

ISBN 84-9705-956-5



9 788497 059565



0256P09

AUTOMATIZACIÓN DE MANIOBRAS INDUSTRIALES MEDIANTE AUTÓMATAS PROGRAMABLES

Juan Pérez Cruz
Manuel Pineda Sánchez

Libro orientado a la iniciación en el mundo de la automatización industrial mediante autómatas programables. Se presenta una breve introducción a la configuración hardware y software de un autómata tipo y de sus modos de funcionamiento. A continuación se analizan en detalle las instrucciones básicas del lenguaje de programación (asignaciones, combinaciones digitales, temporizadores, contadores, activación y desactivación de bits), que se ponen en práctica mediante sencillos ejemplos de maniobras industriales. Finalmente, se propone una serie extensa de maniobras típicas en el ámbito industrial (arranque estrella triángulo, dosificación de depósitos, garaje de dos plantas, etc.), que se resuelven en detalle aplicando las herramientas introducidas en los apartados anteriores. La diversidad de los ejemplos seleccionados, fruto de la experiencia en el terreno de la automatización industrial de los autores, dotan al presente libro de un enfoque eminentemente práctico.



UNIVERSIDAD
POLITÉCNICA
DE VALENCIA
EDITORIAL

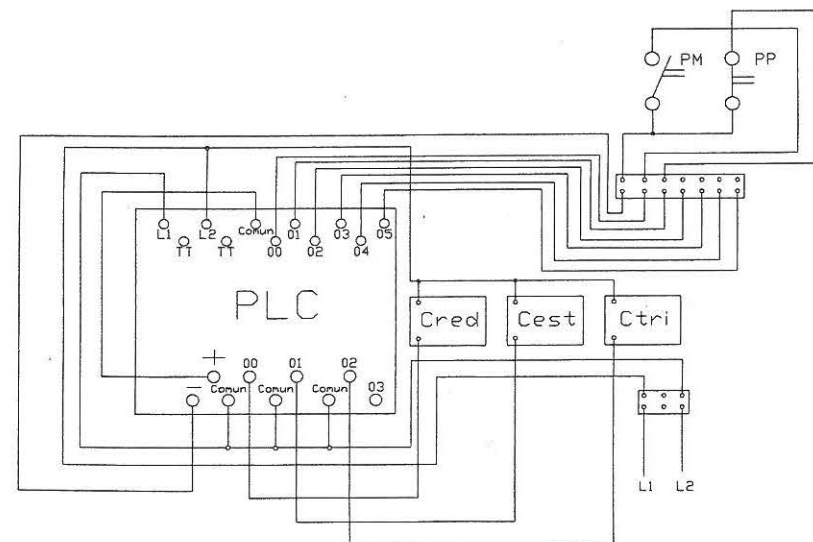
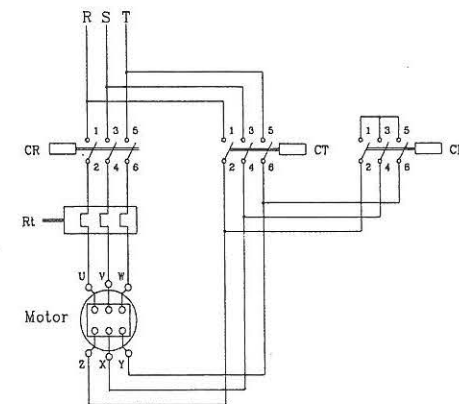
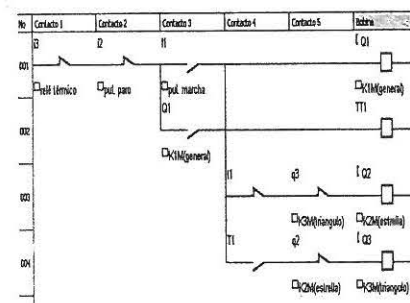
Juan Pérez Cruz
Manuel Pineda Sánchez

AUTOMATIZACIÓN DE MANIOBRAS INDUSTRIALES
MEDIANTE AUTÓMATAS PROGRAMABLES

Ref.: 2006 - 256

AUTOMATIZACIÓN DE MANIOBRAS INDUSTRIALES MEDIANTE AUTÓMATAS PROGRAMABLES

Juan Pérez Cruz
Manuel Pineda Sánchez



EDITORIAL
UNIVERSIDAD POLITÉCNICA DE VALENCIA

PRÓLOGO

Este trabajo se realiza con la finalidad de que sirva como ayuda a los nuevos programadores de las automatizaciones eléctricas industriales mediante autómatas programables

En la primera parte de este trabajo se exponen una serie de conocimientos mínimos necesarios para entender el funcionamiento básico de los autómatas programables, para posteriormente desarrollar un gran número de ejemplos de maniobras reales. La experiencia de los autores en las automatizaciones eléctricas industriales ha servido como base para la elección de los ejemplos desarrollados.

Los ejemplos que se recogen en estos apuntes se han ido generando atendiendo a un orden creciente de complejidad, además de haber sido expuestos y resueltos en los últimos años durante la impartición de más de 15 cursos. La experiencia adquirida durante estos cursos ha dado lugar a los diferentes ejemplos seleccionados.

Los autores piensan, que además de este gran número de ejemplos aportados, los usuarios de este manual deben de intentar complementar con sus propias aportaciones innovando y adaptando los ejemplos que se exponen a aplicaciones similares, además de tratar de implementarlos al menos dos marcas de autómatas programables diferentes. Todo ello dotará al programador de la experiencia necesaria para poder adoptar la solución más propicia en cada automatización eléctrica industrial que se le presente.

Los autores quieren expresar su agradecimiento a la inestimable ayuda recibida por los ingenieros D. Rubén Puche, D. Héctor Llobet y D. Emilio Aragón, en la impartición de algunos de estos cursos, en la exposición y resolución de algunos de los ejemplos, así como en la redacción de los mismos.

© Juan Pérez Cruz
Manuel Pineda Sánchez

Edita: EDITORIAL DE LA UPV
Camino de Vera, s/n
46071 VALENCIA
Tel.96-387 70 12
Fax 96-387 79 12

Imprime: REPROVAL, S.L.
Tel.96-369 22 72

Depósito Legal: V-1418-2006
ISBN: 84-9705-956-5

ÍNDICE

1. OBJETIVO	7
2. INTRODUCCIÓN	7
3. DESCRIPCIÓN DEL EQUIPO	8
3.1. CPU	9
3.2. MEMORIA	9
3.3. FUENTE DE ALIMENTACIÓN	10
3.4. MÓDULOS AUXILIARES	10
3.5. PROGRAMADORA	11
3.6. OTROS	11
4. MODO DE FUNCIONAMIENTO	11
5. CONFIGURACIÓN EXTERNA DEL EQUIPO	12
6. CONFIGURACIÓN INTERNA DEL EQUIPO	13
6.1. MAPA DE MEMORIA	14
6.2. TEMPORIZADORES Y CONTADORES	15
7. INSTRUCCIONES BÁSICAS	16
8. MONTAJE	16
9. COMBINACIONES DIGITALES BÁSICAS	18
9.1. IDENTIFICADORES	18
9.2. COMBINACIONES DIGITALES BÁSICAS	19
9.3. CONEXIONES EN PARALELO	22
9.4. CONEXIONES EN SERIE	23
9.5. CONEXIONES EN SERIE Y PARALELO	24
9.6. LIMITACIONES EN MONTAJE ELEMENTALES	25
9.7. COMPROBACIÓN DE ENTRADA Y DE SALIDAS	26
9.8. PRUEBA DE PILOTOS Y APAGADO GENERAL	27
10. TEMPORIZADORES	27
10.1. RETARDO A LA CONEXIÓN	28
10.2. TEMPORIZADOR DE IMPULSO	29
10.3. RETARDO A LA DESCONEXIÓN	29
11. INTERMITENCIAS	30
12. CONTADORES	31
12.1. CONTADOR EN UN SOLO SENTIDO	31
12.2. CONTADOR EN AMBOS SENTIDOS	32
12.3. EL CONTADOR COMO TEMPORIZADOR	33

13. FUNCIONES BÁSICAS DE ACTIVACIÓN Y DESACTIVACIÓN DE BITS	34
13.1. ACTIVACIÓN/DESACTIVACIÓN DE UNA SALIDA (SET/RESET).....	34
13.2. INSTRUCCIÓN SET-RESET CONJUNTA (KEEP).....	36
13.3. INSTRUCCIÓN DETECCIÓN DE FLANCO DE SUBIDA (DIFU).....	36
13.4. INSTRUCCIÓN DETECCIÓN DE FLANCO DE BAJADA (DIFD).....	37
13.5. BIESTABLE MEDIANTE UNA ÚNICA ENTRADA DE MANDO.....	39
13.6. MEMORIZACIÓN DEL ESTADO DE UNA SEÑAL CON CAÍDAS DE TENSIÓN.....	40
14. COMPARACIONES	41
15. PROGRAMACIÓN DE MANIOBRAS INDUSTRIALES	43
15.1. COMBINACIONES LÓGICAS DE ENTRADAS/SALIDAS.....	45
15.2. COMBINACIÓN ENTRADAS/SALIDAS Y BITS DEL SISTEMA.....	48
15.3. ENCENDIDO DE LUZ DE ESCALERA.....	50
15.4. LIBERACIÓN Y BLOQUEO DE UN MOTOR.....	54
15.5. APERTURA Y CIERRE DE UNA BARRERA DE PARKING.....	56
15.6. AUTOMATIZACIÓN DE UN APARATO ELEVADOR CONVENCIONAL DE DOS PLANTAS.....	58
15.7. ARRANQUE Y PARO DE UN MOTOR CON UN SOLO PULSADOR..	60
15.8. DETECCIÓN DEL FLANCO DE SUBIDA Y BAJADA DE UNA ENTRADA.....	62
15.9. ACTIVACIÓN Y DESACTIVACIÓN DE UNA SALIDA MEDIANTE FLANCO DE SUBIDA Y BAJADA.....	65
15.10. ENCENDIDO Y APAGADO DE UNA LUZ CON TRES PULSADORES.....	69
15.11. CONTROL DE LA CAPACIDAD DE UN DEPÓSITO.....	71
15.12. DETECCIÓN DE NÚMERO DE ENTRADAS ACTIVAS.....	74
15.13. CONTROL DE ACTIVACIÓN DEL MISMO NÚMERO DE ENTRADAS Y SALIDAS.....	77
15.14. REGISTROS DE DESPLAZAMIENTO.....	80
15.15. DESPLAZAMIENTO DE UN DATO A IZQUIERDAS Y DERECHAS.....	82
15.16. CARGA DE UN VALOR EN UNA DIRECCIÓN E INCREMENTO DEL MISMO.....	84
15.17. CARGA E INCREMENTO DE UN VALOR EN UNA DIRECCIÓN MEDIANTE CONTADOR.....	87
15.18. RECORRIDO DE TODA LA MEMORIA DEL AUTÓMATA MEDIANTE PUNTERO.....	90

15.19. ENCODER CON DETECCIÓN DEL SENTIDO DE GIRO.....	93
15.20. ACTIVACIÓN SECUENCIAL DE ETAPAS DE TRABAJO DE UNA LAVADORA.....	97
15.21. PROGRAMACIÓN DE LOS CICLOS DE UNA LAVADORA.....	101
15.22. MOVIMIENTO DE PALABRAS Y DESPLAZAMIENTO DE BITS.....	105
15.23. SIMULACIÓN DE UN TALADRO AUTOMÁTICO.....	111
15.24. PROGRAMACIÓN SECUNECIAL DE UN TALADRO.....	122
15.25. TEMPORIZADORES: INTERMITENCIAS.....	134
15.26. TEMPORIZADORES: RETARDO A LA CONEXIÓN Y DESCONEXIÓN.....	140
15.27. ACCIONAR UN PUNTO DE LUZ DESDE TRES INTERRUPTORES.....	143
15.28. MANTENIMIENTO DEL NIVEL DE UN DEPÓSITO.....	145
15.29. ACCIONAMIENTO DE UN EXPULSOR.....	147
15.30. ACCIONAMIENTO DE UN TALADRO.....	150
15.31. ARRANCADOR ESTRELLA TRIÁNGULO.....	155
15.32. CONTROL DE UN PARKING.....	158
15.33. DOSIFICACIÓN DE UN PRODUCTO.....	162

CONOCIMIENTOS BÁSICOS DE AUTOMATIZACIONES MEDIANTE AUTÓMATAS PROGRAMABLES

1. OBJETIVO

Con el desarrollo de este curso se pretende acercar a los alumnos al elemento de automatización más empleado en la industria actual, el autómata programable (PLC). No se concibe una industria moderna sin estos equipos de automatización industrial. Es por ello que los futuros especialistas dedicados al diseño y mantenimiento de sistemas automatizados deben de tener una serie de conceptos claros de este tipo de equipos, especialmente en cuanto a su programación.

Hoy en día existen en el mercado un gran número de marcas y modelos. Se pretende con el desarrollo de este curso, acercar al alumno a los conceptos generales y comunes de todos ellos. Como finalmente debe de programarse un PLC en concreto, se explicarán oportunamente sus características específicas, aunque los ejemplos se desarrollan empleando las estructuras e instrucciones más comunes a todos los modelos.

2. INTRODUCCIÓN

Las automatizaciones en la industria hasta los años 70 se ejecutaban exclusivamente a base de relés, utilizando lo que se conoce como lógica cableada, pero a partir de estos años irrumpieron con fuerza los Autómatas Programables (PLCs), pasando lógica programada.

Las primeras versiones y modelos de PLCs, eran caros, de difícil programación, con relativa poca memoria y casi sin elementos periféricos. Ya en los años 80 mejoraron estos, tanto en precio como en prestaciones, pero aún la programación era difícil de realizar, lo cual significaba que solo un pequeño grupo de especialistas estuviese capacitado para su puesta en servicio y mantenimiento. Esto hizo que se pusiese mucho empeño en generar una serie de cursos formativos a los profesionales de la automatización. Para ello se necesitó crear alternativas de programaciones que estos especialistas entendieran con claridad.

Así pues se desarrollaron diferentes lenguajes de programación, siendo incluso opcional elegir en cada momento el lenguaje de programación a emplear en un mismo equipo, ya sea a la hora de iniciar un nuevo programa o de poder leer programas ya residentes en la memoria del mismo.

Hoy en día el terreno de la programación de estos equipos se ha facilitado enormemente. Por un lado, existen potentes herramientas de programación, que se pueden ejecutar en entornos gráficos amigables como los de los ordenadores personales. Por otro lado, han aparecido en el mercado programas de simulación que permiten "ejecutar" una automatización sin necesidad del equipo PLC, lo que representa una ventaja importante ya que se puede depurar un programa sin tener que parar la máquina o instalación.

En cuanto al terreno del hardware, la progresión de estos equipos ha sido espectacular, en línea con el incremento sostenido de capacidad experimentado en otros campos como el de la informática personal. No solo se han incrementado las capacidades de los procesadores en término de velocidad de cómputo, capacidad de memoria, etc. También se ha desarrollado de forma paralela el mercado de los periféricos, que permiten la adaptación de las señales para poder integrar al PLC en los procesos productivos: adaptadores de señales analógicas, reguladores PID., comunicación serie/paralelo, telecomunicaciones via MODEM, Internet, etc. Este es un campo abierto actualmente a la investigación, con el resultado final de periféricos cada vez más potentes, autónomos y preparados para su integración en redes de control automatizado, configurando de esta forma sistemas con control distribuido.

Se utilizan en la industria tres formas diferentes de realizar automatizaciones:

- Mediante relés. Este método, empleado de forma clásica, está prácticamente en desuso
- Mediante equipos específicos realizados con microcontroladores. Se utiliza para grandes series de máquinas iguales, pero tiene la dificultad de que su programación es compleja, por lo que solo está al alcance de unos pocos técnicos especialistas su diseño y programación.
- Mediante PLC., la más extendida en la industria actual, dado que se apoya en equipos industriales de amplia difusión, que se pueden programar de forma simple, con lenguajes de programación estandarizados, mediante potentes herramientas de desarrollo ofrecidas por los propios fabricantes.

En los apartados siguientes se expondrá la configuración del Autómata Programable, así como la forma de trabajar de éste.

3. DESCRIPCIÓN DEL EQUIPO

De forma general se puede describir un Autómata Programable como un conjunto de bloques, que asociados y comunicados tiene una configuración análoga a la de un PC. Se irá haciendo mención a las partes de este equipo, comparándolas con las de un ordenador personal, dado que el alumno suele estar familiarizado con él, pensando así que le puede ser más fácil su comprensión.

En la figura 1, se puede observar los bloques principales de un PLC.

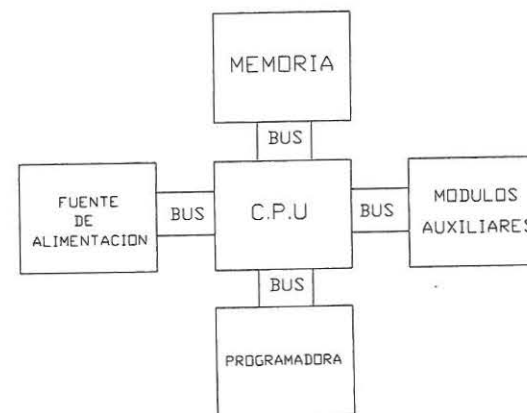


Figura 1

3.1. CPU

Es la parte más importante del equipo, se compone al igual que la del PC de:

- La unidad de control (UC).
- ALU (unidad aritmético/lógica).
- Registros:
 - Registro de control.
 - Puntero de programa (PC).
 - Registro de interrupciones.
 - STACK (zona de memoria interna para salvaguardar registros).
 - Puntero de STACK.

Interesa ver en las bibliografías dedicadas a configuraciones de CPUs, y observar la similitud con los PC.

3.2. MEMORIA

Existe una cantidad de memoria (de forma similar al caso de los PC), dedicada al almacenamiento del programa propio a ejecutar.

Esta memoria suele ser de tipo RAM, la cual mantiene el programa al desconectar la alimentación del suministro eléctrico, por mediación de unas baterías que suelen llevar. Los equipos pequeños y modernos pueden disponer de memoria RAM alimentada mediante condensadores. Permite eliminar la batería, pero en cambio la alimentación no le puede fallar más de un determinado número de horas, ya que tiene el riesgo de perder el programa almacenado.

Existen modelos de PLCs que se le puede adicionar un módulo de memoria EPROM o EEPROM, en la cual se le graba el programa, no existiendo riesgo de pérdida del mismo. Una vez se ha comprobado el buen funcionamiento de un sistema programado, se introduce este programa por mediación de un equipo de grabación bien propio de la marca u otra forma recomendada por el fabricante.

A diferencia de los PCs, la cantidad de memoria que poseen estos equipos es comparativamente reducida. Se puede decir que hoy en día un PLC con 32 K de memoria ya es un equipo que permite la automatización de montajes complejos y grandes, siendo poco usual encontrar equipos con más memoria. Los equipos de bajo nivel, que son los que se emplean para la realización de esta práctica y sistemas pequeños, tales como control de una máquina o procesos pequeños, vienen a tener sobre 1 K de memoria.

3.3. FUENTE DE ALIMENTACIÓN

Cabe resaltar que realmente el equipo interiormente trabaja de forma digital, y con señales de baja tensión (del orden de 5 voltios), mientras que las redes a las que van conectados son de 220 V. O 380 V. por lo general.

Estas fuentes de tensión pueden venir formando parte del equipo propiamente dicho o como un módulo que se le puede adicionar.

Por otro lado es también usual que suministre el PLC, de una tensión de 12 o 24 voltios para la alimentación de módulos auxiliares de entrada o salida de señales, incluso de aparatos que se utiliza, como detectores de posicionamiento, fotocélulas, sensores electrónicos, etc., siendo en todo caso y por lo general de pequeña potencia estas fuentes de tensión.

3.4. MÓDULOS AUXILIARES

En este apartado caben una serie de elementos que se utilizan para acondicionar las señales de entrada y salida del PLC a los distintos actuadores y detectores. Por la gran variedad que hay de ellos se recomienda ir a catálogos especializados de la marca de Autómata a utilizar. Se podría indicar como más generales los siguientes:

- Módulos de Entrada/Salida.
- Módulo convertidores de señales Analógicas/Digitales.
- Módulos de regulación P, PI, PID, etc.
- Módulos de entrada de contadores rápidos.
- Módulos de temporizadores.

En los PLC de alto nivel es fácil encontrar entradas integradas en él que ya poseen alguna aplicación de los tipos indicados. No obstante frecuentemente o bien son insuficientes o no llevan este tipo de entradas/salidas. Por ello permiten adicionarles módulos de los anteriormente descritos.

3.5. PROGRAMADORA

Con este nombre se quiere atender al equipo que sirve para la introducción del programa, o bien para la rectificación de este. Puede ser una programadora de las denominadas de bolsillo, o simplemente un PC con el software de programación adecuado.

3.6. OTROS

A su vez cabe resaltar que se pueden conectar a la CPU. otra serie de periféricos, tales como pantallas táctiles y visualizadoras, empleadas utilizadas para visualizar estados y cambios de algún parámetro, y comunicarse por la entrada serie con otros elementos, tales como módems, variadores de frecuencia, etc.

Estos posibles periféricos no se han colocado en el diagrama de la figura 1, ya que son aplicaciones particulares.

4. MODO DE FUNCIONAMIENTO

Principalmente hay que resaltar que, en funcionamiento normal, estos equipos ejecutan el programa almacenado en su memoria de forma lineal y cíclica. Es decir la CPU lee las instrucciones del programa y las ejecuta de modo secuencial, y al llegar al final del listado de instrucciones vuelve a comenzar la ejecución por la primera de ellas.

Los PLCs pueden trabajar generalmente en tres modos de funcionamiento:

- **Modo Stop;** Bajo esta denominación se está en comunicación el PLC con el elemento de programación (un PC en nuestro caso), pero se encuentra parada la ejecución del programa, con las salidas normalmente a nivel bajo.

De esta forma es como se procede a la carga de los programas y modificaciones importantes de ellos, que requieran que el Autómata mantenga parado el proceso de producción.

Cuando se dan incidencias importantes en las secuencias del programa (tales como interrupciones de programa, ordenes contradictorias o situaciones de indecisión) el PLC, suele pasar a esta forma de funcionamiento para no permitir situaciones peligrosas.

- **Modo Monitor;** Cuando se pasa a esta forma de funcionamiento el PLC y el elemento de programación están transfiriéndose información en tiempo real. De esta manera se puede realizar un análisis del funcionamiento del programa en tiempo real, monitorizando en cada instante el estado de cualquier entrada/salida o elemento programado. También permite el PLC hacer pequeñas correcciones de programa, estando en este modo de funcionamiento. Incluso se permite forzar a un estado lógico entradas o salidas.

Tal vez sea el estado de funcionamiento que permita al alumno un seguimiento de lo proyectado y programado, durante la ejecución de la práctica.

- **Modo Run;** Este es el modo de funcionamiento para el que normalmente está pensado en PLC, es decir desconectado del elemento programador, atendiendo a las entradas y salida ejecutando el programa desarrollado a su máxima velocidad.

Lógicamente es el estado más usual ya que una vez en marcha el sistema de automatizado de un proceso, no tiene sentido estar conectado a ningún elemento capaz de modificar el programa ya que se corre el riesgo de producir modificaciones que pueden llegar a ser peligrosas.

5. CONFIGURACIÓN EXTERNA DEL EQUIPO

Se pueden dividir en dos grandes grupos los tipos de PLCs, en lo que se refiere a sus configuraciones. Por un lado están los denominados de nivel alto, los cuales son prácticamente todos modulares. Esto quiere decir que se venden por separado la CPU, los módulos de entradas y salidas, la fuente de alimentación, etc. En este tipo de autómatas se va montando el equipo según las necesidades, pudiéndose incluso montar unos configurados como maestros y otros (que suelen ser más pequeños) como esclavos. Los primeros controlan el proceso general, mientras los esclavos se encargan de partes de proceso o máquinas individuales, mandando los resultados al Maestro.

Otro grupo son los denominados PLCs compactos, los cuales no necesitan módulos adicionales para su funcionamiento, al llevar incorporados en un mismo conjunto la CPU, la fuente de alimentación y un número limitado de entradas y salidas. Estos equipos están especialmente pensados para procesos pequeños o máquinas simples. De todas formas, aunque con limitaciones, también están disponibles para estos autómatas módulos de expansión y una gama reducida de periféricos.

El equipo a utilizar en esta práctica es del tipo compacto, es de la casa OMRON, el modelo es el CPM 10. Dentro de su configuración se encuentra:

- La fuente de alimentación, para su uso propio y además suministra 24 V. con una intensidad máxima de 300 mA., que está pensada para poder conectar algunas fotocélulas o sensores.
- Módulo de entradas. Tiene 6 entradas, todas son de potencial a (+/-) 24 V. de continua, se configura como un común (+ o -), que va a los detectores o sensores, y 6 retornos que devuelven la tensión aplicada por el común.
- Dos entradas analógicas conectadas a potenciómetros situados en el frontal del PLC.
- Módulo de salidas; Dispone de 4 salidas de tipo binario, estas son de las llamadas de libre potencial, o también se les denominan como salidas de tipo relé. Se conecta un común a cada uno de los relés, de forma que al activarse la salida se obtiene en la salida el potencial que se le está aplicando por el común. En el montaje se ha puesto como común la fase de la red.
- Dispone además (como la gran mayoría de los PLCs) de una vía de comunicación que es específica para conectarle el programador. Para el desarrollo de la práctica esta va conectada con el puerto serie del ordenador.

Tanto las entradas como las salidas llevan incorporado un diodo led para indicar que está activada. Además se aprecian en el frontal otros dos pilotos (diodos led) que indica el estado del PLC, modo Run o Stop.

La alimentación del equipo se realiza a 220 voltios. Todo ello se puede apreciar en el esquema de montaje que viene representado en la figura 5.

6. CONFIGURACIÓN INTERNA DEL EQUIPO

Como se ha indicado en apartados anteriores se puede programar las entradas de diversas maneras, para ello el equipo lleva una serie de registros que nos permiten dicha selección. Todos los PLCs poseen una serie de parámetros configurables, así como un mapa de memoria distribuido para el desarrollo del programa. Todo ello permite una serie de programaciones bastantes extensas que se salen de la intención

de esta práctica. Es por tanto que se desarrollarán en este apartado las configuraciones y distribuciones de memoria de forma muy básica, pero suficientes para el entendimiento de lo que se hace. Remitiéndose a los manuales del equipo para un conocimiento más completo del PLC.

6.1. MAPA DE MEMORIA

Se divide la memoria en varias zonas, pero hay que resaltar que si una de las zonas no se utiliza para su fin determinado, puede programarse para algún otro o como almacenamiento de programa. Para el PLC de que se utiliza en las prácticas, estas son:

Zona de entradas; está prefijada para ello desde la posición 000.00 a la posición 009.15, como se puede apreciar en la figura 2 las tres primeras cifras son las que se refieren a la posición de una palabra de 16 bits, mientras que las dos cifras siguientes representan la posición del bit dentro de cada palabra.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000																
001																
002																
003																
004																
005																
006																
007																
008																
009																

Figura 2

Zona de salida de forma análoga a la zona anterior se asigna desde la 010.00 a la 019.15. Tanto en esta zona como en la anterior, sólo son entradas/salidas físicas un número muy limitado de ellas, el resto son reservadas para equipos de mayor nivel. No obstante se pueden utilizar como bits de consulta y activación por programa, es lo que se entiende como relés internos de memoria.

Zonas de memoria especiales, entre ellas cabe distinguir las áreas de AR, SR, HR. Son para aplicaciones más complejas que se escapan del ámbito de esta práctica. No obstante solo indicar que se entienden como:

- SR; relés especiales que no cambian de estado al desconectar el autómata, es decir, que no se desactivan aunque se le quite la alimentación al PLC.

- AR; son posiciones de memoria que sirven para la conexión paralelo entre PLCs, cuando se programan en comunicación entre ellos.
- HR; se activan o desactivan desde las entradas especiales, tales como las interrupciones permitidas externas al PLC.

6.2. TEMPORIZADORES Y CONTADORES

Dispone el equipo que se utiliza en prácticas de una serie de temporizadores y contadores que se pueden programar por separado o concatenados para la acumulación tanto en tiempo como en contajes. Permite hasta un total de 127 temporizadores/contadores. Es decir que si el llamado como 0 se utiliza como temporizador **NO** se puede habilitar también como contador.

Para una mejor comprensión de lo expuesto se acompaña la un detalle de lo indicado en los párrafos anteriores, en un diagrama de bloques, ver figura 3.

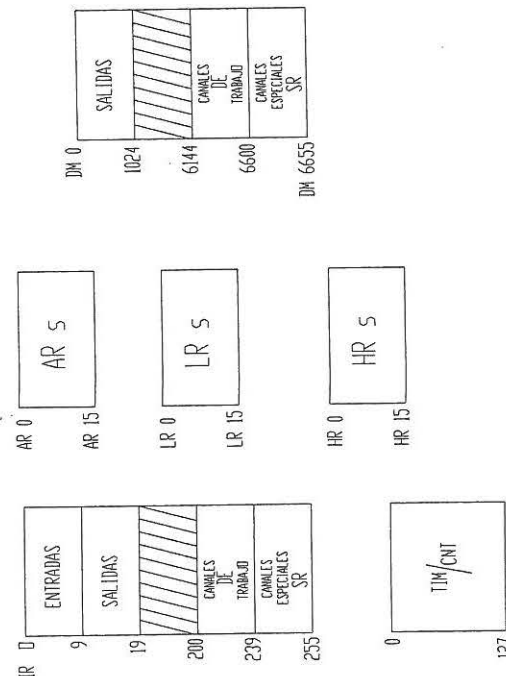


Figura 3

7. INSTRUCCIONES BÁSICAS

De nuevo aquí el hacer una exposición detallada de todas las instrucciones de programación sería muy extenso, quedando fuera del ámbito de esta práctica. Por tanto que se expondrán sólo las más básicas y necesarias para el desarrollo de los ejercicios propuestos, quedando el resto a disposición del alumno, en los manuales del equipo.

Por la simplicidad de las instrucciones que se aportan solo se da la representación gráfica de ellas en la figura 4.

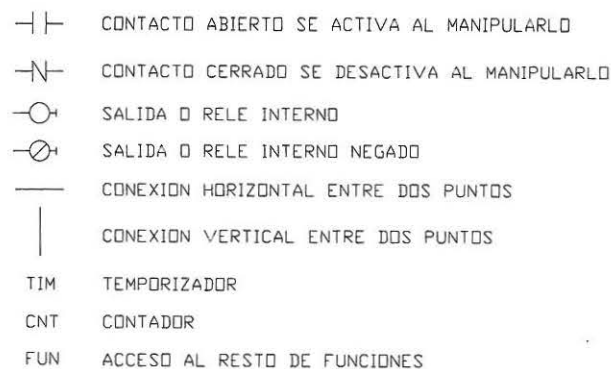


Figura 4

8. MONTAJE

Como ejemplo de montaje se adjunta el de un arrancador estrella-triángulo, en el que se detalla la conexión de las bobinas de los contactores y los interruptores de mando al autómata programable.

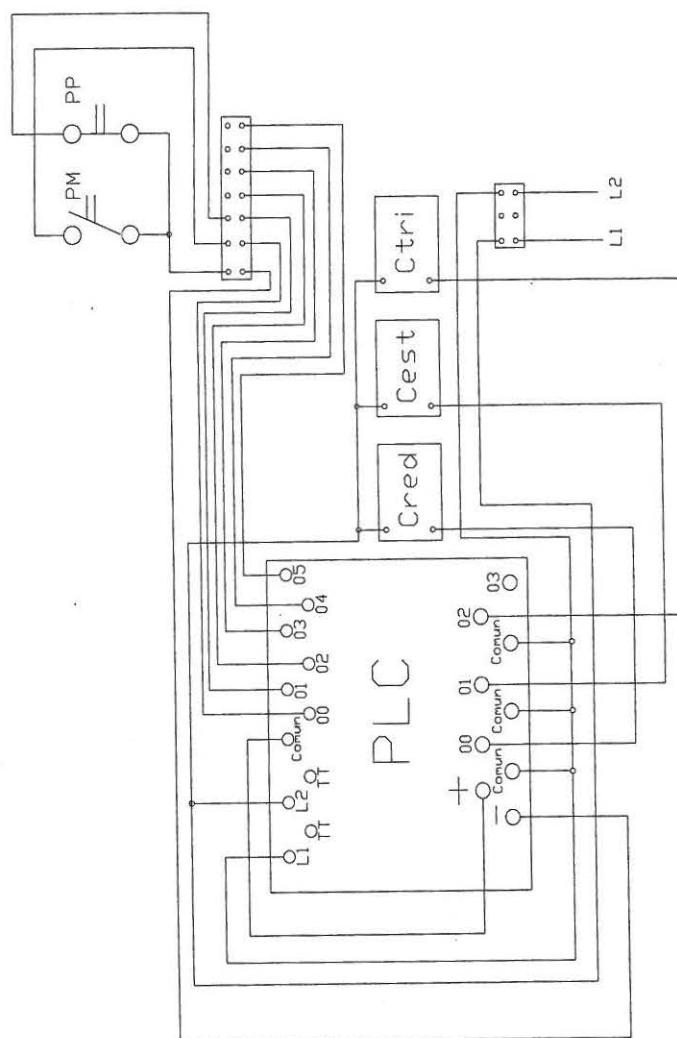


Figura 5

9. COMBINACIONES DIGITALES BÁSICAS

Los ejemplos prácticos a realizar se irán sucediendo de forma que vayan ganando en complejidad, partiendo del diseño e implementación de maniobras fáciles a bases de colocación de entradas y salidas en paralelo para la activación de una o varias salidas, para ir introduciendo funciones más o menos complejas, siempre como ya se ha indicado con el apoyo de ejemplo de tipo industrial.

Se piensa que un verdadero aprendizaje se realiza cuando el alumno realiza por sí solo el diseño de diversas maniobras, su implementación en el PLC, comprobación, corrección y puesta en marcha de los ejercicios diseñados. El simple hecho de copiar un ejercicio de la pizarra y programarlo, es por sí solo insuficiente, ya que se sabe que cuando realmente el alumno aprende es cuando tras varios intentos fallidos consigue poner en marcha un diseño más o menos complejo.

Por ello se piensa que todo lo que se hace en clase, no es nada más que la explicación de un gran número de herramientas a utilizar, que como es lógico si no se usan, rápidamente se olvidan.

Para la realización de los ejercicios se utiliza el PLC de OMRON el modelo CPM1A, que posee 6 entradas y 4 salidas. Ahora se exponen unos ejercicios elementales con los cuales se pretende aproximar al alumno con la forma de representarlos esquemas de contactos implementados con un PLC.

9.1. IDENTIFICADORES

Para trabajar con mayor comodidad y evitar así posibles errores de escritura, se empleará la siguiente tabla de identificadores. En ella se registrarán el nombre de las variables que se emplea tanto como ejecutor de la orden de una acción a tomar, así como la del actuador que ejecuta la orden, correlacionándolos siempre con el número de entrada y salida física del PLC.

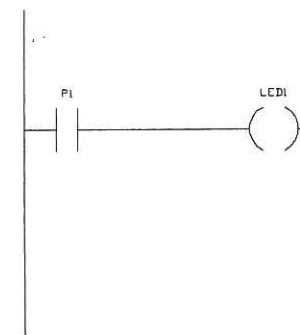
Bit	Identificador	Descripción
0.0	P1	Interruptor nº 1
0.1	P2	Interruptor nº 2
0.2	P3	Interruptor nº 3
0.3	P4	Interruptor nº 4
0.4	P5	Interruptor nº 5
0.5	P6	Interruptor nº 6
10.0	LED1	Diodo LED nº 1
10.1	LED2	Diodo LED nº 2
10.2	LED3	Diodo LED nº 3
10.3	LED4	Diodo LED nº 4
5.0	AUX1	Relé auxiliar nº 1
5.1	AUX2	Relé auxiliar nº 2
5.2	AUX3	Relé auxiliar nº 3
5.3	AUX4	Relé auxiliar nº 4
5.4	AUX5	Relé auxiliar nº 5
5.5	AUX6	Relé auxiliar nº 6
5.6	AUX7	Relé auxiliar nº 7
5.7	AUX8	Relé auxiliar nº 8

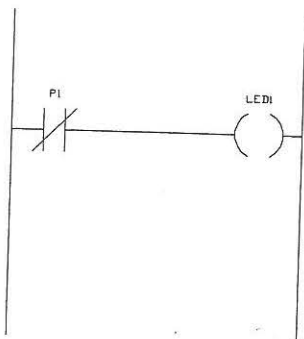
9.2. COMBINACIONES DIGITALES BÁSICAS

Conexión directa entrada - salida

La combinación digital más sencilla, que sirve de paso para comprobación de entradas y salidas digitales, es la de la figura.

El valor del LED1 sigue el estado de la entrada P1.



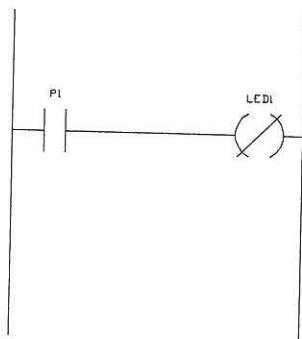


Consulta negada de una entrada

El estado de la señal de entrada puede consultarse de modo negado (ausencia de tensión).

En el ejemplo de la figura, el piloto se encenderá siempre que NO esté accionado el pulsador.

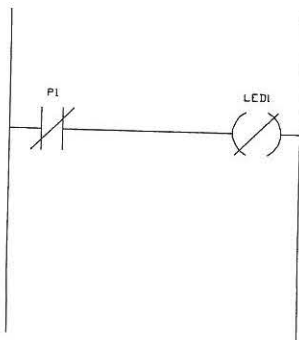
Un caso típico de consulta negada de una entrada es el de una seta de emergencia, con contacto normalmente cerrado.



Activación negada de la salida

El estado de la señal de entrada puede consultarse de modo negado (ausencia de tensión).

En el ejemplo de la figura, el piloto no se encenderá cuando se accione el pulsador.



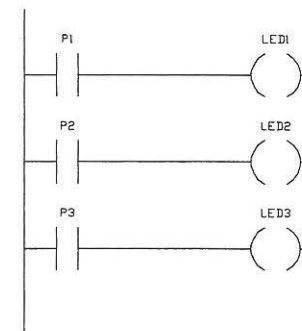
Consultas y activaciones negadas

Se pueden realizar a la vez consultas y activaciones en modo negado.

Conexión de varias líneas

En un programa se analizan de forma secuencial las distintas líneas que lo componen.

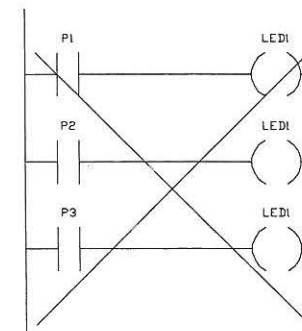
Los pilotos LED1, LED2 y LED3 siguen cada uno el estado de sus correspondientes pulsadores de entrada.



Conexión de varias líneas con la misma salida

NO se debe programar la misma salida en varias combinaciones digitales. SÓLO respondería a la última línea programada.

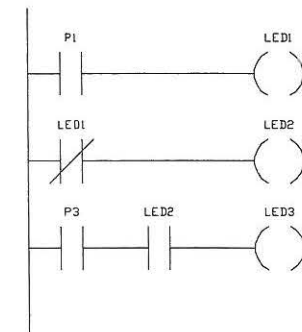
En este ejemplo, el LED1 sólo dependería del estado del pulsador P3. El resto de órdenes no afectan a su estado.

