

Controladores programables para accionamientos

Este trabajo hace parte de la línea de investigación "Adaptación y creación de tecnología en el área del control digital", del Departamento de Ingeniería Eléctrica.

WOLF KERPEL S.
Profesor Departamento de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad Nacional

Antes de la década de los 60 los sistemas de control y accionamientos se venían realizando con elementos electromecánicos (interruptores, pulsadores, relés auxiliares, contactores, etc.) y algunos dispositivos electrónicos especiales (sensores, amplificadores, contadores, etc.).

En la década del 60, firmas importantes en la construcción de accionamientos (Siemens, AEG, BBC, Klockner Moeller y otras) comenzaron a producir sistemas electrónicos completos con base en transistores que permitían reemplazar la lógica convencional de los contactos por el concepto nuevo de las compuertas y memorias electrónicas digitales. Estos sistemas resultaron efímeros, ya que la invención del microprocesador permitió la producción de sistemas de control programables de bajo costo y gran flexibilidad; conduciendo así a un nuevo concepto: la lógica programada.

Refiriéndonos a los controladores programables para accionamientos, básicamente se han desarrollado dos tipos: los de propósito específico y los de propósito general. Los primeros son diseñados específicamente para controlar un equipo particular como por ejemplo Máquinas-Herramientas de Control numérico, telares computarizados, controladores de riego, etc. Los segundos, de propósito general se diseñaron específicamente para reemplazar los sistemas de accionamientos electromecánicos.

EL CONTROLADOR PROGRAMABLE PARA ACCIONAMIENTOS (CPA)

El CPA es un sistema que ejecuta acciones de control lógico (on-off), temporización, conteo y secuenciación. Las acciones que realiza quedan determinadas por el programa (o programas) almacenado en memoria.

Prácticamente todos los grandes fabricantes de equipo de control han desarrollado sus propias versiones. Estos dispositivos, normalmente pueden trabajar como un sistema aislado, o como parte integral de otros sistemas y redes (procesamiento distribuido). Los lenguajes de pro-

TABLA 1
Controladores programables para accionamientos

Sistema	Pequeño	Mediano	Grande
Control lógico	X	X	X
Secuenciación	X	X	X
Temporización	X	X	X
Conteo	X	X	X
E/s. digitales	X	X	X
E/s. análogas		X	X
Manejo de datos		X	X
Máximo número de E/s.	50	250	1000
Comunicación hombre-máquina			
indicadores de led, alarmas	X	X	X
registro de eventos			
reportes			X
Lenguaje de programación:			
instrucciones	X	X	
elementos funcionales		X	
Comunicación con otros sistemas		X	X
Bits/microprocesador	8	8/16	16

gramación disponibles son de gran simplicidad y poder; específicamente orientados para la descripción de sistemas de accionamientos. Se encuentran lenguajes basados en instrucciones y lenguajes gráficos basados en elementos funcionales.

En la tabla 1 se hace una clasificación de los **CPAS** que pretende ser ilustrativa, más no exhaustiva ni precisa.

EL LENGUAJE

Como se mencionó, se encuentran dos tipos básicos de lenguajes:

- de instrucciones
- de elementos funcionales (gráfico)

En el primer caso se cuenta con un número de instrucciones normalmente pequeño, pero suficiente para la descripción del sistema de control. Para la programación se requiere un teclado de unas 20 teclas (incluyendo las numéricas) y un anunciador de elementos de 7 segmentos.

En el segundo caso se trabaja con un conjunto de elementos funcionales. Mediante el uso de una pantalla y un teclado estandar con teclas de funciones. El sistema de control se describe esquematizándolo mediante los elementos funcionales. La gran mayoría de los fabricantes basa su lenguaje gráfico en los llamados diagramas de escalera, o sea que para programar el **CPA**, se dibuja en pantalla el diagrama de escalera del sistema de control. Este lenguaje gráfico es lógicamente el más adecuado, sin embargo, debido a su costo, no es justificable si no se trabaja con sistemas que por su complejidad lo ameriten. Vale la pena recalcar que los lenguajes de instrucciones son muy sencillos, y que al igual que los lenguajes gráficos, también fueron

diseñados específicamente para describir sistemas de control para accionamientos.

