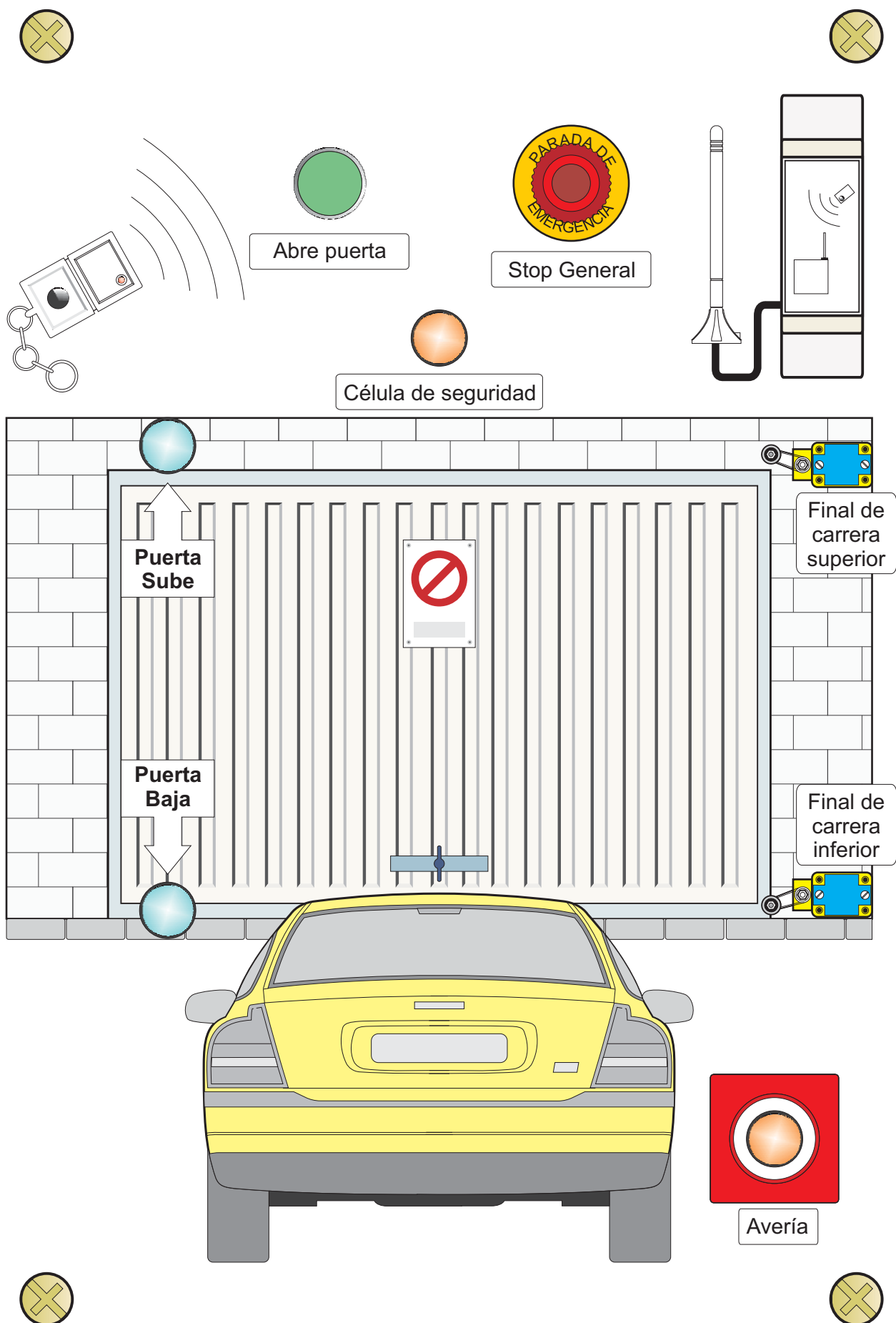
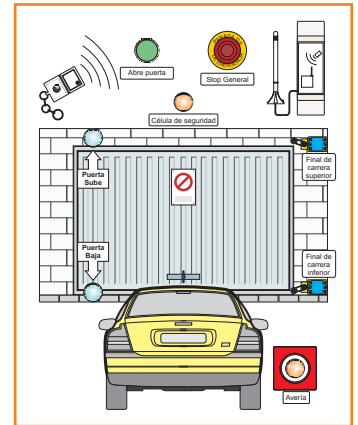


10 Autómatas programables



1 Lógica cableada

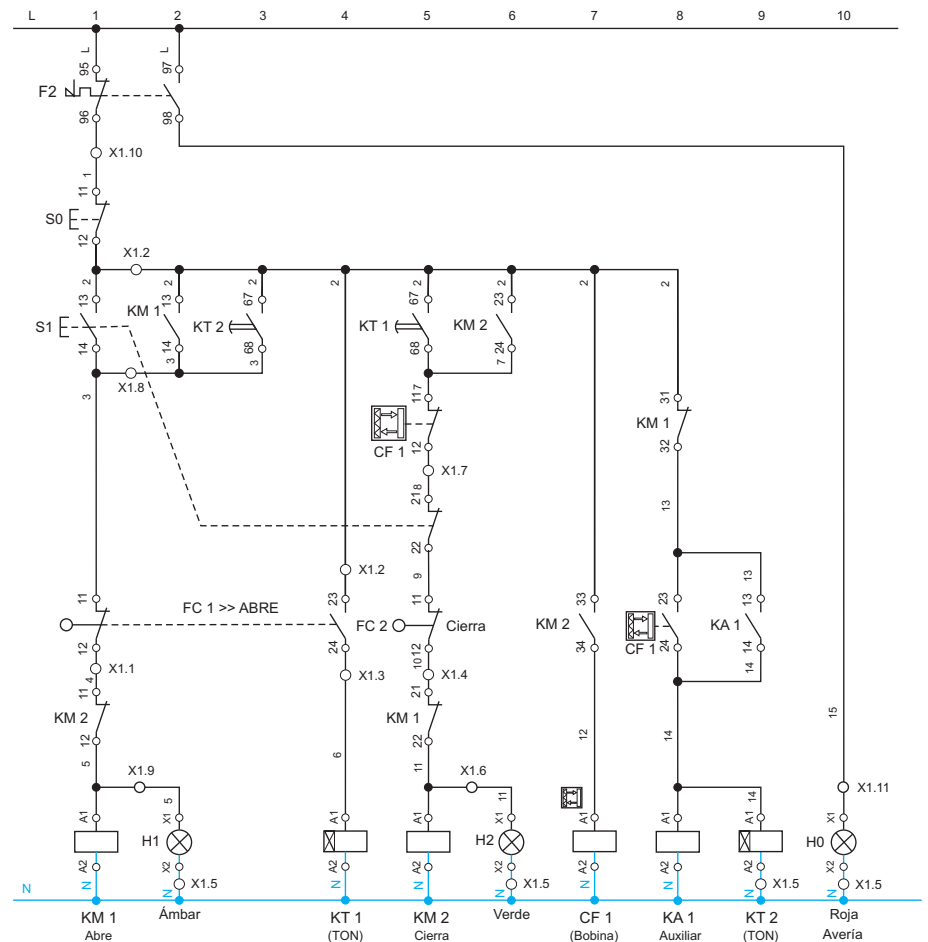
Cuando el funcionamiento de una operación automática se realiza con la alimentación principalmente de relés y contactores, producidas o provocadas por los accionamientos de elementos mecánicos tipo interruptor, pulsador, final de carrera, entre otros, se dice que la lógica del circuito es cableada; de hecho, una modificación en el funcionamiento supone la reestructuración de parte del cableado existente. Si la instalación es permanente con un proceso de funcionamiento sin proyectos de modificación, no es necesario que la gestione un PLC. Note el esquema de mando de la figura 10.1. referida al control motorizado de una puerta de garaje.



El pulsador S1, pone en marcha el sistema, que consiste en la apertura de la puerta (KM1), tiempo de reposo para entrada o salida de vehículos (KT1), y cierre de puerta (KM2). Como medida de seguridad, una célula fotoeléctrica que controla el ancho de la puerta mientras ésta se está cerrando, y en caso de actuación, provoca la parada de cierre de puerta, una pausa de dos segundos (KT2), comenzando el proceso de apertura.

Ahora proponemos algunas modificaciones:

- Si es de noche, además de abrir la puerta, se activarán las luminarias del garaje.
- Desde el interior del garaje existe un interruptor que anula el mando a distancia.
- Si se interrumpe tres veces seguidas el proceso de cierre de la puerta, el sistema se bloquea.



¿Qué problema plantean las modificaciones propuestas?

ii El completo recableado de la instalación !!

La conclusión es, que si la instalación a gobernar cuenta con varios sensores de entrada, y la misma es vulnerable de modificar por razones funcionales o de producción, sale más rentable utilizar un controlador programable, que realizar la instalación con lógica cableada.

El controlador programable modificará la instalación en el programa interno de la máquina, evitando en su mayoría, el recableado general de la instalación.

Figura 10.1. Cuadro sinóptico y esquema de control de una puerta de garaje.

2 Nacimiento del autómata programable

Desde el comienzo de la industrialización, el hombre ha buscado y mejorado constantemente los procedimientos y medios para que las labores de producción se hicieran cada vez más rápidas, menos repetitivas para el operario, mejorando el puesto de trabajo y consiguiendo un rendimiento cada vez más eficaz, ayudado por la tecnología eléctrica basada en control y lógica cableada.

En 1968, una división de una fábrica de automóviles, propone a través de un concurso, la creación de un instrumento tipo controlador electrónico, que sustituyera a los sistemas de control cableados (interruptores, relés, contactores...), por un control programado, con el objetivo de ahorrar costes en los procesos de fabricación. Conocido como el padre del PLC, el desarrollador del proyecto fue Dick Morley.

El MODICON 084 (MODular Digital CONTroller), fue el primer autómata programable comercializado. En 1972, los PLCs, ya comenzaron a programarse con un lenguaje parecido a lo que hoy conocemos como Lenguaje de Contactos Ladder Diagram, con símbolos que provenían del diseño de antiguos cuadros eléctricos.

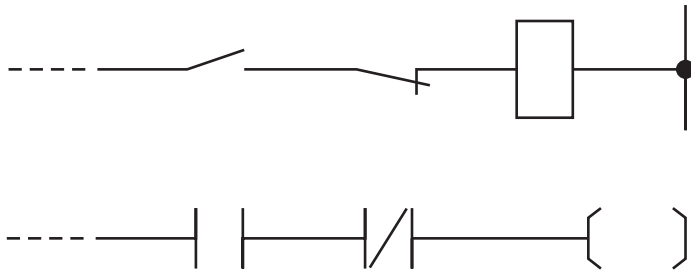


Figura 10.2. El lenguaje de programación “de contactos” deriva de los contactos y bobinas eléctricas habituales en esquemas.

El crecimiento de las máquinas fue rápido, y a partir de 1974, con la incorporación de los microprocesadores, la capacidad de operación fue creciendo exponencialmente -las computadoras iban un paso por delante-, lo que condicionaba de manera directa el desarrollo de los controladores programables.

Los primeros autómatas programables no realizaban complejas operaciones de cálculo, pero comenzaban a manejar operaciones tediosas para el hombre, ya fuera por el ambiente de trabajo, o por la cadencia de las operaciones, por ejemplo, en procesos de soldadura o pintado.

En la siguiente imagen se muestran autómatas programables de carácter industrial, algunos de ellos, ya descatalogados.



Figura 10.3. Autómatas programables industriales.

PLC

El término PLC proviene de “Programmable Logic Controller”, que traducido viene a ser Controlador Lógico Programable, y se usará indistintamente con la designación de Autómata Programable.

El término lo acuñó el austriaco Odo Struger que trabajaba en la empresa Allen-Bradley, hoy Rockwell Automation.

Busca en Internet la vida de los padres del autómata programable:

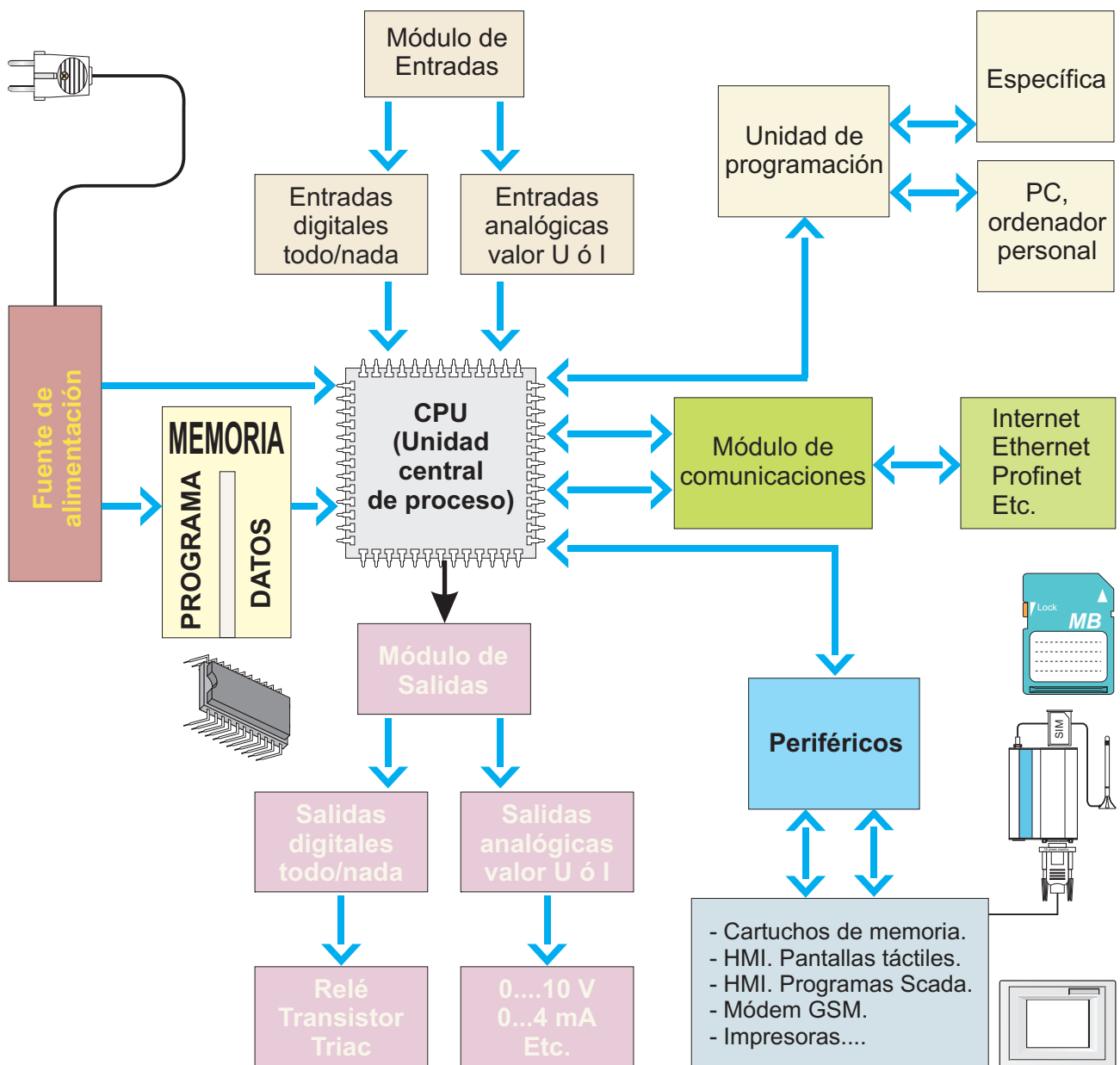
- Dick Morley
- Odo Josef Struger

3 Lógica programada

Si el número de captadores a utilizar en un proyecto es amplio, aunque se estime que la instalación no va a sufrir cambios, el coste del microcontrolador será insignificante si sólo una vez decidiéramos realizar una modificación de control (recableado, pruebas, puesta en marcha, verificación, tiempo perdido, parada de producción, etc.).

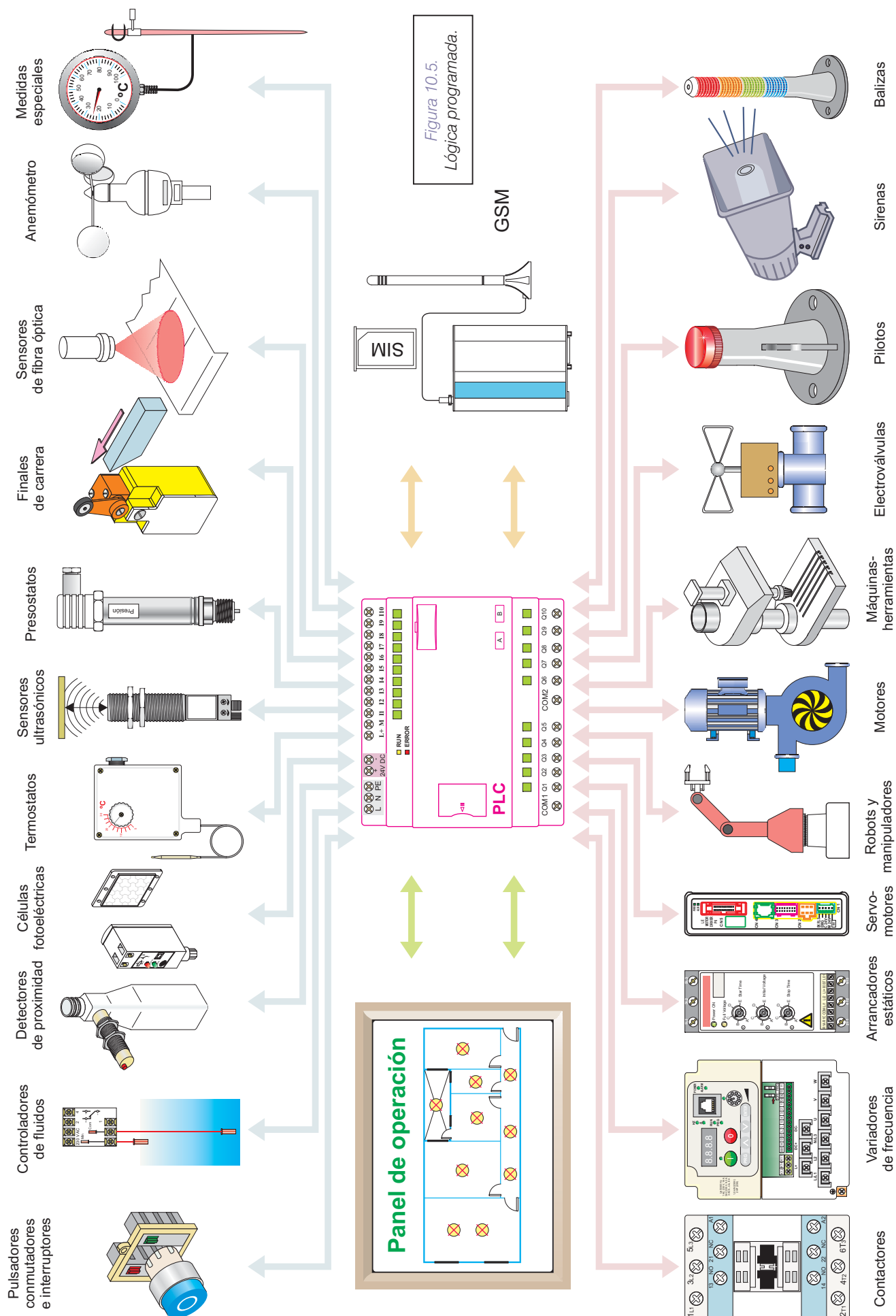
Sirva el ejemplo; si proponemos gestionar las lámparas de los semáforos de un cruce de dos calles, la instalación la pueden realizar automatismos convencionales (relés y temporizadores), en una cantidad determinada; pero si el número de calles se amplía, ya no tiene sentido utilizar automatismos cableados, cuyo volumen sería exagerado; se haría con control programable. La siguiente figura, muestra el esquema de bloques de un autómata programable. Note, la cantidad de interacciones posibles.

Figura 10.4.
Esquema de bloques
de un autómata programable.



En la página siguiente, se muestra un esquema de captadores, actuadores y relaciones que se pueden dar en un PLC. Es por tanto, lógica programada.

236



4 Características de un autómatas programable

Con la propuesta de creación del primer autómatas programable, se indicó que debía cumplir las siguientes características:

- Tenía que usar componentes de estado sólido (transistores, triac, diodos, etc.), para que no sufriera el desgaste que ocasionaban relés y contactores.
- Su configuración tenía que ser flexible, para que se adaptara a cualquier cambio o reutilización, principalmente por programación.
- Debía soportar ambientes de trabajo industriales (horas indefinidas, ambientes tóxicos, temperaturas extremas, etc.).
- La instalación, manejo, reutilización, y mantenimiento en general, debía ser interno, es decir, por los propios operarios de la industria.
- Las funciones a gobernar eran binarias, basadas en 0 / 1.

Ciertamente, la máquina superó con creces los primeros requerimientos, y los resultados quedan anecdóticos. A fecha de hoy, existen transistores capaces de conmutar miles de veces por segundo; el ciclo de programa de los PLCs, se acerca a los nanosegundos; manejan datos de 64 bits, y un largo etcétera.

Operaciones que debe realizar un autómatas programable

La llegada de los microprocesadores a la automatización, hace que las operaciones aumenten y mejoren tan rápidamente, que es difícil enumerar con certeza todas. Destacamos:

- Operaciones lógicas (AND, OR, NOT, NOR, NAND, XOR).
- Operaciones con entradas digitales.
- Operaciones con entradas analógicas.
- Operaciones con salidas digitales.
- Operaciones con salidas analógicas.
- Operaciones con marcas internas.
- Funciones aritméticas (sumas, restas, multiplicaciones, divisiones, raíces cuadradas, trigonometría, logaritmos...).
- Transferencia de datos de diferente tipología.
- Conversión de datos de unos sistemas de numeración a otros (entero, binario, BCD, hexadecimal, entre otros).
- Funciones de tiempo. Temporizadores y relojes.
- Funciones de cómputo. Contadores y contadores rápidos.
- Comparación de datos (mayor que, menor que e igual a.....).
- Creación de subrutinas, que es una forma de dividir el programa en fragmentos más pequeños, y todos juntos forman la programación final.
- Funciones de interrupción del programa.
- Funciones de autochequeo.
- Funciones de red, para la interconexión con entes de origen superior e inferior, pudiendo establecerse comunicaciones tipo maestro-esclavo, esclavo-maestro, o simplemente compartir una red al mismo nivel de control con otros dispositivos.
- Control de visualización y control a través de pantallas y sistemas SCADA.
- Control remoto de los sensores y captadores.
- Control remoto de los sistemas de visualización y control incluso para diagnóstico y reparación.
- Permite la grabación de los programas en varios formatos.
- Permite la redundancia con otros autómatas; esto supone que en caso de avería de uno de ellos, el otro -de idénticas características y misma programación- se hace cargo de la automatización.
- Se pueden programar en diferentes lenguajes.

Domótica

La automatización de viviendas y edificios, se denomina "Domótica".

Algunos de los sistemas domóticos están basados en autómatas programables.

Esto supone la adaptación de PLCs industriales a entornos domésticos, donde cambia la tipología de sensores, actuadores y sobre todo, sistemas de comunicación.

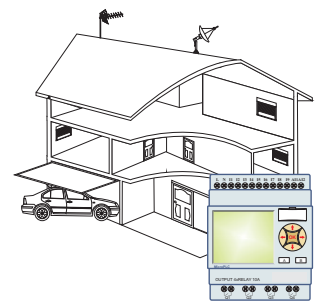


Figura 10.6. Automatización de viviendas con PLC.

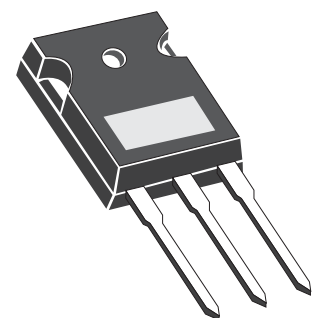
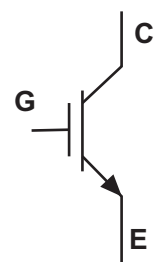


Figura 10.7. Transistor IGBT.

5 MicroPLCs

La referencia de microPLC no está claramente determinada. Algunas empresas afirman que un PLC será micro, si el número de entradas y salidas que gobierna no es superior a 32.

Observe la composición de un microPLC estándar, que podemos buscar en el mercado. Incorpora en un sólo módulo, la mayoría de los componentes básicos para su funcionamiento, como fuente de alimentación, entradas digitales, una o varias entradas analógicas, cartucho de memoria donde guardar los programas de usuario, salidas digitales, y lo más práctico, una pantalla y teclado programador que evita el uso de una unidad de programación. Adicionalmente, el modelo permitirá la ampliación de módulos para entradas/salidas, módulo GSM, módulo de comunicaciones, principalmente.

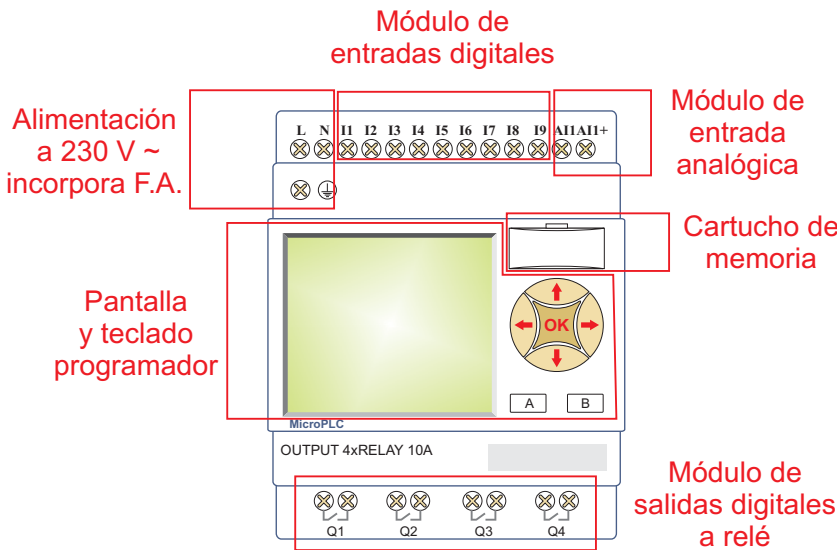


Figura 10.8. MicroPLC, micro autómatas programables.

6 Módulos de entrada

Los módulos de entrada podrán recibir principalmente señales digitales todo/nada, o analógicas en formato tensión (ejemplo 0...10 V DC) o intensidad (ejemplo 0...20 mA).

6.1. Señales digitales (todo-nada)

Los terminales de los módulos de entrada o simplemente los terminales de entrada digitales todo/nada, recibirán un valor de tensión de captadores tales como:

- Pulsadores.
- Interruptores.
- Finales de carrera.
- Termostatos.
- Presostatos y vacuostatos.
- Detectores capacitivos, inductivos o fotoeléctricos.

Ejemplo

Suponemos que el módulo de entradas digitales de un microPLC admite una tensión de 24 V DC. Cada vez que el captador -final de carrera- permite el paso de dicha tensión al micro-autómatas, estará enviando una señal que el programa de usuario tendrá que interpretar y actuar en consecuencia.

Micro autómatas programables

Prácticamente, todas las marcas de autómatas programables, disponen de un modelo de bajo nivel, limitado en entradas, salidas, y bloques operacionales, pero muy útiles en cuadros eléctricos para control discreto de operaciones.

Los nombres para su designación son diversos:

- Relé lógico.
- Caja de relés.
- NanoPLC.
- Mini autómatas.
- Relé programable.
- Relé industrial.
- Etc.

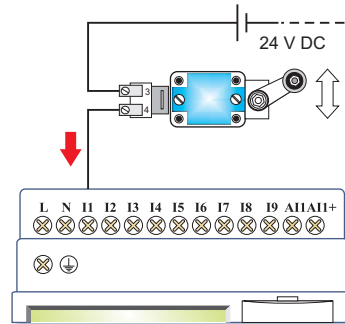


Figura 10.9. El final de carrera implementa un valor de tensión en la entrada, cuando es activado.

Los sensores que aportan señales digitales todo/nada, pueden a su vez ser pasivos y activos.

● Captadores pasivos. Funcionan preferentemente con un movimiento muscular o mecánico y no necesitan de una fuente de energía adicional para estar operativos. Entre ellos, interruptores, pulsadores y finales de carrera. En esencia, el movimiento ejercido sobre el dispositivo conmutará uno o varios contactos, que son los que permiten el flujo de corriente eléctrica.

Note en el gráfico, que cuando es presionado el pulsador, se cierra un contacto del mismo, y a su vez da paso de corriente al PLC, y por tanto, el envío de una señal que será analizada por el programa.

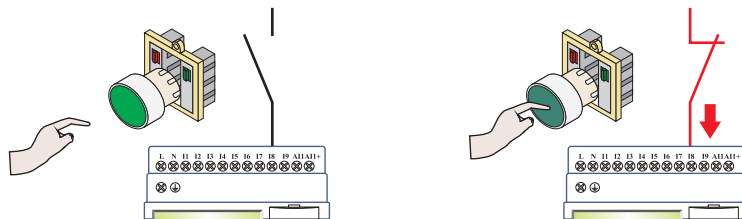


Figura 10.10. El pulsador es un captador pasivo.

● Captadores activos. Requieren de una fuente adicional de energía para operar. Algunos son: detectores capacitivos, inductivos, células fotoeléctricas...

Ejemplo

El siguiente detector capacitivo implementará 24 V (+), sólo cuando se acerque un objeto. Para poder funcionar, el detector tiene que estar alimentado por 24 V DC.

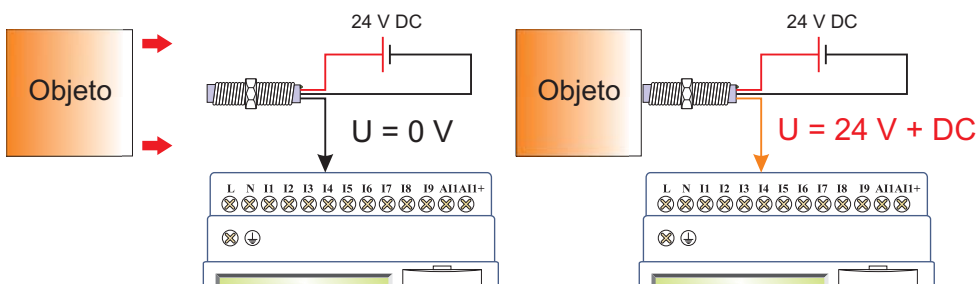


Figura 10.11. El detector capacitivo es activo. Necesita alimentación.

Captadores redundados

Si una aplicación industrial es susceptible de provocar un riesgo, (por ejemplo una prensa) y los sensores que la manejan a nivel de seguridad son activos, se pueden redundar para que en caso de avería de alguno de ellos, el sistema de protección siga funcionando.

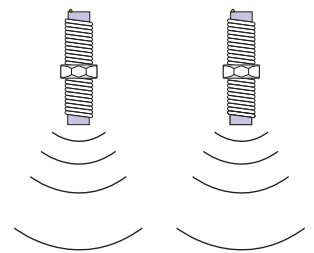


Figura 10.12.

Detectores redundados para aumentar la seguridad de la aplicación.

10 Autómatas programables

6.2. Señales analógicas

Los terminales de los módulos de entrada de señales analógicas, recibirán un valor de tensión o intensidad equivalente a la magnitud real medida. El técnico debe calibrar la señal procedente del sensor de forma correcta para evitar que la lectura sea errónea.

Valores estándar de tensión	Valores estándar de intensidad
-10 V a +10 V. 0 a +10 V c.c. +2 a +10 V c.c.	0 a 20 mA. 4 a 20 mA. +1 a -5 mA 0 a +5 mA.

Tabla 10.1. Valores estándar de tensión e intensidad para sensores analógicos.

Ejemplo

Un anemómetro mide la velocidad del viento, y en su composición, se encuentra una pequeña dinamo solidaria al eje principal del mismo. Según la velocidad de giro, la dinamo generará una determinada tensión; sirvan los valores:

- Anemómetro parado, genera 0 V DC.
- Anemómetro girando a 50 km/h, genera 4 voltios DC.
- Anemómetro girando a 100 km/h, genera 8 voltios DC.
- Etc.

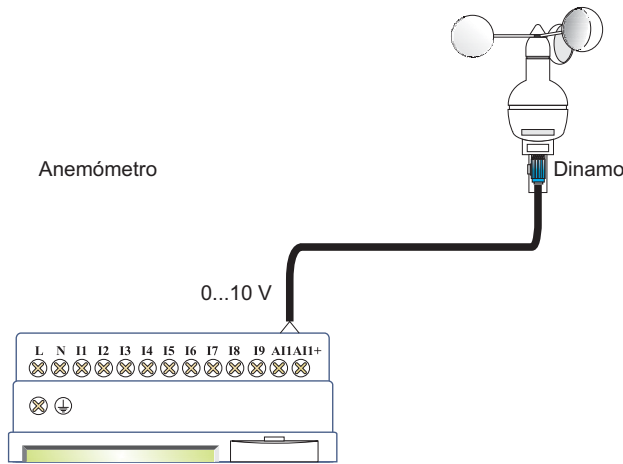


Figura 10.13. El anemómetro se conecta a la entrada analógica del autómata.

7 Módulos de salida

Los módulos de salida permiten alimentar los dispositivos que hacen el trabajo de las instalaciones. Como sucede con los módulos de entradas, las salidas aportarán señales todo/nada, o señales analógicas, como valores de tensión o intensidad variables. Las salidas digitales todo/nada alimentarán principalmente:

- Sistemas de alumbrado.
- Timbres o avisadores acústicos.
- Electroválvulas.
- Contactores.
- Relés.
- Aparatos de caldeo.
- Arrancadores.
- Variadores de frecuencia.

Transductor

Es un dispositivo, que transforma una magnitud física (presión, temperatura, velocidad, longitud, humedad...) en una señal -generalmente de carácter eléctrica-, que posteriormente será tratada.

