

PRACTICA 3

**ENTORNO DE PROGRAMACIÓN STEP7-MICRO / WIN 32:
ESTRUCTURA DE PROGRAMA, TABLA DE SIMBOLOS, TABLA
DE ESTADO, DEPURACIÓN DEL PROGRAMA.**

Indice

1.- Estructura del programa	3
2.- Creación de un proyecto en STEP7-Micro/Win 32	4
3.- Ejemplo de creación de un programa	5
3.1.- Definir el problema.....	5
3.2.- Tabla de símbolos.....	6
3.3.- Tareas básicas del programa para el sistema de alarma	8
3.4.- Diseñar la lógica de control	10
3.5.- Comprobación del funcionamiento del programa	14
4.- Ejercicio de programación	19

PRACTICA 3

ENTORNO DE PROGRAMACIÓN STEP7-MICRO / WIN 32: ESTRUCTURA DE PROGRAMA, TABLA DE SIMBOLOS, TABLA DE ESTADO, DEPURACIÓN DEL PROGRAMA.

1.- Estructura del programa

Los programas para la CPU S7-200 comprenden tres partes básicas: el programa principal, las subrutinas (opcional) y las rutinas de interrupción (opcional). Como muestra la figura 1, la estructura de un programa esta formada por los siguientes elementos:

- **Programa principal:** En esta parte del programa se disponen las operaciones que controlan la aplicación. Las operaciones del programa principal se ejecutan de forma secuencial en cada ciclo de la CPU. Para terminar el programa principal, utilice en KOP una bobina absoluta Finalizar programa principal, o en AWL una operación Finalizar programa principal (MEND). Ver (1) en la figura 1.
- **Subrutinas:** Estos elementos opcionales del programa se ejecutan sólo cuando se llaman desde el programa principal. Se deben añadir siempre al final del programa principal (detrás de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL). Utilice siempre una operación Retorno absoluto (RET) para terminar cada subrutina. Véase (2) en la figura 1.
- **Rutinas de interrupción:** Estos elementos opcionales del programa se ejecutan cada vez que se presente el correspondiente evento de interrupción. Se deben añadir siempre al final del programa principal (detrás de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL). Utilice siempre una operación Retorno absoluto desde rutina de interrupción (RETI) para terminar cada rutina de interrupción. Ver (3) en la figura 1.

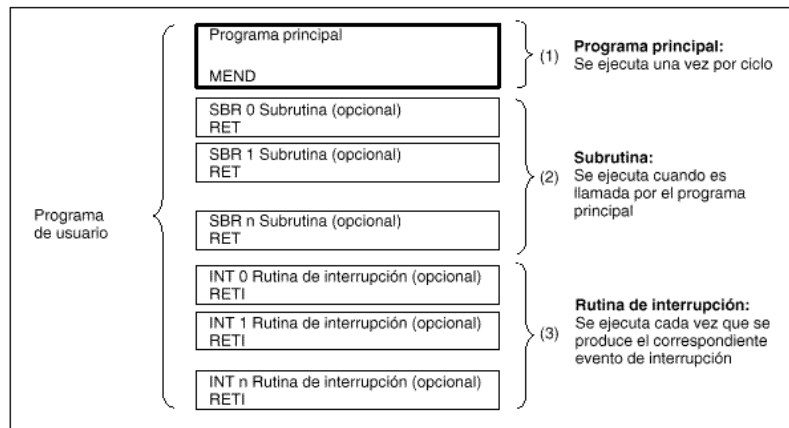


Figura 1

Las subrutinas y las rutinas de interrupción se deben añadir detrás de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL. No hay reglas adicionales en lo relativo a su disposición en el programa de usuario. Las subrutinas y las rutinas de interrupción se pueden mezclar a voluntad después del programa principal. No obstante, para que la estructura del programa sea fácil de leer y comprender, es recomendable agrupar al final del programa principal primero todas las subrutinas y, después, todas las rutinas de interrupción.

