

Práctica 1 – Introducción a la programación del autómatas S7-1200

1. Objetivos

- Conocer cómo se configura y utiliza un entorno de programación de PLCs.
- Saber cómo se organizan las variables y programas en un PLC.
- Ser capaz de realizar programas para PLCs mediante los lenguajes KOP y SCL.
- Se capaz de crear nuevos bloques de funciones y usarlos en un programa.
- Saber simular y depurar un programa, así como cargarlo y ejecutarlo en el PLC real.

2. Programación básica del S7-1200 con el software TIA Portal.

El PLC que se va a emplear en esta asignatura es el S7-1200 de Siemens. Este PLC es un controlador modular para soluciones de automatización discreta y proporciona la potencia y flexibilidad para controlar una amplia variedad de dispositivos de apoyo a necesidades de automatización. En concreto las características principales del PLC a emplear son:

- Posee 14 entradas digitales integradas (de 24 VDC), 10 salidas digitales integradas (por relé), 2 entradas analógicas y 2 salidas analógicas integradas.
- La CPU que posee es el modelo 1215C AC/DC/RLY (6ES7 215-1BG31-0XB0).
- Programación integrada mediante el entorno TIA Portal de Siemens (*Totally Integrated Automation Portal*) versión 15.1.



Figura 1.

A continuación, se describe los pasos a seguir mediante el entorno TIA Portal para la configuración y programación del S7-1200 que se posee en el laboratorio.

2.1. Creación de un proyecto.

Dentro de los programas disponibles en los PCs del laboratorio, hay que iniciar el TIA Portal (se encuentra en el escritorio):

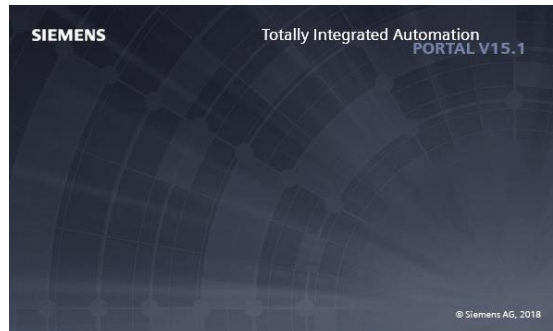


Figura 1. Inicio del software TIA Portal v15.1

En la ventana inicial pulsaremos en la opción "Crear proyecto" del panel central, que permitirá en el panel de la derecha introducir las propiedades básicas del proyecto (nombre, ruta, autor y comentarios), como se observa en la figura siguiente.

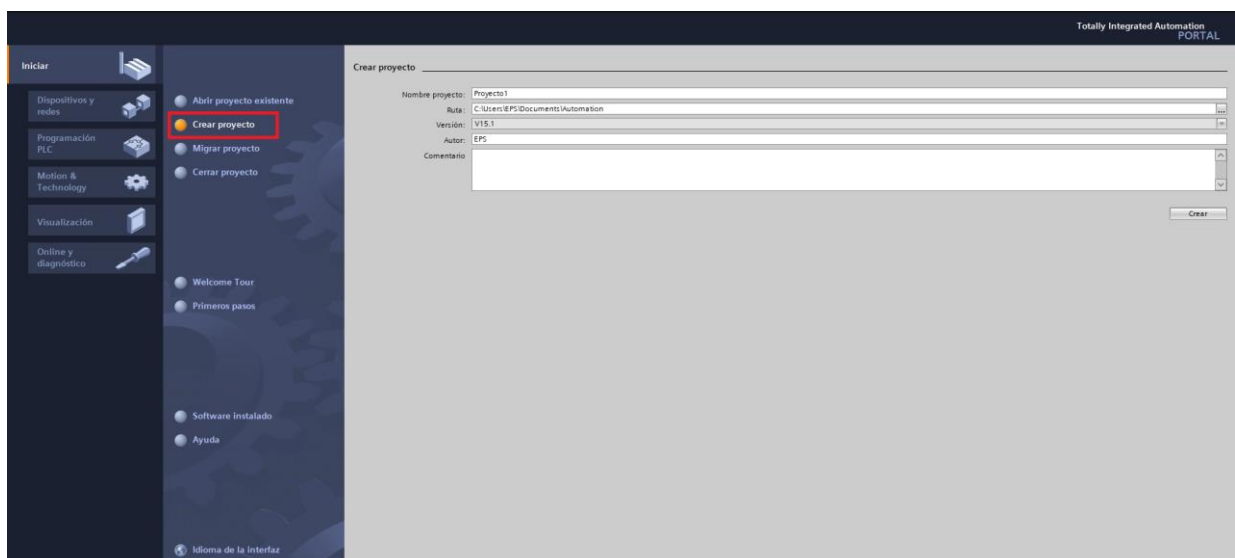


Figura 2. Creación de un proyecto

Al pulsar el botón "Crear", se crea el proyecto y aparece una interfaz de ayuda para la configuración del proyecto. A esta interfaz se le denomina "Vista del portal" (Figura 3):



Figura 3. Vista del portal para la configuración del proyecto

Para realizar programas con mayor comodidad se recomienda pasar a la "Vista del proyecto". Para ello pulsaremos el botón "Vista del proyecto" que aparece en la esquina inferior izquierda, mostrándose la ventana de la figura siguiente.

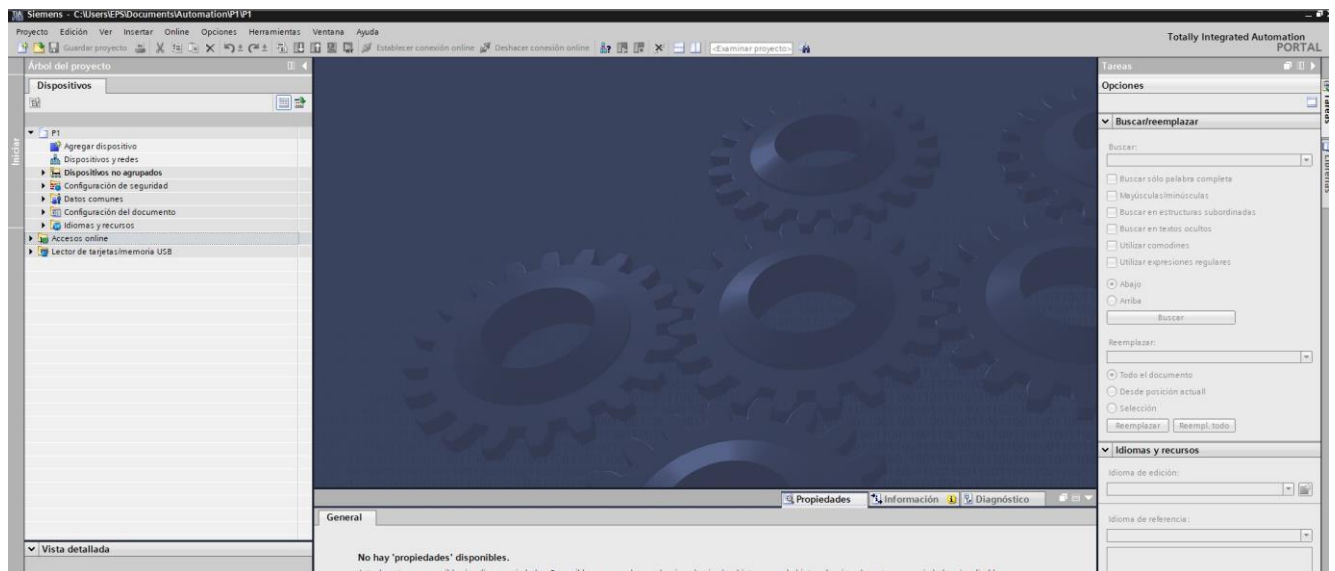


Figura 4. Vista del proyecto

2.2. Configuración del PLC.

Una vez que se ha creado un proyecto nuevo, todavía no existe ningún dispositivo asociado al mismo (en el panel de la izquierda, en el "Árbol del proyecto", no aparece ningún dispositivo). Los PLCs del laboratorio están conectados al PC a través de un cable y tarjeta de red. Para agregar el PLC al proyecto, en primer lugar, deberá alimentarse el PLC mediante el cable de alimentación, y conectarlo a un terminal de red de las mesas de trabajo del laboratorio. A continuación, se debe agregar el PLC al proyecto realizando doble click sobre la opción "Agregar dispositivo" del panel izquierdo.

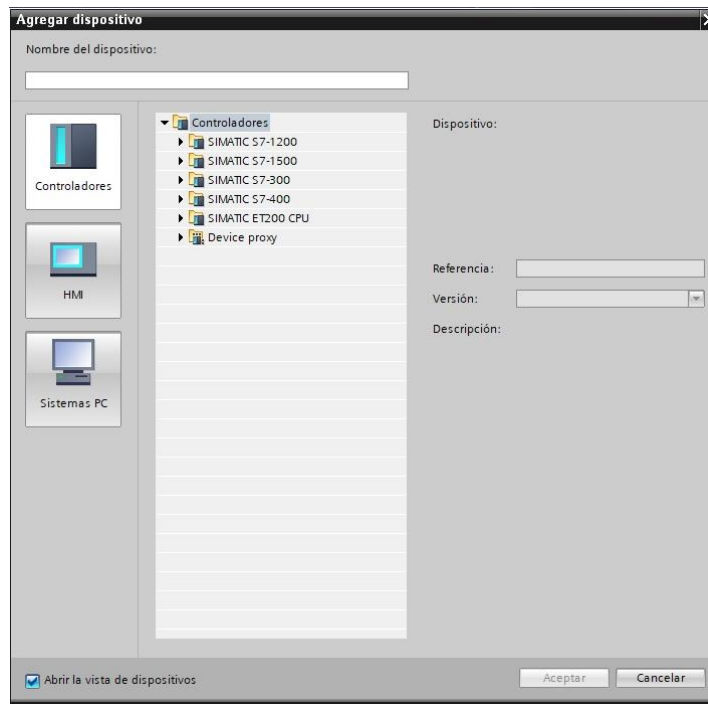


Figura 5. Agregar dispositivo PLC

En esta ventana, se selecciona el botón "Controladores" de la columna de la izquierda y aparecerá en la parte central un listado con todos los posibles modelos de autómatas que pueden conectarse. Existen 2 modos para agregar el autómata: dejando que TIA detecte automáticamente el modelo de autómata, o introduciéndolo manualmente. A continuación, se va a mostrar cómo se agrega el autómata con cada uno de estos modos.

