



MAGNA

INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

CURSO DE FORMACIÓN



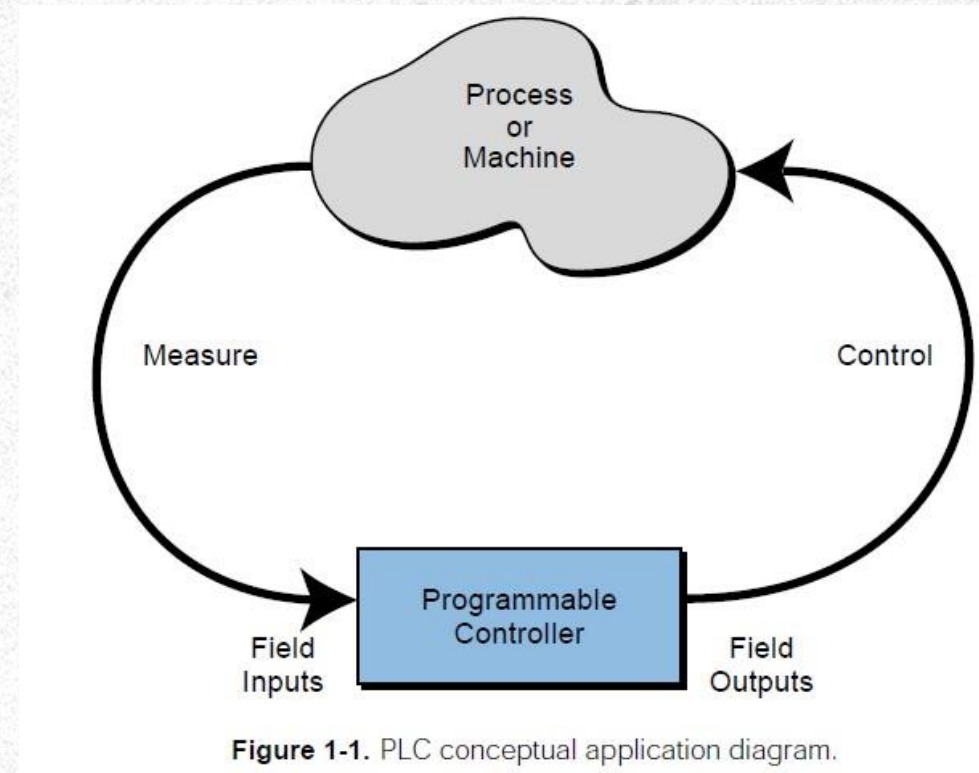
AUTOMATIZACION INSTRUMENTACION Y CONTROL DE PROCESOS INDUSTRIALES

Módulo 3

- Control de Procesos con PLC (Parte 1): Instrumentos de campo
- PLC en la pirámide de la automatización
- Niveles de control por PLC y supervisión
- Introducción a RSLogix5000 y RSLinx
- Funciones booleanas de enclavamiento
- Set, reset, temporizadores on delay, off delay

MODULO 3: PLC:

- Los controladores lógicos programables, también llamados PLC son miembros de la familia de computadoras de estado sólido y utilizan circuitos electrónicos en lugar de dispositivos electromecánicos para implementar funciones de control.
- Son capaces de almacenar instrucciones, como secuenciación, sincronización, conteo, aritmética, manipulación de datos y comunicación. El conjunto de estas instrucciones forman el programa lógico que controlara máquinas y procesos industriales.
- La figura 1-1 ilustra un concepto de PLC.



MODULO 3: PLC:

La historia del PLC (Controlador de Lógica Programable) se remonta a la década de los '70, cuando nace como respuesta a la necesidad creciente de implementar un dispositivo flexible a las modificaciones en terreno, resistente a la contaminación ambiental, humedad, vibraciones, temperaturas extremas y perturbaciones eléctricas, las cuales estaban aumentando con la incorporación de la electrónica de potencia y otras nuevas tecnologías.

Los primeros PLCs tuvieron un éxito inmediato, ya que funcionaban como reemplazo de los paneles basados en relés electromecánicos, permitiendo reducir dramáticamente el elevado costo que significaba la localización de las fallas, la corrección de problemas, las modificaciones en terreno y, por ende, la cantidad de horas-hombre que se ocupaban en este tipo de tareas.

MODULO 3: PLC:

Otros de los factores que influyeron en su aceptación en el mercado fueron su facilidad de programación y contar con un lenguaje conocido y familiar para los electricistas de esa época, ya que estaba basado en diagramas de escalera y símbolos eléctricos comunes para ellos (ver figura 1).

En la actualidad, los principales fabricantes de PLCs han normalizado los lenguajes de programación bajo la norma internacional IEC 61131-3, que, entre otros aspectos, considera cinco lenguajes: Diagrama Bloque Funcionales (FBD), Lista de Instrucciones (IL), Diagrama de Escalera (LD), Texto Estructurado (ST) y Gráfica Funciones Secuenciales (SFC).

MODULO 3: PLC:

Diagrama Eléctrico

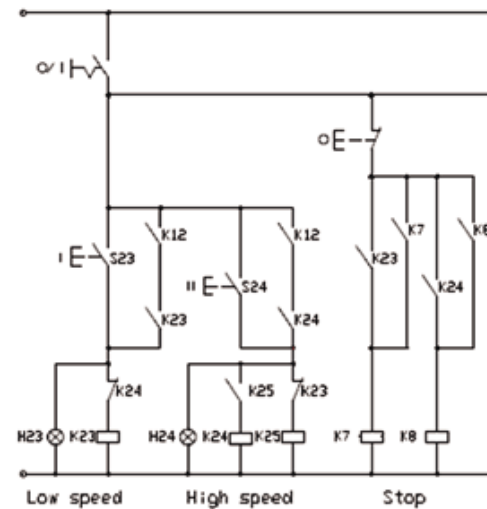
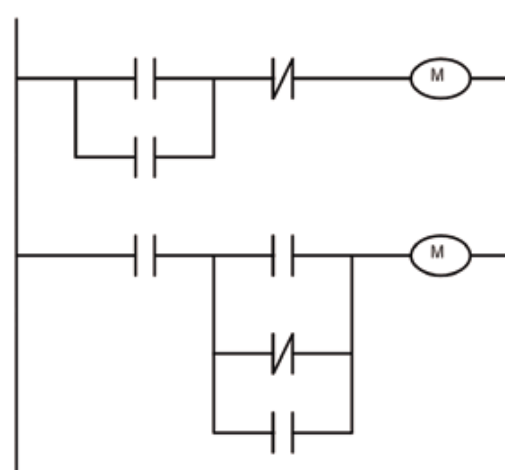


Diagrama Escalera



MODULO 3: PLC:

PARTES DEL PLC

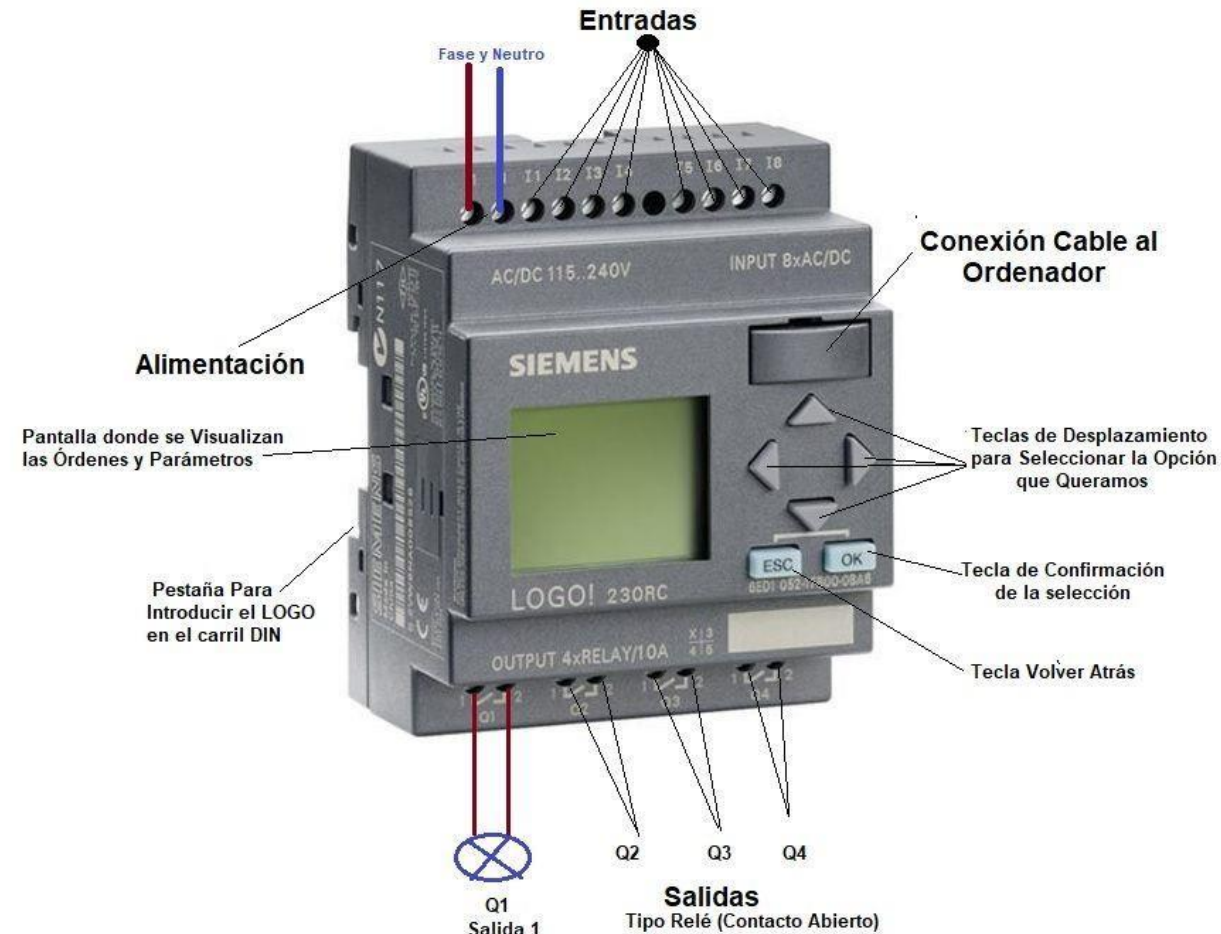
El PLC esta compuesto básicamente de tres partes importantes:

- Módulo de alimentación (Power Supply)
- Unidad central de procesamiento (CPU)
- Módulos de entrada y salida (bloque de entrada y salidas)

Los PLC modulares estan compuestos de otros partes adicionales como:

- Rack o chasis
- Módulo de alimentación
- Unidad central de procesamiento (CPU)
- Módulos de entrada y salida
- Módulos de interfaz de comunicación

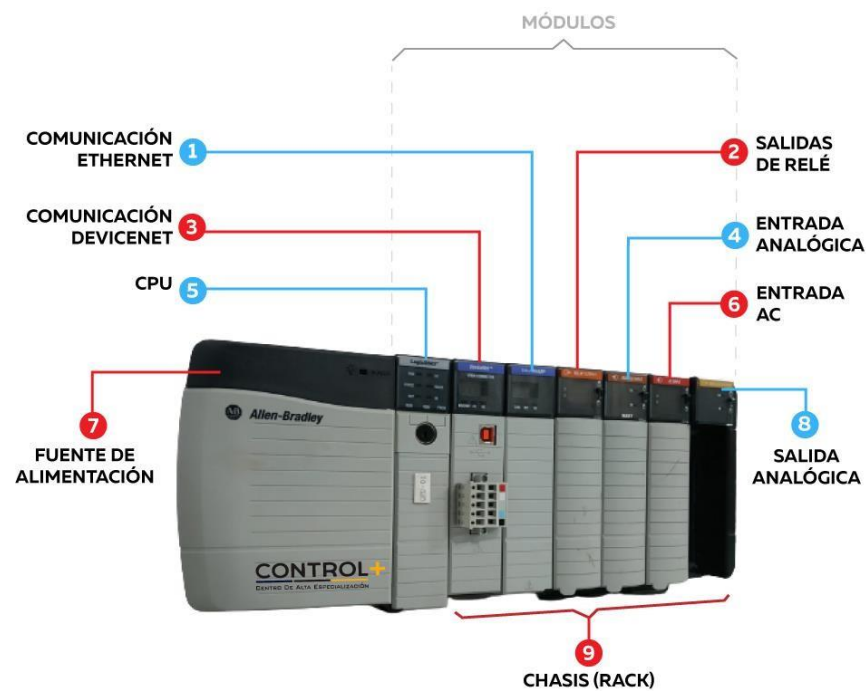
PARTES DEL LOGO! DE SIEMENS



MODULO 3: PLC:



PARTES DE UN PLC MODULAR



PLC:

La Memoria del PLC

La memoria tiene como función almacenar las instrucciones del programa de usuario, datos temporales, estados internos y estados de entrada/salida.

La memoria de un PLC está organizada en zonas de trabajo, denominadas áreas, donde el microprocesador almacenará y consultará la información en cierto momento de su operación.

Áreas de trabajo

Memoria de programa de usuario:
Almacena las instrucciones del programa a ejecutar.

Memoria de datos:
Almacena todos aquellos datos necesarios para la ejecución del programa de usuario por parte del microprocesador.

El área de datos se descompone en:

Área de datos y módulos internos:
Almacena el estado de cada uno de los datos internos con los que trabaja el PLC:

- Imagen de entradas y salidas.
- Temporizadores y contadores.
- Registros de desplazamientos.
- Contadores.
- Etc.

Área de datos internos y variables temporales:
Almacena todos aquellos datos intermedios y temporales que requiera la ejecución del programa de usuario.

Área de Entradas/salidas:
Almacena el estado de las entradas y salidas de que disponga el PLC.

CLASIFICACION DEL PLC

CLASIFICACION POR SU CONSTRUCCION

PLC COMPACTOS

Se estructuran en forma compacta. En un sólo bloque se encuentra: la fuente, el CPU, las entradas y salidas. Están limitados por sus dimensiones y no tienen capacidad de crecer.

