

1 Elementos de un proceso a automatizar

Un proceso a automatizar requiere tener en cuenta un conjunto de elementos, cada uno de los cuales realiza su función dentro del proceso. Podríamos representar el proceso con el sistema de control y estos elementos mediante el gráfico de la figura 1.

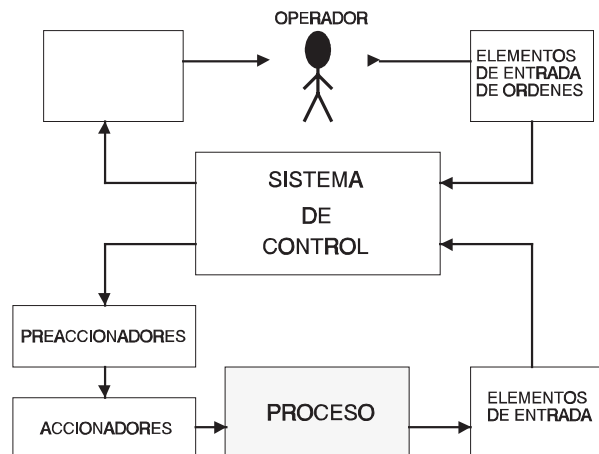


Fig. 1

Seguidamente trataremos cada uno de los elementos que aparecen en este gráfico a fin de tener una visión general de las necesidades y posibilidades de la automatización industrial.

1.1 Elementos de entrada de órdenes

Son los que permiten al operador la entrada de datos y órdenes al sistema. Podemos clasificarlos en dos categorías: binarios y numéricos (o alfanuméricos). Los elementos binarios son los que nos permiten entrar órdenes del tipo sí/no (cierto/falso, activado/desactivado, etc.). Entre ellos destaca el

pulsador que es el más usado en el entorno industrial, sin embargo, podemos destacar también los interruptores, los conmutadores, etc.

Los elementos numéricos permiten la entrada de números (codificados de diversas formas, como ya veremos). Los más corrientes son los preselectores digitales que son elementos con una rueda numerada (habitualmente del 0 al 9) que se puede hacer girar en uno u otro sentido mediante unos botones; el valor que aparece en el frontal del preselector es el que se envía al sistema de control. También destacan los teclados numéricos.

Los elementos alfanuméricos permiten entrar letras y números (a menudo codificados en código ASCII), en la mayor parte de los casos se trata de teclados.

1.2 Elementos de entrada de información

Los elementos de entrada de información se pueden clasificar según el tipo de señal que faciliten o según la magnitud que indiquen.

Según el tipo de señal podemos distinguir los binarios, los numéricos y los analógicos. Los binarios comparan la magnitud con una referencia (umbral) y la salida corresponde al resultado de la comparación (mayor/menor); un ejemplo podría ser un termostato. Los numéricos facilitan un código numérico que corresponde al valor de la magnitud leída; un ejemplo podría ser un codificador de posición axial absoluto (*encoder*).

Los analógicos dan una señal en forma de tensión eléctrica (o de corriente eléctrica) proporcional al valor de la magnitud; por ejemplo podemos citar un transductor de par mecánico. Las informaciones de tipo analógico se presentan habitualmente en uno de los siguientes cuatro rangos: 0 a 10 V, 0 a 20 mA, -10 a 10 V y 4 a 20 mA

Las señales en corriente (0-20 mA y 4-20 mA) tienen la ventaja respecto a los de tensión de no verse afectados por la longitud de los conductores; además el tipo 4-20 mA facilita la detección de averías dado que el valor 0 mA sólo se puede obtener en caso de mal funcionamiento. Por estos motivos el tipo 4-20 mA es el más usado en el entorno industrial.

Las magnitudes a detectar o medir son muchas, podemos destacar algunas: Temperatura, presión, caudal, pH, posición, velocidad, aceleración, fuerza, par mecánico, deformación, corriente eléctrica, tensión eléctrica, potencia, iluminación, presencia (final de carrera), proximidad (inductivos, capacitivos,...), etc.

Además podemos incluir dentro de esta categoría los avisos (todos ellos binarios) procedentes de los preaccionadores; como podrían ser estado de contactores, fusibles, relés térmicos, etc.

1.3 Elementos de salida de información

Se encargan de la comunicación con el operador. Se pueden clasificar de forma similar a los de entrada de órdenes. Los elementos binarios son los que dan informaciones del tipo sí/no (cierto/falso, activado/desactivado, etc.). Entre ellos destaca el piloto visualizador, pero también podemos citar los timbres, las sirenas, etc.

Los elementos numéricos y alfanuméricos permiten la visualización de números o de números y texto. Los más sencillos son los *displays* de 7 segmentos y los *displays* alfanuméricos, pero también destacan las pantallas de cristal líquido (LCD) que permiten mensajes más largos y complejos o los monitores (parecidos a pantallas de ordenador) que permiten presentar gráficos (por ejemplo sinópticos) fijos o en movimiento.

1.4 Preaccionadores y accionadores

Los accionadores son los encargados de actuar sobre el proceso. A menudo los accionadores no son directamente conectables al sistema de control y requieren preaccionadores.

Por ejemplo, un motor eléctrico necesita un contactor (o un interruptor) o un variador de velocidad para poder funcionar, un cilindro neumático necesitará una válvula distribuidora; un cilindro o un motor hidráulico necesitará una válvula distribuidora o una válvula proporcional; a un calentador eléctrico le hará falta un contactor o un variador de tensión; etc. Fijémonos que entre los citados hay elementos binarios (contactor, válvula distribuidora) y elementos de pedido analógico (variador de tensión, válvulas proporcionales).

1.5 Sistema de tratamiento de la información

El sistema de tratamiento de la información establece la forma en que se tienen que combinar las entradas de información a fin de activar las salidas del proceso. Cuando una combinación de entradas siempre da lugar a la misma combinación de salidas se dice que el proceso es de tipo combinacional.

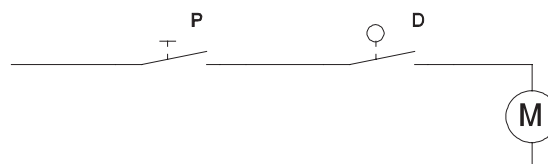


Fig. 2

El circuito de una máquina de picar carne (figura 2) sería un circuito combinacional. Su tabla de funcionamiento sería la tabla 1.

Tabla 1

P	D	M
No	No	No
Si	No	No
No	Si	No
Si	Si	Si

Cuando una combinación de entradas da lugar a combinaciones diferentes de salidas dependiendo de la historia que ha seguido el proceso para llegar donde está, se dice que el proceso es de tipo secuencial. Un ejemplo de circuito secuencial podría ser un circuito marcha-parada (figura 3).

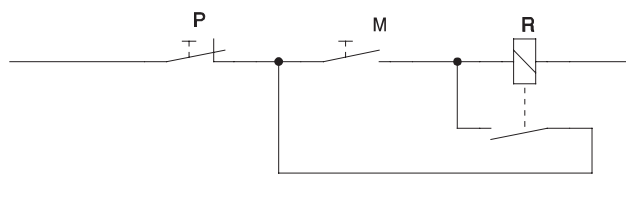


Fig. 3

Tabla 2

P	M	R	R
No	No	No	No
No	No	Si	Si
No	Si	No	Si
No	Si	Si	Si

P	M	R	R
Si	No	No	No
Si	No	Si	No
Si	Si	No	No
Si	Si	Si	No

En este circuito el pulsador M activa un relé R. Un contacto de este relé hace un puente al pulsador M. Esto hace que el pulsador se pueda dejar sin que se desactive el relé. Para desactivar el relé es necesario apretar el pulsador P. Esto se ve reflejado en la tabla de funcionamiento (tabla 2) donde observamos que entre las condiciones aparece el propio relé.