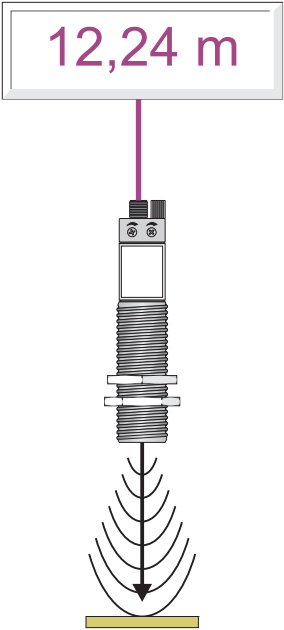


Figura 10.14. El sónar mide la distancia. El transductor convierte el valor en señal eléctrica, que luego podrá ser visualizada.

Las salidas analógicas aportarán señal de control o visualización, principalmente para:

- Displays numéricos.
- Regulación de iluminación.
- Apertura o cierre en % de ciertos conductos.
- Variadores de frecuencia, entre otros.

7.1. Salidas a relé

Probablemente, los autómatas con salidas a relé son los más empleados. Un relé es versátil; por sus contactos puede circular corriente continua, o alterna, y puede manejar valores superiores a 10 amperios. En su contra, la lentitud en las conmutaciones, y al emplear componentes mecánicos, éstos sufren desgaste.

En el siguiente gráfico aparecen dos modelos de salidas a relé. En el primer caso, cada salida es operada por un relé diferente, lo que permite usar diferentes tensiones en los receptores. El segundo caso, muestra una misma línea de alimentación para tres relés, las cuales tendrán obligatoriamente el mismo potencial.

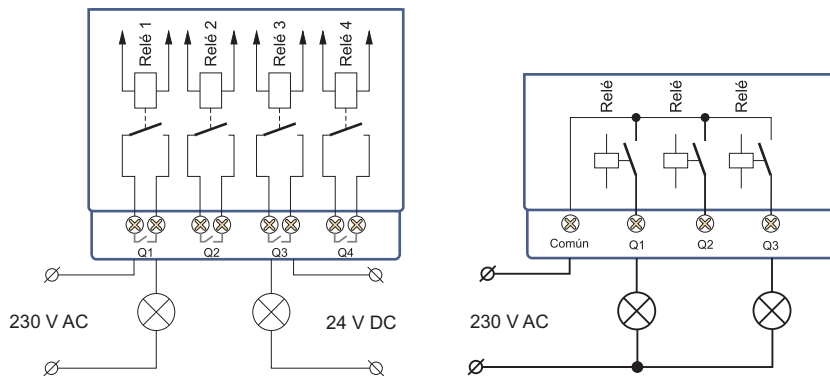


Figura 10.15. Dos modelos internos de salidas a relé.

7.2. Salidas a transistores

Los transistores son dispositivos de estado sólido. Son elementos electrónicos que no tienen partes móviles, por tanto, no tienen desgaste.

Los transistores conmutan corriente continua y son muy rápidos (incluso miles de veces por segundo). En su contra, la corriente de paso.

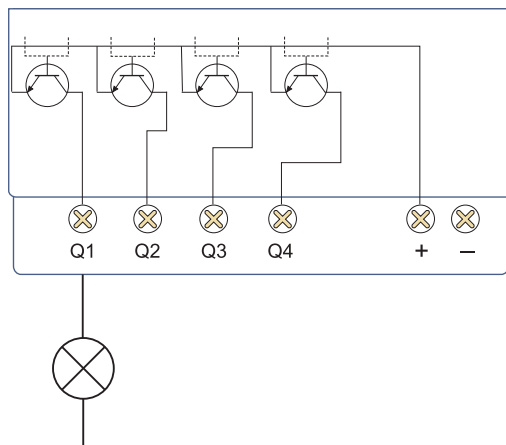


Figura 10.16. Autómata con salidas a corriente continua por transistor.

Acondicionamiento de la señal

Los acondicionadores incluidos en los sensores, son pequeños circuitos de carácter electrónico con la capacidad de entregar una señal de salida apta para ser usada. La principal cualidad de estos circuitos debe ser la correcta calibración.

Por ejemplo

Un termómetro analógico será capaz de emitir a su salida un valor de 0 a 10 V corriente continua, equivalentes a la temperatura de 0 a 100 °C.

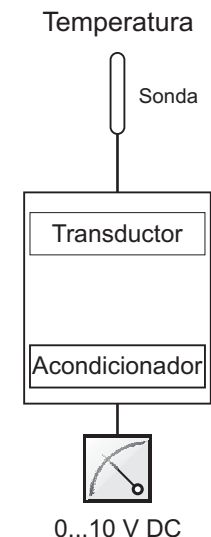


Figura 10.17. Proceso de acondicionamiento de la señal.

10 Autómatas programables

7.3. Salidas a triac

El triac es también un dispositivo de estado sólido sin partes móviles. Al contrario que el transistor, funciona con corriente alterna y se asemeja en la rapidez de sus conmutaciones. Las altas temperaturas son perjudiciales, tanto para triac como para transistor.

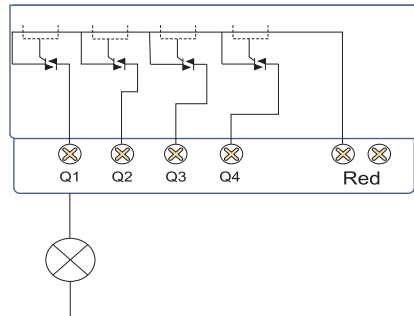


Figura 10.18. Autómata con salidas a corriente alterna por triac.

El contactor como recurso en las salidas del PLC

Si la carga a gobernar por el autómata programable es elevada, tanto para salida a relé, transistor o triac, se puede recurrir al empleo de un contactor y de esta forma, el PLC sólo tendrá que alimentar la bobina del contactor. La operación no debe presentar problemas técnicos ni eléctricos. Figura 10.19.

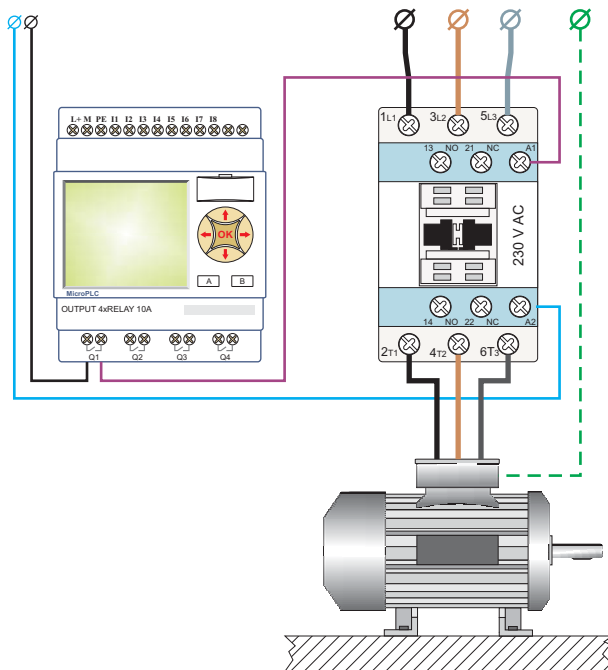


Figura 10.19. El autómata activa el contactor, y éste a su vez excita el motor.

7.4. Módulo de salidas analógicas

Generalmente, los valores de salida serán los estándar para señales analógicas (0...10 V DC; 0...20 mA; 4...20 mA...).

Note el ejemplo de la figura 10.21. La compuerta se abrirá de 0 a 100%, en cualquier posición, según la salida analógica de un PLC, con señal de tensión 0...10 V DC, donde 0 V DC es compuerta cerrada; 5 V DC, es compuerta abierta al 50%; 10 V DC, es compuerta abierta totalmente, por citar algunos valores.

Módulos de entradas/salidas analógicas

Estadísticamente, y por necesidades de las aplicaciones industriales, se consumen más módulos de entradas analógicas que salidas analógicas. Por este motivo, es típico encontrar módulos independientes con las siguientes características:

- Módulo 2xAI. Dos entradas analógicas (0-10 V DC y 4-20 mA).
- Módulo 4xAI. Cuatro entradas analógicas (0-10 V DC y 4-20 mA).
- Módulo AQ. Salida analógica 12 Bits 10 V DC).
- Módulo 4 AI X 1 AQ. Cuatro entradas analógicas más una salida analógica.

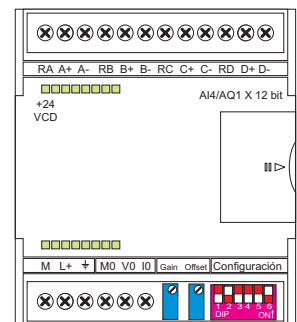


Figura 10.20. Módulo con 4 entradas y una salida analógicas.

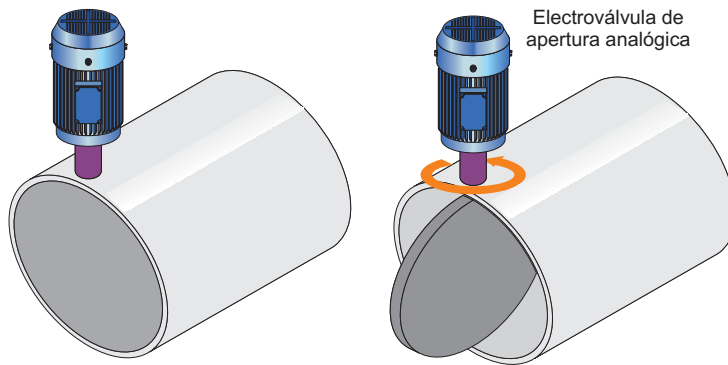


Figura 10.21. La compuerta se abrirá con valores analógicos exactos.

Pilas internas

Los autómatas programables, al igual que los PCs, incorporan una pequeña pila, para que en el momento de apagar la máquina, no se borren los datos.

Los autómatas compactos la suelen llevar en el interior del encapsulado, mientras que los modulares, disponen de una tapa exterior, que permite comprobar su estado y sustitución de forma cómoda.

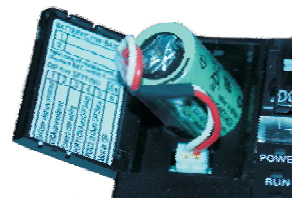


Figura 10.22. Ejemplo de pilas.

8 Unidad central de proceso CPU

Es el cerebro en el más amplio sentido de la palabra del autómata programable. La CPU procesa todas las señales que provienen de los captadores de entrada, los evalúa consultando al programa de aplicación y ordena a las salidas que efectúen sus oportunas aplicaciones.

Consideremos que un autómata tiene un sistema operativo, que es un programa interno que permite que el PLC pueda realizar las operaciones. Este sistema operativo no es modificable por el usuario y viene instalado de fábrica. Asimismo, el programa de usuario que es el programa en el cual el operador diseña y realiza la programación, es modificable y borrable. El programa de usuario se realiza mediante una unidad de programación -por ejemplo un PC- y se transfiere al autómata mediante un cable de conexión; esta acción no es definitiva, es decir, se puede modificar el programa, hasta que el autómata realice las especificaciones deseadas. El encargado de coordinar el programa de usuario con los datos de los sensores de entrada y ordenar que las salidas se activen, es el sistema operativo.

El programa de usuario se realiza con un lenguaje específico más o menos fácil (dependiendo de la profundidad de la programación) que puede ser textual o gráfico. Sin duda, los usuarios acostumbrados a utilizar esquemas eléctricos, prefieren los lenguajes gráficos. Esto no quiere decir que el PLC reconozca esos gráficos que suponen la programación que el autómata tiene que cumplir. Cuando la unidad de programación transfiere el programa -realizado con cualquier lenguaje- los datos que viajan al PLC y por tanto, los que reconoce, están en un lenguaje denominado lenguaje máquina.

Dentro de la CPU encontramos principalmente el microprocesador (μP) y las memorias. El microprocesador que habitualmente no es una unidad física sino varios circuitos integrados, realiza funciones de comparación, cálculo aritmético, entre otros, y en definitiva, la transferencia de información dentro del autómata.

Dentro de éste, aparecen circuitos que desarrollan diferentes funciones: la unidad de control, la unidad aritmética lógica y un circuito de registros.

8.1. Memorias

Las memorias son unos elementos capaces de almacenar información en forma de ceros y unos (bits). Existen varios tipos de memoria:

Memoria RAM; (random access memory) memoria de lectura y escritura. Es el lugar donde reside el programa de usuario. Es requerida por el microprocesador para consultar el programa. Por tanto, es modificable en todos sus aspectos: programación, desprogramación, borrado.

10 Autómatas programables

Si falla la corriente de alimentación se pierde la información que posee; para evitar esto, el autómata está dotado de una pila para que no le falte alimentación a estos componentes.

Memoria EPROM; memoria de sólo lectura, es borrable y programable. Esta memoria es parecida a la ROM (Read Only Memory) pero modificable, esto es para poder cambiar el sistema operativo del autómata por otra versión. En caso de falta de alimentación, no se borra su contenido.

Además, las memorias EPROM Y EEPROM, se dedican a la grabación y archivo de programas. La memoria EEPROM es parecida a la EPROM, pero se programa y se borra eléctricamente mediante una unidad de programación.

La memoria FLASH, derivada de la EEPROM, también utilizada, en PLCs, tiene mayor facilidad (velocidad) para el borrado de datos.

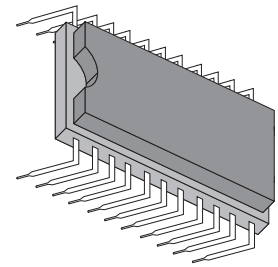


Figura 10.25. Memorias.

8.2. Datos del PLC

Si un pulsador está activado o desactivado, esa información la procesa el autómata como un dato, y ese dato será usado para realizar unas operaciones u otras, según la programación dispuesta para esa entrada en la que está conectado el pulsador.

Un dato ocupa una posición de memoria; ésta puede usar 8 posiciones que son independientes, pero que comparten la misma dirección. Hablamos sólo de ceros y unos (0 y 1).

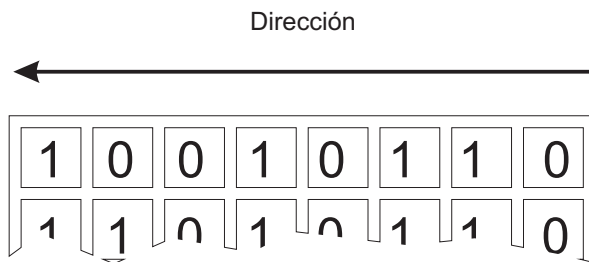


Figura 10.23. Ejemplo de almacén de datos en formato de 8 bits.

Cada cero o uno (0 ó 1) se llama bit. Ocho bits (b) forman un byte (B). El uso del byte es lo que habitualmente determina la capacidad de memoria de un dispositivo, así si decimos que un aparato tiene una memoria de 4 KB, estaremos diciendo que puede almacenar $4 \times 1024 = 4096$ Bytes; ya que 1 KB equivale a 1024 Bytes.

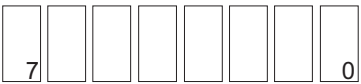
En autómatas programables se habla de palabras (Word) cuando la información o datos utiliza dos bytes de memoria y de palabras dobles (Double Word) cuando la información utiliza cuatro bytes.

Además de lo anterior, existen otros formatos de mayor calibre, como el REAL, de 32 bits, que permite operaciones más precisas.

Un bit es la unidad más pequeña (0 ó 1)



Un conjunto de 8 bits es un byte (B)



Un conjunto de 2 bytes es una palabra (Word)



Un conjunto de 4 bytes es una doble palabra (DW)



Figura 10.24. Representación de bit, byte, palabra y doble palabra.

PLC compacto

Aun así, el usuario en las aplicaciones puede operar con la unidad más pequeña, bit. Pongamos un ejemplo de ocupación de memoria; si un sensor o pulsador es presionado, y da señal eléctrica a una entrada del autómata, esta orden estará ocupando un bit, que indicará ocupado o no ocupado, si le aplicamos corriente o no. Con lo cual, la programación podría ser la siguiente: si el bit 6 tiene valor de 1, entonces que se active la salida 1.

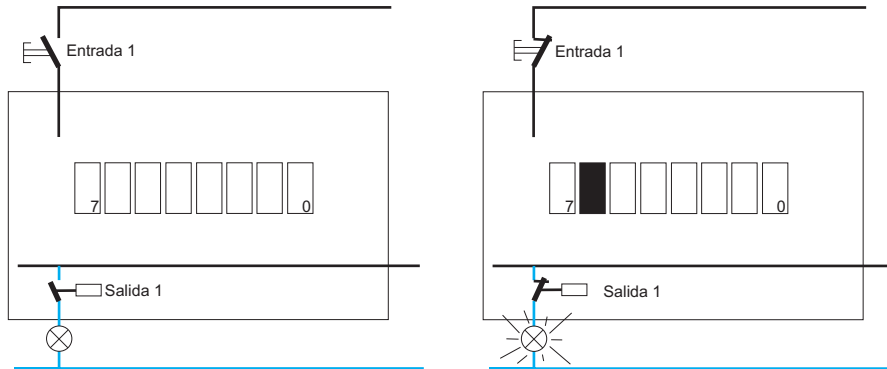


Figura 10.26. "Uso de un bit de entradas, y un bit de salidas".

Este ejemplo, muestra como con un solo bit, podemos realizar una orden de trabajo. Esto no es siempre así, ya que temporizadores, relojes, contadores, contadores rápidos, necesitan ocupar porciones de memoria más grandes (palabras, 16 bit; dobles palabras, 32 bits). Cuando ha de buscarse un dato de la memoria, o guardarlo en ella, tenemos que saber en qué lugar está. Para ello, cada ubicación física tiene una dirección o registro concreto.

Para programar señales procedentes de sensores de entrada o actuadores, tenemos que asignarles una dirección de memoria que ocuparán dentro del autómata. Esta dirección puede ser fija o variable.

- Direccionamiento fijo, es cuando la dirección de una entrada, salida o incluso una marca interna es siempre la misma, esto ocurre en pequeños controladores programables.
- Direccionamiento variable, es cuando la dirección de una entrada, salida o marca interna, se puede adjudicar a voluntad, dentro de los límites del programa.

En la siguiente figura, aparece un módulo de entradas digitales de un micro-controlador programable. Note, que la entrada I1 es inequívoca, es decir, siempre será I1, tanto para el cableado, como para la programación. Para las salidas, lo mismo. También se encuentra un módulo de entradas, con bornes sin determinar, y en el programa, se le ha asignado a una la dirección de entradas 122.1.

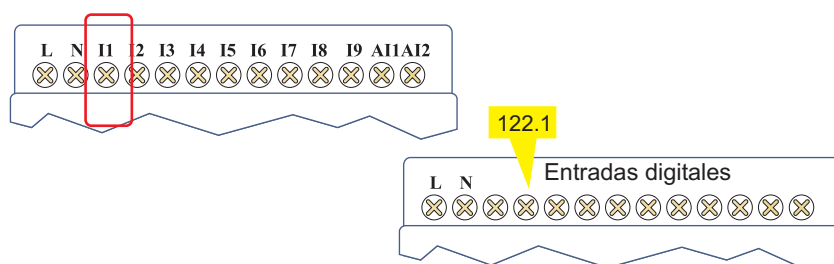


Figura 10.27. Direccionamientos fijo y variable.

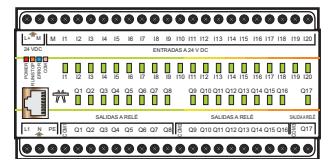


Figura 10.28. PLC compacto.

10 Autómatas programables

8.3. Ciclo de programa

Un ciclo de programa es un proceso en el cual el autómata realiza un análisis del estado de las entradas, consulta al programa, y actualiza el estado de las salidas en función de la información que ha obtenido del análisis de entradas y consulta al programa. Esta operación se realiza periódicamente, y se le llama tiempo de ciclo al tiempo que transcurre durante un ciclo de programa, que ronda los milisegundos, aunque dependerá de la máquina.

- 1- Se analiza el estado de las entradas; las que están activadas (On) y las que están desactivadas (Off), a continuación esa información se guarda de forma temporal en un archivo de memoria denominado imagen de entradas .
- 2- Se analiza y ejecuta el programa de usuario; esto es que el autómata consulta las instrucciones en el programa lógico, y usando el estado de las entradas (el estado que se almacena temporalmente en el archivo imagen de entradas) decide que salida debe ser activada o no. El resultado que tendrán las salidas ahora, se guarda en otro archivo de memoria denominado imagen de salidas .
- 3- Análisis de salidas. Según los datos que se encuentren en la imagen de salidas, el autómata conectará o desconectará los circuitos de salida, realizando de esta forma la gestión de los dispositivos de conexión a los terminales de salida.



Figura 10.29. Ciclo operativo de programa -esquema-.

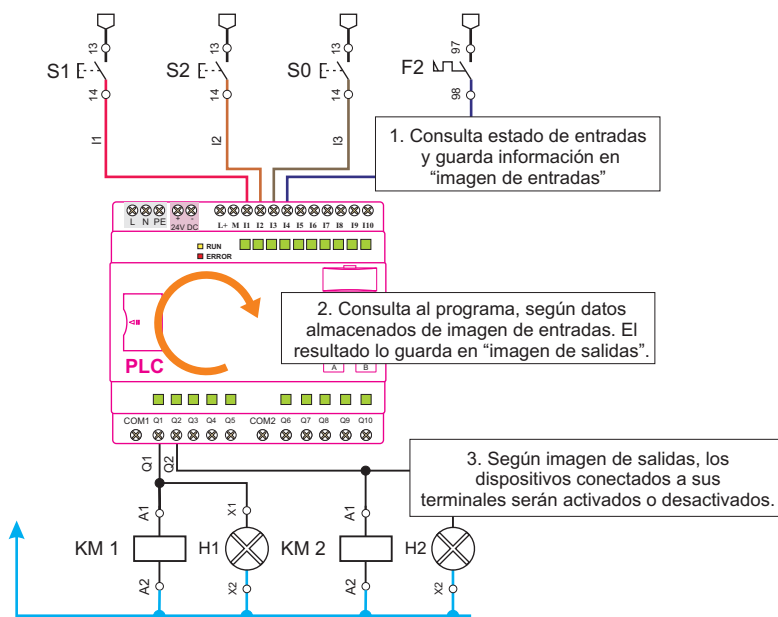


Figura 10.30. Ciclo operativo de programa.

PLC modular

Monta por separado todos los componentes que lo hacen funcionalmente operativo, desde la fuente de alimentación, CPU, memorias, módulos de entradas y salidas digitales, módulos de entradas y salidas analógicas, módulos de comunicaciones, módulos de puertos específicos (por ejemplo, para conexión a dispositivos HMI), principalmente.

Se programará exclusivamente con una unidad específica (un ordenador personal también). A su favor, crece según las necesidades de la instalación.



Figura 10.31. Esquema de bloques. PLC modular.

9 Fuente de alimentación

Es el mecanismo que proporciona la alimentación eléctrica a los componentes internos del autómata programable. La tensión eléctrica que le llega a éstos, será menor que la de red. También, la fuente de alimentación protegerá al autómata contra alteraciones eléctricas en la red.

El dispositivo puede ser interno o externo, y además de alimentar al autómata programable, puede hacerlo hacia dispositivos periféricos como pantallas HMI, o incluso sensores; por este motivo es importante calibrar bien sus dimensiones.

De la fuente de alimentación (si es externa) debemos conocer principalmente la tensión de alimentación, la intensidad de trabajo, y en su caso, la temperatura ambiente que soportará el dispositivo.

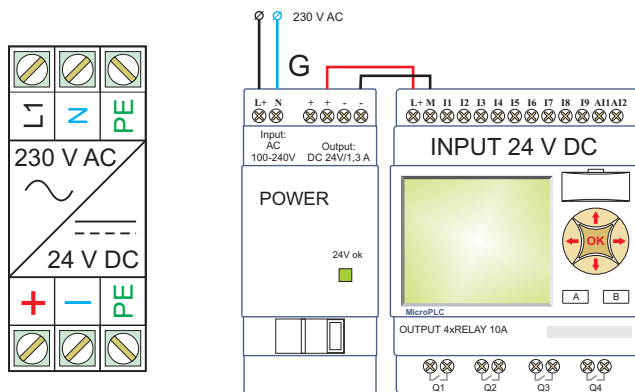


Figura 10.32. Fuente de alimentación.

10 Periféricos

Los periféricos son los dispositivos que se conectan a un autómata programable, y que completan, facilitan y amplían, las labores de automatización. Destacamos entre otros:

- Impresoras. Además de imprimir los programas de usuario, permitirán crear entre otros, históricos de avisos, averías, paradas no programadas, etc.
- Cartuchos de memoria EEPROM. Para guardar el programa de usuario en el propio PLC.
- Visualizadores y pantallas táctiles. Son dispositivos hombre-máquina que sustituyen pulsadores y avisadores convencionales. Con estos dispositivos se podrá controlar uno o varios procesos en tiempo real, incluso se podrá simular la operación en la pantalla, por ejemplo, llenado de botellas.
- Pasarelas de comunicación. Módem GSM, conexión a ethernet, internet, etc. Permiten exportar, importar y compartir los datos que procesa el PLC.

11 Dispositivos de programación de autómatas programables

Originalmente, existían consolas de programación para autómatas programables, aunque hoy prácticamente se utilizan ordenadores personales para esta labor, por varios motivos:

- El software de programación está disponible prácticamente en todos los sistemas operativos.
- Se evita el problema en la programación de PLCs de diferentes firmas (cada uno con su consola específica).

Símbolo F.A.

La fuente de alimentación se identifica por la letra "G", referido al grupo de generadores y dispositivos de alimentación.

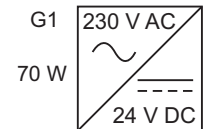


Figura 10.33. Símbolo de fuente de alimentación.

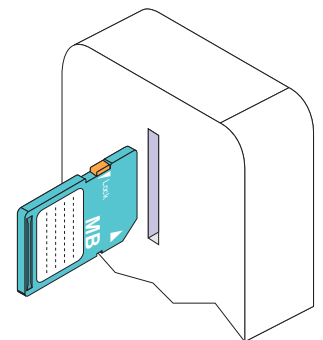


Figura 10.34. Los PLCs, permiten la inserción de una tarjeta de memoria.

10 Autómatas programables

- El mismo software de programación, sirve en algunos casos para simular y parametrizar una instalación de manera virtual, lo que evita tiempo y dinero en la puerta en marcha de los sistemas.

Las consolas siguen existiendo, aunque algo diferentes; están basadas en PC, son muy potentes, y constructivamente son más robustas que un ordenador personal.

¿Para qué sirven?

- Principalmente para crear los programas de funcionamiento que el autómata debe realizar; en ellos, se utilizarán una serie de instrucciones determinadas y se empleará un lenguaje de programación también determinado, pues hay varios tipos.
- Para simular el programa y depurar errores.
- Para transferir el programa al PLC. Se usará un puerto serie, USB o ethernet, preferentemente.
- Para visualizar on-line el funcionamiento. Esto es especialmente importante para depurar errores y realizar ajustes en tiempo real, además de la posibilidad de realizar autochequeos.

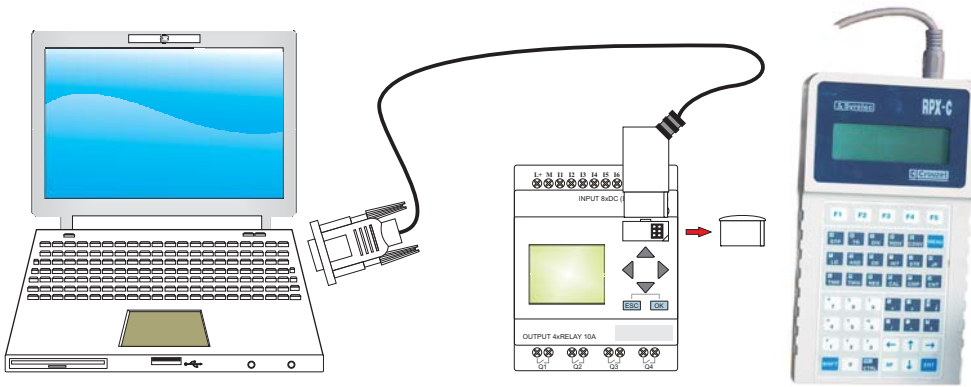


Figura 10.35. Transferencia PC-PLC y consola de programación. (Crouzet).

Cables de programación

No existe un modelo estándar para cable de programación y puesta en marcha de los autómatas programables, por tanto, se hace necesaria la adquisición de un cable específico para un modelo o gama de PLCs.

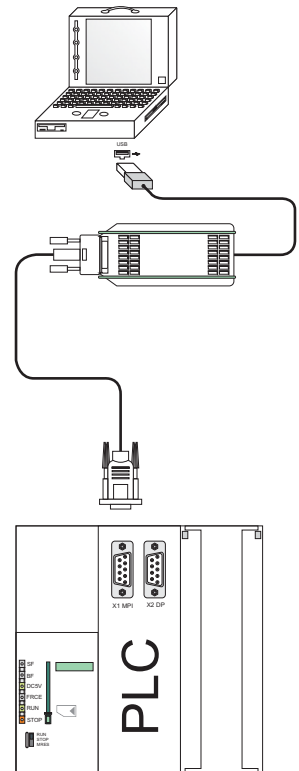


Figura 10.37.

Cable específico para programar PLCs.

12 Variables

En un autómata programable, y en programación en general, una variable es un lugar donde se guardan cierto tipo de datos. Estos datos podrán ser diversos; textuales, imágenes, sonido, etc. En autómatas programables, los datos se guardan en formato tipo: bit, byte, palabra y doble palabra preferentemente, aunque este concepto depende de la tecnología de la firma comercial. Es fundamental comprender que una variable de un tipo, se debe guardar sólo en el lugar destinado a ello, por ejemplo, un dato de Palabra (16 bits), no se puede ubicar en una variable de bit. La llamada a una variable es inequívoca, de tal forma que no existirán dos variables con el mismo nombre.

Entradas digitales	1	0	0	1	0	1	1	0
Salidas digitales	1	1	0	1	0	1	1	0
Marcas	1	1	1	1	0	1	1	0
Entradas analógicas	1	0	0	1	0	1	1	0
Salidas analógicas	0	0	0	1	0	0	0	0
Temporizadores	1	0	1	1	0	1	1	0
Contadores	1	0	0	1	0	1	1	0
Etcétera.	0	0	1	1	1	0	0	1

Figura 10.36. Representación abstracta del almacén de datos.

Ejemplo de variables (tenga en cuenta, que no son equivalentes a todos los autómatas programables, por tanto, sirvan sólo de referencia).

12.1. Variables de entrada digitales

Las variables digitales que relaciona el autómata con los dispositivos de entrada se identifican como **I** de input, por ejemplo: I1, entrada 1; I4 entrada 4; I0.0.1 entrada del módulo 0, del bastidor 0, entrada 1, etc. Estas variables operan con datos tipo bit (0 ó 1), todo o nada, activado o no activado. También se llaman datos Booleanos.

12.2. Variables de entrada analógicas

Las variables analógicas necesitan más capacidad de almacenaje, ya que los valores equivalentes a la magnitud medida pueden ser infinitos. Por ejemplo, la variable AI1 (Analogic Input 1), utilizará almacén de datos de 10 bits. Otro ejemplo, un anemómetro medirá la velocidad del viento a través de la variable AI2 de 16 bits.

El autómata programable dispondrá de las herramientas necesarias para poder convertir, transferir, y en definitiva operar con los diferentes tipos de datos.

12.3. Variables de salida digitales

Las variables de salida son de estado 1 / 0, variables booleanas. Las variables **Q** (Output), se identifican con los dispositivos a los que el autómata alimentará, por tanto, si leemos Q2, entendemos que es la segunda variable de salida y lo que hubiera conectado a ese conector, será excitado.

12.4. Variables de salida analógicas

Como ya ocurre con las entradas analógicas, el valor de salida se mueve en un margen muy amplio de datos, desde 10 a 32 bits, preferentemente. Por ejemplo, una salida analógica regulará la intensidad del alumbrado, a través de la salida AQ4 de 12 bits.

12.5. Variables de marcas (memorias internas)

Prácticamente todos los autómatas programables, disponen de unas variables internas, que sirven de apoyo en los programas. Las variables **M** de memoria (o marca), hacen referencia a salidas internas dentro del PLC; son salidas que no tienen una actuación visible directamente, y actúan como relés dentro del propio dispositivo. Además, las variables **M** pueden ocupar direcciones de memoria mayores, como MB -marca de byte de 8 bits-, MW -marca de palabra de 16 bits-, entre otros.

12.6. Variables específicas

Las que operan con datos internos del PLC, como temporizadores, contadores, contadores rápidos, etc.

Áreas de datos

La tabla que se muestra a continuación es meramente ilustrativa, ya que no existe un modelo común de área de datos para los diferentes autómatas, inclusive dentro de una misma firma comercial.

Área de datos	Identificador
Entradas Digitales	I
Entradas analógicas	AI
Salidas digitales	Q
Salidas analógicas	AQ
Marcas	M
Contadores	C
Contadores rápidos	HC
Temporizadores	T
Relojes	T

Tabla 10.2. Área de datos.

