UNIVERSIDAD MIGUEL HERNÁNDEZ

AUTOMATIZACIÓN INDUSTRIAL

PRÁCTICA 2: OPERACIONES BÁSICAS DE PROGRAMACIÓN

Índice

1. Repaso de algunos conceptos básicos

1.1 Estructura del programa

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.

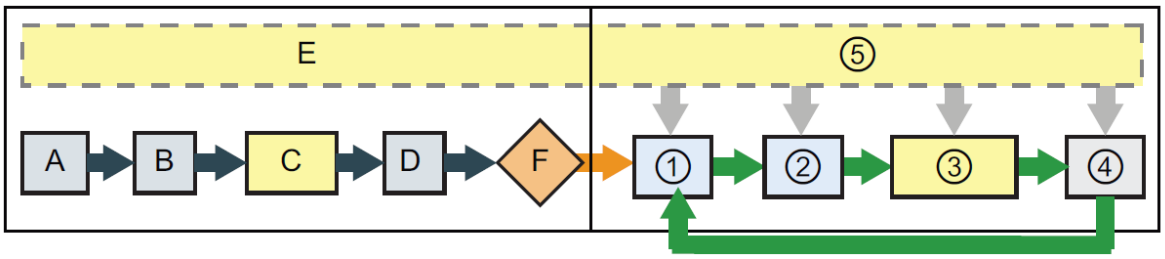
Una función (FC) o un bloque de función (FB) es un bloque de código que puede ser llamado desde un OB o desde otra FC o FB. Las FC no están asociadas con ningún bloque de datos (DB) concreto, mientras que las FB sí están asociadas directamente con un DB, que lo usan para pasar parámetros y almacenar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

1.2 Funcionamiento de la CPU

La CPU tiene tres modos de operación: STOP, STARTUP y RUN:

- En el **modo STOP**, la CPU no ejecuta el programa de usuario. En este modo puede cargarse un proyecto en la CPU. Aunque no se esté ejecutando el programa de usuario, la CPU sí procesa las solicitudes de comunicación y realiza un proceso de autodiagnóstico.
- En el **modo STARTUP**, se ejecutan los OB de inicialización una sola vez (si están presentes en el programa). Durante este modo no se procesan los eventos de interrupción.
- En el **modo RUN**, se ejecuta el ciclo del programa. En este modo sí se pueden procesar los eventos de interrupción y solamente se pueden cargar en la CPU algunas partes del proyecto.

En los modos STARTUP y RUN, la CPU realiza las tareas que se muestran en la figura siguiente:



Las tareas de la figura anterior (letras A-E y números 1-5) se describen a continuación:

Modo STARTUP:

- **A:** Borrado del área de memoria imagen de las entradas (I).
- **B:** Inicialización de las salidas con los últimos valores, o con los valores de sustitución.
- **C:** Se ejecutan los OB de inicialización.
- **D:** Se copia el estado de las entradas físicas en la memoria I.
- **E:** (En paralelo a todas las tareas A-F) Se almacena cualquier evento de interrupción en la cola correspondiente para ser procesados después de entrar en el modo RUN.
- **F:** Se habilita la escritura de la memoria Q en las salidas físicas.

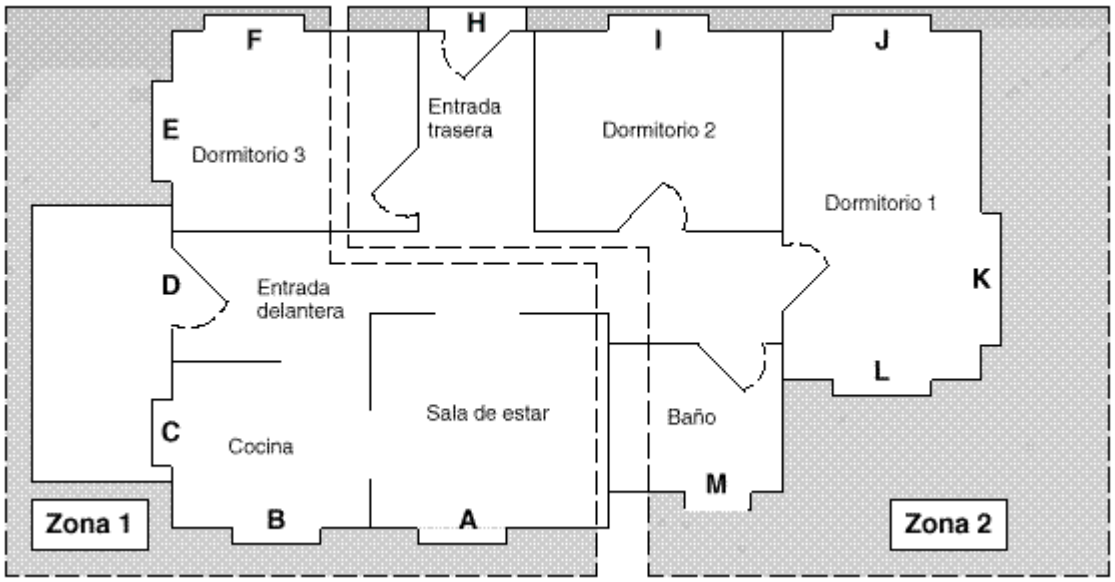
Modo RUN:

- **1:** Se escribe la memoria Q en las salidas físicas.
- **2:** Se copia el estado de las entradas físicas en la memoria I.
- **3:** Se ejecutan los OB del ciclo del programa.
- **4:** Se realiza un autodiagnóstico.
- **5:** (En paralelo a todas las tareas 1-4) Procesa las interrupciones y operaciones comunicación que se produzcan en cualquier instante del ciclo.

2. Ejemplo de creación de un programa

2.1. Definición del problema

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.



En el programa van a estar implicadas las siguientes entradas y salidas:

Entradas:

- La entrada 1 (**I0.0**) vigila la zona 1 (entrada delantera, sala de estar, cocina y dormitorio 3). Entrada normalmente cerrada. (Abierto="0", Cerrado="1").
- La entrada 2 (**I0.1**) vigila la zona 2 (dormitorio 1, dormitorio 2, baño y entrada trasera). Entrada normalmente cerrada. (Abierto="0", Cerrado="1").
- La entrada 3 (**I0.2**) activa o desactiva el sistema de alarma. Activa = "1", Desactivado = "0".
- La entrada 4 (**I0.3**) permite activar manualmente la sirena de alarma. Entrada normalmente abierta. Activa = "1", Desactivado = "0".

Salidas:

- La salida 1 (**Q0.0**) 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 (**Q0.1**) dispara la sirena de alarma.
- La salida 3 (**Q0.2**) activa una señal de alerta baja que indica que la alarma se disparará al cabo de un número predeterminado de segundos.
- La salida 4 (**Q0.3**) activa un relé de interface externo (p.ej. para arrancar una marcación automática).

Funcionamiento del programa de control

La lógica de control de programa debe realizar las siguientes tareas:

- Si el sistema no está activado, el LED (**Q0.0**) parpadea al estar abiertas la zona 1 (**I0.0**) o 2 (**I0.1**).
- Si el sistema está activado (girando la llave a la posición "on", lo que activa la entrada **I0.2**), 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 (**I0.0** ó **I0.1**).
- 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 (**I0.0** ó **I0.1**), el programa arranca una secuencia de notificación que activa la señal de alerta baja (**Q0.2**) 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 **I0.2**), el programa pone a "0" las salidas (**Q0.0** y **Q0.2**) 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 (**Q0.1** y **Q0.3**).
- Si se activa la alarma manual (**I0.3**), el programa dispara la alarma y activa la marcación automática del módem (**Q0.1** y **Q0.3**). Esta tarea se realiza independientemente de la posición del interruptor que activa o desactiva el sistema de alarma (**I0.2**) 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 **I0.2**) una vez disparada la alarma (**Q0.1**), el programa pone a "0" las salidas (**Q0.1** y **Q0.3**) y los temporizadores.