2 Álgebra de Boole

El álgebra de Boole describe el conjunto de propiedades que cumplen las funciones básicas de la lógica. Dado que todos nuestros elementos sólo pueden tener dos valores (1 y 0) podemos decir que nuestra álgebra es de tipo binario.

Los elementos de entrada de señales a un circuito (pulsadores, interruptores, etc.) pueden tener dos estados: conectado (cerrado) y desconectado (abierto). Los elementos de salida (contactores, relés, electroválvulas, etc.) también tienen dos estados: excitado (trabajo) y desexcitado (reposo).

El primer estado (conectado, excitado) lo representamos por 1 y el segundo (desconectado, desexcitado) por 0. Estas cifras (1 y 0) corresponden a la forma que la lógica usa para representar cualquiera de las combinaciones sí-no, cierto-falso, presente-ausente, conectado-desconectado, etc.

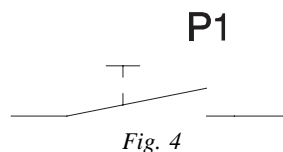
Para describir la ecuación de un circuito, siempre partiremos de la situación de reposo (y, por tanto, la más segura) de sus componentes.

2.1 Elementos básicos de una maniobra eléctrica

Definimos como elementos base de una maniobra eléctrica los siguientes:

Pulsador normalmente abierto

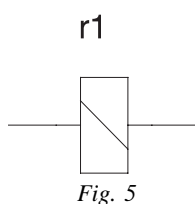
Cuando está apretado, deja pasar corriente por él y podemos escribir $P1=1$. Si no está apretado, no deja pasar corriente y escribimos $P1=0$.



Bajo esta simbología incluimos, además del pulsador, otros elementos que cambian de estado por actuaciones externas al circuito de maniobra (final de carrera, detector de proximidad, etc.).

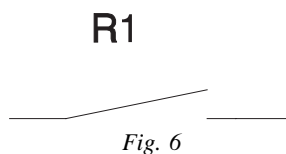
Bobina de relé (o de contactor)

Cuando pasa corriente por él, está excitada y decimos que $r1=1$. Si no pasa corriente por él, está desexcitada y escribimos $r1=0$. Dentro de este elemento consideramos otros elementos de salida (electroválvulas, etc.) que al hacer los esquemas dibujaremos con un símbolo diferente según se tercie.



Contacto normalmente abierto de un relé (o contactor)

Este caso es similar al del pulsador normalmente abierto pero el cambio de estado es por causas internas al circuito de maniobra. Cuando el relé está excitado ($r1=1$), deja pasar corriente por el contacto y decimos $R1=1$. En caso contrario ($r1=0$) tenemos $R1=0$.



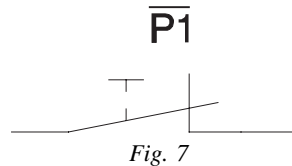
Conviene mencionar que estamos representando las bobinas de los relés por letras minúsculas y sus contactos por mayúsculas.

2.2 Elementos complementarios de una maniobra eléctrica

Además de los elementos citados, conocemos otros dos tipos de elementos: pulsador normalmente cerrado y contacto normalmente cerrado de un relé. Estos elementos son los complementarios de dos de los elementos anteriores y los representaremos con una raya encima. Así:

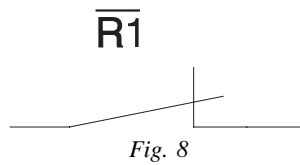
Pulsador normalmente cerrado

Cuando está apretado, es decir cuando $\overline{P1} = 0$, no pasa corriente por él. Si no está apretado, es decir si $\overline{P1} = 1$, pasa corriente por él.



Contacto normalmente cerrado de relé

Cuando el relé está desactivado, es decir si $r1 = 0$ será $\overline{R1} = 1$ y pasa corriente por él. Si está activado, es decir si $r1 = 1$, será $\overline{R1} = 0$ y no pasa corriente por él.



2.3 Elementos básicos del álgebra de Boole

En el caso de las maniobras eléctricas, el álgebra de Boole permite escribir las siguientes expresiones como representación de los esquemas que las acompañen.

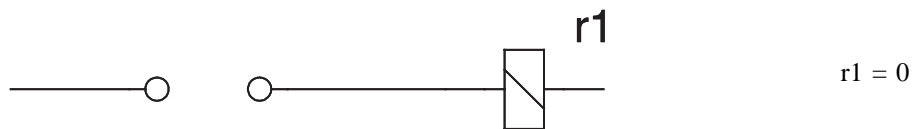


Fig. 9

Relé siempre desactivado

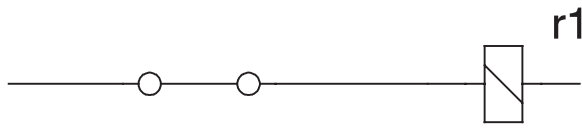


Fig. 10

$$r1 = 1$$

Relé siempre activado

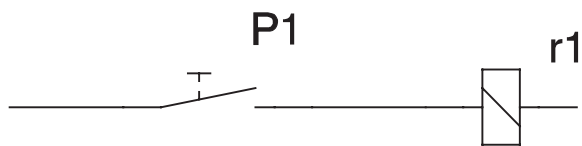


Fig. 11

$$r1 = P1$$

Relé concordante con pulsador

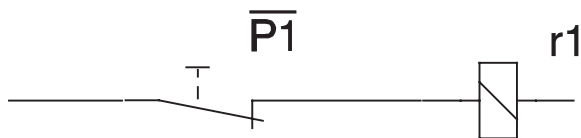


Fig. 12

$$r1 = \overline{P1}$$

Relé discordante con pulsador

2.4 Funciones base del álgebra de Boole

Las funciones base del álgebra son aquellas sobre las cuales se definen las propiedades que caracterizan cada álgebra. Las funciones de nuestra álgebra serán AND (y-lógica) y OR (o-lógica).

Las propiedades de las funciones base se exponen, como se hace con otras álgebras, para los casos de dos y tres elementos, según se tercie. Su generalización es evidente.

Para clarificar conceptos vale la pena mencionar en el caso del álgebra de los números reales, las funciones son adición (suma) y multiplicación (producto).

2.4.1 Función AND

Corresponde al caso en que hay dos pulsadores (P1, P2) en serie. Este circuito deja pasar corriente cuando P1 y P2 están apretados; tal como muestra el circuito de la figura 13 y la tabla de verdad que le acompaña (tabla 3). Para simplificar la función AND a menudo se representa con un punto \cdot .

Tabla 3

P1	P2	r1
0	0	0
0	1	0
1	0	0
1	1	1

$$r1 = P1 \text{ AND } P2 = P1 \cdot P2$$

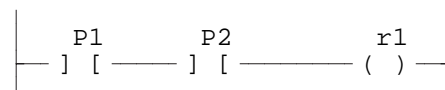


Fig. 13

La función AND viene caracterizada por una serie de propiedades que comentamos a continuación.

