

# Elementos básicos de programación de autómatas

2014

## 1. Introducción

A continuación se ofrece una breve descripción, ilustrada con ejemplos, de las operaciones básicas de todo Autómata Programable:

- Funciones lógicas
- Consideraciones sobre el emisor
- Biestables
- Temporizadores

Para esta introducción se ha elegido el lenguaje gráfico de contactos, también conocido como lenguaje de escalera, *ladder diagram* o LD. Es el lenguaje más intuitivo para personal familiarizado con esquemas eléctricos, y fácilmente aplicable a problemas sencillos. En los equipos de Siemens, con los que se realizarán las prácticas, recibe el nombre de KOP (Kontakt Plan) y su sintaxis será la empleada en el presente documento.

## 2. Funciones lógicas

Los ejemplos del apartado 2.1 muestran la implementación de las funciones lógicas básicas AND y OR usando el lenguaje de programación de contactos (KOP). Para todos ello se han supuesto sensores normalmente abiertos. Obsérvese que la condición AND se corresponde con una configuración de contactos en serie, colocándose éstos en paralelo en el caso de la OR. Asimismo se muestra la solución en KOP para una función más compleja. La resolución de un problema práctico sobre lógica combinacional se describe en el ejemplo del apartado 2.2.

### 2.1. Ejemplos de funciones lógicas

$$Q = A \cdot B$$

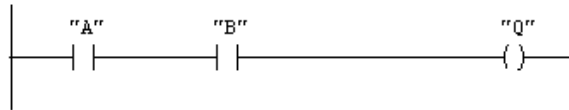


Figura 1: Implementación en KOP de la función AND

$$Q = A + B$$



Figura 2: Implementación en KOP de la función OR

$$Q = (A \cdot \bar{B} + C) \cdot D$$

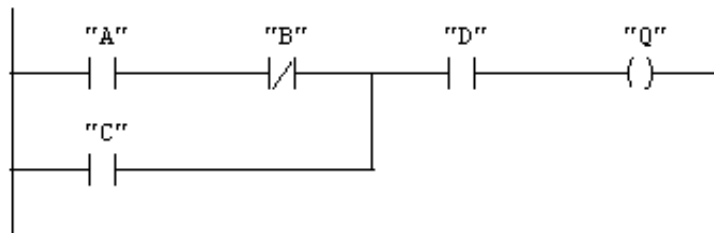


Figura 3: Implementación en KOP de una función lógica

### 2.2. Ejemplo de aplicación de lógica combinatorial

Activación y desactivación de la iluminación de un local, mediante el accionamiento de tres interruptores de configuración normalmente abiertos. Supóngase la sala de un museo en la figura 4. Se quiere que la iluminación no esté activada cuando se encuentre vacía. Para ello, cuando se entra en la sala, se pulsa el interruptor de la puerta por la que se acceda para que se encienda la luz. Cuando se abandone la

sala se debe accionar el interruptor correspondiente a la puerta por la que se sale, para apagar la luz (suponiendo que no quede nadie dentro).

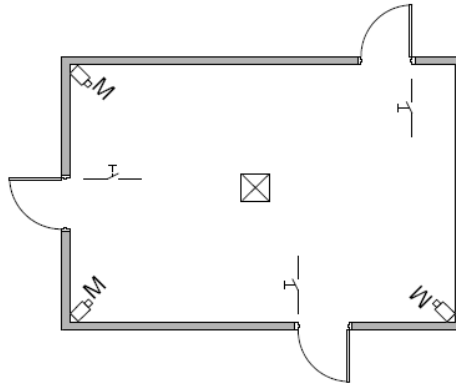


Figura 4: Habitación del museo

La función lógica correspondiente a la tabla de verdad que aparece en la figura 5 está expresada en la ecuación (1). La solución a la implementación de tres interruptores para la iluminación de una habitación se refleja en la figura 5.

