

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA
DIVISION DE ESTUDIOS DE POST-GRADO



PROGRAMACION DE PLC'S

POR:
ING. PATRICIA AGUILERA MARTINEZ

T E S I S

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS
DE LA INGENIERIA DE MANUFACTURA CON
ESPECIALIDAD EN AUTOMATIZACION

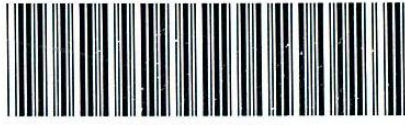
SAN NICOLAS DE LOS GARZA, N. L., JUNIO DE 2002

TM
Z5853
.M2
FIME
2002
.A48

P.A.

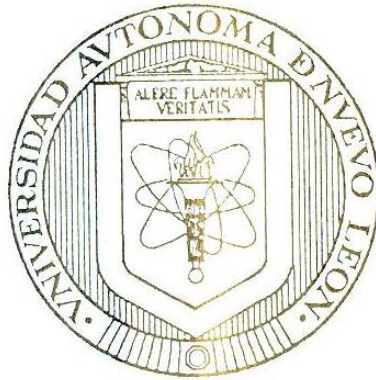
PROCERAMACION DE PLCS

2002



1020148252

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POST-GRADO



PROGRAMACIÓN DE PLC'S

POR

ING. PATRICIA AGUILERA MARTÍNEZ

TESIS

**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS DE LA
INGENIERÍA DE MANUFACTURA CON ESPECIALIDAD EN AUTOMATIZACIÓN**

SAÑ NICOLÁS DE LOS GARZA, N. L.

JUNIO DEL 2002

970 978

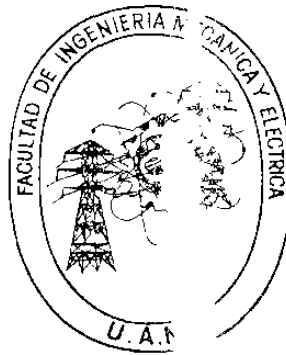
TM
25853

.M2
TIME
2002
.A 48



FONDO
TESIS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POST-GRADO



PROGRAMACIÓN DE PLC'S

POR

ING. PATRICIA AGUIRERA MARTÍNEZ

TESIS

**EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS DE LA
INGENIERÍA DE MANUFACTURA CON ESPECIALIDAD EN AUTOMATIZACIÓN**

SAN NICOLÁS DE LOS GARZA, N. L.

JUNIO DEL 2002

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POST-GRADO**

Los miembros del comité de tesis recomendamos que la tesis “**Programación de PLC’s**”, realizada por la alumna Ing. Patricia Aguilera Martínez, matrícula 411007 sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Ingeniería de Manufactura con Especialidad en Automatización.

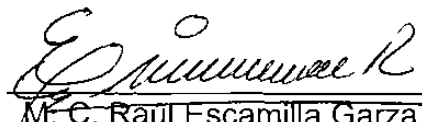
El Comité de Tesis



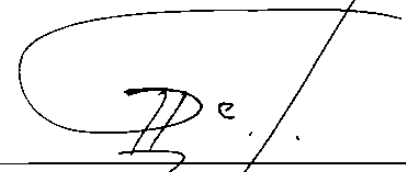
M. C. Juan Ángel Garza Garza
Asesor



M. C. Roberto Villarreal Garza
Coasesor



M. C. Raúl Escamilla Garza
Coasesor



Vq. Bo.
Dr. Guadalupe Alan Castillo Rodríguez
División de Estudios de Postgrado

Agradecimientos

A mis padres, por ser el motor alrededor del cual giran todas mis metas, logros y experiencias tanto laborales como personales. Por darme fuerzas para vencer los retos más importantes en mi vida y sobre todo por haber hecho posible que yo este aquí.

Al personal académico de la FIME por haberme confortado, por haber dejado en mi etapa de estudiante maravillosos momentos, muchos compañeros y grandes amistades.

Al personal del Centro de Escuela Técnica, de Informática y de Postgrado que en lo personal siempre me abrieron sus puertas; por todo su apoyo, muchas gracias.

A ti, que sabes que no necesito escribir tu nombre, pero que te debo todo lo que soy, sin ti no hubiera tenido un camino claro a seguir ni una meta que perseguir, por esto y por mucho más te voy a estar agradecida toda la vida, te quiero y respeto mucho.

Al M. C. Juan Ángel Garza Garza por el apoyo que recibí en mi carrera profesional y ahora en la maestría, muchas gracias.

“La vida es un conjunto de sueños que se van realizando lentamente, solo para que otros sueños ocupen su lugar..... por todo esto gracias FIME”.

Prólogo

Las aplicaciones eléctricas y la lógica de contactos

La lógica de contactos fue desarrollada por la gente del área eléctrica que buscó utilizando relevadores, temporizadores y programadores electromecánicos relacionados unos con otros, secuenciar básicamente el arranque y paro de motores con la finalidad de protegerlos cuando estén operando en líneas de producción industrial.

El uso del álgebra booleana también llamada lógica booleana (falso o verdadero, abierto o cerrado) usando contactos eléctricos en serie, paralelo o la combinación de estos se extendió a otros campos, transformándose en un verdadero lenguaje de programación conocido como lenguaje escalera representado en un diagrama comúnmente llamado circuito escalera.

Durante muchos años fueron usados grandes gabinetes electromecánicos como único medio de efectuar los controles de secuencias y de seguridad en las plantas industriales.

Los inconvenientes de estas instalaciones son las de ocupar un gran espacio, ligada a la vida útil de los contactos de los relevadores, que sólo pueden aceptar una cantidad determinada de aperturas y cierres.

La detección y reparación de fallas que consistían en el reemplazo de piezas electromecánicas y manipulación de alambrado era lenta y normalmente paralizaba el proceso completo, pues todo el gabinete de control debía ser desenergizado, por seguridad.

El desarrollo de los computadores digitales

La idea de utilizar computadores digitales en aplicaciones de control apareció alrededor de 1950, las primeras aplicaciones investigadas fueron en la aeronáutica y los misiles, sin embargo en aquella época, los computadores digitales no habían alcanzado aun el nivel de desarrollo tal que su utilización práctica fuese factible.

Los computadores de la época eran muy grandes, consumían demasiada energía y no eran lo suficientemente confiables, fue por ello que se desarrollaron computadores análogos dedicados a aplicaciones en la aeronáutica.

El computador análogo fue la herramienta principal durante los años 60 en el control de procesos, este logro dio un nuevo impulso a los fabricantes de computadores digitales, quienes se dieron cuenta del potencial del nuevo mercado que se abría a sus productos. Es así como rápidos cambios e innovaciones se sucedieron unas a otras.

Los cambios tecnológicos más relevantes que surgieron posteriormente los podemos dividir básicamente en 6 períodos.

Primer periodo.- Los sistemas pioneros cerca de 1955.

El trabajo llevado a cabo en la planta de Texaco motivo el interés de toda la industria, fabricantes de computadores e instituciones de investigación.

Cumplían principalmente una función de apoyo a los operadores de la planta, sin tomar el control completo de ella. El control análogo continuaba a cargo del proceso.

Segundo periodo - El período del control digital directo.

El primer cambio drástico de controles análogos por controles digitales se produjo en Inglaterra en 1962, con el reemplazo en la Industria química Imperial (ICI) de todos los controladores análogos por un sistema de control por computador.

Tercer periodo - El período de los mini computadores.

El período de los mini computadores, a partir de 1967, con equipos más pequeños, confiables, baratos y más rápidos permitió la utilización en sistemas más pequeños, lo que motivó un rápido aumento del interés de la industria por utilizar control digital en lugar de control análogo.

Cuarto periodo - Los microcomputadores, desde 1980.

Fue la aparición de los microcomputadores, cada vez más confiables, baratos y rápidos año tras año, hasta nuestros días, con tiempos de vida superiores a 100.000 horas la que popularizó definitivamente el control digital.

Quinto periodo - Los DCS.

Los DCS o sistemas de control distribuido son una herencia de los primeros sistemas de control digital y tuvieron su origen en las aplicaciones de control de grandes plantas, principalmente de la industria química.

Sexto periodo - La aparición del PLC.

A fines de los años 70, el desarrollo de los microcomputadores, la confiabilidad cada vez más alta y las tasas de falla menores fomentó la utilización de microprocesadores en reemplazo de los antiguos programadores electromecánicos. Aparecieron los PLC o Controladores Lógicos Programables. Originalmente se pretendió reemplazar con ellos los antiguos gabinetes de control por relevadores.

El avance de la tecnología principalmente en la electrónica permitió el desarrollo de módulos integrados para tarjetas de entradas de contacto y salidas de relevadores, reemplazando de esta manera los relevadores electromecánicos.

Los avances sucesivos en la electrónica y los procesadores permitieron incorporar al mismo equipo lazos de control, ampliando así sus funcionalidades.

Actualmente existen poderosos PLC's en el mercado, capaces de efectuar tanto el control de secuencias como también ejecutar múltiples lazos de control, poderosas funciones de cálculo y tareas de comunicación. Además aparecieron nuevos lenguajes de programación adicionalmente al lenguaje escalera de base, como el lenguaje de secuencias (Grafset) o incluso la programación en códigos más complejos.

Los PLC's están siendo utilizados en todas las áreas de la industria tanto química como minera, producción de alimentos y otros.

Han sido fundamentales en el auge de los PLC's el alto grado de confiabilidad alcanzado, con un tiempo de vida que llega a las 100.000 horas, la posibilidad de tener unidades de procesamiento central redundantes, las facilidades de comunicación vía red con un importante número de equipos en la planta (diferentes protocolos) y características tales como la retención de estados de salidas e información de proceso en memoria en caso de fallas de alimentación.

Otras no menos importante como las facilidades de autodiagnóstico y detección de fallas en circuitos tanto externos como internos, la capacidad de trabajar en forma modular, la robustez mecánica y precios cada vez menores han contribuido también a este auge.

El desarrollo de las aplicaciones de los computadores a la automatización de procesos es un claro ejemplo de como la masificación de un producto fomento la investigación e hizo posible tener costos cada vez más bajos. Este permitió el ingreso de esta tecnología a las aplicaciones industriales.

INDICE

Síntesis	1
CAPITULO 1 Introducción	2
1.1 Descripción del problema	2
1.2 Objetivo de la tesis.....	2
1.3 Hipótesis.....	3
1.4 Límites del estudio	3
1.5 Justificación del trabajo de tesis.....	3
1.6 Metodología.....	4
1.7 Revisión bibliográfica	4
CAPITULO 2 Automatización	7
2.1 Descripción de automatización.....	7
2.2 Objetivos de la automatización	7
2.3 Parte operativa	8
2.3.1 Detectores y Captadores	8
2.3.2 Accionadores y Preaccionadores	9
2.4 Parte de mando	10
2.4.1 Tecnologías cableadas.....	11
2.4.2 Tecnologías programadas	11
CAPITULO 3 Autómatas Programables	13
3.1 Introducción	13
3.2 Definición de un autómata programable	14
3.3 Funciones básicas de un PLC	15
3.4 Campos de aplicación	16
3.4.1 Ejemplos de aplicaciones generales.....	17
3.5 Ventajas de los PLC ´s.....	18
3.6 Inconvenientes de los PLC ´s.....	19
CAPITULO 4 Estructura del PLC.....	20
4.1 Estructura. Conceptos generales	20
4.1.1 Estructura externa.....	21
4.1.2 Estructura interna.....	22
4.2 Memoria	24
4.3 CPU.....	25
4.4 Unidades de E/ S (Entrada y Salida de datos).....	26
4.5 Interfaces	27
4.6 Equipo o unidades de programación	28

4.7	Dispositivos periféricos	29
4.8	Ciclo de trabajo de un autómata	29
CAPITULO 5	Sistemas lógicos	31
5.1	Funciones generales	31
5.2	Funciones especiales	32
5.3	Registros y acumuladores	33
5.4	Temporizadores	33
5.5	Constantes	34
5.6	Estructura del programa	34
5.7	Tipos de módulos	36
Capítulo 6	Lenguajes de programación	37
6.1	Introducción	37
6.1.1	Esquema de contactos	37
6.1.2	MNEMÓNICO	41
6.2	Gráfico secuencial de funciones (Grafcet)	42
6.3	Lista de instrucciones	43
6.4	Texto estructurado	44
6.5	Diagrama de contactos	44
6.6	Diagrama de funciones	45
6.7	Organizaciones de tareas	45
6.8	Bloques de funciones	46
Capítulo 7	Grafcet	47
7.1	Introducción	47
7.2	Antecedentes	47
7.3	Elementos del Grafcet	48
7.3.1	Etapas iniciales	48
7.3.2	Etapas normales	49
7.3.3	Acciones	50
7.3.4	Transiciones	54
7.3.5	Líneas de enlace	55
7.4	Diseño y estructuras del Grafcet	56
7.4.1	Desarrollo del sistema	56
7.4.2	Evolución del sistema	57
7.4.3	Estructuras básicas	58
7.5	Desarrollo de ejemplos con Grafcet	65
Capitulo 8	Conclusiones y recomendaciones	76
	Bibliografía	77
	Listado de figuras	79
	Glosario de términos	81
	Resumen Autobiográfico	83

Síntesis

El desarrollo acelerado en el uso de nuevas tecnologías, sobre todo en los campos de la electrónica, informática y comunicaciones constituyen un constante desafío en la aplicación de estas en los sistemas automáticos de control, provocando un profundo cambio en la concepción y diseño de los procesos, obligándonos a una constante actualización de conocimientos y un continuo replanteamiento de los métodos de diseño empleados.

Otro factor a considerar es que a medida que aparezcan nuevos equipos en el mercado se amplían las posibilidades de comunicación e interconexión entre ellos.

La complejidad del sistema de control alcanzable con los autómatas, hace que sea imprescindible el empleo de métodos de diseño sistemático, frente al método intuitivo que se veía empleado en el diseño de relevadores.

El propósito de esta tesis es el de presentar dichos métodos sistemáticos aprovechando las funciones y posibilidades que ofrecen actualmente los autómatas en comparación a los sistemas clásicos empleados hace unas décadas. Como ejemplo haremos especial hincapié en el diseño mediante los denominados Gráficos Funcionales de Control de Etapas y Transiciones (GRAFCET) que permite con una gran sencillez el desarrollo del proyecto y aprovechamiento de automatismos secuenciales.

