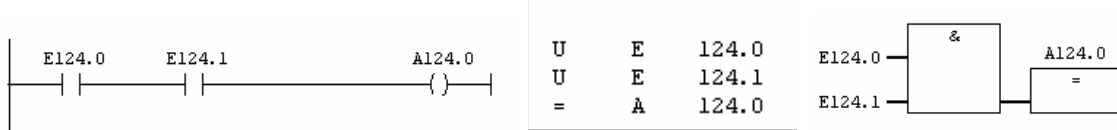
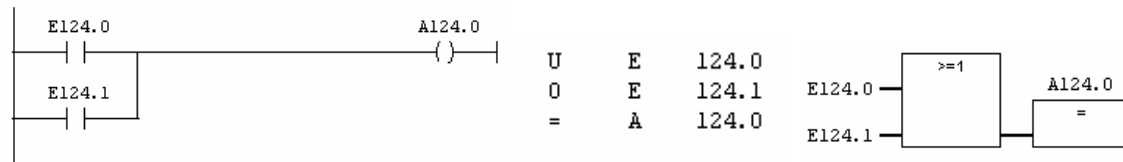


## STEP-7c EJERCICIOS DE INTRODUCCIÓN - REPASO

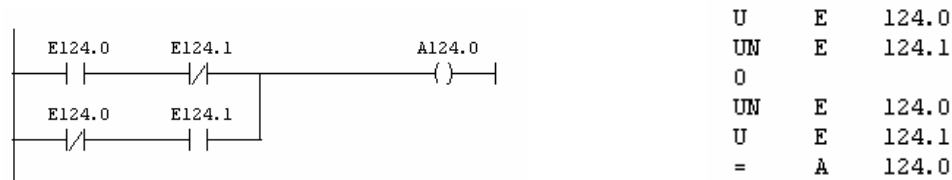
1) Programación en KOP, AWL y FUP, de contactos en serie (U), para activar una Salida.



2) Programación en KOP, AWL y FUP, de contactos en paralelo (O), para activar una Salida.



3) Programación de la O Exclusiva (hay Salida si se activa una u otra Entrada, pero no con las dos).

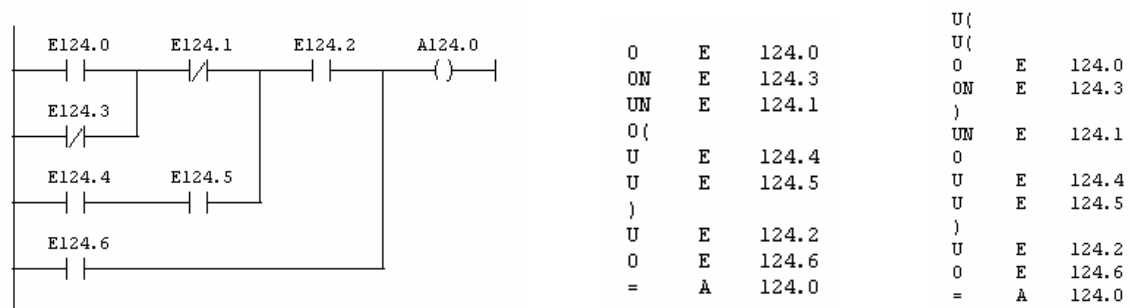


Las dos formas siguientes, son empleando la operación X, que en KOP, no existe.



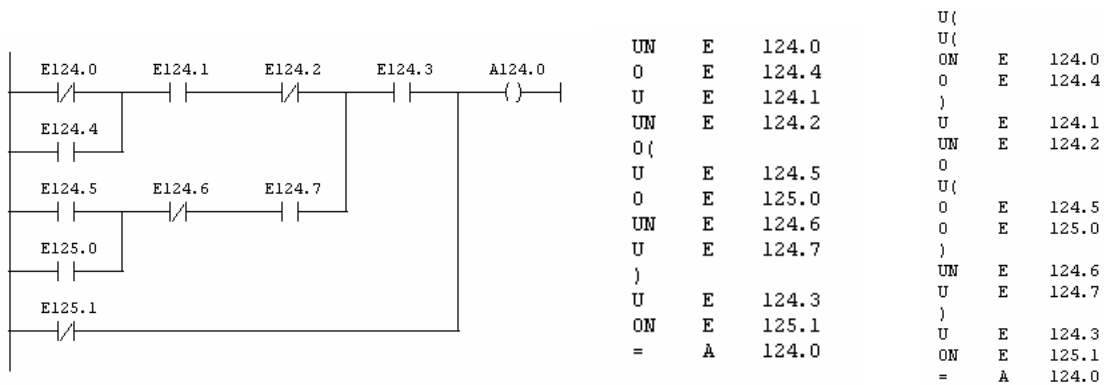
La operación O Exclusiva negada XN (hay Salida si se activan las dos Entradas o ninguna), se realiza negando una de las dos X.

