

11

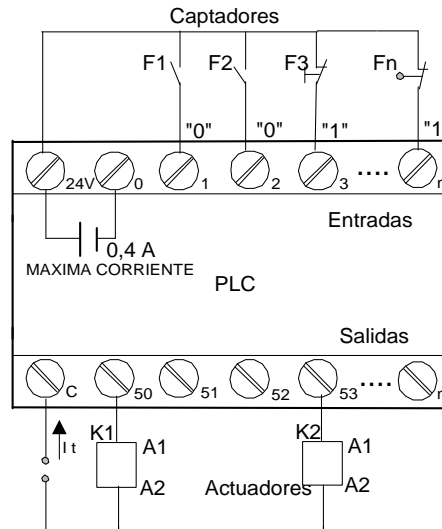
Sistemas de mando. Norma IEC 61131.- Programación del PLC en AWL y FBD.

CONEXIONADO DE ENTRADAS Y SALIDAS

Entradas

- Analógicas, cuya señal es variable en el tiempo y que necesariamente han de acoplarse al mismo tipo de entradas.
- Digitales, en donde la señal responde a:
 - Contacto abierto "0" (nada)
 - Contacto cerrado "1" (todo)

Cuando el contacto está cerrado sucede que la batería alimentará al elemento interno lo que desencadena una señal hacia el circuito de control de entrada del PLC. Si el contacto está abierto no ocasiona ningún efecto; en todo caso *el tipo de señal del captador debe ser igual al tipo de entrada, si no fuera así DEBE INSTALARSE EN LA ENTRADA ANALÓGICO/DIGITAL PARA SU CONVERSIÓN*, pues será la única forma de que el procesador entienda dicha señal.



Debe verificarse el tipo de captador (activo o inactivo), esto es con tensión o sin tensión propia.

- Captadores libres de tensión: pulsadores, interruptores, finales de carrera, contactos de relés, etc
- Captadores con tensión: Detectores de proximidad, de nivel, células fotoeléctricas, etc. En este caso verificar que su *tensión de trabajo coincida con la tensión de entrada* al autómata. Cuando el *requerimiento de intensidad sea mayor a dicha fuente interna* es necesario colocar en paralelo a fin de suplir dicha carencia.

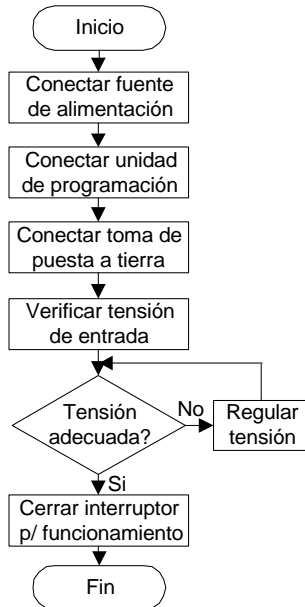
Salidas

Las cargas o actuadores se conectan a las salidas que pueden ser:

- Salidas a relés (c.a. o c.c.) Este tipo de salida suele utilizarse cuando el consumo tiene cierto valor (del orden de amperios) y donde las conmutaciones no son demasiado rápidas. Son empleados en cargas con contactores, electroválvulas, etc.
- Salidas a triacs (c.a. o c.c.) Se usa cuando las conmutaciones son más rápidas, la intensidad de corriente es similar al del relé.
- Salidas a transistores (c.c.) Cuando se usa c.c. y la corriente es pequeña.

Se debe tener en cuenta la tensión para los actuadores que pueden ser distintas para cada grupo y debe estar en el rango que indique el fabricante.

La corriente debe ser sumada y esta no debe sobrepasar la intensidad máxima que nos indique sus características de placa.



Para poner en funcionamiento el PLC seguiremos los pasos indicados en el flujograma de la figura, luego de tener delante de nosotros el cuadro de características o especificaciones del PLC con el vamos a trabajar:

Para iniciar la programación deberemos considerar que es necesario familiarizarse con los *modos y funciones* específicas mediante el manual de instrucciones, teniendo en cuenta que de un autómata a otro aún de la misma marca pueden haber versiones, tanto del hardware como del software.

Como se sabe, la ventaja de los autómatas sobre lo cableado es que se puede *introducir, borrar y modificar los programas*, así como grabarlos.

Esta posibilidad debe ser aprovechada para grabar los archivos en cualquiera de sus modalidades, del mismo modo realizar un hard copy mediante la impresora para su posterior utilización, modificación, actualización, etc. Obviamente el sistema magnético es importante si

se quiere volver a poner en funcionamiento un programa ya utilizado con anterioridad e incluso para realizar alguna modificación sobre él.

Sin embargo antes de ejecutar el programa es necesario realizar las conexiones de los elementos primarios o sensores captadores así como las conexiones de los actuadores mediante las salidas del PLC. La eficaz puesta en funcionamiento depende de un adecuado conexionado que evitará averías y racional uso de las entradas y salidas del autómata.

PROGRAMACIÓN DE UN PLC

Los PLC pueden ser programados para una aplicación o máquina específica que quiera controlarse ya sea trabajando en forma independiente o como parte de un sistema modular. La programación puede realizarse mediante diversos equipos denominados "maleta o terminal de programación". El terminal de programación está diseñado para trabajar en ambientes industriales y consiste en pequeños teclados o paneles de control que tienen, básicamente, teclas alfanuméricas, teclas de función, pantalla de visualización (normalmente de pocas líneas), salida para el PLC, salida para una impresora, salida para una casetera o disk drive para archivar programas en audio-cassetts o floppy disks. Por otro lado, existen paquetes de software que posibilitan la programación de un PLC mediante una PC IBM o compatible, sin embargo si la configuración no es adecuada pueden aparecer dificultades en el desempeño de la PC por las exigencias de una planta industrial.