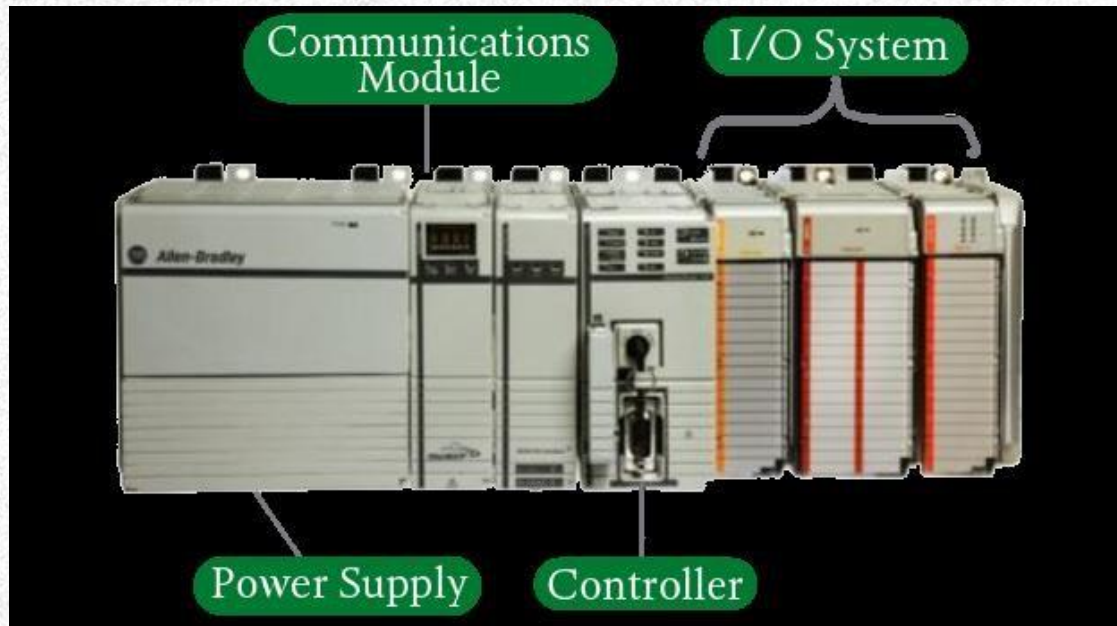
Algunas de sus ventajas son:

- Su precio, ya que son los más económicos dentro de la variedad de PLC.
- Requieren menor espacio por su construcción compacta.
- Su programación es bastante sencilla.
- No se requieren de conocimientos especializados para su manejo (capacitación).
- Son fáciles de instalar.

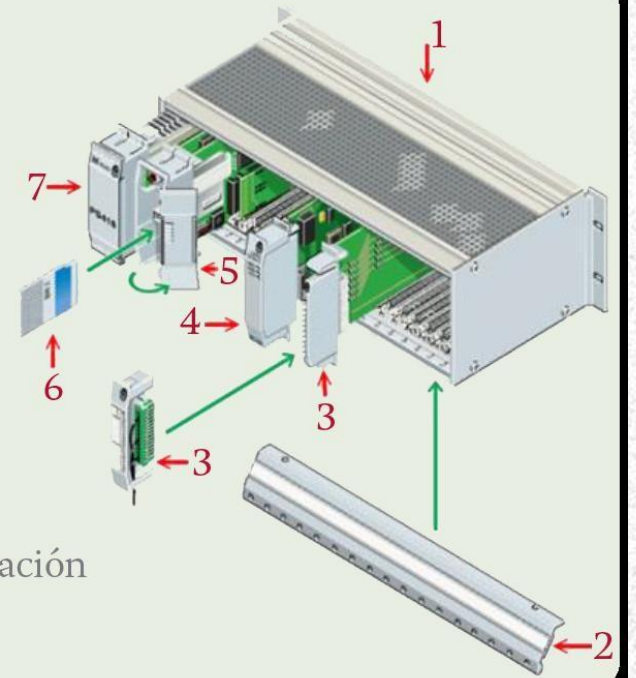
PLC modulares

Los elementos que conforman al PLC se encuentran separados, formando módulos. Ampliable a través de un chasis o panel posterior (back panel), ahora pueden ser montados en un riel y aumentar de acuerdo con las necesidades del cliente, además de las especificaciones del fabricante, incrementando memoria, capacidades de comunicación (ejemplo: módulo ethernet), alimentación, módulos de entrada/salida de diversos tipos (salidas TTL, analógicas, control de movimiento, etcétera).

CLASIFICACION DEL PLC



1. Rack
2. Barra de compensación de potencial
3. Tarjetas de entradas y salidas
4. Tarjetas de comunicación
5. CPU
6. Tarjeta de memoria
7. Tarjeta de fuente de alimentación



CLASIFICACION DEL PLC

PLC modulares

Algunas de sus ventajas y desventajas son:

- Son más caros y varían de acuerdo con la configuración
- Utilizan mayor espacio que los compactos. Aunque hoy en día con la evolución de la tecnología, cada día son más reducidos en sus dimensiones.
- Su mantenimiento y programación requieren más tiempo y de personal capacitado.
- Las ampliaciones se hacen de acuerdo con las necesidades, por lo general, se incrementan los módulos de E/S ya sean discretos o analógicos.
- En caso de avería, puede aislarse el problema, cambiando el modulo averiado, sin afectar la funcionalidad o programación del resto.

PLC

CLASIFICACION POR EL NUMERO DE E/S

Dependiendo del número de Entrada/Salidas que puede manejar un PLC se clasifican en:

- nano (<64 E/S)
- micro (64 E/S)
- pequeño (65 a 255 E/S)
- mediano (256 a 1023 E/S)
- grande (>1024 E/S)

PLC NANO

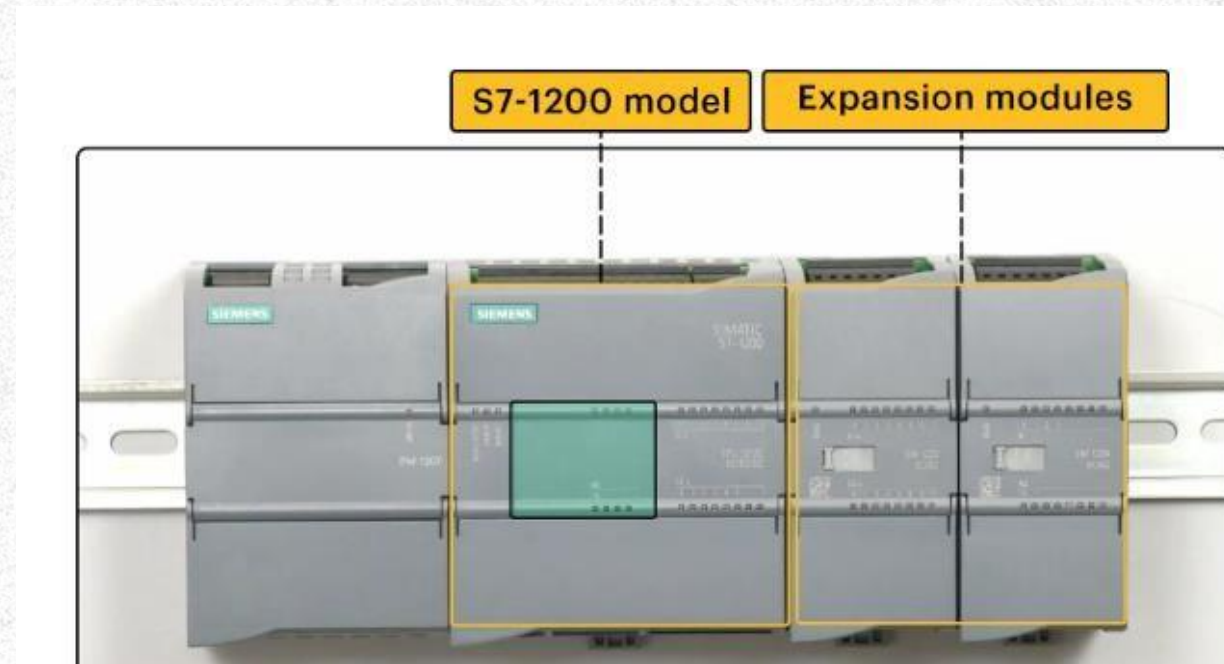


PLC:

PLC Micro



PLC Mediano



PLC:

PLC Grande



PLC:

SECUENCIA DE OPERACIÓN DEL PLC O AUTOMATA

La secuencia básica de operación del autómata PLC se puede dividir en cuatro fases principales:

- Autodiagnóstico. El PLC realiza una autoverificación de hardware y software, revisa las condiciones de memoria, niveles de tensión de alimentación, estado de la batería, conexiones de interfaz y la estructura del programa por errores de sintaxis y códigos incorrectos (este proceso lo realiza completo solo al reinicializar el procesador).
- Lectura de señales desde la interfaz de entradas. El estado de las entradas físicas digitales y analógicas conectadas a los módulos del autómata PLC son leídas y procesadas guardándose en un área llamada imagen de proceso. Esto asegura que los valores de las entradas serán coherentes cuando se ejecute el programa de usuario.
- Procesado del programa para obtención de las señales de control. El programa de usuario ejecuta de forma secuencial las líneas de código con sus instrucciones escribiendo el resultado de esta lógica en la imagen de salidas del proceso. Por ejemplo, en Siemens estaríamos hablando del bloque OB1 donde reside el programa principal.
- Escritura de señales en la interfaz de salidas. Por último, el resultado de las salidas almacenadas en la imagen de proceso se escribe en las salidas físicas del autómata PLC.

