



## AUTÓMATAS Y SISTEMAS DE CONTROL

### PRÁCTICA 1

#### *Creación de proyectos en STEP-7*

---



### **¿Qué hay que hacer en la práctica?**

- 1) Lea los apartados 1 y 2 del guión de prácticas. En ellos se explica las características básicas del entorno de programación de autómatas STEP-7.
- 2) Lea detenidamente el ejemplo de un sistema de alarma descrito en el apartado 3. Cree la tabla de símbolos según se describe en el apartado 3.2.
- 3) Seguidamente, cree un programa en lenguaje de contactos siguiendo las instrucciones del apartado 3.4. Escriba la lógica de cada segmento leyendo la información que se proporciona.
- 4) Compruebe el funcionamiento del sistema de alarma, siguiendo las instrucciones del apartado 3.5. En el autómata se puede emular la activación de un sensor utilizando los interruptores del frontal.
- 5) Siga la evolución del programa de control mediante la función “Estado de programa”.

### **Objetivos**

- Familiarizarse con el entorno STEP-7 y conocer sus posibilidades.
- Ser capaces de escribir un programa en lenguaje KOP.
- Probar el funcionamiento de un programa en un autómata.
- Comprobar la evolución de las variables del programa en respuesta a los cambios en las entradas.

### **Índice**

1.- Estructura del programa .....	
2.- Creación de un proyecto en SETP7-Micro/Win 32 .....	
3.- Ejemplo de creación de un programa .....	
3.1.- Definir el problema.....	
3.2.- Tabla de símbolos.....	
3.3.- Tareas básicas del programa para el sistema de alarma .....	
3.4.- Diseñar la lógica de control .....	
3.5.- Comprobación del funcionamiento del programa .....	

---

## 1.- Estructura del programa

*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. Otros elementos del programa se verán en prácticas siguientes.

## 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 está 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.

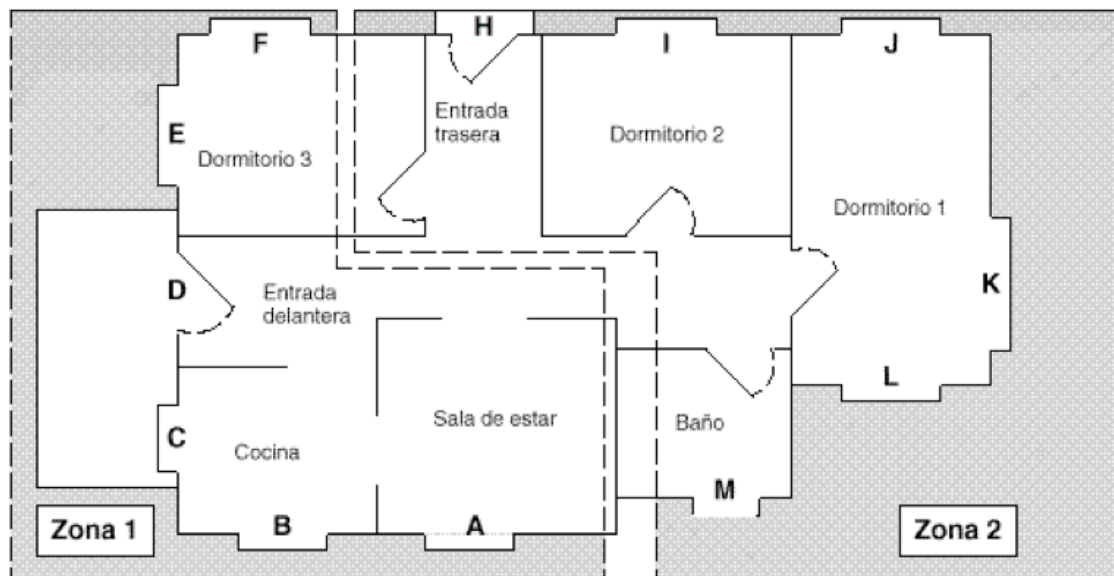


*Tabla de estado* : Las informaciones contenidas en la tabla de estado no se cargan en la CPU.

### 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 siguiente. 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.



