

HISTORIA DE LOS PLC's

En 1969 la División Hydramatic de la General Motors instaló el primer PLC para reemplazar los sistemas inflexibles cableados usados entonces en sus líneas de producción.

Ya en 1971, los PLCs se extendían a otras industrias y, en los ochentas, ya los componentes electrónicos permitieron un conjunto de operaciones en 16 bits,- comparados con los 4 de los 70s -, en un pequeño volumen, lo que los popularizó en todo el mundo.

En los 90s, aparecieron los microprocesadores de 32 bits con posibilidad de operaciones matemáticas complejas, y de comunicaciones entre PLCs de diferentes marcas y PCs, los que abrieron la posibilidad de fábricas completamente automatizadas y con comunicación a la Gerencia en "tiempo real".

Un autómatas programable suele emplearse en procesos industriales que tengan una o varias de las siguientes necesidades:

- Espacio reducido.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos complejos y amplios.
- Chequeo de programación centralizada de las partes del proceso.

Aplicaciones generales:

- Maniobra de máquinas.
- Maniobra de instalaciones.
- Señalización y control.

Entre las ventajas tenemos:

- Menor tiempo de elaboración de proyectos.
- Posibilidad de añadir modificaciones sin costo añadido en otros componentes.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra.
- Mantenimiento económico.

- Posibilidad de gobernar varias máquinas con el mismo autómeta.
- Menor tiempo de puesta en funcionamiento.
- Si el autómeta queda pequeño para el proceso industrial puede seguir siendo de utilidad en otras máquinas o sistemas de producción.

Y entre los inconvenientes:

- Adiestramiento de técnicos.
- Costo.

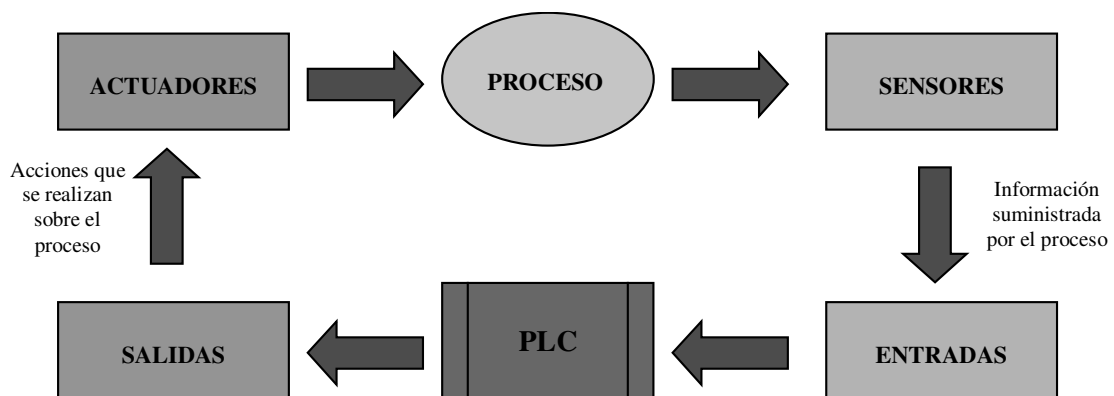
Adaptación al Medio

- § Inmunidad al ruido eléctrico
- § Rigidez dieléctrica
- § Temperatura
- § Humedad
- § Vibraciones
- § Shokc (golpes)
- § Radiofrecuencia
- § Antiexplosivo

Redundancia

- § CPU
- § Fuente
- § Entradas y Salidas

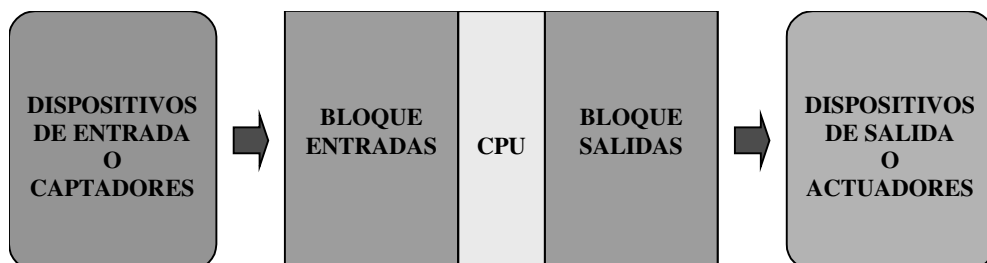
Un PLC o Automata Programable posee las herramientas necesarias, tanto de software como de hardware, para controlar dispositivos externos, recibir señales de sensores y tomar decisiones de acuerdo a un programa que el usuario elabore según el esquema del proceso a controlar



Arquitectura de un Automata Programable

Bloques principales del PLC

- CPU
- Bloque de entrada
- Bloque de salida



CLASIFICACIÓN DE PLC

Debido a la gran variedad de tipos distintos de PLC, tanto en sus funciones, en su capacidad, en el número de I/O, en su tamaño de memoria, en su aspecto físico y otros, es que es posible clasificar los distintos tipos en varias categorías.

PLC tipo Nano:

Generalmente PLC de tipo compacto (Fuente, CPU e I/O integradas) que puede manejar un conjunto reducido de I/O, generalmente en un número inferior a 100. Permiten manejar entradas y salidas digitales y algunos módulos especiales.

PLC tipo Compactos:

Estos PLC tienen incorporado la Fuente de Alimentación, su CPU y módulos de I/O en un solo módulo principal y permiten manejar desde unas pocas I/O hasta varios cientos (alrededor de 500 I/O) , su tamaño es superior a los Nano PLC y soportan una gran variedad de módulos especiales, tales como:

- § entradas y salidas análogas
- § módulos contadores rápidos
- § módulos de comunicaciones
- § interfaces de operador
- § expansiones de i/o

PLC tipo Modular:

Estos PLC se componen de un conjunto de elementos que conforman el controlador final, estos son:

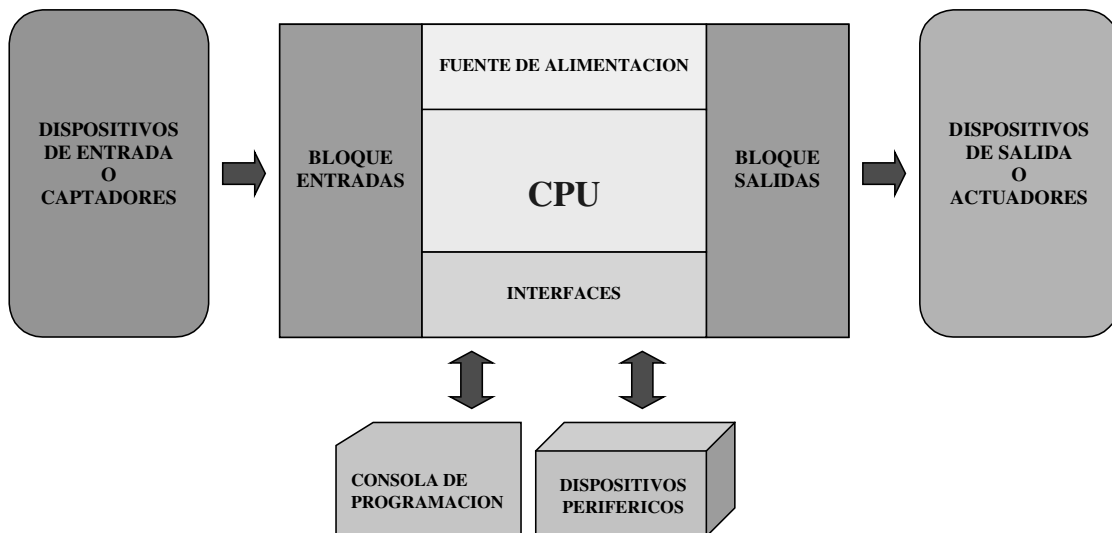
- § Rack
- § Fuente de Alimentación
- § CPU
- § Módulos de I/O
- § Comunicaciones.
- § Contaje rápido.

Funciones especiales

De estos tipos existen desde los denominados MicroPLC que soportan gran cantidad de I/O, hasta los PLC de grandes prestaciones que permiten manejar miles de I/O.

Bloques necesarios para el funcionamiento del PLC

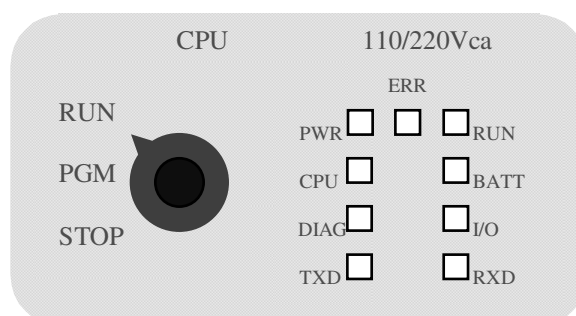
- Fuente de alimentación
- Consola de programación
- Periféricos
- Interfaces



CPU

La Unidad Central de Procesos es el cerebro del sistema. En ella se ejecuta el programa de control del proceso, el cual fue cargado por medio de la consola de programación, lee las entradas. Y posteriormente procesa esta información para enviar respuestas al módulo de salidas. En su memoria se encuentra residente el programa destinado a controlar el proceso.

Indicadores y selectores de modo de operación de la CPU



Selección del modo de operación

RUN: En esta posición, el PLC empieza a ejecutar el programa

STOP: Esta posición detiene el proceso que se esté ejecutando.

PROGRAM: En esta posición, el PLC permite ser programado desde algún dispositivo externo

Funciones de las Indicaciones Lumínicas

PWR: Muestra el estado de la fuente de alimentación.

RUN: Indica que la CPU del PLC se encuentra en modo RUN

CPU: La CPU hace diagnóstico de su estado de funcionamiento en forma automática.

ERR: Señal que hubo un error durante la ejecución del programa. El error puede ser de programa o de funcionamiento del hardware del PLC.

BATT: Indica el estado actual de la batería de respaldo de la fuente de alimentación.

I/O: Esta luz sirve para indicar el estado de los módulos de entrada y salida.

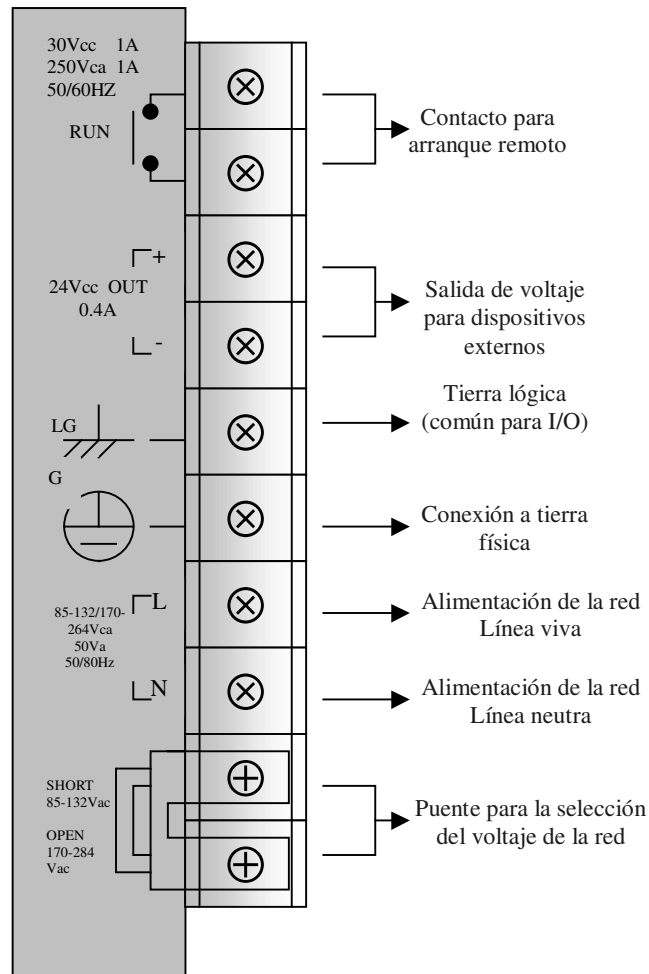
DIAG: Los autómatas pueden tener un sistema de diagnóstico propio.

TXD: Es el indicador de transmisión de datos por los puertos de comunicaciones.

RDX: Es el indicador de recepción de datos por los puertos de comunicaciones.

Fuente de alimentación

A partir de una tensión exterior proporciona las tensiones necesarias para el funcionamiento de los distintos circuitos electrónicos del autómata, además posee una batería para mantener el programa y algunos datos en la memoria si hubiera un corte de la tensión exterior.