2.- Creación de un proyecto en STEP7-Micro/WIN 32

Un proyecto en STEP 7-Micro/WIN 32 integra la información necesaria para comunicarse con una CPU y cargar el programa en ella. Un proyecto esta compuesto por cinco elementos básicos :



Bloque de programa : El bloque de programa incluye el código ejecutable y los comentarios. El código se compila y se carga en la CPU, mas no los comentarios del programa.



Bloque de datos : El bloque de datos comprende datos (valores iniciales de memoria, valores de constantes) y comentarios. Los datos se compilan y se cargan en la CPU, mas no los comentarios.



Bloque de sistema : El bloque de sistema comprende los datos de configuración, tales como los parámetros de comunicación, las áreas remanentes, los filtros de las entradas analógicas y digitales, los valores de las salidas en caso de un cambio a STOP y las informaciones sobre la protección con contraseña. Las informaciones contenidas en el bloque de sistema se cargan en la CPU.



Tabla de símbolos : La tabla de símbolos permite utilizar el direccionamiento simbólico para la programación. En algunos casos es más conveniente programar con símbolos, puesto que facilitan el entendimiento del programa. El programa compilado que se carga en la CPU convierte todos los símbolos a direcciones absolutas. Las informaciones contenidas en la tabla de símbolos no se cargan en la CPU.



Tabla de estado : Las informaciones contenidas en la tabla de estado no se cargan en la CPU.

Todos estos componentes pueden ser visualizados en sus ventanas respectivas sobre el área de trabajo como muestra la Figura 8 (Práctica 1). Para abrir cada una de estas ventanas basta con hacer un doble click sobre el elemento correspondiente en la barra de navegación o bien mediante la opción correspondiente del menú Ver.

Una vez abierto el proyecto, conviene seleccionar el tipo de CPU a la que va destinado el proyecto (En la práctica 1 se vio como seleccionar el tipo de CPU) también es conveniente, aunque se puede dejar para cuando se vaya a cargar el programa en la CPU, configurar la comunicación con la CPU. Estas tareas se pueden realizar más adelante, no es necesario realizarlas nada más abrir o crear un proyecto pero es una buena pauta de trabajo llevar a cabo estas acciones nada más abrir o crear un proyecto.

3.- Ejemplo de creación de un programa

3.1.- Definir el problema

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

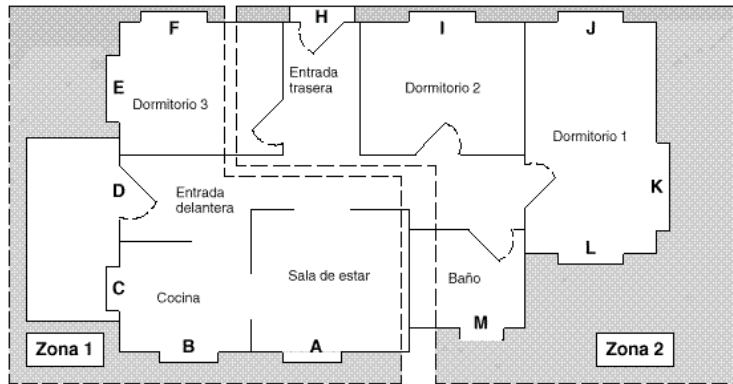


Figura 2

En el programa van a estar implicadas las siguiente 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. Encendido si está activado y parpadeante si está desactivado, estando abiertas las 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 arrancar 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.

3.2.- Tabla de símbolos

Tabla de símbolos (en modo SIMATIC) o una tabla de variables globales (en modo IEC 1131-3).

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 pro-

grama 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).

Se pueden crear nombres simbólicos para los siguientes tipos de memoria: I, Q, M, SM, AI, AQ, V, S, C, T

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

1. Abra la tabla de símbolos/tabla de variables globales.
2. En la columna "Nombre simbólico", teclee el símbolo (máx. 23 caracteres).
3. En la columna "Dirección", teclee la dirección (p.ej. I0.0). (En el modo de programación IEC 1131-3, el prefijo correcto "%" se agrega automáticamente tras introducir la dirección). Si está utilizando la tabla de variables globales, teclee un valor en la columna "Tipo de datos" o seleccione uno del cuadro de lista. (Los usuarios de SIMATIC no deben indicar un tipo de datos.)
4. Introduzca un comentario (opcional, máx. 79 caracteres).