La propiedad conmutativa dice que el orden de los operandos no altera el resultado.

$$P1 \cdot P2 = P2 \cdot P1 \tag{2.1}$$

La propiedad asociativa dice que en caso de tener que operar con tres elementos, no importa el orden con que se hacen las operaciones parciales.

$$(P1 \cdot P2) \cdot P3 = P1 \cdot (P2 \cdot P3) \tag{2.2}$$

El 0 es un elemento nulo ya que operado con cualquier otro el resultado es 0.

$$P1 \cdot 0 = 0 \tag{2.3}$$

El 1 es el elemento neutro ya que operado con una variable la deja invariante.

$$P1 \cdot 1 = P1 \tag{2.4}$$

Presenta idempotencia dado que al operar una variable consigo se obtiene la misma variable.

$$P1 \cdot P1 = P1 \tag{2.5}$$

Cada elemento tiene un elemento simétrico que operado consigo da el elemento nulo.

$$P1 \cdot \overline{P1} = 0 \quad (2.6)$$

2.4.2 Función OR

Corresponde al caso en que hay dos pulsadores (P1, P2) en paralelo. Este circuito deja pasar corriente cuando P1 o P2 (o ambos) están apretados; tal como muestra el circuito de la figura 14 y la tabla de verdad (tabla 4) que le acompaña. Para simplificar la función OR a menudo se representa con una cruz +.

Tabla 4

P1	P2	r1
0	0	0
0	1	1
1	0	1
1	1	1

$$r1 = P1 \text{ OR } P2 = P1 + P2$$

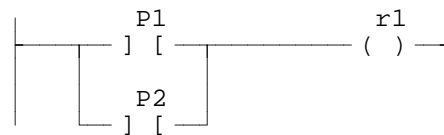


Fig. 14

La función OR viene caracterizada por una serie de propiedades que comentamos a continuación.

La propiedad conmutativa dice que el orden de los operandos no altera el resultado.

$$P1 + P2 = P2 + P1 \quad (2.7)$$

La propiedad asociativa dice que en caso de tener que operar con tres elementos, no importa el orden con que se hacen las operaciones parciales.

$$(P1 + P2) + P3 = P1 + (P2 + P3) \quad (2.8)$$

El 1 es un elemento nulo ya que operado con cualquier otro el resultado es 1.

$$P1 + 1 = 1 \quad (2.9)$$

El 0 es el elemento neutro ya que operado con una variable la deja invariante.

$$P1 + 0 = P1 \quad (2.10)$$

Presenta idempotencia dado que al operar una variable consigo se obtiene la misma variable.

$$P1 + P1 = P1 \quad (2.11)$$

Cada elemento tiene un elemento simétrico que operado consigo da el elemento nulo.

$$P1 + \overline{P1} = 1 \quad (2.12)$$

2.4.3 Propiedades distributivas

Las propiedades distributivas son aquellas en que intervienen las dos funciones sobre tres elementos.

Propiedad distributiva de \cdot respecto de $+$

$$P1 \cdot (P2 + P3) = P1 \cdot P2 + P1 \cdot P3 \quad (2.13)$$

Propiedad distributiva de $+$ respecto de \cdot

$$P1 + (P2 \cdot P3) = (P1 + P2) \cdot (P1 + P3) \quad (2.14)$$

3 Sistemas de numeración y representación

En la vida cotidiana las personas cuentan según el sistema decimal, es decir usando 10 cifras (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9). En vez de este sistema se podía haber usado arbitrariamente cualquier otro pero éste fue el adoptado, probablemente porque en la época en que se hizo se contaba con los dedos de las dos manos.

En el sistema decimal, para expresar un número mayor que 9 se ponen dos cifras de lado, la primera de ellas expresa el número entero de veces que es necesario contar hasta 10 y la segunda el resto que es necesario añadirle; así podemos contar hasta 99. Para contar hasta más arriba sólo es necesario ir aumentando el número de cifras.

Por ejemplo el número cuarenta y siete mil quinientos ochenta y tres se descompone según puede verse en la figura 15.

4 decenas de millar	=	$4 \cdot 10^4$	=	$4 \cdot 10000$	=	40000
7 unidades de millar	=	$7 \cdot 10^3$	=	$7 \cdot 1000$	=	7000
5 centenas	=	$5 \cdot 10^2$	=	$5 \cdot 100$	=	500
8 decenas	=	$8 \cdot 10^1$	=	$8 \cdot 10$	=	80
3 unidades	=	$3 \cdot 10^0$	=	$3 \cdot 1$	=	3
						47583

Fig. 15

Así un mismo número puede ser representado en cualquier sistema de numeración que se nos pueda ocurrir.

Por ejemplo, si decidimos contar en base 5 (con los dedos de una mano) escribiremos el número anterior como 3010313; veámoslo en la figura 16.

$$\begin{array}{rcl}
 3 \cdot 5^6 & = & 3 \cdot 15625 = 46875 \\
 0 \cdot 5^5 & = & 0 \cdot 3125 = 0 \\
 1 \cdot 5^4 & = & 1 \cdot 625 = 625 \\
 0 \cdot 5^3 & = & 0 \cdot 125 = 0 \\
 3 \cdot 5^2 & = & 3 \cdot 25 = 75 \\
 1 \cdot 5^1 & = & 1 \cdot 5 = 5 \\
 3 \cdot 5^0 & = & 3 \cdot 1 = 3 \\
 & & 47583
 \end{array}$$

Fig. 16

Hasta ahora sabemos convertir un número de cualquier base a decimal. Para hacer la conversión en sentido contrario vamos dividiendo el número por la base deseada hasta llegar a un cociente de cero; entonces todos los restos obtenidos leídos en orden inverso dan el número deseado. Así podemos convertir 47583 a base 5 como se observa en la figura 17.

$$\begin{array}{rcl}
 47583 : 5 & = & 9516 \quad 3 \\
 9516 : 5 & = & 1903 \quad 1 \\
 1903 : 5 & = & 380 \quad 3 \\
 380 : 5 & = & 76 \quad 0 \\
 76 : 5 & = & 15 \quad 1 \\
 15 : 5 & = & 3 \quad 0 \\
 3 : 5 & = & 0 \quad 3 \\
 & & \hline
 & & 3010313
 \end{array}$$

Fig. 17

3.1 Sistemas binario, octal y hexadecimal

Los ordenadores y los autómatas son equipos electrónicos y, como tales, sólo son capaces de almacenar dos valores que corresponden a los estados con corriente y sin corriente. Por este motivo estas máquinas trabajan internamente en forma binaria (base 2).

El código binario tiene sólo dos valores (0 y 1), lo que hace que un número escrito en binario tenga muchas más cifras que en decimal. Por ejemplo 47583 escrito en binario será 1011100111011111.

A la hora de trabajar con ordenadores y autómatas no es cómodo trabajar en binario y, por esto, los usuarios pueden trabajar, según los casos, en octal o hexadecimal.

