



industriales  
etsii

Escuela Técnica  
Superior  
de Ingeniería  
Industrial

# UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

## Desarrollo de entornos virtuales para la representación de procesos automatizables

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS INDUSTRIALES

**Autor:** Alberto Nicolás Sánchez  
**Director:** Roque Torres Sanchez

Cartagena, 01/09/2022



Universidad  
Politécnica  
de Cartagena



## Índice

<b>Capítulo 1. Introducción</b> .....	7
<b>Capítulo 2. Objetivo</b> .....	8
<b>Capítulo 3. Material</b> .....	9
3.1 TIA Portal .....	9
3.2 PLC .....	9
3.2.1 Definición .....	9
3.2.2 Arquitectura de un PLC .....	10
3.2.3 Bloques de memoria en un PLC .....	10
3.2.4 Descripción funcional .....	11
3.3 Maquetas Lucas-Nülle .....	12
3.4 HMI .....	13
<b>Capítulo 4. Metodología</b> .....	14
4.1 Metodología de las prácticas de Automatización Industrial .....	14
4.2 Metodología del proceso de virtualización de las prácticas .....	15
4.2.1 PLC .....	15
4.2.2 HMI .....	17
4.2.2.1 Representación gráfica .....	17
4.2.2.2 Variables HMI .....	19
4.2.2.3 Botones e interruptores .....	19
4.2.2.4 Apariencia y visibilidad .....	20
4.2.2.5 Movimiento .....	21
<b>Capítulo 5. Maquetas</b> .....	26
5.1 Plataforma de cambio de mercancía .....	26
5.1.1 Enunciado .....	26
5.1.2 Representación gráfica .....	27
5.1.3 Variables .....	28
5.1.4 Movimiento .....	29
5.1.5 Visibilidad y apariencia .....	31
5.2 Dobladora .....	36
5.2.1 Enunciado .....	36
5.2.2 Representación gráfica .....	37
5.2.3 Movimiento .....	38
5.2.4 Visibilidad y apariencia .....	39
5.3 Semáforo de obra .....	42
5.3.1 Enunciado .....	42
5.3.2 Representación gráfica .....	43



5.3.3 Movimiento .....	43
5.3.4 Visibilidad y apariencia .....	45
5.4 Taladradora .....	46
5.4.1 Enunciado .....	46
5.4.2 Representación gráfica .....	47
5.4.3 Movimiento .....	47
5.4.4 Visibilidad y apariencia .....	50
.....	51
5.5 Habitación de aire limpio .....	53
5.5.1 Enunciado .....	53
5.5.2 Representación gráfica .....	54
.....	54
5.5.3 Movimiento .....	54
5.5.4 Visibilidad y apariencia .....	55
<b>Capítulo 6. Conclusión .....</b>	<b>62</b>



## Índice de figuras

Figura 1. PLC del laboratorio .....	10
Figura 2.Arquitectura de un PLC .....	10
Figura 3.Ciclo de trabajo de un API .....	12
Figura 4.Maqueta de Lucas-Nülle.....	12
Figura 5.Placa base de la maqueta.....	13
Figura 6.HMI.....	13
Figura 7.Diagrama GRAFCET .....	14
Figura 8.Como agregar el bloque Función .....	15
Figura 9.Eschema de la transducción de las variables.....	16
Figura 10.Transducción de variables.....	16
Figura 11.Variables cifradas .....	16
Figura 12.First Scan manual .....	17
Figura 13.Elementos para dibujar en el HMI.....	17
Figura 14.Diferentes tipos de botones.....	17
Figura 15.Ejemplo de representación de un proceso en el HMI.....	18
Figura 16.Ejemplo de plantilla que podemos encontrar en el laboratorio .....	18
Figura 17.Tabla de variables del HMI.....	19
Figura 18.Escalado lineal .....	19
Figura 19.Pestaña de Eventos de un botón .....	20
Figura 20.Opciones de funciones de la ventana eventos.....	20
Figura 21.Opciones de la pestaña 'Procesamiento por bits' .....	20
Figura 22.Opciones del ajuste de Visualización .....	20
Figura 23.Ajustes en la Apariencia .....	21
Figura 24.Ajustes en la Visibilidad.....	21
Figura 25.Ventana para ajustar el movimiento.....	21
Figura 26.Ajuste de Movimiento de un Objeto.....	22
Figura 27.Sensibilidad en el movimiento por el rango de la variable .....	22
Figura 28.Temporizador que rige el movimiento.....	22
Figura 29.Contador CTUD que controla el movimiento.....	23
Figura 30.Temporizador en bucle vinculado al contador CTUD.....	23
Figura 31.Temporizador en bucle con fallo por ciclo de scan .....	24
Figura 32.Temporizador en bucle sin fallo por ciclo de Scan .....	24
Figura 33.Plantilla de la plataforma de cambio de mercancía que encontramos en el laboratorio.....	26
Figura 34.Representacion gráfica de la plataforma de cambio de mercancía realizada en el HMI.....	27
Figura 35.Superposicion de objetos en la representación gráfica .....	27
Figura 36.Transducción de las variables.....	28
Figura 37.Tabla de variables del HMI .....	28
Figura 38.Temporizador del motor H2.....	29
Figura 39.Temporizador del motor H1.....	29
Figura 40.Contador CTUD que controla el movimiento vertical .....	29
Figura 41.Configuración del movimiento .....	30
Figura 42.Posición inicial y final de la caja .....	30
Figura 43.Configuración del final de carrera S14 .....	30
Figura 44.Configuración del final de carrera S13 .....	31
Figura 45.Configuración del movimiento horizontal de la grúa.....	31
Figura 46.Configuración de los botones.....	31
Figura 47.Visibilidad de la rama izquierda de la plataforma.....	32
Figura 48.Configuración de la visibilidad de la rama izquierda de la plataforma .....	32



Figura 49.Visibilidad de la parte derecha de la plataforma .....	32
Figura 50.Visibilidad en la parte central de la plataforma .....	32
Figura 51.Visibilidad de la Caja 1.....	33
Figura 52.Vista de la caja 1 durante el proceso .....	33
Figura 53.Visibilidad de la caja 2 .....	33
Figura 54.Vista de la caja 2 durante el proceso .....	33
Figura 55.Visibilidad de la caja 3 .....	34
Figura 56.Vista de la caja 3 durante el proceso .....	34
Figura 57.Visibilidad de la caja 4 .....	34
Figura 58.Vista de la caja 4 durante el proceso .....	34
Figura 59.Visibilidad de la caja 5 .....	35
Figura 60.Vista de la caja 5 durante el proceso .....	35
Figura 61.Ajuste de apariencia de la señal de final de carrera .....	35
Figura 62.Configuración del final de carrera.....	36
Figura 63.Plantilla de la dobladora que encontramos en el laboratorio .....	37
Figura 64.Representación gráfica de la dobladora en el HMI.....	37
Figura 65.Contadores utilizados para describir el movimiento en la dobladora .....	38
Figura 66.Cilindro empujando la barra .....	38
Figura 67.Ajuste del rango de la variable.....	38
Figura 68.Vista del estado inicial de la barra .....	39
Figura 69.Visibilidad del estado inicial de la barra.....	39
Figura 70.Vista de la barra doblándose.....	39
Figura 71.Visibilidad de la barra doblándose .....	40
Figura 72.Vista de la barra cuando el cilindro está totalmente extendido.....	40
Figura 73.Visibilidad de la barra para cuando el cilindro está totalmente extendido .....	40
Figura 74.Vista de la barra doblándose por el cilindro 3 .....	41
Figura 75.Visibilidad de la barra doblándose por el cilindro 3.....	41
Figura 76.Vista de la barra en su posición final.....	41
Figura 77.Visibilidad de la barra en su posición final.....	42
Figura 78.Plantilla del semáforo de obra que encontramos en el laboratorio.....	42
Figura 79.Representacion del semáforo de obra en el HMI .....	43
Figura 80.Temporizador del que depende el movimiento.....	43
Figura 81.Variable que activa el temporizador .....	44
Figura 82.Escalado lineal .....	44
Figura 83.Configuración del movimiento: Variable y Rango.....	44
Figura 84.Visibilidad de los coches en estado de espera .....	45
Figura 85.Vista de los coches en estado de espera.....	45
Figura 86.Visibilidad de los coches cuando están cruzando la zona de obras.....	45
Figura 87.Vista de los coches cuando están cruzando la zona de obras.....	46
Figura 88.Plantilla de la Taladradora que encontramos en el laboratorio .....	47
Figura 89.Representación de la taladradora en el HMI .....	47
Figura 90.Contadores CTUD de los que depende el movimiento .....	48
Figura 91.Tipos de interacción entre un cilindro y un objeto .....	48
Figura 92.Ajustes del caso ambos elementos están en contacto desde el principio del movimiento .....	48
Figura 93.Ajustes del caso en el que los elementos no están en contacto desde el principio del movimiento .....	49
Figura 94.Temporizador del que depende el movimiento.....	49
Figura 95.Ajuste del movimiento basado en temporizadores: Variable y rango.....	49
Figura 96.Vista de la pieza inicial .....	50
Figura 97.Visibilidad de la pieza inicial.....	50
Figura 98.Vista de la pieza perforada.....	51
Figura 99.Visibilidad de la pieza perforada .....	51



Figura 100. Vista de la pieza movida por el aire comprimido .....	51
Figura 101. Visibilidad de la pieza movida por el aire comprimido .....	52
Figura 102. Visibilidad de la pieza inicial.....	52
Figura 103. Plantilla de la habitación de aire limpio que encontramos en el laboratorio.....	53
Figura 104. Representación de la habitación de aire limpio en el HMI .....	54
Figura 105. Contadores CTUD de los que depende el movimiento .....	54
Figura 106. Ajuste para que las barreras de luz sean normalmente cerradas.....	55
Figura 107. Configuración de los botones de las barreras de luz .....	55
Figura 108. Vista de la persona esperando a entrar .....	55
Figura 109. Visibilidad de la persona esperando a entrar .....	56
Figura 110. Vista de la persona cruzando el haz de luz B1 .....	56
Figura 111. Visibilidad de la persona cruzando el haz de luz B1.....	56
Figura 112. Vista de la persona en el centro de la esclusa .....	57
Figura 113. Vista de la persona cruzando el haz de luz mientras hay una persona en el centro de la esclusa .....	57
Figura 114. Visibilidad de la persona en el centro de la esclusa.....	58
Figura 115. Vista de la persona cruzando el haz de luz B2 .....	58
Figura 116. Visibilidad de la persona cruzando el haz de luz B2.....	58
Figura 117. Vista de la persona que ha cruzado la puerta.....	59
Figura 118. Vista de la persona cruzando el haz de luz mientras que otra ha cruzado la puerta .....	59
Figura 119. Visibilidad de la persona que ha cruzado la puerta .....	59
Figura 120. Momento en el que la persona que ya ha cruzado la puerta se oculta.....	60
Figura 121. Configuración de la variable para que la persona que ha cruzado la puerta se oculte .....	60
Figura 122. Vista del haz de luz ocultándose .....	60
Figura 123. Visibilidad del haz de luz .....	61



## Capítulo 1. Introducción

El periodo de pandemia del COVID-19 resaltó un problema que tiene la asignatura de Automatización Industrial. Este problema es que la parte práctica de la asignatura, basada en programar PLCs a través del programa TIA Portal, solo puede ser realizada en el laboratorio de la universidad, ya que los alumnos no disponen del material necesario en sus casas para poder llevarlas a cabo.

Los materiales utilizados en las prácticas son los siguientes:

- PLC
- Universal Application Simulator (Maqueta)
- Plantillas simuladoras de los procesos
- Licencias de TIA Portal

No son materiales que los alumnos suelen tener en casa y la compra de estos por parte del alumno no es viable ya que cada PLC ronda los 400 € y las maquetas 2500€. Además, por el mismo motivo, en la universidad contamos con una cantidad limitada de dispositivos, problema que se manifiesta incluso sin estar en pandemia.



## Capítulo 2. Objetivo

El objetivo del presente trabajo es desarrollar entornos virtuales para poder realizar prácticas docentes virtuales simulando procesos que sean similares a los utilizados en prácticas de asignaturas de automatización industrial desde casa sin necesidad de usar componentes físicos como los PLCs o las maquetas de simulación.

El uso de licencias itinerantes soluciona la problemática de las licencias necesaria para usar TIA Portal, por lo que desarrollaremos los entornos virtuales bajo estas licencias.

Este proceso de virtualización de las maquetas implica que tanto los PLCs como los simuladores de procesos industriales utilizados en las practicas, dejan de ser necesarios para la realización de estas, por lo que le supone un ahorro económico a la universidad y permite a los alumnos llevar a cabo las practicas fuera del laboratorio.





## Capítulo 3. Material

### 3.1 TIA Portal

*Totally Integrated Automation* es la plataforma de ingeniería de Siemens que ofrece soluciones de automatización en todos los sectores industriales del mundo, integrando todas las tareas de automatización de un proceso industrial. Se trata de una aplicación modular a la que se le pueden ir añadiendo nuevas funcionalidades según las necesidades concretas de cada sector.

Vivimos en un mundo en el que las plantas y procesos industriales son cada vez más complejos. Cada vez más sistemas están conectados en red, y como consecuencia de esto el número de herramientas de ingeniería crece exponencialmente y cada una de ellas es diferente de la otra. Con lo cual, la complejidad para los técnicos es mucho mayor.

Para intentar solucionar esta complejidad, Siemens creó hace 10 años TIA Portal con la idea de combinar todas estas herramientas en una.

TIA Portal incluye:

- Lenguaje de programación para PLCs (autómatas programables) de Siemens STEP 7.
- Software para control de periféricos.
- Software WINCC para la visualización de procesos y dispositivos. Pantallas táctiles.
- Start Drive. Control de servomotores.
- Scout TIA. Motion Control.

Como cualquier programa de Siemens, para poder utilizar TIA Portal necesitamos tener una licencia que nos permita usar el programa.

### 3.2 PLC

#### 3.2.1 Definición

Un autómata programable industrial es un equipo electrónico, programable en lenguaje no informático, diseñado para controlar en tiempo real y en ambiente de tipo industrial procesos secuenciales.

El autómata programable, también conocido como PLC (Programmable Logic Controller), se trata de un computador especial, tanto en el software como en el hardware.

En el software, porque se programa en un lenguaje especial diseñado específicamente para generar de forma sencilla el programa que implementa el algoritmo de control de procesos secuenciales (de sistemas de eventos discretos), y porque el algoritmo de control programado es ejecutado de forma periódica en un ciclo temporal que es lo bastante breve como para poder controlar los procesos en tiempo real.

En el hardware, porque utiliza componentes robustos que soportan condiciones de trabajo adversas, como las que se dan en ambientes industriales (polvo, temperatura, vibraciones, etc.), y porque su constitución física incluye los circuitos de interfaz necesarios para conectarlo de forma directa a los sensores y actuadores del proceso.



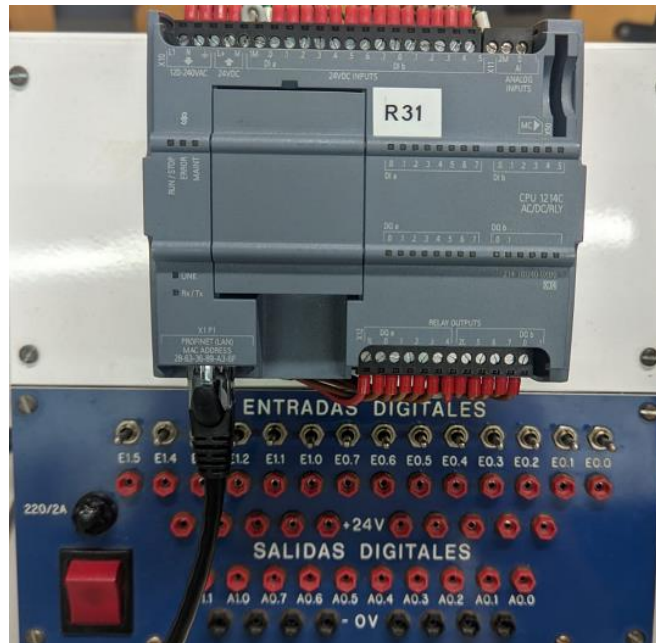


Figura 1. PLC del laboratorio

### 3.2.2 Arquitectura de un PLC

La figura 2 muestra de forma esquemática la arquitectura interna de un autómata programable industrial típico. Como computador que es, tiene un procesador que es el que ejecuta el programa almacenado en la memoria de programa. La memoria de programa y la de datos están físicamente separadas, constituyendo una arquitectura tipo Harvard. Además, la memoria de datos está separada en dos tipos, que en la figura se denominan memoria de datos y memoria interna. Esta última se utiliza para almacenar los valores de las señales de entrada y salida, por lo que están conectadas con los módulos de entradas y salidas, que son los elementos de interfaz donde se conectan los sensores y actuadores del proceso. También dispone de periféricos para comunicar con otros dispositivos, como pantallas táctiles, ordenadores u otros autómatas.

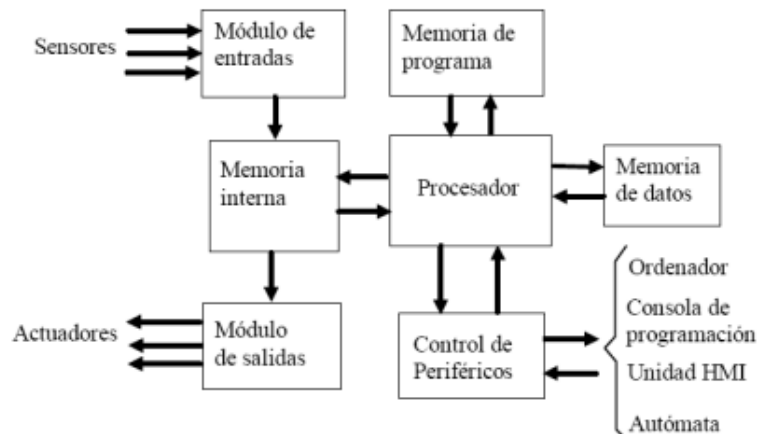


Figura 2. Arquitectura de un PLC

### 3.2.3 Bloques de memoria en un PLC

A continuación, se describen los bloques de memoria que suelen tener los autómatas programables industriales, incluyendo el tipo de memoria física y su función:

- Memoria de programa. Contiene el programa (instrucciones) que se ejecutan en el procesador. Se puede dividir en dos partes:



- Una parte ROM que contiene el programa monitor (para comunicar el autómata con los módulos de programación). Este programa monitor es fijo.
  - Una parte RAM con batería (puede ser también FLASH EEPROM) en la que se almacena el programa del usuario, que implementa el algoritmo de control del proceso. Evidentemente, este programa se mantiene aunque se desconecte el autómata.
- Memoria interna. Almacena los valores de entradas y salidas, además de otras variables internas del autómata. En ella se almacenan variables de 1 solo bit, es decir, variables que, aunque estén organizadas en bytes (grupos de 8 bits), se puede acceder a cada uno de los bits de forma independiente para leer o escribir. En esta zona de memoria se leen los valores de las entradas (donde están conectados los sensores), y se escriben los valores de las salidas (donde están conectados los actuadores).
  - Memoria de datos. Contiene datos de configuración o parámetros de funcionamiento del autómata y del proceso, o datos de propósito general. En ella se almacenan variables tipo byte (8 bits) o word (16 bits).

Tanto la memoria interna como la de datos suelen tener una parte de RAM normal (volátil) y una parte de RAM con batería o EEPROM para almacenar datos que no se deben perder cuando se desconecta el autómata.

### 3.2.4 Descripción funcional

El API es un computador, y por lo tanto, su funcionamiento consiste en la ejecución de un determinado programa almacenado en la memoria de programa. Se puede distinguir dos modos de funcionamiento: modo de programación y modo de ejecución.

- Modo de programación. En este modo de funcionamiento el programa monitor que reside en la parte ROM de la memoria de programa comunica el API con el elemento de programación (normalmente un PC), para que desde éste se le trasvase el programa que implementa el algoritmo de control que se desea ejecutar. En este modo, el API no está controlando el proceso.
- Modo de ejecución (modo RUN). En este modo, se ejecuta el programa del usuario que implementa al algoritmo de control, con lo que el autómata controla el proceso. En este modo, cuando se inicializa el autómata, el programa monitor salta a la dirección donde está el programa de control, y se empieza a ejecutar éste. La ejecución del programa de control se lleva a cabo de forma cíclica, ejecutando ciclos de scan de forma indefinida.

Se llama ciclo de trabajo (o de scan) al conjunto de tareas que el autómata lleva a cabo (instrucciones que se ejecutan) de forma cíclica cuando está controlando un proceso. El ciclo de trabajo más habitual tiene la estructura de la figura 3.



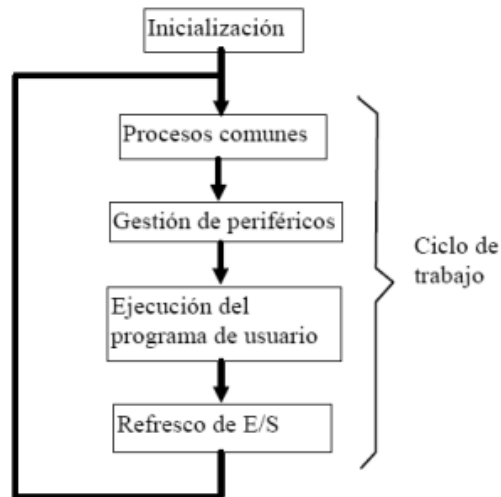


Figura 3. Ciclo de trabajo de un API

### 3.3 Maquetas Lucas-Nülle

Dispositivo que combinado con un PLC sirven para simular procesos industriales mediante el encendido y apagado de luces LED.

La maqueta está equipada con interruptores y pulsadores actuando como sensores, así como LEDs y un circuito electrónico para simular actuadores.

La maqueta es capaz de generar señales de retroalimentación automáticamente. Para simular cada proceso correctamente, primero deberemos colocar la plantilla correspondiente sobre la maqueta  
Características de la maqueta:

- 8 entradas digitales
- 20 salidas digitales
- 2 entradas analógicas
- 2 salidas analógicas
- 3 segmentos de luz móviles
- Conexiones en enchufes de seguridad de 4mm
- Conector 37-pin sub-D para la conexión con el PLC (señales de entrada)
- Enchufe de 37-pin sub-D para la conexión con el PLC (señales de salida)
- Tensión y corriente de funcionamiento: 24 V DC / 0.3 A
- Tamaño: 456 x 297 x 125 mm
- Peso : 1 kg

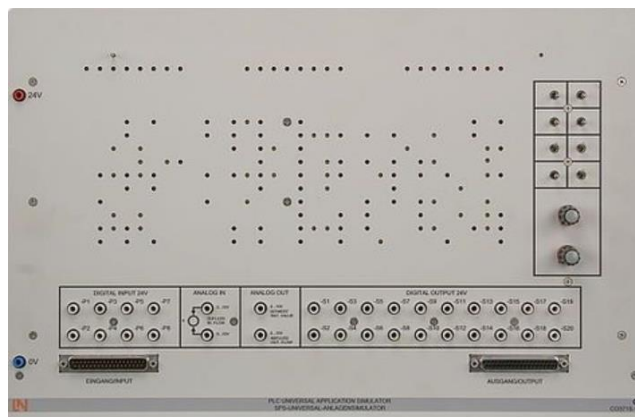


Figura 4. Maqueta de Lucas-Nülle



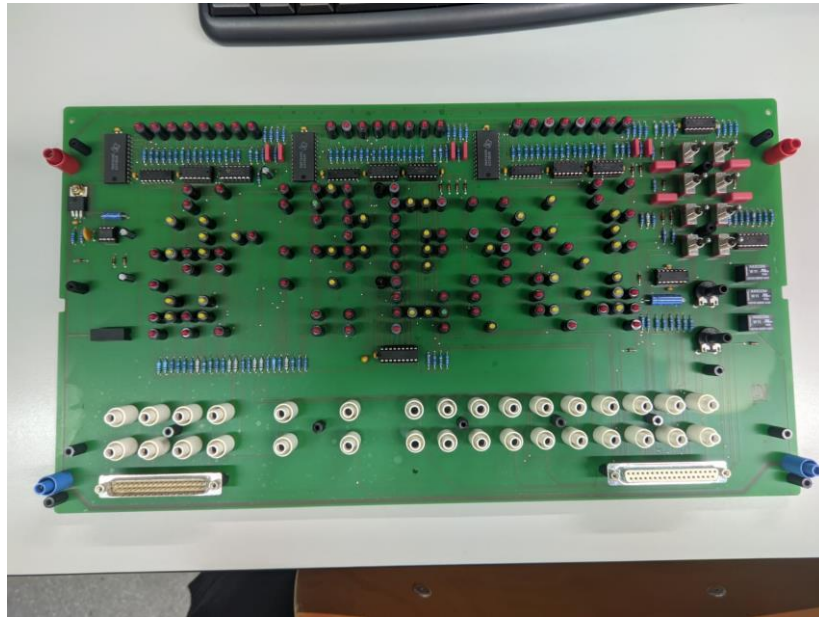


Figura 5. Placa base de la maqueta

### 3.4 HMI

La Interfaz Hombre-Máquina (HMI, por sus siglas en inglés) es la interfaz entre la máquina procesadora y el operario. En esencia, es el panel de control del operario. Es la herramienta principal que los operarios y los supervisores de línea utilizan para coordinar y controlar procesos y máquinas industriales y de fabricación. Los HMI traducen complejas variables de proceso en información utilizable y procesable.

Los sistemas HMI se especializan en mostrar información operativa casi en tiempo real. Los gráficos visuales de proceso dan significado y contexto al estado del motor y de las válvulas, los niveles del tanque y otros parámetros del proceso. Los HMI dan una visión operativa del proceso y permiten su control y optimización mediante el ajuste de la producción y de los objetivos del mismo.

El propósito de los HMI es mostrar información operativa en tiempo real y de forma fácilmente comprensible. Los gráficos visuales de proceso dan significado y contexto al estado del motor y de las válvulas, los niveles del tanque, presión, vibración y otros parámetros del proceso. Los sistemas HMI le introducen en el proceso para que pueda controlarlo y optimizarlo ajustando la producción y los objetivos del mismo. Los HMI también ayudan a los gerentes y supervisores a mejorar el proceso proporcionando datos históricos y de tendencias sobre la eficiencia de las máquinas o la calidad del producto. La flexibilidad y las capacidades avanzadas del software HMI actual presentan nuevas y grandes oportunidades para mejorar la eficiencia de las máquinas y los procesos.

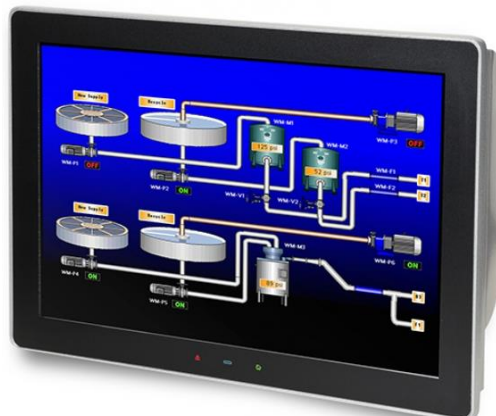


Figura 6. HMI



## Capítulo 4. Metodología

### 4.1 Metodología de las prácticas de Automatización Industrial

La metodología de prácticas de Automatización Industrial consiste en una previa explicación de cómo implementar los conceptos teóricos en el programa TIA Portal necesarios para llevar a cabo la práctica, como puede ser el uso de contadores, temporizadores o comparadores.