4) Programar en AWL el siguiente programa de KOP



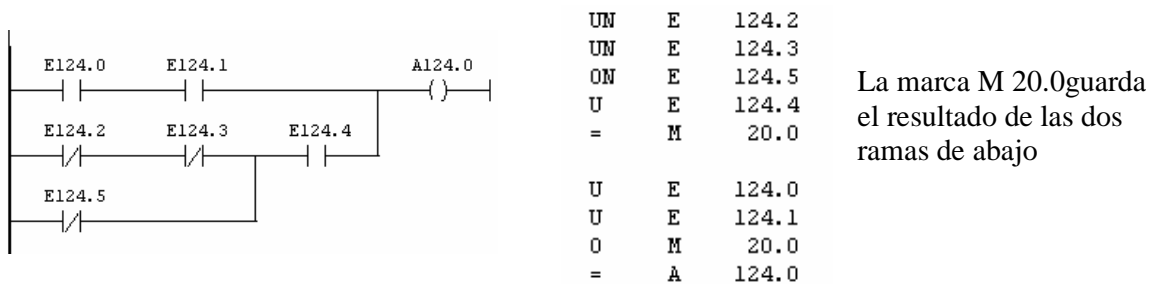
El primer programa es como lo programaríamos lógicamente y el segundo como lo hace el programa Step-7 para que sea convertible a KOP y FUP.

5) Programar en AWL el siguiente programa de KOP.

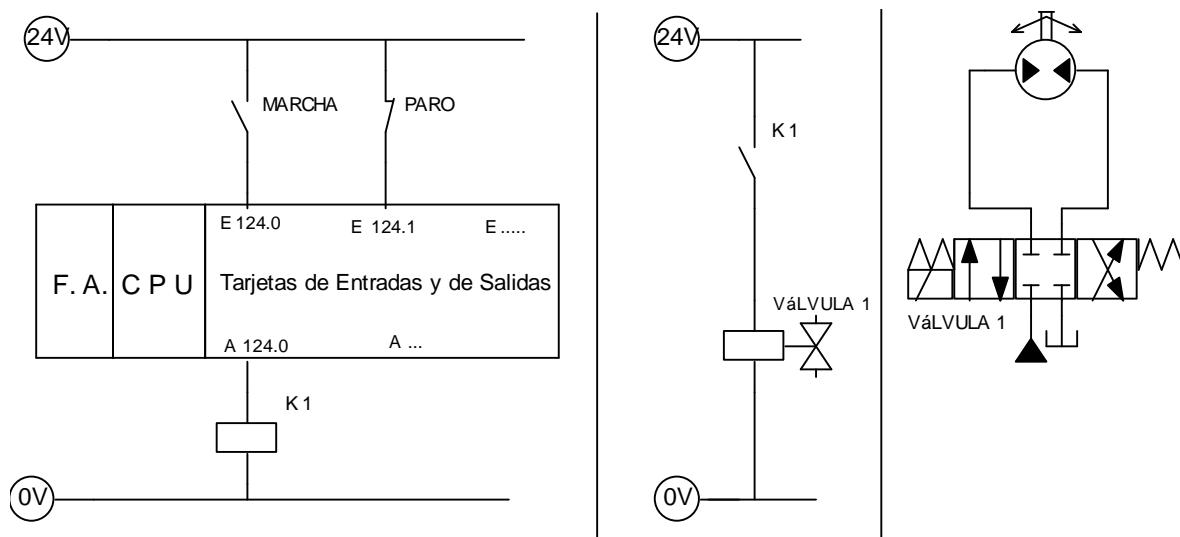


El primer programa es como lo programaríamos lógicamente y el segundo como lo hace el programa Step-7 para que sea convertible a KOP y FUP.