Voltaje de entrada

Oscila entre:
100VCA - 240VCA

Voltaje de salida

Oscila entre:
12VDC - 24VDC

Corriente de salida

1A - 3A

Frecuencia de operación

50Hz ó 60Hz

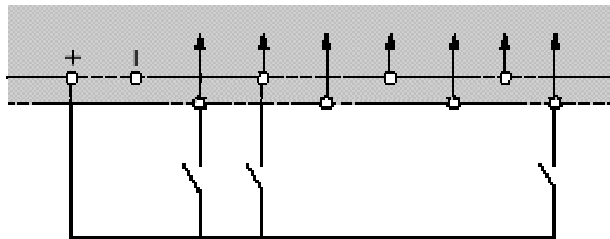
Protecciones

Sobrecorrientes y sobrevoltajes

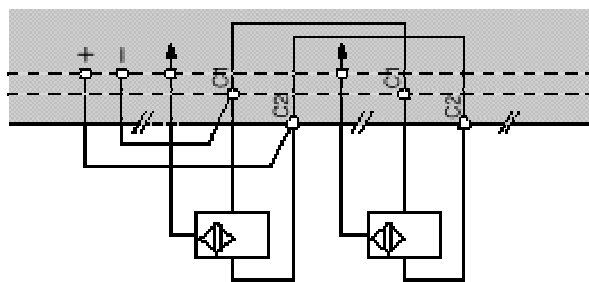
Bloque de entradas

Adapta y codifica de forma comprensible para la CPU las señales procedentes de los dispositivos de entrada o captadores.

- **Captadores Pasivos** son aquellos que cambian su estado lógico, activado - no activado, por medio de una acción mecánica. Estos son los Interruptores, pulsadores, finales de carrera, etc.



- **Captadores Activos** son dispositivos electrónicos que necesitan ser alimentados por una tensión para que varíen su estado lógico. Este es el caso de los diferentes tipos de detectores (Inductivos, Capacitivos, Fotoeléctricos). Muchos de estos aparatos pueden ser alimentados por la propia fuente de alimentación del autómata.

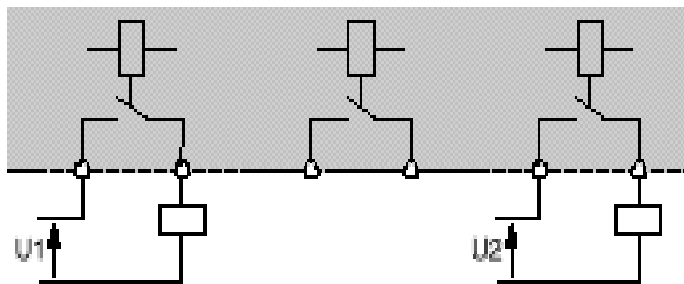


Se puede utilizar como captadores contactos eléctricamente abiertos o eléctricamente cerrados dependiendo de su función en el circuito.

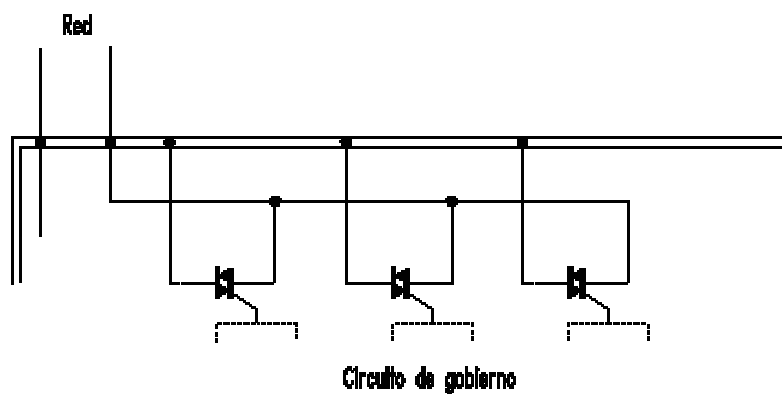
Bloque de salidas

Decodifica las señales procedentes de la CPU, las amplifica y las envía a los dispositivos de salida o actuadores, como lámparas, relés, contactores, arrancadores, electroválvulas, etc.

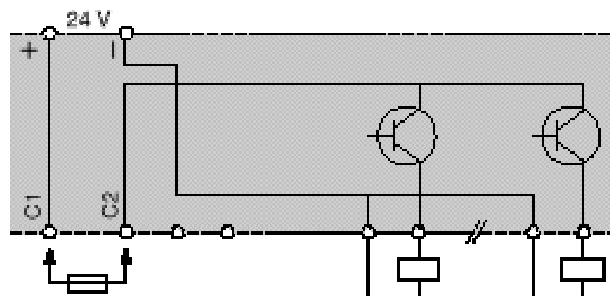
Módulos de salidas a Relés



Módulos de salidas a Triacs



Módulos de salidas a Transistores a colector abierto



Módulos de E/S analógicos

Módulos Especiales

- § BCD Entradas y Salidas
- § Entradas de Termocuplas
- § Entradas de Termorresistencias (PT-100)
- § Salidas a Display
- § Memoria de Datos

Módulos Inteligentes

- § Control de Motores a pasos
- § Control PID
- § Comunicación
- § Lector Óptico
- § Generación de frecuencias
- § Contadores de pulsos rápidos
- § Medidores de frecuencias y/o ancho de pulsos
- § Remotos

Densidad de Modulo

Esta dado por el numero de E/S de cada modulo

Pueden ser de: 4, 8, 16 o 32

Un modulo de 16 E/S se puede dividir en 2 de 8

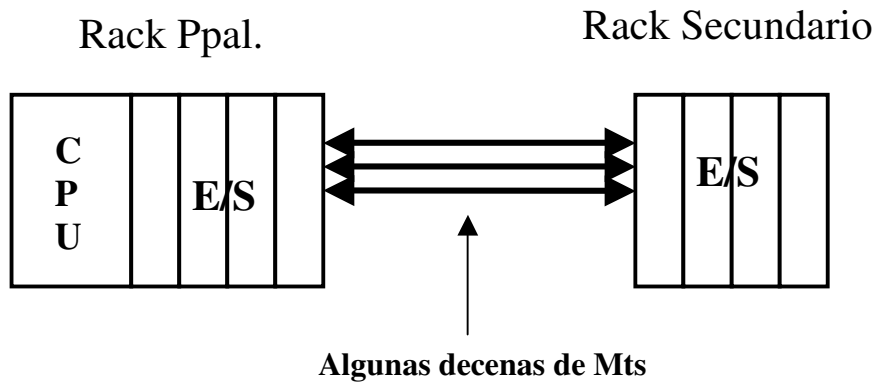
Cada grupo dispone de un solo borne de referencia o común.

Las salidas tienen un fusible de protección

Las borneras son móviles, se desmontan sin tener que desconectar los cables evitando cometer errores

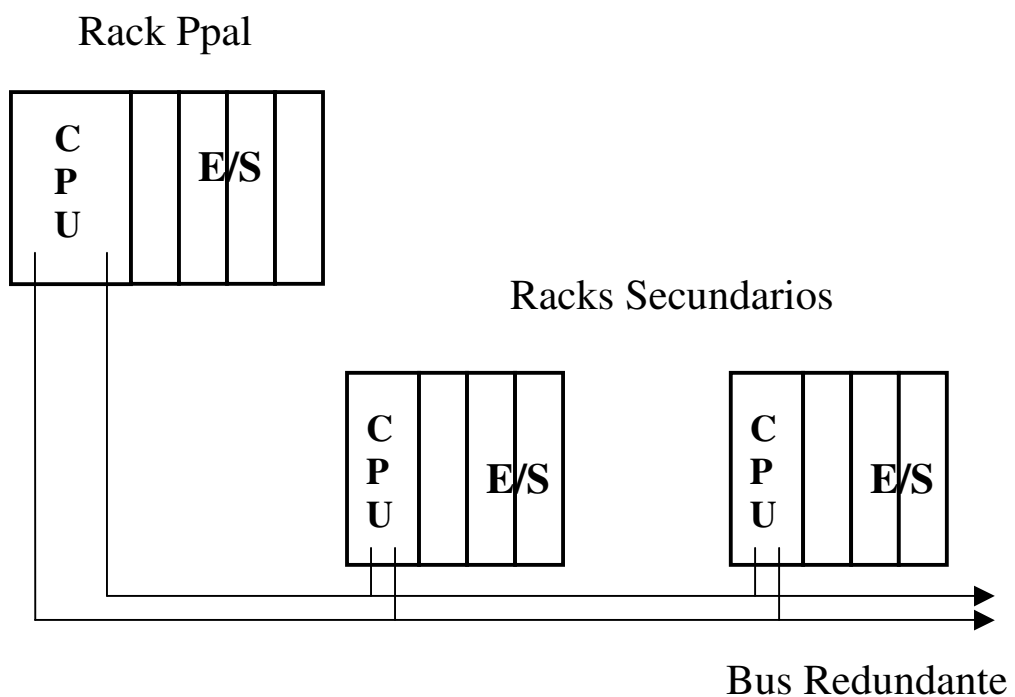
Expansión Local

Se expanden las E/S por el Bus Paralelo



Expansión Remota

Se utilizan procesadores de comunicaciones especiales, uno en cada chasis de expansión



Medio Físico: Coaxil, Fibra Óptica o UTP

Los racks secundarios pueden estar a 5 Kms.

Esto implica un ahorro substancial en el cableado.

Si tengo 200 E/S a 50 mts. => 10 Kms. de cable

Con un rack secundario solo necesito 50 mts de cable coaxil.

Todas las señales viajan por un solo cable => más vulnerable

Se emplean cableados redundantes por distintos caminos

Consola de programación

La consola de programación es la que permite comunicar al operario con el sistema, permitiendo escribir y poner a punto programas. Algunas permiten ensayos de simulación y puesta en servicio de los mismos.

Las consolas pueden estar constituidas por un dispositivo de presentación visual (display) o bien un ordenador personal (PC) que soporte un software especialmente diseñado para resolver los problemas de programación y control.

Las funciones básicas de éste son las siguientes:

1. Transferencia y modificación de programas.
2. Verificación de la programación.
3. Información del funcionamiento de los procesos.



Terminal de programación portátil



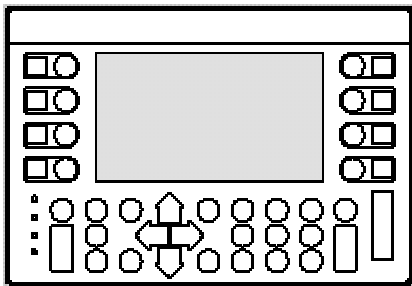
Terminal de programación compatible PC

PERIFÉRICOS

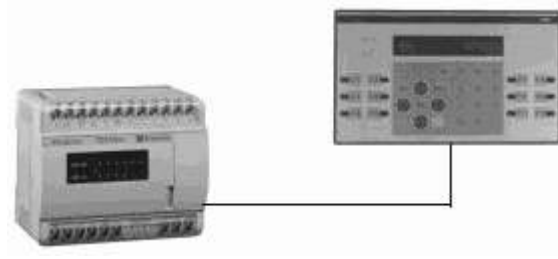
Los periféricos no intervienen directamente en el funcionamiento del autómata, pero sin embargo facilitan la labor del operario.

Los más utilizados son:

- Grabadoras a cassettes.
- Impresoras.
- Cartuchos de memoria EEPROM.
- Visualizadores y paneles de operación OP



Panel de Operación

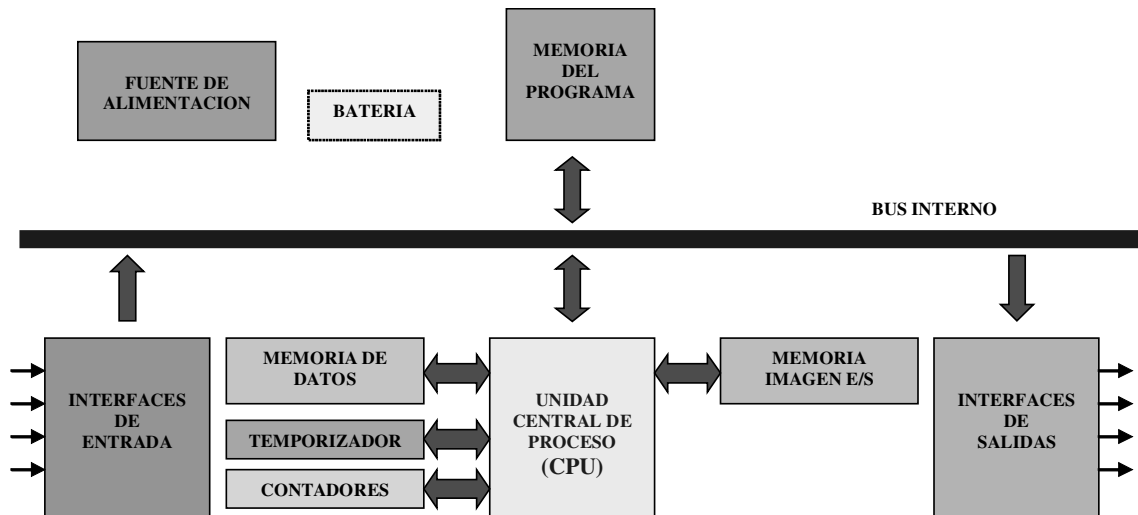


Conexión de un visualizador a un autómata

Arquitectura interna del PLC

Esta constituida por:

- Fuente de alimentación
- Unidad central de proceso
- Memoria:
 - a- Memorias internas
 - b- Memorias de programa
- Interfaces de entrada/salida
- Buses de comunicación
- Batería de respaldo



Conexión mediante buses

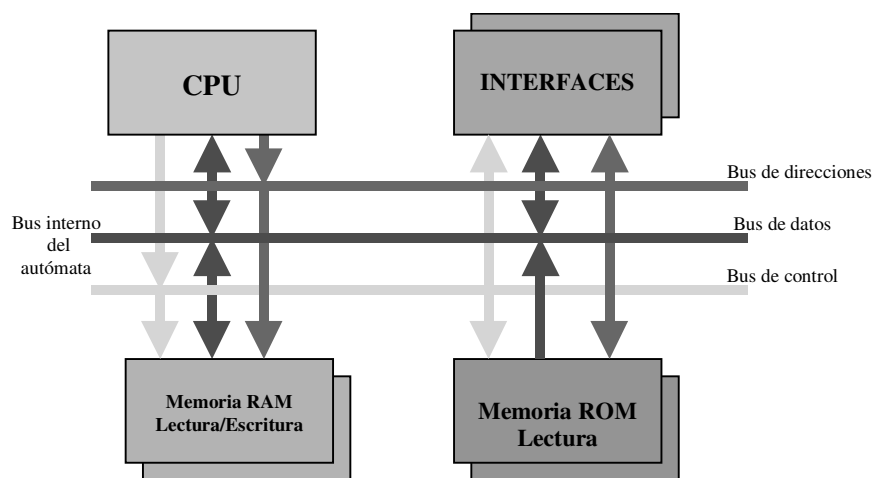
Si el PLC es de tipo modular, los módulos se comunican internamente a través de buses ubicados en el fondo del dispositivo o "rack" donde se ensambla la arquitectura deseada.