El programa utilizará 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.

2.2. Tablas de variables

En cualquier programa de control, si se utilizan símbolos (cada vez que se asigna un nombre simbólico a una dirección se crea un símbolo), el proyecto se podrá crear, mantener y documentar más fácilmente.

En la tabla de símbolos/tabla de variables globales se asignan nombres simbólicos a la memoria de la CPU y a las direcciones de E/S. Si un símbolo se define en la tabla de símbolos/tabla de variables globales, el símbolo tendrá ámbito global. Ello significa que el nombre del símbolo se puede utilizar en cualquier unidad de

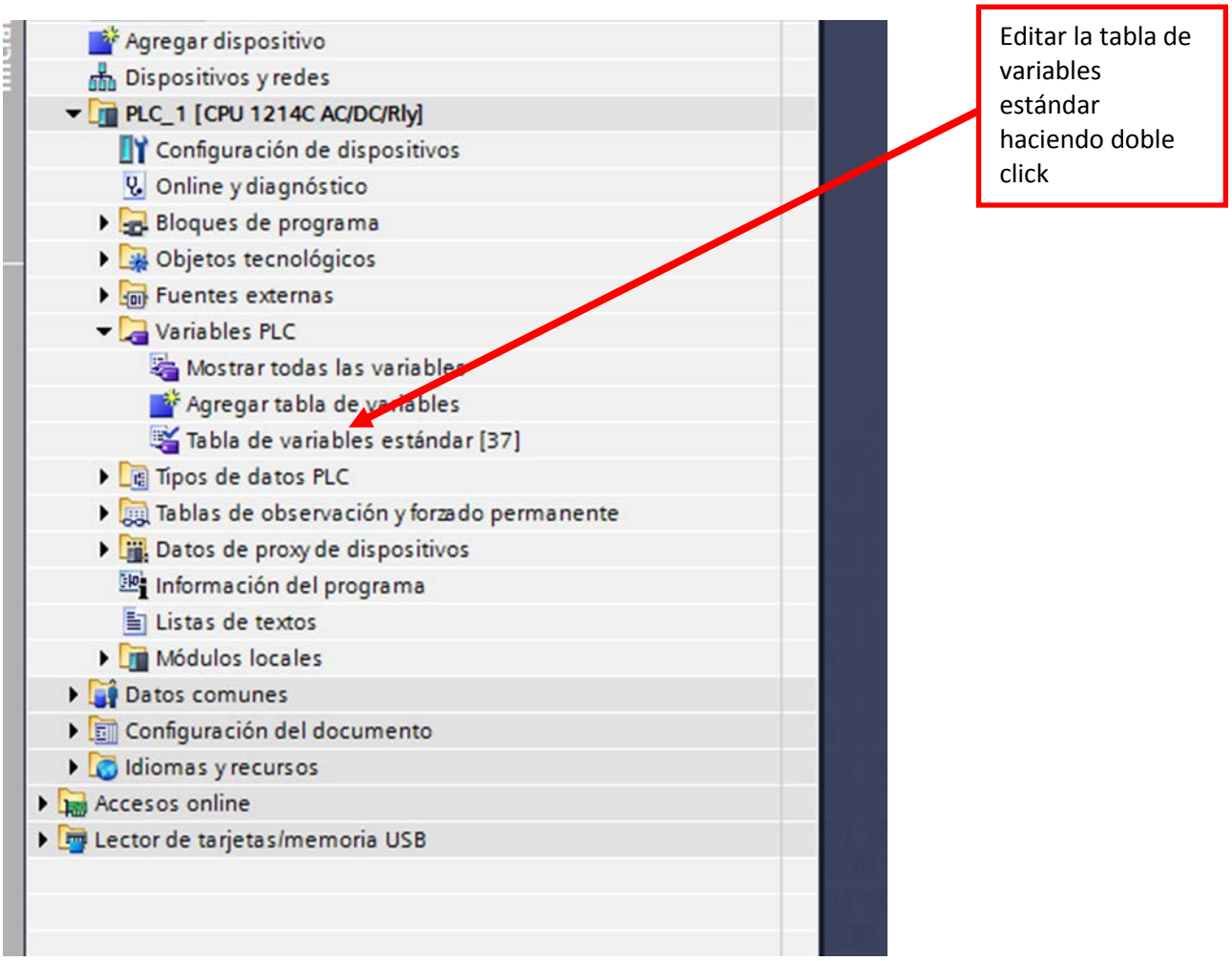
organización del programa como referencia a los datos de la dirección correspondiente a dicho símbolo. En cambio, si asigna un símbolo en una tabla de variables locales, el ámbito de dicho símbolo o "variable local" se limitará a la unidad de organización del programa donde se ha definido.

En la descripción que sigue **se supone que se ha creado un proyecto en el programa TIA y se ha añadido un autómata S7-1200 al mismo, como se explicó en la práctica 1.**

Para asignar un símbolo a una dirección, pueden seguirse los pasos siguientes:

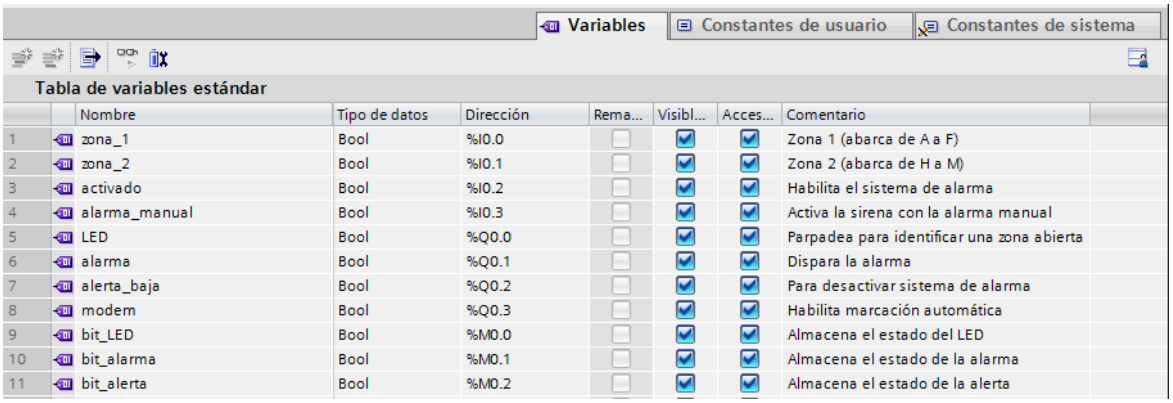
1. **En el panel de la izquierda** (en el árbol del proyecto), en la rama correspondiente al autómata del proyecto, **abrir la opción "Variables PLC"**.

2. **Dentro de "Variables PLC"** vemos que aparece una opción para "Agregar tabla de variables" y, además, aparece una tabla de variables ya creada que se llama "Tabla de variables estándar". En este ejemplo **seleccionaremos haciendo doble click la "Tabla de variables estándar", como se muestra en la figura siguiente** (también podría crearse una tabla nueva si se desea, aunque en este ejemplo no es necesario).



Una vez hecho esto se abrirá el editor de la tabla de variables estándar. Sobre este editor, **introduciremos las variables correspondientes a las entradas/salidas y las marcas que usaremos en el programa**, como se indica en la figura siguiente.