CAPITULO 1 Introducción

1.1 Descripción del problema

El diseño tradicional en la automatización de procesos usando PLC (Controlador Lógico Programable) consiste normalmente en el uso de bloques definidos lo que para adecuaciones o algunas aplicaciones específicas no se aplica metodología alguna, ocasionando dificultad de diseño o diseños exclusivos y poco entendibles, en los que solo el que diseña pueda modificarlos, debido a que a otra persona le tomaría tiempo entender la lógica en la que fue hecho y por consecuencia la dificultad de lograr alguna modificación a este.

1.2 Objetivo de la tesis

Hacer un análisis de las herramientas por eventos discretos y proponer una metodología para el diseño de la automatización de procesos industriales usando para ello lenguajes de programación que se puedan usar en los Controladores Lógicos Programables (PLC´s).

1.3 Hipótesis

Se cree que es más fácil y rentable la programación por eventos discretos de los Controladores Lógicos Programables (PLC´s) a través de lenguajes comparado a la programación tradicional usando bloques definidos, representados normalmente en diagramas de escalera.

1.4 Límites del estudio

Se pretende llegar a desarrollar una de las aplicaciones que existen alrededor de la programación de Controladores lógicos Programables para comprobar la facilidad en el diseño de la automatización de procesos industriales.

1.5 Justificación del trabajo de tesis

En la actualidad la cultura de los Ingenieros dedicados al diseño de sistema de automatización con Controladores Lógicos Programables se basa en módulos o soluciones anteriormente ya planeadas sin conocer la eficiencia de estas o la prueba y error que lleva un tiempo en el diseño; esto no asegura un eficaz funcionamiento así como una lógica en la que otro diseñador pueda comprender o documentar el uso de la programación por eventos discretos.

1.6 Metodología

Primera Fase:

Concepto de automatización.

Búsqueda bibliográfica de autómatas programables.

Sistemas lógicos.

Estructura del PLC.

Lenguajes de programación de un autómata programable.

Segunda Fase:

Basándose en los lenguajes de programación, mostrar la diferencia entre la programación de un PLC por contactos comparada con la de eventos discretos haciendo uso de la programación de diseño y estructura del lenguaje Grafset.

1.7 Revisión bibliográfica

Existen varias publicaciones que se enfocan solo en aclarar el funcionamiento de los PLC's de manera genérica con el fin de mantener los conceptos estables durante el tiempo y solo ir modificando los parámetros de los componentes. Otras publicaciones se preocupan por la optimización del producto de manera integrada y muy pocas por generar una guía de diseño.

La bibliografía presentada en esta tesis se ha desarrollado en base o refiriéndose a las publicaciones que a continuación se mencionan.

[1] Autómatas Programables,

José Luis Romeral, Serie Mundo Electrónico.

Editorial Marcombo, S.A., Boixareu Editores. 1997.

[2] "Programmable Logic Controllers and Petri Nets"

IFAC, Madrid. September 1982

[3] The Design and Modeling of PLC Programs Using Petri Nets, Cambridge, England 1995.

[4] GRAFNET: PRÁCTICA Y APLICACIONES J. C. BOSSY, P. BRARD, P. FAUGERE, C. MERALUD Ediciones UPC, Aula Quadern, 1995.

[5] PROGRAMMABLE LOGIC DEVICES AND LOGIC CONTROLLERS E. MANDADO, J. MARCOS; Prentice Hall International, 1996.

[6] Autómatas Programables, fundamentos, manejo, instalación y prácticas A. Porras / A P. Montanero Ed. Mc Graw Hill, publicación, 1990.

[7] Real Time Systems and their Programming and Languages, Ed. Addison-Wesley. Burns, A. and A. Welling, 1990.

[8] Internet

<http://www.autómatas.org>

<http://www.grupo-maser.com>

<http://www.femz.autómatas.com>

CAPITULO 2 Automatización

2.1 Descripción de automatización

La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.

Un sistema automatizado consta de dos partes principales:

- Parte de operativa
- Parte mando

2.2 Objetivos de la automatización

Los objetivos principales logrados en el proceso de automatización se describen a continuación:

- Mejorar la productividad de la empresa, reduciendo los costos de la producción y mejorando la calidad de la misma.
- Mejorar las condiciones de trabajo del personal, suprimiendo los trabajos pesados e incrementando la seguridad.
- Realizar las operaciones imposibles de controlar intelectual o manualmente.

- Mejorar la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.
- Simplificar el mantenimiento de forma que el operario no requiera grandes conocimientos para la manipulación del proceso productivo.
- Integrar la manufactura y producción.

2.3 Parte operativa

La parte operativa es la parte que actúa directamente sobre la máquina. Son los elementos que hacen que la máquina se mueva y realice la operación deseada. Los elementos que forman la parte operativa son los accionadores de las máquinas como motores, cilindros, compresores y los captadores como fotodiodos, transductores.

2.3.1 Detectores y Captadores

Como las personas necesitan de los sentidos para percibir, lo que ocurre en su entorno, los sistemas automatizados precisan de los transductores para adquirir información de:

- La variación de ciertas magnitudes físicas del sistema.
- El estado físico de sus componentes.

Los dispositivos encargados de convertir las magnitudes físicas en magnitudes eléctricas se denominan transductores.

Los transductores se pueden clasificar en función del tipo de señal que transmiten en:

- Transductores todo o nada: Suministran una señal binaria claramente diferenciada. Los finales de carrera son transductores de este tipo.
- Transductores numéricos: Transmiten valores numéricos en forma de combinaciones binarias. Los encoders son transductores de este tipo.
- Transductores analógicos: Suministran una señal continua que es fiel reflejo de la variación de la magnitud física medida.

Algunos de los transductores más utilizados son: fotoceldas, botones, encoders, etc.

2.3.2 Accionadores y Preaccionadores

El accionador es el elemento final de control que, en respuesta a la señal de mando que recibe, actúa sobre la variable o elemento final del proceso.

Un accionador transforma la energía de salida del automatismo en otra de más utilidad para el entorno industrial de trabajo.

Los accionadores pueden ser clasificados en eléctricos, neumáticos e hidráulicos. Los más utilizados en la industria son: cilindros, motores de corriente alterna, motores de corriente continua, etc.

Los accionadores son controlados por la parte de mando, sin embargo, pueden estar bajo el control directo de la misma o bien requerir algún preaccionamiento para amplificar la señal de mando. Esta preamplificación se traduce en establecer o interrumpir la circulación de energía desde la fuente al accionador.

Los preaccionadores disponen de: parte de mando o de control que se encarga de conmutar la conexión eléctrica, hidráulica o neumática entre los cables o conductores del circuito de potencia.

2.4 Parte de mando

La parte de mando suele ser un autómatas programable (tecnología programada), aunque hasta hace poco se utilizaban relevadores electromagnéticos, tarjetas electrónicas o módulos lógicos neumáticos (tecnología cableada).

En un sistema de fabricación automatizado el autómatas programable esta en el centro del sistema. Este debe ser capaz de comunicarse con todos los elementos que lo constituyen.

2.4.1 Tecnologías cableadas

Con este tipo de tecnología, el automatismo se realiza interconectando los distintos elementos que lo integran. Su funcionamiento es establecido por los elementos que lo componen y por la forma de conectarlos.

Esta fue la primera solución que se utilizó para crear autómatas industriales, pero presenta varios inconvenientes.

Los dispositivos que se utilizan en las tecnologías cableadas para la realización del automatismo son:

- Relevadores electromagnéticos.
- Módulos lógicos neumáticos.
- Tarjetas electrónicas.

2.4.2 Tecnologías programadas

Los avances en el campo de los microprocesadores de los últimos años han favorecido la generalización de las tecnologías programadas en la realización de automatismos. Los equipos realizados para este fin son:

- Las computadoras.
- Los autómatas programables.

La computadora, como parte de mando de un automatismo presenta la ventaja de ser altamente flexible a modificaciones de proceso. Pero, al mismo tiempo, y debido a su diseño no específico para su entorno industrial, resulta un elemento frágil para trabajar en entornos de líneas de producción.

Un autómeta programable industrial es un elemento robusto diseñado especialmente para trabajar en ambientes de talleres, con casi todos los elementos de la computadora.

CAPITULO 3 Autómatas Programables

3.1 Introducción

Hasta no hace mucho tiempo el control de procesos industriales se hacía de forma cableada por medio de contactores y relevadores.

Al operario que se encontraba a cargo de este tipo de instalaciones, se le exigía tener altos conocimientos técnicos para poder realizarlas y posteriormente mantenerlas en funcionamiento.

Además cualquier variación en el proceso suponía modificar físicamente gran parte de las conexiones de los montajes siendo necesario dedicar mucho tiempo además de un gran esfuerzo técnico, por lo que se incide directamente un mayor desembolso económico.

En la actualidad no se puede entender un proceso complejo de alto nivel desarrollado para técnicas cableadas. La computadora y los autómatas programables han intervenido de una forma considerable para que este tipo de instalación se haya visto sustituida por otras de forma programada.

El Autómata Programable Industrial (API) nació como solución al control de circuitos complejos de automatización. Por lo tanto se puede decir que un API no es más que un aparato electrónico que sustituye los circuitos auxiliares o de mando de los sistemas automáticos. A él se le conectan los captadores (transductores botones) por una parte, y los actuadores (bobinas de contactores, lámparas) por otra.

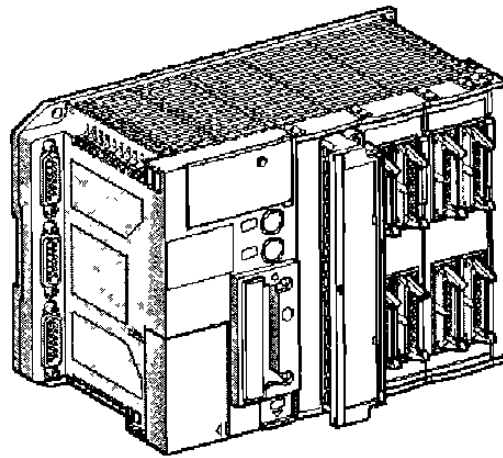


Figura 3.1 Esquema de un autómata programable.

3.2 Definición de un autómata programable

Se entiende por Controlador Lógico Programable (PLC), o Autómata Programable, a toda máquina electrónica diseñada para controlar en tiempo real y en medio industrial procesos secuenciales.

Esta definición se está quedando un poco desfasada, ya que han aparecido los Micro-PLC's, destinados a pequeñas necesidades y al alcance de cualquier persona.

3.3 Funciones básicas de un PLC

Dentro de estas funciones podemos mencionar:

- La detección: Lectura de la señal de los captadores distribuidos por el sistema de fabricación.
- El mando: Elabora y envía las acciones al sistema mediante los accionadores y preaccionadores.
- El diálogo hombre máquina: Mantener un diálogo con los operarios de producción obedeciendo sus consignas e informando del estado del proceso.

La programación: Para introducir, elaborar y cambiar el programa de aplicación del autómata el diálogo de programación debe permitir modificar el programa incluso con el autómata controlando la máquina.

Nuevas Funciones

- Redes de comunicación: Permiten establecer comunicación con otras partes de control. Las redes industriales permiten la comunicación y el intercambio de datos entre autómatas a tiempo real. En unos cuantos milisegundos pueden enviarse telegramas e intercambiar tablas de memoria compartida.
- Sistemas de supervisión: También los autómatas permiten comunicarse con computadoras provistas de programas de supervisión industrial. En comunicación se realiza por una simple conexión por el puerto serie de la computadora.

- Control de procesos continuos: Además de dedicarse al control de sistemas de eventos discretos los autómatas llevan incorporadas funciones que permiten el control de procesos continuos. Disponen de módulos de entrada, salidas analógicas y la posibilidad de ejecutar reguladores PID que están programados en el autómata.
- Entradas-Salidas distribuidas: Los módulos de entrada-salida pueden estar distribuidos por la instalación, se comunican con la unidad central del autómata mediante un cable de red.
- Buses de campo: Mediante un solo cable de comunicación se pueden conectar al bus de captadores y accionadores, remplazando al cableado tradicional

3.4 Campos de aplicación

El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La constante evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc. , por tanto, su aplicación abarca desde procesos de fabricación industriales de cualquier tipo a transformaciones industriales, control de instalaciones, etc.

Sus reducidas dimensiones, la extremada facilidad de su montaje, la posibilidad de almacenar los programas para su posterior y rápida utilización, la modificación o alteración de los mismos, hace que su eficacia se aprecie fundamentalmente en procesos en que se producen necesidades tales como:

- 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.

3.4.1 Ejemplos de aplicaciones generales

A continuación mencionaremos algunos ejemplos de aplicaciones comunes:

- Maniobra de máquinas.
- Maquinaria industrial de plástico.
- Máquinas de transferencia.
- Maquinaria de embalajes.
- Maniobra de instalaciones.

3.5 Ventajas de los PLC´s

No todos los autómatas ofrecen las mismas ventajas sobre la lógica cableada, ello es debido, principalmente, a la variedad de modelos existentes en el mercado y las innovaciones técnicas que surgen constantemente. Tales consideraciones me obligan a referirme a las ventajas que proporciona un autómata de tipo medio.

Dentro de las ventajas que podemos observar es el menor tiempo empleado en la elaboración de proyectos debido a que:

- No es necesario dibujar el esquema de contactos.
- No es necesario simplificar las ecuaciones lógicas, ya que, por lo general la capacidad de almacenamiento del módulo de memoria es lo suficientemente grande.
- La lista de materiales queda sensiblemente reducida, y al elaborar el presupuesto correspondiente se eliminará parte del problema que supone el contar con diferentes proveedores, distintos plazos de entrega.
- Posibilidad de introducir modificaciones sin cambiar el cableado ni añadir aparatos.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra en la instalación.
- Economía de mantenimiento. Además de aumentar la fiabilidad del sistema, al eliminar contactos móviles, los mismos autómatas pueden indicar y detectar averías.

- Posibilidad de gobernar varias máquinas con un mismo autómata.
- Menor tiempo para la puesta en funcionamiento del proceso al quedar reducido el tiempo cableado.

3.6 Inconvenientes de los PLC´s

Como inconvenientes podríamos hablar, en primer lugar, de que hace falta un programador, lo que obliga a capacitar a uno de los técnicos en tal sentido, pero hoy en día ese inconveniente está solucionado porque las universidades ya se encargan de dicho adiestramiento.

El costo inicial también puede ser un inconveniente.

