



**Universidad
de La Laguna**

ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Trabajo de Fin de Grado

**“Desarrollo de Sistema SCADA para la
automatización de células de fabricación”**

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Autor: Javier Rodríguez de la Rosa

Tutor: Roberto Luis Marichal Plasencia

ÍNDICE DE CONTENIDO

| | | |
|---------|--|----|
| 1. | Abstract | 1 |
| 2. | Resumen..... | 1 |
| 3. | Objetivos | 2 |
| 4. | SCADA | 3 |
| 4.1 | Estaciones FESTO..... | 5 |
| 4.1.1 | Estación 0..... | 6 |
| 4.1.2 | Estación 1 | 8 |
| 4.1.3 | Estación 2..... | 10 |
| 4.1.4 | Estación 3..... | 12 |
| 4.1.5 | Estación 4..... | 14 |
| 4.2 | PLC's..... | 16 |
| 4.2.1 | Programación en TIA Portal | 17 |
| 4.2.2 | Lenguajes de programación utilizados | 18 |
| 4.2.3 | Modelos seleccionados | 23 |
| 4.2.4 | Módulos de ampliación..... | 25 |
| 4.3 | COMUNICACIONES..... | 27 |
| 4.3.1 | Protocolo PROFINET | 27 |
| 4.3.2 | Detalles de las comunicaciones | 27 |
| 4.3.3 | Bloques GET y PUT | 28 |
| 4.4 | HMI..... | 31 |
| 4.4.1 | Modelo de HMI utilizado | 32 |
| 4.4.2 | WinCC | 33 |
| 4.4.3 | Creación de la pantalla HMI..... | 35 |
| 4.4.3.1 | Configuración de la pantalla | 35 |
| 4.4.3.2 | Imagen interactiva..... | 36 |
| 4.4.3.3 | Objetos de la pantalla..... | 36 |
| 4.4.3.4 | Método de creación de objetos personalizados..... | 38 |
| 4.4.3.5 | Programación de los objetos | 39 |
| 4.4.3.6 | Elementos generales de las imágenes interactivas..... | 45 |
| 4.4.3.7 | Imagen interactiva manual y automática | 47 |
| 4.4.3.8 | Órdenes de control | 52 |
| 4.4.3.9 | Detalles del funcionamiento de los objetos en modo automático..... | 57 |
| 5. | Presupuesto | 62 |
| 6. | Conclusion | 63 |

| | | |
|----|----------------------|----|
| 7. | Conclusión | 63 |
| 8. | Plan de mejora | 64 |
| 9. | Bibliografía | 65 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Esquema del SCADA | 4 |
| Figura 2. Estaciones FESTO | 5 |
| Figura 3. Imagen de la Estación 0 | 6 |
| Figura 4. Imagen de la Estación 1 | 8 |
| Figura 5. Imagen de la Estación 2 | 10 |
| Figura 6. Imagen de la Estación 3 | 12 |
| Figura 7. Imagen de la Estación 4 | 14 |
| Figura 8. Tabla de observación | 17 |
| Figura 9. Fragmento de código en KOP | 19 |
| Figura 10. Estructura de una instrucción | 20 |
| Figura 11. Estructura de la palabra de estados | 21 |
| Figura 12. Fragmento de código en AWL | 22 |
| Figura 13. Fragmento de código en AWL | 23 |
| Figura 14. Modelos S7-1200 y S7-1500 respectivamente | 25 |
| Figura 15. Estructura de un bloque GET | 29 |
| Figura 16. Estructura de un bloque PUT | 31 |
| Figura 17. Panel HMI utilizado | 32 |
| Figura 18. Diagrama de tipos de WinCC | 34 |
| Figura 19. Conexión entre HMI y PLC | 35 |
| Figura 20. Imagen raíz inicial | 36 |
| Figura 21. Objetos básicos | 37 |
| Figura 22. Objetos específicos | 37 |
| Figura 23. Iconos de la librería | 38 |
| Figura 24. Vista de la ventana de Eventos | 39 |
| Figura 25. Vista de la ventana de Visibilidad | 41 |
| Figura 26. Vista de la ventana de Apariencia | 43 |
| Figura 27. Vista de la trayectoria del movimiento | 43 |
| Figura 28. Fragmento de código con temporizador | 44 |
| Figura 29. Elementos de control internos | 45 |
| Figura 30. Elementos de control externos | 46 |
| Figura 31. Visualización de los objetos programados | 48 |
| Figura 32. Panel de Control y Visualización | 49 |
| Figura 33. Pulsador y Conmutador respectivamente | 49 |
| Figura 34. Fragmento de código de la Estación 0 | 50 |
| Figura 35. Fragmento de código de la Estación 0 | 51 |
| Figura 36. Órdenes de la Estación 0 | 52 |
| Figura 37. Órdenes de la Estación 1 | 53 |
| Figura 38. Órdenes de la Estación 2 | 54 |
| Figura 39. Órdenes de la Estación 3 | 55 |
| Figura 40. Órdenes de la Estación 4 | 56 |
| Figura 41. Imagen Modo Automático Estación 0 | 57 |
| Figura 42. Imagen Modo Automático Estación 1 | 58 |
| Figura 43. Imagen Modo Automático Estación 2 | 59 |
| Figura 44. Imagen Modo Automático Estación 3 | 60 |
| Figura 45. Imagen Modo Automático Estación 4 | 61 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Sensores y Actuadores de la Estación 0..... | 7 |
| Tabla 2. Sensores y Actuadores de la Estación 1..... | 9 |
| Tabla 3. Sensores y Actuadores de la Estación 2..... | 11 |
| Tabla 4. Sensores y Actuadores de la Estación 3..... | 13 |
| Tabla 5. Sensores y Actuadores de la Estación 4..... | 15 |
| Tabla 6. Comparación entre S7-1200 y S7-1500..... | 24 |
| Tabla 7. Selección de módulos para el S7-1500..... | 26 |
| Tabla 8. Selección de módulos para el S7-1200..... | 26 |

1. Abstract

The project consists of the design and development of a *SCADA* (Supervisory Control and Data Acquisition). *SCADA* is the set of software and hardware components that enable the supervision and control of industrial plants.

This system consists of three main elements:

- A central server where the obtained data is processed and stored.
- The transmission with the field network, where are the *PLC*'s (Programmable Logic Controller) that interact with the sensors and actuators of the process.
- Interaction with the user where the *HMI* (Human-Machine Interface) are located, which allow the operator to monitor and control the field devices.

The *SCADA* has been made for the stations of the company *FESTO* [1], these stations are small-scale manufacturing cells representing the main processes in the industry. Regarding its implementation, a type of connection has been proposed that includes an *HMI* screen, programmable controllers and the *FESTO* stations themselves.

In the communication with the devices *HMI* screens and *PLC*'s, a type of communication protocol called *PROFINET* has been used. This protocol stands out for its ease of deployment devices, improving both the speed and security of your communications. *PROFINET* uses real-time Ethernet to communicate the different connected devices via a fieldbus. In a *PROFINET* network, all devices can assume the role of supplier and consumer.

2. Resumen

El proyecto realizado consiste en el diseño y desarrollo de un sistema *SCADA* (Supervisión Control y Adquisición de Datos). Un *SCADA* es el conjunto de componentes de software y hardware que permiten la supervisión y el control de plantas industriales.

Este sistema está formado por tres elementos principales:

- Un servidor central donde se procesan los datos obtenidos y se almacenan.

- La transmisión con la red de campo, donde se encuentran los *PLC's* (Programmable Logic Controller) que interactúan con los sensores y actuadores del proceso.
- La interacción con el usuario donde se encuentran los *HMI* (Human-Machine Interface), que permiten al operario monitorizar y controlar los dispositivos de campo.

El *SCADA* ha sido realizado para las estaciones de la compañía *FESTO* [1], dichas estaciones son células de fabricación a pequeña escala que representan los principales procesos de la industria. En cuanto a su implementación se ha planteado una tipología de conexión donde se incluye una pantalla *HMI*, autómatas programables y las propias estaciones *FESTO*.

En la comunicación de los dispositivos utilizados, esto es, la pantalla *HMI* y los autómatas, se ha utilizado un tipo de protocolo de comunicación denominado *PROFINET*. Este protocolo destaca por su facilidad para la implementación de dispositivos, mejorando tanto la velocidad como la seguridad de sus comunicaciones. *PROFINET* utiliza Ethernet en tiempo real para comunicar mediante un bus de campo los diferentes dispositivos conectados. En una red *PROFINET*, todos los dispositivos pueden asumir el rol de proveedor y consumidor.

3. Objetivos

Los principales objetivos del proyecto planteado son los siguientes:

- Programación de cada una de las cinco estaciones *FESTO* utilizando lenguaje *KOP* y *AWL*.
- Simulación de las cinco estaciones *FESTO* utilizando el *PLCSIM*.
- Testeo de las estaciones sobre las estaciones reales.
- Diseño de las pantallas *HMI* correspondientes para cada una de las estaciones, incluyendo paneles interactivos para la manipulación y visualización directa de los actuadores.
- Testeo de las pantallas *HMI* sobre las estaciones reales.

- Implementación de las comunicaciones entre las distintas estaciones en base al protocolo de comunicación *PROFINET* para lograr una correcta comunicación entre dispositivos.

4. SCADA

Un sistema *SCADA* proviene del acrónimo (Supervisory Control and Data Acquisition o Supervisión Control y Adquisición de Datos). Este sistema se utiliza para monitorizar y controlar los equipos industriales de cualquier sector en distintas fases de un proceso. Gracias al *SCADA* se pueden integrar en un solo lugar los datos que hayan sido recogidos por los distintos *PLC*'s, sensores y actuadores.

Entre sus funciones principales está el examinar, recopilar y procesar datos en tiempo real. Para esta labor el *SCADA* suele contar con un sistema central donde se procesan digitalmente los datos obtenidos y los almacena en un servidor.

Está compuesto por una parte fundamental que es la transmisión con la red de campo, donde se encuentran los *PLC*'s, estos dispositivos facilitan la interacción del sistema con los sensores y actuadores del proceso.

Por otro lado, cuenta con la interacción a nivel de usuario, donde se encuentran los *HMI*. Los *HMI* permiten al *SCADA* comunicarse con los dispositivos de campo de manera sencilla y que el operario disponga de una herramienta para la monitorización y control del proceso a tiempo real.

El *SCADA* dispone de funciones más específicas, posee una gestión de base de datos con bajos tiempos de acceso donde se registran los datos con fines históricos. Además, es capaz de transmitir la información recopilada a los dispositivos de campos y otros *PC*'s (Personal Computers) tanto de forma local como remota.

El esquema del SCADA de este proyecto se muestra a continuación:

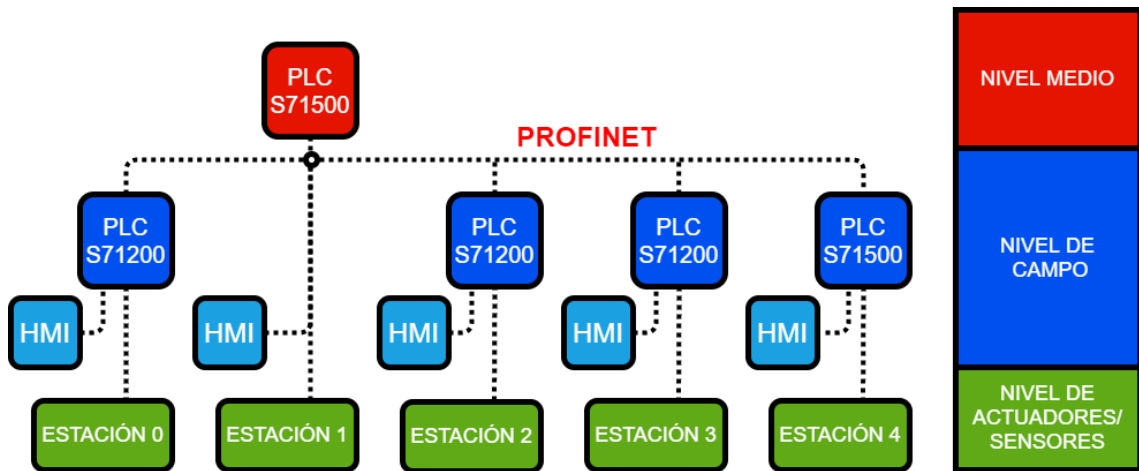


Figura 1. Esquema del SCADA

En el *nivel de actuadores y sensores*, encontramos los sensores y actuadores utilizados para la automatización de las estaciones *FESTO*.

En el *nivel de campo*, se encuentran los *PLC's* y los *HMI*. Estos dispositivos de campo facilitan la interacción del *SCADA* con el nivel de actuadores y sensores. En la *sección 4.2* y *4.4* se profundiza en detalle sobre estos dispositivos.

En el *nivel medio*, se encuentra el *PLC* utilizado como principal para la organización de las comunicaciones, este *PLC S7-1500* es el que tiene mayor jerarquía. En la *sección 4.3* se profundiza sobre la labor de este *PLC* y se explica cómo se realiza la interacción entre los diferentes niveles utilizando el protocolo *PROFINET*.

En la siguiente sección se explicará de forma detallada el funcionamiento de las estaciones *FESTO* que componen el nivel de actuadores y sensores.

4.1 Estaciones FESTO

En este proyecto se emplean como procesos las estaciones *FESTO* [2]. Estas estaciones son células de fabricación a pequeña escala que replican los principales procesos de la industria. En total se dispone de cinco estaciones cuyos objetivos son: “*Almacén y Distribución de Piezas*”, “*Testeo de Piezas*”, “*Procesado de Piezas*”, “*Acarreo de Piezas*” y “*Clasificación de Piezas*” respectivamente. Las estaciones están preparadas para funcionar de forma secuencial y formar una cadena de producción.

Estas estaciones se ubican en el Laboratorio Lorenzo Moreno Ruiz de la *ESIT* (Escuela Superior de Ingeniería y Tecnología).

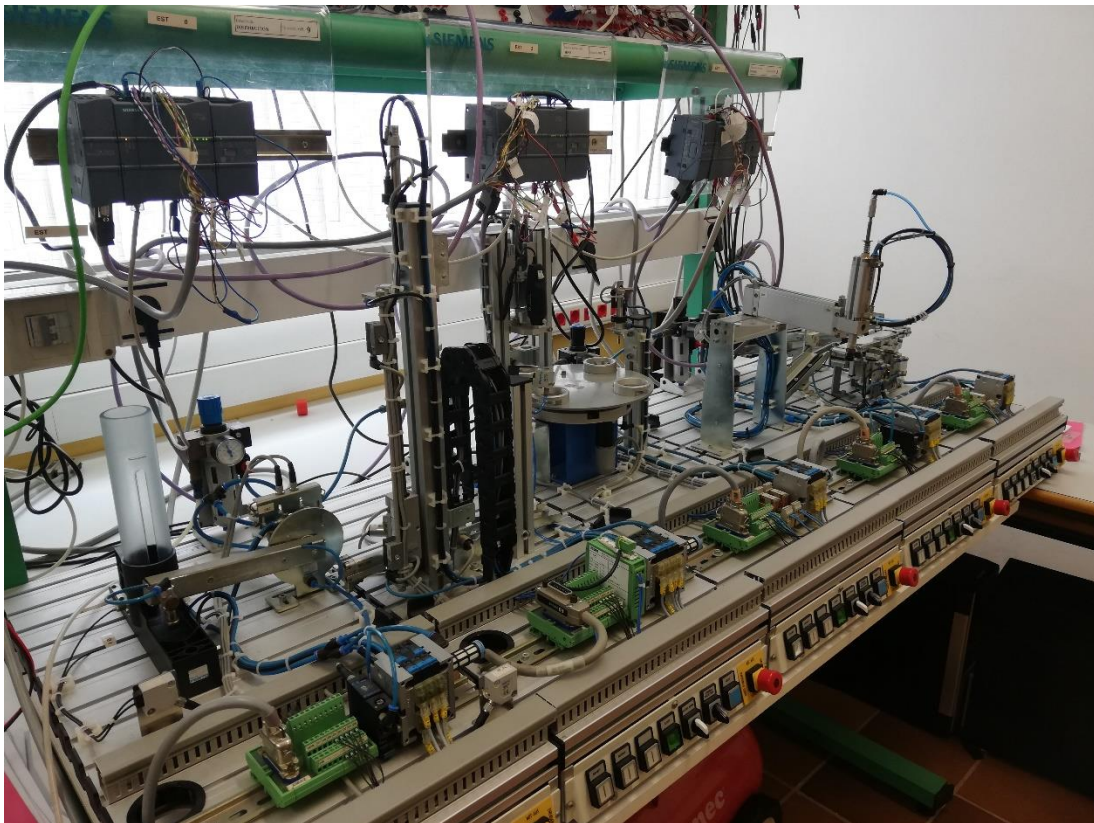


Figura 2. Estaciones FESTO

4.1.1 Estación 0

La estación 0, “*Almacén y Distribución de Piezas*”, tiene como objetivo extraer las piezas apiladas en un almacén, para posteriormente trasladarlas mediante un brazo mecánico a una plataforma.

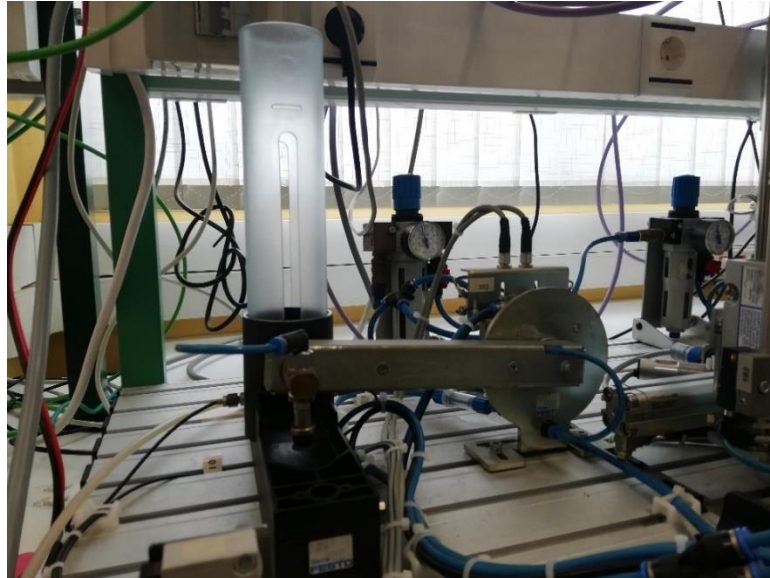


Figura 3. Imagen de la Estación 0

El brazo mecánico intercambia su posición entre el almacén y la estación siguiente para trasladar las piezas. Las piezas son extraídas del almacén mediante un pistón de expulsión.

Cuando el brazo se encuentre en la posición del almacén y tenga pieza disponible para transportar (indicado por un sensor), el brazo activa una ventosa que succiona la pieza y se mueve a la siguiente estación.

Cuando el brazo llegue a la estación siguiente, desactiva la ventosa dejando caer la pieza sobre la plataforma de elevación de la estación 1. Una vez depositada la pieza, el brazo regresa a la posición anterior donde se extraerá de nuevo una pieza del almacén.

En la *Tabla 1* se recogen las direcciones de los diferentes sensores y actuadores que utiliza la estación y una breve descripción de los mismos.

Tabla 1. Sensores y Actuadores de la Estación 0

| DIRECCIÓN S7-1200 | SENSORES | DESCRIPCIÓN |
|----------------------|--------------------------------|--|
| I0.0 | Start | Inicia que el proceso ha comenzado |
| I0.1 | Pistón extendido | Indica si el pistón de expulsión se encuentra en posición extendida |
| I0.2 | Pistón retraído | Indica si el pistón de expulsión está retraído. |
| I0.3 | Pieza sujeta en ventosa | Se activa cuando la ventosa tiene succionada una pieza |
| I0.4 | Brazo en posición de origen | Se activa cuando el brazo permanece en la posición de origen |
| I0.5 | Brazo en posición de destino | Se activa cuando el brazo permanece en la posición de destino |
| I0.6 | Pieza en inicio | Se activa cuando una pieza ha sido extraída del almacén |
| DIRECCIÓN S7-1200 | ACTUADORES | DESCRIPCIÓN |
| Q0.0 | Retraer pistón | Activa el pistón que extrae la pieza del almacén. Normalmente se encuentra extendido |
| Q0.1 | Activar succión | Activa el vacío que succiona la pieza. (Desactivar Q0.2) |
| Q0.2 | Soplar | Activa el aire que sopla la pieza. (Desactivar Q0.1) |
| Q0.3 | Mover brazo a posición previa | Motor que gira el brazo a la posición previa |
| Q0.4 | Mover brazo a posición destino | Motor que gira el brazo a la posición de destino |

4.1.2 Estación 1

La estación 1, “*Testeo de Pieza*”, tiene como objetivo principal testear el tamaño adecuado de una pieza, para ello, mide la pieza con un sensor de longitud, guarda la medida, y una vez se guarda, si la medida se encuentra dentro del rango de medida establecido, la pieza es enviada a la estación siguiente. De lo contrario, es enviada a un depósito de piezas defectuosas.

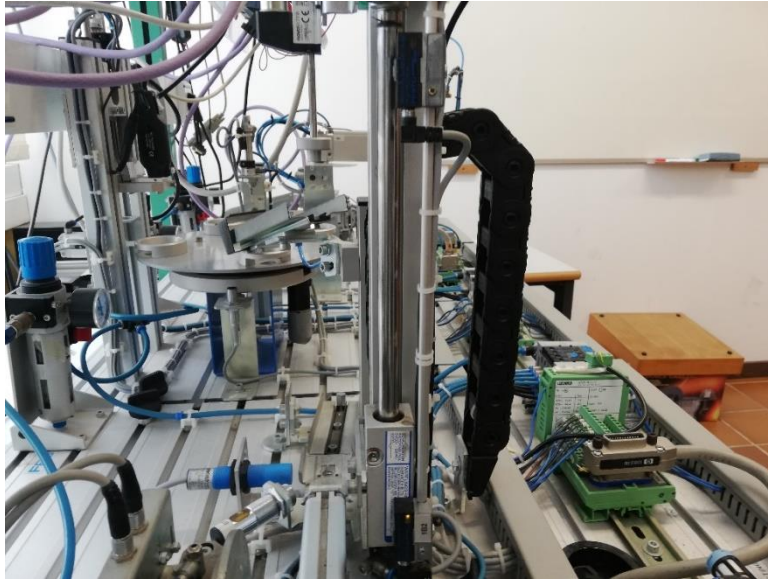


Figura 4. Imagen de la Estación 1

Cuando la estación 1 recibe la pieza de la estación 0, activa un motor que eleva la plataforma hasta alcanzar el final de carrera de su parte superior. Cuando llegue al final de carrera, se baja un sensor que mide durante un periodo de tiempo el tamaño de la pieza. Una vez el tiempo concluye, se sube el sensor y se guarda la medida. Esta medida es comparada entre los límites inferior y superior establecidos por el programador, los límites son de 21900 y 23100 (formato Word) respectivamente.

Si la pieza se encuentra dentro de los límites, es considerada como pieza “**normal**”. Por tanto, se activa el pistón de expulsión y se envía la pieza a la estación siguiente mediante una rampa por la que se desliza, en esta rampa se ubica un pistón de sujeción para que la estación 2 tenga tiempo a confirmar que puede recibir la pieza, una vez comunica que está preparada este pistón se desactiva. Cuando la pieza es enviada a la estación siguiente se baja la plataforma a la posición inicial.

Si la pieza está fuera de los límites, es considerada como pieza “defectuosa”. Por tanto, se baja la plataforma a la posición de origen y se activa el pistón de expulsión para enviar la pieza a un pequeño depósito de piezas defectuosas.

En la **Tabla 2** se recogen las direcciones de los diferentes sensores y actuadores que utiliza la estación y una breve descripción de los mismos.

Tabla 2. Sensores y Actuadores de la Estación 1

| DIRECCIÓN S7-1500 | SENSORES | DESCRIPCIÓN |
|----------------------|--|--|
| I0.0 | Detector pieza metálica | No utilizado |
| I0.1 | Detector de objeto en posición 1 | Indica cuando llega una pieza a la plataforma |
| I0.2 | Detector de color | No utilizado |
| I0.3 | Detector de final de carrera 1 | Indica si la plataforma se encuentra en la posición inferior |
| I0.4 | Detector de final de carrera 2 | Indica si la plataforma se encuentra en la posición superior |
| I0.5 | Extraer/retraer el pistón de expulsión | Se activa cuando el pistón de expulsión está extendido |
| I0.6 | Extender/retraer el sensor de medida | Se activa cuando el sensor de medida está extendido |
| IW6 | Resultado del sensor de medida | Muestra el valor de la medida |
| DIRECCIÓN S7-1500 | ACTUADORES | DESCRIPCIÓN |
| Q0.0 | Bajar la plataforma | Motor para bajar la plataforma |
| Q0.1 | Subir la plataforma | Motor para subir la plataforma |
| Q0.2 | Extender el pistón | Motor para extender el pistón que expulsa la pieza |
| Q0.3 | Bajar/subir sensor que mide la longitud de pieza | Motor para bajar el sensor de longitud, al desactivarse vuelve a subir a su posición inicial |
| Q0.4 | Activar/desactivar pistón de sujeción en la rampa 2. | Activa el pistón de sujeción de la rampa |

4.1.3 Estación 2

La estación 2, “*Procesado de Piezas*”, es el eje principal de la cadena de producción, su funcionamiento se basa en una mesa controlada por un motor. Dicha mesa tiene cuatro posiciones de parada y cada una de ellas tiene una función específica, que son:

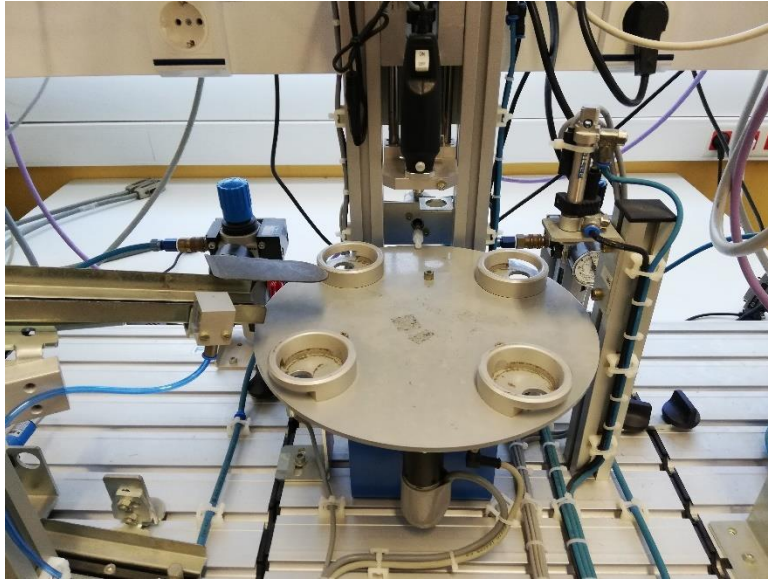


Figura 5. Imagen de la Estación 2

La **posición 1** recibe las piezas procedentes de la estación 1 a través de una rampa, cuando la pieza se encuentre en la posición correcta (indicado por un sensor), se gira la mesa 90°.

La **posición 2** sujeta la pieza con un cilindro de sujeción, a posteriori se baja un taladro realizando un orificio en la parte superior de la pieza. A continuación, el taladro se sube, colocándose en la posición inicial y la mesa gira 90°.

La **posición 3** comprueba, mediante un cilindro extensible, que el orificio realizado en la posición 2 se ha realizado de forma adecuada. Una vez realizada la comprobación, de nuevo la mesa gira 90°.

La **posición 4** es la última posición, es donde la siguiente estación recoge la pieza. Una vez verificada la recepción por parte de la estación, la mesa vuelve a girar 90° quedando en la posición inicial (posición 1).

En la **Tabla 3** se recogen las direcciones de los diferentes sensores y actuadores que utiliza la estación y una breve descripción de los mismos.

Tabla 3. Sensores y Actuadores de la Estación 2

| DIRECCIÓN S7-1200 | SENSORES | DESCRIPCIÓN |
|----------------------|--------------------------------------|--|
| I0.0 | Pieza en posición 1 | Indica que hay una pieza en la posición 1 |
| I0.1 | Mesa posición correcta | Indica que la mesa se encuentra en una de las cuatro posibles posiciones |
| I0.2 | Cilindro de sujeción retraído | Se activa cuando el cilindro de sujeción está retraído |
| I0.3 | Cilindro de sujeción extendido | Se activa cuando el cilindro de sujeción está extendido |
| I0.4 | Cilindro de taladro retraído | Se activa cuando el cilindro del taladro está retraído |
| I0.5 | Cilindro de taladro extendido | Se activa cuando el cilindro del taladro está extendido |
| I0.6 | Cilindro de prueba retraído | Se activa cuando el cilindro de prueba está retraído |
| I0.7 | Cilindro de prueba extendido | Se activa cuando el cilindro de prueba está extendido |
| DIRECCIÓN S7-1200 | ACTUADORES | DESCRIPCIÓN |
| Q0.0 | Cilindro de sujeción (simple efecto) | Motor que activa el cilindro de sujeción |
| Q0.1 | Motor del taladro ON/OFF | Motor que activa el taladro |
| Q0.2 | Motor de la mesa ON/OFF | Motor que hace girar la mesa |
| Q0.3 | Bajar taladro | Motor que baja el taladro |
| Q0.4 | Subir taladro | Motor que sube el taladro |
| Q8.0 | Cilindro de prueba (simple efecto) | Motor para bajar el cilindro de prueba que comprueba la pieza |

4.1.4 Estación 3

La estación 3, “*Acarreo de Piezas*”, tiene como objetivo mover las piezas de la estación 2 a la estación 4.



Figura 6. Imagen de la Estación 3

La estación está compuesta por un brazo giratorio con la posibilidad de extenderse de forma vertical y horizontal, esto le permite acercarse a las estaciones adyacentes para coger o dejar las piezas que procesa.

Inicialmente el brazo es movido por el motor hacia la estación 2, una vez alcanza esta posición (indicado por un sensor), el brazo se extiende horizontalmente y seguidamente de forma vertical. Cuando está extendido verticalmente, recoge la pieza de la estación 2 usando una ventosa para la recogida. Una vez succionada, se retrae de forma vertical y horizontal y gira 180° hacia la siguiente estación. En esta situación se puede producir dos supuestos:

- Si la estación 2 le ha comunicado que la pieza es “normal”, cuando reciba el permiso de la estación 4, se extenderá de nuevo tanto horizontal como verticalmente y se soplará la pieza, dejándola en la posición de inicio de la cinta de la estación 4.
- Si la estación 2 le ha comunicado que la pieza es “defectuosa”, se extenderá solamente el elemento vertical y se soplará la pieza dejándola en un pequeño depósito para las piezas defectuosas.

Después de realizar cualquiera de las dos acciones, el brazo regresa a su posición de origen girando de nuevo 180°.

En la **Tabla 4** se recogen las direcciones de los diferentes sensores y actuadores que utiliza la estación y una breve descripción de los mismos.

Tabla 4. Sensores y Actuadores de la Estación 3

| DIRECCIÓN S7-1200 | SENSORES | DESCRIPCIÓN |
|----------------------|---------------------------------------|---|
| I0.0 | Pieza recogida (sensor de vacío) | Indica que la pieza está en la ventosa |
| I0.1 | Brazo girado en estación posterior | Indica que el brazo se encuentra en la estación posterior |
| I0.2 | Brazo girado en estación previa | Indica que el brazo se encuentra en la estación previa |
| I0.3 | Elemento horizontal extendido | Se activa cuando el elemento horizontal está extendido |
| I0.4 | Elemento horizontal retraído | Se activa cuando el elemento horizontal está retraído |
| I0.5 | Elemento vertical extendido | Se activa cuando el elemento vertical está extendido |
| I0.6 | Elemento vertical retraído | Se activa cuando el elemento vertical está retraído |
| DIRECCIÓN S7-1200 | ACTUADORES | DESCRIPCIÓN |
| Q0.0 | Retraer elemento horizontal | Motor para retraer el elemento horizontal |
| Q0.1 | Extender elemento horizontal | Motor para extender el elemento horizontal |
| Q0.2 | Girar hacia la estación posterior | Motor que gira el brazo a la estación posterior |
| Q0.3 | Girar hacia la estación previa | Motor que gira el brazo a la estación previa |
| Q0.4 | Vacío ON | Activa el Vacío para succionar la pieza (Desactivar Q8.0) |
| Q8.0 | Vacío OFF | Quita el Vacío para depositar la pieza (Desactivar Q0.4) |
| Q8.1 | Bajar/subir elemento vertical | Motor para bajar el elemento vertical, al desactivarse vuelve a subir |

4.1.5 Estación 4

La estación 4, “*Clasificación de Piezas*”, se encarga de clasificar las piezas según el requerimiento establecido por el operario. Dispone de una cinta para mover las piezas hacia los depósitos de almacenamiento y de dos actuadores neumáticos que cortan el paso de la pieza por la cinta.

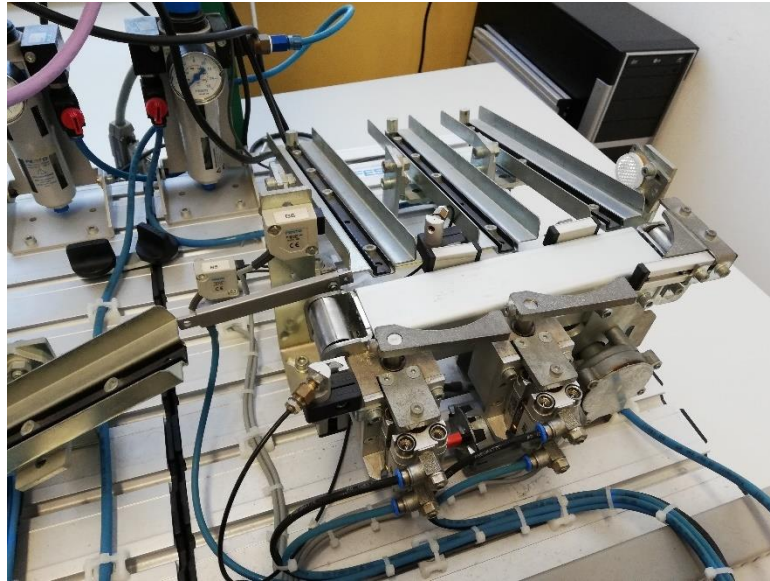


Figura 7. Imagen de la Estación 4

Una vez la estación 3 deposita la pieza al principio de la cinta, la cinta se mueve hasta que la pieza ha entrado (indicado por un sensor) a uno de los tres posibles depósitos de almacenamiento. El recorrido de la pieza viene determinado en función del desviador actuador neumático que se encuentre extendido.

Para evitar conflictos con los actuadores se impone una condición, la cinta no debe moverse hasta que los actuadores neumáticos alcancen la posición de retraído o extendido, esta información es proporcionada por los sensores ubicados en la cinta.

En la *Tabla 5* se recogen las direcciones de los diferentes sensores y actuadores que utiliza la estación y una breve descripción de los mismos.

Tabla 5. Sensores y Actuadores de la Estación 4

| DIRECCIÓN S7-1500 | | |
|-------------------|--|--|
| | SENSORES | DESCRIPCIÓN |
| I0.0 | Detecta desviador 1 en el origen | Se activa cuando el Desv.1 está retraído |
| I0.2 | Detecta desviador 2 en el origen | Se activa cuando el Desv.2 está retraído |
| I1.1 | Modo auto ('0') / Modo manual ('1') | Cuando está desactivado la estación opera en modo auto, cuando se activa, en modo manual |
| I1.0 | Botón START | Inicia el proceso, en modo manual arranca el motor de la cinta |
| I1.2 | Botón STOP | Detiene el proceso |
| I0.1 | Desviador 1 extendido | Se activa cuando el Desv.1 está extendido |
| I0.3 | Desviador 2 extendido | Se activa cuando el Desv.2 está extendido |
| I1.6 | Señal CLAS1 | Interruptor para extender el Desviador 1 |
| I1.7 | Señal CLAS2 | Interruptor para extender el Desviador 2 |
| I0.4 | Detecta pieza al principio de la cinta ('0') | Normalmente se encuentra activo, cuando una pieza lo atraviesa se desactiva |
| I0.5 | Detecta pieza entrando en almacén | Indica que una pieza ha entrado en el almacén |
| DIRECCIÓN S7-1500 | | |
| | ACTUADORES | DESCRIPCIÓN |
| Q0.0 | Extender desviador 1 | Motor que extiende el Desviador 1 |
| Q0.1 | Extender desviador 2 | Motor que extiende el Desviador 2 |
| Q0.2 | Mover cinta | Motor que mueve la cinta |

Una vez explicado cómo funcionan las estaciones *FESTO*, en la siguiente sección veremos los *PLC*'s utilizados para su automatización.

4.2 PLC's

Un autómata o *PLC* (Programmable Logic Controller) es un dispositivo que utiliza el mundo de la ingeniería para la automatización de procesos y tiene como objetivo que las máquinas realicen efectivamente todos los sistemas que la componen. Este dispositivo se encarga de procesar los datos de uno o varios procesos industriales. Estos datos provienen comúnmente de las entradas de los sistemas, es decir, los sensores del proceso. Una obtenidos los datos se ejecuta la lógica programada y se generan señales de salida que activan los diferentes actuadores del proceso.

El funcionamiento básico de un *PLC* consiste en cuatro acciones que se ejecutan cíclicamente:

1. Cada ciclo comienza con un autodiagnóstico interno del sistema, evalúa todos los circuitos y en caso de detectar algún error el dispositivo lo indica mediante una señal.
2. A continuación, se leen las entradas de los módulos de señal (coordinan la entrada y salida de las señales) y se convierten en señales binarias o digitales. Estas señales son enviadas a la *CPU* y se guardan en los datos de la memoria.
3. La *CPU* ejecuta el programa del usuario, el cual ha sido cargado previamente en la memoria y durante la ejecución del programa se generan nuevas señales de salida.
4. Por último, se actualizan las salidas. Tras la ejecución de la última parte del programa, las señales de salida se envían a los módulos de señal desde la memoria. Estas señales son convertidas en las apropiadas para cada actuador y el ciclo comienza de nuevo.

En este proyecto se utilizan cinco dispositivos *PLC's*, uno por cada estación *FESTO*, estos autómatas están conectados a una red local que se establecerá como entorno para la comunicación entre los diferentes dispositivos.

A continuación, veremos el software utilizado para programar los *PLC's* utilizados.

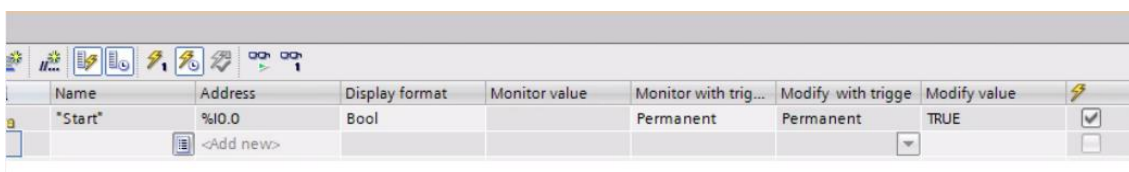
4.2.1 Programación en TIA Portal

El software *TIA Portal* proviene del acrónimo Totally Integrated Automation Portal y pertenece a la marca *SIEMENS*. El *TIA Portal* [3] es la aplicación que optimiza todas las rutinas de operaciones, máquinas y procesos. Surge por efecto del incremento en la complejidad a lo largo del tiempo de los procesos industriales y la dificultad para controlarlos, para solucionar este problema nace el *TIA*, que integra todas las rutinas en una sola aplicación permitiendo una alta accesibilidad a todo lo necesario para el control industrial.

Por otro lado, *TIA Portal* dispone de una herramienta software que permite al programador simular el proyecto creado antes de ser transferido a un *PLC* físico.

Una vez programada la lógica del proceso, se carga la configuración del programa a un *PLC* simulado en *S7-PLCSIM* [4]. Desde el *PLCSIM* se genera un *PLC* artificial que emula el comportamiento de un *PLC* real, este *PLC* ejecuta el código cargado y permite al programador observar el efecto de las entradas y salidas simuladas. En caso de ser necesario, puede cambiar la lógica del programa y volver a simular.

Para realizar la comprobación del código, *TIA Portal* cuenta con unas tablas de observación, estas tablas de observación permiten alterar el estado de las variables de entrada y observar el resultado en las variables de salida. Para modificar el estado de las variables de entrada se utiliza el forzado de variables. En la **Figura 8** se muestra el aspecto de una tabla de observación.



| Name | Address | Display format | Monitor value | Monitor with trig... | Modify with trigge | Modify value |
|-----------|---------|----------------|---------------|----------------------|--------------------|--------------|
| *Start | %I0.0 | Bool | | Permanent | Permanent | TRUE |
| <Add new> | | | | | | |

Figura 8. Tabla de observación

Las entradas del *PLCSIM* son equivalentes a las entradas físicas de una *CPU* real, por tanto, si el comportamiento del *PLCSIM* no es el correcto cuando se transfiera el programa a una *CPU* real, tampoco tendrá el comportamiento correcto.

4.2.2 Lenguajes de programación utilizados

Para la programación de los *PLC*'s de este proyecto se han utilizado dos lenguajes diferentes, el lenguaje *KOP* para el modelo *S7-1200* y el lenguaje *AWL* para el modelo *S7-1500*.

- **Lenguaje *KOP***

El lenguaje *KOP* (*Kontaktplan*) es el más común para la programación de *PLC*'s. Es un lenguaje que se caracteriza por ser visual y orientativo que funciona mediante esquema de contactos.

La lógica de estos contactos se divide en unidades pequeñas y fáciles de comprender llamadas “*Segmentos*”. Estos segmentos son ejecutados por la *CPU* en orden descendente y de izquierda a derecha. Cuando la *CPU* alcanza el final del programa (el último segmento), comienza de nuevo. Esta operación se produce muy rápido debido al ciclo del autómatas, pues en aproximadamente 1 segundo la *CPU* puede haber ejecutado todos los segmentos 50 veces, dependiendo del modelo *PLC* utilizado.

Para programar estos segmentos disponemos de tres elementos básicos:

CONTACTORES: Representan las condiciones lógicas de “entrada”, son similares a interruptores y botones. Estos elementos sirven para crear condiciones de activación de salidas. Como las variables son binarias, los contactores solo pueden tener dos estados, el ‘0’ y el ‘1’. Estos estados provienen de las entradas del *PLC* o de marcas del sistema.

Dentro de los contactos encontramos principalmente dos tipos: los contactores normalmente abiertos, que se activan cuando hay un ‘1’ lógico en el elemento que representan, y los contactos normalmente cerrados, que se activan cuando hay un ‘0’ lógico en el elemento que representan.

Por otro lado, están los flancos de subida y de bajada, estos flancos se activan cuando se produce un cambio en el valor lógico de la variable. Positivo si pasa de ‘0’ a ‘1’ y negativo si pasa de ‘1’ a ‘0’. Cuando se produce este cambio, la variable asignada activa su condición lógica.

BOBINAS: Son los elementos finales de un segmento, estas salidas pueden estar vinculadas a salidas del *PLC* o a marcas del sistema. La bobina es una operación de

asignación, la cual asigna el resultado de la operación lógica previa a la bobina a la posición de memoria correspondiente.

Para que la bobina permanezca activa, aunque la condición de asignación ya no se cumpla, se utilizan las bobinas *SET* y *RESET*, estas bobinas mantienen activadas/desactivadas la bobina hasta que la variable tome el valor contrario.

CUADROS: Los cuadros representan operaciones adicionales, como, por ejemplo, temporizadores, contadores, operaciones aritméticas o comparaciones de bits.

A todos estos elementos se les asocia una variable que en general es obtenida a partir del direccionamiento a la imagen de entradas y salidas del *PLC*.

También es posible asignarle marcas internas del sistema, estas marcas se pueden utilizar como relés de control para almacenar el estado intermedio de una operación o para indicar el estado del proceso en un punto donde no se encuentran disponibles sensores.

Para hacer más legible los diagramas de contactos se suelen utilizar las denominadas “Tablas de variables”, el objetivo de estas tablas es asignar símbolos fácilmente legibles a las entradas, salidas y marcas utilizadas en la programación.

En la **Figura 9** se muestra un ejemplo de aplicación que implica a los diferentes elementos mencionados:

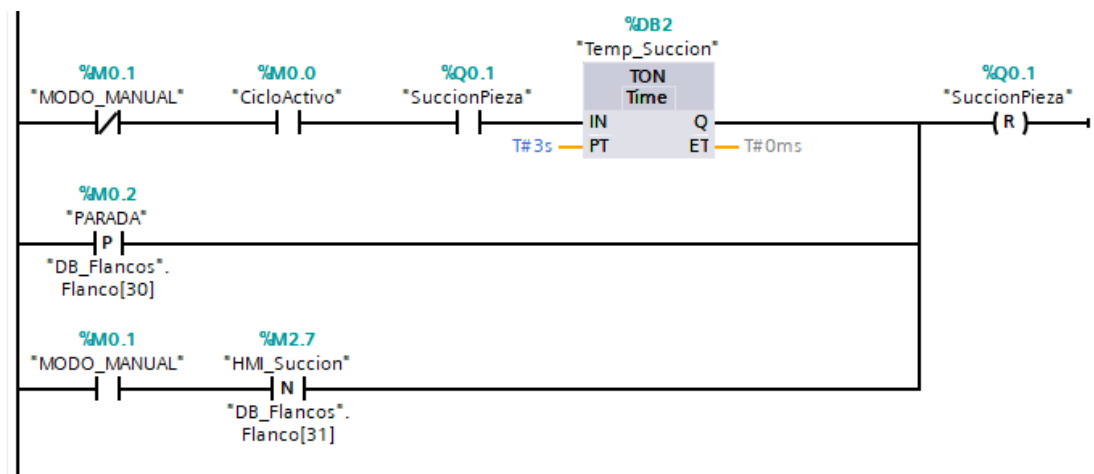


Figura 9. Fragmento de código en KOP

Este segmento posee tres ramas en paralelo conectadas a una bobina con *Reset*, esto significa que cuando cualquiera de las tres posibles ramas se active, el motor *Q0.1* se reseteará.