$$Q = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C \tag{1}$$

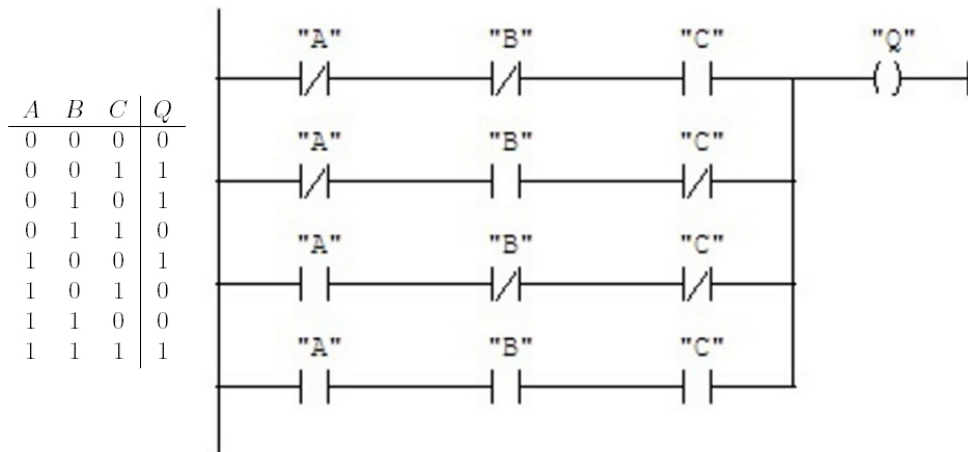


Figura 5: Tabla de verdad y solución en KOP

Si bien puede ser útil el empleo de tablas de verdad, no lo es tanto su simplificación (por Karnaugh u otros medios). No es tan importante la minimización del número de 'puertas lógicas' como lo sería en un circuito implementado en hardware. En la mayor parte de las situaciones con tecnología programada es preferible una

mayor claridad e inteligibilidad del algoritmo y/o un menor tiempo de desarrollo, que el ahorro de unas líneas de código.

### 3. Consideraciones sobre el emisor

A la hora de programar es necesario tener en cuenta la lógica con que están implementados los sensores. Hay que pensar que el autómata no ve más allá de las bornas de sus tarjetas de entrada. Por lo tanto habrá que adecuar el programa según los elementos emisores de señal (sensores) sean de nivel activo alto o bajo.

En los ejemplos anteriores se ha supuesto que todos los sensores eran de nivel activo alto. Pero no siempre es así, ni tan siquiera es lo más habitual. En muchos casos la selección de sensores de nivel activo alto o bajo está condicionada por motivos de seguridad.

A continuación se muestra un ejemplo. Se pretende gestionar el encendido y apagado de una bombilla en función de la información obtenida de los sensores Sa y Sb cuyo estado de reposo se observa en el cableado de las entradas de la figura 6. La ley de control es la siguiente: *la bombilla sólo debe lucir cuando, simultáneamente, Sa esté activo y Sb no lo esté.*

Evidentemente la solución programada será diferente según cual sea la configuración de los sensores. Para resolver el problema conviene fijarse en qué señal aparece en los bornes del autómata (lo que en realidad ve en sus entradas), según el estado de detección de los sensores y la ley de control.

En la situación de la figura 6, Sa es de nivel activo alto, o normalmente abierto (en adelante NA) y Sb es de nivel activo bajo, o normalmente cerrado (en adelante NC). Entonces, si Sa está activo aparecerá un nivel de tensión alto en la entrada %I0.0 y si Sb no detecta, por ser de nivel activo bajo aparecerá también un nivel de tensión alto en la entrada %I0.1. Así pues, la condición que habrá que programar para que se encienda la bombilla es que %I0.0 esté a 1 y %I0.1 esté a 1.