Es común la programación del PLC mediante una técnica de programación simbólica que simula el circuito eléctrico de relés, aquí un diagrama de este tipo es convertido, previo scan, en un diagrama para el PLC codificando la apertura y cierre de los contactos, activación de los temporizadores, nudos o ramales del circuito, etc.; estos circuitos pueden ser fácilmente modificados totalmente o por partes para reprogramar el PLC.

Las funciones principales de las unidades de programación son:

a- Programación

- Introducción de instrucciones
- Búsqueda de instrucciones o posiciones de memoria
- Modificación del programa: Borrado; inserción; modificación
- Detección de errores de sintaxis o formato
- Visualización del programa o parte de él contenido en la memoria del usuario
- Forzamiento del estado de marcas, registros, contadores, temporizadores, etc.

b- Grabación de programas

- En cinta
- En chip de memoria EPROM o EEPROM
- En papel mediante impresora
- En disquet

c- Visualización y verificación dinámica del programa

- Del programa o parte del mismo
- De entradas y salidas
- De temporizadores, contadores, registros, etc.

d- Modos de servicio

- STOP (off-line), o salidas en reposo
- RUN (on-line), o ejecutando el programa
- Otros modos intermedios como monitor, etc.

Las unidades pueden ser del tipo calculadora, consola o trabajar con la computadora personal.

LA NORMA IEC 61131-3

La norma IEC 61131-3 estandariza la programación del PLC ante el incremento de la complejidad de los procesos a controlarse mediante los autómatas programables y la diversidad de lenguajes que se utilizan para cada una de ellas. Si bien es cierto que mediante un programa puede prepararse a varios PLCs es necesario que se estandarice los conceptos y procedimientos.

IEC 61131-3 (Lenguajes de programación) es la tercera parte de la familia **IEC 61131**, la cual consiste de:

- 1.- Vista general
- 2.- Hardware
- 3.- Lenguajes de programación
- 4.- Directrices al usuario
- 5.- Comunicación

La parte 3 dedicada a los lenguajes engloba, en más de 200 páginas de texto y 60 tablas, la especificación de la sintaxis y semántica de una suite unificada del lenguaje de programación. Dentro de esta norma son definidos cinco lenguajes de programación, a saber:

- LISTA DE INSTRUCCIONES
- TEXTO ESTRUCTURADO
- DIAGRAMA DE CONTACTOS
- DIAGRAMA DE FUNCIONES
- GRAFICO SECUENCIAL DE FUNCIONES, SFC O GRAFCET

Los lenguajes mencionados describen un mismo programa y la selección a ser usado depende de la preparación del programador, el problema a resolver, el nivel de descripción del problema, la estructura del sistema de control y la interfaz con otro personal o con otros departamentos.

LISTA DE INSTRUCCIONES (IL o AWL)

Su denominación en Ingles es *Instruction List*, *IL* en el Alemán se conoce como *Anweisungsliste*, *AWL*.

Es un lenguaje textual de bajo nivel, que se asemeja al lenguaje ensamblador y que permite almacenar un valor en una línea. Este lenguaje es adecuado para pequeñas aplicaciones y para optimizar partes de una aplicación.

Ejemplo:

LD	A
ANDN	B
ST	C

TEXTO ESTRUCTURADO (structured text o ST)

En Ingles es *Structured Text*, *ST* en Alemán se conoce como *Strukturierter Text*, *ST*.

Es un lenguaje de alto nivel estructurado por bloques que posee una sintaxis parecida al Pascal y C. Este lenguaje puede ser usado para realizar rápidamente sentencias complejas que manejen variables con un amplio rango de diferentes tipos de datos, incluyendo valores análogos o digitales. También se especifica tipos de datos para el manejo de horas, fechas y temporizaciones, algo importante en procesos industriales.

El lenguaje posee soporte para bucles iterantes como REPEAT UNTIL, ejecuciones condicionales empleando sentencias IF-THEN-ELSE y funciones como SQRT() y SIN()

Ejemplo:

C:=A AND NOT B

DIAGRAMA DE CONTACTOS (Ladder diagram LD o KOP)

Conocido también como diagrama de escalera, en Ingles como *Ladder Diagram*, *LD* y en Alemán como *Kontaktplan*, *KOP*.

Es un lenguaje que utiliza un juego estandarizado de símbolos de programación y que inicialmente se basó en una representación gráfica de la lógica de escalera por relevadores. En el estándar IEC los símbolos han sido racionalizados, reduciéndose el número de ellos.

Ejemplo:

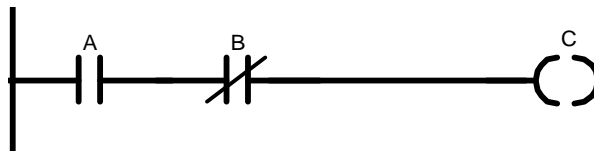


DIAGRAMA DE FUNCIONES (FBD)

En Ingles es *Function Block Diagram*, *FBD* en Alemán se conoce como *Funktionsbausteinsprache*, *FBS*.