- **Configuración manual del dispositivo.** Este modo es cuando conocemos la versión de CPU que incorpora el PLC. Para ello, en primer lugar, se elige el modelo de autómata en la parte central de la ventana anterior: **SIMATIC S7-1200**, apareciendo un listado con todas las posibles CPUs para este modelo. El modelo concreto de CPU puede leerse en el lateral derecho del propio autómata. Para los PLCs del laboratorio, el modelo de CPU es "CPU 1215C AC/DC/Rly", con modelo "6ES7 215-1BG40-0XB0". Se selecciona esta CPU, y se pone como versión del firmware la opción 4.2.

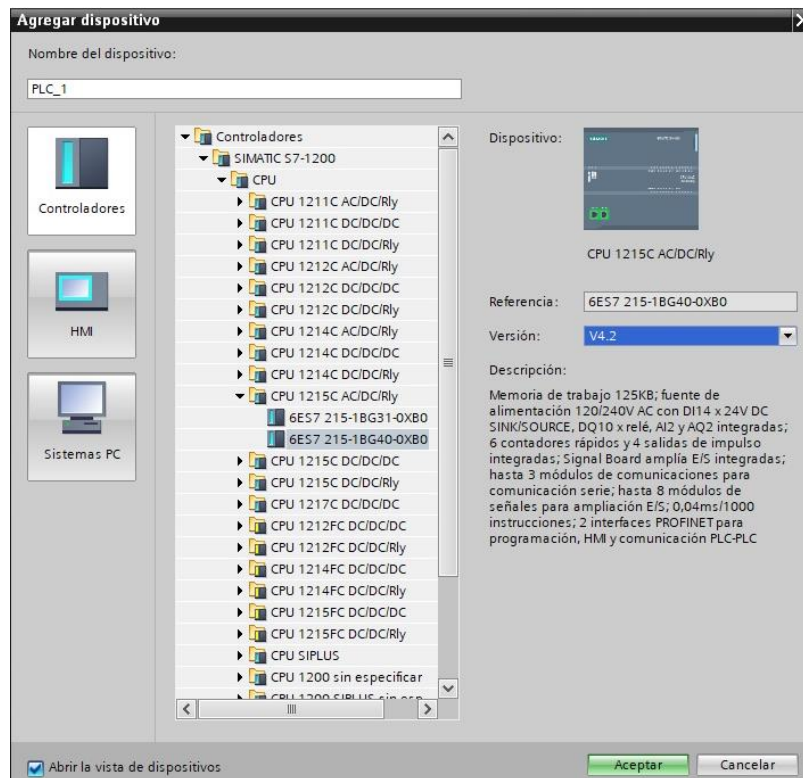


Figura 6. Agregar CPU y modelo del PLC

En la parte superior de la ventana anterior se puede escribir un nombre para el dispositivo. Dado que van a existir en red varios PLCs, lo más conveniente es poner a cada uno un nombre diferente. Por defecto, dejado el nombre PLC_1, que se deberá variar en función del PLC conectado a la red. Al terminar la configuración, se pulsa a “Aceptar” y se mostrará la vista del proyecto con el PLC ya agregado (Figura 7).

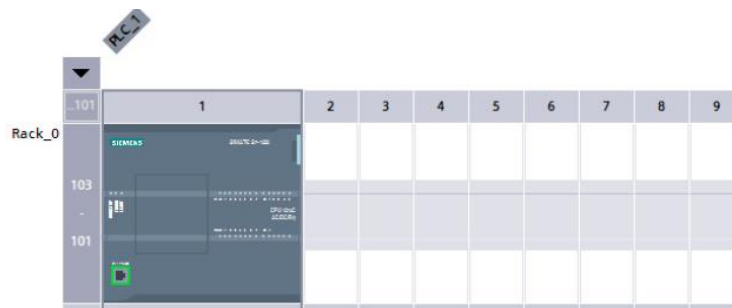


Figura 7. PLC configurado como dispositivo

- **Detección automática del dispositivo.** Para este caso, se selecciona modelo de autómatas “CPU 1200 sin especificar” (en la ventana “Agregar dispositivo”).

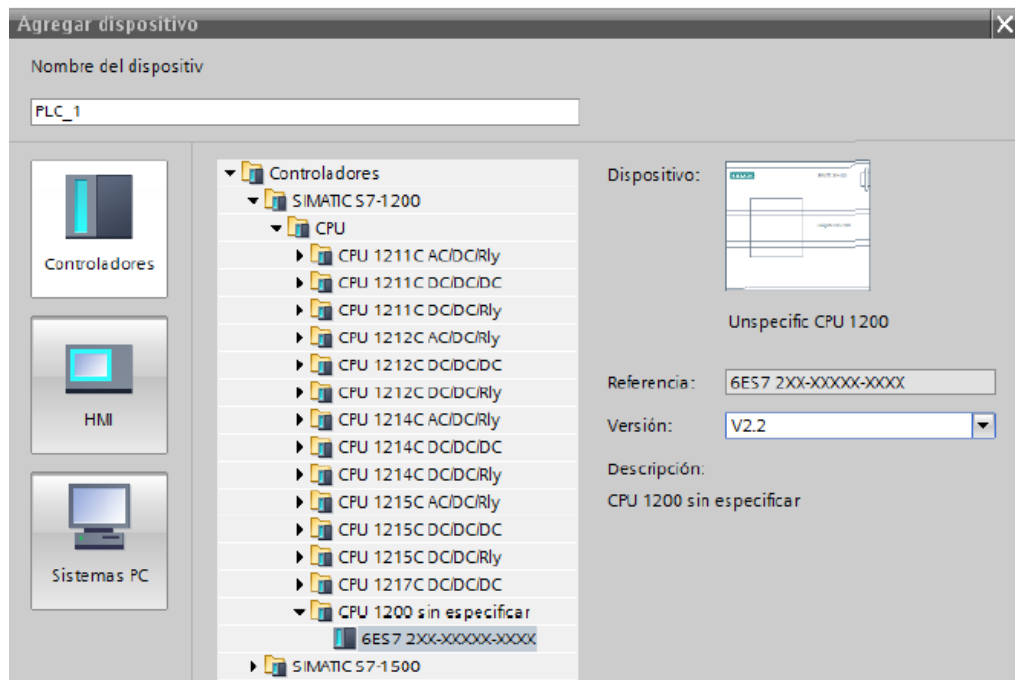


Figura 8. Configuración automática del dispositivo (I)

En esta ventana se selecciona cualquier versión del firmware ya que el programa la detectará, aunque es conveniente elegir la versión más baja de las que aparezca en la lista. Al pulsar el botón Aceptar, aparece la siguiente imagen en la Vista de dispositivo:

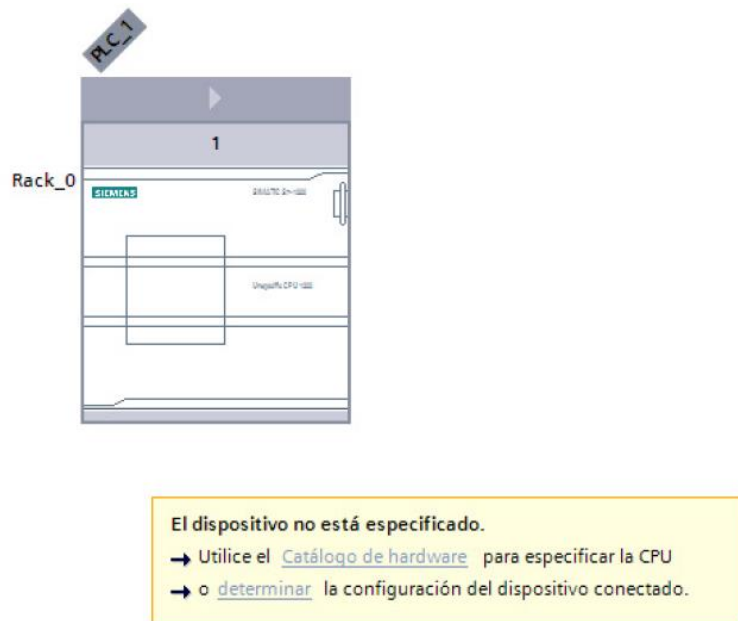


Figura 9. Configuración automática del dispositivo (II)

En esta vista, se debe seleccionar la opción “determinar la configuración del dispositivo conectado” (en el cuadro amarillo de la parte central). A continuación, tendrá que aparecer la ventana de “Detección de hardware”. Aquí, hay que configurar las opciones del interfaz, es decir, indicar al programa cómo está conectado el autómata al ordenador. En el caso de los PLCs del laboratorio están conectados a través de una tarjeta de red (habrá que elegir el tipo de interfaz PN/IE y la tarjeta concreta que esté instalada). La configuración final que se debe introducir es la siguiente:

- Tipo de interfaz PG/PC: **PN/IE**.
- Interfaz PG/PC: **Realtek PCIe GBE Family Controller**

Finalmente, se detecta el PLC y se pulsa el botón “Detección”. Una vez hecho esto, el programa volverá a la vista de dispositivo con el autómata detectado.