Obsérvese que habrá que escribir el nombre de cada variable, el tipo de datos ("Bool" si se trata de un bit) y la dirección correspondiente. El símbolo "%" delante de las direcciones lo introduce automáticamente el editor para indicar que son direcciones de memoria absolutas y no símbolos del usuario. También es posible introducir en la tabla comentarios sobre cada variable (en la última columna).



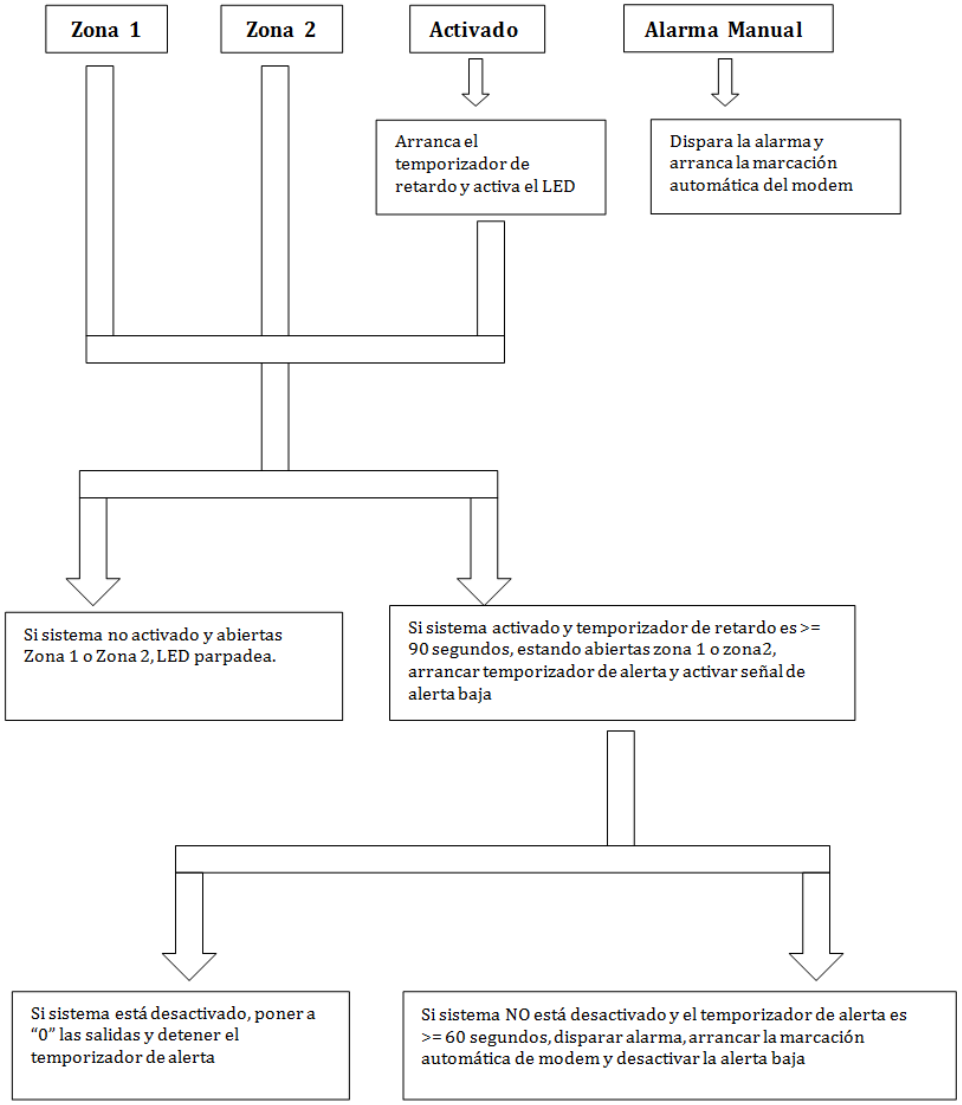
Variables | Constantes de usuario | Constantes de sistema

Tabla de variables estándar

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	zona_1	Bool	%I0.0		☑	☑	Zona 1 (abarca de A a F)
2	zona_2	Bool	%I0.1		☑	☑	Zona 2 (abarca de H a M)
3	activado	Bool	%I0.2		☑	☑	Habilita el sistema de alarma
4	alarma_manual	Bool	%I0.3		☑	☑	Activa la sirena con la alarma manual
5	LED	Bool	%Q0.0		☑	☑	Parpadea para identificar una zona abierta
6	alarma	Bool	%Q0.1		☑	☑	Dispara la alarma
7	alerta_baja	Bool	%Q0.2		☑	☑	Para desactivar sistema de alarma
8	modem	Bool	%Q0.3		☑	☑	Habilita marcación automática
9	bit_LED	Bool	%M0.0		☑	☑	Almacena el estado del LED
10	bit_alarma	Bool	%M0.1		☑	☑	Almacena el estado de la alarma
11	bit_alerta	Bool	%M0.2		☑	☑	Almacena el estado de la alerta

2.3. Tareas básicas del programa para el sistema de alarma

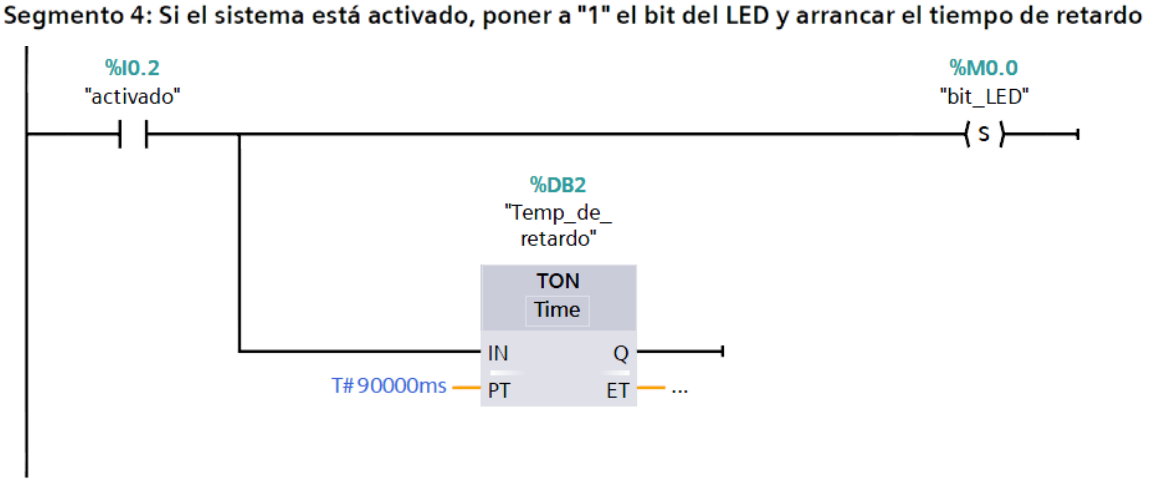
En la figura siguiente se muestra un diagrama de las tareas a realizar por el sistema de control del ejemplo.



2.4. Diseño de la lógica de control

Activar el sistema de alarma

Al activarse el sistema se habilita la marca de memoria M que controla la activación del LED. La lógica de control provee también un tiempo de retardo entre la activación del interruptor y la activación del sistema de alarma. Ello le permite al propietario activar el sistema de alarma y salir de la vivienda. (Hay otro temporizador que controla una señal de alerta baja. Esta le indica al propietario de la vivienda que desactive el sistema).