CAPITULO 4 Estructura del PLC

4.1 Estructura. Conceptos generales

La mejor opción para el control de procesos industriales es el empleo de autómatas programables. Estos aparatos se basan en el empleo de un microcontrolador para el manejo de entradas y salidas.

La memoria del aparato contendrá tanto el programa de usuario que le introduzcamos como el sistema operativo que permite ejecutar en secuencia las instrucciones del programa. Opcionalmente, en la mayoría de los autómatas, también se incluyen una serie de funciones re-implementadas de uso general (como reguladores PID).

La mayor ventaja es que si hay que variar el proceso basta con cambiar el programa introducido en el autómata (en la mayoría de los casos).Otra ventaja es que el autómata también nos permite saber el estado del proceso incluyendo la adquisición de datos para un posterior estudio.

4.1.1 Estructura externa

Todos los autómatas programables, poseen una de las siguientes estructuras:

- Compacta: en un solo bloque están todos los elementos.
- Modular:

Estructura americana: separa la E/ S del resto del autómata.

Estructura europea: cada módulo es una función (fuente de alimentación, CPU, E/ S, etc.).

Exteriormente nos encontramos con cajas que contienen una de estas estructuras las cuales poseen indicadores y conectores en función del modelo y fabricante.

Para el caso de una estructura modular se dispone de la posibilidad de fijar los distintos módulos en rieles normalizados, para que el conjunto sea compacto y resistente.

Los micro-autómatas suelen venir sin caja en formato kit ya que su empleo no es determinado y se suele incluir dentro de un conjunto más grande de control o dentro de la maquinaria que se debe controlar.



Figura 4.1 Estructura externa de un PLC.

4.1.2 Estructura interna

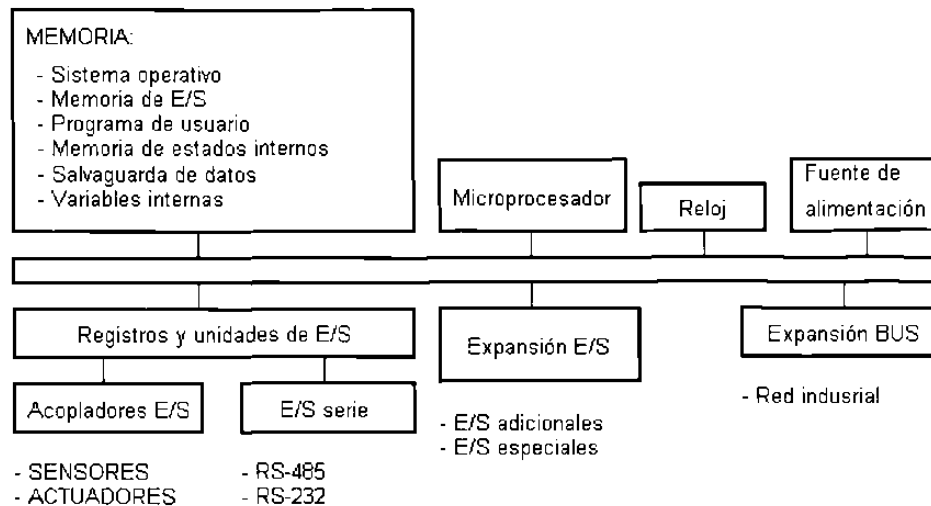


Figura 4.2 Diagrama describiendo la estructura interna de un PLC.

Los elementos esenciales, que como mínimo todo autómata programable posee son:

- Sección de entradas: se trata de líneas de entrada, las cuales pueden ser del tipo digital o analógico. En ambos casos tenemos unos rangos de tensión característicos, los cuales se encuentran en las hojas de características del fabricante. A estas líneas conectaremos los sensores.
- Sección de salidas: son una serie de línea de salida, que también pueden ser de carácter digital o analógico. A estas líneas conectaremos los actuadores.
- Unidad central de proceso (CPU): se encarga de procesar el programa de usuario que introduciremos, para ello disponemos de diversas zonas de memoria, instrucciones de programa y registros.

Adicionalmente en determinados modelos más avanzados, podemos disponer de funciones ya integrados en el CPU; como reguladores PID, control de posición, etc.

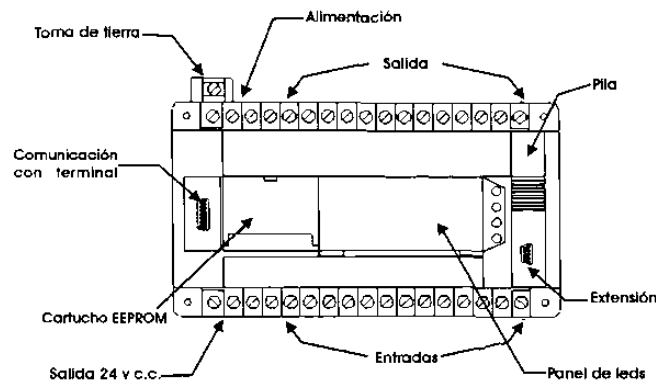


Figura 4.3 Elementos esenciales de un autómata programable.

Tanto las entradas como las salidas están aisladas de la CPU según el tipo de autómata que utilizemos. Normalmente se suelen emplear opto acopladores en las entradas relevadores y salidas.

Aparte de estos elementos podemos disponer de los siguientes:

- Unidad de alimentación (algunas CPU la llevan incluida).
- Unidad o consola de programación: que nos permitirá introducir, modificar y supervisar al programa del usuario.
- Dispositivos periféricos: como nuevas unidades de E/ S, más memoria unidades de comunicación en red, etc.
- Interfaces: facilitan la comunicación del autómata mediante enlace serie con otros dispositivos (como una computadora personal).

4.2 Memoria

Dentro de la CPU vamos a disponer de un área de memoria, la cual emplearemos para diversas funciones:

- Memoria del programa de usuario: aquí introduciremos el programa va a ejecutar cíclicamente.
- Memoria de la tabla de datos: se suele subdividir en zonas según el tipo de datos (como marcas de memoria, temporizadores, contadores, etc.).
- Memoria del sistema: aquí se encuentra el programa en código máquina que monitoriza el sistema (programa del sistema). Este programa es ejecutado directamente por el microprocesador /microcontrolador que posea el autómata.
- Memoria de almacenamiento: se trata de memoria externa que empleamos para almacenar el programa de usuario, y en ciertos casos parte de la memoria de la tabla de datos. Suele ser de uno de los siguientes tipos: EPROM, EEPROM O FLASH.

Cada autómata divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante.

4.3 CPU

La CPU es el corazón del autómata programable. Es la encargada de ejecutar el programa de usuario mediante el programa del sistema (es decir, el programa de usuario es interpretado por el programa del sistema).

Sus funciones son:

- Vigilar que el tiempo de ejecución del programa de usuario no exceda un determinado tiempo máximo (tiempo de ciclo máximo). A esta función se le suele denominar watchdog (perro guardián).
- Ejecutar el programa de usuario.
- Crear una imagen de las entradas, ya que el programa de usuario no debe acceder directamente a dichas entradas.
- Renovar el estado de las salidas en función de la imagen de las mismas obtenida al final del ciclo de ejecución del programa de usuario.
- Verificación del sistema.

El autómata va a poseer un ciclo de trabajo, que ejecutará de forma continua:

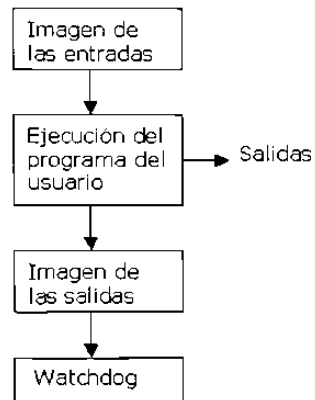


Figura 4.4 Ciclo de trabajo de un autómata.

4.4 Unidades de E/ S (Entrada y Salida de datos)

Generalmente vamos a disponer de dos tipos de E /S:

- Digital
- Analógica

Las E /S digitales se basan en el principio de todo o nada, es decir o no conducen señal alguna o poseen un nivel mínimo de tensión. Estas E /S se manejan en nivel de bit dentro del programa de usuario.

Las E /S analógicas pueden poseer cualquier valor dentro de un rango determinado especificado por el fabricante, se basan en convertidores A /D y D /A aislados de la CPU. Estas señales se manejan en nivel de byte o palabra (8/16) dentro del programa de usuario.

Las E /S son leídas y escritas dependiendo del modelo y del fabricante, es decir pueden estar incluidas sus imágenes dentro del área de memoria o ser manejadas a través de instrucciones específicas de E /S.

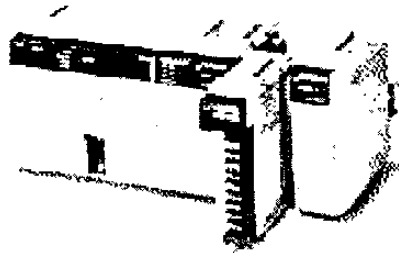


Figura 4.5 Dispositivos conectados como entradas y salidas.

4.5 Interfaces

Todo autómeta, salvo casos excepcionales, posee la virtud de poder comunicarse con otros dispositivos (como una computadora personal).

Lo normal es que posea una E /S serie del tipo RS-232 / RS-422.

A través de esta línea se pueden manejar todas las características internas del autómeta incluida la programación del mismo, y suele emplearse para monitorización del proceso en otro lugar separado.



Figura 4.6 Dispositivo utilizado como interfase.

4.6 Equipo o unidades de programación

El autómata debe de disponer de alguna forma de programación, la cual se suele realizar empleando alguno de los siguientes elementos:

- Unidad de programación: suele ser en forma de calculadora. Es la forma más simple de programar el autómata, y se suele reservar para pequeñas modificaciones del programa o la lectura de datos en el lugar de colocación del autómata.
- Consola de programación: es una terminal a modo de ordenador, que proporciona una forma más cómoda de realizar el programa de usuario y observar parámetros internos del autómata desfasado actualmente.
- PC: es el modo más potente y empleado en la actualidad. Permite programar desde un ordenador personal estándar, con todo lo que ello supone: herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCAD, etc.

Por cada caso el fabricante proporciona lo necesario, el equipo o el software (cables adecuados) cada equipo dependiendo del modelo y fabricante, puede poseer una conexión a uno o varios de los elementos anteriores. En el caso de los Micro-PLC se escoge la programación por PC o por unidad de programación integrada en la propia CPU.

4.7 Dispositivos periféricos

El autómata programable, en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan una gran variedad de posibilidades, que van desde las redes internas, módulos auxiliares de E /S, memoria adicional hasta la conexión con otros autómatas del mismo modelo.

Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.

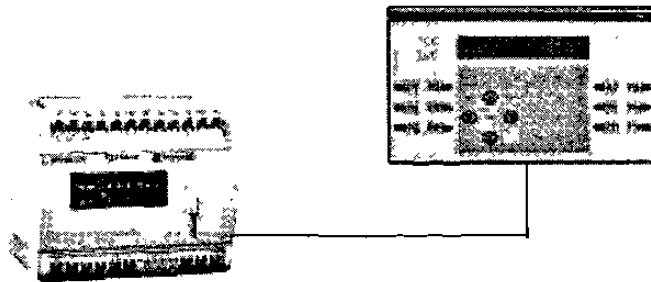


Figura 4.7 Dispositivos periféricos.

4.8 Ciclo de trabajo de un autómata

El autómata va a ejecutar nuestro programa de usuario en un tiempo determinado, el cual va a depender sobre todo de la longitud del programa. Esto debido a que cada instrucción tarda un tiempo determinado en ejecutarse, por lo que en proceso rápido será un factor crítico.

En un sistema de control mediante autómata programable tendremos 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.

La vigilancia y exploración de las entradas, los retardos de entrada y la ejecución del programa del usuario dan como total el tiempo de ciclo del autómata. Al transcurrir este ciclo es cuando se modifican las salidas, por lo que sí 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, si no 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 tapa de vigilancia (watchdog) se comprueba si se sobrepasó el tiempo máximo de ciclo, activándose en caso afirmativo la señal de error correspondiente.

CAPITULO 5 Sistemas lógicos

5.1 Funciones generales

Las operaciones combinacionales más comunes se realizan con los bloques de funciones básicas, conexión serie, paralelo, negación, etc.

	AND	OR	NOT	EXOR	NAND	NOR	EXNOR
	Condición	Alternativa	Negar	Or Exclusivo	AND negado	OR negado	EXOR negado
Símbolo							
Ecuación	$S = A \cdot B$ $S = AB$	$S = A + B$	$S = \bar{A}$	$S = A \oplus B$	$S = \overline{A \cdot B}$ $S = \overline{AB}$	$S = \overline{A + B}$	$S = \overline{A \oplus B}$
Tabla de Verdad							
Circuito eléctrico equivalente							
Simbología NEMA							

Figura 5.1 Tabla de funciones combinacionales básicas y sus diferentes representaciones.

Normalmente las funciones AND, OR, EXOR NAND y NOR tiene tres entradas y una salida. Si deseamos realizar operaciones con más de tres entradas, se conectan en cascada como lo indica la siguiente figura.

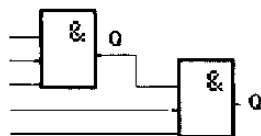


Figura 5.2 Operación And de cinco entradas.

5.2 Funciones especiales

- Temporizador con retardo a la conexión: Activa la salida Q una vez que ha transcurrido el tiempo programado
- Temporizador con retardo a la desconexión: Desactiva la salida una vez transcurrido el tiempo programado.
- Relevador de impulsos: Tiene el mismo funcionamiento que un interruptor. La salida cambia de estado, de 0 a 1, cada vez que cambia la señal de entrada de disparo (Trigger).
- Reloj: Permite controlar los instantes de activación y desactivación de la salida en un día de la semana y una hora determinada.
- Generador de pulsos: Genera pulsos de reloj a intervalos iguales. Funcionamiento similar a un intermitente.

5.3 Registros y acumuladores

Todas las operaciones que hagamos con las entradas y las salidas se deben de efectuar en algún sitio. Para ello se define:

- Registro de entrada (VKE): Su tamaño es de 1 bit. Aquí es donde efectuaremos las instrucciones, a la carga de entradas y a la asignación de salidas en nivel de bit.
- Acumuladores (AKK1 Y AKK2): Sus son de 16 bits cada uno. Cada vez que se almacene un dato en los acumuladores se seguirá la siguiente secuencia.