La **primera rama** incluye un cuadro para programar un temporizador. Este temporizador se activará cuando se encuentren activos los tres contactores previos, uno de ellos está normalmente cerrado (*M0.1*) y los otros dos son contactores normalmente abiertos (*M0.0* y *Q0.1*). Cuando los tres contactores se encuentren activos, el temporizador comenzará la cuenta y cuando llegue al valor establecido (3 segundos) activará su salida, activando así la bobina.

La **segunda rama** incluye un flanco positivo, cuando la variable “*PARADA*” (*M0.2*) se active, en el flanco de subida de la misma se activará la bobina, incluso si la variable deja de valer ‘1’ después de ocurrir el flanco.

La **tercera y última rama** cuenta con un contactor normalmente abierto (*M0.1*) y un flanco de bajada (*M2.7*), cuando el contactor esté activo y la variable asociada al flanco de bajada se desactive se activará la salida, incluso si la variable del flanco vuelve a activarse.

- **Lenguaje AWL**

El lenguaje *AWL* (Anweisungsliste) [5] suele definirse como un lenguaje basado en listas de instrucciones, muy parecido a la programación en Ensamblador.

La lógica de *AWL* al igual que en *KOP* está dividida en pequeños segmentos que son ejecutados en orden descendente. Estos segmentos se componen de instrucciones que ejecutan operaciones.

Las operaciones en *AWL* se dividen en Operación y Operando, como se puede observar en la **Figura 10**:

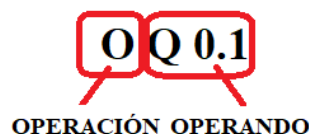


Figura 10. Estructura de una instrucción

La operación es la instrucción que se va a ejecutar, por ejemplo, una instrucción *AND* realiza la operación lógica entre dos variables. El operando indica una constante o dirección del autómata sobre la que ejecuta la instrucción.

Todas las *CPU S7* disponen de dos registros de 32 bits llamados Acumuladores, los acumuladores procesan y almacenan los operandos y resultados de las operaciones lógicas de bytes o palabras. Cada uno de estos acumuladores (*ACU1* y *ACU2*) se compone de dos palabras de 16 bits, una palabra baja y una palabra alta, esto permite operaciones como cargar, transferir e intercambiar el contenido de acumuladores y realizar operaciones entre ellos.

En el caso de bits, el resultado lógico de una operación como el de la *Figura 10*, se almacena en un bit llamado *RLO* (Result Logic Operation). Normalmente el *RLO* se encuentra en estado de primera consulta, la primera consulta ocurre cuando se efectúa una asignación o cuando empieza un ciclo de nuevo.

El *RLO* pertenece al bit 1 de la palabra de estados. La palabra de estados es un registro donde se almacenan los bits de procesamiento y estado de las operaciones que se ejecutan. Esta palabra está compuesta por 9 bits, cada bit dentro de esta palabra cumple una función específica para mantener una trazabilidad de lo que se está ejecutando. En la *Figura 11* se muestra la estructura de la palabra de estados.

| RB | A1 | A0 | OV | OS | OR | STA | RLO /ER |
|----|----|----|----|----|----|-----|---------|
| - | - | - | - | - | X | X | X 1 |

Figura 11. Estructura de la palabra de estados

BIT 0 - ER. Cuando este bit está a '0' indica que la siguiente línea de código se ejecuta como nueva consulta. En este estado la consulta se almacena directamente en el *RLO*.

BIT 1 – RLO. Este bit almacena el valor del resultado lógico de las operaciones actuales a nivel de bit (*AND, OR, XOR...*) y primera consulta.

BIT 2 – STA. Es el bit de estado de la dirección booleana de la memoria. Si está a '1' es que la marca o entrada está a '1', si está a '0', pues al revés.

BIT 3 – OR. Sirve para conocer el valor de la combinatoria de un *AND* antes de hacer un *OR*.

BIT 4 – OS. Es el bit de desbordamiento memorizado. Se activa junto con *OV* e indica que previamente se ha producido un error.

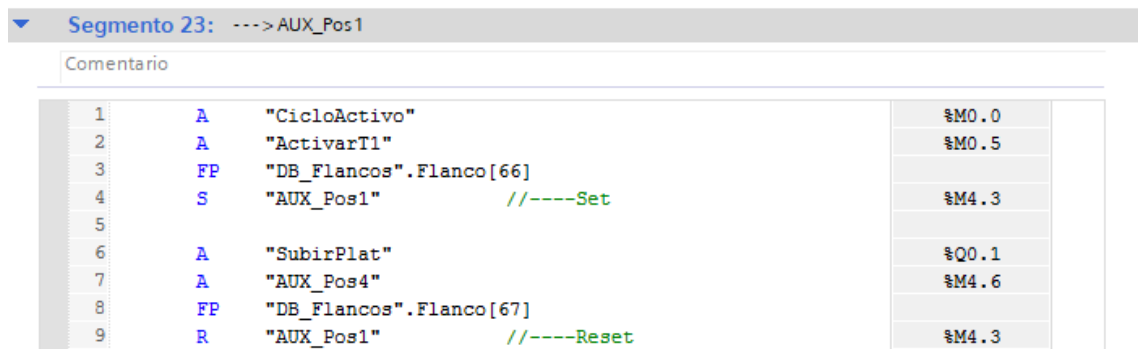
BIT 5 – OV. Es el bit de desbordamiento. Se pone a ‘1’ cuando una operación aritmética da un error por desbordamiento, operación no admisible o relación incorrecta.

BITS 6 (A0) Y 7 (A1). Son los códigos de condición. Dan información sobre los resultados o bits de operaciones aritméticas, comparaciones, operaciones digitales y bits desplazados por una instrucción de desplazamiento o rotación.

BIT 8 – RB. Es el bit de resultado binario. Permite interpretar el resultado de una operación de palabras como resultado binario e integrarlo en la cadena de combinaciones lógicas binarias.

A continuación, se muestra mediante dos ejemplos el funcionamiento básico de este lenguaje:

En la **Figura 12** se muestra un ejemplo de un segmento donde se utilizan dos condiciones de salida para la variable “*Aux_Pos1*” (*M4.3*).



| Segmento 23: --->AUX_Pos1 | | | | |
|---------------------------|----|-------------------------|--|-------|
| Comentario | | | | |
| 1 | A | "CicloActivo" | | %M0.0 |
| 2 | A | "ActivarT1" | | %M0.5 |
| 3 | FP | "DB_Flancos".Flanco[66] | | |
| 4 | S | "AUX_Pos1" //----Set | | %M4.3 |
| 5 | | | | |
| 6 | A | "SubirPlat" | | %Q0.1 |
| 7 | A | "AUX_Pos4" | | %M4.6 |
| 8 | FP | "DB_Flancos".Flanco[67] | | |
| 9 | R | "AUX_Pos1" //----Reset | | %M4.3 |

Figura 12. Fragmento de código en AWL

La primera condición son dos puertas *AND*, una instrucción tiene un flanco, es decir que mientras el valor de “*ActivarT1*” sea ‘0’ la operación lógica será [*I AND 0*] dando resultado ‘0’ siendo el valor del *RLO* también ‘0’ y no activando la salida. En el momento que se detecta un flanco positivo de la variable “*ActivarT1*” la operación lógica será [*I AND I*] dando ‘1’ el resultado, el *RLO* valdrá uno y la variable “*AUX_Pos1*” se activará con la instrucción *SET*.

La segunda condición es similar a la anterior, cuando se detecte un flanco positivo en la variable “AUX_Pos4” la operación AND dará ‘1’, el RLO valdrá ‘1’ y se reseteará la variable “AUX_Pos1”.

En la **Figura 13** se muestra otro ejemplo, en este caso se emplean temporizadores. Los temporizadores en AWL deben definirse cargando en el acumulador el valor de la cuenta a realizar y utilizar una instrucción diferente para cada tipo de temporizador.

▼ **Segmento 12:** Temporizador para expulsar pieza mala

Comentario

| | | | |
|----|----|-------------------------|--------|
| 1 | A | "CicloActivo" | %M0.0 |
| 2 | AN | "PiezaBuena" | %M0.4 |
| 3 | A | "BajarDefectuosa" | %M1.1 |
| 4 | A | "FinalCarr1" | %I0.3 |
| 5 | FP | "DB_Flancos".Flanco[39] | |
| 6 | S | "ActivarT4" //----Set | %M1.0 |
| 7 | | | |
| 8 | | | |
| 9 | A | "ActivarT4" | %M1.0 |
| 10 | L | s5t#2s | s5t#2s |
| 11 | SS | "Temp_ExpulsarMala" | %T4 |
| 12 | | | |
| 13 | O | "SubirPlat" | %Q0.1 |
| 14 | O | "PARADA" | %M0.2 |
| 15 | R | "Temp_ExpulsarMala" | %T4 |
| 16 | R | "ActivarT4" //----Reset | %M1.0 |

Figura 13. Fragmento de código en AWL

La variable “ActivarT4” se activará cuando la operación lógica ($M0.0 \text{ NAND } M0.4 \text{ AND } M1.1 \text{ AND } I0.3$) de resultado 1, el temporizador contará hasta 2 segundos (SS = TON) y cuando alcance el valor de consigna se activará la variable “Temp_ExpulsarMala” (%T4).

4.2.3 Modelos seleccionados

Se dispondrá de dos modelos diferentes de la marca SIEMENS, el modelo SIMATIC S7-1200 y el modelo SIMATIC S7-1500.

Revisando las características técnicas de ambos PLC’s [6], se realiza la siguiente tabla mostrando las diferencias entre ambos modelos:

Tabla 6. Comparación entre S7-1200 y S7-1500

| CARACTERÍSTICAS | S7-1200 | S7-1500 |
|---|---------------|-------------------|
| Tensión de alimentación | 24V DC | 24V DC |
| Consumo (valor nominal) | 0,85A | 0,85 ^a |
| Memoria de trabajo para programa | 75 Kbyte | 1 Mbyte |
| Memoria de trabajo para datos | 4 Mbyte | 5 Mbyte |
| Tiempo de ejecución de la CPU para operaciones a bits | 0,085 μ s | 10 ns |
| Tiempo de ejecución de la CPU para operaciones a palabras | 1,7 μ s | 12 ns |
| Nº de bloques (total) | 6000 | 6000 |
| Nº de interfaces Profinet | 1 | 2 |
| Permite lenguaje KOP | Sí | Sí |
| Permite lenguaje AWL | No | Sí |

La principal diferencia entre ambos modelos se encuentra en la *CPU*, pues el *S7-1500* tiene unos tiempos de ejecución menores.

Según la propia marca el modelo *S7-1500* está enfocado a una gama media, enfocado a aguantar una cierta exigencia de trabajo y requisitos técnicos, mientras que el *S7-1200* se considera de gama baja y está enfocado a una exigencia más pasiva y con poca demanda de trabajo.

A nivel de este proyecto la única diferencia que tendrán los *PLC's* es que los *S7-1200* están programados en *KOP* y los *S7-1500* en *AWL*, puesto que los *S7-1200* no soportan el lenguaje *AWL*.

En la **Figura 14** se muestra el aspecto de cada modelo, en la siguiente sección veremos los módulos de ampliación, que permitirán ampliar el número de entradas y salidas del *PLC*.



Figura 14. Modelos S7-1200 y S7-1500 respectivamente

4.2.4 Módulos de ampliación

Los módulos de ampliación son un hardware que se le añade a los *PLC* para permitirles ampliar el número de entradas y salidas y así evitar su reemplazo por otro modelo con más capacidad.

Los módulos suelen ser de tres tipos:

- Los **módulos de entradas**, que se distinguen en entradas digitales y analógicas. Los módulos de entradas digitales trabajan con señales de tensión que solo tienen dos estados lógicos ('0' y '1'). Mientras que los analógicos traducen una señal de tensión convirtiéndola en un número para que el *PLC* la pueda interpretar mediante el uso de un convertor (A/D).
- Los **módulos de salidas**, agregan un determinado número de salidas al *PLC*, estas salidas pueden ser de tipo relé, salidas a transistor, triac o analógicas. Internamente en el *PLC* se realiza una conversión digital-analógica (D/A), puesto que el autómatas solo trabaja con señales digitales.
- Los **módulos de comunicaciones** permiten añadir al *PLC* las opciones de protocolos disponibles, como Protocolo *Modbus RTU*, *PROFIBUS* y *PROFINET*.

Todos los módulos deberán ser compatibles con el *PLC* al que se acoplan, para ello es recomendable consultar los correspondientes *datasheets* para cerciorarse de su compatibilidad.

En las siguientes tablas se muestran los modelos de módulos de los que se dispondrán:

Tabla 7. Selección de módulos para el S7-1500

| S7-1500 | |
|--------------------------------------|-------------------------|
| MODELO | CPU 1516-3 PN/DP |
| REFERENCIA | 6ES7 516-3AN01-0AB0 |
| VERSIÓN | V2.6 |
| MÓDULO DE ENTRADAS DIGITALES | |
| MODELO | DI 32x24 VDC HF |
| REFERENCIA | 6ES7 521-1BL00-0AB0 |
| VERSIÓN | V2.2 |
| MÓDULO DE SALIDAS DIGITALES | |
| MODELO | DQ 32x24 VDC / 0.5 A HF |
| REFERENCIA | 6ES7 522-1BL01-0AB0 |
| VERSIÓN | V1.1 |
| MÓDULO DE ENTRADAS ANALÓGICAS | |
| MODELO | AI 8xU/ I/ RTD/ TC ST |
| REFERENCIA | 6ES7 531-7KF00-0AB0 |
| VERSIÓN | V2.2 |
| MÓDULO DE SALIDAS ANALÓGICAS | |
| MODELO | AQ 4xU/ I ST_1 |
| REFERENCIA | 6ES7 532-5HD00-0AB0 |
| VERSIÓN | V2.2 |

Tabla 8. Selección de módulos para el S7-1200

| S7-1200 | |
|--------------------------------|------------------------|
| MODELO | CPU 1212C AC/DC/Rly |
| REFERENCIA | 6ES7 212-1BE40-0XB0 |
| VERSIÓN | V4.2 |
| MÓDULO DE E/S DIGITALES | |
| MODELO | SM 1223 DI8/DQ8 x relé |
| REFERENCIA | 6ES7 223-1PH32-0XB0 |
| VERSIÓN | V2.0 |

4.3 COMUNICACIONES

4.3.1 Protocolo PROFINET

Para que todos los dispositivos del *SCADA* funcionen correctamente será necesario que se comuniquen entre ellos, para ello se utilizan los protocolos de comunicación industrial, que permiten el intercambio de información entre los diferentes elementos que conforman una subred industrial.

Los protocolos de comunicación industrial permiten el intercambio de información entre los diferentes elementos que conforman una subred industrial.

El protocolo utilizado en este proyecto es *PROFINET* [7], un estándar abierto de Ethernet Industrial de la asociación *PROFIBUS Internacional* según *IEC 61784-2*.

PROFINET destaca por utilizar Ethernet en tiempo real, donde los dispositivos se comunican por el bus de campo y cooperan en el procesamiento de solicitudes que se realizan dentro del propio bus.

4.3.2 Detalles de las comunicaciones

El mecanismo de intercambio de datos utilizado es el proveedor-consumidor, propio del protocolo *PROFINET*, este mecanismo es muy flexible, pues permite que todos los dispositivos de la red asuman el rol de proveedor y consumidor indistintamente [8].

A este mecanismo le hemos asignado una jerarquía adicional, el *PLC S7-1500* de la estación 1 actúa a un nivel superior del resto, este *PLC* se encarga de iniciar y parar el proceso de todas las estaciones restantes tomando un rol similar al de *Maestro-Esclavo*, propio del protocolo *PROFIBUS*, donde se establece una jerarquía de comunicación que es solo de una dirección.

El *PLC* de mayor jerarquía proporcionará los datos de entrada al resto de consumidores que serán las otras estaciones. Sin embargo, las otras estaciones también están configuradas para su comunicación a un nivel inferior, que permitirá el funcionamiento como cadena de producción.

4.3.3 Bloques GET y PUT

Para que las *CPU's* puedan transferir datos, deberemos utilizar en el *TIA Portal* los bloques *GET* y *PUT*, propios de las comunicaciones tipo *S7*.

Las *comunicaciones S7* son enlaces entre las *CPU's* a través de conexiones *PROFINET* y *PROFIBUS*.

En primer lugar, deberemos activar la pestaña de “*Permitir acceso vía comunicación PUT/GET*” en las opciones de protección local de la *CPU* en la pestaña de propiedades del dispositivo. Esta opción en los *PLC's* de versiones superiores de firmware a la 4.0 por defecto se encuentra desactivada.

Con el surgimiento de nuevos procedimientos Ethernet las comunicaciones *PUT/GET* se han quedado obsoletas. El principal problema que presentan es que son débiles frente a amenazas cibernéticas, con procedimientos externos es posible interrumpir el tránsito de datos en caso de las comunicaciones estén conectadas a la red.

En este proyecto la red es local, por tanto, es posible usar los bloques *PUT/GET* sin ningún problema.

El bloque *GET* permite leer datos desde una *CPU* interlocutora, mientras que el bloque *PUT* permite escribir datos en una *CPU* interlocutora. Estos bloques son tratados por el *TIA Portal* como bloques de datos.

- **Bloque GET**

El bloque *GET* permite leer datos desde una *CPU* interlocutora. De esta manera, la *CPU* que utiliza el bloque obtiene datos de una *CPU* remota.

Dentro de este bloque encontramos cuatro elementos a configurar:

REQ: Habilita la transferencia de datos en el flanco de subida de la variable asociada. Este parámetro cuando está activo establece la conexión con la *CPU* interlocutora para recoger los datos.

ID: El identificador hace referencia a la conexión establecida entre dos *PLC's*. Esta conexión viene indicada mediante un código que simboliza la unión establecida.

Normalmente es un formato de tipo Word. Este *ID* de conexión debe ser único para el enlace entre dos *PLC*'s.

ADDR: Es un puntero que necesita la dirección absoluta de la *CPU* remota para recoger los datos. En la dirección asignada los datos serán leídos cuando se active el *REQ*.

RD: Es un puntero que necesita la dirección absoluta de un espacio de memoria de la propia *CPU* local. En este parámetro se guarda el valor del dato leído de la dirección indicada en el *ADDR*.

Tanto al bloque *GET* como al bloque *PUT* se le pueden añadir hasta cuatro direcciones de envío y lectura. En caso de necesitar más direcciones es conveniente utilizar varios bloques de datos.

En la **Figura 15** se muestra un ejemplo de la estructura de un bloque *GET*:

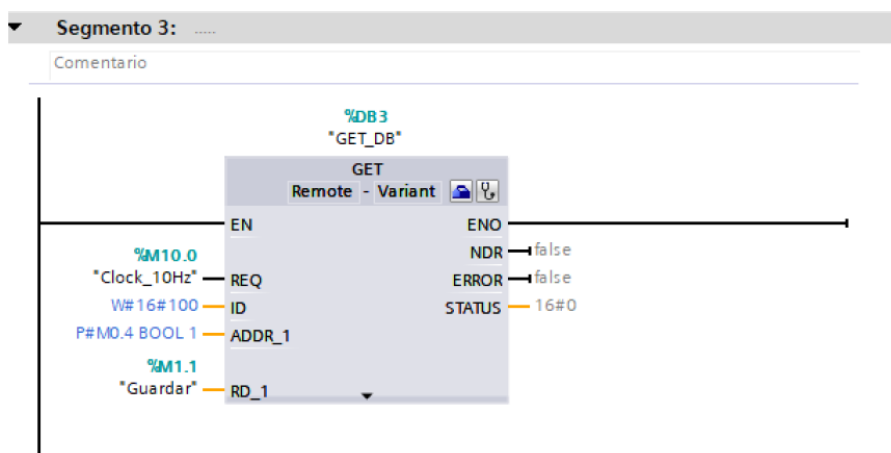


Figura 15. Estructura de un bloque *GET*

En este ejemplo la conexión se establece de forma periódica, ayudado por la señal de reloj a la que se encuentra conectada el *REQ*, esto permite obtener de forma continuada los datos de la *CPU* remota.

Los datos que recogeremos son los de la dirección *M0.4*, estos datos serán de tipo booleano y de tamaño 1. El valor de estos datos se actualizará cada vez que el *REQ* se active.

Los datos los guardamos en la variable “*Guardar*” (*M1.1*). Es importante mencionar que realmente lo que hacemos es guardar el valor que leemos en la *CPU* remota en nuestra *CPU*, por esto es importante que el *ADDR* y *RD* tengan el mismo tipo y longitud de datos para que no exista una pérdida de información.

- **Bloque PUT**

El bloque *PUT* permite escribir datos del *PLC* local en un *PLC* remoto.

Dentro de este bloque encontramos cuatro elementos a configurar:

REQ: Habilita la transferencia de datos en el flanco de subida de la variable asociada. Este parámetro cuando está activo establece la conexión con la *CPU* interlocutora y le envía los datos. El valor de este parámetro deberá ser del mismo tipo y longitud que el *ADDR* para que no se produzca una pérdida de información durante el envío.

ID: El identificador hace referencia a la conexión establecida entre dos *PLC*'s. Esta conexión viene indicada mediante un código que simboliza la unión establecida. Normalmente es un formato de tipo Word. Este *ID* de conexión debe ser único para el enlace entre dos *PLC*'s.

ADDR: Es un puntero que necesita la dirección absoluta de la *CPU* remota para escribir los datos. Los datos son enviados a la dirección remota indica en el *ADDR*.

SD: Es un puntero que necesita la dirección absoluta de un espacio de memoria de la propia *CPU* local. El envío de datos se efectúa mandando el valor de este parámetro cuando se inicia el *REQ*. El valor de este parámetro deberá ser del mismo tipo y longitud que el *ADDR* para que no se produzca una pérdida de información durante el envío.

En la **Figura 16** se muestra un ejemplo de la estructura de un bloque *PUT*:

El identificador tanto para el bloque *PUT* y *GET* se mantiene, pues ambos bloques son del mismo *PLC* y de la misma conexión.

Al igual que antes, el *REQ* establece la conexión en forma de señal de reloj y los datos son enviados a la dirección absoluta que se indica en el *ADDR*. Los enviados serán los pertenecientes a los datos de *SD*.

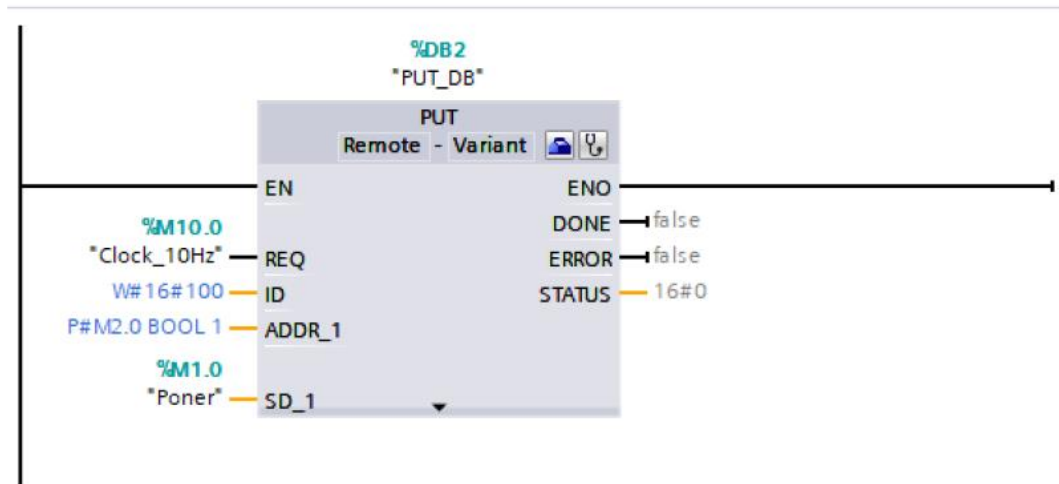


Figura 16. Estructura de un bloque PUT

4.4 HMI

Una pantalla *HMI* (Human-Machine Interface) es una interfaz que conecta un proceso industrial con el operario. Los *HMI* permiten monitorizar y controlar el proceso.

Normalmente los *HMI* son encontrados en forma de “*Imágenes*”, estas imágenes se cargan en el panel táctil y permiten al operario manipular y observar el proceso de forma precisa.

Las imágenes permiten concentrar los diferentes indicadores de la estación, como pueden ser sensores y actuadores. De otra manera, los indicadores estarían dispersos por diferentes zonas de la planta industrial.

Como norma general los *HMI* están conectados a *PLC*'s, esto permite trabajar de forma eficiente con el nivel de actuadores/sensores. De otro modo, si los *HMI* estuvieran directamente conectados a los ordenadores se tendría unos tiempos de espera elevados.

Los *HMI* son programados en software especializados, cada tipo de panel *HMI* utiliza su propio programa de elaboración de pantallas. Los softwares permiten al programador elegir qué es lo que se puede monitorizar en la pantalla y qué es lo que puede controlar.

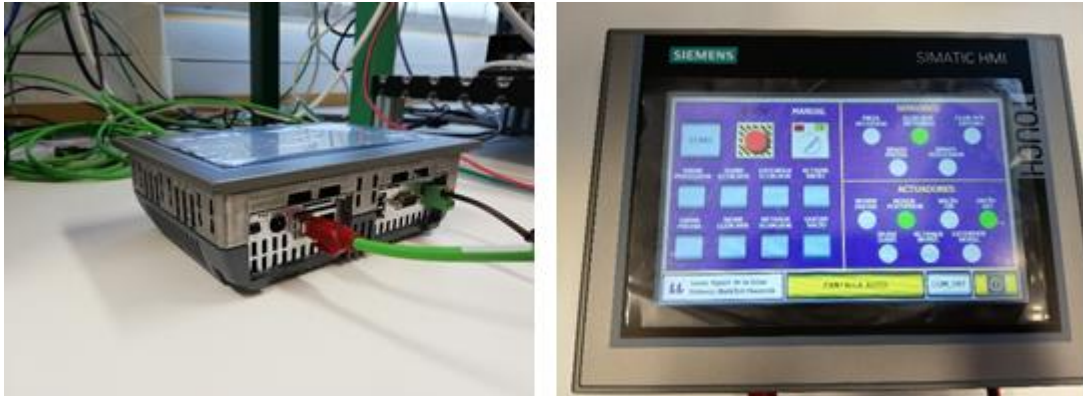


Figura 17. Panel HMI utilizado

En la siguiente sección se muestran las características del modelo de panel utilizado.

4.4.1 Modelo de HMI utilizado

El modelo *HMI* utilizado en este proyecto es el *SIMATIC HMI TP700 Comfort Panel*. [9]

La referencia del producto es: 6ES7 124-0GC01-0AX0 con versión 16.0.0.0 actualizada.

Sus características generales son:

- Tamaño de pantalla de 7 pulgadas y completamente táctil.
- Anchura del display de 152,4 x 91,4 mm.
- La resolución es de 800 x 400 píxeles.
- Este modelo es de 7 pulgadas, con una anchura de display de 152,4 x 91,4 mm.
- La resolución de la imagen es de 800 x 400 píxeles.
- Su tensión de alimentación es de 24 voltios en corriente continua.
- Tiene un consumo de corriente de valor nominal de 0,5 A.

- El consumo de potencia activa es de 12 W.

Si atendemos a características más específicas:

- Dispone de un procesador X86, con memoria RAM y Flash
- Memoria usable para datos de usuario de 12 Mbytes.
- Soporta protocolos de conexión Profinet, Profibus, MPI e IRT.
- Dispone de una capacidad de más de 500 imágenes configurables con más de 400 objetos como máximo por imagen. Dentro de estos objetos engloba campos de texto, campos de E/S gráficos y simbólicos, campos de fecha/hora, interruptores, botones, visor de gráficos, objetos geométricos e iconos.
- Compatible con los modelos S7-1200, S7-1500, S7-200 y S7.300/400.

Para programar las pantallas HMI, utilizamos un software que pertenece al TIA Portal llamado WinCC. En la siguiente sección lo veremos con detalle.

4.4.2 WinCC

“*SIMATIC WinCC* es un sistema de visualización de procesos escalable y dotado de potentes funciones para la supervisión de procesos automatizados. *WinCC* aporta funcionalidad *SCADA* completa en *Windows* para todos los sectores.

El *WinCC* está abierto tanto a aplicaciones monopuesto propias del sector de fabricación de maquinaria, como a soluciones multipuesto más complejas o incluso a sistemas distribuidos con varios clientes y servidores redundantes para la ingeniería de plantas e instalaciones. Las numerosas referencias procedentes de sectores como la industria del automóvil, industria química y farmacéutica, industria alimentaria, suministro y distribución de energía entre muchos otros sectores, testifican su capacidad de uso múltiple y rendimiento eficaz”. [10]

El software *WinCC* permite programar la pantalla *HMI* desde el entorno *TIA Portal*. Cuando el programa es transferido al panel *HMI*, se ejecuta una rutina llamada “*Runtime*”

que activa el funcionamiento de la pantalla para que tenga un carácter dinámico, de lo contrario, la imagen estaría estática y la programación no tendría sentido.

Dentro del *WinCC* encontramos diferentes categorías, cada versión permite ir aumentando el control y la disponibilidad de las herramientas:

- El ***WinCC Basic***, es el más básico de todos, con el *Basic* solo estará permitido trabajar a nivel de configuración y programación de paneles básicos *HMI*.
- El ***WinCC Comfort***, permite trabajar con paneles *Comfort*, que ofrecen algo más complejo y además no solo son *HMI*, sino que también se puede programar paneles móviles y multipaneles.
- El ***WinCC Advanced***, permite programar ya en un nivel superior, permite trabajar con *PLC's* y sistemas monopuestos basados *PC's* directamente.
- El ***WinCC Professional***, permite definir un orden central que globaliza la planta, definir sistemas multipuestos basados en *PC's* y permite la funcionalidad de los *SCADA's*.

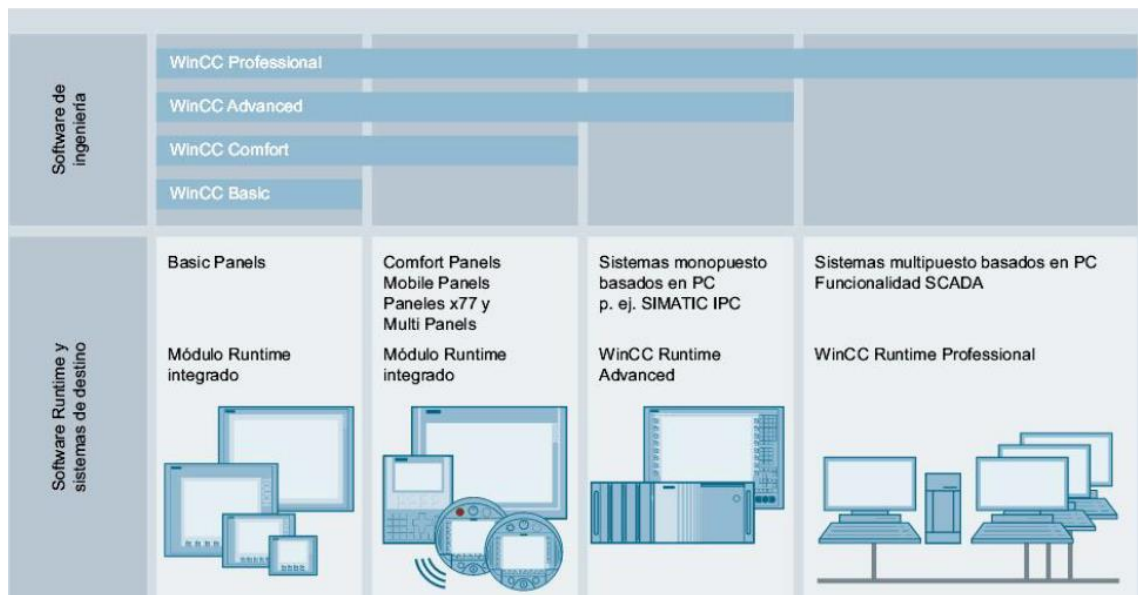


Figura 18. Diagrama de tipos de WinCC

4.4.3 Creación de la pantalla HMI

4.4.3.1 Configuración de la pantalla

Para la creación de una nueva pantalla *HMI* en el *TIA Portal*, iremos al menú de añadir dispositivos y elegimos entre el catálogo disponible de la librería de *TIA Portal* el modelo que tenemos.

Una vez escogido el modelo, se abrirá un menú para configurar aquellos aspectos básicos de la pantalla.

Inicialmente, tendremos que indicarle a la pantalla *HMI* a cuál *PLC* irá conectado, esto permitirá que el *HMI* tenga acceso a las variables de sistema del *PLC* que elijamos.

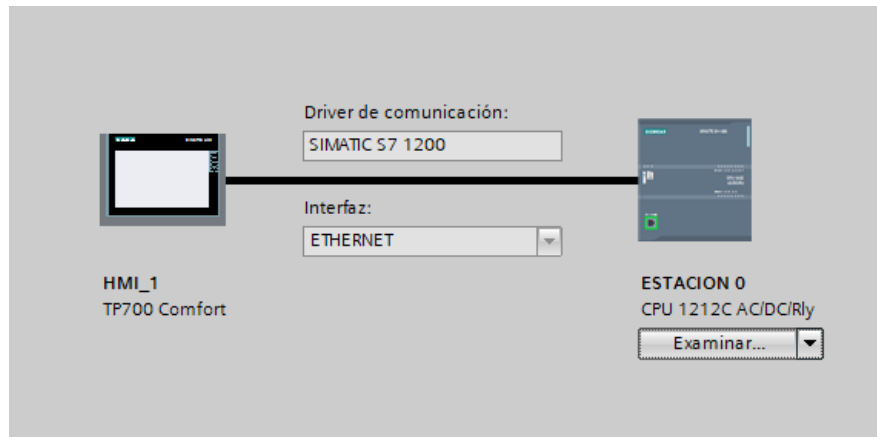


Figura 19. Conexión entre HMI y PLC

Posteriormente, tendremos que indicar que elementos queremos que aparezcan de forma predeterminada en la pantalla *HMI*, en este caso, no seleccionaremos ninguna opción y lo haremos todo desde cero.

Terminada la configuración de la pantalla, se abrirá una nueva pestaña en el proyecto del *TIA Portal* que será llamada “Imagen raíz” (ver *Figura 20*)

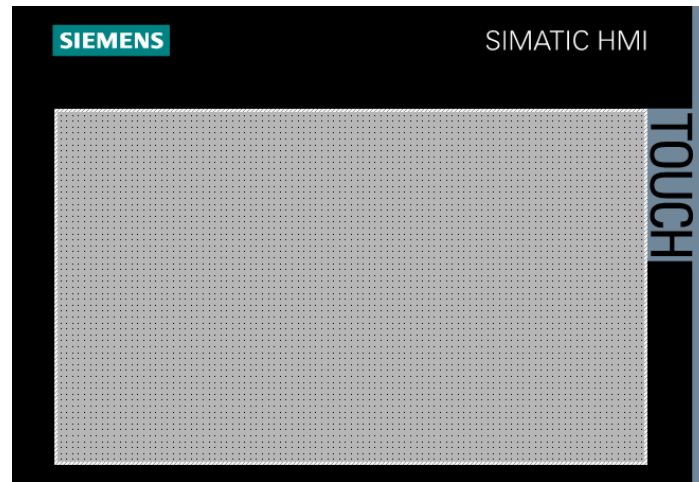


Figura 20. Imagen raíz inicial

4.4.3.2 Imagen interactiva

La imagen raíz es una imagen interactiva con la que se inicia la programación de la pantalla *HMI*. La imagen interactiva es aquella imagen que veremos cuando tengamos cargado nuestro proyecto del *TIA Portal* en el panel táctil.

Una pantalla *HMI* la forman una o varias imágenes interactivas diferentes. Las imágenes interactivas son de carácter dinámico, cambian en función de los sucesos que ocurran en el proceso, expresándolo de otra manera, cambian en función de la información que el *PLC* emparejado esté procesando.

El objetivo principal de las imágenes interactivas es permitir al operario poder observar y controlar el proceso. Para ello deben tener un carácter minimalista y de fácil comprensión. La información debe aparecer de forma clara y concisa, evitando la aglomeración visual de muchos acontecimientos al mismo tiempo o una excesiva cantidad de información irrelevante que pueda confundir al operario.

4.4.3.3 Objetos de la pantalla

Los objetos son pequeños “sprites” que se agregan a la imagen de la pantalla para otorgar un aspecto visual a la programación y permiten tener una funcionalidad a nivel de programación. Estos objetos son considerados *Powertags* según *WinCC*, los *Powertags* permiten establecer un enlace directo entre los objetos del *HMI* y las variables del controlador *PLC*.

Los sprites son un conjunto de mapa de bits que representan un objeto (o parte de ellos) de manera gráfica y que se utiliza para poder crear cualquier efecto de movimiento o para cambiar su estado.

Dentro de estos objetos se engloba campos de texto, campos de E/S gráficos y simbólicos, campos de fecha/hora, interruptores, botones, visor de gráficos, objetos geométricos e iconos. Existen dos tipos principales de objetos, los objetos básicos y los objetos específicos.

Los *objetos básicos*, que comprenden líneas, texto y figuras geométricas básicas o personalizadas. Estos objetos tienen una vista totalmente plana.

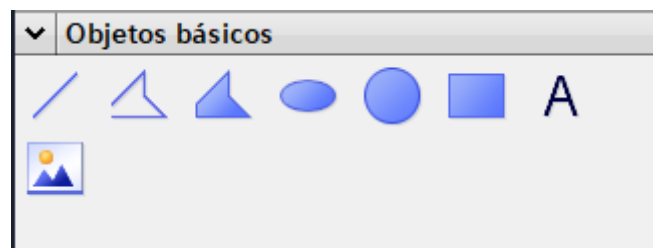


Figura 21. Objetos básicos

Los *objetos específicos*, son campos de entradas/salidas gráficos y simbólicos, dentro de estos objetos se encuentran los campos de fecha y hora, interruptores, botones, visor de gráficos, objetos geométricos e iconos.

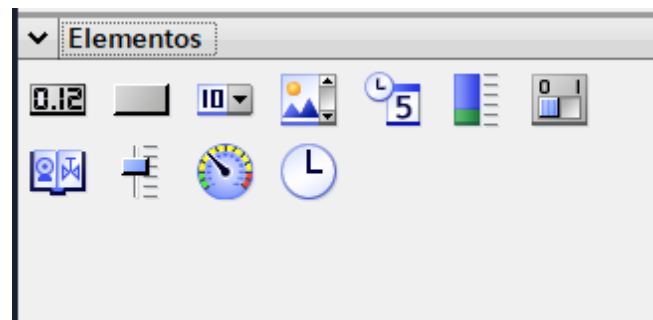


Figura 22. Objetos específicos

Dentro de los objetos específicos también encontramos los *iconos*, los se pueden definir como objetos especiales, estos objetos pertenecen a una librería específica que ha creado la marca *SIEMENS* destinada a la programación de pantallas *HMI* para procesos industriales. A diferencia de los objetos básicos, tienen una apariencia con relieve y efecto

3D. Estos objetos son mucho más complejos que los básicos, comprenden bombas de aguas, grúas, ventiladores, cables, tuberías, señales, etcétera.

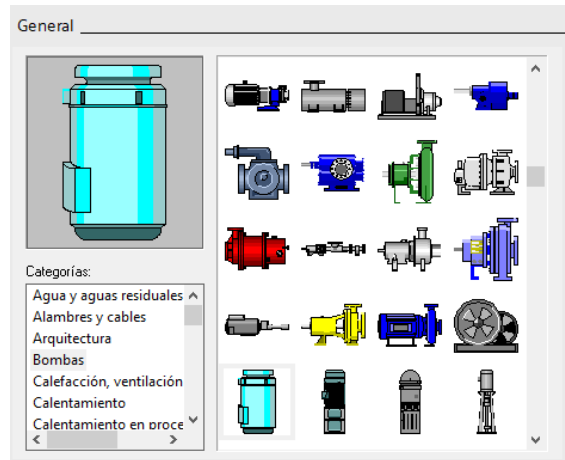


Figura 23. Iconos de la librería

A pesar de que la librería es muy amplia, la mayoría de objetos que necesitamos para el proyecto no los encontraremos en ella. Por este motivo, se crearon los objetos externamente y se importaron a la librería de objetos personalizados del TIA Portal para poder utilizarlos en las imágenes.

4.4.3.4 Método de creación de objetos personalizados

La creación de objetos personalizados consta de 8 pasos: [11]

1. Abrir un procesador de texto con disponibilidad de creación de figuras, como pueden ser *Word* o *Powerpoint*.
2. Crear el objeto que nos interesa, cuanto más pequeño sea, más claro se verá dentro del *TIA Portal*.
3. Guardar el objeto creado en formato de **Windows metafísico mejorado**.
4. Dentro del *TIA Portal* abrir “*Lista de textos y gráficos*”.
5. Definir un gráfico para el objeto creado, arrastrando desde la carpeta donde esté guardado el objeto a la ventana de añadir gráfico.
6. En las herramientas de la pantalla *HMI*, seleccionar “*Campo E/S Gráfico*”.
7. Asignar una variable al objeto y seleccionar en la lista de gráficos el objeto que importamos.
8. En los ajustes de objeto seleccionar *Fondo Transparente* y *Estilo de borde Compacto*.

Los objetos creados mantienen las mismas propiedades de los iconos de la librería, sin embargo, estos objetos siempre tendrán que estar emparejados a una lista de gráficos en concreto. Una vez se tienen todos los objetos necesarios hay que programarlos, esto se verá en la siguiente sección.

4.4.3.5 Programación de los objetos

Como se ha comentado anteriormente, los objetos están emparejados con una variable en concreto del *PLC*, esta variable tiene un comportamiento recíproco con el objeto, es decir, si al objeto se le designa una acción y se activa, afectará a la variable, y al contrario, si la variable modifica su valor, afectará al objeto. Estas acciones varían en función de las preferencias de comportamiento deseadas, los objetos pueden cambiar su aspecto en función del valor que tome la variable, y la variable puede cambiar su valor en función de la acción que tome el objeto.

- *Objeto afecta a la variable*

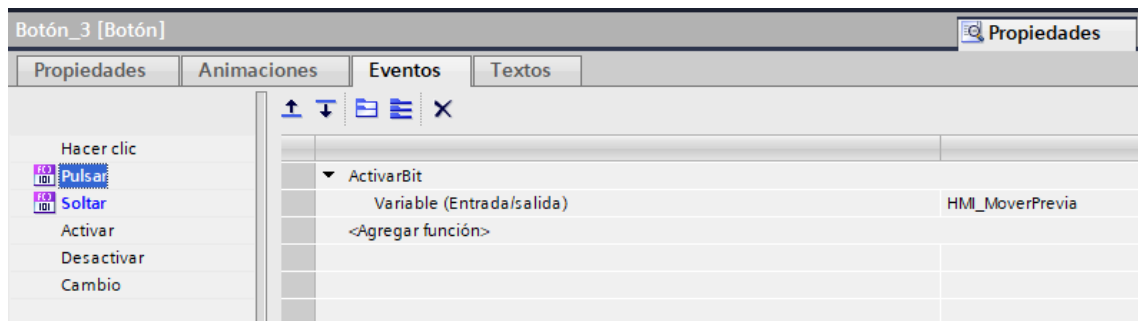


Figura 24. Vista de la ventana de Eventos

Para que un objeto tenga efecto sobre una variable, debemos asignarle lo que se conoce como “*Evento*”, un evento es un suceso relacionado con la pantalla HMI capaz de alterar una variable del *PLC*. A este tipo de objetos se les ha asignado el nombre de “*Elementos de Control*”.

Los *Elementos de Control* están divididos en dos tipos:

1) *Pulsadores*

La variable asociada a un pulsador es activada cuando se detecta una pulsación en el panel táctil sobre el objeto de la imagen, esta variable permanece activa mientras se mantenga pulsado el objeto, una vez se deje de pulsar, volverá a desactivarse la variable.

Los pulsadores están conectados a variables del código donde hay flancos de subida o bajada, puesto que los flancos solo registran el cambio de valor lógico de las variables, no será necesario mantener pulsado el objeto de la pantalla.

La apariencia de estos pulsadores es de color gris cuando permanecen desactivados y obtienen brevemente el color verde en el instante en el que pulsan, volviendo al color gris un segundo más tarde.

2) *Conmutadores*

Permiten que la variable asociada se mantenga activa sin tener que mantener pulsado el objeto en el panel táctil. A diferencia de los pulsadores, para desactivar la variable se deberá volver a presionar sobre el objeto.

La apariencia de los conmutadores es similar a los pulsadores por motivos estéticos y de coherencia, la única diferencia es el color utilizado, por defecto son de color negro y cuando se activan obtienen el color verde.

- *Variable afecta al objeto*

El objetivo de que una variable afecte a un objeto tiene tres posibles funciones: La visibilidad, la apariencia y el movimiento.