Básicamente existen tres tipos de buses

Bus de datos: Es el bus encargado de transportar la información que hace referencia a los datos propiamente dicho, tales como entradas y salidas.

Bus de direcciones: Contiene la información del dispositivo que es afectado por los datos que actualmente viajen por el bus de datos. Esto es necesario ya que el bus de datos es el mismo para todos los dispositivos, pero no todos ellos deben tener en cuenta a la vez la misma información.

Bus de control: Es el bus por donde viaja la información que indica al dispositivo seleccionado con el bus de direcciones, lo que debe hacer con los datos que viajan actualmente por el bus de datos. Por ejemplo, mediante el bus de control se indica si los datos son de entrada o de salida.



Arquitectura de conexión mediante buses

Memoria

Es cualquier tipo de dispositivo que permita almacenar información en forma de bits (unos y ceros), los cuales pueden ser leídos posición a posición (bit a bit), o por bloques de 8 posiciones (byte) o dieciséis posiciones (word).

Clasificación

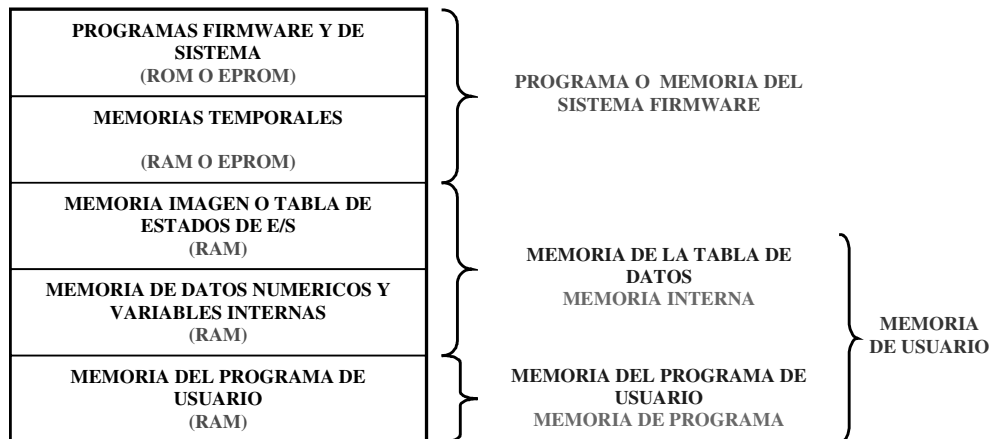
Existen dos tipos fundamentales de memorias fabricadas con semiconductores, estas son:

RAM (Random Access Memory)

ROM (Read Only Memory)

Tipo de memoria	Sistema de programación	Sistema de borrado	Ante el corte de tensión la memoria
RAM (o memoria de lectura – escritura)	Eléctrica	Eléctrica	Se pierde, es volátil
ROM (o memoria de solo lectura)	Durante su proceso de fabricación	Es imposible su borrado	Se mantiene
PROM (o memoria programable)	Eléctrica	Es imposible su borrado	Se mantiene
EPROM (o memoria modificable)	Eléctrica	Por rayos UV	Se mantiene
EEPROM (o memoria modificable)	Eléctrica	Eléctrica	Se mantiene

Utilización de memorias

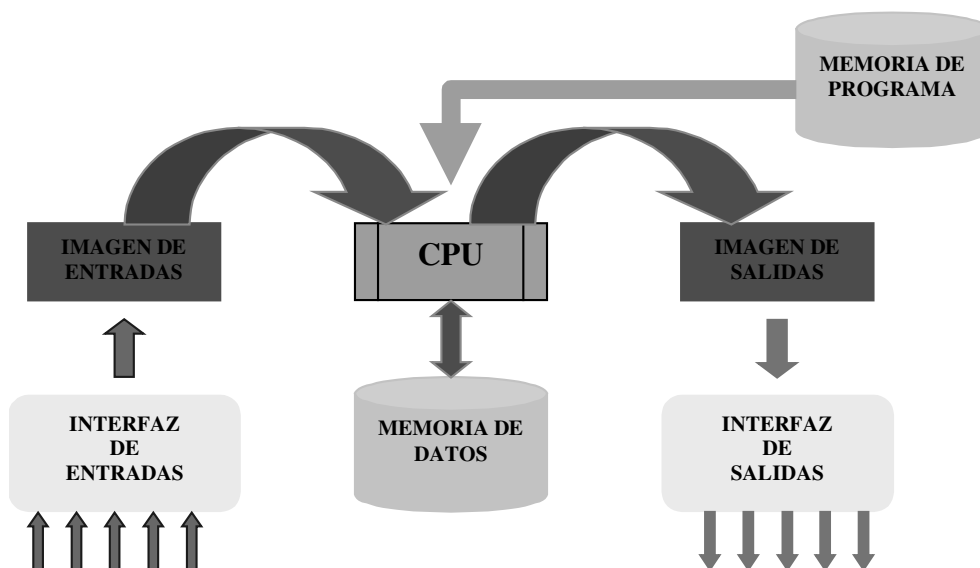


Ciclo de tratamiento de las señales de entrada/salida a través de las memorias imagen:

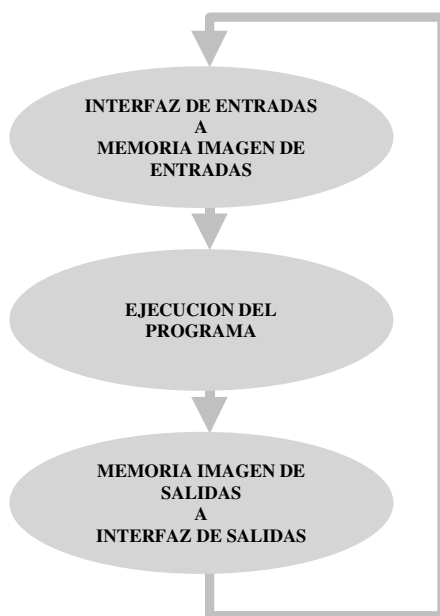
1- Antes de la ejecución del programa de usuario, la CPU consulta los estados de las *entradas físicas* y carga con ellos la *memoria imagen de entradas*.

2- Durante la ejecución del programa de usuario, la CPU realiza los cálculos a partir de los datos de la memoria imagen y del estado de los temporizadores, contadores y relés internos. El resultado de estos cálculos queda depositado en la memoria *imagen de salidas*.

3- Finalizada la ejecución, la CPU transfiere a las *interfaces de salida* los estados de las señales contenidos en la *memoria imagen de salidas*, quedando el sistema preparado para comenzar un nuevo ciclo.



Ciclo real



Otras variables que se almacenan en la memoria interna son:

Relés internos

Ocupan posiciones RAM de 1 bit, y son utilizados como área de datos temporales, como salida de resultados de operaciones intermedias, y para controlar otros bits o registros, temporizadores y contadores.

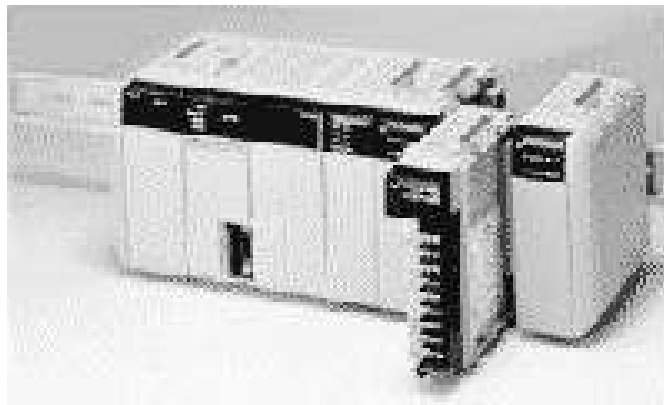
Relés auxiliares/especiales

Se guardan en posiciones de 1 bit, y mantienen información sobre señales necesarias para el sistema, como relojes, bits de control, flags de estados de la CPU, e información sobre el autómata (Run, Stop, Halt, errores, etc.). Estos relés pueden consultarse y utilizarse desde el programa usuario.

Area de temporizadores y contadores

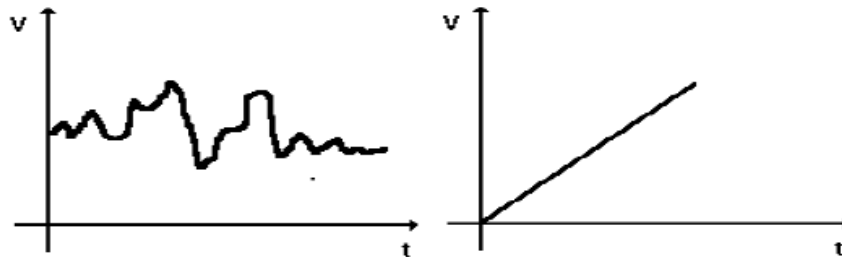
Ocupa posiciones de 16 bits o más bits, capaces de almacenar los valores de preselección y estado actualizados de estos elementos.

Interfaces de entrada y salida

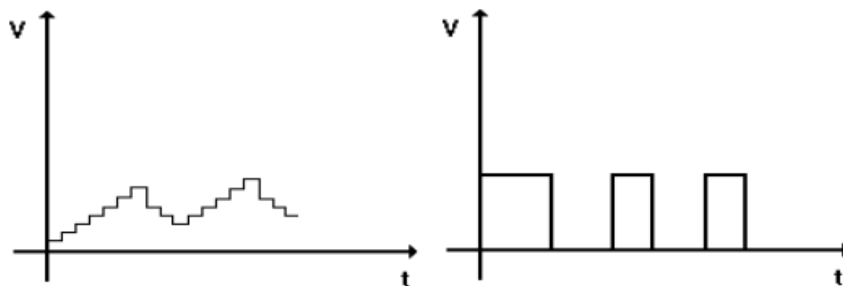


Tipos de Señales

Señales Análogas



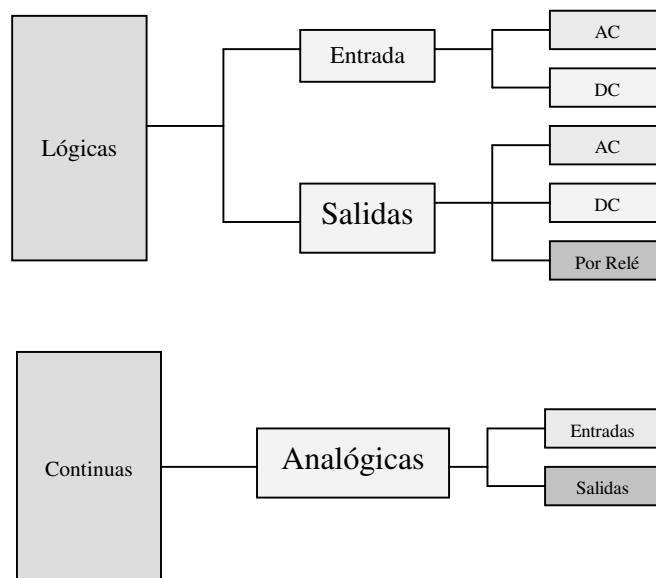
Señales Discretas



Las señales analógicas presentan variaciones de nivel en forma continua, motivo por el cual pueden presentar infinitudes de valores

Las señales del tipo lógicos o Discretas, solamente permiten dos estados, activo o inactivo