En el sistema hexadecimal (base 16) disponemos de 16 cifras (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E i F). Demos cuenta de que para evitar confusiones en la escritura a las 6 cifras superiores a 9 se les asigna una letra. Como ejemplo el número 47583 se escribiría en hexadecimal como B9DF, comprobémoslo en la figura 18.

$$\begin{array}{rclclcl}
 B \cdot 16^3 & = & 11 \cdot 4096 & = & 45056 \\
 9 \cdot 16^2 & = & 9 \cdot 256 & = & 2304 \\
 D \cdot 16^1 & = & 13 \cdot 16 & = & 208 \\
 F \cdot 16^0 & = & 15 \cdot 1 & = & 15 \\
 & & & & 47583
 \end{array}$$

Fig. 18

En el sistema octal (base 8) trabajamos con dígitos entre 0 y 7. El número 134737 escrito en octal, valdrá 47583 en decimal como se ve en la figura 19.

$$\begin{array}{rclclcl}
 1 \cdot 8^5 & = & 1 \cdot 32768 & = & 32768 \\
 3 \cdot 8^4 & = & 3 \cdot 4096 & = & 12288 \\
 4 \cdot 8^3 & = & 4 \cdot 512 & = & 2048 \\
 7 \cdot 8^2 & = & 7 \cdot 64 & = & 448 \\
 3 \cdot 8^1 & = & 3 \cdot 8 & = & 24 \\
 7 \cdot 8^0 & = & 7 \cdot 1 & = & 7 \\
 & & & & 47583
 \end{array}$$

Fig. 19

3.2 Código BCD

A menudo un número tiene que ser representado en un *display* de salida. En este caso es habitual escribir este número en código BCD (decimal codificado en binario). Un número en código BCD consta de una sucesión de grupos de cuatro cifras binarias donde cada bloque es la cifra decimal correspondiente escrita en binario. Así el número 47583 se escribe en BCD como se puede ver en la figura 20.

0100	0111	0101	1000	0011
4	7	5	8	3

Fig. 20

3.3 Bits, bytes, nibbles, etc.

Los ordenadores y autómatas suelen tener la memoria dividida en bloques iguales. Cada celda de memoria puede contener una cifra binaria (pasa corriente o no pasa corriente); es habitual asignar el valor 1 cuando está activada (pasa corriente) y el 0 cuando no lo está (no pasa corriente). A una cifra binaria se la llama bit. La agrupación de 4 bits forma un *nibble* y la de 8 bits forma un *byte* (también llamado octeto). La agrupación de bits funcional de un microprocesador se llama palabra; hay sistemas con palabras de 8 bits, otras de 16 bits, etc.

Normalmente la capacidad de memoria de un equipo es una potencia de 2. Por ejemplo: 1024 *bytes*, 16384 *bytes*, 32768 *bytes*, 65536 *bytes*, etc. Para facilitar el contado de la capacidad de memoria (y también de otros datos) a menudo se cuenta en *kilobytes* o en *megabytes*. En este caso el prefijo kilo no indica mil sino la potencia de dos más próxima, o sea 1024. De la misma manera mega indica 1048576. Así una memoria de 64 kB tiene 65536 *bytes*.

3.4 Código ASCII

Muchas veces se desea representar un texto en un *display* de salida. En este caso es habitual escribir este texto en código ASCII. Un carácter ASCII viene representado por un número entre 0 y 127, por tanto, en un número de 7 bits. Los códigos ASCII son los de la tabla 5 donde los espacios sombreados corresponden a los caracteres de control.

Tabla 5

	0	1	2	3	4	5	6	7	8	9
0										
10										
20										
30				!	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	'	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}				

Dado que habitualmente se trabaja con sistemas hexadecimal u octal se suelen usar 8 bits para designar un carácter, de manera que el bit más significativo es siempre 0; de esta forma en un octeto cabe un carácter y en una palabra de 16 bits caben dos.

4 Evolución de los automatismos industriales

Tomemos el caso de la maniobra de un ascensor como ejemplo de automatismo. Las entradas son los pulsadores de las plantas y la cabina, los detectores de posición de la cabina, los detectores de puerta abierta, el relé térmico, etc. Los elementos de salida son los pilotos de planta actual, ocupado, dirección, etc., el timbre de alarma y el motor.

Para automatizar el proceso podemos usar cualquier sistema que nos permita actuar sobre las salidas. A lo largo de la historia ha habido diversas formas de implantación de los automatismos industriales (no todas ellas se han usado exhaustivamente en la automatización de ascensores).

4.1 La lógica cableada (o electromecánica)

La lógica cableada fue la primera que se usó y la de más gran difusión pues la mayoría de los automatismos se han hecho de esta manera.

Consiste en interconectar relés con los elementos de entrada y salida para que, a base de conexiones en serie y en paralelo de elementos, al final se obtenga el automatismo deseado. Se dispone de diversos elementos: relés de conmutación, contactores, relés de funciones lógicas, temporizadores, relés de control, etc.

Este tipo de automatismo tiene el inconveniente del gran volumen ocupado por el automatismo. Ciertamente, las dimensiones de un relé son importantes y las funciones que permite hacer son pocas (enclavamiento, negación). Esto hace que se intente reducir al máximo su número de manera que los esquemas de conexionado se vuelven difíciles de interpretar. Cuando los automatismos se complican los esquemas son realizados de forma intuitiva. La modificación de un automatismo pasa a menudo por desmontar una buena parte y cablearlo de nuevo. Además, a causa de la presencia de contactos móviles, necesitan un mantenimiento importante.

En caso de automatismos sencillos esta solución continúa teniendo ventajas ya que la lógica cableada es la única que no requiere forzosamente un cambio de los niveles de tensión entre el automatismo y los elementos a controlar.

En cualquier caso las representaciones a base de relés todavía son la forma de representar un automatismo más familiar del personal de mantenimiento, lo cual hace que los autómatas programables usen un lenguaje de programación a base de diagramas de relés.

4.2 La lógica neumática

Se basa en usar aire comprimido y diversos tipos de elementos: válvulas distribuidoras, detectores, pulsadores y pilotos neumáticos, válvulas biestables, cilindros neumáticos, válvulas de funciones lógicas, etc.

Tiene la ventaja de que no se ve afectada por las interferencias electromagnéticas pero necesita mucho espacio y crea un ruido importante; además, es necesario un compresor. La distribución del aire comprimido es más compleja que la de la energía eléctrica a causa del diámetro de los tubos y del radio mínimo de curvatura. Necesita un mantenimiento importante.

La lógica neumática resulta interesante cuando se trata de automatismos sencillos que actúan sobre accionamientos neumáticos.

En caso de que se requiera una potencia o precisión mayor puede usarse la oleohidráulica.

4.3 La lógica estática discreta

Poco después de la aparición de los transistores se empezaron a usar circuitos electrónicos para realizar las funciones. Los diseñadores montaban circuitos con resistencias, transistores y diodos a fin de controlar los automatismos. Este método tenía el inconveniente de que era preciso un cambio de niveles de tensión entre la potencia y la lógica pero en circuitos complejos significaba una importante reducción de volumen; además, el uso de componentes estáticos disminuía los problemas ya que no había ningún contacto móvil que se pudiese desgastar; esto les permitía también una mayor velocidad de respuesta.

Las puertas lógicas aparecieron para simplificar el montaje ya que se podían conectar directamente las entradas de una a la salida de otra sin tomar ningún tipo de precaución. Inicialmente no eran más que cajas de plástico con un cierto número de patas que contenían un circuito impreso con resistencias, diodos y transistores.