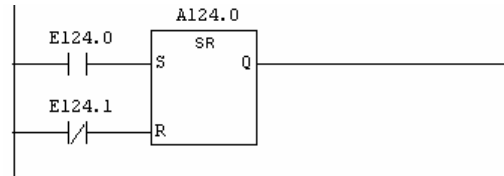
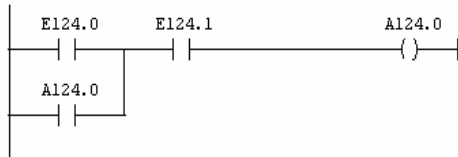
6) Programar en AWL el siguiente programa de KOP, empleando una marca que almacene un resultado intermedio, en vez de usar paréntesis.



7) Programar el accionamiento de un motor hidráulico con pulsadores de marcha n.a. y paro n.c.. Fijarse en el razonamiento empleado, de que son n.a. o n.c. los pulsadores reales y que los contactos de programación los entendemos como afirmados o negados. Suponiendo que las salidas del PLC, no tienen suficiente potencia para accionar directamente a la válvula, lo vamos a hacer a través de un relé con una F.A. externa. Las conexiones son las siguientes:



A continuación va el programa hecho de dos formas, primero con una activación = , y segundo con una activación S-R (fijarse en como está programado el paro).

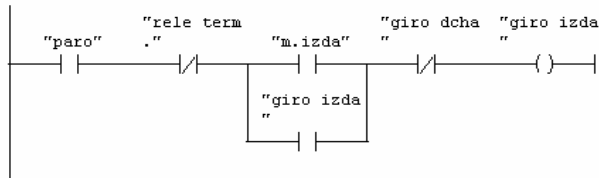


8) Programar las activaciones para el giro de un motor en los dos sentidos, con pulsadores de: m.izda., m.dcha., contacto de relé térmico, normalmente abiertos y pulsador de paro n.c. Para poder cambiar el sentido de giro, hay que pasar por paro.

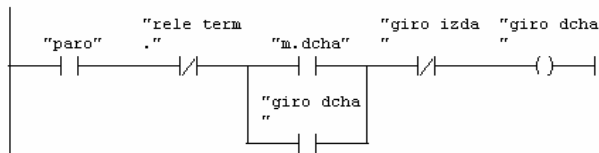
Con activaciones =

OB1 : "Main Program Sweep (Cycle)"

Segm. 1: Activacion del contactor de giro a izquierda



Segm. 2: Activacion del contactor de giro a derecha

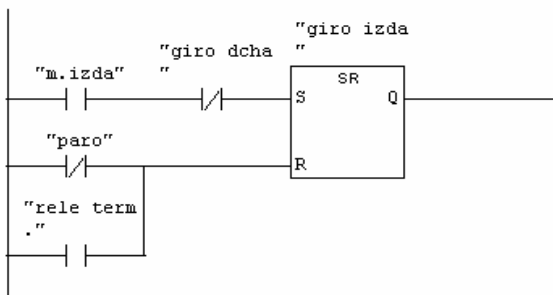


Los enclavamientos de giro dcha para el contactor de giro izda, y viceversa, impiden que cuando el motor esté girando en un sentido, pueda girar en el otro. Para cambiar el sentido, hay que pasar primero por el paro.

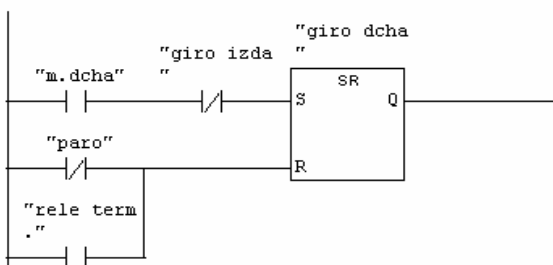
Con activaciones S-R

OB1 : "Main Program Sweep (Cycle)"