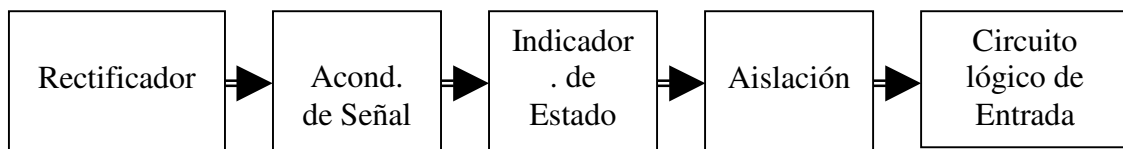
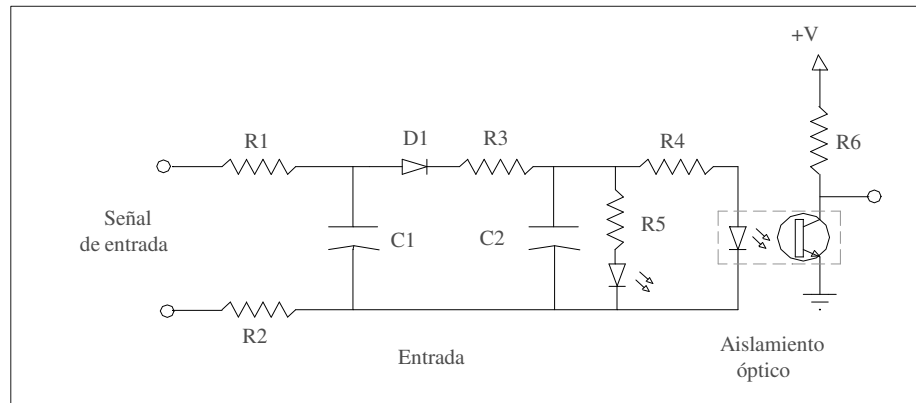
Clasificación de los diferentes tipos de interfaces de entrada y salida utilizados por los PLC.



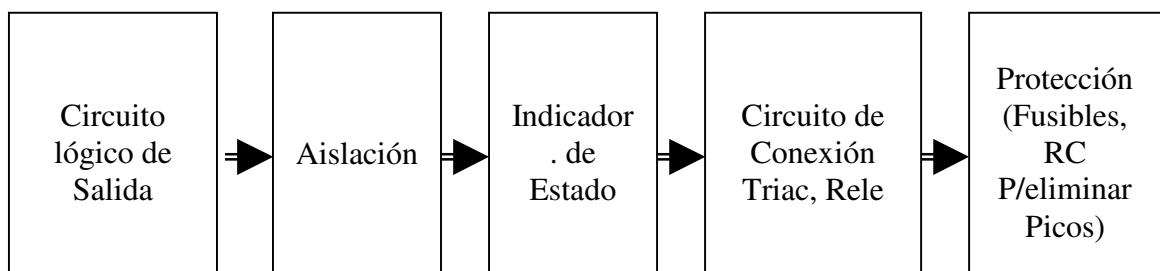
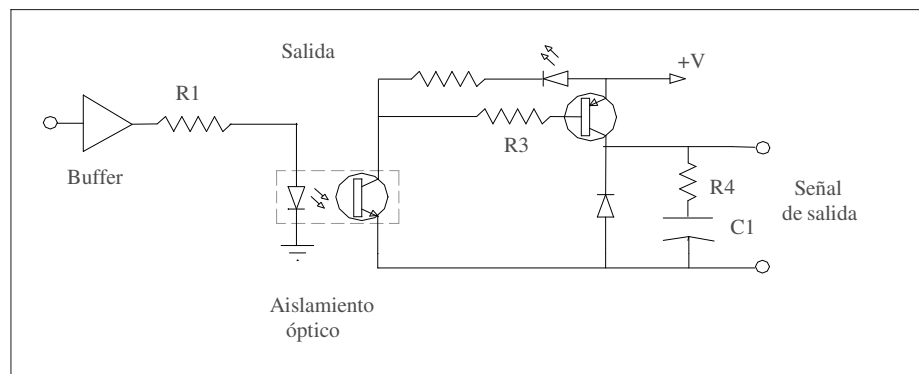
Una vez obtenidas la cantidad necesaria de Entradas y de Salidas es muy recomendable reservar espacio para futuras ampliaciones (entre un 10 y un 20 %).

Circuito típico de aislamiento utilizado para entradas y salidas de un PLC, sin importar el tipo de señal (DC o AC) de entrada

Entrada

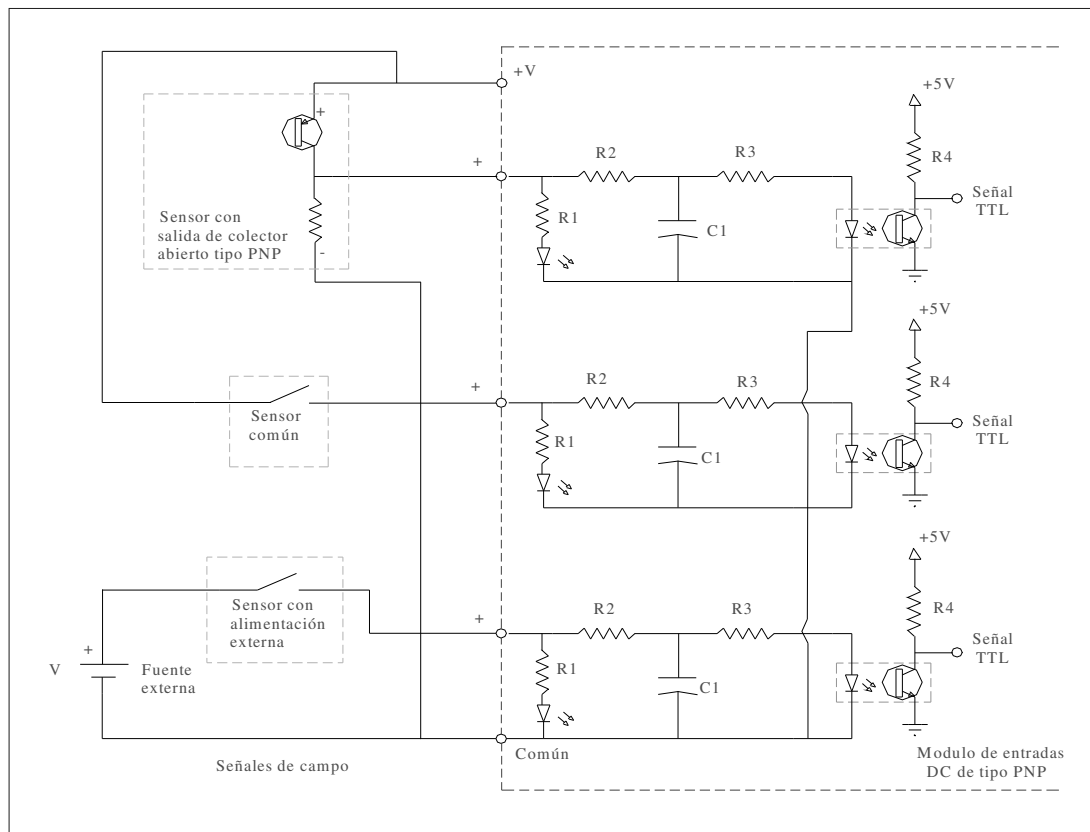


Salida



Entrada lógica DC

Las entradas de este tipo pueden ser PNP o NPN, dependiendo de la configuración que se deba utilizar para la conexión de los sensores de entrada. Las de tipo PNP utilizan la *lógica positiva*, es decir, el terminal común de todos los sensores e interruptores de entrada es el positivo de la fuente de alimentación DC.



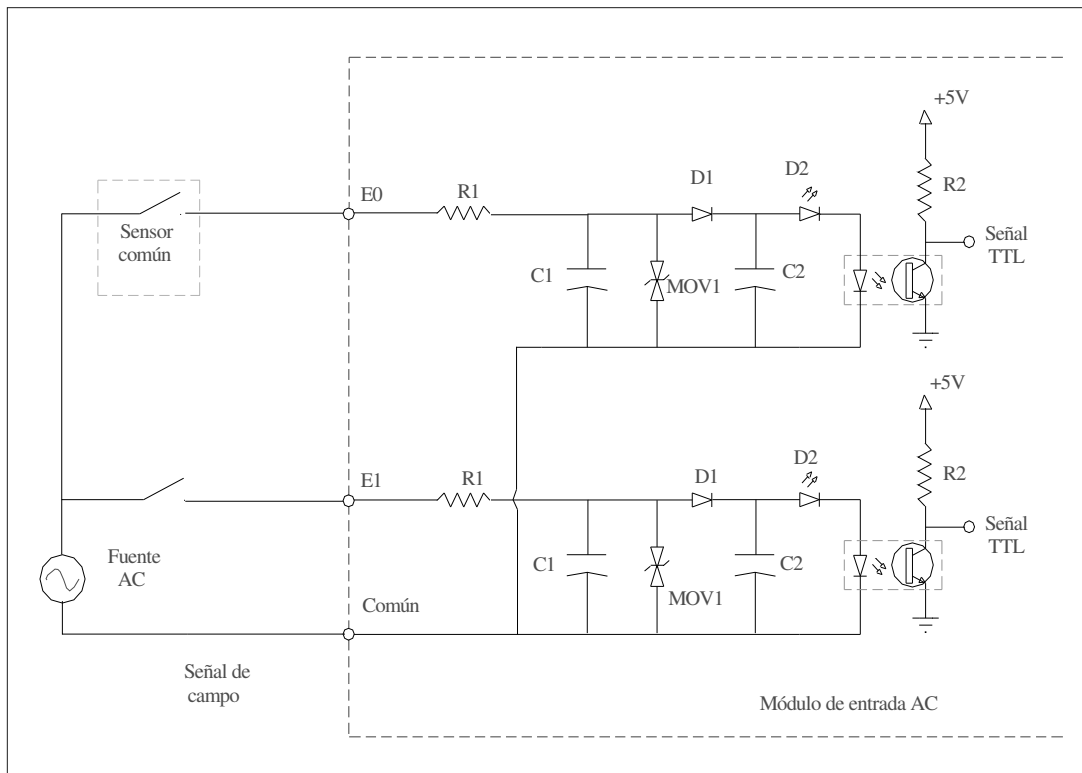
Por otra parte, las entradas DC de tipo NPN utilizan *lógica negativa*, siendo el terminal común entre los sensores, el borne negativo de la fuente de alimentación

Características que se deben tener en cuenta en el momento de adquirir o de utilizar un módulo de entradas de tipo DC.

<i>Características técnicas</i>	
Aislamiento eléctrico entre canales	1500 a 5000VAC
Números de entradas por módulos	4, 8, 16 ó 32
Conexión externa	Lógica positiva (PNP) o Lógica negativa (NPN)
Tensión de entrada	12, 24, 48, 120VDC
Margen de entrada para nivel 1	> 65 % de la tensión nominal
Margen de entrada para nivel 0	< 25 % de la tensión nominal
Corriente en estado 1	2 a 10mA
Corriente en estado 0	0 a 3mA
Impedancia de entrada	800 Ω a 12K Ω

Entrada lógica AC

Este tipo de entradas puede recibir voltajes de corriente alterna, motivo por el cual no requieren una polaridad definida, a diferencia de las entradas de tipo DC. Se caracterizan por sus amplios rangos de voltajes de funcionamiento (hay entradas que funcionan correctamente en un rango de 100 a 220VAC).

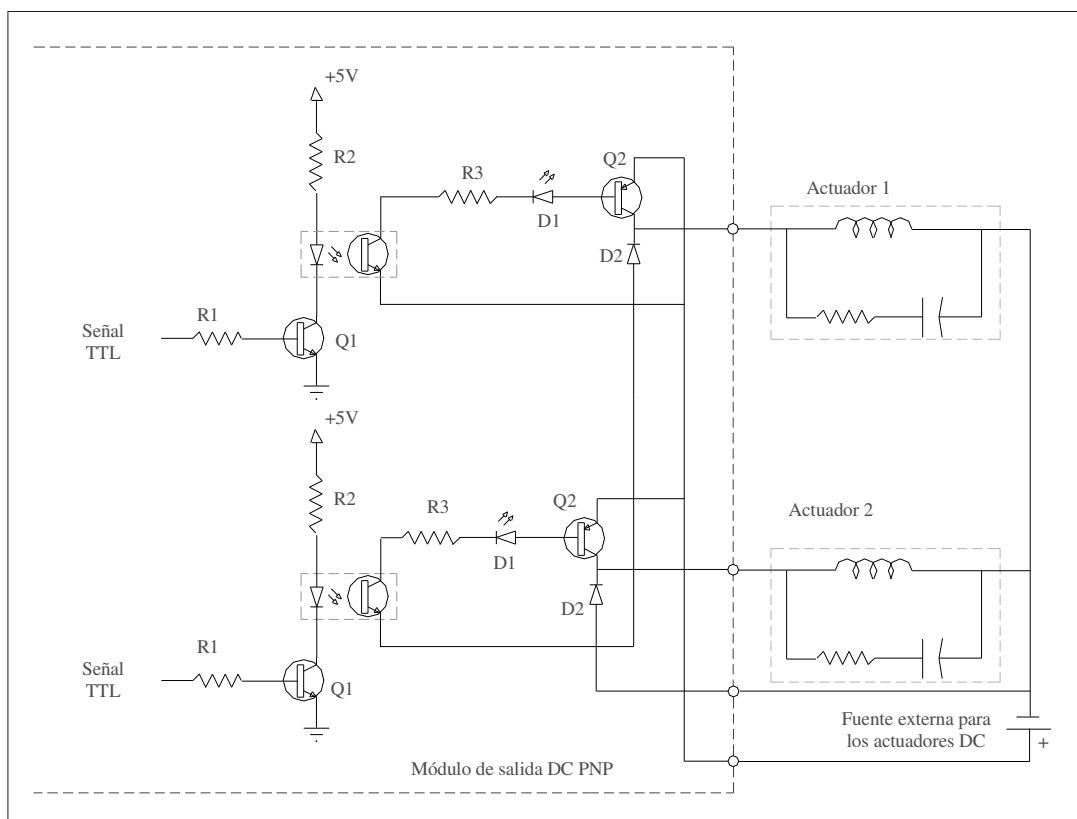


Características que se deben tener en cuenta en el momento de adquirir o de utilizar un módulo de entradas de tipo AC.