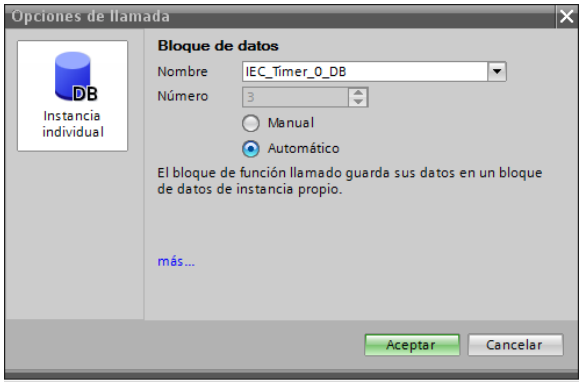


Como se ve en la figura anterior, el contacto normalmente abierto (Activado) pone a 1 la marca Bit_LED e inicia la cuenta en el temporizador "Temp_de_retardo". Se ha seleccionado un temporizador con retardo a la conexión (de tipo TON), de forma que cuando la cuenta (Tiempo_Retardo) sea igual 90000 ms, habrán transcurrido los 90 segundos que el sistema le concede al usuario para activar la alarma y abandonar el edificio.

A continuación se explica **cómo introducir un temporizador en el programa**:

- En el panel de la derecha ("Instrucciones") **seleccionamos un temporizador de tipo TON** (en este caso) **y lo arrastramos con el ratón hasta el punto del programa** donde deseamos colocarlo.

- Se abrirá una ventana en la que podremos especificar el bloque de datos (DB) asociado al temporizador, como se muestra en la figura siguiente.



En esta ventana **escribiremos el nombre que queramos darle al DB del temporizador** (por ejemplo, *Temp_de_retardo*), para luego hacer referencia a las variables del temporizador (bit y tiempo, por ejemplo).

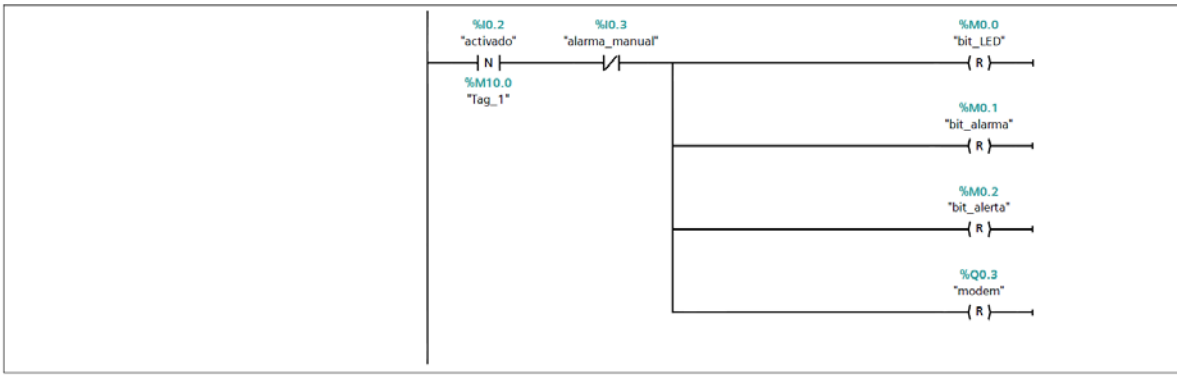
- **Escribimos el valor del parámetro PT del temporizador** (90 segundos). En este ejemplo se ha escrito en milisegundos (90000ms), aunque también puede escribirse en segundos directamente (90s). El símbolo T# lo incluye automáticamente el editor para indicar que se trata de una variable de tiempo.

- En este ejemplo se han dejado sin conectar las salidas Q y ET del temporizador (se ha optado por acceder a las variables Q y ET a partir del bloque de datos del temporizador, como se verá en otros segmentos).

Desactivar el sistema de alarma

Al desactivarse el sistema de alarma se detiene la señal de alerta y la secuencia de alarma. En la figura siguiente (segmento 5) se puede ver como se ha implementado esta parte del programa de control.

Segmento 5: Poner todo a "0" si se desactiva el sistema



En el segmento anterior se ha utilizado la operación de detección de flanco negativo. Para poner a cero las marcas se ha optado por la operación de Reset "simple" (que solo resetea un bit), aunque se podría usar también la instrucción "RESET_BF" para resetear varios bits.

Activar el parpadeo del LED

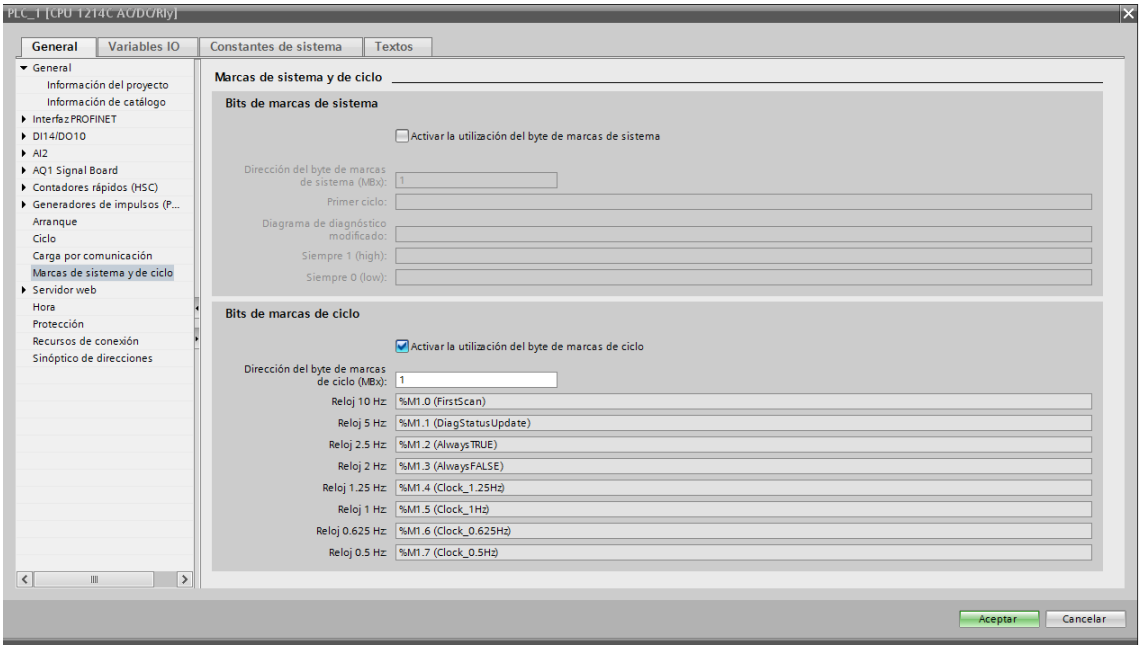
Para realizar este parpadeo se ha utilizado una posición de memoria " especial" del autómata. Estas posiciones de memoria "especiales" ofrecen una serie de funciones de estado y control. A continuación se explica **cómo activar estas posiciones de memoria especiales** en el autómata S7-1200:

- En el panel izquierdo (árbol del proyecto) **hacemos click con el botón derecho del ratón sobre el PLC** y, en el menú contextual que aparece, **seleccionamos "Propiedades"**.

- Aparecerá una ventana para editar las propiedades del autómata. En esta ventana **seleccionamos la solapa General** y, en el menú de la izquierda, **seleccionamos "Marcas de sistema y de ciclo"**, como se muestra en la figura siguiente.

- En la parte de la derecha podremos seleccionar los bits "especiales" que deseamos usar en el programa. En nuestro ejemplo **seleccionamos "Activar la utilización del byte de marcas de ciclo"**, como se muestra en la figura.

- **En "Dirección de byte de marcas de ciclo"** debemos 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 de nuestro programa*. En este ejemplo **escribimos el byte 1** (ya que no se usarán los bits M1.0 a M1.7 en el resto del programa), como se muestra en la figura.