Es un lenguaje gráfico que permite programar elementos que aparecen como bloques para ser cableados entre si de forma análoga al esquema de un circuito. FBD es adecuado para muchas aplicaciones que involucren el flujo de información o datos entre componentes de control. Permite interrelacionar con otros dispositivos similares tales como las compuertas lógicas.

Ejemplo:

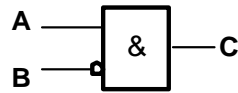


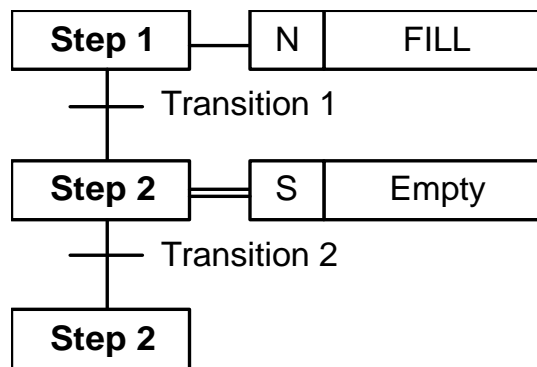
GRAFICO SECUENCIAL DE FUNCIONES (GRAFCET o SFC)

El Grafico secuencial de funciones se denomina en ingles como *Sequential Function Chart, SFC* y en alemán como *Ablaufsprache, AS*.

Es un lenguaje gráfico que proporciona una representación en forma de diagramas de las secuencias del programa. Soporta selecciones alternativas de secuencia y secuencias paralelas. Los elementos básicos son etapas y transiciones. Las etapas consisten de piezas de programa que son inhibidas hasta que una condición especificada por las transiciones es conocida.

Como consecuencia de que las aplicaciones industriales funcionan en forma de pasos o etapas, el SFC es la forma lógica de especificar y programar el mas alto nivel de un programa para PLC.

Ejemplo:



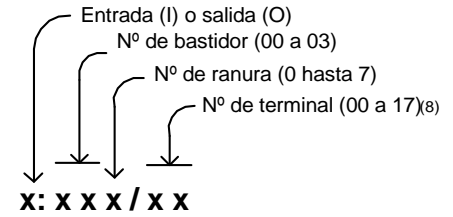
Estas técnicas, como todas, permiten la elaboración y depuración del programa así como la detección de posibles fallas en los sensores y actuadores o autodiagnosticar su funcionamiento tanto en hardware o software.

INSTRUCCIONES Y PROGRAMAS

Asignar direcciones de entrada y de salida

El usuario debe especificar la dirección completa en los archivos de imágenes de E/S a fin de identificar un bit individual en particular de la memoria. Para especificar una dirección se requieren dos partes: La dirección de palabra y el número de bit en la palabra; una dirección de palabra de entrada o salida tiene tres dígitos octales; el número de bit de E/S tiene dos dígitos octales. Como ejemplo, a continuación se visualiza la tabla de memoria para un PLC 5/12 de Allen-Bradley.

Secciones de la memoria del procesador		Rango de direcciones que pueden continuar a los caracteres iniciales	
Archivo de imágenes de salida. Carácter inicial - O:	000/00 037/17	Números octales	
Archivo de imágenes de entrada. Carácter inicial - I:	000/00 037/17	Números octales	
Estado del procesador momento a momento. Carácter inicial - S:	0 31	Números decimales	
Memoria de datos variables			
Subsección (o archivo) de bit binario aislado. Caracteres iniciales - B3/	0 999	Números decimales	
Subsección (o archivo) del temporizador. Caracteres iniciales - T4:	0 999	Números decimales	
Subsección del contador. Caracteres iniciales - C5:	0 999	Números decimales	
Subsección de valores enteros. Caracteres iniciales - N7:	0 999	Números decimales	
Subsección de valores de punto flotante. Caracteres iniciales - F8:	0 999	Números decimales	



Ejemplos:

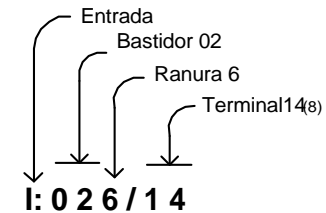
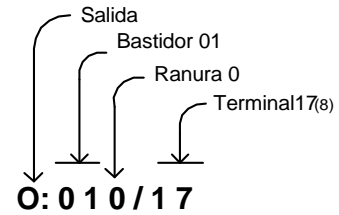


TABLA DE MEMORIA PARA PLC 5/12 DE ALLEN-BRADLEY

Este direccionamiento se puede visualizar como la asignación en una tabla similar a la siguiente:

		BASTIDOR nn							 BASTIDOR 03
		Ranura								
		00	01	02	03	04	05	06	07	
Terminal de Ranura	00	00	00	00	00	00	00	00	00	
	01	01	01	01	01	01	01	01	01	
	02	02	02	02	02	02	02	02	02	
	03	03	03	03	03	03	03	03	03	
	04	04	04	04	04	04	04	04	04	
	05	05	05	05	05	05	05	05	05	
	06	06	06	06	06	06	06	06	06	
	07	07	07	07	07	07	07	07	07	
	10	10	10	10	10	10	10	10	10	
	11	11	11	11	11	11	11	11	11	
	12	12	12	12	12	12	12	12	12	
	13	13	13	13	13	13	13	13	13	
	14	14	14	14	14	14	14	14	14	
	15	15	15	15	15	15	15	15	15	
	16	16	16	16	16	16	16	16	16	
	17	17	17	17	17	17	17	17	17	