A continuación, elaboramos un diagrama etapa-transición (GRAFCET) del proceso industrial en el que estamos trabajando. GRAFCET es un modelo de representación gráfica, de los sucesivos comportamientos de un sistema lógico, predefinido por sus entradas y salidas. Es un diagrama funcional normalizado que permite hacer un modelo del proceso a automatizar, contemplando entradas, acciones a realizar, y los procesos intermedios que provocan estas acciones.

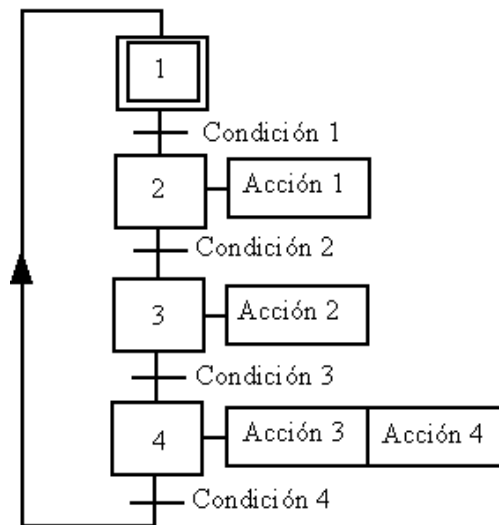


Figura 7. Diagrama GRAFCET

Esta forma de representación del automatismo de alto nivel es muy útil para explicar el funcionamiento del proceso y facilitar el diseño. La implementación final, sin embargo, se hace generalmente con la representación de contactos, por lo que es necesario traducir estos diagramas a esa forma de representación de bajo nivel. Este proceso es el que realizamos a continuación en la práctica, debemos traducir el GRAFCET al lenguaje del TIA Portal.

Una vez hemos traducido el GRAFCET y tenemos el programa del proceso industrial en el TIA Portal, debemos conectar el programa al PLC. La conexión entre PC y PLC se hace a través de un cable Ethernet que permite la comunicación entre ambos dispositivos.

El PLC a su vez debemos conectarlo a las maquetas de Lucas Nülle para poder observar el proceso industrial. El PLC ejecuta el programa que tiene escrito y activa o desactiva los actuadores del proceso según la información proporcionada por los sensores e interruptores del sistema. Esta información se la proporciona la maqueta, la cual es capaz de generar esta información según el estado de los actuadores.

Por lo que debemos conectar las salidas del PLC a las entradas de la maqueta y las salidas de la maqueta a las entradas del PLC para que estos trabajen en conjunto. Las salidas del PLC representan los actuadores del proceso (motores, pistones, válvulas, etc.) y las salidas de la maqueta los sensores e interruptores.

Para la correcta visualización del proceso debemos poner las plantillas del proceso sobre la maqueta.



## 4.2 Metodología del proceso de virtualización de las prácticas

### 4.2.1 PLC

Como cualquier programa de Siemens, para poder utilizar TIA Portal necesitamos disponer de una licencia que nos permita utilizar el programa. Para ellos, la universidad dispone de licencias itinerantes que permiten el uso de los programas de Siemens que vamos a necesitar para realizar este trabajo (TIA Portal y PLCSIM). Las licencias itinerantes son un tipo de licencia que permiten a los usuarios disfrutar de las licencias sin necesidad de tenerla en su dispositivo, si no que se conectan a un servidor en el que están alojadas las licencias permitiendo utilizar sus programas. Este sistema es el ideal para la universidad ya que permite a los alumnos utilizar los programas desde cualquier dispositivo reduciendo así el número de licencias necesita comprar. El inconveniente de las licencias itinerantes es que, si el número de usuarios simultáneos utilizando las licencias es mayor que el número de licencias contratadas, no todos dispondrán de ellas.

El fin de este trabajo es que el alumno sea capaz de realizar las prácticas sin necesidad de ir al laboratorio, por lo que necesitamos herramientas para suplir los materiales que tenemos en el laboratorio y no disponemos en casa, como son los PLCs y las maquetas. Para simular el PLC, utilizamos el programa PLCSIM, que es un programa que permite la simulación de autómatas en cualquier dispositivo.

Tras la virtualización de las prácticas, el alumno debe observar los mismos resultados del programa que ha diseñado tanto en las maquetas del laboratorio como en el HMI, por lo que necesitamos crear un programa que junto al HMI sean capaces de replicar a las maquetas del laboratorio. Para crear este programa utilizamos el bloque 'Función'. El bloque 'Función' nos permite crear un programa que trabaje en conjunto con el que diseñe el alumno y además permite protegerlo mediante una contraseña.

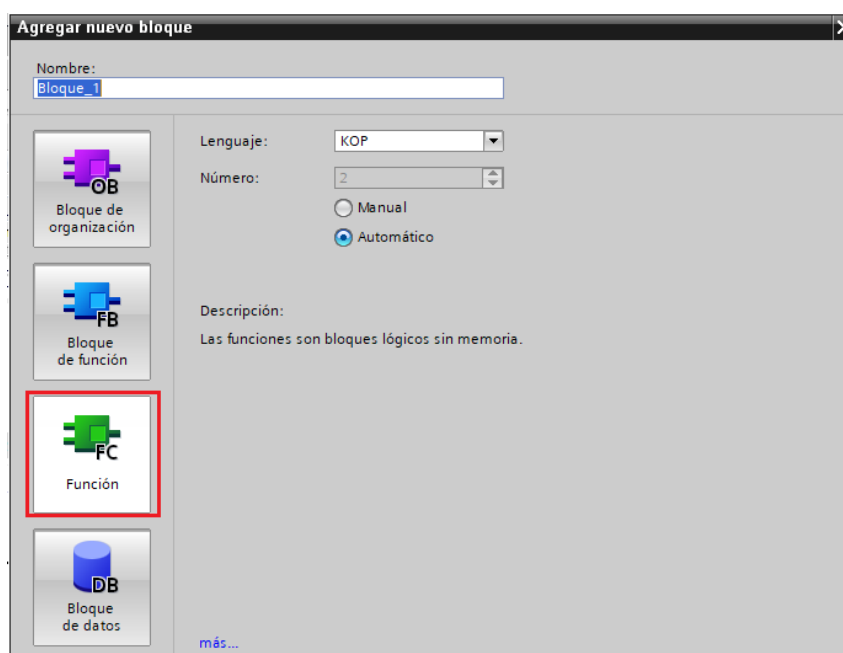


Figura 8. Como agregar el bloque Función

La idea es que cuando el alumno se disponga a realizar una práctica, abra el archivo de TIA Portal que contenga el bloque 'Función' correspondiente a la práctica. El alumno tendrá que realizar el programa que describa el proceso y mediante el HMI observará lo que ha programado.



Por lo tanto, en el mismo archivo de TIA Portal, tendremos dos bloques de programa, el del alumno y el interno, lo que implica dos tablas de variables. Tanto el programa interno como las variables internas no deben ser modificadas por el alumno, por ello es por lo que protegemos el bloque mediante contraseña.

El bloque 'Función' debe contener todas las operaciones lógicas necesarias para que el proceso pueda ser representado por el HMI (gestión de finales de carrera, movimiento, visibilidad de objetos...).

Al tener que representar el mismo proceso, las tablas de variables (la interna y la del alumno) tendrán variables que representen lo mismo (el mismo motor, el mismo sensor...). El objetivo es que el alumno mediante el HMI controle el proceso y vea el desarrollo del mismo. Si el alumno pulsa un botón en del HMI, el programa del alumno reaccionara a ello y provocara cambios en el estado de sus salidas, cambios que serán representados por el HMI, por lo que el esquema de variables queda así:

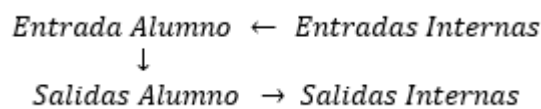


Figura 9. Esquema de la transducción de las variables

Las entradas del programa interno forzarán el estado de las entradas del programa alumno. El programa alumno, tras realizar sus operaciones lógicas, modificará el estado de las salidas del programa alumno. Estos cambios deben ser representados por el HMI, por lo que las salidas del programa alumno indicarán el estado de las salidas del programa interno.

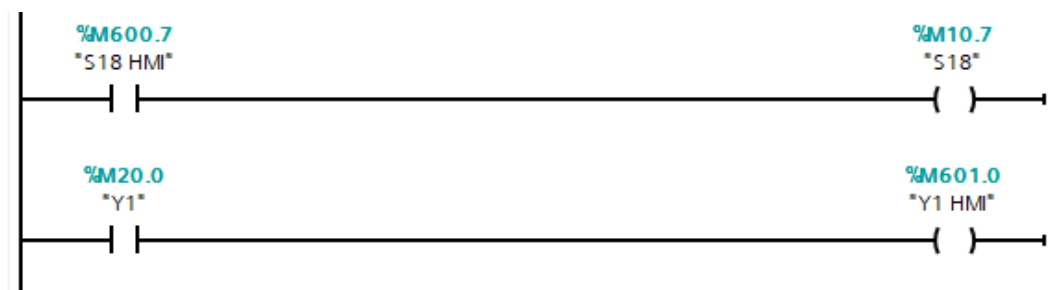


Figura 10. Transducción de variables

Ya que el alumno tiene acceso a las variables internas, hemos modificado los nombres de estas variables para que el alumno no las confunda con sus propias variables. En caso de tener que modificar el programa, en la opción 'Comentario' de la tabla de variables, hemos indicado de que variable se trata.

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Comentario
1	[-] _Entrada_1	Bool	%M600.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S1 Interno
2	[-] _Entrada_2	Bool	%M600.1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S2 Interno
3	[-] _Actuador_1	Bool	%M600.2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Y1 H6 Interno
4	[-] _Actuador_2	Bool	%M600.3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Y2 H5 Interno
5	[-] _Actuador_3	Bool	%M600.4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Y3 H3 Interno
6	[-] _Actuador_4	Bool	%M600.5		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Y4 H4 Interno
7	[-] _Actuador_5	Bool	%M600.6		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Y5 H2 Interno
8	[-] _Actuador_6	Bool	%M600.7		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Y6 H1 Interno
9	[-] _Entrada_3	Bool	%M601.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S13 Interno
10	[-] _Entrada_4	Bool	%M601.1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S14 Interno
11	[-] _Entrada_5	Bool	%M601.2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S15 Interno
12	[-] _Entrada_6	Bool	%M601.3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S16 Interno
13	[-] _Entrada_7	Bool	%M601.4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S17 Interno
14	[-] _Entrada_8	Bool	%M601.5		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S18 Interno

Figura 11. Variables cifradas





Al no poder contener elementos que el alumno pueda modificar, el bloque 'Función' necesita crear su propio First Scan sin utilizar la opción 'First Scan' del PLC. El First Scan creado es el siguiente, donde el bit 1000.2 tendrá un flanco positivo en el segundo ciclo de scan, por lo que nos sirve como sustituto del First Scan original.

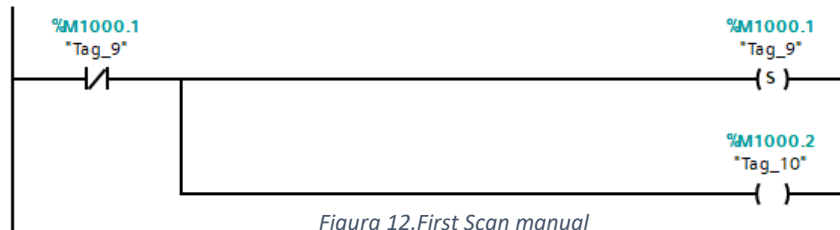


Figura 12. First Scan manual

#### 4.2.2 HMI

Para sustituir a las maquetas emplearemos la herramienta HMI que nos proporciona TIA Portal. Esta herramienta nos permite configurar un HMI y simularlo, lo que permite la visualización del proceso desde el mismo ordenador.

Para configurar el HMI debemos de representar gráficamente el proceso, configurar las variables propias del HMI, configurar los botones necesarios para controlar el proceso y configurar el movimiento, la visibilidad y la apariencia de los objetos.

##### 4.2.2.1 Representación gráfica

Debemos representar de forma aproximada el proceso y añadir los botones e interruptores necesarios para controlarlo, así como indicadores de finales de carrera y luces que indiquen el estado de los actuadores.

Para la representación del proceso usaremos las herramientas de dibujo de TIA Portal.

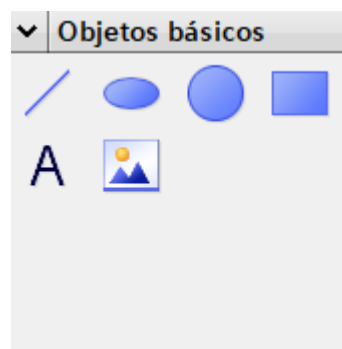


Figura 13. Elementos para dibujar en el HMI

Estas herramientas de dibujo son muy básicas por lo que en algunos procesos necesitaremos de imágenes para poder representar mejor el proceso. Estas imágenes deben de ser imágenes en 2D y con fondo transparente, condición que limita mucho la representación de ciertos objetos.

Para la representación de botones e interruptores utilizamos la librería de TIA Portal 'Buttons and switches' y elegimos entre las diferentes opciones que nos ofrece la librería



Figura 14. Diferentes tipos de botones



Para la representación de finales de carrera y de luces que indican el estado de los actuadores utilizamos los círculos de las herramientas básicas de dibujo a los que les añadiremos animaciones que nos indicaran el estado del proceso.

En todas las maquetas hemos añadido un botón llamado ‘Pulsador extra’, esto es debido a que el alumno es libre de hacer cualquier mejora al proceso y para que pueda hacerlo le proporcionamos este botón.

Para la práctica ‘Plataforma para cambio de mercancía’, la representación sería la siguiente, siendo la primera imagen la representación gráfica en el HMI y la segunda la representación del proceso dada por la plantilla usada en prácticas.

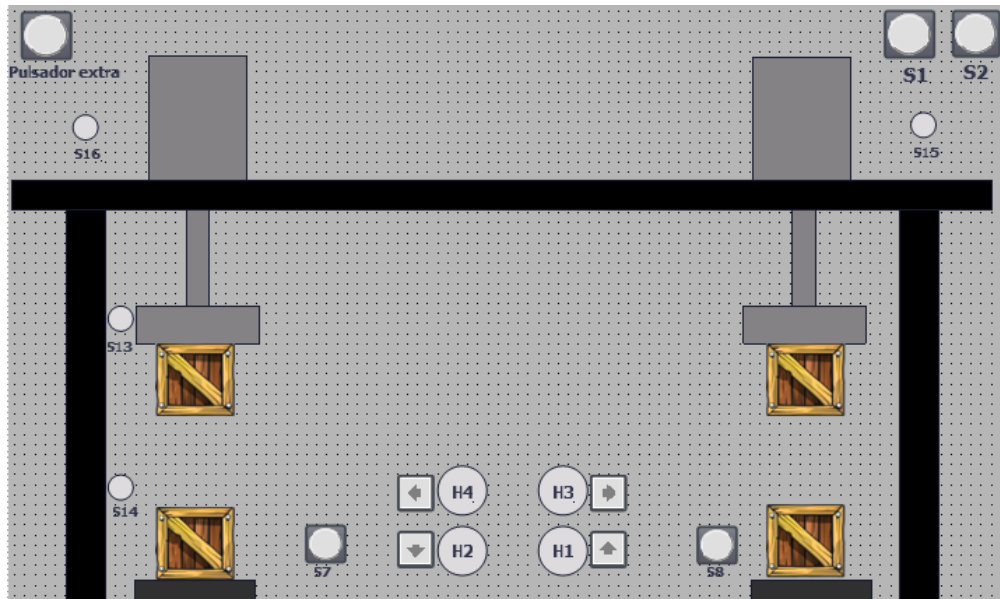


Figura 15. Ejemplo de representación de un proceso en el HMI

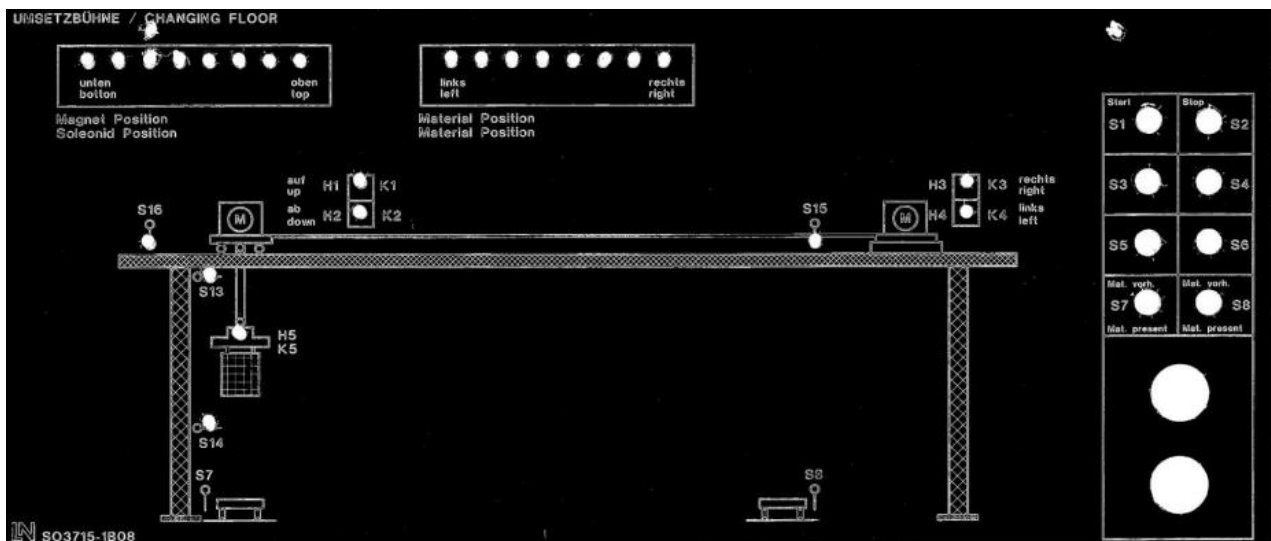


Figura 16. Ejemplo de plantilla que podemos encontrar en el laboratorio



Podemos apreciar ciertas diferencias entre la plantilla de prácticas y la representación hecha en el HMI. Estas diferencias se deben a que debido a las limitaciones de TIA Portal para establecer el movimiento, necesitamos crear varias veces el mismo objeto e ir jugando con la visibilidad para poder representar el proceso.

#### 4.2.2.2 Variables HMI

Los movimientos y animaciones que llevara a cabo el HMI están sujetos a los valores de las variables del HMI. Vincularemos estas variables a las variables ‘internas’ del PLC que nos ayuden a describir el proceso.

Tabla de variables estándar					
Nombre ▲	Tipo de datos	Conexión	Nombre del PLC	Variable PLC	
Cilindro 1 Mov	Int	HMI_Conexión_1	PLC_1	"Movimiento Cilindro 1"	
Cilindro 2 Mov	Int	HMI_Conexión_1	PLC_1	"Movimiento Cilindro 2"	
Cilindro 3 Mov	Int	HMI_Conexión_1	PLC_1	"Movimiento Cilindro 3"	
Posicion de final	Bool	HMI_Conexión_1	PLC_1	"Posicion final"	
Pulsador extra	Bool	HMI_Conexión_1	PLC_1	"Pulsador extra interno"	
S1	Bool	HMI_Conexión_1	PLC_1	"S1 Interno"	
S13	Bool	HMI_Conexión_1	PLC_1	"S13 Interno"	
S14	Bool	HMI_Conexión_1	PLC_1	"S14 Interno"	
S15	Bool	HMI_Conexión_1	PLC_1	"S15 Interno"	
S16	Bool	HMI_Conexión_1	PLC_1	"S16 Interno"	
S17	Bool	HMI_Conexión_1	PLC_1	"S17 Interno"	
S18	Bool	HMI_Conexión_1	PLC_1	"S18 Interno"	

Figura 17. Tabla de variables del HMI

En variables en las que el movimiento se debe a una variable vinculada al tiempo es necesario aplicarle un escalado lineal para controlarla de forma más sencilla. Este escalado se configura en la pestaña ‘escalado lineal’ de la ventana propiedades que aparece si seleccionamos la variable. En la pestaña modificaremos los valores iniciales y finales tanto del PLC y del HMI a nuestro antojo.

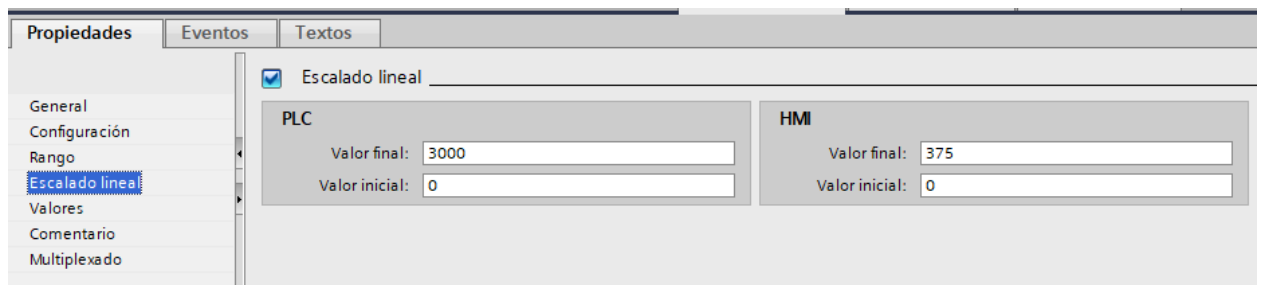


Figura 18. Escalado lineal

#### 4.2.2.3 Botones e interruptores

Antes hemos visto como introducir botones e interruptores en el HMI, pero ahora necesitamos configurarlos. Necesitamos controlar el proceso a través de estos elementos y para ellos deben de ser capaces de cambiar el valor de las variables del HMI.

Para llevar a cabo esto, seleccionaremos el elemento que queremos configurar y nos vamos a la pestaña ‘Eventos’ de la ventana ‘Propiedades’. En ella podemos observar una serie de acciones que podemos hacer sobre el botón, como pulsarlo o soltarlo, y a la derecha podemos configurar las consecuencias de las acciones que hagamos sobre el botón.

En la figura 20 podemos ver las posibilidades que nos da el programa, pero las que vamos a usar son las de la figura 21.



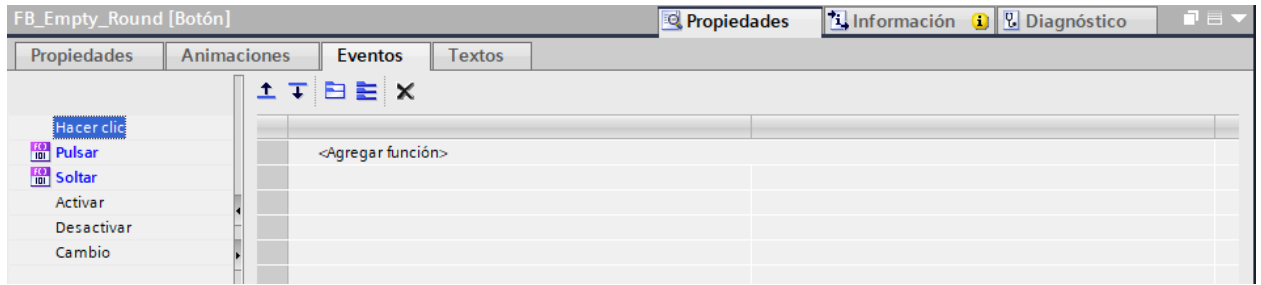


Figura 19. Pestaña de Eventos de un botón



Figura 20. Opciones de funciones de la ventana eventos

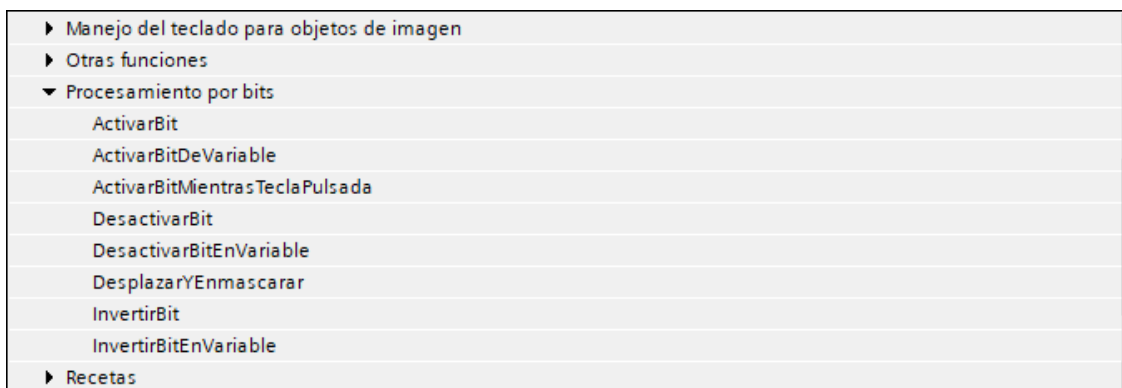


Figura 21. Opciones de la pestaña 'Procesamiento por bits'

#### 4.2.2.4 Apariencia y visibilidad

Para configurar la apariencia y la visibilidad de un objeto, seleccionamos el objeto y nos vamos a la opción 'Visualización' en la pestaña 'Animaciones' de la ventana 'Propiedades'.

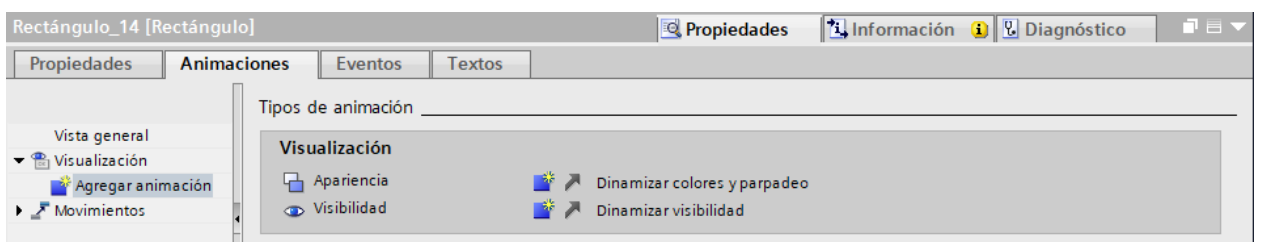


Figura 22. Opciones del ajuste de Visualización



Si queremos configurar la apariencia del objeto seleccionamos 'Apariencia' y seleccionamos la variable de la que dependerá la apariencia del objeto y para que valores de esta queremos que la apariencia cambie.

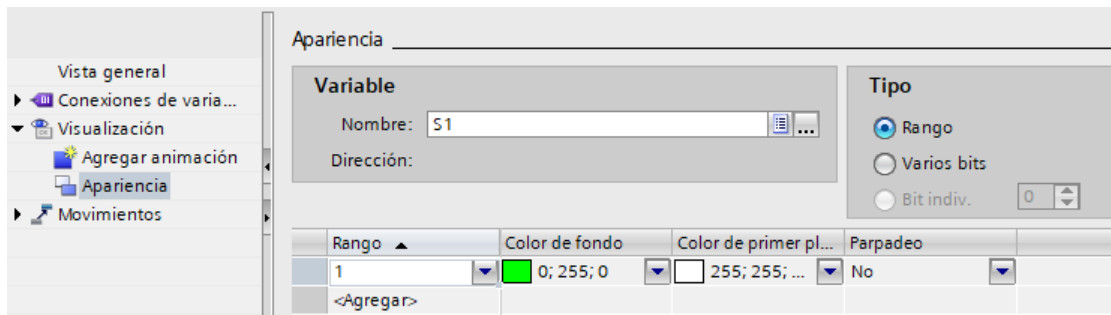


Figura 23. Ajustes en la Apariencia

Si queremos configurar la visibilidad del objeto seleccionamos 'Visibilidad' y seleccionamos la variable de la que dependerá la visibilidad del objeto y para que valores de esta queremos que la visibilidad cambie.