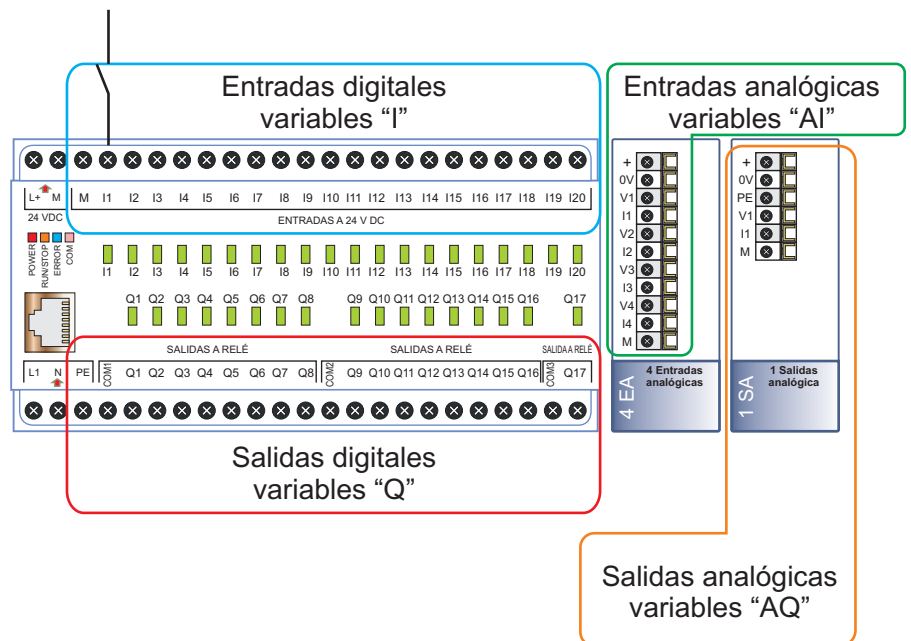


Figura 10.38. Disposición de las variables "físicas" del autómata programable.

13 Álgebra de Boole

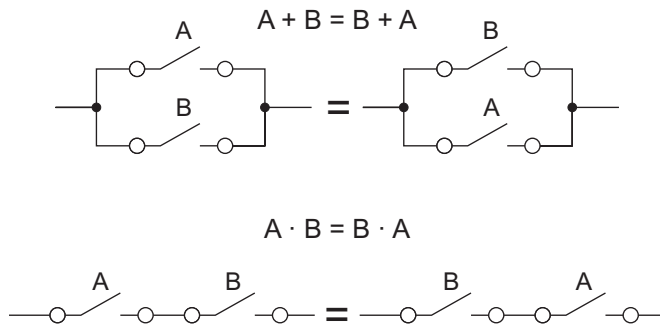
Álgebra, en el que sus componentes sólo pueden tomar dos estados. Se considera un álgebra especial para el sistema binario. Proporciona el modo de expresar el funcionamiento de un circuito lógico compuesto por la combinación de puertas lógicas, en la que se puede saber el resultado de la salida, según la combinación de los valores de entrada.

El álgebra está definido por tres operaciones binarias fundamentales:

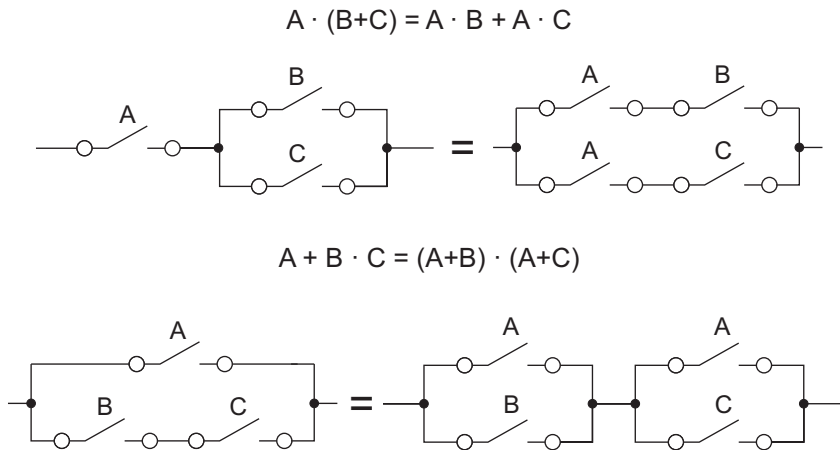
- Suma (+), equivalente a una asociación en paralelo (OR).
- Multiplicación (·), equivalente a una asociación en serie (AND).
- Complemento lógico o negación (NOT).

Se considera, que una operación es álgebra de Boole sólo si se cumplen los 4 postulados siguientes:

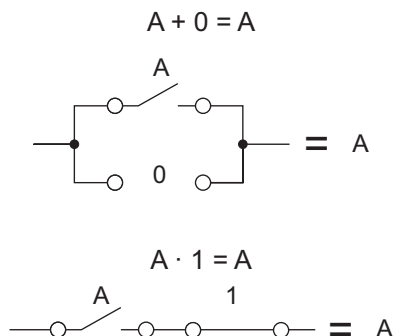
Propiedad CONMUTATIVA:



Propiedad DISTRIBUTIVA:



Existencia de elementos neutros (0 y 1), para las operaciones (+ y ·), respectivamente.



George Boole

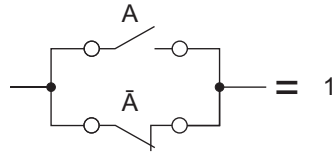
El creador del álgebra de Boole fue el matemático inglés George Boole (2-11-1815 a 8-12-1864).

Se publicó por primera vez en 1847 en un documento llamado "The Mathematical Analysis of Logic".

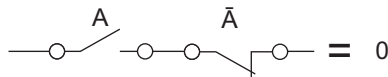
El álgebra de Boole está definida por unos postulados y teoremas.

Para cada elemento A, existe su complemento $\bar{A}$

$$A + \bar{A} = 1$$



$$A \cdot \bar{A} = 0$$



Postulado y teorema

Postulado

Proposición, cuya verdad se admite sin pruebas, aunque éstas son necesarias para posteriores razonamientos. Se admite, mientras no exista razón definitiva.

Teorema

Proposición demostrable.

13.1. Tabla de la verdad

Es una tabla que muestra una serie de valores claramente ordenados y ejecuta uno o varios resultados de la relación entre ellos.

Ejemplo 1. Contacto abierto.

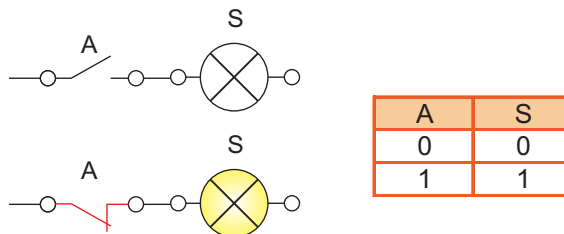


Figura 10.39. Tabla de la verdad de un solo contacto NA.

Ejemplo 2. Dos contactos en paralelo. Operación $A + B$.

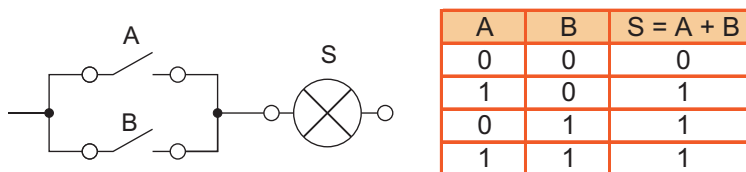


Figura 10.40. Tabla de la verdad de dos contactos NA en paralelo.

Ejemplo 3. Dos contactos en serie. Operación $A \cdot B$.

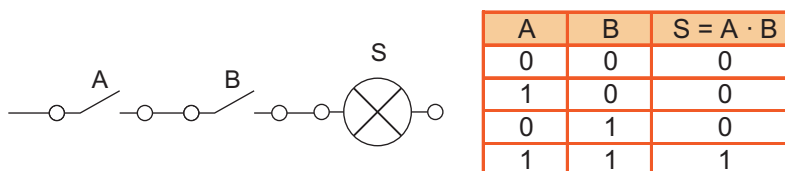
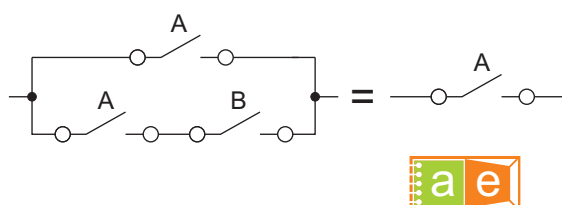


Figura 10.41. Tabla de la verdad de dos contactos NA en serie.

13.2. Teoremas del Álgebra de Boole

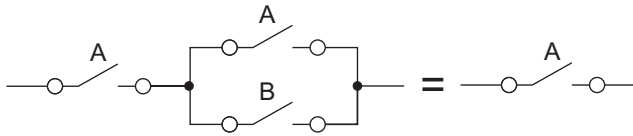
Principio de dualidad. Cualquier identidad algebraica, deducible a los postulados del Álgebra de Boole, permanece válida, si cambiamos $+$ por $\cdot$ y 1 por 0 .

Teorema 1. $A + A \cdot B = A$

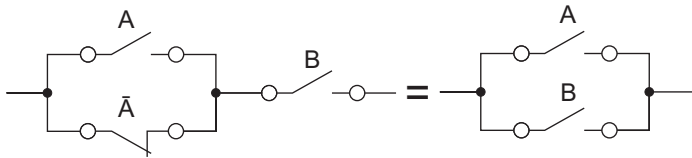


10 Autómatas programables

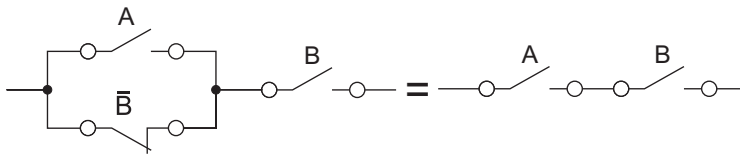
Teorema 2. $A \cdot (A + B) = A$



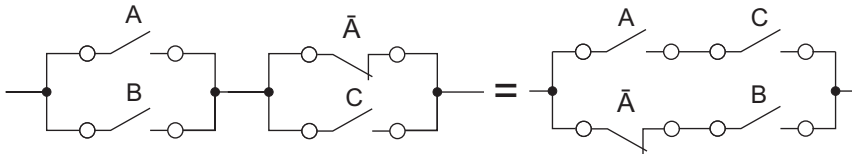
Teorema 3. $A + \bar{A} \cdot B = A + B$



Teorema 4. $(A + \bar{B}) \cdot B = A \cdot B$



Teorema 5. $(A + B) \cdot (\bar{A} + C) = A \cdot C + \bar{A} \cdot B$



Teoremas de Morgan

Teorema 6. Primer teorema de Morgan. $\overline{A + B} = \bar{A} \cdot \bar{B}$

A	B	$\bar{A}$	$\bar{B}$	$A + B$	$\overline{A + B}$	$\bar{A} \cdot \bar{B}$
0	0	1	1	0	1	1
1	0	0	1	1	0	0
0	1	1	0	1	0	0
1	1	0	0	1	0	0

Tabla 10.3. Tabla de la verdad del teorema 6, primer teorema de Morgan.

Teorema 7. Segundo teorema de Morgan. $\overline{A \cdot B} = \bar{A} + \bar{B}$

A	B	$\bar{A}$	$\bar{B}$	$A \cdot B$	$\overline{A \cdot B}$	$\bar{A} + \bar{B}$
0	0	1	1	0	1	1
1	0	0	1	0	1	1
0	1	1	0	0	1	1
1	1	0	0	1	0	0

Tabla 10.4. Tabla de la verdad del teorema 7, segundo teorema de Morgan.

Teoremas de Morgan

Dentro del álgebra de Boole, los teoremas de Morgan, son ampliamente utilizados en operaciones binarias.

Primer teorema:

El complemento (o la inversa), de un producto lógico de variables, es igual a la suma lógica de las inversas de las variables.

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

Segundo teorema:

El complemento (o la inversa), de una suma lógica de variables, es igual al producto lógico de las inversas de las variables.

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

13.3. Puertas lógicas

Para representar las funciones lógicas, se utilizan unos símbolos llamados puertas lógicas. Dentro de este bloque funcional se realizará una operación, que produce un resultado sobre una salida.

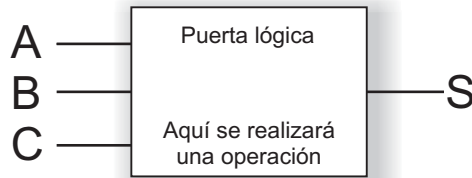


Figura 10.42. Puerta lógica.

13.3.1 Función NO o inversión NOT

La salida Q, tomará el valor contrario a la entrada.

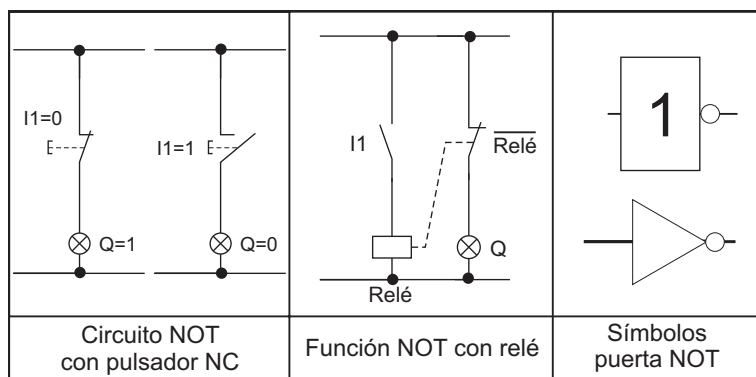


Figura 10.43. Función NOT.

13.3.2. Función Y, AND

La salida tendrá valor 1, cuando todas las variables de entrada tengan valor 1.

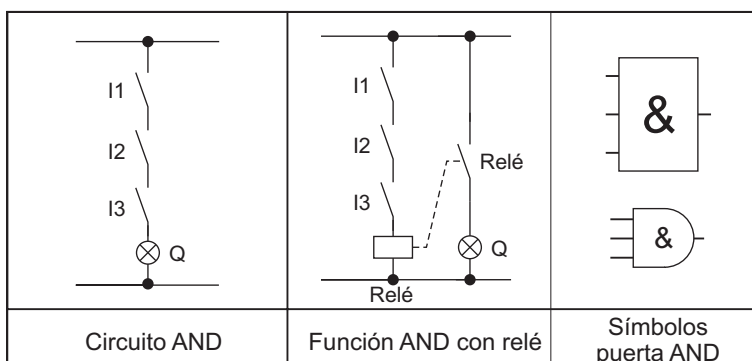


Figura 10.44. Función AND.

Puerta AND

Si una función AND requiere de más variables de las que admite la "puerta", se pueden asociar, siendo el mismo efecto que si fuera una única puerta.

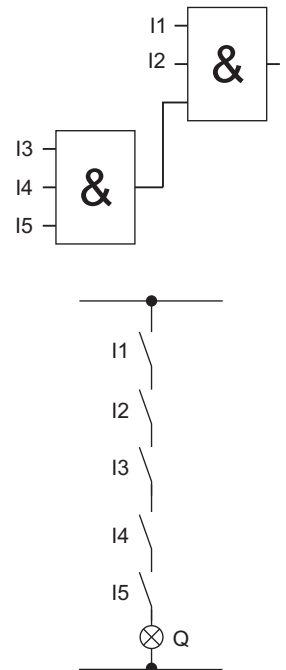


Figura 10.45. Asociación de puertas lógicas de la misma clase.

I1	I2	I3	Q = I1 · I2 · I3
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Ejemplo

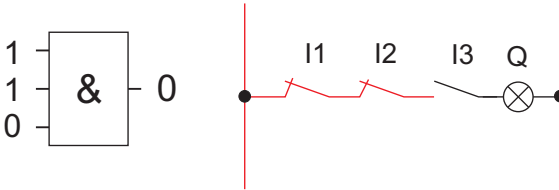


Figura 10.46. Tabla AND y ejemplo.

13.3.3. Función O, OR

La salida tendrá valor 1 , si cualquiera de las variables de entrada tiene valor 1 .

Circuito OR	Función OR con relé	Símbolos puerta OR

I1	I2	I3	Q = I1 + I2 + I3
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Ejemplo

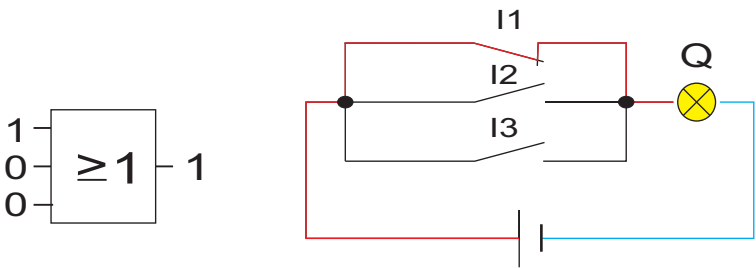


Figura 10.47. Función OR.

Puerta OR

Si una función OR requiere de más variables de las que admite la “puerta”, se pueden asociar, siendo el mismo efecto que si fuera una única puerta.

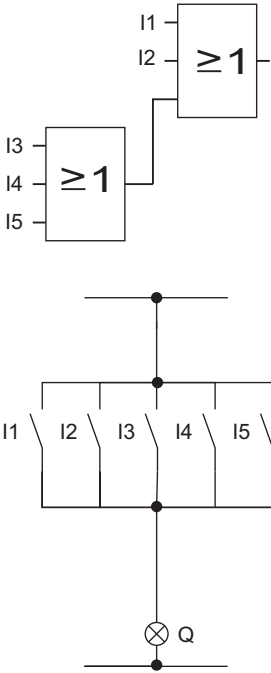
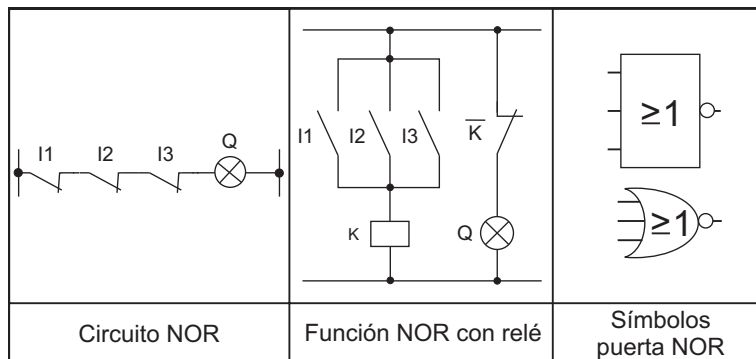


Figura 10.48. Asociación de puertas lógicas de la misma clase.

13.3.4. Funciones lógicas inversas. NO-O; NOR

La función inversa NOR realiza la misma operación que la función OR, sólo que invierte el resultado.



I1	I2	I3	$Q = \overline{I1 + I2 + I3}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Ejemplo

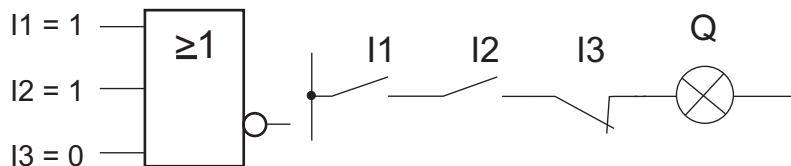


Figura 10.49. Función NOR.

13.3.5 Funciones lógicas inversas. NO-Y, NAND

La función inversa NAND realiza la misma operación que la función AND, sólo que invierte el resultado.

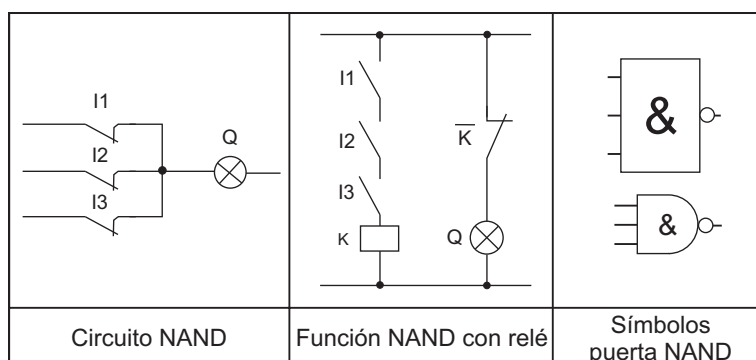


Figura 10.50. Función NAND.

Circuitos integrados

Las puertas lógicas se utilizan para el diseño y definición de circuitos integrados. En la figura, aparece el 7400.

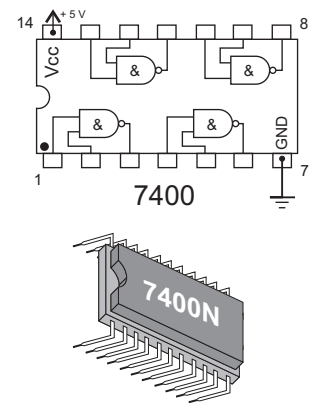


Figura 10.51. Circuito integrado.

10 Autómatas programables

I1	I2	I3	$Q = \overline{I1 \cdot I2 \cdot I3}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Ejercicio

¿Podrías dibujar un circuito eléctrico, que cumpla con la siguiente tabla de la verdad?

I1	I2	Q1
0	0	1
0	1	0
1	0	0
1	1	1

Tabla 10.5.

Ejemplo

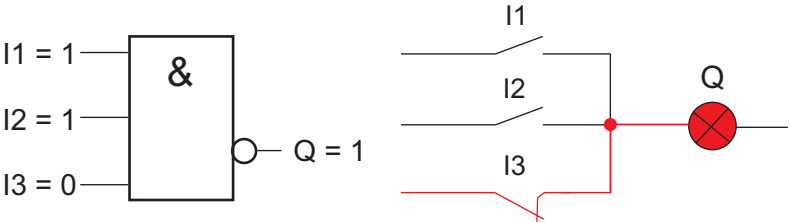
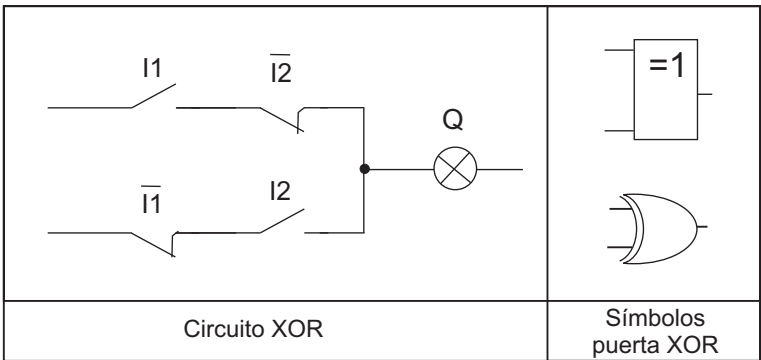


Figura 10.52. Tabla NAND y ejemplo.

13.3.6. Funciones lógicas especiales. OR-Exclusiva; XOR

En primer lugar, esta función sólo tiene dos variables de entrada, y la salida adoptará el valor de 1 cuando el estado de las dos variables sea diferente.



I1	I2	$Q = I1 \oplus I2$
0	0	0
0	1	1
1	0	1
1	1	0

Ejemplo

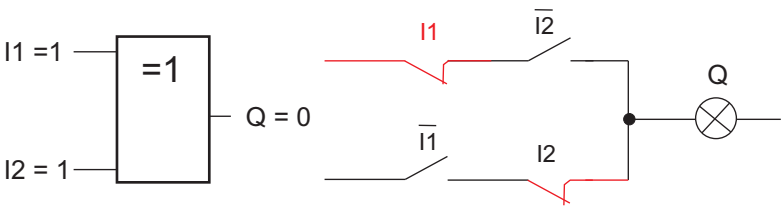


Figura 10.53. Función XOR.

13.3.7. Aplicaciones de las puertas lógicas

Como se verá más adelante, las puertas lógicas además de servir de base en el desarrollo de circuitos electrónicos, permite crear un lenguaje de programación para autómatas programables.

Note el siguiente circuito eléctrico, consistente en dos interruptores en serie (S1 y S2), y a su vez, en paralelo con un tercer interruptor (S3). A la derecha se muestra la equivalencia mediante puertas lógicas.

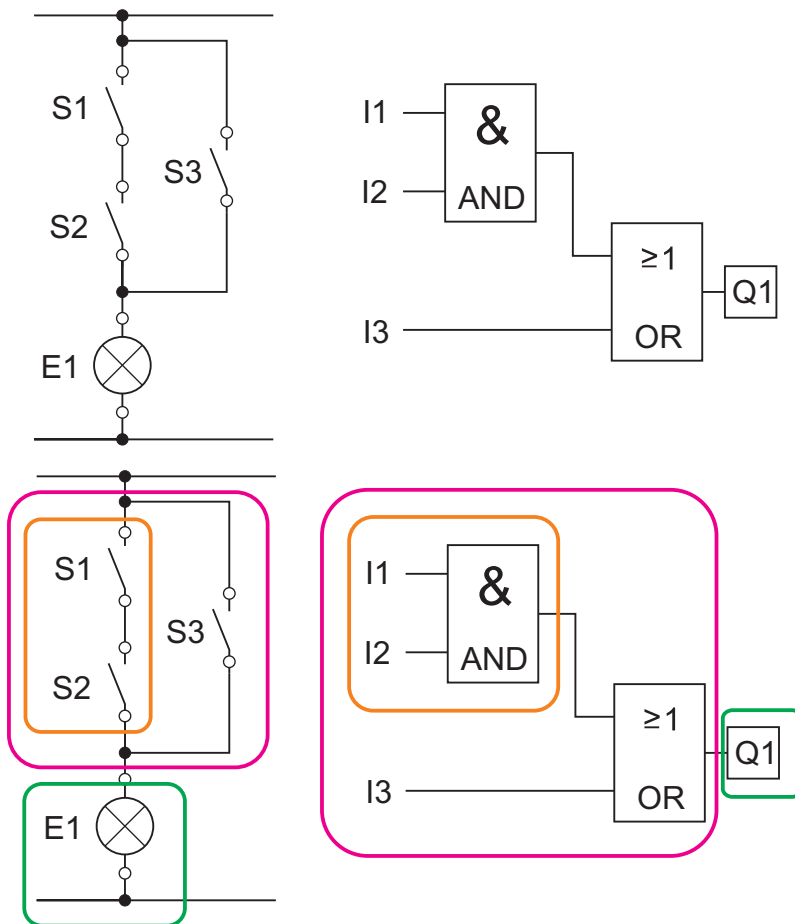


Figura 10.54. Equivalencia entre circuito eléctrico y puertas lógicas.

Tabla de la verdad

Tabla de la verdad del circuito propuesto y equivalencia de sus componentes

I1	I2	I3	Q1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Circuito eléctrico	Funciones lógicas
INTERRUPTOR 1 S1	I1
INTERRUPTOR 2 S2	I2
INTERRUPTOR 3 S3	I3
LÁMPARA E1	Q1

Tablas 10.6 y 10.7.

14 Lenguajes de programación

Para configurar las acciones que ha de desarrollar el autómata programable, el diseñador dispone de unas herramientas que le van a permitir además de programar: diagnosticar, modificar, comprobar, simular las aplicaciones, incluso antes de volcarlas al PLC. El lenguaje de programación maneja el juego de instrucciones que realizarán las funciones lógicas y de cálculo de la unidad central de proceso.

Aunque cada firma aporta funciones concretas de la marca, prácticamente todos los sistemas de programación se acogen a la norma IEC 1131-3, que propone lenguajes de programación comunes para todos. Esto supone para el programador, que las operaciones entre unas marcas comerciales y otras, no le serán desconocidas.

De tediosas consolas de programación hemos pasado a entornos gráficos muy orientativos, que hacen que la programación sea una tarea accesible a técnicos que no tienen por que ser expertos en la materia. Tutoriales interactivos, asistentes, y ayudas directas en las funciones, se incorporan a los programas.

De los cinco lenguajes que define el estándar (cuatro más GRAFCET), tres se desarrollan en forma gráfica y dos en forma textual. Es destacable que algunos sistemas de programación permiten el cambio de un lenguaje de programación a otro de forma directa, facilitando más la labor del programador.

14.1. Funciones de programación

Antes de analizar los lenguajes de programación, se hará un repaso a las principales funciones de programación. La memoria operativa y la riqueza de las funciones dentro de una máquina permite el desarrollo de las grandes instalaciones automatizadas.

Las funciones que se representan a continuación, son acogidas prácticamente por todos los lenguajes de programación, por tanto, se hace necesario el conocimiento de las mismas.

14.1.1. Temporizadores

Temporizador con retardo a la activación TON

Este temporizador activará su salida, si es excitada su entrada (IN) y pasa el tiempo programado, aunque si la entrada es igual a cero, la salida también lo será. Normalmente, de este temporizador usaremos dos variables; el bit digital de salida (T), que podrá ser 0/1, y el dato del tiempo actual, medido entre otros, en formato de 16 bits (Word).

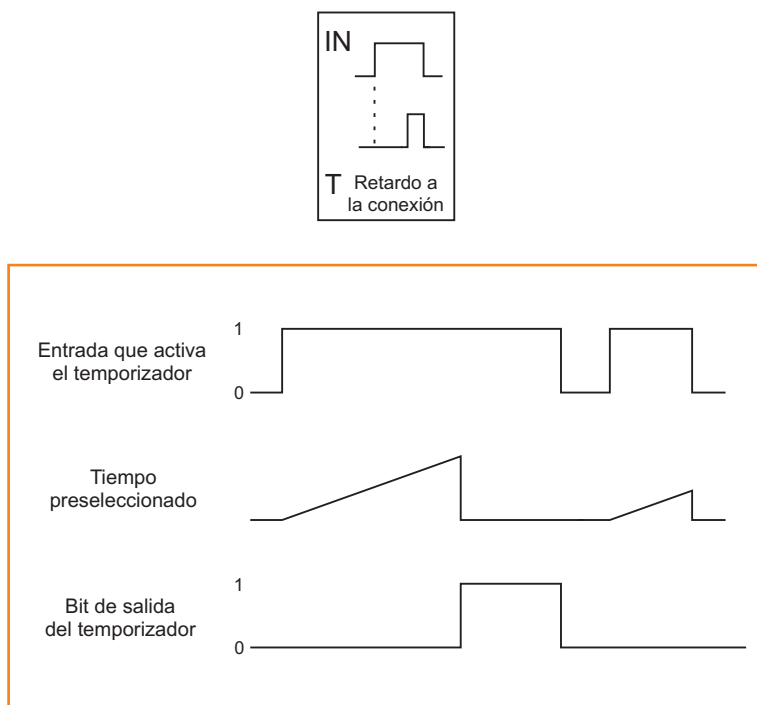


Figura 10.55. Temporizador con retardo a la activación y cronograma.

IEC 1131-3

La norma IEC-1131-3, es el resultado del esfuerzo mostrado por las multinacionales del sector para que los lenguajes de programación de autómatas programables, tengan elementos comunes. Esto supone, que el programador podrá acometer proyectos de diferentes firmas, siguiendo criterios similares en el diseño y programación.

La norma establece cuatro lenguajes de programación, dos textuales, dos gráficos, además del GRAFCET.

Ejemplo

El pulsador S1 activa el temporizador con retardo a la activación. ¿Qué pasará a los 10 segundos de activar la entrada del citado temporizador?

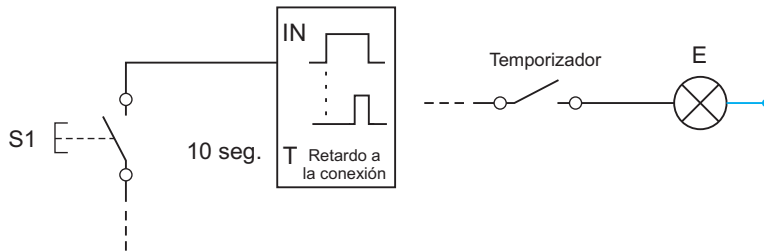


Figura 10.56. Ejemplo del temporizador TON.

Se excitará la lámpara E.

Temporizador con retardo a la desactivación TOF

Este temporizador activará su salida en el mismo instante que se excita su entrada (IN). Una vez la entrada no está activa, comienza a contar el tiempo que resta para la desactivación de la salida. Además, el temporizador puede contar con una entrada RESET (R), es decir, puesta a cero de la salida y el tiempo. Normalmente, de este temporizador usaremos dos variables; el bit digital de salida (T), que podrá ser 0/1, y el dato del tiempo actual, medido entre otros, en formato de 16 bits.

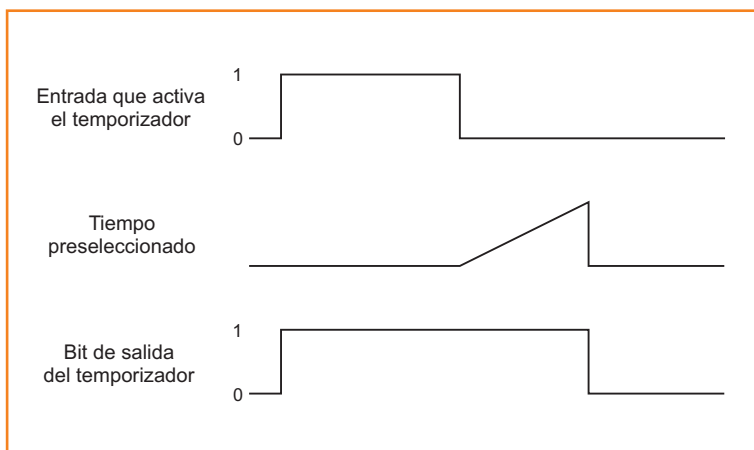
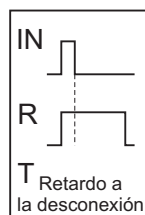


Figura 10.57. Representación y cronograma del temporizador con retardo a la desactivación.

Ejemplo temporizador TOF

Las luminarias de los garajes están gobernadas por temporizadores llamados "relés de escalera", que en realidad son temporizadores con retardo a la desactivación.

Todos los pulsadores se disponen en paralelo, hacia la entrada de activación del temporizador.

10 Autómatas programables

Ejemplo

El pulsador S1 activa el temporizador con retardo a la desactivación, y en ese mismo instante, se activa la lámpara E. ¿Qué pasará a los 10 segundos de activar la entrada del citado temporizador?

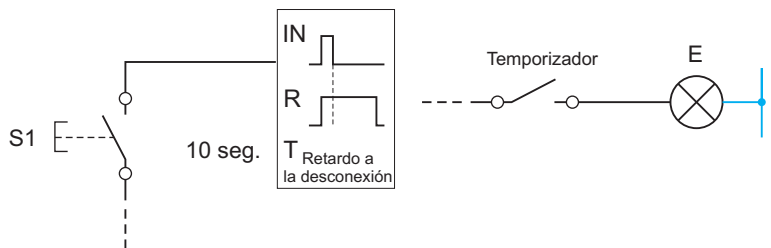


Figura 10.58. Ejemplo del temporizador TOF.