Dicho de otro modo, los archivos contienen la información de estado asociada con las entradas y salidas externas y otras instrucciones que se usan en los archivos de programa escalera principal y de subrutina y como se indica en la tabla de memorias; los tipos de archivos, para la familia SLC de Allen-Bradley, son:

- **SALIDA (Archivo 0)** Almacena el estado de los terminales de salida para el controlador.
- **ENTRADA (Archivo 1)** Almacena el estado de los terminales de entrada para el controlador.
- **ESTADO (Archivo 2)** Almacena la información de operación del controlador y es útil para la localización y corrección de fallos y la operación del programa.
- **BIT (Archivo 3)** Se usa para el almacenamiento de la lógica del relé interno.
- **TEMPORIZADOR (Archivo 4)** Almacena los valores acumulados y predefinidos del temporizador y bits de estado.
- **CONTADOR (Archivo 5)** Almacena los valores acumulados y predefinidos del temporizador y bits de estado.
- **CONTROL (Archivo 6)** Almacena la longitud, posición del puntero y bits de estado para instrucciones específicas tales como registro de desplazamiento y secuenciadores.
- **ENTEROS (Archivo 7)** Se usa para almacenar valores numéricos o información de bits.

En conclusión, se puede tabular como sigue:

TIPO DE ARCHIVO	IDENTIFICADOR	Nº ARCHIVO
Salida	O	0
Entrada	I	1
Estado	S	2
Bit	B	3
Temporizador	T	4
Contador	C	5
Control	R	6
Enteros	N	7

Los delimitadores usados incluyen el signo de dos puntos (:), el signo diagonal (/) y el punto (.). Como por ejemplo: I1:0/4 O0:0/2 T4:3/DN etc. Para el caso de los temporizadores se deberán especificar la dirección, la base de tiempo, el tiempo preseleccionado y el acumulado.

Quando se ejecuta el programa se “barre” con todas las instrucciones y se distinguen los siguientes modos de estructuras de programación:

- Ejecución cíclica lineal, cuando el barrido se realiza línea por línea, sin alterar el orden.
- Ejecución con salto condicional, cuando el programa da salto a otra línea preestablecida.
- Ejecución con salto a subrutinas, cuando saltan a una secuencia o rutina que siempre se repite y culminado dicha rutina vuelven a la línea siguiente para seguir corriendo el programa.
- Ejecución con programas paralelos, cuando se ejecuta el programa controlando varios procesos totalmente independientes.

Ingreso del programa

El procesador debe pasarse al modo programación.

Estando en este modo debe verificarse el software con el que se está trabajando ya que según ello se tendrá menor o mayor facilidad para operar según sea el caso para un programa conforme a las normas IEC 61131-3 normalmente el especialista en electricidad, preferirá trabajar con el KOP o ladder por la similitud de sus trabajos mediante el diagrama de contactos.

En todo caso debe visualizar las teclas de funciones que insertan los contactos normalmente cerrados, normalmente abiertos, las salidas, las bifurcaciones paralelas, las instrucciones de temporización, conteo u otro; según sea el caso el programa podrá estar solicitando las direcciones que inicialmente se asignaron a las entradas y salidas para que vaya verificando la lógica con que se está trabajando; si encontrase una acción ilógica no ingresará el dato y puede aparecer un mensaje

sugiriendo que debe hacer o si la interfaz hombre máquina es muy buena, aparecerán todas las posibilidades para que seleccione el usuario. A continuación se insertará un nuevo escalón para ir insertando cada uno de los componentes de dicho escalón.

Edición del programa

Aun cuando se estuvo verificando cada escalón, es probable que cuando se ejecute el programa aparezcan errores conceptuales o quizá errores tipográficos, en ese caso debemos pasar al modo edición del programa y a medida que se vaya corrigiendo los errores se estará indicando cuantos de ellos fueron corregidos e indicarán los casos que aún faltan corregirse. A medida que se adquiera práctica en el manejo de la programación y en concordancia con los manuales del equipo se estarán culminando con mayor rapidez.