- Una vez pulsado "Aceptar" en la ventana anterior, podemos comprobar que en la tabla de variables estándar se han añadido los "Bits de marcas de ciclo", como se puede observar en la figura siguiente.

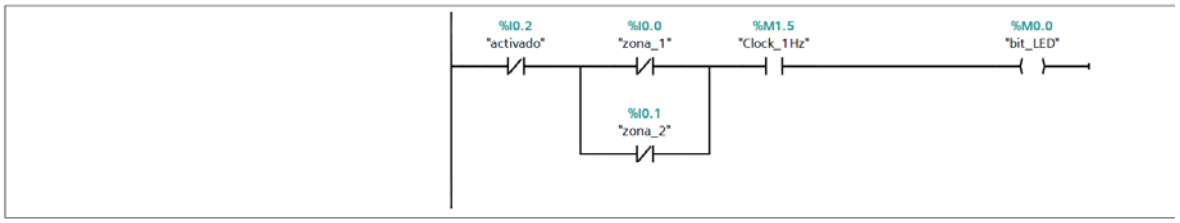
Variables | Constantes de usuario | Constantes de sistema

Tabla de variables estándar

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	zona_1	Bool	%I0.0		☑	☑	Zona 1 (abarca de A a F)
2	zona_2	Bool	%I0.1		☑	☑	Zona 2 (abarca de H a M)
3	activado	Bool	%I0.2		☑	☑	Habilita el sistema de alarma
4	alarma_manual	Bool	%I0.3		☑	☑	Activa la sirena con la alarma manual
5	LED	Bool	%Q0.0		☑	☑	Parpadea para identificar una zona abierta
6	alarma	Bool	%Q0.1		☑	☑	Dispara la alarma
7	alerta_baja	Bool	%Q0.2		☑	☑	Para desactivar sistema de alarma
8	modem	Bool	%Q0.3		☑	☑	Habilita marcación automática
9	bit_LED	Bool	%M0.0		☑	☑	Almacena el estado del LED
10	bit_alarma	Bool	%M0.1		☑	☑	Almacena el estado de la alarma
11	bit_alerta	Bool	%M0.2		☑	☑	Almacena el estado de la alerta
12	Clock_Byte	Byte	%MB1		☑	☑	
13	Clock_1.25Hz	Bool	%M1.4		☑	☑	
14	Clock_1Hz	Bool	%M1.5		☑	☑	
15	Clock_0.625Hz	Bool	%M1.6		☑	☑	
16	Clock_0.5Hz	Bool	%M1.7		☑	☑	

Con la configuración anterior, las marcas M1.4 a M1.7 generan señales cuadradas de distintos periodos. Así, por ejemplo, en la marca M1.5 se genera una onda cuadrada binaria de frecuencia 1 Hz, que es la que utilizaremos en este ejemplo para conseguir el parpadeo del LED, como se muestra en la figura siguiente (segmento 6).

Segmento 6: Activa y desactiva el LED



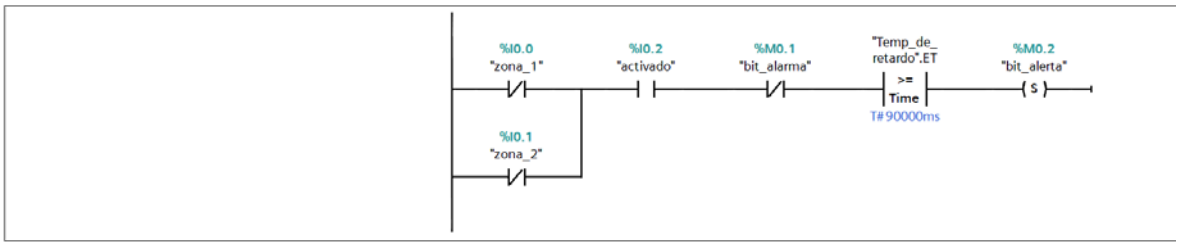
Activar la señal de alerta baja

Cuando se irrumpe en una de las zonas vigiladas (es decir, al abrirse la zona 1 o la zona 2 una vez activado el sistema de alarma), el programa activa la señal de alerta baja. El propietario de la vivienda puede desactivar el sistema en un tiempo determinado (p.ej. cuando regresa a casa).

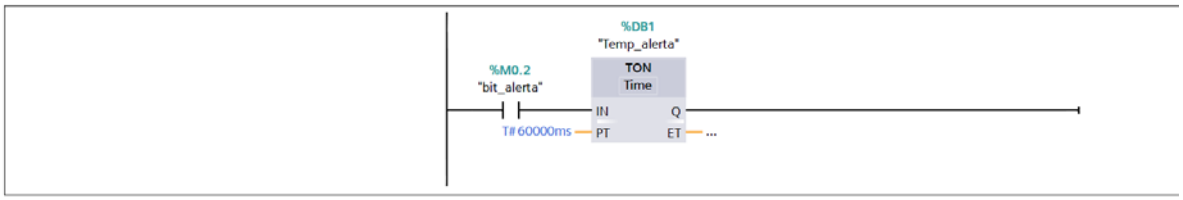
Como muestra la figura siguiente, el programa vigila el estado de ambas zonas, así como el interruptor para activar y desactivar el sistema. Asimismo, dispone de un retardo de activación de la alarma de 90 segundos.

Cuando se detecta una irrupción en la vivienda, el programa arranca el temporizador de alerta baja (temporización de 60 s).

Segmento 2: Evaluar el estado del sistema



Segmento 3: Arrancar el temporizador de alerta



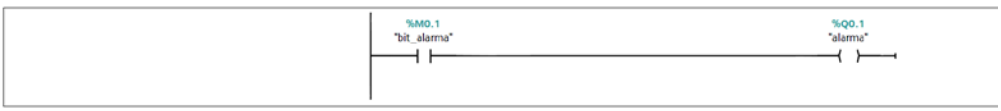
Se ha utilizado en estas líneas de programa una instrucción de comparación, que permite comprobar si el temporizador que mide el tiempo de retardo desde que el sistema ha sido activado ha sobrepasado los 90 segundos.

Disparar la alarma y activar la marcación del modem

El programa utiliza marcas (M) para almacenar los resultados de la lógica de control. Al final del programa, dichas marcas activan (o desactivan) las salidas, como se muestra en la figura siguiente (segmentos 7 a 9).

Segmento 7: Activar el LED del sistema

Segmento 8: Activar la sirena de alarma

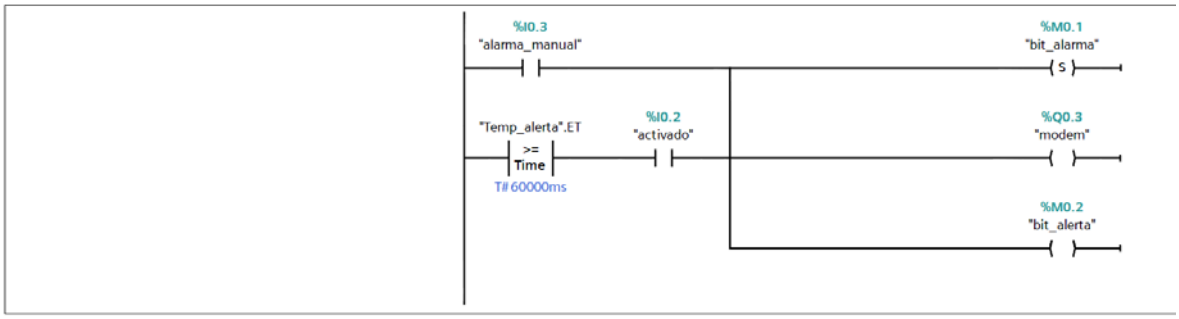


Segmento 9: Activar la señal de alerta baja

Como muestra la figura siguiente, las marcas correspondientes a la sirena de alarma y a la marcación del módem se activan si se presenta una de las siguientes situaciones:

- Alguien activa la alarma manual (sin importar si el sistema de alarma está activado o desactivado y sin que se emita la señal de alerta baja).
- El sistema no se ha desactivado al cabo de 60 segundos después de haberse activado la señal de alerta baja.