No pasará nada, pero al soltar el pulsador, comenzará a contar el tiempo -10 segundos- y pasados los cuales, se desconectará la luminaria E.

Temporizador con salida intermitente parametrizable

Permite un estado on/off a su salida, si la entrada (IN) está activa. Encontramos temporizadores de salida síncrona, es decir, el estado 1 y el 0 de la salida ocupan el mismo espacio de tiempo y temporizadores con salida asíncrona, donde el tiempo de activación de la salida, no tiene por qué coincidir con el de descanso.

Temporizador con retardo a la activación-desactivación

Es una mezcla de los dos anteriores, por este motivo, necesita programar dos bases de tiempo, (TON y TOF).

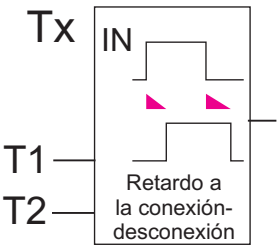


Figura 10.60. Temporizador TON-TOF.

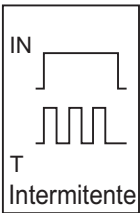
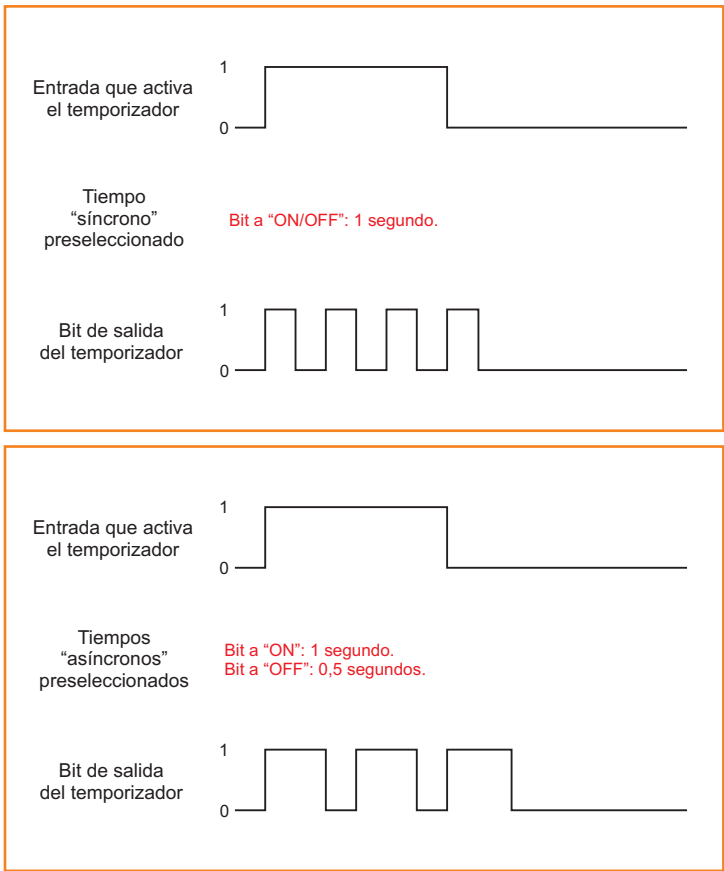


Figura 10.59. Temporizador con salida intermitente parametrizable. Cronograma síncrono y asíncrono.

Ejemplo

El pulsador S1 activa el temporizador con salida intermitente síncrona a 1 segundo. ¿Qué pasará al activar la entrada del citado temporizador?

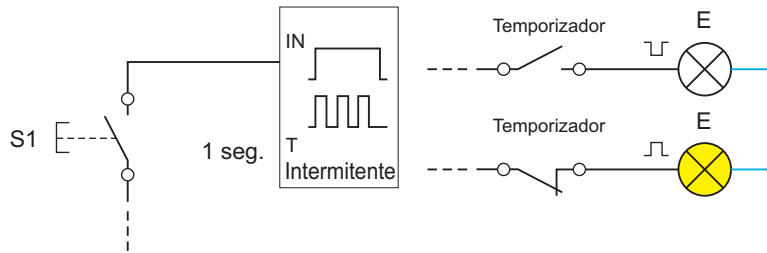


Figura 10.61. Ejemplo del temporizador con salida intermitente.

Se excitará la lámpara E a intervalos de 1 segundo, es decir, un segundo encendida, un segundo apagada, mientras el pulsador esté presionado.

Reloj horario

Activa-desactiva su salida un espacio de tiempo parametrizable, por ejemplo, de las 08:34 hasta las 23:45 horas. La programación por tanto, podrá ser semanal, mensual, incluso anual.

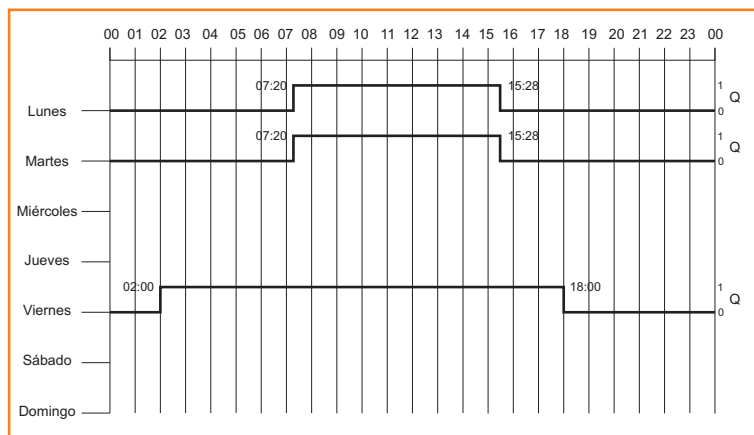
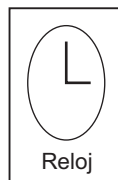


Figura 10.62. Representación y cronograma ejemplo del reloj semanal.

Ejemplo

Según el cronograma, ¿cuándo se activará la salida del reloj, y por tanto, lo que hubiera conectado a ella?

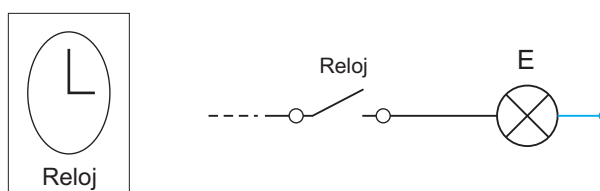


Figura 10.63. Ejemplo del temporizador tipo reloj.

Avisos de advertencia

Los temporizadores con salida intermitente son muy usados en industria en señales acústicas y luminosas de advertencia y peligro.

Si la máquina no dispone del citado temporizador, se puede constituir con dos temporizadores con retardo a la conexión.

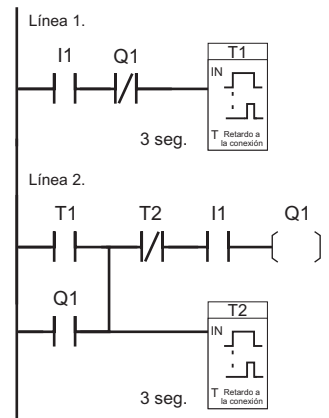


Figura 10.64. Temporizador con salida intermitente. Programación en LD.

Respuesta:

Lunes	ON: 07:20 OFF: 15:28
Martes	ON: 07:20 OFF: 15:28
Miércoles	
Jueves	
Viernes	ON: 02:00 OFF: 18:00
Sábado	
Domingo	

14.1.2. Función SET-RESET (RS)

El bloque funcional set-reset, implica a su salida un 1 , si en la entrada SET ha existido un impulso (1), es decir, el circuito se realimenta. Observe el cronograma.

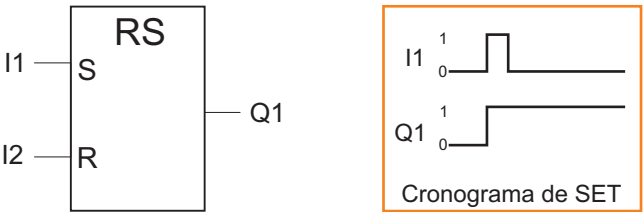


Figura 10.65. Bloque RS, y cronograma función SET.

Sin embargo, si el impulso (1) es aplicado en la entrada RESET, obliga a que su salida sea 0 . Note el cronograma.

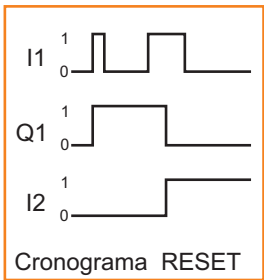


Figura 10.66. Cronograma función RESET.

Aunque la función set-reset es probablemente parametrizable, recuerde que en la mayoría de los casos, RESET tiene prioridad sobre SET.

Ejemplo

El pulsador S1 activa la entrada SET del bloque. Pasado un tiempo, son presionados a la vez los pulsadores S1 y S2. ¿Qué le ha pasado a la lámpara E?

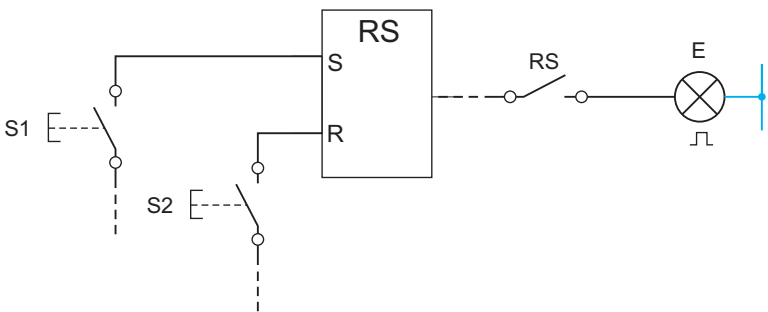


Figura 10.67. Ejemplo de la función SET-RESET.

Realimentación

La función SET-RESET, es en programación el sustituto de un circuito con realimentación. A la puesta en marcha de circuitos con esta función se le suele llamar “modo seguro”.

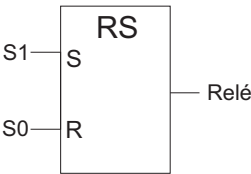
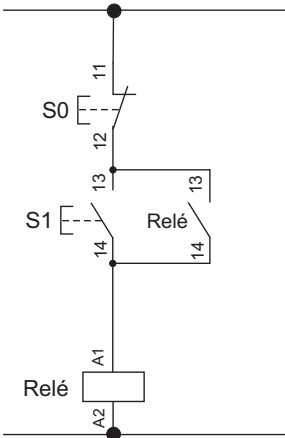


Figura 10.68. Comparación circuito con realimentación a la función SET-RESET.

Respuesta

Al presionar S1 (pulsar y soltar), la salida del bloque funcional RS tiene estado 1, ya que se realimenta, y la lámpara E se enciende. Al presionar de nuevo S1, la lámpara debe seguir conectada, pero como también lo hace S2 sobre la entrada RESET, y ésta tiene preferencia, el bloque tiene salida 0, y la lámpara se apaga. Note el cronograma del ejemplo propuesto.

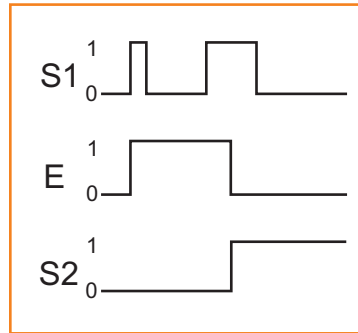


Figura 10.69. Cronograma del ejemplo.

14.1.3. Telerruptor

El bloque funcional telerruptor, tiene la misma utilidad que el telerruptor físico; con un impulso (IN) la salida será 1 y con un nuevo impulso, será 0. Además, cuenta con una entrada RESET, que obliga la salida a 0. Para los sistemas de programación que no dispongan de bloque telerruptor, la construcción de éste con funciones convencionales es simple.

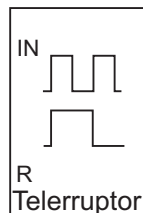


Figura 10.70. Representación de la función telerruptor.

Ejemplo

Desde varios pulsadores en paralelo (S1, S2...), dispuestos en una nave industrial, se podrá conectar-desconectar el alumbrado, que está constituido por luminarias en paralelo (E1, E2...).

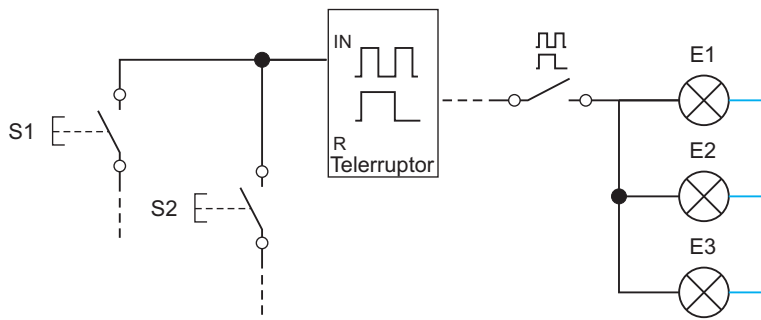


Figura 10.71. Control de alumbrado por función telerruptor.

Telerruptor

El aparato "telerruptor", se suele usar en instalaciones de interior, y viene a sustituir labores de conmutación.

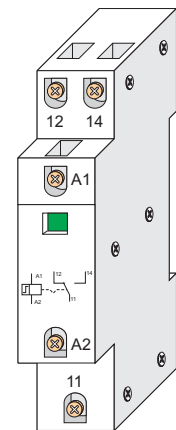
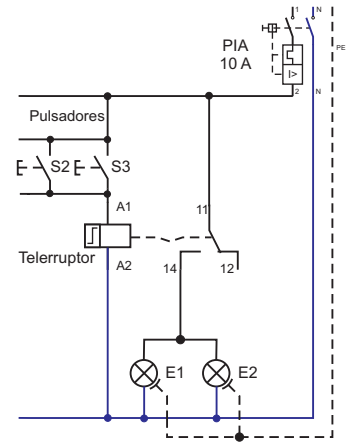


Figura 10.72. Telerruptor.

10 Autómatas programables

14.1.4. Flanco positivo

La función flanco positivo detecta un cambio de 0 a 1 en su entrada, implementando a su salida un 1 durante un ciclo de programa, es decir, un instante.

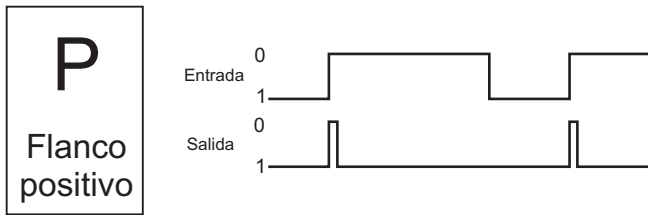


Figura 10.73. Cronograma del flanco positivo.

Ejemplo

Un pulsador efectúa una orden mientras es presionado, normalmente, durante poco tiempo. ¿Qué ocurre si el pulsador se avería y se queda presionado de forma permanente? La instalación, a través de la programación, está recibiendo todo el tiempo una orden de marcha, y esto puede ocasionar el mal funcionamiento del sistema. Con el uso del flanco positivo, aunque el pulsador esté presionado todo el tiempo, la orden sólo se efectuará el tiempo necesario para la llamada, menos de un segundo.

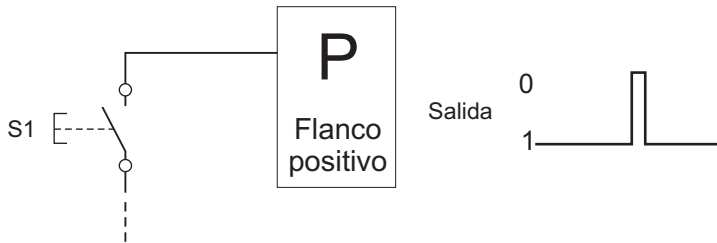


Figura 10.74. El flanco positivo permite órdenes inequívocas.

14.1.5. Flanco negativo

La función flanco negativo implementa a su salida un 1 durante un instante, si su entrada detecta un cambio de 1 a 0.

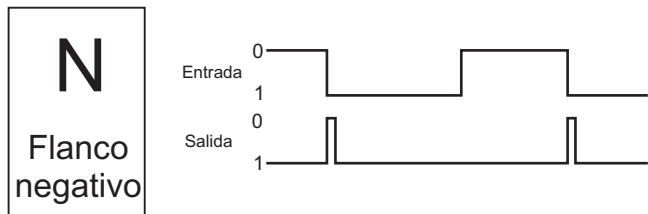


Figura 10.75. Cronograma del flanco negativo.

Ejemplo

Una aplicación requiere que cuando un motor deje de funcionar, en ese momento, se debe efectuar una orden para que se active un segundo motor. Se usará un flanco negativo aplicado al primer motor; así, cuando pare, efectuará la orden de puesta en marcha del siguiente.

Programación telerruptor

En programación, se puede “construir” un telerruptor de varias formas. Una de las más simples, utiliza flancos, positivo y negativo. A continuación, se muestra en lenguaje LD.

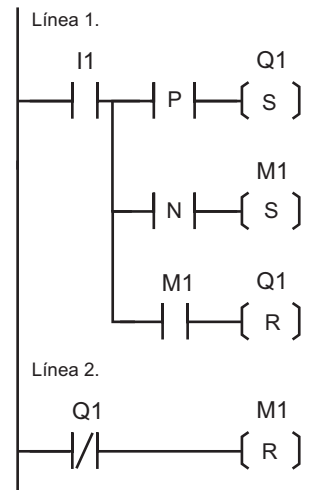


Figura 10.76. Telerruptor en Ladder.

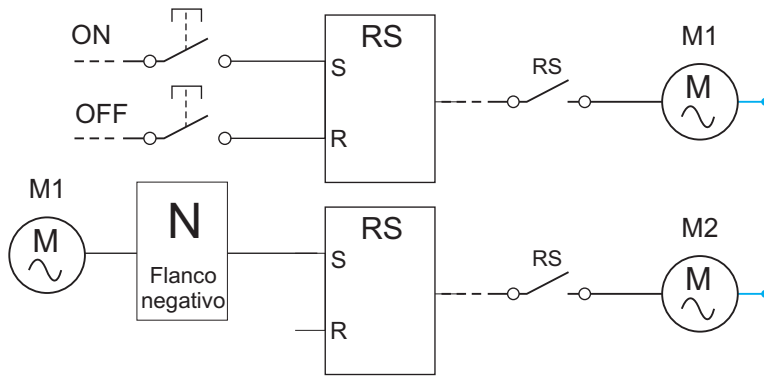


Figura 10.77. Circuito propuesto para el ejemplo.

14.1.6. Contadores

Un contador realizará una acción, cuando una entrada sea activada un número programado de veces. Técnicamente, los contadores activarán o desactivarán una salida, o un bit (bit a 1 ó bit a 0) cuando se alcanza un número predeterminado de conexiones en su entrada. Este número de activaciones, es la programación. Los contadores podrán computar en positivo, en negativo o en ambos sentidos. Además, también pueden contar con una entrada de RESET, que pone el cómputo y la salida a cero.

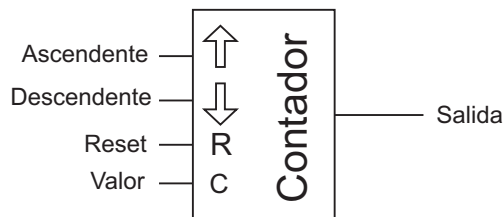


Figura 10.78. Contador que computa de manera ascendente, descendente y con entrada para reset.

Por ejemplo, cuando la entrada del contador se conecte 8 veces, se activará la salida Q.

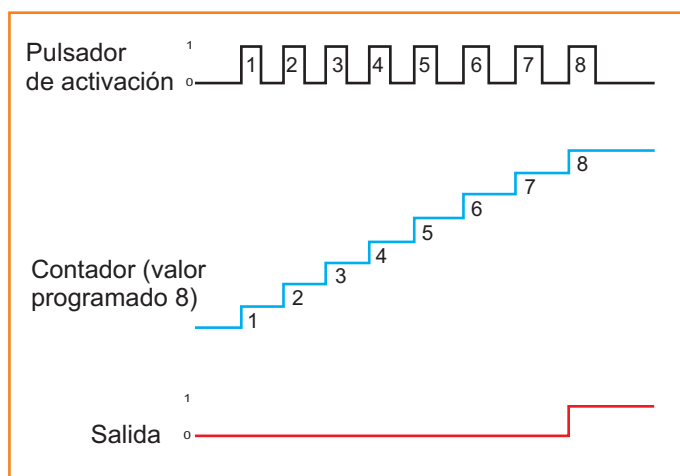


Figura 10.79. Cronograma de cómputo a "8".

- Primero, el contador está programado a 8; esto no quiere decir que la entrada no pueda ser activada más, si no que al llegar a 8, la salida tendrá valor 1.
- Segundo, el contador puede disponer de descontador, lo cual hace que, si el número total del cómputo ascendente no es 8 la salida se desactivará.

Contadores rápidos

Los autómatas programables que disponen de contadores rápidos, que serán capaces de computar velocidades superiores a las del tiempo de ciclo. Además, existen módulos periféricos específicos para estos contadores rápidos.

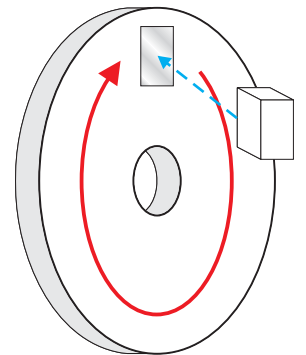


Figura 10.80. Un contador rápido podrá computar las vueltas de un motor.

10 Autómatas programables

En el siguiente gráfico se muestra como el cómputo llega a 8, la salida se activa, pero el descontador baja el cómputo a un valor inferior a 8, con lo cual, la salida se desconecta.

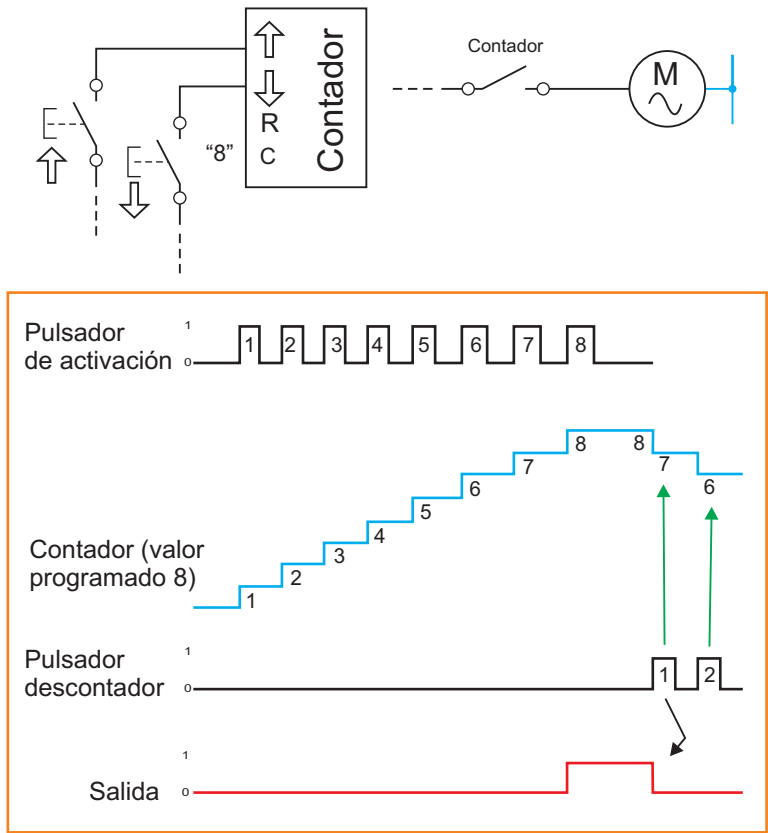


Figura 10.81. Contador en positivo y negativo.

Normalmente, de un contador usaremos dos variables; el bit digital de salida (C), que podrá ser 0/1, y el dato del cómputo, medido en formato de -por ejemplo- 16 bits.

14.1.7. Comparadores

Esta función compara dos valores de entrada y permite el paso de corriente a su salida si se cumple la relación entre ambos valores, que podrá ser:

- Igual que.
- Mayor que.
- Menor que.
- Mayor o igual que.
- Menor o igual que.

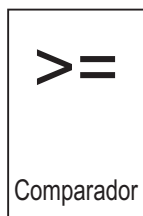


Figura 10.82. Comparador.

Ejemplo

La entrada analógica de un autómata programable está conectada a un anemómetro, que controla la velocidad del viento local. Si la velocidad registrada por el PLC es mayor o igual 100 km/h, se cierran automáticamente los toldos, por peligro de rotura.

Conversión de datos

Las funciones de conversión, permiten modificar un tipo de datos, en otro diferente, por ejemplo, de formato Byte (8 bits) a Word (16 bits).

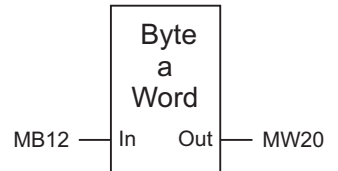


Figura 10.83. Función de conversión de datos.

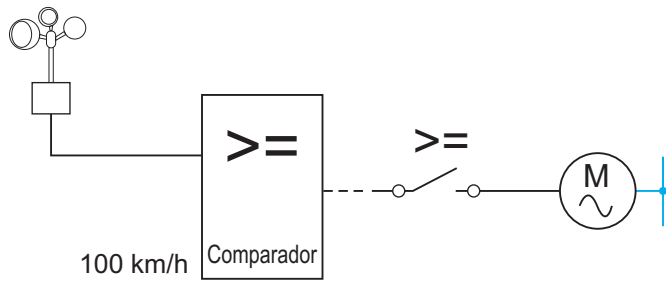


Figura 10.84. La velocidad real, se comparará con un valor establecido.

Suma de tiempos

Los operadores aritméticos pueden sumar “tiempos” de diferentes temporizadores, ya que en el fondo, el tiempo computado son datos.

Por ejemplo, una instalación de semáforos está constantemente barajando cifras de tiempo, y la relación entre ellas, da como resultado el ajuste que observamos en la vía.

14.1.8. Operadores aritméticos

Con los datos que maneja el autómata, procedentes de temporizadores, contadores, entradas analógicas, salidas analógicas, entre otros, se podrá operar aritméticamente. Los principales operadores son: suma, resta, multiplicación, división, raíz cuadrada. Adicionalmente, algunas máquinas operan con trigonometría, logaritmos, incluso en operaciones de fifo y lifo.



Figura 10.85. Operadores aritméticos.

Ejemplo

Dos cintas transportadoras, desplazan piezas de fruta a una caja común. Cuando la suma de las piezas es 100, cantidad estimada para llenar la caja, las cintas se detienen, hasta que el proceso se reinicie de nuevo.

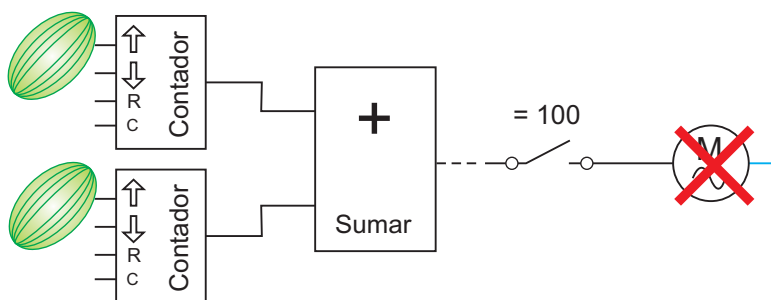


Figura 10.86. Cuando la suma de las piezas de fruta, computadas por los dos contadores, sea igual a “100”, se detiene el motor de las cintas.

14.2. Programación en Diagrama de Contactos (LD, Ladder Diagram)

El lenguaje gráfico LD, es sin duda de los más utilizados para programar autómatas programables, ya que la simbología utilizada, es parecida a los esquemas eléctricos empleados en instalaciones eléctricas con relés o contactores. La utilización de entornos gráficos, hace que la programación se pueda enriquecer con textos complementarios explicativos.

10 Autómatas programables

Note un ejemplo de programación en LD, con comentarios:

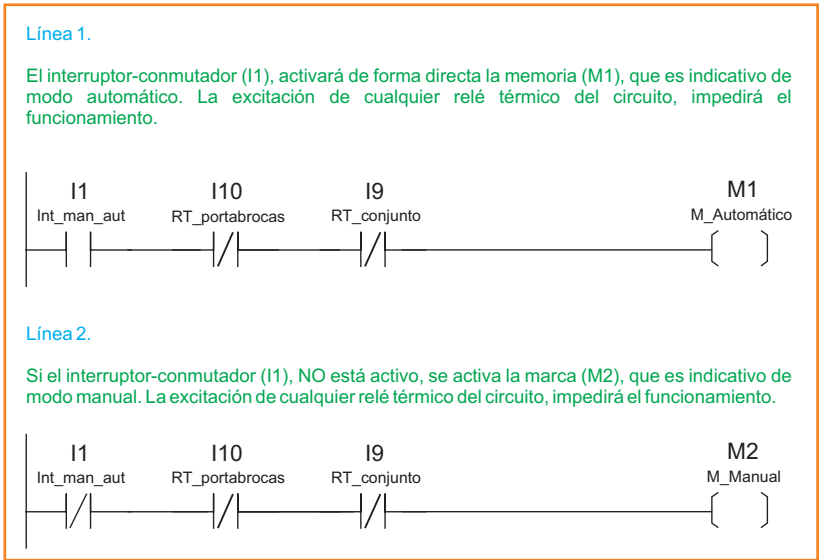


Figura 10.87. Ejemplo de programación en lenguaje de contactos.

La semejanza de este sistema de programación a los circuitos eléctricos basados en relés y contactores, aporta en el diseño cierta facilidad, ya que partimos de la base, de que a la bobina, tiene que llegarle corriente eléctrica. Nos referimos a circuitos en escalera.

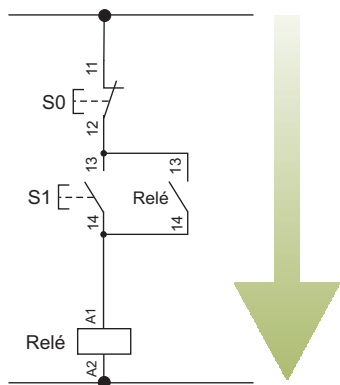


Figura 10.88. En los esquemas eléctricos en escalera, la corriente tiene que llegar a la bobina pasando por los dispositivos.

La conversión por tanto entre un esquema de relés y lenguaje LD, es sencilla, aunque hay que buscar la equivalencia entre los componentes.

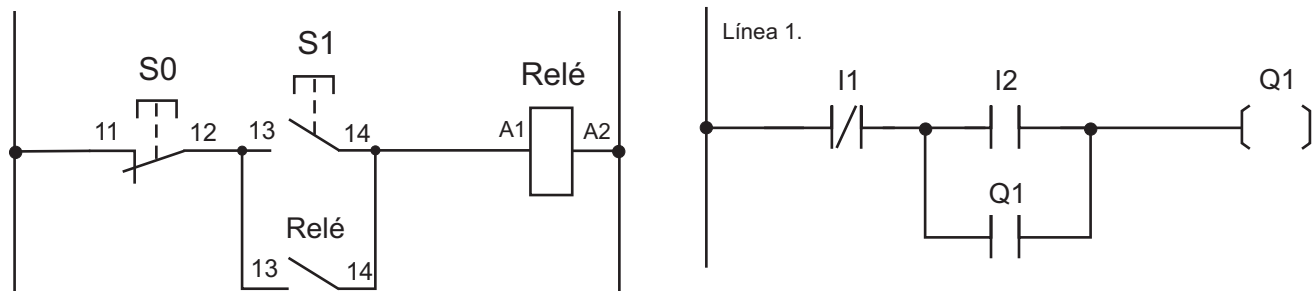


Figura 10.89. Equivalencia entre sistemas.

Símbolos LD

En el lenguaje de programación LD, los símbolos (que son elementos de programación aunque sean parecidos a esquemas eléctricos) más usados son:

- Contacto abierto
- Contacto cerrado
- Negación
- Bobina directa
- Bobina inversa
- Activación de bobina en SET
- Desactivación de bobina en RESET

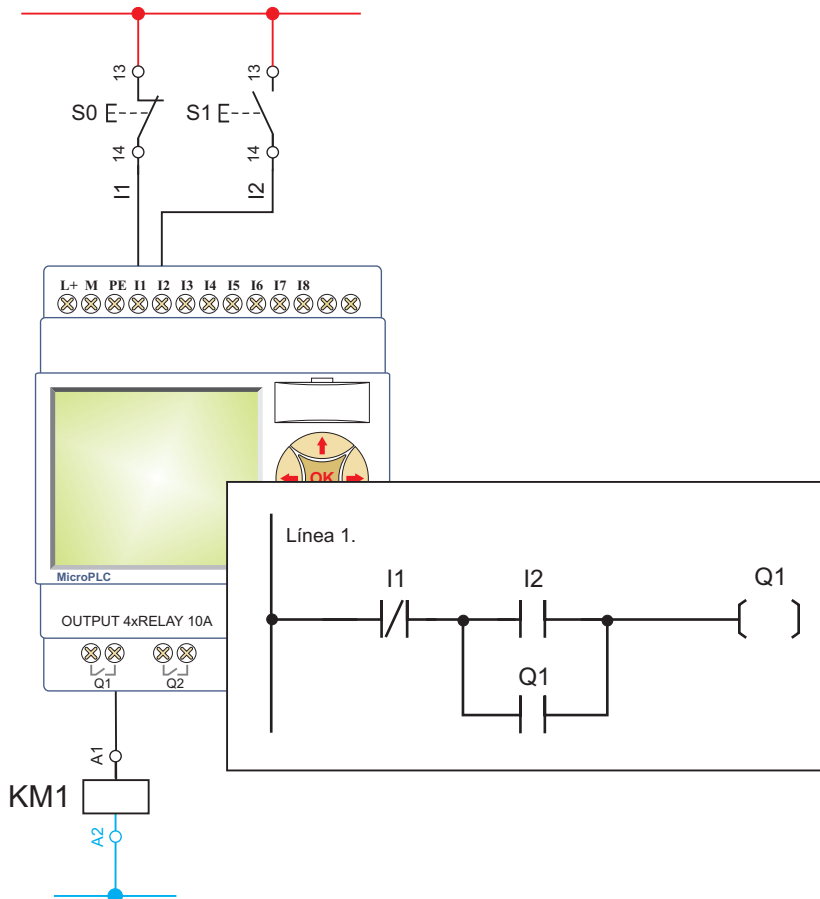


Figura 10.90. Cableado equivalente a la programación dada.