➤ *Visibilidad de los objetos*

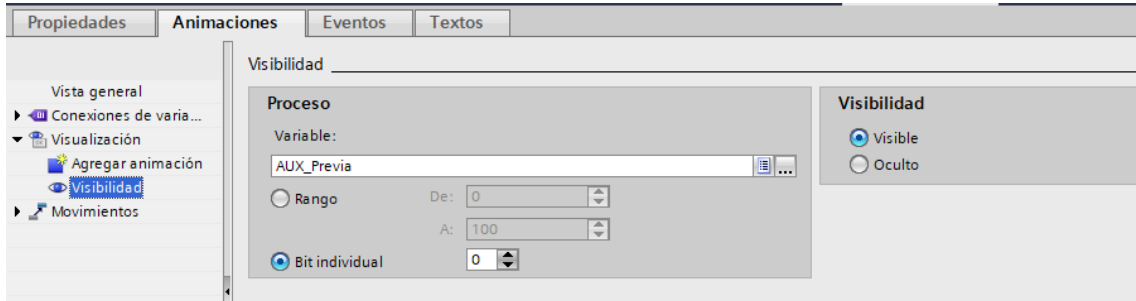


Figura 25. Vista de la ventana de Visibilidad

La visibilidad de objetos está enfocada a que los objetos aparezcan y desaparezcan de la imagen en función del valor actual de la variable emparejada. Esto permite dos usos diferentes:

- Si por defecto se elige la opción de que el objeto sea visible, se verá siempre y cuando la variable emparejada esté desactivada.
- Por el contrario, si por defecto se elige la opción de que el objeto permanezca oculto, hasta que la variable no esté activa no se verá el objeto.

Esta útil herramienta, permite que se vean en pantalla solo los elementos activos del proceso, quitando información innecesaria y evitando el solapamiento de objetos. Sin embargo, que solo se pueda asociar una variable a un objeto genera un problema, ¿Y si necesito que el objeto cambie su visibilidad en función de dos variables diferentes?

Desgraciadamente *TIA Portal* no ofrece esta opción. Por tanto, para solucionar esto nacen las variables (marcas del sistema) que a partir de ahora llamaremos con la nomenclatura “*AUX_XXXX*”.

Estas variables no tienen un significado real en el proceso, pero sí sirven para que en la pantalla HMI los objetos tengan el comportamiento deseado. Estas marcas siempre serán asignaciones que se activarán en función de la lógica programada. Asimismo, permiten solucionar que el mismo problema suceda cuando se quieran asignar varias variables a un mismo objeto en las dos funciones restantes.

Otro problema que solucionamos con estas marcas es el de los flancos, si la visibilidad de un objeto está determinada por una variable que se activa solo durante un flanco, la visibilidad del objeto será de un instante de tiempo, dado que la variable solo estará activa en el instante que se produzca dicho flanco. Utilizando estas marcas, aunque la variable deje de valer '1' la marca permanece activa.

➤ *Apariencia de los objetos*

La apariencia de los objetos consiste en el cambio del color del objeto en función del valor actual que tome la variable emparejada. Además, es posible programar que el objeto parpadee cambiando de un color a otro mientras se encuentre activado/desactivado.

Este cambio de color solo tiene sentido en objetos básicos, cuya forma es plana y sin relieve, o simplemente es texto.

A los objetos que no son básicos, les ocurre un efecto particular si se activa esta función. Al activarse la variable siempre cambiarán parte de su color al azul, iluminando los colores base de azul y dejando las sombras en negro. Si se activa el parpadeo alternará entre el color azul y el color base del objeto.

En este proyecto, se ha empleado esta función en objetos especiales en los que resulta difícil implementar cualquier tipo de movimiento, el parpadeo indicará que el objeto está moviéndose de alguna manera.

Los objetos personalizados no cumplen con estas propiedades, a los objetos personalizados no es posible cambiar su apariencia, dado que están formados a partir de un gráfico.

El cambio de apariencia de un objeto no necesariamente tiene que ser binario, como es el caso del contador del almacén de la estación 0 (ver *Figura 26*), este contador tiene un rango de 0 a 8 y en función del valor actual de la variable emparejada, irá tomando un color diferente, empezará con un color verde y a medida que el almacén tenga menos piezas irá cambiando a un color rojo.

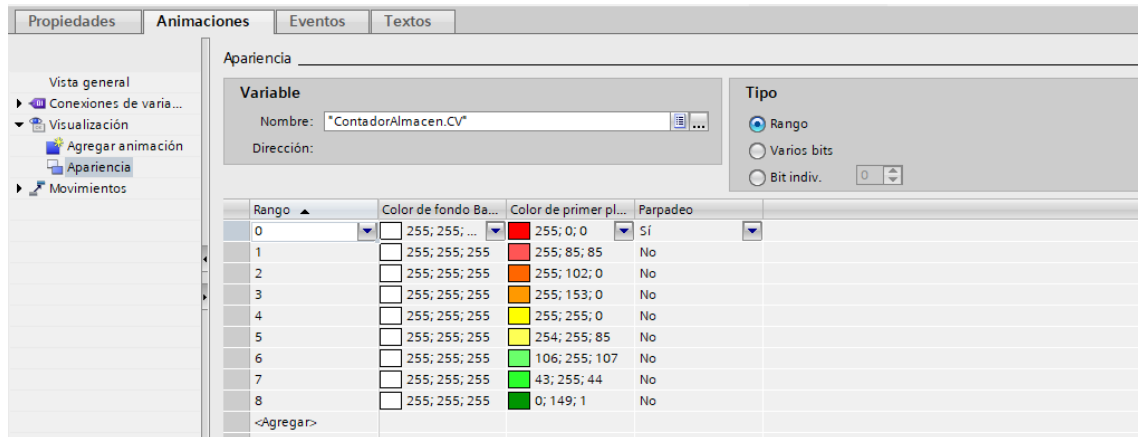


Figura 26. Vista de la ventana de Apariencia

➤ Movimiento de los objetos

El movimiento de los objetos consiste en establecer una trayectoria en línea recta por el espacio de la imagen. Marcando unas coordenadas XY iniciales y unas coordenadas XY finales, el objeto recorrerá esta trayectoria dividiendo el tramo en tantas partes como su rango de valores sea.

La trayectoria de los objetos puede ser de tres tipos diferentes: diagonal, vertical y horizontal.

Para conseguir ver el movimiento no sirve emplear una variable booleana, pues ésta recorrería la trayectoria en un solo instante de tiempo, siendo el valor 0 la posición XY inicial y el valor 1 la posición XY final.

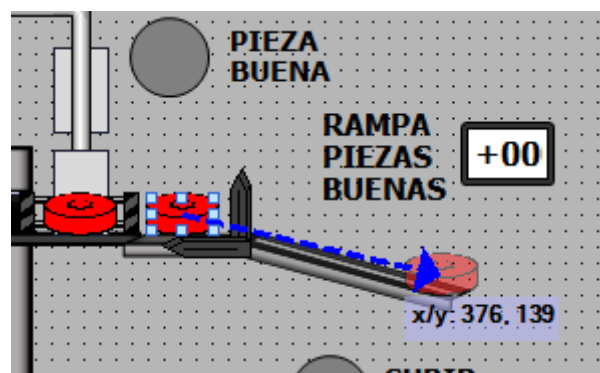


Figura 27. Vista de la trayectoria del movimiento

En el siguiente ejemplo que se muestra en la **Figura 28** se explica cómo lograr que un objeto se mueva:

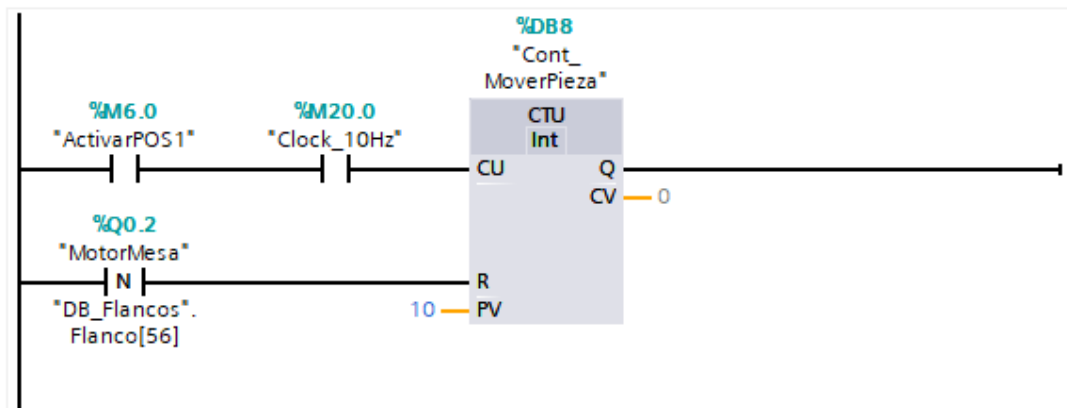


Figura 28. Fragmento de código con temporizador

En la estación 2, hemos utilizado un movimiento diagonal para conseguir que se vea como las piezas bajan por la rampa.

Para ello definimos tres elementos, una variable de activación del contador, un contador y una señal de reloj de 10Hz.

La variable que asociaremos al objeto será el valor actual del contador (CV). Es decir, la variable perteneciente al bloque de datos de instancia llamada “*Cont_MoverPieza.CV*”.

Si establecemos que el contador llegue a 10, la distancia que recorrerá el objeto será dividida en 8 partes. Una vez se quiera mover el objeto se activará la variable llamada “*ActivarPOS1*”.

Cuando la variable “*ActivarPOS1*” está a ‘1’, cada vez que la señal de reloj tome el valor ‘1’, incrementará en uno el contador, con lo cual el objeto se moverá una distancia X respecto a su posición. Como la señal de reloj es rápida, el valor del contador se incrementa a un ritmo que emula movimiento.

Cuanto mayor sea la distancia establecida y menor sea el valor del contador la pieza tendrá un movimiento poco fluido, dado que la pieza recorrerá la distancia en pocos puntos. Para lograr una relativa fluidez de movimiento será necesario comprobar mediante ensayo y error cuántos puntos son necesarios.

4.4.3.6 Elementos generales de las imágenes interactivas

Todas las imágenes interactivas disponen de “*elementos de control*” que ayudan al operario a realizar su labor. Distinguiremos dos tipos: los “*elementos de control internos*” que afectan al proceso de forma directa, y los “*elementos de control externos*” que no afectan al proceso, pero ofrecen diferentes acciones.

➤ *Elementos de control internos*

Los elementos de control internos permiten al operario iniciar, cambiar o parar el estado del proceso mediante una pulsación en el panel táctil.

Estos elementos de control se muestran en la **Figura 29**.



Figura 29. Elementos de control internos

1) *Pulsador de Start*

Este pulsador permite iniciar el proceso desde la pantalla.

La activación de *START* en el *PLC* maestro implica la activación del ciclo en todos los *PLC*'s esclavos. En el caso de estar operando con las comunicaciones deshabilitadas, la pulsación de *START* solo iniciará el ciclo de la estación correspondiente.

Es importante mencionar que este pulsador no es la única manera de iniciar el proceso.

2) *Seta de Emergencia*

La seta de emergencia detiene inmediatamente el proceso una vez ha sido activado, se trata de un conmutador, por tanto, la emergencia se mantendrá activa hasta que el operario desactive el conmutador.

Al igual que con el pulsador de *START*, con las comunicaciones habilitadas solo es necesario activar la parada de emergencia en el *PLC* maestro para detener todas las estaciones.

En el caso de estar operando con las comunicaciones desactivadas solo se detendrá la estación en la que se haya activado la parada de emergencia.

Cada estación tiene su propio procedimiento de actuación en caso de que la parada de emergencia se active. Todas deberán volver al estado inicial y esperarán a que se reanude el ciclo de nuevo.

3) *Conmutador del modo de operación*

Este conmutador otorga al operario la posibilidad de salir del ciclo convencional (modo automático) y entrar en el modo manual.

En el modo manual se le entrega el control absoluto del proceso al operario. Una vez este modo se activa, las comunicaciones entre *PLC's* se desactivan para proteger a la estación frente a posibles colisiones con las estaciones adyacentes.

Aunque el modo manual se encuentre activo, la parada de emergencia permanece operativa, el operario puede considerar necesario que sea activada por razones de seguridad. En caso de que se produzca su activación, el control manual es desactivado y el proceso vuelve a las condiciones iniciales.

➤ *Elementos de control externos*

Los elementos de control externos permiten al operario interactuar con la pantalla *HMI* sin que se traduzca en un cambio sobre el proceso. Cada uno de los tres elementos de control de los que se dispone cumplen un propósito:



Figura 30. Elementos de control externos

1) *Cambiar de imagen*

Pulsador que alterna entre las dos imágenes de las que dispone cada pantalla *HMI*. Esto permite al operario cambiar de la imagen de modo automático a la imagen de modo manual y viceversa cuando crea conveniente

Por defecto, al iniciar el *Runtime* de la pantalla *HMI* aparecerá la imagen de modo automático como imagen inicial.

2) *Quitar las comunicaciones*

Este conmutador activa y desactiva las comunicaciones de la estación correspondiente. Al estar activo, la estación no enviará ni recibirá ningún tipo de dato a las estaciones adyacentes.

Esta útil herramienta proporciona una manera de poder testear individualmente una estación en caso de fallo o de estudio de mejora.

Todas las estaciones están programadas para funcionar según estén activas o desactivas las comunicaciones. Como por defecto las comunicaciones están activas, para poder iniciar el ciclo sin ellas, primero se deberá activar el conmutador y después pulsar *START* para comenzar con el proceso.

3) *Salir (Exit Runtime)*

Pulsador para salir de las imágenes interactivas de la *pantalla HMI*, cuando se pulse se mostrará en el panel táctil el menú predeterminado del dispositivo. Una vez en este menú para volver a ver el proceso se deberá iniciar el *Runtime* pulsando en el panel.

4.4.3.7 Imagen interactiva manual y automática

Cada estación FESTO dispone de dos imágenes interactivas, una imagen para el control en modo manual y otra para el control en modo automático.

Por defecto, al iniciar el *Runtime* de la pantalla *HMI* aparece la imagen del modo automático, debido a que el proceso normalmente se encontrará en modo automático.

➤ **Imagen interactiva del modo automático**

La imagen interactiva del modo automático está enfocada a la visualización en tiempo real del proceso. Para lograr este objetivo, la imagen dispone de una amplia diversidad de objetos de todos los tipos, que se superponen en algunos casos y que cambian su comportamiento (visibilidad, apariencia y movimiento) en función del estado actual del proceso.

Además, se destina una parte de la imagen para que el operario disponga de un pequeño panel de control, donde dispone de los tres elementos de control internos y de una cantidad de pilotos reducida, que ayudan a la monitorización.

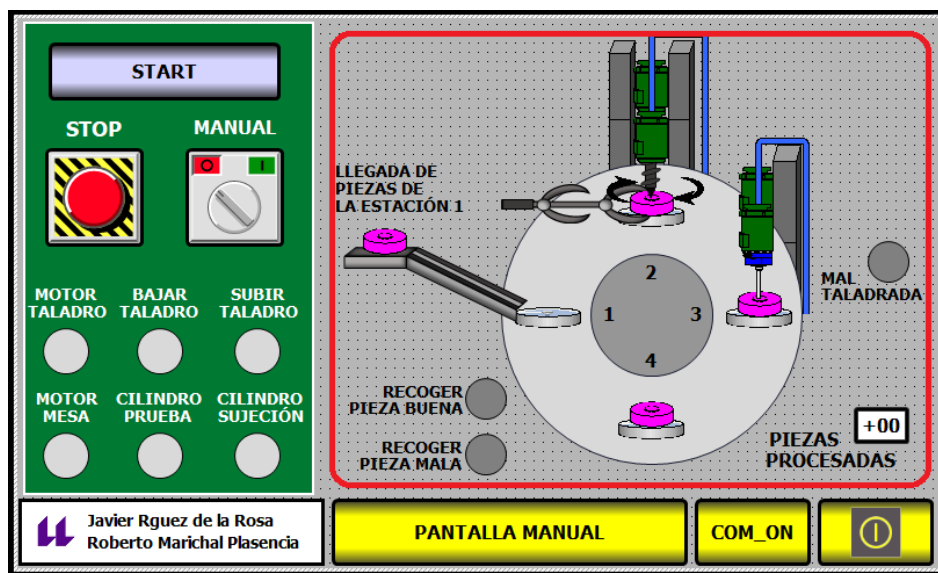


Figura 31. Visualización de los objetos programados

➤ **Imagen interactiva del modo manual**

La imagen interactiva del modo manual comprende el conjunto de panel del operador y el panel de visualización.



Figura 32. Panel de Control y Visualización

El panel de control se compone de los tres elementos de control internos, sumado a unos pulsadores y conmutadores que activan las “Órdenes de control”, el número de ordenes de control viene determinado por la cantidad de actuadores que tenga el proceso.

Las órdenes de control no son más que variables ligadas a los pulsadores y conmutadores que activarán o desactivarán los actuadores. Estas marcas llevan la nomenclatura de “HMI_XXXX”.



Figura 33. Pulsador y Conmutador respectivamente

Las órdenes que utilizan **pulsadores** solo podrán desactivarse cuando otra orden se ejecute y sea contraria a la misma.

En la **Figura 34** se muestra un ejemplo:

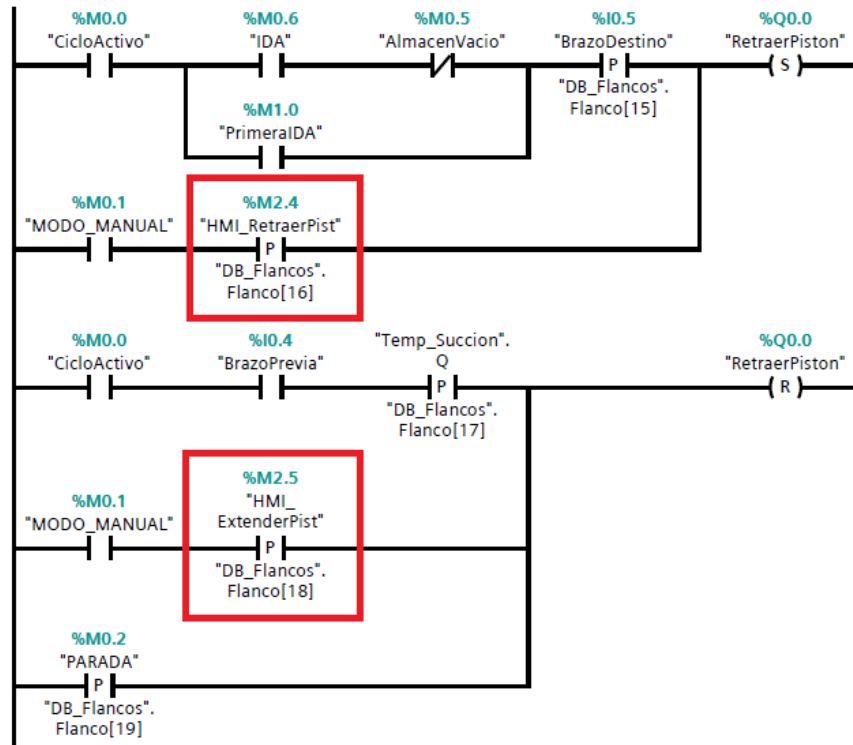


Figura 34. Fragmento de código de la Estación 0

Si activamos la orden “Retraer Pistón” (M2.4) con el pulsador de la pantalla HMI, se activa el motor de Retraer Pistón (Q0.0), este actuador se mantendrá activo hasta que se ejecute la orden de “Extender Pistón” (M2.5) o bien se entre en parada de emergencia.

Normalmente, las órdenes asociadas a pulsadores se deben a que los actuadores son de doble efecto o por motivos de simplificación a la hora de utilizar el panel.

Las órdenes que llevan **conmutadores** se asocian a actuadores que son de simple efecto, donde estos actuadores solo interesa tenerlos activos durante un período de tiempo concreto.

En la **Figura 35** se observa que cuando el conmutador emparejado a la variable “HMI_Soplar” se active, la salida permanecerá activa hasta que se produzca un flanco de bajada de la variable ligada a la orden.

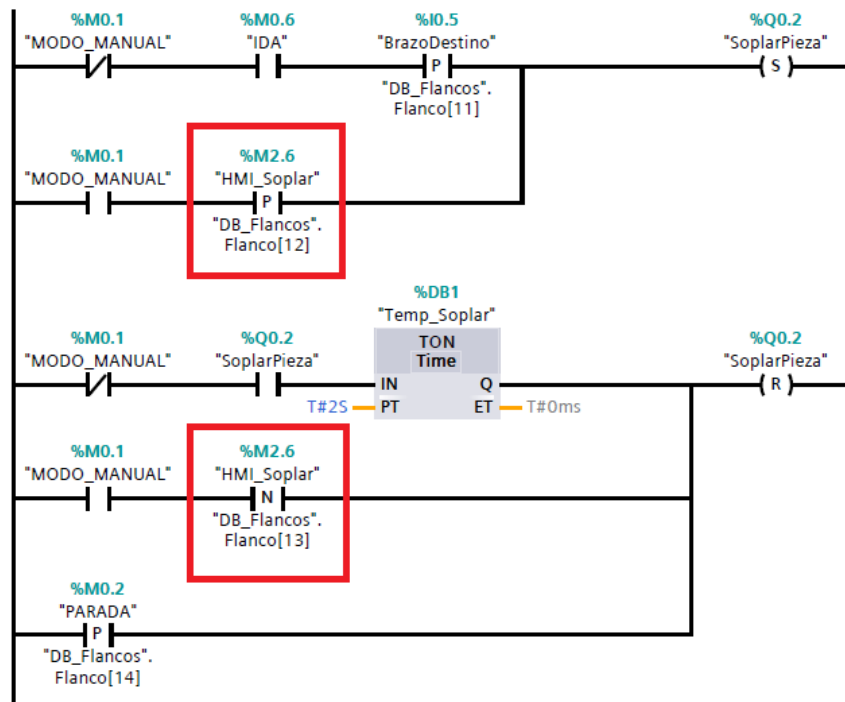


Figura 35. Fragmento de código de la Estación 0

El **panel de visualización** se compone de pilotos que indican cuáles son los sensores y actuadores que se encuentran activos. Estos pilotos son objetos básicos con forma de circunferencia que se emparejan a todos los actuadores y sensores del proceso, y que cambian su apariencia en función del valor que obtenga la variable. Tienen un color gris mientras la variable tiene valor '0' y en el momento que tomen el valor '1' se ponen de color verde.

Por último, hay que destacar que la imagen interactiva del modo manual está parcialmente sincronizada con la imagen en modo automático. Si se activa una orden mientras la imagen del modo manual está activa, se verá reflejado en los pilotos del panel, cambiando a la imagen del modo automática es posible observar el estado de los actuadores del proceso. Solo será posible observar aquellos objetos que están ligados a actuadores, los objetos como las piezas procesadas no serán visibles en el modo manual.

A continuación, se muestran las principales *órdenes de control* de las que dispone cada imagen del modo manual.

4.4.3.8 Órdenes de control

ÓRDENES PRINCIPALES DE LA ESTACIÓN 0



Figura 36. Órdenes de la Estación 0

MOVER PREVIA: (*HMI_MoverPrevia – M2.3*) Activa el motor de girar a la estación previa (*Q0.3*). Desactivará inmediatamente el motor de girar a la estación posterior en caso de estar activa.

MOVER DESTINO: (*HMI_MoverDestino - M2.2*) Similar a la orden anterior. Activará el motor de girar a la estación posterior (*Q0.4*) y desactivará el motor de girar a la estación previa.

RETRAER PISTÓN: (*HMI_RetraerPist – M2.4*) Retraerá el pistón (*Q0.0*) expulsando una pieza del almacén. Se mantendrá retraída hasta que se active la orden de extender el pistón.

EXTENDER PISTÓN: (*HMI_ExtenderPist – M2.5*) Extenderá el pistón de expulsión poniéndolo en posición inicial, cuando se extienda del todo (indicado por un sensor) automáticamente se desactivará.

SUCCIONAR PIEZA: (*HMI_Succion – M2.7*) Orden que se activa por medio de un conmutador, esto quiere decir que el motor de succionar (*Q0.2*) permanecerá activo hasta que el conmutador se desactive.

SOPLAR PIEZA: (*HMI_Soplar – M2.6*) Orden similar a la anterior, el motor de soplar pieza (*Q0.1*) se activará cuando el conmutador esté a 1 y además desactivará el motor de succionar pieza (*Q0.2*) en caso de que estuviera activo.

ÓRDENES PRINCIPALES DE LA ESTACIÓN 1



Figura 37. Órdenes de la Estación 1

SUBIR PLATAFORMA: (*HMI_SubirPlat – M2.3*) Activa el motor de subir plataforma (Q0.1), se mantendrá activo hasta que se active la orden de bajar la plataforma o se pulse la parada de emergencia.

BAJAR PLATAFORMA: (*HMI_BajarPlat – M2.2*) Activa el motor de bajar plataforma (Q0.0), una vez llegue al final de carrera 1 (indicado por un sensor) se parará el motor.

EXTENDER PISTÓN: (*HMI_Expulsar – M2.5*) Activa el motor de extender el pistón (Q0.2) para expulsar la pieza por la rampa, mientras el conmutador permanezca activo el pistón estará extendido.

SUJETAR PIEZA: (*HMI_SujetarPieza – M2.6*) Activa el motor del pistón de sujeción (Q0.4), que sujeta la pieza al inicio de la rampa del final de carrera 2. Este pistón se mantendrá activo hasta que se desactive el conmutador.

MEDIR PIEZA: (*HMI_Medir – M2.4*) Activa el motor que baja el sensor de medida de la pieza (Q0.3), este sensor se mantendrá abajo mientras se mantenga el conmutador activo.

PIEZA BUENA: (*M0.4*) Esta orden sirve para indicarle a la estación que la pieza medida tiene un tamaño correcto.

ÓRDENES PRINCIPALES DE LA ESTACIÓN 2

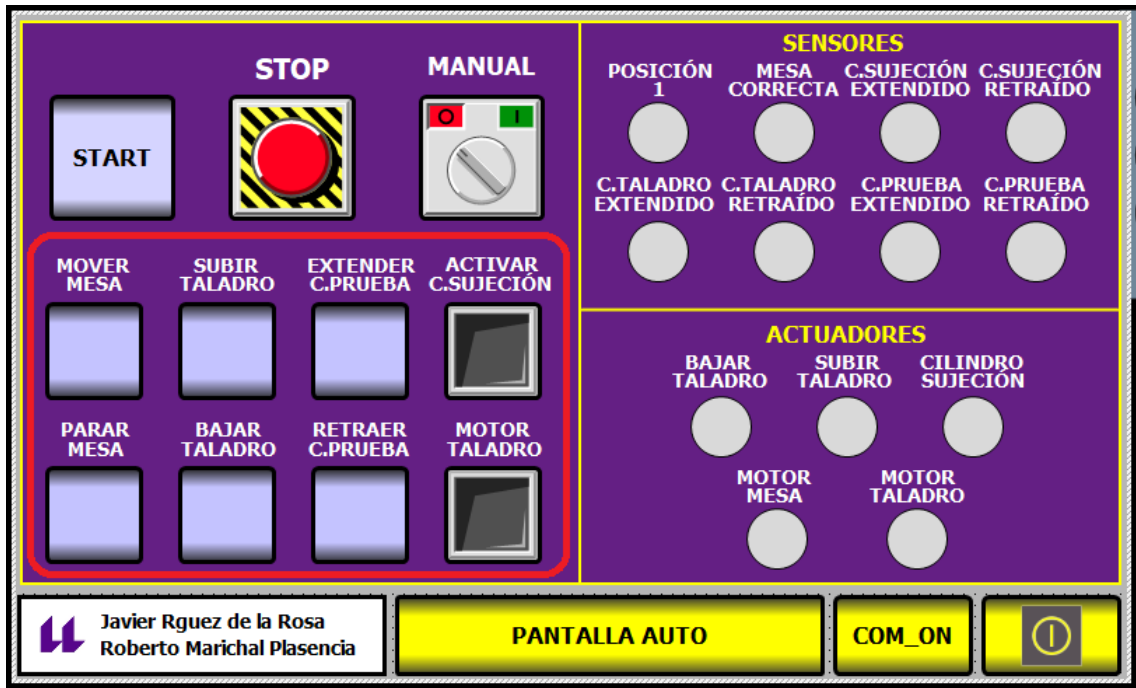


Figura 38. Órdenes de la Estación 2

MOVER MESA: (*HMI_MoverMesa* – M2.2) Activa el motor que mueve la mesa (Q0.2). La mesa permanecerá girando hasta que active la orden de parar.

PARAR MESA: (*HMI_PararMesa* – M2.3) Detiene el giro de la mesa.

SUBIR TALADRO: (*HMI_SubirTal* – M2.6) Activa el motor de subir el taladro (Q0.4).

BAJAR TALADRO: (*HMI_BajarTal* – M2.5) Activa el motor de bajar el taladro (Q0.3), además desactivará el motor de subir.

EXTENDER CILINDRO DE PRUEBA: (*HMI_ExtPrueba* – M3.0) Activa el motor que baja el cilindro que comprueba si el orificio realizado por el taladro es correcto (Q8.0).

RETRAER CILINDRO DE PRUEBA: (*HMI_RetPrueba* – M3.1) Desactiva la orden anterior y devuelve el cilindro a la posición de retraído.

ACTIVAR CILINDRO DE SUJECIÓN: (*HMI_Sujecion* – M2.4) Activa el cilindro de sujeción (Q0.0).

MOTOR TALADRO: (*HMI_ActTaladro* – M2.7) Activar el motor del taladro (Q0.1), permanecerá activo hasta que se desactive el conmutador o se produzca una parada de emergencia.

ÓRDENES PRINCIPALES DE LA ESTACIÓN 3

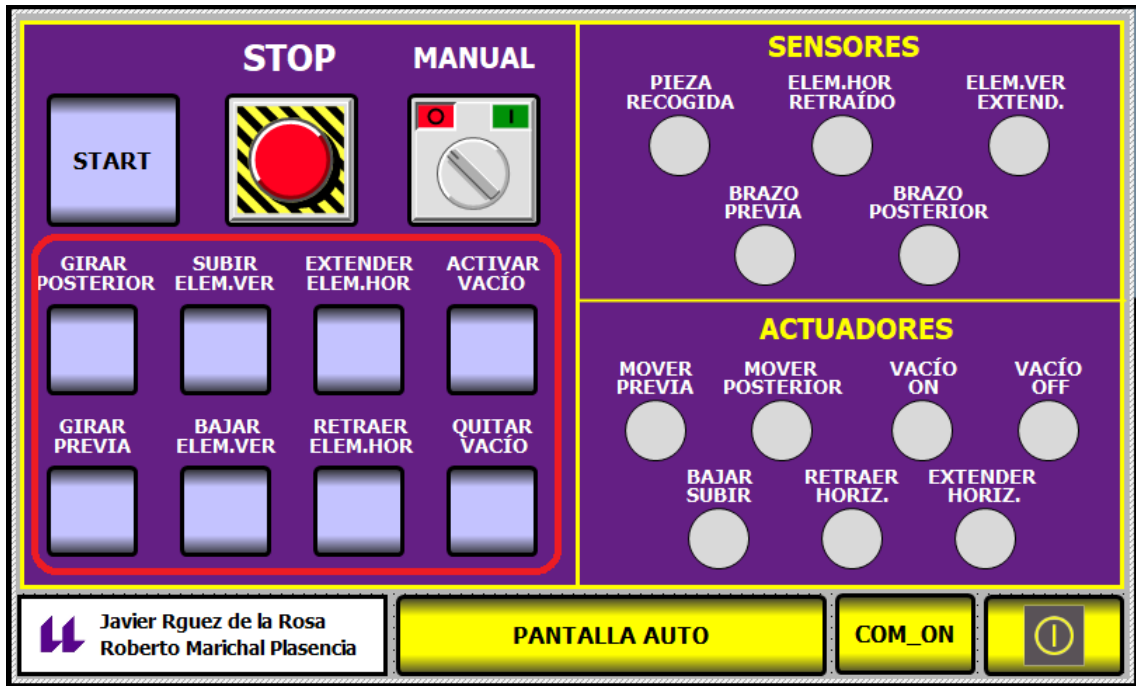


Figura 39. Órdenes de la Estación 3

GIRAR POSTERIOR: (HMI_GirarPost – M2.5) Activar el motor que gira el brazo a la estación posterior (Q0.2). Desactivará el motor de girar a previa en caso de estar activo.

GIRAR PREVIA: (HMI_GirarPrev – M2.4) Activa el motor que gira el brazo a la estación previa (Q0.3). Desactivará el motor de girar a posterior en caso de estar activo.

SUBIR ELEMENTO VERTICAL: (HMI_SubirElem – M3.0) Activa el motor que el elemento vertical del brazo (Q8.1).

BAJAR ELEMENTO VERTICAL: (HMI_BajarElem – M3.1) Desactiva el motor de bajar el brazo (Q8.1) y por tanto volverá a subir a su posición inicial.

EXTENDER ELEMENTO HORIZONTAL: (HMI_ExtElem – M3.2) Activa el motor que extiende el brazo horizontalmente (Q0.1). En caso de que se active la orden de retraer se desactivará el motor.

RETRAER ELEMENTO HORIZONTAL: (HMI_RetElem – M3.3) Activa el motor que retrae el brazo horizontalmente (Q0.0). En caso de que se active la orden de extender se desactivará el motor.

ACTIVAR VACÍO: (HMI_VacioON – M2.6) Activa el vacío (Q0.4). Se mantendrá activo hasta que se active la orden de quitar el vacío.

QUITAR VACÍO: (HMI_VacioOFF – M2.7) Desactive el vacío (Q0.4) activando el (Q8.0) que es el Vacío OFF.

ÓRDENES PRINCIPALES DE LA ESTACIÓN 4

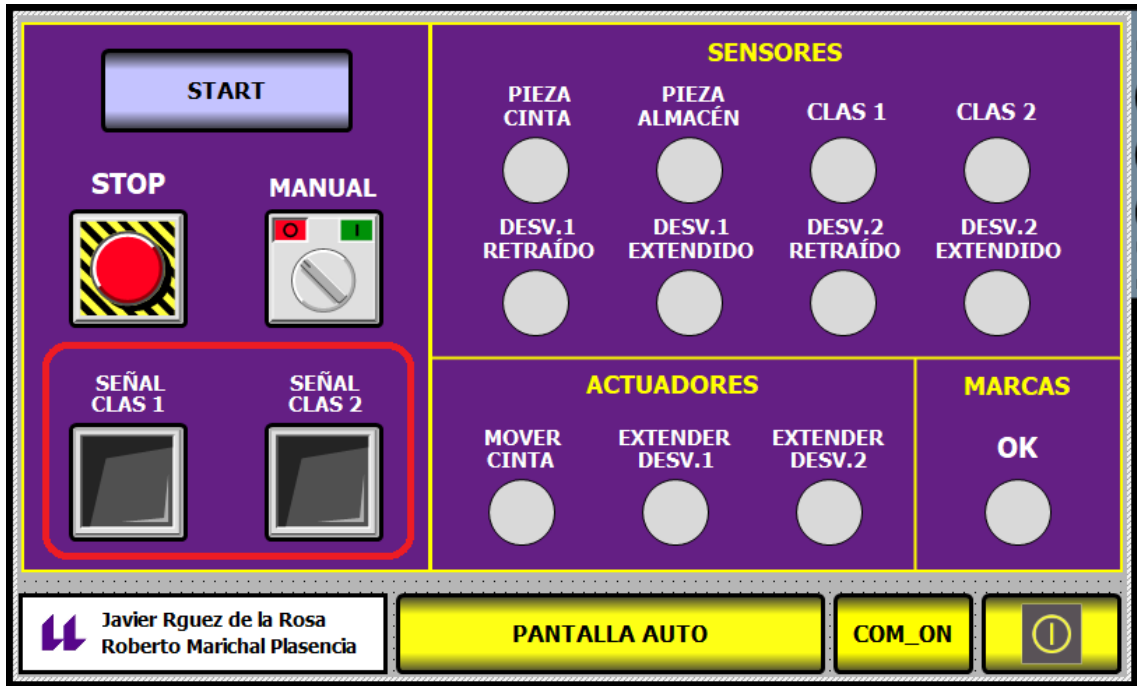


Figura 40. Órdenes de la Estación 4

SEÑAL CLAS1: (*HMI_CLAS1 – M2.2*) Activar el actuador neumático de la posición 1 (*Q0.0*). Se mantendrá extendido hasta que se desactive el conmutador.

SEÑAL CLAS2: (*HMI_CLAS2 – M2.3*) Activar el actuador neumático de la posición 2 (*Q0.1*). Se mantendrá extendido hasta que se desactive el conmutador.

4.4.3.9 Detalles del funcionamiento de los objetos en modo automático

OBJETOS DE LA ESTACIÓN 0

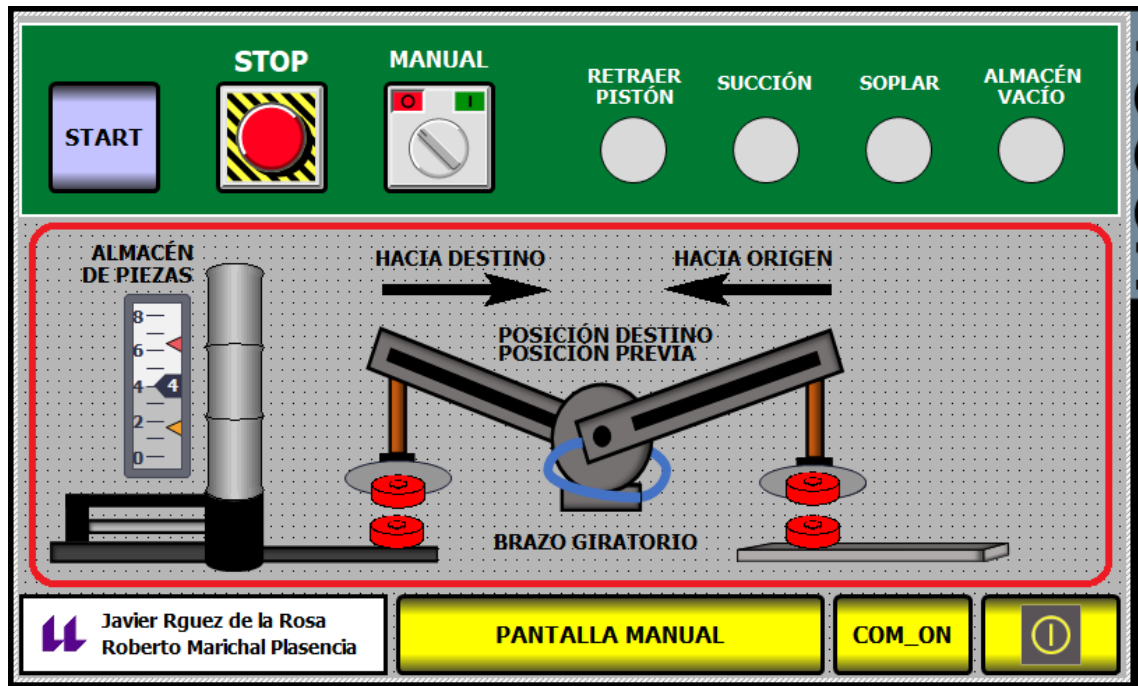


Figura 41. Imagen Modo Automático Estación 0

Los objetos de esta imagen son los más básicos de las cinco estaciones, se tienen dos objetos principales que son las dos posiciones posibles del brazo, cuando una esté activa la otra no será visible. La visualización distinguiremos cuatro posiciones diferentes, cuando se encuentre en la posición previa o en la de destino, y si está succionada por la ventosa o no.

Respecto a las ventosas, cuando se active el motor de succión, la ventosa parpadeará entre un color negro y azul claro, indicando que la pieza está siendo succionada. Por el contrario, si el motor de soplar se activa, parpadeará de color negro y naranja.

OBJETOS DE LA ESTACIÓN 1

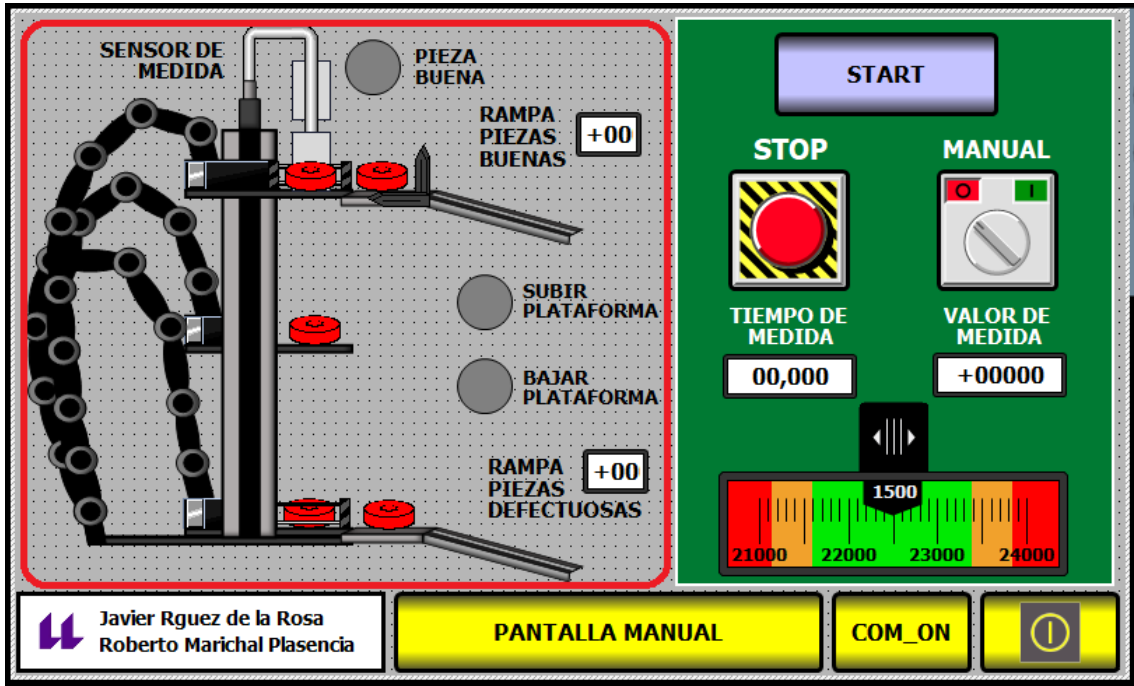


Figura 42. Imagen Modo Automático Estación 1

Esta imagen está compuesta en su mayor parte por la plataforma de elevación, lo que se ha hecho es, crear tres plataformas diferentes para cada posición y superponerlas. Con esto conseguimos que cuando la pieza esté en la posición inferior se vea que la plataforma está abajo, cuando esté en el medio que se vea un punto intermedio, y que cuando llegue a la posición superior se vea en el punto máximo.

Los pistones de expulsión tienen dos objetos, uno para cuando está retraído y otro para cuando está extendido, cuando se extiende el pistón, el objeto de retraído dejará de ser visible.

Por último, las piezas que bajan por la rampa solo serán visibles cuando se den las condiciones necesarias. Una vez haga todo el recorrido de la trayectoria dejará de ser visible de nuevo.

OBJETOS DE LA ESTACIÓN 2

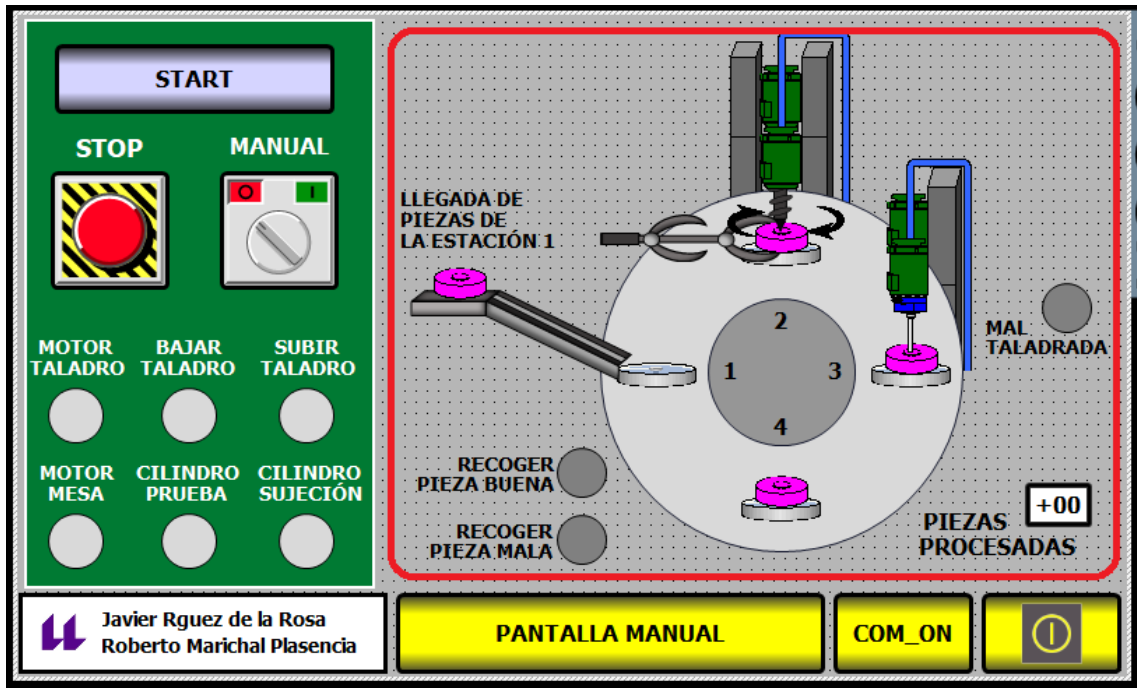


Figura 43. Imagen Modo Automático Estación 2

Los objetos principales de esta imagen son las piezas, para lograr la correcta visualización de las piezas a medida que avanzan posiciones se ha optado por un método bastante peculiar. Se ha creado un contador que aumenta su valor cada vez que la mesa se mueve. En cada posición se ha puesto la misma pieza cambiada de color hasta seis veces, entonces cada vez que el contador cambie su valor, las piezas cambiarán su visibilidad para conseguir un efecto visual de que las piezas realmente cambian su posición respecto al giro anterior de la mesa.