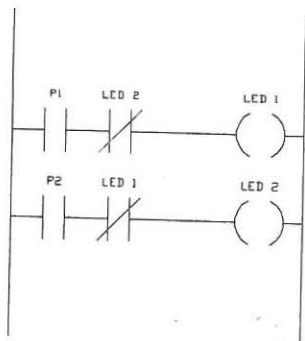


Consulta del estado de las salidas

Se puede consultar no sólo el estado de una entrada, sino también el de una salida, de forma normal o negada.

En este ejemplo, el estado de los pilotos LED2 y LED3 depende a su vez del estado de otras salidas.

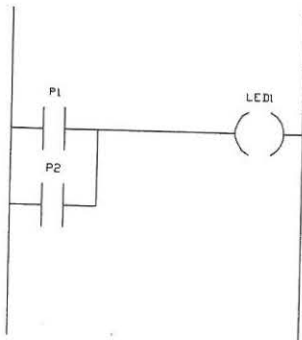




Bloqueo entre salidas

Si dos salidas NO deben estar nunca activas de forma simultánea (por ejemplo, la luz verde de dos semáforos en un cruce), se puede añadir una seguridad como la del ejemplo. No pueden estar nunca las dos salidas de forma simultánea.

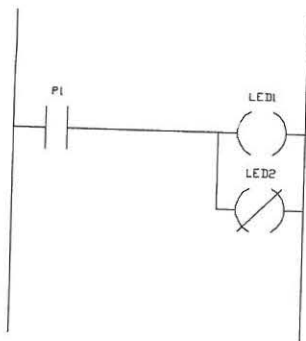
9.3. CONEXIONES EN PARALELO



Conexión de entradas en paralelo

Se puede consultar el estado simultáneo de varias entradas para activar una salida.

El piloto recibe tensión cuando está accionado el pulsador P1 o P2.



Conexión de salidas en paralelo

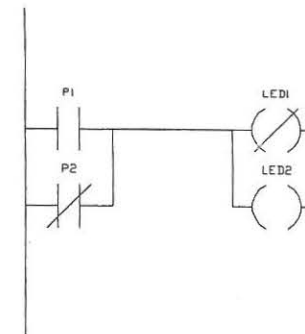
Se pueden activar simultáneamente varias salidas.

El piloto P1 sigue el estado del pulsador P1 y el piloto P2 lo hace de forma negada.

Conexión de entradas y salidas en paralelo

Se pueden realizar a la vez consultas de entradas en paralelo y accionar salidas en paralelo.

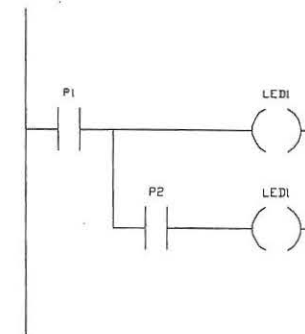
En este montaje cuando esté accionado el pulsador P1 o no lo esté el P2 se activará la salida LED2 y dejará de activarse la salida LED1.



Conexiones serie en cascada

Se pueden activar varias salidas con ramificaciones de una misma serie.

En el ejemplo de la figura, el piloto LED1 se encenderá cuando se activen el pulsador P1. El LED2 lo hará si está además pulsado el pulsador P2.

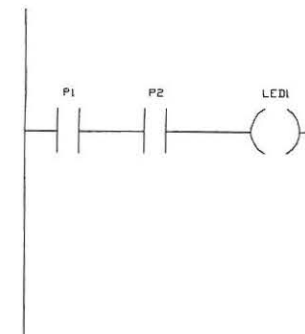


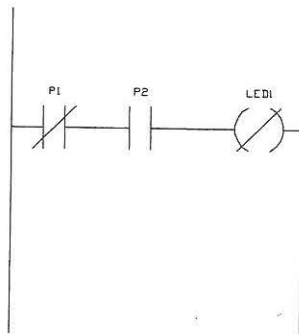
9.4. CONEXIONES EN SERIE

Conexión serie

En la conexión en serie la condición para que la salida tenga alimentación es que TODAS las entradas estén activas a la vez.

En el ejemplo de la figura, el piloto se encenderá cuando se activen a la vez los pulsadores P1 y P2.

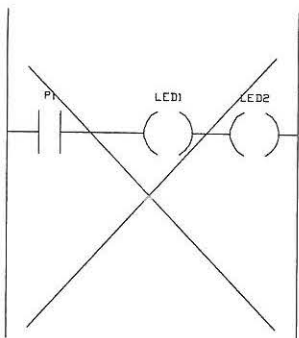




Conexión serie con entradas y salidas negadas

En los esquemas en paralelo se pueden consultar entradas y activar salidas en modo negado.

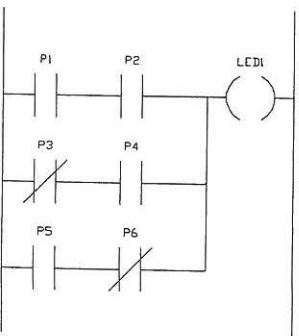
En el ejemplo de la figura, el piloto NO se encenderá cuando NO esté activado el pulsador P1 y sí lo esté el pulsador P2.



Conexión de salidas en serie

NO se pueden conectar en serie salidas.

9.5. CONEXIONES EN SERIE Y PARALELO

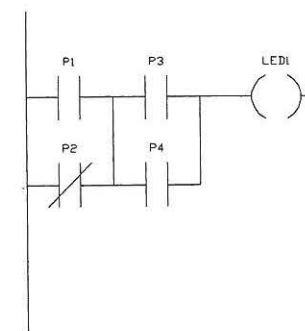


Conexión paralelo-serie

Se pueden conectar en paralelo varias combinaciones de tipo serie.

Conexión serie-paralelo

Se pueden conectar en serie varias combinaciones de tipo paralelo.

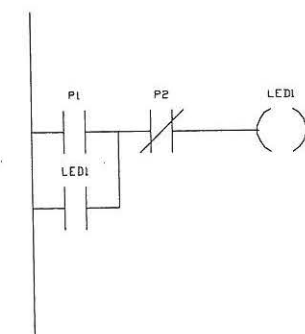


Autor retención

Alimentando la salida en paralelo con su propio estado se consigue mantenerlo aunque desaparezca la entrada del pulsador.

Para poder apagar la salida es necesario conectar en serie otra entrada de reset.

En el ejemplo, el LED1 se enciende con el pulsador P1, y permanece encendido, aunque desaparezca P1, hasta que se apaga con el pulsador P2.



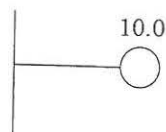
9.6. LIMITACIONES EN MONTAJE ELEMENTALES

Cada marca y/o modelo de PLC, tiene una serie de limitaciones a la hora de programar esquemas de tipo serie paralelo en lógica de relés.

Así, se conocen PLCs que so permiten encadenar en serie de más de 7 elementos (contactos abierto/cerrados), otros permiten algunos elementos más, pero tienen deficiencias a la hora de colocar en paralelo muchas ramas, etc. (Nos remitiremos en cada caso al manual de usuario del PLC empleado).

Pero de forma general hay una serie de maniobras que no se deben programar en la realización de diseños, como se ha indicado en el apartado anterior. Una de las más usuales es pretender activar de modo permanente una salida, por ejemplo un piloto indicador del funcionamiento del autómata.

NO es posible emplear una salida enganchada directamente, sin contacto u orden previa de activación.



Esto es solucionable en la práctica totalidad de PLCs y de varias formas, unas con la ayuda de funciones que siempre actúen de tipo SET, o similares, otra forma de solventarlo en otros PLC, es con el uso de bits especiales, caso de la firma OMRON, etc.

Como ejemplo industrial se puede necesitar este tipo de configuraciones al intentar señalizar que se un PLC está alimentado y en modo de trabajo (RUN), con la ayuda de un piloto en la puerta de un cuadro eléctrico. Ver en la figura 4 una posible solución en la cual se ejecuta esta operación sin necesidad de utilizar ninguna entrada física del PLC.

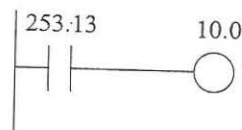
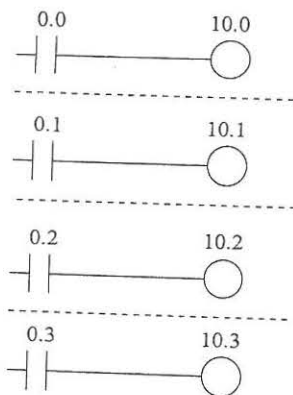


Figura 6. Señalización de PLC en modo RUN

9.7. COMPROBACIÓN DE ENTRADA Y DE SALIDAS



Este montaje simple permite comprobar, en un cuadro de maniobra, el correcto cableado de las entradas y salidas de un PLC. Para ello se puede hacer un ejercicio de correspondencia entre entradas y salidas.

Se diseña por tanto que:

La entrada 1 al activarse active a la salida 1,

La entrada 2 activará a la salida 2.

Así sucesivamente hasta completar todas las salidas.

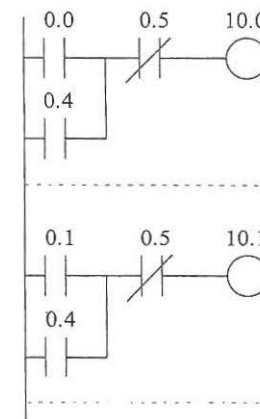
9.8. PRUEBA DE PILOTOS Y APAGADO GENERAL

Un caso práctico muy común es la presencia de un panel de visualización con diferentes pilotos que visualizan el estado lógico de señales de control (bomba en marcha, etc.). En este caso es necesario disponer de una orden de encendido general, que permita comprobar si algún piloto está fundido. Asimismo es conveniente una señal de apagado general de los pilotos. Se realiza el mismo ejercicio anterior con la siguiente variación:

Las 4 primeras entradas accionan las 4 primeras salidas de forma directa.

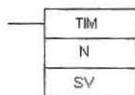
La entrada nº 5 activa todas las salidas.

La entrada nº 6 anula todas las salidas. Esta orden tiene preferencia sobre la anterior.



10. TEMPORIZADORES

Existen marcas de PLCs que poseen una gran batería de modelos de temporizadores programables. Por el contrario otros sólo disponen de un modelo de temporizador programable, lo cual obliga a interconectarlo con otros elementos para conseguir otras formas de funcionamiento. Por otro lado la gran mayoría de PLCs disponen de bits especiales que suelen ser biestables internos o temporizadores de diversas escalas de tiempos. Dado la gran importancia que tiene el control del tiempo y la necesidad de generar diversas formas de temporizar, se dedican varios ejemplos de temporización.

TIM - Temporizador**Descripción:**

Un temporizador se activa cuando su condición de ejecución es ON, y se resetea (a SV) cuando la condición de ejecución se pone en OFF. Una vez activado, TIM mide en unidades de 0,1 segundo desde el SV.

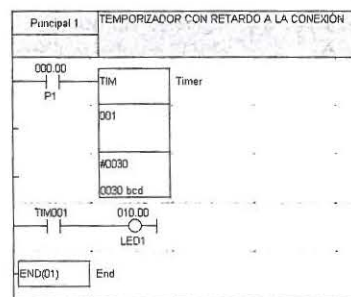
Si la condición de ejecución permanece en ON lo suficiente para que transcurra el tiempo fijado en TIM, se pondrá a ON el indicador de finalización del número de TC utilizado y permanecerá en dicho estado hasta que se resetee TIM (es decir, hasta que su condición de ejecución se ponga en OFF).

Rangos:

N: Número de TC 000 - 511
SV: Valor seleccionado (canal, BCD)
IR, AR, DM, HR, #

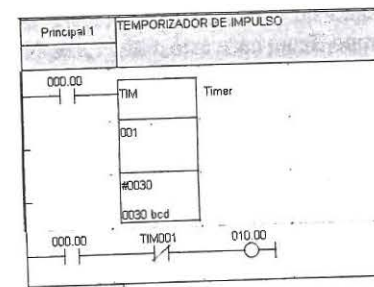
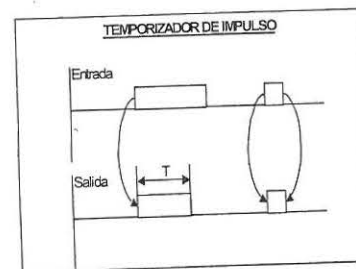
10.1. RETARDO A LA CONEXIÓN

Se activa la salida al cabo de un tiempo de conectar la entrada, permaneciendo la salida el tiempo que esté activa la entrada. Sirva como aplicación la conexión del contactor de triángulo en un arranque estrella triángulo. Otra aplicación puede ser la del retardo de la conexión de una alarma, es decir se pone en marcha un sistema de vigilancia, pero permite un tiempo de desconexión hasta que la persona que la conecta sale del local, a partir de entonces es cuando se queda el sistema vigilante. En el ejemplo se programa el temporizador 1 para que encienda el LED 1 3 segundos después de activarse la entrada P1. Cuando se anula la entrada el temporizador se resetea.

**10.2. TEMPORIZADOR DE IMPULSO**

Se activa en el instante de activar la entrada y se desconecta al cabo de un tiempo si y sólo si la entrada sigue activa.

Como aplicación típica de este temporizador se puede indicar la luz de la escalera, otro ejemplo en el cual se podría programar este tipo de temporizador es el disparo de una alarma, el detector se activa un momento y la alarma suena durante un tiempo más o menos largo.

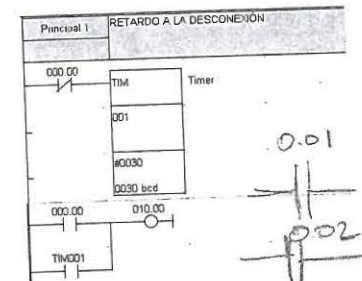


En el ejemplo se programa el temporizador 1 para que encienda el LED 1 durante 3 segundos después de activarse la entrada P1. Cuando se anula la entrada el temporizador se resetea.

10.3. RETARDO A LA DESCONEXIÓN

Se activa en el instante de activar la entrada y se desconecta al cabo de un tiempo después de desaparecer la entrada.

Este tipo de temporizador es el adecuado para arrancar un motor durante el transitorio de estrella. Otro ejemplo práctico de este tipo de temporizador puede ser la desconexión de una sirena de alarma, tras haberse disparado esta ante cualquier anomalía.



En el ejemplo se programa el temporizador 1 para que encienda el LED 1 cuando se activa la entrada P1, permaneciendo encendido durante 3 segundos más.

11. INTERMITENCIAS

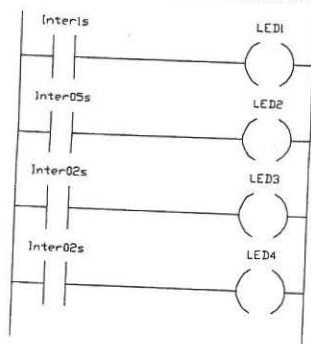
Generalmente los PLCs disponen de unos osciladores internos que permiten con una adecuada utilización generar temporizadores, intermitentes y osciladores, cuyas bases de tiempo serán función del bit que se elija.

Tal como se ha especificado con anterioridad, para la realización de las prácticas se utiliza en PLC de OMRON modelo CPM1A. En este autómata se dispone de varios bits especiales, donde algunos de ellos son del tipo oscilador, el funcionamiento de ellos es idéntico, solo los diferencian la base de tiempo, la mitad del tiempo de su base está a nivel alto (ON) y la otra mitad a nivel bajo (OFF). Veamos la tabla siguiente, en la cual aparecen algunos bits del sistema.

Bit del sistema	Identificador	Descripción
253.13	ON	Siempre a uno
253.14	OFF	Siempre a cero
253.15	Inicio	A uno en el primer ciclo del programa
254.0	Inter1m	Intermitente de 1 minuto (30 s ON- 30 s OFF)
255.2	Inter1s	Intermitente de 1 segundo (0'5 s ON- 0'5 s OFF)
255.1	Inter0.2s	Intermitente de 0'2 segundos
255.0	Inter0.1s	Intermitente de 0'1 segundos
254.1	Inter0.02s	Intermitente de 0'02 segundos

Como ejemplos a implementar con este tipo de señales tendremos:

Visualizar intermitencias



BITS del sistema

Se puede consultar el estado de bits particulares del sistema. Para ello se introduce la siguiente tabla de identificadores.

Los LED2 1 A 4 muestran el estado de los distintos intermitentes del sistema. Se suelen gastar como señales de intermitencias, o bien como se verá más adelante combinándolos con contadores, se pueden obtener temporizadores incluso de tiempos muy largos.

12. CONTADORES

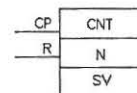
Un segundo elemento clásico en las automatizaciones es el contador. El uso general de este tipo de herramienta es contar una serie de impulsos generados desde alguna de las entradas o bien de cualquier bit de los denominados especiales.

Generalmente es los PLCs, existe la posibilidad de programar dos tipos de contador, uno de conteo en un solo sentido (que suele ir descontando desde el valor prefijado), que al finalizar la cuenta pone a nivel alto su salida. Otro tipo de contador usual de que se dispone, es que puede contar en dos sentidos (contador/descontador), al llegar al valor prefijado pondrá su salida a nivel alto, como en el caso anterior.

Otra aplicación típica de estos elementos es su utilización como temporizadores de secuencias de tiempo largas. Imaginemos que se quiere contar un tiempo de varias horas, para su implementación se necesitaría ve muchos temporizadores en cascada. Sin embargo si se pone uno de los bits especiales (ejemplo el de 1 minuto) como entrada de un contador, y este se programa con un conteo de 360 pulsos, se conseguiría de forma fácil un temporizador de 3 horas. Véase unos ejemplos con contadores.

12.1. CONTADOR EN UN SOLO SENTIDO

CNT - Contador



Descripción:

CNT se utiliza para descontar a partir de SV cuando la condición de ejecución en el impulso de conteo, CP, pase de OFF a ON, es decir el valor presente (PV) será reducido en uno, siempre que CNT se ejecute con una condición de ejecución ON para CP y la condición de ejecución fuera OFF para la última ejecución. Si la condición de ejecución no cambió o cambió de ON a OFF, el PV de CNT no cambiará. El indicador de finalización para un contador se pone a ON cuando el PV alcanza cero y permanecerá en ON hasta que el contador se resetee.

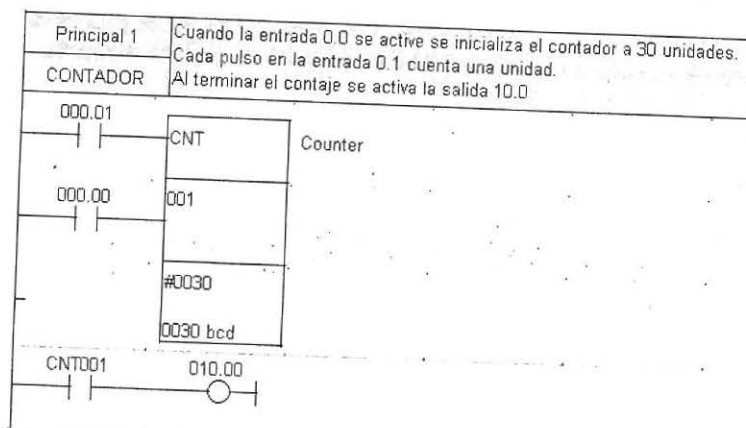
CNT se resetea con una entrada de reset, R. Cuando R pasa de OFF a ON, el PV se resetea a SV. El contador no descuenta mientras la entrada R está en ON. El PV para CNT no se resetea en secciones de programa enclavadas o por cortes de alimentación.

Rangos:

N: Número de TC 000 - 511
SV: Valor seleccionado (CH, BCD)
IR, AR, DM, HR, #

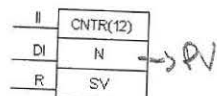
Un ejemplo industrial de este tipo, podría ser el conteo de piezas hasta formar un lote (un lote puede ser el llenado de una caja, etc.). Otro ejemplo sería los aparatos de contar coches que utiliza tráfico, para obtener la densidad de circulación en

una vía determinada. Se programará como ejemplo básico la cuenta de pulsos desde una entrada del PLC.



12.2. CONTADOR EN AMBOS SENTIDOS

CNTR(12) - Contador Reversible



Descripción:

CNTR(12) es un contador reversible, es decir se utiliza para contar entre cero y SV de acuerdo con los cambios de dos condiciones de ejecución, la entrada de conteo adelante (DI) y la entrada de conteo atrás (DI).

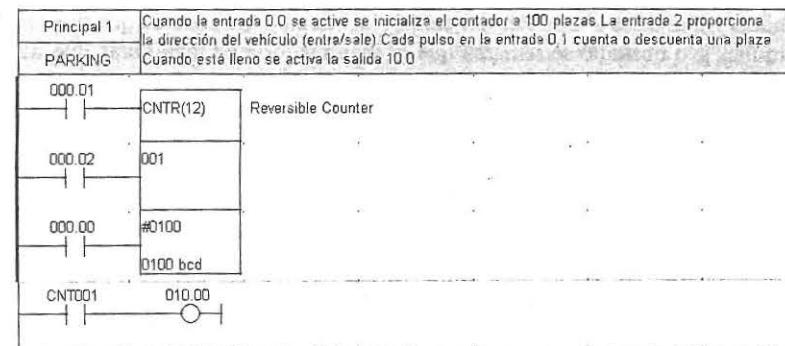
El valor presente (PV) aumentará en uno cuando se ejecute CNTR(12) con una condición de ejecución ON para DI y la última condición de ejecución para DI sea OFF. El valor presente (PV) disminuirá en uno siempre que se ejecute CNTR(12) con una condición de ejecución ON para DI y la última condición de ejecución para DI sea OFF. Si la transición de OFF a ON se produce en ambas entradas DI y DI, el PV no cambiará.

Si las condiciones de ejecución no han cambiado o lo han hecho de ON a OFF para ambas DI y DI, el PV de CNT no cambiará.

En conteo descendente, de 0000 pasa al SV y se pone a ON el indicador de finalización hasta que el PV se descuenta de nuevo. En conteo ascendente, del SV se pasa a un PV 0000 y el indicador de finalización se pone a ON hasta que se incremente de nuevo el PV.

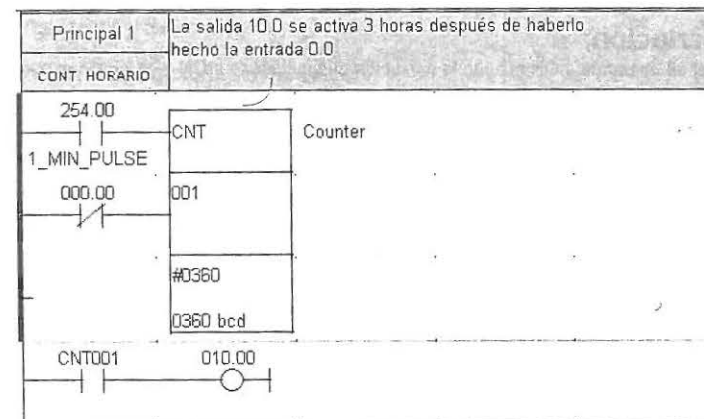
CNTR(12) se resetea con la entrada de reset, R. Cuando R pasa de OFF a ON, el PV se resetea a cero. No se realiza ningún conteo mientras la entrada de reset está en ON. El PV de CNTR(12) no se reseteará en secciones de programa enclavadas o por cortes de alimentación.

Se puede colocar varios ejemplos donde se utilice este tipo de contador, uno sería el control de personas dentro de un local en un momento determinado, cada persona que entra incrementa el número, cada persona que sale lo decremента, en cualquier momento revisando el estado del contador se puede saber la cantidad de personas que hay. Otro ejemplo típico sería el control de un parking, este avisaría de si existen plazas libres o no. Veamos un ejemplo básico de un contador/descontador, con un parking de 100 plazas.



12.3. EL CONTADOR COMO TEMPORIZADOR

Si se quisiera programar un riego que durase varias horas, se haría de forma muy fácil con este tipo de secuencias. Se puede colocar como ejercicio para programar el activar una salida al cabo de varios minutos. En este ejemplo se programa un contador para que dé una señal a las 3 horas de activar una entrada, empleando un intermitente de 1 minuto del sistema.



13. FUNCIONES BÁSICAS DE ACTIVACIÓN Y DESACTIVACIÓN DE BITS

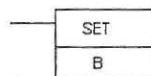
Además de disponer de los elementos vistos hasta ahora para realizar el diseño de distintas maniobras, la práctica totalidad de PLCs poseen una serie de funciones que permiten agilizar y esclarecer los esquemas de maniobra habituales empleados en la mayoría de automatismos. En este apartado solo se van a recoger una parte de las funciones de que disponen los PLCs. Si bien se quiere resaltar que existe un gran abanico de herramientas/funciones que se puede hacer uso en el diseño con autómatas y que dado al carácter del curso y del tiempo limitado de que dispone, no se estudian. No obstante se remarca que la gran mayoría de maniobras que funcionan en la industria, se puede llevar a cabo con la materia tratada en este curso.

13.1. ACTIVACIÓN/DEACTIVACIÓN DE UNA SALIDA (SET/RESET)

Estas son dos funciones que van prácticamente siempre ligadas, ya que no se entiende que una salida (o un bit cualquiera) se active y no exista ningún resorte para desactivarla. Se utiliza para poner en marcha y parar una salida bajo ciertas condiciones. Veamos un ejemplo.

Se pueden programar varios SET para una misma salida y que se programe una sola función de RSET.

SET - Set



Descripción:

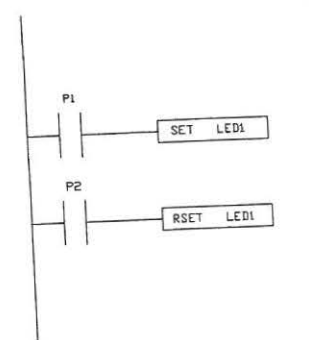
Pone el bit operando a ON cuando la condición de ejecución es ON y no afecta al estado del operando cuando la condición es OFF.

Rangos:

B: Bit IR, AR, HR, LR.

Se puede conseguir el efecto de auto retención mediante el empleo de las funciones SET y RESET del sistema.

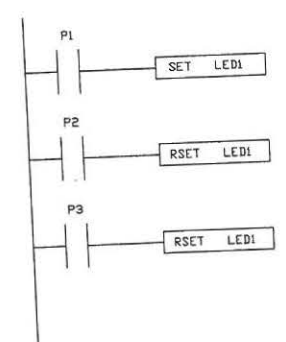
En este ejemplo, el pulsador P1 enciende el LED1 y el pulsador P2 lo apaga. ¿Qué ocurre si ambos pulsadores están accionados a la vez?



Funciones Set y Reset múltiples

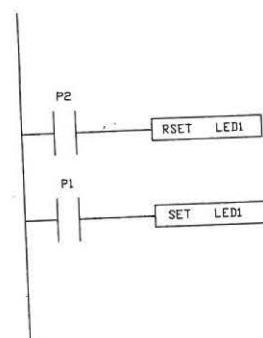
Se pueden programar diversas líneas de SET o RSET sobre una misma salida en distintos puntos del programa. El efecto sería comparable al de una combinación en paralelo de las distintas condiciones programadas.

En este ejemplo, el pulsador P1 enciende el LED1 y el pulsador P2 o el P3 lo apagan.



Funciones Reset y Set

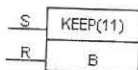
Si se programa primero la función RSET, y están activados ambos pulsadores, el reset no se produce.



13.2. INSTRUCCIÓN SET-RESET CONJUNTA (KEEP)

Con esta función se agrupa en una sola las dos funciones anteriores, la única diferencia es que aquí solo existe un SET asociado a cada RESET.

KEEP(11) - Relé de Enclavamiento

**Descripción:**

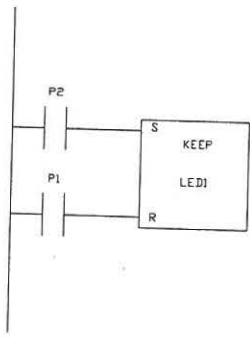
Define un bit (B) como de enclavamiento, activado por una entrada de set (S) y desactivado por una entrada de reset (R).

Rangos:

B: Bit IR, AR, HR, LR.

Función KEEP

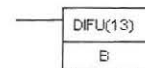
La función KEEP, o biestable, implementa en un solo paso el efecto de las funciones SET y RSET.



13.3. INSTRUCCIÓN DETECCIÓN DE FLANCO DE SUBIDA (DIFU)

Cuando se ejecuta esta instrucción lo que se pretende es que una señal de entrada, quede constituida en un pulso de un ciclo de scan (es decir el tiempo que dure un ciclo de programa). Esta función tiene un importancia vital, ya sirve para la eliminación del efecto rebote de los elementos de conexión.

DIFU(13) - Detección de Flancos

**Descripción:**

DIFU(13) pone en ON el bit designado durante un solo ciclo de scan, cuando la condición precedente pasa de OFF a ON.

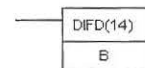
Rangos:

B: Bit IR, AR, HR, LR.

13.4. INSTRUCCIÓN DETECCIÓN DE FLANCO DE BAJADA (DIFD)

El comportamiento y uso de esta señal es prácticamente igual al anterior, pero cuando se pretende detectar un flanco de bajada. Se compruebe de forma similar al caso anterior, sólo se cambia el flanco empleado.

DIFD(14) - Detección de Flancos

**Descripción:**

DIFD(14) pone en ON el bit designado durante un solo ciclo de scan, cuando la condición precedente pasa de ON a OFF.

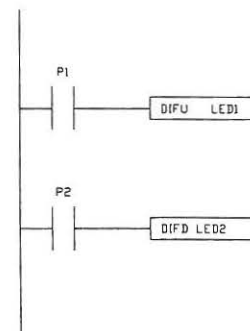
Rangos:

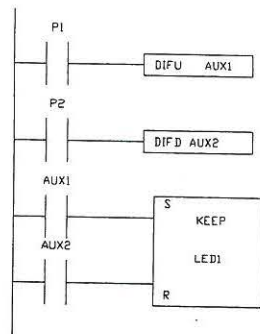
B: Bit IR, AR, HR, LR.

Cambios de estado

En este ejemplo, el LED1 se enciende cuando la entrada P1 pasa de 0 a 1, y el LED2 cuando su entrada P2 pasa de 1 a 0.

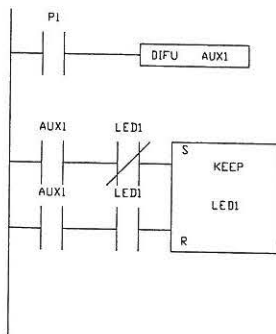
¿Qué se observa al realizar el ejemplo? Ningún LED se enciende, porque el cambio de estado sólo dura un ciclo, y es demasiado rápido para poder observarlo.





Cambios de estado con retención

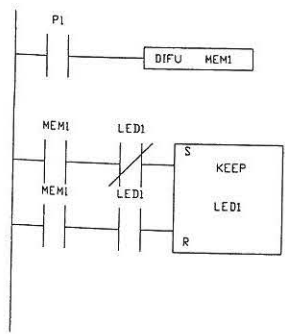
En este ejemplo, el LED1 se enciende cuando la entrada P1 pasa de 0 a 1, y se apaga cuando la entrada P2 pasa de 1 a 0.



Salida biestable

Se desea pilotar la salida LED1 mediante el pulsador P1, de modo que en cada activación del mismo (paso de 0 $\rightarrow$ 1) la salida cambie de estado.

Cada vez que el pulsador P1 pasa de 0 a 1 el estado el LED 1 cambia.



Salida biestable con memoria

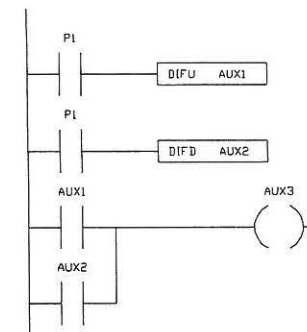
Si se quiere que la salida LED1 se memorice frente a una reconexión del autómata, basta con sustituir el relé auxiliar AUX1 por otro con memoria.

Consulta de flanco positivo o negativo

En ocasiones interesa consultar cualquier cambio de estado (0 $\rightarrow$ 1 ó 1 $\rightarrow$ 0).

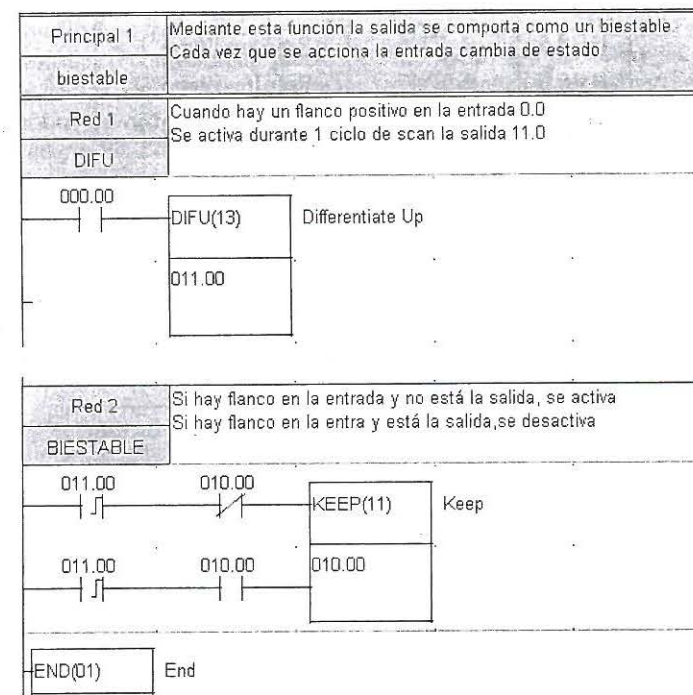
Hay que hacer en este caso la consulta en paralelo a dos funciones DIFU y DIFD.

En el ejemplo el relé auxiliar AUX3 se activa en cada cambio del estado del pulsador P1.



13.5. BIESTABLE MEDIANTE UNA ÚNICA ENTRADA DE MANDO

Veamos un ejemplo conjunto de estas funciones, de uso común en entornos industriales: una salida que cambia de estado cada vez que se acciona un pulsador.



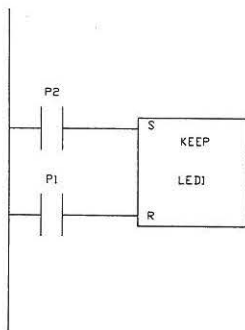
13.6. MEMORIZACIÓN DEL ESTADO DE UNA SEÑAL CON CAÍDAS DE TENSIÓN

¿Qué ocurre cuando se desconecta el autómata, bien por avería bien por apagado del equipo?. Al arrancar de nuevo, todas las salidas se ponen a 0, incluso aquellas que estaban a 1 por una función SET o KEEP.

Sin embargo, existen salidas en el autómata cuyo estado se “recuerda” después de una desconexión, gracias a una pila interna. Son los relés con memoria, HR00 a HR19.

Emplearemos los siguientes relés con memoria:

Bit	Identificador	Descripción
HR0.0	MEM1	Relé auxiliar nº 1
HR0.1	MEM2	Relé auxiliar nº 2
HR0.2	MEM3	Relé auxiliar nº 3
HR0.3	MEM4	Relé auxiliar nº 4
HR0.4	MEM5	Relé auxiliar nº 5
HR0.5	MEM6	Relé auxiliar nº 6
HR0.6	MEM7	Relé auxiliar nº 7
HR0.7	MEM8	Relé auxiliar nº 8

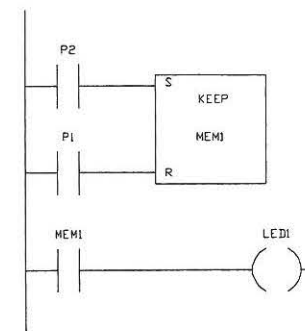


Función KEEP sin memoria

Activar el LED1 con el pulsador de SET. Apagar y encender el autómata. El LED1, al reconectar, está apagado.

Función KEEP con memoria

Activar el LED1 con el pulsador de SET. Apagar y encender el autómata. El LED1, al reconectar, sigue encendido.



14. COMPARACIONES

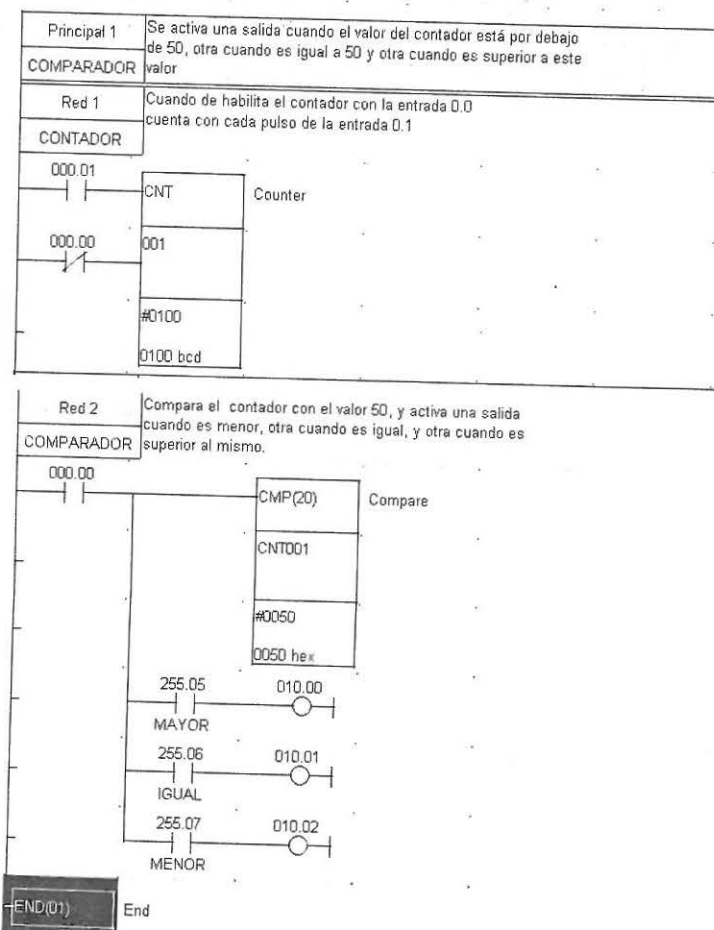
Una de la potencialidad de los PLCs es el poder almacenar en memoria muchos datos, estos nos pueden determinar una serie de situaciones del estado de la máquina o proceso a controlar. Enumeremos algunos casos, en una máquina herramienta se pueden programar los diversos posicionamientos de la máquina, lo que permite memorizar trabajos de precisión y complejos, que irá tomando la máquina para desarrollar o confeccionar piezas en serie. Otro caso puede ser las cantidades de pesaje de una mezcladora de harinas u otro ingrediente, lo cual nos permite automatizar una fábrica de piensos, laboratorio de todo tipo, etc. Se necesita en estos caso acceder a las posiciones de memoria y grabar en ella una serie de datos que previamente se han calculado o experimentado, para posteriormente ir comparando los lectores de posición, pesaje, etc., se irá comparando y tomando decisiones con el resultado.

Todos los PLCs, disponen de la función de comparación, disponen así mismo unos bits asociados al resultado de la comparación, que nos permite la toma de decisiones. Para el autómata de la práctica, se tiene:

Indicador	Dirección	C1 < C2	C1 = C2	C1 > C2
GR	255.05	OFF	OFF	ON
EQ	255.06	OFF	ON	OFF
LE	255.07	ON	OFF	OFF

Por lo tanto consultando el estado de estos tres bits especiales se puede saber el resultado de la comparación.