Funciones de control

Las siguientes son las funciones básicas con que cuentan los **CPAS** (se referirán con base en la lógica de contactos):

1. Funciones de contacto: **NA, NC**.
 repetición ilimitada de las funciones de contacto
2. Funciones lógicas: (serie, paralelo) **Y O**
 Negación
3. Funciones de temporización
4. Funciones de conteo de eventos
5. Funciones de registro de desplazamiento para secuenciación
7. Memorias auxiliares (reemplazan a los relés auxiliares).

Ventajas

Entre las ventajas que ofrece un **CPA** sobre un sistema convencional electromecánico se tienen:

1. Mayor confiabilidad
2. Mayor facilidad de operación
3. Consumo de potencia reducido
4. Espacio requerido reducido
5. Facilidad de modificación
6. Facilidad de monitoreo, prueba, y detección de fallas

Un caso hipotético

A continuación se describe un **CPA** hipotético (basado en los reales), junto con su lenguaje de Instrucciones y con ejemplos para su programación.

El **CPA** hipotético cuenta con:

30 Entradas:	E1-E30
20 Salidas:	S1-S20
10 Temporizadores:	T1-T10

10 Contadores: C1-C10
 64 Memorias de trabajo M1-M64

partir de un valor prefijado de dos dígitos que puede variar-entre:

1 - 99

8 Registros de desplazamiento, así agrupables en
 M1 -M8
 M9 -M16
 M17-M24
 M25-M32
 M33-M40
 M41-M49
 M49-M56
 M57-M64

Las memorias de trabajo realizan las funciones de los relés auxiliares y se pueden agrupar en conjuntos de a 8 para formar registros de desplazamiento con el fin de secuenciar eventos. Cualquier "contacto" se puede repetir un número ilimitado de veces.

Las entradas y salidas son digitales de dos niveles (on-off). Los temporizadores permiten obtener tiempos de retardo especificados con dos dígitos:

0.1 - 0.9 seg.
 1 - 99 seg.

Se pueden tener programas de hasta 200 instrucciones.

El funcionamiento del CPA es como sigue:

1. Lee las entradas
2. Corre el programa de control
3. Actualiza las salidas
4. Volver a 1.

O sea, que pueden programarse retardos entre 0.1 y 99 segundos. Los contadores sirven para contar eventos. Son descendentes. El conteo comienza a

El lenguaje permite describir fácilmente las funciones que debe realizar el CPA. Cuenta con 14 instrucciones que se describen en la tabla 2.

TABLA 2
Instrucciones del lenguaje de control

Instrucción	Aplicación	Símbolo en diagrama de contactos
CAR	CARGUE Comienzo de rama con un contacto NA	
CARN	CARGUE NEGADO Comienzo de rama con un contacto NC	
SALE	FIN de una rama. Señal de salida.	
Y	Conexión en serie con un contacto NA	
O	Conexión en paralelo con un contacto NA	
YN	Y NEGADO Conexión en serie con un contacto NC	
ON	O NEGADO Conexión en paralelo con un contacto NC	
YB	Y BLOQUE Conexión en serie de dos ramas	
OB	O BLOQUE Conexión en paralelo de dos ramas	
PLSO	Pulso a una memoria de trabajo (dura un ciclo de corrida)	
RPS	Reposición de un contador a su valor Prefijado o de un registro de desplazamiento a ceros.	
DESP	Pulso (orden) de desplazamiento a un registro	
NOP	No opere, avance a la siguiente instrucción.	
FIN	Fin del programa	

PROGRAMACION

El punto de partida para programar el CPA es una descripción del sistema de control la cual puede efectuarse mediante un diagrama en escalera (o diagrama de contactos), un diagrama lógico (compuertas) o simplemente fórmulas de Algebra Booleana. Una vez que se tiene la descripción del

sistema de control se procede a expresarla en función del lenguaje del CPA; luego de lo cual se puede proceder a cargar el programa en la memoria del dispositivo.

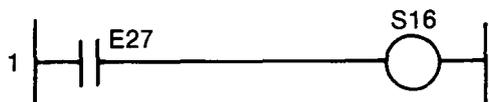
A continuación se explicarán las instrucciones del CPA por medio de ejemplos:

Instrucciones básicas: CAR, CARN, Y, O, YN, ON.