Errores de introducción de símbolos

Los errores de introducción se marcan en color para poder visualizarlos fácilmente:

- Sintaxis no válida - texto rojo
- Uso no admisible - subrayado rojo en dientes de sierra
- Símbolo no definido - subrayado verde en dientes de sierra

	Nombre	Dirección	Comentario
1	ENTRADA	I0.0	
2	PULSADOR	I0.0	
3	ALARMA	3Q01	
4			
5			

Figura 3

En la figura 3 se puede observar un detalle de los errores en la introducción de símbolos en la Tabla de símbolos.

Ahora se pretende crear la tabla de símbolos correspondiente al ejemplo. En la figura 4 se muestra un detalle de dicha tabla.

	Nombre	Dirección	Comentario
1	Zona_1	I0.0	Zona 1 (abarca de A a F)
2	Zona_2	I0.1	Zona 2 (abarca de H a M)
3	Activado	I0.2	Habilita el sistema de alarma
4	Alarma_Manual	I0.3	Activa la sirena con la alarma manual
5	LED	Q0.0	Parpadea para identificar una zona abierta
6	Alarma	Q0.1	Dispara la alarma
7	Alerta_Baja	Q0.2	Para desactivar sistema de alarma
8	Modem	Q0.3	Habilita marcación automática
9	Bit_LED	M0.0	Almacena el estado del LED
10	Bit_Alarma	M0.1	Almacena el estado de la alarma
11	Bit_Alerta	M0.2	Almacena el estado de la alerta
12	Temp_Alerta	T0	Retarda la alarma (permite desactivarla)
13	Tiempo_de_Retardo	T1	Tiempo para salir de la vivienda

Figura 4

T38 T37

3.3.- Tareas básicas del programa para el sistema de alarma

En la figura 5 de la página siguiente, se puede apreciar un diagrama de las tareas a realizar por el sistema de control.

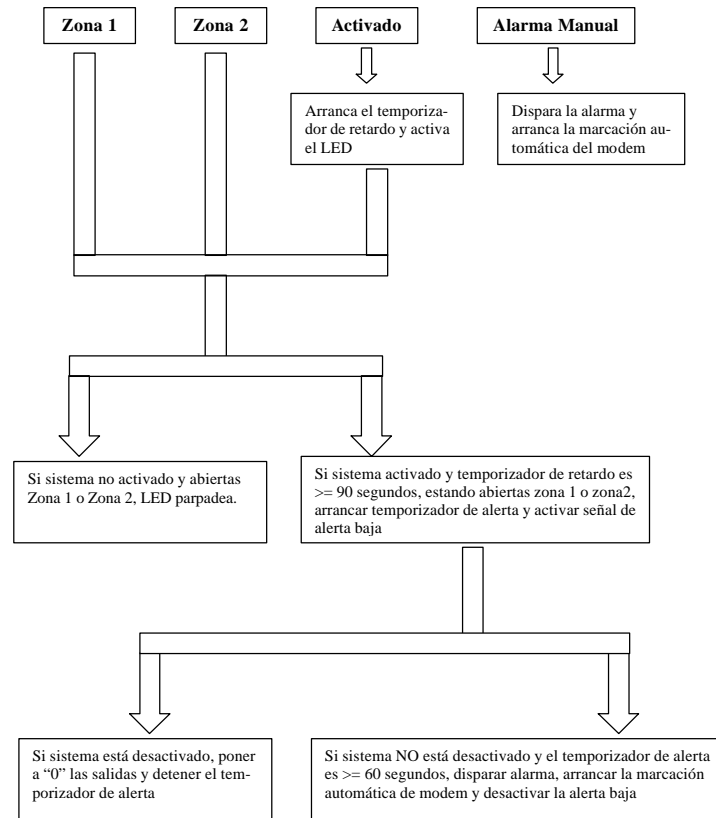


Figura 5

3.4.- Diseñar 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.)