Es importante decir que la instrucción de comparación (CMP), se ejecuta entre dos posiciones o palabras de memoria (16 bits), entendiéndose que la posiciones de memoria están referidas a toda la distribución del PLC, es decir, contempla la memoria de datos, la zona de memoria reservada a las entradas y a las de las salidas, memorias de seguridad (lactes internos), etc. Véase en los manuales específicos de cada autómata las distribuciones que poseen. En el que se emplea en práctica cabe distinguirse las siguientes distribuciones o zonas de memoria; IR (entradas y Salidas), SR (registros especiales), DM (memoria de datos), HR (memoria no volátil), TC (memoria reservada a temporizadores y contadores), LR (registros de comunicación especial con otros PLCs de la misma marca).



15. PROGRAMACIÓN DE MANIOBRAS INDUSTRIALES

Una vez entendido el proceso de introducción de los datos en la memoria y el de comparación, se pasa a ejecutar ejemplos de tipo industrial con las funciones expuestas.

Se indicarán en los primeros ejercicios las instrucciones en dos formatos: diagrama de contactos y mnemónico, con el fin de mostrar la equivalencia entre los mismos. Sin embargo, en el resto de ejercicios se empleará únicamente el sistema de lógica de contactos para la representación del programa, al ser la más extendida.

Cuando se pretende resolver un problema de automatización industrial, bien mediante esquemas convencionales a base de contactores y relés, bien como se hace hoy en día, con la ayuda de los autómatas programables, facilita la resolución del problema adoptando métodos de trabajo organizados, y que el proyectista esté familiarizado con ellos. Cualquier método empleado, si es dominado por el operario es válido. No obstante suele ser habitual seguir los siguientes pasos:

- Estudio completo de los requerimientos de la maniobra o automatización.
- Asignación de las variables de entrada y salida, adaptándolas al autómata.
- Etiquetar o nombrar de forma lógica las distintas variables.
- Planteamiento de un posible cronograma.
- Definición de las señales que actúan en la activación o parada de cada una de las salidas.
- Conocer o estudiar las distintas funciones programables en el autómata elegido.
- Desarrollar el programa de forma sencilla. Desglosando el programa en diversas partes, para evitar secuencias muy largas y/o repetitivas.
- Describir lo más claramente las distintas partes del programa, anotando todos los comentarios necesarios.
- Una vez acabado el programa, si se dispone de simulador, se procederá a la simulación.
- Se transfiere el programa al autómata y se pone en funcionamiento, procurando a ser posible que el proceso controlado no efectúe maniobras peligrosas.
- Se ajusta y corrigen las deficiencias encontradas y se procede a la puesta en marcha definitiva.

Algunas normas útiles para programar:

- No se deben usar dos veces seguidas una misma salida dentro de una misma secuencia de programa. Siempre prevalecerá la última modificación o actuación sobre ella.
- No interesa colgar de una misma entrada muchas ramificaciones. Es más práctico utilizar una misma entrada o posición de memoria repetidas veces.
- Para maniobras repetitivas es conveniente utilizar subrutina.
- El programa debe estructurarse en bloques lógicos que reflejen en la medida de lo posible la instalación.
- No se pueden colocar funciones programables directamente conectadas a la columna de la izquierda.
- No se puede acabar una línea de programación hacia la derecha sin conectar una salida física o posición interna de memoria utilizada como salida u otra función específica.

Naturalmente no todos los pasos son necesarios en todas las automatizaciones efectuadas. La experiencia del programador va descartando o añadiendo distintos pasos en cada proyecto.

El programa con este autómata (así como con otros de otras marcas) se tiene que estructurar formando redes. Cada red queda formada por los elementos que estén interconectados entre sí. Dos elementos o líneas de programación que no tenga ninguna vinculación deben programarse en redes distintas. Para poder pasar a la programación de una red es necesario terminar la red anterior.

El programa debe finalizar con la instrucción propia de finalización (END).

Se emplearán en los ejemplos los siguientes identificadores para las señales de entrada y de salida.

Bit	Identificador
0.0	ENT1
0.1	ENT2
0.2	ENT3
0.3	ENT4
0.4	ENT5
0.5	ENT6
10.0	SAL1
10.1	SAL2
10.2	SAL3
10.3	SAL4

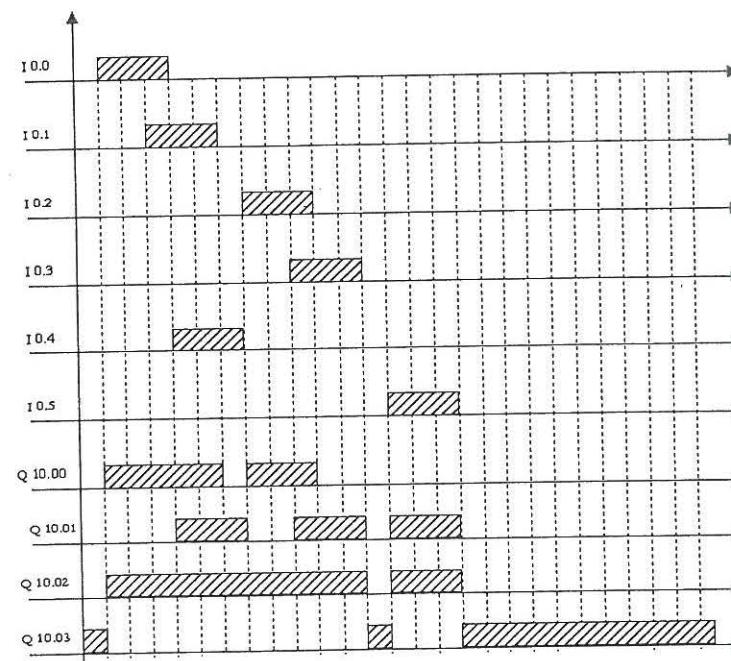
15.1. COMBINACIONES LÓGICAS DE ENTRADAS/SALIDAS

1. Objetivo

Con este ejercicio se pretende introducir al alumno en la combinación de las señales de entrada y salida, bien formando circuitos en serie o en paralelo. Además debe servir para familiarizarse con dos de los lenguajes de programación de automatismos industriales, que son los más usuales en los autómatas programables, tales lenguajes son el de contactos y el de mnemónicos.

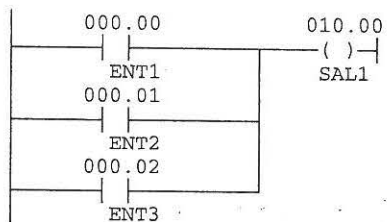
2. Planteamiento

Con las entradas ENT1, ó ENT2, ó ENT3 se activa la SAL1. Con las entradas ENT4, ó ENT5, ó ENT6 se activa la SAL2. Si la SAL1 y la SAL2 están activas se activa la SAL3. Si no están activadas ni la SAL1 ni la SAL2 ni la SAL3, se activará la SAL4. El cronograma que representa la combinatoria descrita es el siguiente:



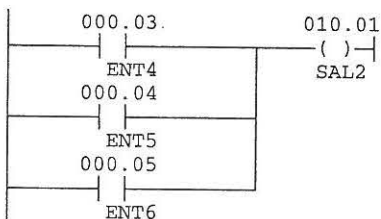
3. Programación

Red 1.- La entrada ENT1, ENT2 o ENT3 activan la salida SAL1.



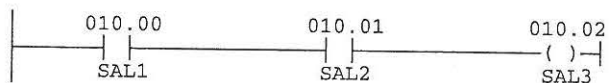
00000	LD	000.00	ENT1
00001	OR	000.01	ENT2
00002	OR	000.02	ENT3
00003	OUT	010.00	SAL1

Red 2.- La entrada ENT4, ENT5 O ENT6 activan la salida SAL2.



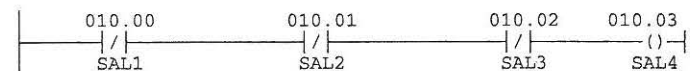
00004	LD	000.03	ENT4
00005	OR	000.04	ENT5
00006	OR	000.05	ENT6
00007	OUT	010.01	SAL2

Red 3.- Si está activa la salida SAL1 o la SAL2 se activa la salida SAL3.



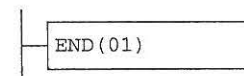
00008	LD	010.00	SAL1
00009	AND	010.01	SAL2
00010	OUT	010.02	SAL3

Red 4.- Si no está activa la salida SAL1 ni la SAL2 se activa la salida SAL4.



00011	LD NOT	010.00	SAL1
00012	AND NOT	010.01	SAL2
00013	AND NOT	010.02	SAL3
00014	OUT	010.03	SAL4

Red 5.- Fin del programa.



00015	END
-------	-----

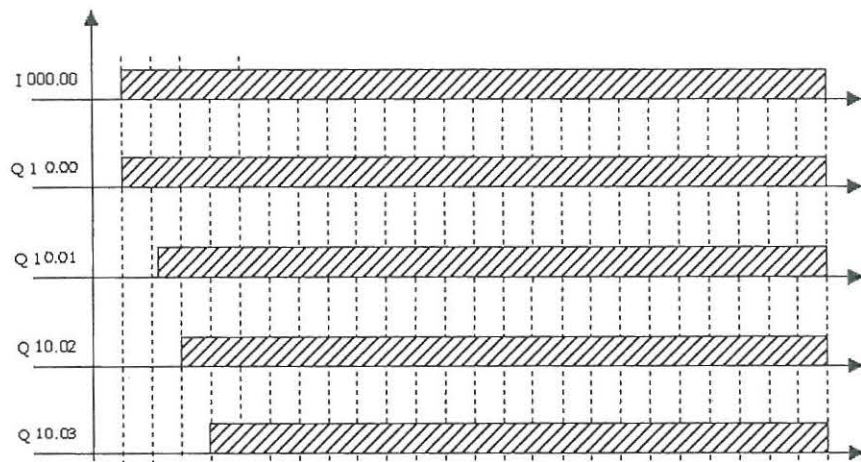
15.2. COMBINACIÓN ENTRADAS/SALIDAS Y BITS DEL SISTEMA

1. Objetivo

Con este ejercicio se pretende que el alumno además de practicar la combinación de las señales de entrada, que se familiarice y conozca alguno de los bits del sistema importantes que este equipo y otros disponen para facilitar los procesos de programación.

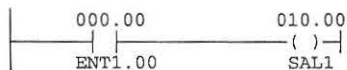
2. Planteamiento

Con la primera entrada (000.00) se activa la salida primera (010.00) y siempre que se mantenga esta salida activa se irán conectando consecutivamente las distintas salidas con una cadencia de un segundo de una a otra. El cronograma sería:



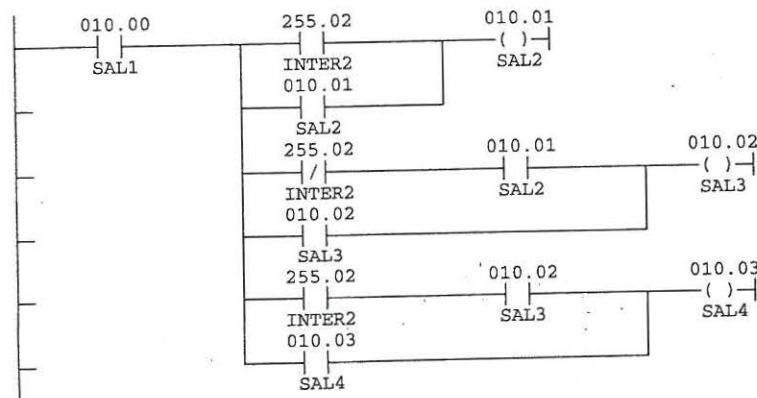
3. Programación

Red 1.- La entrada ENT1 activa la salida SAL1.



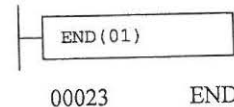
Paso de programa	Instrucción	Dirección	n°E/S
00000	LD	000.00	ENT1.00
00001	OUT	010.00	SAL1

Red 2.- Con salida SAL1 se activa cada segundo una nueva salida siguiente.



Paso de programa	Instrucción	Dirección	n° E/S
00002	LD	010.00	SAL1
00003	OUT	TR0	TR0
00004	AND	255.02	INTER2
00005	LD	TR0	TR0
00006	AND	010.01	SAL2
00007	OR LD		
00008	OUT	010.01	SAL2
00009	LD	TR0	TR0
00010	AND NOT	255.02	INTER2
00011	AND	010.01	SAL2
00012	LD	TR0	TR0
00013	AND	010.02	SAL3
00014	OR LD		
00015	OUT	010.02	SAL3
00016	LD	TR0	TR0
00017	AND	255.02	INTER2
00018	AND	010.02	SAL3
00019	LD	TR0	TR0
00020	AND	010.03	SAL4
00021	OR LD		
00022	OUT	010.03	SAL4

Red 3.- Fin de programa.



00023 END

15.3. ENCENDIDO DE LUZ DE ESCALERA

1. Objetivo

Con este ejercicio se pretende que el alumno practique con los temporizadores y con las instrucciones básicas de tipo SET-RESET. A la vez que programa controles que le resulten familiares, dado que son automatismo que regularmente ve funcionar a diario.

2. Planteamiento

Con la primera entrada (000.00) que quiere representar la función de los pulsadores de encendido puestos todos en paralelo, se activa la salida primera (010.00) que viene a representar el encendido de la luz de la escalera, la cual se apaga al cabo de un tiempo programado. Se van a desarrollar tres modos de funcionamiento para que el alumno vaya ejecutándolos sucesivamente. El primero el más simple, que queda determinado en la explicación anterior.

Un segundo ejemplo que permite un redisparo del tiempo de luz encendida cada vez que se pulse uno de los pulsadores del encendido de la luz.

Un tercer ejercicio permite solucionar el problema que aparece si un pulsador se qued a permanentemente en posición de encendido, bien sea por avería del mecanismo o por un acto vandálico.

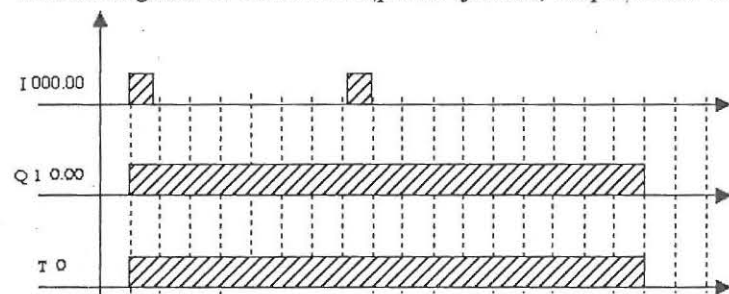
Se proporcionan los cronogramas de los dos primeros ejercicios, quedando el tercero como ejercicio para el alumno:

3. Programación

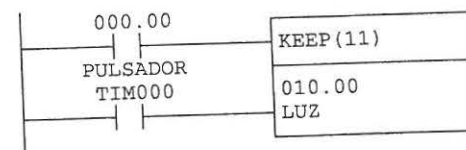
Se asignan los siguientes identificadores a las entradas y salidas:

PULSADOR	←	0.0
LUZ	→	10.00
BIT_PULSADOR		11.0

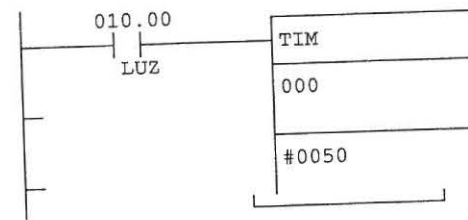
Caso 1. Diagrama de contactos del primer ejercicio, temporización simple.



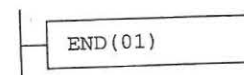
Red 1.- El PULSADOR activa mediante la función KEEP la LUZ. Se desactiva mediante el final del tiempo del temporizador T0.



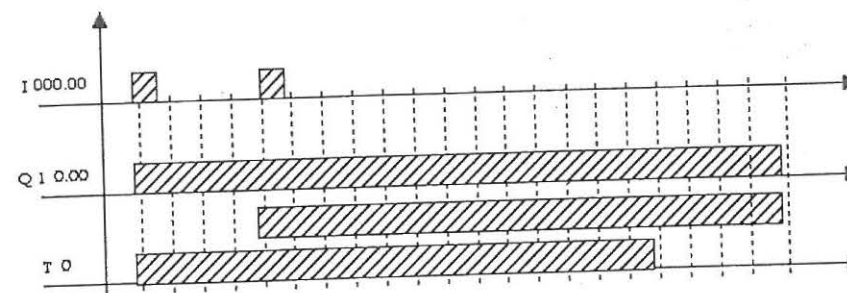
Red 2.- Mientras esté activa la LUZ el temporizador T0 realiza su cuenta.



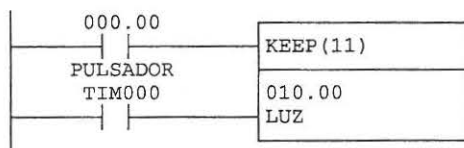
Red 3.- Fin de programa.



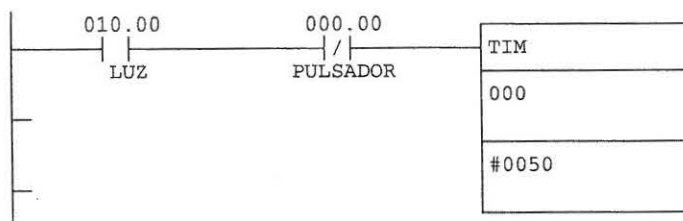
Caso 2. Diagrama de contactos del primer ejercicio, pero cada vez que se activa el PULSADOR se reinicia la temporización.



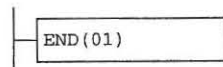
Red 1.- El PULSADOR activa mediante la función KEEP la LUZ. Se desactiva mediante el final del tiempo del temporizador T0.



Red 2.- Mientras esté activa la LUZ y no se acciona el PULSADOR, el temporizador T0 realiza su cuenta.

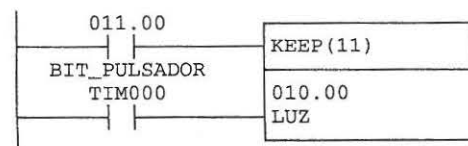


Red 3.- Fin de programa.

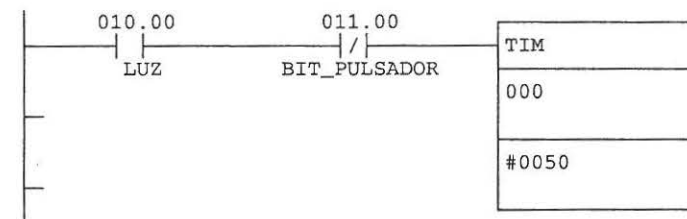


Caso 3. Diagrama de contactos del primer ejercicio, temporización redispensible y seguridad antivandalismo del pulsador.

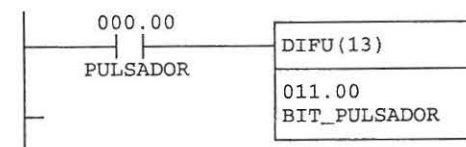
Red 1.- El PULSADOR activa mediante la función KEEP la LUZ. Se desactiva mediante el final del tiempo del temporizador T0.



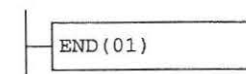
Red 2.- Mientras esté activa la LUZ y no se detecta un pulso de accionamiento el PULSADOR, el temporizador T0 realiza su cuenta.



Red 3.- La entrada E0 dispara una función de flanco de subida (11.00).



Red 3.- Fin de programa.



15.4. LIBERACIÓN Y BLOQUEO DE UN MOTOR

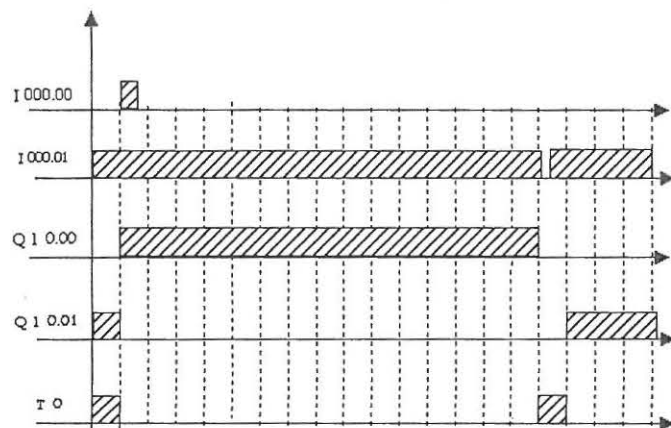
1. Objetivo

Se pretende realizar una maniobra sencilla de tipo industrial con la cual el alumno pueda ir entendiendo la aplicación real de las automatizaciones. En este ejercicio se interconectarán temporizadores y entradas/salidas digitales.

2. Planteamiento

La maniobra representa el bloqueo de un motor, por ejemplo mediante un freno mecánico o un mecanismo de enclavamiento. Al arrancar el motor se libera el bloqueo, tras la orden de parada del motor y al cabo de un breve tiempo se vuelve a accionar el bloqueo. Esta maniobra puede emplearse en grúas, elevadores, etc.

El cronograma se corresponde con el de la figura siguiente:

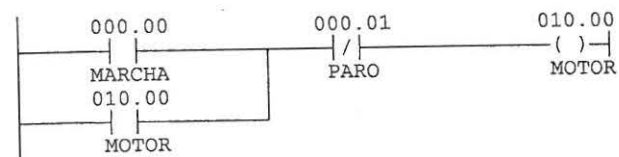


Se asignan los siguientes identificadores a las entradas y salidas:

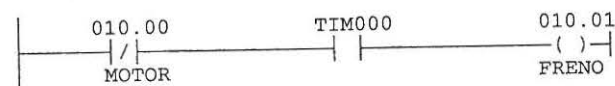
MARCHA	←	0.0	Pulsador de marcha de la maniobra.
PARO	←	0.1	Pulsador de paro, de tipo normalmente cerrado.
MOTOR	→	10.0	Activación del motor.
FRENO	→	10.1	Activación del freno de bloqueo.

3. Programación

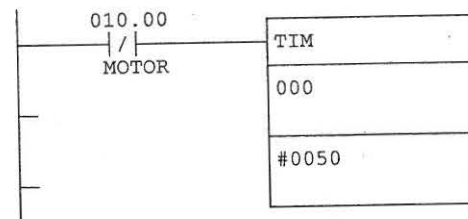
Red 1.- Si se activa la MARCHA no está activado el PARO se pone el MOTOR en marcha, con autoenclavamiento.



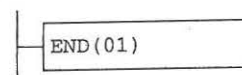
Red 2.- Al pararse el MOTOR y transcurrir un tiempo T0 se activa el FRENO.



Red 3.- Al pararse el MOTOR se activa el temporizador T0.



Red 4.- Fin de programa.



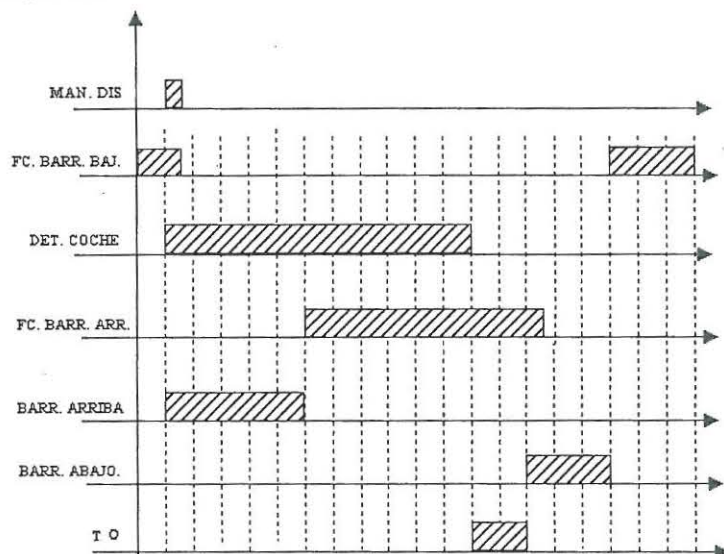
15.5. APERTURA Y CIERRE DE UNA BARRERA DE PARKING

1. Objetivo

Además de poder ejecutar maniobras de tipo industrial con la correspondiente combinación de señales de tipo digital y temporizadores, se quiere aprovechar para ir introduciendo otros elementos normales en las maniobras usuales como son los finales de carrera, los sensores de proximidad, etc., para que los alumnos aprendan a tratarlos como entradas lógicas en las automatizaciones. En este ejercicio se interconectarán temporizadores y entradas/salidas digitales.

2. Planteamiento

Se quiere automatizar la apertura y cierre de una barrera de un parking, gobernado mediante un mando a distancia. Su cronograma se corresponde con el de la figura siguiente:

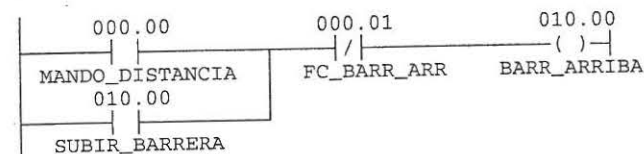


Se asignan los siguientes identificadores a las entradas y salidas:

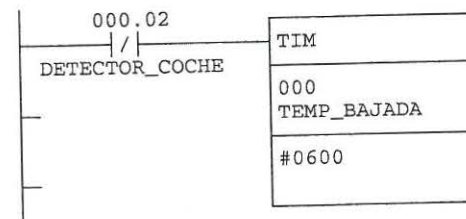
MANDO_DISTANCIA	←	0.0	Señal de inicio del mando a distancia.
FC_BARR_ARR	←	0.1	Final de carrera de barrera arriba.
DET_COCHE	←	0.2	Detector de coche cruzando.
FC_BARR_BAJ	←	0.3	Final de carrera de barrera abajo.
BARR_ARRIBA	→	10.0	Orden de subir la barrera.
BARR_ABAJO	→	10.0	Orden de bajar la barrera.

3. Programación

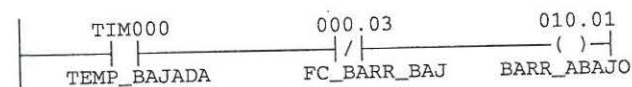
Red 1.- Si se activa el mando a distancia se da la orden de subir la barrera hasta que se activa el final de carrera de barrera arriba.



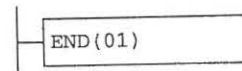
Red 2.- Una vez ha pasado el coche y no lo detecta el sensor DETECTOR_COCHE se activa el temporizador para contar un tiempo de seguridad, durante el cual se mantiene la barrera arriba.



Red 3.- Una vez ha pasado el tiempo se pone en marcha la barrera hasta tocar el final de carrera que detecta que ha bajado la barrera.



Red 4.- Fin de programa.



15.6. AUTOMATIZACIÓN DE UN APARATO ELEVADOR CONVENCIONAL DE DOS PLANTAS

1. Objetivo

Se pretende realizar una maniobra un poco más compleja que las anteriores, pero que todavía se puede realizar mediante combinación lógica de señales digitales. Puede ser una maniobra aplicable a montacargas de empresas de doble planta y como introducción a la automatización de elevadores y ascensores de varias alturas.

2. Planteamiento

En un aparato elevador convencional se tienen las siguientes señales de control:

- una por cada puerta,
- una al menos, por cada altura en su final de carrera,
- una de llamada o envío (pulsador) a cada planta.

Así pues, para este ejercicio son necesarias al menos 6 señales de entrada. Como estos ejercicios se plantean para que puedan ser implementados en autómatas de bajo nivel, que suelen disponer solamente de 6 entradas, no se emplean más señales de control, que en realidad si que existen, tales como finales de carrera de seguridad uno superior y otro inferior.

La programación directa a partir del cronograma no es la más adecuada para maniobras como la presentada en este ejercicio. En lugar de ello, el planteamiento de las redes de control se realizará del modo siguiente:

- Se emplea una instrucción KEEP (set-reset) para cada salida a controlar.
- Se programan las condiciones lógicas necesarias para activar (entrada SET) y desactivar (RESET) cada una de estas salidas.

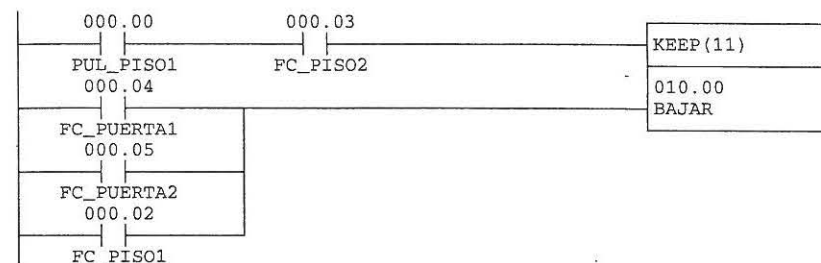
Se asignan los siguientes identificadores a las entradas y salidas:

PUL_PISO1	←	0.0	Pulsador llamada piso 1.
PUL_PISO2	←	0.1	Pulsador llamada piso 2.
FC_PISO1	←	0.2	Final de carrera de elevador en el piso 1.
FC_PISO2	←	0.3	Final de carrera de elevador en el piso 2.
FC_PUERTA 1	←	0.4	Puerta abierta en piso 1.
FC_PUERTA 1	←	0.5	Puerta abierta en piso 2.

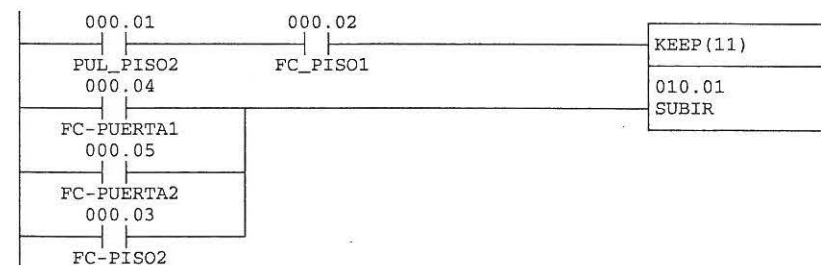
BAJAR	→	10.0	Orden de bajar el elevador.
SUBIR	→	10.1	Orden de subir el elevador.
PILOTO_BAJAR	→	10.2	Piloto de elevador bajando.
PILOTO_SUBIR	→	10.3	Piloto de elevador subiendo.

3. Programación

Red 1.- Activación de la salida de bajada del elevador.



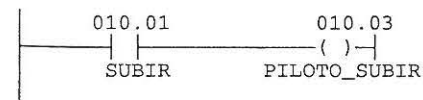
Red 2.- Activación de la salida de subida del elevador.



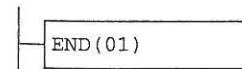
Red 3.- Activación de la salida del piloto de bajada.



Red 4.- Activación de la salida del piloto de subida.



Red 5.- Fin de programa.



15.7. ARRANQUE Y PARO DE UN MOTOR CON UN SOLO PULSADOR

1. Objetivo

Maniobra sencilla por mediación de la cual se pretende manejar dos de las funciones básicas de la programación (KEEP, DIFU), una es una función de SET/RESET, y la otra es una función que se utiliza para convertir una señal de entrada de tiempo variable en un pulso de un ciclo de programa.

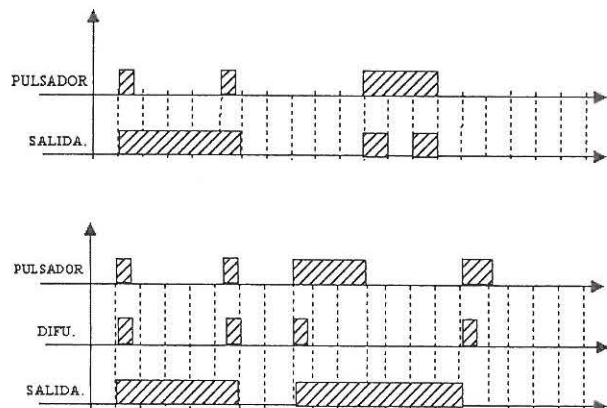
2. Planteamiento

Se desarrolla un mismo ejercicio pero de dos formas distintas, la primera se trata de generar una maniobra sencilla de arranque y paro de un motor con un solo pulsador: en el primer pulso activa el motor. Con cada siguiente actuación sobre el pulsador la salida cambia de estado.

Esta maniobra realizada como se expone en el primer ejercicio puede tener un problema importante. Si el tiempo de actuación sobre el pulsador es mayor que el de un ciclo de trabajo del autómata, puede que la salida actúe pero un periodo de tiempo tan pequeño que realmente no llegue a funcionar el contactor alimentado a la salida.

Para evitar este tipo de anomalías, se corrigen las señales de entrada de forma que dure lo que dure el tiempo de actuación sobre el pulsador, tomando como entrada válida el flanco de subida del pulsador de entrada. Para volver a validar la entrada descrita como una nueva pulsación hay que soltar y volver a actuar sobre el pulsador. Se programa así el segundo ejercicio de este ejemplo.

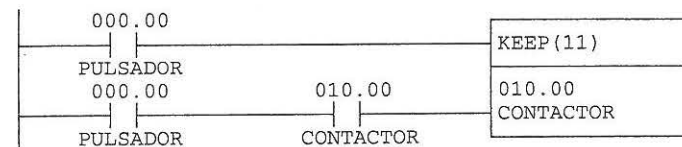
El cronograma para este ejemplo sería doble, pero teniendo en cuenta el tiempo de ciclo, cosa que puede confundir al alumno. Se realiza dos cronogramas, en el primero la señal de entrada, más larga, se considera que ocupa dos ciclos de programa y un segundo cronograma en el que se ha tenido en cuenta la aportación de la función DIFU. Así quedaría:



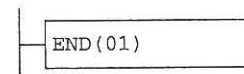
3. Programación

Caso 1.

Red 1.- La entrada del pulsador actúa sobre la función que controla la salida del contactor.

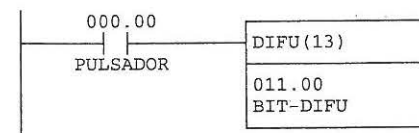


Red 4.- Fin de programa.

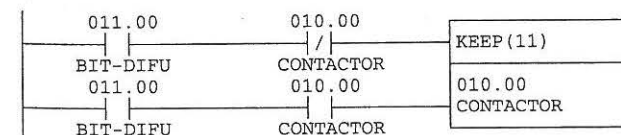


Caso 2.

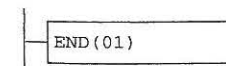
Red 1.- La entrada del pulsador se aplica sobre la función DIFU.



Red 2.- El bit actuado por mediación del DIFU actúa sobre la función que controla la salida SAL1 (10.00).



Red 3.- Fin de programa.



15.8. DETECCIÓN DEL FLANCO DE SUBIDA Y BAJADA DE UNA ENTRADA

1. Objetivo

Con este ejercicio se pretende que el alumno conozca funciones muy útiles para la programación como son el (DIFU (13)), la cual se encarga de detectar flancos de subida y (DIFD (14)), encargada de detectar los flancos de bajada. También la función KEEP, encargada de mantener a '1' la dirección de memoria en ella indicada la salida deseada. Mediante su entrada de RESET se puede conseguir devolver a cero dicha dirección de memoria o salida.

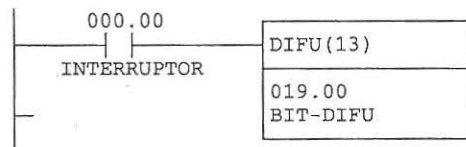
2. Planteamiento

Se pretende detectar el flanco de subida de la primera entrada (000.00) cuando esta se active y también el flanco de bajada cuando esta se desactive activando en ambos casos la dirección de memoria correspondiente (019.00) y consiguiendo que dicha salida sea intermitente.

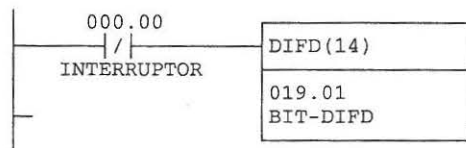
3. Programación

Caso 1.

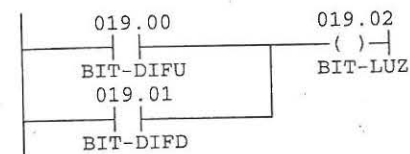
Cuando la primera entrada (000.00) se activa se detecta su flanco de subida y la dirección de memoria correspondiente (019.00) se pone a '1' mientras dure ese flanco de subida, indicando que el flanco ha sido detectado.



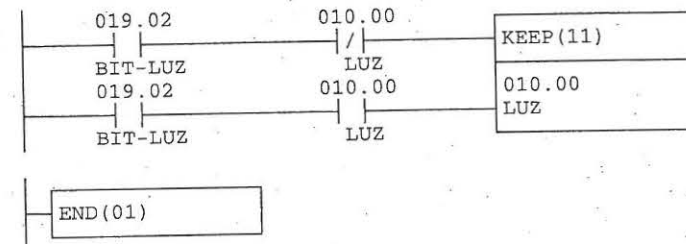
Cuando la primera entrada (000.00) se desactiva se detecta su flanco de bajada y la dirección de memoria correspondiente (019.01) se pone a '1' mientras dure ese flanco de bajada, indicando que el flanco ha sido detectado.



Tanto la dirección de memoria (019.00) como la (019.01) me activan la dirección (019.02), la cual al ser activada por dos posiciones de memoria que actúan como pulsos actuará de forma intermitente.

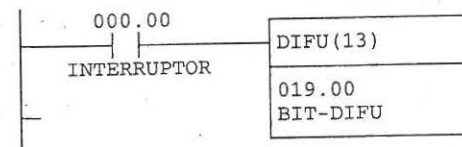


Por otra parte la posición de memoria (019.02) se encarga, mientras la primera salida (010.00) no esté activa, de que esa primera salida pase a '1' mediante la función KEEP, y cuando esta primera salida está a '1' y también el bit (019.02), esta salida (010.00) pasa a '0' al activarse la entrada RESET de la función KEEP. De este modo se consigue el efecto intermitente.

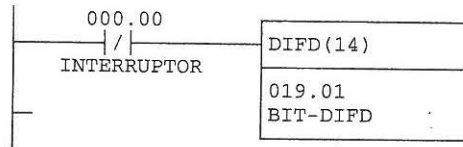


Caso 2.

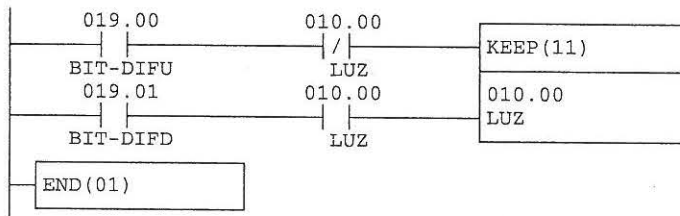
Esto se puede realizar de una forma más sencilla, evitando el paso por la zona de memoria (019.02) anteriormente utilizada. Cuando la primera entrada (000.00) se activa se detecta su flanco de subida y la dirección de memoria correspondiente (019.00) se pone a '1' mientras dure ese flanco de subida, indicando que el flanco ha sido detectado.



Cuando la primera entrada (000.00) se desactiva se detecta su flanco de bajada y la dirección de memoria correspondiente (019.01) se pone a '1' mientras dure ese flanco de bajada, indicando que el flanco ha sido detectado.



Por otra parte mediante la función KEEP y usando las dos posiciones de memoria anteriormente utilizadas (019.00 y 019.01) se consigue realizar el mismo efecto intermitente de la salida (010.00). Si la salida (010.00) no está activa cuando se activa la primera entrada y se detecta un flanco de subida (019.00), la salida se activará. Al estar la salida (010.00) activa cuando la primera entrada se desactive y se detecte el flanco de bajada (019.01) dicha salida (010.00) pasará a '0' al activarse la entrada RESET de la función KEEP.