5.4 Temporizadores

Varían en función de marcas y modelos, pero los más usados suelen incorporar 32 temporizadores:

T0....T31 y 32 contadores: Z0....Z31

De los 32 contadores, 8 no se borran al desconectar el autómata (son remanentes), dichos contadores son Z0 a Z27. Para consultar el estado de cada uno de ellos podremos usarlos como si fueran entradas (mediante operaciones combinacionales) o almacenando su valor en los AKKU.

5.5 Constantes

A la hora de descargar datos en acumuladores, temporizadores, registros, etc., tendremos varias posibilidades en la forma de introducir el dato:

- KB: 8 bits (0 a 255 en decimal).
- KC: 8 bits (2 caracteres alfanuméricos).
- KF: 16 bits (2 bytes, 0 a 255 en decimal cada uno).
- KB: 16 bits (binario natural).

5.6 Estructura del programa

Dentro de la estructura nos encontramos con dos opciones para escribir el programa:

- Lineal: Se emplea un único módulo de programa (OB1). Este módulo se procesa cíclicamente, es decir, tras la última instrucción se volverá a ejecutar la primera. Si la tarea a controlar es simple esta es la mejor forma.
- Estructurada: Para el caso de tareas complejas es más conveniente dividir el programa en módulos. Mediante esta forma logramos un programa más claro y adquiriremos la posibilidad de poder llamar a un módulo desde cualquier parte del programa (lo que evita repetir código).

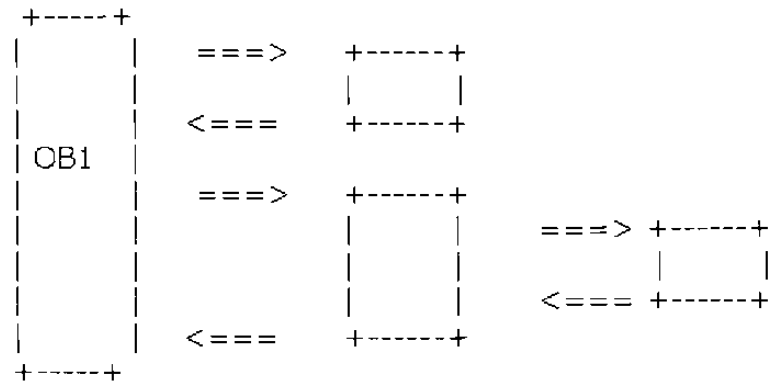


Figura 5.3 Estructura representando un módulo.

En la programación estructurada se comienza y termina en el módulo OB1, desde el cual saltaremos y retornaremos a los módulos que nos interesen. Por supuesto se podrá saltar desde un módulo a otro (anidado), siempre que nos superemos los 16 bits de salto que permite como máximo el autómata. Otras limitaciones son:

- El salto de un módulo a otro debe ser siempre hacia delante (Ej. Se podrá saltar de PB1 a PB2, pero no a la inversa).
- No se pueden dar dos saltos a un mismo módulo desde el módulo actual (Ej. No se podrá saltar dos veces a PB3 desde PB2 pero si puede saltarse a PB3 desde distintos módulos).
- Tanto en la programación lineal como en la estructura los módulos acabarán mediante la instrucción BE.
- La memoria del autómata esta limitada, por lo que dispondremos de 1000 líneas de programación aproximadamente.

5.7 Tipos de módulos

Generalmente, existen cuatro tipos de módulos en cualquier autómata programable:

- Módulos de organización (OB): Son los que gestionan el programa del usuario. Numerados OB1, OB3 y OB22.

Debemos destacar el OB1, que es el módulo del programa principal, el OB3, que es el que contiene el programa controlado por alarma y el OB13 que es el módulo para programas controlados por tiempo.

- Módulos de programa (PB): Son los que incluyen el programa de usuario dividido normalmente según sus aspectos funcionales o tecnológicos. PB0PB3.
- Módulos funcionales (FB): Son módulos de programa especiales aquí se introducen las partes del programa que aparecen con frecuencia o poseen gran complejidad. Poseen un juego de instrucciones ampliado FB0...FB3.
- Módulo de datos (DB): En ellos se almacenan datos para la ejecución del programa como valores reales, textos, etc., adoptan los valores: DB0...DB3.

Los módulos DB1 y DB2 se emplean para definir las condiciones internas del autómata por lo que no deben utilizarse.

La mayor ventaja que aportan es la facilidad para variar el proceso que controlan, ya que basta con cambiar el programa introducido en el autómata (en la mayoría e los casos).

Capítulo 6 Lenguajes de programación

6.1 Introducción

Cuando hablamos de los lenguajes de programación nos referimos a diferentes formas de poder escribir el programa usuario.

El software actual nos permiten traducir el programa usuario de un lenguaje a otro, pudiendo así escribir el programa en el lenguaje que más nos conviene.

Existen varios tipos de lenguajes de programación:

- Esquema de contactos
- Mnemónico o listado de contactos
- Esquema funcional

6.1.1 Esquema de contactos

Es un lenguaje gráfico, derivado del lenguaje de relevadores, que mediante símbolos representa contactos, solenoides. Su principal ventaja es que los símbolos básicos están normalizados según normas NEMA y son empleados por todos los fabricantes.

Los símbolos básicos son:



Figura 6.1 Símbolos básicos del esquema de contactos.

Los elementos básicos que configuran la función se representan entre dos líneas verticales que simbolizan las líneas de alimentación.

Para las líneas de función más complejas como temporizadores, registros de desplazamiento, etc. , se hace uso del formato de bloques. Estos no están formalizados, aunque guardan una gran similitud entre sí para distintos fabricantes y resultan mucho más expresivos que si se utiliza para el mismo fin el lenguaje en lista de instrucciones o mnemónico.

Conceptos generales

- Un programa en esquema de contactos, la constituyen una serie de ramas de contactos.
- Una rama esta compuesta de una serie de contactos, conectados en serie o en paralelo que dan origen a una salida que bien puede ser una bobina o una función especial.

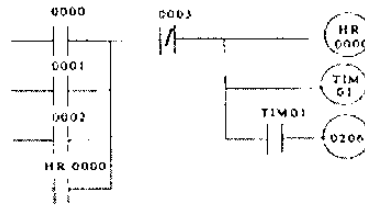


Figura 6.2 Rama de contactos.

- El flujo de la señal va de izquierda a derecha y de arriba abajo.
- A una rama de circuito en esquema de contactos le corresponde una secuencia de instrucciones en forma mnemónica.
- Todas las ramas de circuito se inician con una instrucción LOAD.

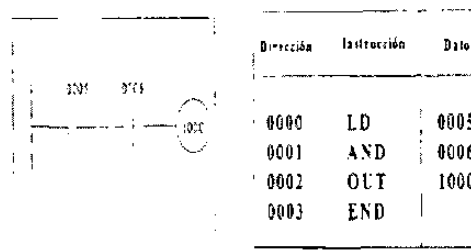


Figura 6.3 Estructura de una instrucción LOAD.

- Una bobina no puede venir conectada directamente de la barra de inicio. En tal caso es necesario interponer un contacto siempre cerrado.

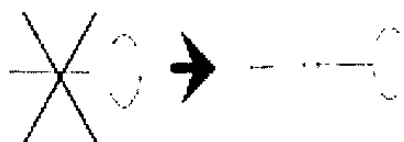


Figura 6.4 simbología de un contacto cerrado.

- A la derecha de una bobina no es posible programar ningún contacto.
- El número de contactos posibles en serie o en paralelo es prácticamente ilimitado.
- Es posible colocar en paralelo 2 o más bobinas.



Figura 6.5 Paralelo de dos o más bobinas.

6.1.2 MNEMÓNICO

Un lenguaje en mnemónico o lista de instrucciones consiste en un conjunto de códigos simbólicos, cada uno de los cuales corresponde a una instrucción.

Cada fabricante utiliza sus propios códigos, y una nomenclatura distinta para nombrar las variables del sistema.

El lenguaje en mnemónico es similar al lenguaje ensamblador del micro.

La complejidad en la programación de los autómatas programables requiere más que nunca de la estandarización de la misma. Bajo la dirección del IEC el estándar IEC 1131-3 (IEC 65) para la programación de PLC's ha sido definida. Alcanzó el estado de Estándar Internacional en Agosto de 1992. Los lenguajes gráficos y textuales definidos en el estándar son una fuerte base para entornos de programación potentes en PLC's. Con la idea de hacer el estándar adecuado para una gran cantidad de aplicaciones, cinco lenguajes han sido definidos en total:

- Gráfico secuencial de funciones (Grafcet)
- Lista de instrucciones (LDI o AWL)
- Texto estructurado
- Diagrama de flujo
- Diagrama de contactos

6.2 Gráfico secuencial de funciones (Grafcet)

El gráfico secuencial de funciones (SFC o Grafcet) es un lenguaje gráfico que proporciona una representación en forma de diagrama de las secuencias del programa. Soporta selecciones, alternativas de secuencias paralelas. Los elementos básicos son pasos y transiciones. Los pasos 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, el SFC es la forma lógica de especificar y programar el más alto nivel de un programa de PLC.

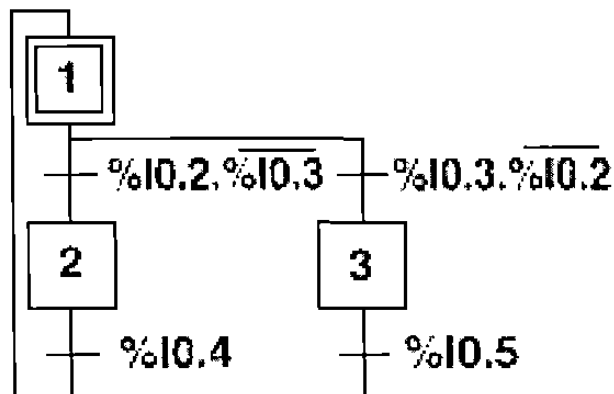


Figura 6.6 Pasos y transiciones de un Grafcet.

6.3 Lista de instrucciones

La lista de instrucciones (IL o AWL) es un lenguaje de bajo nivel, similar al lenguaje ensamblador. Con IL solo una operación es permitida por línea (Ej. Almacenar un valor en un registro). Este lenguaje es adecuado para pequeñas aplicaciones y para optimizar partes de una aplicación.

000	LD	%I0.1	Bp. inicio ciclo
	AND	%I0.0	Dp. presencia vehículo
	AND	%M3	Bit autorización reloj calendario
	AND	%I0.5	Fc. alto rodillo
	AND	%I0.4	Fc. detrás pórtico
005	S	%M0	Memo inicio ciclo
	LD	%M2	
	AND	%I0.5	
	OR	%I0.2	Bp. parada ciclo
	R	%M0	
010	LD	%M0	
	ST	%Q0.0	Piloto ciclo

Figura 6.7 Secuencia que describe un listado de instrucciones.

6.4 Texto estructurado

El texto estructurado (ST) es un lenguaje de alto nivel estructurado por bloques que posee una sintaxis parecida al pascal. El ST puede ser empleado para realizar rápidamente sentencias complejas que manejen variables con un amplio rango de diferentes tipos de datos, incluyendo valores analógicos y digitales.

6.5 Diagrama de contactos

El diagrama de contactos (ladder diagram LD) es un lenguaje que utiliza un juego estandarizado de símbolos de programación. En el estándar IEC los símbolos han sido racionalizados (se ha reducido su número).

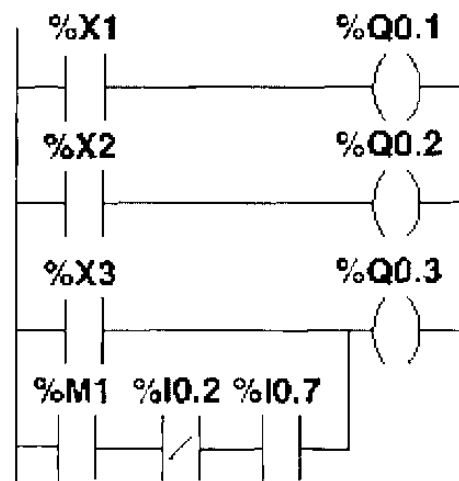


Figura 6.8 Diagrama de contactos escalera.

6.6 Diagrama de funciones

El diagrama de funciones (function block diagram o FBD) es un lenguaje gráfico que permite programar elementos que aparecen como bloques para ser cableados entre sí de forma análoga al esquema de un circuito. Es adecuado para muchas aplicaciones que involucren el flujo de información o datos entre componentes de control.

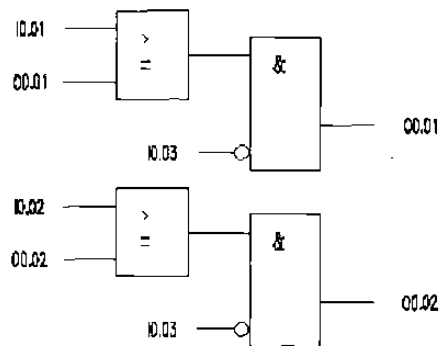


Figura 6.9 Diagrama de función modular o de block.

6.7 Organizaciones de tareas

El estándar también define una nueva arquitectura para la organización de tareas con PLC's. Una tarea controla la ejecución de un programa ejecutándolo periódicamente o en respuesta a un específico evento. Para optimizar los recursos del controlador, una aplicación puede ser fragmentada en un número de pequeños programas concretos. Cada programa está bajo control de una tarea que se ejecuta a la velocidad que requiera la E/S asociada.

6.8 Bloques de funciones

Los bloques de funciones (FB's) son bloques estándar que ejecutan como reguladores PID. El estándar IEC asegura que los bloques de funciones son definidos empleando una metodología estándar. Hay controles empleando parámetros externos, mientras que los algoritmos internos permanecen ocultos empleando programación orientado a objetos.

Capítulo 7 Grafcet

7.1 Introducción

La programación del autómatas consiste en el establecimiento de una sucesión ordenada de instrucciones, escritas en un lenguaje de programación concreto. Estas instrucciones están disponibles en el sistema de programación y resuelven el control de un proceso determinado. Es por ello que hemos seleccionados al Grafcet por su lógica amigable en la programación y aprovechamiento del autómatas, para describir un poco mas de sus características.

7.2 Antecedentes

El Grafcet surge en Francia a mediados de los años 70, debido a la colaboración de algunos fabricantes de autómatas, como Telemecanique y Aper con dos organismos oficiales, AFCET (Asociación francesa para la cibernética, economía y técnica) y ADEPA (Agencia nacional para el desarrollo de la producción automatizada), homologado en Francia, Alemania, y posteriormente por la comisión Electrónica Internacional (IEC 848, año1988).