PLC:

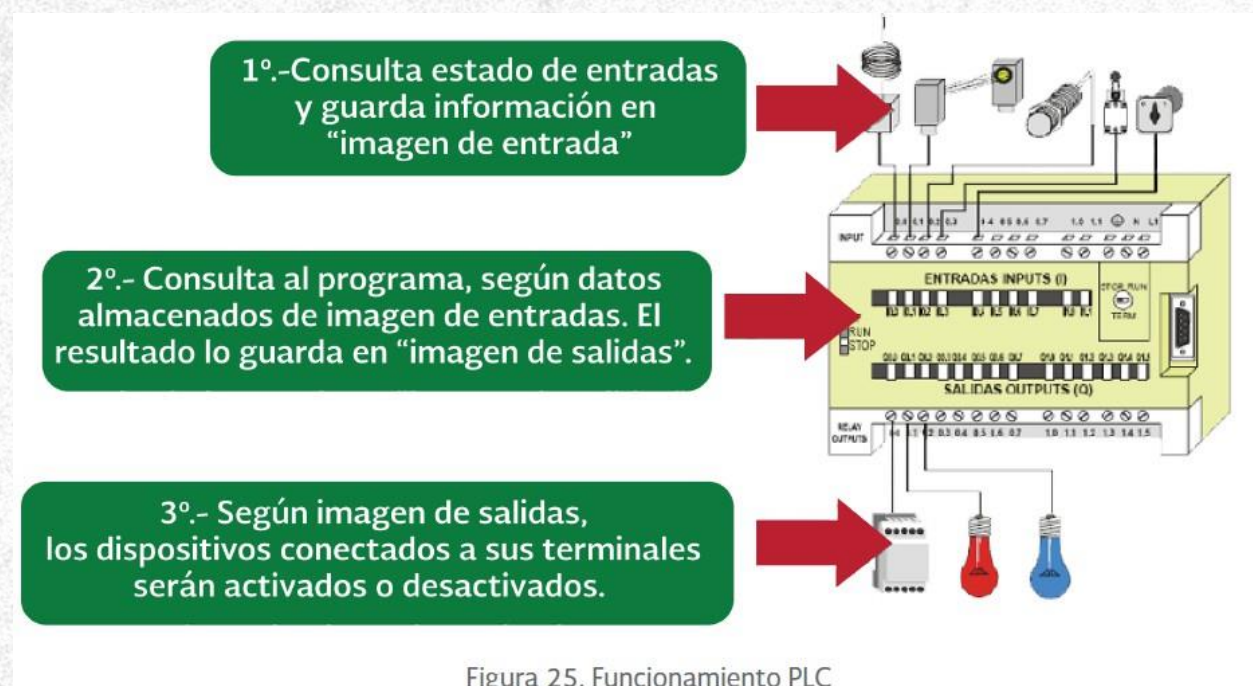


Figura 25. Funcionamiento PLC

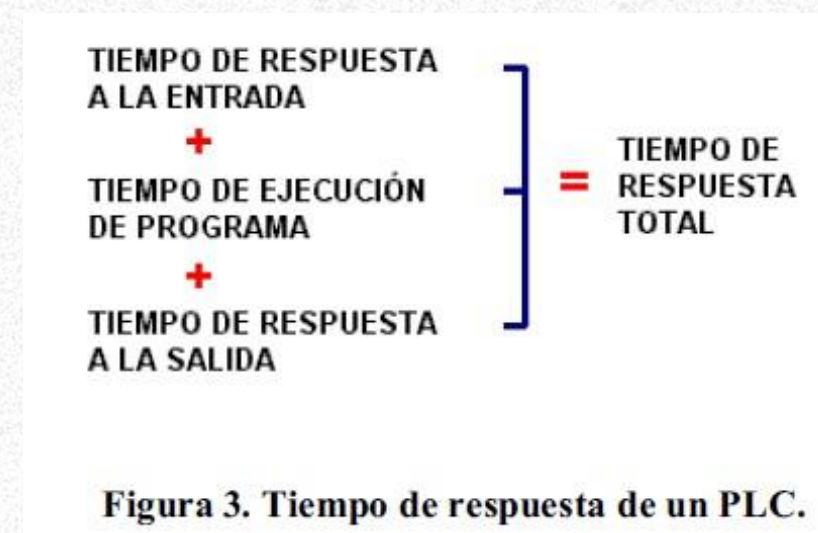
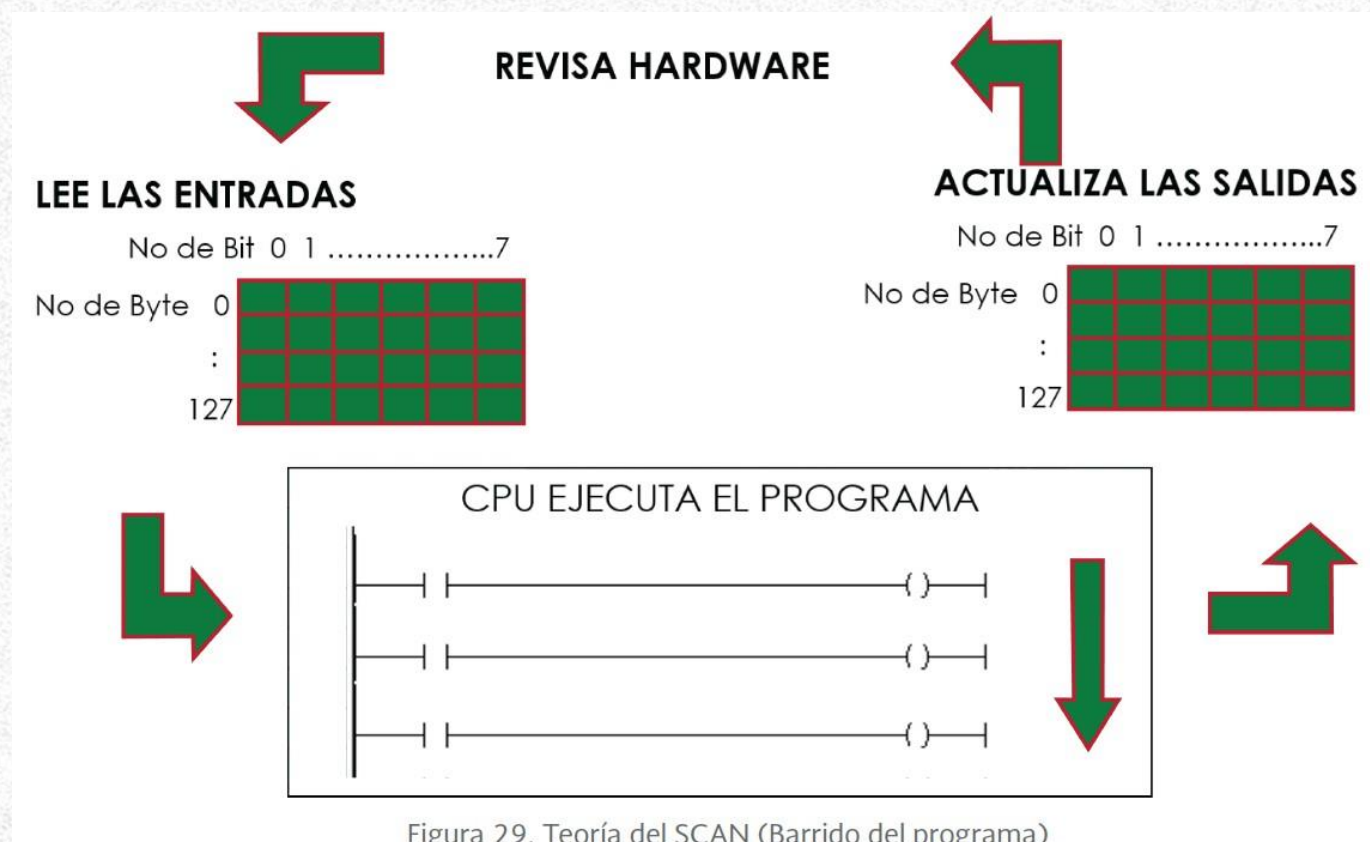