En cada posición se han definido varias posibilidades para los objetos en función de si están extendidos o retraídos. En cuanto a la mesa, simplemente es una circunferencia que parpadea en azul cuando está girando.

OBJETOS DE LA ESTACIÓN 3

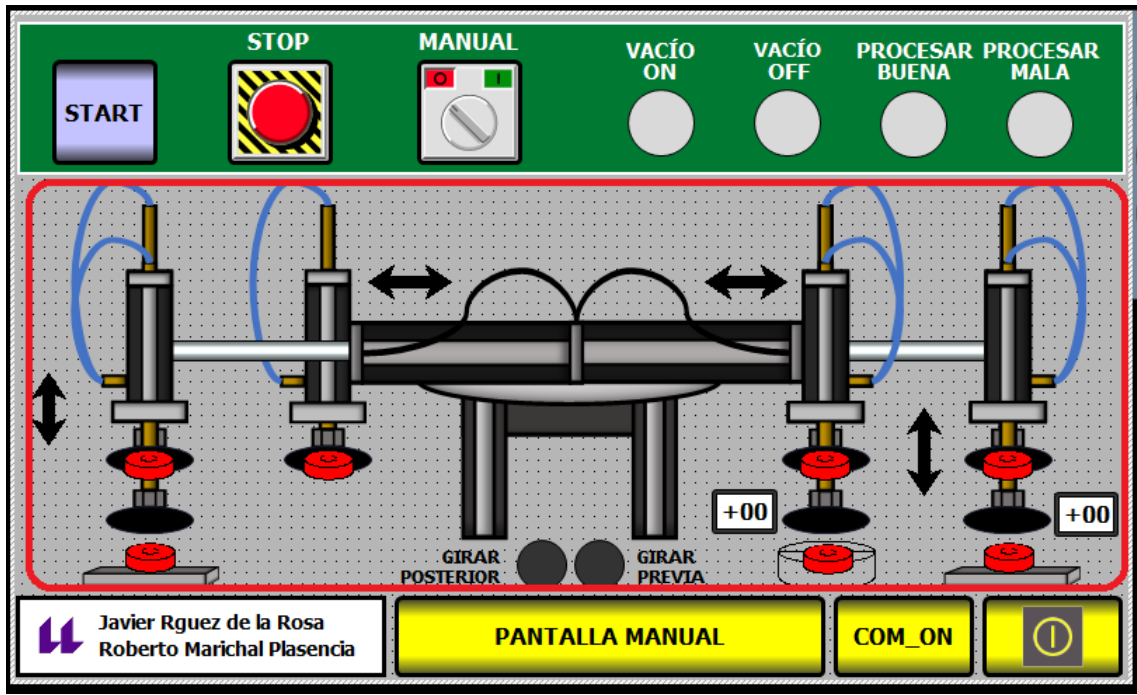


Figura 44. Imagen Modo Automático Estación 3

Esta imagen probablemente es la más difícil de entender del conjunto. El principal problema de esta imagen es que es necesario que para la extensión de los actuadores del brazo no se deje de ver su elemento fijo. Para lograr esto, lo que se ha hecho es crear todas las posibles extensiones por separado y superponerlas. Como programar el movimiento a raíz de estas condiciones es bastante complicado (dado que también se movería el elemento fijo y sería un problema), se han utilizado flechas que parpadean para indicar si alguno de los elementos extensibles está activo.

En cuanto a las piezas, se ha elegido una visualización de cara al transporte de las mismas, la visualización seguirá un orden que va a la par con la posición actual del brazo.

OBJETOS DE LA ESTACIÓN 3

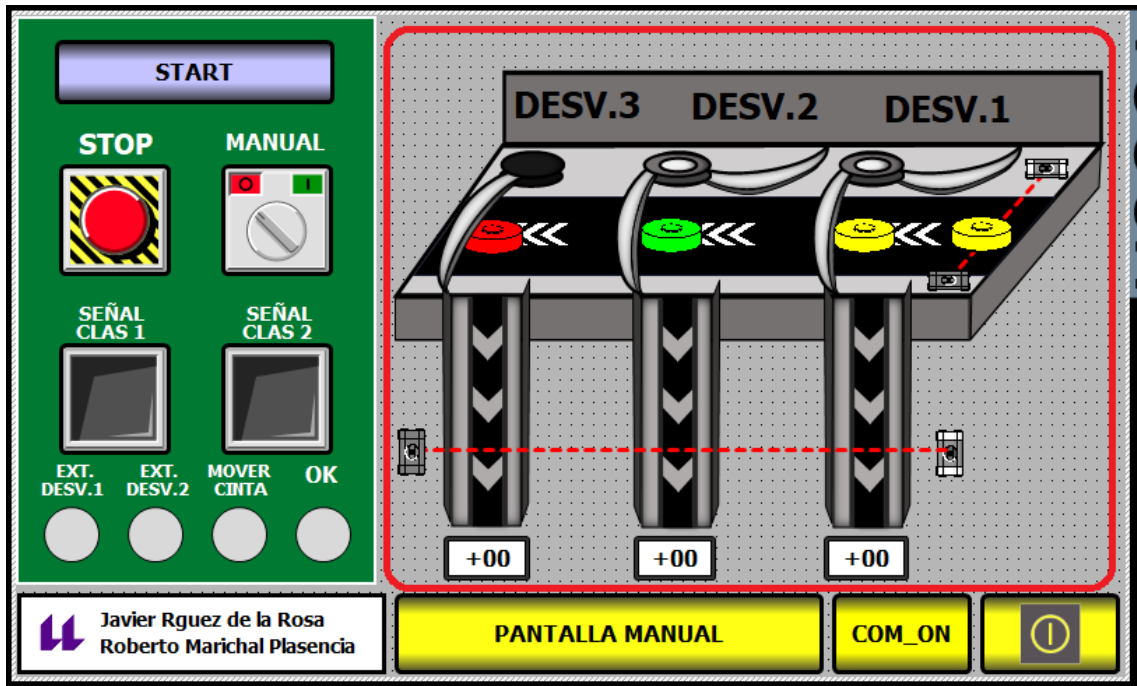


Figura 45. Imagen Modo Automático Estación 4

El funcionamiento de los objetos de esta imagen es muy sencillo. Se tiene una base fija que es la plataforma de la cinta, tres rampas para las piezas y los desviadores de piezas.

Los desviadores de piezas tienen dos posibles posiciones, la de extendido y la de retraído, cuando se activa la señal de extender los desviadores el objeto del desviador retraído desaparece.

Para las piezas se ha optado por un movimiento horizontal y vertical, se han establecido tres contadores para cada una de las piezas en función del desviador activo. El movimiento y visualización de las piezas estará en dividido en dos partes iguales, si el valor establecido para el contador es 30, se tendrá un movimiento horizontal en el rango de 1 a 15 y un movimiento vertical en el rango de 16 a 30.

5. Presupuesto

La elaboración de este presupuesto no contempla el precio ni la implementación de las estaciones, pues es la base sobre la que el cliente pide la creación de este proyecto.

| MATERIAL | UNIDADES | PRECIO TOTAL |
|--|--------------|----------------|
| PLC S7-1500. Modelo CPU 1516-3 PN/DP (con módulos incluidos) | 2 (1393 €/u) | 2786 € |
| PLC S7-1200. Modelo CPU 1212C AC/DC/Rly (sin módulos) | 3 (265 €/u) | 795 € |
| Módulo SM1223 DI8/DQ8 x relé (S7-1200) | 3 (150€/u) | 450 € |
| Módulo de comunicaciones PROFINET/PROFIBUS | 3 (350€/u) | 1050 € |
| HMI Tp700 Comfort Panel | 1 | 549 € |
| Switch D-LINK | 1 | 25 € |
| SIMATIC WinCC Basic con TIA Portal integrado | 1 | 2500 € |
| Horas de trabajo invertidas | 350 (20€/h) | 7000 € |
| TOTAL | - | 15155 € |

6. Conclusion

First of all, the automation of the *FESTO*'s stations was elaborated in two different types of languages; *KOP* and *AWL*. Specifically, the stations connected to *S7-1200* were programmed in contact diagrams (*KOP*). The great advantage of this language is the visual design, for this reason was much easier to implement automations with this language. On the other hand, the second language was used in the stations connected to *S7-1500*, with this language is harder to spot the errors because is a language based on instruction lists that does not use icons. For programming the *PLC*'s connected to the stations was used *TIA Portal*, this software allowed logical programming in both languages. In addition, using a simulation tool called *PLCSIM* was possible to simulate the behavior of the stations. This behavior shows the correct functioning of the process.

Once the operation of the stations was checked, the design of the *HMI* screens was made, using the *WinCC* tool integrated into the *TIA Portal*. This tool allows you to create interactive images and transfer them to a touch panel that allows the operator to interact with the processes. The main problema encountered when making these screens, is that there were no objects similar to the actuators of the stations. In the first instance, an attempt was made to program the interactive images with the objects available in the *TIA Portal* library, but ended up being intelligible and un-visual. As a result, was necessary to investigate how to create objects externally and import them into the object library.

Finally, communications between *PLC*'s have been a big problema. To ensure that the stations operate sequentially it is necessary that there are no conflicts between them when operating. To achieve this objective it is necessary that in addition to communicating, a priority is established between request and shipments.

7. Conclusión

En primer lugar, la automatización de las estaciones *FESTO* se realizó en dos lenguajes diferentes; *KOP* y *AWL*. En concreto, en las estaciones conectadas con los autómatas *S7-1200* se utilizó el lenguaje de diagrama de contactos o *KOP*, la gran ventaja de éste era su diseño visual, y por este motivo fue más sencillo implementar las automatizaciones con este lenguaje. Por otra parte, el segundo lenguaje, el *AWL*, se utilizó en los autómatas

S7-1500, con este lenguaje se tuvo mayor dificultad a la hora de detectar los errores, ya que es poco visual al no manejar iconos sino líneas de instrucciones. Para la programación de los *PLC's* correspondientes a las estaciones se utilizó el entorno *TIA Portal*, este software permitió la programación lógica en ambos lenguajes, y una vez terminado, mediante la herramienta de simulación *PLCSIM*, se simuló la lógica programada para comprobar que el comportamiento de las estaciones era el correcto.

Una vez comprobado el funcionamiento de las estaciones, se continuó con el diseño de las pantallas *HMI*, mediante la herramienta *WinCC* integrado en el *TIA Portal*. Esta herramienta permite elaborar imágenes interactivas y transferirlas a un panel táctil que permite al operario interactuar con los procesos. El principal problema encontrando a la hora de elaborar estas pantallas, es que no se disponía de objetos parecidos a los actuadores de las estaciones. En primera instancia, se intentó programar con los objetos disponibles en la librería del *TIA Portal* las imágenes interactivas, pero terminó quedando inteligible y poco visual. A raíz de esto, se investigó la manera de crear objetos de manera externa e importarlos a la librería de objetos.

Finalmente, las comunicaciones entre *PLC's* han supuesto un gran problema. Para lograr que las estaciones funcionen de forma secuencial es necesario que no existan conflictos entre ellas a la hora de operar. Para conseguir este objetivo es necesario que además de comunicarse se establezca una prioridad entre las peticiones y los envíos.

8. Plan de mejora

La principal propuesta para mejorar este proyecto es la creación de un servidor central donde el *SCADA* almacene todos los datos, como puede ser el caso de los servidores *OPC*, este servidor hace de interfaz, comunicando, por un lado, con una o más fuentes de datos utilizando sus protocolos nativos (*PLC's*, Módulos *I/O*, controladores, etcétera) y, por otro lado, Clientes *OPC (SCADA)*. Este tipo de servidores permite comunicaciones bidireccionales y funciones de lectura y escritura para los dispositivos emparejados.

Gracias a la implementación de este servidor no sólo mejoramos el aspecto de las comunicaciones entre los componentes del sistema, sino que además mejoramos el funcionamiento del *SCADA*, que tendrá acceso a toda la información del sistema en tiempo real.

9. Bibliografía

- [1] La empresa FESTO. (s.f.). *FESTO*. Recuperado en Julio de 2021, de:
<https://www.festo.com/group/es/cms/10054.htm>
- [2] Sigut, M. Marichal, R. (2021). *Guiones de las estaciones FESTO*. Universidad de La Laguna. Automatización Industrial Avanzada. Grado de Ingeniería Electrónica Industrial y Automática.
- [3] SIEMENS. (s.f.). *Totally Integrated Automation*. Recuperado en Julio de 2021, de:
<https://new.siemens.com/ar/es/productos/automatizacion/software-industrial/tia-portal.html>
- [4] TecnoPLC. (s.f.). *Forzar entradas en TIA Portal simulando proyecto*. Recuperado en Julio de 2021, de:
<https://www.tecnopl.com/forzar-entradas-en-tia-portal/>
- [5] Virgilio Vázquez López. (s.f.). *Programación en AWL*. Recuperado en Julio de 2021, de:
http://homepage.cem.itesm.mx/vlopez/lista_de_instrucciones.htm
- [6] SIEMENS. (s.f.). *Datasheets del producto*. Recuperado en Julio de 2021, de:
<https://support.industry.siemens.com/cs/products/6es7212-1be40-0xb0/cpu-1212c-ac-dc-relay-8di-6do-2ai?pid=279623&mlfb=6ES7212-1BE40-0XB0&mf=ps&lc=en-US>

https://cache.industry.siemens.com/dl/files/914/59191914/att_915308/v1/s71500_cpu1516_3_pn_dp_manual_es-ES_es-ES.pdf
- [7] Incibe-cert. (s.f.). *Protocolo Profinet*. Recuperado en Julio de 2021, de:
<https://www.incibe-cert.es/blog/caracteristicas-y-seguridad-profinet>
- [8] Nelly Ayllon. (12/08/2020). *Diferencia entre PROFINET y PROFIBUS*. Recuperado en Julio de 2021, de:
<https://us.profinet.com/la-diferencia-entre-profibus-y-profinet/>
- [9] SIEMENS. (s.f.). *Detalles del producto*. Recuperado en Julio de 2021, de:
<https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6AV2124-0GC01-0AX0>
- [10] Grupdap. (s.f.). *Descripciones técnicas WinCC*. Recuperado en Julio de 2021, de:
http://www.grupdap.es/ficheros/descrip-tecnicas/WinCC_Flexible_Abril_2009.pdf
- [11] Sabaridhasan Pillai. (28/03/2020). *HMI Create Customized Graphic*. Recuperado en Julio de 2021, de:
https://www.youtube.com/watch?v=o10qFtl1-kk&ab_channel=SabaridhasanPillai

ANEXOS

ANEXO 1.
CÓDIGO DE LA
ESTACIÓN 0
(S7-1200)

Startup [OB100]

Startup Propiedades

General

| | | | | | |
|---------------|---------|-------------------|------------|-------------|----|
| Nombre | Startup | Número | 100 | Tipo | OB |
| Idioma | KOP | Numeración | Automático | | |

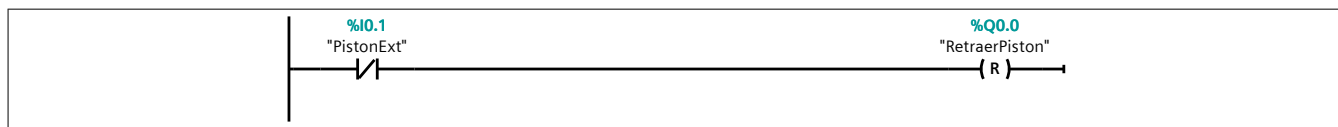
Información

| | | | | | |
|----------------|--------------------|----------------|-----|-------------------------|--|
| Título | "Complete Restart" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

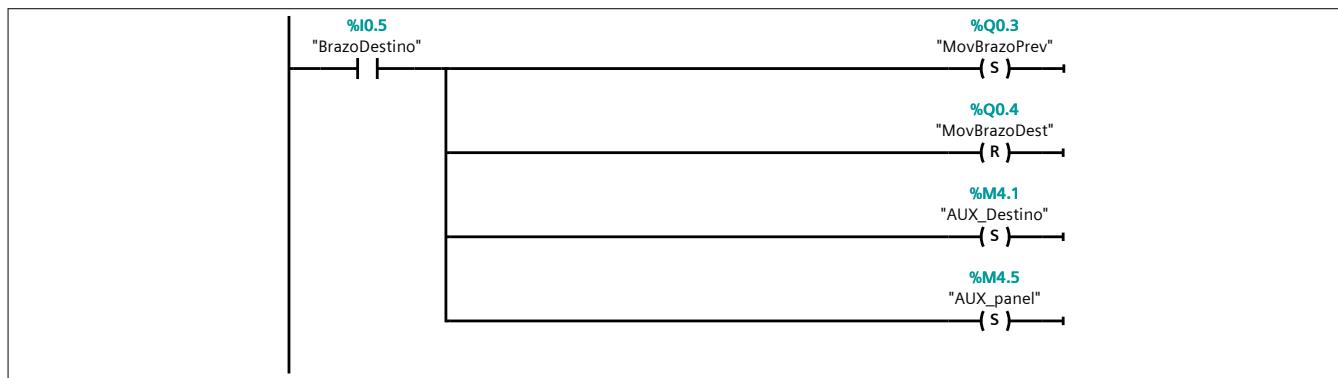
| Nombre | Tipo de datos | Valor predet. | Comentario |
|---------------|---------------|---------------|---------------------------------|
| ▼ Input | | | |
| LostRetentive | Bool | | True if retentive data are lost |
| LostRTC | Bool | | True if date and time are lost |
| Temp | | | |
| Constant | | | |

Segmento 1: PISTON DE EXPULSION

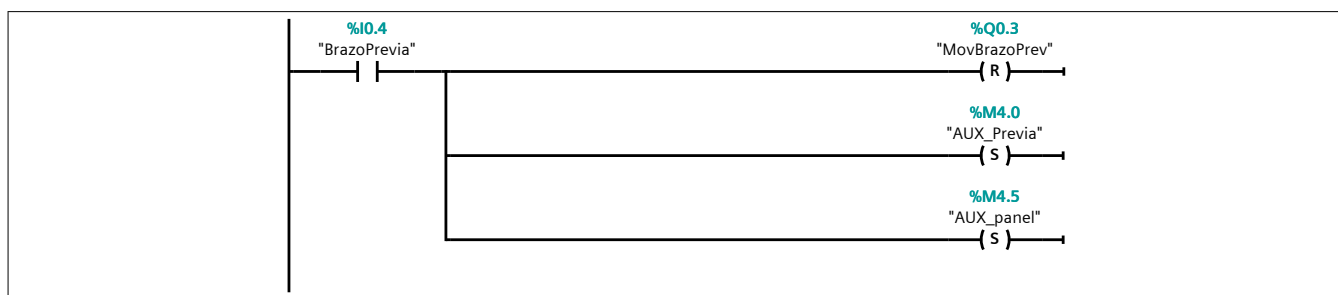
Si el pistón no está extendido reseteamos Q0.0 para que deje de retraerse.



Segmento 2: BRAZO EN POSICION DE DESTINO

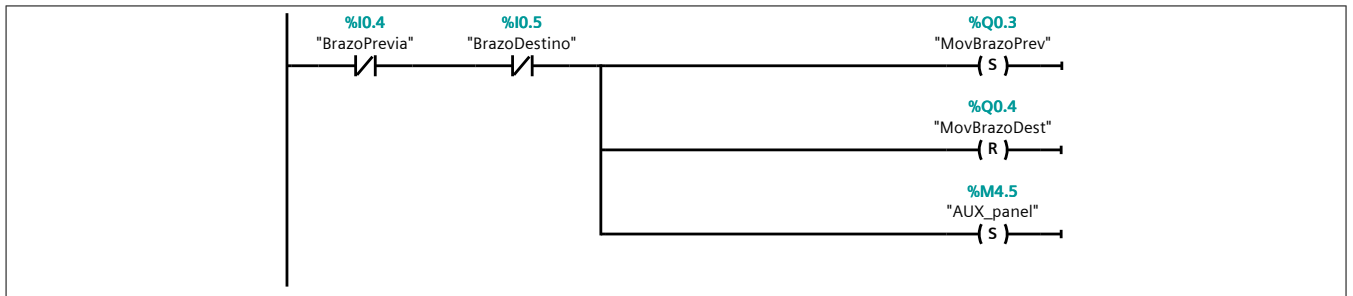


Segmento 3: BRAZO EN POSICION PREVIA

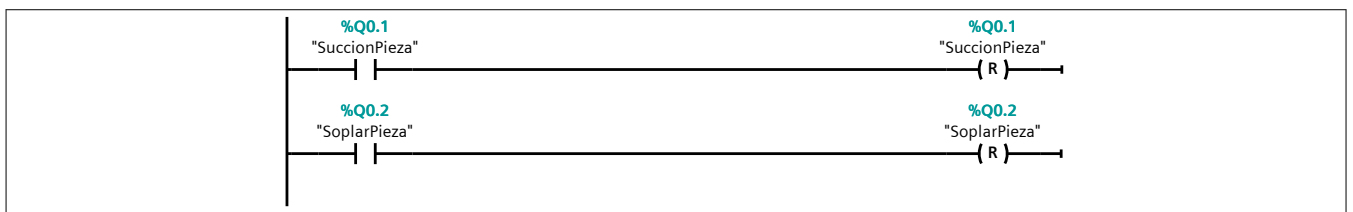


Segmento 4: BRAZO EN ALGUN PUNTO INTERMEDIO

Si no se detecta el brazo en ninguno de los dos sensores se activa siempre el Mover Brazo Previa.



Segmento 5:



Main [OB1]

Main Propiedades

General

| | | | | | |
|---------------|------|-------------------|------------|-------------|----|
| Nombre | Main | Número | 1 | Tipo | OB |
| Idioma | KOP | Numeración | Automático | | |

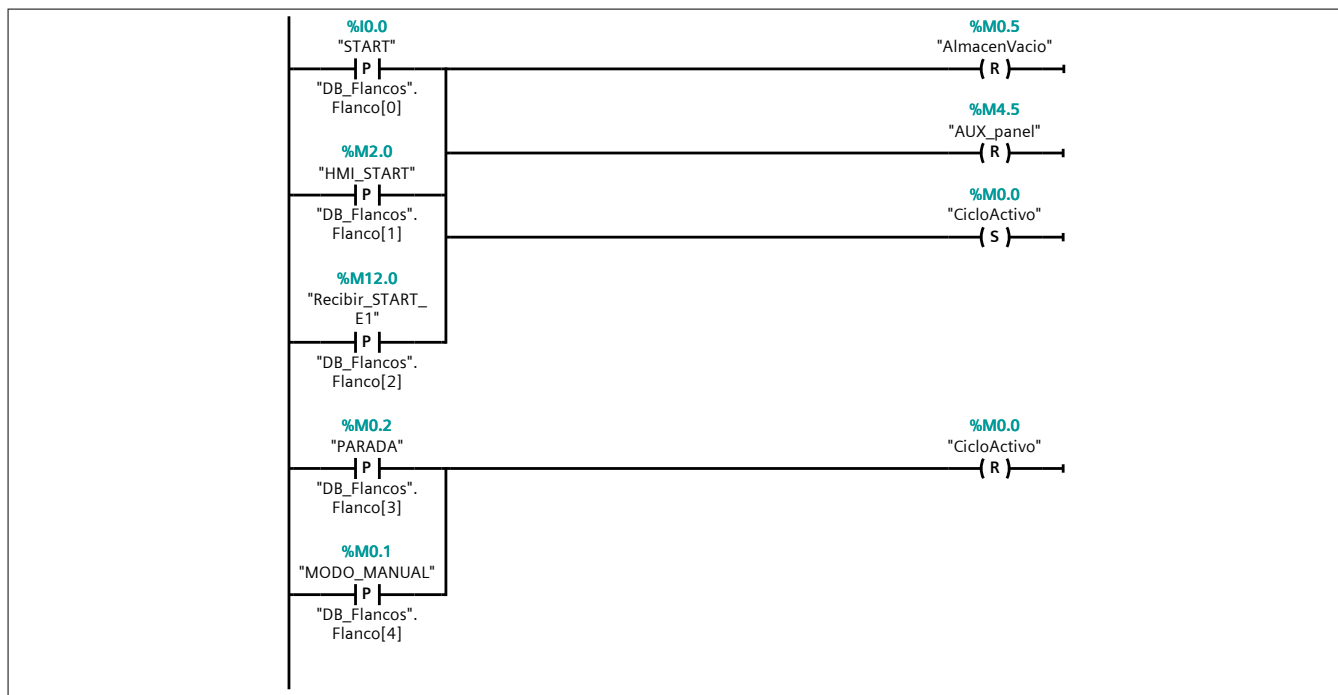
Información

| | | | | | |
|----------------|------------------------------|----------------|-----|-------------------------|--|
| Título | "Main Program Sweep (Cycle)" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

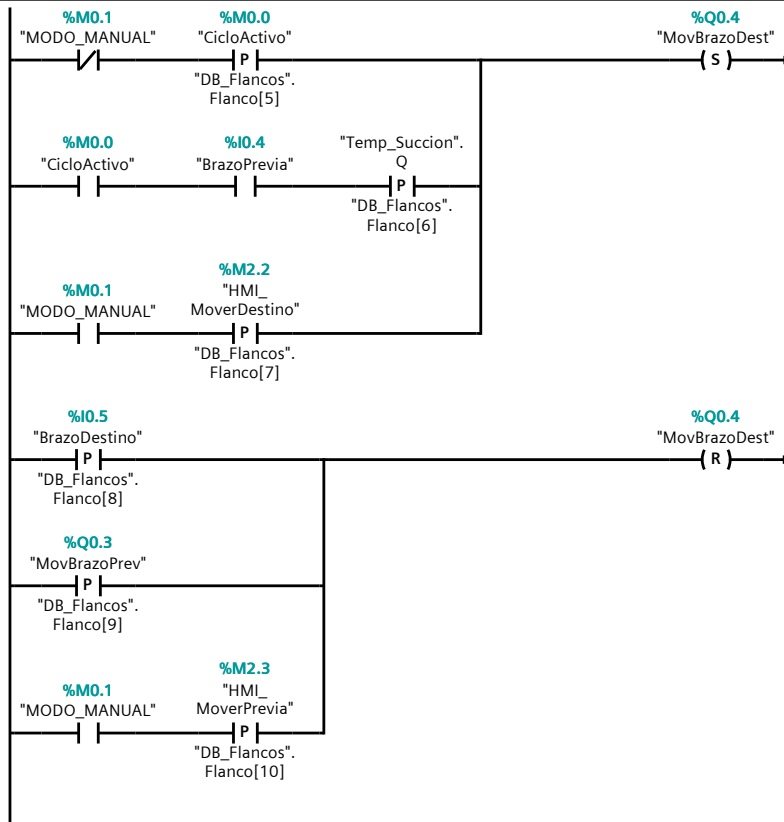
| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|---------------------------------------|
| ▼ Input | | | |
| Initial_Call | Bool | | Initial call of this OB |
| Remanence | Bool | | =True, if remanent data are available |
| Temp | | | |
| Constant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

Segmento 1: Inicio del Ciclo

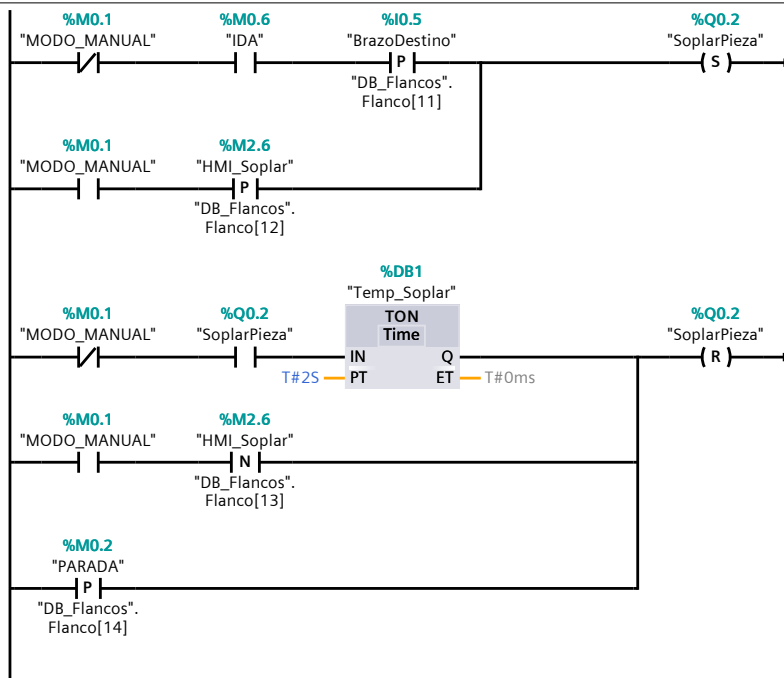
AUTOR: JAVIER RODRÍGUEZ DE LA ROSA - - - TFG - - - 2021



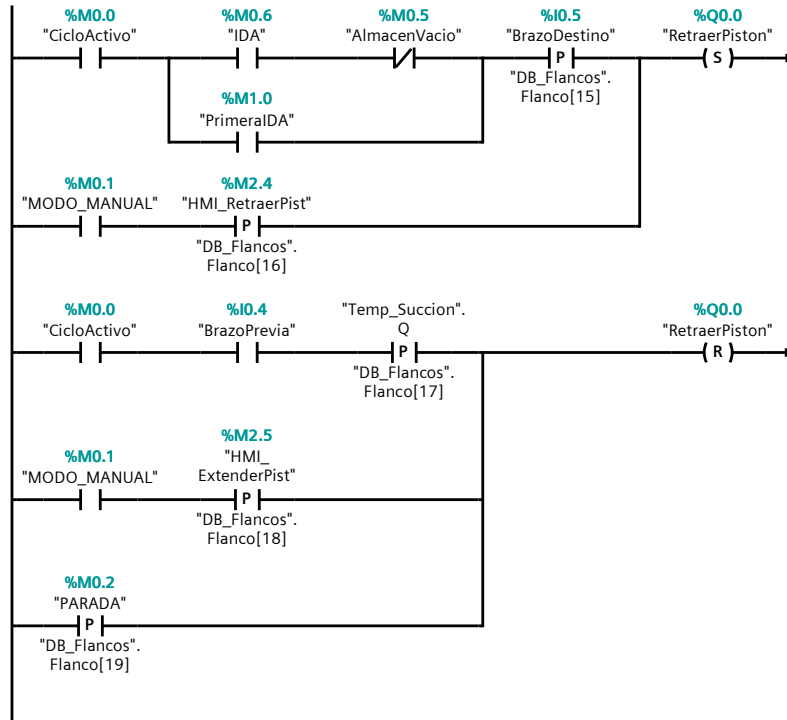
Segmento 2: MoverBrazoDestino (Q0.4)



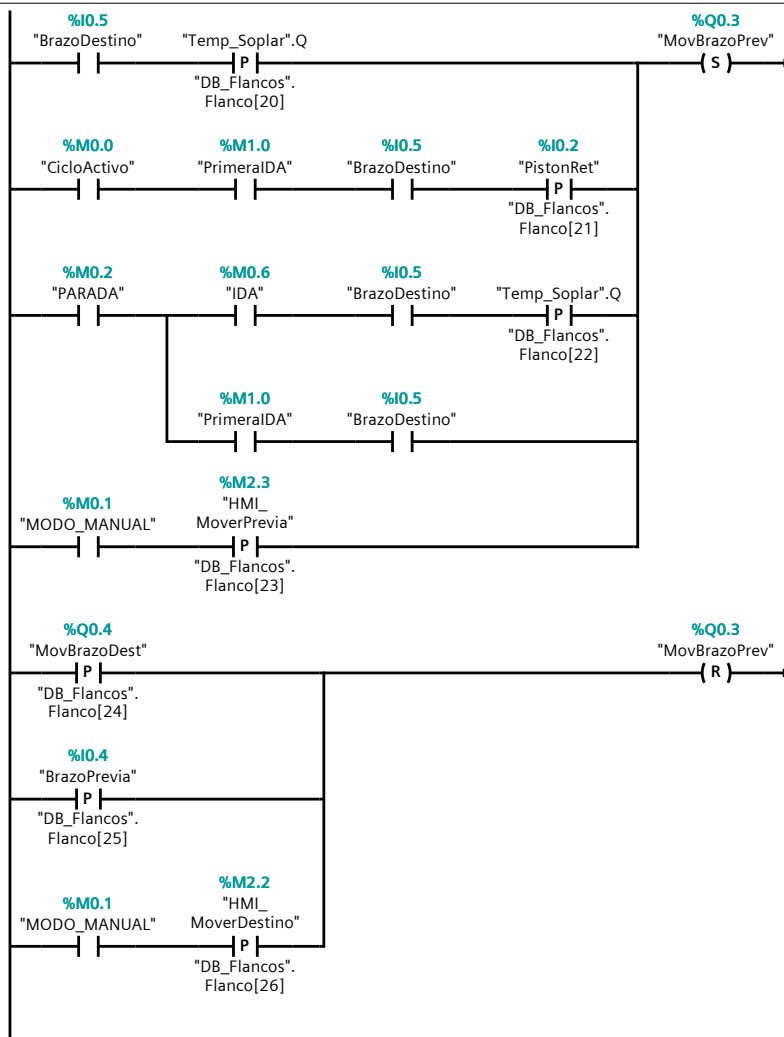
Segmento 3: SoplarPieza (Q0.2)



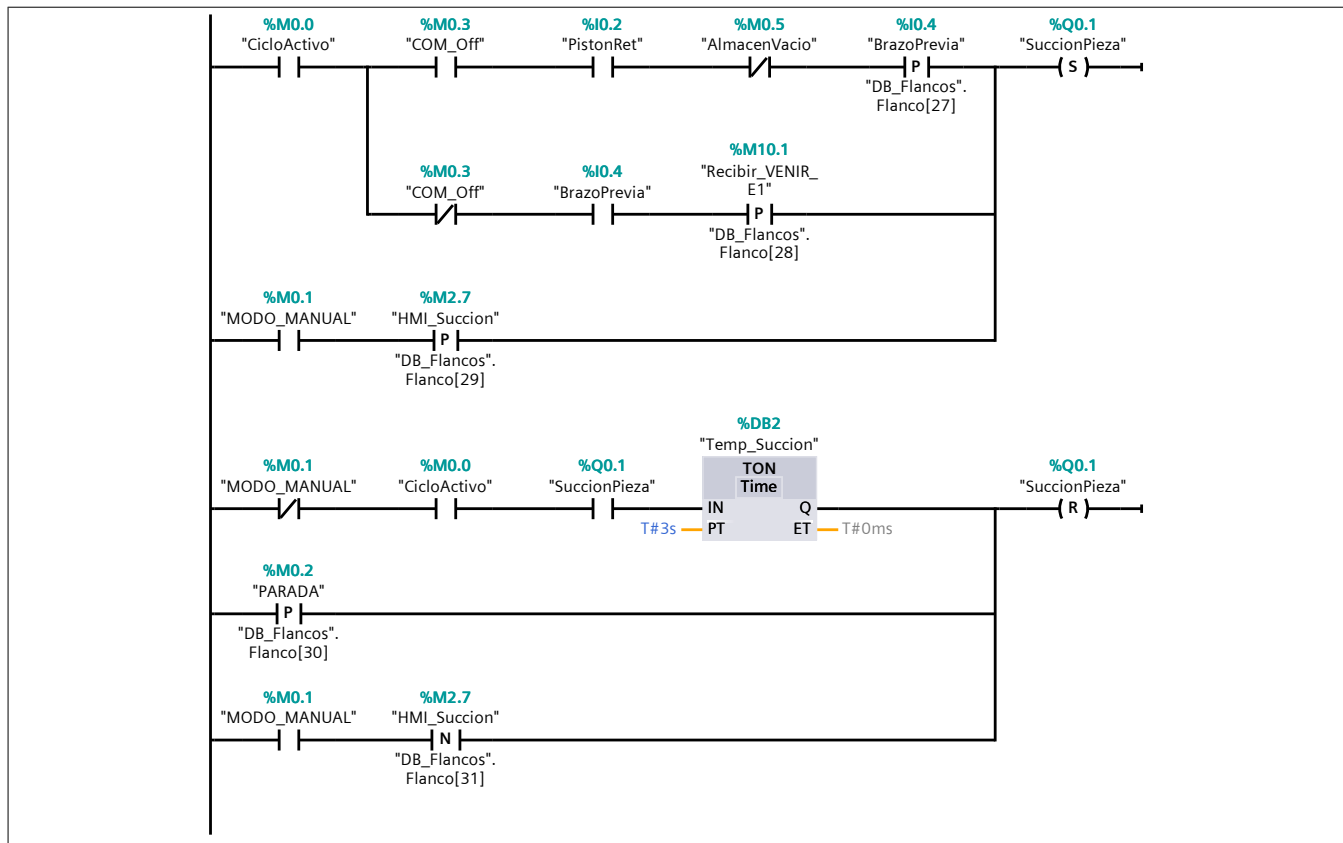
Segmento 4: RetraerPiston (Q0.0)



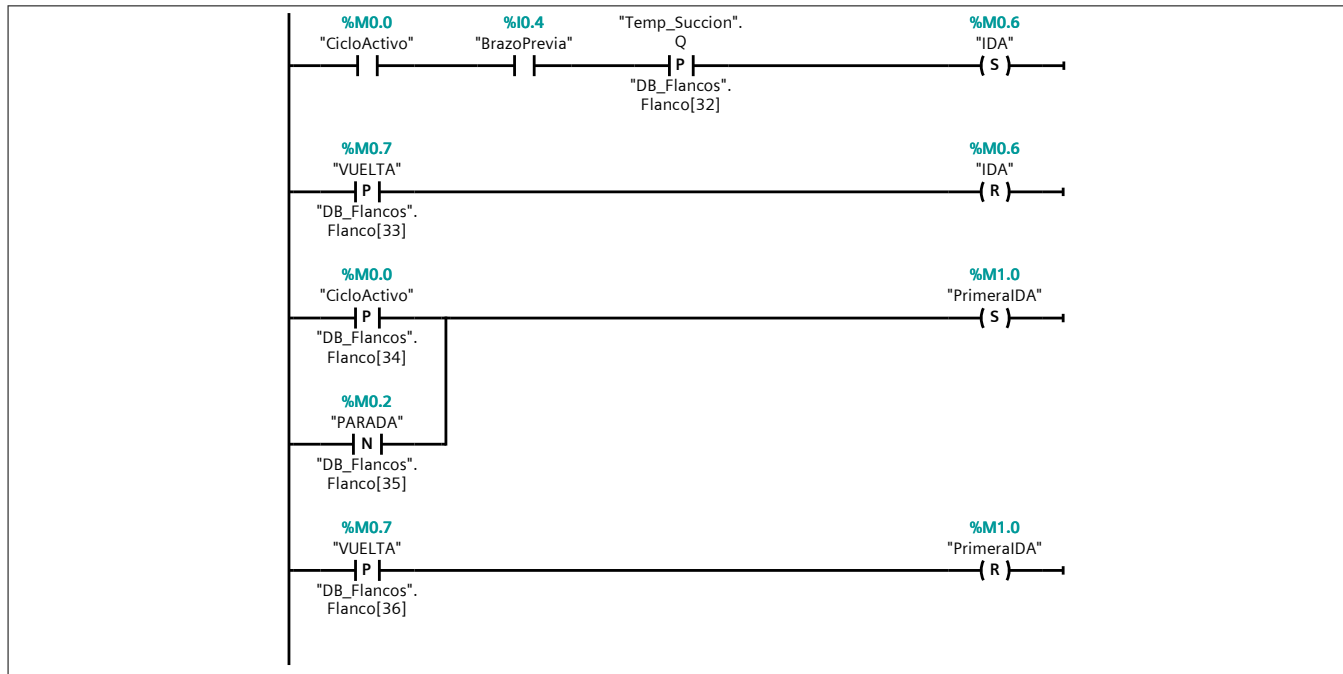
Segmento 5: MoverBrazoPrevia (Q0.3)



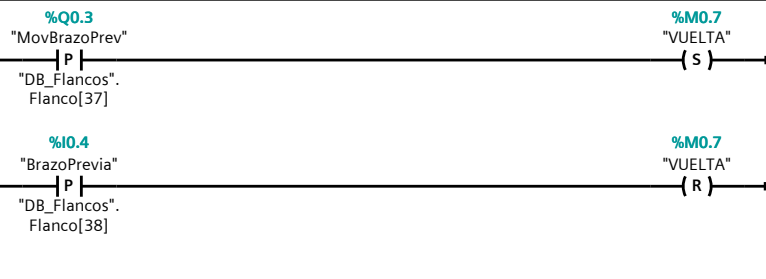
Segmento 6: SuccionPieza (Q0.1)



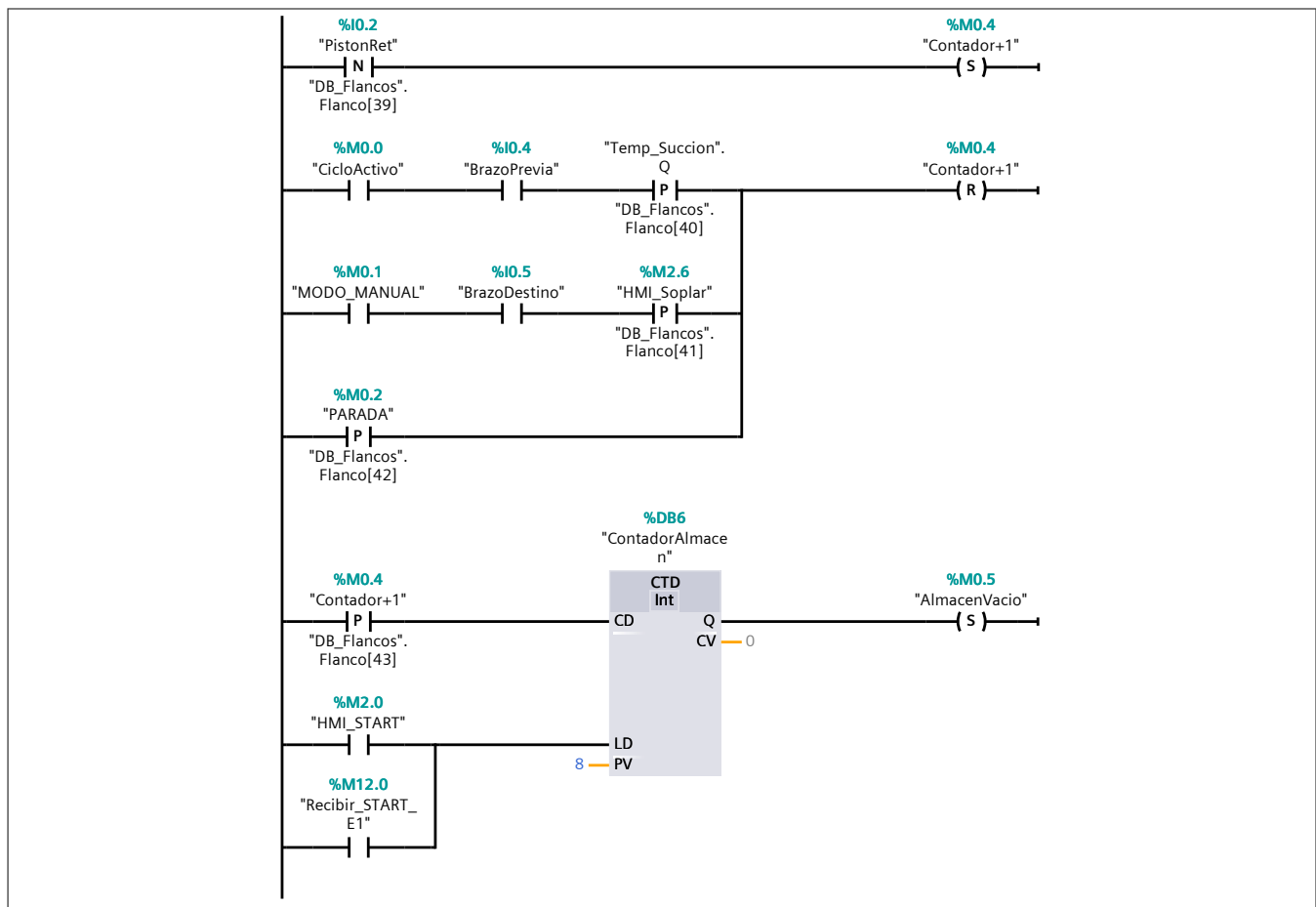
Segmento 7: IDA + PrimeralDA



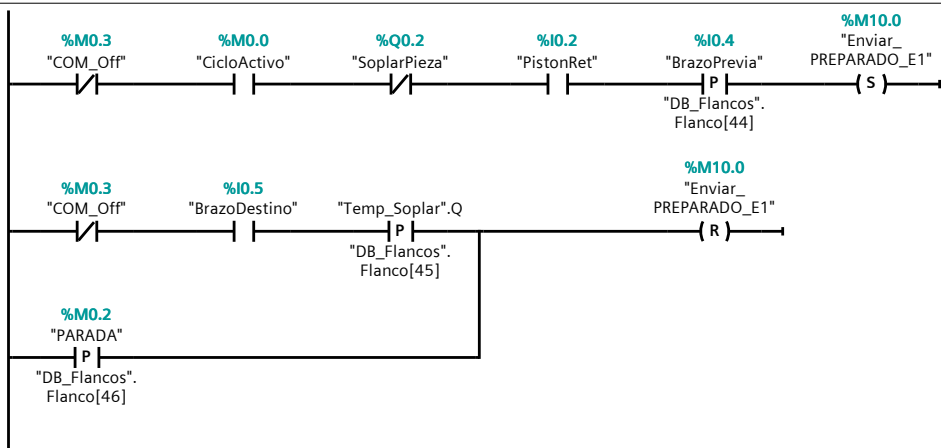
Segmento 8: VUELTA



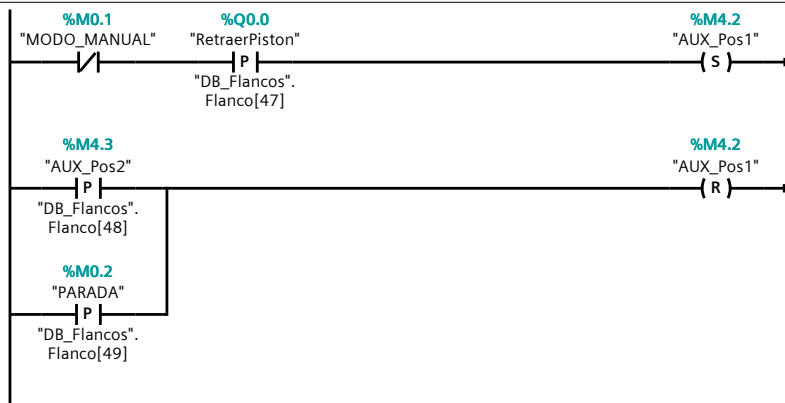
Segmento 9: Contador del Almacen



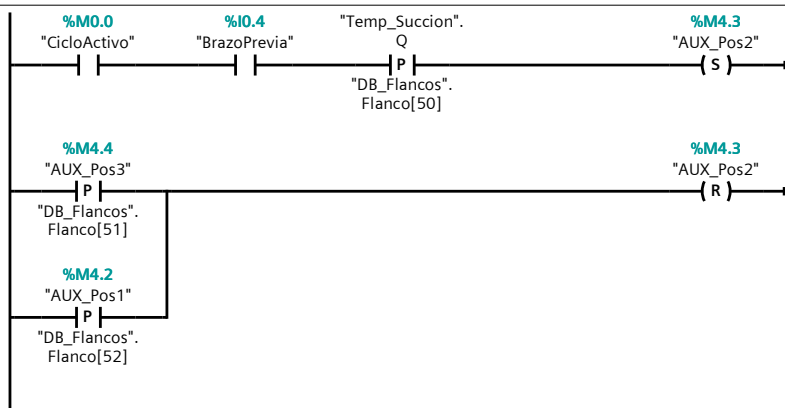
Segmento 10: COMUNICACIONES CON LA ESTACION 1