Segm. 1: Activacion del contactor de giro a izquierda



Segm. 2: Activacion del contactor de giro a derecha



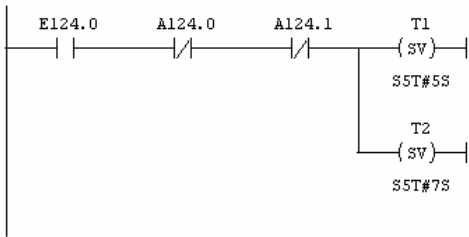
Observar que los paros (paro y relé térmico), van programados al revés que en el caso anterior; en vez de en serie y afirmado-negado, ahora van en paralelo y negado-afirmado. Los enclavamientos de giro dcha para el contactor de giro izda y viceversa, van igual

9) Hacer el programa para que al activar un pulsador (E 124.0) se active una salida (A 124.0), durante 5 seg, y seguidamente se active otra (A 124.1) durante 2 seg.

Funcionaria con lo siguiente. Si queremos que una vez iniciado el ciclo, no se pueda volver a reiniciar hasta que no acabe, metemos un “contacto” de A 124.0 y otro de A 124.1 negados y en serie después de la marcha.

OB1 : Título:

Segm. 1: 5 seg 2 seg (7 seg.)



**Segm. 2:** Activaciones

U	T	1
=	A	124.0
U	T	2
UN	A	124.0
=	A	124.1

Otra solución sería la siguiente:

OB1 : Título:

Segm. 1: 5 seg 2 seg



Segm. 2: Título:

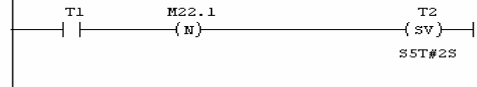


**Segm. 3:** Activaciones

U	T	1
=	A	124.0
U	T	2
=	A	124.1

Aunque el temporizador T2, parece que se va a arrancar al inicio del ciclo, por que el T1 no esta activado; no se arranca por que los temporizadores, arrancan en el cambio de RLO, no cuando el RLO anterior es 1. Sí se activa la primera vez que arrancamos el PLC; esta posibilidad la eliminamos si ese segmento lo programamos como está a continuación, empleando la operación FN.

**Segm. 2:** Título:



Sí además queremos que no se pueda reanunciar el ciclo hasta que no acabe, tendremos que poner alguna condición más, en serie con la marcha; esta condición la podemos generar con una Marca (M 20.0 en este caso). A continuación, hay dos formas de generarla, la primera basándonos en el diagrama de mando o de estados representado, y la segunda con flancos que es la que empleamos en el segmento 4 del final de este mismo ejercicio.

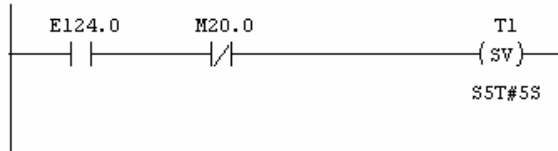
	b	1	a		b	a
E	0	1				
T1	0					
T2	0					
	1		▼ S			▼ S
M R	0			▼ R		

UN E 124.0  
 U T 1  
 UN T 2  
 S M 20.0  
 UN E 124.0  
 UN T 1  
 UN T 2  
 R M 20.0  
 NOP 0

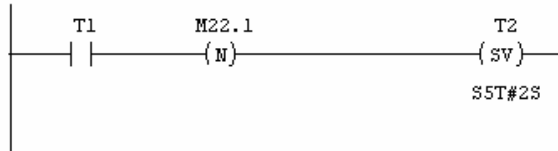
Para programar esta marca con el grafico de estados, nos fijamos en los estados de la E, T1 y T2 en las **zonas estables a y b**. Con esto en el segmento 4, funcionaria igualmente el programa.

OB1 : Título:

Segm. 1: 5 seg 2 seg



Segm. 2 : Título:



Segm. 3): Título:

U	T	1
=	A	124.0
U	T	2
=	A	124.1

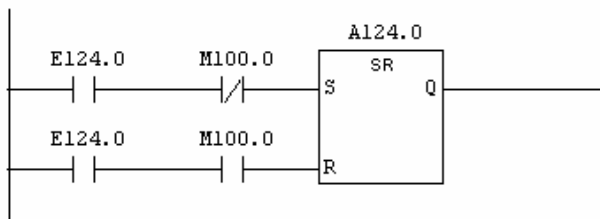
Segm. 4 : Título:

U	T	1
FP	M	21.0
S	M	20.0
U	T	2
FN	M	21.1
R	M	20.0

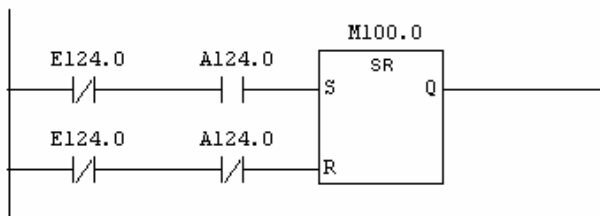
10) Hacer el programa para que al accionar un pulsador se active una salida, al volverlo a accionar se desactive; y así sucesivamente (Divisor Binario).

OB1 : Título:

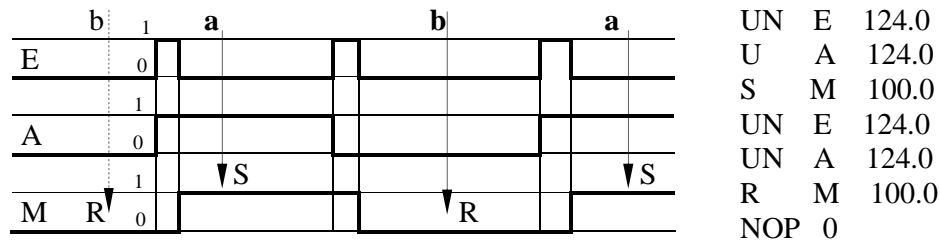
Segm. 1): Título:



Segm. 2 : Título:



En el diagrama siguiente, se ve como se ha programado la marca M 100.0 del Segm. 2; que es la que hace la condición adicional para que en Segm. 1, el pulsador, no haga a la vez el S y el R.



11) Intermitencias.- Sobre todo para activar lámparas de aviso, se necesita que haya salidas que se activen intermitentemente; a continuación, vamos a ver tres formas de conseguirlo:

- a) Empleando los bits de la Marca de Ciclo, para lo cual hay que activarla yendo a: *HW-Config -> seleccionar el slot de la CPU-> seleccionar la ficha Ciclo/Marca de Ciclo -> activar la casilla de Marca de Ciclo y poner la dirección del byte que queremos usar para marca de Ciclo -> ir saliendo, guardando los cambios y al cargar el programa en el PLC, cargar también los datos del sistema.* Con esto si por ejemplo hemos cogido el MB 40 para Marca de Ciclo (ver menú contextual cuando estemos en la fichaCiclo/Marca de Ciclo), haciendo:

```
U M 40.0
= A 124.0 //tendremos intermitencias de 0,1 seg. en A 124.0
```

```
U M 40.7
= A 124.1 //tendremos intermitencias de 2 seg. en A 124.1
```

- b) Generar un temporizador que se autoarranque (UN T 0) y cargar la palabra de tiempo en una Marca, en los bits de esta Marca tenemos distintas intermitencias para activar salidas. Podemos poner previamente un interruptor (S-R de una Marca) hecho con marcha y paro o poner una entrada que haga de interruptor directamente. Ejemplo:

```
U E 124.0 //marcha del interruptor M 42.0
S M 42.0
U E 124.1 //paro del interruptor M 42.0
R M 42.0
```

```
U M 42.0 //si no ponemos nada delante de UN T 0, T 0 se
autoarranca y estaría contando siempre.
```

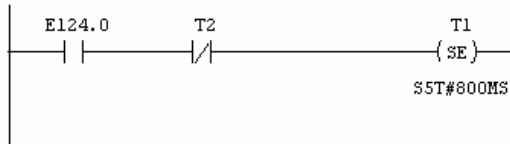
```
UN T 0
L S5T#10S
SE T 0
```

```
L T 0 // cargar la palabra de tiempo del T 0
T MW 40 //transferirla a una MW
```

```
U M 41.3
= A 124.0 //tendremos una de las intermitencias generadas.
```

- c) Intermitencias con dos temporizadores.