Actualmente es una herramienta imprescindible cuando se trata de automatizar procesos secuenciales de cierta complejidad con autómatas programables.

El Grafcet es un diagrama funcional que describe la evolución del proceso que se quiere automatizar. Está definido por elementos gráficos y reglas de evolución que reflejan la dinámica del comportamiento del sistema.

Todo automatismo secuencial o concurrente se puede estructurar en una serie de etapas que representan estados o subestados del sistema en los cuales se realiza una o más acciones, así como transiciones, que son las condiciones que deben darse para pasar de una etapa a otra.

7.3 Elementos del Grafcet

7.3.1 Etapas iniciales

- Una etapa inicial se representa con un doble cuadrado.
- Las etapas iniciales de un sistema se activan al iniciar el Grafcet.
- Una vez iniciado las etapas iniciales tienen el mismo tratamiento que las otras etapas.
- Un sistema debe tener como mínimo una etapa inicial.

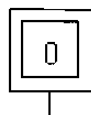


Figura 7.1 Etapa inicial sin retorno.

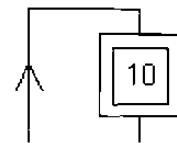


Figura 7.2 Etapa inicial con retorno.

7.3.2 Etapas normales

- Las etapas representan los estados estables del sistema.
- Las etapas del Grafcet se representan mediante un cuadrado numerado
- Las etapas deben estar numeradas; aunque no necesariamente de forma correlativa. No puede haber dos etapas con el mismo número
- Las etapas pueden ser activas o inactivas. Al representar el estado del Grafcet en un momento dado, se puede iniciar que una etapa activa con un punto de color (etapa 4).
- Las etapas, puede o no haber acciones asociadas.

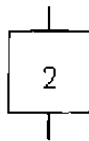


Figura 7.3 Etapa normal.



Figura 7.4 Etapa normal activa.

7.3.3 Acciones

- Asociadas

Una etapa sin ninguna acción asociada (etapa 2) puede servir para hacer detener una acción monoestable que se realizaba en la etapa anterior, o como una etapa de espera.

Una acción asociada (etapa 3). Nos indica que al estar activa la etapa, el motor girará a la derecha.

En una etapa puede haber múltiples acciones como las asociadas (etapa 4). Al estar la etapa 4 activa, el motor girará a la derecha y al mismo tiempo el ventilador estará funcionando.

Si en un sistema en un momento concreto solo hay una sola etapa activa, entonces solo estarán funcionando los elementos activados por las acciones asociadas en esa etapa. (Al menos que en otra etapa se haya activado de forma biestable otra acción).

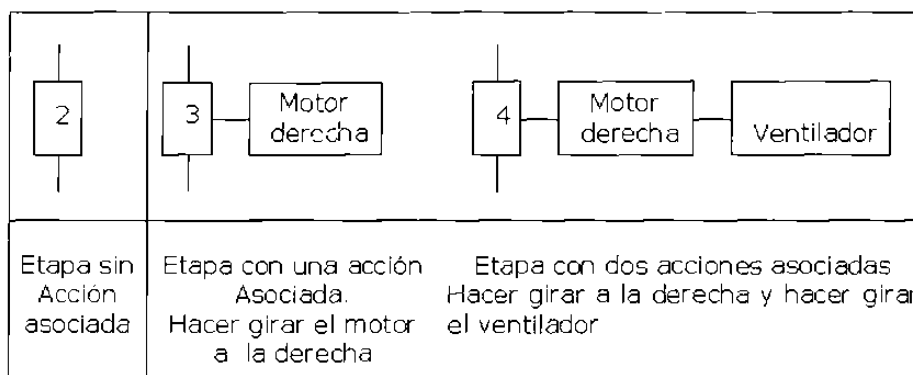


Figura 7.5 Simbología de una etapa activa para una acción asociada.

- Condicionadas

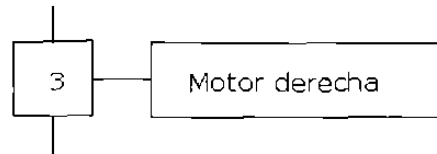


Figura 7.6 Esquema de una etapa condicionada.

La acción a realizar de una o más de las acciones asociadas a una etapa, puede estar condicionada a una función booleana adicional.

En este caso el motor girará a la derecha mientras esté activa la etapa 3.

El rectángulo donde se representa la acción asociada, hay una entrada para las condiciones.

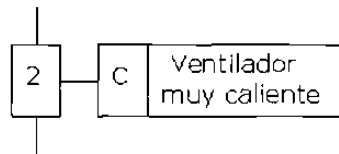
La norma IEC-848 propone las siguientes representaciones para las acciones asociadas.

C	Acción Condicionada
D	Acción retarda
L	Acción limitada en el tiempo
P	Impulso
S	Acción memorizada

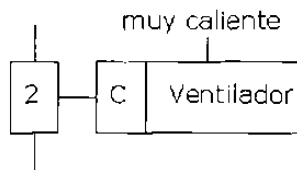
Figura 7.7 Simbología de la norma IEC-848 para las acciones.

- Acción condicionada

Supongamos un sistema en que tenemos un cuadro electrónico, para la regulación de unas máquinas. Si estando activa la etapa de espera 2, y el termostato indica un sobrecalentamiento, el ventilador entrará en marcha.

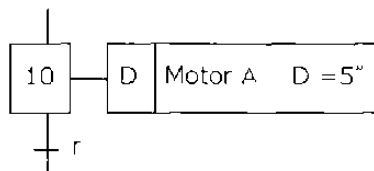


Esta misma acción, la podemos representar dentro del recuadro de la acción, o bien fuera.



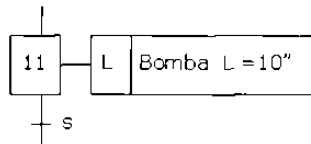
- Acción retardada

El motor A entrará en marcha 5 segundos después de activarse la etapa 10; si la transición " r " se activa antes de ese tiempo el motor no llegará a ponerse en marcha.



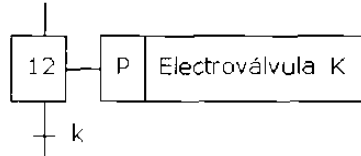
- Acción limitada

La bomba se pondrá en funcionamiento en 10" después de haberse activado la etapa 11, pasado este tiempo aún que no se active la transición " s ", la bomba dejará de funcionar.



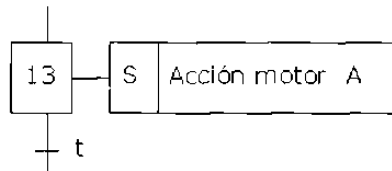
- Acción de impulso

Al activarse la etapa 12, se activará la electro válvula K con un impulso de señal.



- Acción memorizada

Cuando se activa la etapa 13, el motor A se pondrá en marcha de forma biestable (set), y al salir de la etapa, continuará funcionando hasta que se haga un reset a la acción.



- Receptividades asociadas a las transiciones

La condición o condiciones que se deben superar para poder pasar una transición reciben el nombre de receptividades.

Es una transición podemos tener:

Una condición simple [P m]

Una función booleana [(P m + P k) * P p ' *]

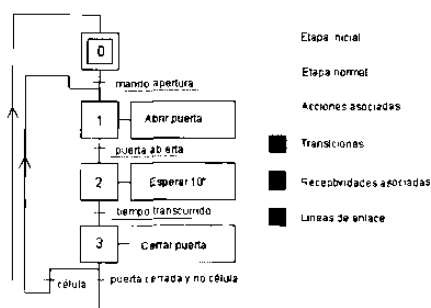
La señal de un temporizador o contador [T03].

La activación de otra etapa del Grafcet [X 12]

Donde X nos indica que la receptividad esta condicionada al hecho que la etapa (en este caso la 12) este activa.

7.3.5 Líneas de enlace

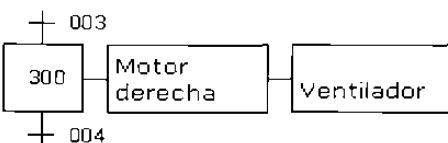
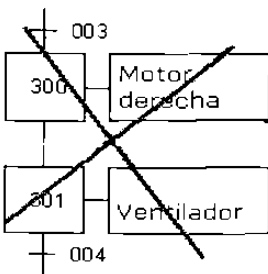
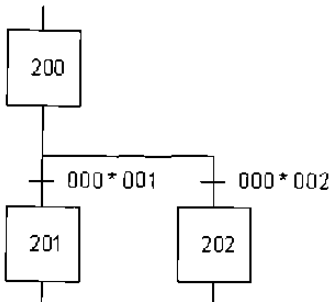
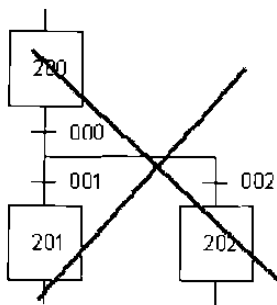
Las líneas de enlace son líneas verticales u horizontales que unen con una dirección significativa (a no ser que indique lo contrario de arriba abajo), las distintas etapas con las transiciones, y las transiciones con las etapas como lo muestra la siguiente figura.



7.4 Diseño y estructuras del Grafcet

7.4.1 Desarrollo del sistema

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



Entre las etapas 200 y 201 o entre las etapas 200 y 202 hay dos condiciones para la transición (000 y 001 o 000 y 002).

En este caso esto se puede resolver haciendo que la receptividad de la transición se cumpla si es válida la función And (000 * 001) o la (000 * 002).

Al superar la condición 003 de la transición, el motor debe girar a la derecha y también se debe accionar el ventilador.

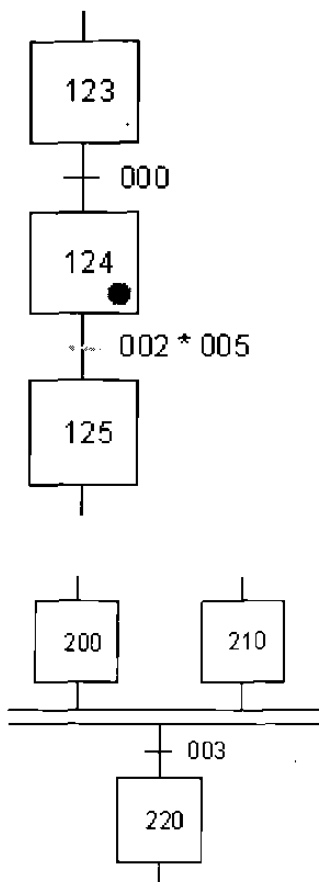
Para realizar esto se han de poner todas las acciones asociadas en la misma etapa.

No puede haber dos etapas seguidas, ni tampoco dos transiciones seguidas.

7.4.2 Evolución del sistema

Para que el sistema pueda evolucionar es necesario:

- La validación de la transición: Todas las etapas de entrada a la transición deben estar activas.
- Que sea cierta la receptividad asociada: Deben ser ciertas las condiciones de la transición.



La primera transición se podrá validar, si la etapa 123 esta activa, y además se cumple la condición 000.

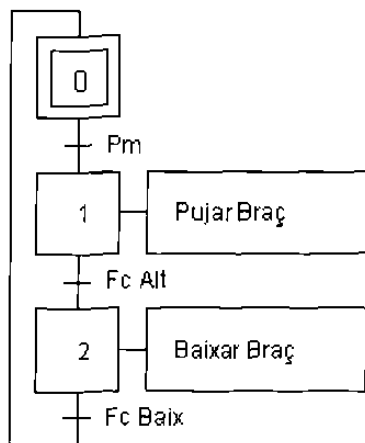
En este momento deja de estar activa la etapa 123, y toma el relevo la 124. Las etapas 200 y 210 son etapas de entrada a la transición.

Para validar la transición, deben estar activas las dos etapas.

Para poder entrar a la etapa 220, la transición tiene que estar validada y se debe de cumplir la receptividad asociada (003) a la transición.

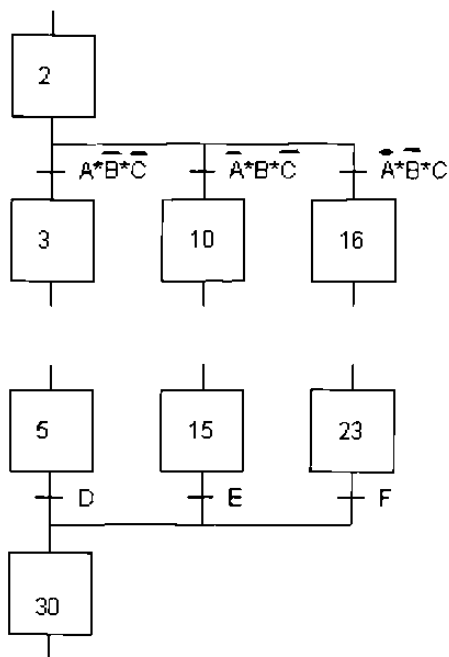
7.4.3 Estructuras básicas

- Secuencia única



Un Grafset 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 a las transiciones.

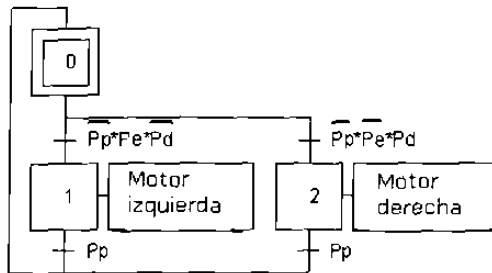
- Bifurcación en O. Selección de secuencia.



Habrà una selecci3n de secuencias, cuando al llegar a un punto encontremos una bifurcaci3n en O, en el serà necesario escoger cual, de las distintas sucesiones de etapas y transiciones se debe seguir.

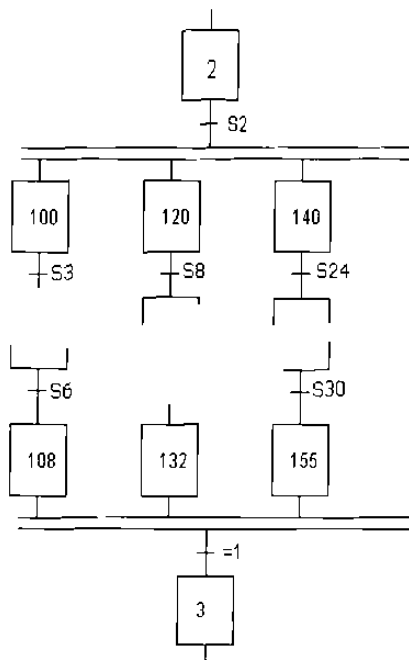
No es necesario que diferentes caminos tengan el mismo nùmero de etapas; pero si conviene que las receptividades asociadas a las transiciones, sean excluyentes.