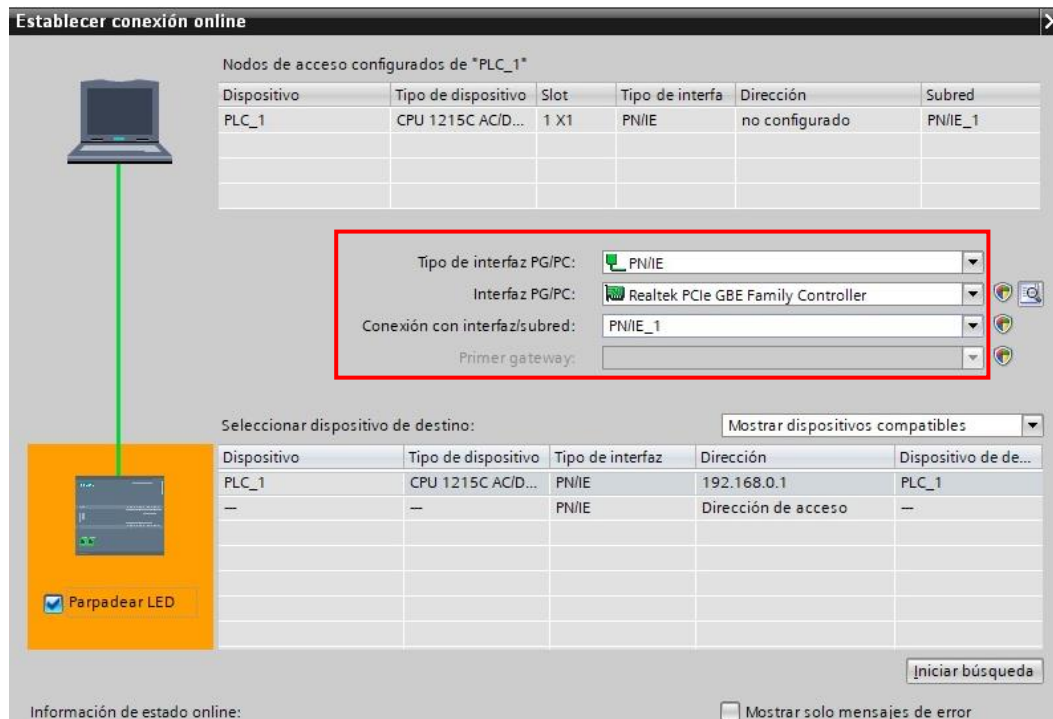


Figura 10. Configuración del PLC

2.3. Creación de un programa ejemplo

Para este primer programa, se trabajará con el lenguaje de programación KOP. De este modo, los elementos que aparecerán en el programa serán contactos, bobinas y bloques (temporizadores, contadores, etc). Dado que en la parte teórica no se ha profundizado en los lenguajes de programación, se ha realizado un pequeño anexo de explicación del lenguaje KOP.

Para crear nuestro programa, se accede a la opción "Bloques de programa" del menú de la izquierda. En esta opción existe la posibilidad de "Agregar un nuevo bloque" o acceder directamente al bloque "Main [OB1]" que ya está creado, pero vacío. Este es el bloque principal del programa del PLC, que se ejecuta cíclicamente como se ha explicado en clase de teoría.

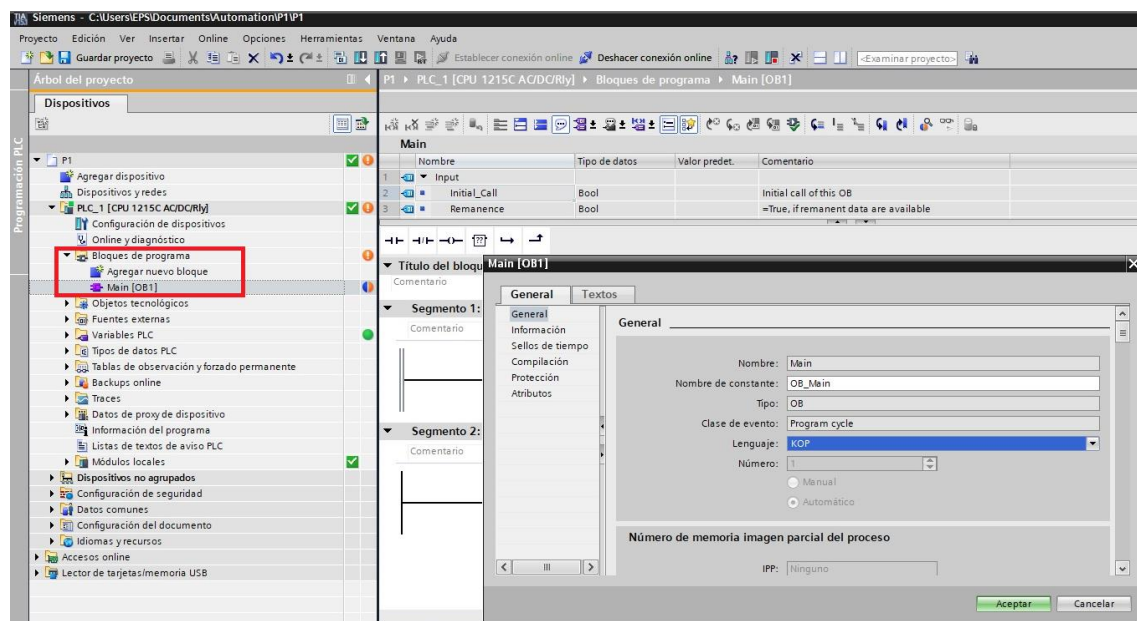


Figura 11. Creación de un programa en KOP (I)

En este ejemplo escribiremos el programa directamente sobre el main que ya está creado por defecto, así que haremos doble click sobre "Main [OB1]" y se abrirá el editor de KOP en la parte central. Las instrucciones del programa (contactos, bobinas, etc.) pueden arrastrarse directamente con el ratón, bien desde la barra de instrucciones de uso frecuente (encima del "Título del bloque"), o bien desde el panel de la derecha, donde aparecen todas las instrucciones (Figura 12).

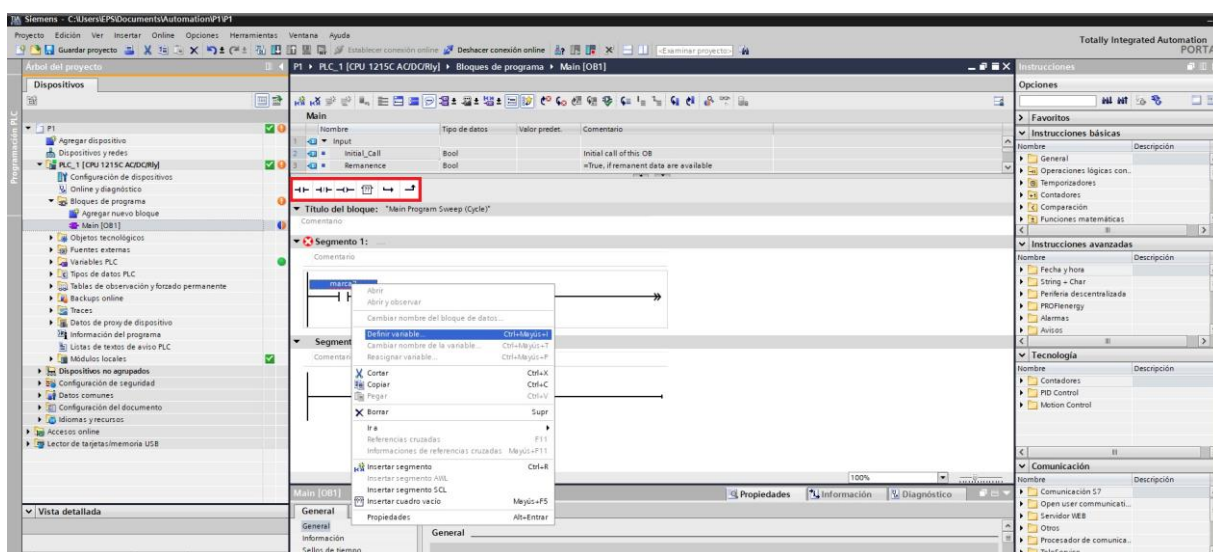
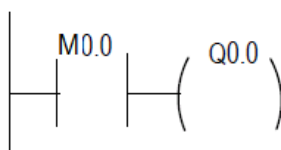


Figura 12. Creación de un programa en KOP (II)

Se creará el siguiente programa:



El programa indica que la salida Q0.0 estará activa cuando lo esté la variable M0.0 (M es la parte de memoria del PLC). Cuando se arrastra hacia el programa con el ratón cualquier operación

(contacto, bobina, etc.), aparece un pequeño cuadrado de color verde indicando lugares donde puede colocarse dicha operación en un segmento. Se recomienda practicar con el editor para familiarizarse con su funcionamiento y ver otras posibilidades de este. Por ejemplo, también pueden introducirse operaciones seleccionándolas primero con el ratón y pulsando después sobre el punto del programa donde se desea colocar.

Antes de dibujar el programa es conveniente crear una tabla de variables en la que se asocie una etiqueta a cada entrada, salida y variable que usemos inicialmente en nuestro programa. Sin embargo, en este primer ejemplo dibujaremos directamente el programa sin crear la tabla de variables (posteriormente se verá cómo crearla). Una vez introducido el componente, y habiéndole dado un nombre, hay que asignarle si pertenece a las entradas (Global Input - %I), salidas (Global Output - %Q), o la memoria interna del PLC (Global Memory - %M). Para ello, se pulsa botón derecho sobre la variable no asignada y se le da una parte de la memoria.



Figura 13. Asignación de la sección a las variables

En la figura siguiente se muestra el aspecto de la ventana con el programa ya introducido.



Figura 14. Programa introducido en el software

Esta asignación realizada sobre los segmentos hace crear de forma automática la tabla de variables. Si se desea visualizar dicha tabla, se encuentra dentro de *Variables del PLC* → *Tabla de variables estándar* (Figura 15).