Segmento 1: Disparar alarma

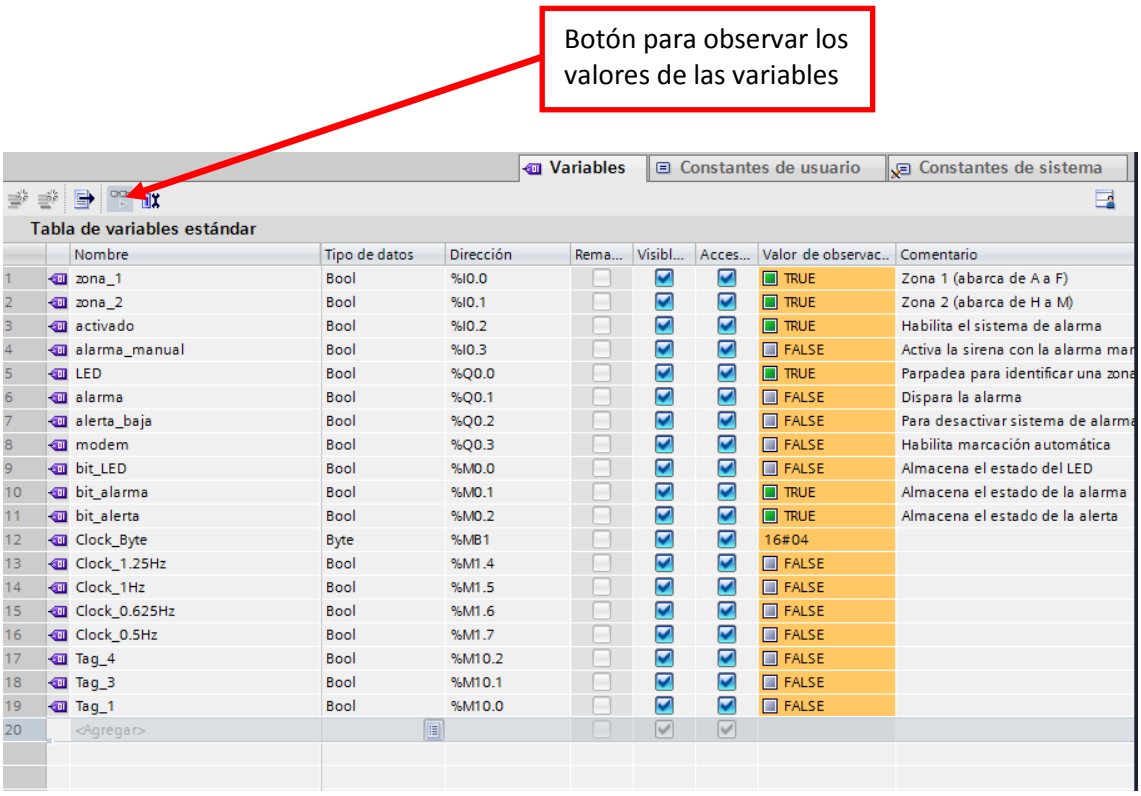


Al dispararse la alarma se desactiva también la señal de alerta baja.

2.5. Comprobación del funcionamiento del programa

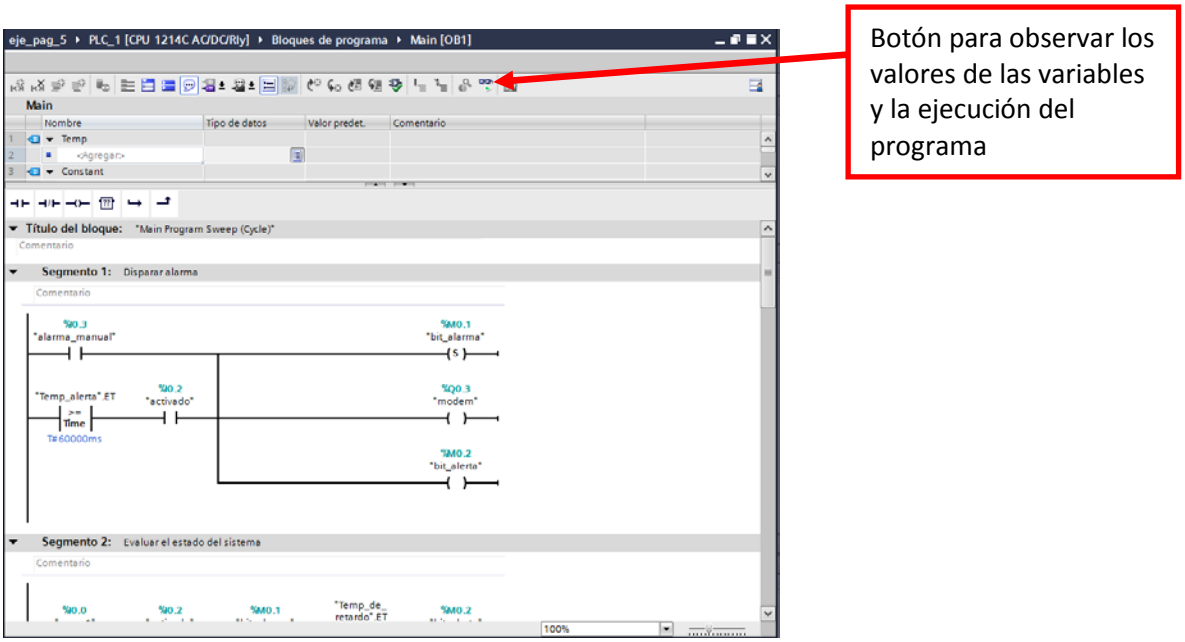
Una vez cargado el programa en la CPU, hay varias formas de llevar un seguimiento de su correcto funcionamiento. En las siguientes líneas se describen algunas de ellas (se supone que el autómata está en modo RUN).

- **Primera forma: En la tabla de variables estándar**. En esta tabla pueden verse los valores que toman las variables en cada instante de ejecución del programa. Para ello, abrimos la tabla de variables y pulsamos el botón cuyo icono tiene unas "gafas" que aparece en la parte superior, como se muestra en la figura siguiente. En ese momento, TIA establece una "conexión online" con el autómata para observar los valores de las variables.