- Giro a derecha o a izquierda de un motor



Para seleccionar el sentido de giro de un motor utilizaremos la bifurcación en O. Un motor puede girar a la derecha o a la izquierda.

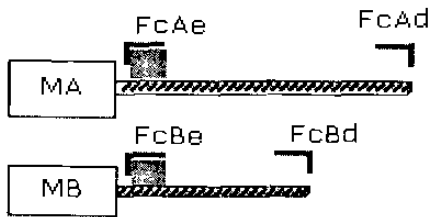
- Bifurcación en Y. Trabajos en paralelo



En automatismo, habrá una bifurcación en Y o "Trabajos paralelos", cuando a partir de un punto, debe evolucionar de forma simultánea por todas las ramas. Al final de estas, encontraremos unas etapas de espera (108,132,155). El sistema continuará su evolución, cuando cada una de las ramas haya llegado a su etapa de espera.

El nombre de las etapas de las diferente ramas puede ser distinto de una a la otra.

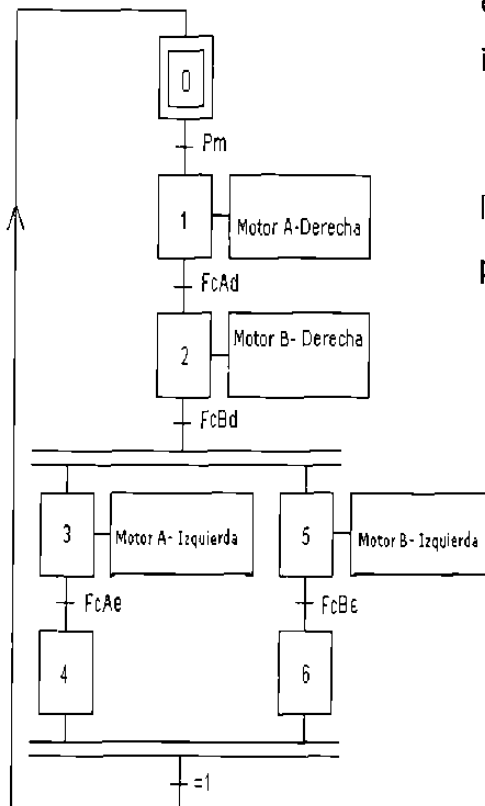
- Motores con trabajos simultáneos



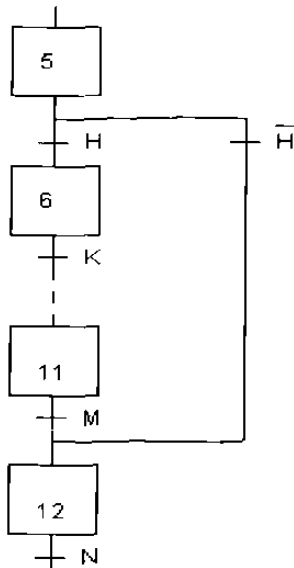
Dos motores MA y MB, desplazan unas piezas.

Primero el motor MA va desde FcAe FcAd, entonces es el MB quien lo hace desde FcBe hasta FcBd. Después los dos vuelven a las posiciones iniciales FcAe y FcBe.

El ciclo se reinicia cuando los dos están de nuevo en las posiciones iniciales.



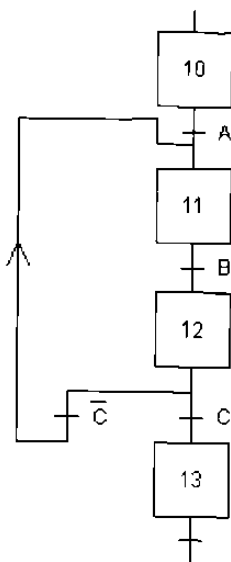
- Saltos de etapas



En un punto puede haber una bifurcación que provoque un salto sobre el conjunto de etapas, que se siga o no la secuencia completa o bien el salto esta determinado por el estado de la condición a la transición (H).

También puede realizarse el salto en sentido ascendente (en este caso lo indicaremos en las líneas de enlace) como pasa en los bucles.

- Bucles

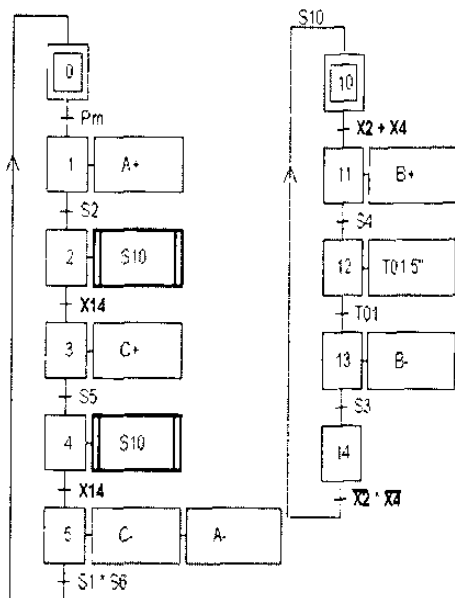


Habrá un bucle o estructura repetitiva (mientras), cuando una o un conjunto de etapas se repitan varias veces, controladas por un contador, temporizador o hasta que se cumpla una condición determinada.

El ciclo de lavado de una lavadora repite varias veces esta estructura (giro a la derecha espera, giro a la izquierda, espera).

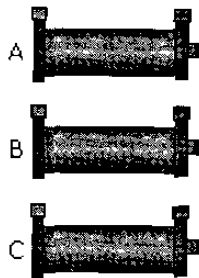
- Subrutinas

Una subrutina es una parte de un programa que realiza una tarea concreta, a la que se puede una o varias veces por parte del programa principal. Una vez realizadas las acciones de la subrutina el programa continúa en el punto donde estaba.



Los trabajos a desarrollar en automatismo se pueden dividir entre diferentes diagramas.

Puede haber un diagrama principal(0-5) y otros secundarios (10-14) que hacen determinadas funciones que una vez realizadas devuelven el control al diagrama principal.



Al llegar a la etapa 2 o 4 del primer diagrama se valida la transición $X2 + X4$ y empieza la subrutina. Al llegar a la etapa 14 se valida la transición $X1$. Continúa la evolución del diagrama principal a las etapas 3 o 5 respectivamente.

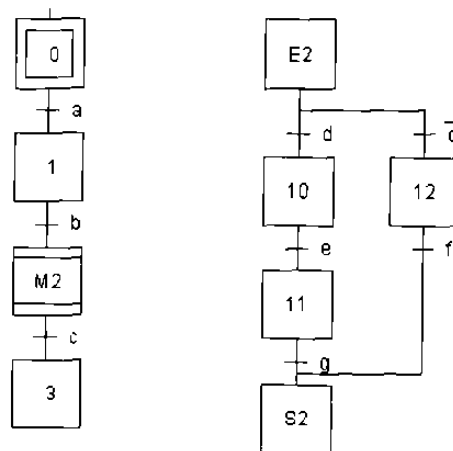
- Macro-etapas

Al hacer la descripción del automatismo, el Grafcet permite empezar desde un punto de vista general ya partir de él hacer descripciones cada vez más concretas del proceso de control.

El diseño se realiza de forma descendente, en grandes bloques que es van resolviendo de forma modular.

Una Macro-etapa es la representación mediante una única etapa, de un conjunto de etapas, transiciones y acciones asociadas, a las que llamamos expansión de la macro-etapa.

La expansión de la macro-etapa, es en realidad una parte del diagrama del Grafcet, con sus transiciones y normas de evolución, pero que en un diseño descendente hemos englobado en macro-etapa. Podríamos decir que al hacer la expansión de la macro etapa, en realidad lo que hacemos es una especie de zoom, que nos enseña en detalle, etapas, transiciones y acciones concretas, a las antes nos hemos referido de forma general.

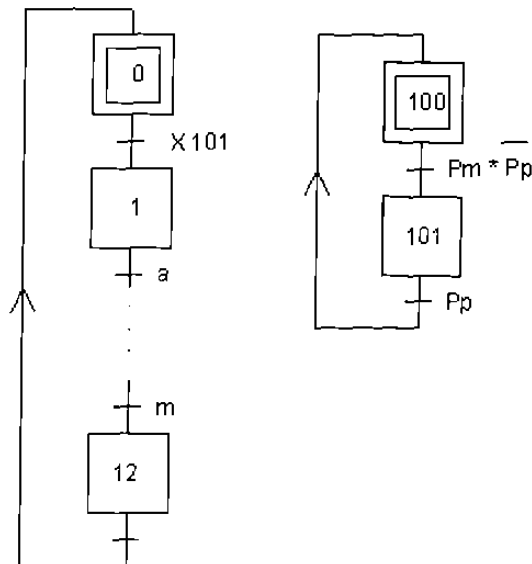


El diagrama principal evoluciona a partir de la tapa y la transición a, una vez que este activa la etapa 1 la transición b estará receptiva, y al validarse, entraremos a la macro etapa M2, la etapa E2 estará activa y según el estado de la transición d, evolucionará hacia la etapa 10 a la 12 y al llegar a la etapa 52 volverá al diagrama principal.

La etapa E2 es la etapa de entrada a la macro 2, la etapa 52 es la etapa de la salida.

- Diagramas paralelos

Para resolver un automatismo, se pueden describir diferentes diagramas paralelos, que evolucionaran cada uno de ellos por separado y a su ritmo. Estos, pueden tener en varios puntos relación entre sí.



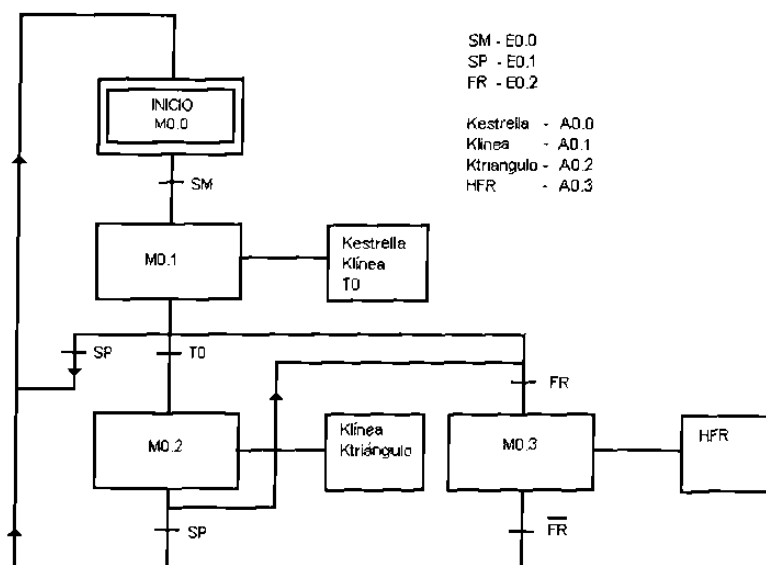
7.5 Desarrollo de ejemplos con Grafcet.

Ejemplo #1:

Como ejemplo práctico vamos a realizar un sencillo arranque Estrella - Triángulo (delta) de un motor trifásico asíncrono. El ciclo de funcionamiento se inicia con el botón de marcha (SM) que activa los contactores de estrella y línea, para que pasado un tiempo (T0) se desconecte la estrella y entre el triángulo. Además tendrá un relevador térmico (FR) que parará la maniobra en caso de avería y activará una luz de emergencia, y un botón de paro (SP).

En el diagrama se tienen que considerar todas las posibilidades de evolución del automatismo, en cada etapa las posibles transiciones que se pueden cumplir y como deben actuar. Debemos realizar el diagrama de tal manera que de una etapa pase a otra, considerando que cuando paremos la maniobra, este estado también será una etapa que no conectará nada (en nuestro caso).

El Grafcet asociado será el siguiente:



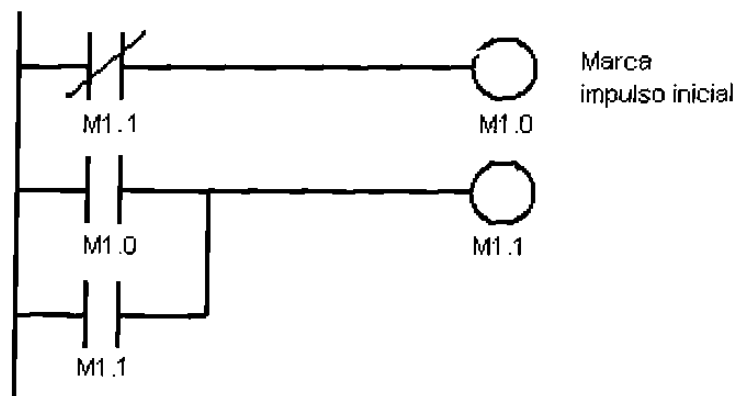
- Traducción a contactos

Constará de dos partes: la primera es la que va a describir el Grafcet y la segunda las acciones asociadas a realizar en cada etapa.

Grafcet

El Grafcet debe hacer seguir el camino de por donde tiene que ir el diagrama saltando de etapa a etapa cuando se cumpla la transición (o transiciones), y todo ello eléctricamente.