Siemens - C:\Users\EPS\Documents\Automation\IP1\IP1

Proyecto Edición Ver Insertar Online Opciones Herramientas Ventana Ayuda

Guardar proyecto Establecer conexión online Deshacer conexión online Examinar proyecto

Árbol del proyecto P1 PLC_1 [CPU 1215C AC/DC/Rly] Variables PLC Tabla de variables estándar [33]

Dispositivos

Programación PLC

Variables

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Comentario
1	marca1	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	salida	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	marca2	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	salida1	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	<Agregar>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 15. Tabla de variables del PLC

2.4. Carga del programa en el PLC.

El programa se ha creado en el TIA Portal, y es necesario enviarlo a la memoria del autómata. Para ello es recomendable primero compilar el programa en el PC y, posteriormente, cargarlo en el autómata. Aunque en el menú de la parte superior existen botones directos para compilar y cargar el programa, la primera vez que se compila y carga el programa de un proyecto es recomendable hacerlo como se indica a continuación, para evitar errores de inconsistencias entre la configuración del proyecto y las características del autómata concreto. Las cargas posteriores del programa del proyecto sí pueden hacerse más cómodamente desde los botones del menú superior.

Para ello, se pulsa el botón derecho del ratón sobre la opción del panel izquierdo correspondiente al autómata ("PLC_1 ..."), y se selecciona la opción "Compilar->Hardware y software (solo cambios)", como se muestra en la figura siguiente.

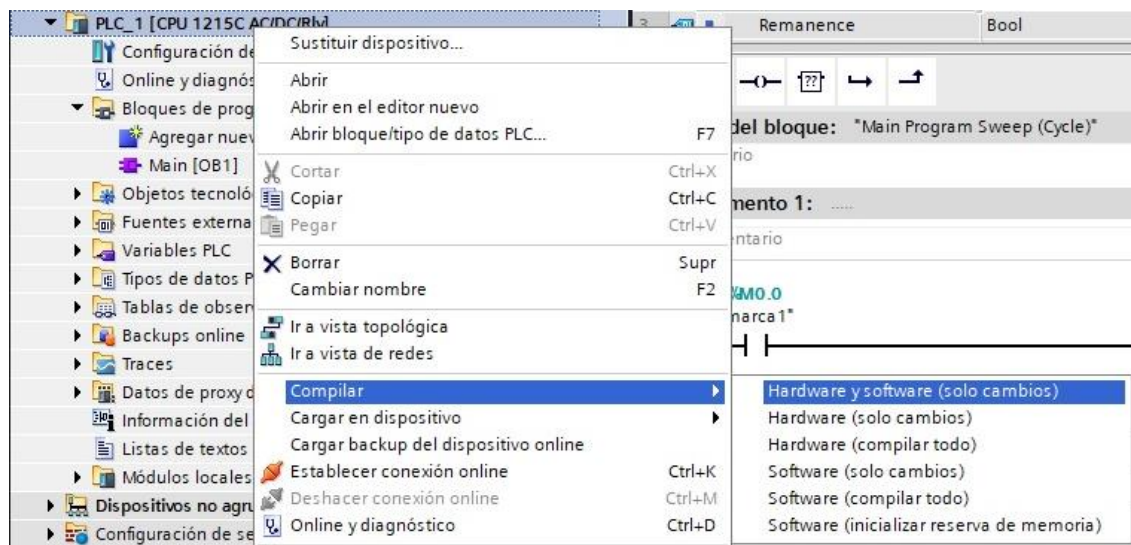


Figura 16. Compilación del programa

2.5. Comprobación del funcionamiento del programa.

Si no se ha producido ningún error, en la ventana inferior central aparecerán mensajes indicando que el software y hardware se han compilado correctamente. A continuación, se carga el programa en el autómata, pulsando el botón derecho del ratón sobre la opción del panel izquierdo correspondiente al autómata ("PLC_1..."), y seleccionando la opción "Cargar en dispositivo->Hardware y software (solo cambios)". Aparecerá una ventana nueva de en la que se debe seleccionar el tipo de interfaz entre el ordenador y el autómata ("Tipo de interfaz PG/PC") y la tarjeta concreta que se está utilizando en el ordenador ("Interfaz PG/PC").

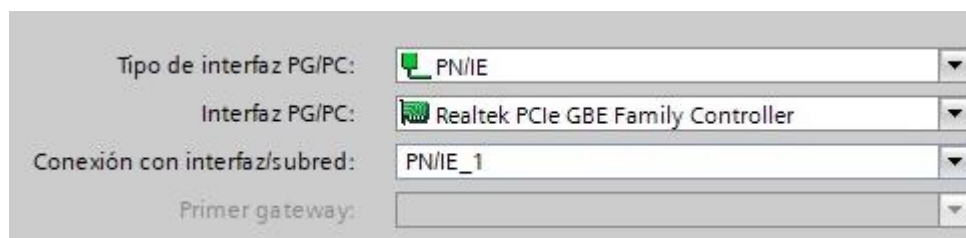


Figura 17. Configuración de la conexión on-line

En el siguiente paso, se pulsa el botón "Iniciar búsqueda" y, después de un tiempo, se muestra en la parte inferior el PLC encontrado en la red, como se puede ver en la figura siguiente.

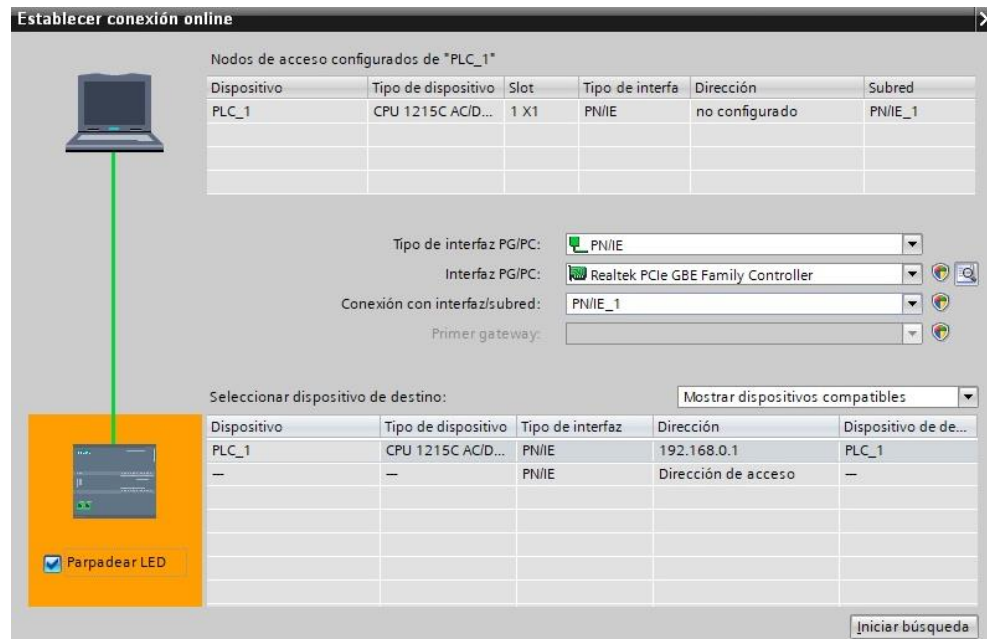


Figura 18. Configuración de la conexión on-line

A continuación, se pulsa el botón "Cargar". Al iniciar la carga del programa es posible que se encuentre algún tipo de error. Por ejemplo, es frecuente que el autómata esté en modo "Run" cuando queremos cargar el programa, y debería estar en modo "Stop". Si ocurre esto, aparecerá una ventana como la que se muestra a continuación.

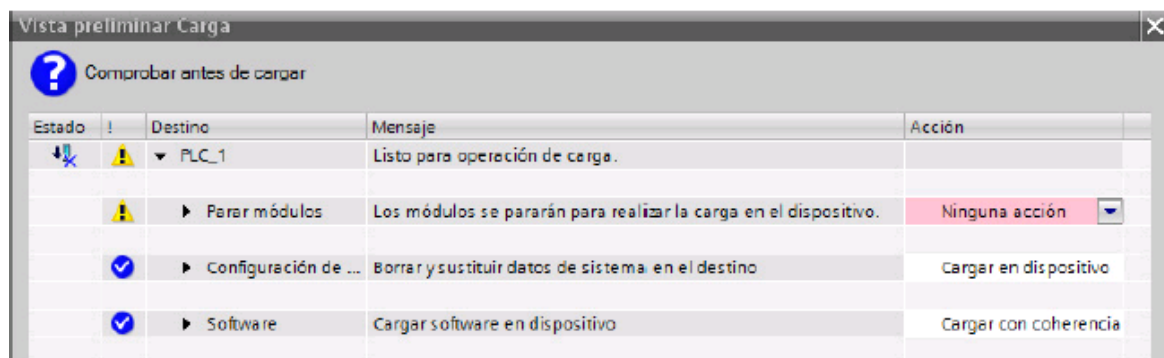


Figura 19. Carga del programa

En esta ventana, se debe elegir la acción "Parar todos" en el menú de acciones del mensaje "Parar módulos", para que se paren todos los módulos activos del autómata y pueda cargarse el programa.

Una vez puesto en marcha el programa, se probará el funcionamiento del mismo desde el software. Para ello, se forzará la marca a un valor de 1 emulando una entrada física del PLC (pulsando botón derecho sobre la variable). A continuación, se deberá activar la salida física del PLC, activando su relé de salida. Esto es debido a que el programa indica que la salida Q0.0 tomará el valor que la marca M0.0.

El procedimiento de prueba de programas desde el software será siempre el mismo, mientras no se conecte el autómata a un sistema físico. Cuando se prueben programas más complejos será necesario simular mediante los interruptores el funcionamiento de entradas cualesquiera (finales de

carrera, sensores de proximidad sin contacto, botoneras, etc); y se deberá interpretar el encendido o apagado de los leds como puesta en marcha o paro de salidas cualesquiera (motores, electroválvulas, calentadores, etc).