Network 4 Si el sistema está activado, poner a "1" el bit del LED y arrancar el tiempo de retardo

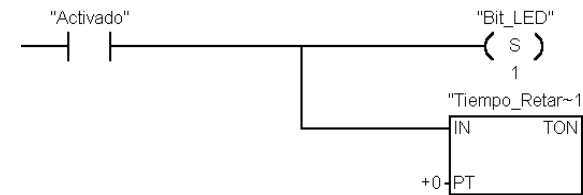


Figura 6

Como se ve en la figura el contacto normalmente abierto (Activado) pone a 1 la marca Bit_LED e inicia la cuenta en el temporizador T38. Se ha seleccionado un temporizador con retardo a la conexión de una resolución de 100 ms, de forma que cuando la cuenta (Tiempo_Retardo) sea igual a 900 habrán transcurrido 90000 ms, los 90 segundos que es sistema le concede al usuario para activar la alarma y abandonar el edificio.

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 7 se puede ver como se ha implementado esta parte del programa de control.

Network 5 Poner todo a "0" si se desactiva el sistema

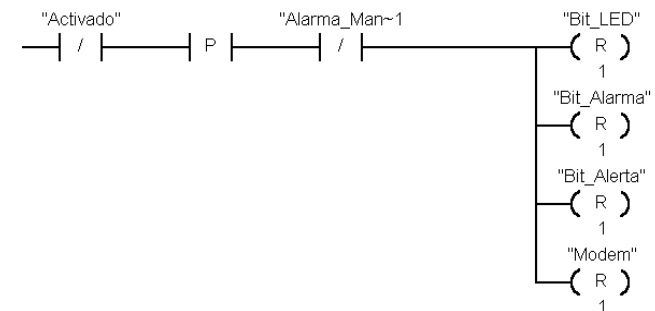
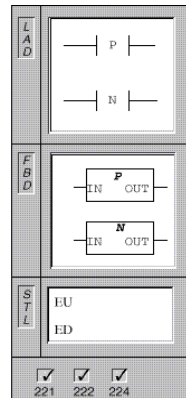


Figura 7

Se ha utilizado el contacto "Detectar flanco positivo". El contacto Detectar flanco positivo permite que fluya la corriente durante un ciclo cada vez que se produce un cambio de 0 a 1 (de "off" a "on").



En AWL, dicho contacto se representa con la operación Detectar flanco positivo (EU). Cuando se detecta un cambio de señal de 0 a 1 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

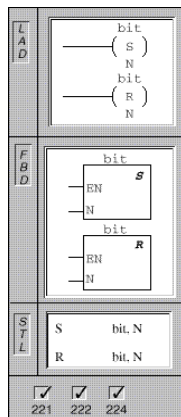
El contacto Detectar flanco negativo permite que fluya la corriente durante un ciclo cada vez que se produce un cambio de 1 a 0 (de "on" a "off").

En AWL, dicho contacto se representa con la operación Detectar flanco negativo (ED). Cuando se detecta un cambio de señal de 1 a 0 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

Figura 8 Operandos: ninguno

Con este contacto se consigue que cuando se desactiva el sistema, el contacto Activado normalmente cerrado pasa de 0 a 1, y no esta activada la alarma manual se ponen a cero las marcas correspondientes.

Se han utilizado también instrucciones de Poner a 1 o Poner a 0.



Al ejecutar las operaciones Poner a 1 y Poner a 0, se activa (se pone a 1) o se desactiva (se pone a 0) el número indicado de entradas y/o salidas (N) a partir de S_BIT, respectivamente.

Operandos:

S_BIT: I, Q, M, SM, T, C, V, S

N: IB, QB, MB, SMB, VB, AC, constante, *VD, *AC, SB

El margen de entradas y/o salidas que se pueden activar o desactivar está comprendido entre 1 y 255. Al utilizarse la operación Poner a 0, si S_BIT es un bit T (bit de temporización) o un bit C (bit de conteo), se desactivará dicho bit y se borrará el valor del temporizador o contador, respectivamente.

