, J.A., “Apuntes de Automatización Industrial”, pág. 339. Asignatura de Automatización Industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

[MOND20] Rodríguez Mondéjar, J.A., “Introducción”, pág. 17. Apuntes de Automatización industrial. Departamento de Electrónica, Automática y Comunicaciones. Escuela Técnica Superior de Ingeniería ICAI. 2018.

Apuntes.

Acceso restringido.

[OMAC21] Organization for Machine Automation and Control (OMAC), “What is Pack ML?”

[PackML \(omac.org\)](http://omac.org)

Página web.

Último acceso: 18/07/2021

[SIEM15] a. Siemens, “OMAC PackML V3.0 for S7-1200/S7-1500”, Library description, figura 1-2, pág. 7. Mayo 2015

[LPMLV30 OMAC library SIMATIC V3 0.pdf](#)

Página web.

Último acceso: 18/07/2021

b. Siemens, “OMAC PackML V3.0 for S7-1200/S7-1500”, Library description, tabla 1-2, pág. 9-10. Mayo 2015

[LPMLV30 OMAC library SIMATIC V3 0.pdf](#)

Página web.

Último acceso: 18/07/2021

## Descripción de la minifábrica ICAI

- [MOND20] c. Rodríguez Mondéjar, J.A., “Figura 1 Minifábrica ICAI”, COMILLAS-ICAI Prácticas de Automatización Avanzada, Madrid. 2020.
- Práctica de laboratorio.
- Acceso restringido.
- d. Rodríguez Mondéjar, J.A., COMILLAS-ICAI Prácticas de laboratorio. Asignatura de Automatización Avanzada, Madrid. 2020.
- Práctica de laboratorio.
- Acceso restringido.
- [SIGM20] SIGMA IMECSA, “Robot ABB IRB 120-Parte Principales de un Robot Industrial ¿Cómo funcionan?”
- [Robot ABB IRB 120 - Partes Principales de un Robot Industrial ¿Cómo funcionan? - SIGMA IMECSA - Bing video](#)
- Contenido audiovisual.
- Último acceso: 18/07/2021
- [FLEX16] Flex Link, “Sistema de palets con módulos estándar y RFID”. 2016
- [5147ES6.book\(3X85PAL.fm\) \(flexlink.com\)](#)
- Página web.
- Último acceso: 18/07/2021
- [FLEX21] a. Flex Link, “Conveyor system X85, links and chains”.2021
- [X85.fm \(flexlink.com\)](#)
- Página web.
- Último acceso: 18/07/2021

b. Flex Link, “Basic locating module”, User documentation ver. 1.0.

[ManualPosicionadorSimple\\_Basic-locating-module-XBUL11.pdf](#)

Página web.

Último acceso: 18/07/2021

c. Flex Link, “Tunnel locating module”, User documentation ver. 1.0.

[ManualPosicionadorTúnel\\_Tunnel-locating-module.pdf](#)

Página web.

Último acceso: 18/07/2021

#### Descripción del modelo desarrollado

##### Estado Execute

[MOND20] Rodríguez Mondéjar, J.A., “Figura 1 Minifábrica ICAI”, COMILLAS-ICAI Prácticas de Automatización Avanzada, Madrid. 2020.

Práctica de laboratorio.

Acceso restringido.

##### Simulación del estado Execute

[MOND20] Rodríguez Mondéjar, J.A. “Tecnologías-Cableada”, Apuntes de Automatización Industrial, Escuela Técnica Superior de Ingeniería ICAI, pág. 34. Enero 2020.

Apuntes.

Acceso restringido.

##### Parada de emergencia

[MOND20] Rodríguez Mondéjar, J.A., “Seta de emergencia”, Escuela Técnica Superior de Ingeniería ICAI, Apuntes de Automatización Industrial, Introducción. Enero 2020.

Apuntes.

Acceso restringido.

## Objetivos de Desarrollo Sostenible

- [BBC21] BBC News Mundo, “[Cómo consiguió China erradicar la pobreza extrema \[...\]](#)”. Febrero 26, 2021
- [Cómo consiguió China erradicar la pobreza extrema \(y las dudas que despierta ese triunfal anuncio del gobierno de Xi\) - BBC News Mundo](#)
- Artículo periodístico.
- Último acceso: 18/07/2021
- [ONU21] ONU (Organización de las Naciones Unidas), Objetivos de Desarrollo Sostenible, “[Hambre cero](#)”
- [Objetivos de Desarrollo Sostenible | PNUD \(undp.org\)](#)
- Página web.
- Último acceso: 18/07/2021
- [CNN21] McPhilips,D., “El 10% de la población mundial está completamente vacunada contra el covid-19, según datos”, [cnn.com](#). Junio 22, 2021.
- [El 10% de la población mundial completamente vacunada contra covid-19 \(cnn.com\)](#)
- Artículo periodístico.
- Último acceso: 18/07/2021

---

<sup>i</sup> La versión mínima admitida por los PLCs del laboratorio es la 14.

<sup>ii</sup> Esta es la denominación que recibe este lenguaje en los programas de Siemens. KOP figura en la Comisión Electrotécnica Internacional (IEC) como LD (Ladder diagram).

<sup>iii</sup> SFC(Sequential Function Chart) en IEC.

<sup>iv</sup> ST( Structured Text) en IEC.

<sup>v</sup> FBD(Function block diagram) en IEC

<sup>vi</sup> IL (Instruction List) en IEC.

<sup>vii</sup> Acrónimo que en español se traduce como Gráfico Funcional de Control de Etapas y Transiciones.

<sup>viii</sup> SFC es el acrónimo de Sequential Function Chart (Gráfico de Funciones Secuenciales). La versión de este lenguaje en los PLC de Siemens se conoce como S7-GRAPH.

<sup>ix</sup> ADEPA es el acrónimo de *Agence Nationale pour le DEveloppement de la Productique Appliquée à l'industrie* , Agencia Nacional para el desarrollo de la producción aplicada a la industria.

<sup>x</sup> Se ha modificado el plano original para añadir el recorrido del palé.

<sup>xi</sup> Este tiempo se ha utilizado para comprobar el programa, pero podría ser diferente al tiempo de subida/bajada del elevador.