2.6. Creación de tablas de variables. Tabla de forzado.

En el programa desarrollado con anterioridad, como entrada se ha utilizado la memoria del PLC. Si se desea utilizar entradas de la parte de Global Input (%I) no es tan evidente la función de forzado, ya que dichas entradas supuestamente están conectadas a la entrada física. Si no se tiene nada conectado a las entradas reales, y se desea forzar el valor, es necesario crear una tabla de forzado permanente. Para ello, en la parte del menú de *Tablas de observación y forzado permanente*, se debe crear las variables conectadas a las entradas físicas que se desean forzar.

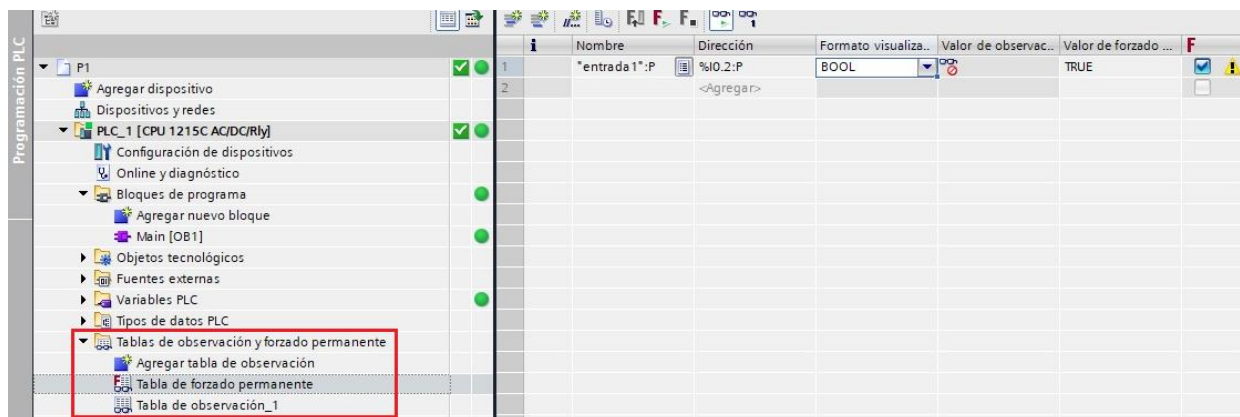


Figura 20. Tabla de forzado permanente

2.7. Activación de funciones especiales.

EL PLC S7-1200 posee una serie de funciones especiales (funciones de estado y control que emplean la memoria interna del PLC), como por ejemplo la activación en forma de parpadeo de salidas. A continuación, se explica cómo activar estas funciones especiales mediante la escritura en la memoria del PLC S7-1200:

1. Botón derecho sobre el dispositivo PLC del árbol del proyecto y, en el menú contextual que aparece, se selecciona "Propiedades".
2. En esta ventana, en la solapa General y, en el menú de la izquierda, se selecciona "Marcas de sistema y de ciclo", como se muestra en la figura siguiente:

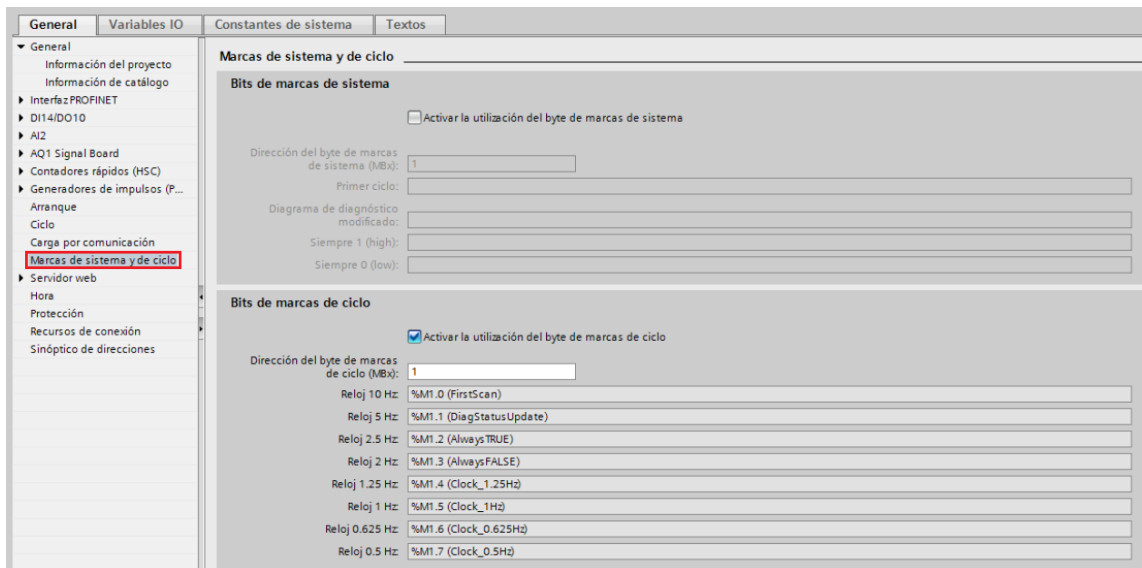


Figura 21. Marcas especiales

- En la parte de la derecha, se puede seleccionar los bits "especiales" que deseamos usar en el programa. Para ciclos de parpadeo, se debe seleccionar "Activar la utilización del byte de marcas de ciclo", como se muestra en la figura 19.
- En "Dirección de byte de marcas de ciclo", se debe indicar en qué dirección de memoria queremos que comience el byte de marcas de ciclo. En este punto debemos tener cuidado seleccionando un byte que no usemos como marcas auxiliares o posiciones de memoria en cualquier punto del programa.

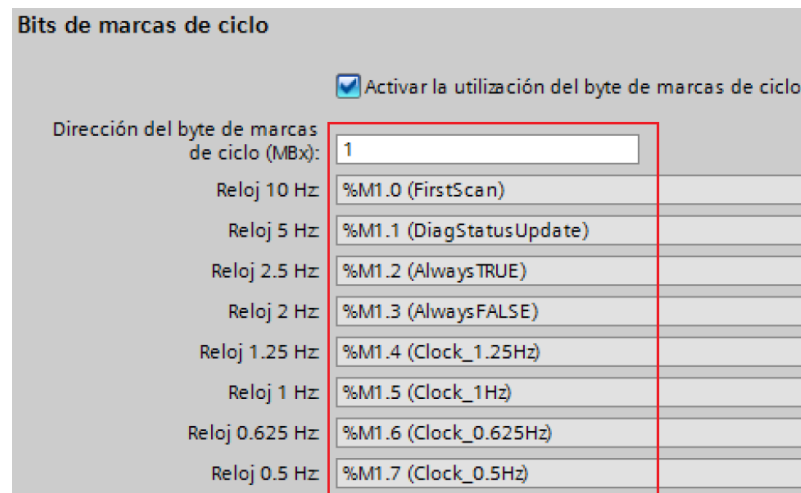


Figura 22. Memoria de 1 byte (M1.0) asignada a las marcas especiales

2.8. Desarrollo de una interfaz HMI.

Un dispositivo HMI (*Human Machine Interface*) proporciona una interfaz para poder monitorizar y controlar las variables del sistema a controlar. Para el desarrollo de una interfaz gráfica con el TIA Portal, se deben seguir los siguientes pasos:

1. Seleccionar la pantalla HMI a utilizar, agregando un dispositivo nuevo (*Insertar → Dispositivo*). En nuestro caso, no se posee las pantallas físicas, por lo que se seleccionará

una básica para desarrollar una interfaz sencilla. En concreto, se puede seleccionar cualquiera de la KTP700 Basic.

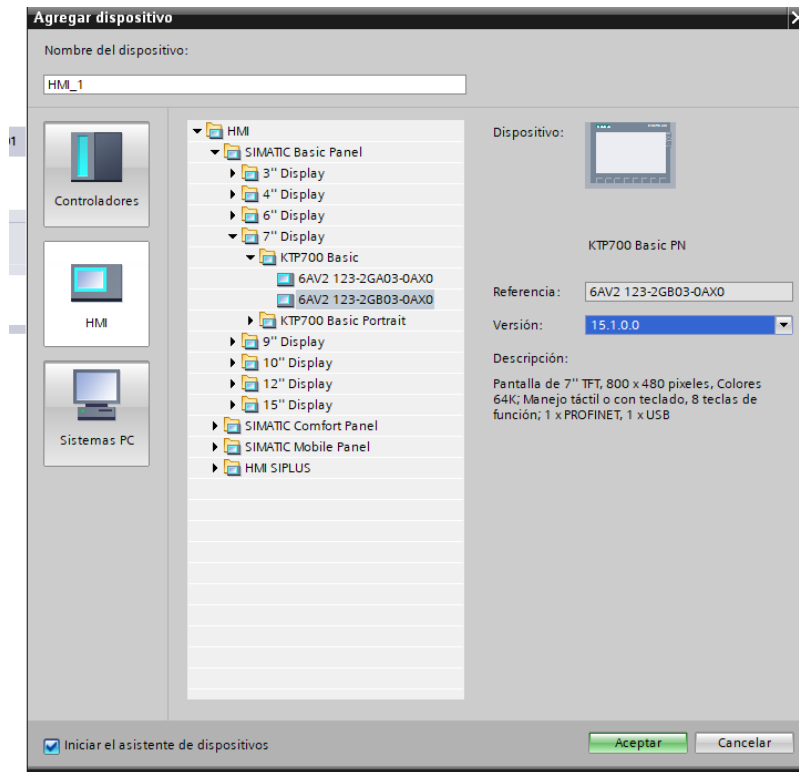


Figura 23. Pantalla HMI

2. En segundo lugar, se debe configurar la pantalla (comunicación, resolución, formato de imagen, etc) a través de una interfaz de configuración:

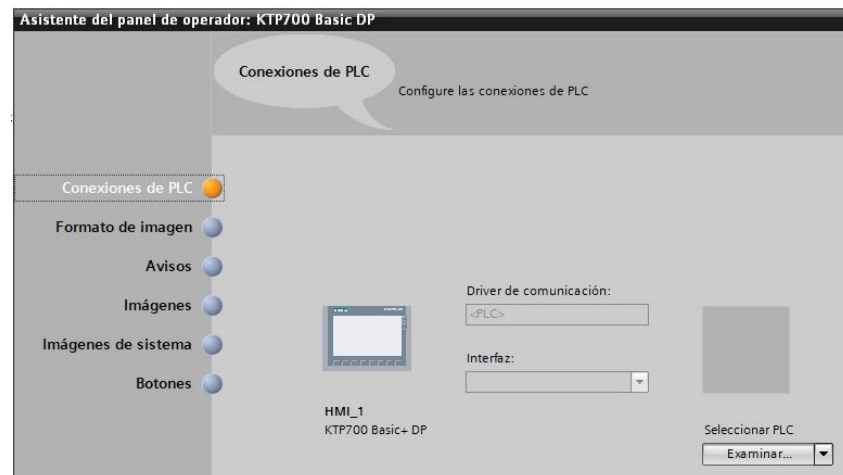


Figura 24. Configuración de la pantalla.