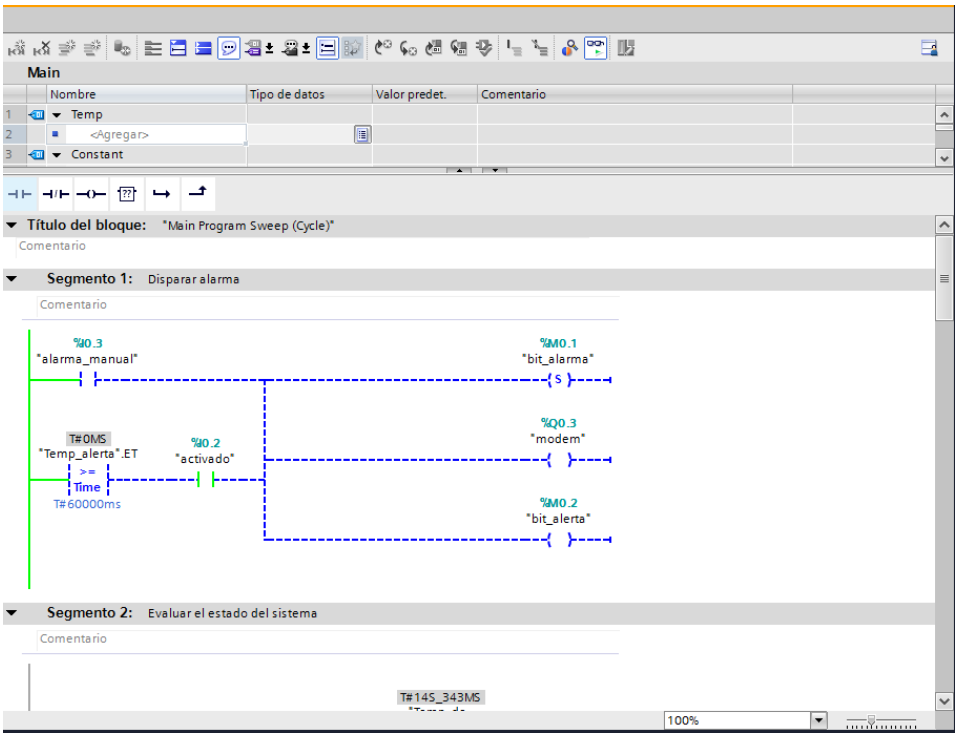
	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Valor de observac...	Comentario
1	zona_1	Bool	%I0.0		☑	☑	TRUE	Zona 1 (abarca de A a F)
2	zona_2	Bool	%I0.1		☑	☑	TRUE	Zona 2 (abarca de H a M)
3	activado	Bool	%I0.2		☑	☑	TRUE	Habilita el sistema de alarma
4	alarma_manual	Bool	%I0.3		☑	☑	FALSE	Activa la sirena con la alarma mar
5	LED	Bool	%Q0.0		☑	☑	TRUE	Parpadea para identificar una zona
6	alarma	Bool	%Q0.1		☑	☑	FALSE	Dispara la alarma
7	alerta_baja	Bool	%Q0.2		☑	☑	FALSE	Para desactivar sistema de alarma
8	modem	Bool	%Q0.3		☑	☑	FALSE	Habilita marcación automática
9	bit_LED	Bool	%M0.0		☑	☑	FALSE	Almacena el estado del LED
10	bit_alarma	Bool	%M0.1		☑	☑	TRUE	Almacena el estado de la alarma
11	bit_alerta	Bool	%M0.2		☑	☑	TRUE	Almacena el estado de la alerta
12	Clock_Byte	Byte	%MB1		☑	☑	16#04	
13	Clock_1.25Hz	Bool	%M1.4		☑	☑	FALSE	
14	Clock_1Hz	Bool	%M1.5		☑	☑	FALSE	
15	Clock_0.625Hz	Bool	%M1.6		☑	☑	FALSE	
16	Clock_0.5Hz	Bool	%M1.7		☑	☑	FALSE	
17	Tag_4	Bool	%M10.2		☑	☑	FALSE	
18	Tag_3	Bool	%M10.1		☑	☑	FALSE	
19	Tag_1	Bool	%M10.0		☑	☑	FALSE	
20	<Agregar>							

- **Segunda forma: Sobre los bloques de programa.** Sobre el editor del programa KOP también puede comprobarse de forma gráfica el funcionamiento del programa y los valores de las variables a medida que avanza la ejecución. Para ello, accedemos al editor del programa principal (main, OB1) y, estando el autómata en modo RUN, pulsamos sobre el botón cuyo icono tiene unas "gafas" que aparece en la parte superior, como se muestra en la figura siguiente.

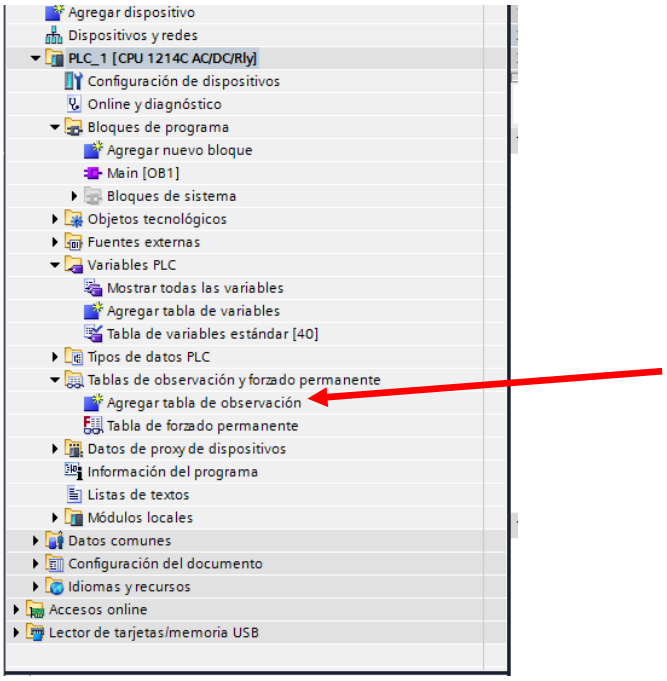


Si pulsamos el botón indicado se mostrará sobre el programa lo siguiente (véase la figura siguiente):

- En color verde las líneas del programa KOP que tienen valor lógico 1 ("circula corriente").
- En color azul y trazo discontinuo, las líneas del programa KOP que tienen valor lógico 0 ("no circula corriente").
- Los valores de las variables en tiempo real. En el caso de los bits, la operación correspondiente aparece en color verde si está activada la operación (es decir, si la operación "deja pasar corriente") y en color azul si está desactivada (es decir, si la operación "no deja pasar corriente").



- **Tercera forma: Usando una tabla de observación y forzado.** Como indica su nombre, estas tablas permiten tanto observar como forzar (asignar) valores a las variables. Para crear una tabla de este tipo, accedemos al panel de la izquierda (árbol del proyecto) y **hacemos doble click sobre "Tablas de observación y forzado permanente"->"Agregar tabla de observación"**, como se muestra en la figura siguiente.

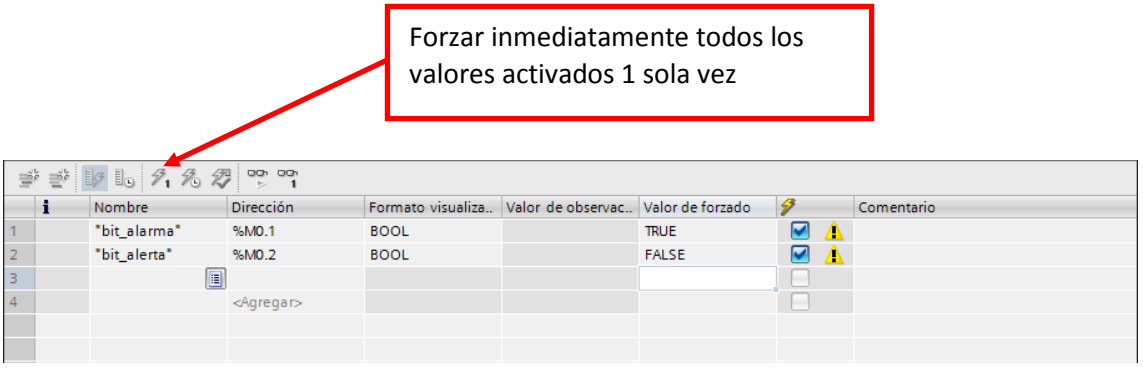


En la tabla que aparece podemos añadir las variables cuyos valores queremos observar y/o forzar. Los valores "observados" aparecerán en la columna "Valor de observación". Si deseamos asignar algún valor a alguna variable para realizar pruebas, debemos seguir los pasos siguientes:

1. Activar la casilla correspondiente a la variable en la penúltima columna (cuyo icono es un rayo de color amarillo, como se muestra en la figura siguiente).