15.9. ACTIVACIÓN Y DESACTIVACIÓN DE UNA SALIDA MEDIANTE FLANCO DE SUBIDA Y BAJADA

1. Objetivo

Con este ejercicio se pretende que el alumno conozca funciones muy útiles para la programación como son el (DIFU (14)), la cual se encarga de detectar flancos de subida y (DIFD (14)), encargada de detectar los flancos de bajada.

2. Planteamiento

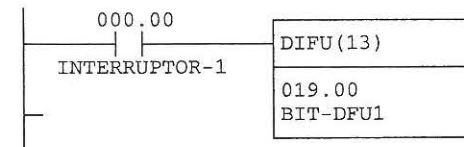
En este ejercicio se pretende que con 3 interruptores diferentes (000.00, 000.01, 000.02) se conecte la salida deseada (010.00) cuando detecte el flanco de subida, y se desconecte cuando detecte el flanco de bajada, ambas cosas con cualquiera de los 3 interruptores.

3. Programación

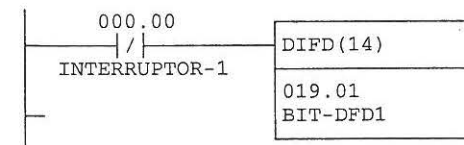
Se realizará la programación mediante dos sistemas diferentes, lo que permite comparar ambas técnicas presentadas

Caso 1.

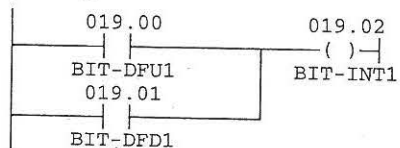
Cuando la primera entrada (000.00) se activa se detecta su flanco de subida y la dirección de memoria correspondiente (019.00) se pone a '1' mientras dure ese flanco de subida, indicando que el flanco ha sido detectado.



Cuando la primera entrada (000.00) se desactiva se detecta su flanco de bajada y la dirección de memoria correspondiente (019.01) se pone a '1' mientras dure ese flanco de bajada, indicando que el flanco ha sido detectado.

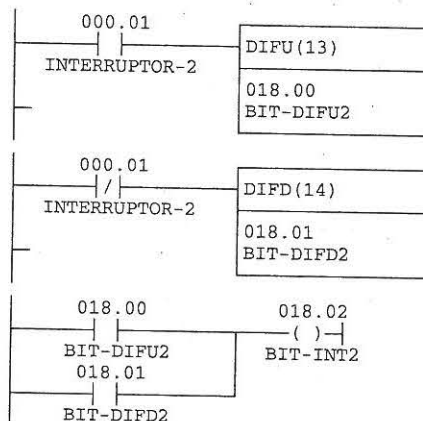


Tanto la dirección de memoria (019.00) como la (019.01) activan la dirección (019.02),

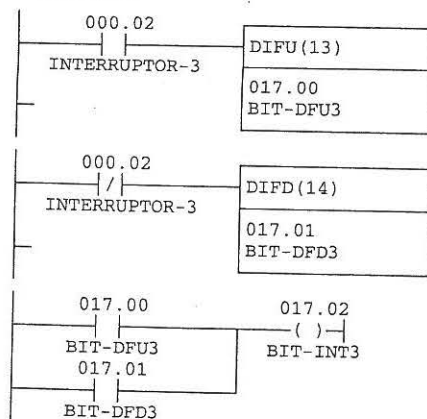


Este mismo esquema se repite para las otras dos entradas cuyos flancos se desea controlar:

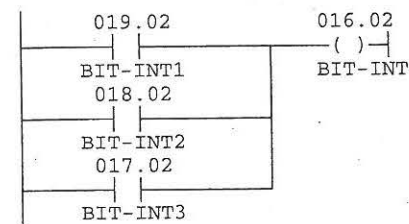
Entrada 2



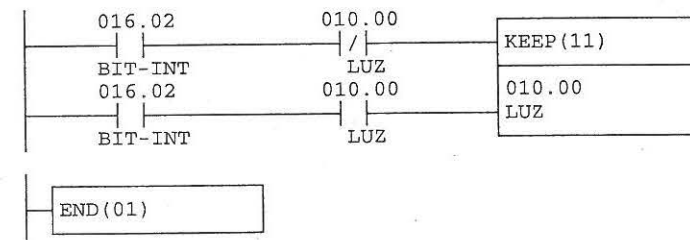
Entrada 3



Por último, cualquier cambio de estado en los bits generados para cada interruptor actúan sobre la salida deseada.



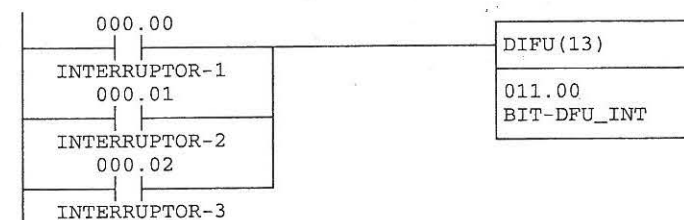
Mediante esta posición de memoria (016.02) se consigue que la salida deseada (010.00) haga el efecto intermitente utilizando la función KEEP.



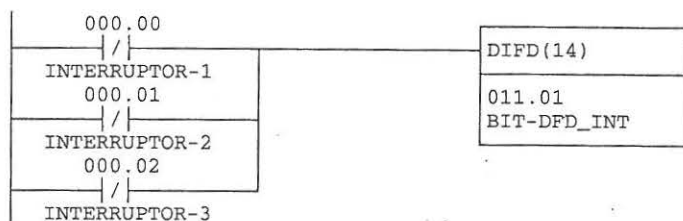
Caso 2.

En lugar de realizar la detección de flancos de cada señal de entrada de forma individualizada, se realiza en paralelo.

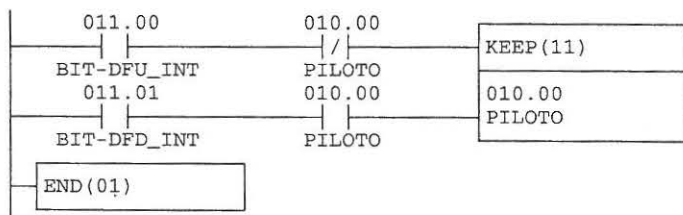
Cuando cualquiera de los 3 interruptores pasa a '1' se detecta su flanco de subida con el DIFU y activa la posición de memoria (011.00).



Cuando cualquiera de los 3 interruptores pasa a '0' se consigue detectar su flanco de bajada por medio del DIFD y activar la posición de memoria (011.01).



Con memoria (011.00), que detecta los flancos de subida, la salida (010.00) se fija a '1'. Con la posición de memoria (011.01) se detectan los flancos de bajada, que resetea la salida (010.00).



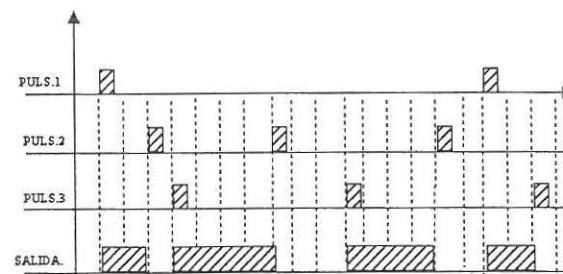
15.10. ENCENDIDO Y APAGADO DE UNA LUZ CON TRES PULSADORES

1. Objetivo

Se pretende realizar una maniobra sencilla que represente una aplicación de tipo domestico que con las funciones de SET/RESET (KEEP), con la cual el alumno pueda ir entendiendo con aplicaciones cercanas el funcionamiento de estas funciones.

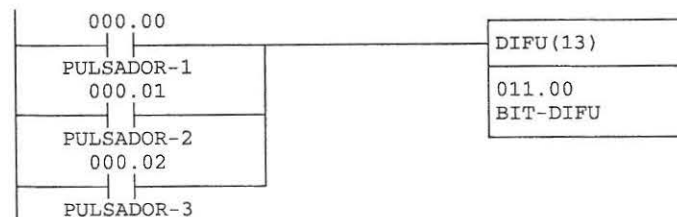
2. Planteamiento

El presente ejercicio realiza la maniobra del encendido y apagado de una luz o motor desde distintos puntos, en este caso desde 3 puntos diferentes. Puede entenderse como la luz conmutada del pasillo o de un dormitorio. Cada pulso invierte el estado de la salida, es decir si la luz se encuentra encendida al actuar sobre cualquier de los pulsadores de entrada se apaga y viceversa. Su cronograma es el de la figura siguiente:

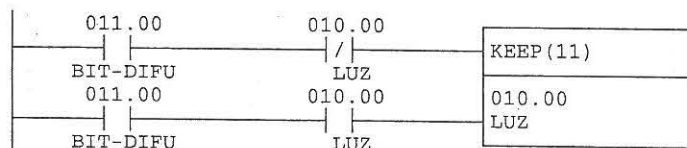


3. Programación

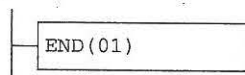
Red 1.- Cualquier entrada actúa sobre el bit (11.00), para conseguir hacer las entradas reales de cambio de estado de la luz independientes de la duración de la entrada.



Red 2.- El bit de cambio de estado se aplica a una función KEEP, que se encarga de controlar la salida de la luz SAL1 (10.00).



Red 3.- Fin de programa.



15.11. CONTROL DE LA CAPACIDAD DE UN DEPÓSITO

1. Objetivo

Con este ejercicio se pretende que el alumno conozca las funciones (ADD (30)), la cual se encarga de realizar sumas entre valores de dos canales diferentes, un valor determinado con el valor de un canal, etc. y la función (SUB (31)), encargada de realizar la resta con las mismas características que la suma. Del mismo modo también el alumno utilizará la función de comparación (CMP(20)), la cual se encarga de realizar la comparación entre dos canales, un canal y un valor determinado, etc. Al alumno también se le indica en este ejercicio como utilizar la función BSET(71) para mover un valor deseado a un canal o canales deseados.

2. Planteamiento

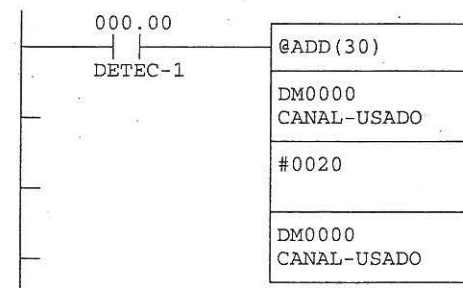
Se pretende controlar la capacidad de un depósito mediante dos detectores, uno de ellos que detecte que el depósito se encuentra por debajo de los 100 litros y el otro que indique que se encuentra por encima de los 500 litros. En ambos casos se enciende el piloto correspondiente y se apaga otro que indica depósito en condiciones normales.

Por otra parte se precisa limpiar (poner a cero) desde el DM0000 al DM1000.

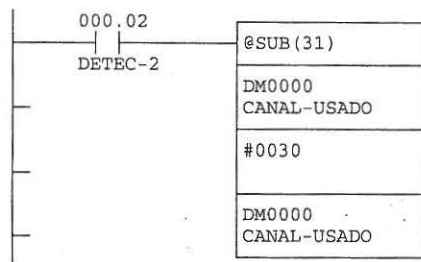
3. Programación

CON DOS DETECTORES SABER EN TODO MOMENTO EL NIVEL DEL DEPOSITO.

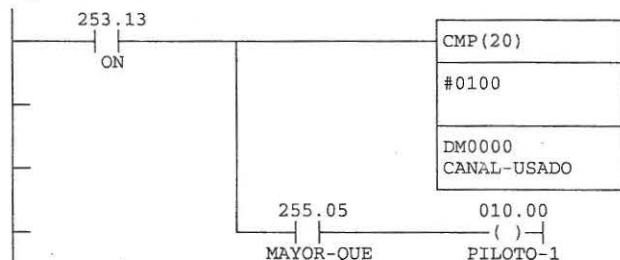
Cuando el detector_1 (000.00) es activado, este activa el módulo ADD de suma y al DM0000 le suma 20, indicando que han entrado 20 litros, e introduciendo el resultado en el mismo canal DM0000.



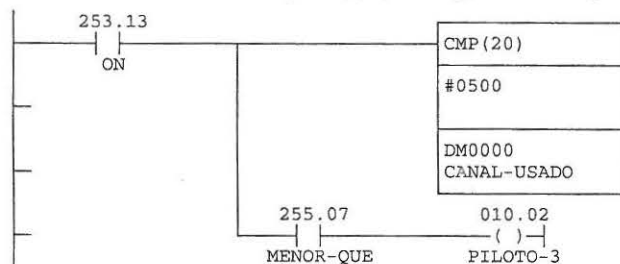
Al activarse el detector_2 (000.02), este activa el módulo SUB de resta y al DM0000 le resta 30, indicando que han salido 30 litros, e introduciendo el resultado en el mismo canal DM0000.



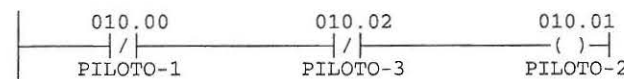
El bit (253.13) es un bit que siempre está a ON, por lo que el bloque del comparador (CMP) siempre estará activo para realizar la comparación. En este caso compara el valor 100 con el valor del canal DM0000 y en caso de que 100 sea mayor que el valor del canal DM0000 se activa el piloto_1 (010.00), indicando que hay menos de 100 litros.



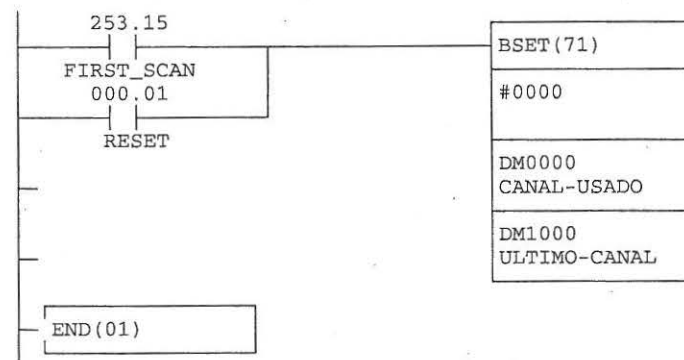
El bit (253.13) siempre está a ON, por lo que el bloque del comparador (CMP) siempre estará activo para realizar la comparación. En este caso compara el valor 500 con el valor del canal DM0000 y en caso de que 500 sea menor que el valor del canal DM0000 se activa el piloto_3 (010.02), indicando que hay más de 500 litros.



Mediante el piloto_1 (010.00) y el piloto_3 (010.02) se controla el piloto_2 (010.01). El piloto_2 normalmente está activado, pero en el caso de que uno de los otros dos pilotos se active, indicando que el depósito está por debajo de los 100 litros o por encima de los 500 litros, el piloto_2 se desactiva.



La función BSET(71) se encarga de dar el valor que se seleccione a los canales indicados en dicha función. En el primer ciclo SCAN, o cuando se active el RESET (000.01) esta función procederá a dar el valor 0 desde el DM0000 al DM1000, borrando así los valores anteriores.



15.12. DETECCIÓN DE NÚMERO DE ENTRADAS ACTIVAS

1. Objetivo

El objetivo del ejercicio es comprender el funcionamiento de la instrucción "INC" que se utiliza para incrementar en una unidad el valor de una palabra (16 bits). Se hace uso también de la instrucción "BSET" cuya función es rellenar una tabla de palabras con un valor dado. Finalmente se utiliza la función "CMP" para realizar comparaciones.

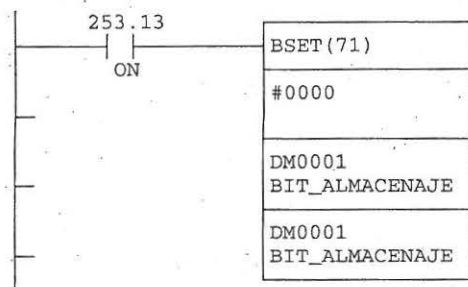
2. Planteamiento

El primer paso consistirá en poner a cero, en el primer ciclo de ejecución, el valor de la palabra que vamos a incrementar (DM0001) utilizando para ello la instrucción BSET.

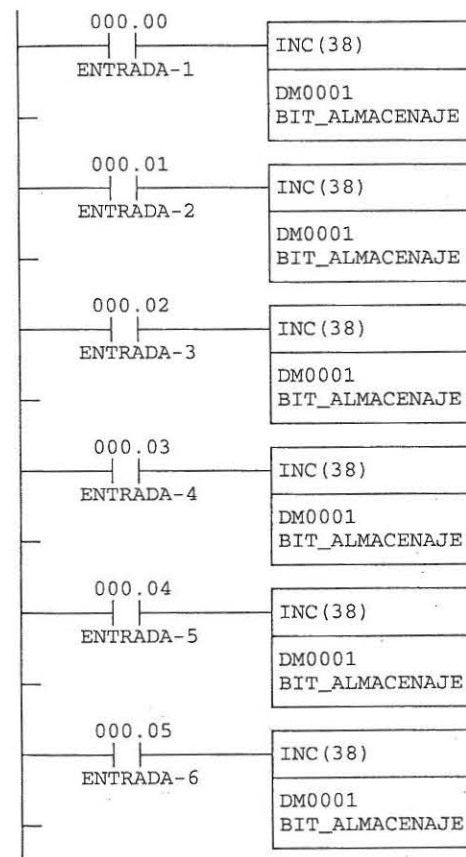
Cada vez que activamos una entrada se incrementa el valor de DM0001 de tal forma que podemos comparar el valor de esta palabra y saber cuántas entradas hay activas. 1 entrada activa pondrá en funcionamiento 1 salida, 2 entradas activarán 2 salidas, 3 entradas activarán 3 salidas y 4 o más entradas activas pondrán en funcionamiento 4 salidas.

3. Programación

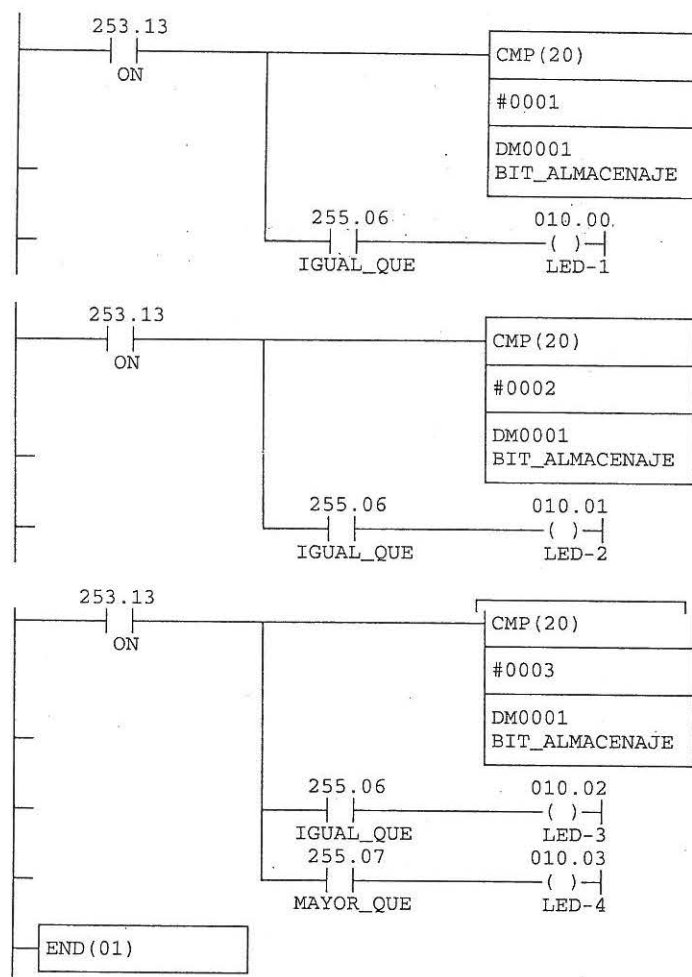
Ponemos a cero la palabra que indica el número de entradas activas (BIT_ALMACENAJE).



La activación de una entrada incrementa BIT_ALMACENAJE en una unidad.



Realizamos la comparación para saber cuántas entradas están activadas, activando las salidas correspondientes.



15.13. CONTROL DE ACTIVACIÓN DEL MISMO NÚMERO DE ENTRADAS Y SALIDAS

1. Objetivo

El objetivo del ejercicio es comprender el funcionamiento de la instrucción "INC" que se utiliza para incrementar en una unidad el valor de una palabra (16 bits).

Hacer uso de los comparadores así como de las variables de control asociadas a estos, recordemos que existen diferentes palabras de control como son: El indicador de "igual que", cuya dirección es 255.06 (P_EQ). El indicador de "mayor que", cuya dirección es 255.05 (P_GT). Y el indicador de "menor que", cuya dirección es 255.07 (P_LT).

2. Planteamiento

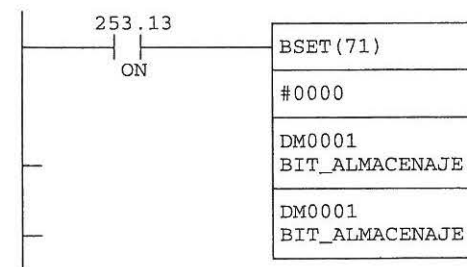
Este ejemplo es similar al anterior y difiere en el enunciado del problema. Aquí lo que se pretende realizar es un programa en el cual haya activas el mismo número de entradas que de salidas. Como el autómata de posee 6 entradas y 4 salidas, a partir de más de 4 entradas seguirá habiendo activas 4 salidas.

El primer paso consistirá en poner a cero, en el primer ciclo de ejecución, el valor de la palabra que vamos a incrementar (DM0001) utilizando para ello la instrucción BSET.

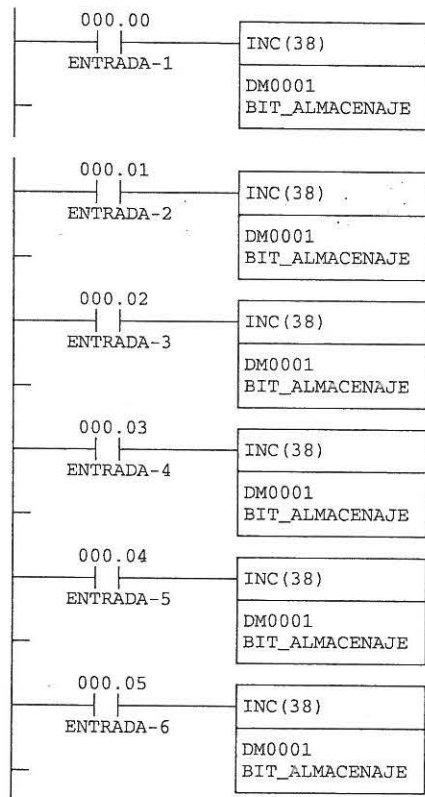
Cada vez que activamos una entrada se incrementa el valor de DM0001 de tal forma que podemos comparar el valor de esta palabra y saber cuántas entradas hay activas. 1 entrada activa pondrá en funcionamiento 1 salida, 2 entradas activarán 2 salidas, 3 entradas activarán 3 salidas y 4 o más entradas activas pondrán en funcionamiento 4 salidas.

3. Programación

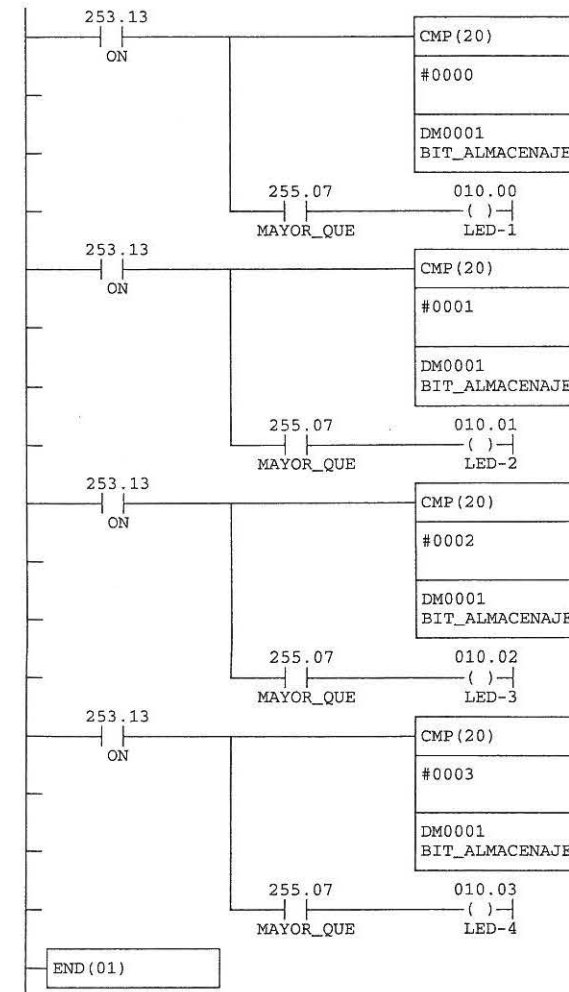
Ponemos a cero la palabra que nos indica el número de entradas activas (BIT_ALMACENAJE).



A continuación encontramos una serie de redes donde cada entrada incrementa en una unidad la palabra BIT_ALMACENAJE, que nos indica el número de entradas activas en ese momento.



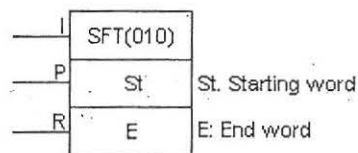
En función del número de entradas activas, las cuales se encuentran almacenadas en el `BIT_ALMACENAJE`, se activan las diferentes salidas. El primer comparador realiza la comparación del `BIT_ALMACENAJE` con "cero" y si es mayor enciende la primera salida. Así sucesivamente, el segundo comparador lo comparamos con "uno" y si es mayor se activa la segunda salida. Para terminar el último comparador lo comparamos con "tres" y si es mayor activamos la cuarta salida. Con esto si el número de entradas es 2, como es mayor que "cero" y "uno" habré activado dos salidas y así sucesivamente.



15.14. REGISTROS DE DESPLAZAMIENTO

1. Objetivo

El objetivo del ejercicio es comprender el funcionamiento de la instrucción "SFT" que es el registro de desplazamiento. Dicha instrucción posee 3 condiciones de ejecución I, P y R.



Si se ejecuta "SFT" y la condición de ejecución P está en ON pero estaba en OFF durante la última ejecución y R está en OFF, la condición de ejecución I se desplaza al bit más a la derecha del registro de desplazamiento definido entre St y E, es decir, si I está en ON, se desplaza un 1 en el registro; si I está en OFF, se desplaza un 0. Al desplazar I en el registro, todos los bits existentes anteriormente en el registro se desplazan hacia la izquierda y se pierde el bit más a la izquierda del registro.

2. Planteamiento

Lo que se pretende probar en este ejercicio es el uso de la instrucción "SFT" y la comprensión de su uso.

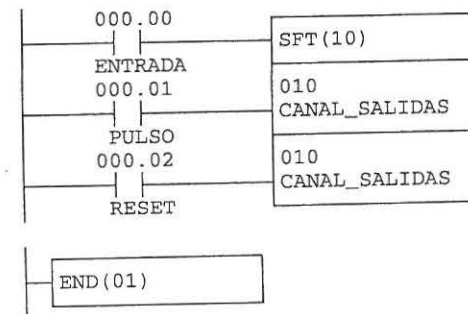
Las tres condiciones de ejecución son "ENTRADA", la segunda condición es "PULSO" y la tercera condición es "RESET". Cuando se produzca un flanco ascendente de la señal "PULSO" y siempre que este a "OFF" la condición de "RESET" se pasará el valor de la condición "ENTRADA", ya sea "ON" un "1" o "OFF" un "0" al bit más bajo del "Starting Word" perdiéndose el bit más significativo de dicha palabra, o sea se produce un desplazamiento de la palabra hacia la izquierda. Al siguiente flanco ascendente se desplazará la palabra hacia la izquierda tomando de nuevo el valor de la entrada en el bit menos significativo. Así hasta rellenar la palabra entera entonces se continuará por la siguiente hasta llegar a la "End Word".

3. Programación

Caso 1.

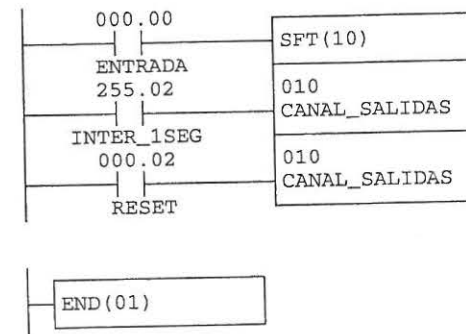
En la programación de esta red se usan las 3 condiciones antes mencionadas, y como "Starting Word" y "End Word" usamos el canal de salidas. Es decir, el valor que vayamos pasando a la palabra es el que se vera representado en las salidas.

Por ejemplo, si ponemos a "ON" la "ENTRADA" y el "PULSO" se pasará un 1 al canal de salidas (que por defecto lo vamos a tomar como que tiene un "0") por tanto el autómata se verá iluminada la primera lámpara. Si ponemos a "OFF" ambas señales anteriores y después ponemos a "ON" la entrada "PULSO" pasaríamos un "0" al canal de salidas que con lo existente nos daría un "...0010" con lo que se iluminaría la segunda lámpara apagándose la primera. Si volviésemos a poner a "OFF" la señal "PULSO" y pusiésemos a "ON" la "ENTRADA" y el "PULSO" se pasará un 1 al canal de salidas quedándonos éste "...0101" con lo que se apagará la lámpara dos y se encenderá la primera y la tercera lámpara.



Caso 2.

Este caso es idéntico al anterior pero se cambia la segunda condición de ejecución haciéndola periódica de 1 segundo, para ello se utiliza un bit de control interno del autómata que se activa cada segundo. De esta forma se le dota al ejercicio de una secuencialidad en las acciones y en función de la "ENTRADA" conseguimos que se iluminen unas u otras lámparas.

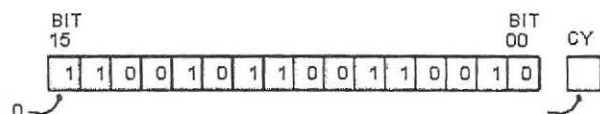


15.15. DESPLAZAMIENTO DE UN DATO A IZQUIERDAS Y DERECHAS

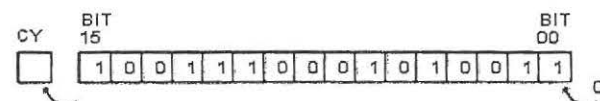
1. Objetivo

Este ejercicio permite usar las instrucciones "BSET" que es rellenar un bloque de datos correlativos con el mismo valor, para ello debemos darle el valor al cual queremos rellenarlo así como el dato inicial y el dato final del bloque de datos. Sobre todo, nos introduce dos nuevas instrucciones referidas a los registros que son "ASR" desplazamiento aritmético de bits hacia la derecha y "ASL" Desplazamiento aritmético de bits hacia la izquierda. Lo que realiza el ASR es introducir un "cero" en el "bit 15" de la palabra y el "bit 0" lo pasa al "CY" los demás los desplaza una posición hacia la derecha. A continuación se muestra un gráfico indicativo:

ASR:



ASL:

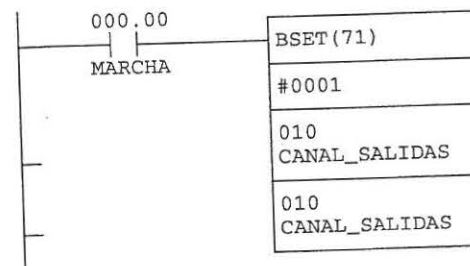


2. Planteamiento

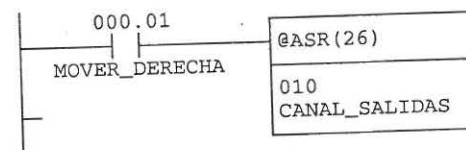
El planteamiento de este problema es el de desplazar tanto a izquierda como a derechas el dato introducido en primera instancia. Para ello, tenemos 3 entradas "MARCHA", "MOVER_DCH", y "MOVER_IZQ". Al accionar la entrada "MARCHA" se cargará el valor 1 en el canal de salidas, es decir se iluminará la primera lámpara del autómata. Con las otras entradas "MOVER_DCH" y "MOVER_IZQ" desplazaremos la activación de las demás salidas, o sea, si pulsamos "MOVER_IZQ" se iluminará la segunda lámpara.

3. Programación

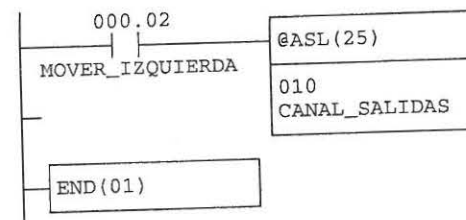
3.1.- Al activar la entrada "MARCHA" se activa la instrucción "BSET" que realiza la carga del valor 1 al canal de Salidas.



Cada flanco de la entrada "MOVER_DERECHA" activa la instrucción "ASR" que realiza el desplazamiento a la derecha de los bits. Cuando una instrucción va precedida por el símbolo "@" significa que detecta los flancos de la entrada. Por ejemplo, esta instrucción solo se ejecutaría cada cambio de "OFF" a "ON" que se produjese en la entrada "MOVER_DERECHA" porque si no se estaría desplazando todo el rato.



Similar a la red anterior pero el desplazamiento lo realizas hacia la izquierda.



15.16. CARGA DE UN VALOR EN UNA DIRECCIÓN E INCREMENTO DEL MISMO

1. Objetivo

Este ejercicio permite usar las instrucciones "BSET" que es rellenar un bloque de datos correlativos con el mismo valor, para ello debemos darle el valor al cual queremos rellenarlo así como el dato inicial y el dato final del bloque de datos. Otra de las instrucciones utilizadas es "MOV" que se utiliza para mover de un canal fuente a un canal destino por tanto habrá que darle los dos canales, pudiéndose ser el canal fuente una constante. Por ultimo también se utiliza la instrucción "INC" que incrementa en una unidad el valor donde se almacena.

2. Planteamiento

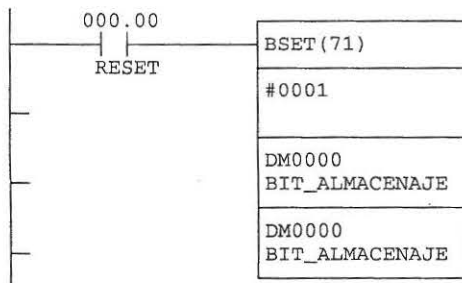
Poseemos dos entradas el "RESET" y la entrada "MARCHA". La entrada "RESET" realiza la carga del valor "0001" al BIT_ALMACENAJE, mientras que la entrada "MARCHA" se encarga de incrementar en una unidad el BIT_ALMACENAJE. En todo instante se esta pasando al canal de salidas el valor ubicado en las direcciones a las cuales apunta el BIT_ALMACENAJE, direccionamiento indirecto.

El direccionamiento indirecto se caracteriza por el uso de los operadores "*" y "@". Se basa en direccionar sin poner la dirección real a la cual se le quiere direccionar, es decir se parte de una dirección base y de hay se le suma: DIR_BASE +3, por ejemplo.

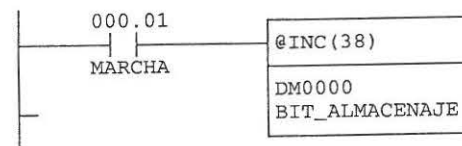
3. Programación

Caso 1.

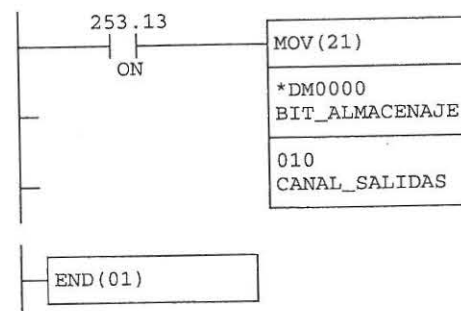
Al activar la entrada "RESET" cargamos el valor 1 a BIT_ALMACENAJE. Quedando almacenado el valor 1 en el BIT_ALMACENAJE y si no hacemos nada más, al canal de salidas se le pasara el valor donde apunta el BIT_ALMACENAJE que en este caso seria el 1, o sea el DM0001.



Al activar la entrada "MARCHA" incrementamos en una unidad el valor del BIT_ALMACENAJE. Con lo cual ya variamos el dato a transferir al canal de las salidas. Por ejemplo si solo pulsásemos una vez el botón de marcha ahora el valor del BIT_ALMACENAJE seria 2, y pasaríamos al canal de salidas el valor donde apunta el BIT_ALMACENAJE que en este caso seria el 2, o sea el valor del DM0002.

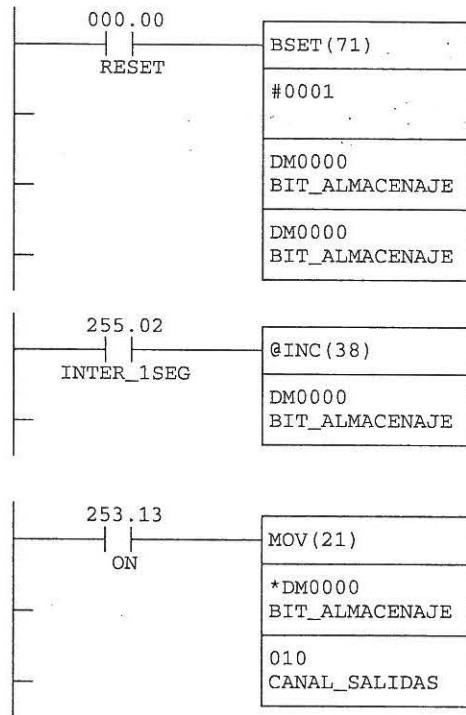


Esta red nos indica que siempre esta activa, para ello hacemos uso del BIT interno del autómata programable que siempre esta activo, cuya dirección es "253.13". Esta red es la encargada del direccionamiento indirecto, donde mueve el valor de donde apunta el "BIT_ALMACENAJE" al canal de salidas.



Caso 2.

A continuación se plantea el mismo ejercicio que anteriormente pero a diferencia de que el incremento del "BIT_ALMACENAJE" lo realizamos de forma automática, es decir usando el BIT interno que oscila cada segundo, cuya dirección es la "255.02".



15.17. CARGA E INCREMENTO DE UN VALOR EN UNA DIRECCIÓN MEDIANTE CONTADOR

1. Objetivo

Este ejercicio permite usar las instrucciones "BSET" que es rellenar un bloque de datos correlativos con el mismo valor, para ello debemos darle el valor al cual queremos rellenarlo así como el dato inicial y el dato final del bloque de datos. También hace uso de los "Contadores" que funcionan descontando el SV en función de los flancos que se producen a la entrada del Contador.

Otra de las instrucciones utilizadas es "INC" que incrementa en una unidad el valor donde se almacena. Por último también se utiliza la instrucción "MOV" para mover de un canal fuente a un canal destino.

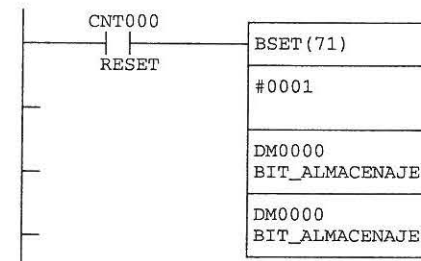
2. Planteamiento

Mediante la utilización de un contador como la entrada "RESET" del sistema, es decir, dicho contador cuenta 10 y resetea el sistema, contador y acumulador que en este caso es el "BIT_ALMACENAJE". Cuando se produce el "RESET" se pasa el valor "0001" al "BIT_ALMACENAJE" se empieza la cuenta en el contador, la señal interna del autómata que se activa cada segundo, cuya dirección es 255.02 será la encargada de incrementar en una unidad el "BIT_ALMACENAJE" y descontar en una unidad el contador hasta llegar a 10 donde se resetea y comienza de nuevo. Todo el rato estamos pasando donde apunta el "BIT_ALMACENAJE" al canal de salidas, utilizamos indexado o direccionamiento indirecto.