<i>Características técnicas</i>	
Aislamiento entre canales	2500 a 5000VAC
Números de entradas por módulos	4, 8, 16 ó 32
Tensión de entrada	24, 48, 120, 220VAC
Margen de entrada para nivel 1	> 65 % de la tensión nominal
Margen de entrada para nivel 0	< 20 % de la tensión nominal
Tiempo de respuesta	5 a 20ms
Corriente en estado 1	8 a 10mA
Corriente en estado 0	3 a 5mA
Impedancia de entrada	1 a 10K Ω

Salidas lógicas de tipo DC

Al igual que las entradas de este tipo, las salidas lógicas DC pueden ser del tipo PNP o NPN dependiendo de la forma en que se deben conectar los circuitos de control que se vayan a utilizar. Las de tipo PNP utilizan lógica positiva, lo que significa que las salidas activas del módulo suministran la polaridad positiva de la fuente de alimentación que se este usando.



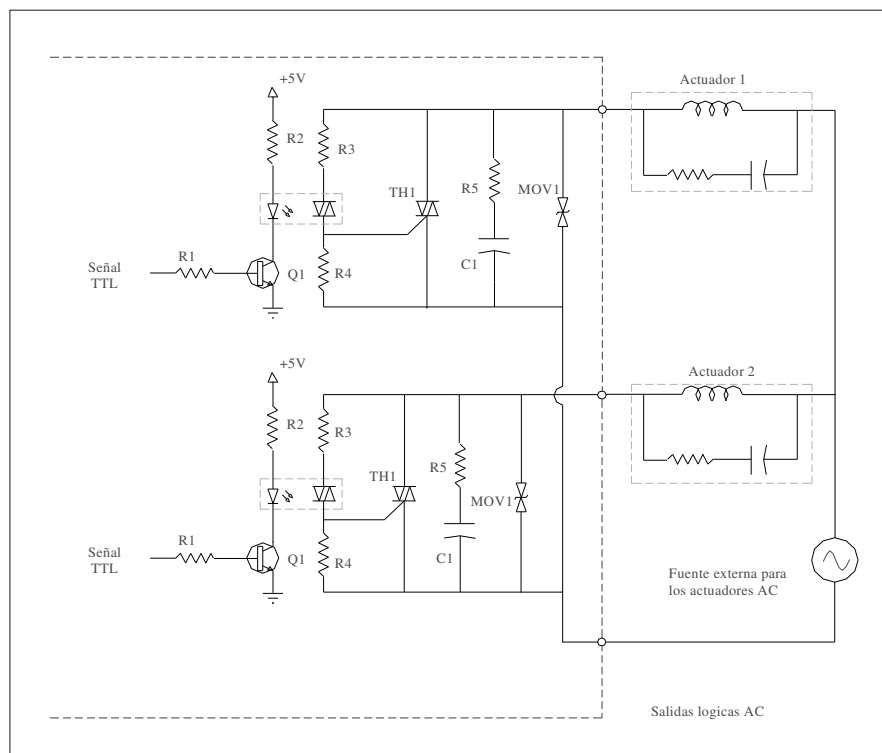
Características que se deben tener en cuenta en el momento de adquirir o de utilizar un módulo de salidas de tipo DC.

<i>Características técnicas</i>	
Aislamiento entre canales	1500VAC
Números de salidas por módulos	4, 8, 16 ó 32
Tensión de salida	5 – 30VDC
Corriente de salida	100mA a 1A
Caída de tensión	0,2V a 1,5V
Tiempo de respuesta	100 a 500µs
Frecuencia máxima de trabajo	10 a 100Hz
Temperatura de trabajo	5 a 55 °C

Salidas lógicas de tipo AC

Las salidas lógicas de tipo AC manejan voltajes y corrientes superiores a las salidas de tipo DC. Como mencionamos anteriormente, el elemento semiconductor en este tipo de salidas suelen ser *un triac*, el cual permite el flujo de corriente AC entre dos de sus terminales, controlándola mediante otro terminal.

La conexión de los elementos externos de control a las salidas lógicas de tipo AC no tienen polaridad alguna.



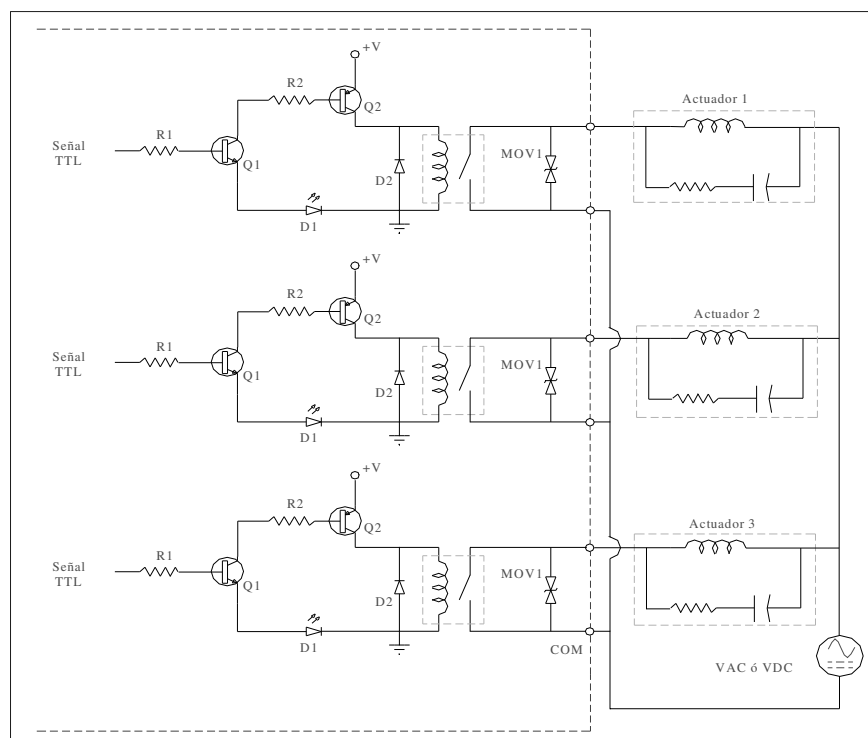
Características técnicas relacionadas con las salidas AC

<i>Características técnicas</i>	
Aislamiento entre canales	1500VAC
Números de salidas por módulos	4, 8 ó 16
Tensión de salida	Hasta 250VAC
Corriente de salida	1 a 2A
Intensidad transitoria	20A por 100ms
Caída de tensión	3V a 6V
Tiempo de respuesta	10 a 20ms
Frecuencia máxima de trabajo	20Hz
Temperatura de trabajo	5 a 55 °C

Salidas lógicas por relé

Este tipo de salidas puede ser utilizado para manejar cargas de corrientes AC y de corrientes DC, ambas con diversos niveles de voltaje. Las principales desventajas contra los demás tipos de salidas son el tiempo de respuesta y el tamaño físico que ocupan dentro del circuito electrónico.

Como ventaja, la separación eléctrica entre la carga y los circuitos de control es altamente confiable, ya que los contactos del relevo nada tiene que ver, con la alimentación o excitación de su bobina.

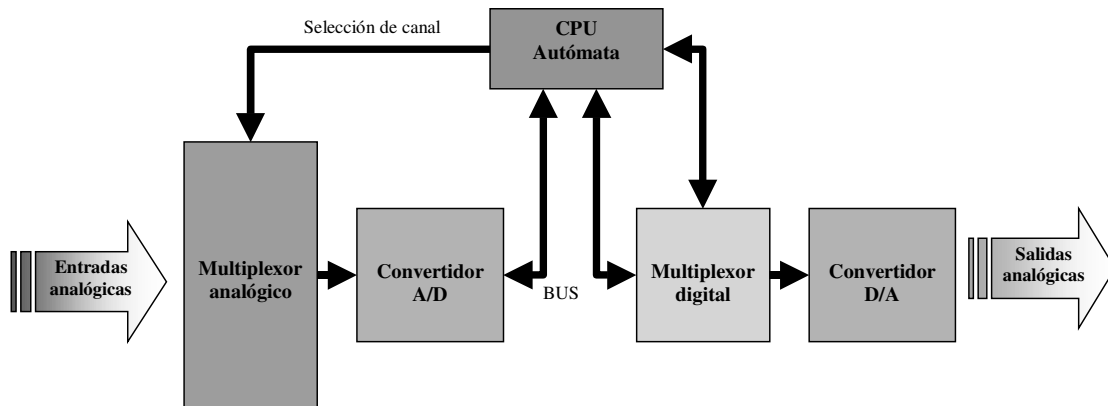


Características técnicas relacionadas con las salidas AC

<i>Características técnicas</i>	
Números de salidas por módulos	4 ó 8
Corriente de salida	1 a 2A
Corriente transitoria	10A durante 40ms
Corriente residual	0V
Caída de tensión	0V
Tiempo de respuesta	Típico 20ms
Frecuencia máxima de trabajo	1 a 5Hz
Temperatura de trabajo	5 a 55 °C

Entradas y salidas analógicas

Secuencia recorrida por un autómata durante la captura, procesamiento y generación de señales analógicas.



Por costo y espacio varias Entradas de utilizan un mismo circuito conversor. Por lo tanto se utiliza un circuito multiplexor.

Los niveles de voltaje y corriente estándares manipuladas por los autómatas son los siguientes:

Señales de voltajes	De 0 a 5V De 0 a 10V
Señales de corriente	De 0 a 20mA De 4 a 20mA

La conversión A/D y D/A puede ser tan precisa o tan inexacta como se desee dependiendo de la cantidad de bits utilizados durante el proceso de conversión. Por lo tanto la resolución es un parámetro importante a tener en cuenta con los módulos de entradas o salidas analógicas de un PLC.

Con 8 bits podemos representar $2^8 = 256$ valores

Con 10 bits podemos representar $2^{12} = 4096$ valores

Si tenemos una señal de salida de un transmisor de presión por ejemplo de 16 mA

Resolución $16 \text{ mA} / 256 = 0,0256 \text{ mA}$ por bit

$16 \text{ mA} / 4096 = 0,0039 \text{ mA}$ por bit

En la figura se observa una señal analógica y su representación luego de haber tomado varias muestras durante un ciclo de la misma. Como puede verse, si se toman más muestras, la señal se puede reproducir con mayor similitud a la original.