3. A continuación, cuando se crea la interfaz preparada para introducir elementos (botones, campos de texto, etc.) en la parte de la imagen raíz del dispositivo introducido.

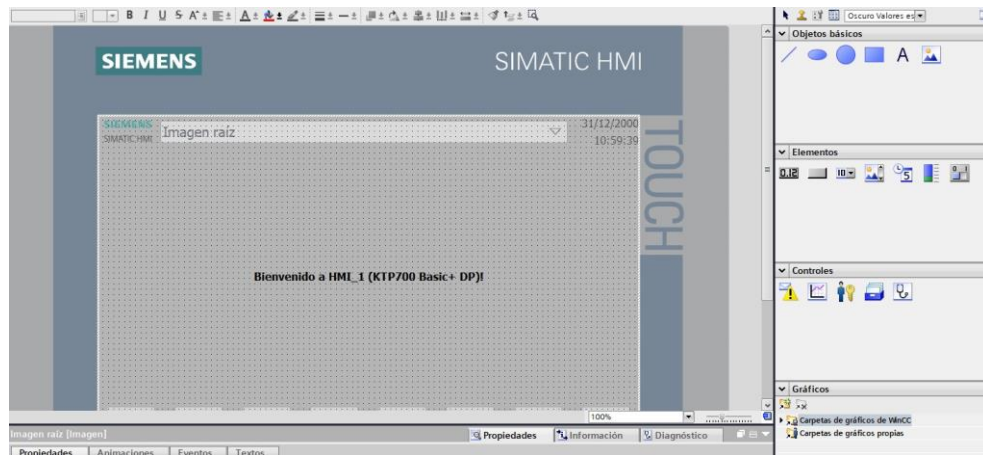


Figura 25. Configuración de la pantalla.

4. Para poder conectar las variables de la pantalla con el PLC, es necesario crear una conexión entre el PLC y el dispositivo HMI. Para ello, se crea una nueva conexión desde el HMI, dentro de la opción de “Conexiones” en la parte de Variables HMI.

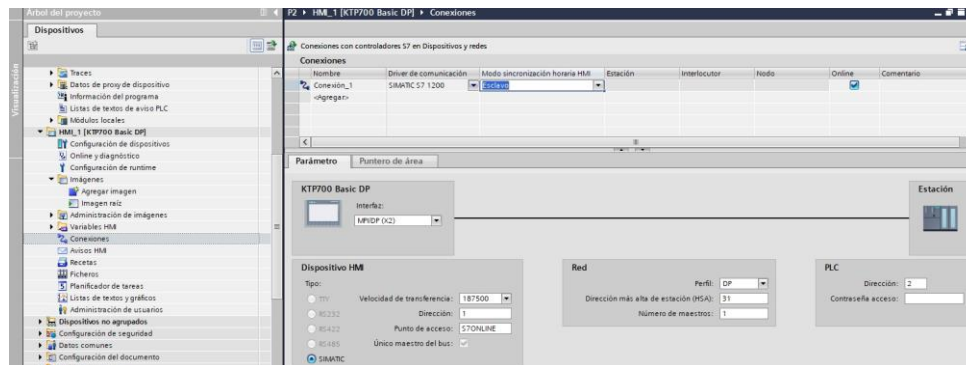


Figura 26. Creación de una conexión con el PLC.

5. Una vez conectados a través de la red, se conectarían los controles de la interfaz (botones, campos, etc.) con las variables del PLC. Para ello, se crean eventos asociados a variables del PLC. A continuación, se muestra una imagen de un campo numérico, asociado a la variable del temporizador ET.

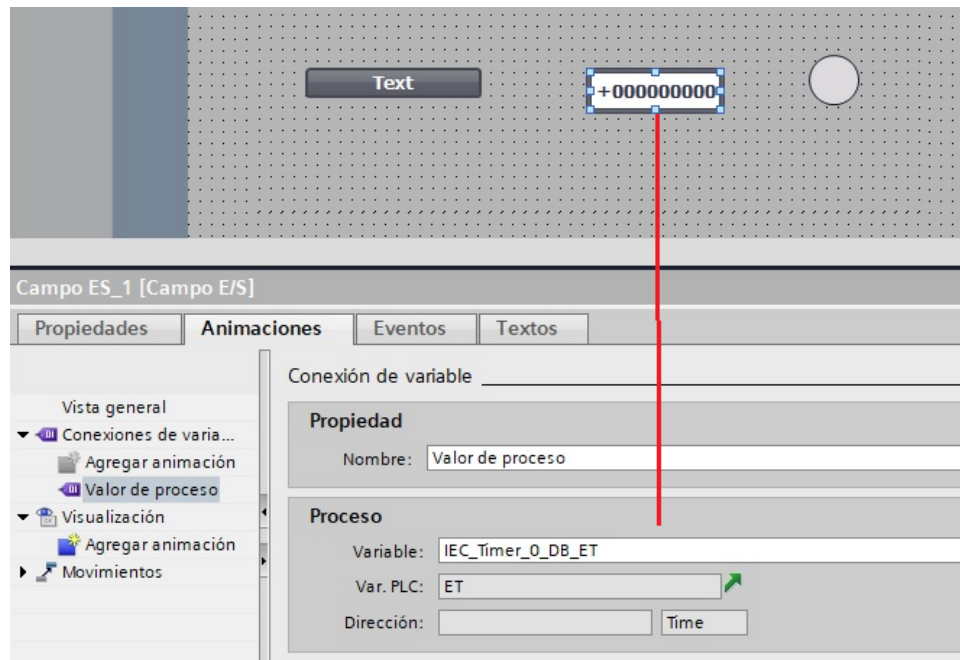


Figura 27. Asociación de una variable del PLC al campo numérico.

3. Estructura del programa en el S7-1200

Los programas para la CPU de los autómatas S7-1200 pueden tener los siguientes tipos de bloques de código:

- Bloques de organización (**OB**), que definen la estructura del programa.
- Funciones (**FC**) y bloques de funciones (**FB**), que contienen el código que se corresponde con tareas específicas.
- Bloques de datos (**DB**), que almacenan datos que pueden ser usados por el resto de bloques de programas.

La ejecución del programa de usuario comienza con uno o más OB opcionales de inicialización que se ejecutan una única vez cuando la CPU pasa a modo de ejecución (RUN), seguidos por la ejecución de uno o más OB cíclicos. Un OB puede estar también asociado a un evento de interrupción, que puede ser un evento estándar o un evento de error. En estos casos, el OB se ejecutará cuando ocurra el evento correspondiente.

Una función (FC) o un bloque de función (FB) es un bloque de código que puede ser llamado desde un OB o desde otra FC o FB. Las FC no están asociadas con ningún bloque de datos (DB) concreto, mientras que las FB sí están asociadas directamente con un DB, que lo usan para pasar parámetros y almacenar resultados de operaciones.

Si se desea crear un bloque de función FC, se agrega un nuevo bloque de programa como función FC. Este bloque puede programarse en cualquier lenguaje compatible con el S7-1200.

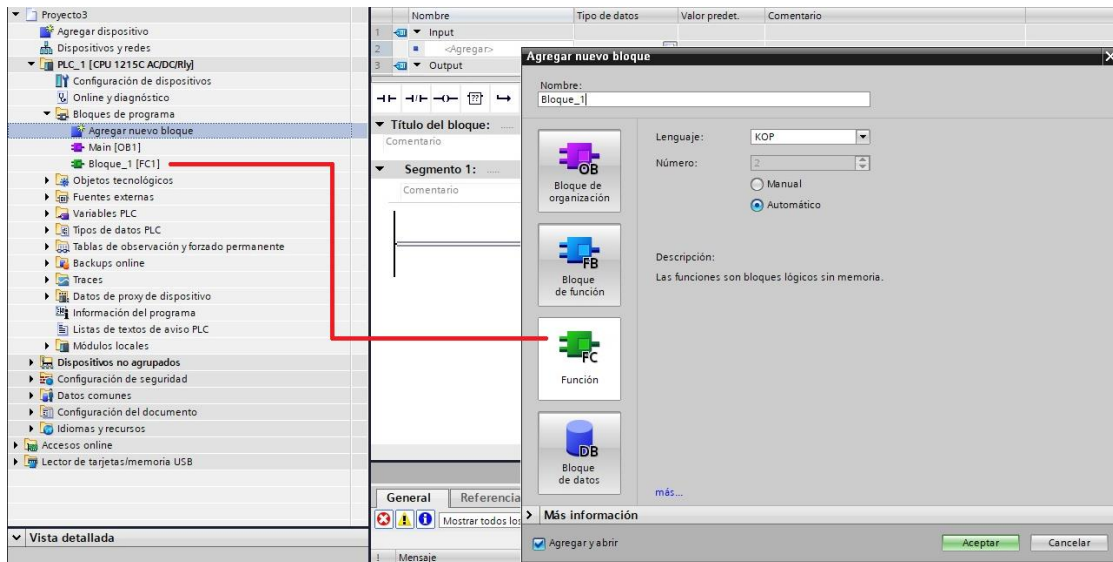


Figura 28. Creación de un bloque FC.