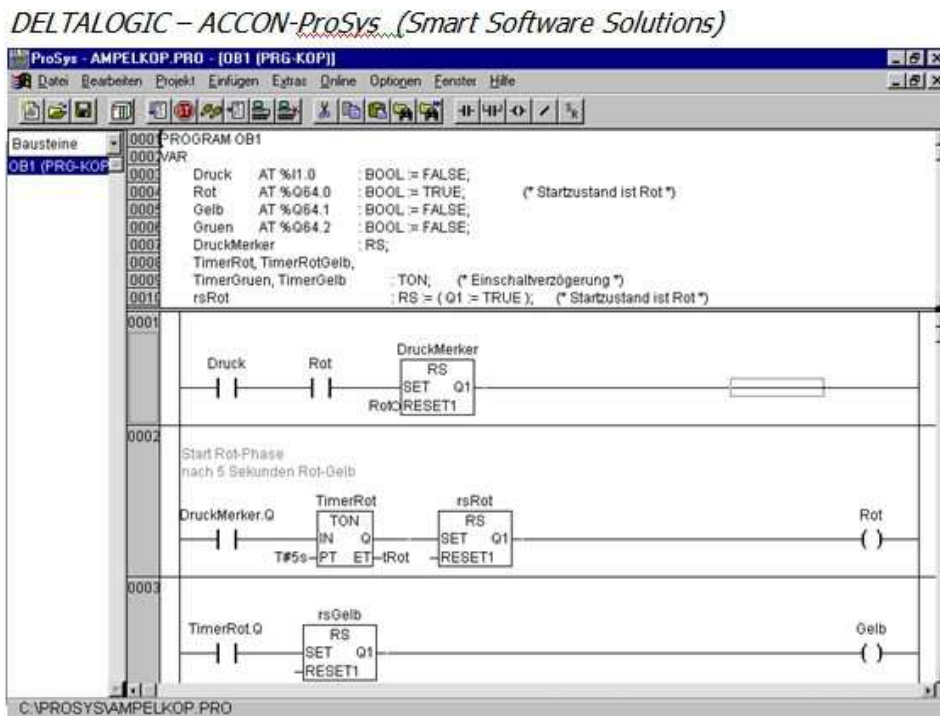
Prueba del programa

Los editores de programas siempre consideran que es necesario verificar que el programa realmente está libre de errores y puede ejecutarse sin problemas. Por ello cuentan con un modo de Prueba de Programa o en otros se denominan simulación de tal modo que el programa se ejecute como si estuviese trabajando industrialmente, en este test pueden aparecer aún problemas dentro del programa, sin embargo los problemas que con mayor frecuencia aparecen son los relacionados con los periféricos que deben configurarse adecuadamente para un interfaceamiento pleno.

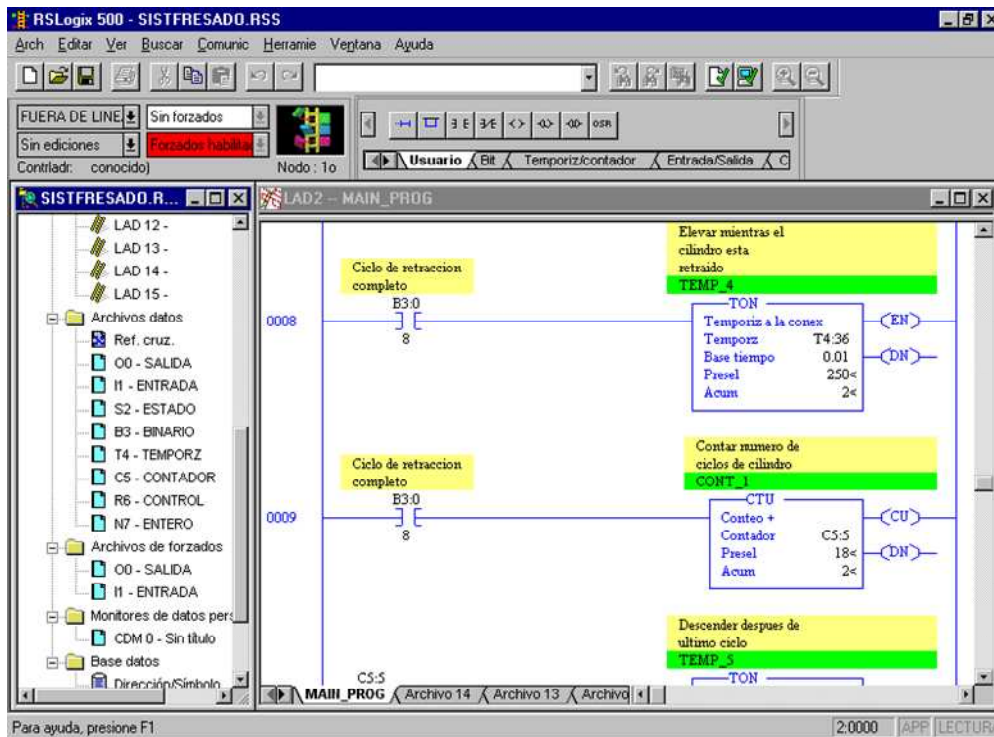
Cuando se ha culminado satisfactoriamente con la prueba se conmutará el procesador al modo EJECUCIÓN o RUN.

En la actualidad existe una gran diversidad de programas y soluciones informáticas que permiten desarrollar fácilmente programas y secuencias para los PLC considerando que es uno de los elementos principales en el sistema de automatización, existe tal diversidad de software que en la red INTERNET puede encontrarse variedad, siendo algunos de características de demostración así como otros en calidad de prueba. Los hay exclusivamente para la familia de una marca así como compatibles con otras marcas y últimamente están adecuados a la norma IEC 61131 de tal modo que puede intercambiarse información de una familia de marcas a otras.