Según este esquema eléctrico de cableado del PLC, deben quedar claros unos conceptos:

- El pulsador S0 **NO ES** el contacto I1, ya que S0 es físico e I1 es la representación virtual de S0 en el programa. De otro modo; un contacto físico exterior de un pulsador, final de carrera, detector, etc. No tiene porqué representarse con la misma condición de abierto o cerrado en el programa. Por tanto, I1, es el terminal o conector que une el pulsador de paro S0 al autómatas, y por donde el pulsador le aporta información de su estado -abierto o cerrado- al PLC.
- Del mismo modo, I2 **NO ES** S1, sino una representación del estado que tenga actualmente el pulsador S1.

Ejemplo

En reposo, tenemos este cableado y esta programación. ¿Qué ocurre si presionamos a la vez los pulsadores S1 y S2?

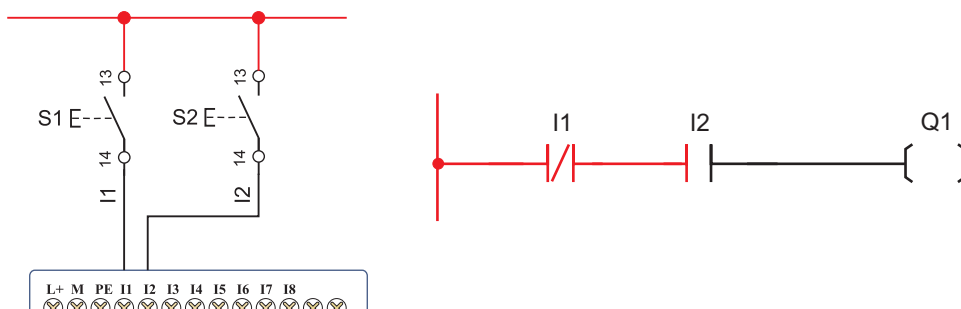


Figura 10.91. Circuito propuesto para el ejemplo.

Temporizadores en LD

Se representa el símbolo del temporizador, y su contacto o contactos, que podrán ser abiertos o cerrados.

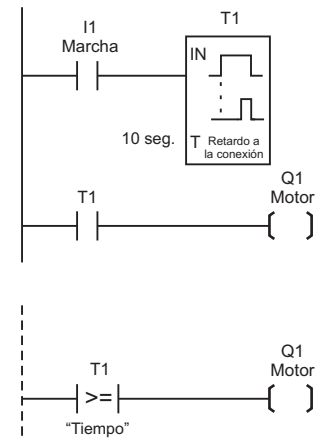


Figura 10.92. Temporizador en LD.

10 Autómatas programables

Respuesta

No se activará Q1, ya que al presionar S1, el contacto I1 se ABRE, aunque al presionar S2, el contacto I2 se CIERRA.

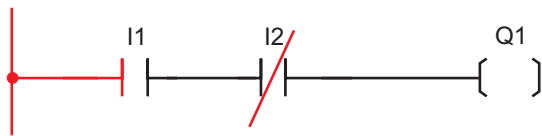


Figura 10.93. Efecto de la programación.

Ejemplo

Según este cableado y esta programación. ¿Cómo estará actualmente -en reposo- la salida Q1?

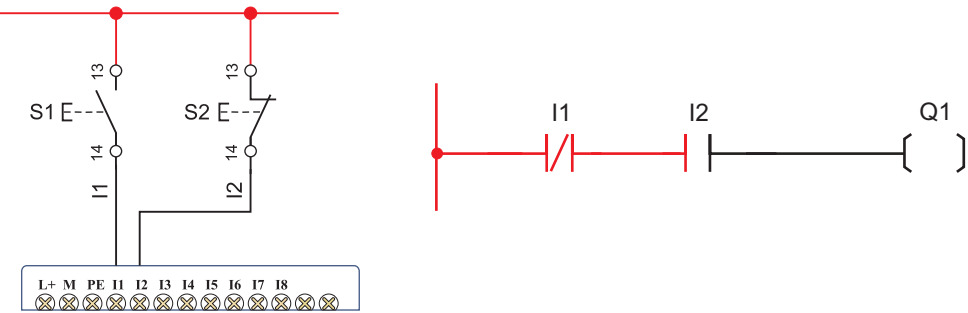


Figura 10.94. Circuito propuesto para el ejemplo.

Respuesta

La salida Q1 estará inicialmente activa, ya que S2 está implementando corriente al borne I2, por tanto, I2 cambia su estado en la programación.

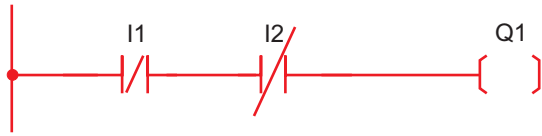
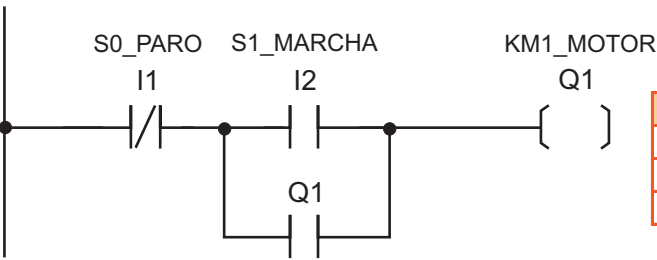


Figura 10.95. Resultado de la programación, según el circuito eléctrico.

Identificación de variables

No sólo para este lenguaje de programación. Para una programación extensa, se hace necesario una identificación de variables de manera ordenada, que facilite sobre todo la puesta en marcha y depuración de errores. El programa, sabrá identificar la variable por el nombre, además de por el indicador propio.



SÍMBOLO	VARIABLE	COMENTARIO
S0_PARO	I1	Pulsador de parada
S1_MARCHA	I2	Pulsador de marcha
KM1_MOTOR	Q1	Motor trifásico en caldera

Figura 10.96. El programa permitirá crear listado de variables con nombres simbólicos.

Contador en LD

Se representa el símbolo del contador, y su contacto o contactos, que podrán ser abiertos o cerrados. En el ejemplo, el motor se activará cuando el contador compute de forma positiva "10".

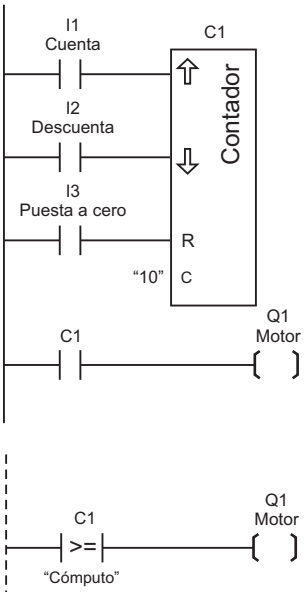


Figura 10.97. Contador en LD.

Los contactos en lenguaje LD, se podrán asociar en serie, paralelo, y mixto.

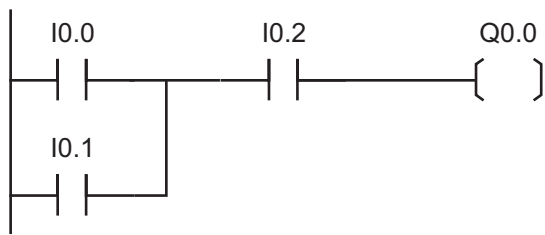


Figura 10.98. Circuito mixto de contactos en LD.

Ejercicio

Un pulsador S1, pone en marcha un sistema de riego, que consiste en la activación de un motor-bomba trifásico, por un tiempo total de 5 horas, pasadas las cuales, se conectará un segundo motor-bomba monofásico, al tiempo que se detiene el primer motor. El segundo motor funcionará un tiempo máximo de 4 horas. Existirá un pulsador de parada S2, además de protección por relé térmico de cada motor.

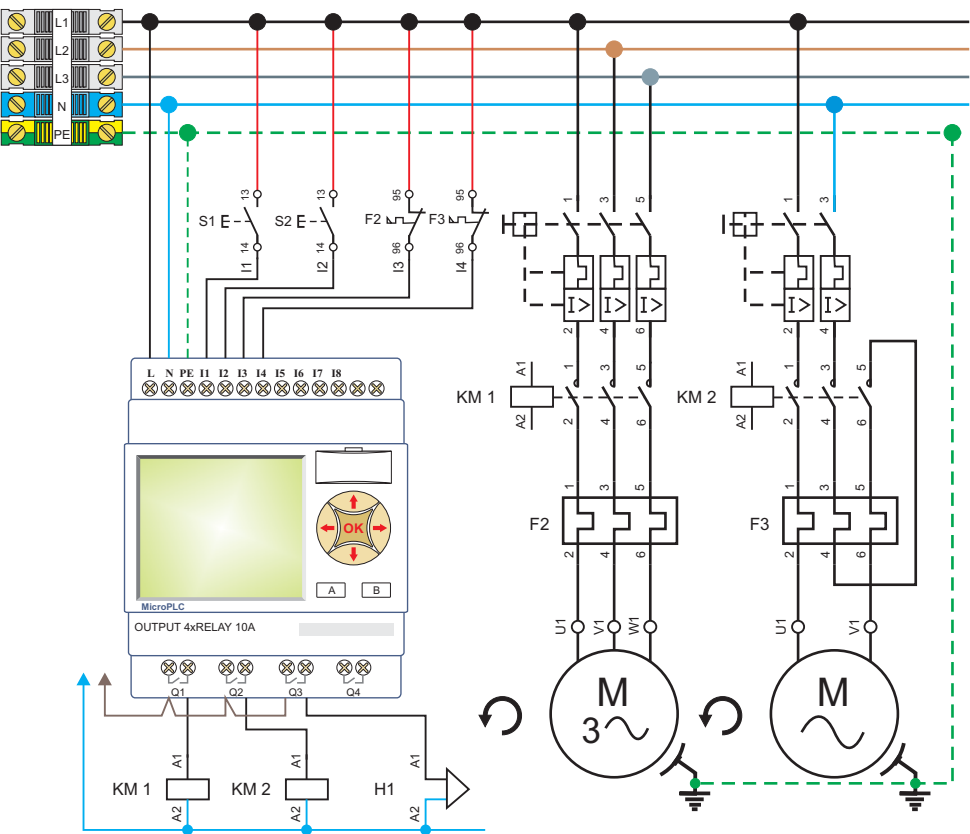


Figura 10.99. Cableado del ejercicio propuesto.

Ejercicio

¿En qué estado se encuentra la lámpara E?

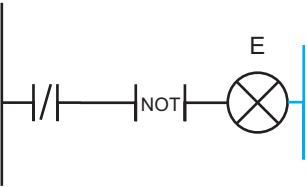


Figura 10.100. Programación.

SÍMBOLO	VARIABLE	COMENTARIO
S1_MARCHA	I1	Pulsador de marcha
S2_PARO	I2	Pulsador de parada
F2_RT_MOT_TRIFÁS.	I3	Relé térmico motor trifásico
F3_RT_MOT_MONO.	I4	Relé térmico motor monofásico
MOTOR1	Q1	Motor trifásico
MOTOR2	Q2	Motor monofásico
AVERÍA	Q3	Aviso por avería

Tabla 10.8. Lista de símbolos.

10 Autómatas programables

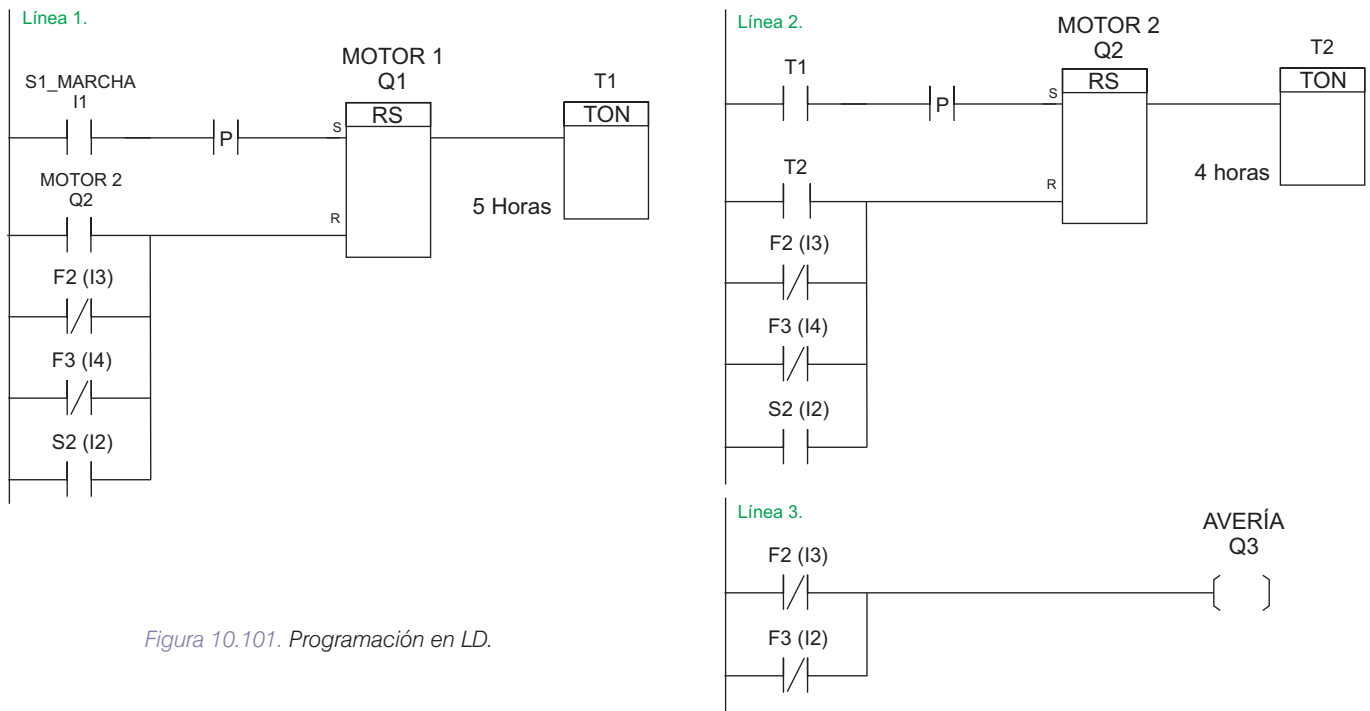


Figura 10.101. Programación en LD.

14.3. Programación en Lenguaje de Funciones Lógicas (FBD, Function Block Diagram)

El lenguaje gráfico FBD es parecido a los esquemas empleados en electrónica digital, utilizando funciones lógicas. También se le pueden insertar textos explicativos. Recuerde las puertas lógicas usadas en el apartado del álgebra de Boole. En la siguiente figura, se aprecia la comparación de una programación entre LD y FBD.

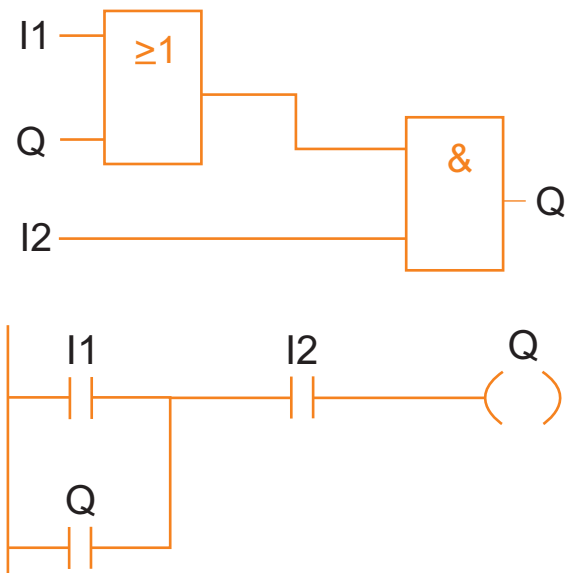


Figura 10.102. Comparación entre FBD y LD.

De la cual, conocemos las entradas (I1 e I2) y la salida (Q). Recuerde el significado de las puertas lógicas (&, puerta AND; ≥ 1 , puerta OR; $= 1$, puerta XOR, etc.). De la programación en FBD, podemos convertir a LD, de manera simple. Si el autómata programable admite estos dos (o más) sistemas de programación, el resultado final funcional es el mismo.

En la siguiente figura, aparece la equivalencia de la misma programación, con un esquema eléctrico puro.

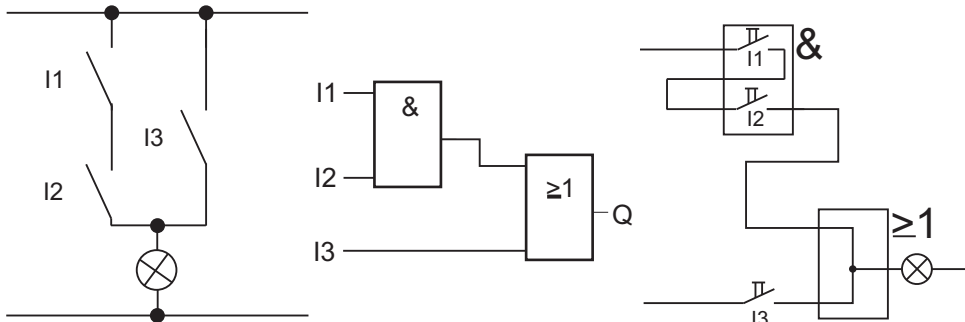


Figura 10.103. Comparación entre contactos eléctricos y FBD.

Ejemplo

La entrada I0.0, activará la salida Q0.0, que a su vez se realimentará a través de un contacto. Para desactivar el circuito se usará un contacto cerrado I0.1.

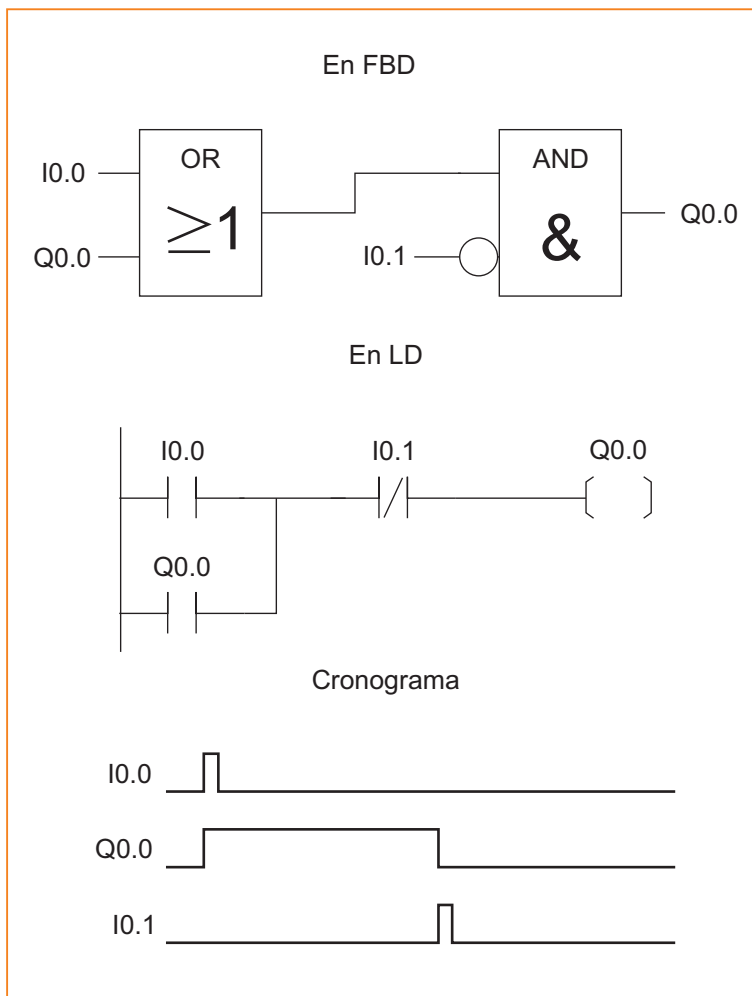


Figura 10.104. Ejemplo FBD, comparación y cronograma.

Función NOT

Para simplificar las programaciones, la función inversora NOT, se puede contraer con un círculo como símbolo, antes del siguiente bloque funcional.

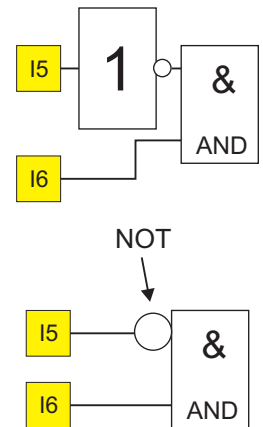


Figura 10.105. Función NOT. Bloque normativo y reducido.

Temporizadores en FBD

Normalmente permite la conexión directa del temporizador a otros componentes, al no usar contactos abiertos o cerrados.

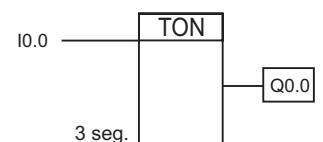


Figura 10.106. Temporizador en FBD.

10 Autómatas programables

Ejemplo

Una vez se active la entrada I0.0, la salida Q0.0, hará lo mismo pasados tres segundos. Si la entrada I0.0 = 0, la salida también adoptará ese estado.

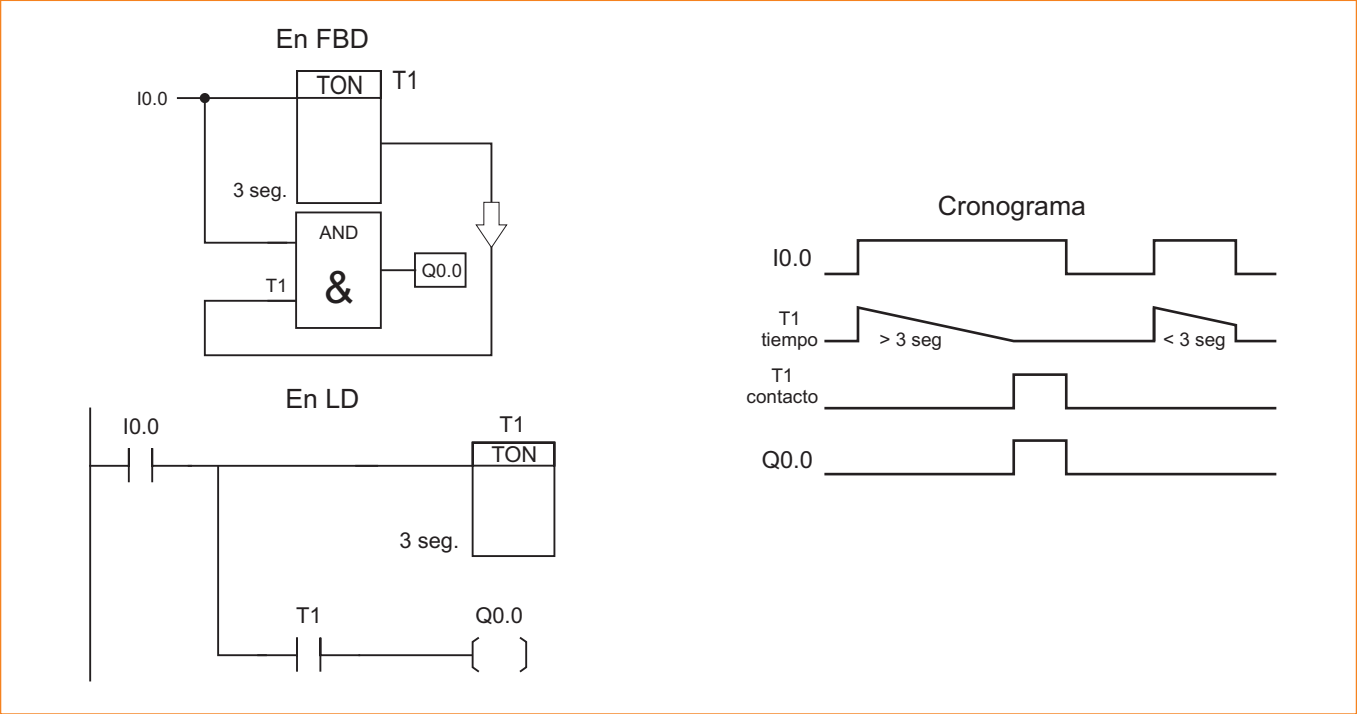


Figura 10.107. Ejemplo FBD, comparación y cronograma.

Ejemplo

La salida Q0.0 se activará en SET, si se excitan a la vez las entradas I0.0 e I0.1. Para anular a Q0.0 en RESET, se deben excitar a la vez las entradas I0.0 e I0.2.

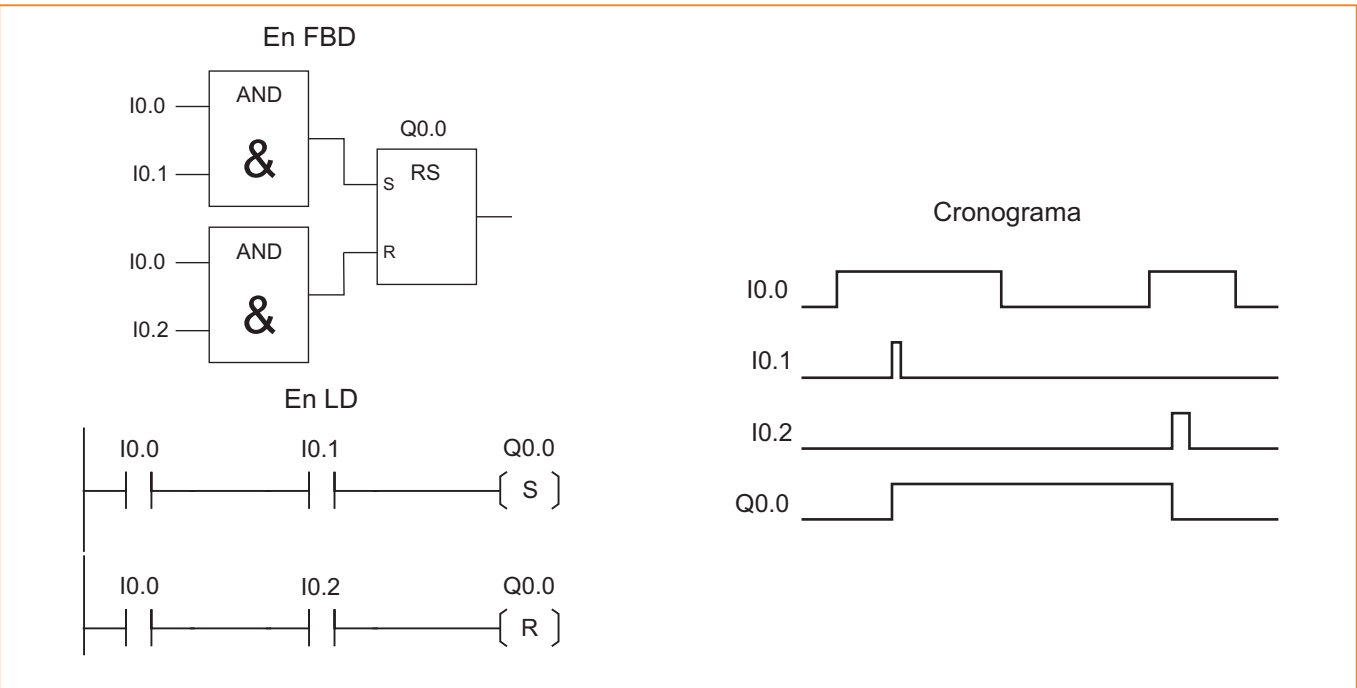


Figura 10.108. Ejemplo FBD, comparación y cronograma.

Ejercicio con lenguaje FBD

Trituradora de grava

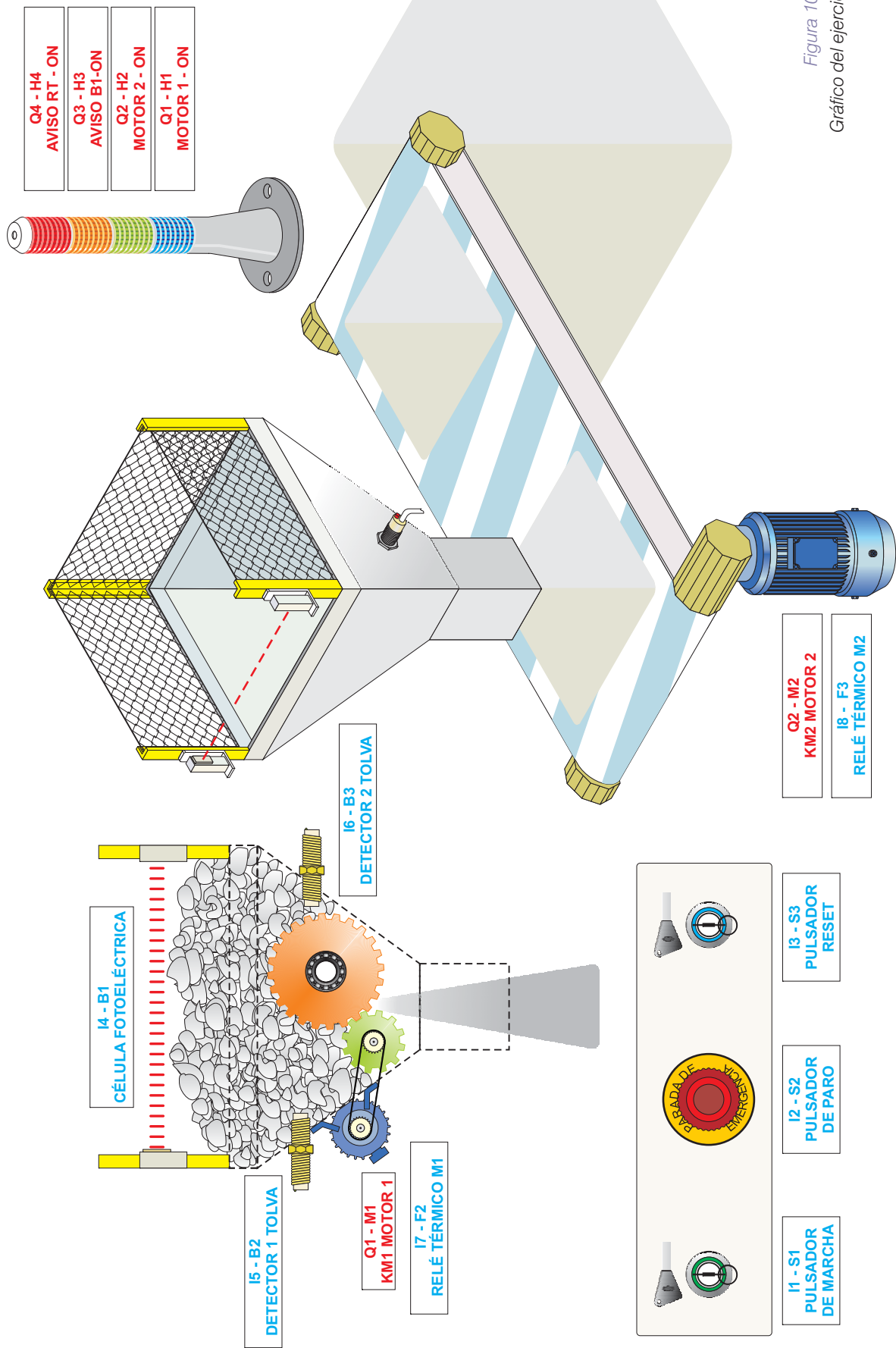


Figura 10.109.

Gráfico del ejercicio propuesto.

10 Autómatas programables

Argumento

Una tolva, admite grava de diferente calibre para ser sometida a un proceso de trituración, donde se tienen que cumplir las siguientes condiciones:

- 1. La grava sólo entra a la tolva por un lateral, estando los otros tres, protegidos por una valla.
- 2. El proceso de moledura, se pone en marcha con el pulsador de llave S1 (I1), que activa en primer lugar el motor 1 de la tolva (Q1), y a los 5 segundos el motor de la cinta transportadora 2 (Q2). Para que comience el proceso, la tolva debe estar llena de grava.
- 3. Dos detectores B1 y B2 (I3 e I4), advierten el llenado de la tolva; si ello ocurre, se puede iniciar el proceso, activando los motores de la tolva y cinta respectivamente. El motor de la cinta permite que la grava molida se transporte a un recipiente mayor.

- 4. Si uno de los dos detectores, o los dos, no advierten grava síntoma de que la tolva se está vaciando, le resta un tiempo de 10 segundos de funcionamiento al motor 1 (Q1).
- 5. La zona de entrada de grava cuenta con una célula fotoeléctrica B1 (I4), que estará activa sólo en el proceso de moledura (cuando Q1 = activa) para que, en caso de intrusión de un objeto, el sistema se detenga automáticamente. Para reactivar el proceso, se debe presionar el pulsador de llave S3 (I3), que es un reset, y después presionar de nuevo el pulsador de marcha S1 (I1).

- 6. Una vez se detiene en motor 1, el motor 2 hará lo propio 20 segundos después, tiempo suficiente para vaciar la cinta.
- 7. Cada motor tiene protección por relé térmico F2 y F3 (I7 e I8). En caso de actuación de alguno de ellos, además de detener el proceso, se activará un aviso luminoso H4 (Q4).
- 8. El pulsador de paro S2 (I2), detiene todo. Es una seta.

SÍMBOLO	VARIABLE	COMENTARIO
S1	I1	Pulsador de marcha
S2	I2	Pulsador de paro. Seta
S3	I3	Pulsador de reset
B1	I4	Célula fotoeléctrica
B2	I5	Primer detector tolva
B3	I6	Segundo detector tolva
F2	I7	Relé térmico motor tolva
F3	I8	Relé térmico motor cinta
MOTOR1	Q1	Motor tolva
MOTOR2	Q2	Motor cinta transportadora
H3	Q3	Aviso, célula fotoeléctrica activa
H4	Q4	Aviso, algún relé térmico activo

Tabla 10.9. Listado de variables del ejercicio propuesto.