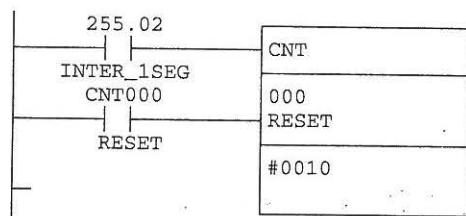
Entonces cada segundo estaremos pasando el valor del DM0001 al segundo después el valor del DM0002 al siguiente el valor del DM0003 y así hasta alcanzar el valor del contador.

3. Programación**Caso 1.**

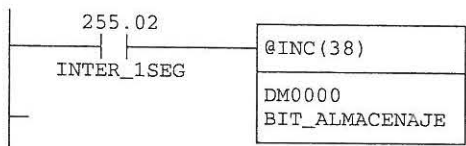
Cuando se activa la salida del contador "RESET" se transfiere el valor "0001" al "BIT_ALMACENAJE".



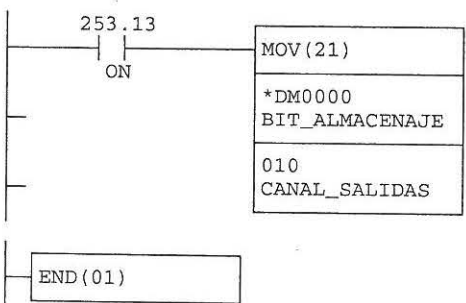
La señal interna que se activa cada segundo es la entrada del contador y el reset del mismo sería su propia salida. Así cuando alcance la cuenta se reseteará y comenzará otra vez a contar.



Esta red realiza el incremento en una unidad del valor del "BIT_ALMACENAJE". Como se observa la instrucción posee el símbolo "@" delante del nombre de la instrucción lo cual nos garantiza que solo se producirá cuando haya un flanco.

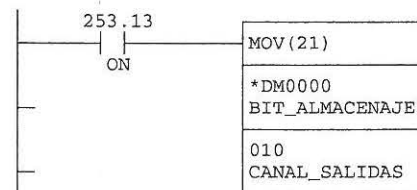
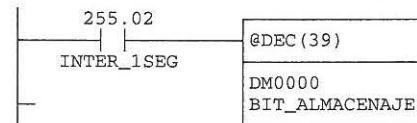
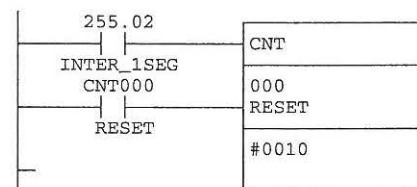
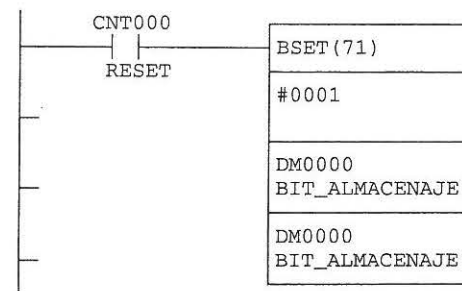


Esta red es la encargada de transferir el valor de la dirección donde apunta el "BIT_ALMACENAJE" al canal de salidas.



Caso 2.

Este es un nuevo ejercicio similar al anterior pero con la salvedad que decreenta el valor del "BIT_ALMACENAJE" en una unidad.



15.18. RECORRIDO DE TODA LA MEMORIA DEL AUTÓMATA MEDIANTE PUNTERO

1. Objetivo

Este ejercicio permite usar diferentes instrucciones anteriormente utilizadas para afianzarlas y comprobar sus ejecuciones.

Se pretende usar "BSET", "INC", "DEC", "MOV", y "CMP" así como los bits internos del autómata de control de "primer ciclo de ejecución", el de "activación cada segundo" o el de "igual que".

Se pretende realizar un ejemplo automático de pasar toda la memoria DM del autómata al canal de salidas, claro esta, palabra a palabra, primero de forma ascendente y después de forma descendente.

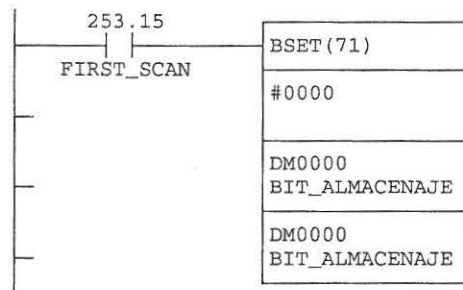
2. Planteamiento

Se pretende recorrer toda la memoria del autómata de forma ascendente y descendente, para visualizarlo lo haremos a través del canal de salidas.

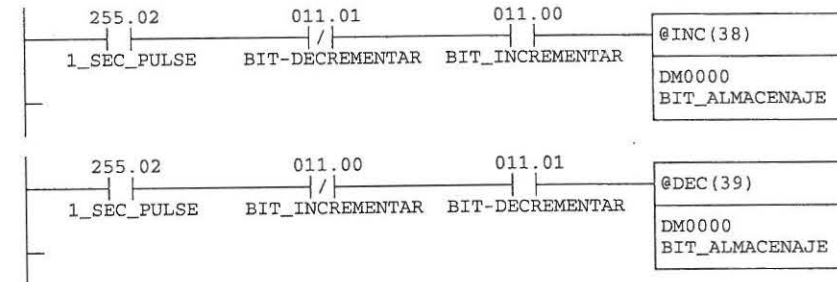
Para ello en el primer ciclo de ejecución se carga el valor inicial "0000" en el "BIT_ALMACENAJE", encargado del almacenamiento de la cuenta. A continuación haciendo uso del bit interno de activación cada segundo se va incrementando una unidad el "BIT_ALMACENAJE" hasta llegar al final de la memoria que se resetea el incremento y se pasa al decremento de dicho bit. A continuación se explicarán las diferentes redes necesarias para su programación.

3. Programación

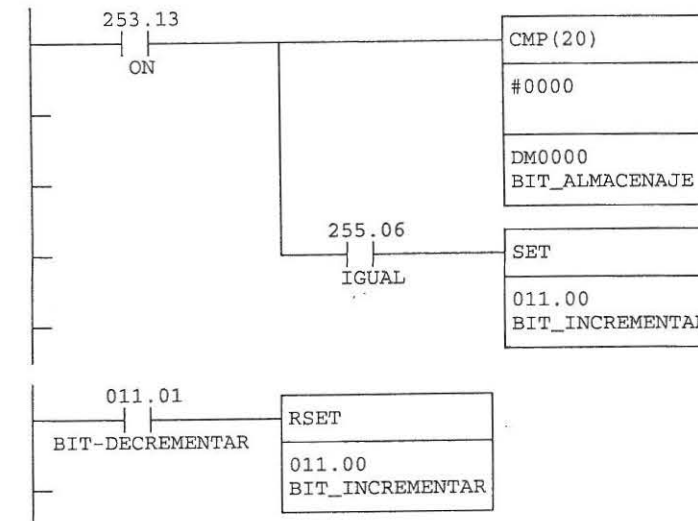
La primera red, hace uso del bit de "activación en el primer ciclo de ejecución" cuya dirección es "253.15". El cual hace que se cargue el valor inicial "0000" en el "BIT_ALMACENAJE", lo cual provoca como se verá en una red posterior que cuando comparamos dicho bit con el valor "0000" se activa el bit incremento y este resetea el decremento.



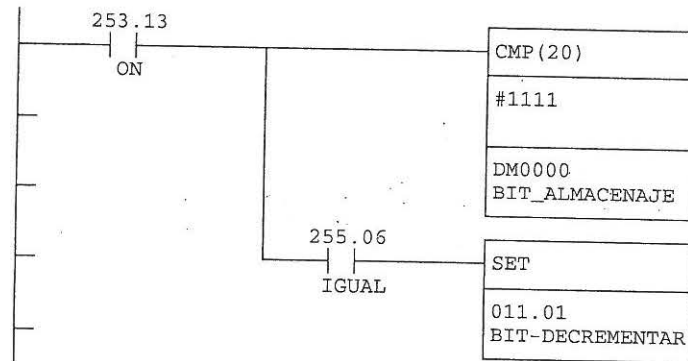
Estas dos redes son la encargadas de incrementar o decrementar en una unidad el valor del "BIT_ALMACENAJE" cada segundo, siempre y cuando este activo el "BIT_INCREMENTAR" y desactivado el "BIT_DECREMENTAR" en el caso del incremento y al revés e el otro. Este último es un poco redundante porque como se verá mas adelante cuando el "BIT_INCREMENTAR" esta a SET el otro esta a RESET.



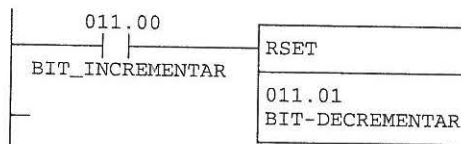
Esta red esta siempre activa y es la encargada de comparar el valor del "BIT_ALMACENAJE" con el valor "0000" y si es igual pone a SET (activo) el bit "BIT_INCREMENTAR".



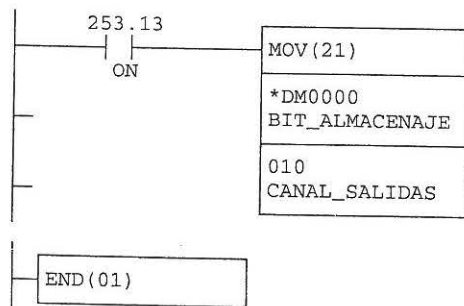
Esta red esta siempre activa y es la encargada de comparar el valor del "BIT_ALMACENAJE" con el valor "1111" y si es igual pone a SET (activo) el bit "BIT_DECREMENTAR".



Esta red desactiva el "BIT_DECREMENTAR" cuando se activa el "BIT_INCREMENTAR".



Esta red siempre activa, se encarga de pasar el valor de la dirección donde apunta el "BIT_ALMACENAJE" al canal de salidas.



15.19. ENCODER CON DETECCIÓN DEL SENTIDO DE GIRO

1. Objetivo

Este ejercicio permite usar diferentes instrucciones para afianzarlas y comprobar sus ejecuciones.

Se pretende usar "BSET", "INC", "DEC", "MOV", "DIFU", "DIFD", "KEEP", "MUL" y "DIV".

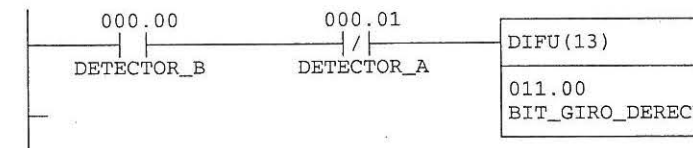
Se pretende realizar el funcionamiento de un encoder.

2. Planteamiento

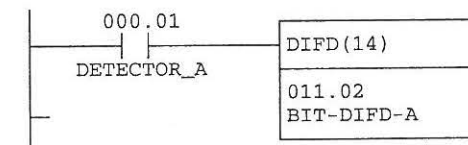
Se pretende realizar un encoder el cual actúa con dos sensores que nos indica el sentido de giro del encoder "izquierda" o "derecha".

3. Programación

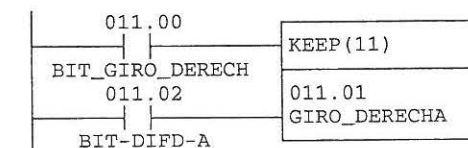
En esta red, se hace uso de la instrucción "DIFU". Para detectar los flancos de subida de la entrada "DETECTOR_B", siempre que no este el "DETECTOR_A".



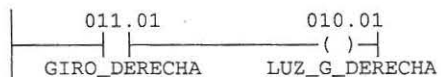
En esta red, se hace uso de la instrucción "DIFD". Para detectar los flancos de bajada de la entrada "DETECTOR_A".



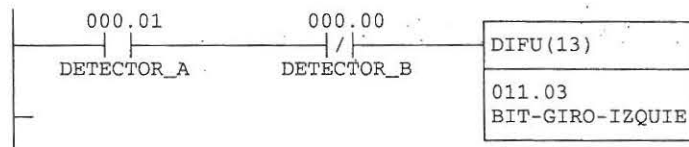
Para Mantener activo el bit "GIRO_DERECHA" lo hacemos mediante el bit "BIT-GIRO- DERECH" y cuando lo queremos desactivar lo realizamos mediante "BIT-DIFD- A".



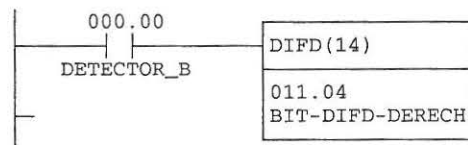
Cuando se activa la señal de "GIRO_DERECHA", se debe de activar la salida "LUZ_G_DERECHA" encargada de visualizar el movimiento hacia la izquierda.



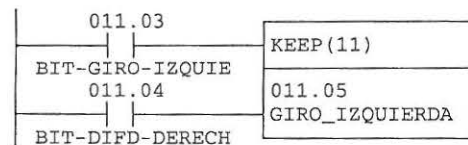
En esta red, se hace uso de la instrucción "DIFU". Para detectar los flancos de subida de la entrada "DETECTOR_A", siempre que no este el "DETECTOR_A".



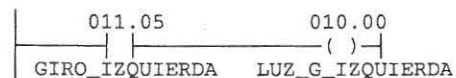
En esta red, se hace uso de la instrucción "DIFD". Para detectar los flancos de bajada de la entrada "DETECTOR_B".



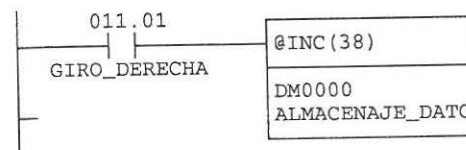
Para Mantener activo el bit "GIRO_IZQUIERDA" lo hacemos mediante el bit "BIT-GIRO-IZQUIE" y cuando lo queremos desactivar lo realizamos mediante "BIT-DIFD-DERECH".



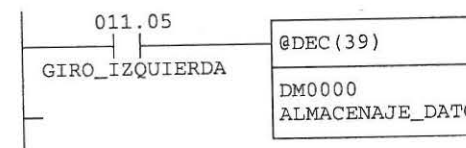
Cuando se activa la señal de "GIRO_IZQUIERDA", se debe de activar la salida "LUZ_G_IZQUIERDA" encargada de visualizar el movimiento hacia la izquierda.



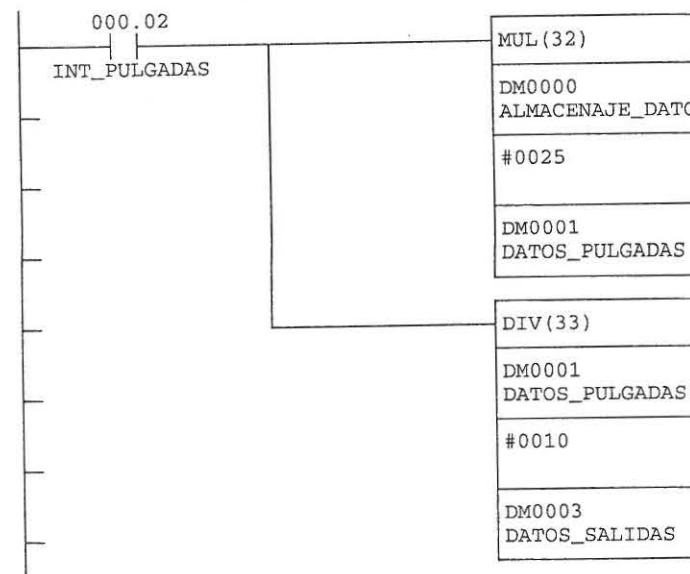
Cada vez que detecta un flanco en la entrada "GIRO_DERECHA" se incrementa en una unidad la palabra "ALMACENAJE_DATO".



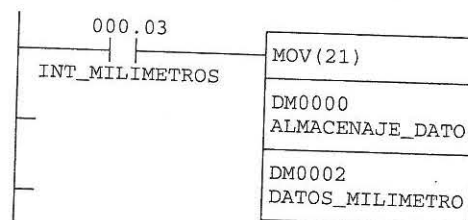
Cada vez que detecta un flanco en la entrada "GIRO_IZQUIERDA" se decrementa en una unidad la palabra "ALMACENAJE_DATO".



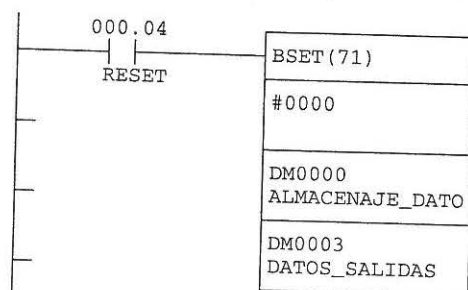
Cuando se pulsa la entrada "INT_PULGADAS" se mueve el canal de "ALMACENAJE_DATO" al canal "DATOS_SALIDAS". Pero para ello, debemos hacer algunas conversiones para sacar el resultado en pulgadas.



Cuando se pulsa la entrada "INT_MILIMETROS" se mueve el canal de "ALMACENAJE_DATO" al canal "DATOS_MILIMETRO".



Esta red, se encarga de realizar la puesta a cero de las palabras "ALMACENAJE_DATO", "DATOS_PULGADAS", "DATOS_MILIMETRO" y "DATOS_SALIDAS" cuando pulsamos la entrada reset.



END(01)

15.20. ACTIVACIÓN SECUENCIAL DE ETAPAS DE TRABAJO DE UNA LAVADORA

1. Objetivo

Este ejercicio permite usar diferentes instrucciones anteriormente utilizadas para afianzarlas y comprobar sus ejecuciones. Se pretende usar "BSET", "INC", "MOV", y "TIM" así como el bit interno del autómata de control de "primer ciclo de ejecución".

Se pretende simular el funcionamiento de una lavadora. En el cual mientras esta el botón de marcha comienzan los programas de la lavadora (centrifugado, lavado, secado,...) y van cambiando cada cierto tiempo.

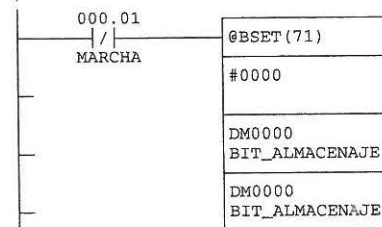
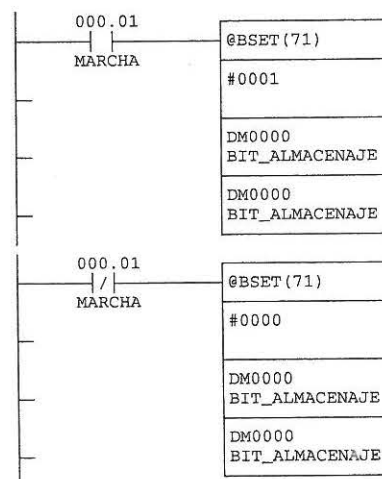
2. Planteamiento

Se pretende simular el funcionamiento de una lavadora, para ello cuando pulsemos la entrada de "MARCHA", cargaremos en el "BIT_ALMACENAJE" el valor "0001" y cuando no este la entrada cargaremos la "0000". Mientras este la entrada de "MARCHA" el temporizador esta en funcionamiento, este temporizador simularía los diferentes programas de la lavadora. Cada vez que se cumpla el temporizador incrementará en una unidad el valor del "BIT_ALMACENAJE" realizando la operación de cambio de programa en la lavadora. Por ultimo para visualizarlo se pasará donde apunta el "BIT_ALMACENAJE" al canal de salidas.

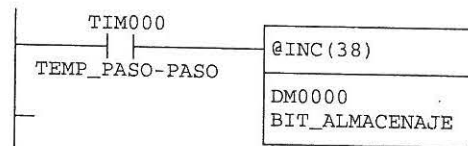
3. Programación

Caso 1.

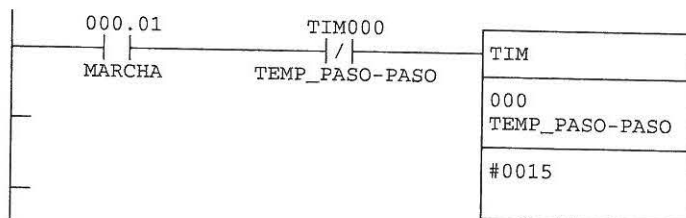
La primera red, realiza la carga del valor "0001" al "BIT_ALMACENAJE" siempre y cuando este activa la entrada "MARCHA". Si no esta activa la entrada "MARCHA" se realiza la carga del valor "0000", que seria la lavadora parada.



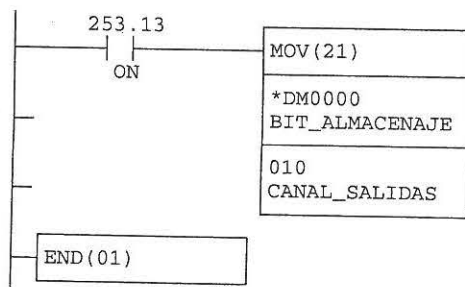
Cada vez que se cumple el temporizador, que equivaldría a la finalización de un programa de la lavadora (centrifugado, secado,...) activa la instrucción "INC" encargada de incrementar el "BIT_ALMACENAJE" que sería como la secuencia de programas a seguir en el proceso de lavado realizado por la lavadora.



Mientras este activa la entrada "MARCHA" y no este activo el temporizador, activo el temporizador para que realice la cuenta.



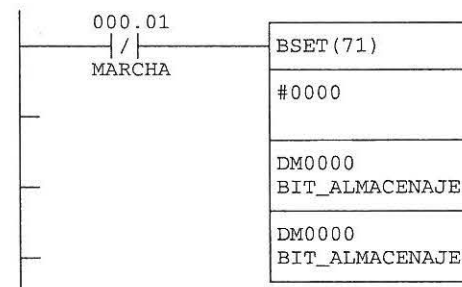
Por ultimo, se realiza el paso del valor de la dirección a la que apunta el "BIT_ALMACENAJE" al canal de salidas para poder visualizar los diferentes programas (centrifugado, lavado, secado,...) de la lavadora.



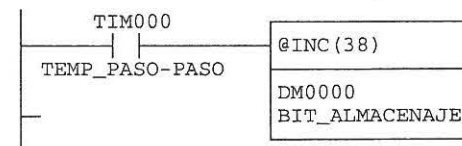
Caso 2.

La diferencia con el caso anterior es que antes de cargar el primer programa realice un retardo, por ejemplo para cargar agua.

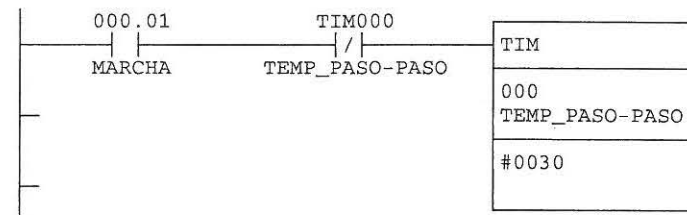
La primera red, realiza la carga del valor "0000" al "BIT_ALMACENAJE" siempre y cuando este desactive la entrada "MARCHA".



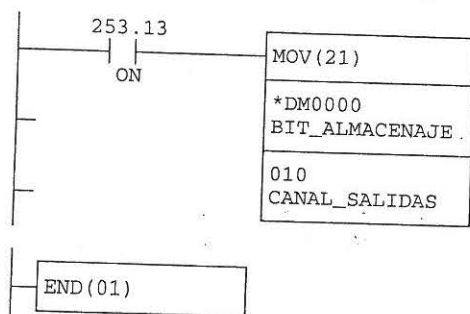
Cada vez que se cumple el temporizador, que equivaldría a la finalización de un programa de la lavadora (centrifugado, secado,...) activa la instrucción "INC" encargada de incrementar el "BIT_ALMACENAJE" que sería como la secuencia de programas a seguir en el proceso de lavado realizado por la lavadora.



Mientras este activa la entrada "MARCHA" y no este activo el temporizador, activo el temporizador para que realice la cuenta.



Por ultimo, se realiza el paso del valor de la dirección a la que apunta el "BIT_ALMACENAJE" al canal de salidas para poder visualizar los diferentes programas (centrifugado, lavado, secado,...) de la lavadora.



15.21. PROGRAMACIÓN DE LOS CICLOS DE UNA LAVADORA

1. Objetivo

Este ejercicio permite conocer el funcionamiento y la programación de los temporizadores mediante la función "TIM" así como el movimiento de un dato de una palabra otra mediante el comando "MOV".

2. Planteamiento

En el ejercicio anterior se activan los pasos del programa de una lavadora. En este ejercicio se asignan a cada uno de estos pasos las correspondientes salidas de mando. A cada paso del ciclo programado se le asocia un estado del puerto de salida contenido en una palabra de memoria. Se han supuesto 14 pasos distintos. Para cambiar las salidas asociadas a cada paso bastaría con cambiar el contenido de la palabra de memoria correspondiente. Se asignan los siguientes bits del canal de salidas:

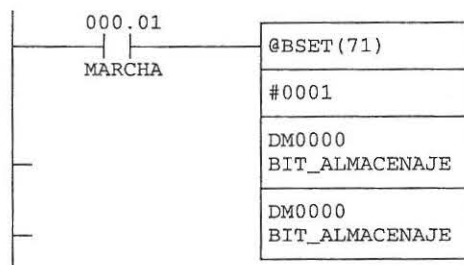
Mientras la entrada "MARCHA" no esté activa se mantendrá a cero la palabra que contiene el número de pasos realizados (BIT_ALMACENAJE). Al activar "MARCHA" pasaremos el valor 1 a la palabra BIT_ALMACENAJE. Simultáneamente activaremos el temporizador 0 con un tiempo de 1.5 seg. Al transcurrir este tiempo el BIT_ALMACENAJE se incrementará en una unidad con lo cual realizaremos el paso siguiente. Cuando hayamos realizado todos los pasos el ciclo de lavado habrá concluido con lo cual podemos desactivar la entrada "MARCHA" y el sistema volverá al estado inicial.

En todo momento pasamos el BIT_ALMACENAJE directamente a las salidas mediante la instrucción "MOV" de forma que las salidas se activarán de la siguiente manera:

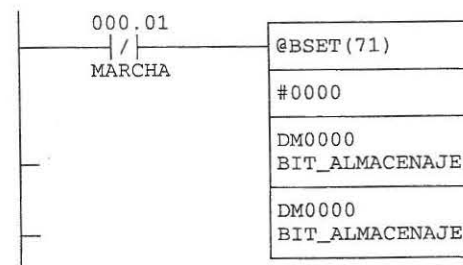
CANAL_SALIDAS (HEX)	CANAL_SALIDAS (BIT) (OUT16.. OUT1)
00	0000 0000 0000 0000
01	0000 0000 0000 0001
02	0000 0000 0000 0010
03	0000 0000 0000 0011
04	0000 0000 0000 0100
05	0000 0000 0000 0101
06	0000 0000 0000 0110
07	0000 0000 0000 0111
08	0000 0000 0000 1000
09	0000 0000 0000 1001
0A	0000 0000 0000 1010
0B	0000 0000 0000 1011
0C	0000 0000 0000 1100
0D	0000 0000 0000 1101
0E	0000 0000 0000 1110
0F	0000 0000 0000 1111
10	0000 0000 0001 0000
11	0000 0000 0001 0001
12	0000 0000 0001 0010
13	0000 0000 0001 0011

3. Programación

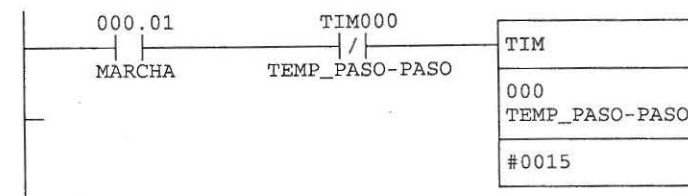
Al activar la entrada "MARCHA" transferimos el valor 1 a BIT_ALMACENAJE. Éste será el primer movimiento que realice la lavadora.



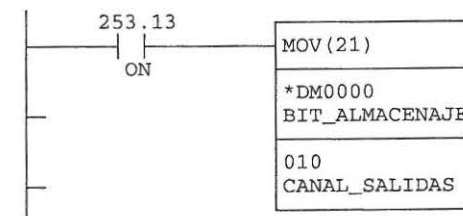
Al desactivar la entrada "MARCHA" transferimos un 0 a BIT_ALMACENAJE con lo cual todas las salidas estarán desactivadas.



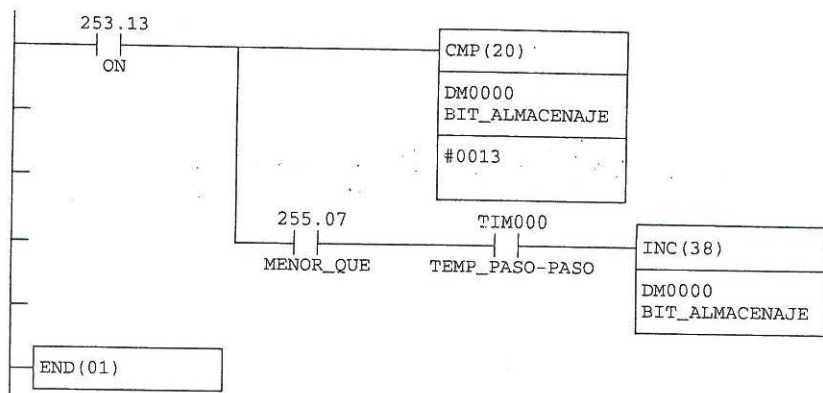
Mediante la temporización establecemos el tiempo que debe invertir la lavadora en realizar cada paso. La condición para que temporice es que "MARCHA" esté activo y cuando el tiempo haya transcurrido volverá a iniciarse la temporización. Cuando el tiempo ha transcurrido la variable asociada al temporizador se activa (en nuestro caso el temporizador cero tiene asociada la salida TIM000).



En todo momento transferimos el valor de BIT_ALMACENAJE a las salidas (CANAL_SALIDAS) activándolas según muestra la tabla anterior.



Mientras el número de pasos realizados (contenido en BIT_ALMACENAJE) sea inferior a 13 (en hexadecimal), cada vez que se produzca la temporización (TEMP_PASO-PASO), se incrementará en 1 el número de pasos.



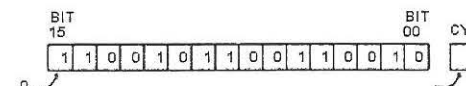
15.22. MOVIMIENTO DE PALABRAS Y DESPLAZAMIENTO DE BITS

1. Objetivo

En este ejercicio se hará uso de la función MOV para transferir datos a una zona de memoria así como de la función ASR de desplazamiento de bits. También se hará uso del temporizador TIM, de la función de INC y de la función de tablas BSET.

2. Planteamiento

Mientras el sistema esté detenido, es decir la entrada "MARCHA" este desactivada, transferiremos el estado de las entradas (CANAL_ENTRADAS) a la palabra "BIT-BCD". Al activar "MARCHA" desplazaremos los bits de la palabra "BIT-BCD" una posición a la derecha de la siguiente manera:



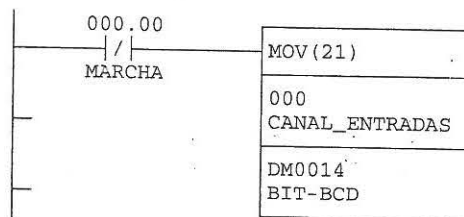
CODIFICACIÓN DECIMAL	CODIFICACIÓN BCD (BIT-BCD) ORIGINAL	CODIFICACIÓN BCD (BIT-BCD) ORIGINAL TRAS DESPLAZAMIENTO	CODIFICACIÓN DECIMAL
0	0000	0000	0
1	0001	0000	0
2	0010	0001	1
3	0011	0001	1
4	0100	0010	2
5	0101	0010	2
6	0110	0011	3
7	0111	0011	3
8	1000	0100	4
9	1001	0100	4

Tras el desplazamiento transferimos "BIT-BCD" a la palabra "BIT_ALMACENAJE" cuyo contenido irá incrementándose en una unidad cada vez que se produzca la temporización determinada por el temporizador TIM000 (TEMP_PASO-PASO). En todo momento se transferirá el contenido de "BIT_ALMACENAJE" a las salidas (CANAL_SALIDAS). Al desactivar "MARCHA" se volverá a las condiciones iniciales.

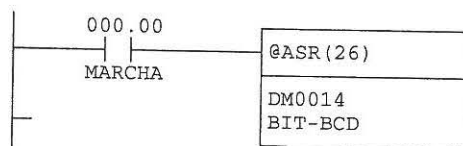
3. Programación

Caso 1.

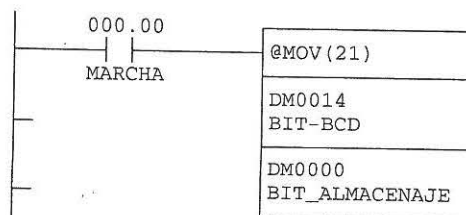
Mientras "MARCHA" esté desactivada transferiremos el estado de las entradas a la palabra "BIT-BCD".



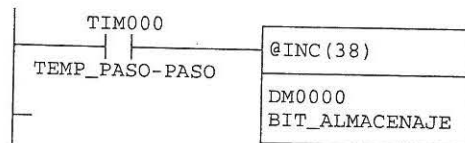
Al pulsar "MARCHA" desplazaremos los bits de "BIT-BCD" una posición a la derecha.



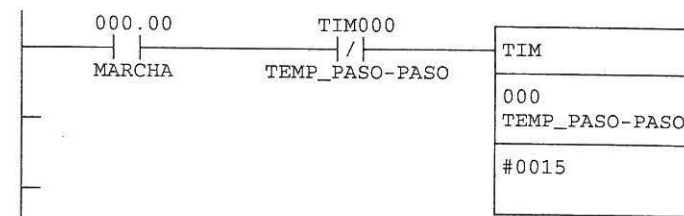
Transferimos el valor de "BIT-BCD" a la palabra "BIT_ALMACENAJE".



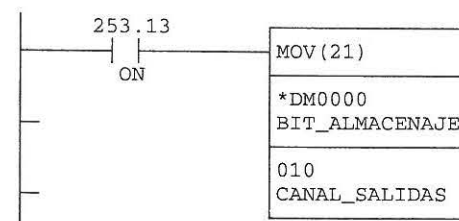
Cada vez que se produce la temporización se incrementa en una unidad el contenido de "BIT_ALMACENAJE".



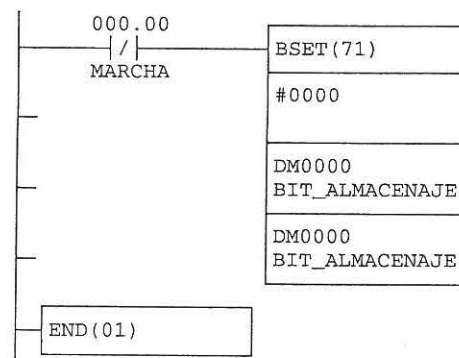
Mientras "MARCHA" esté activo se realiza la temporización. Cuando la temporización ha concluido se activa la variable asociada al temporizador (TEMP_PASO-PASO), en este momento la temporización se reinicia.



En todo momento transferimos la palabra "BIT_ALMACENAJE" a las salidas (CANAL_SALIDAS).



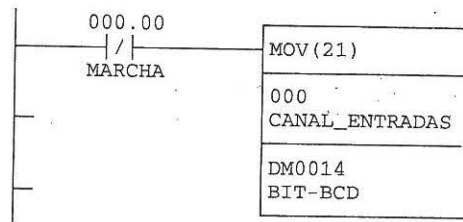
Cuando "MARCHA" está desactivada reiniciamos el contenido de "BIT_ALMACENAJE" transfiriéndole un 0.



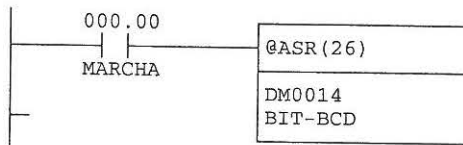
Caso 2.

Ésta es una variación del caso anterior; la diferencia es que si el sistema detecta que iniciamos el programa desde el valor 0 (BIT_BCD = 0) no debe realizar el primer paso.

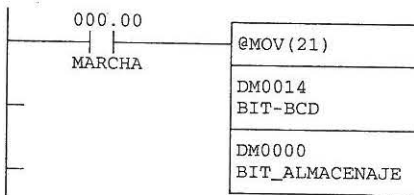
Mientras "MARCHA" esté desactivada transferiremos el estado de las entradas a la palabra "BIT-BCD".



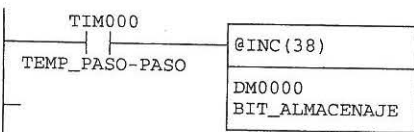
Al pulsar "MARCHA" desplazaremos los bits de "BIT-BCD" una posición a la derecha.



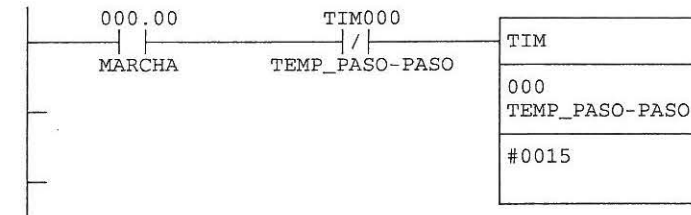
Transferimos el valor de "BIT-BCD" a la palabra "BIT_ALMACENAJE".



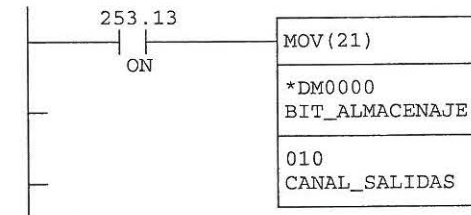
Cada vez que se produce la temporización se incrementa en una unidad el contenido de "BIT_ALMACENAJE".



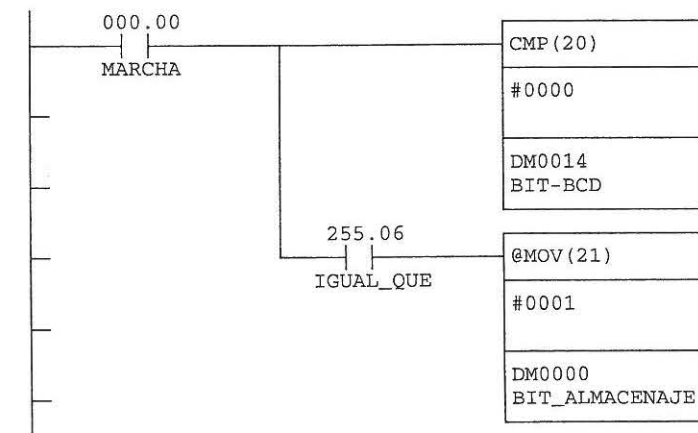
Mientras "MARCHA" esté activo se realiza la temporización. Cuando la temporización ha concluido se activa la variable asociada al temporizador (TEMP_PASO-PASO), en este momento la temporización se reinicia.



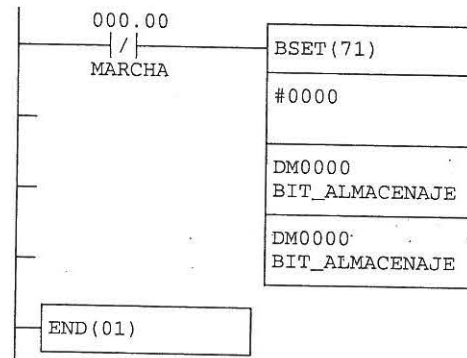
En todo momento transferimos la palabra "BIT_ALMACENAJE" a las salidas (CANAL_SALIDAS).