Ejemplos:

Diagrama de escalera

Programa



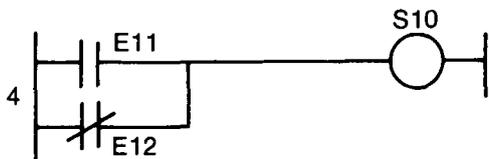
CAR	E27
SALE	S16



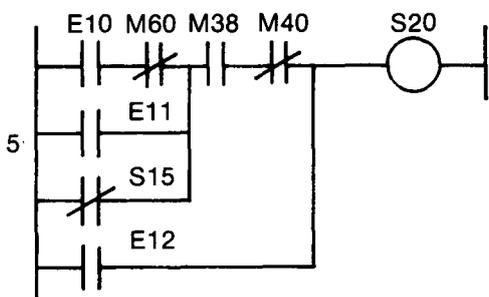
CARN	E2
SALE	S3



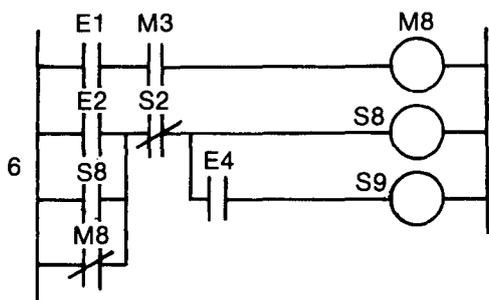
CAR	E5
Y	M4
SALE	M8



CAR	E11
ON	E12
SALE	S10



CAR	E10
YN	M60
O	E11
ON	S15
Y	M38
YN	M40
O	E12
SALE	S20



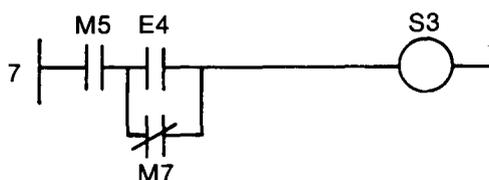
CAR	E1
Y	M3
SALE	M8
CAR	E2
O	S8
ON	M8
YN	S2
SALE	S8
Y	E4
SALE	S9

Instrucciones de bloque: YB, OB

Ejemplos:

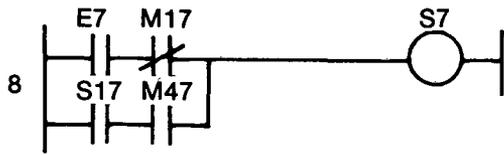
Diagrama de escalera

Programa

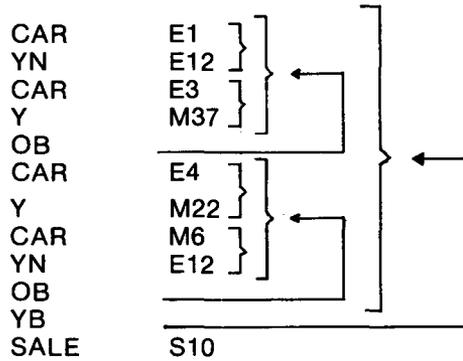
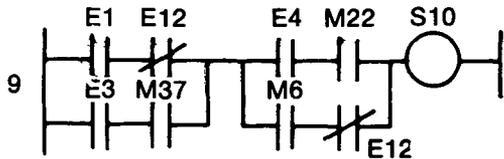
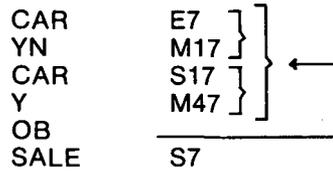