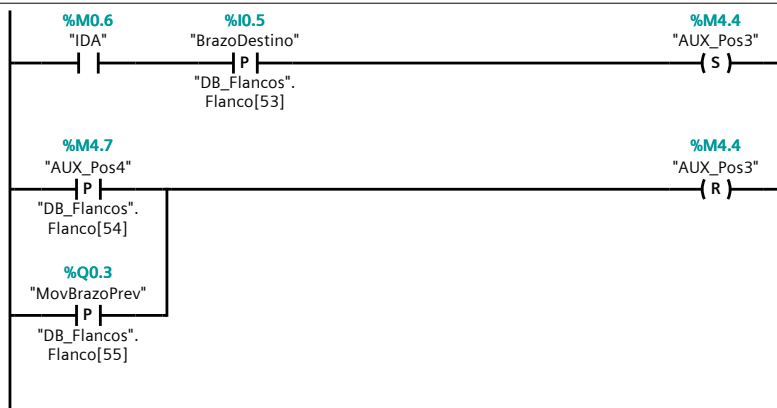
Segmento 11: --- > AUX Pos1



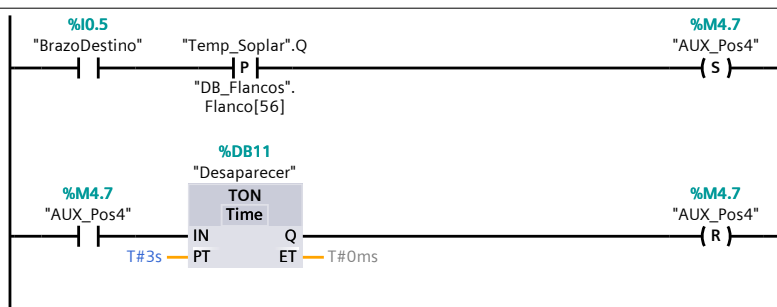
Segmento 12: --- > AUX Pos2



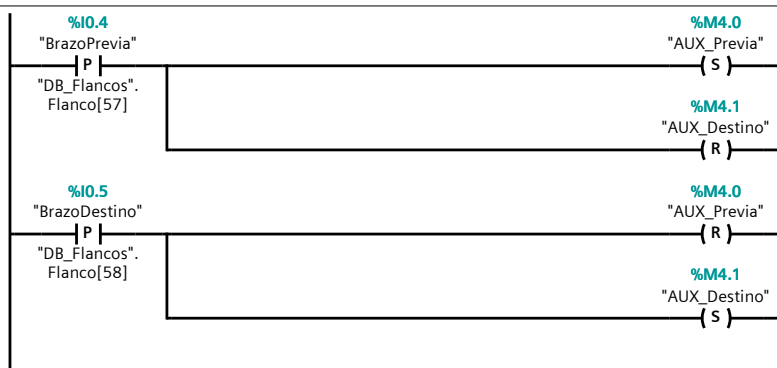
Segmento 13: --- > AUX Pos3



Segmento 14: --- > AUX Pos4



Segmento 15: -- -> AUX Brazo Previa



Segmento 16: PARADA DE EMERGENCIA

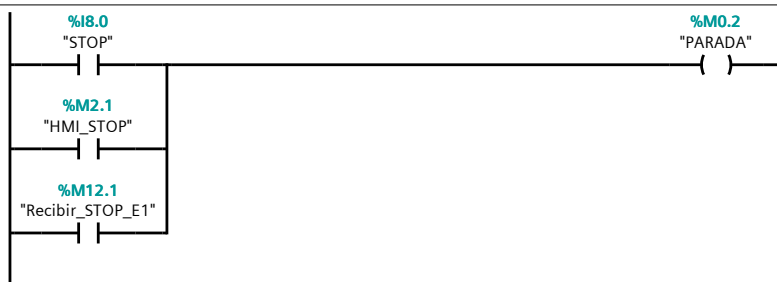










Tabla de variables estándar [73]

Variables PLC

| Variables PLC | | | | | | | | | |
|---------------|------------------|---------------|-----------|------------|-------------------------------------|--------------------------------------|----------------------------|-------------|------------|
| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/Web API | Escribible desde HMI/O PC UA/Web API | Visible en HMI Engineering | Supervisión | Comentario |
| | START | Bool | %I0.0 | False | True | True | True | | |
| | PistonExt | Bool | %I0.1 | False | True | True | True | | |
| | PistonRet | Bool | %I0.2 | False | True | True | True | | |
| | PiezaVentosa | Bool | %I0.3 | False | True | True | True | | |
| | BrazoPrevia | Bool | %I0.4 | False | True | True | True | | |
| | BrazoDestino | Bool | %I0.5 | False | True | True | True | | |
| | Piezalnicio | Bool | %I0.6 | False | True | True | True | | |
| | STOP | Bool | %I8.0 | False | True | True | True | | |
| | RetraerPiston | Bool | %Q0.0 | False | True | True | True | | |
| | SuccionPieza | Bool | %Q0.1 | False | True | True | True | | |
| | SoplarPieza | Bool | %Q0.2 | False | True | True | True | | |
| | MovBrazoPrev | Bool | %Q0.3 | False | True | True | True | | |
| | MovBrazoDest | Bool | %Q0.4 | False | True | True | True | | |
| | Contador+1 | Bool | %M0.4 | False | True | True | True | | |
| | AlmacenVacio | Bool | %M0.5 | False | True | True | True | | |
| | IDA | Bool | %M0.6 | False | True | True | True | | |
| | VUELTA | Bool | %M0.7 | False | True | True | True | | |
| | HMI_START | Bool | %M2.0 | False | True | True | True | | |
| | HMI_STOP | Bool | %M2.1 | False | True | True | True | | |
| | PrimerIDA | Bool | %M1.0 | False | True | True | True | | |
| | AUX_Destino | Bool | %M4.1 | False | True | True | True | | |
| | AUX_Pos2 | Bool | %M4.3 | False | True | True | True | | |
| | AUX_Pos3 | Bool | %M4.4 | False | True | True | True | | |
| | AUX_Panel | Bool | %M4.6 | False | True | True | True | | |
| | AUX_Previa | Bool | %M4.0 | False | True | True | True | | |
| | AUX_Pos1 | Bool | %M4.2 | False | True | True | True | | |
| | MODO_MANUAL | Bool | %M0.1 | False | True | True | True | | |
| | ActivarSuccion | Bool | %M1.1 | False | True | True | True | | |
| | ActivarSoplar | Bool | %M1.2 | False | True | True | True | | |
| | PARADA | Bool | %M0.2 | False | True | True | True | | |
| | HMI_MoverDestino | Bool | %M2.2 | False | True | True | True | | |
| | HMI_MoverPrevia | Bool | %M2.3 | False | True | True | True | | |
| | HMI_RetraerPist | Bool | %M2.4 | False | True | True | True | | |
| | HMI_ExtenderPist | Bool | %M2.5 | False | True | True | True | | |
| | HMI_Soplar | Bool | %M2.6 | False | True | True | True | | |

| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/Web API | Escribible desde HMI/O PC UA/Web API | Visible en HMI Engineering | Supervisión | Comentario |
|---|---------------------|---------------|-----------|------------|-------------------------------------|--------------------------------------|----------------------------|-------------|------------|
|  | HMI_Succion | Bool | %M2.7 | False | True | True | True | | |
|  | Recibir_START_E1 | Bool | %M12.0 | False | True | True | True | | |
|  | Recibir_STOP_E1 | Bool | %M12.1 | False | True | True | True | | |
|  | CicloActivo | Bool | %M0.0 | False | True | True | True | | |
|  | AUX_Pos4 | Bool | %M4.5 | False | True | True | True | | |
|  | COM_Off | Bool | %M0.3 | False | True | True | True | | |
|  | Enviar_PREPARADO_E1 | Bool | %M10.0 | False | True | True | True | | |
|  | Recibir_VENIR_E1 | Bool | %M10.1 | False | True | True | True | | |

ANEXO 2.
CÓDIGO DE LA
ESTACIÓN 1
(S7-1500)

Startup [OB100]

Startup Propiedades

General

| | | | | | |
|---------------|---------|-------------------|------------|-------------|----|
| Nombre | Startup | Número | 100 | Tipo | OB |
| Idioma | AWL | Numeración | Automático | | |

Información

| | | | | | |
|----------------|--------------------|----------------|-----|-------------------------------|--|
| Título | "Complete Restart" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personali- zado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|---------------|---------------|---------------|---------------------------------|
| ▼ Input | | | |
| LostRetentive | Bool | | True if retentive data are lost |
| LostRTC | Bool | | True if date and time are lost |
| Temp | | | |
| Constant | | | |

Segmento 1: QUE ESTÉ EN FC2

```

0001      A      "FinalCarr2"
0002      S      "BajarPlat"
0003      R      "SubirPlat"
0004      R      "ExtPiston"
0005      R      "SensorMedida"
0006      R      "PistonSujec"
0007      S      "AUX_Final"
0008      R      "CicloActivo"
0009      S      "AUX_Panel"
    
```

Segmento 2: QUE ESTÉ EN EL MEDIO

```

0001      AN     "FinalCarr1"
0002      AN     "FinalCarr2"
0003      S      "BajarPlat"
0004      R      "SubirPlat"
0005      R      "ExtPiston"
0006      R      "SensorMedida"
0007      R      "PistonSujec"
0008      S      "AUX_Medio"
0009      S      "AUX_Panel"
0010      R      "CicloActivo"
    
```

Segmento 3: QUE ESTÉ EN FC1

```

0001      A      "FinalCarr1"
0002      R      "BajarPlat"
0003      R      "SubirPlat"
0004      R      "ExtPiston"
0005      R      "SensorMedida"
0006      R      "PistonSujec"
0007      S      "AUX_Inicial"
0008      S      "AUX_Panel"
    
```

| | | |
|---|--|--|
| Totally Integrated Automation Portal | | |
|---|--|--|

| | | |
|------|---|---------------|
| 0009 | R | "AUX_Final" |
| 0010 | R | "AUX_Medio" |
| 0011 | R | "CicloActivo" |

Main [OB1]

Main Propiedades

General

| | | | | | |
|---------------|------|-------------------|------------|-------------|----|
| Nombre | Main | Número | 1 | Tipo | OB |
| Idioma | AWL | Numeración | Automático | | |

Información

| | | | | | |
|----------------|---------------------------------|----------------|-----|-------------------------------|--|
| Título | "Main Program Sweep (Cycle)" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personal- izado | |

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

Segmento 1: Inicio del Ciclo

AUTOR: JAVIER RODRÍGUEZ DE LA ROSA - - - TFG - - - 2021

```

0001      O      "HMI_START"
0002      FP     "DB_Flancos".Flanco[0]
0003      O      "START"
0004      FP     "DB_Flancos".Flanco[1]
0005      O      "Interruptor_START"
0006      FP     "DB_Flancos".Flanco[2]
0007      S      "CicloActivo"      //-----Set
0008      R      "AUX_Panel"
0009
0010      O      "PARADA"
0011      FP     "DB_Flancos".Flanco[3]
0012      O      "MODO_MANUAL"
0013      FP     "DB_Flancos".Flanco[4]
0014      R      "CicloActivo"      //-----Reset
    
```

Segmento 2: COMUNICACION CON LA ESTACION 0

```

0001      O (
0002      A      "CicloActivo"
0003      AN     "Orden_VENIR_E0"
0004      AN     "CicloCerrado"
0005      A      "Leer_PREPARADO_E0"
0006      FP     "DB_Flancos".Flanco[5]
0007      A      "FinalCarr1"
0008      AN     "COM_Off"
0009      )
0010      S      "CicloCerrado"
0011      S      "Orden_VENIR_E0"
0012
0013      O      "Leer_NuevoCiclo_E4"
0014      FP     "DB_Flancos".Flanco[6]
0015      O      "Leer_NuevoCiclo_E3"
0016      FP     "DB_Flancos".Flanco[7]
    
```



```

0017      O      "Temp_ExpulsarMala"
0018      FP     "DB_Flancos".Flanco[8]
0019      O      "PARADA"
0020      FP     "DB_Flancos".Flanco[9]
0021      R      "CicloCerrado"
0022
0023      A      "Orden_VENIR_E0"
0024      FP     "DB_Flancos".Flanco[10]
0025      L      s5t#4s
0026      SS     "Temp_Espera"
0027
0028      A      "Temp_Espera"
0029      S      "Listo_SUBIR"
0030
0031      A      "FinalCarr2"
0032      R      "Temp_Espera"
0033      R      "Listo_SUBIR"
0034      R      "Orden_VENIR_E0"

```

Segmento 3: Temporizador para Subir

```

0001      O(
0002      A      "COM_Off"
0003      A      "ObjPos1"
0004      FP     "DB_Flancos".Flanco[11]
0005      )
0006      O(
0007      AN     "COM_Off"
0008      A      "Listo_SUBIR"
0009      FP     "DB_Flancos".Flanco[12]
0010      )
0011      A      "CicloActivo"
0012      A      "FinalCarr1"
0013      S      "ActivarT1"          //----Set
0014
0015      A      "ActivarT1"
0016      L      s5t#5s
0017      SS     "Temp_SubirPlat"
0018
0019      O      "FinalCarr2"
0020      O      "PARADA"
0021      R      "Temp_SubirPlat"
0022      R      "ActivarT1"          //----Reset

```

Segmento 4: Subir Plataforma (Q0.1)

```

0001      O(
0002      A      "CicloActivo"
0003      A      "Temp_SubirPlat"
0004      FP     "DB_Flancos".Flanco[13]
0005      )
0006      O(
0007      A      "MODO_MANUAL"
0008      A      "HMI_SubirPlat"
0009      FP     "DB_Flancos".Flanco[14]
0010      )
0011      S      "SubirPlat"          //----Set

```

```

0012
0013      A      "BajarPlat"
0014      FP     "DB_Flancos".Flanco[15]
0015      R      "SubirPlat"          //----Reset
    
```

Segmento 5: Temporizador para Medir

```

0001      AN     "ExtPiston"
0002      A      "CicloActivo"
0003      A      "FinalCarr2"
0004      FP     "DB_Flancos".Flanco[16]
0005      S      "ActivarT2"          //----Set
0006
0007      A      "ActivarT2"
0008      L      s5t#8s
0009      SS     "Temp_Medir"
0010
0011      O      "FinalCarr1"
0012      O      "PARADA"
0013      R      "Temp_Medir"
0014      R      "ActivarT2"          //----Reset
    
```

Segmento 6: Sensor de Medida (Q0.3)

```

0001      O(
0002      A      "CicloActivo"
0003      AN     "ExtPiston"
0004      AN     "Temp_Medir"
0005      A      "FinalCarr2"
0006      FP     "DB_Flancos".Flanco[17]
0007      )
0008      O(
0009      A      "MODO_MANUAL"
0010      A      "HMI_Medir"
0011      FP     "DB_Flancos".Flanco[18]
0012      )
0013      S      "SensorMedida"        //----Set
0014
0015      O(
0016      A      "Temp_Medir"
0017      FP     "DB_Flancos".Flanco[19]
0018      )
0019      O(
0020      A      "MODO_MANUAL"
0021      A      "HMI_Medir"
0022      FN     "DB_Flancos".Flanco[20]
0023      )
0024      O(
0025      A      "PARADA"
0026      FP     "DB_Flancos".Flanco[21]
0027      )
0028      R      "SensorMedida"        //----Reset
    
```

Segmento 7: Criterio de medida

```

0001      A      "Temp_Medir"
    
```

```

0002      FP      "DB_Flancos".Flanco[22]
0003      A(
0004      L        "leer"
0005      L        21900
0006      >=I
0007      )
0008      A(
0009      L        "leer"
0010      L        23100
0011      <=I
0012      )
0013      S        "PiezaBuena"      //----Set
0014
0015      A        "FinalCarr1"
0016      FP      "DB_Flancos".Flanco[23]
0017      R        "PiezaBuena"      //----Reset

```

Segmento 8: Temporizador para subir el sensor de medida

```

0001      A        "Temp_Medir"
0002      S        "ActivarT3"      //----Set
0003
0004      A        "ActivarT3"
0005      L        s5t#1.5s
0006      SS       "Temp_PausaFC2"
0007
0008      O        "BajarPlat"
0009      O        "PARADA"
0010      R        "Temp_PausaFC2"
0011      R        "ActivarT3"      //----Reset

```

Segmento 9: Extender Piston (Q0.2)

```

0001      O(
0002      A        "CicloActivo"
0003      A        "PiezaBuena"
0004      A        "Temp_PausaFC2"
0005      FP      "DB_Flancos".Flanco[24]
0006      )
0007      O(
0008      A        "CicloActivo"
0009      AN       "PiezaBuena"
0010      A        "BajarDefectuosa"
0011      A        "FinalCarr1"
0012      FP      "DB_Flancos".Flanco[25]
0013      )
0014      O(
0015      A        "MODO_MANUAL"
0016      A        "HMI_Expulsar"
0017      FP      "DB_Flancos".Flanco[26]
0018      )
0019      S        "ExtPiston"      //----Set
0020
0021      O(
0022      AN       "MODO_MANUAL"
0023      A        "PistonExtendido"
0024      FP      "DB_Flancos".Flanco[27]

```

```

0025     )
0026     O(
0027     A     "Temp_ExpulsarMala"
0028     FP    "DB_Flancos".Flanco[28]
0029     )
0030     O(
0031     A     "PARADA"
0032     FP    "DB_Flancos".Flanco[29]
0033     )
0034     O(
0035     A     "MODO_MANUAL"
0036     A     "HMI_Expulsar"
0037     FN    "DB_Flancos".Flanco[30]
0038     )
0039     R     "ExtPiston"           //----Reset
    
```

Segmento 10: Piston de Sujecion (Q0.4)

```

0001     O(
0002     A     "CicloActivo"
0003     A     "PiezaBuena"
0004     A     "ExtPiston"
0005     FP    "DB_Flancos".Flanco[31]
0006     )
0007     O(
0008     A     "MODO_MANUAL"
0009     A     "HMI_SujetarPieza"
0010     FP    "DB_Flancos".Flanco[32]
0011     )
0012     S     "PistonSujec"       //----Set
0013
0014     O(
0015     AN    "COM_Off"
0016     A     "Leer_LISTO_E2"
0017     A     "FinalCarr1"
0018     FP    "DB_Flancos".Flanco[33]
0019     )
0020     O(
0021     A     "COM_Off"
0022     A     "FinalCarr1"
0023     FP    "DB_Flancos".Flanco[34]
0024     )
0025     O(
0026     A     "MODO_MANUAL"
0027     A     "HMI_SujetarPieza"
0028     FN    "DB_Flancos".Flanco[35]
0029     )
0030     O(
0031     A     "PARADA"
0032     FP    "DB_Flancos".Flanco[36]
0033     )
0034     R     "PistonSujec"       //----Reset
    
```

Segmento 11: COMUNICACION CON LA ESTACION 2

```

0001     A     "CicloActivo"
0002     AN    "COM_Off"
    
```

| | | | |
|---|----|---------------------------------|--|
| Totally Integrated Automation Portal | | | |
| 0003 | A | "PistonSujec" | |
| 0004 | FN | "DB_Flancos".Flanco[37] | |
| 0005 | S | "Orden_COMENZAR_E2" //----Set | |
| 0006 | | | |
| 0007 | A | "CicloActivo" | |
| 0008 | A | "SubirPlat" | |
| 0009 | FP | "DB_Flancos".Flanco[38] | |
| 0010 | R | "Orden_COMENZAR_E2" //----Reset | |
| Segmento 12: Temporizador para expulsar pieza mala | | | |
| 0001 | A | "CicloActivo" | |
| 0002 | AN | "PiezaBuena" | |
| 0003 | A | "BajarDefectuosa" | |
| 0004 | A | "FinalCarr1" | |
| 0005 | FP | "DB_Flancos".Flanco[39] | |
| 0006 | S | "ActivarT4" //----Set | |
| 0007 | | | |
| 0008 | | | |
| 0009 | A | "ActivarT4" | |
| 0010 | L | s5t#2s | |
| 0011 | SS | "Temp_ExpulsarMala" | |
| 0012 | | | |
| 0013 | O | "SubirPlat" | |
| 0014 | O | "PARADA" | |
| 0015 | R | "Temp_ExpulsarMala" | |
| 0016 | R | "ActivarT4" //----Reset | |
| Segmento 13: Bajar Plataforma (Q0.0) | | | |
| 0001 | O(| | |
| 0002 | A | "CicloActivo" | |
| 0003 | AN | "PiezaBuena" | |
| 0004 | A | "Temp_PausaFC2" | |
| 0005 | FP | "DB_Flancos".Flanco[40] | |
| 0006 |) | | |
| 0007 | O(| | |
| 0008 | A | "CicloActivo" | |
| 0009 | A | "PiezaBuena" | |
| 0010 | A | "ExtPiston" | |
| 0011 | FN | "DB_Flancos".Flanco[41] | |
| 0012 |) | | |
| 0013 | O(| | |
| 0014 | A | "MODO_MANUAL" | |
| 0015 | A | "HMI_BajarPlat" | |
| 0016 | FP | "DB_Flancos".Flanco[42] | |
| 0017 |) | | |
| 0018 | O(| | |
| 0019 | AN | "FinalCarr1" | |
| 0020 | A | "PARADA" | |
| 0021 | FP | "DB_Flancos".Flanco[43] | |
| 0022 |) | | |
| 0023 | S | "BajarPlat" //----Set | |
| 0024 | | | |
| 0025 | O(| | |
| 0026 | A | "FinalCarr1" | |
| 0027 | FP | "DB_Flancos".Flanco[44] | |

```

0028      )
0029      O(
0030      A      "PARADA"
0031      FN     "DB_Flancos".Flanco[45]
0032      )
0033      R      "BajarPlat"      //----Reset
    
```

Segmento 14: BajarDefectuosa (Necesaria para evitar conflicto)

```

0001      AN     "PiezaBuena"
0002      A      "Temp_PausaFC2"
0003      FP     "DB_Flancos".Flanco[46]
0004      S      "BajarDefectuosa" //----Set
0005
0006      O(
0007      A      "CicloActivo"
0008      A      "COM_Off"
0009      A      "FinalCarr1"
0010      A      "ObjPos1"
0011      FP     "DB_Flancos".Flanco[47]
0012      )
0013      O(
0014      A      "CicloActivo"
0015      AN     "COM_Off"
0016      A      "FinalCarr1"
0017      A      "Listo_SUBIR"
0018      FP     "DB_Flancos".Flanco[48]
0019      )
0020      O(
0021      A      "PARADA"
0022      FP     "DB_Flancos".Flanco[49]
0023      )
0024      R      "BajarDefectuosa" //----Reset
    
```

Segmento 15: MOVER PIEZA C1 + C2

```

0001      A      "CicloActivo"
0002      A      "PistonSujec"
0003      FN     "DB_Flancos".Flanco[50]
0004      S      "C1_MoverPieza"
0005
0006      A      "CicloActivo"
0007      AN     "PiezaBuena"
0008      A      "BajarDefectuosa"
0009      A      "FinalCarr1"
0010      FP     "DB_Flancos".Flanco[51]
0011      S      "C2_MoverPieza"
    
```

Segmento 16: ANIMACIÓN DE PIEZA RAMPA 1

```

0001      A      "C1_MoverPieza"
0002      A      "Clock_5Hz"
0003      =      "C1_Activar"
0004
0005      CALL   CTU , "MoverPiezaBuena"
0006      Int
    
```

```

0007      CU := "C1_Activar"
0008      R  := "C1_Reset"
0009      PV := 15
0010      Q  := "C1_Increm"
0011      CV :=
0012
0013      A    "Temp_Medir"
0014      =    "C1_Reset"
0015
0016      A    "C1_Reset"
0017      R    "C1_MoverPieza"
0018
0019      CALL CTU , "CuentaPiezasBuenas"
0020      Int
0021      CU := "C1_Increm"
0022      R  := "AUX_Panel"
0023      PV := 10
0024      Q  :=
0025      CV :=

```

Segmento 17: ANIMACIÓN DE PIEZA RAMPA 2

```

0001      A    "C2_MoverPieza"
0002      A    "Clock_5Hz"
0003      =    "C2_Activar"
0004
0005
0006      CALL CTU , "MoverPiezaMala"
0007      Int
0008      CU := "C2_Activar"
0009      R  := "C2_Reset"
0010      PV := 15
0011      Q  := "C2_Increm"
0012      CV :=
0013
0014      A    "Temp_SubirPlat"
0015      =    "C2_Reset"
0016
0017      A    "C2_Reset"
0018      R    "C2_MoverPieza"
0019
0020
0021      CALL CTU , "CuentaPiezasMalas"
0022      Int
0023      CU := "C2_Increm"
0024      R  := "AUX_Panel"
0025      PV := 10
0026      Q  :=
0027      CV :=

```

Segmento 18: PARADA EMERGENCIA

```

0001      O    "STOP"
0002      O    "HMI_STOP"
0003      O    "Interruptor_STOP"
0004      =    "PARADA"

```

Segmento 19: COMUNICACION GLOBAL A LOS ESCLAVOS

```

0001      O      "Interruptor_START"
0002      FP     "DB_Flancos".Flanco[52]
0003      O      "HMI_START"
0004      FP     "DB_Flancos".Flanco[53]
0005      S      "Orden_START"      //----Set
0006
0007      O      "Interruptor_START"
0008      FN     "DB_Flancos".Flanco[54]
0009      O      "HMI_START"
0010      FN     "DB_Flancos".Flanco[55]
0011      R      "Orden_START"      //----Reset
0012
0013      O      "Interruptor_STOP"
0014      FP     "DB_Flancos".Flanco[56]
0015      O      "HMI_STOP"
0016      FP     "DB_Flancos".Flanco[57]
0017      S      "Orden_STOP"
0018
0019      O      "Interruptor_STOP"
0020      FN     "DB_Flancos".Flanco[58]
0021      O      "HMI_STOP"
0022      FN     "DB_Flancos".Flanco[59]
0023      R      "Orden_STOP"
0024
0025      A      "Int.PiezaMala"
0026      FP     "DB_Flancos".Flanco[60]
0027      S      "Orden_PIEZA_MALA_E2"
0028
0029      A      "Int.PiezaMala"
0030      FN     "DB_Flancos".Flanco[61]
0031      R      "Orden_PIEZA_MALA_E2"

```

Segmento 20: LLAMADA A LOS BLOQUES FC DE COMUNICACIONES

```

0001      CALL   "Conex1_ID100"
0002
0003      CALL   "Conex2_ID101"
0004
0005      CALL   "Conex3_ID102"
0006
0007      CALL   "Conex4_ID103"

```

Segmento 21: ---> AUX plataforma

```

0001      A      "FinalCarr1"
0002      =      "AUX_Inicial"
0003
0004      A      "FinalCarr2"
0005      =      "AUX_Final"
0006
0007      AN     "FinalCarr1"
0008      AN     "FinalCarr2"
0009      =      "AUX_Medio"

```


Segmento 22: --- > AUX expulsar pieza

```

0001      A      "ExtPiston"
0002      FP     "DB_Flancos".Flanco[62]
0003      A      "FinalCarr1"
0004      S      "AUX_Expulsar_FC1"
0005
0006      A      "ExtPiston"
0007      FN     "DB_Flancos".Flanco[63]
0008      A      "FinalCarr1"
0009      R      "AUX_Expulsar_FC1"
0010
0011      A      "ExtPiston"
0012      FP     "DB_Flancos".Flanco[64]
0013      A      "FinalCarr2"
0014      S      "AUX_Expulsar_FC2"
0015
0016      A      "ExtPiston"
0017      FN     "DB_Flancos".Flanco[65]
0018      A      "FinalCarr2"
0019      R      "AUX_Expulsar_FC2"
    
```

Segmento 23: --- > AUX_Pos1

```

0001      A      "CicloActivo"
0002      A      "ActivarT1"
0003      FP     "DB_Flancos".Flanco[66]
0004      S      "AUX_Pos1"           //----Set
0005
0006      A      "SubirPlat"
0007      A      "AUX_Pos4"
0008      FP     "DB_Flancos".Flanco[67]
0009      R      "AUX_Pos1"           //----Reset
    
```

Segmento 24: --- > AUX_Pos2

```

0001      A      "CicloActivo"
0002      A      "FinalCarr2"
0003      FP     "DB_Flancos".Flanco[68]
0004      S      "AUX_Pos2"           //----Set
0005
0006      O(
0007      A      "CicloActivo"
0008      A      "ExtPiston"
0009      FP     "DB_Flancos".Flanco[69]
0010      )
0011      O(
0012      A      "BajarPlat"
0013      FP     "DB_Flancos".Flanco[70]
0014      )
0015      R      "AUX_Pos2"           //----Reset
    
```

Segmento 25: --- > AUX_Pos3

```
0001      A      "CicloActivo"  
0002      A      "FinalCarr2"  
0003      A      "ExtPiston"  
0004      FP     "DB_Flancos".Flanco[71]  
0005      S      "AUX_Pos3"          //----Set  
0006  
0007      A      "PistonSujec"  
0008      FN     "DB_Flancos".Flanco[72]  
0009      R      "AUX_Pos3"          //----Reset
```

Segmento 26: --- > AUX_Pos4

```
0001      O(  
0002      AN     "PiezaBuena"  
0003      A      "BajarPlat"  
0004      )  
0005      O      "SubirPlat"  
0006      AN     "MODO_MANUAL"  
0007      A      "AUX_Medio"  
0008      =      "AUX_Pos4"
```

Conex1_ID100 [FC1]

Conex1_ID100 Propiedades

General

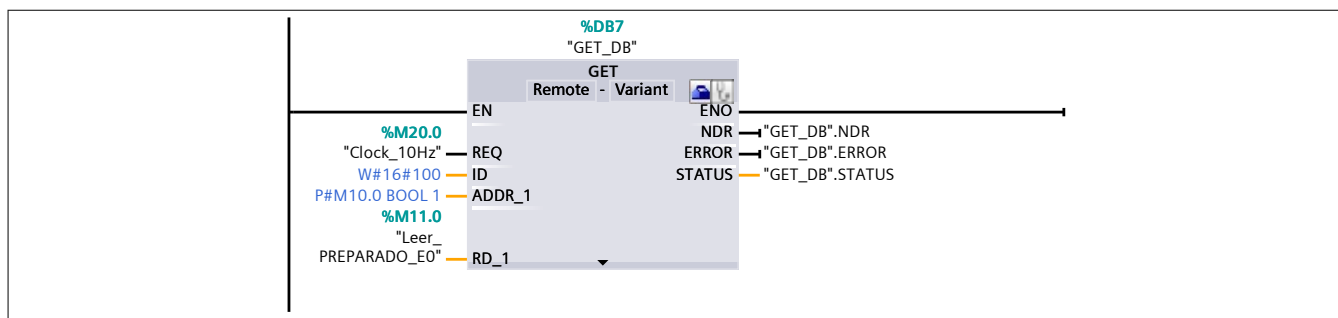
| | | | | | |
|---------------|--------------|-------------------|------------|-------------|----|
| Nombre | Conex1_ID100 | Número | 1 | Tipo | FC |
| Idioma | KOP | Numeración | Automático | | |

Información

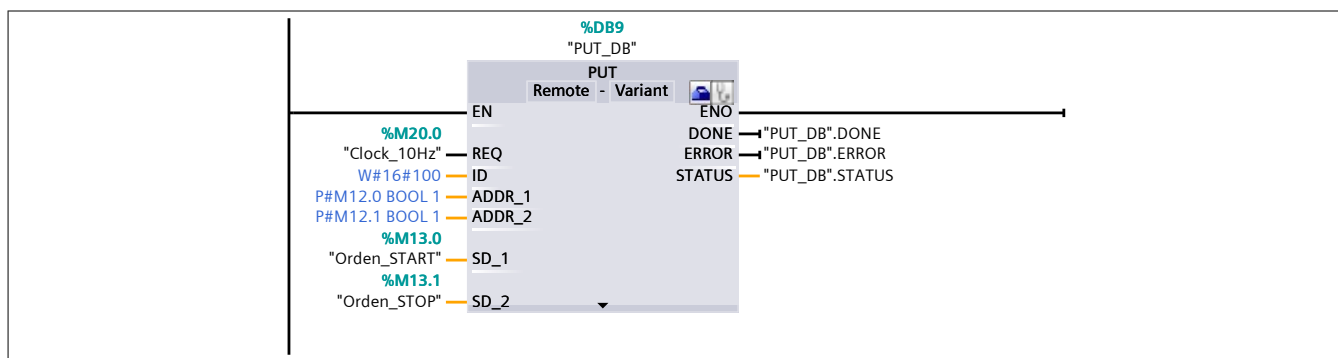
| | | | | | |
|----------------|--|----------------|-----|-------------------------|--|
| Título | | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|------------|
| Input | | | |
| Output | | | |
| InOut | | | |
| Temp | | | |
| Constant | | | |
| ▼ Return | | | |
| Conex1_ID100 | Void | | |

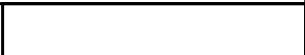
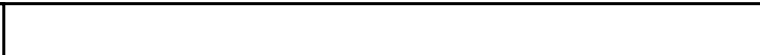
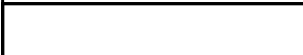
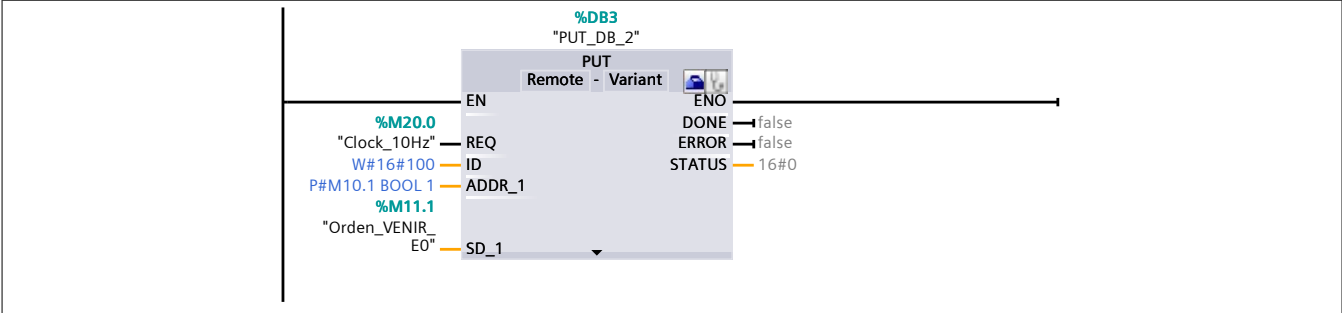
Segmento 1:



Segmento 2:



Segmento 3:



Conex2_ID101 [FC2]

Conex2_ID101 Propiedades

General

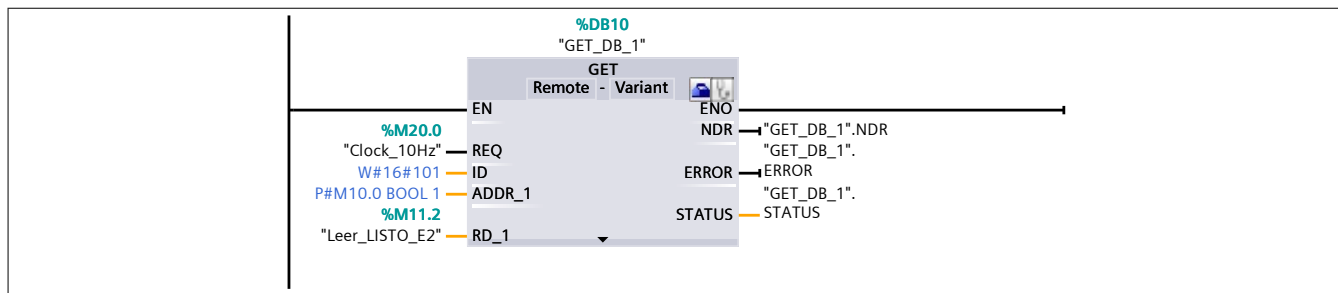
| | | | | | |
|---------------|--------------|-------------------|------------|-------------|----|
| Nombre | Conex2_ID101 | Número | 2 | Tipo | FC |
| Idioma | KOP | Numeración | Automático | | |

Información

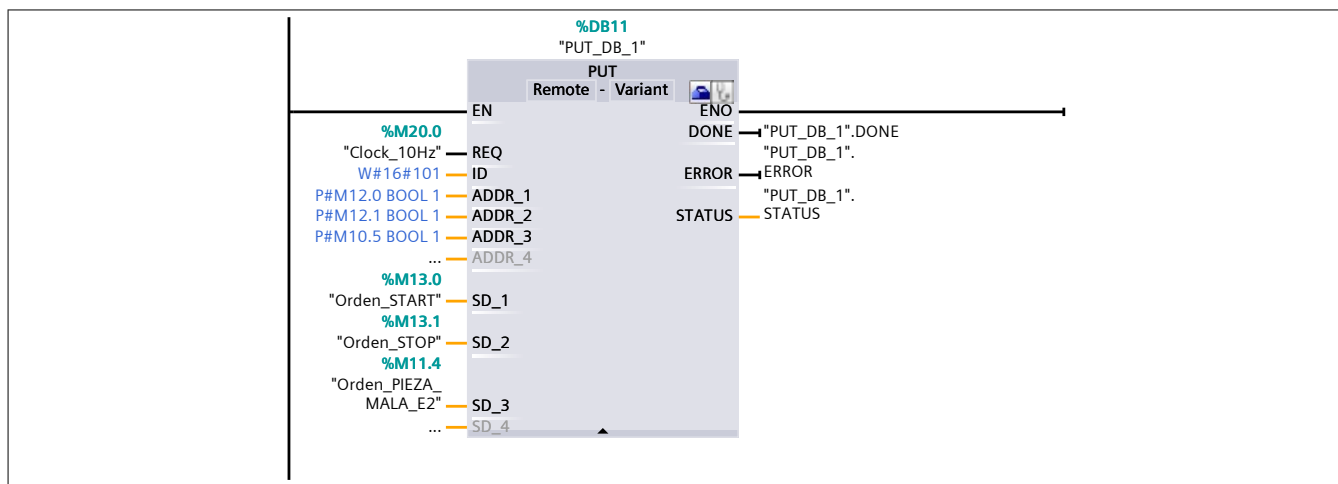
| | | | | | |
|----------------|--|----------------|-----|-------------------------|--|
| Título | | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|------------|
| Input | | | |
| Output | | | |
| InOut | | | |
| Temp | | | |
| Constant | | | |
| ▼ Return | | | |
| Conex2_ID101 | Void | | |

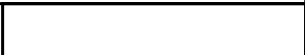
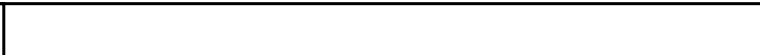
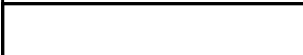
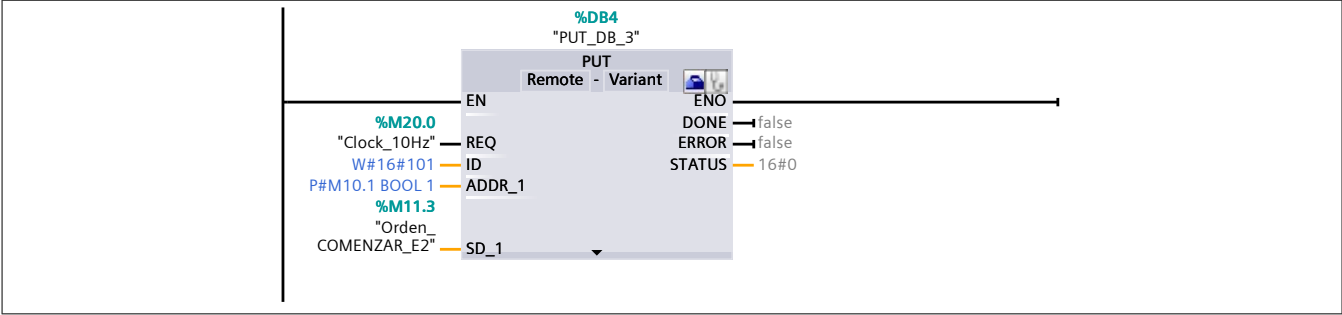
Segmento 1:



Segmento 2:



Segmento 3:



Conex3_ID102 [FC3]

Conex3_ID102 Propiedades

General

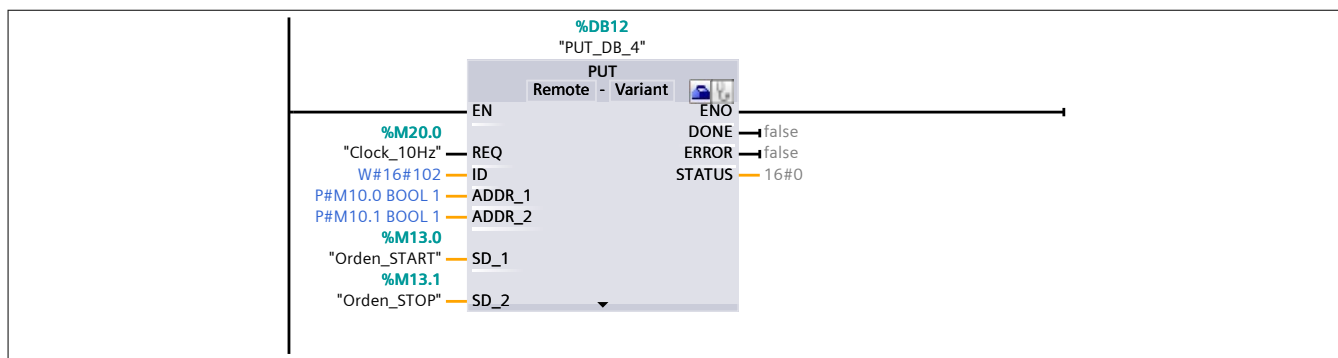
| | | | | | |
|---------------|--------------|-------------------|------------|-------------|----|
| Nombre | Conex3_ID102 | Número | 3 | Tipo | FC |
| Idioma | KOP | Numeración | Automático | | |