Comparamos si "BIT-BCD" es 0, en caso afirmativo incrementamos en una unidad el valor de "BIT_ALMACENAJE".



Cuando "MARCHA" está desactivada reiniciamos el contenido de "BIT_ALMACENAJE" transfiriéndole un 0.



15.23. SIMULACIÓN DE UN TALADRO AUTOMÁTICO

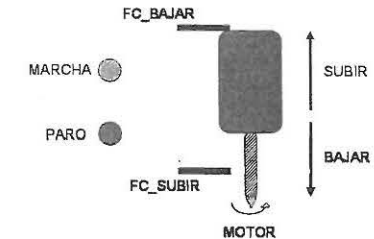
1. Objetivo

Comprender el funcionamiento de la instrucción KEEP (enclavamiento) y del temporizador (TIM) en un posible ejemplo real de un taladro.

2. Planteamiento

Inicialmente partiremos de las condiciones siguientes: el "MOTOR" del taladro estará parado y el taladro estará situado arriba (FC_BAJAR activo).

Al pulsar "MARCHA" se activará la salida "MOTOR" con lo cual el taladro empezará a girar. La salida "BAJAR" también se activará comenzando así el proceso de taladro.



Cuando la herramienta alcance su posición máxima de descenso (FC_SUBIR) se activará la temporización (TEMP_SUBIDA). Transcurrido este tiempo se activará la salida "SUBIR" con lo cual la herramienta subirá hasta que llegue a "FC_BAJAR"; en este momento el "MOTOR" del taladro deberá dejar de girar.

Si durante el proceso de taladrado se activa "PARO" se deben detener todos los movimientos y el taladro quedará bloqueado.

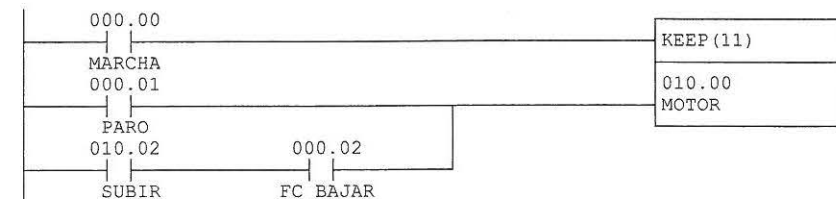
3. Programación

Caso 1

Red Activación del motor.

Condición de SET: Al activar "MARCHA" se activa la salida "MOTOR" (giro del taladro). Esta salida permanecerá activa aunque se desactive "MARCHA" y no se produzca la condición de RESET.

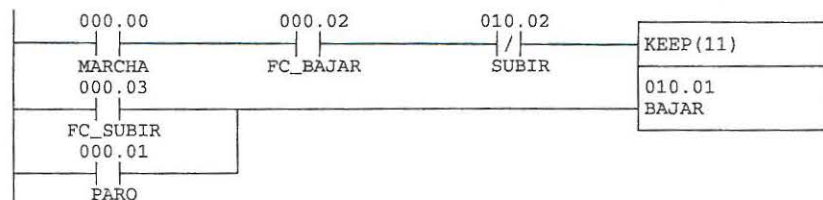
Condición de RESET: Si pulsamos "PARO" o la herramienta está subiendo (SUBIR) y llega al final del movimiento de ascenso (FC_BAJAR) la salida "MOTOR" se desactivará (el taladro se detiene). Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Red Orden de bajada.

Condición de SET: Al activar "MARCHA" y se cumplen las condiciones iniciales (herramienta en posición inicial -FC_BAJAR- y no está subiendo -SUBIR desactivado-) la salida "BAJAR" se activará con lo cual comenzará el descenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

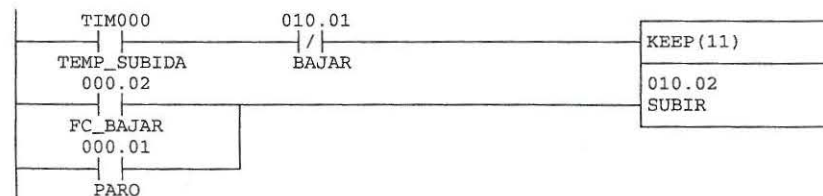
Condición de RESET: Si pulsamos "PARO" o la herramienta ha llegado a "FC_SUBIR" la salida "BAJAR" se desactivará y la herramienta detendrá su descenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



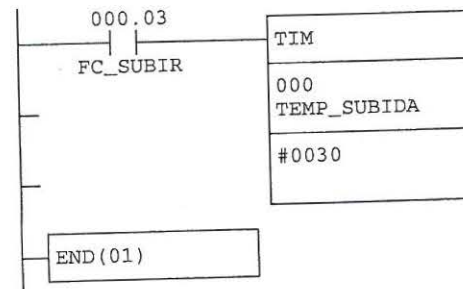
Red Orden de subida.

Condición de SET: Cuando la herramienta ha terminado su descenso (BAJAR) permanecerá en esta posición hasta que termine la temporización (TEMP_SUBIDA), en este momento la salida "SUBIR" se activará con lo cual comenzará el ascenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

Condición de RESET: Si pulsamos "PARO" o la herramienta ha llegado a "FC_BAJAR" la salida "SUBIR" se desactivará y la herramienta detendrá su ascenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Cuando la herramienta ha descendido completamente (FC_SUBIR activo) debe permanecer en esta posición durante 3 segundos y sólo podrá comenzar el ascenso cuando este tiempo se haya cumplido, es decir, "TEMP_SUBIDA" se active. En el momento que "FC_SUBIDA" se desactive la temporización volverá a su estado inicial.



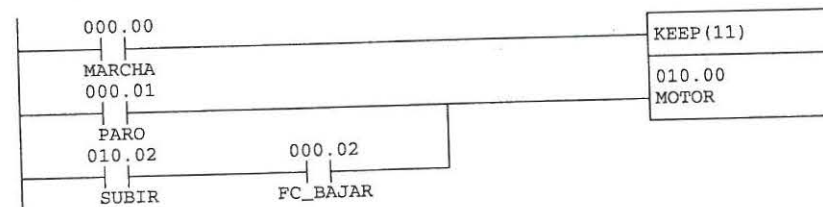
Caso 2.

Cumple las mismas especificaciones que el caso anterior con la variación siguiente: si durante el ciclo de taladrado se pulsa el "PARO" al volver a pulsar "MARCHA" el sistema vuelve a las condiciones iniciales de forma automática.

Red Activación del motor.

Condición de SET: Al activar "MARCHA" se activa la salida "MOTOR" (giro del taladro). Esta salida permanecerá activa aunque se desactive "MARCHA" y no se produzca la condición de RESET.

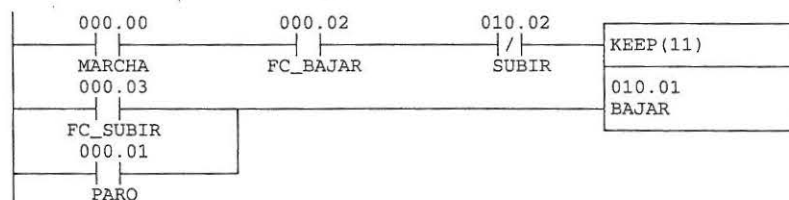
Condición de RESET: Si pulsamos "PARO" o la herramienta está subiendo (SUBIR) y llega al final del movimiento de ascenso (FC_BAJAR) la salida "MOTOR" se desactivará (el taladro se detiene). Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Red Orden de bajada.

Condición de SET: Al activar "MARCHA" y se cumplen las condiciones iniciales (herramienta en posición inicial -FC_BAJAR- y no está subiendo -SUBIR desactivado-) la salida "BAJAR" se activará con lo cual comenzará el descenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

Condición de RESET: Si pulsamos "PARO" o la herramienta ha llegado a "FC_SUBIR" la salida "BAJAR" se desactivará y la herramienta detendrá su descenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.

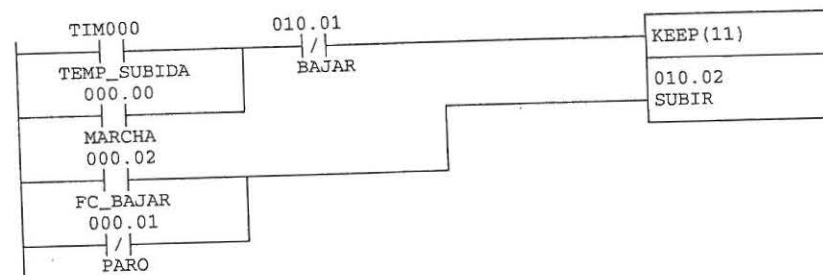


Red Orden de subida.

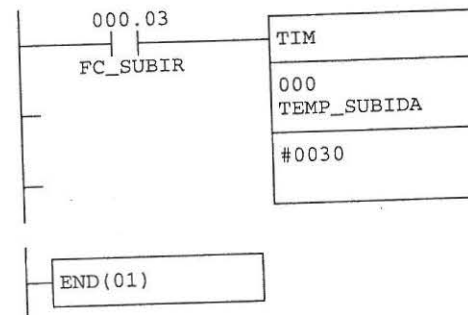
Condición de SET:

- Opción 1: Cuando la herramienta ha terminado su descenso (BAJAR) permanecerá en esta posición hasta que termine la temporización (TEMP_SUBIDA), en este momento la salida "SUBIR" se activará con lo cual comenzará el ascenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.
- Opción 2: Si durante el ciclo se produce el "PARO" todas las salidas se desactivarán (MOTOR, SUBIR y BAJAR) y la temporización se detendrá. Al volver a activar "MARCHA" la salida "SUBIR" se activará con lo cual comenzará el ascenso de la herramienta hasta la posición inicial. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

Condición de RESET: Si pulsamos "PARO" o la herramienta ha llegado a "FC_BAJAR" la salida "SUBIR" se desactivará y la herramienta detendrá su ascenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Cuando la herramienta ha descendido completamente (FC_SUBIR activo) debe permanecer en esta posición durante 3 segundos y sólo podrá comenzar el ascenso cuando este tiempo se haya cumplido, es decir, "TEMP_SUBIDA" se active. En el momento que "FC_SUBIDA" se desactive la temporización volverá a su estado inicial.



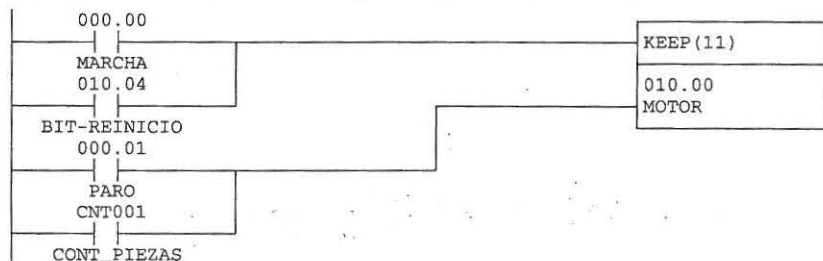
Caso 3.

Cumple las mismas especificaciones que el caso anterior con la variación siguiente: el ciclo se repite 20 veces.

Red Activación del motor.

Condición de SET: Al activar "MARCHA" o no haya completado todos los ciclos (BIT-REINICIO a 1) se activa la salida "MOTOR" (giro del taladro). Esta salida permanecerá activa mientras no se produzca la condición de RESET.

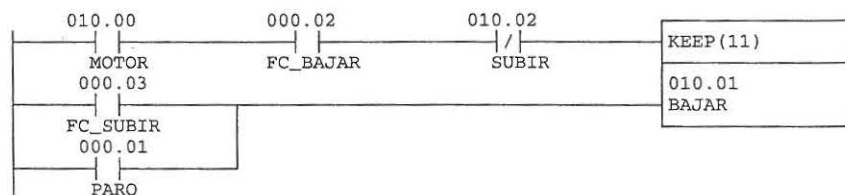
Condición de RESET: Si pulsamos "PARO" o hemos completado todos los ciclos la salida "MOTOR" se desactivará (el taladro se detiene). Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Red Orden de bajada.

Condición de SET: Al activar "MARCHA" y se cumplen las condiciones iniciales (herramienta en posición inicial -FC_BAJAR- y no está subiendo -SUBIR desactivado-) la salida "BAJAR" se activará con lo cual comenzará el descenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

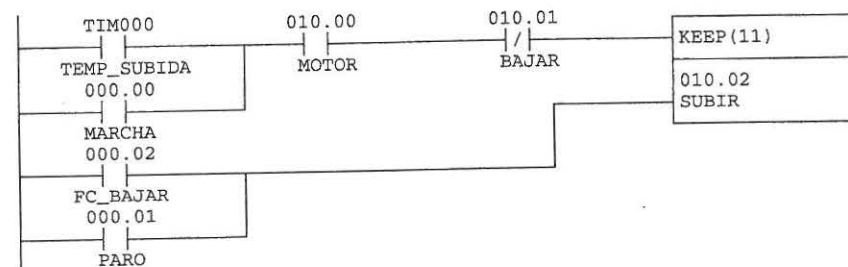
Condición de RESET: Si pulsamos "PARO" o la herramienta ha llegado a "FC_SUBIR" la salida "BAJAR" se desactivará y la herramienta detendrá su descenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



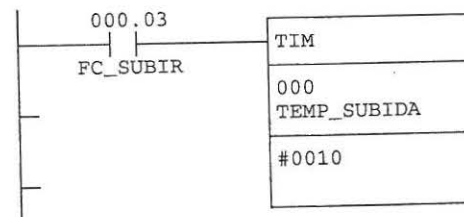
Red Orden de subida.

Condición de SET: Cuando la herramienta ha terminado su descenso (BAJAR) permanecerá en esta posición hasta que termine la temporización (TEMP_SUBIDA) y mientras el "MOTOR" esté activo la salida "SUBIR" se activará con lo cual comenzará el ascenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

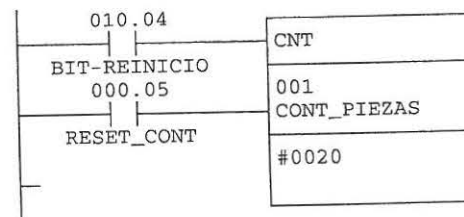
Condición de RESET: Si pulsamos "PARO" o la herramienta ha llegado a "FC_BAJAR" la salida "SUBIR" se desactivará y la herramienta detendrá su ascenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Cuando la herramienta ha descendido completamente (FC_SUBIR activo) debe permanecer en esta posición durante 3 segundos y sólo podrá comenzar el ascenso cuando este tiempo se haya cumplido, es decir, "TEMP_SUBIDA" se active. En el momento que "FC_SUBIDA" se desactive la temporización volverá a su estado inicial.



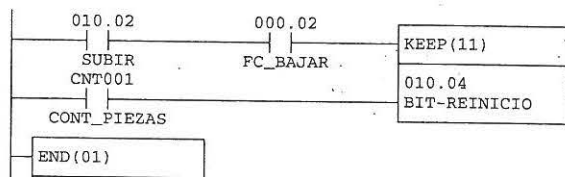
Cada vez que se activa "BIT-REINICIO" (comienza un nuevo ciclo) el sistema aumenta en una unidad el contenido de "CONT_PIEZAS", de esta forma se sabe cuántos ciclos ha realizado hasta el momento. Con la entrada "RESET_CONT" podemos poner a cero la cuenta del contador.



Red Reinicio.

Condición de SET: Una vez se haya completado un ciclo el taladro comienza a subir, en este momento se activa "BIT-REINICIO" que es la condición para aumentar la cuenta del contador de ciclos.

Condición de RESET: Cuando el contador llegue a su máxima cuenta (20 ciclos) se activará la salida asociada al contador 1 (CNT001) que hace el RESET de "BIT-REINICIO".



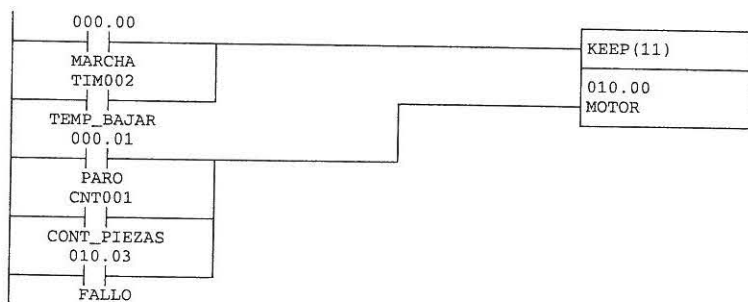
Caso 4.

Cumple las mismas especificaciones que el caso anterior con las siguientes variaciones: se activará una señal de error (FALLO) si tarda más de 30 segundos en realizar el ciclo (BAJAR-SUBIR), si están activados simultáneamente "FC_BAJAR" y "FC_SUBIR" o si el motor no está girando (MOTOR) y estamos realizando el ciclo (BAJAR-SUBIR).

Red Activación del motor.

Condición de SET: Al activar "MARCHA" o al finalizar el tiempo para iniciar el descenso (TEMP_BAJAR) se activa la salida "MOTOR" (giro del taladro). Esta salida permanecerá activa mientras no se produzca la condición de RESET.

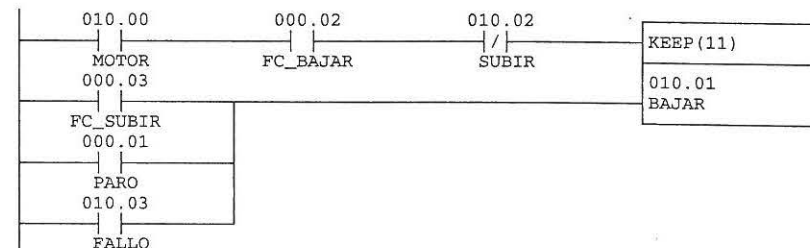
Condición de RESET: Si pulsamos "PARO", hemos completado todos los ciclos la salida "MOTOR" se desactivará (el taladro se detiene) o se produce la condición de "FALLO". Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Red Orden de Bajada.

Condición de SET: Mientras gire el motor (MOTOR) y se cumplen las condiciones iniciales (herramienta en posición inicial -FC_BAJAR- y no está subiendo -SUBIR desactivado-) la salida "BAJAR" se activará con lo cual comenzará el descenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

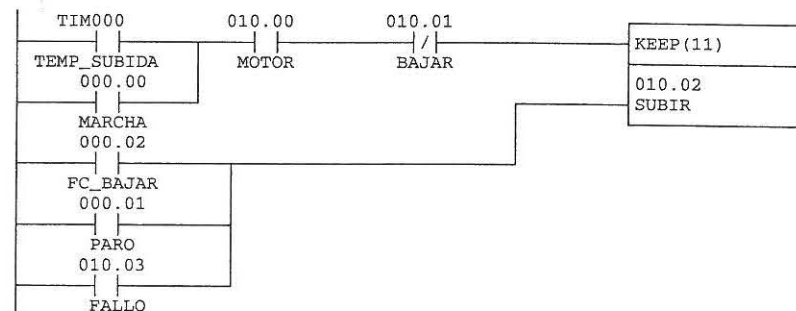
Condición de RESET: Si pulsamos "PARO", la herramienta ha llegado a "FC_SUBIR" o se produce un "FALLO" la salida "BAJAR" se desactivará y la herramienta detendrá su descenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



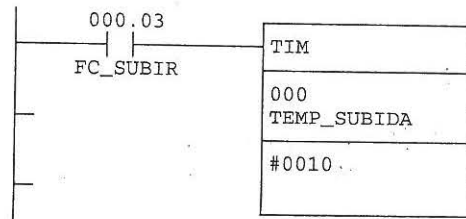
Red Orden Subida.

Condición de SET: Cuando la herramienta ha terminado su descenso (BAJAR) permanecerá en esta posición hasta que termine la temporización (TEMP_SUBIDA) o se active "MARCHA" y mientras el "MOTOR" esté activo la salida "SUBIR" se activará con lo cual comenzará el ascenso de la herramienta. Esta salida permanecerá activa mientras no se produzca la condición de RESET.

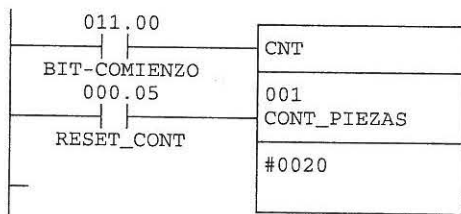
Condición de RESET: Si pulsamos "PARO", la herramienta ha llegado a "FC_BAJAR" o se produce un "FALLO" la salida "SUBIR" se desactivará y la herramienta detendrá su ascenso. Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



Cuando la herramienta ha descendido completamente (FC_SUBIR activo) debe permanecer en esta posición durante 1 segundo y sólo podrá comenzar el ascenso cuando este tiempo se haya cumplido, es decir, "TEMP_SUBIDA" se active. En el momento que "FC_SUBIDA" se desactive la temporización volverá a su estado inicial.



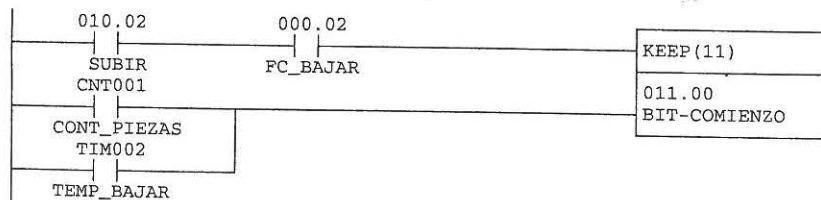
Cada vez que se activa "BIT-REINICIO" (comienza un nuevo ciclo) el sistema aumenta en una unidad el contenido de "CONT_PIEZAS", de esta forma se sabe cuántos ciclos ha realizado hasta el momento. Con la entrada "RESET_CONT" podemos poner a cero la cuenta del contador.



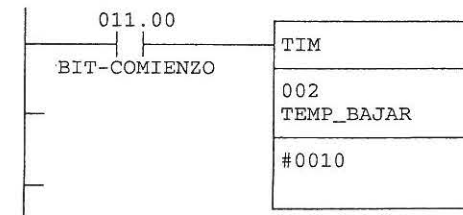
Red de Comienzo de ciclo.

Condición de SET: Una vez se haya completado un ciclo el taladro comienza a subir, en este momento se activa "BIT-COMIENZO" que es la condición para aumentar la cuenta del contador de ciclos.

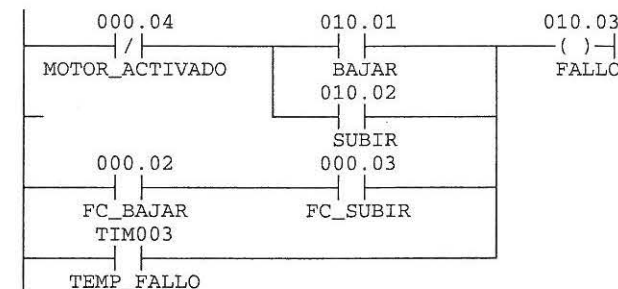
Condición de RESET: Cuando el contador llegue a su máxima cuenta (20 ciclos) o haya concluido la temporización para el descenso (TEMP_BJAR). Esta salida permanecerá desactivada mientras no se produzca la condición de SET.



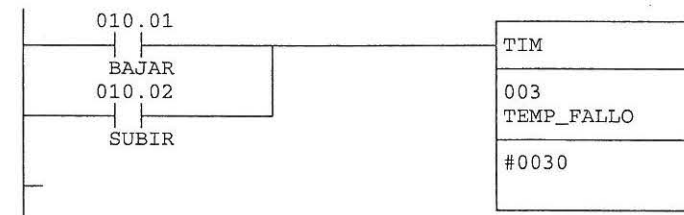
Una vez se inicie un nuevo ciclo (BIT-COMIENZO se activa cuando termina el ascenso y llega al FC_BAJAR) debe permanecer en esta posición durante 1 segundo y sólo podrá comenzar el ascenso cuando este tiempo se haya cumplido, es decir, "TEMP_BAJAR" se active. En el momento que "BIT-COMIENZO" se desactive la temporización volverá a su estado inicial.



La salida "FALLO" se activa si estamos en el ciclo (BAJAR o SUBIR) y el motor deja de girar (MOTOR_ACTIVADO), o se activan simultáneamente los dos finales de carrera (FC_BAJAR y FC_SUBIR) o se rebasa el tiempo máximo para realizar el ciclo (TEMP_FALLO).



Si estamos realizando el ciclo (SUBIR o BAJAR) y tarda más de 3 segundos en hacer el movimiento se activa la variable asociada al temporizador 3 (TEMP_FALLO).



15.24. PROGRAMACIÓN SECUNECIAL DE UN TALADRO

1. Objetivo

Esta variación sobre el ejercicio del taladro permite mostrar una nueva manera de enfocar el diseño de los ejercicios de una forma más estructurada y secuencial. De esta forma la programación se convierte en una serie de subprogramas anidados unos a otros en el que la activación de una variable permite la activación de la etapa o etapas siguientes.

2. Planteamiento

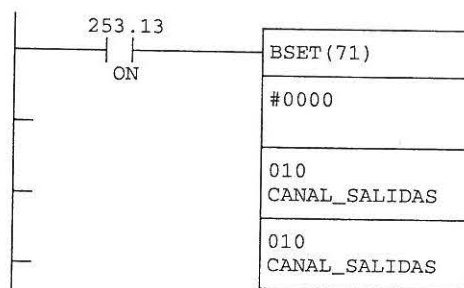
Los pasos a seguir serán los siguientes:

- En el primer ciclo de trabajo todas las salidas se pondrán a cero.
- Si no existe ningún movimiento realizándose se activará una variable que indicara el estado de "REPOSO".
- Estando en "REPOSO" si activamos "MARCHA" comienza a girar la carga para ponerla en la posición adecuada.
- Cuando la pieza está en la posición adecuada (FC_PIEZA) comienza el giro y el descenso del taladro.
- Una vez el taladro ha descendido completamente (FC_SUBIDA), tras un tiempo de espera el taladro comienza a subir.
- El ascenso termina cuando el taladro llega a "FC_BAJADA" y la pieza debe girar (GIRO_PIEZA) hasta que esté en la posición requerida para ser extraída.

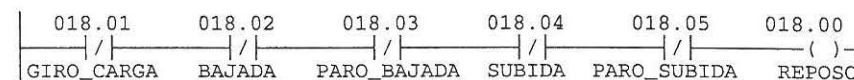
3. Programación

Caso 1.

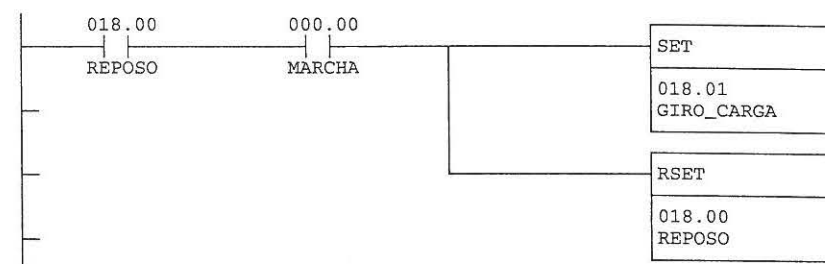
En el primer ciclo de trabajo se desactivan todas las salidas (CNAL_SALIDAS).



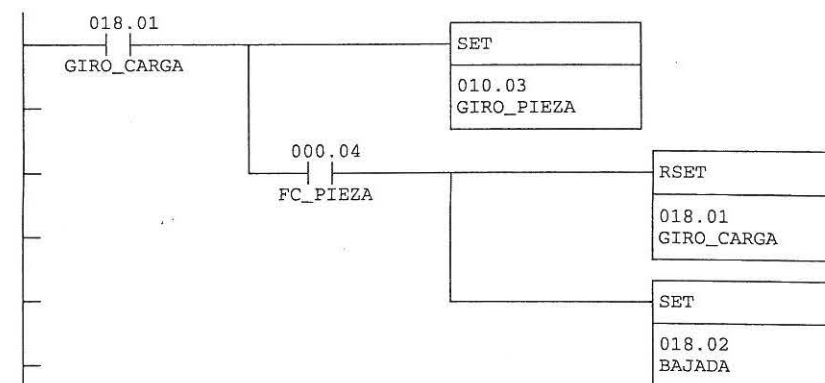
Si no se realiza ningún movimiento se activa la variable "REPOSO", indicando al sistema que se puede iniciar el ciclo.



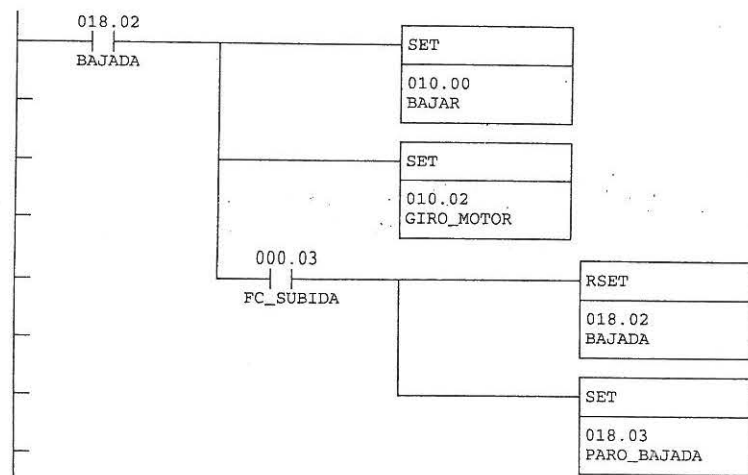
Estando en "REPOSO" si activamos "MARCHA" se activa "GIRO_CARGA" y se desactiva "REPOSO".



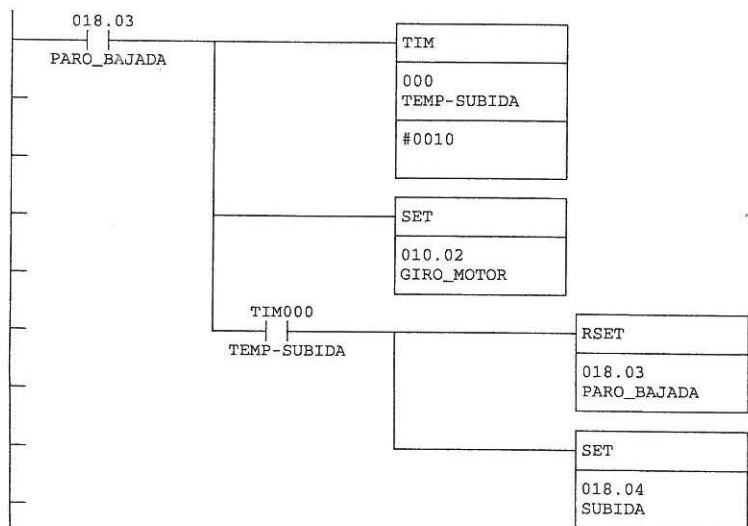
La activación de "GIRO_CARGA" activa el giro de la pieza. En el momento que la pieza esté colocada en la posición adecuada (FC_PIEZA) desactivamos "GIRO_CARGA" y activamos "BAJADA".



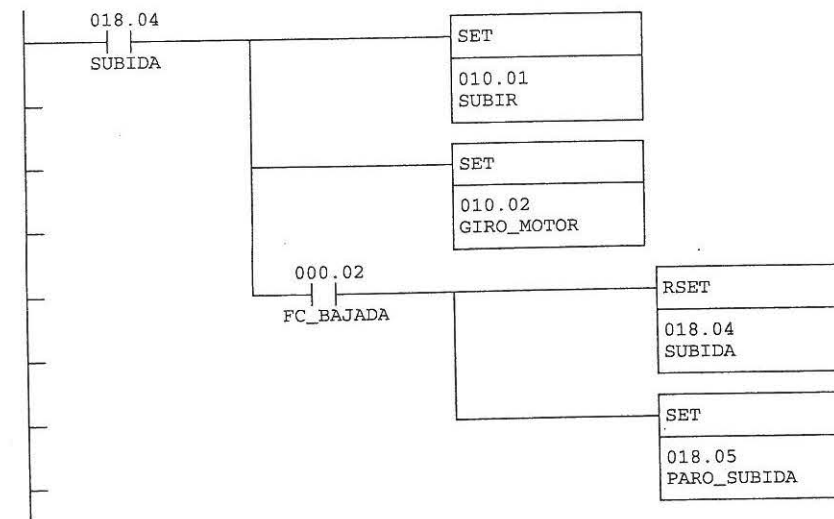
La activación de "BAJADA" inicia el giro y el descenso del taladro. El descenso termina cuando la herramienta llega a "FC_SUBIDA", en este momento se activa "PARO_BAJADA".



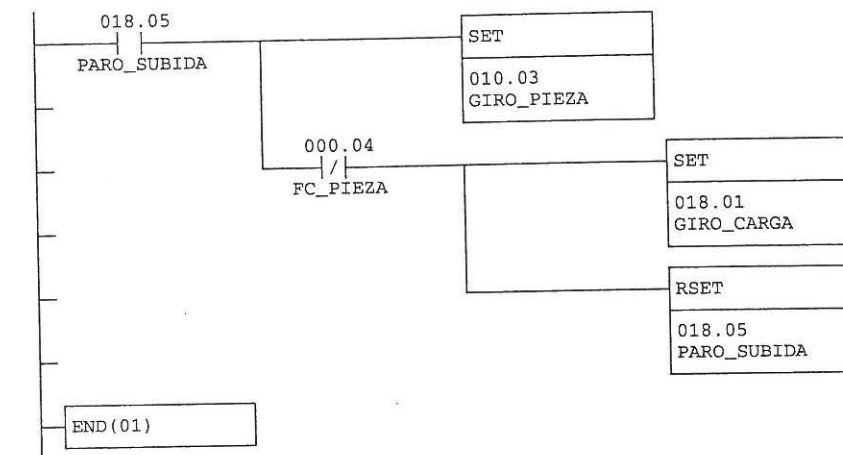
La activación "PARO_BAJADA" inicia el giro del motor y la temporización para iniciar la subida. Cuando se cumple el tiempo de espera se activa la variable "SUBIDA".



La activación de "SUBIDA" activa el giro del motor y el proceso de ascensión del taladro. Una vez el taladro ha llegado a "FC_BAJADA" se desactiva "SUBIDA" y se activa "PARO_SUBIDA".



La activación de "PARO_SUBIDA" activa "GIRO_PIEZA". Cuando "FC_PIEZA" se desactive se desactivará "PARO_SUBIDA" y se activará "GIRO_CARGA" con lo cual el ciclo volverá a iniciarse.



Caso 2.

Las especificaciones son las del caso anterior, pero añadiendo:

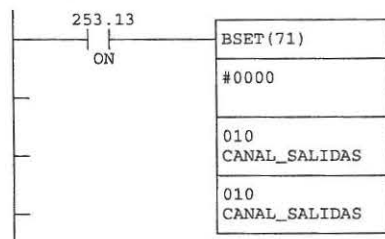
- Un pulsador de paro, cuya función es la siguiente:

Si se pulsa paro una vez el taladro debe quedarse en pausa permitiendo dos opciones:

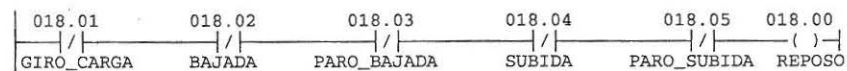
- Si se pulsa marcha continuar donde estaba trabajando.
- Si se pulsa paro de nuevo volver a la posición de inicio (arriba).

- Un bloqueo de la máquina en caso de detección de fallo o anomalía en su funcionamiento.

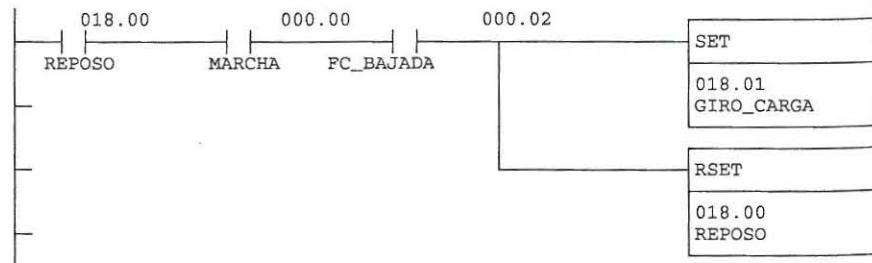
Se pone inicialmente a cero el canal de salidas (010).



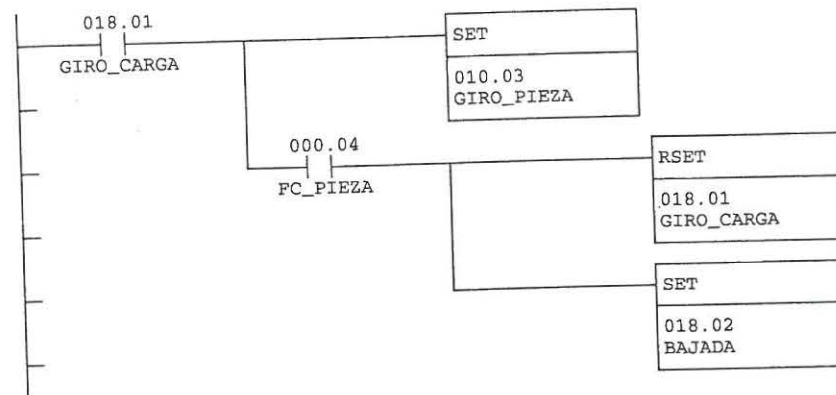
En este bloque se indica si el taladro se encuentra en pausa o no dependiendo del estado de los bits GIRO_CARGA, BAJADA, PARO_BAJADA, SUBIDA, PARO_SUBIDA.



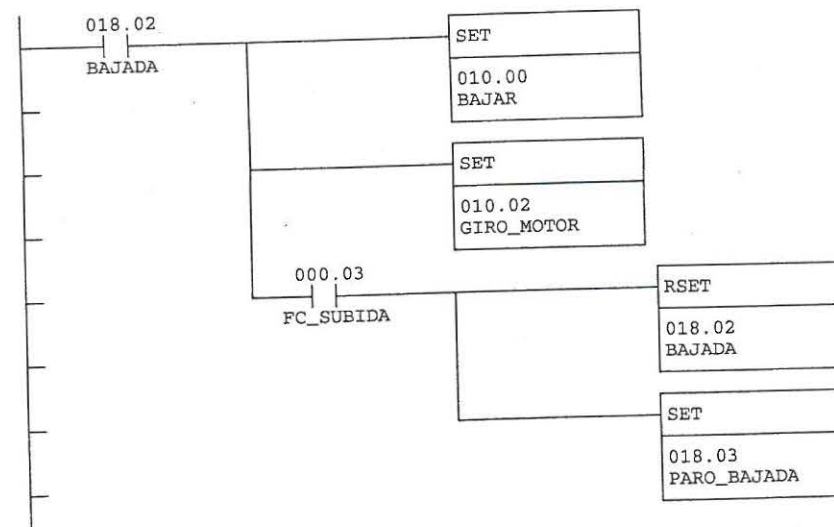
Si el taladro se encuentra en reposo, activado el bit (018.00), y el FC_BAJADA (000.02) activado al pulsar marcha se activa el bit GIRO_CARGA (018.01) y se desactiva el bit de reposo (018.00).