Figura 3. Tiempo de respuesta de un PLC.

PLC:



PLC:

MODOS DE OPERACIÓN DEL PLC

Los cuatro modos de operación genéricos de los PLC son: RUN, STOP, ERROR y POWER - ON.

- **RUN.** El PLC ejecuta el programa de usuario como modo normal de operación.
- **STOP.** El PLC ignora el programa de usuario y efectúa sus programas internos. En este modo el usuario puede monitorear y/o programar el PLC desde una PC.
- **ERROR.** El PLC detiene la ejecución del programa de usuario, pues sus programas internos encuentran un error de programación o de hardware. Como en el caso del modo STOP, se inactiva todas las salidas y no se permite llevar a cabo el modo RUN hasta que no se resuelva el problema que causó el error.
- **POWER-ON.** Este modo ocurre a partir del momento en que se energiza el PLC; el cual utiliza el modo POWER-ON para auto configurarse y hacer comprobaciones de estado del sistema. Una vez realizadas las rutinas de inicio asumen según las condiciones del sistema, uno de los otros tres modos.
- Por supuesto cada fabricante tiene sus variantes para operar.

PLC:

En los PLC Allen-Bradley (Rockwell) los define así:

- **PROG:** la lógica del programa no se ejecuta en el procesador, las entradas no se monitorean ni se actualizan las salidas, el modo edición se encuentra habilitada.
- **RUN:** se ejecuta la lógica del programa, se monitorean las entradas y se actualizan las salidas.
- **FAULT:** cuando ocurre un Major Fault (error) en el procesador y se detiene el proceso.
- **REM:** En este modo es posible cambiar el modo de operación del controlador de manera remota. Más que un modo es el permiso para realizar cambios en configuración o programación.



PLC:

LENGUAJES DE PROGRAMACION DE PLC ´S

1. PROGRAMA Y LENGUAJE DE PROGRAMACIÓN

Se puede definir un programa como un conjunto de instrucciones, órdenes y símbolos reconocibles por el PLC, a través de su unidad de programación, que le permiten ejecutar una secuencia de control deseada. El Lenguaje de Programación en cambio, permite al usuario ingresar un programa de control en la memoria del PLC, usando una sintaxis establecida.

Al igual como los PLCs se han desarrollado y expandido, los lenguajes de programación también se han desarrollado con ellos. Los lenguajes de hoy en día tienen nuevas y más versátiles instrucciones y con mayor poder de computación.

Adicionalmente a las nuevas instrucciones de programación, el desarrollo de nuevos módulos de entradas y salidas también ha obligado a cambiar las instrucciones existentes.

2. PROGRAMAS DE APLICACIÓN Y DEL SISTEMA

Los programas de aplicación que crean los usuarios están orientados a ejecutar, a través del controlador, tareas de automatización y control. Para ello, el usuario escribe el programa en el lenguaje de programación que mejor se adapte a su trabajo y con el que sienta poseer un mejor dominio. En este punto es importante señalar, que algunos fabricantes no ofrecen todas las formas de representación de lenguajes de programación, por lo que el usuario deberá adaptarse a la representación disponible.

Por otro lado, el conjunto de programas que realizan funciones operativas internas del controlador, incluyendo los traductores de lenguaje, reciben la denominación de programas del sistema o software del sistema.

PLC:

3. TIPOS DE LENGUAJES DE PROGRAMACIÓN DE PLCS

- En la actualidad cada fabricante diseña su propio software de programación, lo que significa que existe una gran variedad comparable con la cantidad de PLCs que hay en el mercado. No obstante, actualmente existen tres tipos de lenguajes de programación de PLCs como los más difundidos a nivel mundial; estos son:
 - Lenguaje de contactos o Ladder
 - Lenguaje Booleano (Lista de instrucciones)
 - Diagrama de funciones

4. LA NORMA IEC 1131-3

- La Comisión Electrotécnica Internacional (IEC) desarrolló el estándar IEC 1131, en un esfuerzo para estandarizar los Controladores Programables. Uno de los objetivos del Comité fue crear un conjunto común de instrucciones que podría ser usado en todos los PLCs.
- El estándar IEC 1131 para controladores programables consiste en cinco partes, una de las cuales hace referencia a los lenguajes de programación y es referida como la IEC 1131-3.

PLC:

- **Lenguajes Gráficos**

- Diagrama Ladder (LD)
- Diagrama de Bloques de Funciones (FBD)

- **Lenguajes Textuales**

-
- Lista de Instrucciones (IL)
- Texto Estructurado (ST)
-

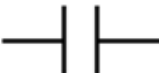





- Adicionalmente, el estándar IEC 1131-3 incluye una forma de programación orientada a objetos llamada **Sequential Function Chart (SFC)**. SFC es a menudo categorizado como un lenguaje IEC 1131-3, pero éste es realmente una estructura organizacional que coordina los cuatro lenguajes estándares de programación (LD, FBD, IL y ST). La estructura del SFC tuvo sus raíces en el primer estándar francés de **Grafcet** (IEC 848).

PLC:

5. LENGUAJE LADDER

- El LADDER, también denominado lenguaje de contactos o de escalera, es un lenguaje de programación gráfico muy popular dentro de los PLC, debido a que está basado en los esquemas eléctricos de control clásicos. De este modo, con los conocimientos que todo técnico eléctrico posee, es muy fácil adaptarse a la programación en este tipo de lenguaje. Su principal ventaja es que los símbolos básicos están normalizados según normas NEMA y son empleados por todos los fabricantes.
- Para programar un PLC con LADDER, además de estar familiarizado con las reglas de los circuitos de conmutación, es necesario conocer cada uno de los elementos de que consta este lenguaje. En la siguiente tabla podemos observar los símbolos de los elementos básicos junto con sus respectivas descripciones.