Entradas Analógicas

Las características técnicas relacionadas con las Entradas Analógicas son las siguientes

<i>Características técnicas</i>	
Cantidad de entradas	4 a 16
Márgenes de entrada	Desde $\pm 25\text{mV}$ hasta $\pm 10\text{V}$ y 0 a 20mA ó 4 a 20mA
Impedancia de entrada	Desde $50\text{K}\Omega$ hasta $10\text{M}\Omega$ (entradas de voltaje) Desde $20\text{K}\Omega$ hasta $50\text{M}\Omega$ (entradas de corriente)
Resolución	8, 10, 12 ó más bits
Principio de conversión	Por integración, por aproximaciones sucesivas o por conversión flash
Tiempo de conversión	$1\mu\text{s}$, $10\mu\text{s}$, $16\mu\text{s}$, etc.
Tensión de alimentación	24V, 48V
Consumo de corriente	0,1 a 0,3A

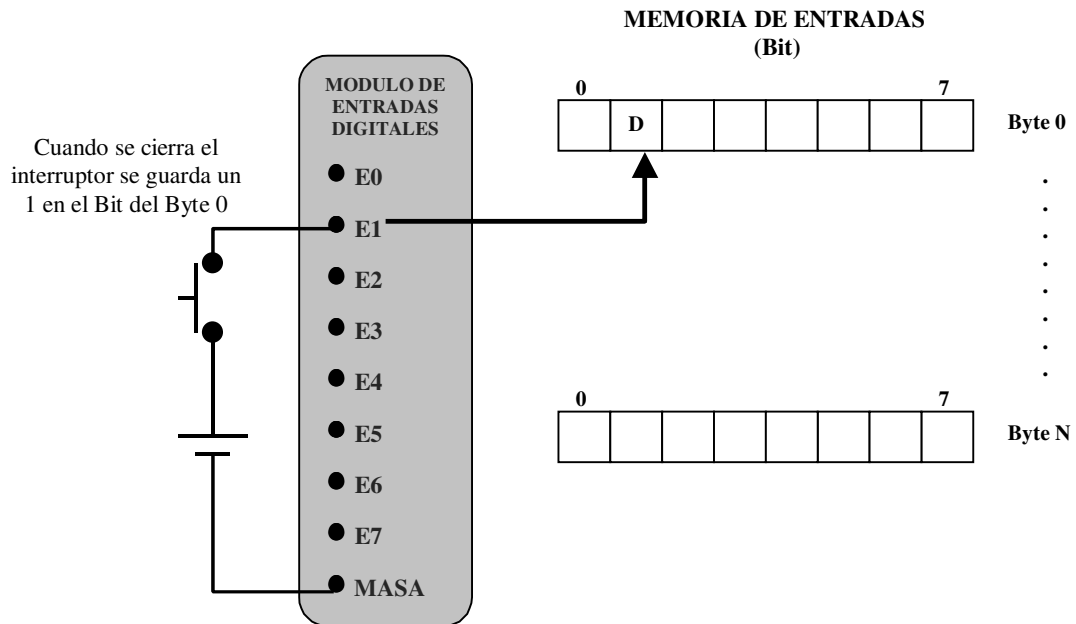
Salidas Analógicas

Las características técnicas relacionadas con las Salidas Analógicas son las siguientes

<i>Características técnicas</i>	
Cantidad de salidas	4 a 16
Márgenes de salida	Desde $\pm 5V$ hasta $\pm 10V$ y 0 a 20mA ó 4 a 20mA
Impedancia de salida	Desde 500Ω hasta $1K\Omega$ (entradas de voltaje) Desde 300Ω hasta 500Ω (entradas de corriente)
Resolución	8, 10, 12 ó más bits
Principio de conversión	Red R-2R, escalera binaria.
Tiempo de estabilización	10ms, 15ms, etc..
Tensión de alimentación	24V, 48V

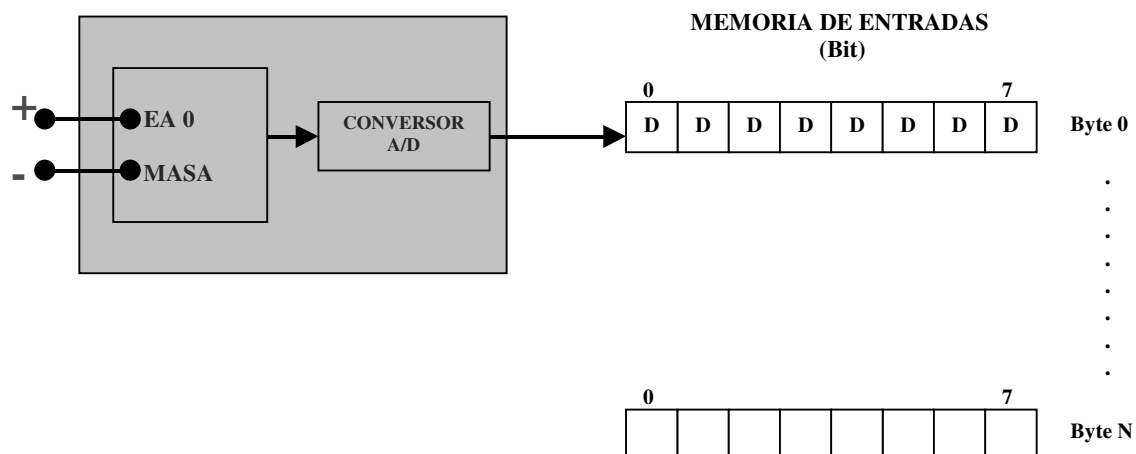
Existe una relación directa entre la posición física que ocupa la conexión de la variable en el módulo de entradas o salidas conectado al autómata y la posición de memoria donde se almacena el dato, por ejemplo:

▼ Carga en la memoria imagen de entradas de las entradas digitales



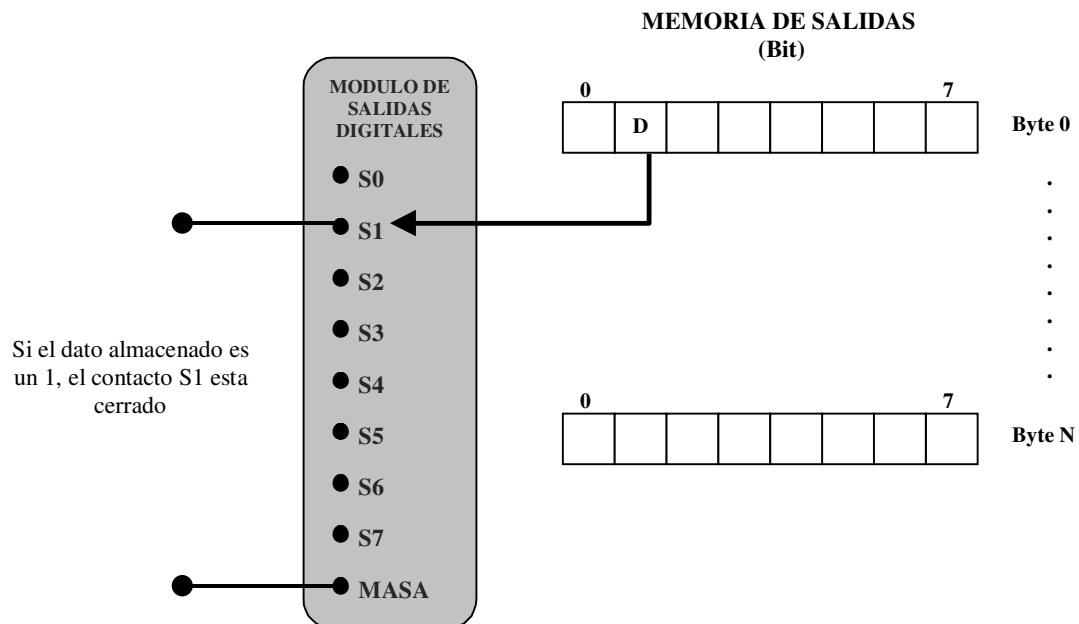
▼ Carga en la memoria imagen de entradas de las entradas analógicas

La entrada analógica cero almacena un valor en código binario natural en el byte cero de la memoria de entrada de byte. Previamente el convertor analógico/digital transforma a código binario natural la señal de tensión que estará comprendida en el margen 0-10V.

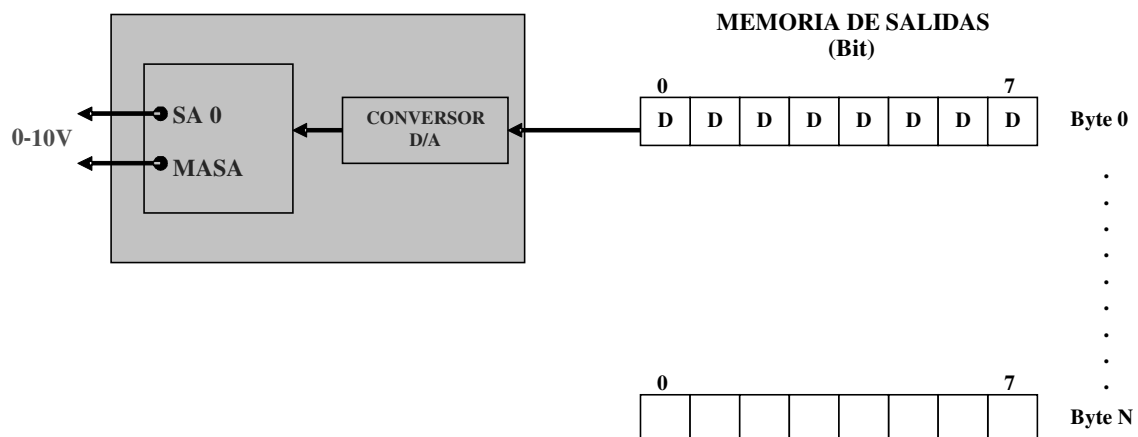


También existe una relación directa entre la memoria dedicada a la variable de salida y la posición de las conexiones asociadas a ella con los módulos de salida.

∇ **Transferencia de los datos contenidos en la memoria de salida a las salidas digitales**



∇ **Transferencia de los datos contenidos en la memoria de salida a las salidas analógicas**



CICLO DE FUNCIONAMIENTO DEL PLC

A nivel de software ejecuta una serie de rutinas de forma secuencial

- De auto diagnóstico e inicialización
- De procesamiento o ejecución del programa, esta última de manera cíclica.

Etapa de auto chequeo e inicialización

Al encender un PLC se ejecuta inicialmente una rutina que verifica ciertos elementos antes de comenzar a ejecutar el programa que haya elaborado el usuario.

Las rutinas de inicio y de programa de un PLC típico son:

1. Verificación del hardware

- Mal estado de la Fuente de Alimentación, CPU u otros Módulos y indicación de error existente en el panel frontal
- Estado de la batería de respaldo
- Integridad del programa almacenado en la memoria del PLC.

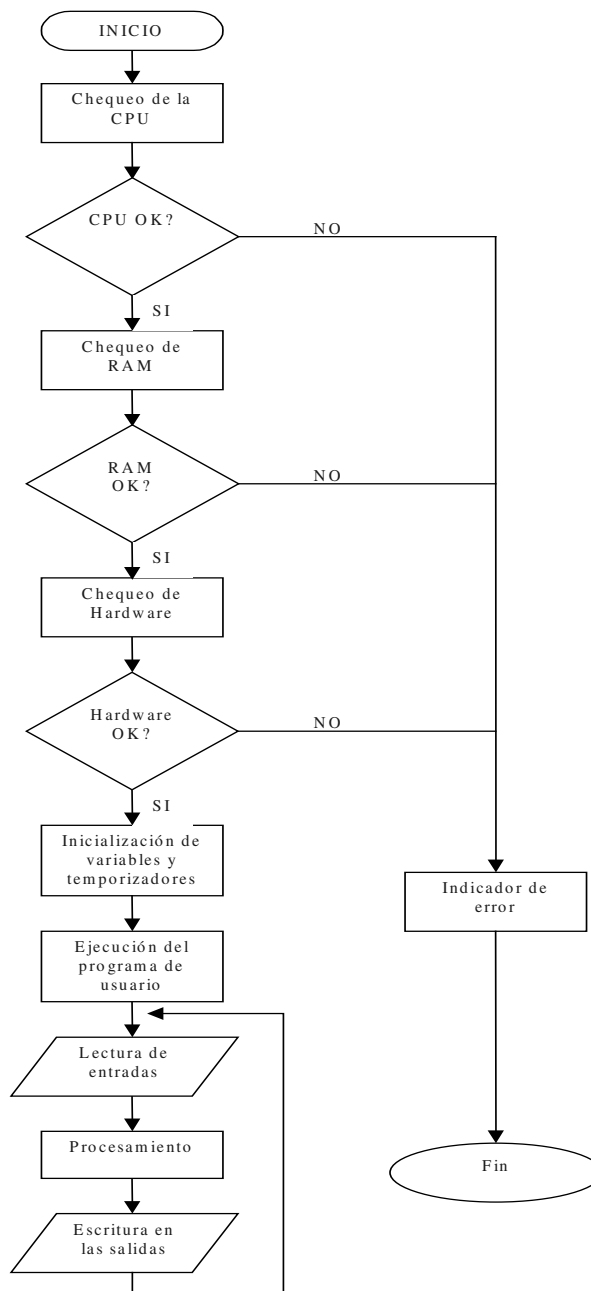
2. Inicialización de las imágenes de entradas y salidas

Ya que estas son posiciones de memoria, antes de actualizar el estado de las entradas y salidas, deben ponerse en "0" cada una de dichas posiciones.

3. Inicialización de posiciones de memoria

Establecer condiciones iniciales de posiciones de memoria. Poner en cero ("0") todos los contadores y temporizadores del sistema.