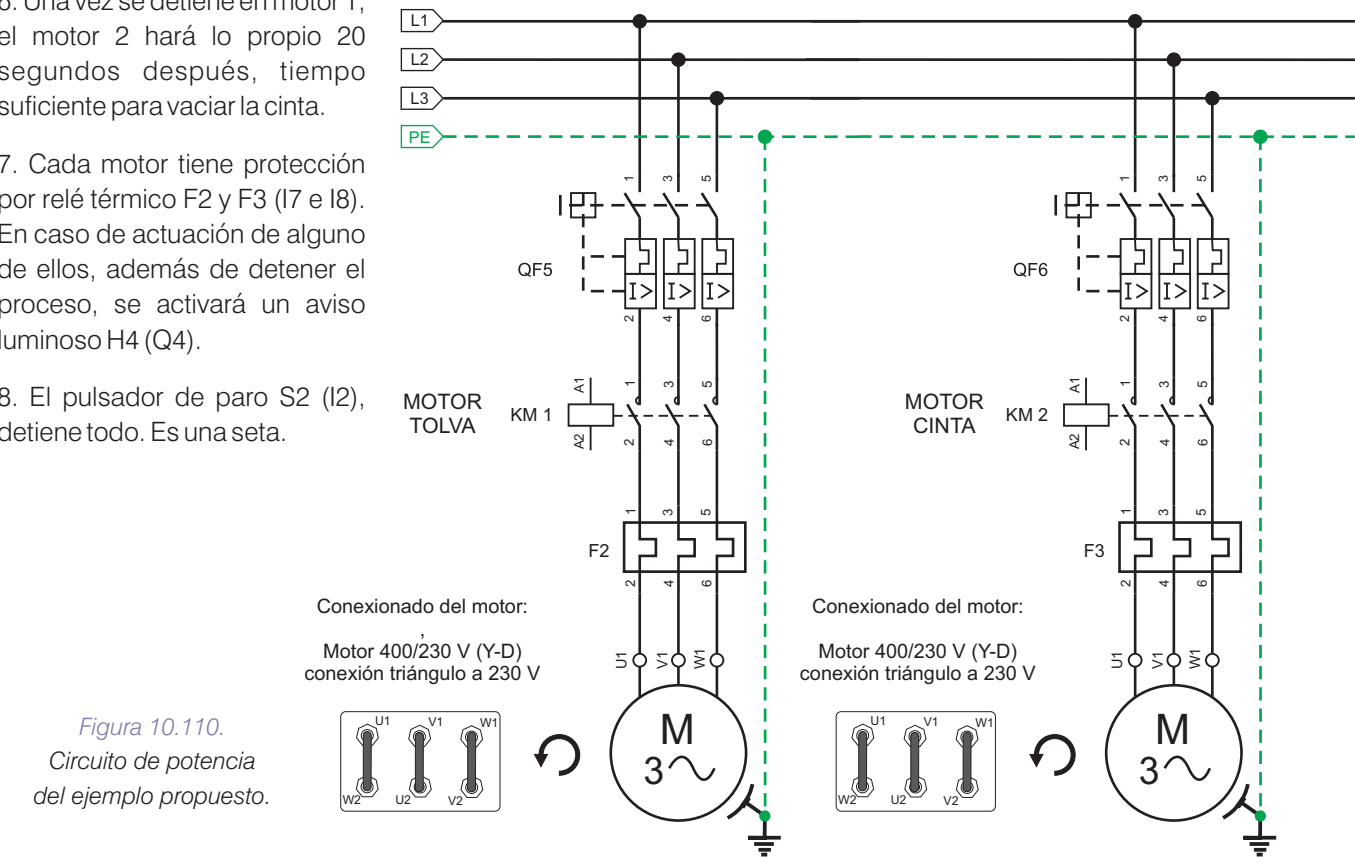


Figura 10.110. Circuito de potencia del ejemplo propuesto.



Programación en lenguaje de funciones lógicas FBD

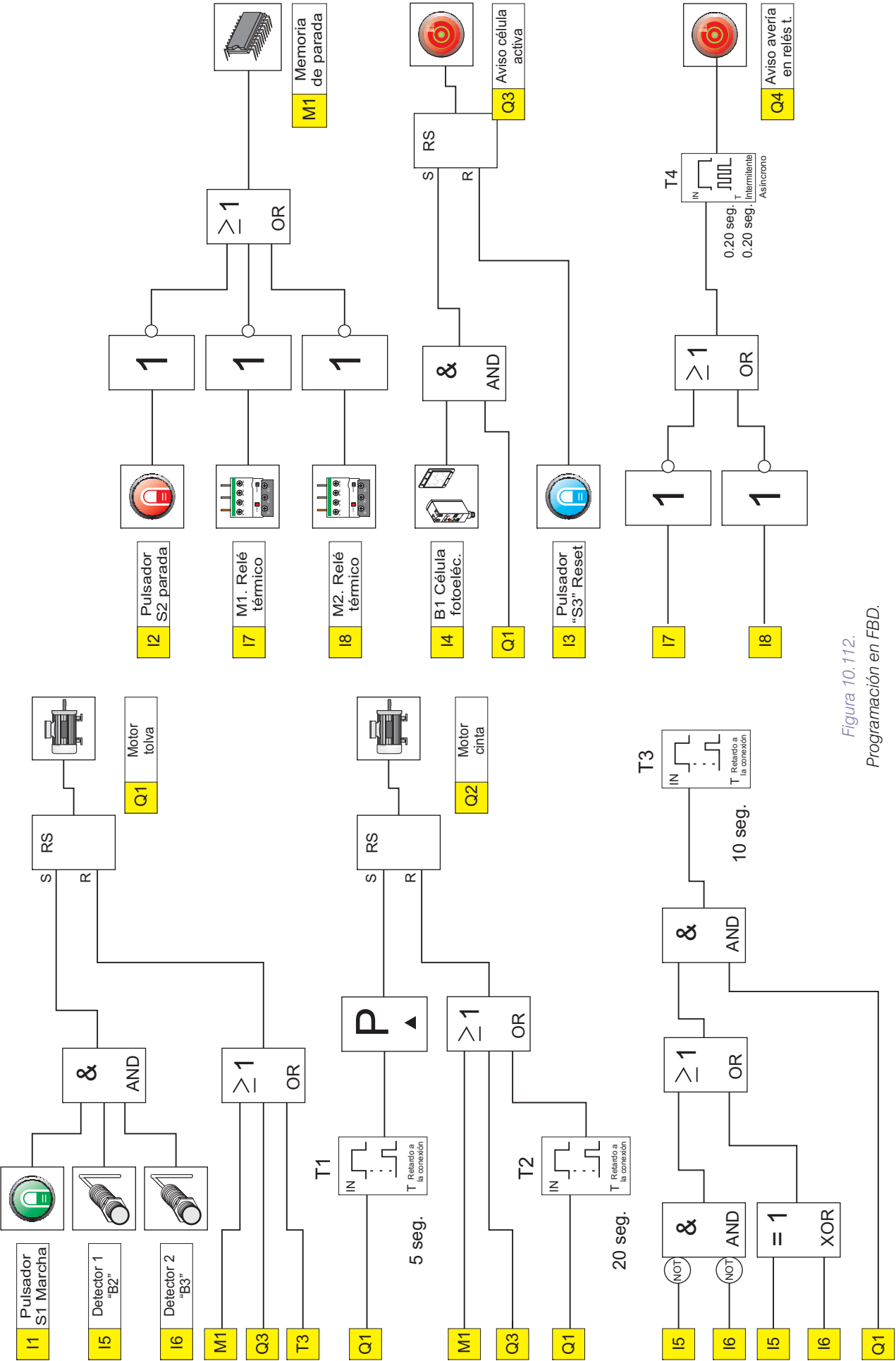


Figura 10.112.
Programación en FBD.

14.4. Programación en lista de instrucciones (IL, Instruction List)

Es un lenguaje textual -no gráfico- tipo ensamblador de bajo nivel usado en computación, compuesto por una serie reducida de instrucciones. Por este motivo, los diferentes fabricantes de autómatas, desarrollan un lenguaje basado en lista de instrucciones, con una riqueza de símbolos superior, que le permiten corresponderse con el resto de lenguajes. Note en las siguientes tablas, las instrucciones básicas del lenguaje.

Programa	
LD	Carga valor
ST	Almacenar resultado

Operaciones lógicas con bits	
AND	Función AND
OR	Función OR
XOR	Función XOR
S	Puesta a 1
R	Puesta a 0

Control del programa	
JMP	Salto
CAL	Llamada
RET	Retorno

Operaciones aritméticas	
ADD	Suma
SUB	Resta
MUL	Multiplicación
DIV	División

Comparadores	
GT	Mayor que
GE	Mayor o igual a
EQ	Igual a
NE	Mayor o menor a
LE	Menor o igual a
LT	Menor que

Temporizadores	
TON	Temporizador con retardo a la activación
TOF	Temporizador con retardo a la desactivación

Tablas 10.10. Operadores y nemónicos básicos IEC 1131.

Ejemplo

Dos entradas digitales conectadas en serie, y como resultado, la activación de una salida digital.

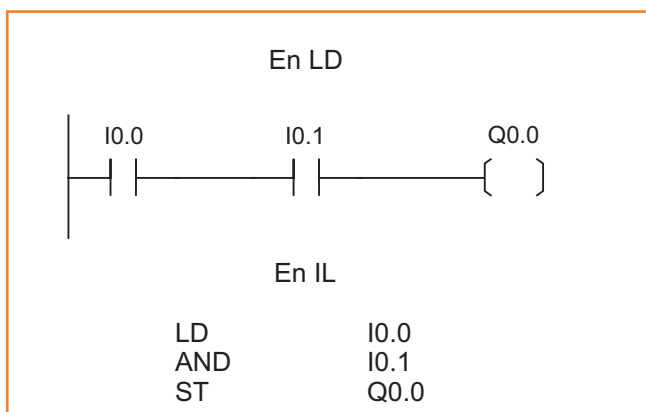


Figura 10.113. Ejemplo IL de dos entradas en serie.

Ejemplo

Dos entradas digitales conectadas en serie, una de ellas negada, y como resultado, la activación de una salida digital.

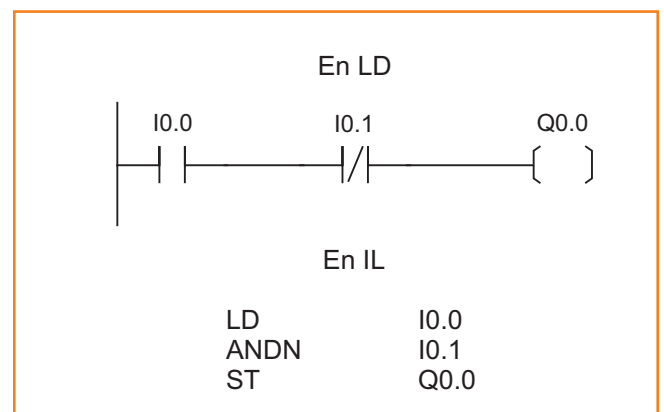


Figura 10.114. Ejemplo IL de dos entradas en serie, una negada.

Ensamblador

El lenguaje de bajo nivel ensamblador, trabaja directamente con los microprocesadores. Se basa en instrucciones que se implementan al microcontrolador directamente, siendo por tanto el lenguaje -del programador- más cercano a la máquina.

Ocupa pocos recursos, ya que no necesita librerías como ocurre con los lenguajes de alto nivel. Emplea nemónicos, que son palabras que sustituyen a un código determinado de operación, -en lenguaje máquina- con lo cual se reduce el espacio.

Por ejemplo, la instrucción MOVE, que desplaza datos de un lugar de memoria a otro.

10 Autómatas programables

El sistema que desarrolle programas en lista de instrucciones, permitirá realizar operaciones típicas como dividir las programaciones en diferentes líneas, crear listas de variables, comentar las líneas de programación, entre otros.

Línea 3.

Primera línea de programación

Modo automático: Si el detector (I5), advierte una pieza, se activa la marca auxiliar (M3), para orden de bajada, a través de la salida (Q1). Esta marca podrá ser anulada por:

- La activación del relé térmico del conjunto arriba-abajo.
- Final de carrera inferior.
- Un contacto de seguridad de motor sube (Q2).
- La memoria (M1), automático debe estar activa.

LD

Detec_piezas

LD

RT_conjunto

O

FC_inferior

O

Motor_sube

ON

M_automático

NOT

A

M_de_Q01

=

M_de_Q01

Comentarios a la primera línea de programación

Programación

Figura 10.115. Programa en lista de instrucciones con comentarios.

14.5. Texto estructurado (ST, Structured Text)

Es un lenguaje de alto nivel, utilizado para programaciones complejas. Los lenguajes de alto nivel se desarrollan para que la forma de programación sea más adecuada al conocimiento del hombre, que a un lenguaje cercano al de la máquina. Internamente, la máquina debe realizar complejos cálculos, que requieren de capacidades de memoria considerables. Como ejemplo de algunos lenguajes de alto nivel: C++, BASIC, COBOL, FORTRAN, JAVA, PASCAL, PHP, SQL.

En la siguiente tabla aparecen algunos operadores y sentencias típicas de un lenguaje de alto nivel.

Operadores y sentencias	
*	Multiplicación
/	División
+	Suma
-	Resta
=	Igualdad
>, <, >=, <=	Comparaciones
< >	Desigualdad
&	Operación AND
OR	Operación OR
XOR	Operación XOR
IF, THEN, ELSE	
CASE	
FOR	
WHILE	
REPEAT UNTIL	

Tabla 10.11. Algunos operadores en lenguaje de alto nivel.

Lenguaje de alto nivel en dispositivos móviles

Algunos juegos que disfrutamos en dispositivos como teléfonos móviles, smartphones, ordenadores personales, tabletas, etc. Están desarrollados por un lenguaje de alto nivel como JAVA.

Figura 10.116. Lenguaje de alto nivel para la creación de juegos.

En LD

En ST

Q0.0:= I0.0 AND I0.1

En LD

En ST

Q0.0:= I0.0 AND NOT I0.1

Figura 10.117. Ejemplos de programación en lenguaje ST.

280

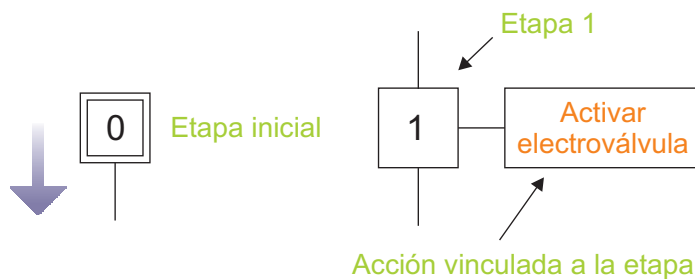
14.6. Gráfico secuencial de funciones (GRAFCET)

El GRAFCET es un lenguaje gráfico que representa las secuencias del programa en forma de diagrama. Para conocer este lenguaje, es necesario comprender los circuitos secuenciales.

14.6.1. Etapas

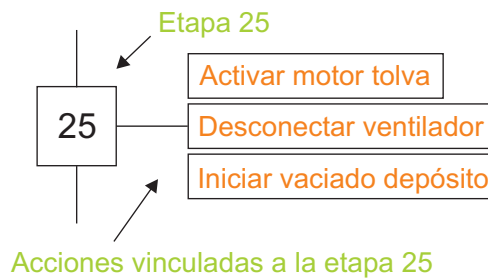
Una etapa implica una acción, dentro del proceso secuencial. Abrir una válvula, activar un motor, cerrar una compuerta, etc. En GRAFCET, se sucederán diferentes etapas, y la suma de todas ellas será el proceso productivo. Cuando una etapa está activa, la anterior lo ha sido previamente.

La etapa se representa con un cuadrado, y dentro del mismo, se indica el número de etapa. La primera etapa o etapa inicial, se representa con un cuadrado doble, y tiene que ser activada en primer lugar, antes del resto de etapas.



Tablas 10.118. Etapa inicial y etapa 1, asociada a una acción.

Una etapa puede asociarse a una o más acciones, que estarán operativas cuando la etapa está activa.



Tablas 10.119. Una etapa puede contener varias acciones.

14.6.2. Transiciones

En el proceso secuencial, para que se ejecute una acción, ha tenido que efectuarse alguna condición que permita - o no - que se desarrolle la etapa. Estas condiciones se denominan transiciones, y son las que permiten que las etapas se desarrollen una tras otra. Cuando se produce una transición, se activa la etapa siguiente y se desactiva la anterior.

En el siguiente gráfico, la condición es un final de carrera, que permite la activación de la etapa 15, que a su vez tiene asociadas dos acciones. No obstante, la transición pertenece a la etapa anterior, en este caso, a la 14.

Grafcet y SFC

GRAFCET

GRAfco Funcional de
Control Etapa Transición

SFC

Esquema secuencial de
funciones

Sequential Function Chart

Secuencial

En un sistema secuencial el valor del estado actual de una secuencia depende, además de valores de entrada de dicha secuencia, del estado de secuencias o etapas anteriores.

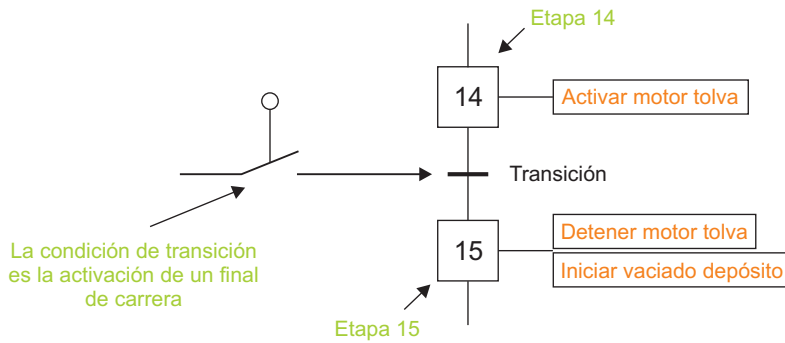


Figura 10.120. Transición entre etapas.

Condiciones de transición

Las condiciones de transición pueden ser varias y efectuar diferentes operaciones:

- Temporizadores.
- Contadores.
- Comparadores.
- Asociaciones en serie.
- Asociaciones en paralelo.
- Flancos positivos.
- Flancos negativos.
- Negaciones.

14.6.3. Operaciones en GRAFCET

Salto condicional de etapa

Se podrá dar en sentido ascendente, o descendente, y supone que se omite la ejecución de ciertas etapas.

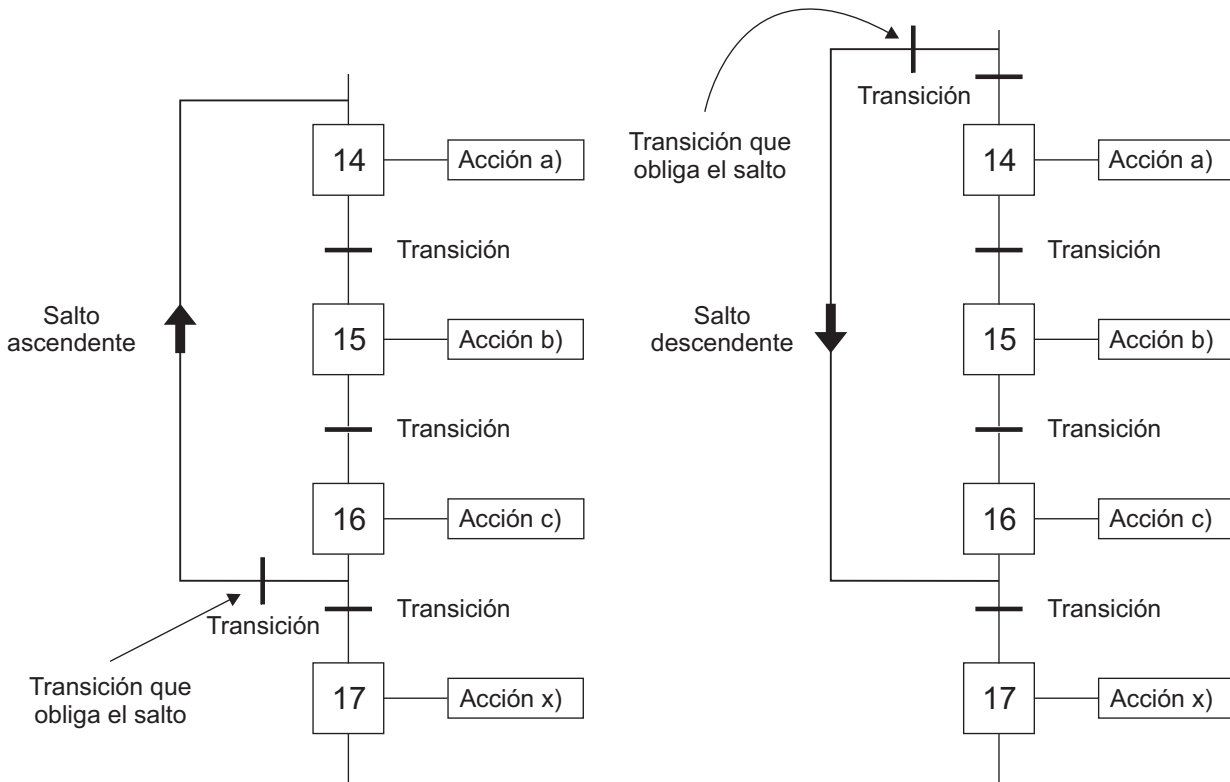


Figura 10.121. Operaciones de salto condicional.

Direccionamiento condicional

Las transiciones determinarán el camino de desarrollo de las etapas. En el siguiente gráfico, se muestra un proceso de embotellado, taponado y etiquetado. Unos sensores determinarán el tamaño de la botella, y en consecuencia, el volumen de líquido, tamaño de tapón y dimensión de la etiqueta.

Según este detector, se efectuará un camino u otro de secuencias. Note, que las diferentes secuencias pueden culminar en una única línea de programación.

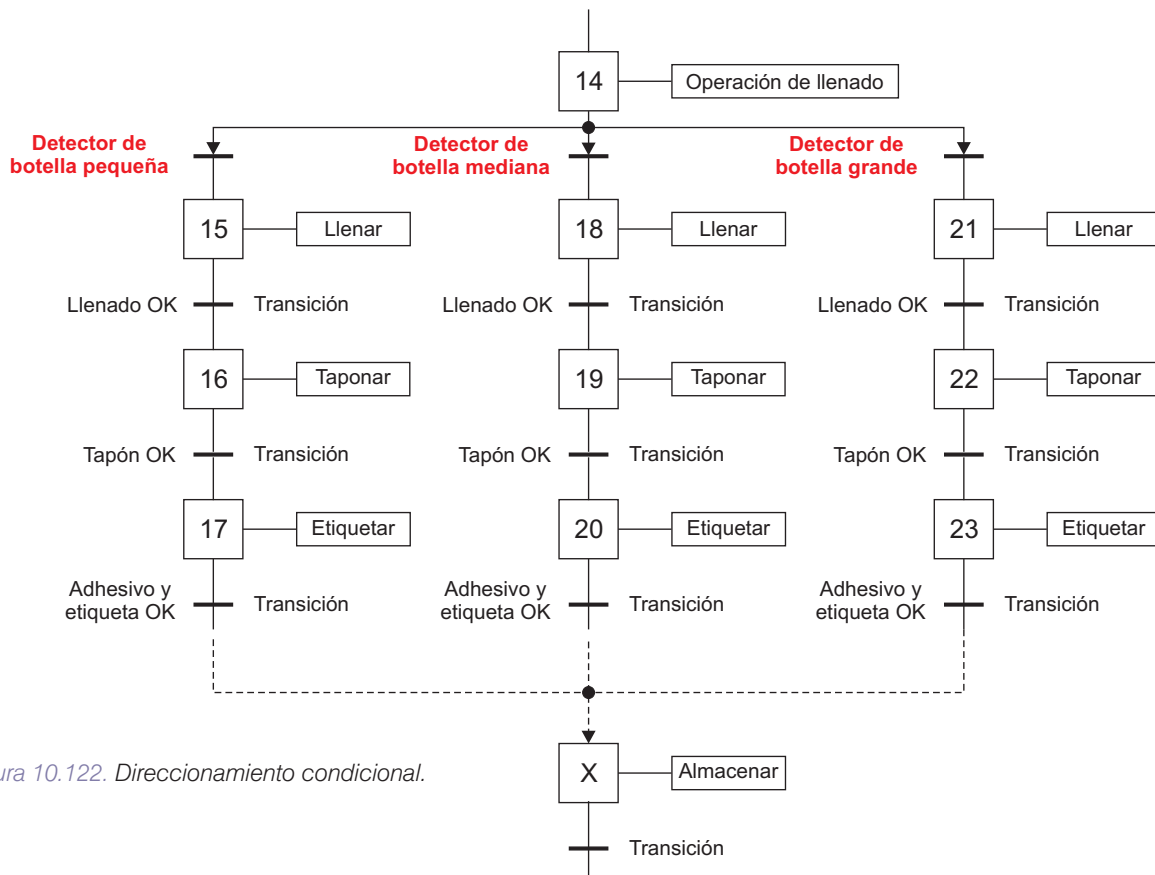


Figura 10.122. Direccionamiento condicional.

Secuencias simultáneas

Dos trazos en paralelo, nos indican que se van a realizar varios estados operativos al mismo tiempo. El retorno de las diferentes secuencias también pueden converger en una línea.

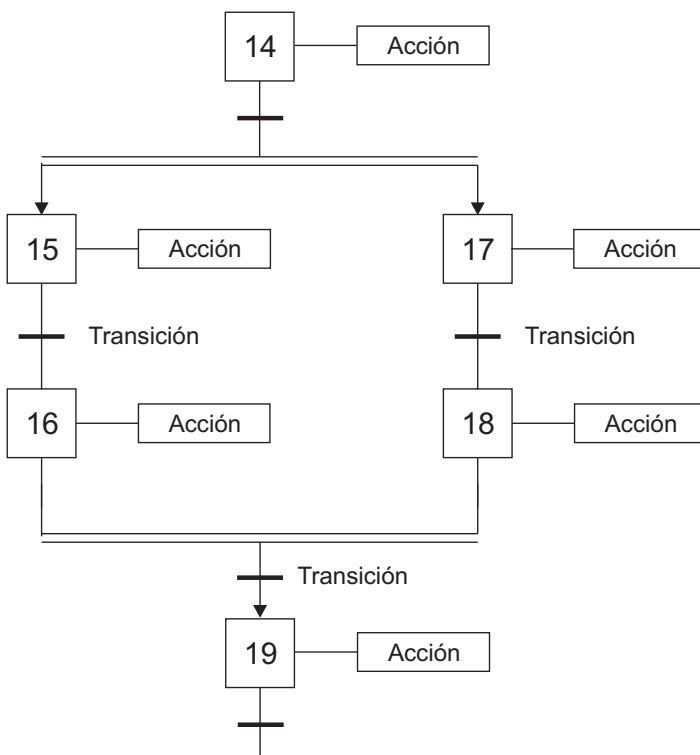


Figura 10.123. Secuencias simultáneas.

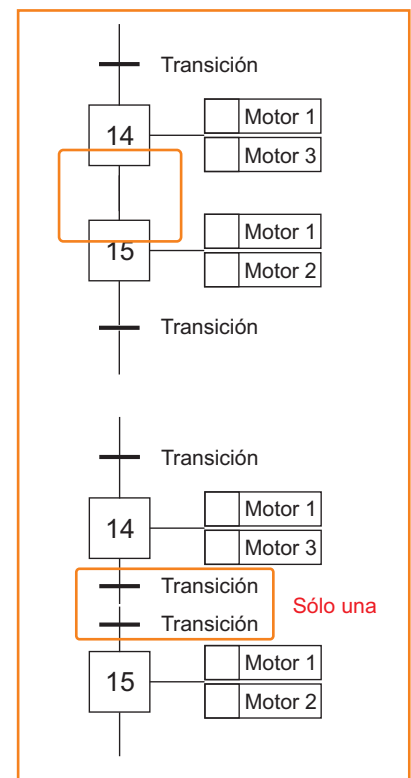


Figura 10.124. Recuerde, que entre etapa y etapa debe existir una transición, no permitiéndose más.

10 Autómatas programables

Ejercicio

Semáforo. Con la actuación de un pulsador, se activará de forma directa la lámpara ámbar; a los tres segundos, se desconecta ámbar y se activa la luminaria roja; a los 10 segundos, se apaga ésta, y se excita la lámpara verde; pasados 7 segundos, el proceso se repite con el encendido de la luz ámbar.

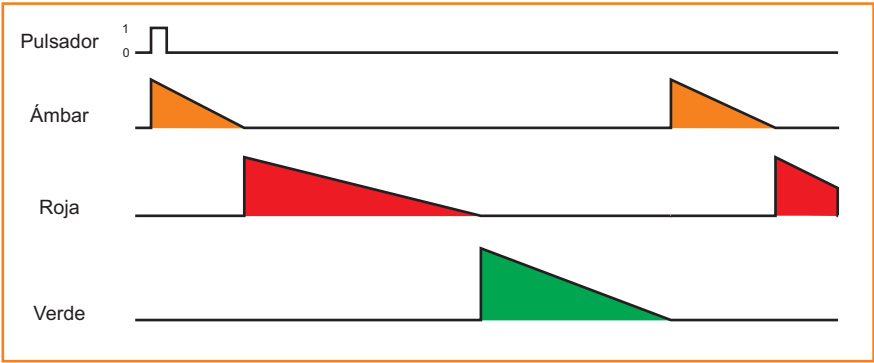


Figura 10.125. Cronograma del ejercicio propuesto.

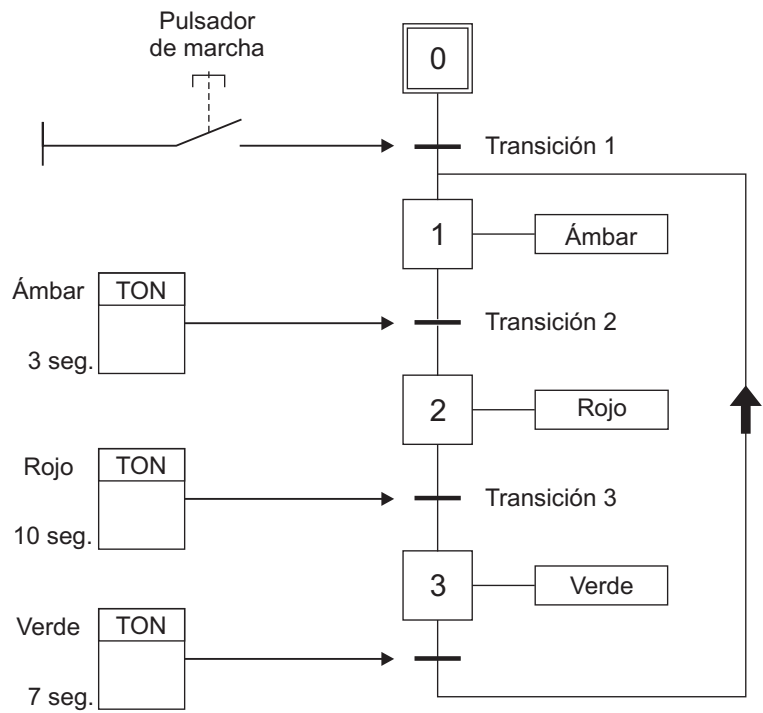


Figura 10.126. GRAFCET del ejercicio propuesto.

Ventajas del grafcet

- Permite un diagnóstico rápido del proceso.
- Simplifica los esquemas lineales.
- Permite distribuir un problema a otras secuencias.
- Se puede realizar la conversión a otros lenguajes de programación.

SET-RESET dentro de GRAFCET

Los sistemas de programación en GRAFCET, permitirán asociar directamente a las acciones de las etapas, funciones de SET- RESET.

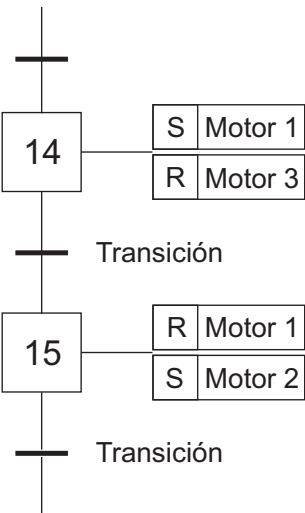


Figura 10.127. Las acciones de las etapas podrán usar operaciones SET-RESET

El GRAFCET, se puede representar en otros lenguajes de programación, siendo los más representativos FBD y LD. En la siguiente figura, aparece la programación del ejercicio propuesto anterior en lenguaje LD, con funciones SET-RESET.

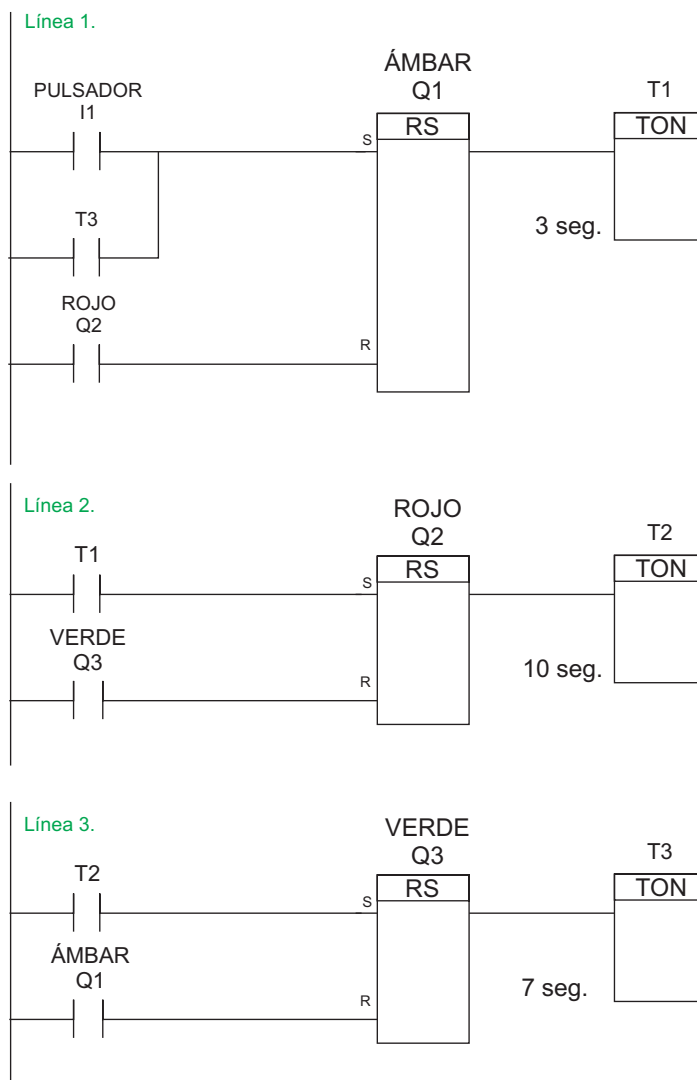


Figura 10.128. Programación en LD del ejercicio propuesto.