4.4 La lógica estática integrada

Con la aparición de los circuitos integrados las antiguas puertas con circuitos compactos se substituyeron por circuitos integrados con una mayor reducción de volumen. Con el tiempo la mayor parte de la lógica se redujo a dos familias: la TTL (a 5 volts) y la CMOS (habitualmente a 12 volts).

Se pueden encontrar circuitos integrados con las principales funciones: puertas lógicas, biestables, temporizadores, contadores, selectores, decodificadores, etc. Con la ventaja de que el volumen ocupado por el automatismo es menor que en todos los casos anteriores.

Presenta dos inconvenientes importantes, por un lado la necesidad de un cambio de niveles de tensión y por otro la imposibilidad de modificación; si es necesario hacer un cambio en el automatismo hay que tirar el circuito impreso y hacer uno nuevo.

4.5 La lógica estática programada

La lógica estática permite compactificar mucho los circuitos pero tiene el inconveniente, como ya hemos visto, de la gran dificultad de modificación.

Para solucionar los problemas de la lógica estática (y de la lógica cableada) se usan los sistemas basados en microprocesador que permiten una mayor reducción del circuito electrónico y que sea programable; de esta forma la modificación de las relaciones lógicas es relativamente sencilla.

Continúa presentando el inconveniente de la dificultad de modificación dado que añadir una entrada o una salida adicional implicará confeccionar un nuevo circuito impreso.

4.6 El ordenador de proceso

Como mejora de los sistemas basados en microprocesador aparece el ordenador de proceso, parecido al ordenador de gestión pero preparado para funcionar en ambiente industrial y equipado con entradas y salidas. Presenta la ventaja adicional de estar capacitado para realizar cálculos complejos.

Tiene un inconveniente importante derivado del hecho de necesitar personal informático pero con conocimientos de automatización industrial y del proceso que se quiere automatizar para su programación.

4.7 El autómatas programable industrial

Ante esta problemática aparecieron los autómatas programables (o PLC, *Programmable Logic Controller*, controlador programable por lógica). Inicialmente se concibieron como circuitos electrónicos basados en un microprocesador que tenían que funcionar como una lógica estática pero de manera que las funciones a realizar fuesen programadas y, por tanto, fácilmente modificables.

A fin de que la programación y el mantenimiento fuesen posibles sin una formación informática del personal, la inmensa mayoría de los equipos permitían una programación a base de reproducir un diagrama de relés.

5 Definición y arquitectura del autómeta programable industrial

5.1 Definición histórica

Un autómeta programable industrial es una máquina electrónica, históricamente programable por personal no informático, preparada para realizar, en ambiente industrial, automatismos combinatorios y secuenciales en tiempo real.

Se trata, pues, de un ordenador pensado, tanto en los aspectos de *software* como de *hardware*, para la automatización industrial.

Actualmente, dada la evolución de estos equipos, a pesar de que pueden ser programados por personal no informático es conveniente que el personal que tiene que hacerlo tenga unos ciertos conocimientos de informática.

5.2 Arquitectura

Un autómeta programable consta (figura 21) de tres partes fundamentales: Unidad de memoria, unidad de control y elementos de entrada y salida.

5.2.1 Unidad de memoria

La memoria de un autómeta programable sirve para almacenar el programa y los datos del proceso. En muchos autómetas el usuario puede trabajar con la configuración de memoria base o puede añadir más memoria (hasta un cierto límite) en función de sus necesidades.

Dentro de la memoria de datos tenemos una parte fija que es la tabla de imágenes de entradas y salidas; cuya medida viene ya definida mientras que el resto de la memoria de datos puede ser variable en función de las necesidades de cada programa.

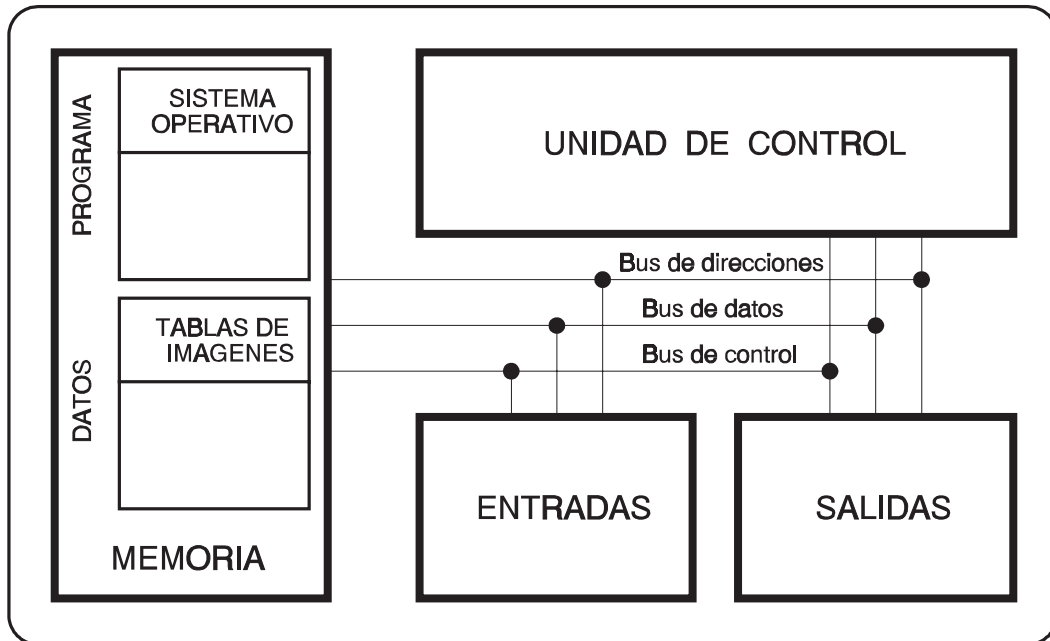


Fig. 21

Los autómatas pueden tener una asignación de memoria fija es decir que la parte destinada a programa y la parte destinada a datos (incluso el número de temporizadores, contadores, etc.) viene fijada de fábrica o con asignación dinámica de la memoria de forma que a medida que se va haciendo el programa se asigna a cada necesidad la parte requerida.

La memoria de programa tiene una parte fija: el sistema operativo. Esta parte viene programada de fábrica y es la que se encarga de la lectura de entradas/salidas, efectuar el *scan* de programa, gestionar los posibles errores de funcionamiento, etc.

Tipos de memorias

Hay dos tipos básicos de memoria: volátiles y no volátiles. Las memorias volátiles se pueden leer, escribir y borrar fácilmente por el propio programa. Tienen el inconveniente de que pierden la información grabada cuando se desconecta la alimentación. Se llaman memorias RAM *Random Access Memory* (Memoria de acceso aleatorio).

En muchos casos se pone una pequeña batería o un condensador que alimenta exclusivamente la memoria RAM cuando falla la alimentación. Según los fabricantes, el programa y los datos pueden mantenerse horas, días o años.

Las memorias no volátiles pueden ser leídas a voluntad pero difieren en la forma en que se pueden escribir. Todas ellas mantienen la información aunque se pierda la alimentación eléctrica. A continuación hay una clasificación de algunos tipos.