Figura 9

Activar el parpadeo del LED

Para realizar este parpadeo se ha utilizado un marca especial (SM0.5). Las marcas especiales (SM) ofrecen una serie de funciones de estado y control. Sirven para intercambiar informaciones entre la CPU y el programa, pudiéndose utilizar en formato de bits, bytes, palabras o palabras dobles.

El bit (SM0.5) ofrece un reloj que está activado durante 0,5 segundos y desactivado durante 0,5 segundos, siendo el tiempo de ciclo de 1 segundo. Ofrece un reloj que está activado durante 0,5 segundos y desactivado durante 0,5 segundos, siendo el tiempo de ciclo de 1 minuto.

Utilizando esta marca especial (SM0.5) y un detector de flanco, negativo para poner a cero la marca correspondiente al LED a cero y positivo para ponerla a uno, se consigue el parpadeo del LED.

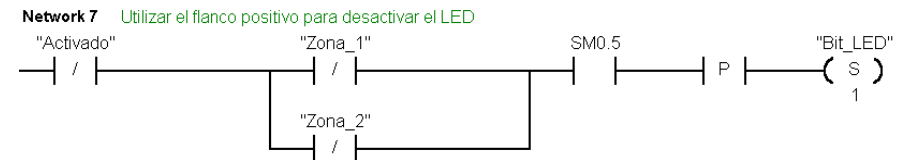
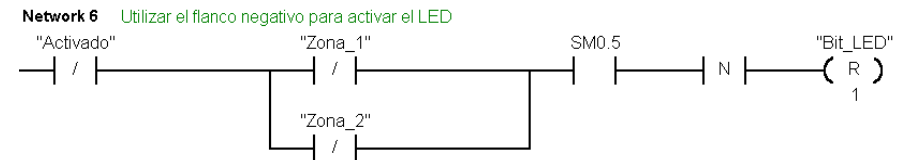


Figura 10

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 11, 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.

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.

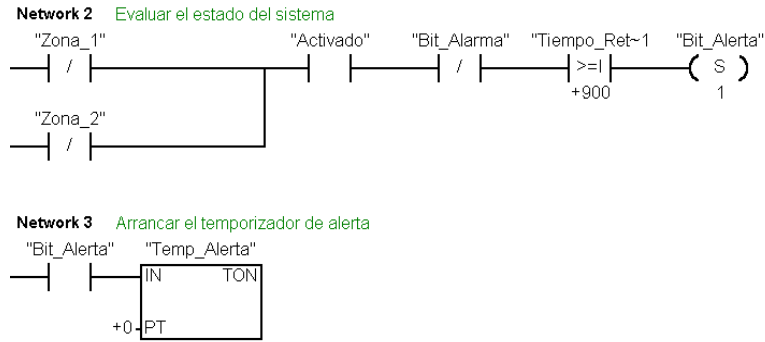


Figura 11

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 puede ver en la figura 12.

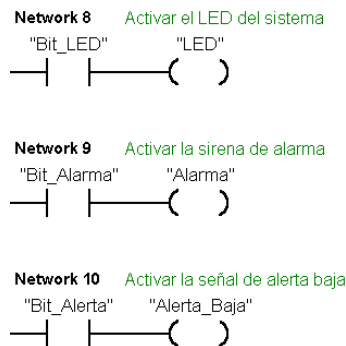


Figura 12

Como muestra la figura 13, 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.

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

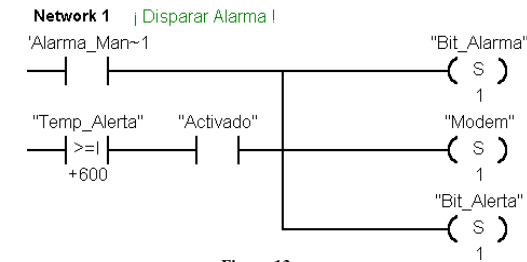


Figura 13

3.5.- 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 cada una de ellas :

Simulación de entradas y salidas

Cambiando la CPU a modo RUN, el programa allí cargado se ejecuta y se pueden ver los cambios de estado que tienen las salidas ante las variaciones simuladas mediante interruptores de las entradas de la CPU.