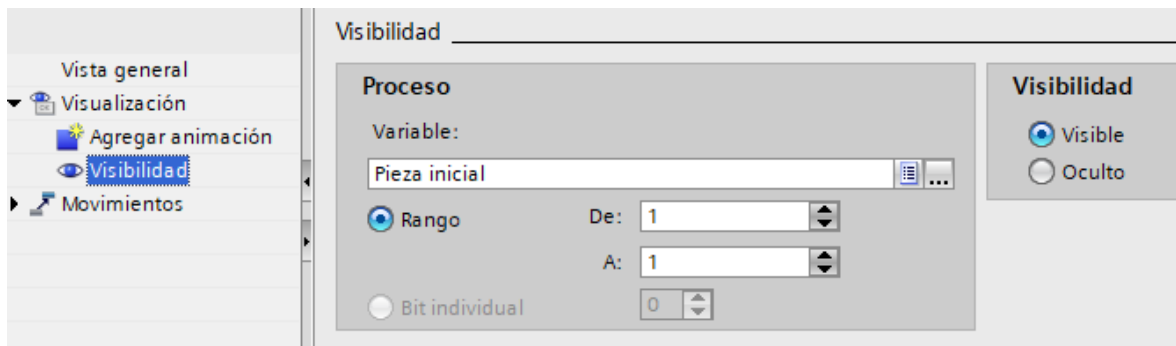


Figura 24. Ajustes en la Visibilidad

#### 4.2.2.5 Movimiento

Para añadirle movimiento a un objeto en el HMI debemos seleccionar un objeto e irnos a la pestaña 'Animaciones' de la ventana 'Propiedades'. En ella podemos elegir entre los diferentes movimientos que nos ofrece TIA Portal. El programa solo permite asignar un tipo de movimiento a un objeto, por lo que solo es posible que los objetos se muevan en un eje (x o y). Por lo que si queremos que el objeto se mueva en ambos ejes tenemos que jugar con la visibilidad del mismo para hacerlo posible.

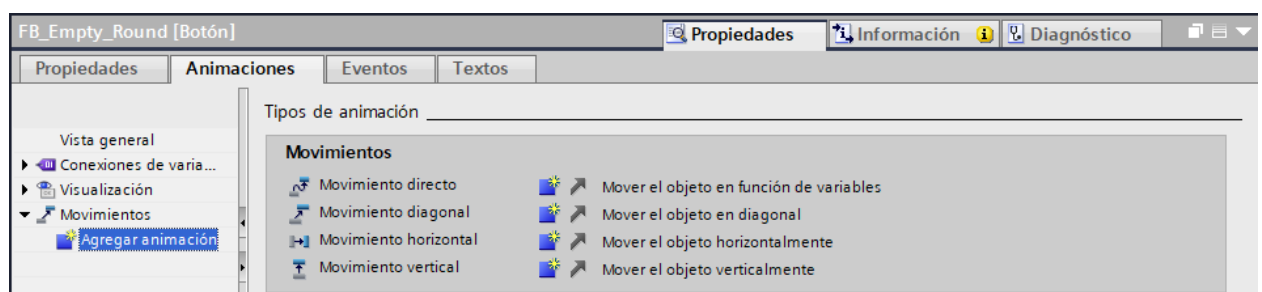


Figura 25. Ventana para ajustar el movimiento

Para configurar el movimiento de un objeto debemos establecer la posición inicial y final del mismo, asignarle una variable y un rango a esta variable. La variación del valor de la variable producirá el



movimiento. En el caso de la figura 26, cuando la variable tiene un valor de 0, el objeto se encuentra en su posición inicial mientras que, si tiene un valor de 15, la variable estará en su posición final. Para los valores intermedios de la variable, el programa establece una relación entre el rango de la variable y la distancia entre inicio y final del movimiento, es decir, si el valor de la variable es la mitad del valor del rango, la posición del objeto será la mitad de la distancia entre la posición final e inicial.

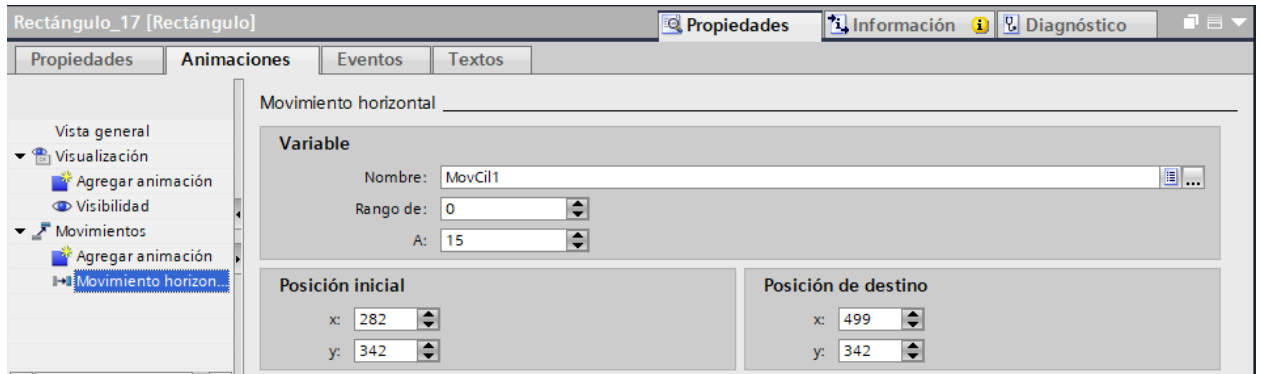


Figura 26. Ajuste de Movimiento de un Objeto

Cuanto mayor sea el rango de la variable mayor será la sensibilidad y fluidez tendrá el movimiento. En las siguientes figuras podemos apreciar como un proceso de rango 3 va a tener menor precisión a la hora de describir la posición de un objeto que uno rango 5.



Figura 27. Sensibilidad en el movimiento por el rango de la variable

La actualización del movimiento respecto a la variable no es instantánea, el HMI tiene una 'tasa de refresco' de las variables lo que hace que no todos los valores de la variable sean representados si la tasa de refresco es más lenta que la variación de la variable. Esta tasa de refresco no es siempre es la misma, lo que implica que el HMI no va a representar siempre el movimiento de la misma forma. Incrementar el rango de la variable hace que las diferencias entre los distintos movimientos representados por el HMI sean menores.

Como hemos explicado antes, el movimiento se genera por la variación de una variable que está vinculada a la posición de un objeto. Para configurar el movimiento de los procesos, hemos desarrollado dos métodos para controlar la variación de la variable de la que depende el movimiento, una basada en temporizadores y otra en contadores:

-Basada en temporizadores: la variable vinculada al movimiento será el valor ET (tiempo transcurrido desde que la entrada del temporizador detecta un flanco positivo) de un temporizador TON.

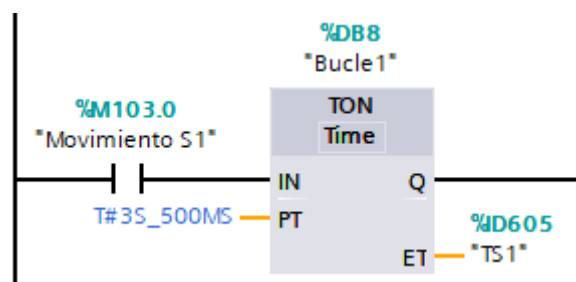


Figura 28. Temporizador que rige el movimiento



Un temporizador Ton comienza a cronometrar cuando el parámetro de entrada (IN) cambia de 0 a 1. Cuando el tiempo transcurrido (ET) alcanza el tiempo preestablecido (PT), el parámetro de salida (Q) cambia de 0 a 1.

Cuando el temporizador detecte un flanco positivo en la entrada el tiempo transcurrido se almacenará en la variable que le asignemos, y esta variable aumentara hasta que haya un flanco negativo en la entrada del temporizador, de esta forma generamos que la variable asignada a ET incremente su valor y generar movimiento.

Como antes hemos explicado, necesitamos aplicarles un escaldo lineal a las variables vinculadas al tiempo para que trabajar con ellas sea más sencillo.

Este método de generar movimiento es útil cuando queremos representar movimiento en una sola dirección, como, por ejemplo, representar el avance de un coche en una carretera. Por otra parte, este método presenta limitaciones si queremos representar un movimiento de dos direcciones, ya que no es capaz de disminuir la variable progresivamente, sino que lo hace de forma instantánea, de cierto valor a 0, por lo que no podemos usarlo para representar movimiento en dos direcciones.

-Basada en contadores: la variable vinculada al movimiento será la variable CV (valor del conteo) de un contador CTUD.

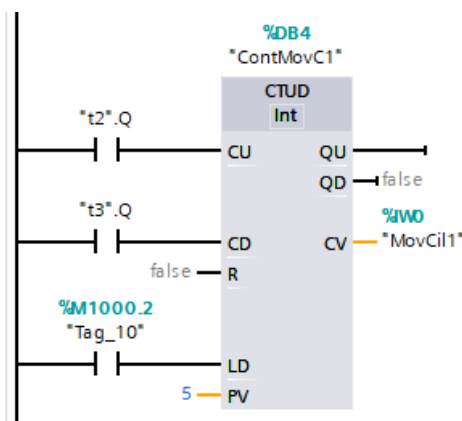


Figura 29. Contador CTUD que controla el movimiento

Un Contador CTUD empieza a contar adelante cuando se produce un flanco positivo en la entrada CU, y empieza a contar atrás cuando se produce un flanco positivo en la entrada CD. El valor actual CV del contador conserva el contaje actual. Los flancos en las entradas CU y CD lo proporcionan temporizadores en bucle configurados para que se ajusten a las necesidades del proceso.

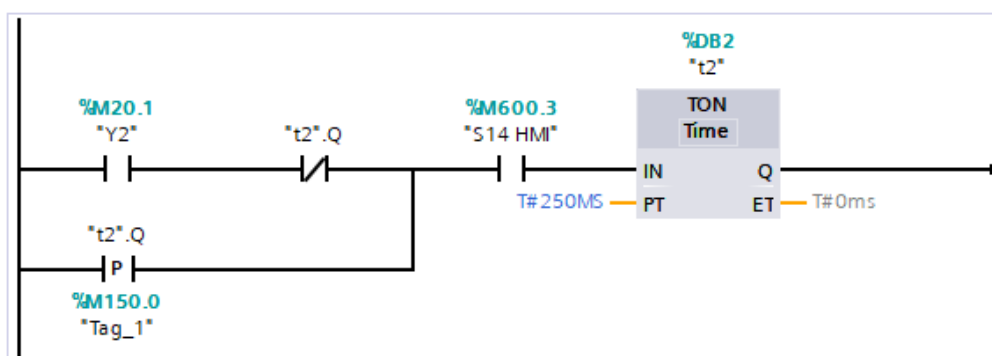


Figura 30. Temporizador en bucle vinculado al contador CTUD



Para explicar el funcionamiento de este método, seguiremos las figuras 29 y 30. Podemos observar que la señal de salida del temporizador 't2.Q' está conectada a la entrada CU del contador. El contador incrementará el valor de CV cada vez que en 't2.Q' haya un flanco positivo, por lo que necesitamos generar flancos positivos en 't2.Q' en bucle para que CV este continuamente incrementando y así generar movimiento.

Para generar el bucle en el temporizador debemos configurarlo como en la figura 31, en ella, 'Y2' representa el actuador que genera el movimiento en el proceso real y 'S14 HMI' representa el final de carrera (normalmente cerrado).

Cuando el actuador se activa, la entrada IN detecta un flanco y transcurrido el tiempo indicado en PT, la salida del temporizador será 1. El actuador Y2 permanece activo y conectado a la entrada IN, por lo que si queremos un generador de bucle en el temporizador debemos desconectarlo de la entrada IN una vez la salida del temporizador sea 1, por eso le conectamos en serie el valor negado de 't2.Q'.

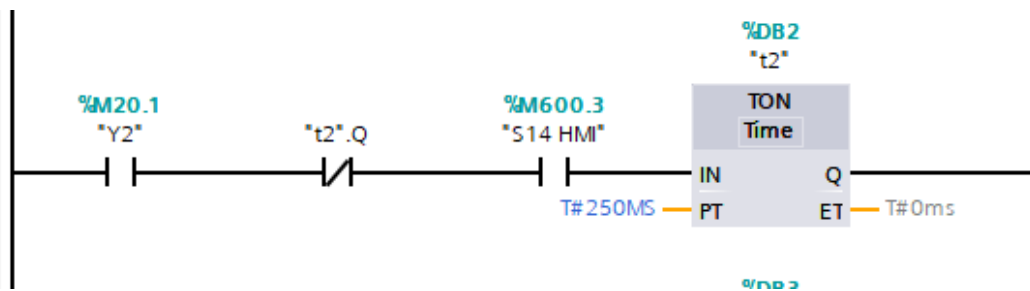


Figura 31. Temporizador en bucle con fallo por ciclo de scan

Al hacer esto, cuando la salida del temporizador sea 1, en el siguiente ciclo de scan se desactivará la entrada y salida del temporizador y en el siguiente, la entrada volverá a ser 1, por lo que el bucle está creado.

Esta configuración presenta un problema. El contador necesita que las señales de sus terminales de entrada duren al menos un ciclo de scan para poder procesarlas y la señal de salida de temporizador que se encuentra en bucle no dura un ciclo de scan, por lo que el contador no llega a procesar la señal y no el valor de CV no varía. Para corregir este problema añadimos en paralelo la señal 'flanco positivo' de salida del temporizador. Esto prolonga un ciclo de scan la duración de señal 't2.Q' cuando es 1, por lo que el contador procesará la señal y podrá modificar el valor de CV.

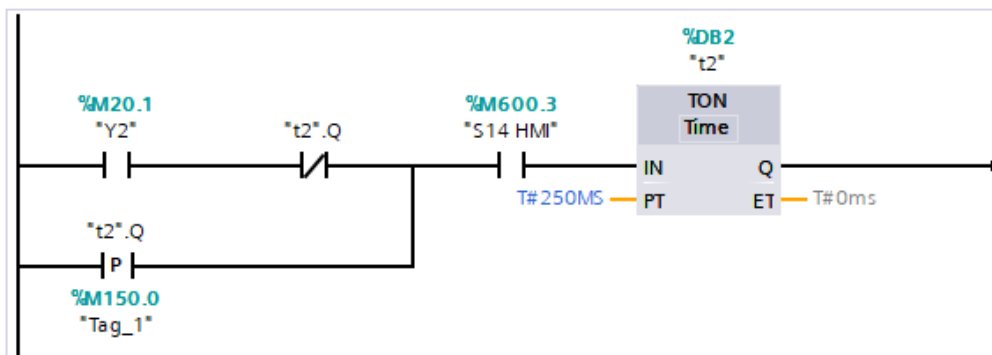


Figura 32. Temporizador en bucle sin fallo por ciclo de Scan





Este método es útil cuando queremos generar movimiento en dos sentidos. Cuando el actuador vinculado a la entrada CU está activo, el objeto avanzará, mientras que, si está activo el actuador de la entrada CD, el objeto retrocederá. Podemos ajustar la velocidad del movimiento variando el valor del tiempo TP de los temporizadores, cuanto menor sea el tiempo, más rápido ira el objeto.

Si la entrada LD del contador detecta un flanco positivo cargara el valor de PV en el conteo CV. Esto lo utilizaremos para establecer la posición inicial del objeto. A la entrada PV le asignaremos un valor que se sitúe dentro del rango establecido para el movimiento y conectaremos el 'first scan casero' a la entrada LD. En el primer ciclo de scan, PV cargara su valor en CV estableciendo la posición del objeto al inicio del sistema.



## Capítulo 5. Maquetas

Las maquetas que conforman las practicas de la asignatura 'Automatización industrial' son un total de 11. Para llevar a cabo este trabajo, hemos elegido las cinco maquetas más representativas y mas complejas en cuanto a recurso gráficos se refiere, por lo que el futuro desarrollo de los entornos virtuales de las maquetas restantes resultaría más sencillo.

### 5.1 Plataforma de cambio de mercancía

#### 5.1.1 Enunciado

Una plataforma de puente de grúa se utiliza para transferir piezas de trabajo desde una cinta transportadora, a otra. Está compuesto por un carro de desplazamiento y una grúa con un imán. La plataforma está en su estado inicial, cuando:

- el carro está parado en la posición de la izquierda;
- el imán está arriba;
- el imán está apagado.

El sistema se pone en funcionamiento con el botón S1 inicio en la consola del operador.

Tan pronto como una pieza de trabajo en la cinta 1 alcanza el final de carrera S7, el imán se baja por la activación de contactor K2 (H2).

Cuando se alcanza el final de carrera S14, el movimiento hacia abajo se detiene y se enciende el imán mediante K5 (H5).

Después de un tiempo de retardo de 2 segundos, el motor se acciona a través de K1 (H1) y la pieza de trabajo se levanta de la cinta.

Cuando se llega a final de carrera S13, el movimiento ascendente de la plataforma se detiene y el accionamiento del husillo se activa a través de K3 hacia la derecha.

Cuando el carro llega al final de carrera S15, se detiene y el imán se baja mediante K2 (H2).

Cuando se alcanza el final de carrera S14, el movimiento hacia abajo se detiene y la pieza de trabajo se posiciona en la segunda cinta desconectando el imán.

Cuando la señal S8 se activa, indicando que la pieza de trabajo está sobre la segunda cinta, se levanta el imán después de 2 segundos.

El movimiento ascendente se detiene cuando se activa el final de carrera S13 y el eje se mueve hacia la izquierda a través de K4 (H4).

El accionamiento del husillo se apaga cuando se alcanza el final de carrera S16 y el sistema se queda en su estado inicial.

La llegada de una pieza de trabajo con la activación de S7, hace comenzar el proceso de nuevo.

Debe ser posible apagar el sistema en cualquier momento con el botón S2.

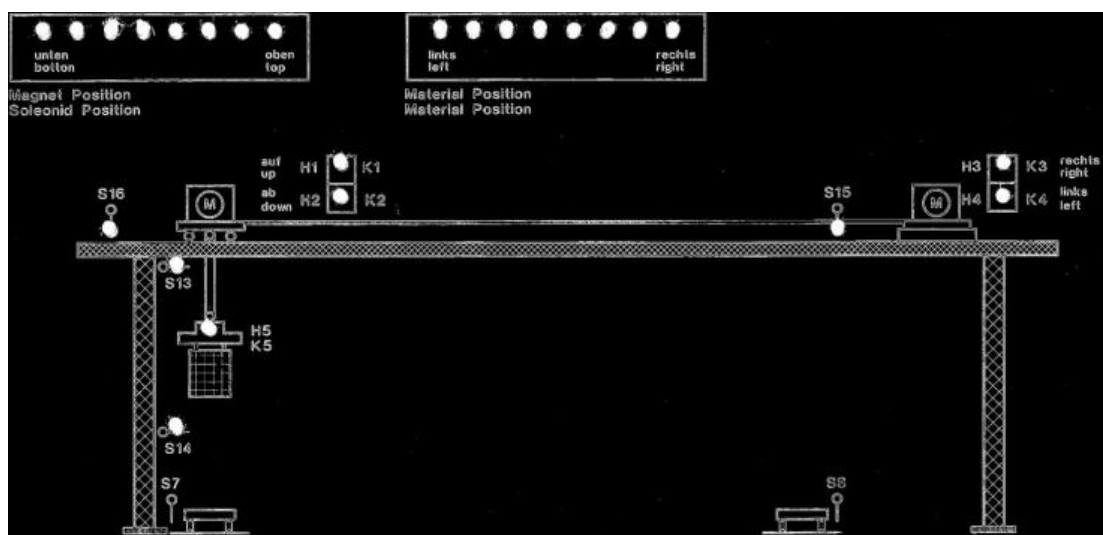


Figura 33. Plantilla de la plataforma de cambio de mercancía que encontramos en el laboratorio



### 5.1.2 Representación gráfica

La representación que hemos desarrollado para el proceso es la siguiente:

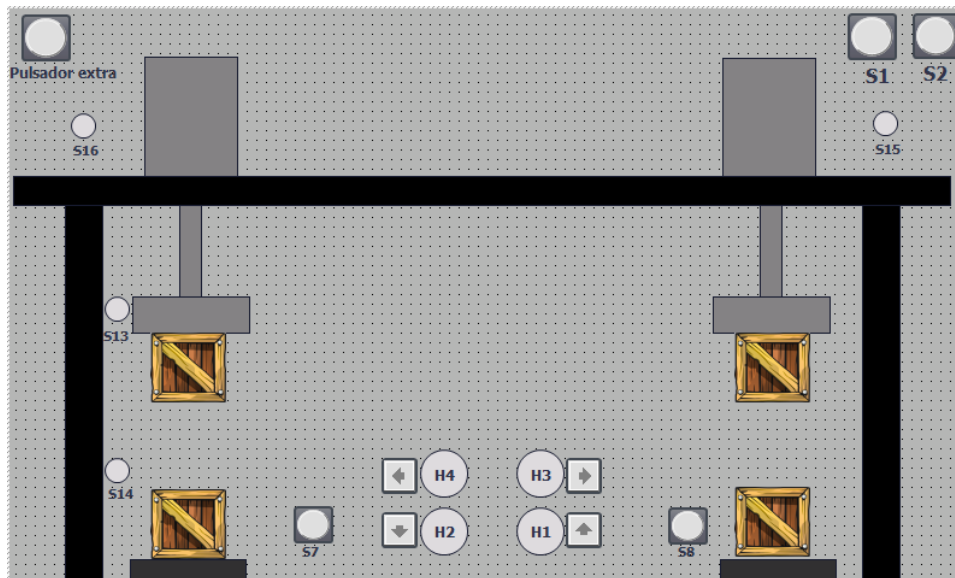


Figura 34. Representación gráfica de la plataforma de cambio de mercancía realizada en el HMI

En ella podemos observar los botones S1, S2, S7 y S8, los cuales activan las entradas correspondientes. Además, tenemos los indicadores de final de carrera (normalmente cerrados) S13, S14, S15 y S16 y los indicadores del estado de las salidas H1, H2, H3 y H4.

Como antes hemos explicado, para elaborar un movimiento fluido de un objeto que se mueve en dos direcciones debemos jugar con su visibilidad, por lo que debemos de incluir en la representación gráfica todos los elementos para poder crear un movimiento fluido. Es por esto por lo que hay 5 cajas (2 superpuestas) y 3 grúas (2 superpuestas) diferentes.

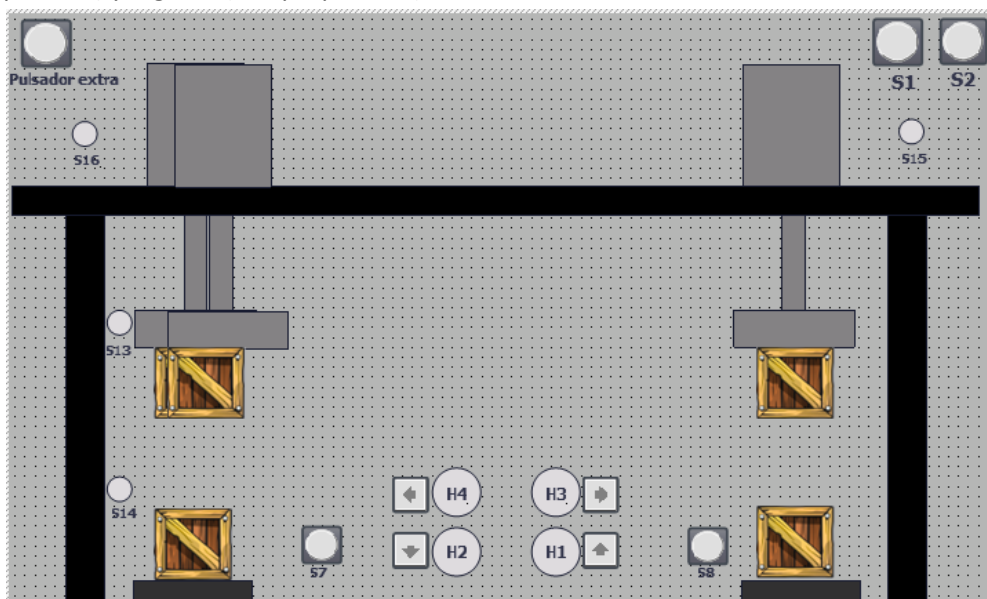


Figura 35. Superposición de objetos en la representación gráfica



### 5.1.3 Variables

Para trabajar de la forma adecuada con las variables, debemos establecer una conexión entre las variables creadas por el alumno y las internas (como hemos explicado en el apartado 4.1.2). Para realizar esta conexión correctamente, las entradas internas determinarán el estado de las entradas creadas por el alumno y las salidas creadas por el alumno lo harán con las salidas internas:

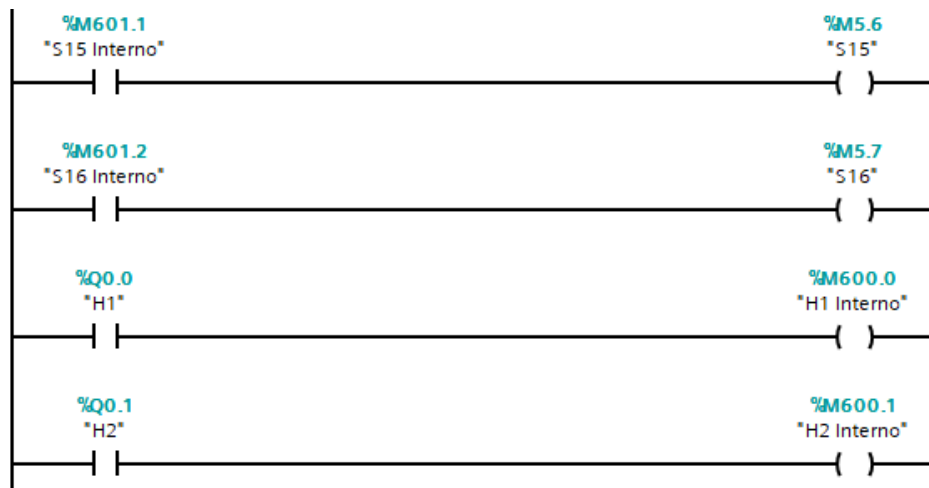


Figura 36. Transducción de las variables

Para que el HMI pueda operar con las variables del PLC debemos vincularlas desde la tabla de variables del HMI. Al vincular las variables, todo cambio que se produzca en la variable del PLC se reproducirá en la del HMI.

Variables HMI						
Nombre	Tabla de varia...	Tipo de datos	Conexión	Nom...	Variable PLC	
Caja Preparada	Tabla de varia...	Bool	HMI_...	PLC_1	*Caja Preparada para ...	
H1	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Actuador_1	
H2	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Actuador_2	
H3	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Actuador_3	
H4	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Actuador_4	
H5	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Actuador_5	
Movimiento Vertical	Tabla de varia...	Int	HMI_Con...	PLC_1	*Movimiento Vertical*	
MovLateral	Tabla de varia...	Int	HMI_Con...	PLC_1	MovLateral	
Pulsador extra	Tabla de varia...	Bool	HMI_Con...	PLC_1	*Pulsador extra interno*	
S1	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_1	
S13	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_3	
S14	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_4	
S15	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_5	
S16	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_6	
S2	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_2	
S7	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_7	
S8	Tabla de varia...	Bool	HMI_Con...	PLC_1	_Entrada_8	
VCaja1	Tabla de varia...	Bool	HMI_Con...	PLC_1	VCaja1	
VCaja2	Tabla de varia...	Bool	HMI_Con...	PLC_1	VCaja2	
VCaja3	Tabla de varia...	Bool	HMI_Con...	PLC_1	VCaja3	

Figura 37. Tabla de variables del HMI

La transducción de las variables alumno a internas y viceversa y la vinculación entre las variables del HMI y las del PLC se realizan de la misma forma en todas las maquetas.