ROM *Read Only Memory* (Memoria de sólo lectura). Es una memoria que se programa en el momento de fabricación y que puede ser leída pero no escrita ni borrada.

PROM *Programmable ROM* (ROM programable). Es una memoria que se puede programar eléctricamente en cualquier momento pero no se puede borrar (sólo se puede programar una vez).

EPROM *Erasable PROM* (PROM borrable). Se puede programar eléctricamente y borrar (todo el contenido de golpe) exponiéndola a una luz UV un número muy elevado de veces.

EEPROM *Electrically Erasable PROM* (PROM borrable eléctricamente). Se programa y se borra (todo el contenido de una vez) eléctricamente.

EAROM *Electrically Alterable ROM* (ROM alterable eléctricamente). Se comporta como una memoria RAM pero los datos no se pierden al cortar la alimentación. Es la única memoria no volátil en que se puede modificar una parte del contenido sin borrarla completamente.

5.2.2 Unidad de control

La unidad de control, también llamada CPU *Central Processing Unit* (unidad central de proceso) es la parte inteligente del autómata. Su función es ejecutar las instrucciones del programa. También se encarga de las comunicaciones con los equipos de programación y de la gestión de los estados de error.

Su elemento base es el microprocesador. La capacidad de cálculo y la velocidad de procesamiento dependen del número y tipo de procesadores que tenga. La mayor parte de los autómatas tienen una CPU con un solo procesador pero cada vez hay más que tienen las funciones descentralizadas entre diversos procesadores a menudo diferentes.

La ejecución del programa sigue un ciclo llamado *scan* que consiste en:

1. Lee las entradas y guarda sus estados en la tabla de imágenes de entrada.
2. Hace una ejecución del programa cogiendo los datos necesarios de la tabla de entradas, los contadores, los temporizadores, etc. y dejando lo que convenga en la tabla de salidas, contadores, etc.

3. Copia la tabla de imágenes de salida sobre las salidas.
4. Vuelve a empezar el ciclo leyendo las entradas.

El uso de las tablas de entradas y salidas es muy importante. Si las entradas se leyeran cada vez que se necesitan y las salidas se escribieran también cada vez, resultaría que una misma entrada que se usa más de una vez dentro de un *scan* de programa tendría estados diferentes con lo que el automatismo no funciona correctamente.

El programa se va repitiendo en forma cíclica ya que las modificaciones que vayan apareciendo en las entradas tienen que ir modificando los estados de los relés internos y las salidas.

5.2.3 Elementos de entrada y salida

Los elementos de entrada y salida son los que permiten comunicar el autómatas con el proceso que está controlando y con el operador. Mediante los elementos de entrada el autómatas se entera del estado en que se encuentra el proceso (posiciones, velocidades, niveles, temperaturas, elementos activados, elementos desactivados, etc.) a partir de los captadores que el diseñador ha situado para las señales que interesan. Los elementos de salida permiten que el PLC actúe sobre el proceso (electroválvulas, motores, pilotos, etc.).

Elementos de entrada

Las entradas digitales son del tipo sí/no, es decir reciben señales de tipo cualitativo pero no cuantitativo. Por ejemplo un termostato da una señal digital (temperatura mayor o menor que 22°C). Las entradas digitales se corresponden con un bit en la tabla de imágenes de entrada.

Las entradas de un autómatas se clasifican según el tipo y la polaridad de conexión. Pueden ser de corriente alterna, de corriente continua con común positivo (tipo *sink*) y de corriente continua con común negativo (tipo *source*).

Los principales elementos que se conectan en este tipo de entrada son:

- a) Final de carrera. Es un elemento que detecta un objeto por contacto físico con el mismo.
- b) Detectores de proximidad magnéticos. Se coloca un relé Reed cerca del lugar donde tiene que pasar la pieza a detectar, a la que se le ha adosado un imán permanente. Cuando el imán pase cerca del relé provocará el accionamiento del mismo.

- c) Detectores de proximidad inductivos. Constan de un circuito electrónico oscilante basado en una inductancia que deja de oscilar cuando ésta se encuentra en las proximidades de una pieza metálica.
- d) Detectores de proximidad capacitivos. Constan de un circuito electrónico oscilante basado en un condensador que no oscila en condiciones normales pero sí cuando un objeto no metálico pasa cerca suyo. A pesar de la descripción, si se ajustan convenientemente, también pueden detectar objetos metálicos.
- e) Detectores de proximidad fotoeléctricos de reflexión. Hay un elemento que emite un haz de luz y lo tiene que volver a recibir. El objeto a detectar tiene que reflejar el haz sobre el detector.
- f) Detectores de proximidad fotoeléctricos de interposición. Hay un elemento que emite un haz de luz y otro que lo tiene que recibir. El objeto a detectar tiene que interponerse entre el emisor y el receptor. Se llaman *reflex* aquellos en que el emisor y el detector están juntos y el haz de luz se refleja en un reflector catadióptrico; en este caso si hay posibilidades de que el objeto a detectar refleje el rayo de luz es necesario usar haces polarizados dado que los reflectores catadióptricos giran el plano de la luz.

Conviene distinguir entre aquellos captadores o detectores que necesitan una alimentación exterior para funcionar (detectores de tres hilos) y los que no necesitan (detectores de dos hilos). Entre los detectores de dos hilos distinguimos aquellos que se basan en un sencillo contacto eléctrico y los que llevan circuitería electrónica.

Entre los detectores electrónicos de dos hilos podemos distinguir entre los no polarizados (figura 22) que, con el mismo principio, pueden funcionar en corriente continua o en corriente alterna (y algunos indistintamente en corriente continua y corriente alterna) y los polarizados (figura 23) que sólo pueden funcionar en corriente continua y vigilando la polaridad de conexión.

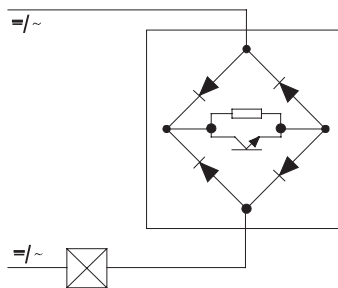


Fig. 22

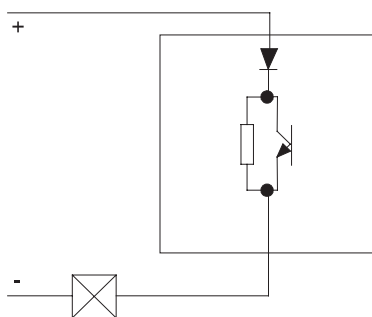


Fig. 23

Los captadores de dos hilos tienen la ventaja de que un mismo tipo de captador se puede usar con autómatas diferentes. Es necesario vigilar que la corriente residual (en estado abierto) que necesitan para funcionar no pueda activar la entrada del autómata.

Los captadores de tres hilos no tienen corrientes residuales pero necesitan alimentación externa. Hay dos tipos según la lógica del autómata donde se tienen que conectar. Los captadores de tipo NPN (figura 24) se usan en autómatas con el negativo como común de entrada (tipo *source*) y los de tipo PNP (figura 25) se usan en autómatas con el positivo como común de entrada (tipo *sink*).