PLC:

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa, esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema.
	Contacto NC	Su función es similar al contacto NA anterior, pero en este caso se activa cuando hay un cero lógico, cosa que deberá de tenerse muy en cuenta a la hora de su utilización.
	Bobina NA	Se activa cuando la combinación que hay a su entrada (izquierda) da un uno lógico. Su activación equivale a decir que tiene un uno lógico. Suele representar elementos de salida, aunque a veces puede hacer el papel de variable interna.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y usada junto con la bobina RESET dan una enorme potencia en la programación.
	Bobina SET	Permite desactivar una bobina SET previamente activada.

PLC:

6. LENGUAJE BOOLEANO (Lista de Instrucciones)

El lenguaje Booleano utiliza la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control.

Consiste en elaborar una lista de instrucciones o

nemónicos, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones nemónicas, para implementar el circuito de control. El lenguaje

“Lista de Instrucciones” (IL) de la Norma IEC 1131-3, es una forma de lenguaje Booleano.

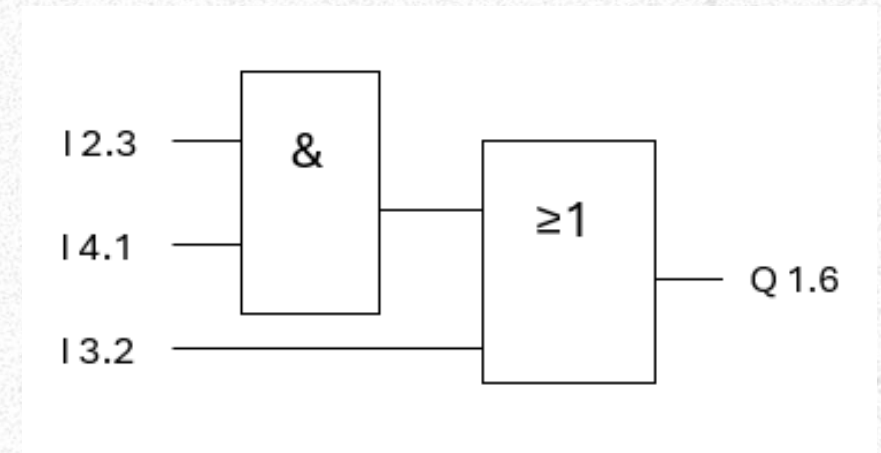
- Ejemplo de programación Booleana:

• A	I	2.3
• A	I	4.1
• O	I	3.2
• =	Q	1.6

PLC:

7. DIAGRAMA DE FUNCIONES (FBD)

- Es un lenguaje gráfico que permite al usuario programar en bloques de funciones del PLC, en tal forma que ellos aparecen interconectados al igual que un circuito eléctrico. Generalmente utilizan símbolos lógicos para representar al bloque de función. Las salidas lógicas no requieren incorporar una bobina de salida, porque la salida es representada por una variable asignada a la salida del bloque.
- Ejemplo de programación mediante diagrama de funciones:



PLC:

8. LENGUAJE DE TEXTO ESTRUCTURADO (ST)

Texto estructurado (ST) es un lenguaje de alto nivel que permite la programación estructurada, lo que significa que muchas tareas complejas pueden ser divididas en unidades más pequeñas. ST se parece mucho a los lenguajes de computadoras BASIC o PASCAL, que usa subrutinas para llevar a cabo diferentes partes de las funciones de control y paso de parámetros y valores entre las diferentes secciones del programa.

Incluye estructuras de cálculo repetitivo y condicional, tales como: FOR ... TO; REPEAT..... UNTIL X; WHILE X... ; IF ... THEN ...ELSE. Además soporta operaciones Booleanas (AND, OR, etc.) y una variedad de datos específicos, tales como fecha, hora.

Ejemplo:

- IF Manual AND Alarm THEN
 - Level = Manual_Level;
 - Mixer = Start AND NOT Reset
- ELSE IF Other_Mode THEN
 - Level = Max_level;
- ELSE Level = (Level_Indic X100)/Scale;
- END IF;

PLC:

9. SEQUENTIAL FUNCTION CHART (SFC)

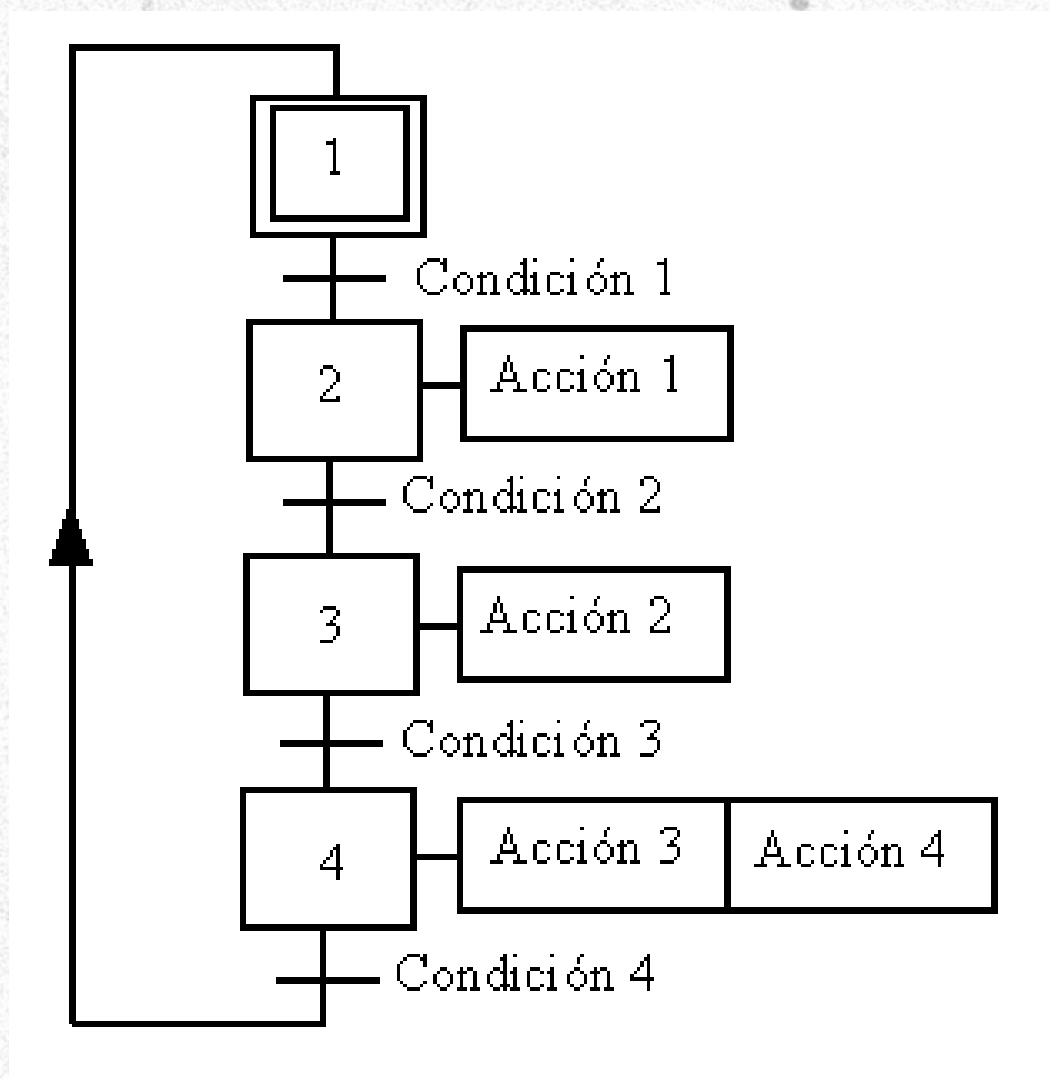
Es un “lenguaje” gráfico que provee una representación diagramática de secuencias de control en un programa. Básicamente, SFC es similar a un diagrama de flujo, en el que se puede organizar los subprogramas o subrutinas (programadas en LD, FBD, IL y/o ST) que forman el programa de control. SFC es particularmente útil para operaciones de control secuencial, donde un programa fluye de un punto a otro una vez que una condición ha sido satisfecha (cierta o falsa).