Posteriormente, para llamar a esta función desde el main de la aplicación se introduce un bloque de función en el que se inserta el nombre del bloque FC creado. De

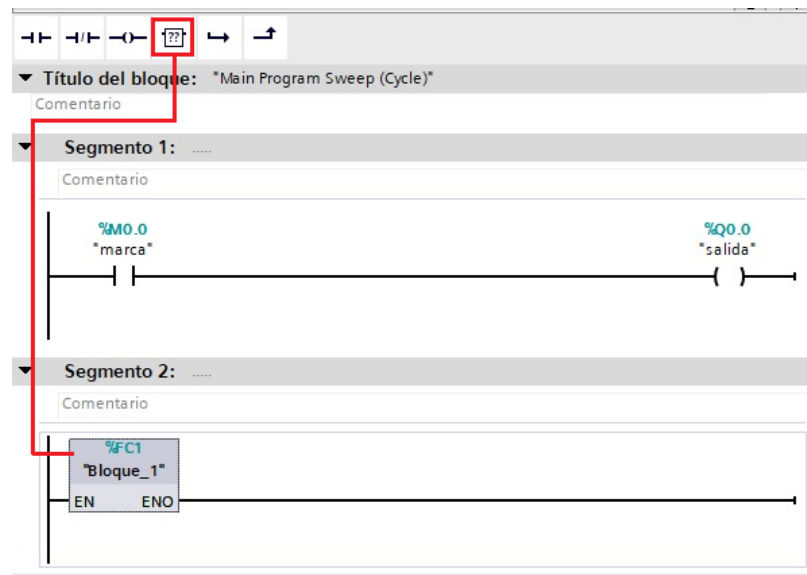


Figura 29. Llamada a un bloque FC.

4. Ejercicios de la práctica.

Ejercicio 1: CONTROL DE UN MOTOR DESDE UN PANEL DE MANDO

Las entradas y salidas del sistema para el control del motor que se encuentran conectadas al PLC son:

Descripción	Área de memoria	Estados
Interruptor On/Off	%M 0.0	On = 24v, Off = 0v.
Palanca de giro positivo motor	%M 0.1	Marcha = 24v, Paro = 0v.
Palanca de giro negativo motor	%M 0.2	Marcha = 24v, Paro = 0v.
Lámpara funcionamiento	%M 10.0	Encendido = 24v, Apagado = 0v.
Lámpara sentido de positivo giro	%M 10.1	Encendido = 24v, Apagado = 0v.
Lámpara sentido negativo de giro.	%M 10.2	Encendido = 24v, Apagado = 0v.

El motor se conecta al autómatas mediante dos contactores que lo conectan a la red de modo que gire en un sentido u otro, y que se comandan desde las salidas M10.3 y M10.4 del PLC, según aparece en la siguiente tabla:

Descripción	Área de memoria	Estados
Contactor giro positivo motor	%M 10.3	Marcha = 24v, Paro = 0v.
Contactor giro negativo motor	%M 10.4	Marcha = 24v, Paro = 0v.

El funcionamiento del sistema debe ser el siguiente:

- El interruptor ON/OFF pone en marcha o para el sistema y enciende o apaga la lámpara de funcionamiento.
- Si el interruptor está en ON y se actúa sobre la palanca de giro positivo, el motor gira a derechas y se activa la lámpara indicativa de ese sentido de giro.
- Si el interruptor está en ON y se actúa sobre la palanca de giro negativo, el motor gira a izquierdas y se activa la lámpara indicativa de ese sentido de giro.
- Si, con el interruptor en ON, se actúa sobre ambas palancas, el motor no girará en ningún sentido, pero sí se encenderán las dos lámparas indicativas del sentido de giro.

Tareas a realizar:

- Desarrollo del programa en KOP para el control del motor.
- Interfaz HMI para la monitorización y control del motor.

Parte optativa:

- Se añade un sensor para medir la temperatura en el bobinado del motor. Cuando la temperatura sea excesiva, el motor debe detenerse y se debe encender un piloto en el panel de mando; una vez restablecida la temperatura normal se deberá apagar el piloto y el motor

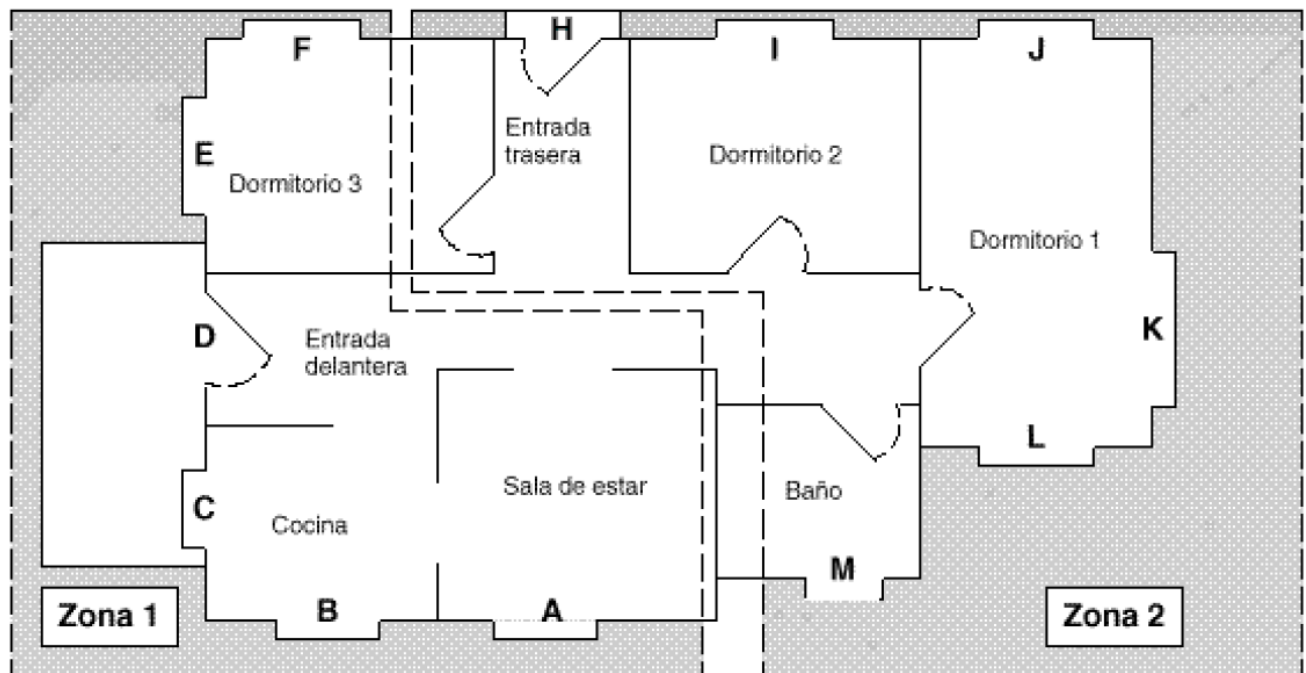
continuará moviéndose. Las conexiones de sensor y piloto con el autómata se muestran en la tabla siguiente:

Descripción	Área de memoria	Estados
Sensor temperatura	%M 0.3	Temperatura OK = 24v, alta = 0v.
Piloto panel de mando	%M 10.5	Encendido = 24v, Apagado = 0v.

Se pide modificar el programa anterior y la interfaz para que contemple estos nuevos elementos y probar el resultado sobre el autómata.

Ejercicio 2: CONTROL DE UN SISTEMA DE ALARMA DE UNA VIVIENDA

Se pretende diseñar el programa de control de un sistema de alarma de una vivienda, como la que se muestra en la figura siguiente. En el programa se vigilan dos zonas (Zona 1 y Zona 2), de manera que cuando se irrumpe en una de ellas, se dispara una alarma.



Entradas:

- La entrada 1 (%M0.0) vigila la zona 1 (entrada delantera, sala de estar, cocina y dormitorio 3). Entrada normalmente cerrada. (Abierto="0", Cerrado="1").
- La entrada 2 (%M0.1) vigila la zona 2 (dormitorio 1, dormitorio 2, baño y entrada trasera). Entrada normalmente cerrada. (Abierto="0", Cerrado="1").
- La entrada 3 (%M0.2) activa o desactiva el sistema de alarma. Activa = "1", Desactivado = "0".
- La entrada 4 (%M 0.3) permite activar manualmente la sirena de alarma. Entrada normalmente abierta. Activa = "1", Desactivado = "0".

Salidas:

- La salida 1 (%M10.0) controla el LED del sistema de alarma. Estará encendido si está activado y parpadeante si está desactivado, estando abiertas la zona 1 o la zona 2.
- La salida 2 (%M10.1) dispara la sirena de alarma.
- La salida 3 (%M10.2) activa una señal de alerta baja que indica que la alarma se disparará al cabo de un número predeterminado de segundos.
- La salida 4 (%M10.3) activa un relé de interface externo (p.ej. para arrancar una marcación automática).