EJEMPLO DE LAS PANTALLAS DE SOFTWARE PARA PROGRAMAR PLC



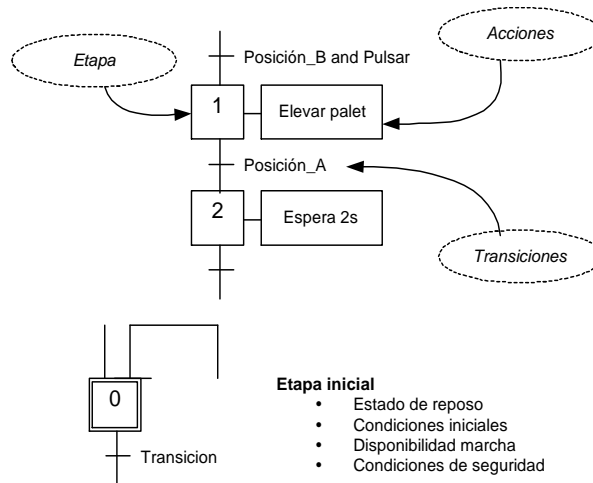
RSLogix500 (Rockwell Software) Para PLC de la familia SLC500 de Allen-Bradley



DESARROLLANDO EL SFC DEL ESTANDAR DE PROGRAMACIÓN – IEC 61131-3

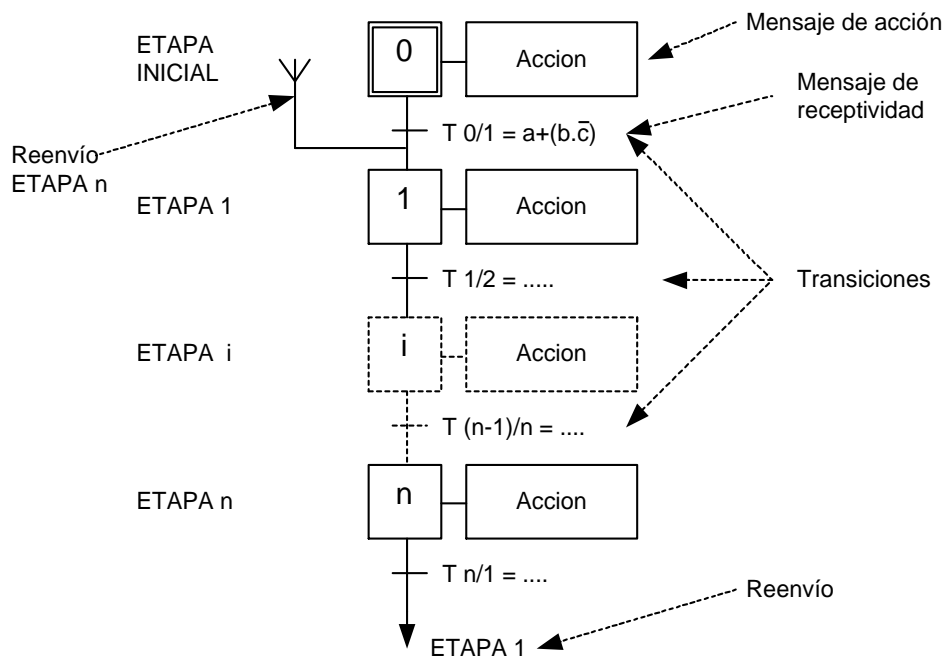
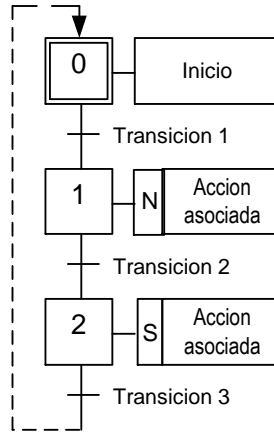
El Graphe de Comande Etapa-Transition o GRAFICO Funcional de Control de Etapas y Transiciones - GRAFCET

A nivel de estándar es el grafico secuencial de funciones (SFC) es un lenguaje grafico que proporciona una representación en forma de diagrama de las secuencia y secuencias paralelas. Los elementos básicos son etapas y transiciones. Las etapas consisten de partes de programa que son inhibidas hasta que se da una condición especificada por las transiciones.



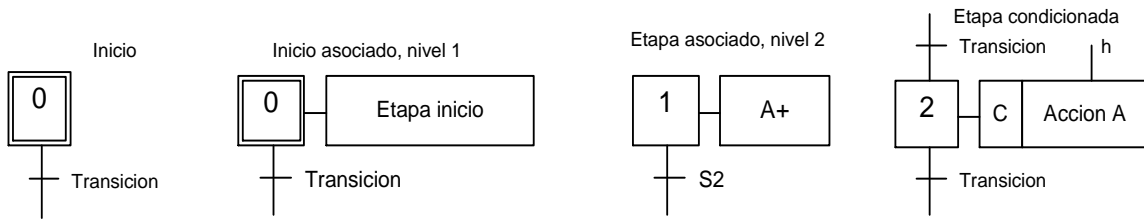
GRACET: ETAPAS Y TRANSICIONES. ETAPA DE INICIO
 Adaptado de: Desarrollo de Sistemas Secuenciales. Rodríguez Mata

Gráficos Funcionales Secuenciales (SFC)

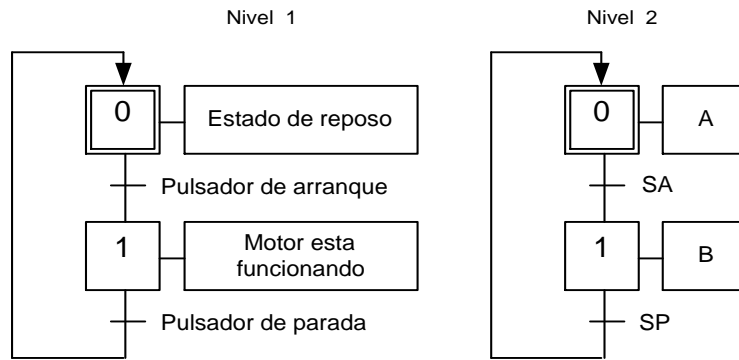


ELEMENTOS GRAFICOS DEL GRAFCET

Adaptado de Automatas Programables, Balcells Josep



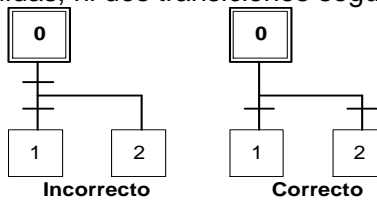
Ejemplo:



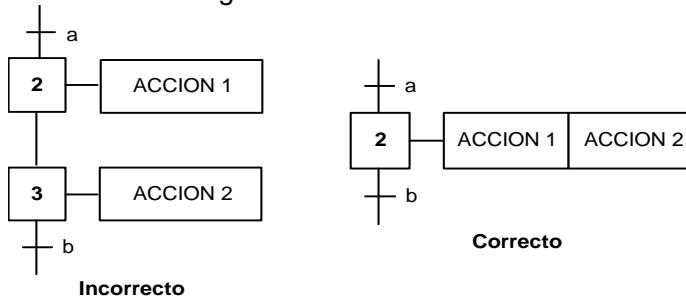
COMANDO DE UN MOTOR ELECTRICO

REGLAS FUNDAMENTALES PARA PROGRAMAR CON EL GRAFCET

REGLA 1. El diagrama se dibuja con una sucesión alternada de etapas y transiciones. No puede haber dos etapas seguidas, ni dos transiciones seguidas.

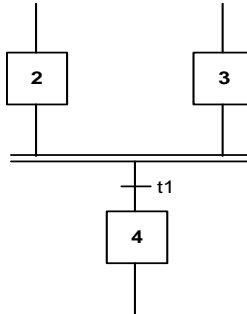


REGLA 2. Al superar una transición y debe hacerse dos acciones simultaneas, debe utilizarse una etapa con dos acciones sucesivas. No puede haber dos etapas seguidas ni dos transiciones seguidas.

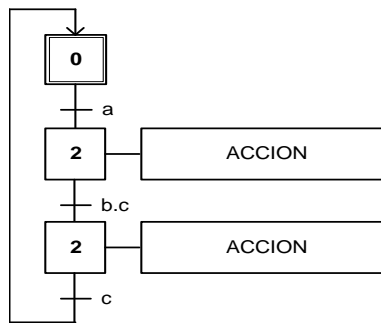


REGLA 3. la primera transición se podrá validar si la etapa anterior esta activa y además cumple lo establecido en la transición. En este momento deja de estar activa la etapa anterior y le toma el relevo la etapa siguiente. Para cada etapa se repite esta regla.

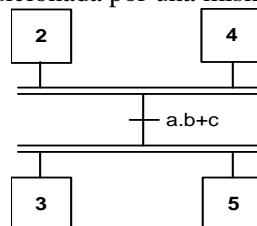
REGLA 4. Las etapas 2 y 3 son etapas de entrada a la transición. Para validar la transición, deben estar activas las dos etapas. Para pasar a la etapa 4 la transición tiene que estar validada y se debe cumplir la receptividad asociada $t1$



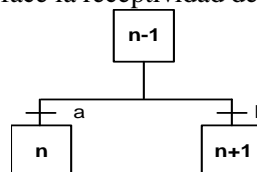
REGLA 5. Un GRAFCET será de secuencia única, cuando en el diagrama solo hay una sola rama; el conjunto de etapas se irán activando una tras la otra, después de validarse las recepciones asociadas a las transiciones.



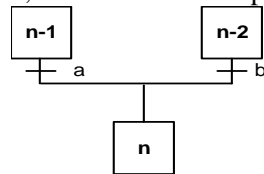
TRAZOS PARALELOS. Se utilizan para representar varias etapas cuya evolución está condicionada por una misma transición.



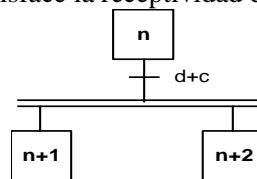
DIVERGENCIA EN OR. La etapa n pasa a ser activa la etapa $n-1$, se satisface la receptividad de la transición a . La etapa $n+1$ pasa a ser activa si, activa la etapa $n-1$, se satisface la receptividad de la transición b .



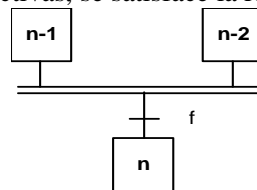
CONVERGENCIA EN OR. La etapa n pasa a ser activa si, estando activa la etapa $n-1$, se satisface la receptividad de la transición a ; o si estando activa la etapa $n-2$, se satisface la receptividad de la transición b .



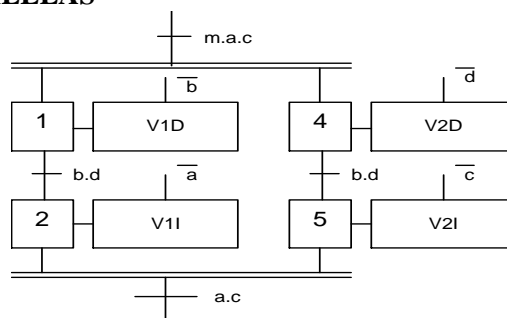
DIVERGENCIA EN AND. Las etapas $n+1$ y $n+2$ pasan al estado activo, si estando activa n se satisface la receptividad de la transición t cuya receptividad es $d+c$.