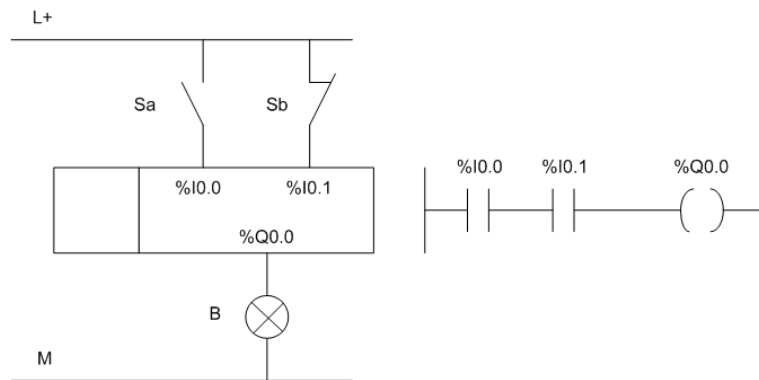


Figura 6: La bombilla luce si Sa está activo y Sb no

Para evitar problemas, es necesario hablar con propiedad. Se puede decir que los sensores conectados al AP son NA o NC, pero cuando se habla de los elementos del programa es peligroso emplear esa denominación. En este caso es más apropiado hablar de *consultas de estado* y decir que si %I0.0 está en estado alto y %I0.1 está también en estado alto se activará la salida %Q0.0.

## 4. Biestables

Los biestables permiten mantener el estado de una variable aún cuando sus entradas se desactiven. Son muy usados cuando se dispone de pulsadores, para memorizar las órdenes.

Poseen dos entradas (Set y Reset) que se usan para poner a nivel alto o a nivel bajo la variable asociada al biestable.

En KOP se encuentran disponibles dos tipos de biestables: de inscripción prioritaria (RS) y de borrado prioritario (SR) que se diferencian en la forma de resolver la contradicción que se produce cuando ambas entradas están activas. La figura 7 ilustra su funcionamiento.

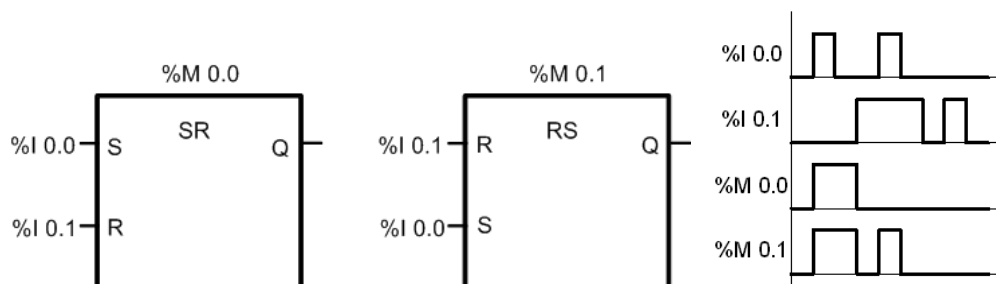


Figura 7: Biestables SR, RS y su cronograma

*Nota sobre la implementación de los biestables en STEP 7.* Obsérvese que la variable que se desea poner a uno o a cero se escribe sobre la caja del biestable. La patilla Q se puede emplear opcionalmente si se quiere enlazar la operación con otras, o bien asignar el mismo resultado a una segunda variable, si no se deja en blanco.

### 4.1. Ejemplo del uso de biestables

La figura 8 muestra un depósito llenado por una electrobomba que pretende ser controlada mediante el uso de dos sensores de nivel (NA). El estado de disponibilidad o no de servicio lo proporciona un selector de dos posiciones: Marcha-Paro. La electrobomba se pondrá en marcha cuando el nivel descienda por debajo del mínimo y se apagará cuando: bien se alcance el máximo, bien salte el térmico (NC), o bien sea desconectada mediante el selector. Si la electrobomba está en servicio deberá lucir una lámpara indicadora. Si salta el térmico se encenderá una lámpara de aviso.