2. Escribir el valor deseado para la variable en la columna "Valor de forzado" (en el caso de bits, podemos escribir un 1 o un 0).

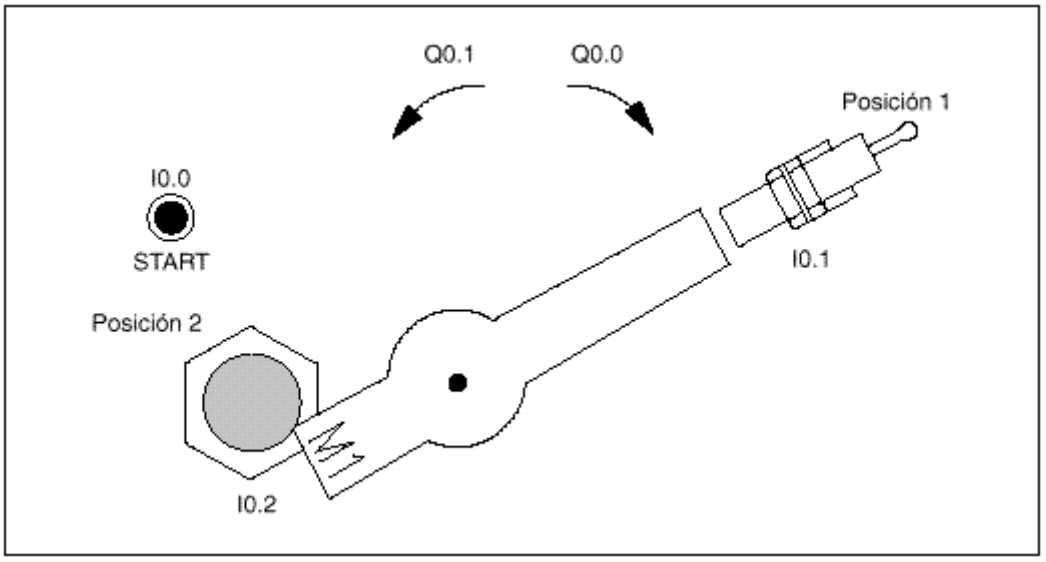
3. Pulsar el botón de forzado inmediato 1 sola vez (botón cuyo icono es un rayo y un 1, como se muestra en la figura siguiente).



3. Ejercicios de programación

Ejercicio 1

La figura representa una máquina que tiene un brazo motorizado. Cuando se pulsa el botón de arranque con el brazo en la posición 1, el brazo gira en sentido horario y detiene su rotación una vez que ha llegado a la posición 2. Transcurridos 5 segundos, el brazo gira en sentido antihorario hasta la posición 1 y se para. El ciclo se puede repetir de nuevo, cuando pulse el botón de arranque.



Se requieren los siguientes elementos para posicionar el motor correctamente.

Entradas :

- I0.0 está vinculado al botón de arranque.
- I0.1 está vinculado al micro---interruptor en la posición 1.
- I0.2 está vinculado al micro---interruptor de la posición 2.

Salidas :

- Q0.0 está vinculada al motor cuando gira en sentido horario.
- Q0.1 está vinculada al motor cuando gira en sentido antihorario.

Marcas:

- M0.0 secuencia de arranque del motor : Si ((I0.0 y I0.1) o M0.0) y no M0.1 entonces M0.0
- M0.1 secuencia terminada : Si Q0.1 y I0.1 entonces M0.1

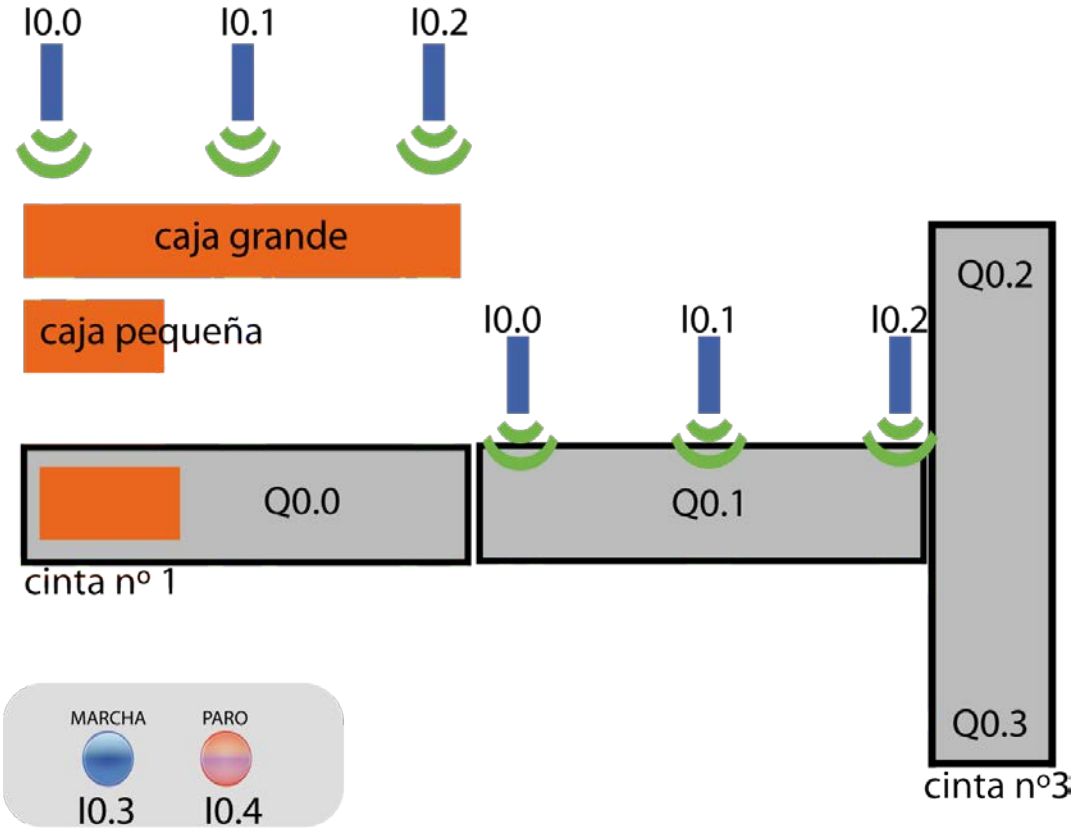
Temporizador:

- Debe usarse un temporizador de retardo a la conexión (TON).

Ejercicio 2

CINTAS TRANSPORADORAS

Tenemos tres cintas transportadoras dispuestas de la siguiente manera:



Por las cintas transportadoras van a circular cajas grandes y pequeñas indistintamente. El tamaño de las cajas con respecto a los sensores ópticos que tenemos en la segunda cinta se muestra en la figura.

El funcionamiento que queremos es el siguiente:

Cuando le demos al pulsador de marcha queremos que se ponga en marcha la cinta nº 1 (Q0.0). Cuando llegue la primera caja a la cinta nº 2, queremos que se pare la cinta no 1 y que se ponga en marcha la cinta nº 2 (Q0.1). En la cinta nº 2 detectamos si la caja es grande o pequeña. Si es grande, queremos que se ponga en marcha la tercera cinta hacia arriba (Q0.2), y si es pequeña queremos que se ponga en marcha la tercera cinta hacia abajo (Q0.3). La cinta nº 2 se para cuando la caja ya esté abandonando la cinta n°2. La cinta nº 3 se para a los 10 seg. de haberse puesto en marcha. A continuación se pone en marcha de nuevo la primera cinta y vuelve a comenzar el ciclo.