Funcionamiento del programa de control

- Si el sistema no está activado, el LED (%M10.0) parpadea al estar abiertas la zona 1 (%M0.0) o 2 (%M0.1).
- Si el sistema está activado (girando la llave a la posición “on”, lo que activa la entrada %M0.2), el programa arranca un temporizador de retardo de 90 segundos para que el propietario pueda salir de la vivienda. Durante ese tiempo de retardo, el programa no reacciona si se abre alguna de las zonas (%M0.0 ó %M0.1).
- Si el sistema está activado y ha transcurrido el tiempo de retardo para salir de la vivienda, el programa evalúa el estado de ambas zonas. Si se abre alguna de ellas (%M0.0 ó %M0.1), el programa arranca una secuencia de notificación que activa la señal de alerta baja (%M10.2) y arranca un temporizador. Ello le recuerda al propietario que debe desactivar el sistema de alarma al regresar a casa.
- Una vez arrancada la secuencia de notificación, el programa tiene dos opciones:
- Si se desactiva el sistema (girando la llave a la posición “off”, lo que desactiva %M0.2), el programa pone a “0” las salidas (%M10.0 y %M10.2) y los temporizadores.
- Si el sistema no se desactiva al cabo de 60 segundos a más tardar, el programa dispara la alarma y activa la marcación automática del módem (%M0.1 y %M0.3).
- Si se activa la alarma manual (%M0.3), el programa dispara la alarma y activa la marcación automática del módem (%M10.1 y %M10.3). Esta tarea se realiza independientemente de la posición del interruptor que activa o desactiva el sistema de alarma (%M0.2) y no ejecuta la secuencia de notificación que ofrece un tiempo de retardo para desactivar el sistema.
- Si se desactiva el sistema (girando la llave a la posición “off”, lo que desactiva %M0.2) una vez disparada la alarma (%M0.1), el programa pone a “0” las salidas (%M0.1 y %M0.3) y los temporizadores.

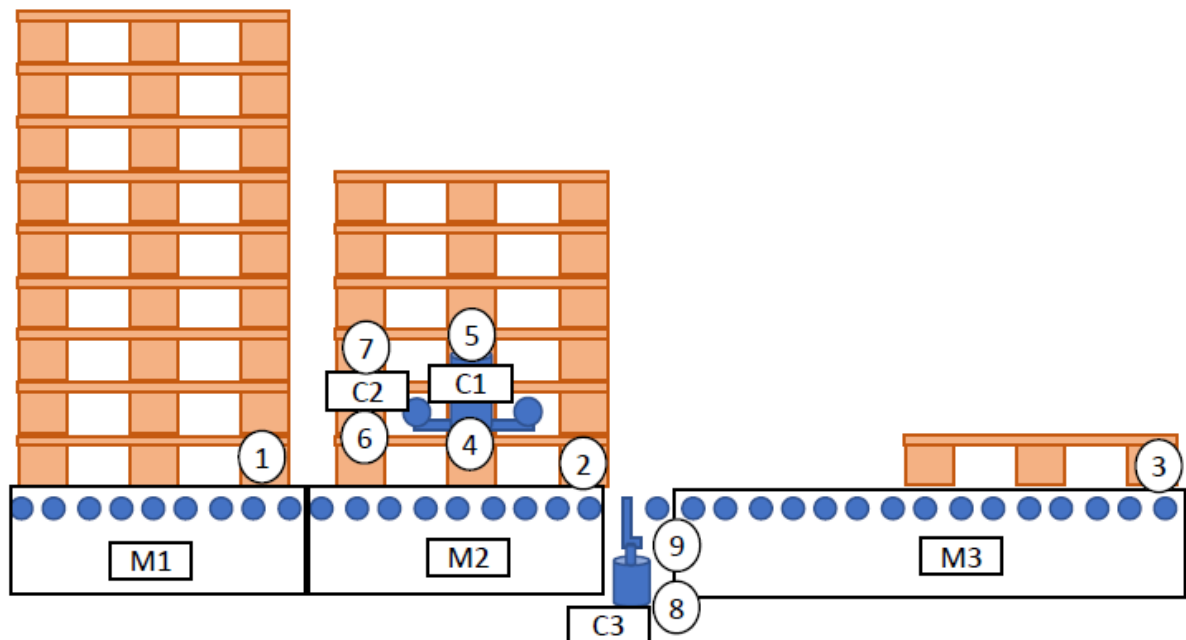
El programa deberá utilizar las marcas internas (memoria M) para almacenar los estados intermedios de la lógica por lo que respecta a las salidas físicas. Una vez evaluada la lógica de control, el programa usa los estados de dichas marcas para activar o desactivar las salidas.

Tareas a realizar:

- Desarrollo del programa en KOP.
- Interfaz HMI para la monitorización y control.

Ejercicio 3: CONTROL DE UN DESAPILADOR AUTOMÁTICO

Un dispositivo automático destinado a desapilar pales de forma automática. Recibe pilas de pales vacíos y los desapila, suministrándolos a una célula robotizada de paletizado.



–Un pulsador de Marcha (%M0.0) y otro de Paro(%M0.1).

Los círculos corresponden con las entradas y los rectángulos con las salidas.

Entradas:

- 1.- Pila de pales en la entrada(%M0.2).
- 2.- Pila de pales en desapilador(%M0.3).
- 3.- Palé en rodillos de salida(%M0.4).
- 4.- Cilindro subir pila pales reposo(%M0.5).
- 5.- Cilindro subir pila pales arriba(%M0.6).
- 6.- Cilindro uña sujetar pila reposo(%M0.7).

7.- Cilindro uña sujetar pila extendida(%M1.0).

8.- Freno pila recogido(%M1.1).

9.- Freno pila extendido(%M1.2).

Salidas:

M1.- Motor rodillos entrada pilas(%M10.0).

M2.- Motor rodillos desapilador(%M10.1).

M3.- Motor rodillos salida de palé(%M10.2).

C1.- Cilindro subir pila(%M10.3).

C2.- Cilindro uñas sujetar pila(%M10.4).

C3.- Cilindro freno(%M10.5).

Funcionamiento:

A la maquina entran pilas de pales y los desapila de forma automática entregándolos de uno en uno.

Tenemos un pulsador de marcha uno de paro para el sistema. Si se pulsa el paro la secuencia se detiene continuando en el mismo punto cuando se pulse marcha de nuevo.

El motor M1 está en marcha hasta que detectamos que la pila alcanza la fotocélula 1. La pila permanecerá en esta posición con el motor M1 parado hasta que el desapilador quede vacío, dejando pasar la pila.

La secuencia del desapilador es la siguiente. Entra una pila con el freno C3 arriba y el motor M2 activo hasta que es detectada en 2. En ese momento se para el motor M2. Activamos el cilindro C2 que sujeta las paletas que se encuentran sobre la que queremos extraer. Una vez extendido C2, C1 sube generando el hueco necesario en la pila para que solo salga un palé. En este punto el desapilador espera a tener permiso para extraer palé.

El permiso para extraer palé lo da el no detectar palé en 3 durante más de tres segundos.

M3 está en marcha siempre que la maquina lo esté (marcha/paro).

Cuando el desapilador tiene permiso para extraer el palé baja el freno C3 activa el motor M2 hasta que 2 no detecte paleta mas de 3 segundos. En ese momento entendemos que el palé ha salido y bajamos C1 hasta depositar la pila en el camino de rodillos (4) y entonces desactivamos C2.

Si tras desactivar C2 no detectamos paleta en 2 entendemos que el desapilador está vacío por lo que se reinicia el ciclo con una nueva entrada de pales.



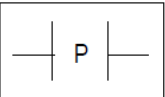
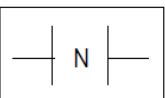
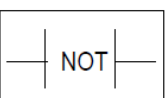
Tareas a realizar:

- Desarrollo del programa en KOP.
- Interfaz HMI para la monitorización y control.

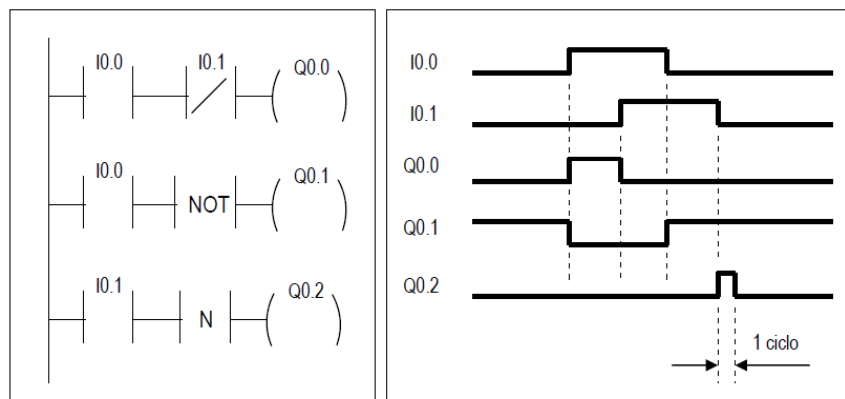
Anexo. Resumen programación en esquema de contactos (KOP).

A continuación, se muestran los elementos más importantes y necesarios de la programación en esquema de contactos (KOP) para esta primera práctica.

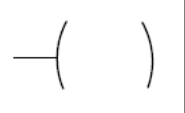
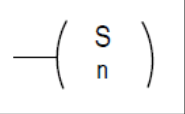
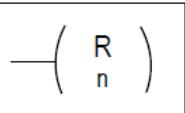
Elementos tipo contacto

	Contacto normalmente abierto: deja circular la corriente cuando la entrada o variable x está activa (valor 1)
	Contacto normalmente cerrado: deja circular la corriente cuando la entrada o variable x está inactiva (valor 0)
	Detección de flanco positivo: deja circular la corriente cuando se detecta un cambio de cero a uno.
	Detección de flanco negativo: deja circular la corriente cuando se detecta un cambio de uno a cero.
	Negación: cambia el valor de la corriente de cero o uno o viceversa (de circular corriente a no circular o viceversa).

A continuación, se muestra un ejemplo del funcionamiento de cada uno de estos elementos mediante un programa y un cronograma con el valor de las variables.



Elementos tipo bobina

	Bobina de asignación: la salida o variable x se activa si llega corriente y se inactiva si no llega corriente.
	Set: cuando llega corriente se activan n bits a partir de la entrada o variable x. No se desactivan cuando deja de llegar corriente (su valor queda enclavado a uno).
	Reset: cuando llega corriente se desactivan n bits a partir de la entrada o variable x. No se activan cuando deja de llegar corriente (su valor queda enclavado a cero).

A continuación, se muestra un ejemplo del funcionamiento de las bobinas.