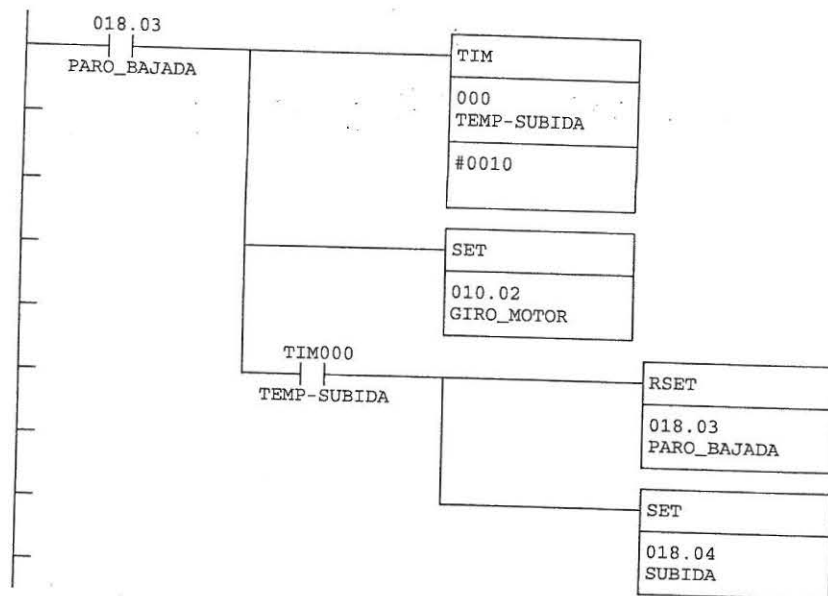
Al estar activo el bit GIRO_CARGA (018.01) se activa la salida GIRO_PIEZA (010.03) con lo que la pieza gira hasta detectar el FC_PIEZA (000.04), momento en el que se desactiva el bit GIRO_CARGA (018.01) y se activa el de BAJADA (018.02).



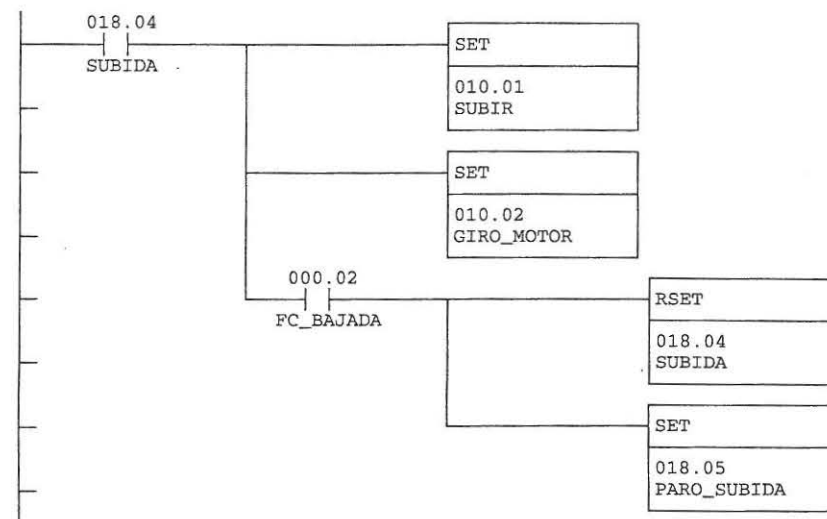
Al activarse el bit BAJADA (018.02) permite que se active la salida BAJAR (010.00) y al mismo tiempo el GIRO_MOTOR (010.02), con lo cual el taladro empieza a bajar girando la herramienta (broca). Cuando el taladro activa la entrada FC_SUBIDA (000.03) se desactiva el bit BAJADA (018.02) y se activa el bit PARO_BAJADA (018.03), indicando que ha dejado de bajar el taladro.



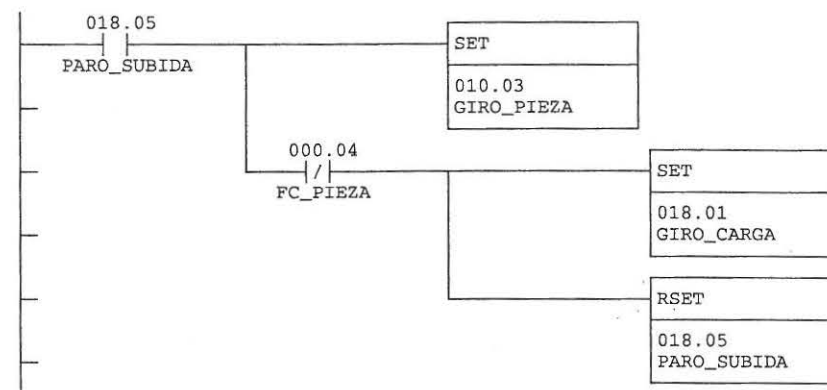
Al estar activo el bit PARO_BAJADA (018.03) se activa un temporizador el cual temporiza el tiempo que el taladro está taladrando, pasado este tiempo el taladro subirá gracias a la activación de los bits correspondientes, y al mismo tiempo se activa la salida GIRO_MOTOR (010.02) siguiendo este girando. Al pasar el tiempo de subida se desactiva el bit PARO_BAJADA (018.03) y se activa el bit SUBIDA (018.04).



Activado este bit SUBIDA (018.04) se activa la salida SUBIR (010.01) y también la salida GIRO_MOTOR (010.02) siguiendo este girando. Cuando el taladro llega arriba activando el FC_BAJADA (000.02) se desactiva el bit SUBIDA (018.04) y se activa al mismo tiempo el bit PARO_SUBIDA (018.05).

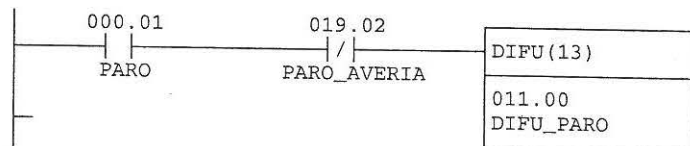


La activación del bit PARO_SUBIDA (018.05) permite activar la salida GIRO_PIEZA (010.03), al girar esta el FC_PIEZA se desactiva y esto permite que se active el bit GIRO_CARGA (018.01) y se desactive el bit PARO_SUBIDA (018.05), con lo cual la máquina queda de nuevo en reposo y lista par realizar otro ciclo.

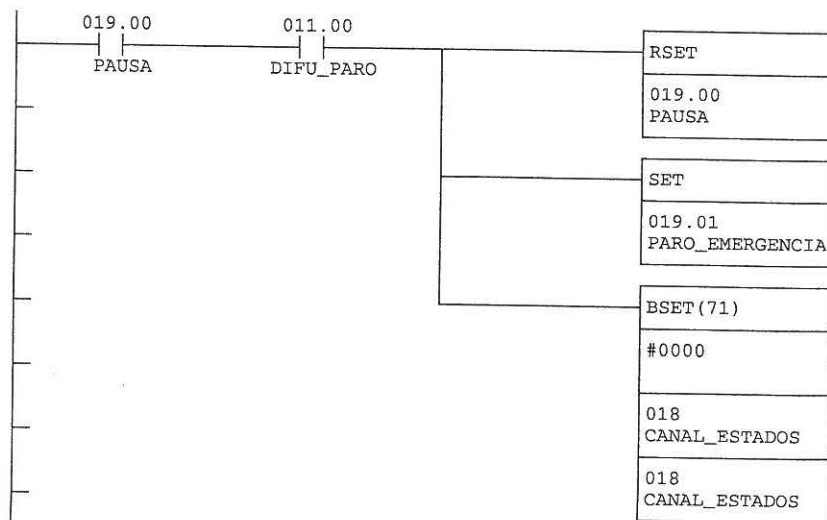


En los siguientes bloques se procede al paro del taladro como especifica el enunciado.

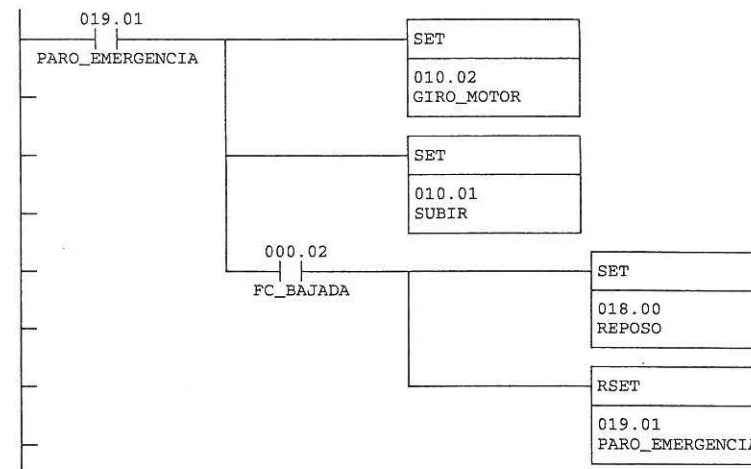
En este bloque si el bit PARO_AVERIA (019.02) no está activo y se pulsa PARO se activa el bit DIFU_PARO (011.00).



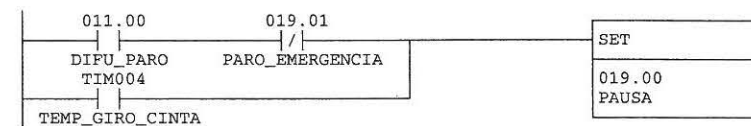
En el caso de que el taladro se encuentre en PAUSA (019.00) y el bit DIFU_PARO (011.00) esté activo, es decir, se ha pulsado PARO, se produce la desactivación del bit PAUSA (019.00) y al mismo tiempo se activa el bit PARO_EMERGENCIA (019.01) y se pone todo a ceros el canal (018).



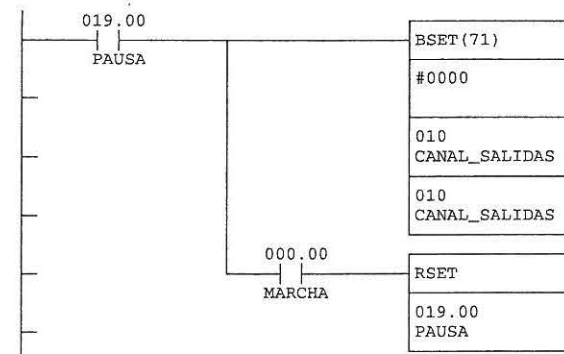
La activación del bit PARO_EMERGENCIA provoca que se activen las salidas GIRO_MOTOR (010.02) y SUBIR (010.01), subiendo de este modo el taladro a la posición inicial hasta que es detectado el FC_BAJADA (000.02), momento en el que se activa el bit de REPOSO (018.00) y al mismo tiempo se desactiva el bit PARO_EMERGENCIA (019.01).



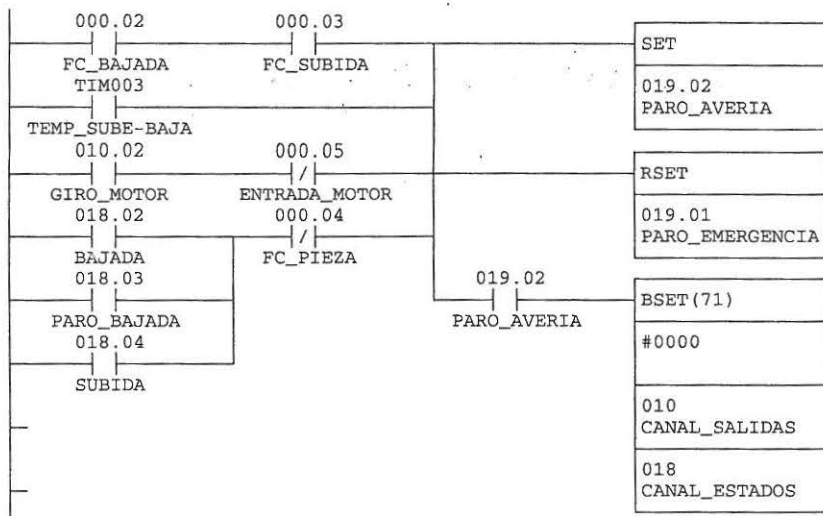
En el caso de que se pulse PARO y el bit PARO_EMERGENCIA no esté activo, o que se de el TEMP_GIRO_CINTA (TIM004) se activa la posición de memoria PAUSA (019.00).



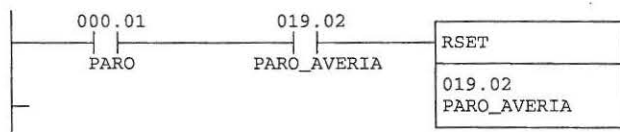
Al activarse el bit de pausa el canal de salidas (010) pasa a valer todo cero, con lo que todas las salidas están desactivadas y el taladro completamente parado, en PAUSA. Si por consiguiente se pulsa de nuevo MARCHA (000.00) el bit de PAUSA (019.00) queda desactivado y la máquina continúa trabajando donde se había quedado.



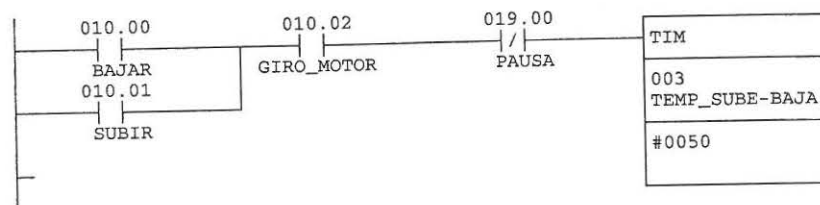
En este bloque como se puede apreciar se dan las diferentes condiciones que se pueden presentar para un paro de emergencia de la máquina. Que estén los dos finales de carrera activados FC_BAJADA y FC_SUBIDA. Del mismo modo cuando empieza a subir o a bajar se activa un temporizador TEMP_SUBE_BAJA (TIM0003) el cual si pasado este tiempo el taladro no ha parado se activa produciendo la parada de emergencia. Del mismo modo si sube o baja y el final de carrera pieza no está activado también provoca el paro de emergencia.



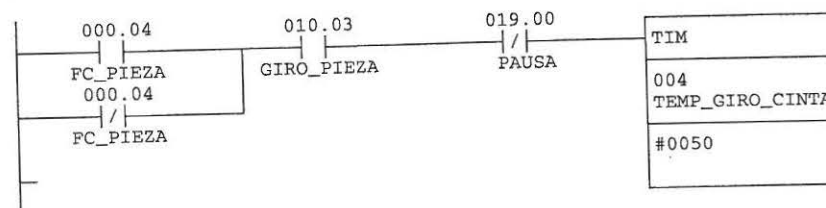
En el siguiente bloque al pulsar paro y estar activado el bit PARO_AVERIA (019.02) se resetea ese mismo bit.



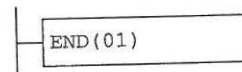
En el caso de que el taladro esté bajando o subiendo y esté girando la herramienta (broca), se activa el temporizador que me indica el tiempo máximo que este debe estar bajando o subiendo. Como se ha podido apreciar en un bloque anterior, si pasado este tiempo no está parado, se produce una parada de emergencia.



En el caso de que se active el final de carrera de la pieza FC_PIEZA (000.04) y la entrada GIRO_PIEZA (010.03) esté activa y el bit PAUSA (019.00) no lo esté, se activa el temporizador que me indica el tiempo máximo que este debe estar bajando o subiendo. Como se ha podido apreciar en un bloque anterior, si pasado este tiempo no está parado, se produce una parada de emergencia.



La siguiente función indica el final de programa y es necesaria para que pueda funcionar el programa transferido al autómata.



15.25. TEMPORIZADORES: INTERMITENCIAS

1. Objetivo

Entre las funciones que se dispone en todos los autómatas programables, existe una muy importante que es la del temporizador. En este ejercicio se trata de poner en práctica el temporizador con retardo a la conexión, funcionando en modo intermitente.

2. Planteamiento

En este ejemplo se van a estudiar el temporizador de retardo a la conexión, para entender con mayor claridad su funcionamiento se puede ver el cronograma correspondiente, pero además sería el que utiliza los sistemas de alarma para una vez conectada, esta, nos permita un tiempo para salir del recinto de actuación de la alarma y transcurrido este tiempo quedaría activada. Posteriormente en sucesivos apartados de este ejercicio se desarrollará la combinatoria necesaria para obtener los modos de trabajo del resto de temporizadores más usuales.

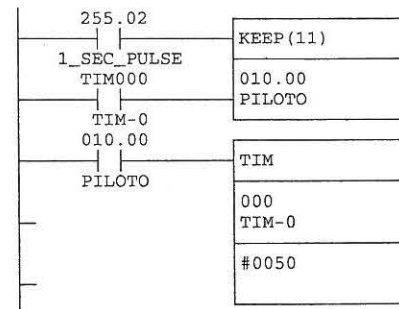
En un segundo ejemplo se diseña el temporizador que se conoce como retardo a la desconexión como ejemplo de uso sería el que se emplearía en la automatización de un freno o bloqueo de un elevador. Es decir al activar el motor se desbloquea el freno, al parar el motor el freno o bloqueo no actúa de forma inmediata si no transcurrido un tiempo, para que se pare por inercia o reducción de velocidad.

El tercer ejemplo de este ejercicio el temporizador a la conexión, la forma de funcionamiento se deduce del tercer cronograma.

El cuarto ejercicio de este ejemplo es el temporizador a la desconexión, la salida se encuentra activa siempre que la entrada no esté activa y además al activar la entrada no se desactiva de forma inmediata la salida si no transcurrido un tiempo.

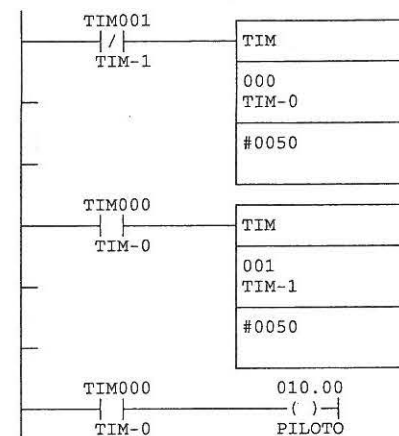
CONVERTIR UN TREN DE IMPULSOS EN UNA ONDA RECTANGULAR

Mediante el bit (255.02) se consigue que cada segundo se produzca un pulso. Cuando se produce este pulso la salida PILOTO (010.00) se activa y esto provoca que se active el temporizador TIM000 de 50 segundos el cual cuando realiza la temporización se encarga de desactivar la salida PILOTO (010.00). Este ciclo se repite indefinidamente consiguiendo una onda rectangular.



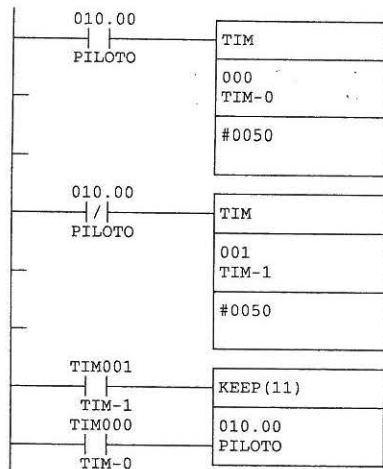
ONDA RECTANGULAR MEDIANTE DOS TEMPORIZADORES

Al estar el TIM001 cerrado en un principio se consigue que el TIM000 empiece su temporización de 50 segundos. Una vez realizada este TIM000 activa el TIM001 y este empieza su temporización provocando al mismo tiempo que la salida PILOTO (010.00) se active. Cuando el TIM001 finalice la temporización de 50 segundos se desactiva el TIM000 con lo cual la salida PILOTO (010.00) se desactiva y provoca también que se desactive el TIM001 provocando que el TIM000 vuelva a realizar la temporización, repitiéndose el ciclo continuamente.



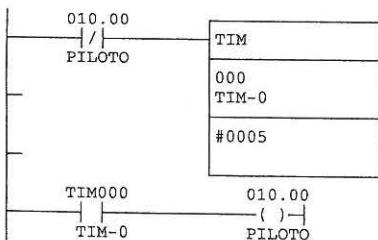
ONDA RECTANGULAR MEDIANTE DOS TEMPORIZADORES

En un principio el TIM001 empieza su temporización de 50 segundos. Una vez finalizada provoca la activación de la salida PILOTO (010.00) con lo que se resetea el temporizador TIM001 y se activa la temporización del TIM000. Una vez realizada su temporización se desactiva la salida PILOTO (010.00) provocando que se desactive el TIM000 y se active la temporización del TIM001. El ciclo se repite seguidamente consiguiendo el efecto deseado.



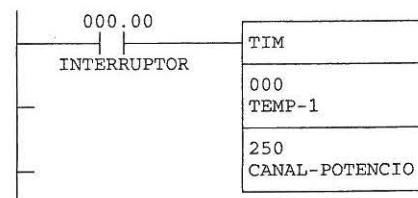
CONEXIÓN Y DESCONEXIÓN CON UN TEMPORIZADOR (INTERMITENCIA)

En un principio al estar la salida PILOTO (010.00) desactivada el TIM000 empieza su temporización de 5 segundos. Una vez realizada esta temporización el TIM000 activa la salida PILOTO (010.00) con lo cual la entrada que controla el temporizador abre su contacto reseteando dicho temporizador y provocando esto que la salida PILOTO (010.00) se desactive, consiguiendo así que la entrada que controla el temporizador se cierre y este vuelva a temporizar. Este ciclo se repite continuamente consiguiendo el efecto intermitente.

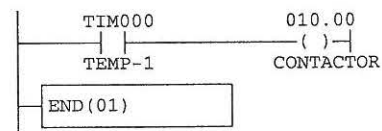


REALIZAR TEMPORIZADOR REGULABLE EXTERIORMENTE (A TRAVÉS DEL POTENCIOMETRO EXTERIOR)

Mediante el INTERRUPTOR (000.00) se consigue controlar el temporizador. Si el INTERRUPTOR está activo el temporizador cuenta, si está desactivo no cuenta. Por medio de un potenciómetro externo conectado al canal 250 de la memoria se consigue regular el tiempo de la temporización a realizar.

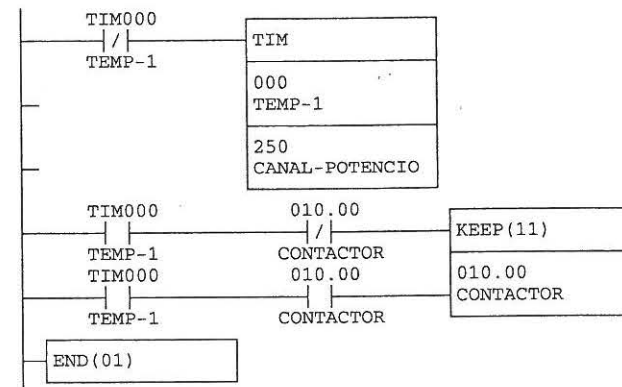


Una vez realizada la temporización activa la salida CONTACTOR (010.00).



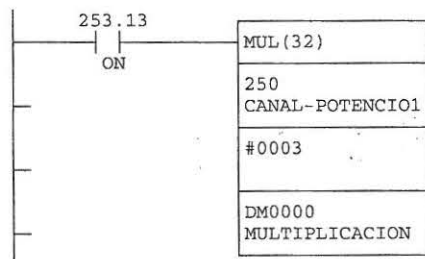
REALIZAR UN TREN DE IMPULSOS REGULABLE EXTERIORMENTE

En un principio el TIM000 empieza a realizar la temporización, siendo esta la correspondiente según la posición del potenciómetro externo que controla el tiempo. Una vez realizada esta temporización se activa la salida CONTACTOR (010.00) y al activarse la salida automáticamente se desactiva y se vuelve a activar. Esto se realiza mediante la función KEEP(11), y se repite sucesivamente.

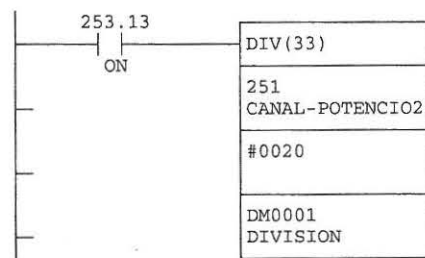


REGULAR UNA TEMPORIZACIÓN A TRAVÉS DE LOS TEMPORIZADORES EXTERIORES, PERO HACIENDO QUE UNO DE ELLOS REGULE EN SEGUNDOS Y EL OTRO EN DECIMAS

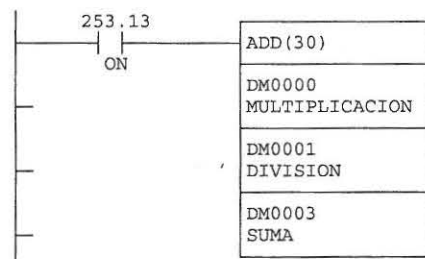
En este bloque se realiza la multiplicación entre el canal donde se guarda el valor del potenciómetro 1 (250) por 3, guardando el resultado en el DM0000.



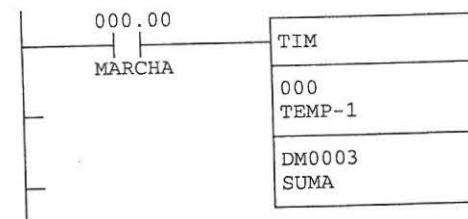
En el bloque siguiente realiza la división entre el canal donde se guarda el valor del potenciómetro 2 (251) entre 20, guardando el resultado en el DM0001.



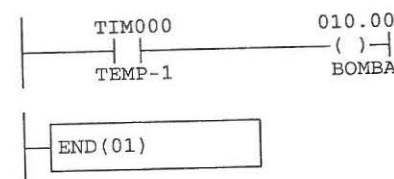
La función ADD(30) realiza la suma entre el valor guardado en el DM0000 del resultado de la multiplicación y el valor guardado en el DM0001 del resultado de la división, guardando el resultado en el DM0003.



Al pulsar marcha se activa el temporizador TIM000, realizando como temporización el valor guardado en el DM0003, resultado de la suma anterior.



Una vez el temporizador a realizado ya la temporización por completo se activa la salida BOMBA (010.00).



15.26. TEMPORIZADORES: RETARDO A LA CONEXIÓN Y DESCONEXIÓN

1. Objetivo

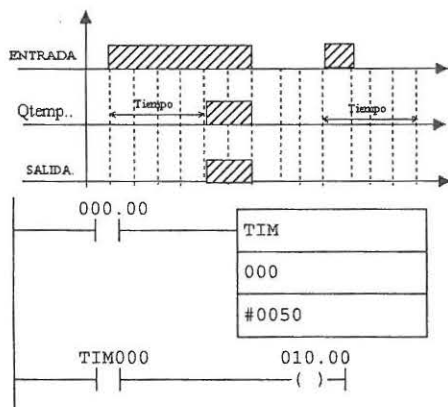
Entre las funciones que se dispone en todos los autómatas programables, existe una muy importante que es la del temporizador. Es tan importante que es usual que se pueda elegir su programación de directa, tal como se hace con las entradas y las salidas. Cada tipo, marca o modelo de autómata puede llevar diferentes clases de temporizadores. No obstante el que llevan todos los autómatas es el que se conoce como retardo a la conexión. Tomaremos este como base y se deducirán otros modos de actuar el temporizador. Hay modelo de autómata que puede llevar implementados varios temporizadores que actúan de forma distinta, estos equipos facilitan la aplicación de cualquiera de ellos. No obstante, interesa saber como desde un temporizador base se puede obtener otro, dado que a veces el que se necesita para una aplicación concreta no se tiene.

2. Planteamiento

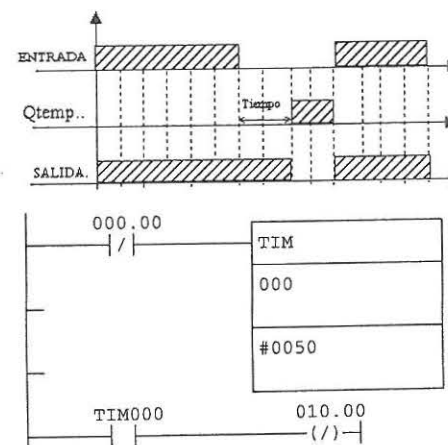
Se plantean diferentes retardos a la conexión y desconexión de una señal basados en el único tipo de temporizador disponible en la mayoría de los autómatas, el temporizador a la conexión.

3. Programación

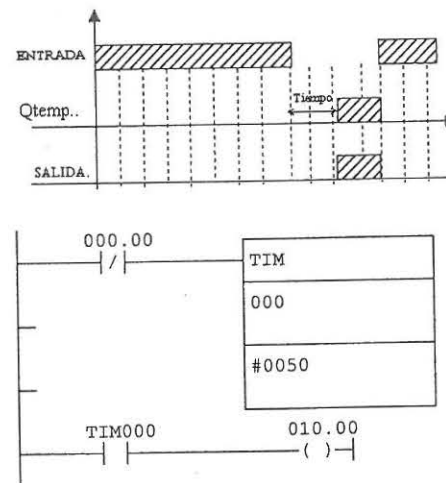
3.1. Retardo a la conexión



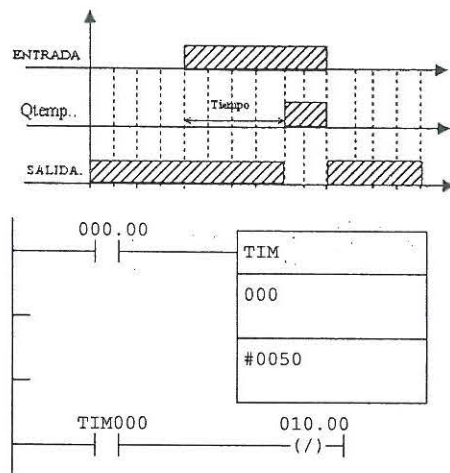
3.2. Retardo a la desconexión



3.3. Conexión temporizada a la pérdida de señal



3.4. Desconexión temporizada



15.27. ACCIONAR UN PUNTO DE LUZ DESDE TRES INTERRUPTORES

1. Planteamiento

Cada vez que un cualquiera de los tres interruptores cambie de estado ($0 \rightarrow 1$, o $1 \rightarrow 0$), el punto de luz cambiará de estado. Se desea asimismo que este estado se memorice, aunque el autómata se desconecte temporalmente.

ASIGNACIÓN DE ENTRADAS Y SALIDAS

Bit	Identificador	Descripción
0.0	INT1	Interruptor nº 1
0.1	INT2	Interruptor nº 2
0.2	INT3	Interruptor nº 3
5.0	ON_P1	Cambio $0 \rightarrow 1$ del interruptor 1
5.1	OFF_P1	Cambio $1 \rightarrow 0$ del interruptor 1
5.2	ON_P2	Cambio $0 \rightarrow 1$ del interruptor 2
5.3	OFF_P2	Cambio $1 \rightarrow 0$ del interruptor 2
5.4	ON_P3	Cambio $0 \rightarrow 1$ del interruptor 3
5.5	OFF_P3	Cambio $1 \rightarrow 0$ del interruptor 3
5.6	CAMBIO	Ha habido algún cambio en algún interruptor
10.0	LUZ	Punto de luz a controlar
HR0.0	AUX_LUZ	Relé auxiliar con memoria

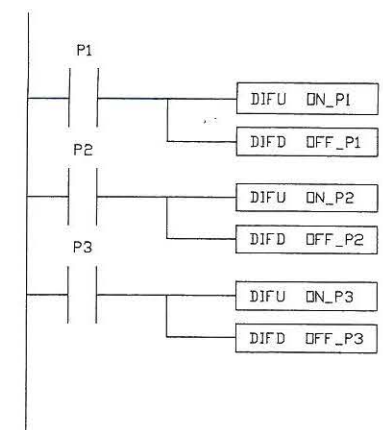
PROGRAMA

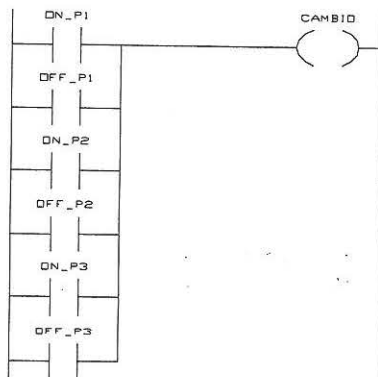
Detección de flancos de subida y bajada

Flancos de subida y bajada del interruptor 1.

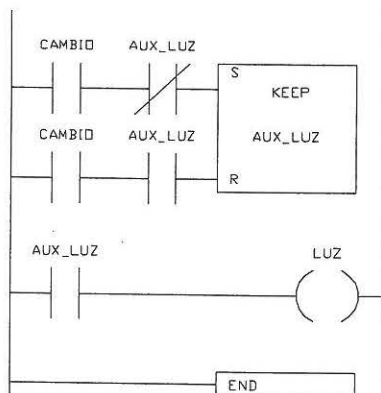
Flancos de subida y bajada del interruptor 2.

Flancos de subida y bajada del interruptor 3.



Determinación de la presencia de un flanco

El relé auxiliar CAMBIO se activa cuando ha habido un cambio de estado en cualquiera de los interruptores.

Activación del punto de luz con memorización a la desconexión del autómata

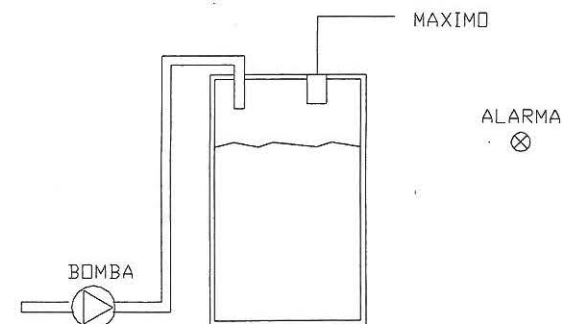
Cambio de estado del relé auxiliar (con memoria) con cada cambio de cualquier interruptor.

Activación del punto de luz.

Fin del programa.

15.28. MANTENIMIENTO DEL NIVEL DE UN DEPÓSITO**1. Descripción de la maniobra**

Se desea automatizar el llenado de un depósito, que cuenta con los siguientes elementos de control:



Bomba de llenado.

Detector de máximo, de tipo capacitivo, con un rango de detección de 20 mm.

Piloto de alarma.

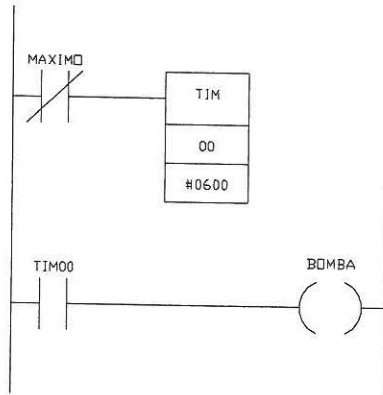
El depósito ha de mantenerse siempre al nivel máximo indicado por el detector de nivel, con las siguientes consideraciones:

1. El detector ha de permanecer siempre fuera del líquido.
2. Para evitar arranques y paradas intempestivas de la bomba, una vez que deje de funcionar habrá de pasar por lo menos 1 minuto antes de que vuelva a arrancar.
3. Si estando la bomba en marcha durante más de 10 minutos el nivel del depósito no alcanza el máximo, se supondrá que existe una avería o bien en la bomba o bien en el detector. Dicha avería se da por concluida cuando se recibe señal de máximo. Durante el tiempo en que esté presente se encenderá el piloto de alarma.

2. Asignación de entradas y salidas

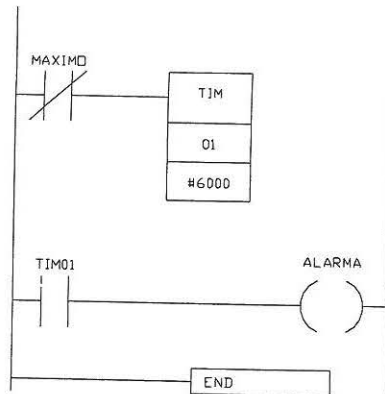
Bit	Identificador	Descripción
0.0	MÁXIMO	Detector de nivel máximo
10.0	BOMBA	Salida de mando de la bomba
10.1	ALARMA	Piloto de alarma
TIM00	TARRANQUE	Temporizador de arranque
TIM01	TALARMA	Temporizador de alarma

Accionamiento bomba



Si está más de 1 minuto sin detectar nivel máximo se da la orden de arranque a la bomba.

Alarma



Si está más de 10 minutos sin detectar nivel máximo se indica una alarma.

15.29. ACCIONAMIENTO DE UN EXPULSOR

1. Descripción de la maniobra

Se dispone de un expulsor de piezas defectuosas compuesto por los siguientes elementos:

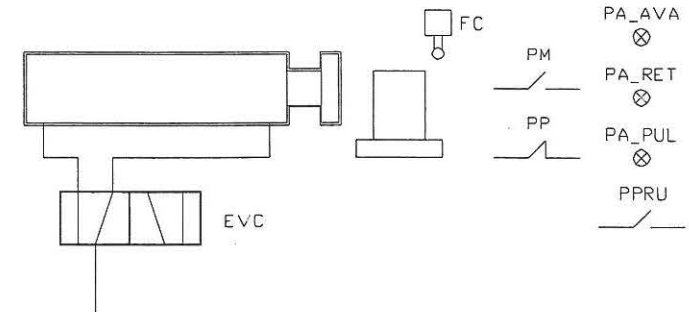
Cilindro de aire comprimido.

Electroválvula neumática de tres vías.

Final de carrera que limita la carrera del expulsor.

Pulsadores de marcha (PM), paro (PP) y prueba de pilotos (PPRU).

Pilotos de alarma de avance (PA_AVA), retroceso (PA_RET) y pulsadores (PA_PUL).



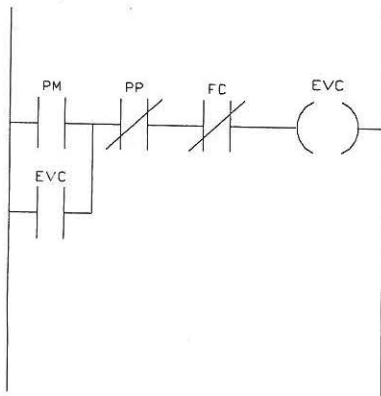
1. Si la pieza que pasa por delante del expulsor es defectuosa se acciona el pulsador de marcha (PM). Se activa la electroválvula EV, el cilindro sale y empuja la pieza fuera de la cinta transportadora. Al tocar el final de carrera se desactiva la electroválvula EV y el cilindro retrocede a su posición de reposo.
2. Si se acciona el pulsador de paro el cilindro debe volver inmediatamente a su posición de reposo.
3. Si el cilindro, con la electroválvula EV accionada más de 30 segundos, no alcanza el final de carrera, se encenderá un piloto de alarma del cilindro. El piloto se apagará tan pronto se alcance el final de carrera.
4. Si el final de carrera se mantiene pulsado más de 3 segundos, se encenderá un piloto de alarma del final de carrera. El piloto se apagará tan pronto se desactive el final de carrera.

5. Si se quedan atascados o bien el pulsador de marcha o el de paro durante más de 3 segundos, se encenderá un piloto de alarma de pulsadores. El piloto se apagará tan pronto los dos pulsadores se encuentren liberados.
6. Se habilitará un pulsador de prueba de pilotos. Al accionarlo, todos los pilotos se encenderán.

2. Asignación de entradas y salidas

Bit	Identificador	Descripción
0.0	FC	Final de carrera
0.1	PM	Pulsador marcha
0.2	PP	Pulsador paro
0.3	PPRU	Pulsador de prueba de pilotos
10.0	EVC	Electroválvula del cilindro
10.1	PA_AVA	Piloto de alarma de avance
10.2	PA_RET	Piloto de alarma de retroceso
10.3	PA_PUL	Piloto de alarma de pulsadores
TIM00	TA_AVA	Temporizador de alarma de avance
TIM01	TA_RET	Temporizador de alarma de retroceso
TIM02	TA_PUL	Temporizador de alarma de pulsadores

Maniobra del cilindro



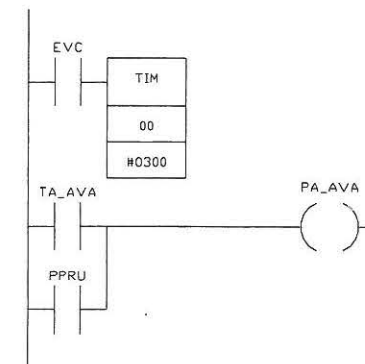
Si se acciona PM se mantiene EVC hasta que se accione PP o FC.