El marco de programación de SFC contiene tres principales elementos que organizan el programa de control:

- Pasos (etapas)
- Transiciones (condiciones)
- Acciones

El programa irá activando cada una de las etapas y desactivando la anterior conforme se vayan cumpliendo cada una de las condiciones. Las acciones se realizarán en función de la etapa activa a la que están asociadas. Por ejemplo, la etapa 1 activa tras arrancar el programa, al cumplirse la "Condición 1", se activará la etapa 2, se desactivará la 1, y se realizará la "Acción 1".

PLC:



PLC:

Tipos de Datos que maneja el PLC

Controladores Logix (Controllogix, Compactlogix, Flexlogix, Micro)

En un controlador Logix5000 se utiliza un tag (nombre alfanumérico) para dirigirse a los datos (variables).

Tag: Un tag es un nombre basado en texto para un área de la memoria del controlador donde se almacenan datos.

Los tags son el mecanismo básico para asignar memoria, hacer referencia a datos de la lógica y monitorear los datos.

La asignación mínima de memoria para un tag es de cuatro bytes.

Cuando se crea un tag con datos que requieren menos de 4 bytes, el controlador asigna 4 bytes pero los datos solo ocupan la parte que necesitan

El controlador utiliza el nombre del tag internamente y no necesita comprobar una dirección física.

EJEMPLO

Tags

Dispositivo de E/S
analógico →

Valor de número entero →

Bit de almacenamiento →

Contador →

Temporizador →

Dispositivo de E/S
digital →

Program Tags - MainProgram				
Scope: MainProgram		Show: Show All		Sort: Tag Name
Tag Name	Alias For	Base Tag	Type	
north_tank_mix			BOOL	
north_tank_pressure			REAL	
north_tank_temp			REAL	
+one_shots			DINT	
+recipe			TANK[3]	
+recipe_number			DINT	
replace_bit			BOOL	
+running_hours			COUNTER	
+running_seconds			TIMER	
start			BOOL	
stop			BOOL	

PLC:

Tipo de datos

El tipo de tag define cómo funciona el tag dentro del proyecto.

Si desea que el tag	Entonces seleccione este tipo
Almacene uno o varios valores para que la lógica los use en el proyecto	Base
Represente otro tag	Alias
Envíe datos a otro controlador	Producido
Reciba datos de otro controlador	Consumido

Si prevé que va a utilizar tags producidos o consumidos, debe seguir pautas adicionales al organizar los tags.

Tipos de datos

Término	Definición
Tipo de datos	El tipo de datos que el tag almacena, como bit, número entero, valor con punto flotante, cadena, etc.
Estructura	<p>Un tipo de datos que es una combinación de otros tipos de datos.</p> <ul style="list-style-type: none"> • Una estructura se formatea para crear un tipo de datos único que coincida con una necesidad específica. • Dentro de una estructura, cada tipo de datos individual se conoce como miembro. • Al igual que los tags, los miembros tienen un nombre y un tipo de datos. • Todos los controladores Logix contienen estructuras predefinidas (tipos de datos) que deben utilizarse con instrucciones específicas como temporizadores, contadores, bloques de funciones, etc. • Puede crear sus propias estructuras, denominadas tipos de datos definidos por el usuario.

PLC:

En la siguiente tabla se presentan los tipos de datos más comunes y cuándo hay que usar cada uno.

Para	Seleccione
Dispositivo analógico en modo con punto flotante	REAL
Dispositivo analógico en modo con número entero (para frecuencias de muestreo muy rápidas)	INT
Caracteres ASCII	Cadena
Bit	BOOL
Contador	CONTADOR
Punto de E/S digital	BOOL
Número con punto flotante	REAL
Número entero	DINT
Secuenciador	CONTROL
Temporizador	TIMER

La asignación mínima de memoria para un tag es de cuatro bytes. Cuando se crea un tag con datos que requieren menos de 4 bytes, el controlador asigna 4 bytes pero los datos solo ocupan la parte que necesitan.

Tipo de datos	Bits						
	31	16	15	8	7	1	0
BOOL	no se usa						0 o 1
SINT	no se usa				−128...+127		
INT	no se usa		−32,768...+32,767				
DINT	−2,147,483,648...+2,147,483,647						
REAL	−3.40282347E ³⁸ ... −1.17549435E ^{−38} (valores negativos)						
	0						
	1.17549435E ^{−38} ... 3.40282347E ³⁸ (valores positivos)						

PLC:

Los tipos de datos del CONTADOR y TEMPORIZADOR son ejemplos de estructuras usadas habitualmente.

Para expandir una estructura y visualizar sus miembros, haga clic en el símbolo +.

Para contraer una estructura y ocultar sus miembros, haga clic en el símbolo -.

Miembros de running_seconds

Program Tags - MainProgram				
Scope:	MainProgram	Show:	Show All	Sort: T.
Tag Name	Alias For	Base Tag	Type	
+ -running_hours			COUNTER	
- -running_seconds			TIMER	
+ -running_seconds.PRE			DINT	
+ -running_seconds.ACC			DINT	
- running_seconds.EN			BOOL	
- running_seconds.TT			BOOL	
- running_seconds.DN			BOOL	
- running_seconds.FS			BOOL	
- running_seconds.LS			BOOL	
- running_seconds.OV			BOOL	
- running_seconds.ER			BOOL	