 Mediante estos dos botones cambiamos el modo de operación de la CPU de RUN (ejecuta el programa de control) a STOP (detiene la ejecución).

Tabla de estado

En una tabla de estado se pueden introducir direcciones para observar o modificar los valores del programa. Los valores de los temporizadores y contadores se pueden visualizar en formato binario o de palabras. Si el valor de un temporizador o contador se visualiza en formato binario, se muestra el estado de la salida ("on" u "off"). Si el valor de un temporizador o de un contador se visualiza en formato de palabra, se utiliza el valor actual.

Para crear una tabla de estado, hay que seguir los pasos siguientes:

- 1.- En el campo "Dirección", introduzca la dirección de cada valor deseado.

La mayoría de los tipos de memoria indicados en el tema Márgenes de direcciones de la CPU son válidos, con excepción de las constantes de datos, los acumuladores y los contadores rápidos.


Utilice las teclas con flecha o el ratón para seleccionar la celda que desea editar. Al comenzar a teclear su texto, el campo se borrará y se introducirán los nuevos caracteres.


Para destacar el campo, haga clic con el ratón o pulse la tecla "F2". Entonces puede utilizar las teclas con flecha para desplazar el cursor de edición hasta la posición que desea editar.

2.- Si el elemento es un bit (p.ej. I, Q ó M), en la columna "Formato" se ajusta el formato binario. Si el elemento es un byte, una palabra o una palabra doble, seleccione la celda en la columna "Formato" y haga doble clic o pulse la BARRA ESPACIADORA o la tecla INTRO para desplazarse por los formatos válidos hasta que se visualice el apropiado.

3.- Notas:

- Las direcciones se pueden seleccionar en la tabla de símbolos y copiarse luego en la tabla de estado para crear ésta última más rápidamente.
- Es posible crear varias tablas de estado. Ello permite dividir los elementos en grupos lógicos para poder visualizar cada grupo en una tabla por separado. Así se evita la necesidad de desplazarse por una sola tabla larga que incluya todos los elementos.

 Los botones de la barra de herramientas relacionados con la tabla de estado son los que aparecen en la figura.

 Este botón conmuta entre ON (iniciar la actualización continua de las tablas desde la CPU) y OFF (detener el estado de tabla). El bucle de lectura continua del estado de tabla se encuentra separado del bucle de lectura continua del estado del programa, por lo que ambos bucles se pueden ejecutar simultáneamente.


La CPU S7-200 permite forzar una o todas las E/S (bits I y Q). Además, se pueden forzar hasta un total de 16 valores internos de la memoria (V o M) o valores de E/S analógicas (AI o AQ). Los valores de las áreas de memoria V o M se pueden forzar en formato de bytes, palabras o palabras dobles. Los valores analógicos sólo se pueden forzar como palabras en bytes pares (p.ej. AIW6 o AIW14). Todos los valores forzados se almacenan en la memoria EEPROM no volátil de la CPU.


Puesto que los datos forzados se pueden modificar durante el ciclo (bien sea por el programa, por el ciclo de actualización de E/S o por el ciclo de procesamiento de las comunicaciones), la CPU aplica los valores forzados varias veces durante el ciclo.


La función "Forzar" tiene prioridad sobre las operaciones directas de lectura o de escritura. Asimismo, tiene prioridad sobre las salidas que deban adoptar un valor específico


cuando la CPU pase a modo STOP: en tal caso, la salida se ajusta al valor forzado y no al valor configurado.


A las funciones de test (Lectura sencilla, Escribir todo, Forzar, Desforzar, Desforzar todo y Leer todo) se puede acceder partiendo del menú Test o de la barra de herramientas Test.