### 5.1.4 Movimiento

Para elaborar el movimiento de la grúa y de la caja hemos creado dos variables: 'Movimiento Lateral' y 'Movimiento Vertical'.

La variable 'Movimiento Vertical' será la encargada de los movimientos verticales del proceso y tendrá como valor máximo 15 y como mínimo 0. Si H1 está activo, el bucle que genera el movimiento ira incrementando el valor de la variable hasta llegar a 15. Si lo está H2, el valor se irá reduciendo:

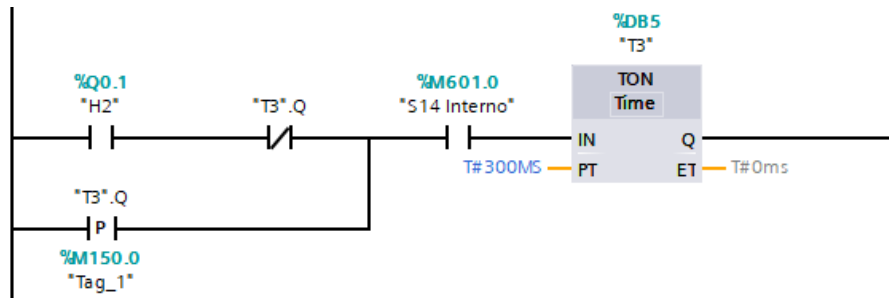


Figura 38. Temporizador del motor H2

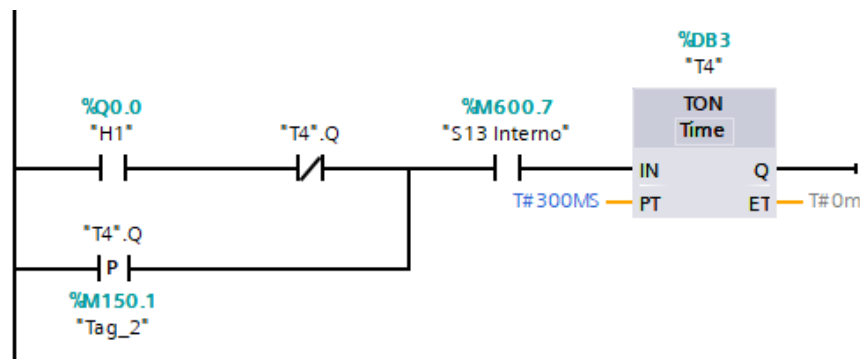


Figura 39. Temporizador del motor H1

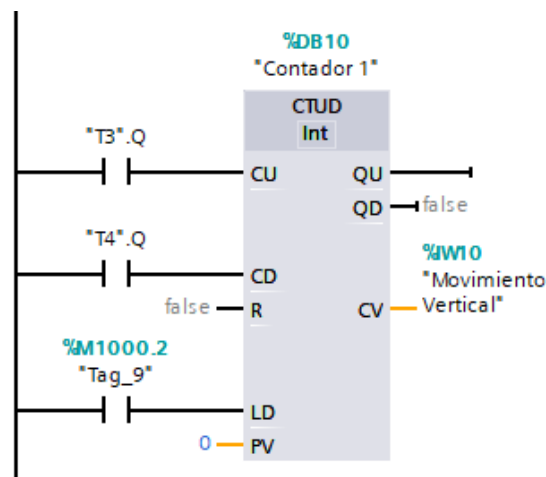


Figura 40. Contador CTUD que controla el movimiento vertical

Para agregarle movimiento a los objetos tenemos que elegir la opción movimiento vertical, asignarle la variable correspondiente y establecer el rango de valores que lo definen.

Si queremos modificar la velocidad con la que se mueven los objetos tenemos que cambiar el tiempo de los temporizadores. Cuanto más pequeño sea el tiempo del temporizador, más rápido será el movimiento.



Cuando el valor de la variable sea el del extremo inferior del rango, la posición del objeto será la inicial, mientras que, si el valor es el del extremo superior del rango, la posición del objeto será la establecida como final. Los valores intermedios del rango me proporcionan las posiciones proporcionales intermedias entre la posición inicial y final.



Figura 41. Configuración del movimiento

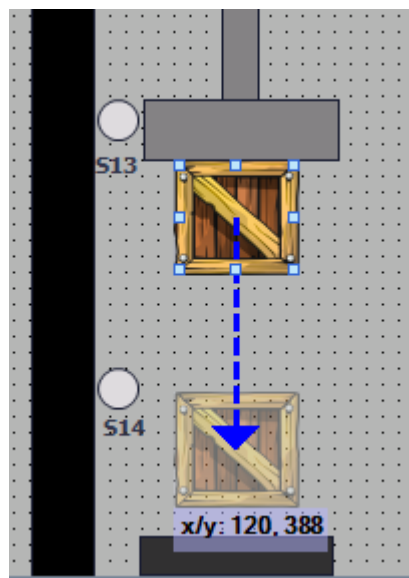


Figura 42. Posición inicial y final de la caja

Cuando la variable llega a su valor máximo, el objeto llega al final de carrera y lo desactiva (contacto normalmente cerrado). Para programar esto, procedemos de la siguiente forma:

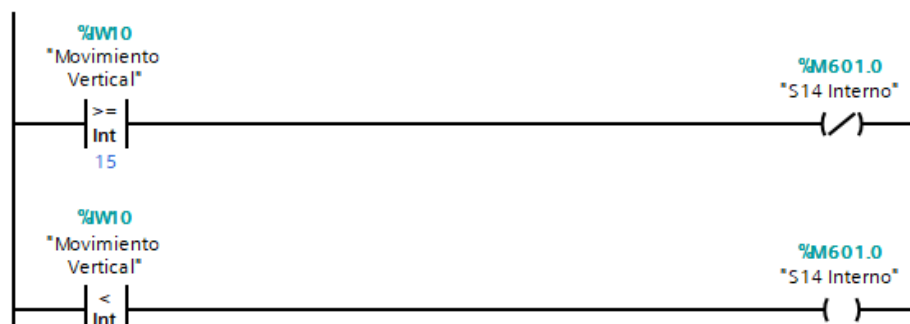


Figura 43. Configuración del final de carrera S14



Lo mismo ocurrirá con el final de carrera opuesto, pero de forma contraria:

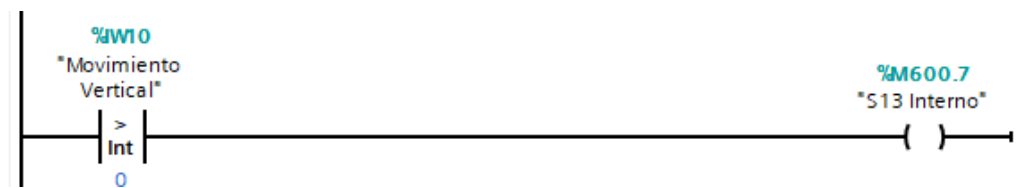


Figura 44. Configuración del final de carrera S13

La variable 'Movimiento Lateral' será la encargada de los movimientos horizontales del proceso y tendrá como valor máximo 30 y como mínimo 0. Si H3 está activo, el bucle que genera el movimiento ira incrementando el valor de la variable hasta llegar a 30. Si lo está H4, el valor se irá reduciendo.

En este caso, seleccionaremos el movimiento horizontal para describir el movimiento de la caja y de la grúa y elegiremos el rango de valores correspondiente.

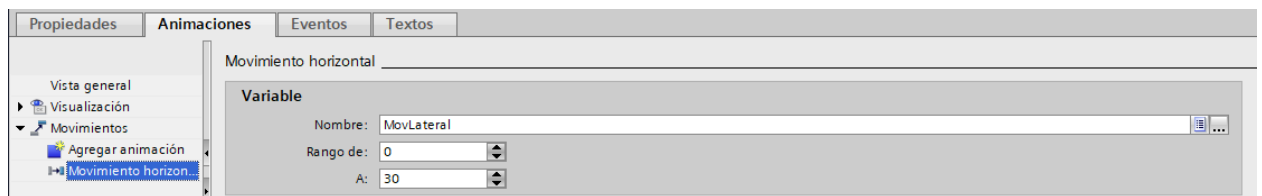


Figura 45. Configuración del movimiento horizontal de la grúa

Para configurar los finales de carrera, procederemos de la misma forma que para el caso del movimiento vertical, pero modificando los valores.

Para configurar los botones lo hacemos de la siguiente manera:

Seleccionamos el botón que queremos configurar y nos vamos a la pestaña 'Eventos' de la ventana 'Propiedades'. En ella, en la opción 'Pulsar', agregamos la función 'ActivarBit' y seleccionamos la variable que queremos activar. Para terminar de configurarlo, en la opción 'Soltar', agregamos la función 'DesactivarBit' y seleccionamos la variable que antes hemos seleccionado.

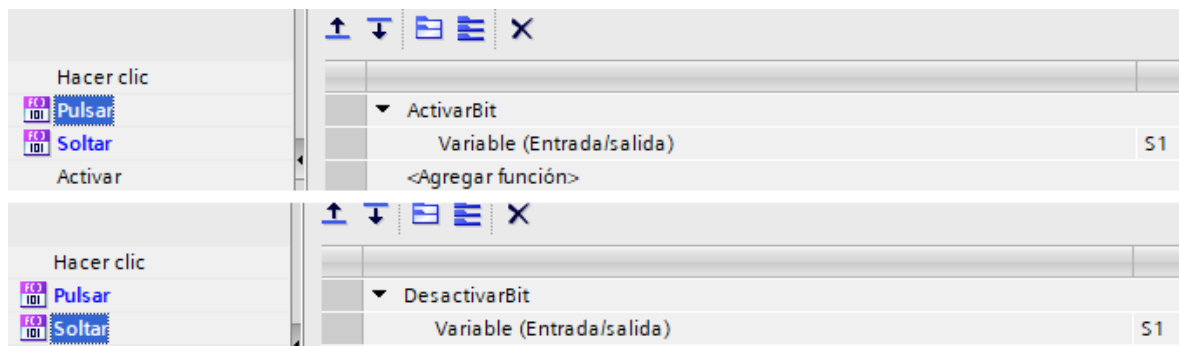


Figura 46. Configuración de los botones

### 5.1.5 Visibilidad y apariencia

Para poder describir el proceso correctamente, debemos combinar de forma adecuada el movimiento con la visibilidad de los objetos y su apariencia.

Durante el proceso, los objetos que necesitaran de ajustes de visibilidad son la grúa y la caja.



- Grúa:

La grúa primero desciende desde la parte izquierda hasta donde está situada la caja, y después de recoger la caja, asciende hasta su posición inicial. Después, se desplaza hacia la derecha para luego descender y dejar la caja en su lugar. Por lo tanto, necesitamos tres grúas diferentes: dos grúas con movimiento vertical, una para el lado izquierdo y otra en el derecho, y otra que realice el movimiento horizontal.

Para configurar la visibilidad de la grúa cuando está en la parte izquierda de la plataforma creamos la variable 'Visibilidad Izquierda' y la configuramos como vemos en la figura 47, la variable estará activa cuando la variable asignada al movimiento lateral sea igual a 0, lo que me indica que estará en la parte izquierda de la plataforma.



Figura 47. Visibilidad de la rama izquierda de la plataforma

A continuación, vamos al HMI y seleccionamos la grúa y en la pestaña 'Visibilidad' de la ventana 'Visualización' elegimos la variable que hemos creado y la configuramos como en la figura 48, de tal modo que cuando la variable no este activa, la grúa encargada del movimiento de la parte izquierda de la plataforma va a estar oculta.

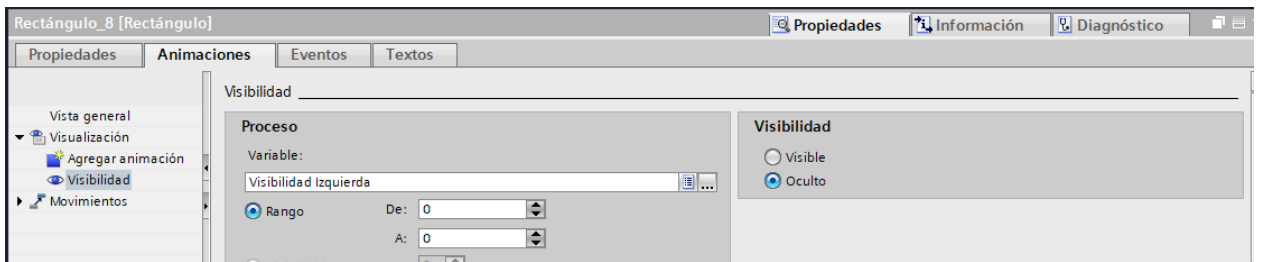


Figura 48. Configuración de la visibilidad de la rama izquierda de la plataforma

Procedemos de la misma forma para las demás grúas, pero cambiando la variable de la que dependen, como indican la figura 49 y 50.

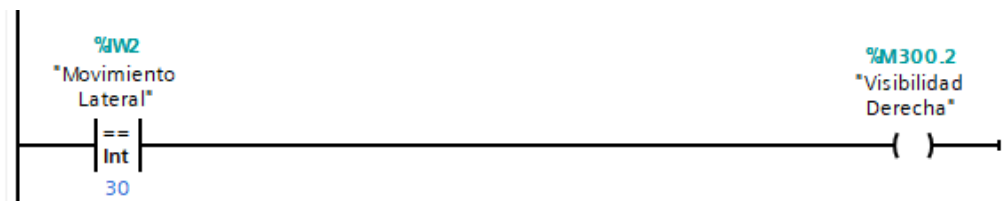


Figura 49. Visibilidad de la parte derecha de la plataforma

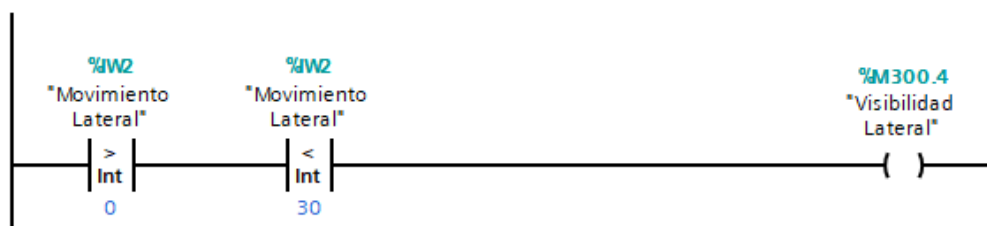


Figura 50. Visibilidad en la parte central de la plataforma





- Caja

En el estado inicial, no hay ninguna caja, pero al pulsar el botón S7, la caja aparece en la zona de recogida. Al llegar la grúa a la posición de recogida de la caja, el imán se activa y la grúa recoge la caja. La caja asciende, luego se desplaza hasta la derecha y acaba bajando hasta la posición final, donde es dejada por la grúa después de desactivar el imán.

Para describir todo el proceso, hemos usado 5 cajas diferentes:

La primera caja es la que situamos en el punto de recogida y solo será visible desde que pulsamos el botón S7 hasta que la caja es recogida por el imán

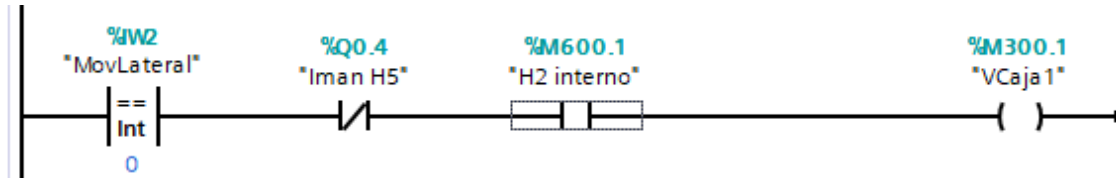


Figura 51. Visibilidad de la Caja 1

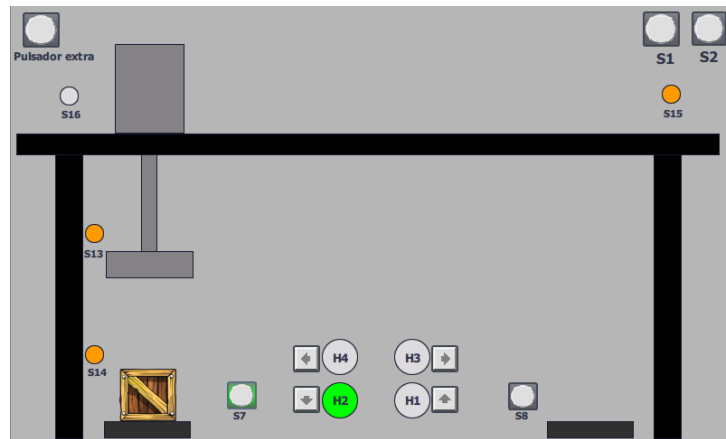


Figura 52. Vista de la caja 1 durante el proceso

La segunda caja es la que asciende desde el punto de recogida hasta la posición inicial:

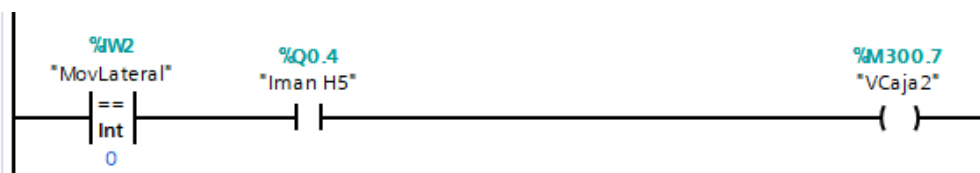


Figura 53. Visibilidad de la caja 2

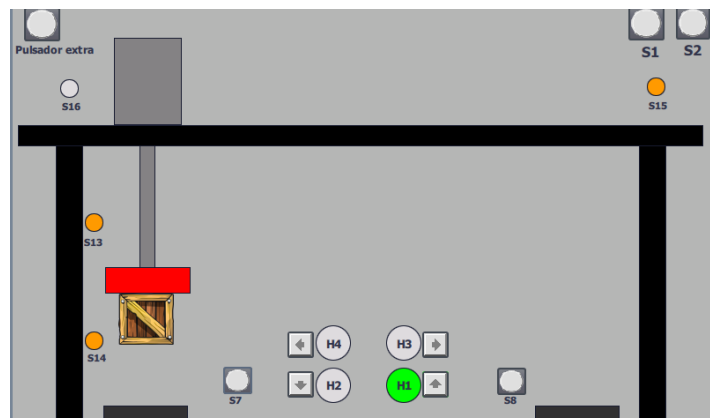


Figura 54. Vista de la caja 2 durante el proceso



La tercera es caja es la que se desplaza desde la parte izquierda de la plataforma a la derecha

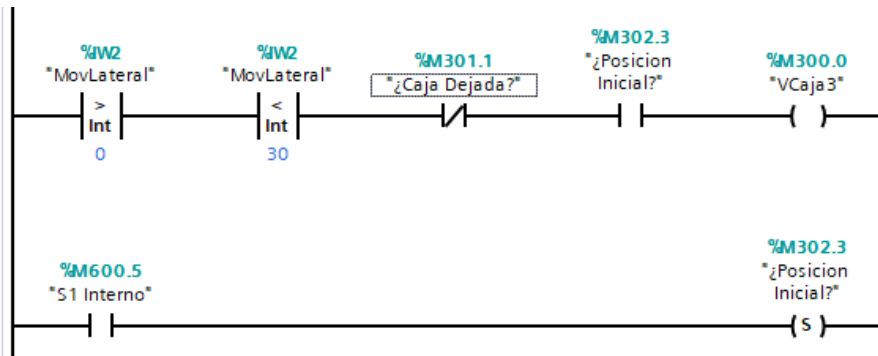


Figura 55.Visibilidad de la caja 3

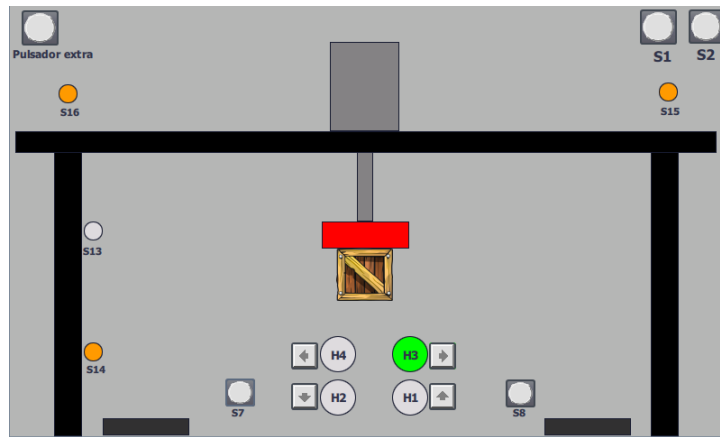


Figura 56.Vista de la caja 3 durante el proceso

La cuarta caja es la que desciende desde la parte derecha de la plataforma hasta la zona donde se deja la caja:

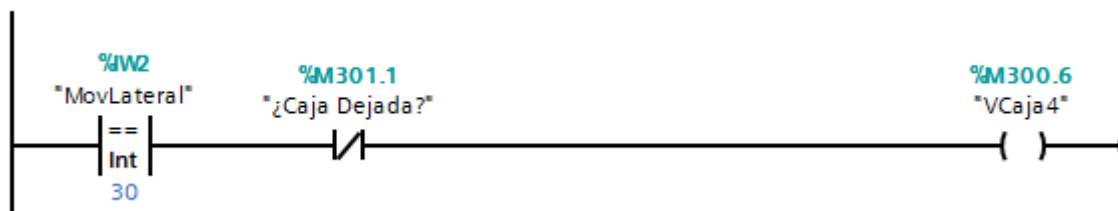


Figura 57.Visibilidad de la caja 4

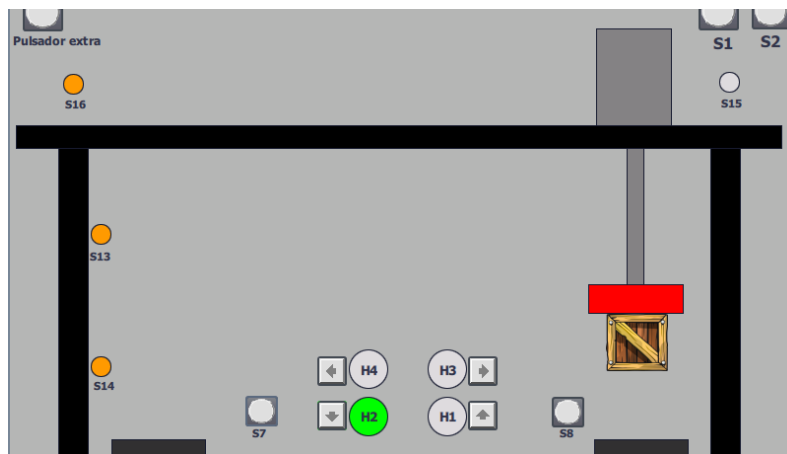


Figura 58.Vista de la caja 4 durante el proceso



La quinta caja es la caja dejada en la zona de descarga:

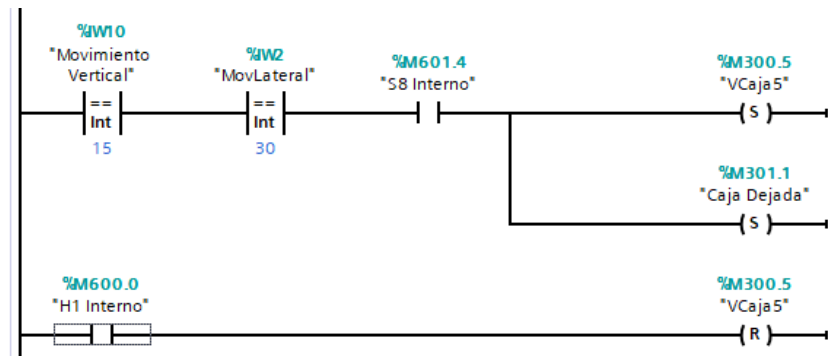


Figura 59. Visibilidad de la caja 5

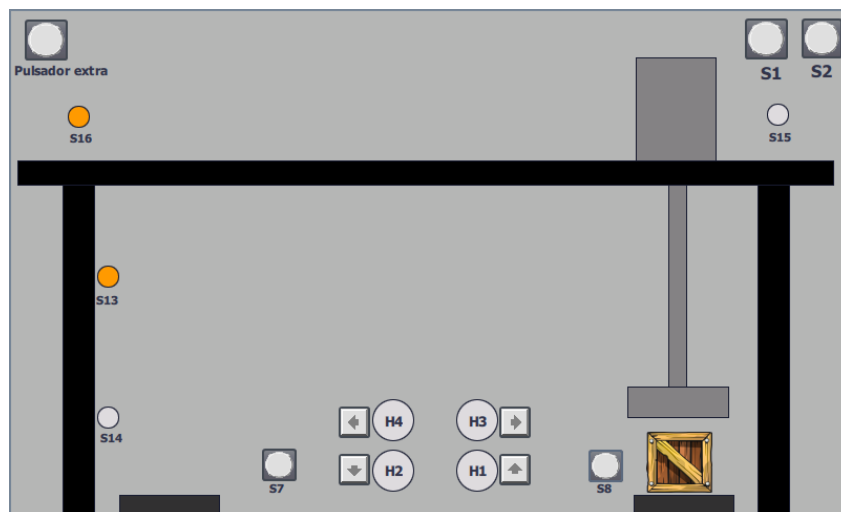


Figura 60. Vista de la caja 5 durante el proceso

Para configurar las visibilidades de las cajas hay que operar de la misma forma que con las grúas, le asignamos la variable a la visibilidad y cuando el valor de la variable sea 0, la caja estará oculta.

Además de los ajustes de visibilidad, los ajustes de apariencia también nos ayudan a describir el proceso. Nos permite cambiar el color de los objetos en función del valor de la variable que le asignamos. Esto es útil para todas aquellas indicaciones luminosas que requiera el proceso, como puede ser el estado de los finales de carrera.

Para configurar la apariencia de cualquier objeto procedemos de la siguiente forma: Seleccionamos el objeto del que queremos modificar la apariencia. En la ventana 'Animaciones', seleccionamos la pestaña 'Visualización' y en la opción 'Apariencia' le asignamos una variable y establecemos la apariencia del objeto para los diferentes valores de la variable.

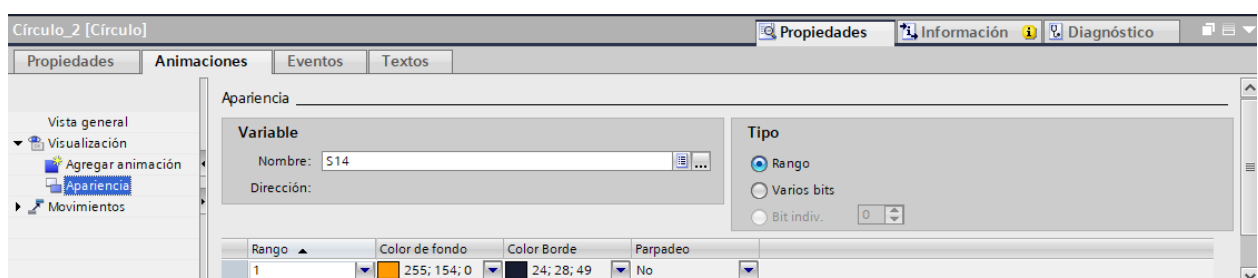


Figura 61. Ajuste de apariencia de la señal de final de carrera



Ahora debemos programar la variable acorde al proceso. En este caso, S14 es un final de carrera por lo que:

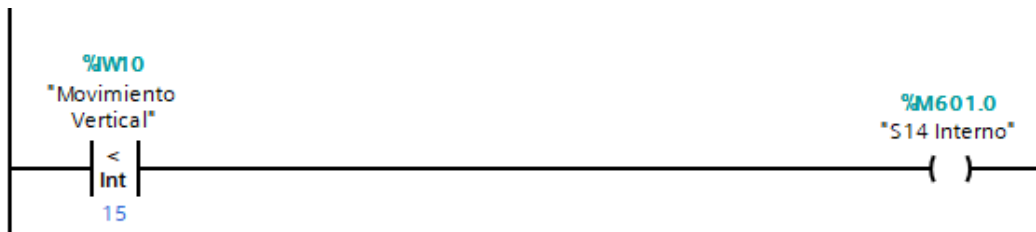


Figura 62. Configuración del final de carrera

Procedemos de la misma forma con las demás señales luminosas del proceso, como son la iluminación de los botones, las de los indicadores de funcionamiento de los motores y la señal que indica que el imán está activo.