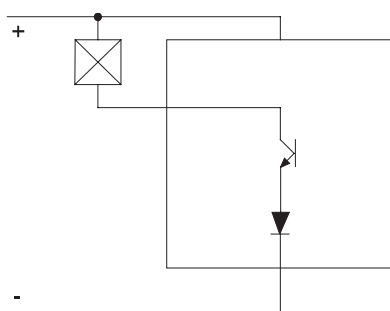


Fig. 24

También hay posibilidades de entradas digitales codificadas como pueden ser teclados, preseleccionadores digitales, *encoders*, etc. Estos elementos convierten señales de tipo numérico en un conjunto de bits en codificación hexadecimal, BCD, etc.

Las entradas analógicas son capaces de aceptar señales de tensión o corriente entre dos valores (0..10 V, -10..10 V, 4..20 mA, 0..20 mA) y asociarles un número de 8, 10, etc. bits que indique el valor leído. Se usan para la medida de temperaturas, presiones, caudales, tensiones, corrientes, pares, etc.

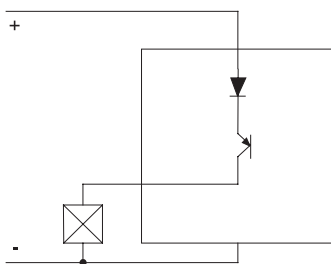


Fig. 25

Conviene usar hilo trenzado y apantallado para las entradas analógicas y corto circuitar aquellas entradas por tensión que no se tengan que usar. Habitualmente la pantalla suele conectarse a tierra en el lado del autómata (figura 26) y al negativo en el extremo donde está el captador. En algunos casos se conecta a tierra y al negativo en el extremo correspondiente al captador (figura 27) pero es necesario vigilar que la tensión que puede haber entre los dos negativos que entran en el autómata no supere el límite admisible (habitualmente de pocos volts).

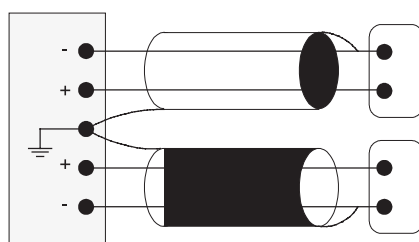


Fig. 26

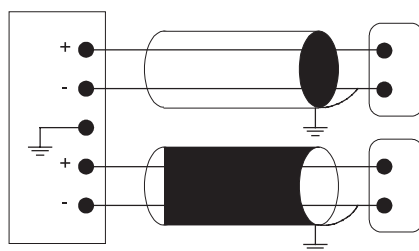


Fig. 27

Elementos de salida

Las salidas digitales también son de tipo sí/no. En el caso de salidas en corriente continua pueden ser por transistor o por relé mientras que si la salida es en corriente alterna puede ser por triac o por relé.

En las salidas por transistor, éste entra en conducción cuando la salida se activa. Son apropiadas para accionar elementos de pequeña potencia en corriente continua (electroválvulas, contactores, pilotos, etc.). Pueden ser con el negativo común (tipo *source*) o con el positivo común (tipo *sink*).

En las salidas por triac, éste entra en conducción cuando la salida se activa. Cuando la salida se desactiva el triac continuará conduciendo hasta el primer paso por cero de la corriente. Por este motivo las salidas por triac no funcionan en corriente continua.

Las salidas por relé son las más versátiles ya que el usuario dispone de un contacto de relé libre de tensión (aislado de cualquier otro circuito) de manera que cada salida puede accionar elementos diferentes a tensiones diferentes. Tienen el inconveniente de que la velocidad de respuesta es pequeña, por tanto, no pueden usarse en aplicaciones que requieren cambios rápidos en las salidas.

También hay elementos de salida codificada como *displays* de 7 segmentos, *displays* alfanuméricos y *displays* de mensajes.

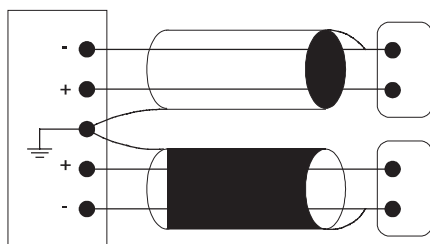


Fig. 28

Las salidas analógicas pueden dar señales de tensión o de corriente variables (0..10 V, -10..10 V, 4..20 mA, 0..20 mA) que permiten accionar válvulas proporcionales, dar consignas a variadores de velocidad para motores, etc. Conviene usar hilo trenzado y apantallado para las salidas analógicas. La pantalla suele conectarse a tierra en el lado del autómeta (figura 28) y al negativo en el extremo donde está el preaccionador dado que el negativo de las diferentes salidas del mismo autómeta suele ser común.

5.2.4 Buses de comunicación

Son el medio físico a través del cual el procesador se comunica con el resto de elementos del sistema (entradas y salidas, memoria, periféricos). Hay normalmente tres buses: Direcciones, datos y control. Cada uno de ellos está formado por un conjunto de cables, o mejor dicho un conjunto de pistas de circuito impreso. Cada uno de los elementos conectados al bus tiene una dirección.

El bus de direcciones es por donde el procesador envía la dirección del elemento al que quiere enviar o que quiere que le envíe información. Esta dirección llegará a todos los elementos pero sólo tiene que haber un elemento que se identifique.

El bus de datos es por donde todos los elementos enviarán los datos. En una escritura, el procesador pondrá los datos que quiere que lea el elemento señalado con el bus de direcciones. En el caso de una lectura, el procesador leerá los datos que haya puesto el elemento señalado. El bus de datos es, por tanto, bidireccional.

El bus de control es aquel mediante el cual el procesador explica qué operación se está efectuando. Las operaciones más corrientes son leer y escribir.

5.3 Sistema operativo

El sistema operativo se encarga de ejecutar las funciones del autómata, tanto si son en tiempo real como si no. En programas sencillos se ejecutan todas las funciones dentro de un solo ciclo. En programas más complejos nos podemos encontrar que el tiempo de ejecución sea inaceptable. En estos casos a menudo se hace un fraccionamiento del programa en módulos (subrutinas) de manera que no todos los módulos se ejecutan en todos los ciclos.

Se encuentran también casos en los que se ejecuta una parte de cada módulo en cada ciclo de programa. Este método tiene serios problemas de interpretación de programas y depuración y corrección de errores de los mismos. Algunos autómatas incorporan ya subrutinas de interrupción por tiempo que permiten ejecutar algunas partes de programa cada un cierto tiempo.

6 Configuración y programación

6.1 Configuración del mapa de memoria

Configurar el mapa de memoria quiere decir indicar, antes de empezar la programación, qué parte de la memoria se reserva para cada cosa. Es necesario definir las direcciones donde se encontrarán la tabla de imágenes de entrada, la tabla de imágenes de salida, los relés internos, los registros de control, las variables enteras, las variables reales (coma flotante), los datos de contadores, los datos de temporizadores, etc. No todos los autómatas tienen todos estos tipos de registros, otros tienen más.

Los relés internos son direcciones binarias que se usan como relés que no afectan a las salidas para realizar automatismos.

En algunos autómatas el mapa de memoria es fijo. En otros es necesario configurarlo antes de programar o se autoconfigura automáticamente a medida que se va programando.

6.2 Lenguajes y sistemas de programación

El técnico que prepara un automatismo debe diseñar primero la lógica que éste tiene que seguir y, una vez acabado el diseño, tiene que explicarlo al autómatas a fin de que éste pueda entenderlo y llevarlo a la práctica.