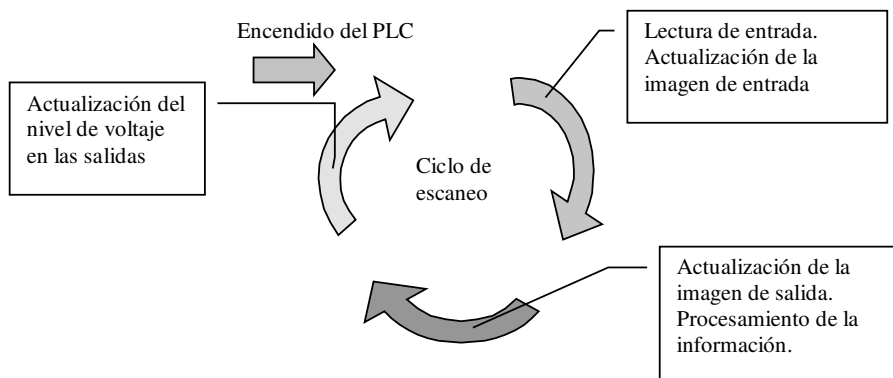
Etapa de auto chequeo e inicialización



Etapa de procesamiento o ejecución del programa

Luego de la rutina de inicio, la cual se ejecuta solamente una vez, el PLC entra en un proceso cíclico que consta básicamente de tres pasos:

1. La lectura de las entradas
2. El procesamiento de la información según el programa del usuario
3. La modificación de las salidas



Cada ciclo es denominado *scan* o barrido, y el tiempo de su duración *scan time* o tiempo de escaneo.

Finalizado el ciclo de escaneo, el PLC inicia un nuevo ciclo a partir del primer paso, repitiéndolos una y otra vez mientras se encuentre en operación.

El tiempo de escaneo redunda en la velocidad de procesamiento del PLC y es afectado por varios factores como son:

- I. La longitud del programa
- II. LA velocidad del microprocesador
- III. El número de entradas y salidas del PLC
- IV. Rutinas de auto chequeo y vigilancia del programa

En un Autómata de mediano desempeño y con un programa de 500 instrucciones, el tiempo de escaneo puede estar alrededor de 10mseg.

Dentro del ciclo de escaneo, además del programa del usuario, también se ejecutan pequeñas rutinas de chequeo, verificando permanentemente que el programa esté corriendo de manera adecuada, las cuales generan una alarma en caso de que el programa del usuario se haya detenido.

Si el programa general se detiene, el PLC puede ejecutar ciertas acciones de prevención, como desenergizar todas sus salidas, con el fin de proteger los dispositivos eléctricos involucrados en el proceso y por qué no, la integridad física de los operarios puesto que las máquinas han podido perder el control.

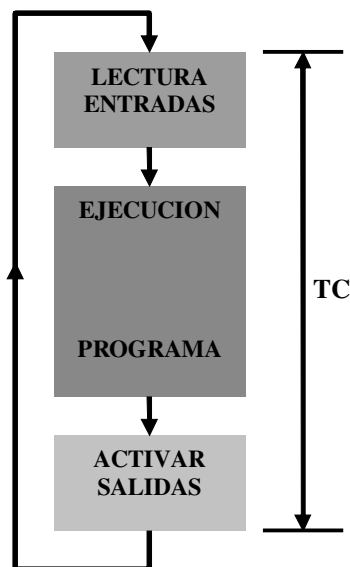
En un sistema de control mediante autómatas programables se tendrán los siguientes tiempos:

1. Retardo de entrada.
2. Vigilancia y exploración de las entradas.
3. Ejecución del programa de usuario.
4. Transmisión de las salidas.
5. Retardo en salidas.

Los puntos 2, 3 y 4 sumados dan como total el tiempo de ciclo del autómata. Tras este ciclo es cuando se modifican las salidas, por lo que si varían durante la ejecución del programa tomarán como valor, el último que se haya asignado.

Esto es así debido a que no se manejan directamente las entradas y las salidas, sino una imagen en memoria de las mismas que se adquiere al comienzo del ciclo (2) y se modifica al final de éste (retardo).

En la etapa de vigilancia se comprueba si se sobrepasó el tiempo máximo de ciclo, activándose en caso afirmativo la señal de error correspondiente.



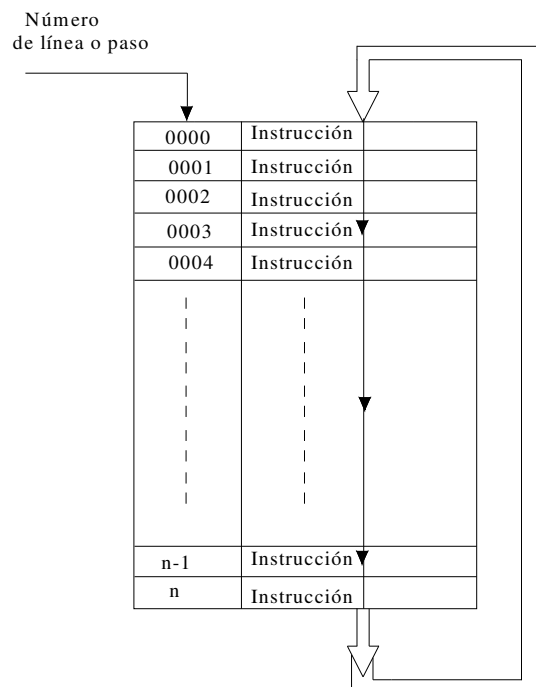
Ejecución del programa

En función de cómo se efectúe la ejecución o barrido del programa, se distinguen los siguientes sistemas, modos o estructuras de programación:

- Ejecución cíclica lineal
- Ejecución con salto condicional
- Ejecución con salto a subrutina
- Ejecución con programas paralelos

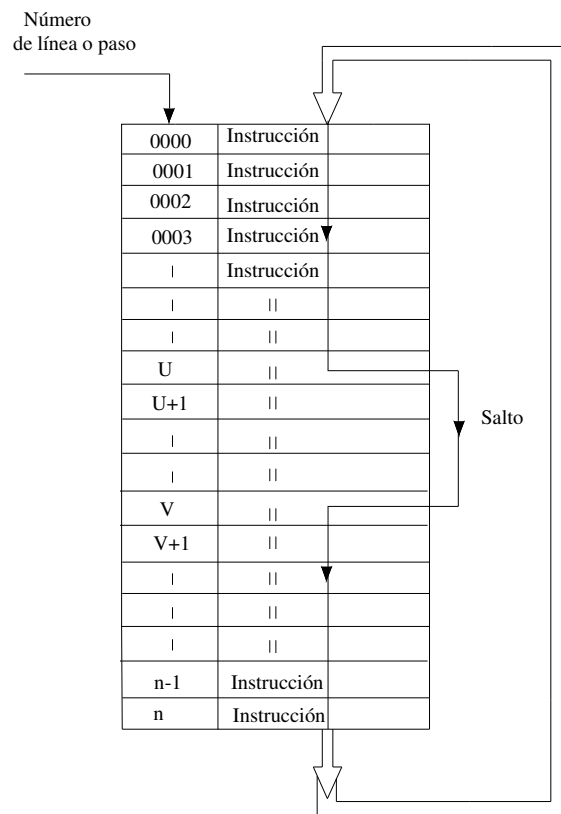
Ejecución lineal

Cuando el ciclo de barrido de la memoria de usuario se realiza línea a línea sin solución de continuidad, se dice que la programación es lineal, y la CPU consulta las instrucciones contenidas en la memoria secuencialmente, una a continuación de la otra, sin alterar este orden.



Salto condicional

Cuando el ciclo de ejecución de un programa tiene la posibilidad, previa condición establecida, de alterar la secuencia línea a línea del mismo y dar un salto a otras líneas de programa, dejando x líneas sin ejecutar, se dice que se ha realizado un salto condicional. Su funcionamiento es el siguiente:

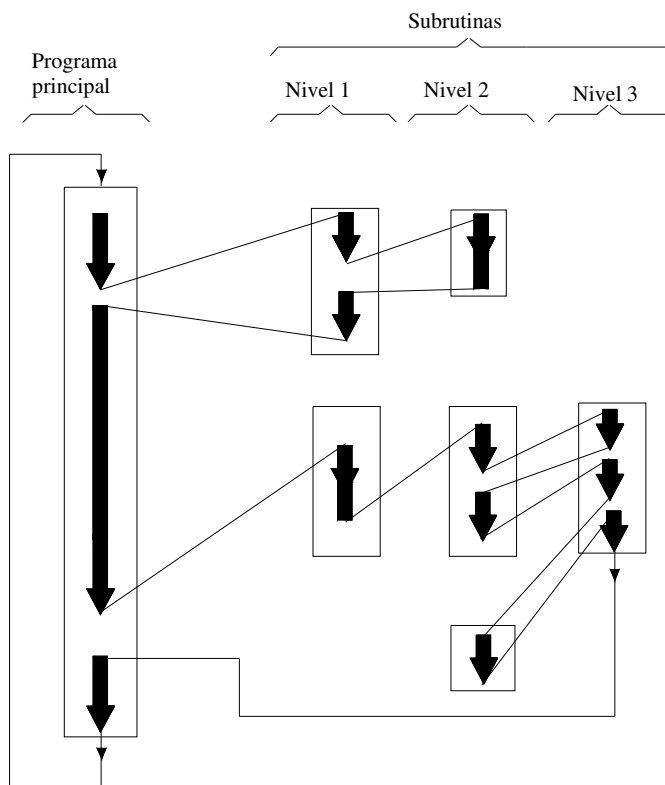


Si al llegar en el proceso de ejecución del programa a la instrucción U, se cumple la condición en ella indicada, se salta a V continuando el barrido en V+1 hasta n. Si por el contrario al llegar a U no se cumple la condición, el programa se ejecuta linealmente continuando en U+1.

Esta posibilidad que poseen muchos Autómatas permite reducir el tiempo de ejecución del ciclo y es aplicable en aquellos casos en que las instrucciones contenidas en el salto sólo son necesarias cuando se dan ciertas condiciones impuestas por el programa.

Salto a subrutina

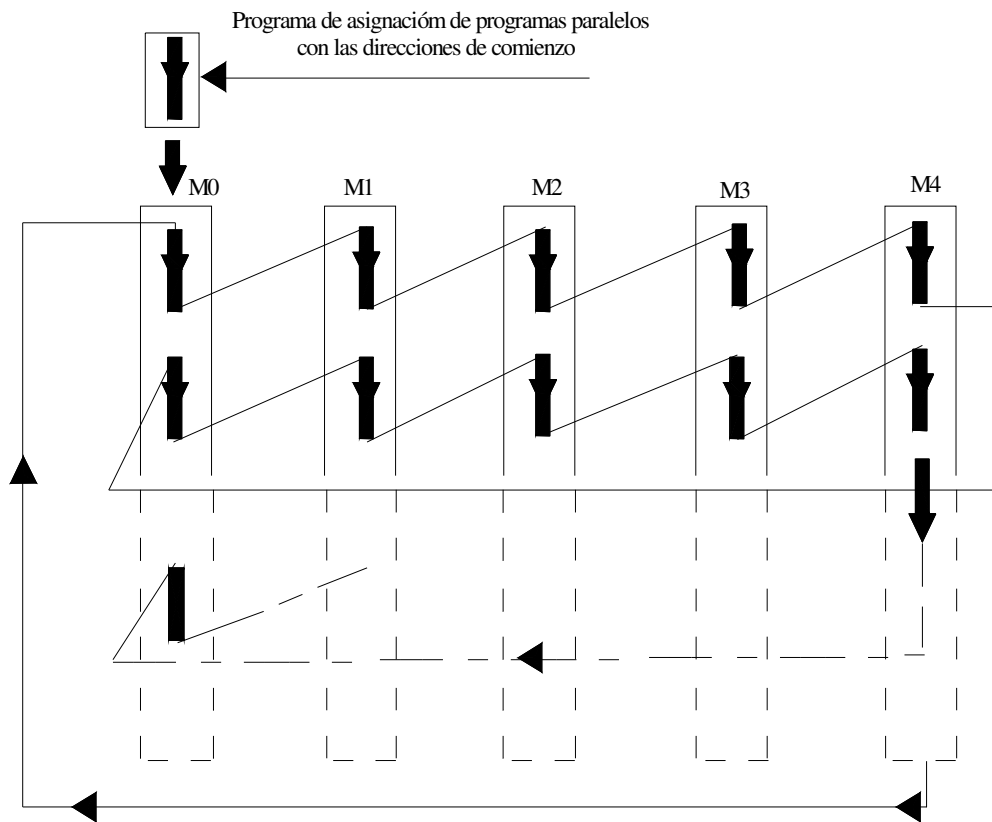
En algunas ocasiones ocurre que en un programa hay uno o más grupos de secuencias de instrucciones idénticas que se repiten y que habrá que reescribir tantas veces como éstas se repitan en dicho programa principal. En estos casos, es muy útil escribir una sola vez esta secuencia o subrutina, e ir a ella cuando se requiera. Los Autómatas de gama baja no suelen incorporar esta posibilidad.



Programas paralelos

En este sistema, utilizado por algunos fabricantes, el procesamiento se realiza paralelamente y en forma asincrónica. En aquellos casos en que con un único Autómata se quiera controlar varios procesos totalmente independientes, este sistema es muy útil, aunque también se utiliza controlando funciones de un proceso único. En este tipo de ejecución es posible el uso de subrutinas en cada programa paralelo.