## 5.2 Dobladora

### 5.2.1 Enunciado

Una herramienta de doblado consta de 3 cilindros neumáticos y se utiliza para doblar chapa metálica. Después de que la lámina de metal se haya colocado a mano, en la mesa de trabajo, se presiona el botón S1 de inicio.

La chapa queda sujeta por el pistón 1. La herramienta del pistón 2, dobla la hoja de metal 90 °, posteriormente la herramienta del pistón 3 conforma el metal a su forma final.

Cuando el sistema se activa mediante el pulsador de inicio S1, la máquina herramienta se reposiciona automáticamente a su estado inicial, es decir, los tres pistones se retraen por la conexión de las válvulas de solenoide Y1, Y4 y Y6.

Una vez que el sistema está en su estado inicial, el pistón 1 se extiende por medio de la activación de la válvula Y2 y mantiene la hoja de metal sobre la mesa de trabajo.

Cuando se alcanza el final de carrera S17, se desactiva Y2, se activa Y3 y el pistón 2 se extiende para doblar el metal 90°.

Cuando se alcanza el final de carrera S15, se desactiva Y3. Después de un retraso de 2 segundos, el pistón 2 se activa mediante la válvula Y4 subiendo la herramienta.

Cuando se alcanza el final de carrera S16, la válvula Y4 se desactiva, la Y5 se activa y el pistón 3 se extiende para doblar la chapa metálica a su forma final.

Cuando se alcanza el final de carrera S14, la válvula Y5 se desactiva. Después de un retraso de 2 segundos, el pistón 3 se retrae mediante la activación de la válvula Y6.

Cuando se alcanza el final de carrera S13, Y6 será desactivada, el pistón 1 se retrae por medio de Y1 y cuando se alcanza el final de carrera S18, Y1 se desconecta.

Tan pronto como se pulsa el botón de inicio, se repite el proceso.

Debería ser posible detener el sistema en cualquier momento con el pulsador de parada S2. La pulsación de S1, activa los movimientos necesarios para alcanzar el estado inicial.



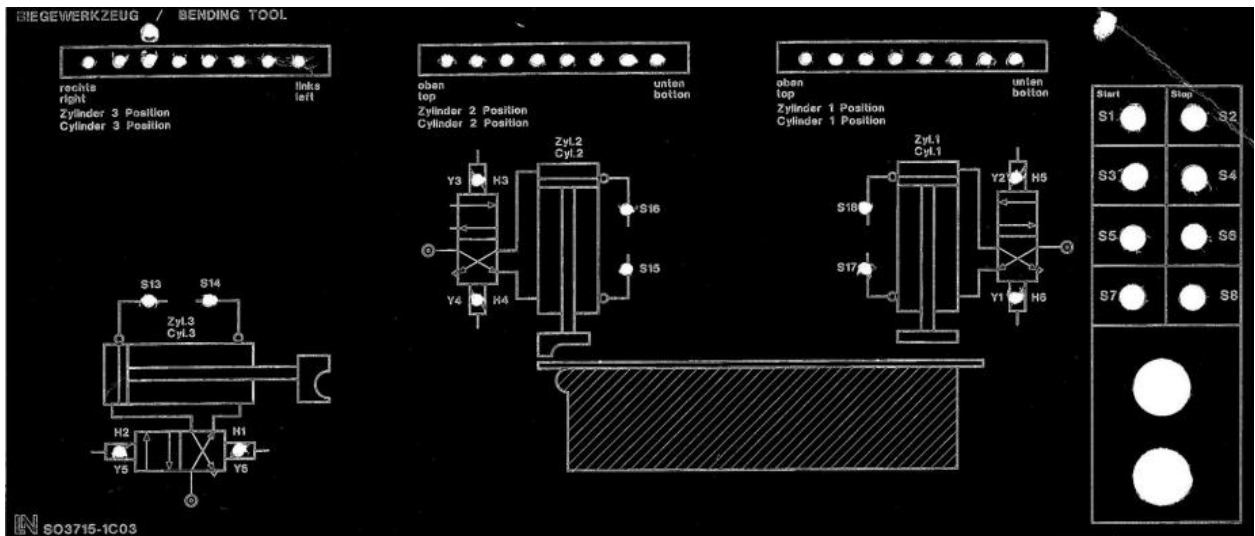


Figura 63. Plantilla de la dobladora que encontramos en el laboratorio

## 5.2.2 Representación grafica

La representación que hemos desarrollado para el proceso es la siguiente:

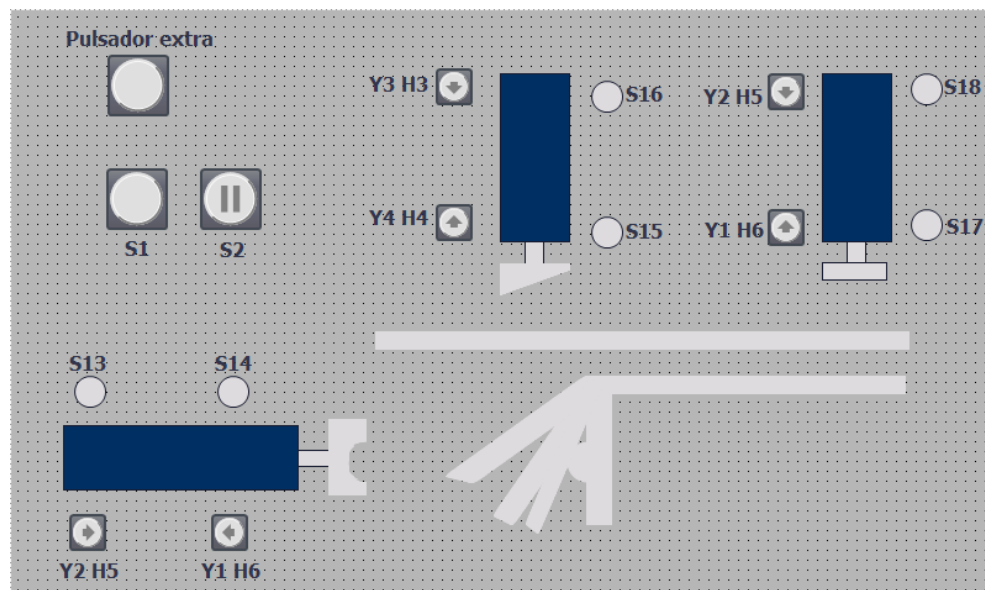


Figura 64. Representación gráfica de la dobladora en el HMI

En ella podemos ver todos los botones que controlan el proceso (S1, S2 y el pulsador extra), los indicadores de final de carrera S13, S14, S15, S16, S17 y S18 (normalmente cerrados) y los indicadores del estado de las salidas H1, H2, H3, H4, H5 y H6.

Además, podemos observar los cilindros que doblarán la barra y los diferentes estados de la barra a lo largo del proceso.



### 5.2.3 Movimiento

Para generar el movimiento utilizamos el mismo método que en la maqueta del apartado 5.1, donde el movimiento es generado a partir de temporizadores en bucle conectados a un contador CTUD. Cada cilindro tiene su propia variable y se mueve forma independiente con respecto a los demás cilindros

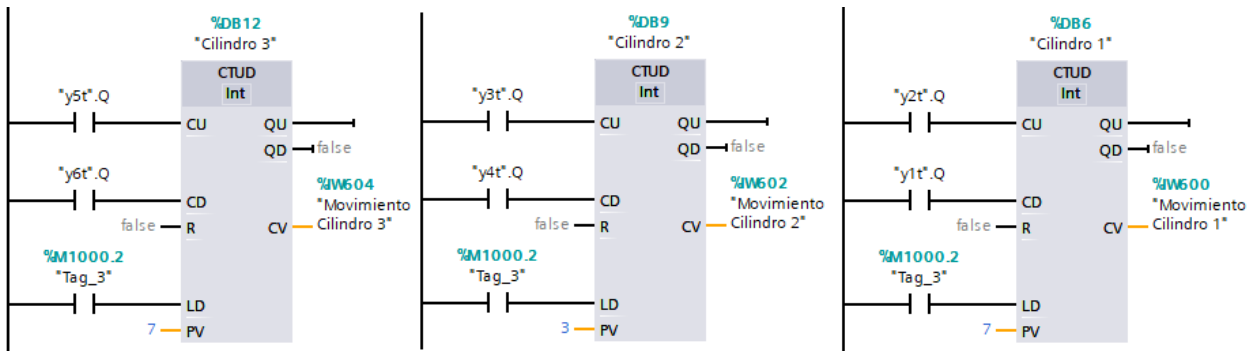


Figura 65. Contadores utilizados para describir el movimiento en la dobladora

Un aspecto a destacar es la configuración del movimiento de la barra.

Durante el proceso, el cilindro empuja la barra hasta la posición deseada. El movimiento de la barra está vinculado a la variable que controla el movimiento del cilindro 1 'Movimiento Cilindro 1', por lo que debemos de configurar el movimiento de forma que de la sensación de que el cilindro empuja la barra.

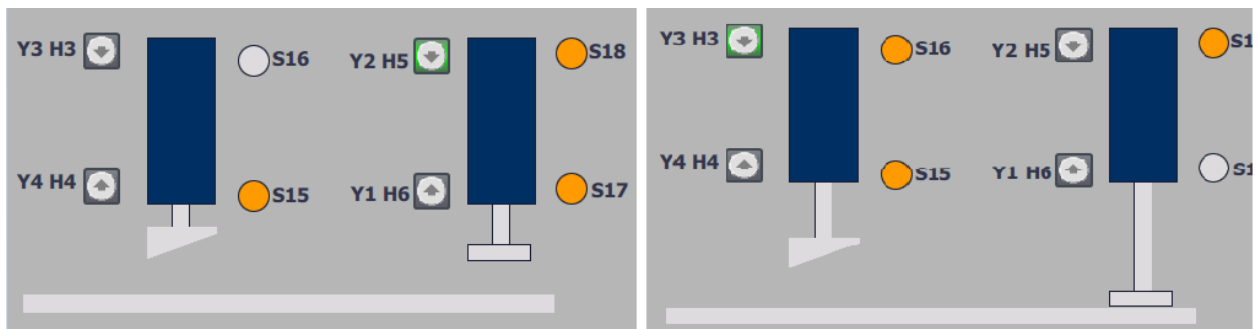


Figura 66. Cilindro empujando la barra

Para realizar esto, debemos cambiar el rango en el que tiene que estar la variable para que se mueva la barra. Como podemos ver en la figura 67, en este caso el rango es de 11 a 15, donde 11 es el valor de la variable en el que el cilindro está a la altura de la barra y 15 es el valor final de la variable.

Si hubiésemos configurado el movimiento de la barra como hacemos normalmente, donde el rango de movimiento abarca todos los valores de la variable, el movimiento de la barra y el del cilindro iría en conjunto y llegarían a tocarse nunca.

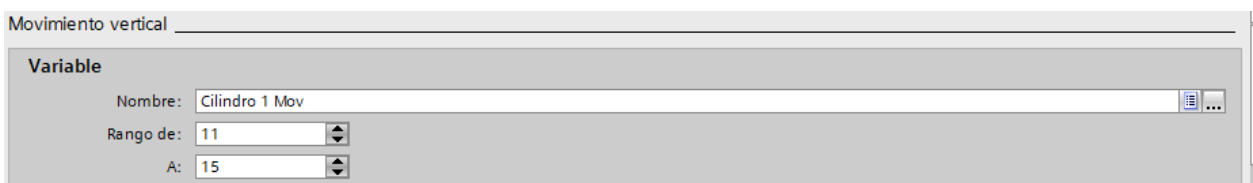


Figura 67. Ajuste del rango de la variable

Cuando necesitemos describir un movimiento en el que un objeto empuje a otro, tendremos que recurrir a este método para describir el proceso correctamente.

Para configurar los finales de carrera procedemos de la misma forma que en el apartado 5.1.4.



## 5.2.4 Visibilidad y apariencia

El único objeto que demandara ajustes de visibilidad durante el proceso es la barra. Una vez el sistema ha alcanzado su posición inicial, se coloca la barra en la posición en la que será doblada. Esta barra la haremos visible desde que el sistema este en su posición inicial hasta que el cilindro 2 alcance a la barra y pueda doblarla, donde haremos visible a la barra doblándose.

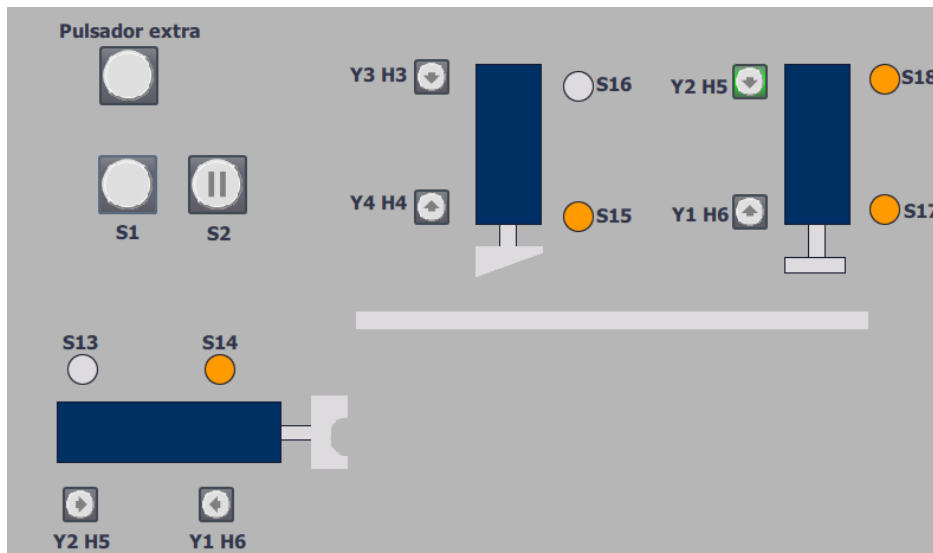


Figura 68. Vista del estado inicial de la barra

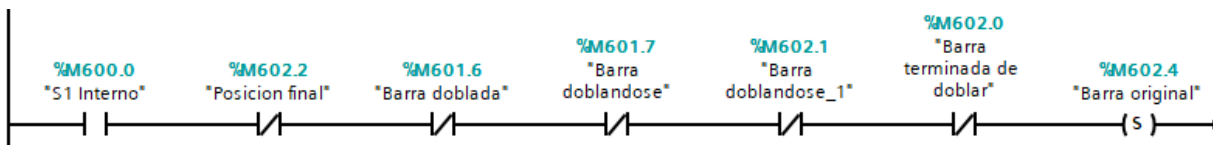


Figura 69. Visibilidad del estado inicial de la barra

Cuando el cilindro alcanza la barra, la barra original se oculta y se hacen visibles dos partes de la barra: la parte de la barra que va a permanecer sin ser doblada, la parte derecha, y la que va a sufrir los cambios durante el proceso, la parte izquierda.

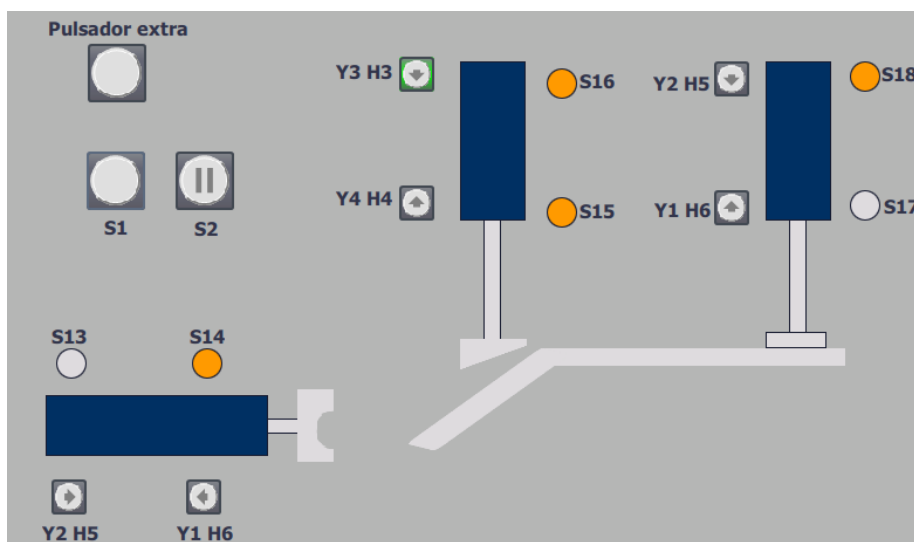


Figura 70. Vista de la barra doblándose



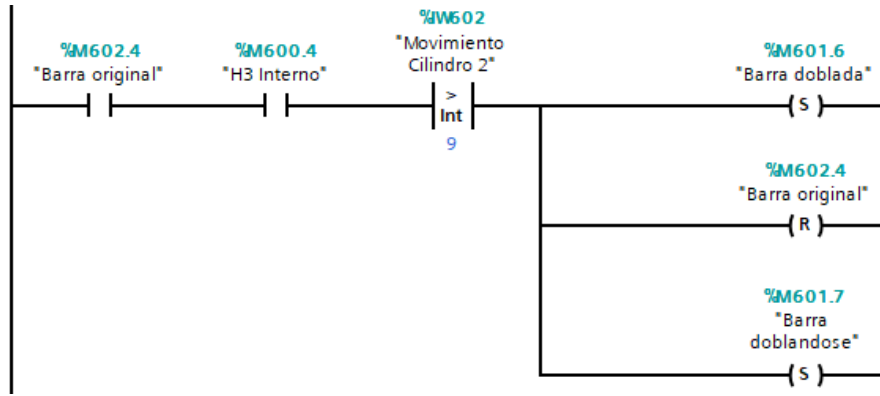


Figura 71. Visibilidad de la barra doblándose

Cuando el cilindro 2 llega a su máximo recorrido, la barra se doblará aún más, como podemos ver en la figura 72.

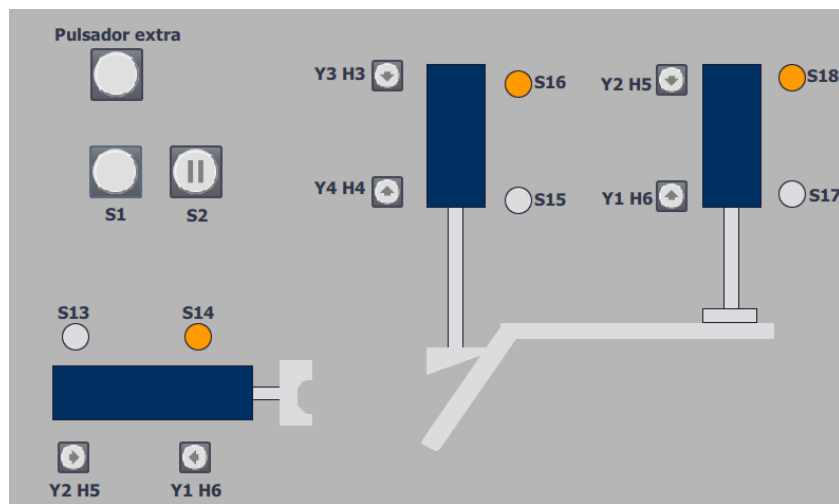


Figura 72. Vista de la barra cuando el cilindro está totalmente extendido

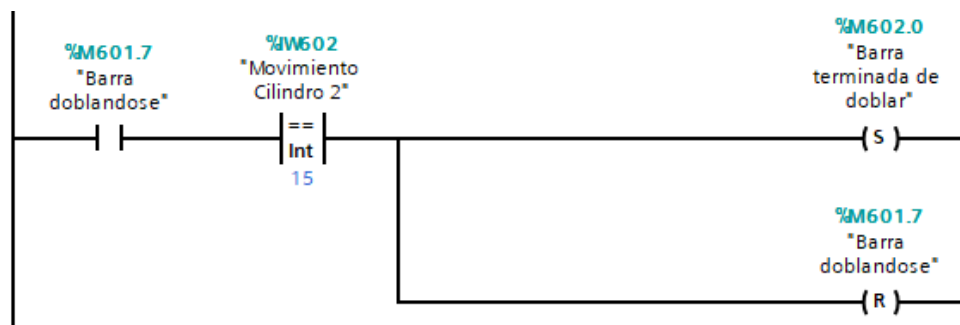


Figura 73. Visibilidad de la barra para cuando el cilindro está totalmente extendido





Después de que el cilindro 2 llegue hasta el final de su recorrido, retrocede, y a continuación el cilindro 3 avanza doblando la pieza hasta su posición final. Para dar más sensación de fluidez durante el proceso de doblado de la barra, entre la posición de doblado inicial y final, creamos una intermedia:

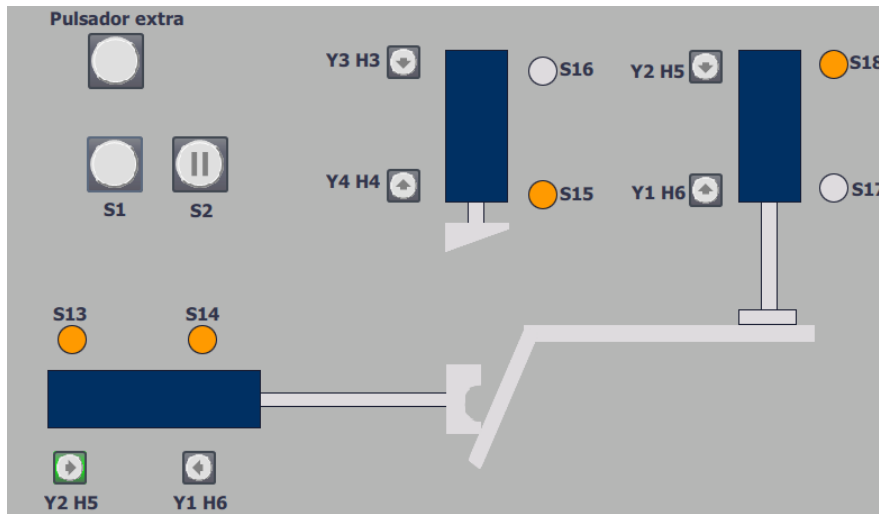


Figura 74. Vista de la barra doblándose por el cilindro 3

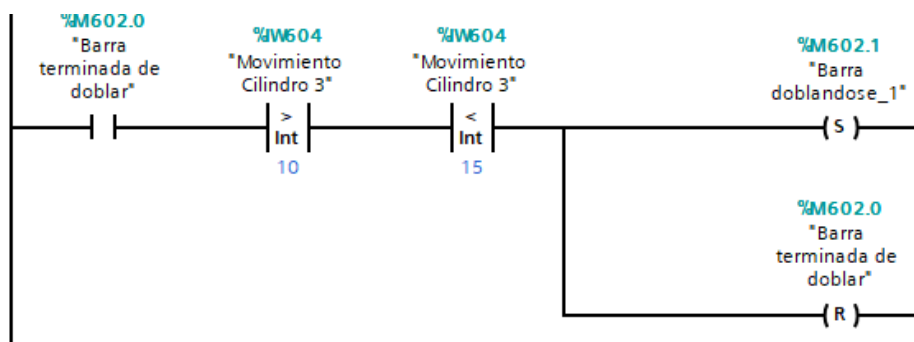


Figura 75. Visibilidad de la barra doblándose por el cilindro 3

Para terminar, cuando el cilindro 3 llega hasta el final de su recorrido, dobla la barra hasta su posición final y termina el proceso de doblado.

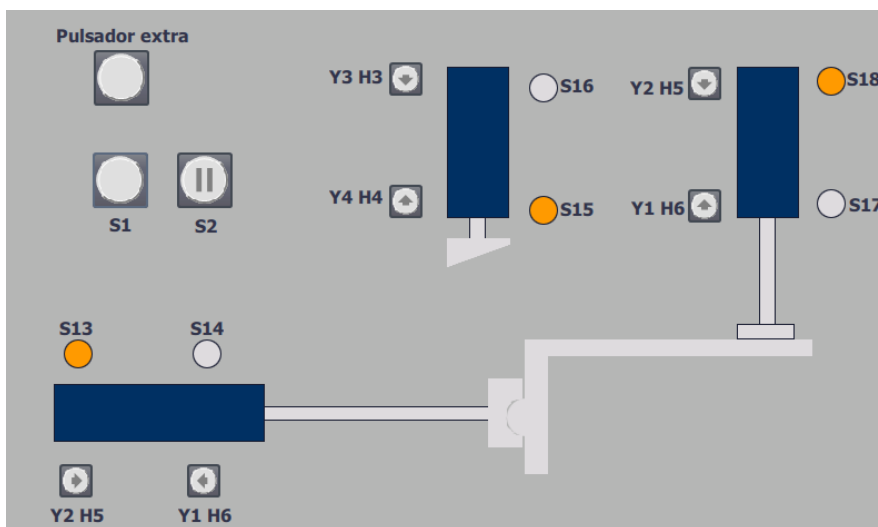


Figura 76. Vista de la barra en su posición final



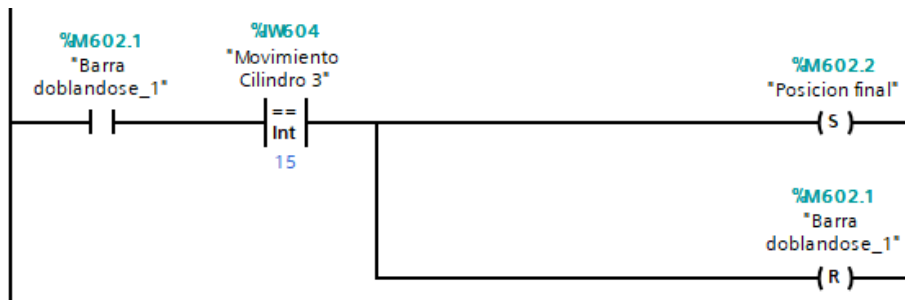


Figura 77. Visibilidad de la barra en su posición final

Para configurar las visibilidades de las cajas hay que operar de la misma forma que en el apartado 5.1.4, le asignamos la variable correspondiente a la visibilidad y cuando el valor de la variable sea 0, el objeto estará oculto.

Los elementos que requieren de ajustes de apariencia son las señales luminosas (indicadores del estado de los finales de carrera y del estado de los actuadores). Para configurarlos, procedemos de la misma forma que en el apartado 5.1.4

### 5.3 Semáforo de obra

#### 5.3.1 Enunciado

Debido a obras de construcción, el tráfico rodado debe limitarse a un solo carril. El flujo de tráfico está regulado por un sistema de luces de tráfico.

Cuando el semáforo se enciende con el interruptor S3, las dos luces deben estar en rojo (H2 y H3). Dos bucles de inducción (S1 y S2) indican la presencia de cualquier tráfico.

La fase verde de las luces debe ser de al menos 20 segundos, antes de que el otro iniciador cambie las dos luces a rojo.