 *Lectura sencilla* Utilice esta función si desea obtener una "instantánea", es decir, una sola actualización de todos los valores del programa. Por defecto, el estado de tabla rastrea continuamente la CPU para actualizar el estado. Si el estado de tabla está OFF y se hace clic en una tabla de estado, se habilita el botón "Lectura sencilla".


 *Escribir todo* Tras modificar los valores en la columna "Nuevo valor" de la tabla de estado, el botón "Escribir todo" se puede utilizar para transmitir los cambios deseados a la CPU.


 *Forzar* Para forzar una dirección a un valor determinado es preciso determinar primero el valor deseado, bien sea leyéndolo (si se desea forzar el valor actual) o introduciéndolo (si se desea forzar la dirección a un nuevo valor). Cuando se utiliza la función "Forzar", el valor se aplica a la dirección en cada ciclo hasta que ésta se desfuerce nuevamente. STEP 7-Micro/WIN 32 ofrece la posibilidad de forzar las direcciones tanto desde el estado del programa como desde el estado de tabla.


 *Desforzar* Seleccione una dirección y elija la función "Desforzar" para retirar el forzado de dicha dirección en particular. Alternativamente puede seleccionar (hacer clic en) un parámetro y pulsar luego el botón derecho del ratón para visualizar el menú emergente de las funciones Forzar y Desforzar.

 *Desforzar todo* Elija esta función si desea desforzar todas las direcciones. No es necesario que seleccione las direcciones individuales antes de utilizar la función "Desforzar todo".

 *Leer todo* Cuando se utiliza la función "Leer todo", en la columna "Valor actual" de la(s) tabla(s) de estado se visualiza un icono junto a cada una de las direcciones que se hayan forzado de forma explícita o implícita, o bien que se hayan forzado parcialmente de forma implícita.

 Este icono indica que esta dirección se ha forzado explícitamente. El valor de esta dirección no se puede modificar hasta que se desfuerce de nuevo.

 Este icono indica que esta dirección se ha forzado implícitamente. Una dirección se considera forzada de forma implícita si forma parte de una dirección mayor que se haya forzado explícitamente. Por ejemplo, si se fuerza VW0, entonces VB0 se forzaría implícitamente (VB0 es el primer byte de VW0). Un valor forzado implícitamente no se puede desforzar por separado. Es preciso desforzar la dirección mayor antes de poder modificar el valor de la dirección que se ha forzado implícitamente.

 Este icono indica que una parte de esta dirección se ha forzado implícitamente. Por ejemplo, si VW0 se ha forzado explícitamente, una parte de VW1 está forzada implícitamente (el primer byte de VW1 es el segundo byte de VW0). Un valor forzado implícitamente no se puede desforzar. Es preciso desforzar la dirección mayor antes de

poder modificar el valor de la dirección que se ha forzado implícitamente. Si ninguno de los tres iconos mencionados aparece en la columna "Valor actual" junto a una dirección cuando se elige el comando "Leer todo", significa que la dirección no está forzada.



Figura 14

Como ejemplo de utilización de la tabla de estado, se puede ver en la siguiente figura un detalle de la tabla de estado del programa ejemplo cuando se ha activado la entradas de Alarma_Manual (I0.3).

	Dirección	Formato	Valor actual
1	"Zona_2"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0000
2	"Zona_1"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0000
3	"Tiempo_Retardo"	Con sig	+0
4	"Temp_Alerta"	Con sig	+173
5	"Modem"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0001
6	"LED"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0000
7	"Bit_LED"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0000
8	"Bit_Alerta"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0001
9	"Bit_Alarma"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0001
10	"Alerta_Baja"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0001
11	"Alarma_Manual"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0001
12	"Alarma"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0001
13	"Activado"	Bit	2#0000_0000_0000_0000_0000_0000_0000_0000

Figura 15

En la figura 15 se puede observar que al activar la Alarma_Manual se activan los Bits de Alerta y Alarma, el Modem, la Alerta_Baja y la Alarma que corresponde con la situación descrita en el programa de control que si se activaba la alarma manual se activaban inmediatamente las alarmas y la marcación automática.