Si la electroválvula del cilindro EVC está accionada durante más de 30 segundos sin que se alcance el final de carrera se dispara la alarma y se enciende el piloto.

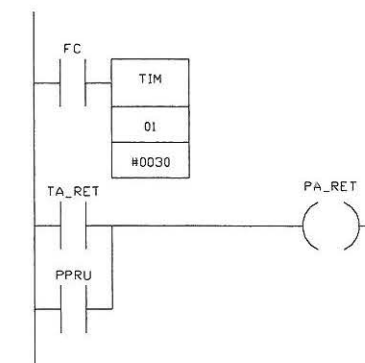
Se apaga cuando desaparece EVC, bien porque se alcanza el final de carrera o porque se anula la maniobra.

Si se acciona el pulsador de prueba se enciende el piloto de alarma.

Alarma de avance



Alarma de retroceso

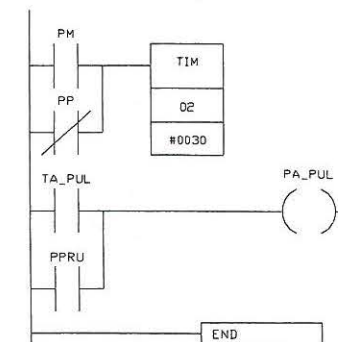


Si el final de carrera FC del cilindro está accionado durante más de 3 segundos se dispara la alarma y se enciende el piloto.

Se apaga cuando desaparece la señal del final de carrera FC.

Si se acciona el pulsador de prueba se enciende el piloto de alarma.

Alarma de pulsadores



15.30. ACCIONAMIENTO DE UN TALADRO

1. Descripción de la maniobra

Se dispone de una máquina de taladrar compuesto por los siguientes elementos:

Motor del taladro (MT).

Detector de marcha del motor del taladro. (DMT).

Variador de frecuencia con inversión de giro para el desplazamiento vertical del taladro, con dos entradas, taladro arriba (TA) y taladro abajo (TB):

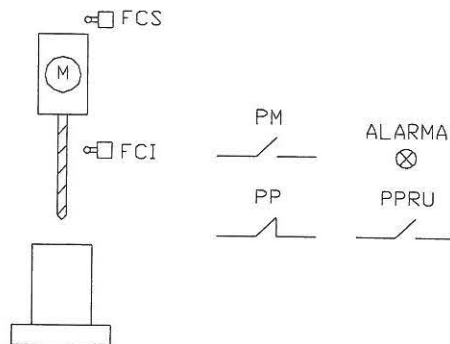
Final de carrera para detectar la posición superior del taladro (FCS).

Final de carrera para detectar la posición inferior del taladro (FCA).

Pulsador de marcha (PM).

Pulsador de paro (normalmente cerrado) (PP).

Piloto de alarma (PALA).



1. Al accionar el pulsador de marcha (PM), y estando el taladro en su posición superior (FCS), se acciona el motor del taladro (MT).
2. Una vez se tiene confirmación de que el taladro está en marcha (CMT), se da la orden de descenso (TB).
3. Cuando el taladro llega a su posición inferior (FCI), se invierte su desplazamiento y se da la orden de ascenso (TA).
4. Al llegar a la posición superior (FCS) se detienen el motor del taladro (MT) y el ascenso del cabezal (TA).

5. En cualquier momento se puede parar la maniobra mediante el pulsador de paro (PP). Su funcionamiento será el siguiente:

- Al pulsar el botón de paro (PP) se detiene el movimiento vertical del taladro (TA o TB) y el motor del taladro.
- Si estando pulsador el botón de paro se acciona el de marcha, la maniobra se considera anulada.
- Al soltar el botón de paro (PP) la máquina sigue parada.
- Al volver a pulsar el botón de marcha (PM),
 - si la maniobra ha sido anulada, el taladro vuelve arriba,
 - si la maniobra no ha sido anulada, continúa normalmente.

6. Se contemplarán las siguientes causas de posible avería:

- Que no se reciba confirmación de marcha del taladro (CMT) transcurridos 30 segundos desde que se dio la orden de marcha (MT).
- Que desaparezca la confirmación de marcha del taladro mientras tiene orden de marcha.
- Que el taladro no alcance su posición inferior (FCI) transcurridos 30 segundos desde que se le dio orden de descenso (TB).
- Que el taladro no alcance su posición superior (FCS) transcurridos 30 segundos desde que se le dio orden de ascenso (TA).

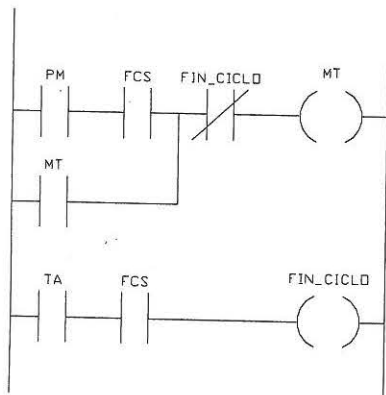
Cuando se presente cualquiera de estas averías, se encenderá un piloto de alarma (PALA) con intermitencia de 1 segundo (0.5 segundos on, 0.5 segundos off), y se anulará la maniobra. Al pulsar el botón de paro se anulará la señal de alarma y se volverá el taladro a su posición de reposo.

Se dispondrá también de un pulsador de prueba de pilotos. Al activarlo, se encienden todos los pilotos.

2. Asignación de entradas y salidas

Bit	Identificador	Descripción
0.0	FCS	Final de carrera superior
0.1	FCI	Final de carrera inferior
0.2	CMT	Confirmación de marcha del taladro
0.3	PM	Pulsador marcha
0.4	PP	Pulsador paro
0.5	PPRU	Pulsador de prueba de pilotos
10.0	MT	Motor del taladro
10.1	TB	Taladro abajo
10.2	TA	Taladro arriba
10.3	ALARMA	Piloto de alarma
INTER	255.2	Intermitente 1 segundo
TIM00	TALA	Temporizador de alarmas
5.0	FIN_CICLO	Relé auxiliar que indica final del ciclo
5.1	AUX_ALARMA	Relé auxiliar de alarma
5.2	AUX_TB	Relé auxiliar de orden de bajada
5.3	AUX_TA	Relé auxiliar de orden de subida

Maniobra del taladro



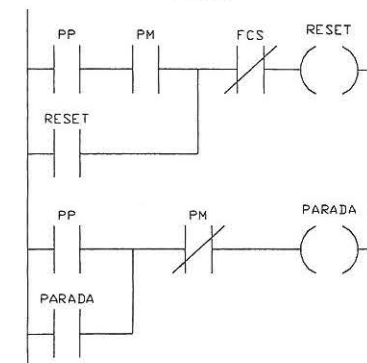
El ciclo se termina cuando el taladro llega al final de carrera superior cuando está subiendo.

Si se acciona pulsador de marcha se acciona el motor del taladro hasta que termina el ciclo.

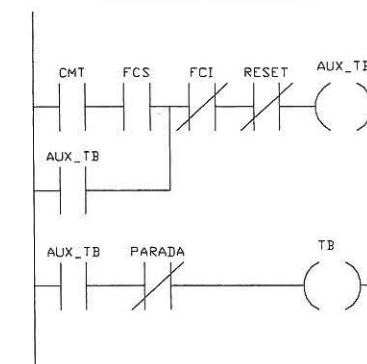
Si estando el pulsador de paro accionado se acciona el de marcha, se resetea la maniobra.

Si se acciona pulsador de paro se interrumpen todos los movimientos. Se reinician al accionar de nuevo el pulsador de marcha.

Reset



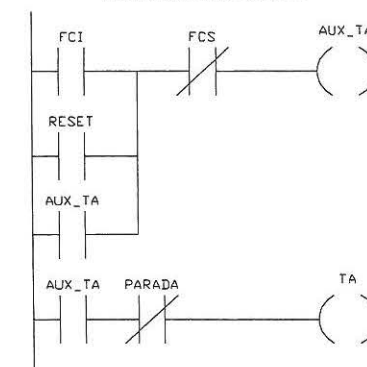
Descenso del taladro



Al iniciar el giro del taladro, estando el taladro arriba (FCS), se da la orden al taladro de bajar. Cuando llega al final de carrera inferior, o se anula la maniobra, se desconecta la orden.

Para poder parar el movimiento, manteniendo al mismo tiempo el enclavamiento de la señal, se utiliza un relé auxiliar AUX_TB

Ascenso del taladro

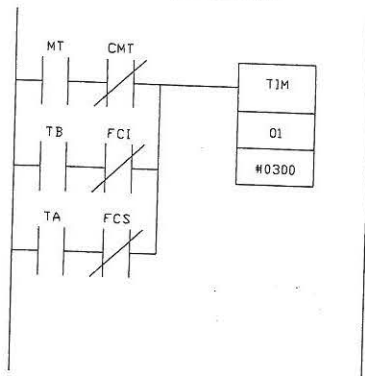


Cuando el taladro llega a la posición inferior (FCI), o bien se anula la maniobra, se da la orden de subir al taladro.

El taladro detiene el ascenso al llegar al final de carrera superior.

Para poder parar el movimiento, manteniendo al mismo tiempo el enclavamiento de la señal, se utiliza un relé auxiliar AUX_TA

Alarmas temporizadas



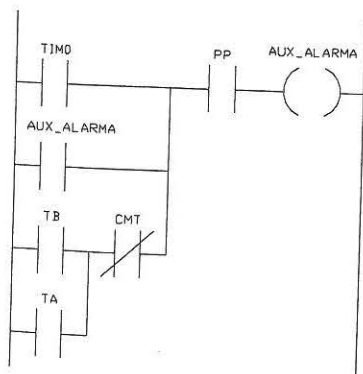
Si estando el motor del taladro en marcha no tengo confirmación de que gira:

O bajando el taladro no toca el FCI.

O subiendo el taladro no toca el FCS.

Antes de 30 segundos, hay alarma.

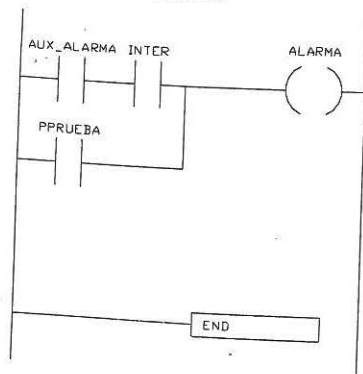
Alarma



Si hay alguna alarma anterior, o bien desaparece la señal de confirmación del motor del taladro mientras asciende o descende:

Se activa el relé auxiliar de alarma, con autorretención y se libera cuando se pulsa el botón de paro.

Piloto de alarma



Si hay alguna alarma se enciende el piloto con intermitencia.

Si se activa el pulsador de prueba de pilotos, se enciende el piloto sin intermitencia.

15.31. ARRANCADOR ESTRELLA TRIÁNGULO

1. Objetivo:

1. Programación de una secuencia de mando con entradas y salidas digitales.
2. Empleo de temporizadores.
3. Detección de errores.

2. Descripción del proceso:

1. Al activar el pulsador de marcha entran los contactores de línea y de estrella.

Al detectar que entra el contactor de línea se arranca el tiempo de espera como retardo a la conexión.

Al finalizar el tiempo de espera sale el contactor de estrella.

Cuando ha salido el contactor de estrella entra el de triángulo.

2. Se comprobarán adicionalmente las siguientes condiciones de trabajo:

Que un contactor sin señal de mando no entre.

Que un contactor con señal de mando haya entrado.

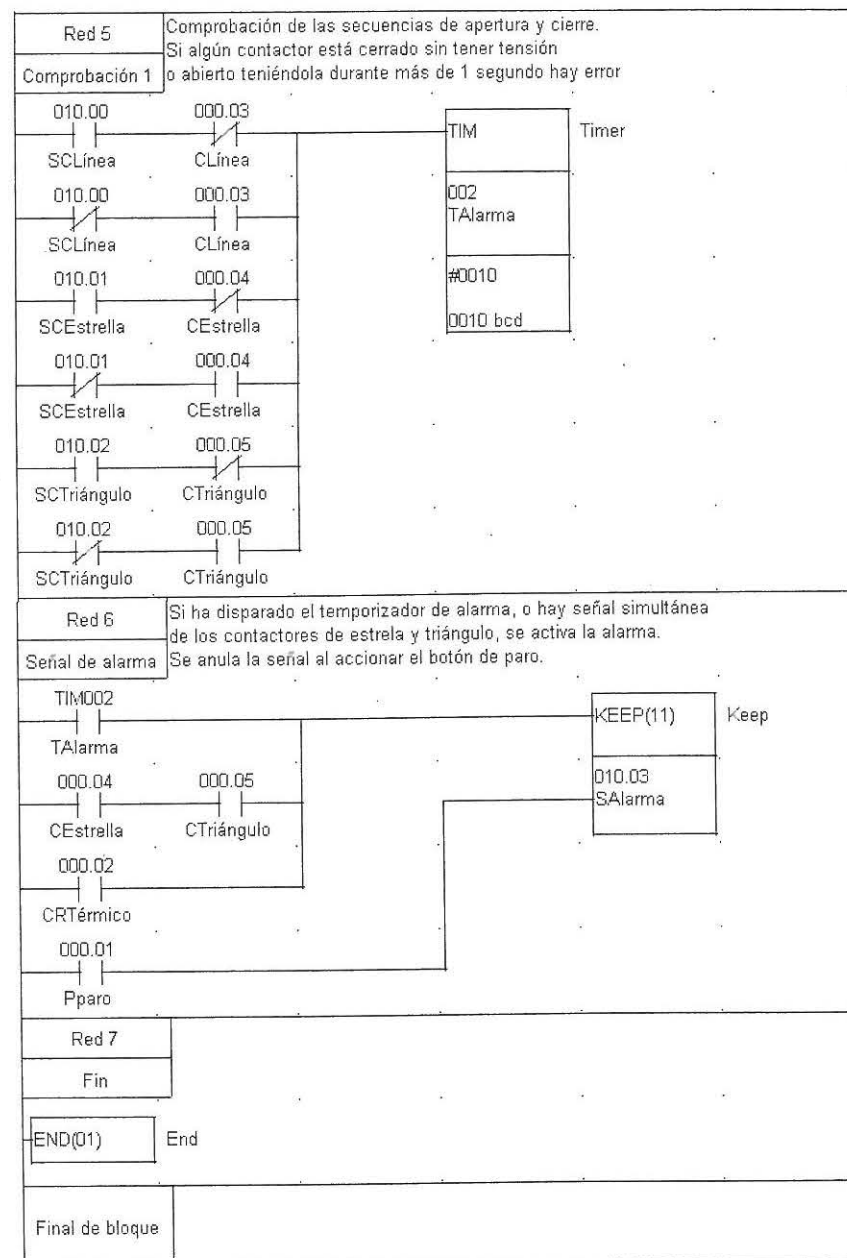
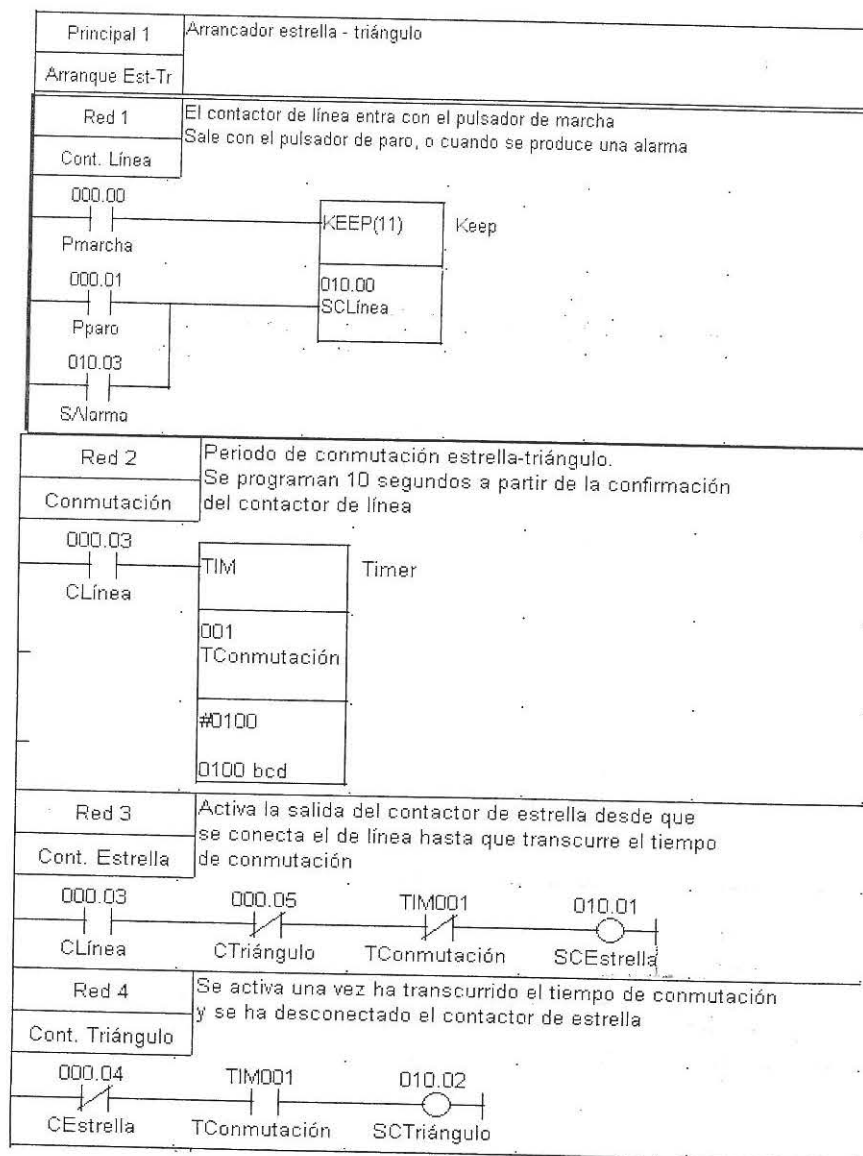
No simultaneidad de los contactores de estrella y triángulo.

Que no haya disparado el relé térmico.

Nº de entrada	Descripción	Comentario
0.00	Pmarcha	Pulsador de puesta en marcha
0.01	Pparo	Pulsador de paro
0.02	CRtérmico	Contacto del relé térmico
0.03	CLínea	Contacto auxiliar contactor de línea
0.04	CEstrella	" " " estrella
0.05	CTriángulo	" " " triángulo

Nº de salida	Descripción	Comentario
10.00	SCLínea	Contactador de línea
10.01	SCEstrella	Contactador de estrella
10.02	SCTriángulo	Contactador de triángulo
10.03	SAlarma	Salida de alarma

Nº de Temporizador	Descripción	Comentario
T1	TConmutación	Tiempo de conmutación de estrella a triángulo
T2	TAlarma	Tiempo de alarma



15.32. CONTROL DE UN PARKING

1. Descripción de la maniobra

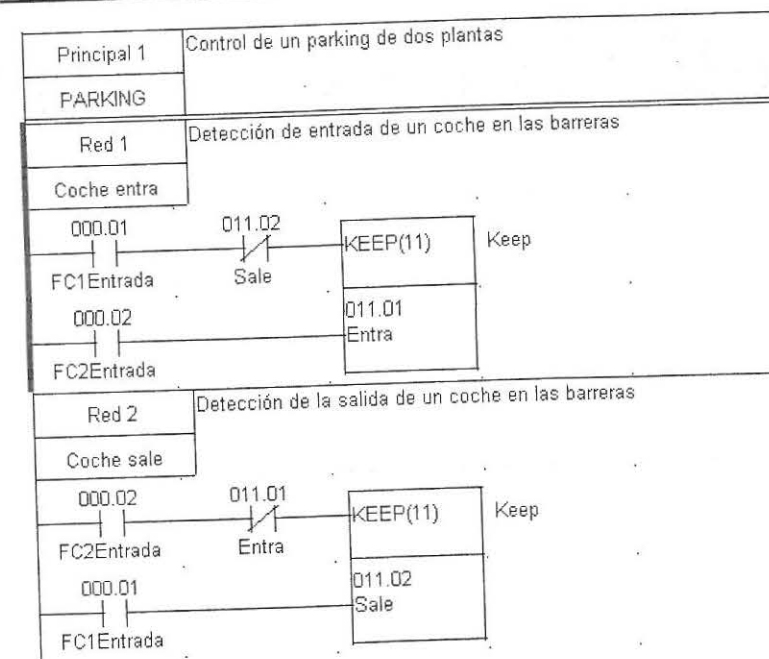
Se trata de un parking de 2 plantas, en cada planta caben 15 coches. El sistema consta de los siguientes elementos:

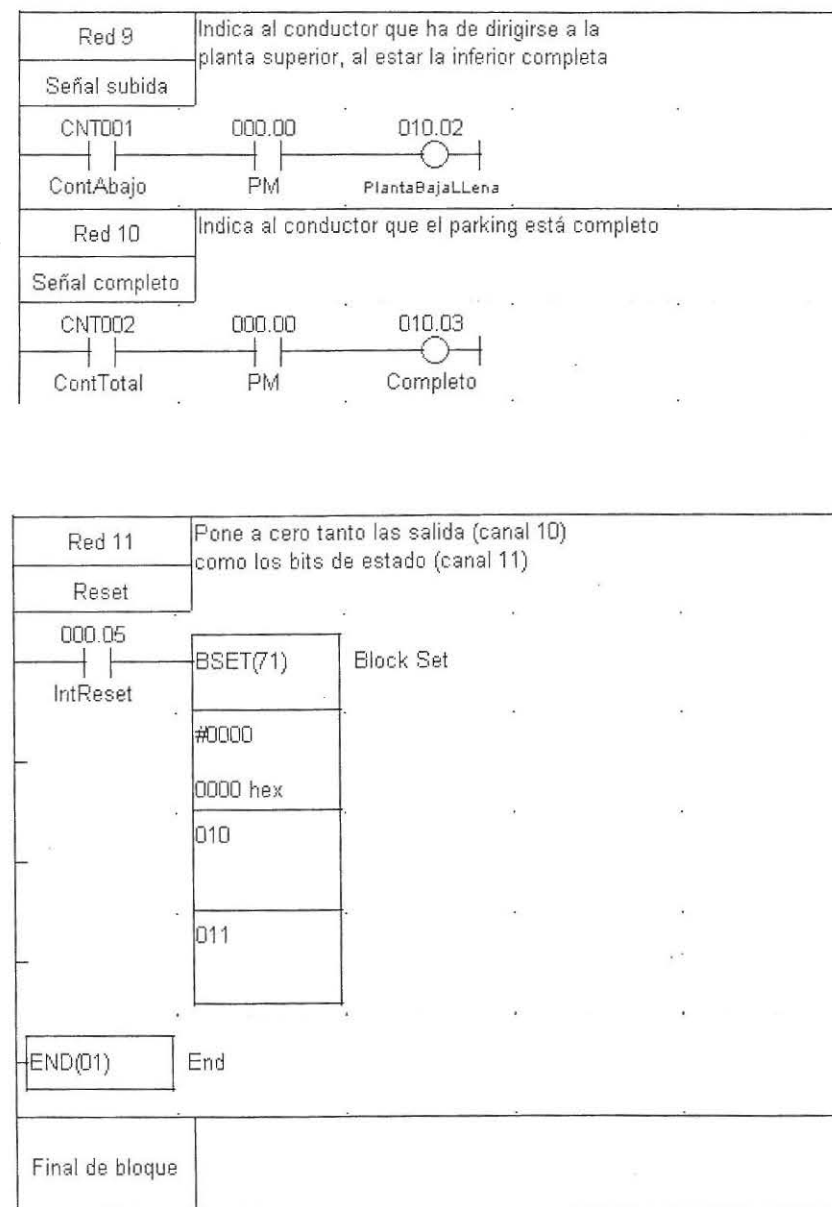
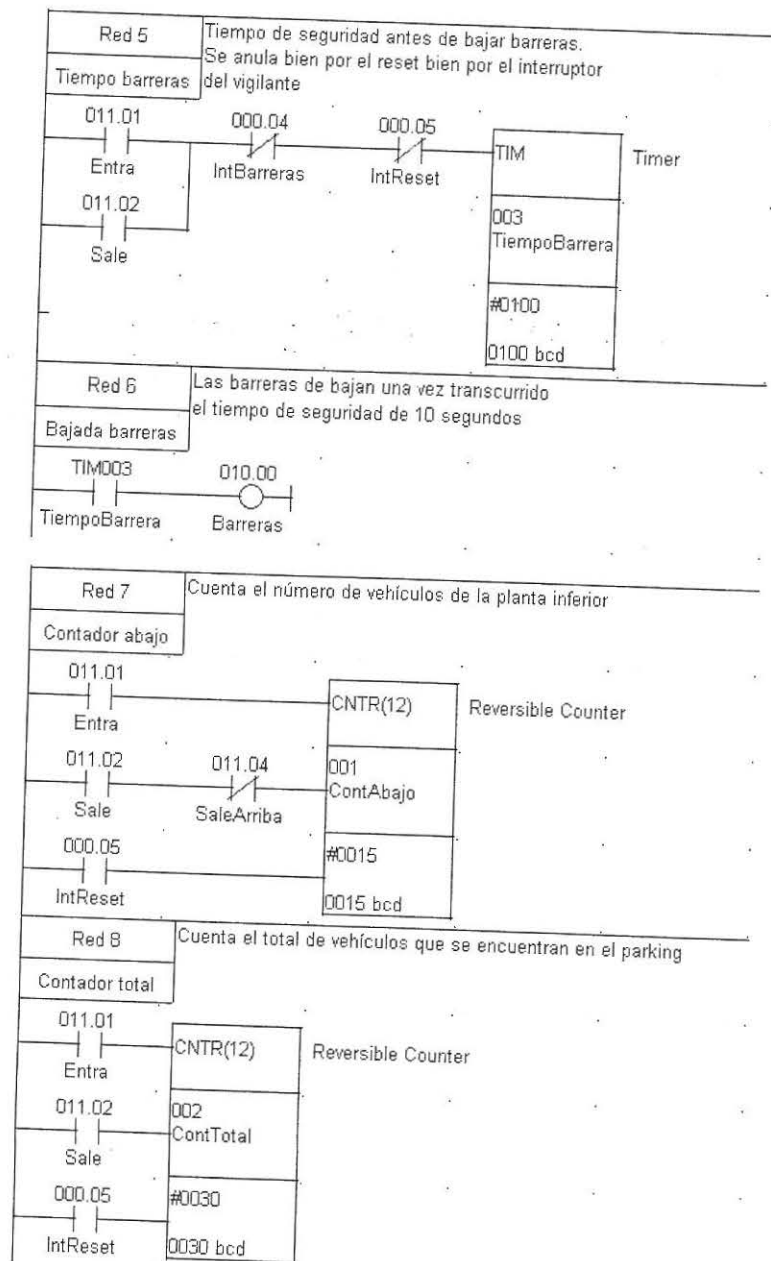
- 1 Interruptor de puesta en marcha.
- 1 Interruptor de reset de todo el sistema.
- 1 Interruptor para bajar las barreras por parte del vigilante.
- 2 Sensores ubicados en la entrada del parking.
- 1 Sensor ubicado en la zona de acceso de los vehículos a la planta 1ª.
- 2 Carteles luminosos que indique la planta o plantas llenas de vehículos.
- 1 Cartel de que indique completo, este se puede combinar con otro que indique plazas libres, ambos controlados con la misma salida.
- 1 Salida que controle la bajada de las barreras

1. Al accionar el interruptor de puesta en marcha (PM), se activan todas las salidas de los carteles correspondientes al estado del parking.
2. Al llegar un coche y ser este detectado por el sensor correspondiente, de entrada o de salida se activan las barreras cerrándose estas dejando el coche entre ellas.
3. Al llegar el coche al segundo sensor, que según el sentido de marcha será el más cercano a la calle si está saliendo, o el más interno al parking si está entrando, las barreras se abren dejando pasar al vehículo. El vigilante tiene potestad a no dejar subir las barreras con la ayuda de un interruptor, por si el pasajero no paga o algo similar.
4. Cuando, al ir entrando vehículos se llene una de las plantas se encenderá el luminoso de planta llena. Si una vez llena la planta sale de ella algún coche se apagará de nuevo el letrero.
5. Al llenarse la otra planta se encenderán el letrero correspondiente a dicha planta y si además está el parking completo se indicará en el cartel indicador de completo. Este se apagará al salir un vehículo de cualquier planta.
6. Por si existiese alguna anomalía, el vigilante dispondrá de un interruptor que resetea y pone a cero todo el sistema.

2. Asignación de entradas y salidas

Bit	Identificador	Descripción
0.0	PM	Puesta en marcha del sistema por el vigilante
0.1	FC1Entrada	Final de carrera de entrada más cercano a la calle
0.2	FC2Entrada	Final de carrera de entrada más lejano a la calle
0.3	FC3Subida	Final de carrera acceso a planta alta
0.4	IntBarreras	Interruptor de bajada de barreras del vigilante
0.5	IntReset	Interruptor de reinicialización
10.0	Barreras	Bajada de barreras
10.1	AbajoLleno	Planta baja completa, sin plazas libres
10.2	Completo	Parking completo, sin plazas libres
11.1	Entra	Coche entrando en zona de barreras
11.2	Sale	Coche saliendo en zona de barreras
11.3	HaEntrado	Coche ha entrado en el parking
11.4	SaleArriba	Coche sale de la planta de arriba
Cnt1	ContAbajo	Contador de plazas ocupadas en planta baja
Cnt2	ContTotal	Contador de plazas totales ocupadas
Tim3	TieBarreras	Tiempo de seguridad de las barreras





15.33. DOSIFICACIÓN DE UN PRODUCTO

En una instalación de llenado de envases se desea incorporar un autómatas que gestione el proceso de llenado del envase.

La instalación consta de un silo de material, con dos válvulas de salida, una de dosificación gruesa y otra de dosificación fina. En el conducto de salida se ha colocado un contador de palas que proporciona un pulso por cada gr. de material dosificado.

El proceso se iniciará cuando se reciba una señal de marcha, y al finalizar lo indicará con una salida de estado, que mantendrá fija si el proceso ha transcurrido normalmente, o con intermitencia de 1 segundo si por cualquier causa no ha podido completarse la operación de llenado en menos de 1 minuto.

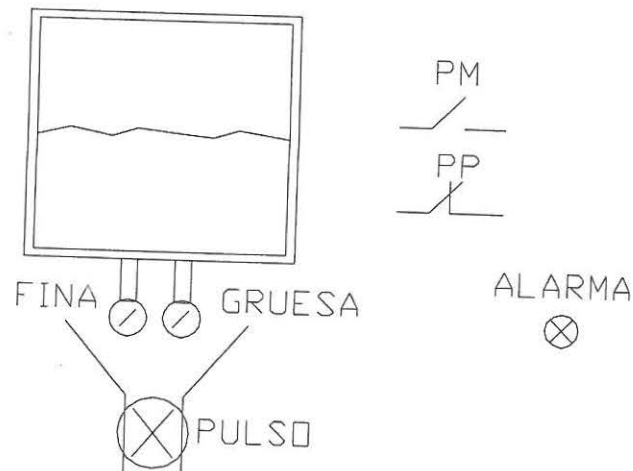
Cuando se inicia el proceso de pesada se abren las dos válvulas, gruesa y fina.

Cuando falten menos de 100 grs para completar la pesada, se cierra la válvula gruesa.

Cuando falten menos de 10 grs para el final de la pesada, se cierra también la válvula fina, y se temporizan 3 segundos para que descienda la cola de salida. Al cabo de dicho tiempo, se activa la señal de proceso terminado.

La orden de activación del proceso se dará mediante un contacto normalmente abierto. Un contacto normalmente cerrado anulará la maniobra en caso de desaparecer su señal (seta de emergencia).

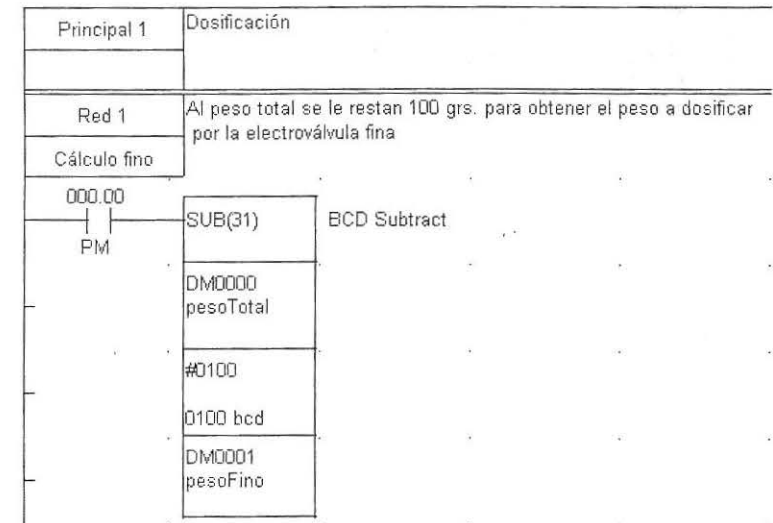
Se dispara una alarma si no se alcanza la dosificación antes de 1 minuto.

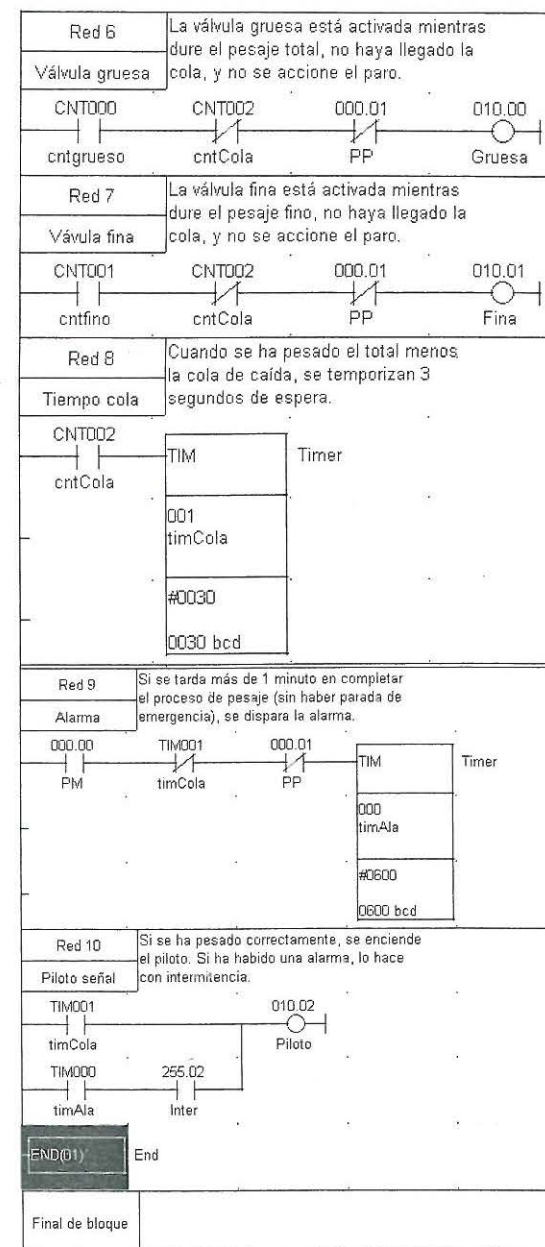
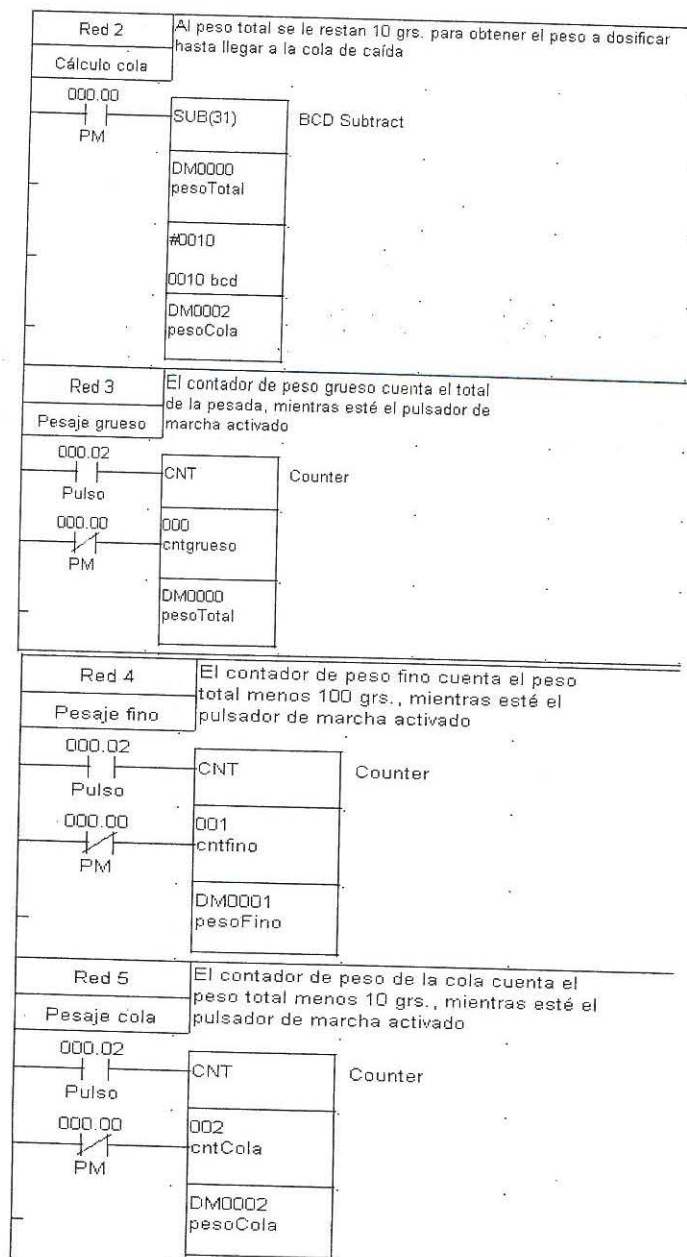


La automatización del proceso se realizará de dos modos, que clarifican conceptos desarrollados anteriormente: empleando contadores y empleando comparadores.

1. Resolución con empleo de contadores

Bit	Identificador	Descripción
DM0	Peso	Valor a dosificar
DM1	PesoFino	Valor a dosificar en ajuste fino
DM2	PesoCola	Valor a dosificar sin cola de caída
10.0	Gruesa	Válvula de dosificación gruesa
10.1	Fina	Válvula de dosificación fina
10.2	Piloto	Piloto de estado
0.0	PM	Pulsador marcha
0.1	PP	Pulsador paro
0.2	Pulso	Pulso de contaje
INTER	255.2	Intermitente 1 segundo
TIM00	timCola	Temporizador de cola de caída
TIM01	timAla	Temporizador de alarma
CNT0	CntGrueso	Contador de pesaje grueso
CNT1	CntFino	Contador de pesaje fino
CNT2	CntCola	Contador de cola de caída





2. Resolución mediante comparadores

Bit	Identificador	Descripción
DM0	Peso	Valor a dosificar
DM1	faltaPesar	Valor a pesar
10.0	Gruesa	Válvula de dosificación gruesa
10.1	Fina	Válvula de dosificación fina
10.2	Piloto	Piloto de estado
0.0	PM	Pulsador marcha
0.1	PP	Pulsador paro
0.2	Pulso	Pulso de conteo
INTER	255.2	Intermitente 1 segundo
TIM00	timCola	Temporizador de cola de caída
TIM01	timAla	Temporizador de alarma