Información

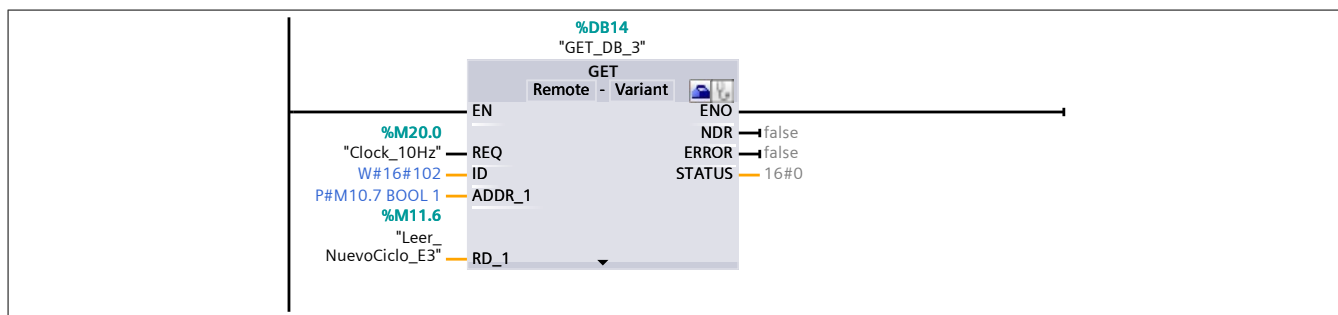
| | | | | | |
|----------------|--|----------------|-----|-------------------------|--|
| Título | | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|------------|
| Input | | | |
| Output | | | |
| InOut | | | |
| Temp | | | |
| Constant | | | |
| ▼ Return | | | |
| Conex3_ID102 | Void | | |

Segmento 1:



Segmento 2:



Conex4_ID103 [FC4]

Conex4_ID103 Propiedades

General

| | | | | | |
|---------------|--------------|-------------------|------------|-------------|----|
| Nombre | Conex4_ID103 | Número | 4 | Tipo | FC |
| Idioma | KOP | Numeración | Automático | | |

Información

| | | | | | |
|----------------|--|----------------|-----|-------------------------|--|
| Título | | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|------------|
| Input | | | |
| Output | | | |
| InOut | | | |
| Temp | | | |
| Constant | | | |
| ▼ Return | | | |
| Conex4_ID103 | Void | | |

Segmento 1:

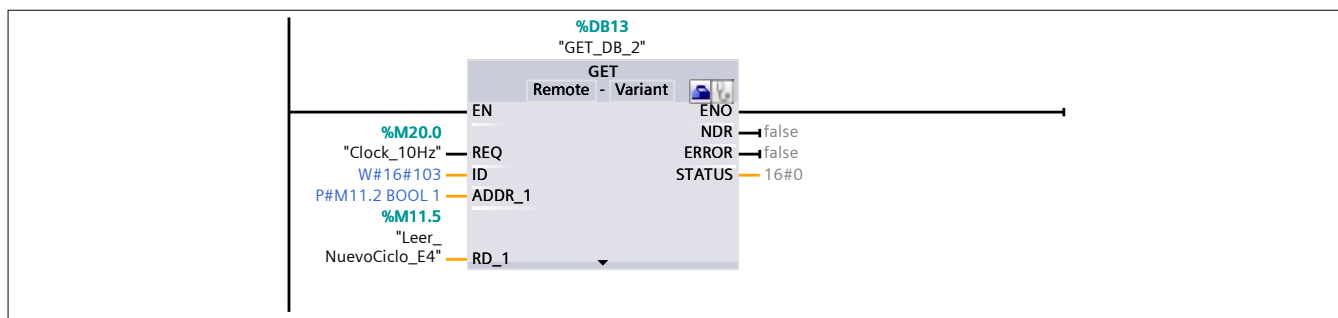










Tabla de variables estándar [135]

Variables PLC

| Variables PLC | | | | | | | | | |
|---------------|-------------------|---------------|-----------|------------|-------------------------------------|--------------------------------------|----------------------------|-------------|------------|
| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/Web API | Escribible desde HMI/O PC UA/Web API | Visible en HMI Engineering | Supervisión | Comentario |
| | PiezaMetal | Bool | %I0.0 | False | True | True | True | | |
| | ObjPos1 | Bool | %I0.1 | False | True | True | True | | |
| | DetectColor | Bool | %I0.2 | False | True | True | True | | |
| | FinalCarr1 | Bool | %I0.3 | False | True | True | True | | |
| | FinalCarr2 | Bool | %I0.4 | False | True | True | True | | |
| | PistonExtendido | Bool | %I0.5 | False | True | True | True | | |
| | SensorMidiendo | Bool | %I0.6 | False | True | True | True | | |
| | STOP | Bool | %I0.7 | False | True | True | True | | |
| | BajarPlat | Bool | %Q0.0 | False | True | True | True | | |
| | SubirPlat | Bool | %Q0.1 | False | True | True | True | | |
| | ExtPiston | Bool | %Q0.2 | False | True | True | True | | |
| | SensorMedida | Bool | %Q0.3 | False | True | True | True | | |
| | PistonSujec | Bool | %Q0.4 | False | True | True | True | | |
| | leer | Int | %IW6 | False | True | True | True | | |
| | PiezaBuena | Bool | %M0.4 | False | True | True | True | | |
| | Clock_Byte | Byte | %MB20 | False | True | True | True | | |
| | Clock_10Hz | Bool | %M20.0 | False | True | True | True | | |
| | Clock_5Hz | Bool | %M20.1 | False | True | True | True | | |
| | Clock_2.5Hz | Bool | %M20.2 | False | True | True | True | | |
| | Clock_2Hz | Bool | %M20.3 | False | True | True | True | | |
| | Clock_1.25Hz | Bool | %M20.4 | False | True | True | True | | |
| | Clock_1Hz | Bool | %M20.5 | False | True | True | True | | |
| | Clock_0.625Hz | Bool | %M20.6 | False | True | True | True | | |
| | Clock_0.5Hz | Bool | %M20.7 | False | True | True | True | | |
| | HMI_START | Bool | %M2.0 | False | True | True | True | | |
| | AUX_ObjPos1 | Bool | %M4.0 | False | True | True | True | | |
| | AUX_FC2 | Bool | %M4.2 | False | True | True | True | | |
| | AUX_Pos2 | Bool | %M4.4 | False | True | True | True | | |
| | AUX_Inicial | Bool | %M4.7 | False | True | True | True | | |
| | AUX_Pos1 | Bool | %M4.3 | False | True | True | True | | |
| | AUX_Final | Bool | %M5.0 | False | True | True | True | | |
| | ActivarT4 | Bool | %M1.0 | False | True | True | True | | |
| | Temp_ExpulsarMala | Timer | %T4 | False | True | True | True | | |
| | AUX_Pos3 | Bool | %M4.5 | False | True | True | True | | |
| | AUX_Medio | Bool | %M5.1 | False | True | True | True | | |

| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/W eb API | Escribible desde HMI/O PC UA/W eb API | Visible en HMI Engineering | Supervisión | Comentario |
|--|-------------------|---------------|-----------|------------|--------------------------------------|---------------------------------------|----------------------------|-------------|------------|
| | AUX_Panel | Bool | %M5.2 | False | True | True | True | | |
| | HMI_STOP | Bool | %M2.1 | False | True | True | True | | |
| | C1_Activar | Bool | %M6.0 | False | True | True | True | | |
| | C1_Reset | Bool | %M6.2 | False | True | True | True | | |
| | C1_Increm | Bool | %M6.3 | False | True | True | True | | |
| | C1_MoverPieza | Bool | %M6.1 | False | True | True | True | | |
| | START | Bool | %I1.0 | False | True | True | True | | |
| | C2_Activar | Bool | %M6.4 | False | True | True | True | | |
| | C2_Reset | Bool | %M6.6 | False | True | True | True | | |
| | C2_Increm | Bool | %M6.7 | False | True | True | True | | |
| | C2_MoverPieza | Bool | %M6.5 | False | True | True | True | | |
| | AUX_FC1 | Bool | %M4.1 | False | True | True | True | | |
| | Leer_PREPARADO_E0 | Bool | %M11.0 | False | True | True | True | | |
| | Orden_START | Bool | %M13.0 | False | True | True | True | | |
| | Interruptor_START | Bool | %I1.7 | False | True | True | True | | |
| | Interruptor_STOP | Bool | %I1.6 | False | True | True | True | | |
| | Orden_STOP | Bool | %M13.1 | False | True | True | True | | |
| | CicloActivo | Bool | %M0.0 | False | True | True | True | | |
| | MODO_MANUAL | Bool | %M0.1 | False | True | True | True | | |
| | PARADA | Bool | %M0.2 | False | True | True | True | | |
| | BajarDefectuosa | Bool | %M1.1 | False | True | True | True | | |
| | HMI_BajarPlat | Bool | %M2.2 | False | True | True | True | | |
| | HMI_SubirPlat | Bool | %M2.3 | False | True | True | True | | |
| | HMI_Medir | Bool | %M2.4 | False | True | True | True | | |
| | HMI_Expulsar | Bool | %M2.5 | False | True | True | True | | |
| | HMI_SujetarPieza | Bool | %M2.6 | False | True | True | True | | |
| | Orden_VENIR_E0 | Bool | %M11.1 | False | True | True | True | | |
| | AUX_Expulsar_FC1 | Bool | %M5.3 | False | True | True | True | | |
| | AUX_Expulsar_FC2 | Bool | %M5.4 | False | True | True | True | | |
| | Listo_SUBIR | Bool | %M11.7 | False | True | True | True | | |
| | COM_Off | Bool | %M0.3 | False | True | True | True | | |
| | Leer_LISTO_E2 | Bool | %M11.2 | False | True | True | True | | |
| | PruebaLatch | Bool | %I1.5 | False | True | True | True | | |
| | Orden_COMENZAR_E2 | Bool | %M11.3 | False | True | True | True | | |
| | Int.PiezaMala | Bool | %I1.4 | False | True | True | True | | |
| | ActivarT1 | Bool | %M0.5 | False | True | True | True | | |
| | Temp_SubirPlat | Timer | %T1 | False | True | True | True | | |
| | ActivarT2 | Bool | %M0.6 | False | True | True | True | | |
| | Temp_Medir | Timer | %T2 | False | True | True | True | | |

| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/Web API | Escribible desde HMI/O PC UA/Web API | Visible en HMI Engineering | Supervisión | Comentario |
|---|---------------------|---------------|-----------|------------|-------------------------------------|--------------------------------------|----------------------------|-------------|------------|
|  | Temp_PausaFC2 | Timer | %T3 | False | True | True | True | | |
|  | ActivarT3 | Bool | %M0.7 | False | True | True | True | | |
|  | Temp_Espera | Timer | %T6 | False | True | True | True | | |
|  | Leer_NuevoCiclo_E4 | Bool | %M11.5 | False | True | True | True | | |
|  | CicloCerrado | Bool | %M12.0 | False | True | True | True | | |
|  | AUX_Pos4 | Bool | %M4.6 | False | True | True | True | | |
|  | Orden_PIEZA_MALA_E2 | Bool | %M11.4 | False | True | True | True | | |
|  | Leer_NuevoCiclo_E3 | Bool | %M11.6 | False | True | True | True | | |

ANEXO 3.
CÓDIGO DE LA
ESTACIÓN 2
(S7-1200)

Startup [OB100]

Startup Propiedades

General

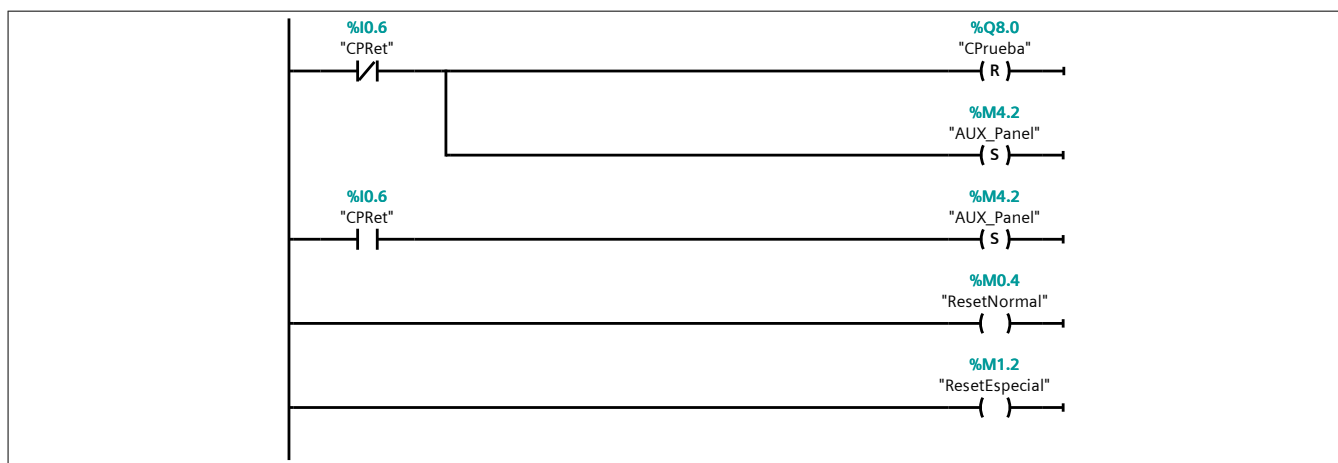
| | | | | | |
|---------------|---------|-------------------|------------|-------------|----|
| Nombre | Startup | Número | 100 | Tipo | OB |
| Idioma | KOP | Numeración | Automático | | |

Información

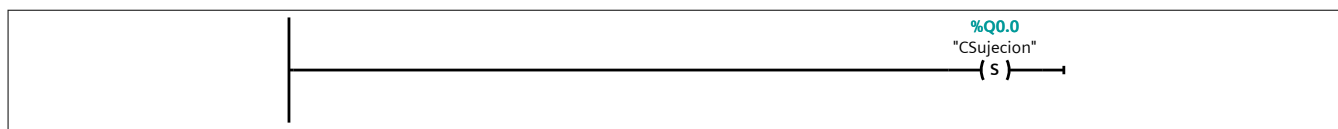
| | | | | | |
|----------------|--------------------|----------------|-----|-------------------------|--|
| Título | "Complete Restart" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|---------------|---------------|---------------|---------------------------------|
| ▼ Input | | | |
| LostRetentive | Bool | | True if retentive data are lost |
| LostRTC | Bool | | True if date and time are lost |
| Temp | | | |
| Constant | | | |

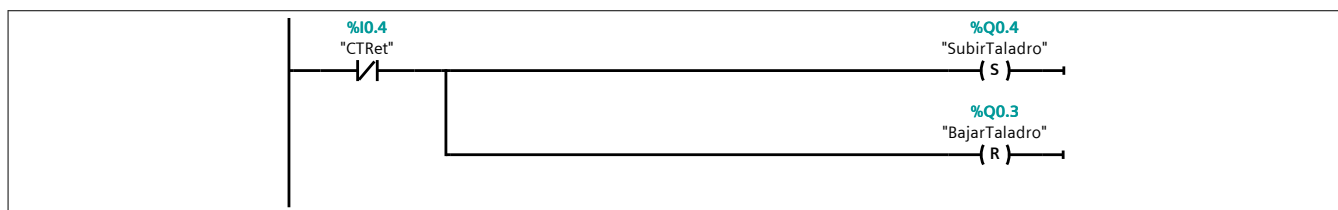
Segmento 1:



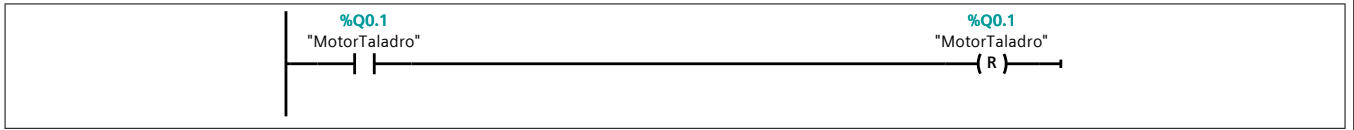
Segmento 2:



Segmento 3:



Segmento 4:



Main [OB1]

Main Propiedades

General

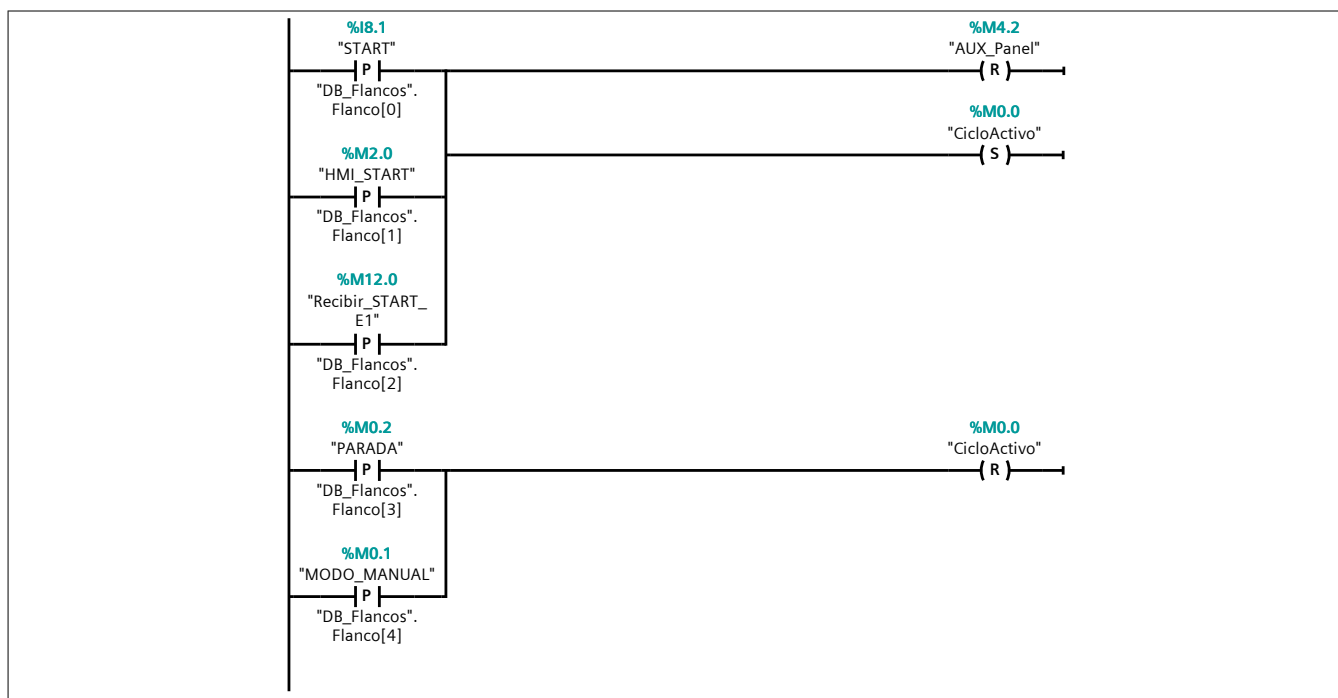
| | | | | | |
|---------------|------|-------------------|------------|-------------|----|
| Nombre | Main | Número | 1 | Tipo | OB |
| Idioma | KOP | Numeración | Automático | | |

Información

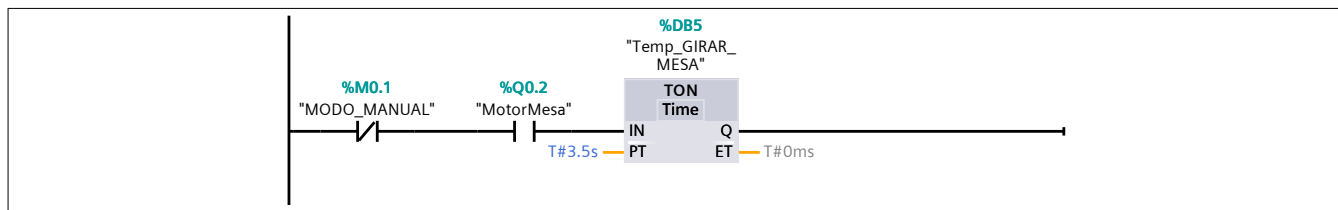
| | | | | | |
|----------------|------------------------------|----------------|-----|-------------------------|--|
| Título | "Main Program Sweep (Cycle)" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

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

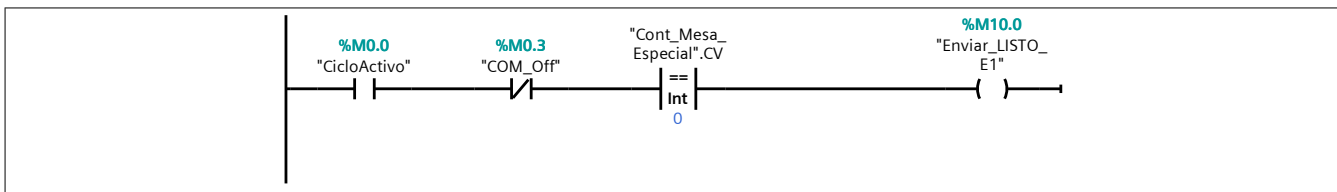
Segmento 1: INICIO



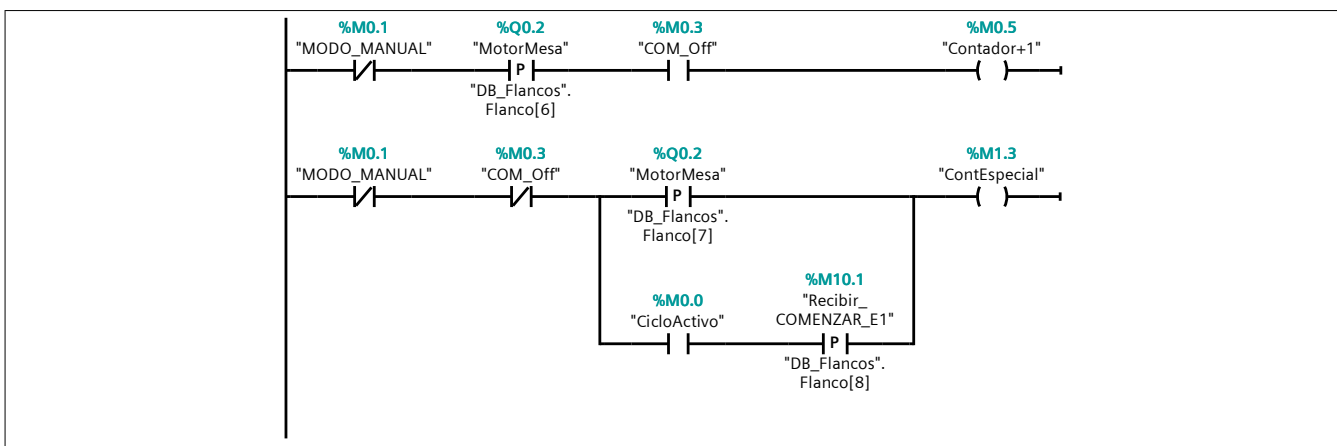
Segmento 2: MOTOR MESA TEMPORIZADOR



Segmento 3: COMUNICACIONES CON LA ESTACIÓN 1

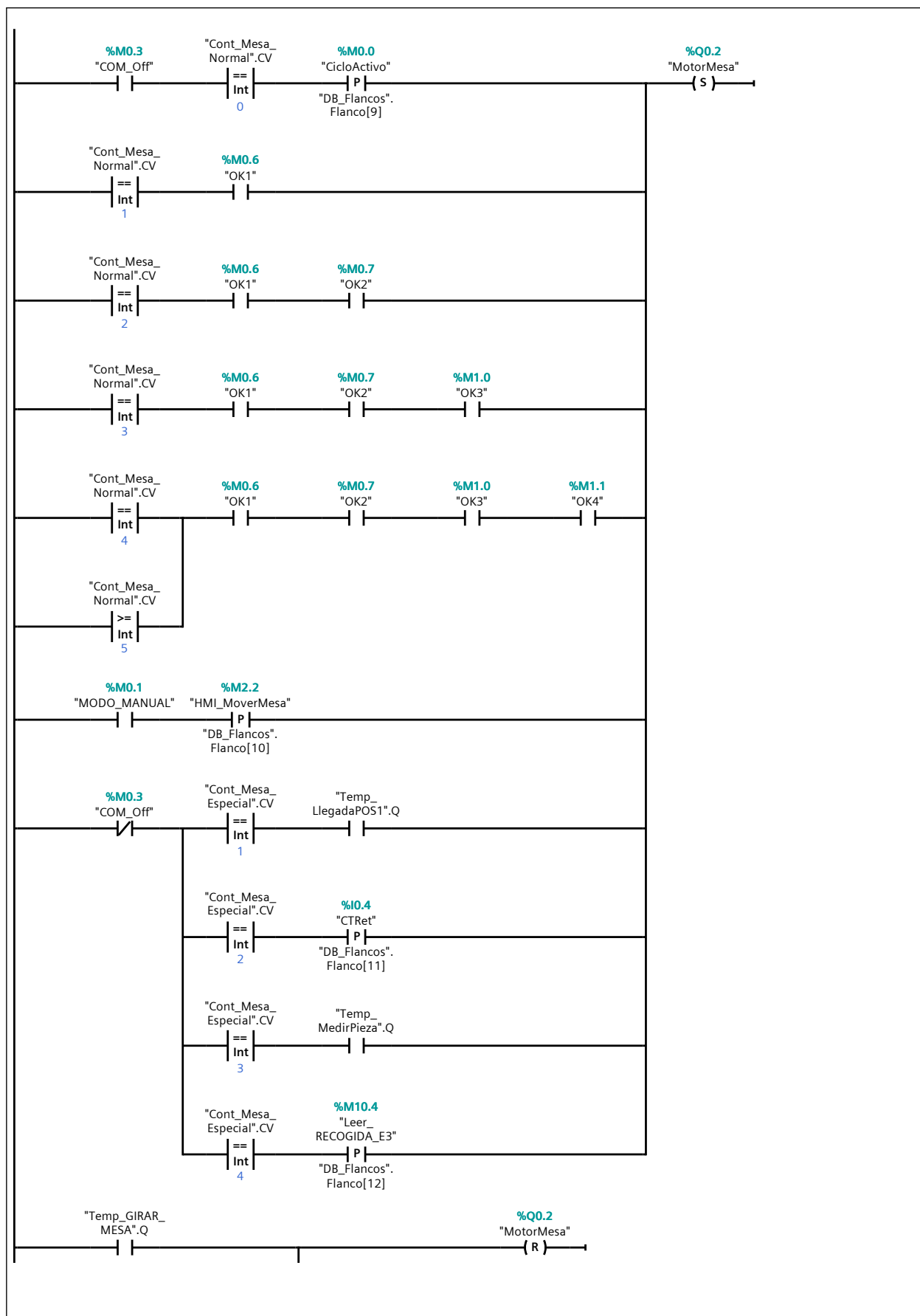


Segmento 4: --- > Criterio para aumentar el Contador



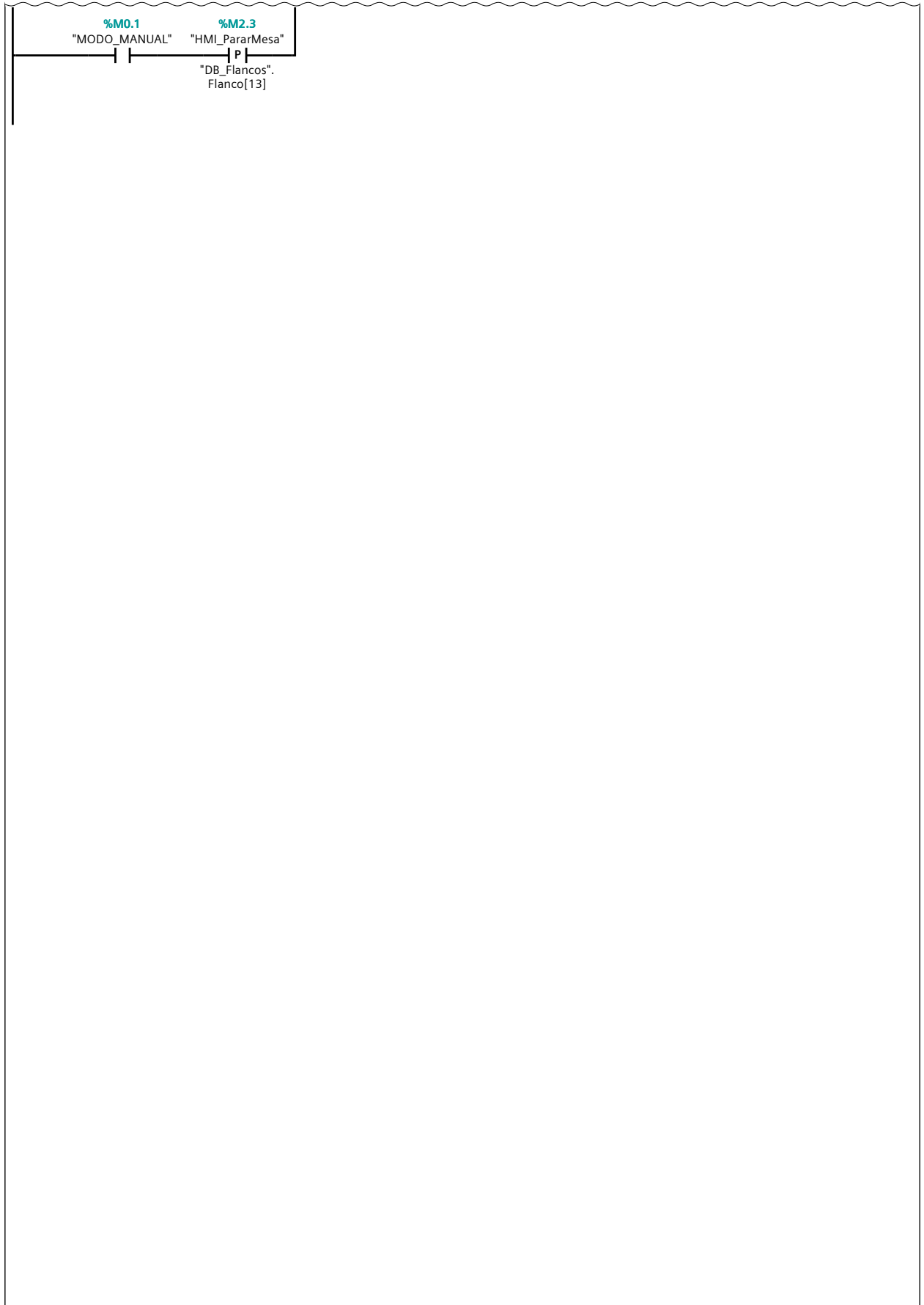
Segmento 5: --- > (Q0.2) MOTOR MESA

Segmento 5: ---> (Q0.2) MOTOR MESA (1.1 / 2.1)

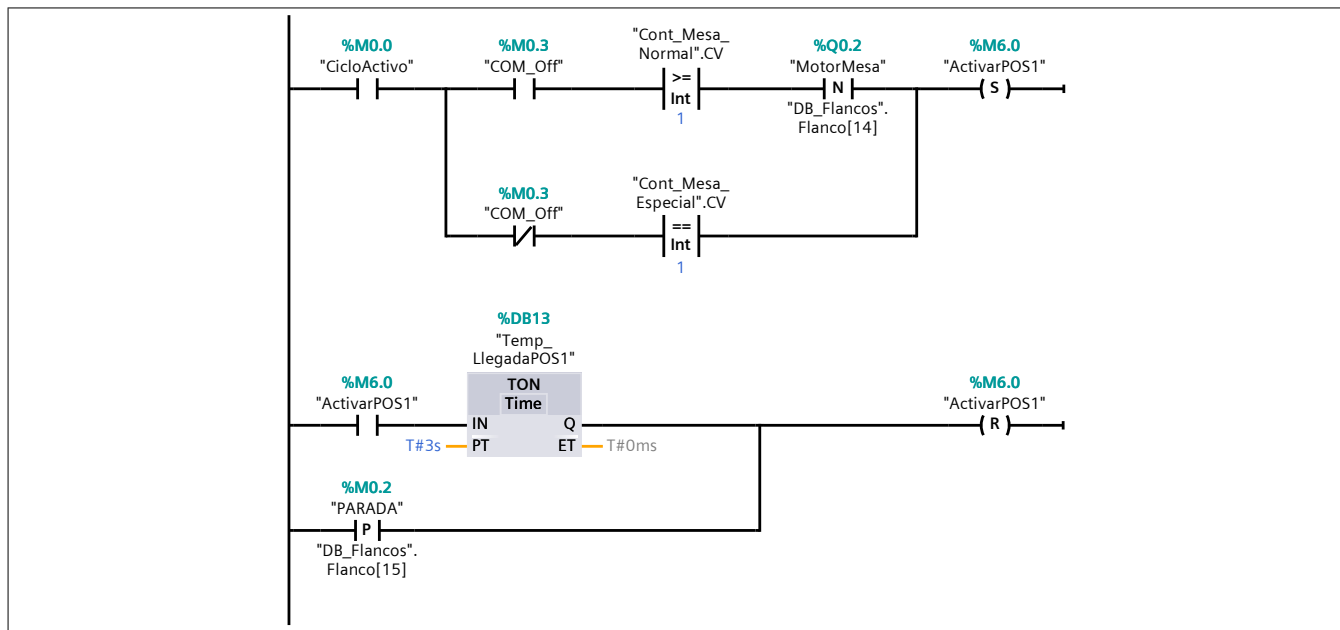


Segmento 5: --- > (Q0.2) MOTOR MESA (2.1 / 2.1)

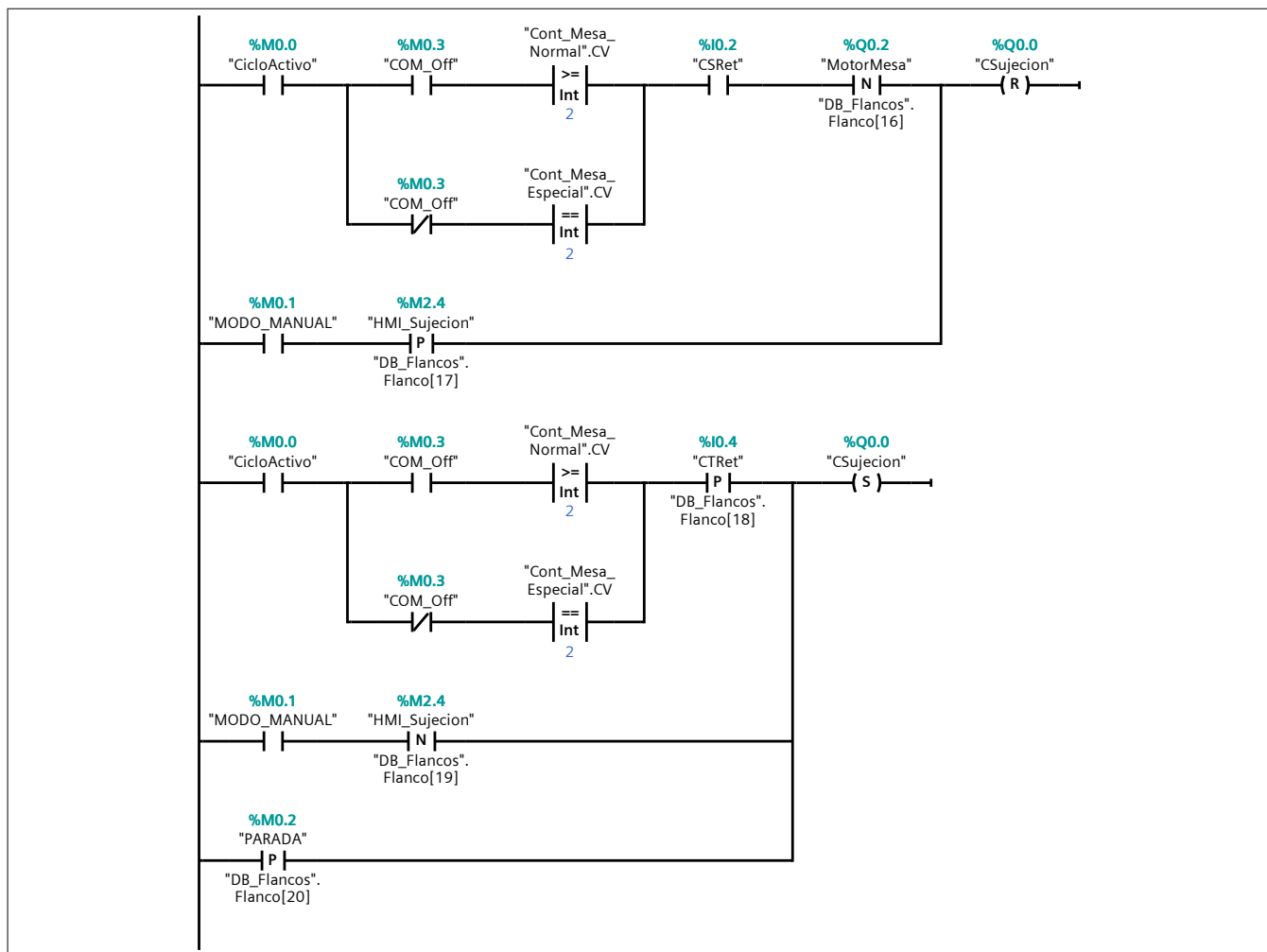
1.1 (Página1 - 3)



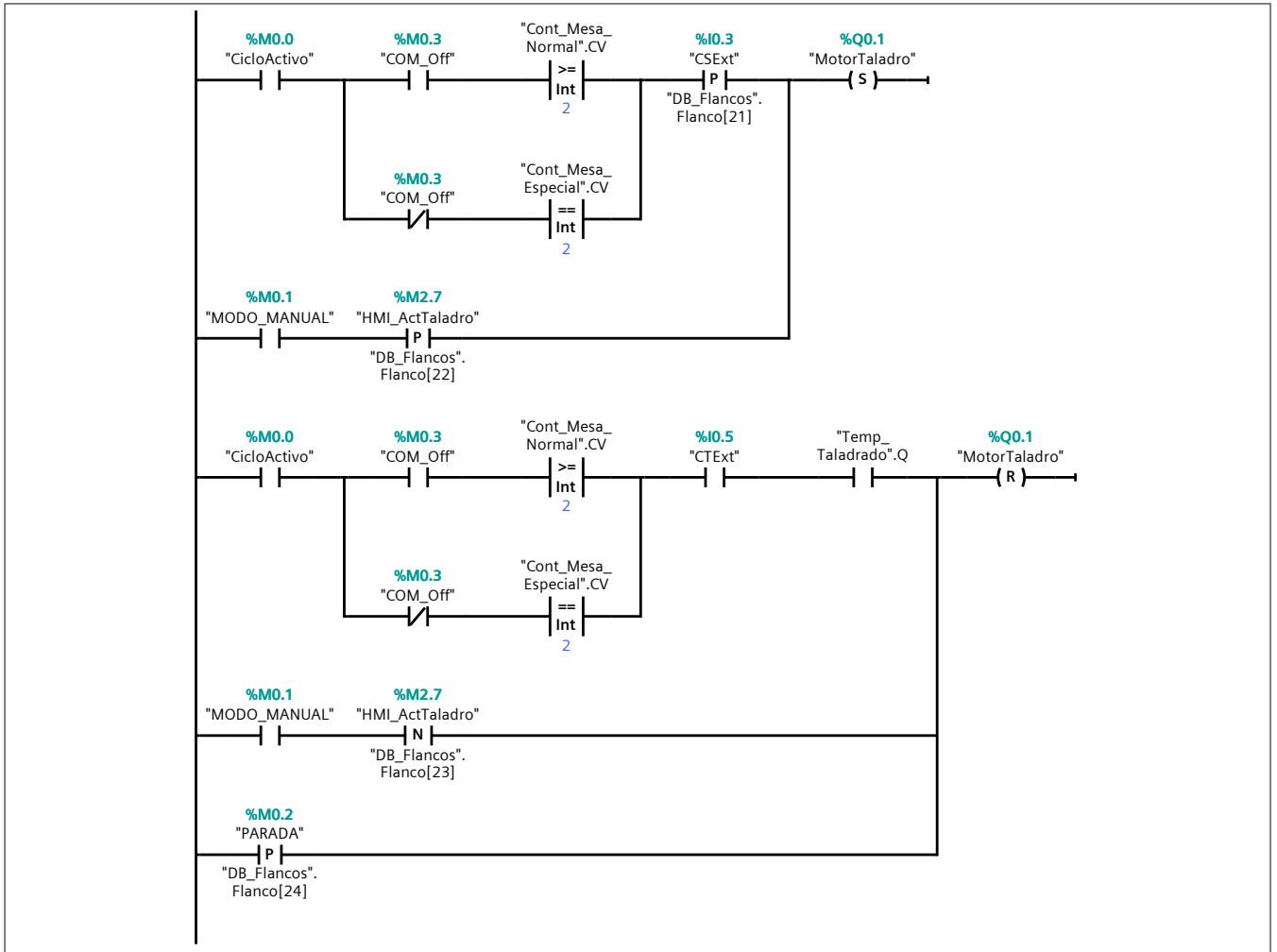
Segmento 6: POSICION 1



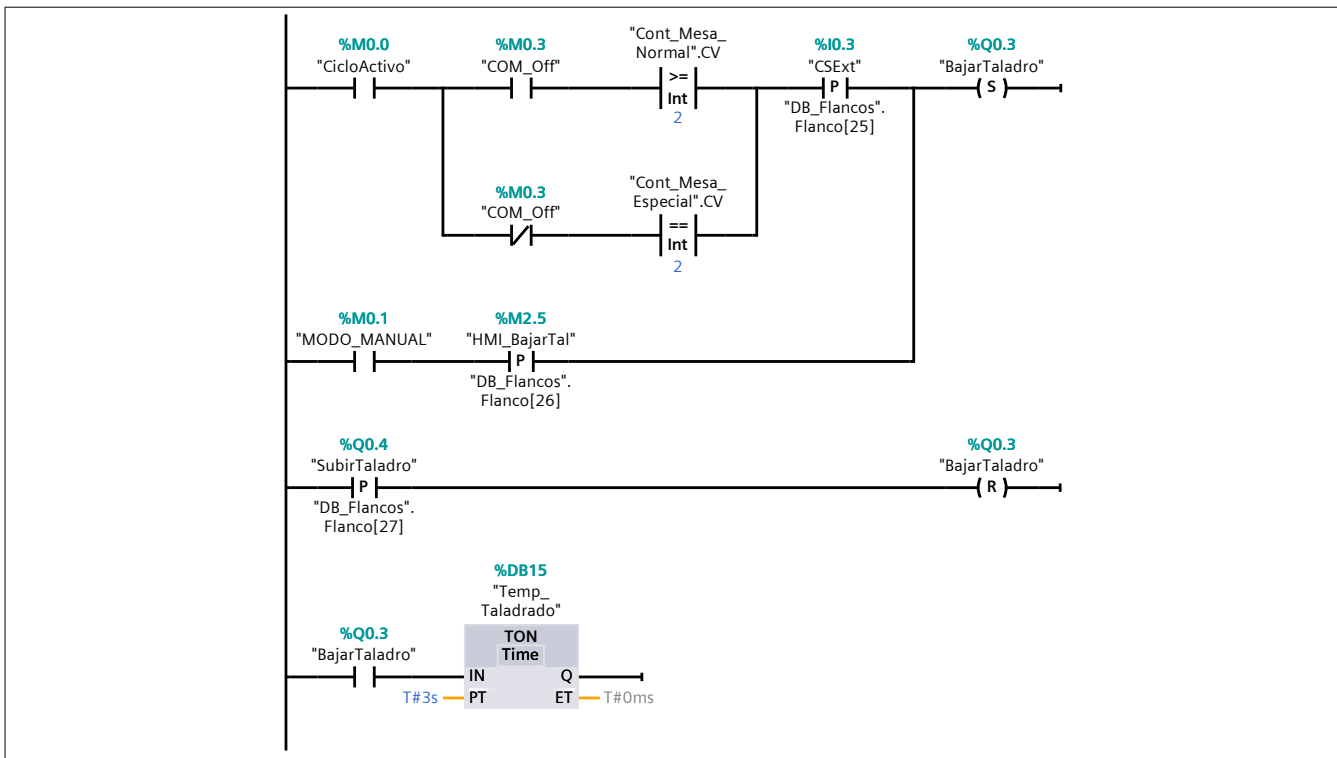
Segmento 7: POSICION 2 --> CILINDRO SUJECION



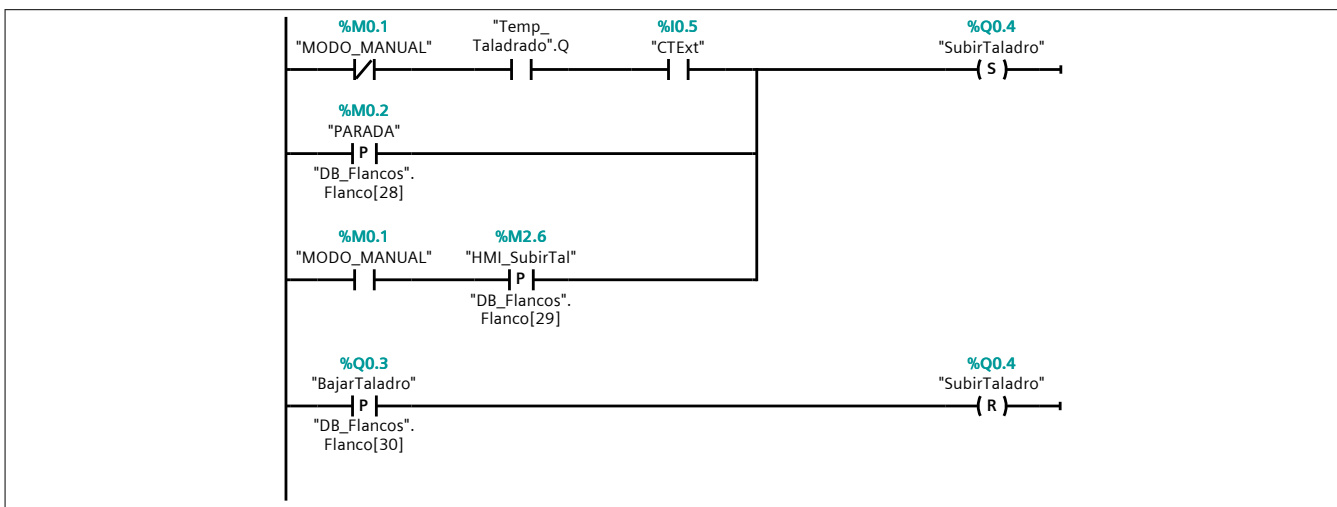
Segmento 8: POSICION 2 --> MOTOR TALADRO