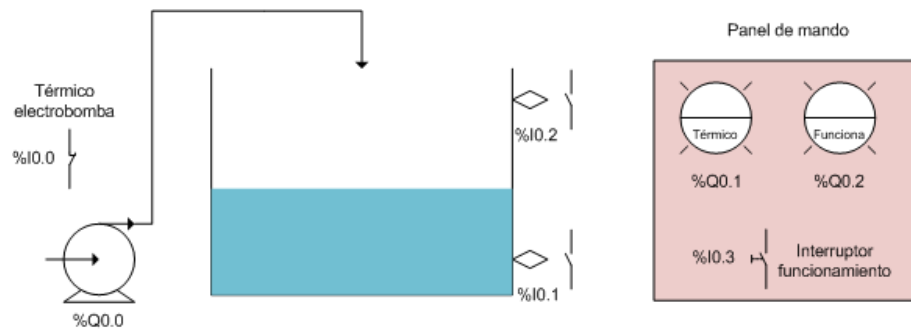


Figura 8: Depósito que se pretende controlar

La tabla de símbolos que permiten etiquetar las direcciones absolutas del autómata aparece en la figura 9 y la solución al ejemplo planteado en la figura ??.

Tabla de variables estándar			
	Nombre	Tipo de datos	Dirección ▲
◀	Térmico	Bool	%I0.0
◀	Nivel Mínimo	Bool	%I0.1
◀	Nivel Máximo	Bool	%I0.2
◀	Marcha-Paro	Bool	%I0.3
◀	Electrobomba	Bool	%Q0.0
◀	Lámpara Térmico	Bool	%Q0.1
◀	Lámpara Servicio	Bool	%Q0.2