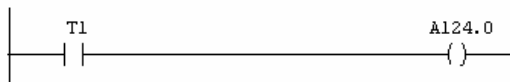
OB1 : Título:  
**Segm. 1:** Intermi



**Segm. 2:** Intermi



**Segm. 3:** Título:



Algo análogo a esto se puede hacer modificando a nuestro gusto los tiempos del ejercicio 9 – b, y metiendo después de la marcha un contacto negado del T2, para que se reinicie el ciclo automáticamente

12) (Bimanual de seguridad). Programa para bajar y subir una puerta. La bajada se realiza al pulsar con una diferencia menor de 500 ms. y mantener dos pulsadores. Al accionar el fc-abajo, se desactiva la bajada y además suena un timbre de aviso para soltar los pulsadores. La subida se realiza automáticamente al pulsar “subir” hasta que se accione el fc-arriba.

OB1 : "Main Program Sweep (Cycle)"

**Segm. 1:** Arrancar T1



**Información del símbolo:**

E124.0	bajar1
--------	--------

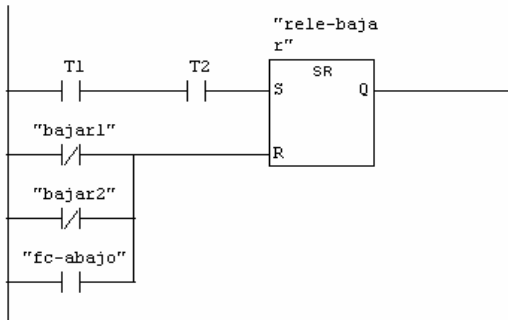
**Segm. 2:** Arrancar T2



**Información del símbolo:**

E124.1	bajar2
--------	--------

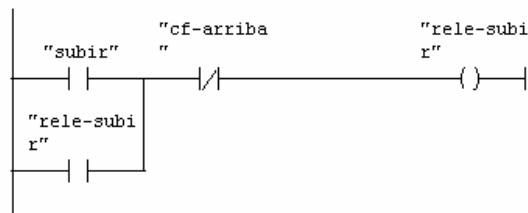
**Segm. 3:** Bajar puerta



**Información del símbolo:**

E124.0	bajar1
E124.1	bajar2
E124.2	fc-abajo
A124.0	rele-bajar

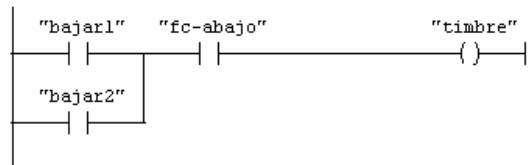
**Segm. 4:** Subir puerta



**Información del símbolo:**

E124.3	subir
A124.1	rele-subir
E124.4	cf-arriba

**Segm. 5:** Timbre



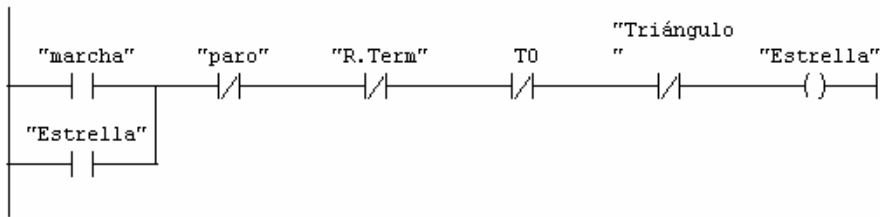
**Información del símbolo:**

E124.0	bajar1
E124.1	bajar2
E124.2	fc-abajo
A124.2	timbre

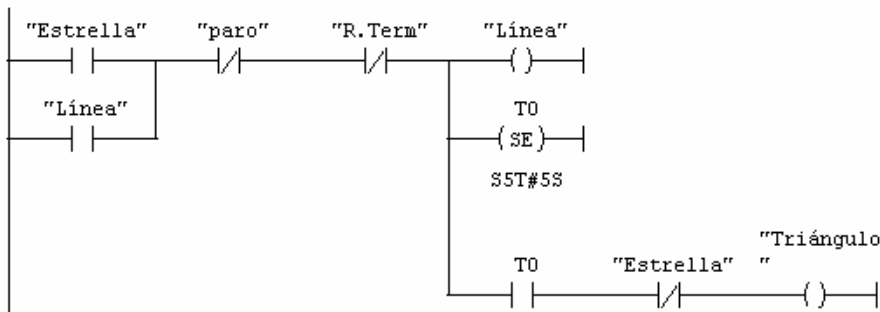
13) Realizar el programa de mando de un arranque estrella-triángulo; con: marcha (E 124.0), paro (E 124.1) y relé térmico (E 124.2), n.a. Contactores de: Estrella (A 124.0), Línea (A 124.1) y Triángulo (A 124.2).