CONVERGENCIA EN AND. La etapa n pasa al estado activo, si estando las etapas $n-1$ y $n-2$ activas, se satisface la receptividad f .



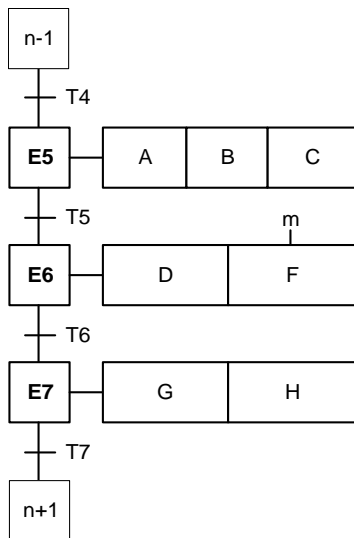
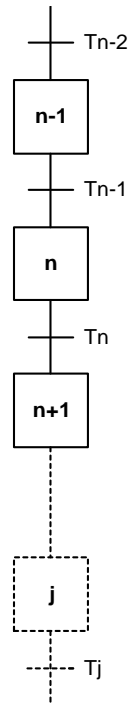
SECUENCIAS PARALELAS



PARA GENERAR ECUACIONES Y PROGRAMAR EL PLC

La activación o desactivación asociada a la etapa enésima será:

$$E_n = E_{n-1}T_{n-1} + \bar{E}_{n+1}E_n$$



Las ecuaciones por cada etapa

$$E_5 = E_4T_4 + E_5\bar{E}_6$$

$$A = E_5$$

$$B = E_5$$

$$C = E_5$$

$$E_6 = E_5T_5 + E_6\bar{E}_7$$

$$D = E_6$$

$$F = mE_6$$

$$E_7 = E_6T_6 + E_7\bar{E}_8$$

$$G = E_7$$

$$H = E_7$$

Con las ecuaciones halladas elaborar el KOP en el programador del PLC.

PROGRAMANDO UN AUTOMATA PROGRAMABLE DESDE EL GRAFCET

GRAFCET

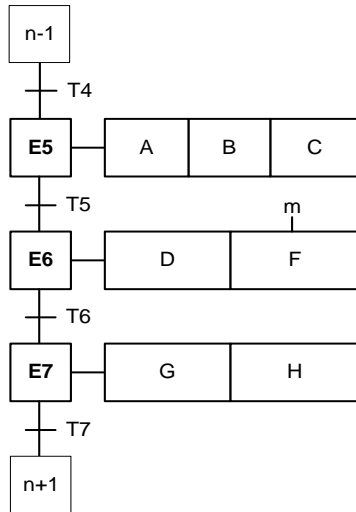
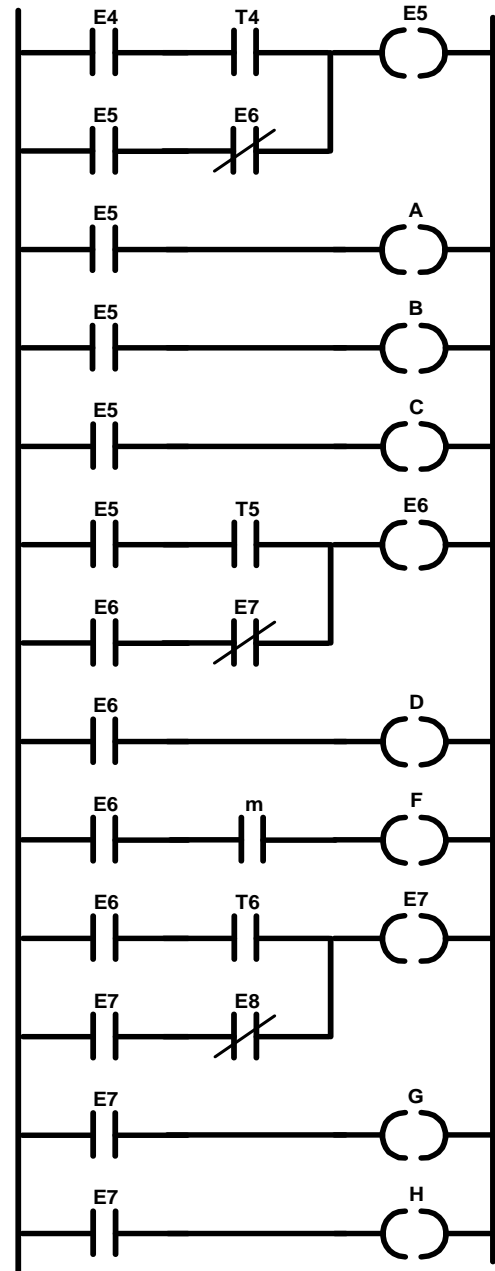


DIAGRAMA DE CONTACTOS



Las ecuaciones por cada etapa

$$E_5 = E_4 T_4 + E_5 \bar{E}_6$$

$$A = E_5$$

$$B = E_5$$

$$C = E_5$$

$$E_6 = E_5 T_5 + E_6 \bar{E}_7$$

$$D = E_6$$

$$F = m E_6$$

$$E_7 = E_6 T_6 + E_7 \bar{E}_8$$

$$G = E_7$$

$$H = E_7$$