Figura 9: Tabla de símbolos

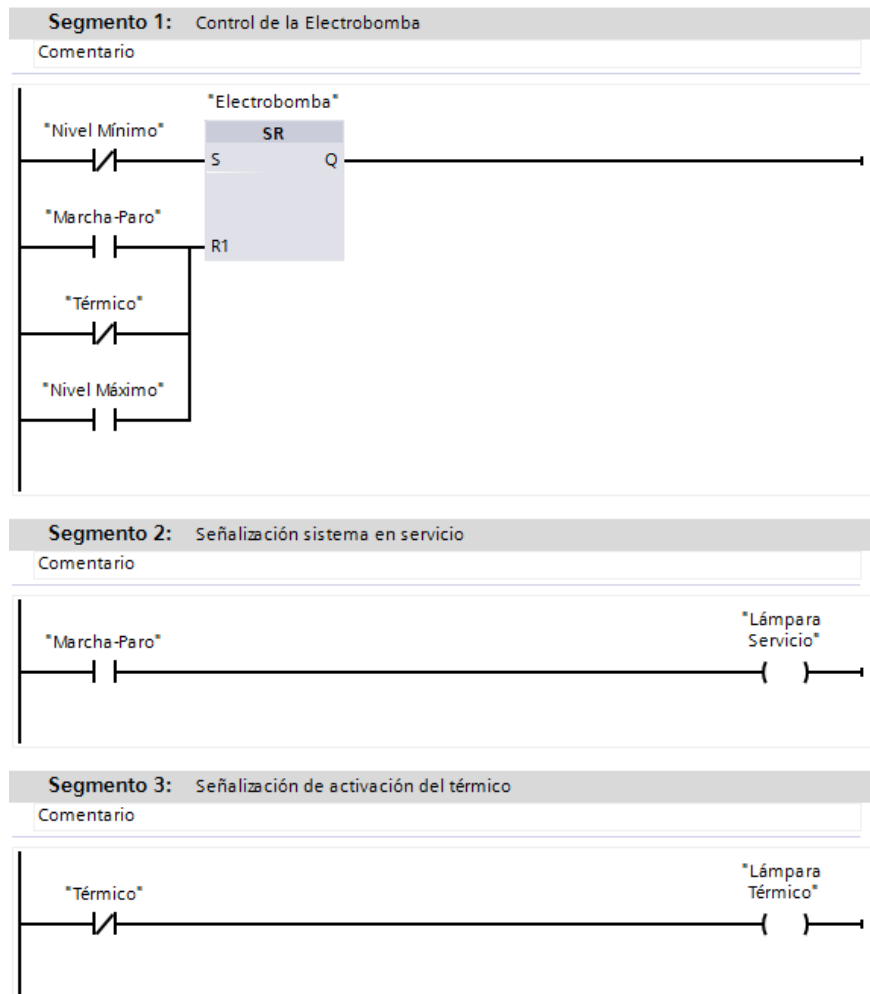


Figura 10: Solución KOP

## 5. Temporizadores

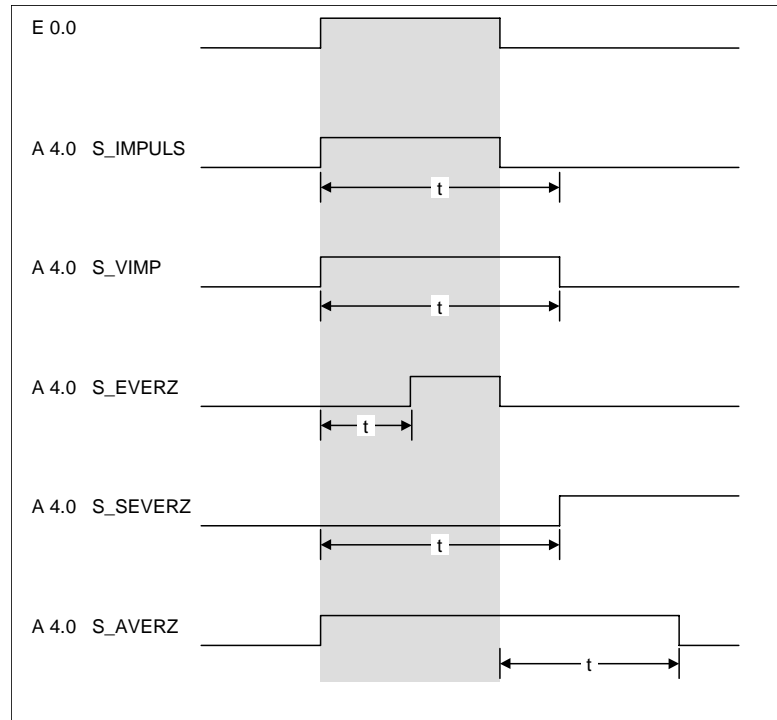
Las instrucciones adecuadas para gestionar la activación o desactivación de determinadas acciones en función del tiempo transcurrido desde un determinado evento son los *temporizadores*.

En el autómata, un temporizador se denomina con la letra **T** y un número a continuación (e.g. T1). Ocupan una palabra de memoria, donde almacenan la información de su estado. *Error típico: declarar T1 como un símbolo, cuando en realidad es el nombre de una variable, como por ejemplo E1.0.*

STEP 7 dispone de 5 tipos de temporización (SI, SV, SE, SS y SA). En la figura 11 se muestra su funcionamiento. A partir de ahí se puede elegir el más adecuado en cada caso.

La activación de los temporizadores se produce cuando ocurre un flanco ascen-

dente (la señal en el ciclo anterior valía cero y en el actual uno) en su patilla de entrada. Esto es cierto en todos los casos excepto en el SA, que arranca con flanco descendente.



Temporizadores	Descripción
<b>S_IMPULS</b> Temporizador de impulso	El tiempo máximo que la señal de salida permanece a 1 corresponde al valor de temporización t programado. La señal de salida permanece a 1 durante un tiempo inferior si la señal de entrada cambia a 0.
<b>S_VIMP</b> Temporizador de impulso prolongado	La señal de salida permanece a 1 durante el tiempo programado, independientemente del tiempo en que la señal de entrada esté a 1.
<b>S_EVERZ</b> Temporizador de retardo a la conexión	La señal de salida es 1 solamente si ha finalizado el tiempo programado y la señal de entrada sigue siendo 1.
<b>S_SEVERZ</b> Temporizador de retardo a la conexión con memoria	La señal de salida cambia de 0 a 1 solamente si ha finalizado el tiempo programado, independientemente del tiempo en que la señal de salida esté a 1.
<b>S_AVERZ</b> Temporizador de retardo a la desconexión	La señal de salida es 1 cuando la señal de entrada es 1 o cuando el temporizador está en marcha. El temporizador arranca cuando la señal de entrada cambia de 1 a 0.