Después de un retraso de 10 segundos, para permitir el tráfico para despejar la zona de restricción, se permite el flujo contrario de tráfico a moverse a lo largo de la restricción al cambiar las luces de color verde (H1 o H4).

Si no hay ningún vehículo de espera en las luces, el sistema permanece en su último estado. El sistema puede ser apagado, sólo cuando esté en verde de uno de los carriles.

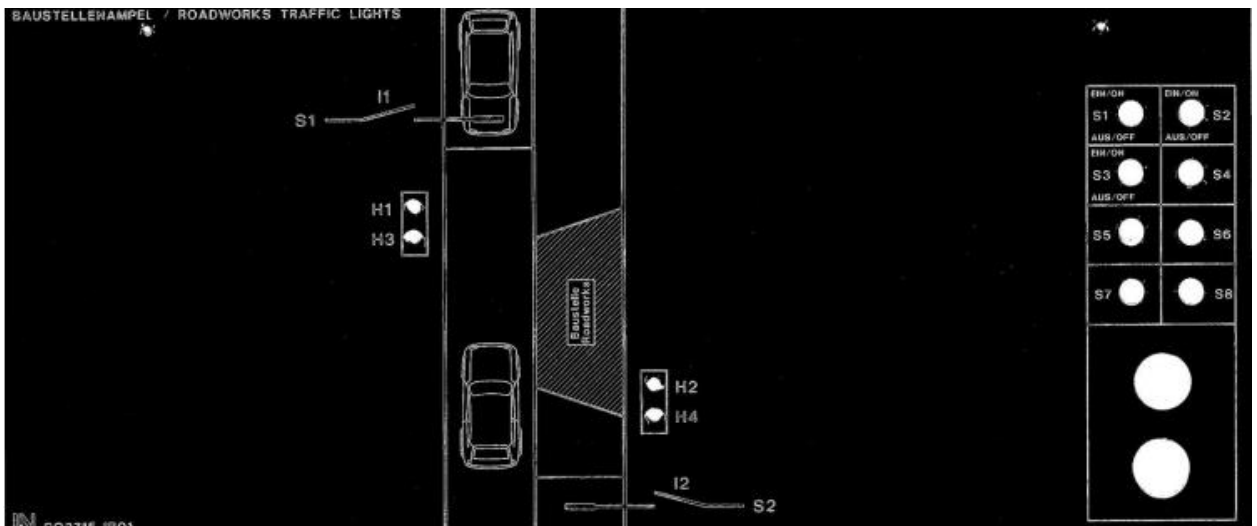


Figura 78. Plantilla del semáforo de obra que encontramos en el laboratorio



### 5.3.2 Representación gráfica

La representación que hemos desarrollado para el proceso es la siguiente



Figura 79. Representación del semáforo de obra en el HMI

En ella podemos observar los botones S1, S2 y S3, que activan las entradas correspondientes. Podemos observar cómo tanto S1 y S2 tienen dos botones, donde el botón rectangular actúa como un interruptor (activa o desactiva la variable) mientras que el botón circular activará la variable mientras este pulsado.

El interruptor S3 es un interruptor de la librería 'Toggle Switches' dentro de la librería 'Buttons and Switches' que tiene sus propios ajustes de apariencia.

Las luces de los semáforos están vinculadas con las salidas H1, H2, H3 y H4. Por ejemplo, si H1 está activa, se iluminará la luz verde del semáforo de arriba.

### 5.3.3 Movimiento

El movimiento de los coches en los dos sentidos está regido por el método de temporizadores explicado en el apartado 4.2.2.5.

En este método, la variable asociada al movimiento se vincula con la lectura interna de un temporizador.



Figura 80. Temporizador del que depende el movimiento



En la figura 81 podemos ver cómo está configurada la variable que activa el temporizador. Gracias a la primera rama del paralelo, cuando el semáforo este en verde y el coche deje de estar en la posición de espera, la variable se activará y el coche empezara a moverse. La segunda rama del paralelo activara el movimiento justo cuando el semáforo se ponga en verde. La tercera rama mantiene la variable activa, sin esta rama, la salida no estaría activa el tiempo necesario para describir correctamente el proceso. La entrada negada 'Bucle1'.Q desactivara la variable movimiento cuando el temporizador alcance el tiempo estipulado en la entrada PT del temporizador.

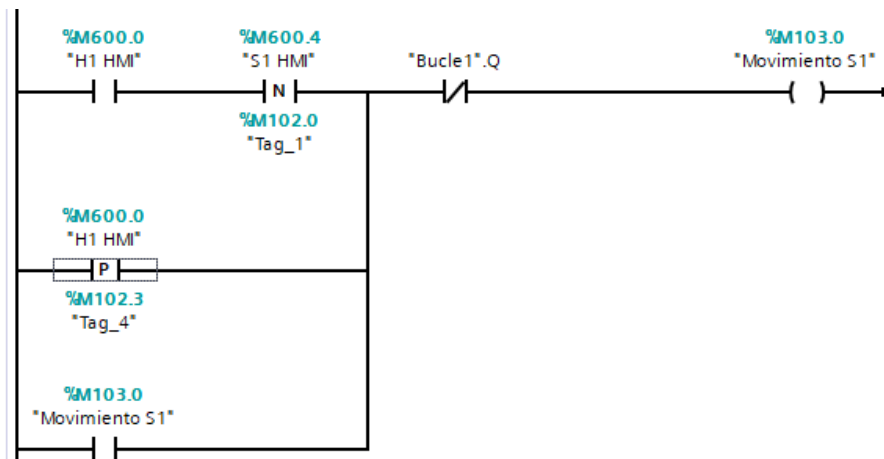


Figura 81. Variable que activa el temporizador

Para poder trabajar de forma más cómoda con esta variable será necesario hacerle un escalado lineal a la variable (apartado 4.2.2.2)

Para ello, en la tabla de variables del HMI, seleccionamos la variable a la que le queremos aplicar el escalado lineal. En el rango de la variable del PLC insertamos el valor máximo y mínimo del temporizador, mientras que en la del HMI insertamos el rango que consideremos oportuno.



Figura 82. Escalado lineal

Después, asociamos la variable al movimiento del coche y en los rangos de la variable, insertamos los rangos que hemos puesto en el escalado lineal.

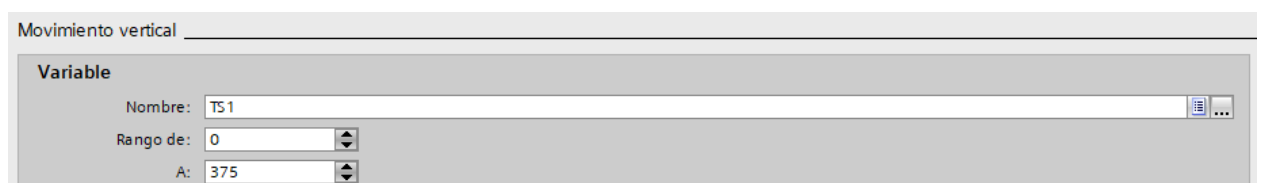


Figura 83. Configuración del movimiento: Variable y Rango



### 5.3.4 Visibilidad y apariencia

Los elementos que necesitan ajustes de visibilidad en esta maqueta son los coches. En cada sentido intervienen dos coches, uno de ellos solo será visible cuando esté avanzado por la carretera y el otro mientras este esperando a que el semáforo se ponga en verde.

Cuando un coche se encuentre en posición de espera, estará activando el sensor correspondiente (S1 o S2), por lo que ajustaremos su visibilidad como en la figura 84.

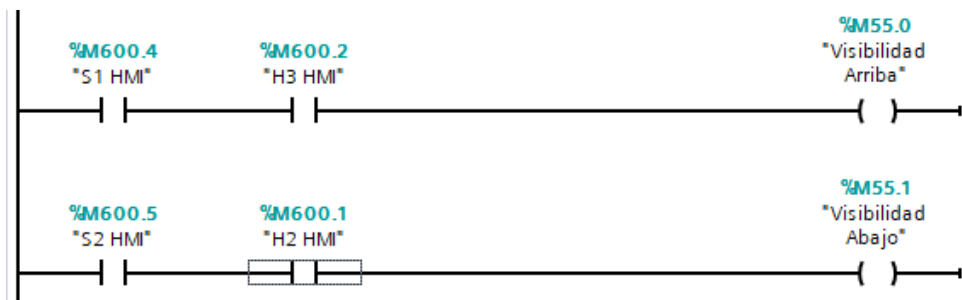


Figura 84. Visibilidad de los coches en estado de espera

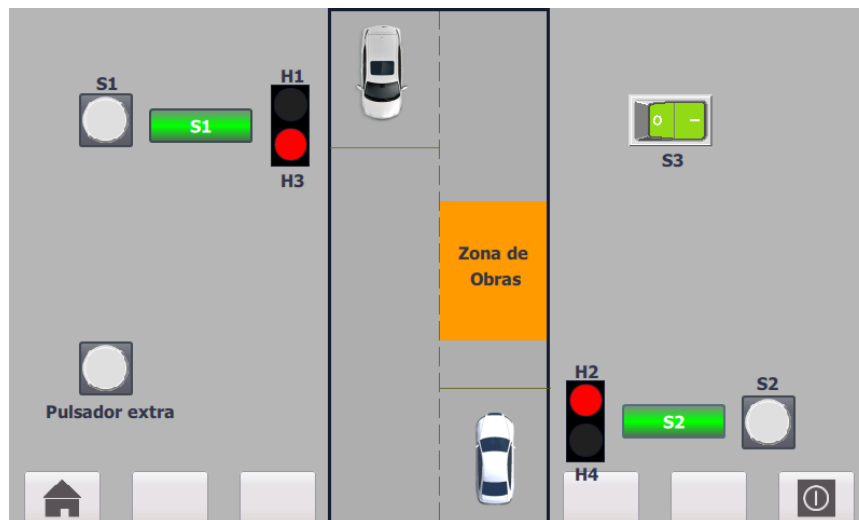


Figura 85. Vista de los coches en estado de espera

Cuando el coche se encuentre en movimiento, el valor variable que rige su movimiento será distinta de 0, por lo que ajustamos su visibilidad como en la figura 86.



Figura 86. Visibilidad de los coches cuando están cruzando la zona de obras



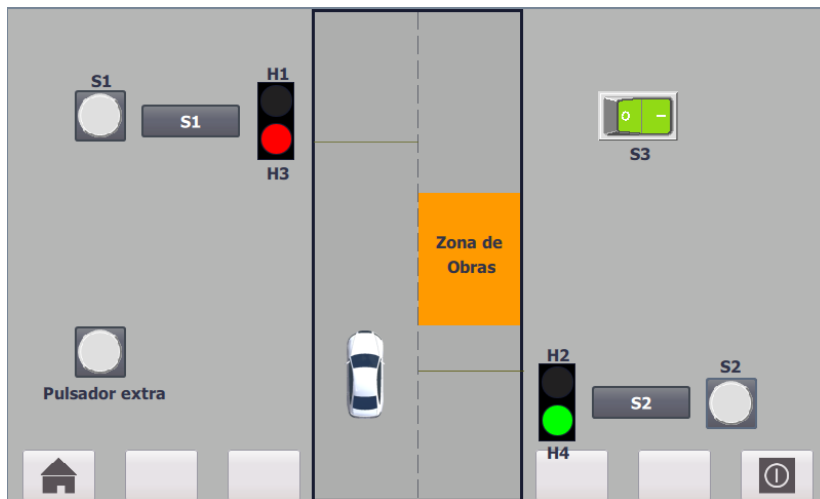


Figura 87. Vista de los coches cuando están cruzando la zona de obras

Los elementos que requieren de ajustes de apariencia son las señales luminosas (indicadores del estado de los botones y del estado de las salidas). Para configurarlos, procedemos de la misma forma que en el apartado 5.1.4

## 5.4 Taladradora

### 5.4.1 Enunciado

Un sistema de taladrado automático se utiliza para perforar agujeros en una pieza de trabajo. Un cilindro se utiliza para recuperar las piezas de trabajo de un alimentador y empujarlas al lugar de taladrado. Allí, un segundo cilindro acciona la acción de perforación. Después de que el proceso de perforación haya terminado, un tercer cilindro levanta la pieza de trabajo, que posteriormente se limpia desde el banco de perforación con aire comprimido.

Cuando el sistema se activa mediante el pulsador de inicio S1, el sistema se pone automáticamente a su estado inicial, es decir, los tres pistones se retraen por la conducción de las válvulas Y1, Y3 e Y6.

Una vez que el sistema está en su estado inicial, la válvula Y2 se activa, el pistón 1 se extiende y se fija la pieza de trabajo en la posición de perforación.

Cuando las señales S14 de final de carrera indican que el pistón se ha extendido, Y2 se desactiva, se enciende Y5 y el pistón 3 "Taladro" se extiende y se perfora la pieza de trabajo.

Cuando se alcanza el final de carrera S17, Y5 se desactiva, se conecta Y6 y el cilindro de perforación se retrae a la posición inicial.

Cuando se alcanza el final de carrera S18, Y6 se apaga, se enciende Y1 está encendido y el pistón 1 libera la pieza de trabajo.

Cuando se alcanza el final de carrera S13, Y1 se desactiva, Y4 se enciende y el pistón 2 levanta la pieza de trabajo.

Cuando se alcanza el final de carrera S16, Y4 se apaga, se activa Y7 y el aire comprimido sopla la pieza de trabajo desde el banquillo de perforación. Después de un retraso de 2 segundos, se desactiva Y7. Al mismo tiempo, Y3 se activa para retraer el pistón 2 hasta que se alcanza el final de carrera S15.

El proceso descrito anteriormente, se repite de forma sucesiva sin necesidad de pulsar nada.

Debería ser posible detener el sistema en cualquier momento, con el botón de parada S2. Un nuevo comienzo con el botón de inicio S1 entonces activa los movimientos necesarios para alcanzar un estado inicial y comenzar el proceso.



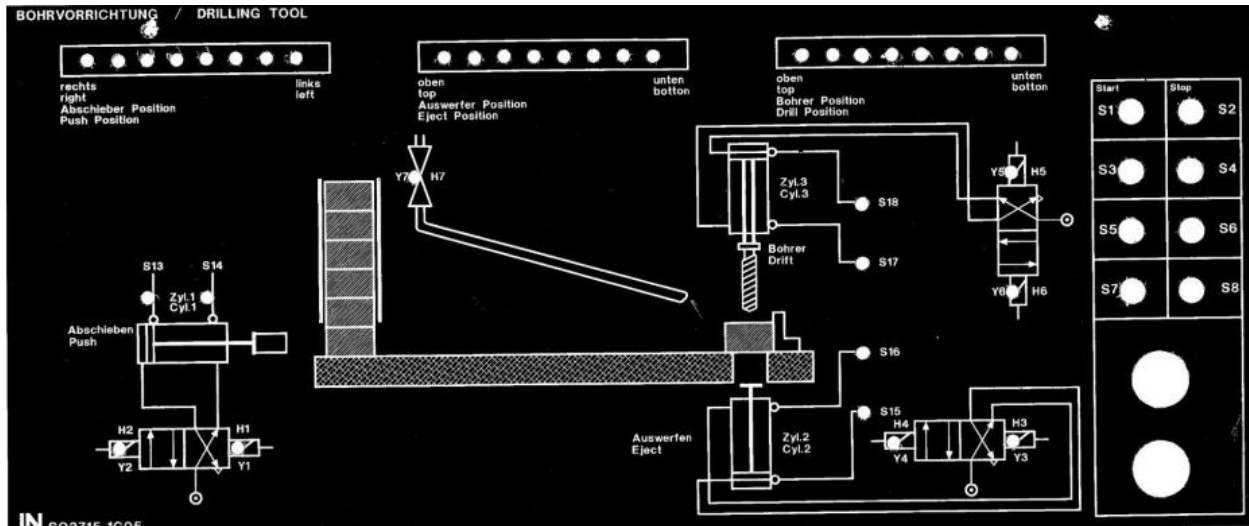


Figura 88. Plantilla de la Taladradora que encontramos en el laboratorio

### 5.4.2 Representación gráfica

La representación que hemos desarrollado para el proceso es la siguiente:

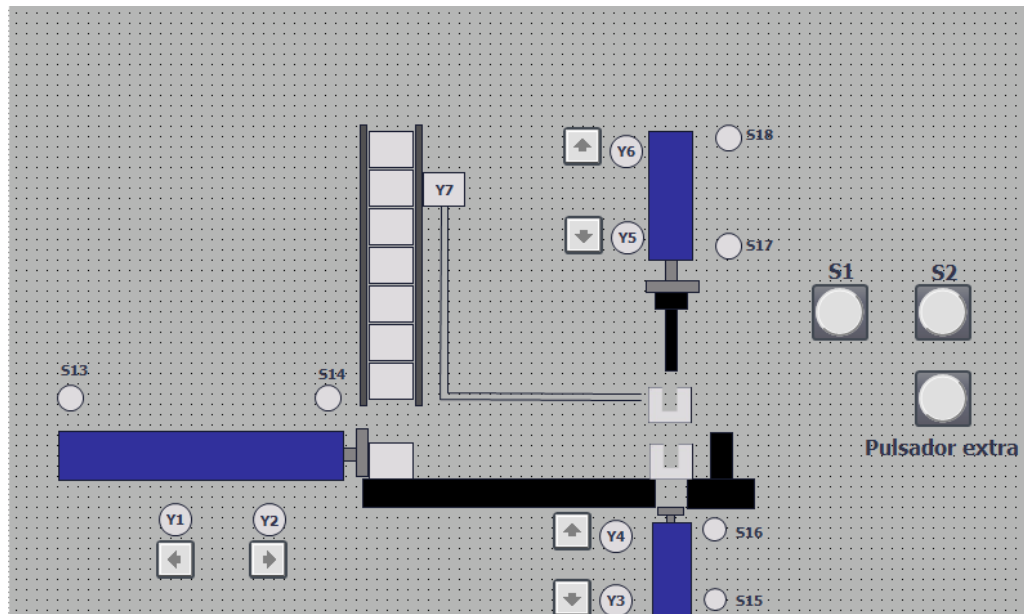


Figura 89. Representación de la taladradora en el HMI

En ella podemos ver todos los botones que controlan el proceso (S1, S2 y el pulsador extra), los indicadores de final de carrera S13, S14, S15, S16, S17 y S18 (normalmente cerrados) y los indicadores del estado de las salidas Y1, Y2, Y3, Y4, Y5, Y6 y Y7.

Además, podemos observar los cilindros que participan en el proceso y los diferentes estados de la pieza a lo largo del proceso.

### 5.4.3 Movimiento

Para desarrollar el movimiento utilizaremos los dos métodos explicados en el apartado 4.2.2.5. Para elaborar el movimiento de los cilindros y de la caja en casi todos sus estados, utilizaremos el método de los contadores, donde el movimiento es generado a partir de temporizadores en bucle conectados a un contador CTUD. Cada cilindro tiene su propia variable y se mueve forma independiente con respecto a los demás cilindros.



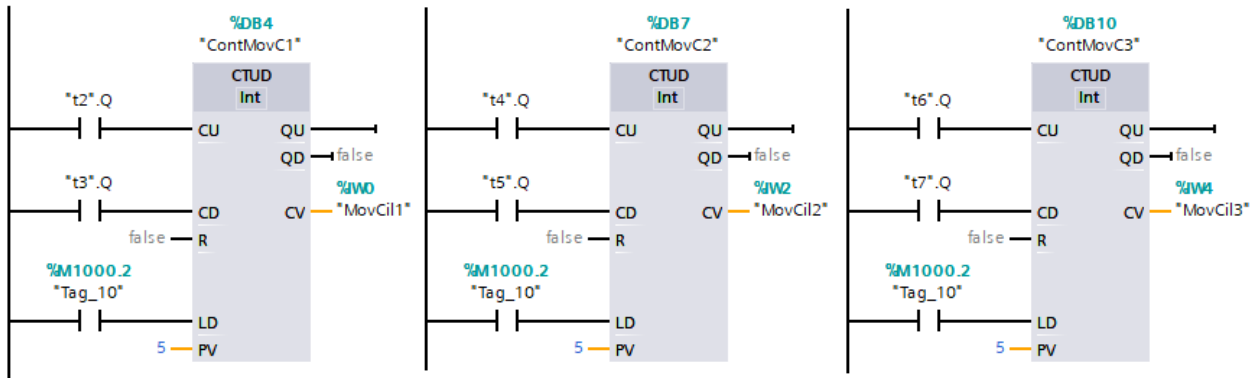


Figura 90. Contadores CTUD de los que depende el movimiento

En la figura X podemos apreciar los dos tipos de interacción entre un cilindro y un objeto. En la imagen de la izquierda tenemos el caso de un objeto que es empujado por el cilindro desde que el cilindro empieza a moverse mientras que en la imagen de la derecha observamos el caso en el que el cilindro cuando empieza a moverse no está en contacto con el objeto que tiene que mover.

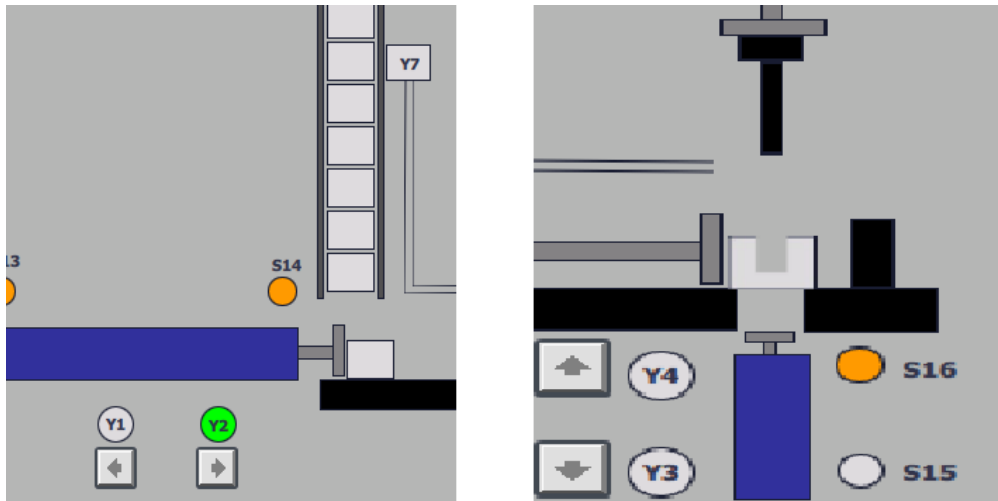


Figura 91. Tipos de interacción entre un cilindro y un objeto

Para el primer caso, cuando configuremos el movimiento de la pieza, el rango de la variable tendrá que ser el mismo que el del cilindro.

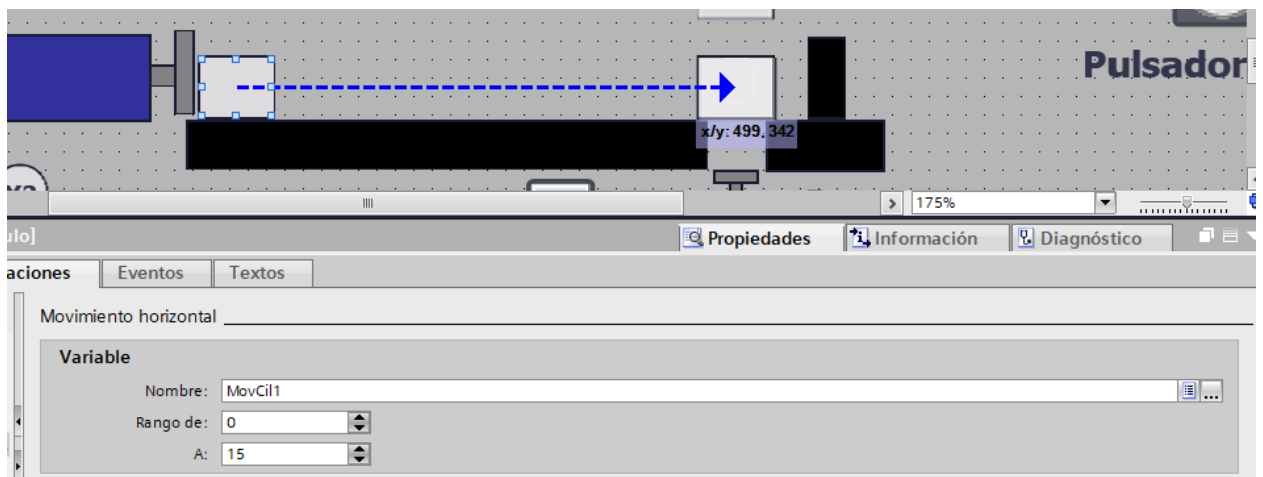


Figura 92. Ajustes del caso ambos elementos están en contacto desde el principio del movimiento





Para el segundo caso, como ya explicamos en el apartado 5.2.3, el rango de la variable tiene que ajustarse a cuando el cilindro este en contacto con la pieza.

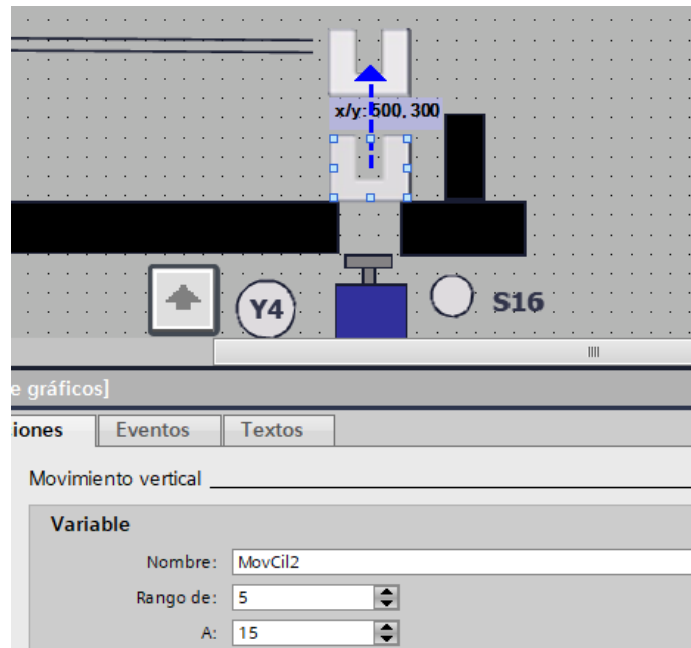


Figura 93. Ajustes del caso en el que los elementos no están en contacto desde el principio del movimiento

Para el último estado de la caja, donde es soplado por aire comprimido, utilizaremos el método de los temporizadores.

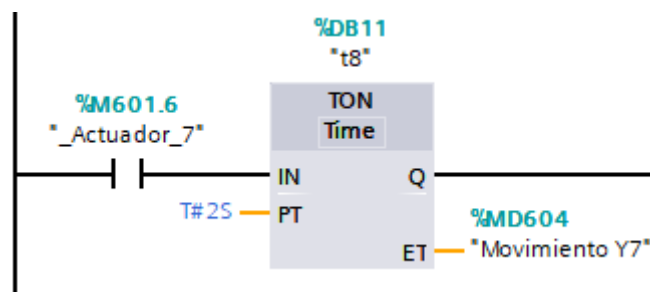


Figura 94. Temporizador del que depende el movimiento

Para poder ajustar con mayor facilidad la variable, le aplicaremos un escalado lineal.