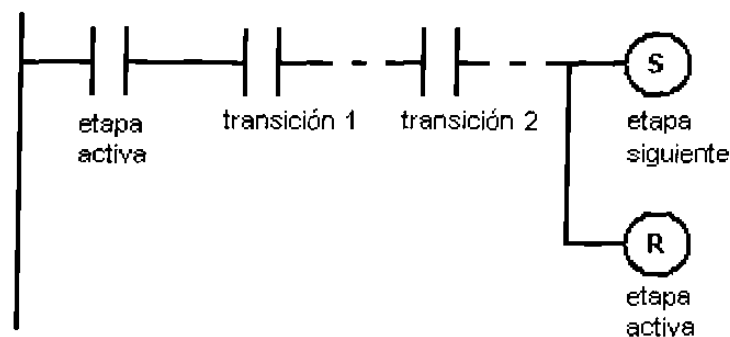
La etapa inicial es la única diferente por que debe activarse en el primer ciclo SCAN de la CPU al ponerse el autómata en RUN, y es la etapa de la cual partirá todo el Grafcet. Para iniciar esta etapa necesitamos un impulso inicial para activar esta etapa, podemos utilizar una instrucción especial (que no tiene el SIEMENS S-5) o generarlo nosotros mismos de la siguiente manera:



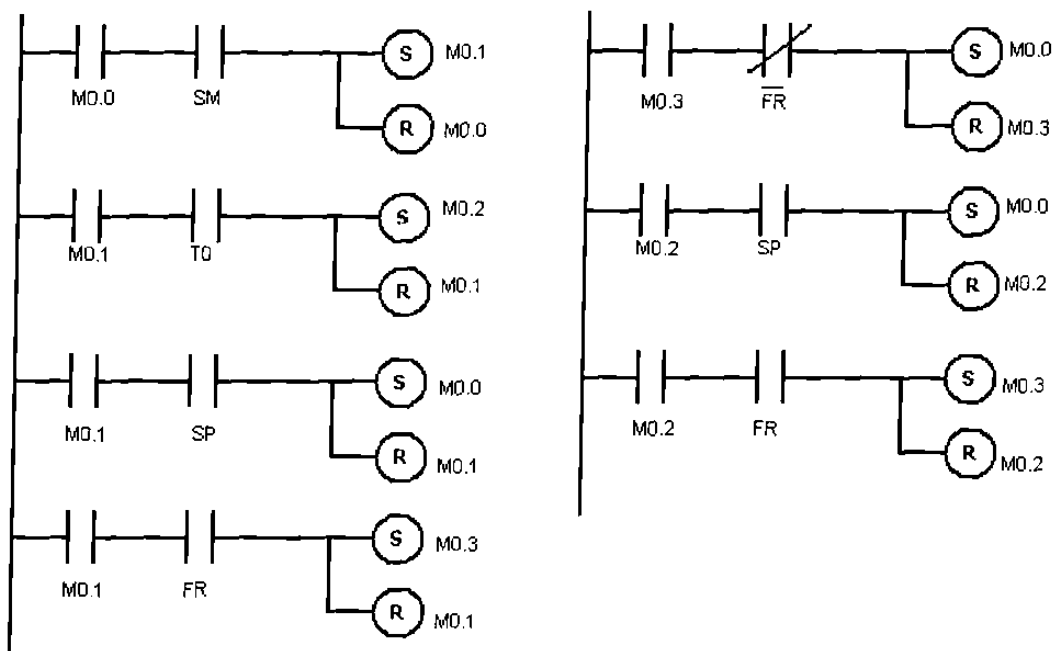
Este impulso inicial (conseguido con la marca M1.0) sólo hará que la etapa inicial se conecte al pasar la CPU a RUN, después de esto, las marcas M1.0 y M1.1 no harán nada más. En nuestro caso el contacto M1.0 activará M0.0. como la siguiente figura:



Con esta etapa ya hemos iniciado el Grafcet, para pasar a la siguiente etapa debemos esperar a la transición (una o varias) y cuando se cumpla cambiar a la etapa siguiente y desactivar donde estábamos, dejando SIEMPRE UNA SOLA ETAPA ACTIVA, así sucesivamente durante todo el diagrama. Una manera de hacerlo es el siguiente esquema:



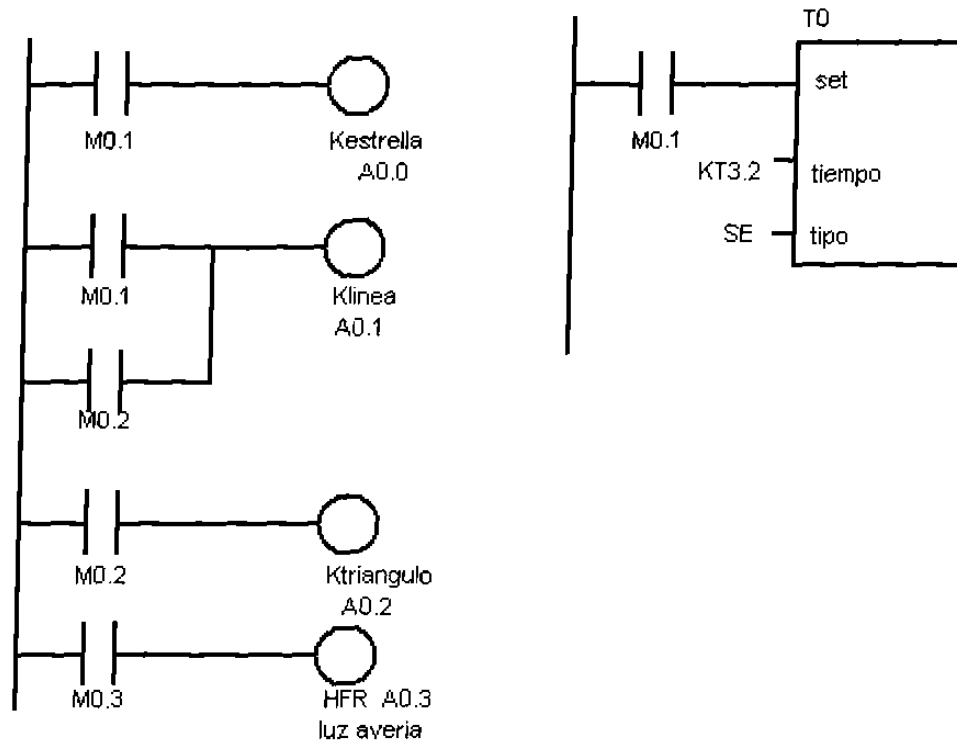
Si hay más de una transición que se deban cumplir todas se conectarán en serie (figura anterior), si se debe cumplir una de ellas solamente las pondremos en paralelo. El resto del Grafcet sigue de esta manera:



De cada etapa pueden derivar varios caminos en función de la transición que se cumpla, nosotros debemos contemplarla y activar donde vamos y desactivar donde estábamos, de esta manera sólo haremos caso a las entradas necesarias en cada etapa y el resto no influirá en la maniobra facilitando el seguimiento en caso de avería. Después de realizarlo podemos simplificar el esquema, pero lo dejaremos de esta manera para una mejor clarificación.

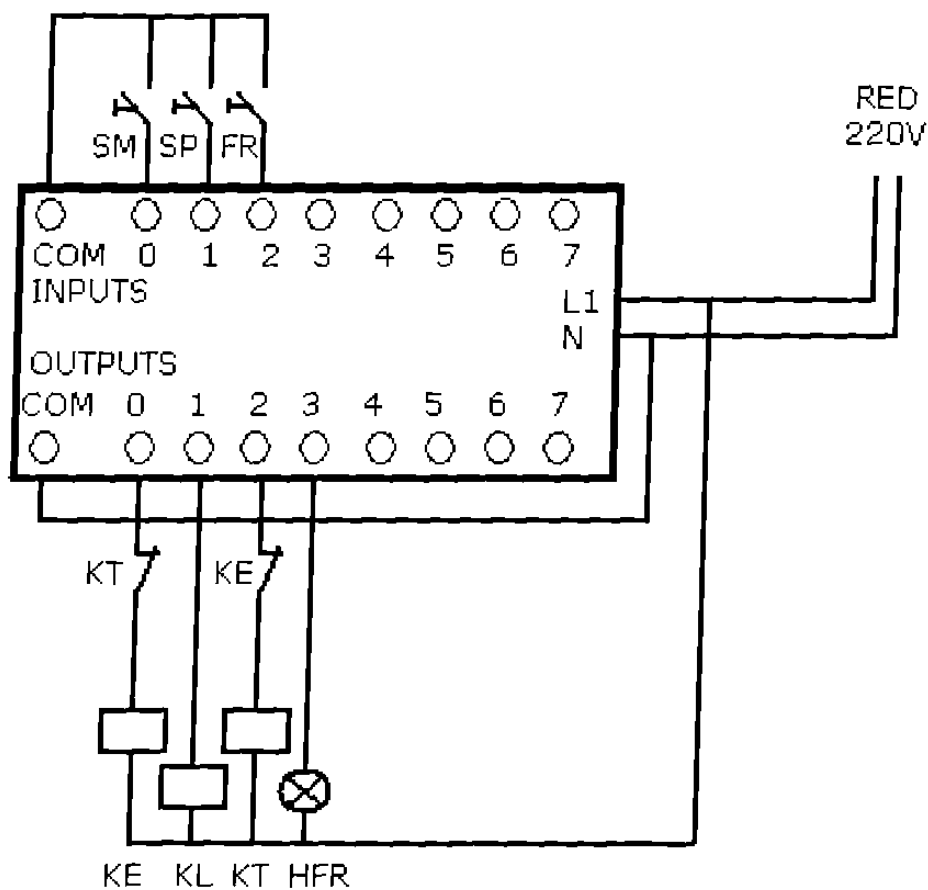
- Acciones asociadas

Ahora lo que debemos hacer es conectar las acciones asociadas a cada etapa de manera que sólo funcionen cuando la etapa este activa:



La salida A0.1 la conectan dos etapas, entonces la activaremos con un contacto de cada etapa en paralelo. El temporizador sólo funciona en la etapa M0.1.

El esquema de potencia queda igual. El de mando queda como indícala figura, se le han añadido dos contactos N. C., cruzados entre las bobinas K Estrella y K Triángulo para evitar que puedan funcionar ambas a la vez en el caso de que una de ellas quede clavada.



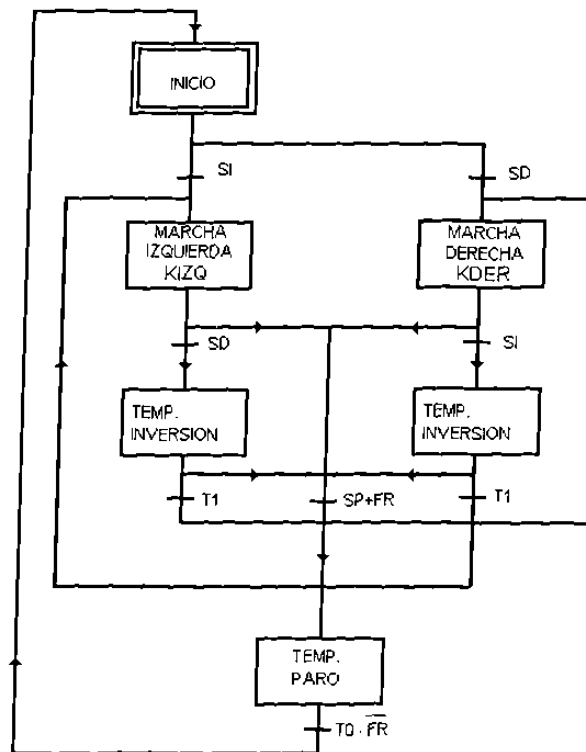
Ejemplo #2:

En este ejemplo analizaremos el Grafset y el esquema asociado a un inversor de giro de un motor con las siguientes condiciones:

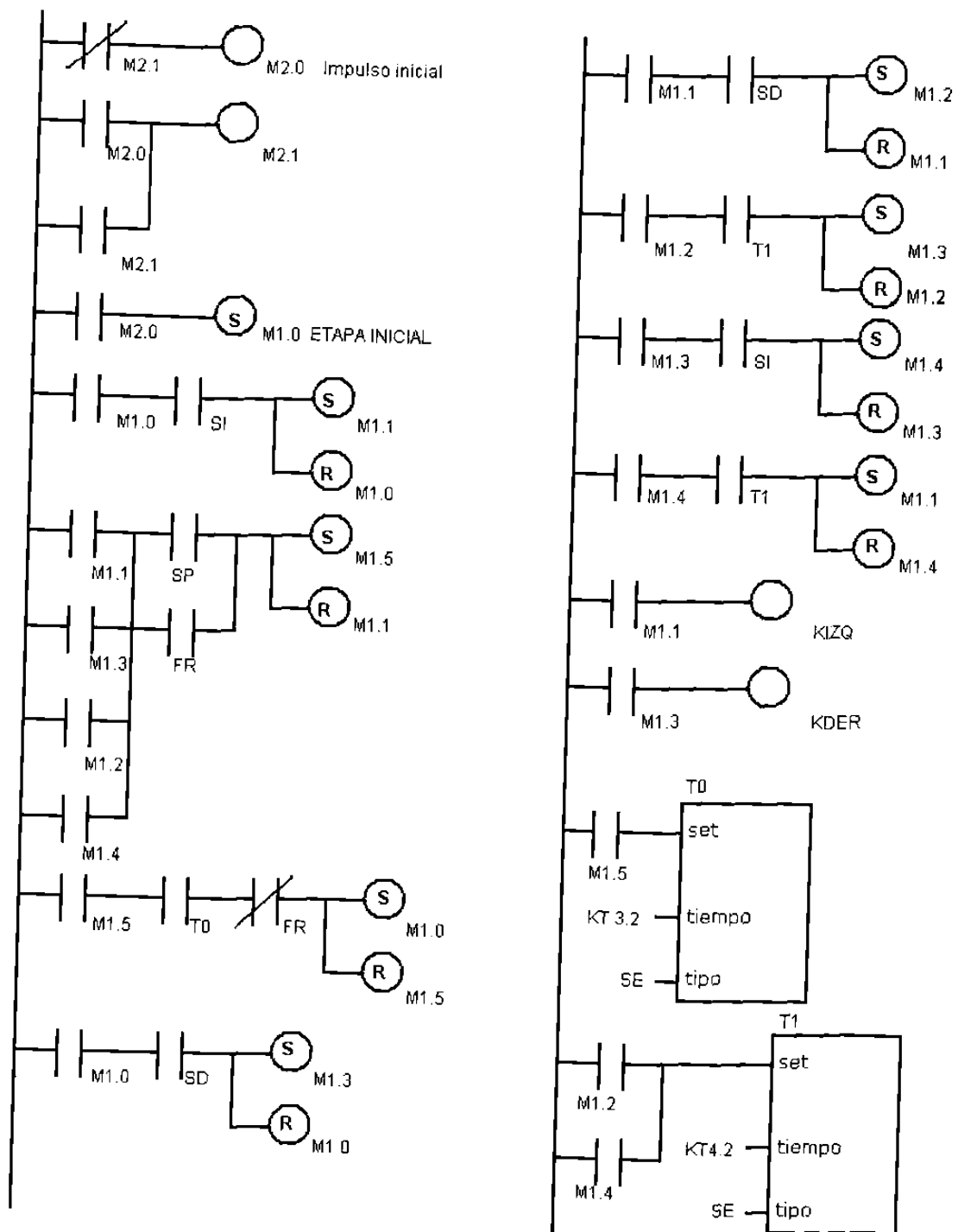
Botones de paro, izquierda, derecha y relevador térmico.