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

---



## Área de Ingeniería de Sistemas Y Automática

### Ingeniería Industrial

#### 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.e. para arrancar una marcación automática).

#### La lógica del control 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:
    - o 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
-



## Área de Ingeniería de Sistemas Y Automática

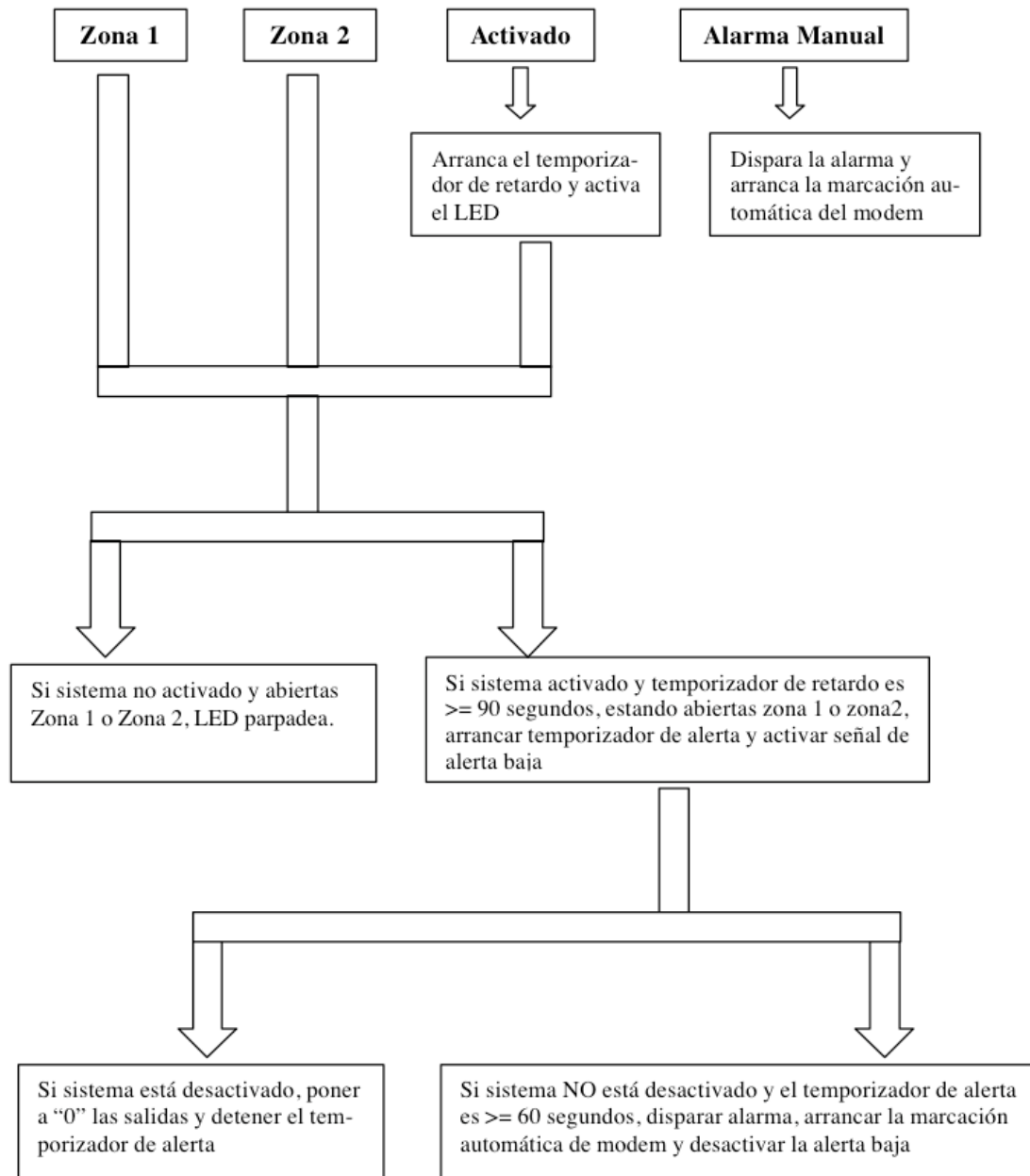
### Ingeniería Industrial

**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, una vez disparada la alarma (**Q0.1**), se desactiva el sistema (girando la llave a la posición "off", lo que desactiva **I0.2**), 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. El siguiente diagrama puede ayudar a comprender la lógica que se pretende programar:

---



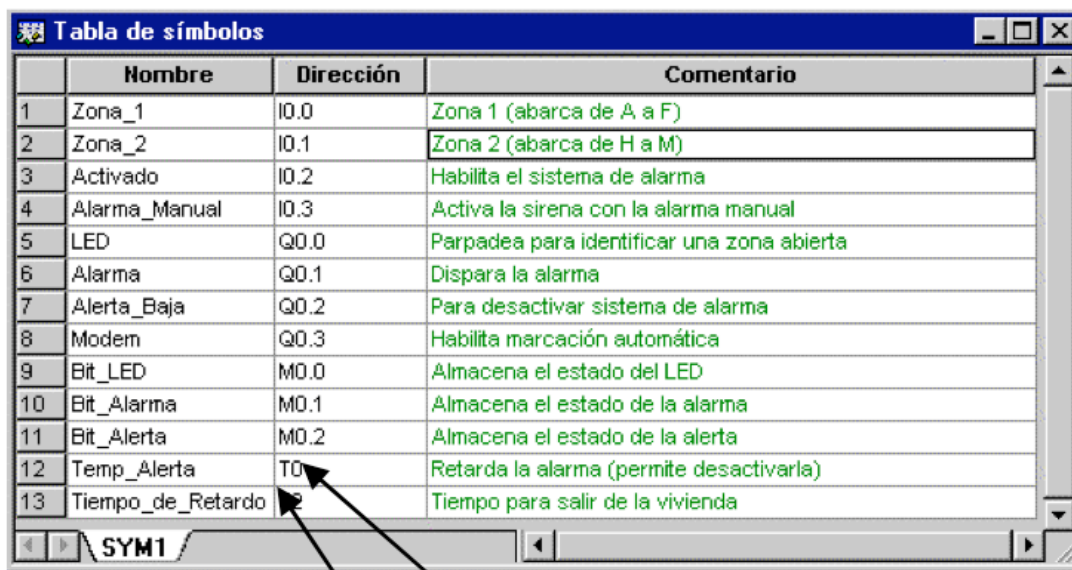
### 3.2.- Tabla de símbolos

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. Abra la tabla de símbolos y cree las siguientes:



	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

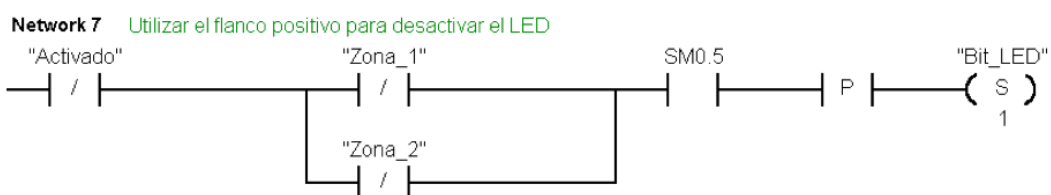
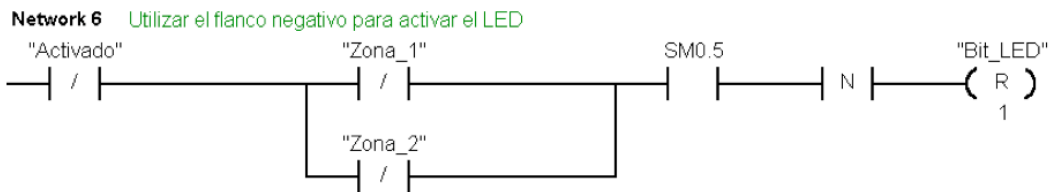
T38    T37

### 3.3.- Diseñar la lógica de control

Se propone ahora que el alumno escriba el programa de control para conseguir que el sistema funcione según las indicaciones. Se insta al alumno a que realice primero una parte sencilla del programa de control. A continuación, deberá probar su funcionamiento sobre el autómeta (apartado 3.4). De esta manera será fácil corregir pequeños fallos de programación.

Como ejemplo se detalla el código KOP para hacer que el LED parpadee cuando el sistema de alarma no esté activado. Recuérdese que la marca especial SM0.5 está activada durante 0,5s y desactivada durante 0,5s.





### 3.4.- Comprobación del funcionamiento del programa

Se propone aquí descargar el programa en el autómatas y, a continuación, ejecutarlo. Se podrá comprobar cómo reacciona el programa ante determinadas entradas.



**Cargar programa en la CPU del autómatas:** Se debe indicar la descarga del programa al autómatas, conectado al PC mediante cable serie y convertidor RS-232/RS-485. Ojo: cargar desde CPU carga el programa residente en la memoria del autómatas, y sobrescribe el que existiera en nuestro proyecto.



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



**Visualización de las variables de programa** de forma continua. A través de la comunicación RS-485 del autómatas se pasa en "tiempo real" el valor de las variables en la CPU del autómatas.



## Área de Ingeniería de Sistemas Y Automática

### Ingeniería Industrial

Con estas directrices el alumno debe ser capaz de terminar el programa en KOP que se ha encomendado, comprobando su funcionamiento sobre el autómeta.

---