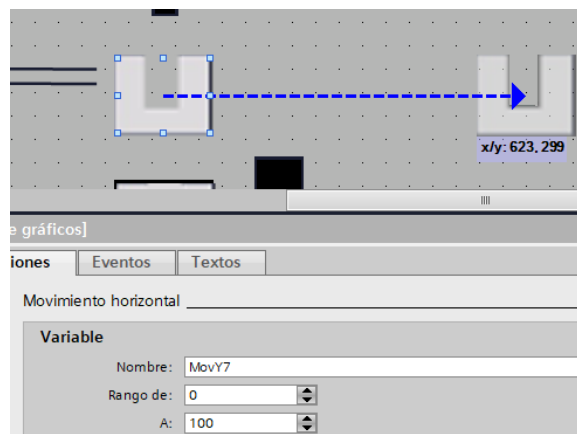


Figura 95. Ajuste del movimiento basado en temporizadores: Variable y rango



En esta maqueta podemos apreciar la utilidad de los dos tipos de formas de generar el movimiento: El método de los temporizadores es útil cuando necesitas generar un movimiento en un solo sentido y sin necesidad de controlar de forma precisa su posición. En esta maqueta, en la etapa en la pieza es soplada, solo necesitamos que la pieza avance un poco, no necesitamos situarla en ningún lugar en concreto, por lo que el método de los temporizadores es el idóneo.

El método de los contadores es útil cuando necesitamos generar un movimiento en dos sentidos (avance y retroceso) y controlar exactamente su posición. Este método es el idóneo para representar el movimiento de los cilindros.

Para configurar los finales de carrera procedemos de la misma forma que en el apartado 5.1.4

#### 5.4.4 Visibilidad y apariencia

El objeto que necesitara ajustes de visibilidad es la pieza que va a ser perforada.

Cuando el sistema alcanza su posición inicial, la pieza se coloca junto al cilindro 1 y será llevada hasta la posición donde será taladrada.

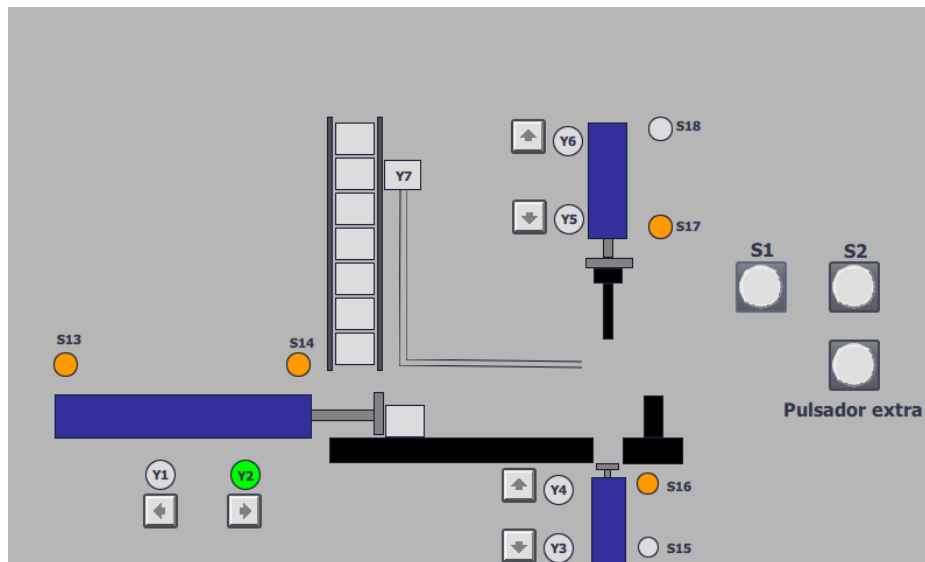


Figura 96. Vista de la pieza inicial

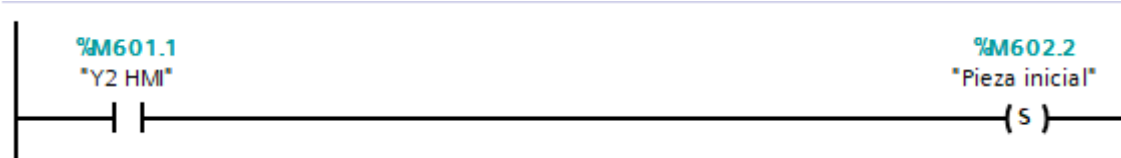


Figura 97. Visibilidad de la pieza inicial

Cuando el cilindro llega al final de su recorrido, la pieza queda fijada y es taladrada. Al ser taladrada tenemos que ocultar la pieza original y hacer visible la pieza taladrada



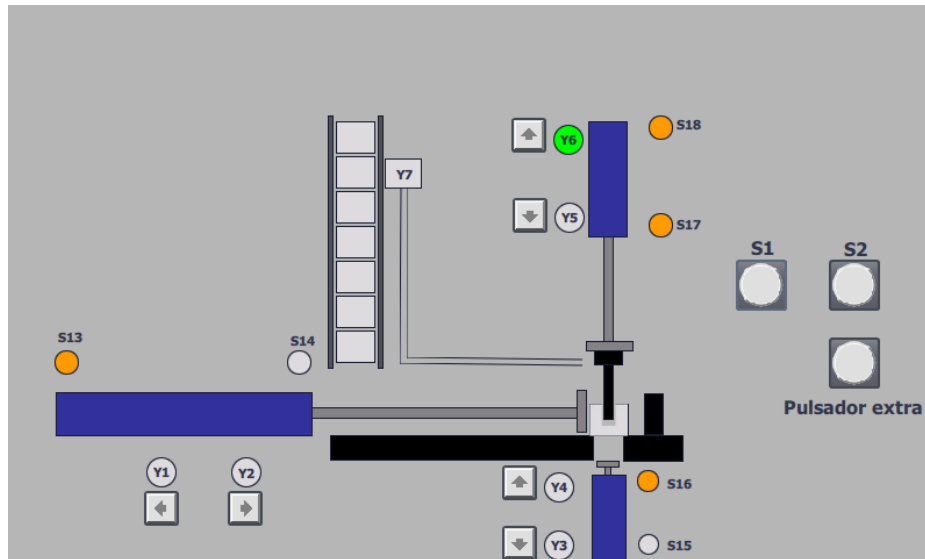


Figura 98. Vista de la pieza perforada

La pieza será completamente taladrada alcance el final de carrera S17, por lo que procedemos con la figura 99.

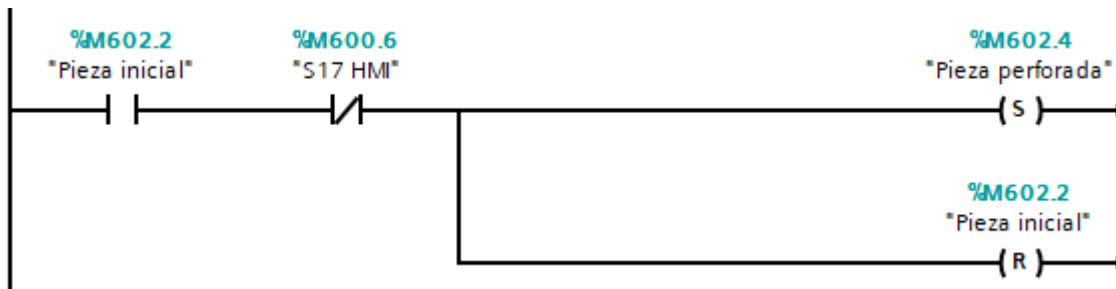


Figura 99. Visibilidad de la pieza perforada

Cuando el cilindro que eleva la pieza perforada alcanza el final de carrera S16, el aire comprimido empuja la pieza, por lo que tenemos que ocultar la pieza que se ha movido en verticalmente y debemos hacer visible la pieza que realizara el movimiento horizontal:

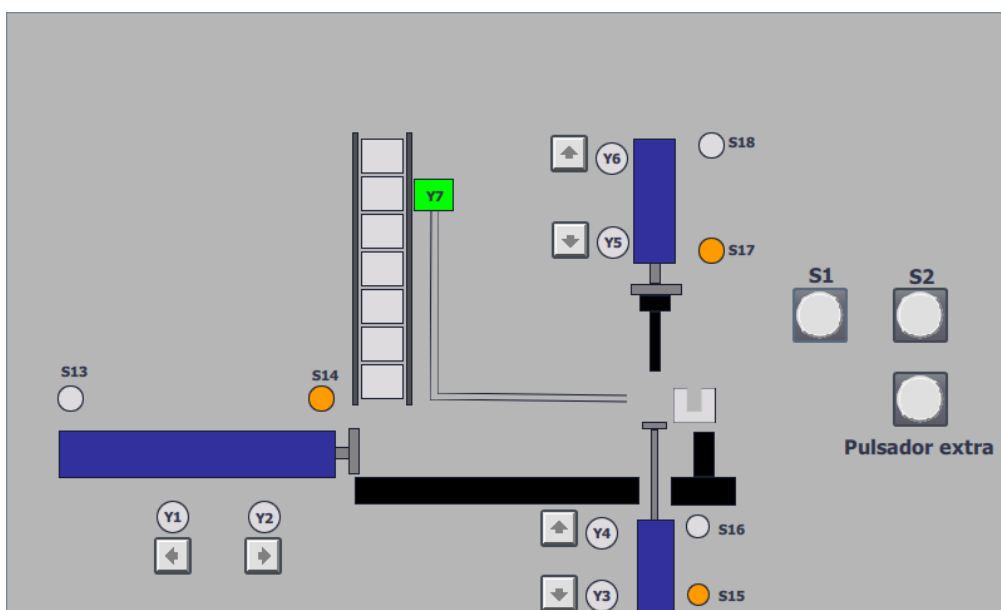


Figura 100. Vista de la pieza movida por el aire comprimido



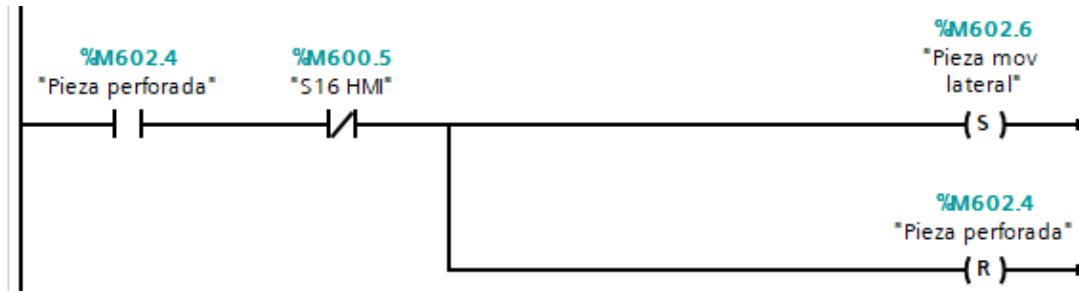


Figura 101. Visibilidad de la pieza movida por el aire comprimido

Cuando Y7 se desactiva, el cilindro que había elevado la pieza vuelve a su posición inicial, haciendo visible la pieza inicial y ocultando la pieza que ya ha sido perforada y desplazada por el aire comprimido, volviendo al estado inicial.

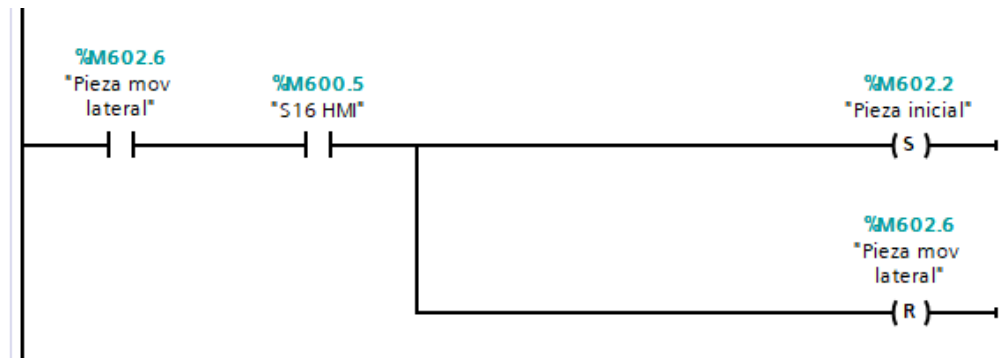


Figura 102. Visibilidad de la pieza inicial

Los elementos que requieren de ajustes de apariencia son las señales luminosas (indicadores del estado de los finales de carrera y del estado de los actuadores). Para configurarlos, procedemos de la misma forma que en el apartado 5.1.4



## 5.5 Habitación de aire limpio

### 5.5.1 Enunciado

Para mantener una habitación con aire tan limpio como sea posible, se ha instalado una cámara para un hombre con puerta corredera. Similar a la de algunos bancos.

Para entrar a la habitación, hay que pasar a través de la esclusa de aire limitada por 2 puertas correderas. Para entrar hay que presionar los botones S1 o S2 en función si se entra a la habitación o se sale. Dentro de la esclusa no hay interruptor de seguridad para abrir las puertas desde el interior. Por lo tanto, es esencial que ambas puertas nunca permanezcan cerradas al mismo tiempo.

Si una persona desea entrar en la esclusa de aire por la puerta 1 y está cerrado, se presiona el botón S1. La Puerta 2 se cierra inmediatamente y la puerta 1 se abre tan pronto como el final de carrera S15 de la puerta 2 ha indicado que la puerta 2 está completamente cerrada.

Al entrar en la esclusa de aire, la barrera B1 luz produce un pulso y después de un retraso de 3 segundos, la puerta 1 se cierra. Tan pronto como finales de carrera S14 indica que la puerta 1 está completamente cerrada, la puerta 2 se abre.

La secuencia correspondiente se aplica, cuando una persona entra en la puerta del lado 2.

Si se introduce la esclusa de aire a través de una puerta abierta, sin pulsar previamente el botón S1 o S2, las puertas no deben reaccionar.

Cada entrada en la esclusa de aire se controla mediante una barrera de luz. Mientras la barrera de luz este interrumpida, la puerta no se debe cerrar. Si la barrera de la luz es interrumpida mientras que la puerta se cierra, la acción debe ser inmediatamente detenida y cuando la obstrucción se elimina de la barrera de la luz, la acción debe continuar.

Las luces de los botones de solicitud indican que el control ha identificado la solicitud. Las lámparas se apagan cuando se introduce la esclusa de aire (Se traspasa la barrera)

Si el botón S1 o S2 se pulsa durante el cambio de aire en la cámara de aire, esto no debería tener ningún efecto sobre el procesamiento del sistema. La solicitud se memoriza y luego se procesa.

NOTA: Los finales de carrera de las puertas son NC, funcionan en modo inverso, es decir, se iluminan ("1") cuándo no están pulsados y viceversa

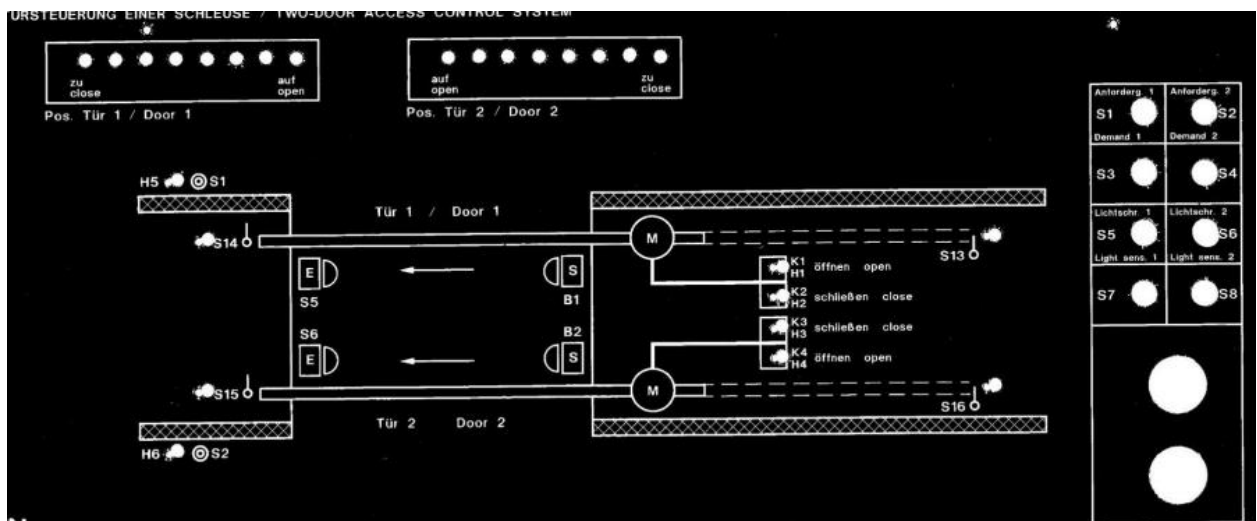


Figura 103. Plantilla de la habitación de aire limpio que encontramos en el laboratorio



### 5.5.2 Representación grafica

La representación que hemos desarrollado para el proceso es la siguiente:

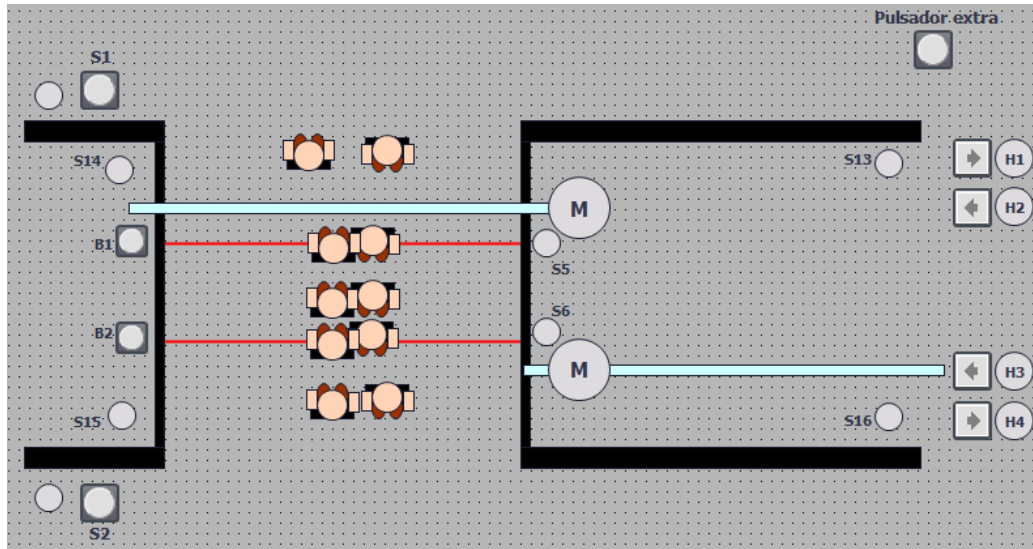


Figura 104. Representación de la habitación de aire limpio en el HMI

En ella podemos ver todos los botones que controlan el proceso (S1, S2, B1, B2 y el pulsador extra), los indicadores de final de carrera S13, S14, S15 y S16 (normalmente cerrados), los indicadores del estado de las barreras luminosas S5 y S6 y los indicadores del estado de las salidas H1, H2, H3 y H4. Además, podemos observar los diferentes estados en los que se pueden encontrar las personas durante el proceso.

### 5.5.3 Movimiento

Para generar el movimiento utilizamos el mismo método que en la maqueta del apartado 5.1, donde el movimiento es generado a partir de temporizadores en bucle conectados a un contador CTUD. Cada puerta tiene su propia variable se mueve de forma independiente respecto a la otra puerta

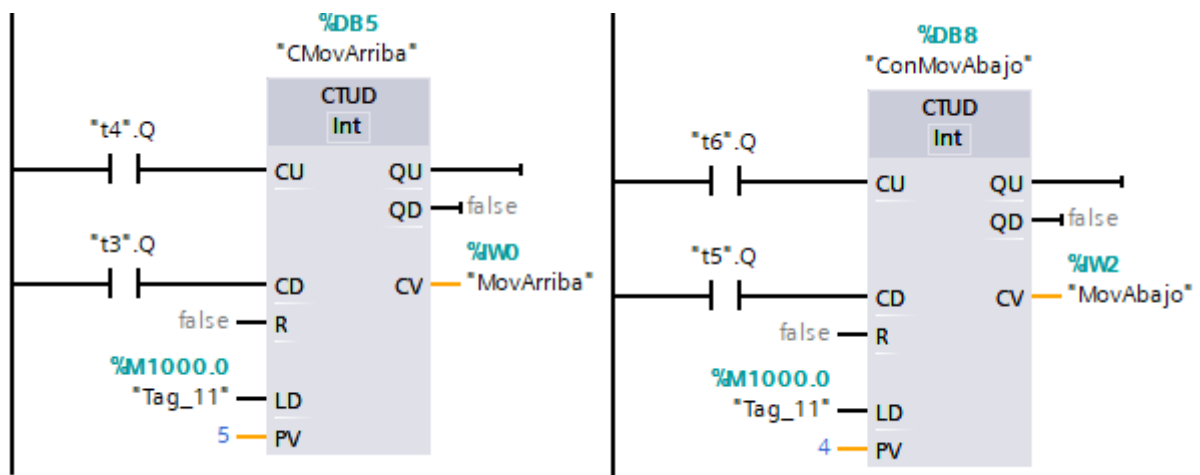


Figura 105. Contadores CTUD de los que depende el movimiento

Para configurar las barreras luminosas procedemos de la siguiente forma: Con el first scan creado manualmente, fijamos las variables a 1, como podemos ver en la figura 106.



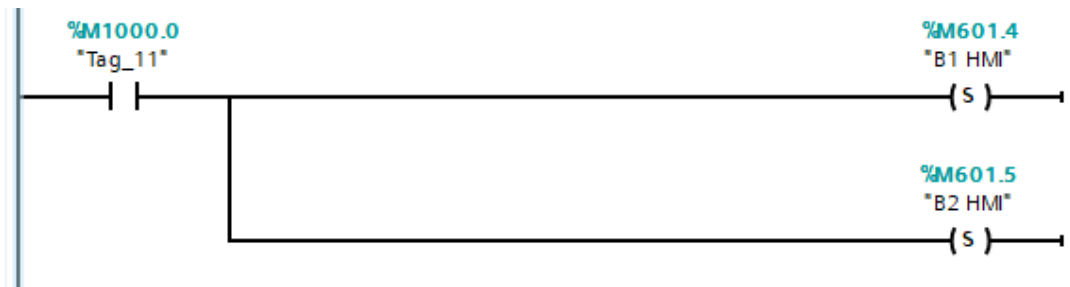


Figura 106. Ajuste para que las barreras de luz sean normalmente cerradas

Después en los eventos del botón (explicado en el apartado 4.2.2.3), lo configuramos para que cuando se pulse el botón, la variable se desactive y cuando se suelte el botón, la variable se vuelva a activar

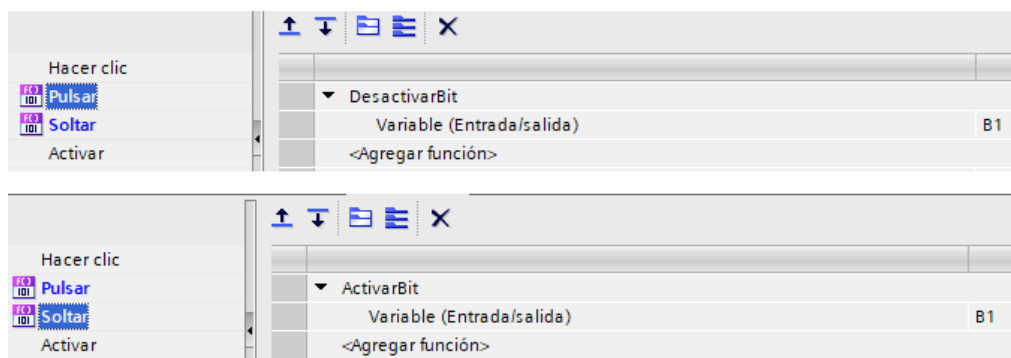


Figura 107. Configuración de los botones de las barreras de luz

Para configurar los finales de carrera procedemos de la misma forma que en el apartado 5.1.4.

### 5.5.4 Visibilidad y apariencia

Para representar de forma adecuada el proceso vamos a necesitar ajustar la visibilidad de las personas a lo largo del proceso. Solo vamos a explicar cómo hemos configurado la visibilidad de la persona entrando por la puerta de arriba ya que la visibilidad de la persona entrando por la puerta de abajo se realiza de la misma forma.

Cuando la persona quiere entrar, debe pulsar el botón S1 y esperar a que la puerta se abra. Por lo que debemos hacer visible a la persona que está esperando en la puerta.

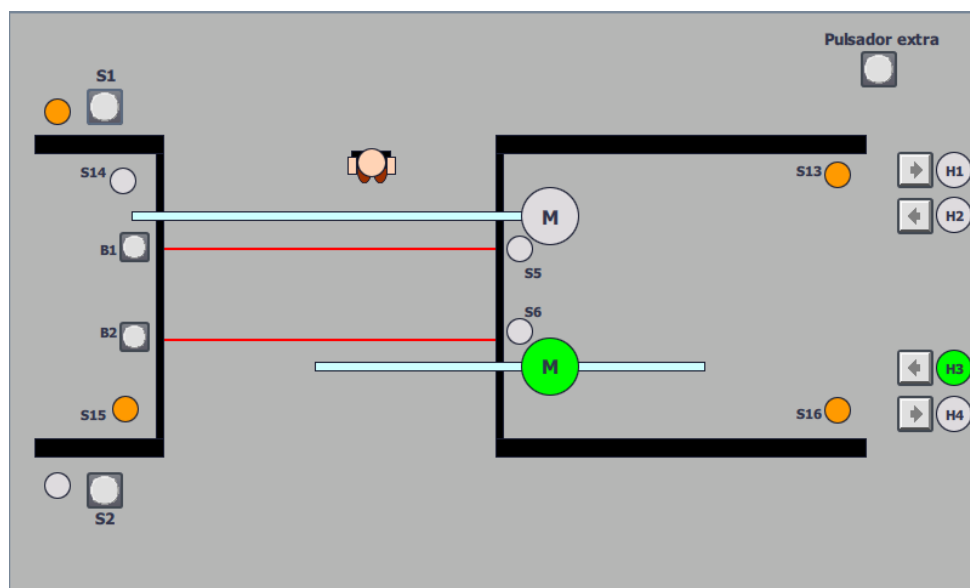


Figura 108. Vista de la persona esperando a entrar



Como más adelante vamos a tener el caso en el que haya más de una persona en pantalla, necesitamos crear otra variable que explicaremos en el siguiente paso.