OB1 : "Main Program Sweep (Cycle)"

**Segm. 1:** Activación del contactor de Estrella



**Segm. 2:** Activación del contactor de Línea, Temporizador y Triángulo



14) Haz el programa para que al accionar un pulsador la primera vez, se active una salida; al accionarlo otra vez, se active una segunda salida, y al accionarlo otra vez, se desactiven las dos. A continuación, van dos soluciones parecidas.

OB1 : "Main Program Sweep (Cycle)"

**Segm. 1:** Título:

```

U   E   124.0
ZV  Z   0

L   Z   0
L   1
==I
S   A   124.0

L   Z   0
L   2
==I
S   A   124.1

L   Z   0
L   3
==I
R   A   124.0
R   A   124.1
R   Z   0

```

OB1 : "Main Program Sweep (Cycle)"

**Segm. 1:** Título:

```

U   E   124.0
ZV  Z   0

U   Z   0
=   A   124.0

L   Z   0
L   2
==I
=   A   124.1

L   Z   0
L   3
==I
R   Z   0

```

15) Generador de Impulsos. Cada cierto tiempo, y solo en un ciclo de sacan, se produce un impulso o activación de lo que programemos; en este caso se activa la Marca M 20.0. Con este impulso, podemos contar espacios de tiempo, llamar a un bloque de programa para que se lea solo cada cierto tiempo, etc. Con lo siguiente, ya está programado:



```

UN  M  20.0  //Al inicio, no está activada esta marca, pero como la negamos, el RLO es 1
L   S5T#3S   y se arranca el temporizador.
SE  T  1
U   T  1      //Cuando se cumpla el tiempo, se va a activar la salida del T1, y también la
=   M  20.0   M 20.0, con lo cual el RLO de la primera instrucción es 0 y se para el T1
                        y se desconectan la salida del T1 y la M 20.0. Con lo cual estamos como al
                        principio y se reinicia otra vez el Ciclo.

```

16) Temporizar mas de 2h46m30s. Por ejemplo activar una salida durante 8 horas; lo vamos a ver de dos formas: a) Con temporizadores en serie, b) Contado 4 impulsos de 2 horas.

Soluciones en fotocopias aparte

17) Programar el automatismo de un garaje sencillo para 5 coches, que tiene a la entrada una barrera y un sensor de apertura, y lo mismo a la salida; además tiene dos indicadores de Ocupado o Libre. La barrera de entrada, sube solo si esta activado su sensor y hay sitio libre. Si activamos el sensor de entrada y el garaje esta lleno, no se modifica el contador.

**Segm. 1:** Subir barreras, (la de la entrada; sube, solo si hay sitio).

La barrera de la entrada sube, solo si esta activado su sensor y no esta lleno el garaje. Al no subir (bajar) la barrera cuando el contador esta a 5, aunque se active el sensor, tampoco se modifica el contador.

```

UN  A  4.3      //señal, de Lleno
U   E  0.0      //sensor entrada
=   A  4.0      //barrera de entrada
U   E  0.1      //sensor de salida
=   A  4.1      //barrera de salida

```

**Segm. 2:** Señales de Lleno y de Libre.

Comentario:

```

L   Z  1        //contador
L   5
<I
=   A  4.2      //señal de Libre
UN  A  4.2      //señal de Libre
=   A  4.3      //señal de Lleno

```

**Segm. 3:** Según la barrera que baje, contar o descontar

Se emplea FN para la barrera de la entrada, por que si pone a 5 el contador nada mas empezar a subir la barrera, se activaría la señal de Lleno, y según el segmento 1, se bajaría la barrera casi sin haber empezado a subir, con lo cual no podría entrar el 5º coche. Para la salida, no haría falta poner el FN.

```

U   A  4.0      //barrera de entrada
FN  M  100.0
ZV  Z  1        //contador
U   A  4.1      //barrera de salida
FN  M  100.1
ZR  Z  1

```