Estado del programa

Se puede hacer un seguimiento del programa de control visualmente, observando en cada momento la variación de las entradas, salidas, marcas,... en los editores KOP o FUP. Un detalle de la comprobación del programa de control mediante este método se puede observar en la figura siguiente.

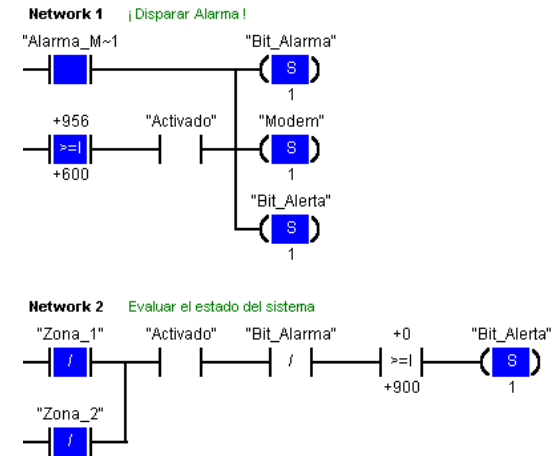


Figura 16

En la Figura 16 se puede ver un detalle de los dos primeros segmentos del programa de control cuando se ha activado la entrada de Alarma_Manual (I0.3). Los recuadros indican cuales son las variables activas en ese momento.

Mediante el siguiente botón se activa o desactiva esta opción :

Este botón conmuta entre ON (iniciar la actualización continua desde la CPU) y OFF (detener la actualización del estado del programa). Cuando el estado del programa está activado (ON), los editores KOP y FUP muestran los estados lógicos y los valores de los parámetros. Si el programa ocupa más espacio de lo que se visualiza en la pantalla, es posible utilizar la barra de desplazamiento para poder observar las diferentes secciones. Las funciones del editor se encuentran desactivadas cuando el estado del programa está ON. Para proseguir con la edición es preciso cambiar el estado del programa a OFF.

4.- Ejercicio de programación

La Figura 17 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.

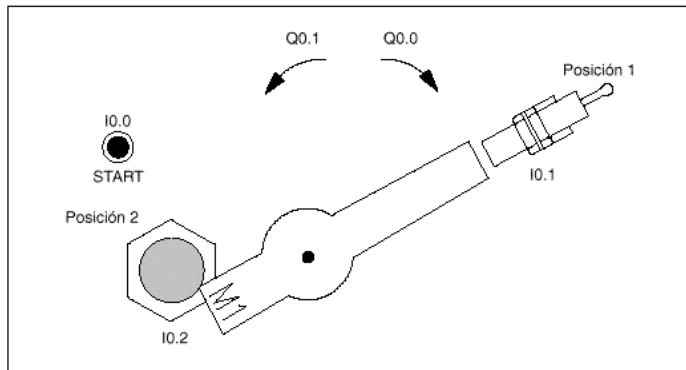


Figura 17

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:

- T37 temporizador de retardo a la conexión de resolución 100ms

Ejercicio

Ejercicio Mezclador de Pintura

La figura muestra un mezclador de pintura. En la parte superior del mezclador hay dos tuberías por donde se suministran dos componentes diferentes. En la parte inferior se aprecia una sola tubería que transporta la mezcla de pintura preparada. Por una tubería en la parte inferior del tanque se conduce la mezcla ya lista. El programa de ejemplo controla la operación de llenado, supervisa el nivel del tanque y controla un ciclo de mezcla y de calefacción como se describe a continuación:

Paso 1: Llenar el mezclador con el componente 1.

Paso 2: Llenar el mezclador con el componente 2.

Paso 3: Supervisar el nivel del mezclador para cerrar el interruptor de nivel superior.

Paso 4: Mantener el estado de la bomba si se abre el interruptor de arranque.

Paso 5: Arrancar el ciclo de mezcla y calefacción.

Paso 6: Poner en marcha el motor del mezclador y la válvula de vapor.

Paso 7: Vaciar el mezclador.

Paso 8: Contar cada ciclo.