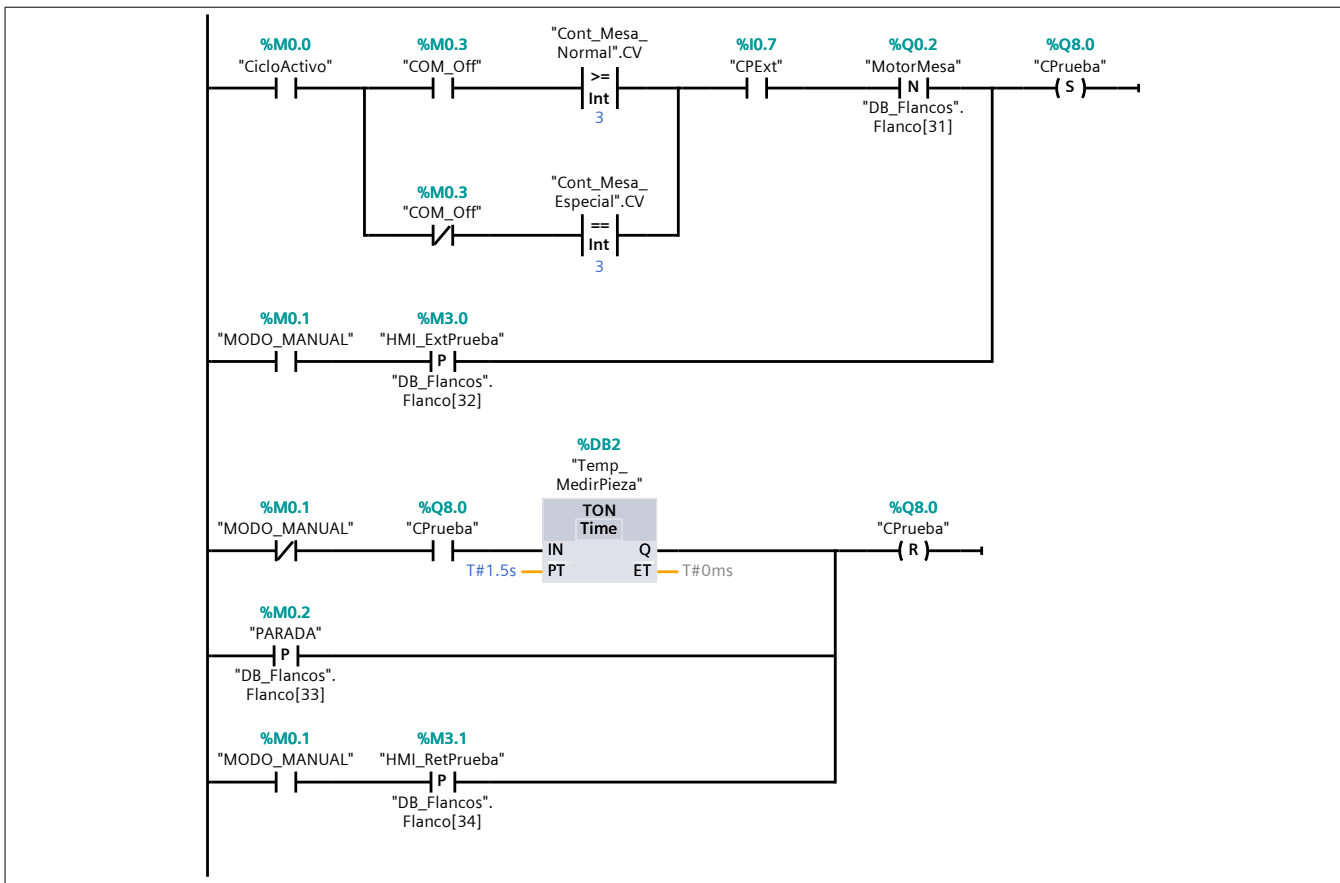
Segmento 9: POSICION 2 --> BAJAR TALADRO



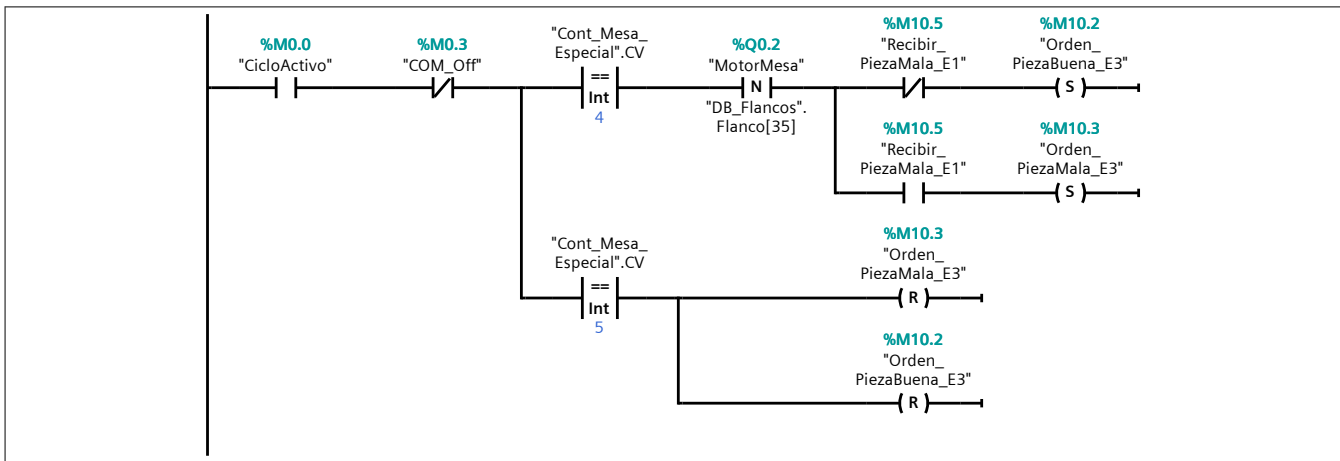
Segmento 10: POSICION 2 --> SUBIR TALADRO



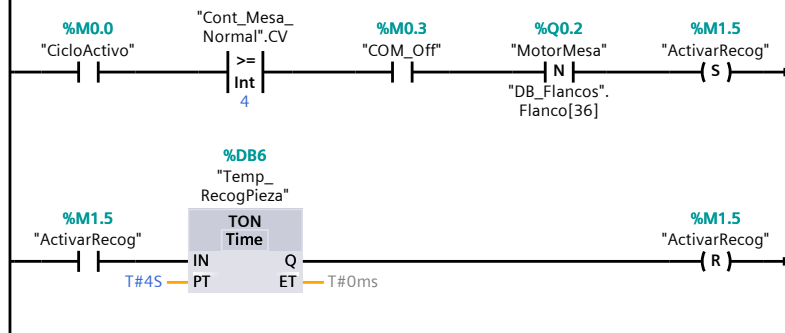
Segmento 11: POSICION 3



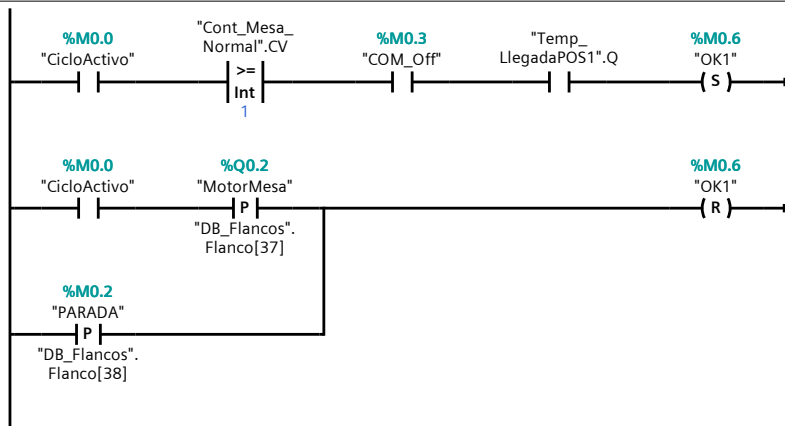
Segmento 12: COMUNICACIONES CON LA ESTACIÓN 3



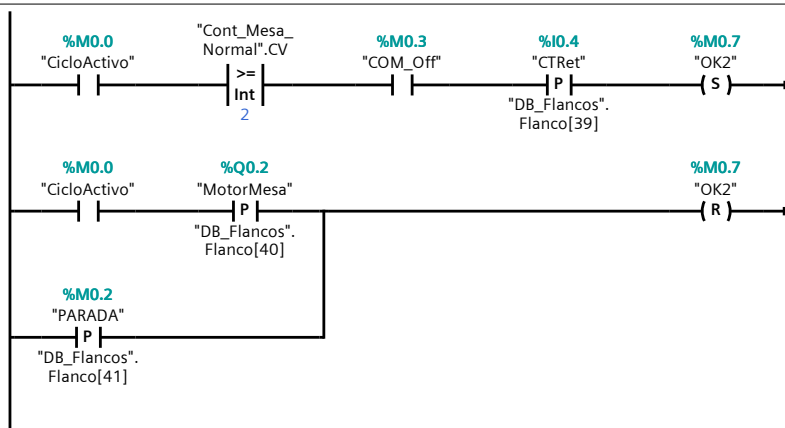
Segmento 13: POSICION 4



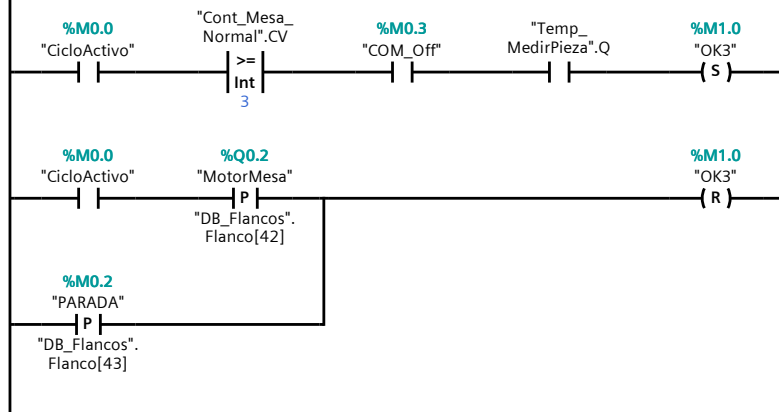
Segmento 14: ---> OK1



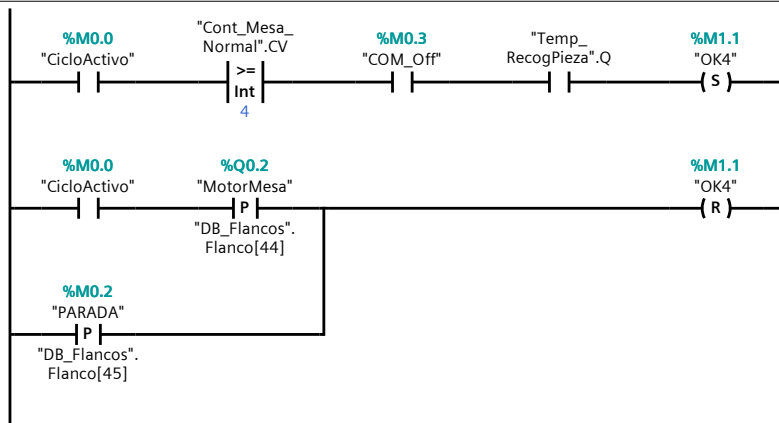
Segmento 15: ---> OK2



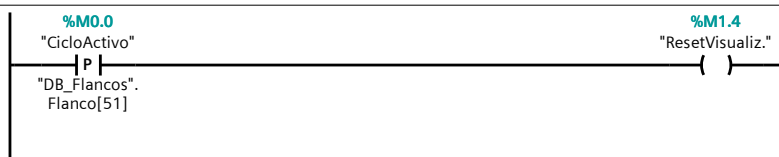
Segmento 16: ---> OK3



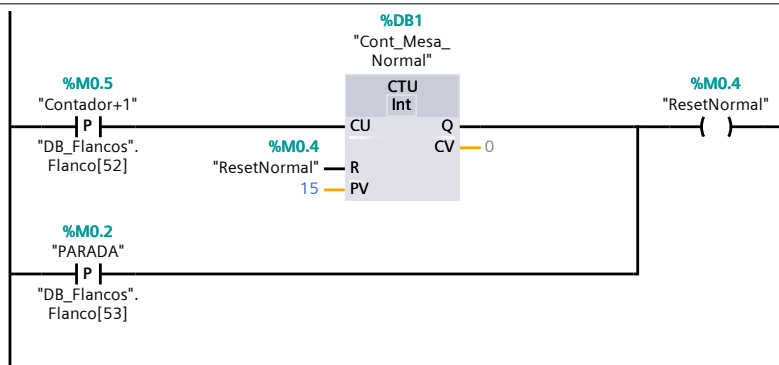
Segmento 17: ---> OK4



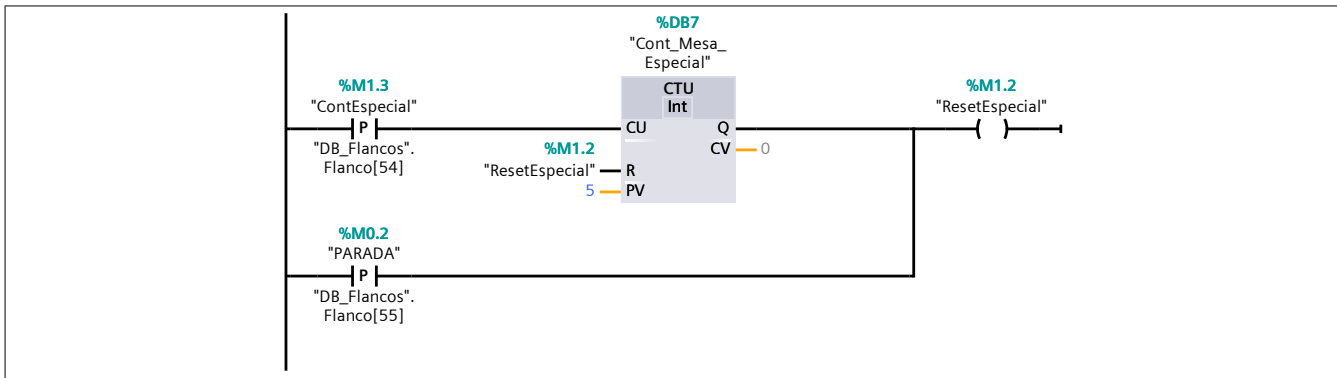
Segmento 18:



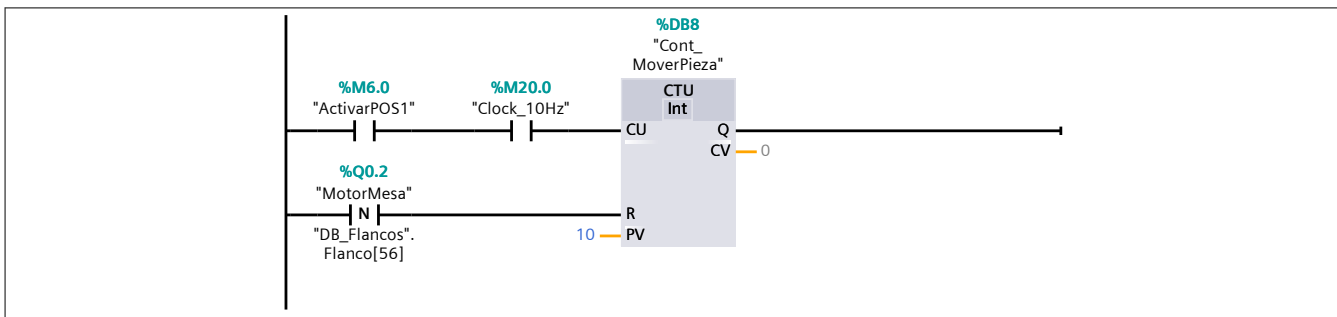
Segmento 19:



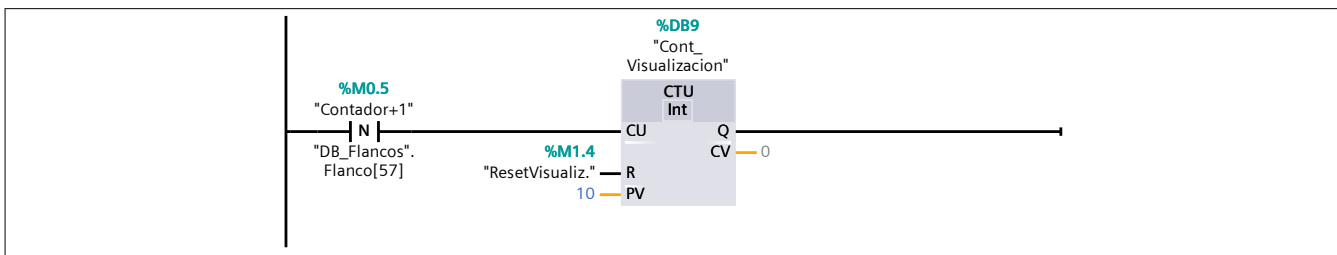
Segmento 20:



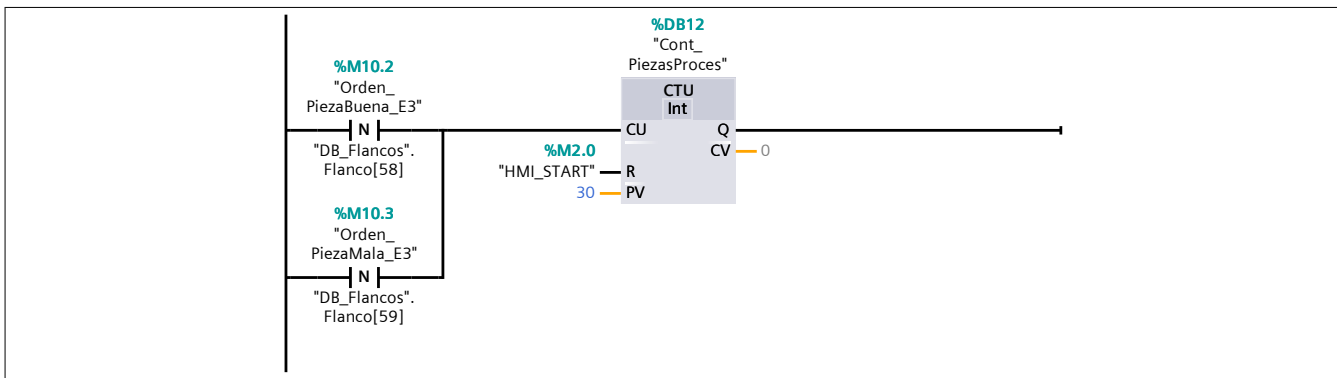
Segmento 21: CONTADOR QUE MUEVE LAS PIEZAS POR LA RAMPA



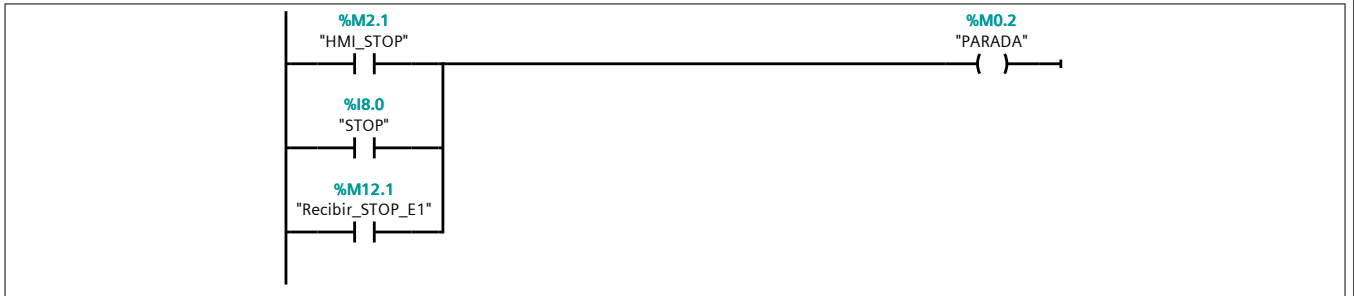
Segmento 22: CONTADOR QUE DETERMINA LA VISUALIZACION DE LA PIEZA



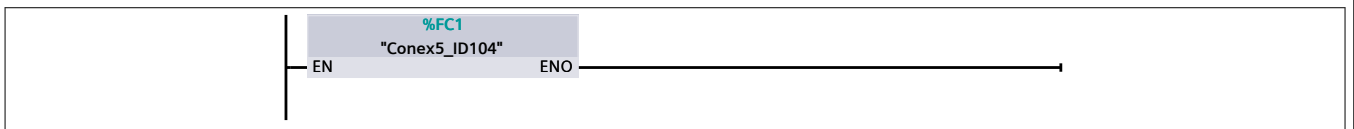
Segmento 23:



Segmento 24:



Segmento 25:



Conex5_ID104 [FC1]

Conex5_ID104 Propiedades

General

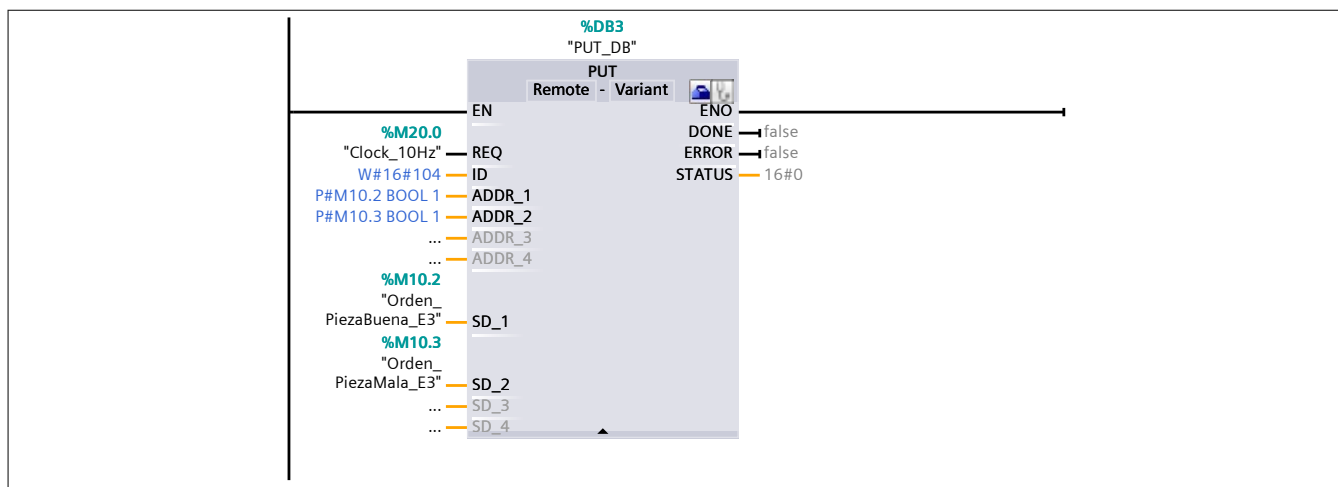
| | | | | | |
|---------------|--------------|-------------------|------------|-------------|----|
| Nombre | Conex5_ID104 | Número | 1 | Tipo | FC |
| Idioma | KOP | Numeración | Automático | | |

Información

| | | | | | |
|----------------|--|----------------|-----|-------------------------|--|
| Título | | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|------------|
| Input | | | |
| Output | | | |
| InOut | | | |
| Temp | | | |
| Constant | | | |
| ▼ Return | | | |
| Conex5_ID104 | Void | | |

Segmento 1:



Segmento 2:

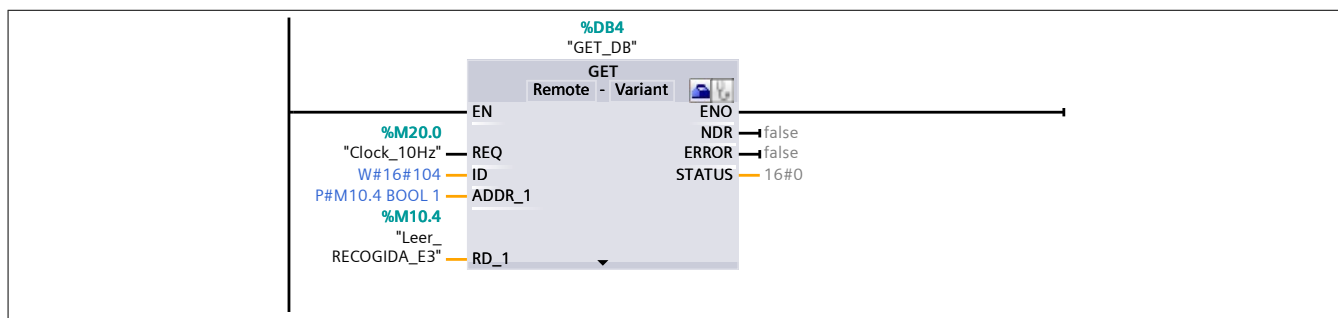


Tabla de variables estándar [91]

Variables PLC

| Variables PLC | | | | | | | | | |
|---------------|------------------|---------------|-----------|------------|-------------------------------------|--------------------------------------|----------------------------|-------------|------------|
| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/Web API | Escribible desde HMI/O PC UA/Web API | Visible en HMI Engineering | Supervisión | Comentario |
| | Posicion1 | Bool | %I0.0 | False | True | True | True | | |
| | MesaCorrecta | Bool | %I0.1 | False | True | True | True | | |
| | CSRet | Bool | %I0.2 | False | True | True | True | | |
| | CSExt | Bool | %I0.3 | False | True | True | True | | |
| | CTRet | Bool | %I0.4 | False | True | True | True | | |
| | CTExt | Bool | %I0.5 | False | True | True | True | | |
| | CPRet | Bool | %I0.6 | False | True | True | True | | |
| | CPExt | Bool | %I0.7 | False | True | True | True | | |
| | CSujecion | Bool | %Q0.0 | False | True | True | True | | |
| | MotorTaladro | Bool | %Q0.1 | False | True | True | True | | |
| | MotorMesa | Bool | %Q0.2 | False | True | True | True | | |
| | BajarTaladro | Bool | %Q0.3 | False | True | True | True | | |
| | SubirTaladro | Bool | %Q0.4 | False | True | True | True | | |
| | CPrueba | Bool | %Q8.0 | False | True | True | True | | |
| | ResetNormal | Bool | %M0.4 | False | True | True | True | | |
| | Contador+1 | Bool | %M0.5 | False | True | True | True | | |
| | STOP | Bool | %I8.0 | False | True | True | True | | |
| | START | Bool | %I8.1 | False | True | True | True | | |
| | Clock_Byte | Byte | %MB20 | False | True | True | True | | |
| | Clock_10Hz | Bool | %M20.0 | False | True | True | True | | |
| | Clock_5Hz | Bool | %M20.1 | False | True | True | True | | |
| | Clock_2.5Hz | Bool | %M20.2 | False | True | True | True | | |
| | Clock_2Hz | Bool | %M20.3 | False | True | True | True | | |
| | Clock_1.25Hz | Bool | %M20.4 | False | True | True | True | | |
| | Clock_1Hz | Bool | %M20.5 | False | True | True | True | | |
| | Clock_0.625Hz | Bool | %M20.6 | False | True | True | True | | |
| | Clock_0.5Hz | Bool | %M20.7 | False | True | True | True | | |
| | HMI_START | Bool | %M2.0 | False | True | True | True | | |
| | OK1 | Bool | %M0.6 | False | True | True | True | | |
| | OK2 | Bool | %M0.7 | False | True | True | True | | |
| | OK3 | Bool | %M1.0 | False | True | True | True | | |
| | OK4 | Bool | %M1.1 | False | True | True | True | | |
| | HMI_STOP | Bool | %M2.1 | False | True | True | True | | |
| | AUX_Panel | Bool | %M4.2 | False | True | True | True | | |
| | Recibir_START_E1 | Bool | %M12.0 | False | True | True | True | | |

| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/W eb API | Escribible desde HMI/O PC UA/W eb API | Visible en HMI Engineering | Supervisión | Comentario |
|--|----------------------|---------------|-----------|------------|--------------------------------------|---------------------------------------|----------------------------|-------------|------------|
| | MODO_MANUAL | Bool | %M0.1 | False | True | True | True | | |
| | PARADA | Bool | %M0.2 | False | True | True | True | | |
| | ActivarPOS1 | Bool | %M6.0 | False | True | True | True | | |
| | HMI_MoverMesa | Bool | %M2.2 | False | True | True | True | | |
| | HMI_PararMesa | Bool | %M2.3 | False | True | True | True | | |
| | ActivarSujecion | Bool | %M1.7 | False | True | True | True | | |
| | ActivarTalad | Bool | %M1.6 | False | True | True | True | | |
| | HMI_Sujecion | Bool | %M2.4 | False | True | True | True | | |
| | HMI_BajarTal | Bool | %M2.5 | False | True | True | True | | |
| | HMI_SubirTal | Bool | %M2.6 | False | True | True | True | | |
| | HMI_ActTaladro | Bool | %M2.7 | False | True | True | True | | |
| | HMI_ExtPrueba | Bool | %M3.0 | False | True | True | True | | |
| | HMI_RetPrueba | Bool | %M3.1 | False | True | True | True | | |
| | ActivarRecog | Bool | %M1.5 | False | True | True | True | | |
| | COM_Off | Bool | %M0.3 | False | True | True | True | | |
| | ResetVisualiz. | Bool | %M1.4 | False | True | True | True | | |
| | Enviar_LISTO_E1 | Bool | %M10.0 | False | True | True | True | | |
| | ContEspecial | Bool | %M1.3 | False | True | True | True | | |
| | ResetEspecial | Bool | %M1.2 | False | True | True | True | | |
| | Leer_RECOGIDA_E3 | Bool | %M10.4 | False | True | True | True | | |
| | Recibir_STOP_E1 | Bool | %M12.1 | False | True | True | True | | |
| | Recibir_COMENZAR_E1 | Bool | %M10.1 | False | True | True | True | | |
| | CicloActivo | Bool | %M0.0 | False | True | True | True | | |
| | Orden_PiezaBuena_E3 | Bool | %M10.2 | False | True | True | True | | |
| | Orden_PiezaMala_E3 | Bool | %M10.3 | False | True | True | True | | |
| | Recibir_PiezaMala_E1 | Bool | %M10.5 | False | True | True | True | | |

ANEXO 4.
CÓDIGO DE LA
ESTACIÓN 3
(S7-1200)

Startup [OB100]

Startup Propiedades

General

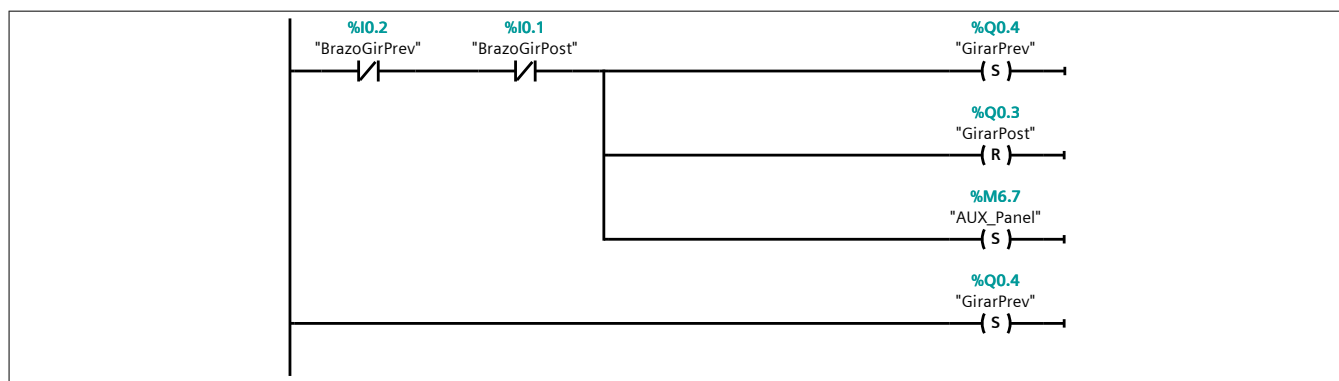
| | | | | | |
|---------------|---------|-------------------|------------|-------------|----|
| Nombre | Startup | Número | 100 | Tipo | OB |
| Idioma | KOP | Numeración | Automático | | |

Información

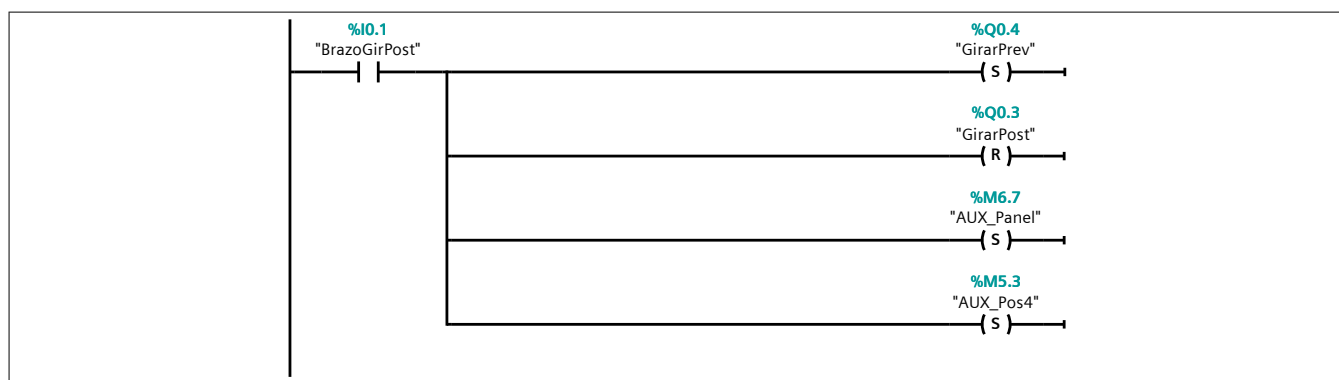
| | | | | | |
|----------------|--------------------|----------------|-----|-------------------------|--|
| Título | "Complete Restart" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|---------------|---------------|---------------|---------------------------------|
| ▼ Input | | | |
| LostRetentive | Bool | | True if retentive data are lost |
| LostRTC | Bool | | True if date and time are lost |
| Temp | | | |
| Constant | | | |

Segmento 1:

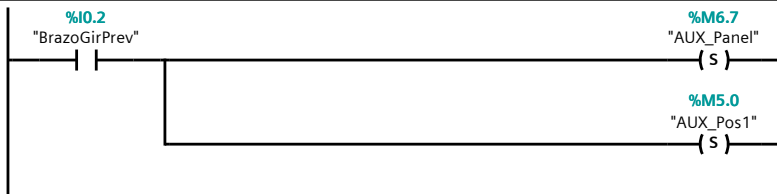


Segmento 2:

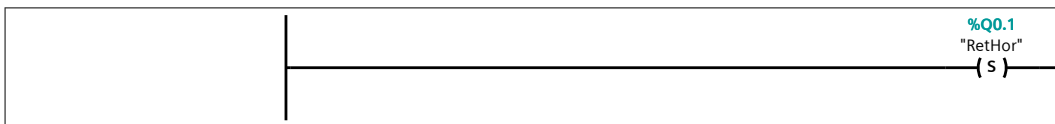


Segmento 3:

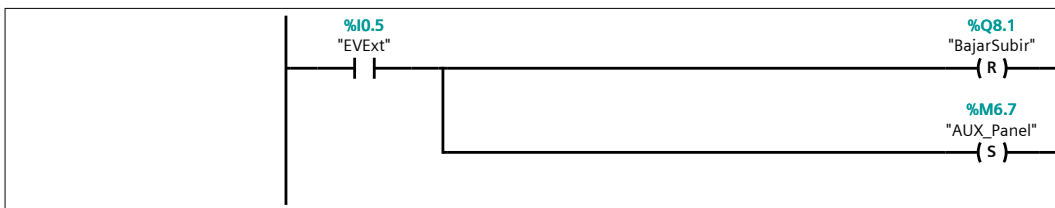
| | | |
|--|--|--|
| | | |
|--|--|--|



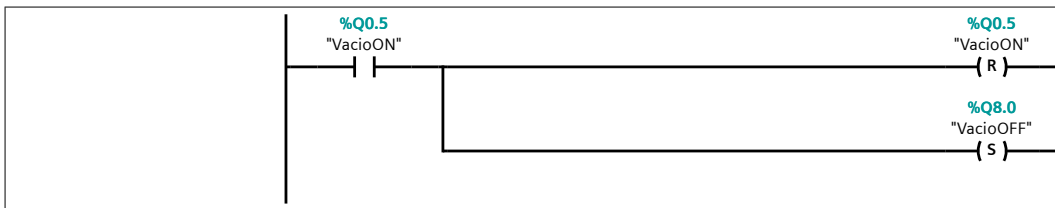
Segmento 4:



Segmento 5:



Segmento 6:



Main [OB1]

Main Propiedades

General

| | | | | | |
|---------------|------|-------------------|------------|-------------|----|
| Nombre | Main | Número | 1 | Tipo | OB |
| Idioma | KOP | Numeración | Automático | | |

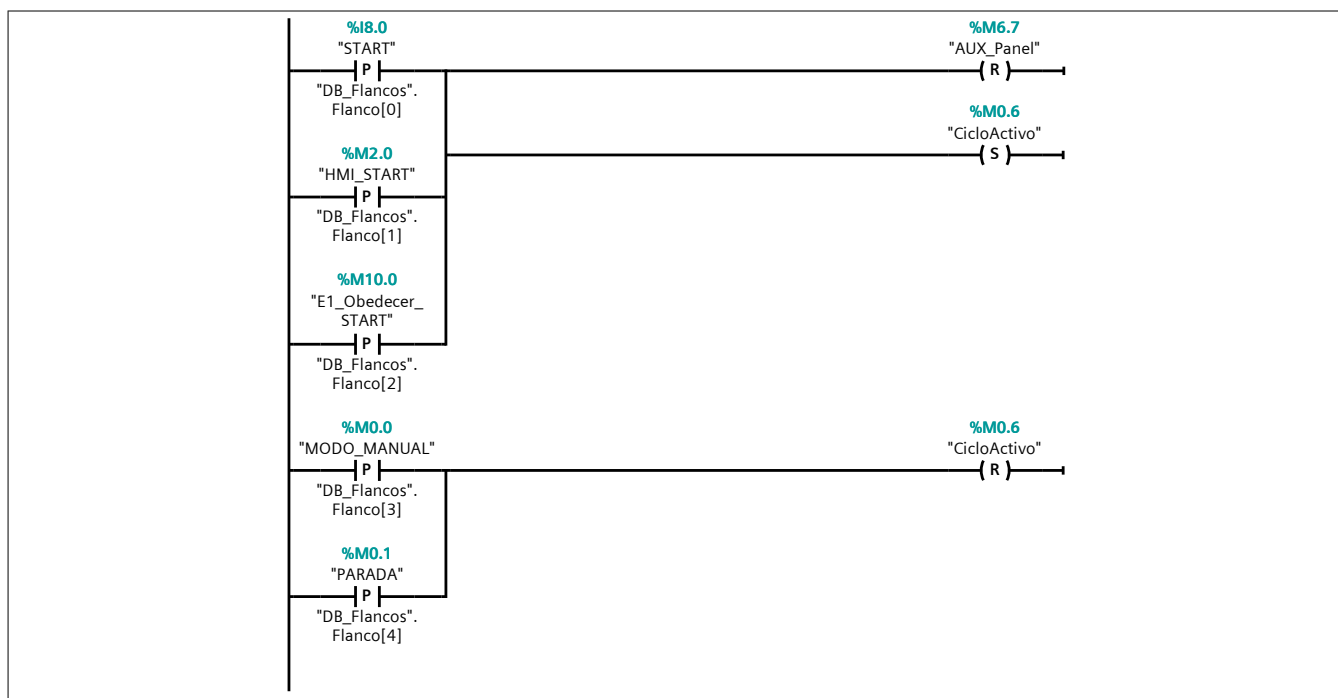
Información

| | | | | | |
|----------------|------------------------------|----------------|-----|-------------------------|--|
| Título | "Main Program Sweep (Cycle)" | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

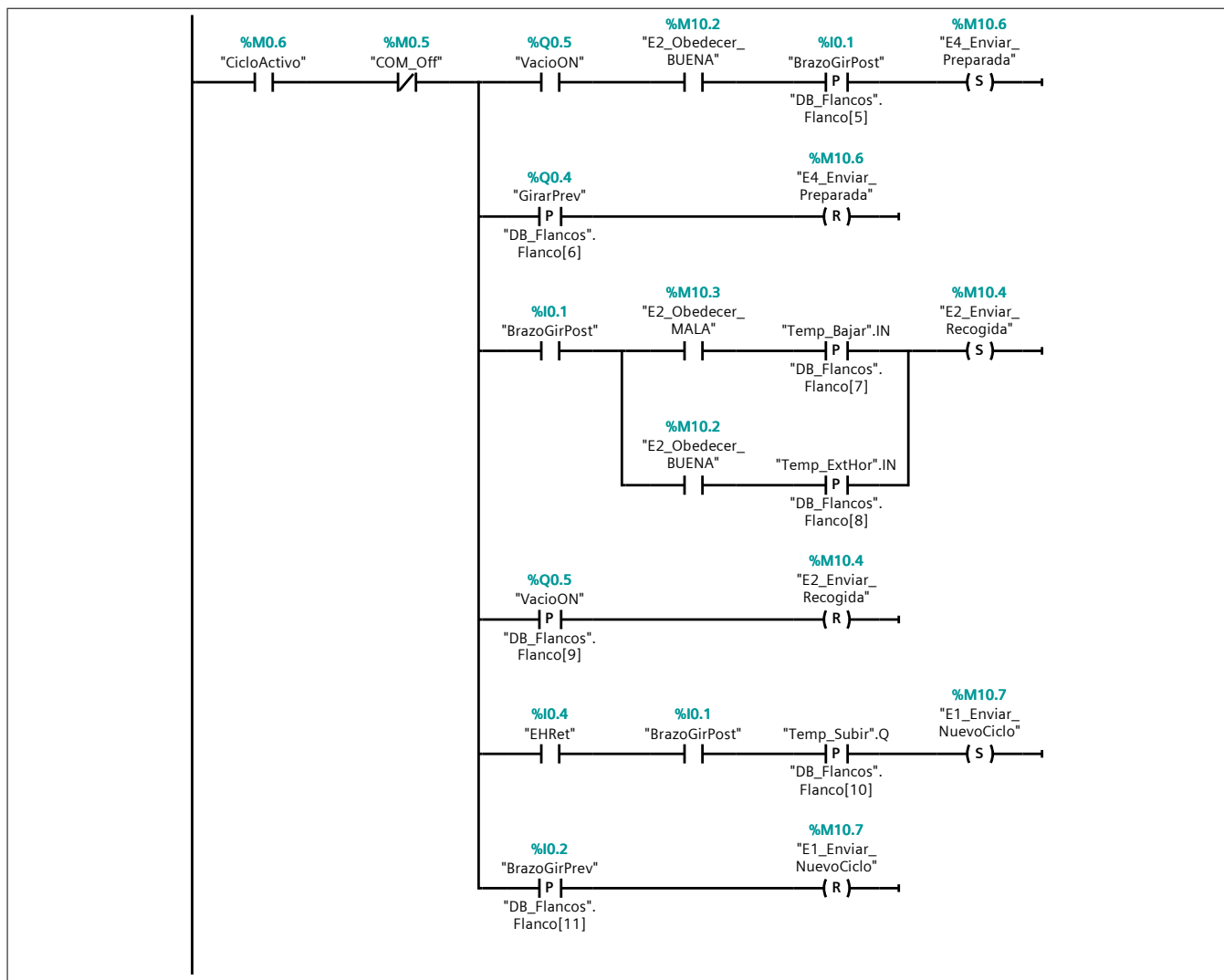
| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|---------------------------------------|
| ▼ Input | | | |
| Initial_Call | Bool | | Initial call of this OB |
| Remanence | Bool | | =True, if remanent data are available |
| Temp | | | |
| Constant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