CAR	M
CAR	E4
ON	M7
YB	}
SALE	

DIAGRAMA DE ESCALERA



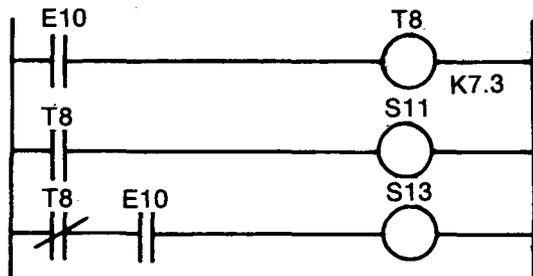
Programa



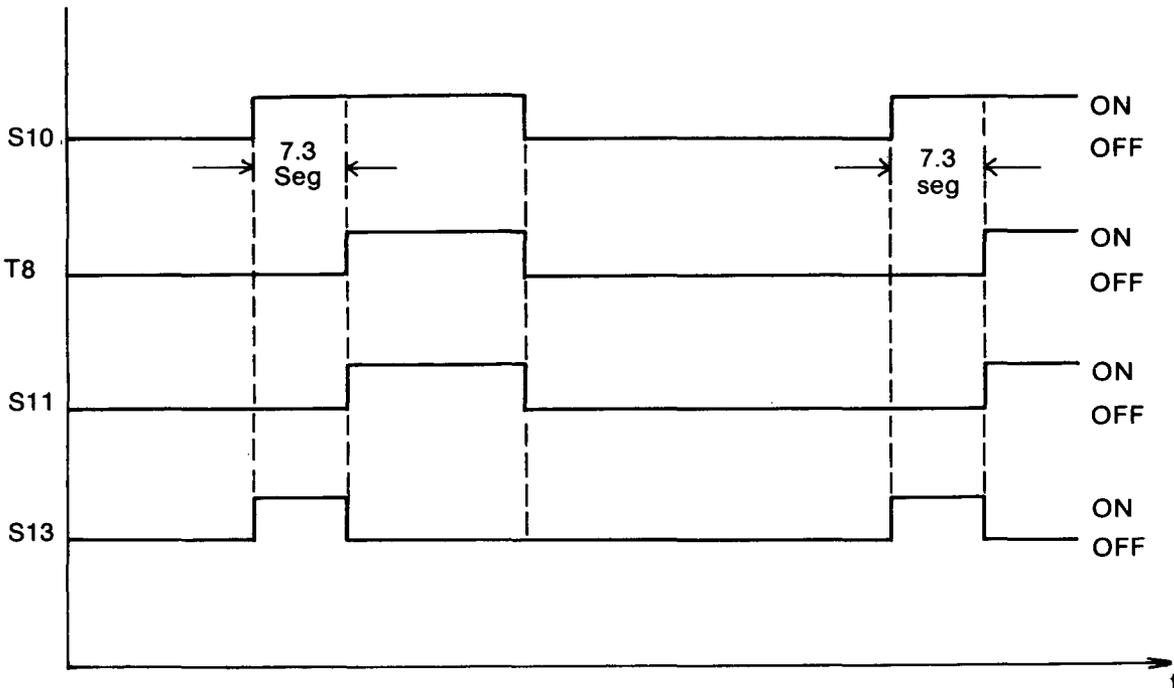
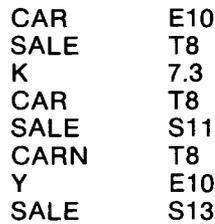
Temporizadores

Ejemplos

Diagrama de escalera



Programa



La constante de tiempo del temporizador K define el retardo de operación. La constante de tiempo se especifica con dos dígitos y opcionalmente el punto decimal.

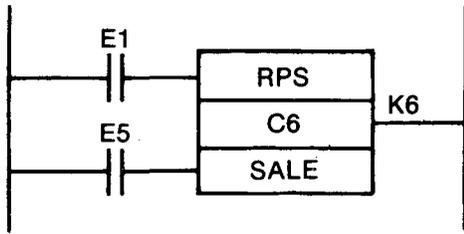
Contadores

Un contador queda definido por su dirección y constante de inicio de cuenta K. Todo contador tiene una entrada de reposición RPS que cuando

está activa retorna la cuenta al valor **K** prefijado y una entrada **SALE** que cada vez que está activa hace que el contador decremente su cuenta en una unidad.

Ejemplo:

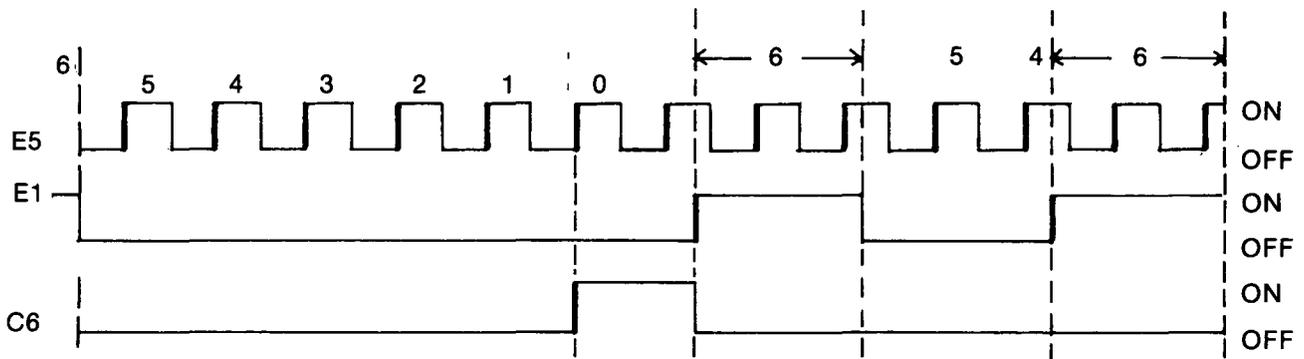
Diagrama de escalera



Programa

CAR	E1
RPS	C6
K	6
CAR	E5
SALE	C6

Después de que **E5** haya sido activada 6 veces la salida del contador es activada. Cuando se activa **E1** el contador devuelve su cuenta a 6.

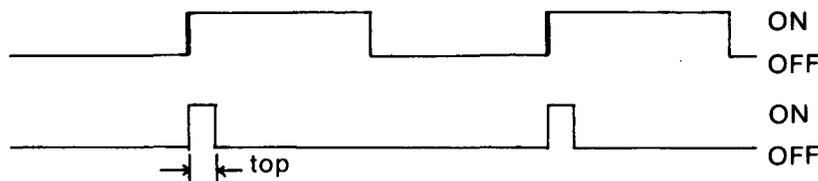
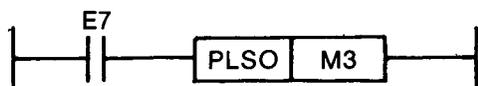


La instrucción **RPS** tiene mayor prioridad que la instrucción **SALE**.

Impulso a una memoria de trabajo

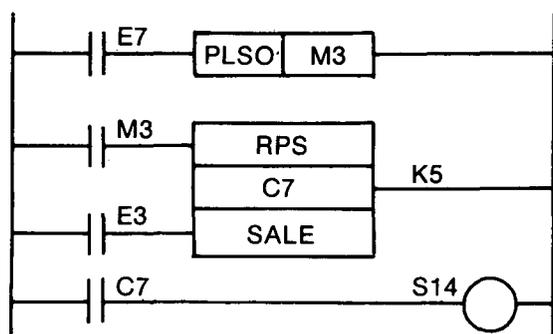
La instrucción **PLSO** se utiliza cuando se requiere

Ejemplos

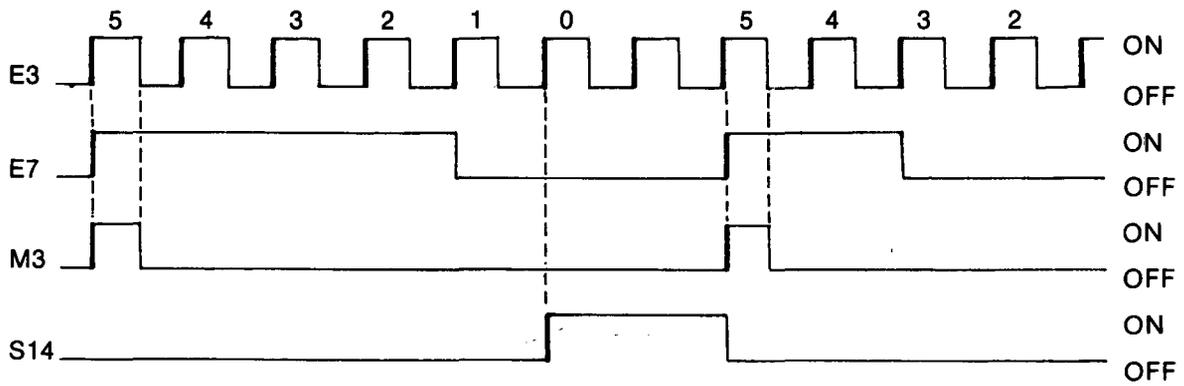


CAR	E7
PLSO	M3

top: tiempo de ejecución de un ciclo del programa.