La ejecución de este tipo de programas se realiza de la siguiente forma:



Cada uno de los tramos en la línea gruesa contiene sólo algunas líneas de programa, de tal forma que la secuencia consiste en el procesamiento de, por ejemplo, diez líneas de programa M0, o lo que es lo mismo, una pequeña parte de un programa; a continuación el barrido salta al programa M1 para procesar sus diez primeras líneas, pasando a continuación al M2 realizando el mismo proceso, etc. Cuando ha barrido todos los programas paralelos, incluso las subrutinas, si se encuentran adscritas a esas líneas, vuelve al programa M0 para repetir el ciclo en el siguiente grupo de diez líneas.

Los Autómatas de gama baja, no suelen incorporar esta posibilidad.

LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación permiten al usuario generar rutinas o secuencias, que una máquina pueda entender y ejecutar de manera automática.

Programa: conjunto de instrucciones, órdenes y símbolos reconocibles por el autómatas que le permiten ejecutar la secuencia de control deseada.

Lenguaje de programación: conjunto total de estas instrucciones, órdenes y símbolos.

El software o conjunto de programas son la forma básica de comunicación, en la cual el operario le indica a la máquina lo que desea que ella haga.

Para que un PLC pueda realizar algún proceso industrial se debe introducir un programa que tenga todas las instrucciones que debe seguir para ejecutar una labor específica.

Tipo	Lenguajes de Programación
Algebraicos	Lenguajes Booleanos
	Lista de Instrucciones (IL ó AWL)
	Lenguajes de Alto Nivel (Texto estructurado)
Gráficos	Diagrama de contactos (Ladder)
	Plano de funciones (Símbolos lógicos)
	Intérprete GRAFCET

INTRODUCCIÓN AL ÁLGEBRA DE BOOLE

Muchos componentes utilizados en sistemas de control, como contactores y relés, presentan dos estados claramente diferenciados (abierto o cerrado, conduce o no conduce). A este tipo de componentes se les denomina componentes todo o nada o también componentes lógicos.

Para estudiar de forma sistemática el comportamiento de estos elementos, se representan los dos estados por los símbolos 1 y 0 (0 abierto, 1 cerrado). De esta forma podemos utilizar una serie de leyes y propiedades comunes con independencia del componente en sí; da igual que sea una puerta lógica, un relé, un transistor, etc...

Atendiendo a este criterio, todos los elementos del tipo todo o nada son representables por una variable lógica, entendiendo como tal aquella que sólo puede tomar los valores 0 y 1.

El conjunto de leyes y reglas de operación de variables lógicas se denomina álgebra de Boole, ya que fué George Boole el que desarrolló las bases de la lógica matemática.

Operaciones lógicas básicas

Sea un conjunto formado por sólo dos elementos que designaremos por 0 y 1. Llamaremos variables lógicas a las que toman sólo los valores del conjunto, es decir 0 o 1.

En dicho conjunto se definen tres operaciones básicas:

SUMA LOGICA

Denominada también operación "**O**" (**OR**). Esta operación responde a la siguiente tabla:

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

PRODUCTO LOGICO

Denominada también operación **"Y" (AND)**. Esta operación responde a la siguiente tabla:

a	b	a*b
0	0	0
0	1	0
1	0	0
1	1	1

NEGACION LOGICA

Denominada también operación **"N" (NOT)**. Esta operación responde a la siguiente tabla:

a	a'
0	1
1	0

Otras operaciones lógicas

A partir de las operaciones lógicas básicas se pueden realizar otras operaciones booleanas, las cuales son:

NAND (AND Negada), cuya tabla correspondiente es

a	b	(a*b)'
0	0	1
0	1	1
1	0	1
1	1	0

NOR (OR Negada), cuya tabla correspondiente es

a	b	$(a+b)'$
0	0	1
0	1	0
1	0	0
1	1	0

XOR, también llamada función OR-EXCLUSIVA. Responde a la tabla

a	b	$a(+)b$
0	0	0
0	1	1
1	0	1
1	1	0

Lenguajes Booleanos y lista de Instrucciones (IL ó AWL)

El lenguaje booleano esta constituido por un conjunto de instrucciones que son transcripción literal de las *funciones del álgebra de Boole o álgebra lógica*.

A continuación figura una relación de mnemónicos, con indicación de lo que representan:

STR: Operación inicio contacto abierto

STR NOT: Operación inicio contacto cerrado

AND (Y): Contacto serie abierto

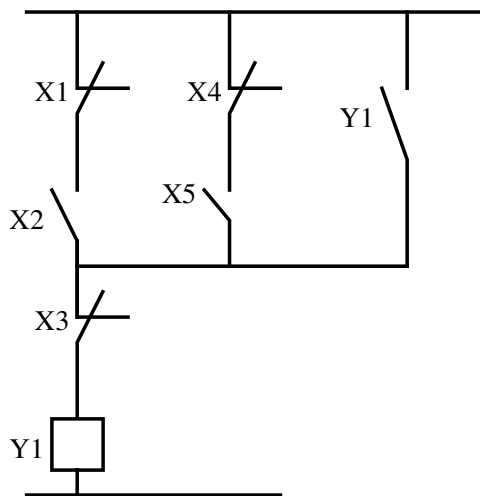
AND NOT: Contacto serie cerrado

OR(O): Contacto paralelo abierto

OR NOT: Contacto paralelo cerrado

OUT: Bobina de relé de salida

Un ejemplo sería:



Esquema de relés

Ecuación lógica:

$$[(\overline{X1} \bullet X2) + (\overline{X4} \bullet X5) + Y1] \overline{X3} = Y1$$

Nemónicos de la ecuación:

```
STR NOT X1
AND X2
STR NOT X4
AND X5
OR STR
OR Y1
AND NOT X3
OUT Y1
```

En una operación normal el Autómata utiliza algunas otras instrucciones del lenguaje *booleano* que le permiten manejar elementos de automatización y que son las siguientes instrucciones secuenciales:

TMR: Temporizador

CNT: Contador

MCS: Conexión de una función a un grupo de salidas

MCR: Fin de la conexión del grupo de salidas

SFR: Registro de desplazamiento

SET: Activar una variable binaria (unidad de memoria)

RST: Desactivar una variable binaria

También existen otros tipos de instrucciones como las siguientes:

DD: Sumar

BB: Restar

MUL: Multiplicar

DIV: Dividir

CMP: Comparar variables digitales

FR: Rotaciones de bits (variables de binarias)

HIFT: Rotaciones de palabras (variables digitales)

MOV: Transferencias de datos
CD / BIN: Conversiones de códigos numéricos
END: Fin de programa
JMP: Salto de bloque de programa
MCS: Habilitación de bloque de programa
JMPSUB: Salto a subrutina, etc.

A este lenguaje no se le puede llamar ya *booleano*, después de ampliarlo con estas extensiones se lo denomina ***Lista de instrucciones***

La lista de instrucciones (IL o AWL) es un lenguaje de bajo nivel. Este lenguaje es adecuado para pequeñas aplicaciones. En los Autómatas de gama baja, este es el único modo de programación.

Lenguajes de Alto Nivel (Texto estructurado)

En ellos las instrucciones son líneas de texto que utilizan palabras o símbolos reservados

SET, AND, FOR, etc.

Las operaciones se definen por los símbolos matemáticos habituales

+, *, <, >, -, /, etc.

También se dispone de funciones trigonométricas, logarítmicas y de manipulación de variables complejas

Sen, Cos, pi, real, img, etc.

Sin embargo, lo que distingue realmente estos lenguajes avanzados de las listas de instrucciones (IL ó AWL) son las tres características siguientes:

1. Son *lenguajes estructurados*, donde es posible la programación por bloques con definición de variables locales o globales.
2. Incluyen estructuras de cálculo repetitivo y condicional tales como:

- a. **FOR ... TO**
- b. **REPEAT ... UNTIL X**
- c. **WHILE X...**
- d. **IF ... THEN ...ELSE**

3. Disponen de instrucciones de manipulación de cadenas de caracteres, muy útiles en aplicaciones de gestión, estadística, etc.

Convenientemente adaptado a las aplicaciones del Autómata, el **BASIC** se conoce como el lenguaje de alto nivel más extendido.

También se pueden encontrar intérpretes o compiladores de C, PASCAL, FORTRAN, etc., lo que permite resolver tareas de cálculo científico en alta resolución, clasificaciones de datos, estadísticas, etc., con total facilidad, y con acceso además a módulos y subrutinas específicos ya escritos en estos lenguajes y de uso general en aplicaciones informáticas.

Dado lo específico de su aplicación un programa escrito en alto nivel necesita para su edición de una unidad de programación avanzada o de un software de desarrollo de programas que corra sobre una computadora.

Adicionalmente, es frecuente que el empleo de estos lenguajes estructurados obligue además a utilizar no solo una unidad de programación tipo PC, sino incluso una CPU especial en el Autómata (coprocesadora), capaz de interpretar y ejecutar las nuevas instrucciones.

En cualquier caso, los lenguajes de alto nivel son posibilidades adicionales al alcance del programador, que puede si así lo desea, utilizar solo las formas básicas de contactos, bloques o lista de instrucciones para escribir sus aplicaciones: en otras palabras, los lenguajes avanzados nunca constituyen el lenguaje básico de un Autómata o familia de Autómatas, papel que queda reservado a la lista de instrucciones o al diagrama de contactos.

Una ventaja adicional del programa en alto nivel es que a él se puede transcribir, casi literalmente el diagrama de flujos que constituye la primera aproximación a la representación del sistema de control.

Plano de funciones (Símbolos lógicos)

La programación por símbolos lógicos incluye como bloques normalizados algunas funciones secuenciales típicas en automatización como:

- Temporizadores
- Contadores
- Bloques combinacionales
- Bloques de tratamiento numéricos
 - Multiplexores
 - Demultiplexores
 - Sumadores
 - Multiplicadores
 - etc.

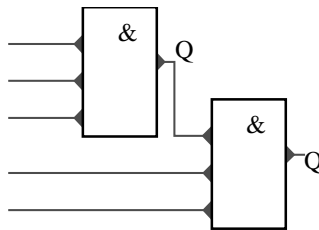
Nº menor de funciones que Las listas de instrucciones
 Diagramas de contactos

Debido a esto la programación por símbolos lógicos queda reservada a aplicaciones en las que solo intervengan

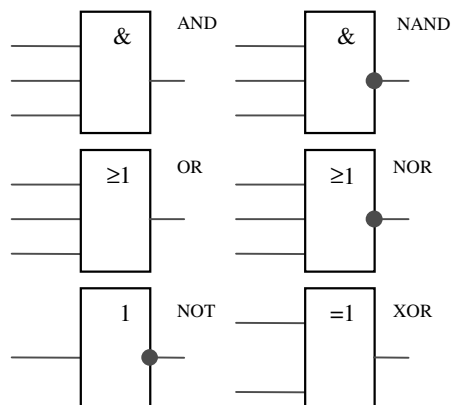
variables booleanas **todo - nada**, y algunos bloques secuenciales elementales: temporizadores, contadores, registros de desplazamiento, etc. Si a esto le sumamos que el usuario habitual del autómat desconoce los símbolos lógicos \perp baja difusión de estos lenguajes de programación.

Todas las funciones **AND**, **OR**, **XOR**, **NAND** y **NOR** tienen tres entradas y una salida.

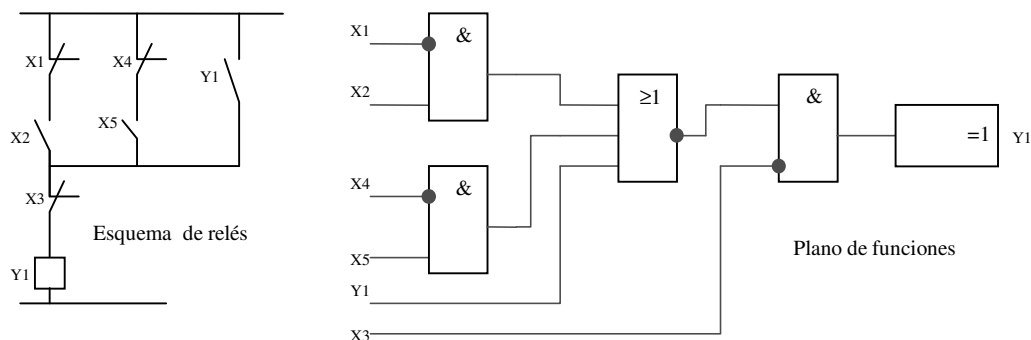
Si se desea realizar operaciones con más de tres entradas, se conectan varios bloques en cascada



La función inversora, **NOT**, tiene una entrada y una salida, y la función **OR** exclusiva (**XOR**) posee dos entradas y una salida.



Un ejemplo de los lenguajes vistos hasta ahora es:



Ecuación lógica:

$$[(\overline{X1} \bullet X2) + (\overline{X4} \bullet X5) + Y1] \overline{X3} = Y1$$

Nemónicos de la ecuación:

```
STR NOT X1
AND X2
STR NOT X4
AND X5
OR STR
OR Y1
AND NOT X3
OUT Y1
```