Segmento 1: Inicio del Ciclo

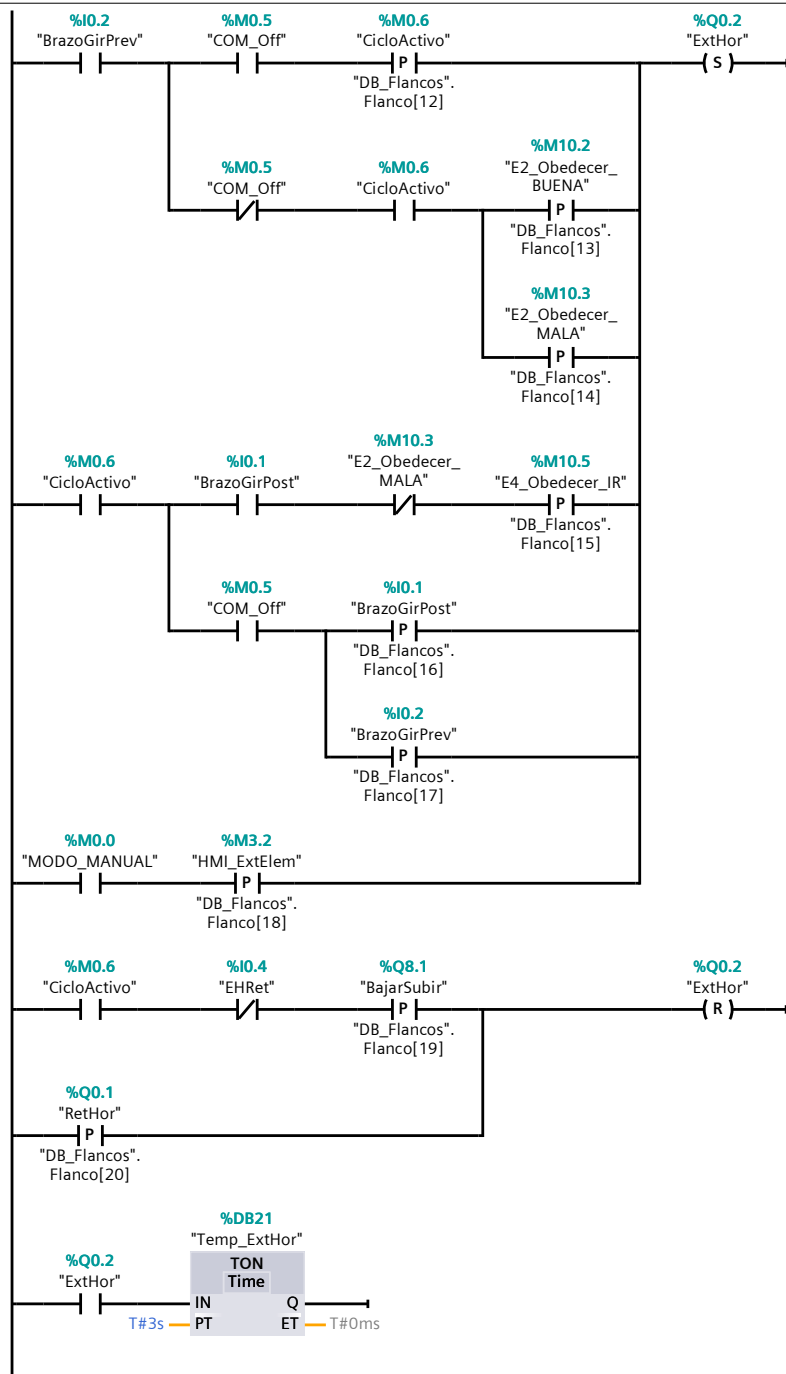
AUTOR: JAVIER RODRÍGUEZ DE LA ROSA



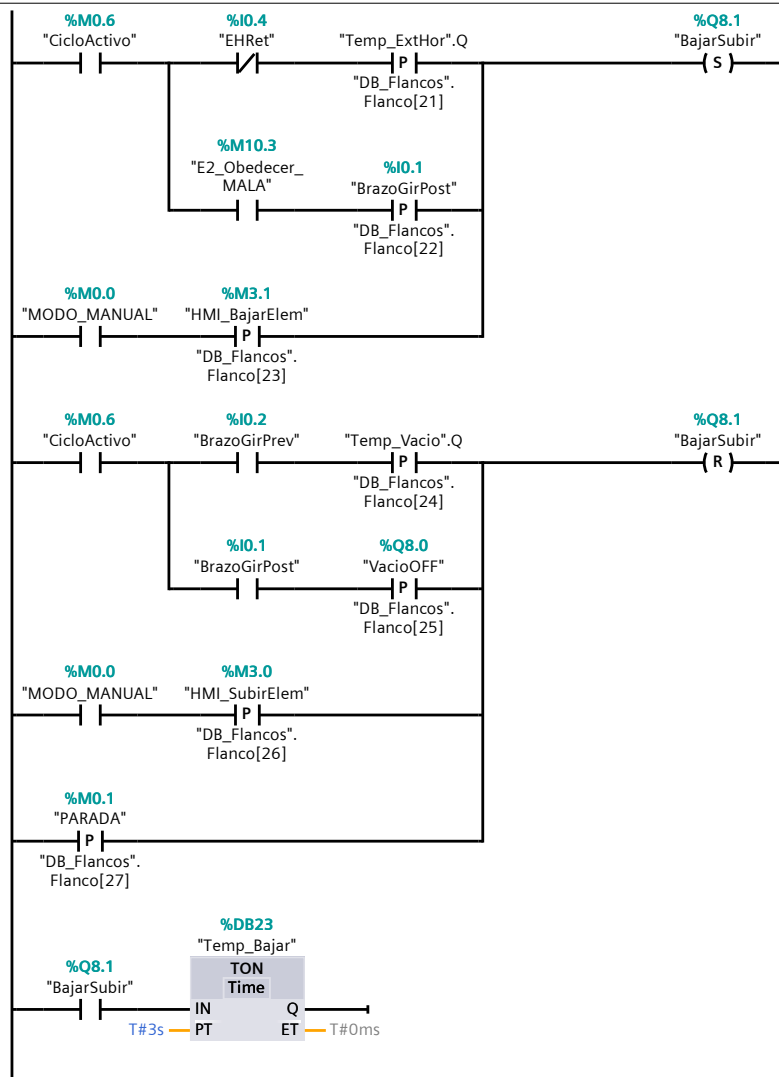
Segmento 2: COMUNICACIONES



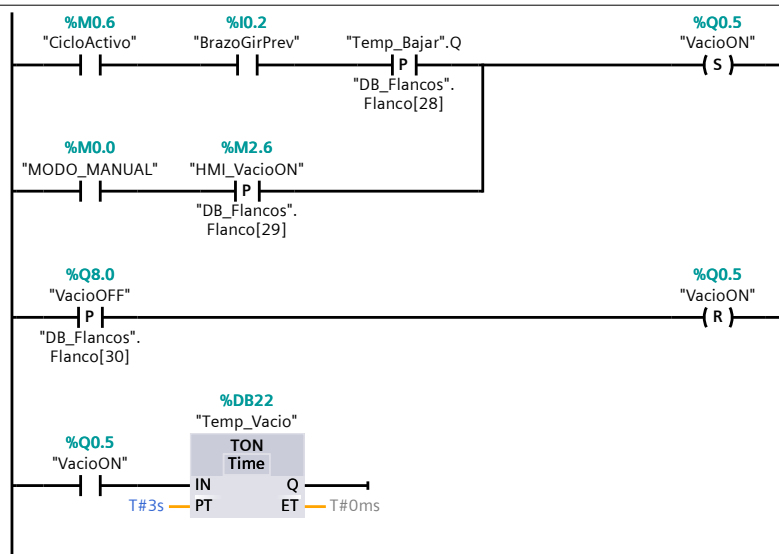
Segmento 3: --- > (Q0.2) ExtHor



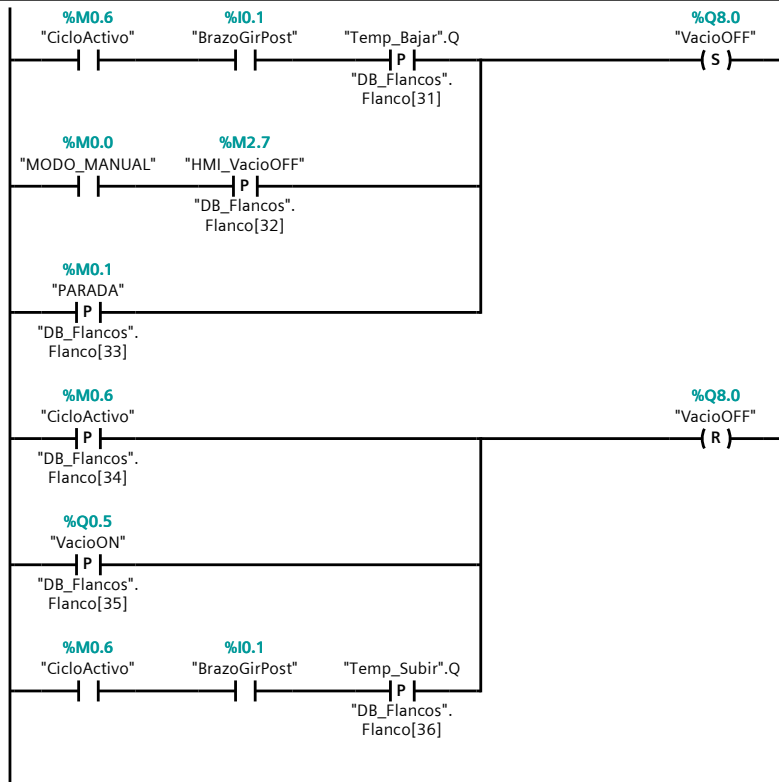
Segmento 4: ---> (Q8.1) BajarSubir



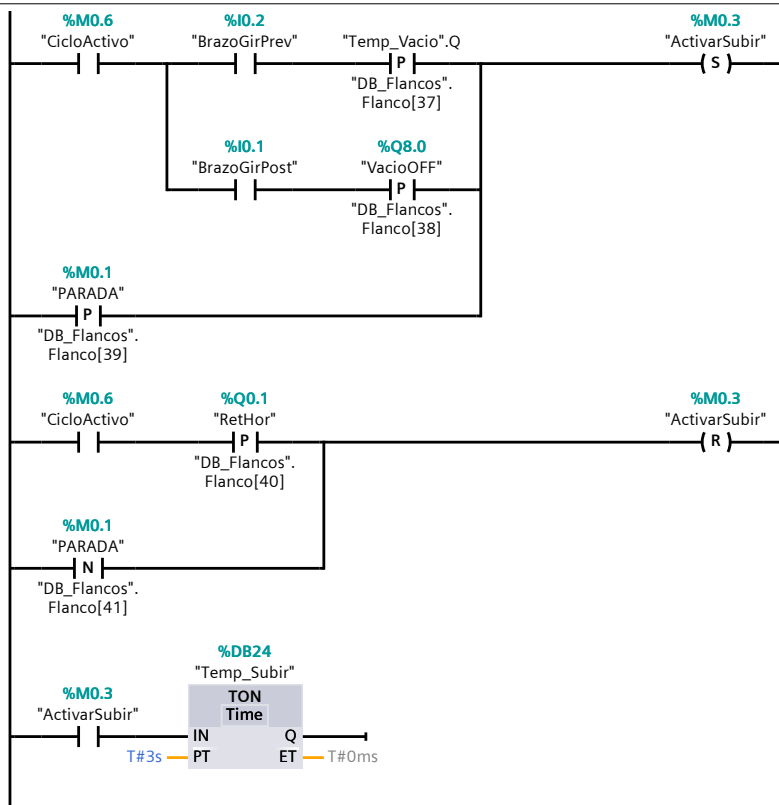
Segmento 5: ---> (Q0.5) VacioON



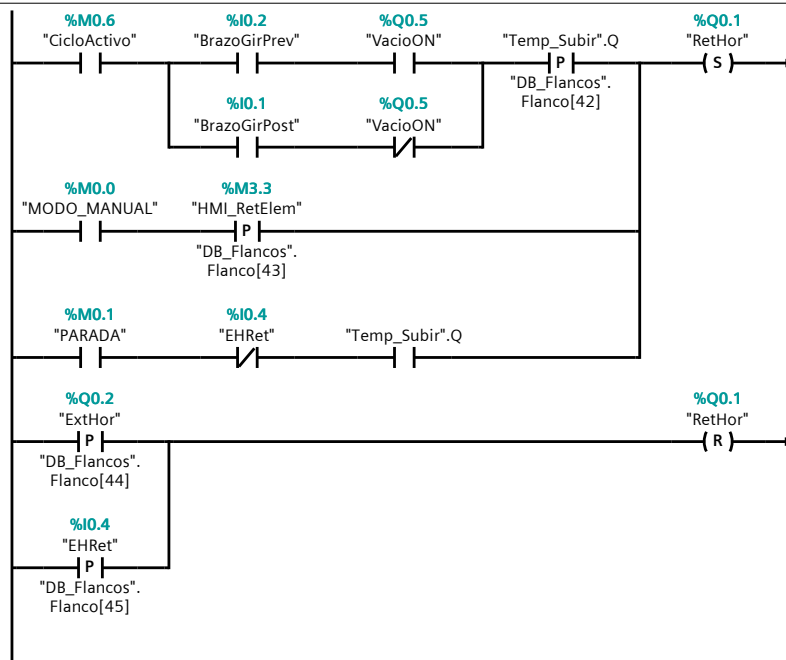
Segmento 6: ---> (Q8.0) VacioOFF



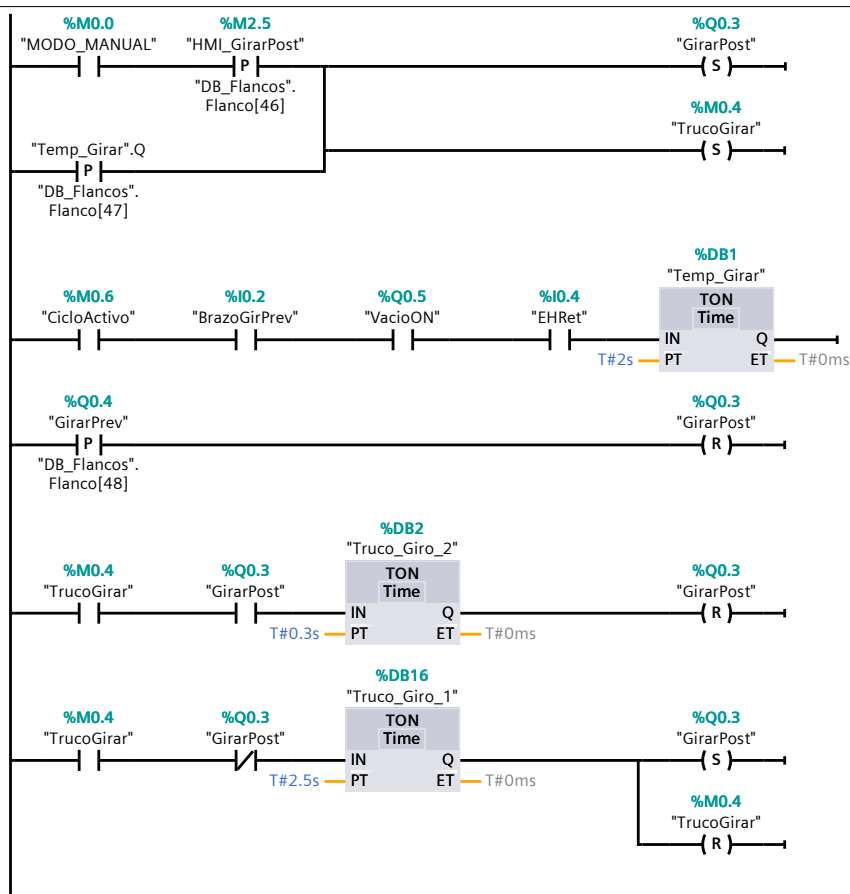
Segmento 7: --- > Activar Subir



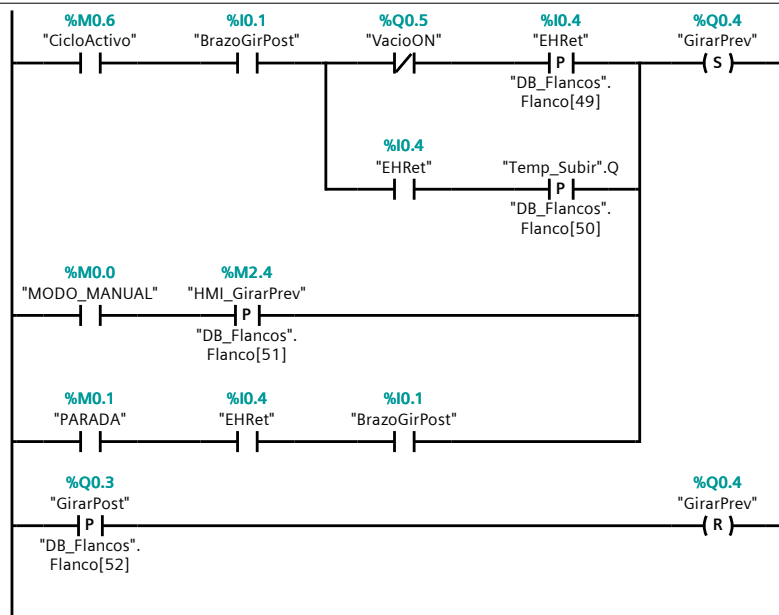
Segmento 8: --- > (Q0.1) RetHor



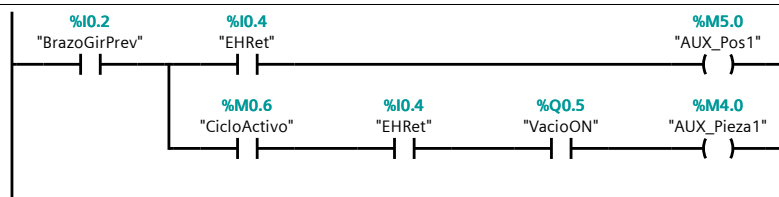
Segmento 9: --- > (Q0.3) GirarPosterior



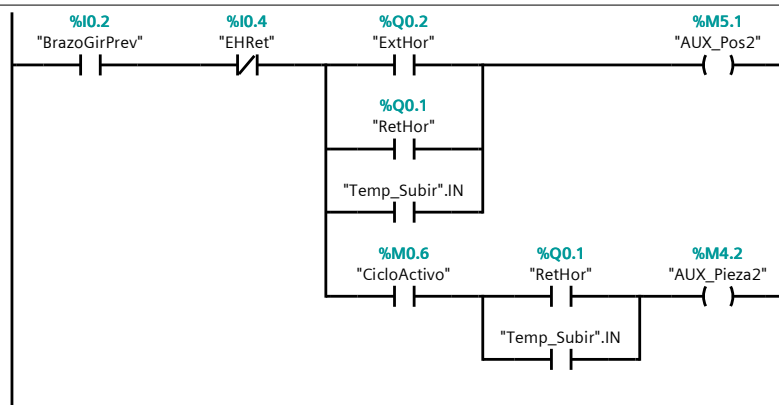
Segmento 10: --- > (Q0.4) GirarPrevia



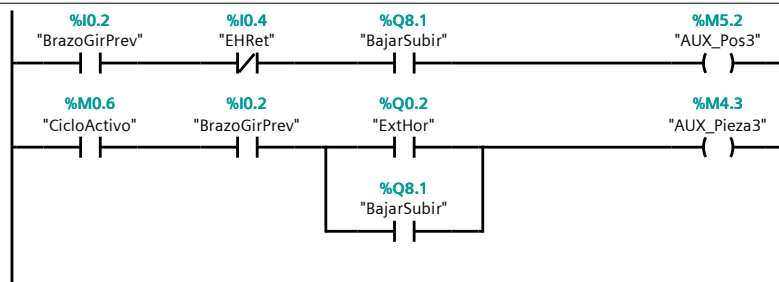
Segmento 11: --- > AUX Pos1 + Pieza1



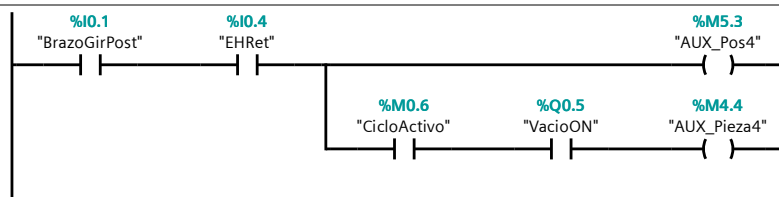
Segmento 12: --- > AUX Pos2 + Pieza2



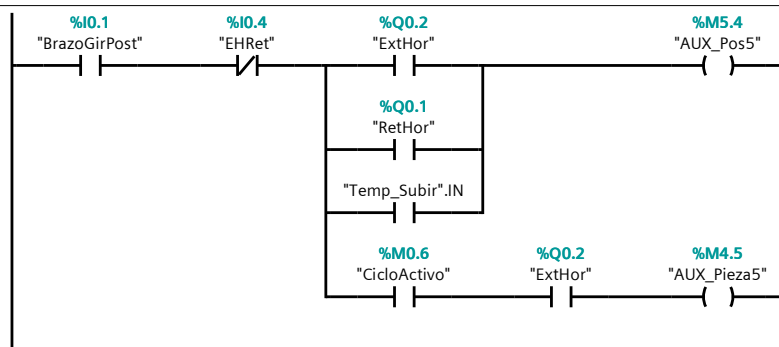
Segmento 13: --- > AUX Pos3 + Pieza3



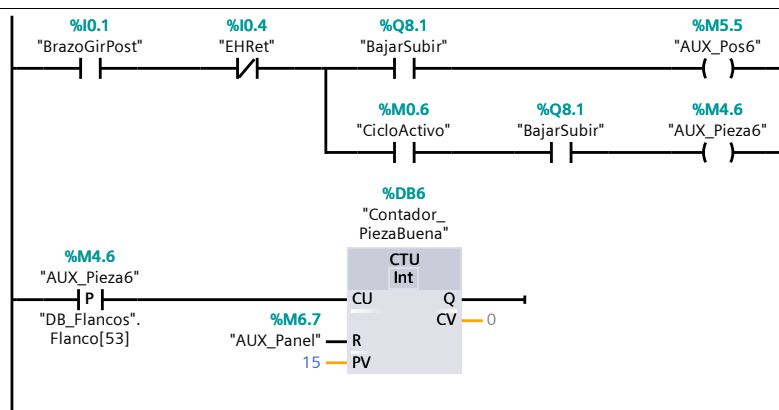
Segmento 14: --- > AUX Pos4 + Pieza4



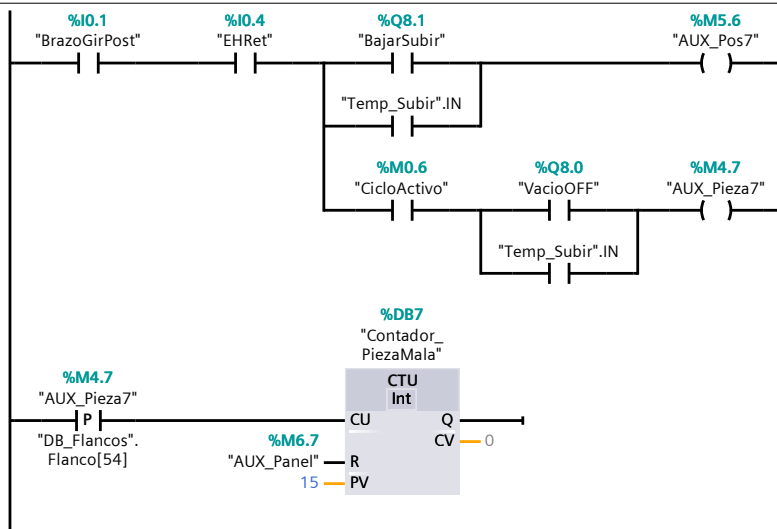
Segmento 15: --- > AUX Pos5 + Pieza5



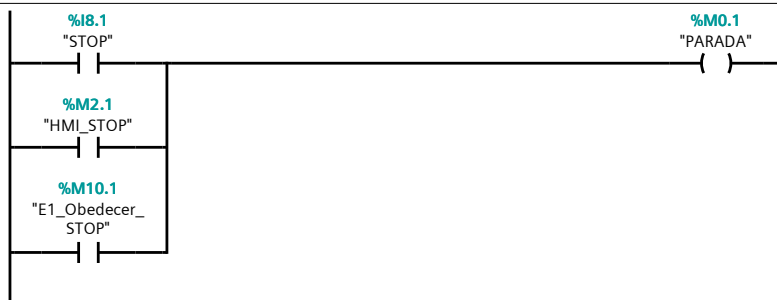
Segmento 16: --- > AUX Pos6 + Pieza6



Segmento 17: --- > AUX Pos7 + Pieza7



Segmento 18: ---> PARADA



Segmento 19: ---> AUX Flechas

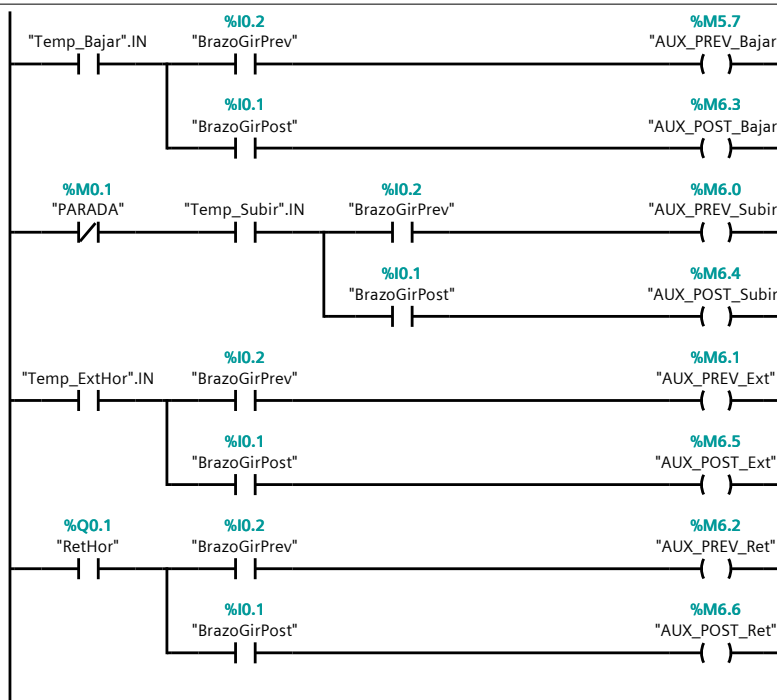


Tabla de variables estándar [95]

Variables PLC

| Variables PLC | | | | | | | | | |
|---------------|----------------|---------------|-----------|------------|-------------------------------------|--------------------------------------|----------------------------|-------------|------------|
| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/Web API | Escribible desde HMI/O PC UA/Web API | Visible en HMI Engineering | Supervisión | Comentario |
| | PRecogida | Bool | %I0.0 | False | True | True | True | | |
| | BrazoGirPost | Bool | %I0.1 | False | True | True | True | | |
| | BrazoGirPrev | Bool | %I0.2 | False | True | True | True | | |
| | EHRet | Bool | %I0.4 | False | True | True | True | | |
| | EVExt | Bool | %I0.5 | False | True | True | True | | |
| | RetHor | Bool | %Q0.1 | False | True | True | True | | |
| | ExtHor | Bool | %Q0.2 | False | True | True | True | | |
| | GirarPost | Bool | %Q0.3 | False | True | True | True | | |
| | GirarPrev | Bool | %Q0.4 | False | True | True | True | | |
| | VacioON | Bool | %Q0.5 | False | True | True | True | | |
| | VacioOFF | Bool | %Q8.0 | False | True | True | True | | |
| | BajarSubir | Bool | %Q8.1 | False | True | True | True | | |
| | START | Bool | %I8.0 | False | True | True | True | | |
| | STOP | Bool | %I8.1 | False | True | True | True | | |
| | HMI_START | Bool | %M2.0 | False | True | True | True | | |
| | AUX_Pieza1 | Bool | %M4.0 | False | True | True | True | | |
| | AUX_Pieza2 | Bool | %M4.2 | False | True | True | True | | |
| | AUX_Pieza3 | Bool | %M4.3 | False | True | True | True | | |
| | AUX_Pieza4 | Bool | %M4.4 | False | True | True | True | | |
| | AUX_Pieza5 | Bool | %M4.5 | False | True | True | True | | |
| | AUX_Pieza6 | Bool | %M4.6 | False | True | True | True | | |
| | AUX_Pos1 | Bool | %M5.0 | False | True | True | True | | |
| | AUX_Pos2 | Bool | %M5.1 | False | True | True | True | | |
| | AUX_Pos3 | Bool | %M5.2 | False | True | True | True | | |
| | AUX_Pos4 | Bool | %M5.3 | False | True | True | True | | |
| | AUX_Pos5 | Bool | %M5.4 | False | True | True | True | | |
| | AUX_Pos6 | Bool | %M5.5 | False | True | True | True | | |
| | AUX_Pos7 | Bool | %M5.6 | False | True | True | True | | |
| | AUX_PREV_Bajar | Bool | %M5.7 | False | True | True | True | | |
| | AUX_PREV_Subir | Bool | %M6.0 | False | True | True | True | | |
| | AUX_PREV_Ext | Bool | %M6.1 | False | True | True | True | | |
| | AUX_PREV_Ret | Bool | %M6.2 | False | True | True | True | | |
| | AUX_POST_Bajar | Bool | %M6.3 | False | True | True | True | | |
| | AUX_POST_Subir | Bool | %M6.4 | False | True | True | True | | |
| | AUX_POST_Ext | Bool | %M6.5 | False | True | True | True | | |

| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/W eb API | Escribible desde HMI/O PC UA/W eb API | Visible en HMI Engineering | Supervisión | Comentario |
|--|----------------------|---------------|-----------|------------|--------------------------------------|---------------------------------------|----------------------------|-------------|------------|
| | AUX_POST_Ret | Bool | %M6.6 | False | True | True | True | | |
| | AUX_Pieza7 | Bool | %M4.7 | False | True | True | True | | |
| | AUX_Panel | Bool | %M6.7 | False | True | True | True | | |
| | HMI_STOP | Bool | %M2.1 | False | True | True | True | | |
| | AUX_Succion_Pieza | Bool | %M7.0 | False | True | True | True | | |
| | MODO_MANUAL | Bool | %M0.0 | False | True | True | True | | |
| | PARADA | Bool | %M0.1 | False | True | True | True | | |
| | HMI_ExtenderHor | Bool | %M2.2 | False | True | True | True | | |
| | HMI_RetraerHor | Bool | %M2.3 | False | True | True | True | | |
| | HMI_GirarPrev | Bool | %M2.4 | False | True | True | True | | |
| | HMI_GirarPost | Bool | %M2.5 | False | True | True | True | | |
| | HMI_VacioON | Bool | %M2.6 | False | True | True | True | | |
| | HMI_VacioOFF | Bool | %M2.7 | False | True | True | True | | |
| | HMI_SubirElem | Bool | %M3.0 | False | True | True | True | | |
| | HMI_BajarElem | Bool | %M3.1 | False | True | True | True | | |
| | HMI_ExtElem | Bool | %M3.2 | False | True | True | True | | |
| | HMI_RetElem | Bool | %M3.3 | False | True | True | True | | |
| | E4_Obedecer_IR | Bool | %M10.5 | False | True | True | True | | |
| | E1_Obedecer_STOP | Bool | %M10.1 | False | True | True | True | | |
| | CicloActivo | Bool | %M0.6 | False | True | True | True | | |
| | COM_Off | Bool | %M0.5 | False | True | True | True | | |
| | ActivarSubir | Bool | %M0.3 | False | True | True | True | | |
| | TrucoGirar | Bool | %M0.4 | False | True | True | True | | |
| | E2_Obedecer_BUENA | Bool | %M10.2 | False | True | True | True | | |
| | E2_Obedecer_MALA | Bool | %M10.3 | False | True | True | True | | |
| | E1_Obedecer_START | Bool | %M10.0 | False | True | True | True | | |
| | E4_Enviar_Preparada | Bool | %M10.6 | False | True | True | True | | |
| | E2_Enviar_Recogida | Bool | %M10.4 | False | True | True | True | | |
| | E1_Enviar_NuevoCiclo | Bool | %M10.7 | False | True | True | True | | |
| | GIRAR | Bool | %M11.0 | False | True | True | True | | |

ANEXO 5.
CÓDIGO DE LA
ESTACIÓN 4
(S7-1500)


```

0014      A      "ExtenderD2"
0015      A      "Desv2_EXT"
0016      =      "Pos2"
0017      )
0018      =      "OK"
    
```

Segmento 3: PARAR CINTA

```

0001      O      "STOP"
0002      O      "HMI_STOP"
0003      O      "PiezaAlmacen"
0004      R      "MovCinta"
0005      R      "OK"
0006      =      "ResetC1"
0007      =      "ResetC2"
0008      =      "ResetC3"
    
```

Segmento 4: EXTENDER D1

```

0001      AN     "MovCinta"
0002      O      "CLAS1"
0003      FP     "DB_Flancos".Flanco[3]
0004      O      "HMI_CLAS1"
0005      FP     "DB_Flancos".Flanco[4]
0006      S      "ExtenderD1"
0007      S      "AUX_d1ext"
0008      R      "AUX_d1ret"
    
```

Segmento 5: RETRAER D1

```

0001      AN     "MovCinta"
0002      O      "CLAS1"
0003      FN     "DB_Flancos".Flanco[5]
0004      O      "HMI_CLAS1"
0005      FN     "DB_Flancos".Flanco[6]
0006      R      "ExtenderD1"
0007      S      "AUX_d1ret"
0008      R      "AUX_d1ext"
    
```

Segmento 6: EXTENDER D2

```

0001      AN     "MovCinta"
0002      O      "CLAS2"
0003      FP     "DB_Flancos".Flanco[7]
0004      O      "HMI_CLAS2"
0005      FP     "DB_Flancos".Flanco[8]
0006      S      "ExtenderD2"
0007      S      "AUX_d2ext"
0008      R      "AUX_d2ret"
    
```

Segmento 7: RETRAER D2

```

0001      AN     "MovCinta"
    
```

```

0002      O      "CLAS2"
0003      FN     "DB_Flancos".Flanco[9]
0004      O      "HMI_CLAS2"
0005      FN     "DB_Flancos".Flanco[10]
0006      R      "ExtenderD2"
0007      S      "AUX_d2ret"
0008      R      "AUX_d2ext"

```

Segmento 8:

```

0001      AN     "COM_Off"
0002      A      "PiezaAlmacen"
0003      FP     "DB_Flancos".Flanco[11]
0004      S      "E1_Enviar_NuevoCiclo"
0005
0006      A      "MovCinta"
0007      FP     "DB_Flancos".Flanco[12]
0008      R      "E1_Enviar_NuevoCiclo"
0009
0010      AN     "COM_Off"
0011      AN     "MovCinta"
0012      A      "E3_Leer_Datos"
0013      =      "E3_Orden_VENIR"

```

Segmento 9: --- > PARADA

```

0001      O      "STOP"
0002      O      "HMI_STOP"
0003      O      "E1_Recibir_STOP_remoto"
0004      =      "PARADA"

```

Segmento 10: --- > MOVIMIENTO DE PIEZA EN POSICION 3

```

0001      A      "Pos3"
0002      A      "MovCinta"
0003      A      "Clock_5Hz"
0004      =      "ActivarC3"
0005
0006      CALL  CTU , "Posicion3"
0007      Int
0008      CU := "ActivarC3"
0009      R  := "ResetC3"
0010      PV := 30
0011      Q  := "IncremD3"
0012      CV := "CuentaC3"

```

Segmento 11: --- > MOVIMIENTO DE PIEZA EN POSICION 2

```

0001      A      "Pos2"
0002      A      "MovCinta"
0003      A      "Clock_5Hz"
0004      =      "ActivarC2"
0005
0006      CALL  CTU , "Posicion2"

```

```

0007      Int
0008      CU := "ActivarC2"
0009      R  := "ResetC2"
0010      PV := 30
0011      Q  := "IncremD2"
0012      CV := "CuentaC2"

```

Segmento 12: ---> MOVIMIENTO DE PIEZA EN POSICION 1

```

0001      A      "Pos1"
0002      A      "MovCinta"
0003      A      "Clock_5Hz"
0004      =      "ActivarC1"
0005
0006      CALL CTU , "Posicion1"
0007      Int
0008      CU := "ActivarC1"
0009      R  := "ResetC1"
0010      PV := 20
0011      Q  := "IncremD1"
0012      CV := "CuentaC1"

```

Segmento 13: ---> CUENTA DE PIEZAS PROCESADAS

```

0001      CALL CTU , "ProcesadasD1"
0002      Int
0003      CU := "IncremD1"
0004      R  :=
0005      PV := 30
0006      Q  :=
0007      CV :=
0008
0009      CALL CTU , "ProcesadasD2"
0010      Int
0011      CU := "IncremD2"
0012      R  :=
0013      PV := 30
0014      Q  :=
0015      CV :=
0016
0017      CALL CTU , "ProcesadasD3"
0018      Int
0019      CU := "IncremD3"
0020      R  :=
0021      PV := 30
0022      Q  :=
0023      CV :=

```

Segmento 14: COMUNICACIÓN ESTACIÓN 3

```

0001      CALL "Conex6_ID105"

```


Conex6_ID105 [FC1]

Conex6_ID105 Propiedades

General

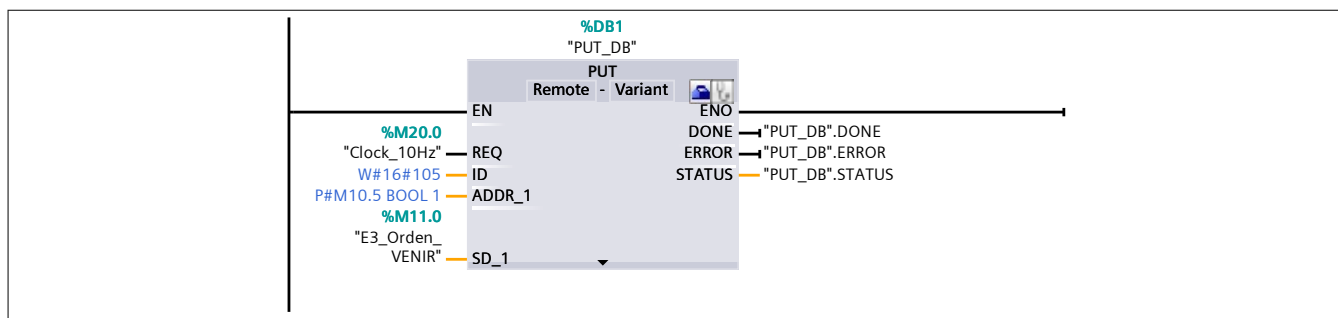
| | | | | | |
|---------------|--------------|-------------------|------------|-------------|----|
| Nombre | Conex6_ID105 | Número | 1 | Tipo | FC |
| Idioma | KOP | Numeración | Automático | | |

Información

| | | | | | |
|----------------|--|----------------|-----|-------------------------|--|
| Título | | Autor | | Comentario | |
| Familia | | Versión | 0.1 | ID personalizado | |

| Nombre | Tipo de datos | Valor predet. | Comentario |
|--------------|---------------|---------------|------------|
| Input | | | |
| Output | | | |
| InOut | | | |
| Temp | | | |
| Constant | | | |
| ▼ Return | | | |
| Conex6_ID105 | Void | | |

Segmento 1:



Segmento 2:

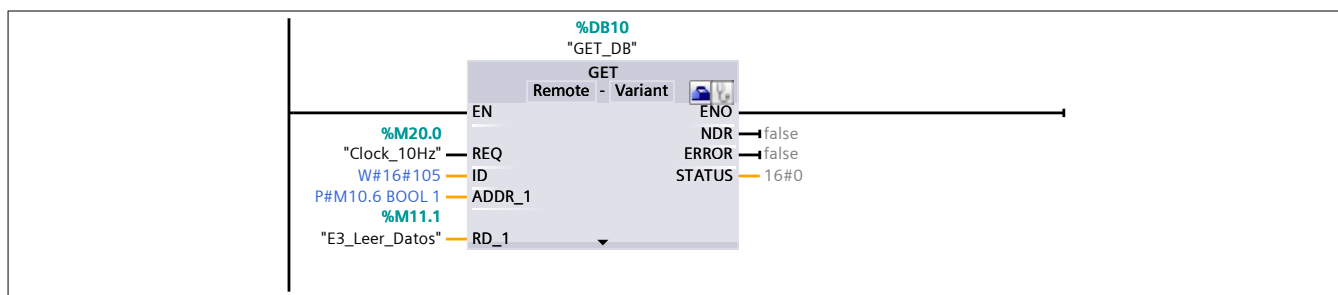


Tabla de variables estándar [111]

Variables PLC

| Variables PLC | | | | | | | | | |
|---------------|-----------------|---------------|-----------|------------|--------------------------------------|---------------------------------------|----------------------------|-------------|------------|
| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/W eb API | Escribible desde HMI/O PC UA/W eb API | Visible en HMI Engineering | Supervisión | Comentario |
| | Desv1_RET | Bool | %I0.0 | False | True | True | True | | |
| | Desv2_RET | Bool | %I0.2 | False | True | True | True | | |
| | Int_MODO_MANUAL | Bool | %I1.1 | False | True | True | True | | |
| | START | Bool | %I1.0 | False | True | True | True | | |
| | STOP | Bool | %I1.2 | False | True | True | True | | |
| | Desv1_EXT | Bool | %I0.1 | False | True | True | True | | |
| | Desv2_EXT | Bool | %I0.3 | False | True | True | True | | |
| | CLAS1 | Bool | %I1.3 | False | True | True | True | | |
| | CLAS2 | Bool | %I1.4 | False | True | True | True | | |
| | PiezaCinta | Bool | %I0.4 | False | True | True | True | | |
| | PiezaAlmacen | Bool | %I0.5 | False | True | True | True | | |
| | MovCinta | Bool | %Q0.2 | False | True | True | True | | |
| | ExtenderD1 | Bool | %Q0.0 | False | True | True | True | | |
| | ExtenderD2 | Bool | %Q0.1 | False | True | True | True | | |
| | OK | Bool | %M0.0 | False | True | True | True | | |
| | Clock_Byte | Byte | %MB20 | False | True | True | True | | |
| | Clock_10Hz | Bool | %M20.0 | False | True | True | True | | |
| | Clock_5Hz | Bool | %M20.1 | False | True | True | True | | |
| | Clock_2.5Hz | Bool | %M20.2 | False | True | True | True | | |
| | Clock_2Hz | Bool | %M20.3 | False | True | True | True | | |
| | Clock_1.25Hz | Bool | %M20.4 | False | True | True | True | | |
| | Clock_1Hz | Bool | %M20.5 | False | True | True | True | | |
| | Clock_0.625Hz | Bool | %M20.6 | False | True | True | True | | |
| | Clock_0.5Hz | Bool | %M20.7 | False | True | True | True | | |
| | HMI_START | Bool | %M2.0 | False | True | True | True | | |
| | MODO_MANUAL | Bool | %M0.1 | False | True | True | True | | |
| | HMI_CLAS1 | Bool | %M2.2 | False | True | True | True | | |
| | HMI_CLAS2 | Bool | %M2.3 | False | True | True | True | | |
| | AUX_d1ext | Bool | %M4.0 | False | True | True | True | | |
| | AUX_d1ret | Bool | %M4.1 | False | True | True | True | | |
| | AUX_d2ext | Bool | %M4.2 | False | True | True | True | | |
| | AUX_d2ret | Bool | %M4.3 | False | True | True | True | | |
| | Pos1 | Bool | %M0.2 | False | True | True | True | | |
| | Pos2 | Bool | %M0.3 | False | True | True | True | | |
| | Pos3 | Bool | %M0.4 | False | True | True | True | | |

| | Nombre | Tipo de datos | Dirección | Remanencia | Accesible desde HMI/O PC UA/W eb API | Escribible desde HMI/O PC UA/W eb API | Visible en HMI Engineering | Supervisión | Comentario |
|--|-------------------------|---------------|-----------|------------|--------------------------------------|---------------------------------------|----------------------------|-------------|------------|
| | ActivarC1 | Bool | %M0.5 | False | True | True | True | | |
| | ActivarC2 | Bool | %M0.6 | False | True | True | True | | |
| | ActivarC3 | Bool | %M0.7 | False | True | True | True | | |
| | CuentaC1 | Int | %MW1 | False | True | True | True | | |
| | CuentaC2 | Int | %MW2 | False | True | True | True | | |
| | CuentaC3 | Int | %MW3 | False | True | True | True | | |
| | ResetC1 | Bool | %M1.0 | False | True | True | True | | |
| | ResetC2 | Bool | %M1.1 | False | True | True | True | | |
| | ResetC3 | Bool | %M1.2 | False | True | True | True | | |
| | IncremD1 | Bool | %M1.3 | False | True | True | True | | |
| | IncremD2 | Bool | %M1.4 | False | True | True | True | | |
| | IncremD3 | Bool | %M1.5 | False | True | True | True | | |
| | HMI_STOP | Bool | %M2.1 | False | True | True | True | | |
| | Establecer_COM | Bool | %I1.7 | False | True | True | True | | |
| | PARADA | Bool | %M1.6 | False | True | True | True | | |
| | E3_Orden_VENIR | Bool | %M11.0 | False | True | True | True | | |
| | Temp1 | Timer | %T1 | False | True | True | True | | |
| | E1_Enviar_NuevoCiclo | Bool | %M11.2 | False | True | True | True | | |
| | COM_Off | Bool | %M1.7 | False | True | True | True | | |
| | E1_Recibir_START_remoto | Bool | %M11.3 | False | True | True | True | | |
| | E1_Recibir_STOP_remoto | Bool | %M11.4 | False | True | True | True | | |
| | E3_Leer_Datos | Bool | %M11.1 | False | True | True | True | | |
| | E1_NuevoCiclo | Bool | %M11.5 | False | True | True | True | | |
| | ActivarT1 | Bool | %M11.6 | False | True | True | True | | |