CAR	E7
PLSO	M3
CAR	M3
RPS	C7
K	5
CAR	E3
SALE	C7
CAR	C7
SALE	S14



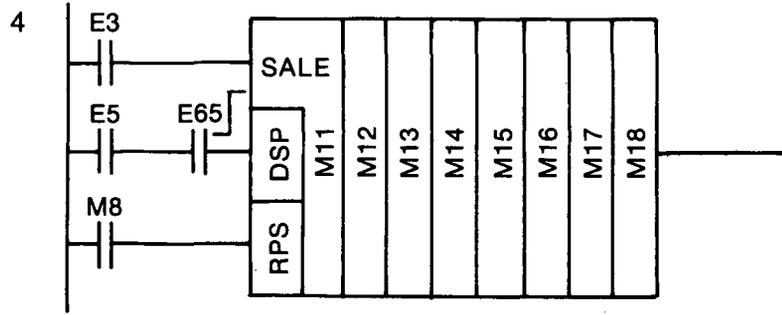
Registros de desplazamiento

Se utilizan para secuenciar eventos. Un registro tiene 3 entradas: **SALE**, **DSP** y **RPS**. **SALE** es el dato de la primera memoria del registro. **DSP** desplaza el contenido de cada memoria un lugar a la derecha. El dato inicial que tenía la última se pierde. **RPS** (reposición) coloca en ceros los

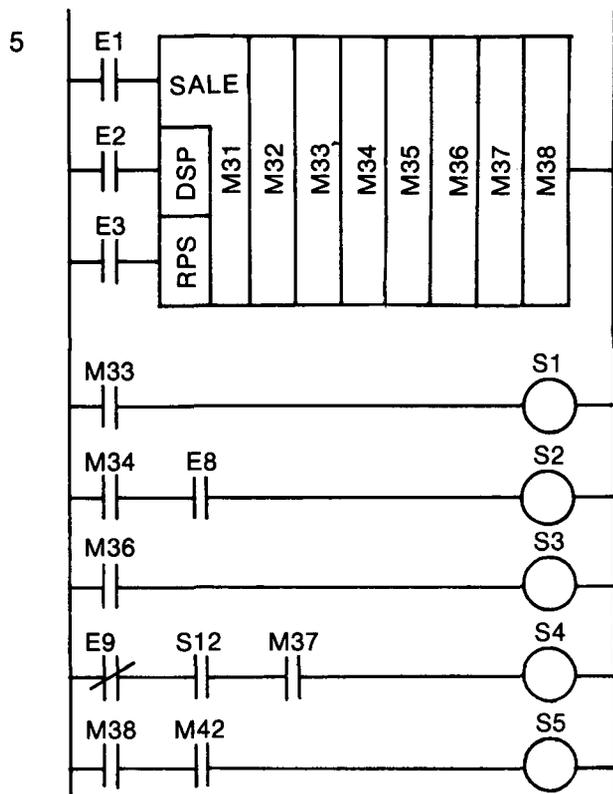
datos de las últimas 7 memorias. El dato de la primera memoria es independiente de las entradas **RPS** y **DSP**.

El estado de las 8 memorias puede utilizarse como entrada en cualquier número de lugares necesarios.

Ejemplos:



CAR	E3
SALE	M11
CAR	E5
Y	E6
DSP	M11
CAR	M8
RPS	M11



CAR	E1
SALE	M31
CAR	E2
DSP	M31
CAR	E3
RPS	M31
CAR	M31
SALE	S1
CAR	M32
Y	E8
SALE	S2
CAR	M33
SALE	S3
CARN	E9
Y	S12
Y	M34
SALE	S4
CAR	M36
Y	M42
SALE	S5

Ejemplo final

Se tiene una máquina que opera en forma secuencial, según la siguiente matriz:

hay 2 entradas:

E1: arranque

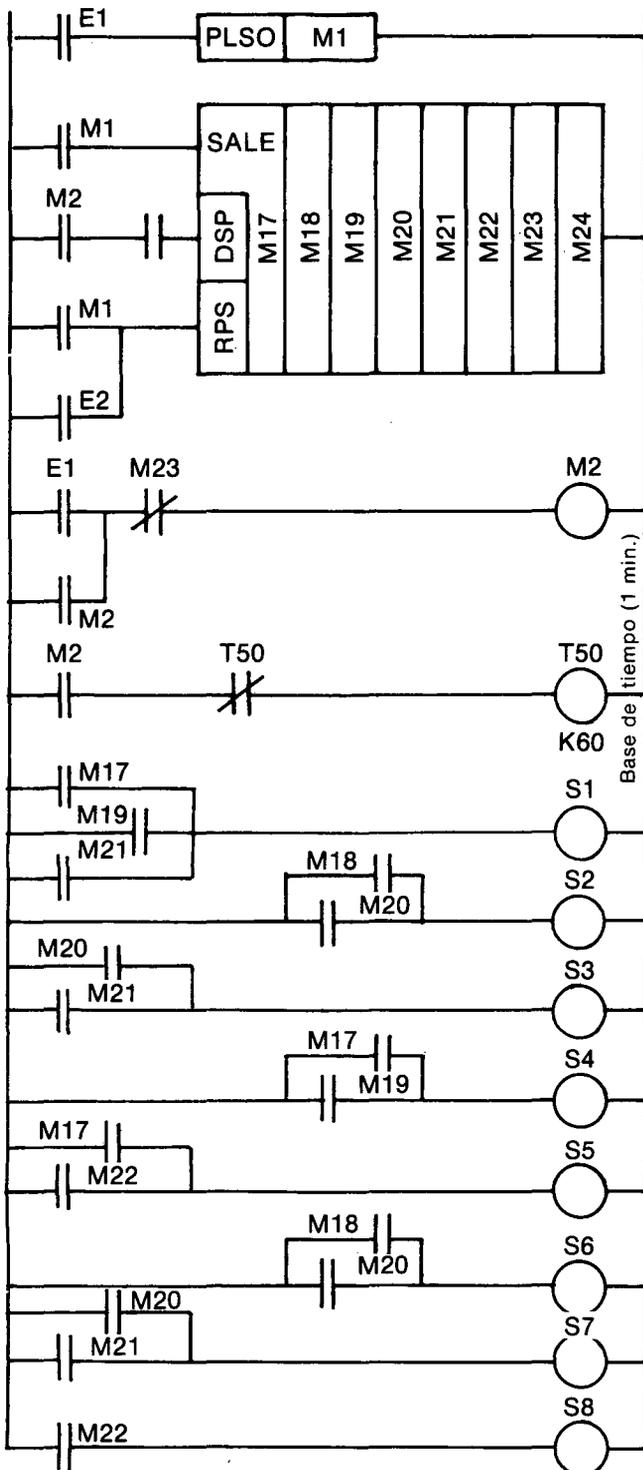
E2: parada

cada paso dura un minuto

Programación:

A cada paso se le asigna una memoria de trabajo:

Salida	S1 - Motor 1	S2 - Motor 2	S3 - Motor 3	S4 - Válvula 1	S5 - Válvula 2	S6 - Válvula 3	S7 - Válvula 4	S8 - Válvula 5
	Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso 6		
Paso 1	X			X	X			
Paso 2		X				X		
Paso 3	X		X	X				
Paso 4		X	X	X	X	X		
Paso 5	X		X			X	X	
Paso 6				X				X



Paso	Memoria
1	M17
2	M18
3	M19
4	M20
5	M21
6	M22
7	M23 indicador de fin de ciclo.

Programa	Programa
CAR	SALE S1
PLSO	CAR M18
CAR	O M20
SALE	SALE S2
CAR	CAR M20
Y	O M21
DSP	SALE S3
CAR	CAR M17
O	O M19
RPS	SALE S4
CAR	CAR M17
O	O M22
YN	SALE S5
SALE	CAR M18
CAR	O M20
YN	SALE S6
SALE	CAR M20
K	O M21
CAR	SALE S7
O	CAR M22
O	SALE S8
	FIN

REFERENCIAS

1. Documentos internos de trabajo de la línea de investigación "Adaptación y creación de tecnología en el área del control digital".
2. ASEA. "Asea master piece. Programmable control systems". Folleto de presentación.
3. ASEA. 4'Asea master piece 100. Manual CA06-1000EL2".
4. WESTINGHOUSE. "Westinghouse quick selector. Catalog 25-000". Edición 9.