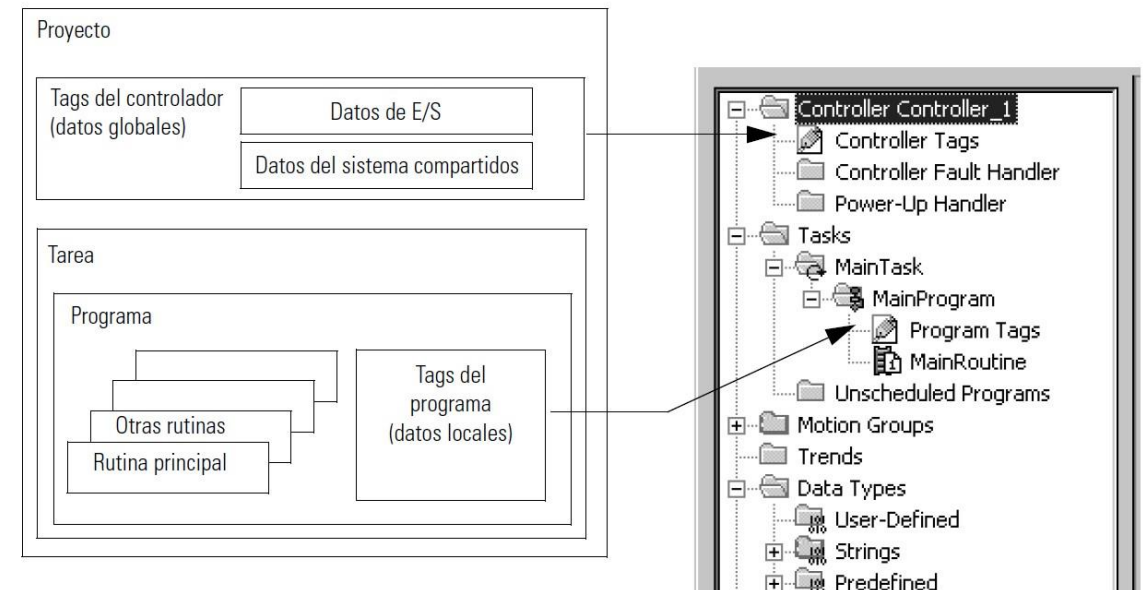
Estructura de CONTADOR

Estructura de TEMPORIZADOR

Miembros de tipos de datos

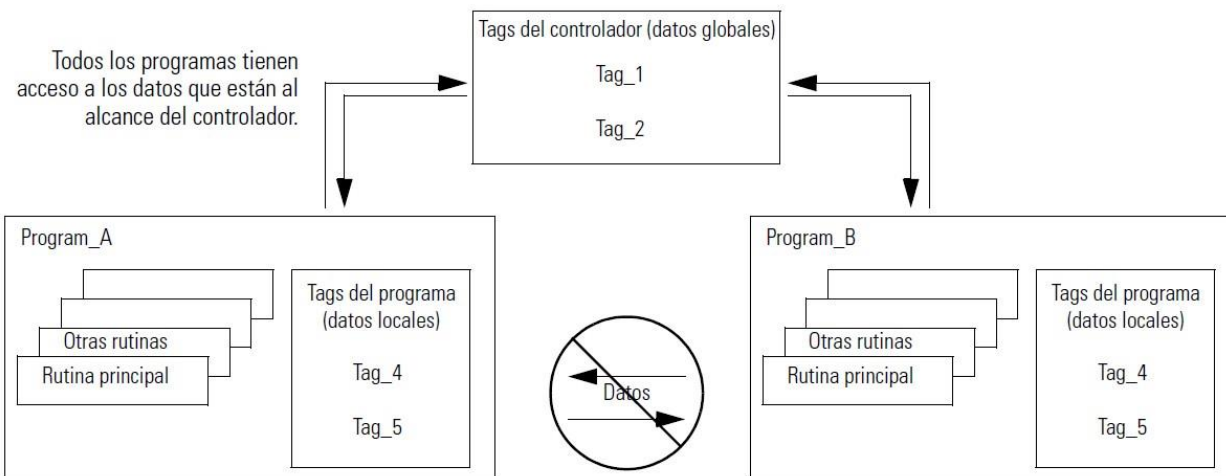
Alcance

Cuando crea un tag, lo tiene que definir como tag del controlador (datos globales) o como tag del programa para un programa específico (datos locales).



PLC:

Un controlador Logix5000 le permite dividir la aplicación en múltiples programas, cada uno con sus propios datos. No es necesario gestionar nombres de tags conflictivos entre programas. Esto facilita la reutilización de nombres tanto de códigos como de tags en múltiples programas.



Los datos de alcance del programa se aíslan de otros programas.

- Las rutinas no pueden acceder a los datos que están al alcance de otro programa.
- Puede reutilizar el nombre de un tag aislado por un programa en varios programas.
Por ejemplo, Program_A y Program_B pueden tener un tag de programa denominado Tag_4.

Evite usar el mismo nombre para un tag de controlador y un tag de programa. Dentro de un programa no es posible hacer referencia a un tag de controlador si existe un tag con el mismo nombre.

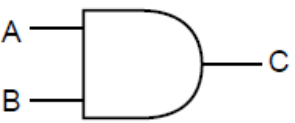
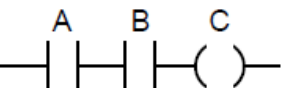
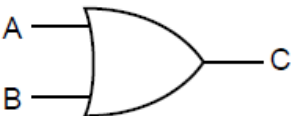
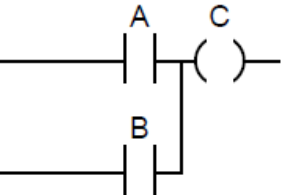
Determinados tags deben ser de alcance del controlador (tag del controlador).

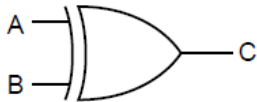
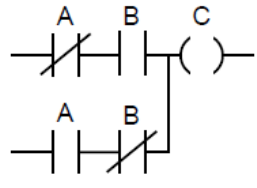

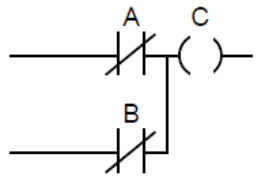
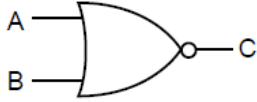
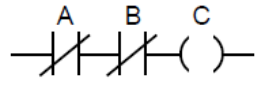
Tags al alcance del controlador

Si desea utilizar el tag	Entonces asigne este alcance
En más de un programa del proyecto	Alcance de controlador (tags de controlador)
En una instrucción Message (MSG)	
Para producir o consumir datos	
En cualquiera de los siete tipos de datos AXIS	
Para comunicarse con un terminal PanelView	Alcance del programa (tags de programa)
En ningún caso de los anteriores	

PLC:

Diagramas lógicos y su equivalencia en Ladder

Logic Diagram	Truth Table	Ladder Diagram															
 <p>AND Gate</p>	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	0	0	0	0	1	0	1	0	0	1	1	1	 <p>AND Equivalent Circuit</p>
A	B	C															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
 <p>OR Gate</p>	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	0	0	0	0	1	1	1	0	1	1	1	1	 <p>OR Equivalent Circuit</p>
A	B	C															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

 <p>Exclusive-OR Gate</p>	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	C	0	0	0	0	1	1	1	0	1	1	1	0	 <p>Exclusive-OR Equivalent Circuit</p>
A	B	C															
0	0	0															
0	1	1															
1	0	1															
1	1	0															
 <p>NAND Gate</p>	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	C	0	0	1	0	1	1	1	0	1	1	1	0	 <p>NAND Equivalent Circuit</p>
A	B	C															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
 <p>NOR Gate</p>	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	C	0	0	1	0	1	0	1	0	0	1	1	0	 <p>NOR Equivalent Circuit</p>
A	B	C															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

PLC:

STUDIO 5000

El Studio 5000 Automation Engineering C Design Environment® combina los elementos de ingeniería y diseño en un entorno común. El primer elemento es la aplicación Studio 5000 Logix Designer®. La aplicación Logix Designer es el nuevo nombre de marca asignado al software RSLogix 5000® y continuará siendo el producto para programar los controladores LOGIX 5000™ en soluciones discretas, de proceso, de lotes, de movimiento, de seguridad y basadas en variadores.



PLC:

¿Qué es RSLinx Classic?

Permite que el controlador programable Allen-Bradley acceda a una amplia variedad de aplicaciones de Rockwell Software y Allen-Bradley. Entre estas aplicaciones se incluyen desde aplicaciones de configuración y programación tales como RSLogix y RSNetWorx hasta aplicaciones HMI (interfaz operador-máquina) como RSView32, hasta sus propias aplicaciones de adquisición de datos mediante Microsoft Office, páginas Web o Visual Basic®. Además, RSLinx Classic utiliza técnicas de optimización de datos avanzadas y dispone de una serie de diagnósticos. La interfaz de programación de aplicaciones (API) admite aplicaciones personalizadas creadas con RSLinx Classic SDK. RSLinx Classic es un servidor compatible con OPC Data Access y un servidor DDE.

Factory Linx

Esta solución, antes conocida como RSLinx Enterprise, es el servicio de comunicación central en el portafolio de software FactoryTalk. Actúa como el centro de una empresa conectada al enviar datos donde se necesita. Esto incluye extraer datos de servidores OPC UA de terceros, entregarlos al software FactoryTalk y proporcionar una interfaz OPC para que el software de terceros pueda acceder a los datos.