El método que use para crear el automatismo no tiene ninguna importancia mientras después sea capaz de traducirlo a alguna forma inteligible para el autómatas.

Los métodos de entrar el automatismo al autómatas (lenguajes de programación) son diversos. A continuación describimos los más corrientes. En todos los casos pondremos como ejemplo el mismo automatismo (que en lógica cableada podríamos definir con el circuito de la figura 29) que corresponde al control de una puerta automatizada para un edificio. En este ejemplo K1 será el contactor que hace abrir la puerta y K2 el que la hace cerrar. P será el detector situado debajo la alfombra y FdCO y FdCT son respectivamente los finales de carrera de abrir y de cerrar.

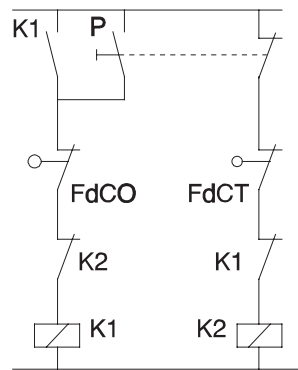


Fig. 29

6.2.1 Diagrama de contactos

También conocido como diagrama de relés, es la forma más corriente de programar un autómata. Se trata de hacer un esquema como si se tuviese que hacer un automatismo con relés y esto se entra gráficamente en el *software* del autómata. Este lenguaje tiene la ventaja de que los técnicos de mantenimiento están acostumbrados a dibujar circuitos lógicos con relés. En la mayor parte de los autómatas el circuito se dibuja según el método americano en que los símbolos son diferentes y las líneas lógicas van horizontales con las salidas a la derecha, tal como se ve en la figura 30.

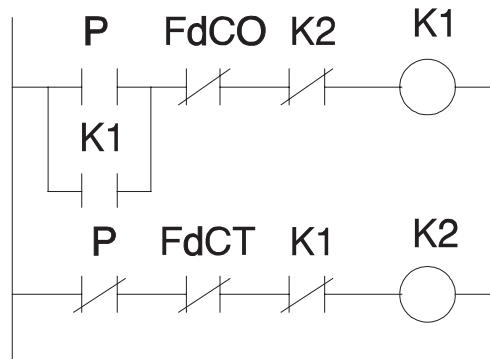


Fig. 30

6.2.2. Puertas lógicas

Consiste en hacer un esquema como si se tuviese que hacer un automatismo electrónico. Este método tiene la ventaja de ser sencillo para aquellos que han trabajado antes con puertas lógicas.

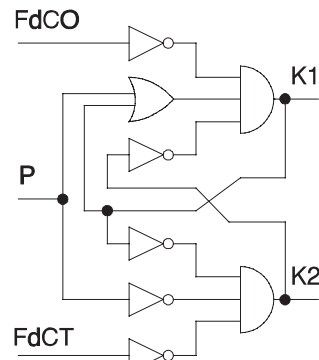


Fig. 31

6.2.3 Diagrama funcional

Consiste en hacer un circuito similar al de las puertas lógicas pero con bloques funcionales. Los símbolos que se usan habitualmente en los bloques funcionales son:

&	Función y (<i>AND</i>)
≥ 1	Función o (<i>OR</i>)
=1	Función o-exclusiva (<i>EXOR</i>)
=	Función igual

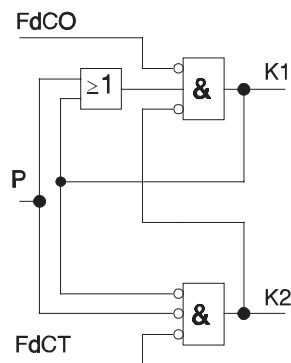


Fig. 32

6.2.4 Diagrama de flujo

Es un método parecido a los árboles de decisión que se usan también en algorítmica.

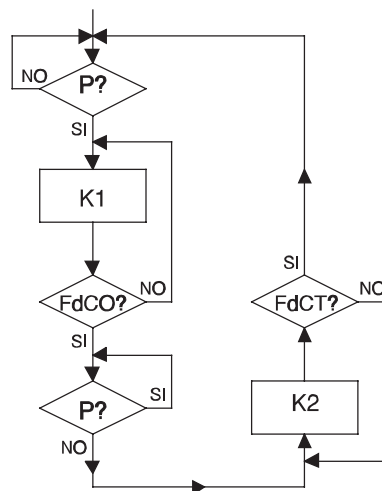


Fig. 33

Consta de cajas en forma de rombo y de rectángulo. Los rombos son preguntas con respuesta sí o no y los rectángulos son acciones.

6.2.5. GRAFCET

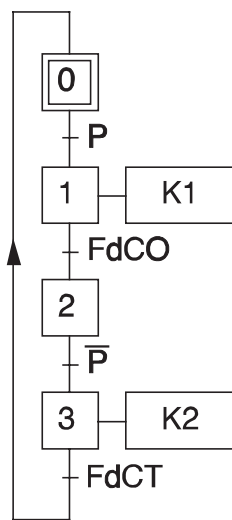


Fig. 34

El GRAFCET (*Grphe de commande etape-transition*, grafo de pedido con etapas y transiciones) es un método similar a los árboles de decisión en que puede haber etapas simultáneas (en paralelo). Antes de cada etapa hay una transición (figura 34).

En el GRAFCET las etapas se representan con cuadrados, que son en línea doble si se trata de etapas iniciales. Las acciones a realizar en cada etapa se representan con rectángulos que salen lateralmente de las etapas.

Las líneas simples son los caminos según los cuales evoluciona el automatismo y las dobles indican que los caminos se bifurcan para dar lugar a etapas en paralelo (simultáneas).

Una rayita horizontal que cruza la línea simple representa una transición. No se puede franquear la transición hasta que no se ha cumplido la condición que se especifica en él.

6.2.6. Lenguaje booleano

Consiste en escribir directamente las ecuaciones booleanas que representan el automatismo. En el ejemplo que venimos representando sería el de la figura 35.

$$K1 = (P + K1) \cdot \overline{FdCO} \cdot \overline{K2}$$

$$K2 = \overline{P} \cdot \overline{FdCT} \cdot \overline{K1}$$

Fig. 35

6.2.7. Lista de instrucciones

Se trata de describir las ecuaciones booleanas con una lista de instrucciones de un solo operando. Cada autómeta tiene su forma particular de hacerlo. Una podría ser la de la figura 36.

LOAD	K1
OR	P
AND NOT	FdCO
OUT	K1
LOAD NOT	P
AND NOT	FdCT
AND NOT	K1
OUT	K2

Fig. 36

6.2.8 Lenguajes de alto nivel

Hay autómatas que admiten programación con lenguajes como BASIC o PASCAL con la ventaja de que los lenguajes son muy parecidos a los que se usan en la programación de ordenadores. En algunos autómatas el lenguaje es interpretado (no compilado) lo cual da lugar a tiempos lentos de ejecución. Un ejemplo en lenguaje BASIC podría ser el de la figura 37.

```
10 IF ((P OR K1) AND (NOT FdCO) AND (NOT K2)) THEN SET K1 ELSE RES K1
20 IF ((NOT P) AND (NOT FdCT) AND (NOT K1)) THEN SET K2 ELSE RES K2
```

Fig. 37