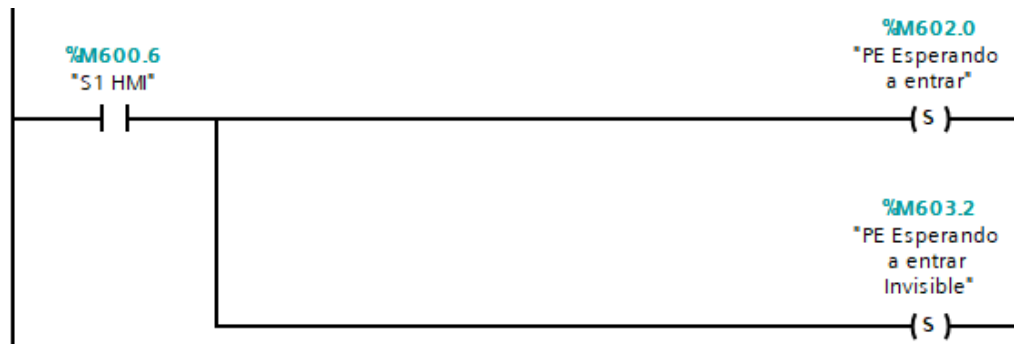


Figura 109. Visibilidad de la persona esperando a entrar

Cuando la puerta se abre, la persona entra a la esclusa (pulsando el botón B1), pasando por la barrera de luz :

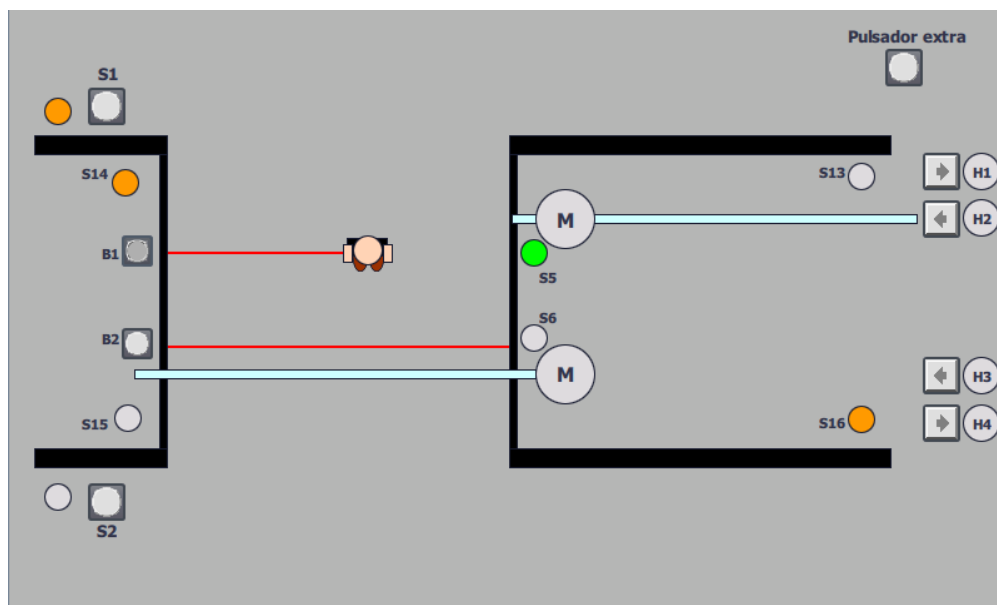


Figura 110. Vista de la persona cruzando el haz de luz B1

En este paso haremos invisible a la persona que está esperando a que la puerta se abra (resetearemos su variable), pero gracias a que no hemos reseteado la variable extra 'PE Esperando a entrar invisible' podemos hacer que aun entren personas a la esclusa.

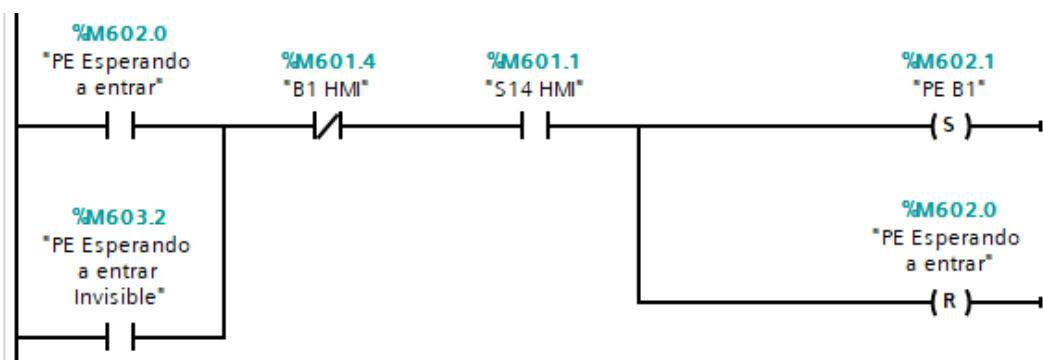


Figura 111. Visibilidad de la persona cruzando el haz de luz B1





Cuando la persona cruza la barrera de luz (soltando el botón B1), la persona está en la posición central de la esclusa esperando a que la puerta inferior se abra. En este momento, hasta que la puerta superior se cierre, aún pueden entrar personas por la puerta superior, por eso mantenemos la variable 'PE Esperando a entrar invisible' activa.

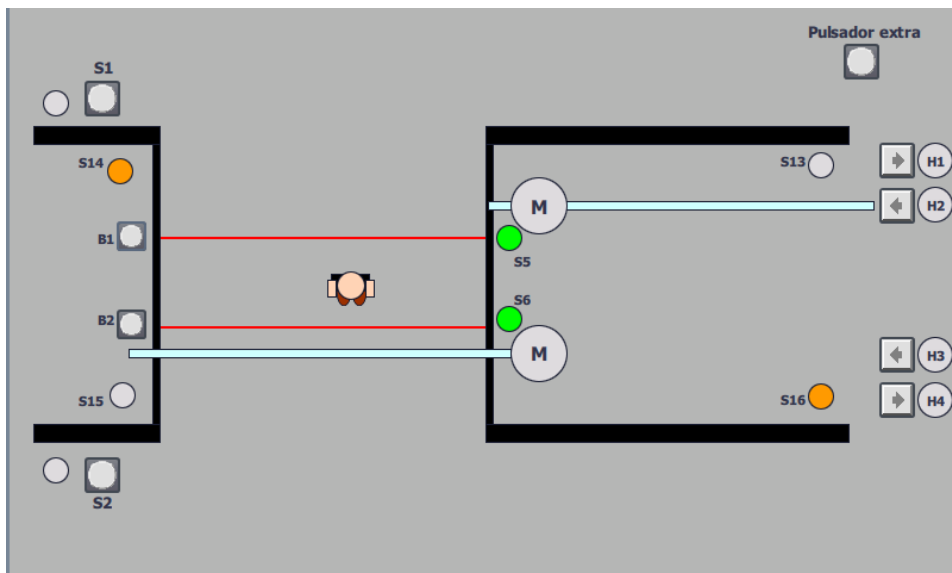


Figura 112. Vista de la persona en el centro de la esclusa

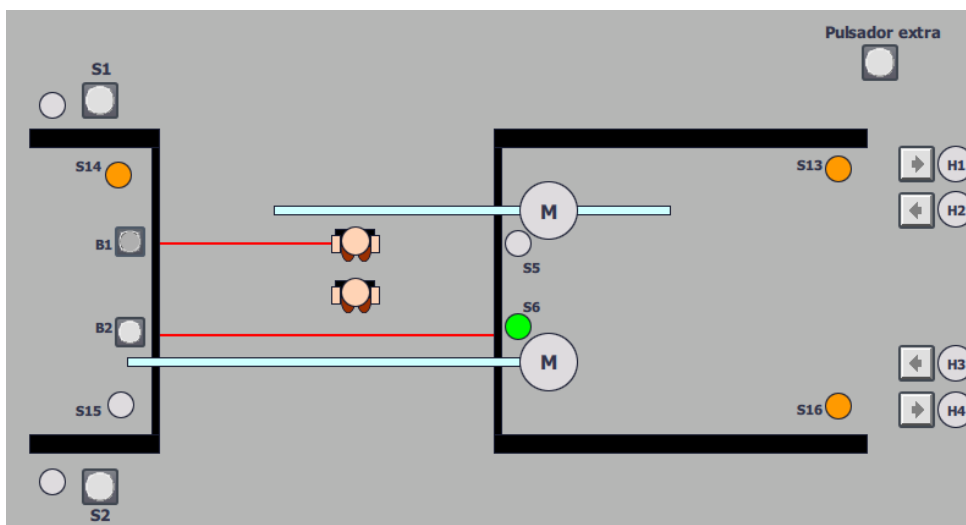


Figura 113. Vista de la persona cruzando el haz de luz mientras hay una persona en el centro de la esclusa

En la figura 114 podemos ver como cuando la barrera no detecta ningún cuerpo (B1 normalmente cerrado), activa la visualización de la persona en el centro de la esclusa y oculta la persona sobre la barrera de luz. La activación de la variable 'PE En medio Invisible' tiene la misma función que la variable 'PE Esperando a entrar invisible' que antes hemos creado, pero esta vez para salir por la puerta inferior.



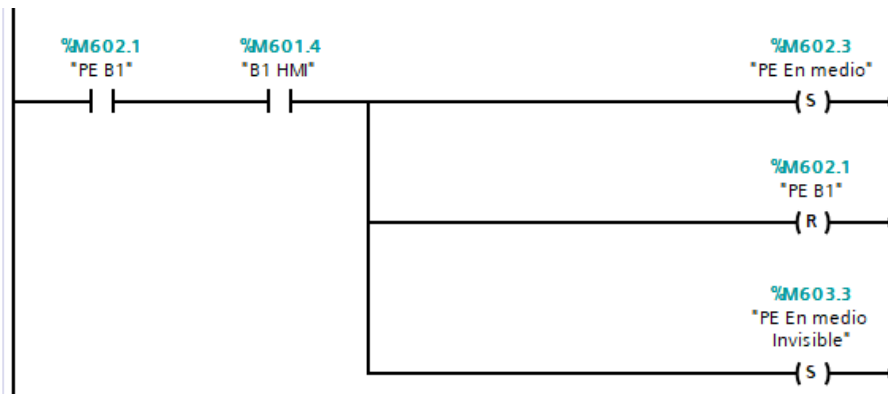


Figura 114. Visibilidad de la persona en el centro de la esclusa

Cuando la puerta inferior se abre y la persona cruza la barrera luminosa (pulsando B2), la persona sobre la luz luminosa se hace visible y la persona del centro de la esclusa se oculta:

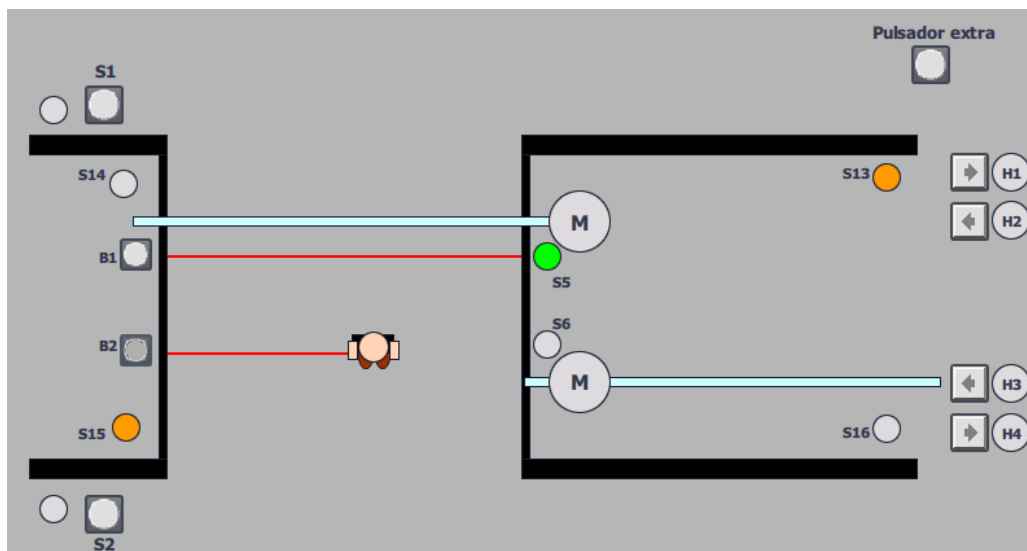


Figura 115. Vista de la persona cruzando el haz de luz B2

En este punto ya no pueden entrar más personas por la puerta superior, por lo que la variable 'PE Esperando a entrar invisible' tenemos que resetearla.

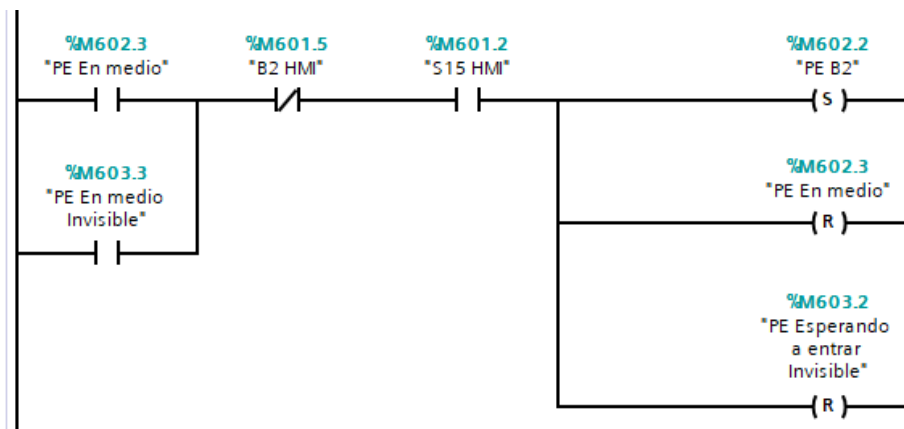


Figura 116. Visibilidad de la persona cruzando el haz de luz B2



Una vez la persona a salido de por la puerta (soltando B2), tenemos que ocultar la persona sobre la barrera luminosa y hacer visible la persona fuera de la esclusa. En este punto, pueden seguir saliendo personas de dentro de la esclusa, por lo que la variable que nos permitía que varias personas salieran de la esclusa, 'PE En medio Invisible', aún sigue activa:

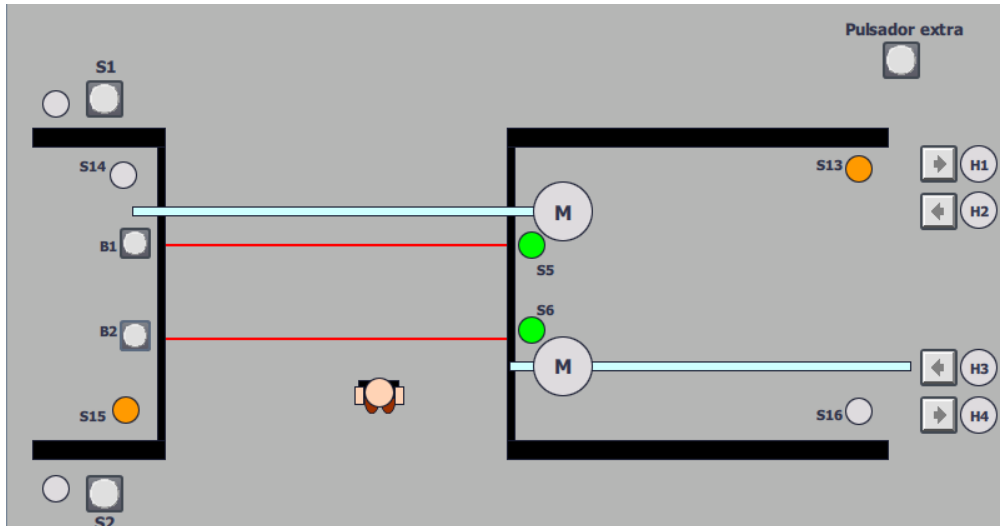


Figura 117. Vista de la persona que ha cruzado la puerta

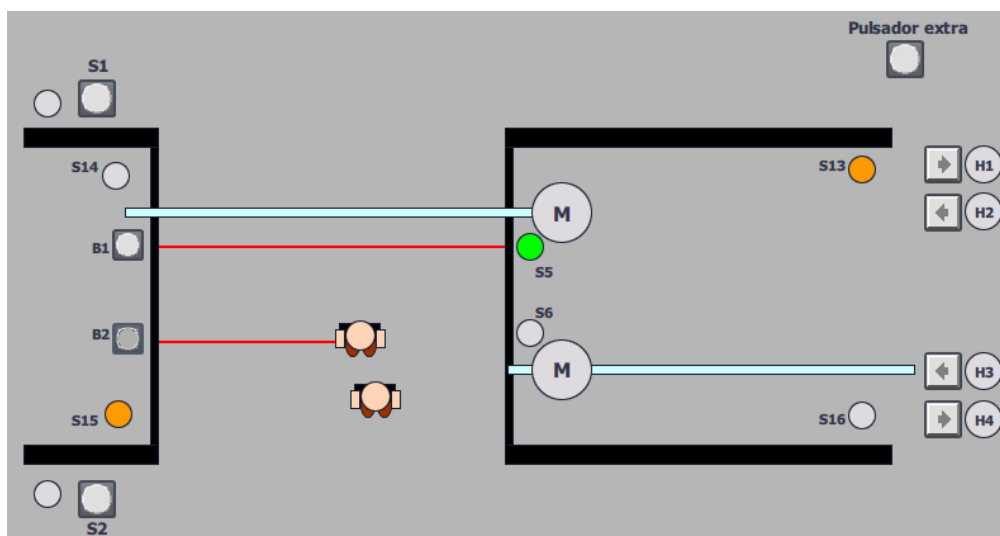


Figura 118. Vista de la persona cruzando el haz de luz mientras que otra ha cruzado la puerta

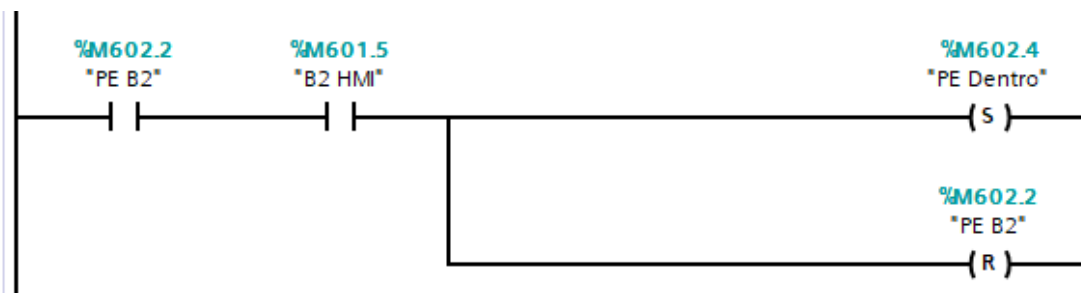


Figura 119. Visibilidad de la persona que ha cruzado la puerta

En el momento en el que la puerta inferior comience a cerrarse, la persona que acaba de entrar se ocultara y, cuando se cierre completamente, dejaran de poder salir personas.



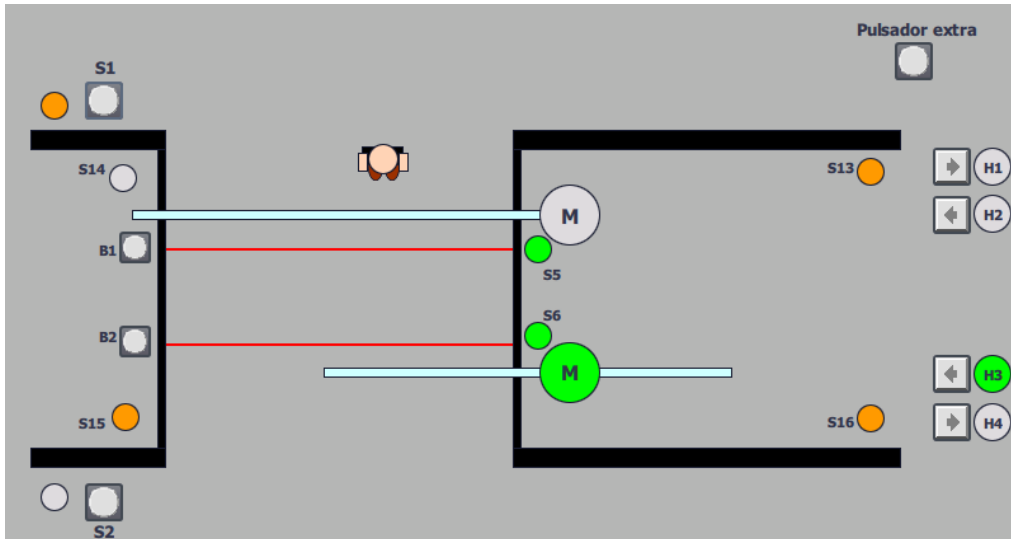


Figura 120. Momento en el que la persona que ya ha cruzado la puerta se oculta

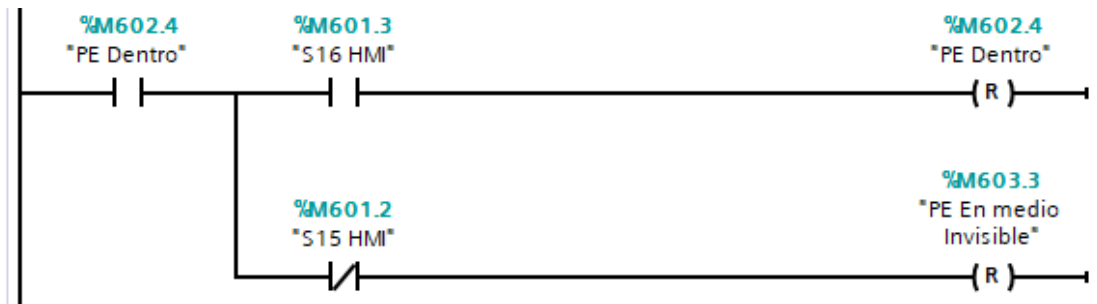


Figura 121. Configuración de la variable para que la persona que ha cruzado la puerta se oculte

Las barreras luminosas también necesitan ajuste de visibilidad para que, cuando una persona las esté cruzando, estas actúen como tal. Para ello hemos dividido la línea roja que representa el haz de luz en dos. La parte derecha del haz de luz será el único que se hará invisible cuando una persona lo cruce.

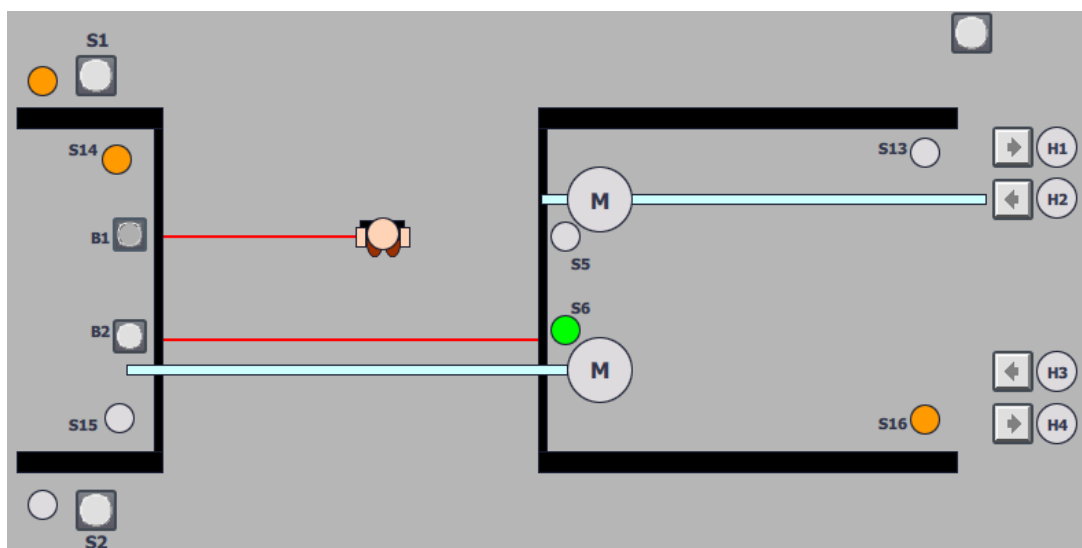


Figura 122. Vista del haz de luz ocultándose



Crearemos una variable de la que dependerá la visibilidad del haz de luz como en la figura 123. Esta variable estará activa cuando una persona cruce el haz de luz, ya sea entrando o saliendo. Cuando la variable este activa ocultaremos la parte derecha del haz de luz

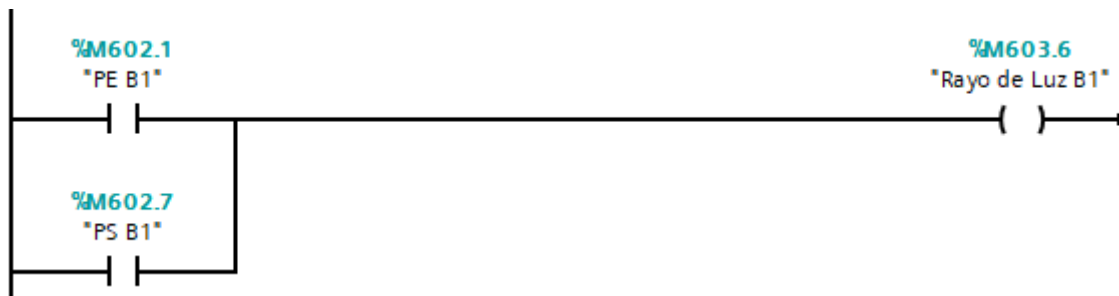


Figura 123. Visibilidad del haz de luz

Los elementos que requieren de ajustes de apariencia son las señales luminosas (indicadores del estado de los finales de carrera y del estado de los actuadores). Para configurarlos, procedemos de la misma forma que en el apartado 5.1.4



## Capítulo 6. Conclusión

El objetivo principal de este trabajo era desarrollar los entornos virtuales que reproduzcan los procesos industriales que simulamos en las prácticas de la asignatura Automatización Industrial. Esto habilitaría a los alumnos a trabajar y estudiar la asignatura desde cualquier ordenador, solucionando los problemas que presentaba el solo poder trabajar la asignatura desde el laboratorio y del reducido número de maquetas y PLCs que encontramos en el mismo.

Hemos conseguido un manejo adecuado del programa TIA Portal y de las herramientas que nos ofrece para desarrollar una metodología que representa y reproduce los diferentes casos que podemos encontrarnos en el programa de prácticas y que, además, nos permite crear nuevos casos.

Esta metodología consiste en combinar los ajustes de movimiento y de visibilidad para representar los procesos industriales. En el apartado de movimiento hemos desarrollado dos técnicas que usaremos según lo que requiera el proceso. El apartado de visibilidad soluciona las carencias que presenta el sistema de movimiento de TIA Portal. Ajustando su visibilidad, podemos representar un objeto moviéndose en las dos dimensiones de la pantalla. Además, gracias a los ajustes de apariencia, podemos configurar las señales luminosas del proceso.

Gracias al desarrollo de esta metodología, cualquier alumno podrá estudiar la asignatura y llevar a cabo las prácticas desde cualquier ordenador. Esto supone un avance en la enseñanza de la asignatura y hará más accesible la asignatura a los alumnos.

El método de los entornos virtuales frente al método de las maquetas de Lucas Nuelle presentan las siguientes ventajas e inconvenientes:

### -Ventajas

- Los entornos virtuales que hemos desarrollado, las licencias itinerantes y el programa de simulador de PLCs (PLCSIM), nos permiten trabajar y practicar con TIA Portal desde cualquier sitio.
- La metodología que hemos desarrollado para elaborar los entornos virtuales de los diferentes procesos industriales nos permite desarrollar todo tipo de procesos, por lo que podríamos ampliar el catálogo de prácticas de la asignatura.
- Con el método de los entornos virtuales es mucho más fácil ver que está pasando durante el proceso.

### -Inconvenientes

- El sistema que tiene TIA Portal para representar los procesos gráficamente es muy simple, al igual que el sistema de movimiento, lo que hace que a veces los procesos no puedan ser representados fielmente a como son en realidad.
- Los programas requeridos para utilizar los entornos virtuales que hemos desarrollado son muy pesados, por lo que necesitamos un ordenador con buen hardware para poder mover el programa de forma adecuada.



## Bibliografía

[1] '¿Qué es TIA Portal?'

[https://www.festo.com/es/es/e/tendencias/tia-portal-id\\_828990/](https://www.festo.com/es/es/e/tendencias/tia-portal-id_828990/)

[2] 'Que es un PLC y como funciona'

<https://industriasgsl.com/blogs/automatizacion/que-es-un-plc-y-como-funciona/>

[3] 'La interfaz entre el proceso y los operarios'

<https://www.aveva.com/es->

[es/solutions/operations/hmi/#:~:text=La%20Interfaz%20Hombre%2DM%C3%A1quina%20\(HMI,panel%20de%20control%20del%20operario.](https://www.aveva.com/es-solutions/operations/hmi/#:~:text=La%20Interfaz%20Hombre%2DM%C3%A1quina%20(HMI,panel%20de%20control%20del%20operario.)

[4] 'Temporizadores en TIA Portal-Como funcionan'

<https://ingelearn.com/blog/temporizadores-en-tia-portal-como-funcionan/>