Figura 11: Tipos de temporización



### 5.1. Ejemplo del uso de temporizadores

Se pretende controlar una cinta de transporte mediante un interruptor que gestione su arranque y parada. La cinta debe comenzar a funcionar 30 segundos después que se active el interruptor. Además, se debe de hacer sonar una bocina durante los 10 segundos previos al arranque de la cinta.

La tabla de símbolos que permiten etiquetar las direcciones absolutas del autómeta aparece en la figura 12.

Símbolo $\Delta$	Dirección	Tipo de dato
Cycle Execution	OB 1	OB 1
Marcha-Paro	E 124.7	BOOL
Cinta	A 124.7	BOOL
Bocina	A 124.6	BOOL

Figura 12: Tabla de símbolos

La solución al ejemplo planteado aparece en la figura 14. El cronograma asociado aparece en la figura 13.

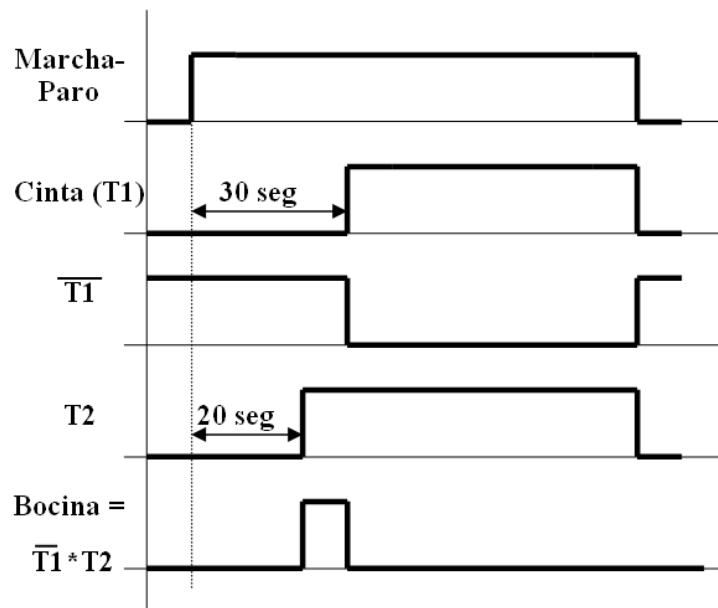


Figura 13: Cronograma

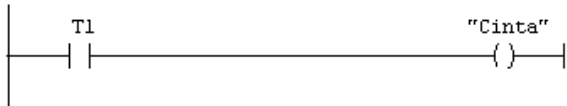
**Segm. 1 : Título:**

Temporización tipo SE de 30 segundos en la variable T1



**Segm. 2 : Título:**

Control de la cinta



**Segm. 3 : Título:**

Temporización tipo SE de 20 segundos en la variable T2



**Segm. 4 : Título:**

Control de la bocina

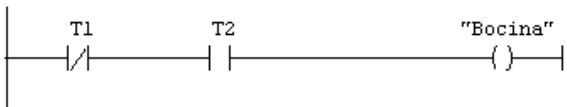


Figura 14: Solución KOP

## 6. Para saber más

Puede descargar el manual: *Esquema de contactos (KOP) para S7-300 y S7-400*

[http://support.automation.siemens.com/WW/llisapi.dll/csfetch/18654395/KOP\\_s.pdf](http://support.automation.siemens.com/WW/llisapi.dll/csfetch/18654395/KOP_s.pdf)

?func=cslib.csFetch&nodeid=18652981&forcedownload=true