- Inversión sin pasar por paro y temporizada, o sea, al accionar cualquier botón para invertir el sentido se parará el motor y luego arrancará en el sentido contrario.
- Al accionar el paro también temporizará antes de poder arrancar de nuevo.

Una posible solución es:



El esquema asociado será:



Ejemplo #3:

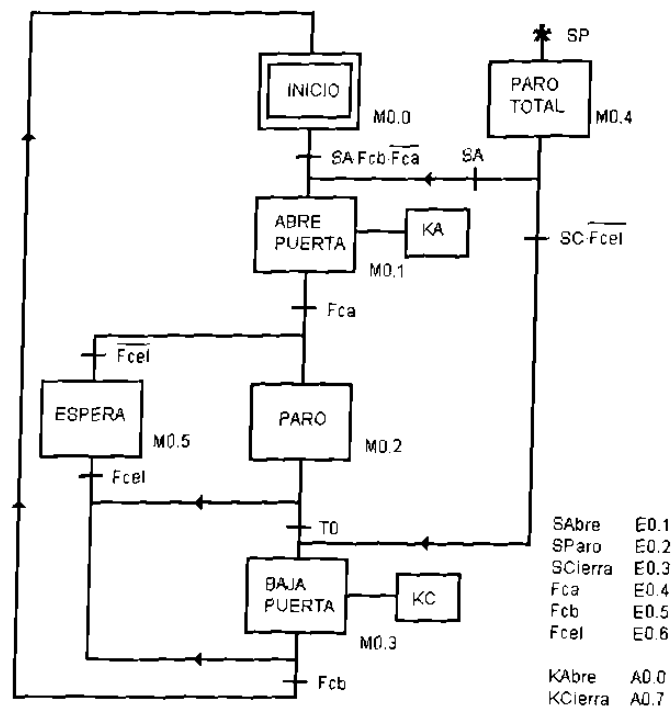
Describiremos el Grafset de una puerta de un garaje que cumpla las siguientes condiciones:

Botones de paro, de cerrar, abrir y sensores

Al accionar el botón de abrir (SAbrire) sube la puerta hasta tocar el sensor (Fca), una vez pasado un tiempo la puerta bajará automáticamente hasta tocar el sensor (Fcb) que se parará.

Si el sensor Fcel se interrumpe cuando la puerta está arriba o ha iniciado la bajada, ésta se parará y volverá a contar el tiempo programado para iniciar de nuevo el descenso.

En el caso de accionar botón de paro (SParo), la puerta se detiene y no funcionará hasta que se pulse del botón de abrir (SAbrire) o cerrar (SCerrar).



Ejemplo #4:

Programación de un caso real: Veamos como se realiza el programa para gestionar los movimientos de un Ascensor. Los botones de llamada, de cada una de las puertas y del interior de la cabina, gobiernan el sentido de giro del motor para la subida y la bajada. Los finales de carrera, situados en cada una de las plantas, detectan la posición del ascensor. La solución se realiza con un Grafcet de seis secuencias opcionales como el de la figura. Las transiciones de la parte superior, establecen las condiciones de funcionamiento del motor con productos de las variables. Las transiciones de la parte inferior, definen las paradas del motor al llegar a cada planta.

Ascensor:

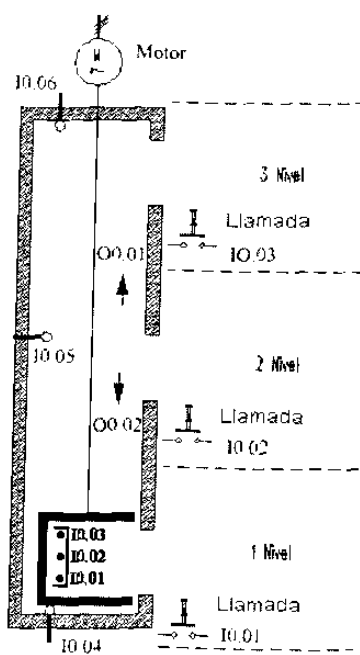
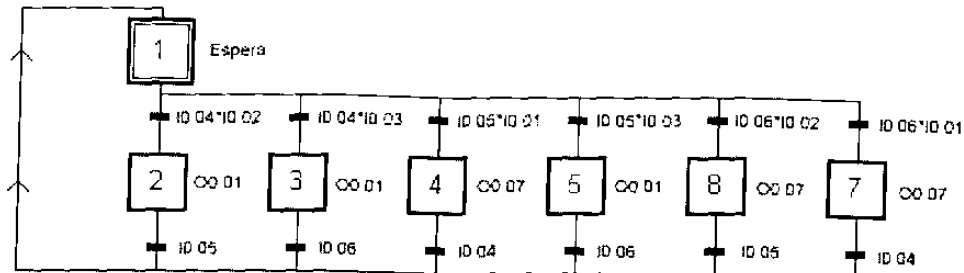
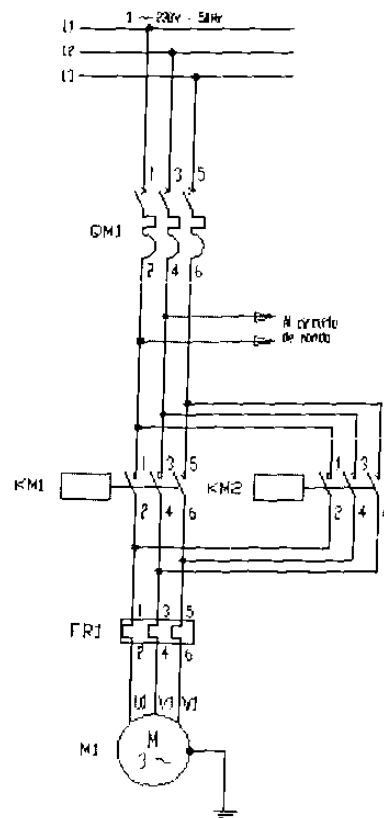


Gráfico Secuencial:



De esta forma es imposible que los dos contactores, que invierten el sentido de giro del motor trifásico, se activen a la vez.

Esquema de fuerza:



Capitulo 8 Conclusiones y recomendaciones

El uso de estas tecnologías en la automatización de procesos industriales no ha alcanzado su madurez ni popularidad debido a que los fabricantes de Controladores Lógicos Programables (PLC's) y los desarrolladores de lenguajes de programación no se ponen de acuerdo en la estandarización; que representaría un mejor aprovechamiento de los recursos y ventajas significativas en el desarrollo de nuevos diseños como en la actualización de sistemas ya implementados.

Se recomienda seguir muy de cerca los avances que se presenten en este campo ya que la tecnología cambia constantemente, la evolución es importante sobre todo en el campo de la automatización.

Seria interesante comprobar si con el paso del tiempo otros fabricantes además de Telemecanique y Aper adoptaran el uso del lenguaje Grafcet u otro lenguaje que ofrezca mas ventajas, entre ellas el de una programación lógica y amigable, que no requiriera de una gran experiencia o capacitación especial.

Bibliografía

- [1] Autómatas Programables,
José Luis Romeral, Serie Mundo Electrónico.
Editorial Marcombo, S.A., Boixareu Editores. 1997.
- [2] "Programmable Logic Controllers and Petri Nets"
IFAC, Madrid. September 1982
- [3] The Design and Modeling of PLC Programs Using Petri
Nets, Cambridge, England 1995.
- [4] GRAFCET: PRÁCTICA Y APLICACIONES J. C. BOSSY, P.
BRARD, P. FAUGERE, C. MERALUD Ediciones UPC, Aula Quadern,
1995.
- [5] PROGRAMMABLE LOGIC DEVICES AND LOGIC
CONTROLLERS E. MANDADO, J. MARCOS; Prentice Hall
International, 1996.

[6] Autómatas Programables, fundamentos, manejo, instalación y prácticas A. Porras / A P. Montanero Ed. Mc Graw Hill, publicación, 1990.

[7] Real Time Systems and their Programming and Languages, Ed. Addison-Wesley. Burns, A. and A. Welling, 1990.

[8] Internet

<http://www.autómatas.org>

<http://www.grupo-maser.com>

<http://www.femz.autómatas.com>

Listado de figuras

Figura 3.1 Esquema de un autómata programable.	12
Figura 4.1 Estructura externa de un PLC.	19
Figura 4.2 Diagrama describiendo la estructura interna de un PLC.	20
Figura 4.3 Elementos esenciales de un autómata programable.	21
Figura 4.4 Ciclo de trabajo de un autómata.	24
Figura 4.5 Dispositivos conectados como entradas y salidas.	25
Figura 4.6 Dispositivo utilizado como interfase.	25
Figura 4.7 Dispositivo periféricos.	27
Figura 5.1 Tabla de funciones combinacionales básicas y sus diferentes representaciones.	29
Figura 5.2 Operación And de cinco entradas.	30
Figura.5.3 Estructura representando un módulo.	33
Figura 6.1 Símbolos básicos de un esquema de contactos.	36
Figura 6.2 Rama de contactos.	37
Figura 6.3 Estructura de una instrucción LOAD.	37
Figura 6.4 Simbología de un contacto cerrado.	38
Figura 6.5 Paralelo de dos o más bobinas.	38
Figura 6.6 Pasos y transiciones de un Grafcet.	40
Figura 6.7 Secuencia que describe un listado de instrucciones.	41

Figura 6.8 Diagrama de contactos escalera.	42
Figura 6.9 Diagrama de función modular o block.	43
Figura 7.1 Etapa inicial sin retorno.	46
Figura 7.2 Etapa inicial con retorno.	46
Figura 7.3 Etapa normal.	47
Figura 7.4 Etapa normal activa.	47
Figura 7.5 Simbología de una etapa para una acción asociada.	48
Figura 7.6 Esquema de una etapa condicionada.	49
Figura 7.7 Simbología de la Norma IEC-848 para las acciones.	49
Figura 7.8 Etapa de salida de una transición.	52

Glosario de términos

Álgebra de Boole: Sistema algebraico donde las proposiciones lógicas se indican por medio de símbolos y se relacionan mediante operadores que corresponden a las leyes de la lógica.

API: (Autómata Programable Industrial). Aparato electrónico que sustituye circuitos auxiliares por sistemas automáticos.

Bifurcación: Punto donde se separan dos o más vías o caminos.

Bit: Contracción de Dígito Binario (Binary Digit).

Bloque: Está formado por varios elementos lógicos con interconexión programable entre sí. Varios bloques interconectados forman el dispositivo.

BYTE: Grupo de ocho bits.

Contacto: Dispositivo que abre y/ o cierra un circuito eléctrico.

CPU: Unidad Central de Proceso.

EEPROM:(Electrically Erasable Programmable Read-OnlyMemory): Memoria programable de sólo lectura eléctricamente borrable. Un tipo de memoria semiconductor.

ENCODER: Dispositivo que transforma señales de línea a código.

EPROM: (Erasable Programmable Read-Only Memory): Memoria de sólo lectura programable y borrrable.

Evento: Acontecimiento significativo que tiene una ubicación en el tiempo y en el espacio, puede tener parámetros.

Flash: Tecnología de memorias no volátiles. Los dispositivos con esta tecnología son borrados y programados eléctricamente.

Hardware: Parte física o tangible de un equipo de computo.

Kit: Conjunto de partes listas para ensamblar.

Load: Instrucción de carga del programa.

PLC: (Controlador Lógico Programable).

Método: La implementación de una operación. Procedimiento que permite llegar al resultado de una operación.

NEMAC: National Electrical Manufacturer Association.

Relevador: Dispositivo electromecánico, compuesto de una bobina o electroimán y contactos.

Software: Programas o aplicaciones utilizadas por la computadora.

SPC: (Gráfico Secuencial de Funciones).

ST: Texto estructurado, lenguaje de alto nivel estructurado por bloques.

TRANSDUCTORES: Convierten las magnitudes físicas en magnitudes eléctricas.

Resumen Autobiográfico

Patricia Aguilera Martínez, nacida el 26 de Agosto de 1968, en la ciudad de Monterrey, N. L. México siendo sus padres el Sr. Adolfo Aguilera Rangel y Sra. Demetria Martínez Pérez.

Sus estudios de primaria y secundaria fueron en la escuela Federico Herrera González y Profesor Sigifredo H. Rodríguez, en Monterrey, N. L., posteriormente estudio en la Preparatoria #2 de la U. A. N. L.

Cursó sus estudios en la Facultad de Ingeniería Mecánica y Eléctrica, de la Universidad Autónoma de Nuevo León, logrando obtener el título de la Licenciatura de Ingeniero en Control y Computación, perteneciente a la generación de Agosto de 1985 a Enero de 1990.

En Diciembre de 1990 ingresa a trabajar en la empresa estadounidense Música Intercomunicación y Sonido de México, en Apodaca, N. L. durante un período de 2 años.

Realizó en 1994 sus estudios del idioma Inglés en la institución Harmon Hall obteniendo el nivel profesional satisfactoriamente, logrando también con esto poder aplicar a las diferentes plazas vacantes donde se requería un segundo idioma.

Recibe la oportunidad de viajar a los Estados Unidos en representación de la empresa Inoac Polytec de México en 1997, realizando importantes logros en el área de Ingeniería con un producto de la compañía SARP .

Comienza un nuevo reto para la empresa Nortel en el año de 1997, logrando con esto que su carrera vaya en ascenso, perfilándose a nuevos retos en el ramo de la Manufactura.

En el año de 1998 comienza sus estudios de maestría en la carrera de Ingeniería de Manufactura con especialidad en Automatización.

En 1997 le invitan a elaborar en una nueva área de interés personal, Calidad, para la empresa SQUARE D. Viaja en representación de la compañía a Estados Unidos y Canadá, logrando con ello que su interés profesional por desarrollarse en las áreas relacionadas con Ingeniería, Manufactura y Calidad crezca.

Actualmente labora en la empresa canadiense Triton ubicada en Apodaca N. L., como responsable del área de Aseguramiento de la Calidad realizando nuevas metas, siendo la certificación de la empresa en las normas ISO 9002-94 y la IEC 17025 la próxima a lograr.

Es por ello que opto como requisito para obtener el grado de Maestro en Ciencias de la Ingeniería de Manufactura con Especialidad en Automatización en la Facultad de Ingeniería Mecánica y Eléctrica, FIME, de la Universidad Autónoma de Nuevo León, UANL, Mexico, desarrollar la tesis "Programación de PLC´s".