Los sistemas de programación en GRAFCET en tiempo real, identifican la etapa activa con colores llamativos.

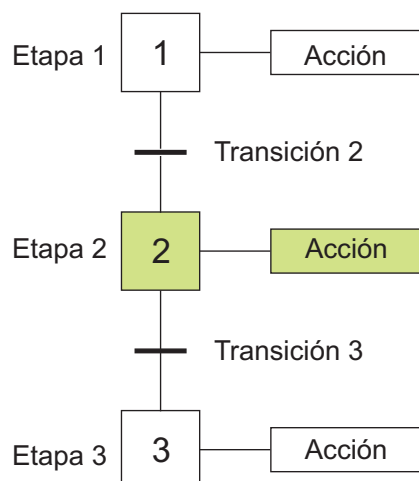


Figura 10.129. Identificación "on-line" de las etapas activas.

Representación de funciones SET-RESET

Las funciones SET- RESET, se pueden representar en forma de bobina, o a través de un bloque funcional, que recuerda a las funciones en lenguaje FBD.

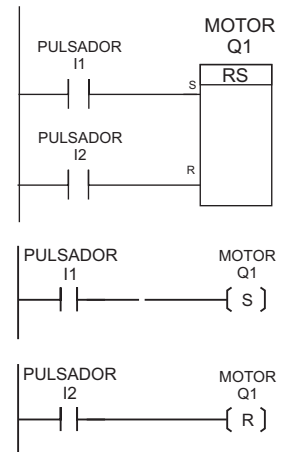


Figura 10.130. Representación de funciones SET-RESET.

10 Autómatas programables

El comportamiento del GRAFCET respecto a funciones SET-RESET, es el siguiente:

- La etapa en curso habilita el circuito.
- La condición de transición alimenta las funciones SET-RESET.
- La función SET, habilita la etapa siguiente y la función RESET anula la etapa actual.
- El ciclo se repite con la siguiente etapa-transición.

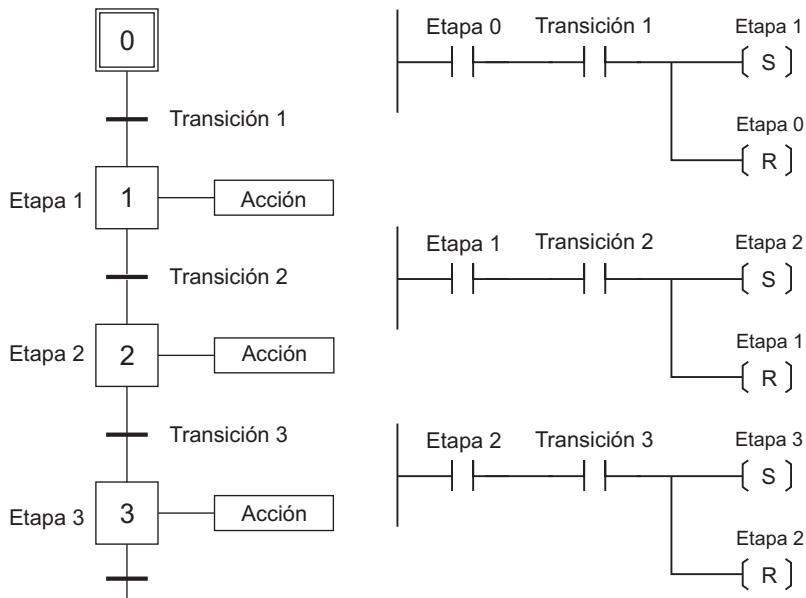


Figura 10.131. GRAFCET equivalente con funciones SET-RESET.

15 Diálogo Hombre-máquina, HMI (Human Machine Interface)

En esencia, nos referimos a dispositivos HMI, cuando el sistema se pone en contacto de manera sensorial con el hombre.

Básicamente podemos afirmar que pulsadores, pilotos o un simple potenciómetro son HMI, pero de forma generalizada hablamos de dispositivos HMI con visualizadores, pantallas táctiles, sistemas scada, consolas de programación, incluso ordenadores personales, usando comunicaciones cableadas o inalámbricas, con las cuales se puede programar, diagnosticar, recibir información en tiempo real, modificarla, etc.

15.1. Paneles, pantallas táctiles y visualizadores

Se conectan de forma directa al autómatas programable o a través de una red, basada en algún tipo de comunicación industrial y tienen características claramente definidas.

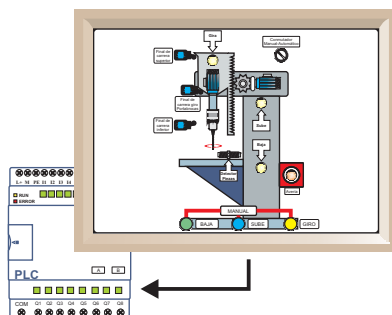


Figura 10.132. Manejo de una aplicación, a través de una pantalla.

Mecanismos "virtuales"

Pulsadores, interruptores, pilotos, deslizadores... Serán un "dibujo" o una "imagen" en algunas pantallas táctiles o paneles de control, aunque su comportamiento será el mismo como si fuera un mecanismo real, activándose, y en definitiva realizando la misma función. La ventaja, es que se pueden insertar componentes según necesidad en diferentes pantallas de control.

Algunos paneles de control incorporan botones físicos que también serán usados en el proceso. Insertar comentarios textuales o datos en tiempo real, será otra de las funciones importantes de los paneles.



Figura 10.133. Paneles de control con botones virtuales y físicos.

- Manejan variables de la memoria del PLC, por tanto, sustituyen de forma directa a pulsadores, interruptores, conmutadores, etc.
- Por tanto, modifican parámetros en el sistema.
- Advierten sensorialmente de diversos modos el estado de la programación, y por consiguiente del sistema en general (gráficos, avisos acústicos, alarmas visuales...).
- Se pueden conectar con elementos de ámbito superior, como ordenadores centrales tipo host.
- El control de las instalaciones se puede pormenorizar y detallar en diferentes pantallas.
- Muestran parámetros de los procesos.

Existen gran variedad de dispositivos para control y visualización, adaptándose a la dimensión de la instalación. Los paneles de pequeño calibre se conectarán directamente al autómata programable, aunque requiera de software específico de programación, en cambio, la puesta en marcha de grandes paneles requiere un conocimiento más especializado.

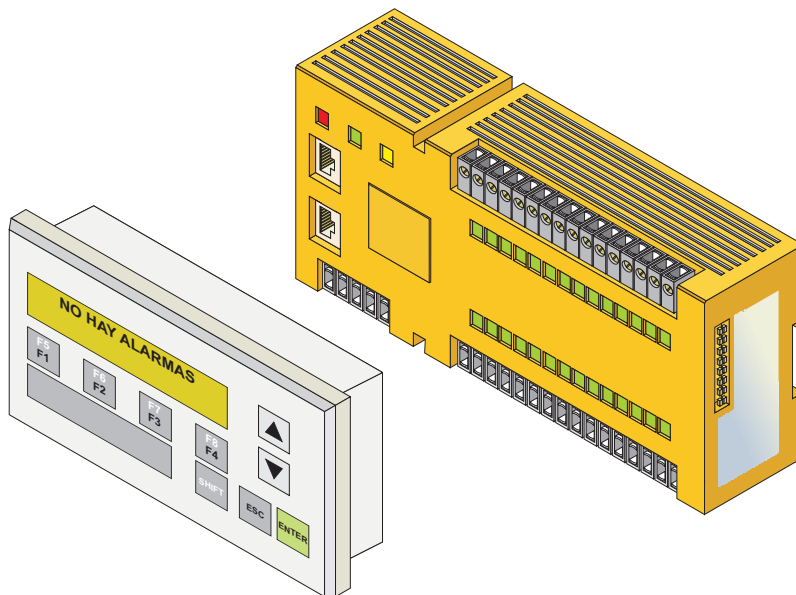


Figura 10.134. Pequeña pantalla de control y visualización.

PC industrial (ordenador industrial)

En este tema se está tratando el autómata programable como controlador de aplicaciones industriales, pero no es el único procedimiento. Los PCs industriales (ordenadores industriales), son máquinas desarrolladas para trabajar en ambientes industriales, y a través de tarjetas de entradas-salidas, realizan un control exhaustivo de los sistemas automatizados.

Inicialmente, el modo de programación es propietario, basado en lenguajes de alto nivel, pero cada vez más, la programación de estas máquinas se realiza con los lenguajes definidos por el estándar IEC 1131.

La robustez y características internas de un PC industrial, dista bastante de un ordenador doméstico.

15.2. Programas SCADA

La palabra SCADA significa Supervisory Control and Data Acquisition, supervisión, control y registro de datos. Un SCADA es un programa que tiene como trabajo, la comunicación en tiempo real con los dispositivos que realizan aplicaciones industriales.

Desde el programa SCADA, se podrá gobernar el proceso productivo por varios factores:

- Recibe información pormenorizada en tiempo real, por ejemplo, a través de señales analógicas o digitales. En la pantalla se visualizarán las señales en diferentes formatos gráficos.
- Se pueden ejecutar órdenes directas desde la pantalla.
- Almacena los datos recibidos, y genera documentos de control (históricos, gráficas, estadísticas, etc.).
- Puede mantener comunicación con otros soportes.

10 Autómatas programables

El programa SCADA no gobierna de manera directa la aplicación. Para interconectar funcionalmente sensores y actuadores, se podrá utilizar un autómata programable o un PC industrial, y el programa SCADA se unirá bidireccionalmente para el fluido de datos.

Una duda razonable que aparece cuando se trata un programa SCADA, es la compatibilidad con el autómata programable al que está conectado, sobre todo, si ambos son de diferentes marcas. Existe un programa intermedio OPC, que es un driver para la comunicación entre el SCADA y el PLC. La misión de este driver consiste en que exista tráfico de datos entre un dispositivo y otro, para que la transferencia de información se dé con fluidez.

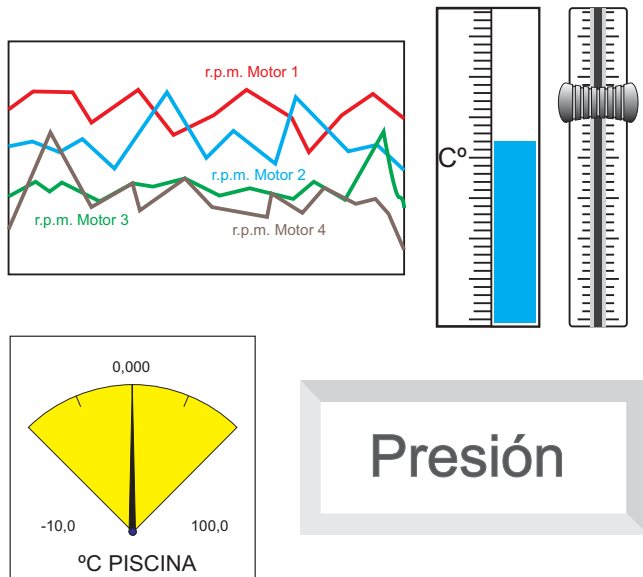


Figura 10.135. Controles analógicos típicos de programas Scada.

Depósitos en pantallas

El control del nivel de los depósitos, es una de las aplicaciones típicas para un sistema de seguimiento visual. Se consigue colocando un depósito -a medir- junto a una barra gráfica, o incluso el depósito "hueco", permitiendo advertir el nivel equivalente al real.

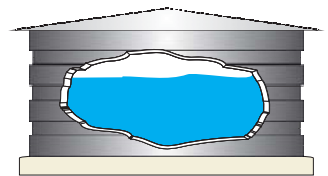


Figura 10.137. Control gráfico de un depósito.

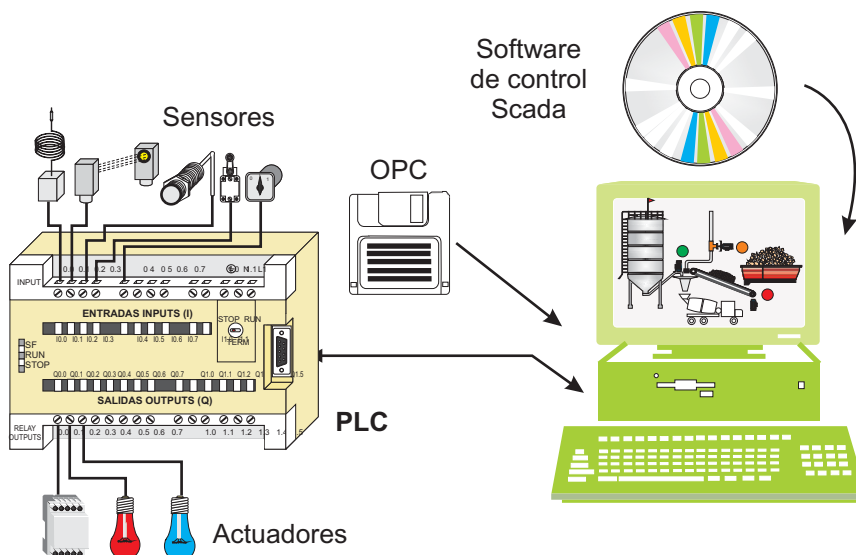


Figura 10.136. Esquema de conexionado de un programa SCADA a través de un driver OPC.

Además de los botones y pilotos virtuales antes mencionados para paneles de operación, los programas SCADA, manejan de manera fluida, principalmente para señales analógicas, barras gráficas, medidores analógicos, curvas, y para el manejo directo de valores se emplean deslizadores -slider-, que son como botones potenciómetros.

15.3. Sinópticos

En primer lugar, hay que reseñar que cualquier elemento dentro y fuera de un cuadro eléctrico, debe estar bien indicado. En el caso de pilotos, pulsadores, conmutadores, aparatos de medida, y cualquier mecanismo que se sitúe en el exterior, debe llevar una etiqueta que aporte la información suficiente de cuál es su cometido.



Figura 10.138. Etiquetas identificativas permanentes en aluminio.

Cuando el propio armario de automatismos es el que muestra el proceso, se puede recurrir al uso de sinópticos informativos, que a través de gráficos, muestran la secuencia de trabajo de las máquinas, incluyendo dentro del mismo mecanismos, que serán vía de conexión con la instalación.

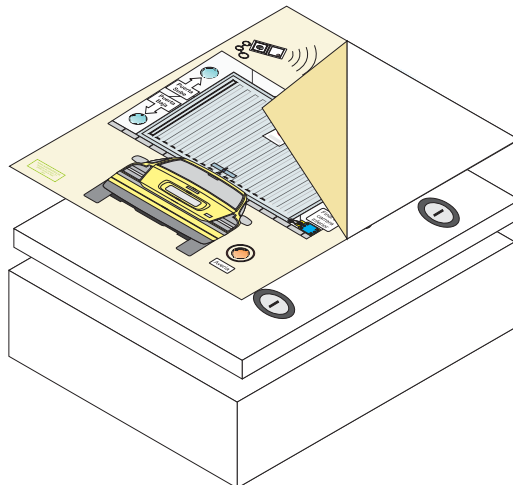


Figura 10.139. Sinóptico adhesivo para un cuadro eléctrico.

Redes industriales

EtherCat

www.ethercat.org

Profinet

www.profinet.com

DeviceNet

www.anybus.com

Modbus

www.modbus.org

Profibus

www.profinet.com

AS-I

as-interface.net

IO-LINK

www.io-link.com

16 Comunicaciones industriales

Si observamos el siguiente gráfico, podemos encontrar algunas apreciaciones.

- La red de color verde, une un ordenador central, con otros dispositivos, como pueden ser autómatas programables de gran potencia, visualizadores, y otros dispositivos similares. Por esta red, viajan datos a alta velocidad, y suele ser del tipo Ethernet.
- La red de color amarillo, une autómatas programables con visualizadores, otros ordenadores, pantallas táctiles, con una velocidad de transferencia de datos menor.
- La red color azul, une algunos de dispositivos de control con elementos de campo, como micro-autómatas, programables, variadores de frecuencia, arrancadores estáticos, etc.

10 Autómatas programables

Bus de comunicaciones

Un bus es el camino físico por el cual viajan los datos en una comunicación industrial. Los cables suelen ser específicos y en algunos casos apantallados y/o blindados.

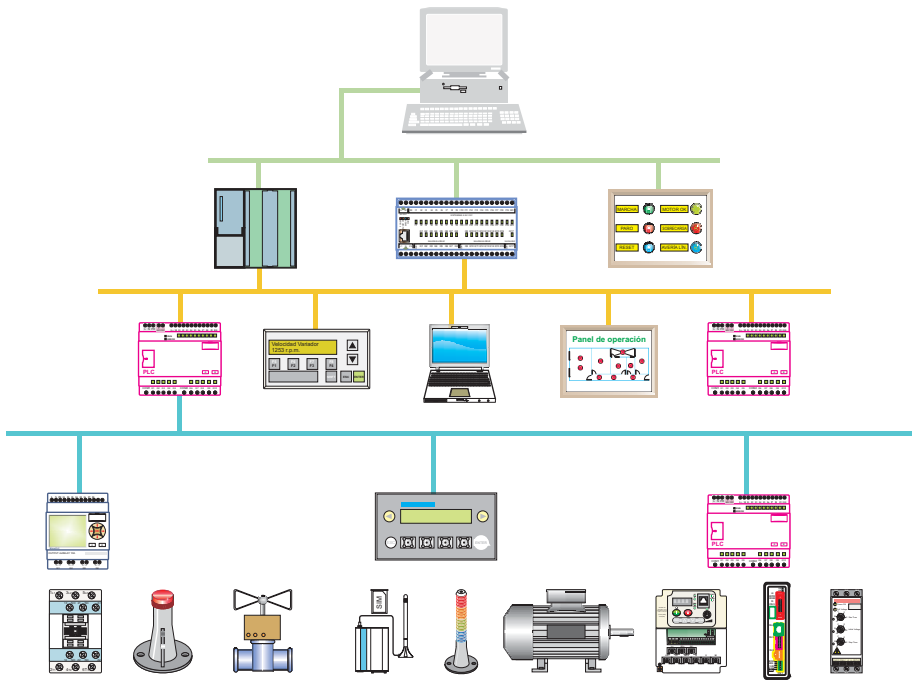


Figura 10.140. Diferentes redes de comunicaciones industriales.

Las comunicaciones industriales son protocolos de comunicación de dispositivos de carácter industrial, con el propósito de compartir información para que exista fluidez en el proceso productivo.

Unas redes son más complejas que otras, y por tanto, el tráfico de datos difiere. Las redes más básicas, unen sensores y actuadores a un controlador, y la información que se dirige entre ellos se realiza en forma de bit a modo de maestro-esclavo. Por otro lado, las redes de más nivel, permiten tráfico de datos en formatos tipo 16, 32 y 64 bits, en modos maestro-esclavo, maestro-maestro.

En la siguiente figura aparece un esquema de una red AS-interface, donde los componentes se comportan como esclavos del PLC.

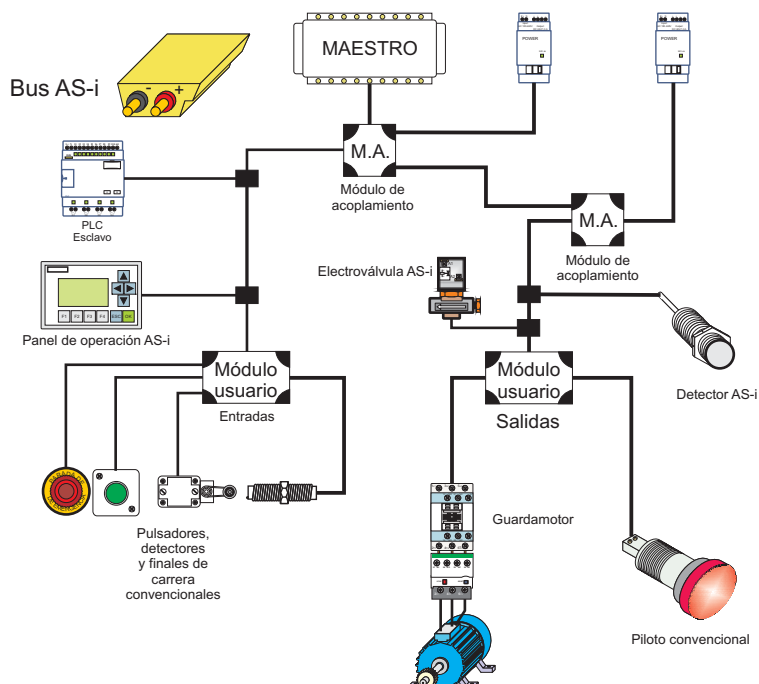


Figura 10.141. Red AS-interface.

Módulos de comunicaciones

Al existir variados sistemas de comunicaciones industriales, se hace necesario el uso de tarjetas o módulos convertidores de un sistema en concreto. En el siguiente gráfico se muestra un módulo de comunicaciones de un PLC que gobierna variadores de frecuencia. El módulo en este caso es el comienzo de una red industrial.

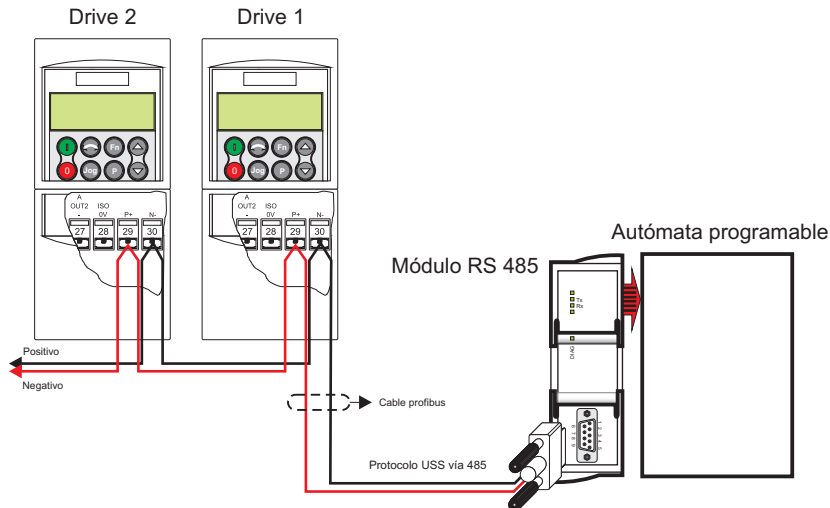


Figura 10.142. Módulo de comunicaciones.

Convertidor

En el siguiente ejemplo, un convertidor permitirá que un autómata programable que se programa con un protocolo concreto, sea programado por otro protocolo diferente.

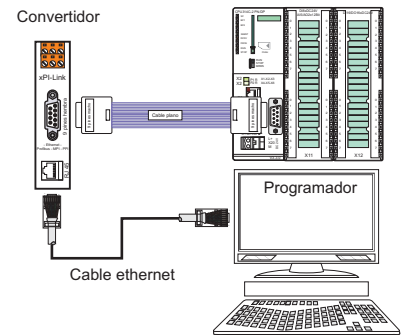


Figura 10.145. Convertidor de señales.

Convertidores de protocolos

Si disponemos de una red de carácter industrial, pero no tenemos elementos capaces de hablar el idioma de la citada red, es muy probable que encontremos un elemento convertidor de un protocolo a otro. Estos dispositivos son de carácter electrónico.

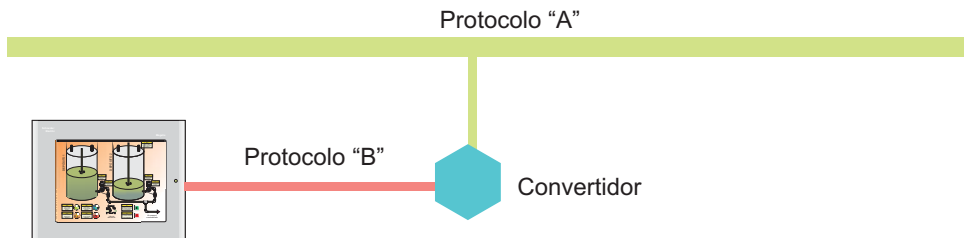


Figura 10.143. Convertidor de protocolos.

Redes industriales inalámbricas

La comunicación entre dispositivos automáticos se puede realizar con comunicaciones inalámbricas robustas, asegurando el flujo de datos ininterrumpidamente.



Figura 10.144. Comunicación inalámbrica entre componentes.

10 Autómatas programables

17 Sensores analógicos y autómatas programables

Un apartado importante dentro de la instalación y programación de autómatas programables, son los sensores con salida analógica a tensión o intensidad. Pondremos algunos ejemplos.

Sensor de presión con salida a intensidad

El módulo de entrada analógica recoge la intensidad que se dispone en serie con el sensor.

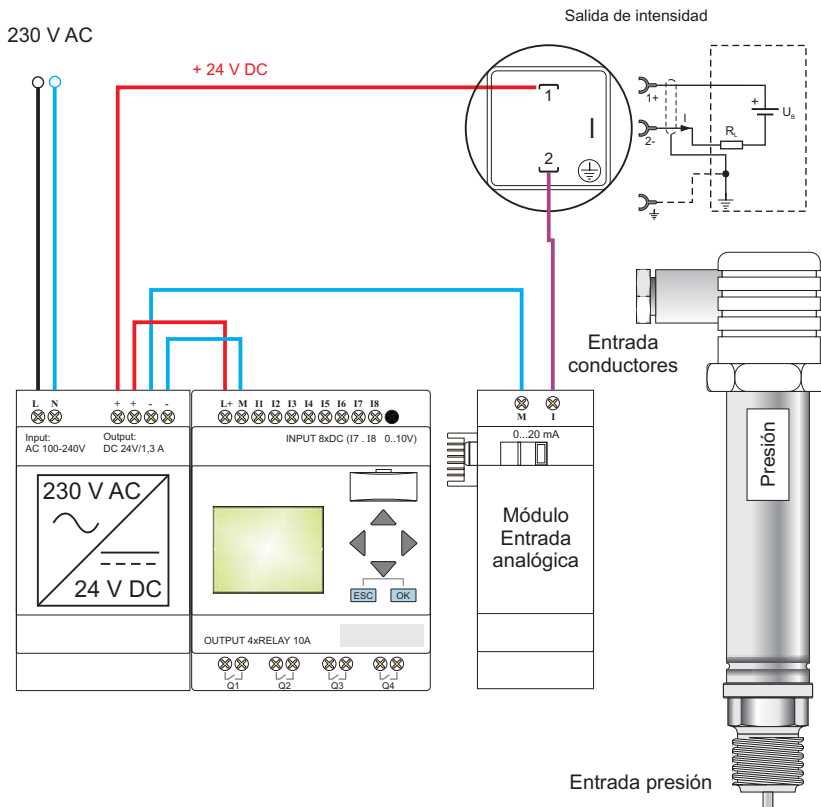


Figura 10.146. Conexión de un sensor de presión analógico.

Sensor resistivo con salida 0 - 10 V DC

Por ejemplo, un potenciómetro. Si se dispone de una tensión de 10V DC, la conexión se puede hacer de manera directa. En el ejemplo siguiente, partimos de una fuente de alimentación de 24V DC, y para reducir el valor hasta 10 V, se conecta una resistencia en serie con el potenciómetro. El resultado, con el potenciómetro se obtiene un valor variable de 0 a 10V DC.

El valor final de las resistencias dependerá en todo caso del módulo del PLC.

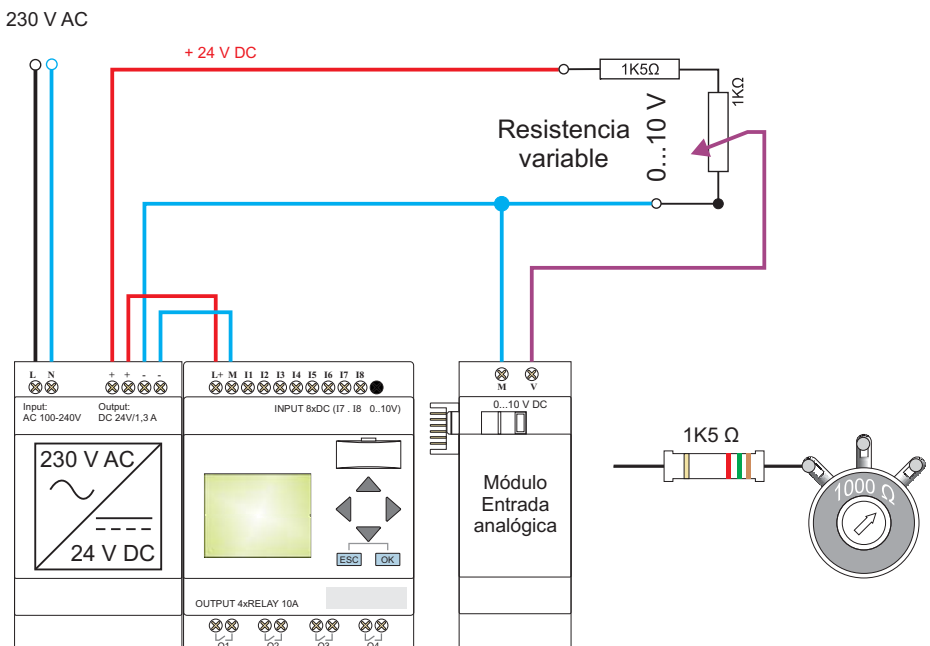


Figura 10.147. Conexión de un potenciómetro, como sensor resistivo.

Programación

Los módulos de entrada analógicos, suelen trabajar con valores de 16 bits. Un ejemplo típico para tratar los datos es el siguiente, el valor del sensor que oscila entre 0 y 32768 es dividido por 327'68, para que el valor final oscile de 0 a 100.

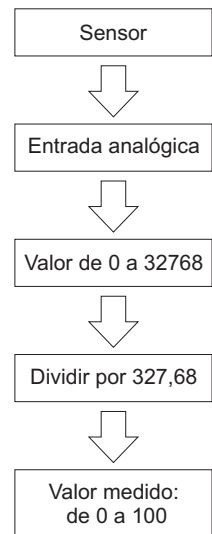


Figura 10.148. Conversión de datos de un Sensor analógico.

Sensores específicos

Si disponemos de un sensor analógico, pero cuyos valores de salida no son los estándar que admiten los módulos de entradas analógicas de los PLC, debemos recurrir al acondicionamiento de la señal para que la reconozca a través de transductores. Será fácil encontrar un dispositivo conversor en el mercado. En el siguiente gráfico se muestra una sonda de temperatura PT100, que puede ser a dos, tres o cuatro hilos. Para conectarla al PLC, debemos buscar un módulo específico para PT100, o un conversor de la misma, a valores de tensión o intensidad.

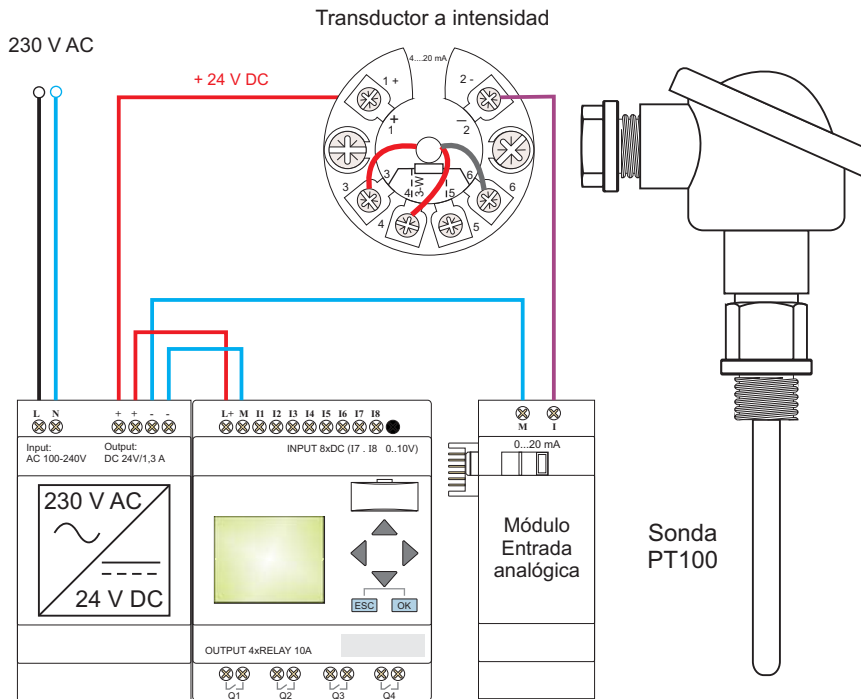


Figura 10.149. Conexión de una sonda de temperatura PT100 a un módulo de entrada analógica de intensidad, a través de un transductor.

Programación en Ladder

Ejemplo

La entrada analógica se lleva a una variable en formato de 16 bits. Se convierte a número real para poder trabajar con decimales. Se divide por 327,68 y el resultado se "guarda" en una variable de 32 bits. Desde ésta se podrá comprobar el estado actual de la entrada analógica AI1 con valores de 0 a 100.

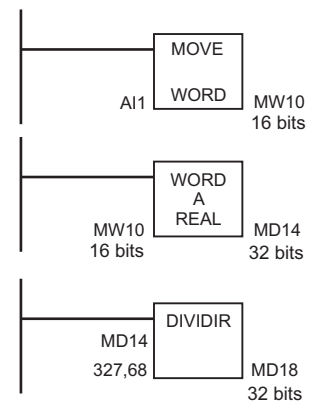


Figura 10.151. Ejemplo de programación en LD para una medida analógica.

18 Ubicación del autómata programable dentro del cuadro

Aunque partimos de la base en ubicar el PLC en el lugar del cuadro que reciba menor temperatura, cada fabricante aconseja en el manual del mismo, el lugar idóneo para su colocación.

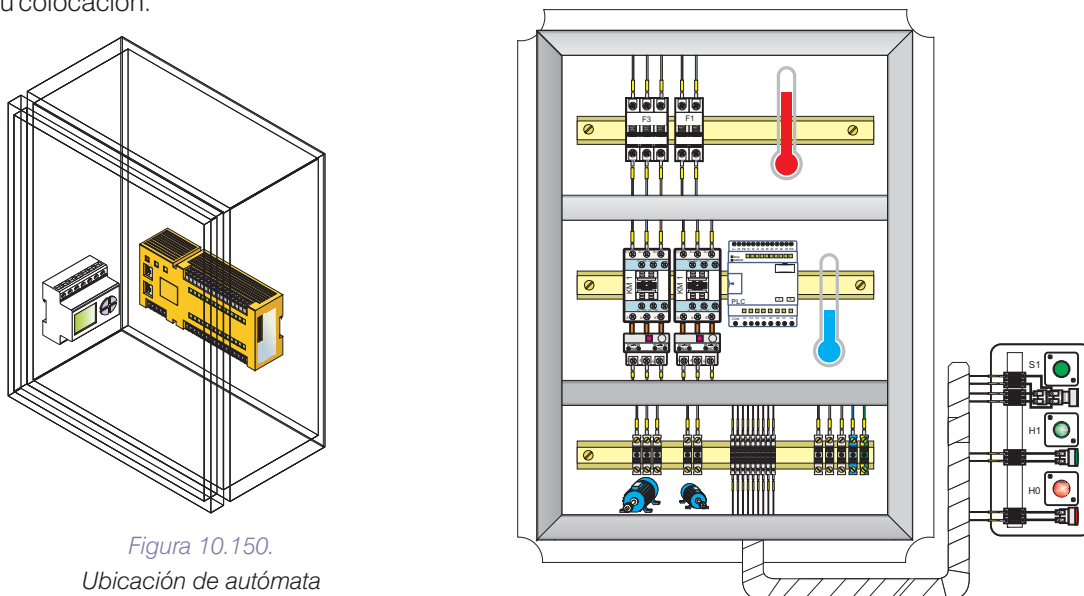


Figura 10.150. Ubicación de autómata dentro del cuadro.

19 Diagnóstico de autómatas programables

19.1. Comprobación de la alimentación

Tanto si dispone de fuente de alimentación, como si se conecta directamente a la red, debemos advertir que le llega corriente eléctrica.



Figura 10.152. El led de la fuente de alimentación es un indicador de que la alimentación está correcta.

19.2. Comprobación de entradas digitales

Para comprobar si las entradas digitales imprimen señal al autómata programable, se puede forzar directamente la alimentación de éstas, y hacerlas llegar a los bornes. Los led de registro de entrada se deben encender, y en programación debe suceder lo mismo. El conexionado se hará sin corriente.

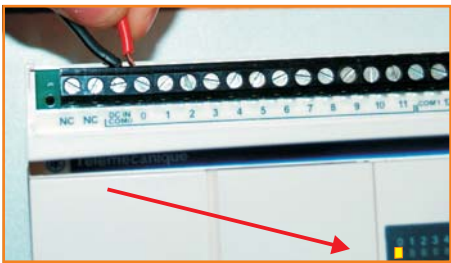


Figura 10.153. Al forzar la conexión de una entrada, la programación debe reflejarlo.

19.3. Comprobación de entradas analógicas

En el módulo de entrada analógica debemos medir tensión (en paralelo) o intensidad (en serie), con el sensor que deba enviar la señal. La programación debe reflejar la variación: AI1 = valor.

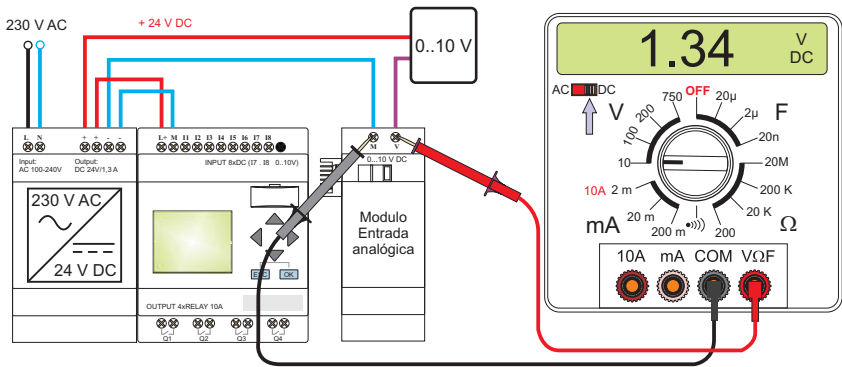


Figura 10.154. Comprobar que le llega tensión al módulo de entrada.

Medidas de seguridad

Todas las operaciones de cableado, conexonado, tanto para una puesta definitiva como para una comprobación de puesta en marcha, se deben realizar sin corriente eléctrica.

Además de esto, se tendrá en cuenta el manual del fabricante del dispositivo que se tiene entre manos, ya que no debemos olvidar que los autómatas programables son aparatos electrónicos y un error en el conexionado de los mismos puede ocasionar el deterioro inmediato.

Se realizarán, por tanto, todas las comprobaciones previas antes de aplicar corriente.



19.4. Comprobación de salidas digitales a relé

Forzando una programación, en la cual se deba activar una salida en concreto, y sin nada conectado a ella, se utilizará un multímetro, para medir continuidad. El relé interno debe actuar cerrando el circuito, si todo está correcto.

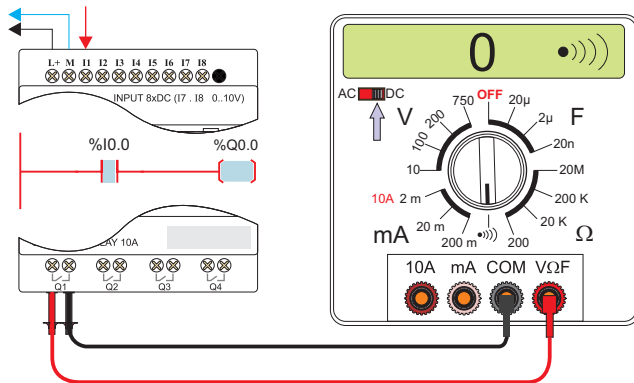


Figura 10.155. Comprobar continuidad en el relé interno.

19.5. Comprobación de salidas digitales a transistor

Forzando una programación, en la cual se deba activar una salida en concreto, y sin nada conectado a ella, se utilizará un multímetro, para medir corriente continua. Recuerde que el módulo de salidas a transistor deberá estar alimentado con corriente continua.

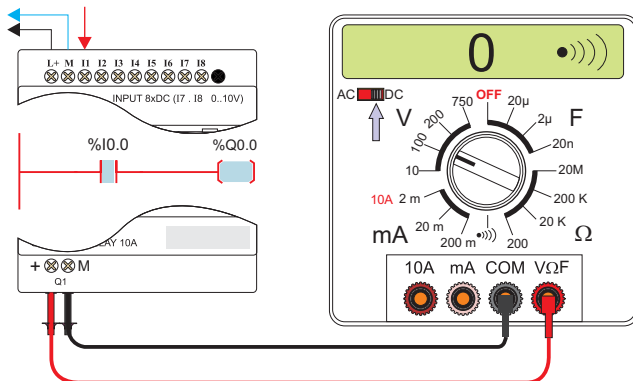


Figura 10.156. Comprobar tensión en la salida a transistor.

19.6. Comprobación de salidas analógicas

El módulo de salidas analógicas de un PLC ofrecerá típicamente una señal de tensión de 0 a 10 V DC, o una señal de intensidad de 0 a 20 mA, ó 4 a 20 mA.

En el primer caso, y suponiendo una programación que habilite cierto valor de salida, bastará con comprobar tensión en los bornes de salida; en el segundo caso, se hace necesario conectar un receptor para situar el amperímetro en serie con la carga.



Figura 10.157. Comprobar tensión o intensidad en la salida analógica.

Programación

Programación para forzar un valor de salida analógica.

Suponiendo que la salida analógica es capaz de procesar datos de 16 bits, movemos el máximo valor a dicha salida, con lo cual la tensión o intensidad de salida, será el valor máximo.

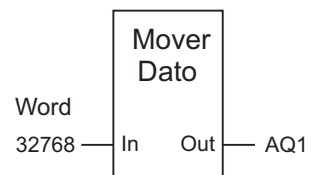


Figura 10.158.

Mover el máximo valor a una salida analógica.

10 Comprueba tus conocimientos

1. Cuando el funcionamiento de una operación automática se realiza con la alimentación principalmente de relés y contactores:

- ☐ Es lógica programada.
- ☐ Es lógica cableada.
- ☐ Es lógica automática.
- ☐ Ninguna de las anteriores.

2. ¿Cómo se llamó el primer autómatas comercializado?

- ☐ ADLON.
- ☐ MODLON.
- ☐ MODICON.
- ☐ Ninguna de las anteriores.

3. El término "PLC", quiere decir:

- ☐ Controlador lógico de cables.
- ☐ Controlador lógico de relés.
- ☐ Controlador lógico de memorias.
- ☐ Controlador lógico programable.

4. Si una instalación cuenta con un número de sensores superior a 5, y es susceptible de modificar con frecuencia, se debe utilizar:

- ☐ Lógica cableada, basada en relés y contactores.
- ☐ Lógica programada, basada en autómatas programables.
- ☐ Si la instalación tiene menos de 50 dispositivos de entrada-salida, se usarán relés y contactores.
- ☐ Lo que resulte más económico en la instalación inicial.

5. De las afirmaciones que aparecen a continuación, referidas a las operaciones que puede realizar un PLC, indica cuáles son falsas. Elige dos.

- ☐ Realiza operaciones aritméticas.
- ☐ Realiza operaciones con salidas analógicas.
- ☐ Su CPU es muy superior a la de un PC.
- ☐ Se puede programar en diferentes lenguajes de programación.
- ☐ Puede manejar un número ilimitado de entradas y salidas.
- ☐ Se puede utilizar de manera remota.

6. Aunque no es una afirmación que esté claramente definida, según algunas empresas, un autómatas será "micro":

- ☐ Cuando el número de entradas-salidas no es superior a 128.
- ☐ Cuando dispone de menos de 2 entradas analógicas.
- ☐ Cuando dispone de una salida analógica.
- ☐ Cuando el número de entradas-salidas no es superior a 32.

7. La señal de un anemómetro, utilizará el módulo del autómatas:

- ☐ Módulo de salida analógica.
- ☐ Módulo de comunicaciones.
- ☐ Módulo de entrada analógica.
- ☐ Módulo de entrada digital.

8. La señal de un final de carrera, utilizará el módulo del autómatas:

- ☐ Módulo de salida analógica.
- ☐ Módulo de comunicaciones.
- ☐ Módulo de entrada analógica.
- ☐ Módulo de entrada digital.

9. La señal de un detector capacitivo, utilizará el módulo del autómatas:

- ☐ Módulo de salida analógica.
- ☐ Módulo de comunicaciones.
- ☐ Módulo de entrada analógica.
- ☐ Módulo de entrada digital.

10. De los sensores que se muestran a continuación, elige tres que son activos.

- ☐ Detector de posición.
- ☐ Detector inductivo.
- ☐ Célula fotoeléctrica.
- ☐ Interruptor.

- ☐ Sónar.
- ☐ Pulsador.

11. Un dispositivo que transforma una magnitud física, en una señal -generalmente eléctrica- que posteriormente será tratada, es:

- ☐ Conversor de señales.
- ☐ Transductor.
- ☐ Autómatas programable.
- ☐ Transistor.

12. En un módulo de salidas de un PLC, ¿cuál es el dispositivo principal en su constitución, para que los receptores conectados funcionen a corriente continua, y tengan conmutaciones muy rápidas?

- ☐ Relés.
- ☐ Triacs.
- ☐ Transistores.
- ☐ Diodos.

13. Un Byte, ¿cuántos bits, tiene?

- ☐ 1.
- ☐ 4.
- ☐ 8.
- ☐ 16.

14. Una palabra (word), ¿cuántos bits tiene?

- ☐ 8.
- ☐ 16.
- ☐ 32.
- ☐ 64.

15. Direccionamiento de entradas. Cuando una entrada tiene siempre la misma dirección, por ejemplo I0.0., se trata de:

- ☐ Direccionamiento fijo.
- ☐ Direccionamiento variable.

16. El proceso en el cual el autómatas realiza un análisis del estado de las entradas, consulta al programa, y actualiza el estado de las salidas, es:

- ☐ Proceso de actualización.
- ☐ Ciclo de programa.
- ☐ Ciclo actualizador.
- ☐ Proceso interno de actualización.

17. Para programar autómatas programables. Elige tres.

- ☐ Se puede usar una consola de programación de la marca.
- ☐ Se puede usar en un PC estándar, con el software adecuado.
- ☐ Tiene que hacerse siempre con unidades de programación específicas de la marca del autómatas.
- ☐ Existe un cable común para todos los autómatas.
- ☐ Algunos microPLCs, se programan directamente sin necesidad de programas.

18. Si leemos I10.4.

- ☐ Es una variable de entradas digitales.
- ☐ Es una variable de marcas.
- ☐ Es una variable de salidas analógicas.
- ☐ No es ninguna variable.

19. Las variables, que se usan como relés internos dentro del autómatas, se suelen llamar:

- ☐ Memorias internas.
- ☐ Relés internos.
- ☐ Marcas.
- ☐ Temporizadores.

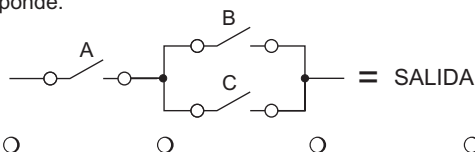
20. ¿Qué muestra la propiedad Conmutativa, según el álgebra de Boole?

- ☐ A + B = A · B
- ☐ A · B = A + B
- ☐ A + B = B + A
- ☐ A · B = B · A

21. Una proposición, cuya verdad se admite sin pruebas, aunque éstas son necesarias, para posteriores razonamientos, es:

- ☐ Postulado.
- ☐ Teorema.

22. Según este circuito, elige la tabla de la verdad que se corresponde.



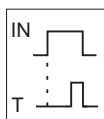
A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

23. Si encontramos un bloque funcional con el siguiente cronograma, ¿de qué elemento se trata?

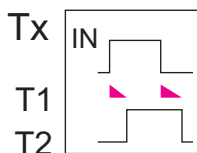


- ☐ Un temporizador con retardo a la activación.
- ☐ Un temporizador con retardo a la desactivación.
- ☐ Un telerruptor.
- ☐ Un contador ascendente-descendente.

24. ¿Qué norma IEC, establece 4 lenguajes de programación, dos textuales y dos gráficos, más el grafcet?

- ☐ 1131.
- ☐ 1160.
- ☐ 1134.

25. Si encontramos un bloque funcional con el siguiente cronograma, ¿de qué elemento se trata?



- ☐ Un temporizador con retardo a la activación.
- ☐ Un temporizador con retardo a la desactivación.
- ☐ Un temporizador con retardo a la activación-desactivación.
- ☐ Un contador ascendente-descendente.

26. ¿Qué bloque funcional implica a su salida un "1" lógico permanente, si en su entrada ha existido un impulso?

- ☐ Un temporizador con retardo a la activación.
- ☐ Un bloque SET-RESET.
- ☐ Un temporizador con retardo a la activación-desactivación.
- ☐ Un contador ascendente-descendente.

27. La operación flanco positivo, implementa un "1" lógico a su salida:

- ☐ Cuando la entrada pasa de "1" a "0".
- ☐ Cuando la entrada pasa de "0" a "1".

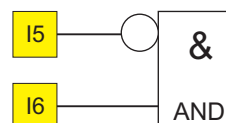
28. Comparadores. Elige dos que no son.

- ☐ Mayor o igual que.
- ☐ Menor que, más 1.
- ☐ Igual que.
- ☐ Igual a, menos 1.
- ☐ Menor o igual que.

29. Un pulsador S1 con contacto NA, conectado a la entrada digital I1 de un autómata programable. ¿Es lo mismo, el contacto abierto del pulsador S1, que el contacto en programación I1?

- ☐ Sí, a todos los efectos.
- ☐ Sí, solo mientras I1 sea abierto.
- ☐ Sí, sólo mientras I1 sea cerrado.
- ☐ No. I1 es una representación de S1, pero no tiene porqué ser Igual.

30. ¿Qué significa este "círculo"?



- ☐ Un borne de conexión.
- ☐ Una función inversora NOT, en representación reducida.
- ☐ Un borne que está siempre a "1".
- ☐ Un borne que está siempre a "0".

31. Si en un lenguaje de programación de PLCs, encuentro el texto "WHILE", el lenguaje es:

- ☐ Ladder.
- ☐ Texto estructurado ST.
- ☐ Lista de instrucciones IL.
- ☐ Grafcet.

32. Dentro del Grafcet, ¿qué implica una acción, dentro del proceso secuencial?

- ☐ Una etapa.
- ☐ Una transición.

33. ¿Qué quiere decir HMI?

- ☐ Diálogo máquina-máquina.
- ☐ Programación de máquinas.
- ☐ Diálogo hombre-máquina.
- ☐ Es un tipo de alarma en programación de autómatas.

34. ¿Qué es un OPC?

- ☐ Es un lenguaje de programación para autómatas.
- ☐ Es una marca de autómatas.
- ☐ Es un programa Scada.
- ☐ Es un programa "driver", para que un autómata se pueda comunicar con un sistema basado en Scada.

35. ¿Qué es un deslizador "slider"?

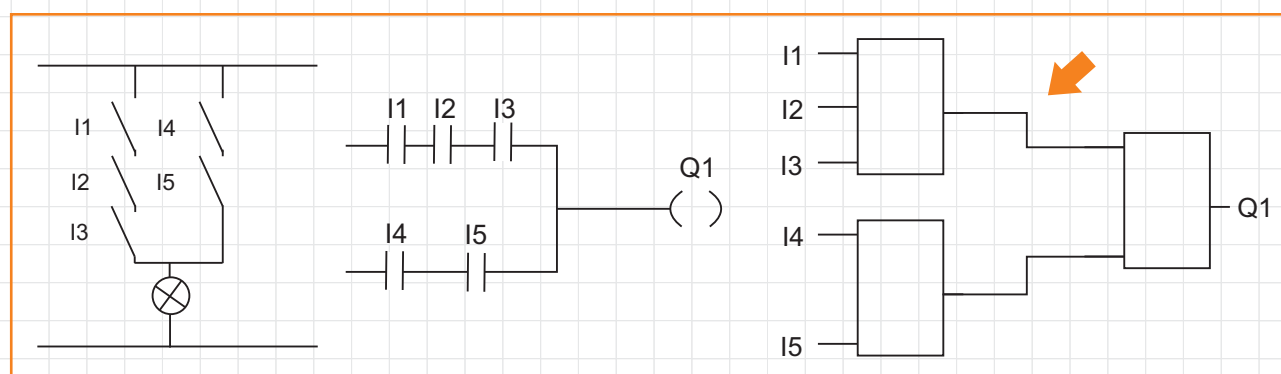
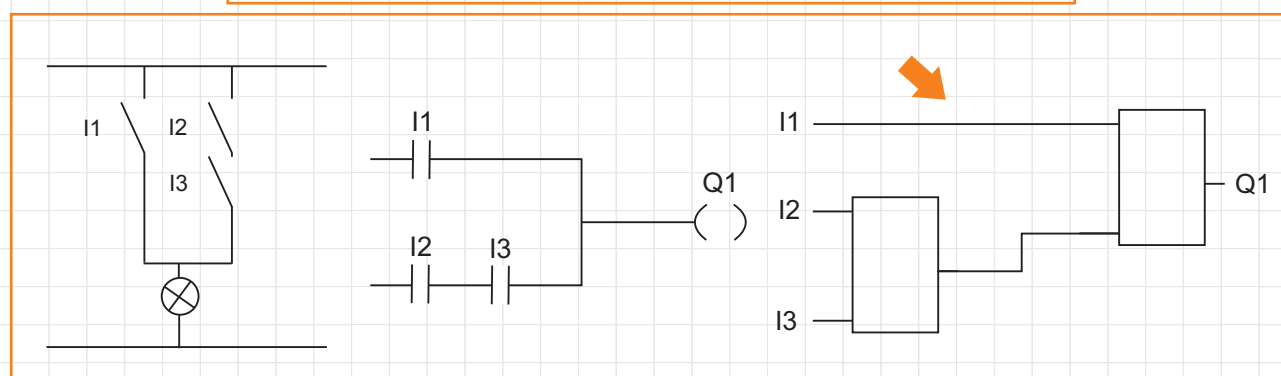
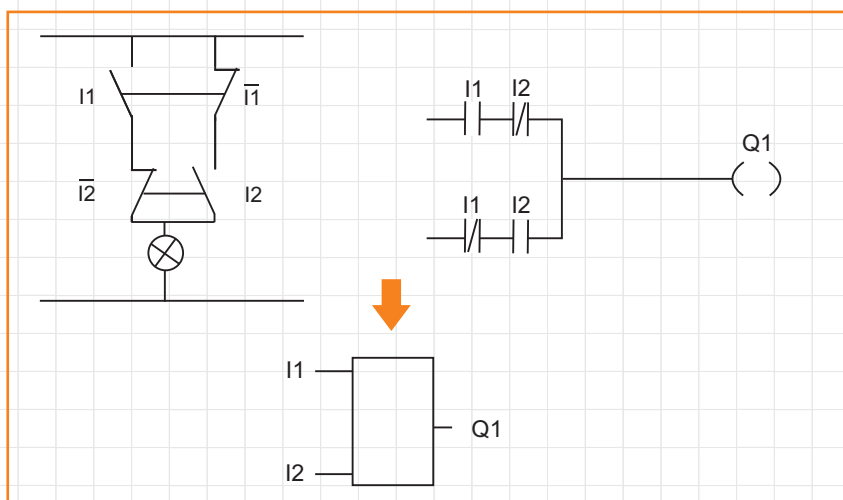
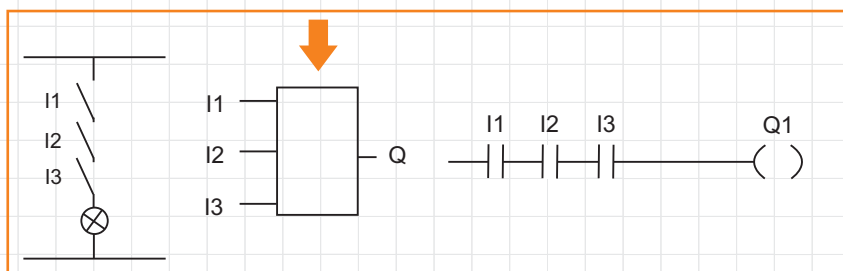
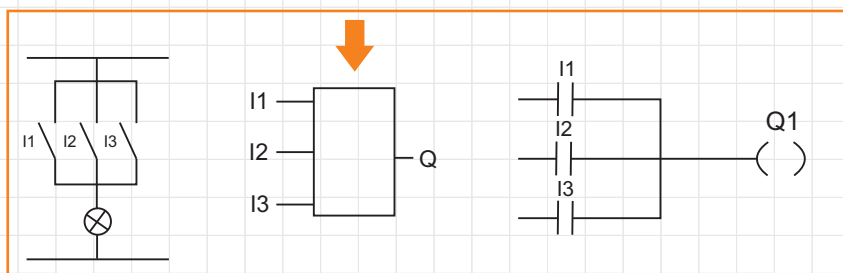
- ☐ Una representación gráfica de una señal analógica variable, por ejemplo el estado de llenado de un depósito.
- ☐ Una imagen dentro de una pantalla táctil o programa Scada.
- ☐ Un control virtual, para programas Scada, similares a los Potenciómetros.
- ☐ Un programa de pantallas táctiles.

36. Sensores analógicos industriales. Elige tres afirmaciones.

- ☐ Pueden tener salida a intensidad.
- ☐ El autómata mide sus datos en formato bit.
- ☐ El programa del autómata mide sus datos en formatos tipo 16 bits.
- ☐ Su valor dentro del programa se puede convertir y acondicionar aritméticamente.
- ☐ Se tienen que alimentar obligatoriamente a corriente continua.
- ☐ Cuando tienen salida a tensión los valores oscilan de 100 a 150 V DC.

10 Ejercicios

1 ¿Qué símbolo corresponde a las “cajas” vacías?



2 Según el siguiente esquema de mando, referido a la inversión de sentido de giro “brusca” de un motor trifásico, determina una programación afín para los lenguajes LD y FBD. Realiza una tabla de equivalencia entre dispositivos (pulsadores, contactores, relés, lámparas...), y el tipo de variables empleadas (I1, I0.4, etc.).

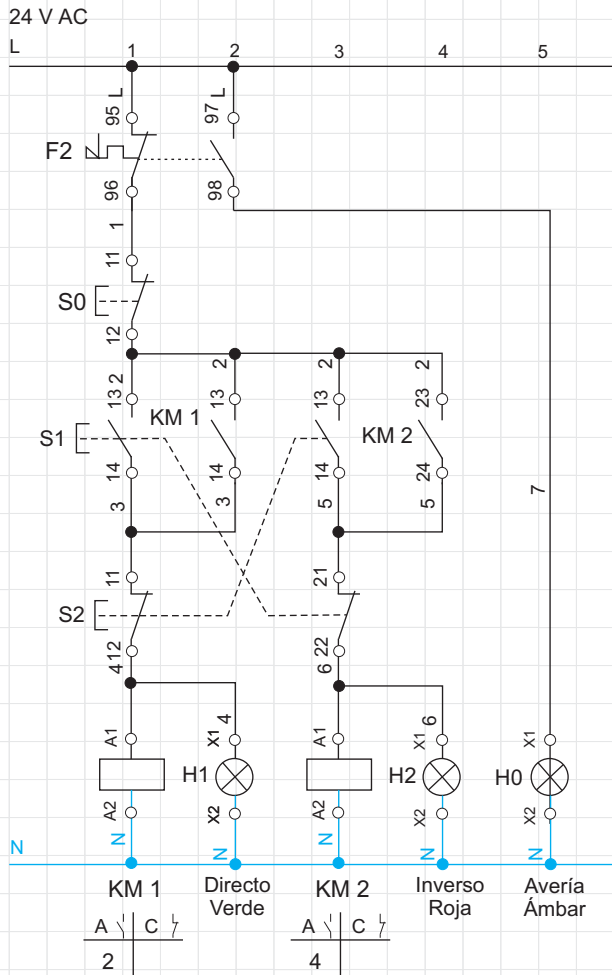


Figura 10.159. Esquema de mando del circuito a automatizar.

3 Según el esquema empleado en el ejercicio anterior, realiza el cableado en el PLC propuesto de la siguiente figura. El conexionado de componentes debe corresponderse con las programaciones descritas anteriormente.

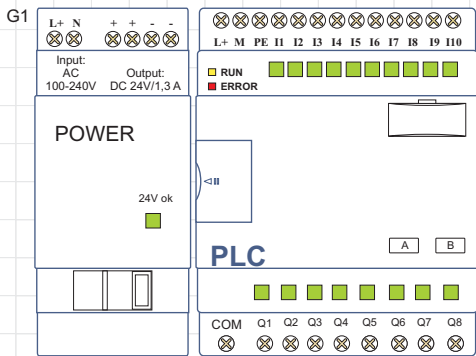


Figura 10.160. PLC propuesto para el ejercicio.

4 Realiza el Grafcet, de una programación que cumpla lo siguiente:

- Al presionar un pulsador S1, se activará el motor 1.
- Pasados 5 segundos, hará lo propio el motor 2, desactivándose el primero.
- Pasados 10 segundos, se excitará el motor 3, anulándose el segundo.
- Pasados 15 segundos, se conectionará el motor 4, parándose el tercero.
- Pasados 20 segundos, se detiene el motor 4. El proceso comenzará presionado de nuevo S1.

5 Realiza una programación en LD, usando funciones SET-RESET, que cumplan los condicionantes del ejercicio anterior.

6 Un sensor de presión analógico con salida a tensión 0-10 V DC, aporta al módulo de entrada analógica del PLC una señal que oscila entre 0 y 27500 bits. ¿Qué operaciones aritméticas se han de hacer para que el valor de la presión oscile entre 0 y 20?

7 Según el cableado de los sensores y actuadores del autómatas programable, mostrado en la siguiente página, determina los esquemas de los regleteros de los dispositivos que aparecen.

Alimentación	Cable							
	REGLETERO X1	Destino A						
		Nº	1	2	3	4	5	6
Cable	Destino B							
	Nº							

Figura 10.161. Esquema de un regletero.

8 Del libro “Cuaderno de prácticas para automatismos cableados y programados”, realiza los siguientes montajes prácticos.

- A. Puesta en marcha de un motor trifásico, mediante controlador programable.
- B. Juego de semáforos para educación vial.
- C. Automatización programada de una puerta de garaje.
- D. Aprovechamiento de aguas naturales.
- E. Máquina de lavado de vehículos manual, gestionada por PLC.
- F. Automatización de tres pozos para riego.
- G. Limpieza automática de aceitunas.
- H. Control automatizado de una puerta doble, de acceso a vehículos.
- I. Control automático de una prensa industrial.
- J. Programación para la puesta en marcha de un montacargas de tres plantas.
- K. Control automático de la temperatura de un invernadero.

Completando la correspondiente memoria descriptiva:

- Listado de variables definitivas, según el autómatas usado.
- El esquema eléctrico destacado, según el PLC usado.
- La programación en un lenguaje reglamentario.
- Puesta en marcha y depuración de errores.

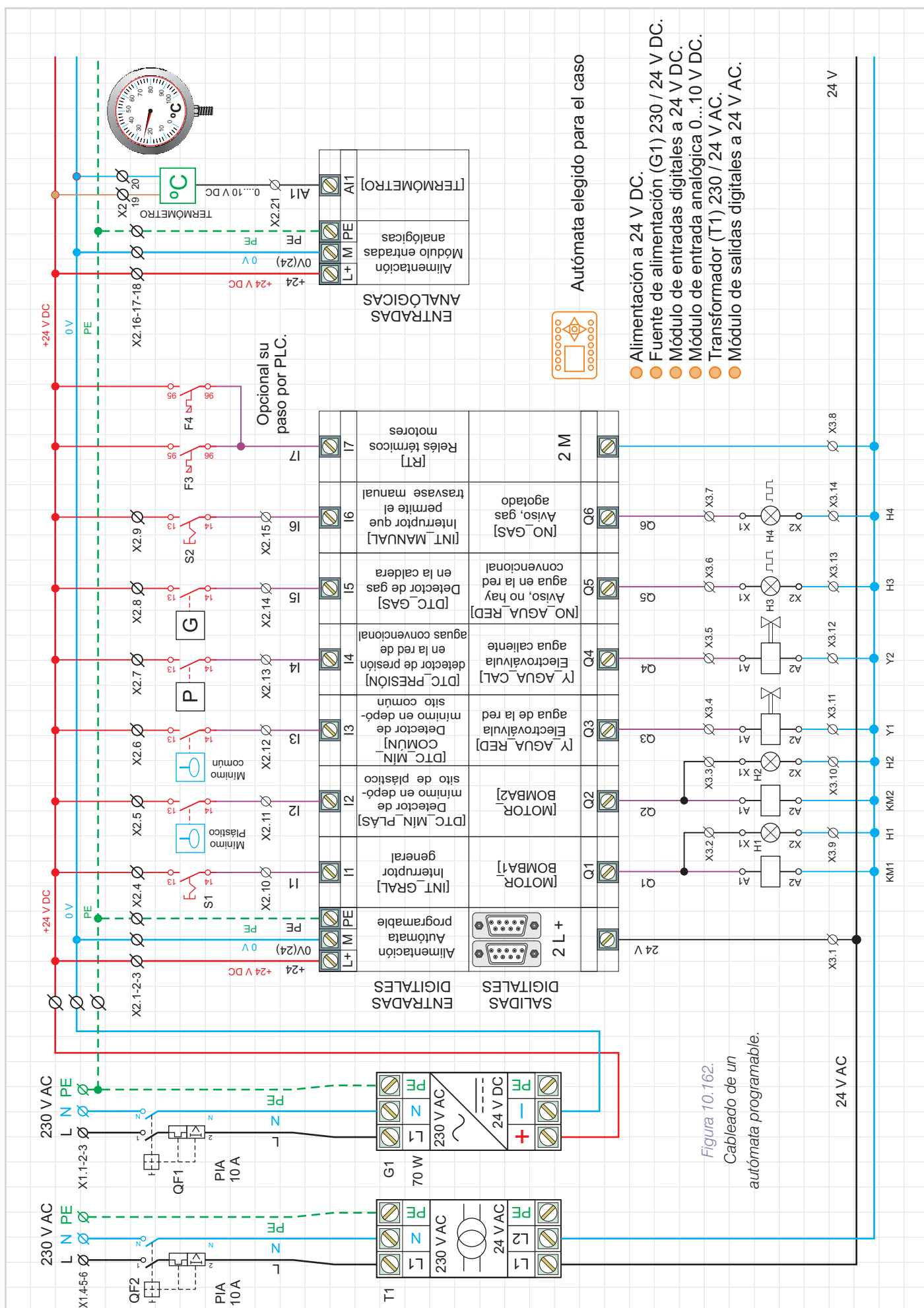


Figura 10.162.
Cableado de un
autómata programable.

