TABLA DE CONTENIDO

LISTADOS DE FIGURAS

RESUMEN

El término PLC de amplia difusión en el medio significa en inglés, Controlador Lógico Programable. Originalmente se denominaban PCs (Programmable Controllers), pero, con la llegada de las IBM PCs, para evitar confusión, se emplearon definitivamente las siglas PLC.

En Europa, el mismo concepto es llamado Autómata Programable. La definición más apropiada es: Sistema Industrial de Control Automático que trabaja bajo una secuencia almacenada en memoria, de instrucciones lógicas. Es un sistema porque contiene todo lo necesario para operar, e industrial por tener todos los registros necesarios para operar en los ambientes hostiles encontrados en la industria.

Esta familia de aparatos se distingue de otros controladores automáticos en que puede ser programado para controlar cualquier tipo de máquina, a diferencia de otros muchos que, solamente pueden controlar un tipo específico de aparato. Un programador o Control de Flama de una caldera, es un ejemplo de estos últimos. Además de poder ser programados, se insiste en el término "Control Automático", que corresponde solamente a los aparatos que comparan ciertas señales provenientes de la máquina controlada de acuerdo con algunas reglas programadas con anterioridad para emitir señales de control para mantener la operación estable de dicha máquina.

Las instrucciones almacenadas en memoria permiten modificaciones así como su monitoreo externo. Los controladores PLC fueron desarrollados

como una alternativa flexible y económica a los viejos controladores lógicos de relevo utilizados en procesos y controles industriales automatizados. Tienen algunas ventajas sobre los controladores lógicos de relevo, entre ellas, el costo, el tamaño, la fidelidad y otras características que los antiguos sistemas carecen. Su desarrollo se debe a la misma tecnología del microchip y del microprocesador. Esto permite una más fácil programación de los programas de control almacenados dentro de la memoria del PLC y elimina la necesidad de alterar el hardware cada vez que se tenga que implementar un nuevo proceso de control. Los controladores PLC son económicos y compactos, son unidades versátiles basadas en la arquitectura estándar de microprocesadores, utilizados en el control de procesos o de maquinas. Fueron diseñados para facilitar su programación y mantenimiento. Los sistemas PLCs reemplazan de forma rápida, fácil y eficiente a los antiguos y aparatosos sistemas controladores lógicos de relevo utilizados en manufactura automatizada. Los PLC's, fueron definidos en 1978 por la Asociación Nacional (EUA) de Manufactureros Eléctricos (NEMA) como: "Un aparato electrónico de operación digital, el cual utiliza una memoria programable que almacena internamente las instrucciones para implementar funciones especificas, tales como, lógicas, secuenciales, de tiempo, de conteo, y aritméticas; Para controlar por medio de entradas y salidas digitales o analógicas, varios tipos de maquinas o procesos.".

Los PLC's operan básicamente detectando las señales de entrada lógicas (on/off) o analógicas, y dependiendo de los programas de control o diagramas de escala (Ladder), produce señales de salidas del mismo tipo (normalmente lógico). En la implementación de PLCs, el campo de cableado o conexiones entre los elementos lógicos permanece inalterado, pero no se necesitaran mas conexiones físicas. Por el contrario, las conexiones son almacenadas en memoria de computadora, lo cual permite la programación

de dichas conexiones y por ende facilita más las cosas al pasar al ámbito o escala lógica. La escala lógica (Ladder logic) es una técnica de programación que requiere un mínimo entrenamiento de programación.

Los sistemas PLC's, sobre los antiguos sistemas de relevo, tienen ventajas considerables:

- Realizan todas las capacidades y funciones de los sistemas antiguos.
- Mucho mayor y mejor desempeño.
- Mayor fidelidad.
- Requieren poco mantenimiento ya que no utilizan partes movibles.
- No requiere habilidades especiales de programación para su mantenimiento.
- Su tamaño físico es mucho menor que los antiguos sistemas convencionales.
- Y lo más importante, un costo mucho menor.

Aunque los sistemas PLCs tienen muchas ventajas, también tienen sus desventajas. Entre ellas esta la localización de fallas debido a su diseño más complejo que los anteriores. Segundo, la falla del PLC puede detener por completo los procesos que controla, mientras que una falla en un sistema convencional solo lo interrumpe parcialmente. Y tercero, la interferencia eléctrica puede alterar o interrumpir la memoria del PLC.

INTRODUCCIÓN

Hasta no hace mucho tiempo el control de procesos industriales se venia haciendo de forma cableada por medio de contactores y relés. Al operario que se encontraba a cargo de este tipo de instalaciones, se le exigía tener altos conocimientos técnicos para poder realizarlas y adicionalmente mantenerlas. Además cualquier variación en el proceso suponía modificar físicamente gran parte de las conexiones de los montajes, para lo cual era necesario un gran esfuerzo técnico y un mayor desembolso económico.

Los Controladores Lógicos Programables, PLC, son unidades de control con un hardware estándar y modular, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente industriales) y programable por el usuario. El Controlador Lógico Programable, PLC, nació como solución al control de circuitos complejos de automatización. Un Controlador Lógico Programable, PLC, no es más que un aparato electrónico que sustituye los circuitos auxiliares o de mando de los sistemas automáticos. Aunque en realidad sus prestaciones actuales van mucho mas allá

A lo largo de este Tutorial, se intentara lograr que las personas que lo utilicen tengan una herramienta de fácil entendimiento, que les sea útil durante el aprendizaje del uso y programación de Controladores lógicos Programables.

Capitulo1. INTRODUCCIÓN A LOS CONTROLADORES LÓGICOS PROGRAMABLES

1.1 QUE SON LOS AUTÓMATAS PROGRAMABLES

Un Controlador lógico Programable, es un dispositivo que fue inventado para reemplazar los circuitos de reles secuenciales necesarios para maquinas de control. El Controlador Lógico Programable trabaja leyendo sus entradas y dependiendo de su estado, varia las salidas de acuerdo a la lógica que se plantee[1].

Un controlador lógico programable es una maquina electrónica programable, por personal no informático, destinada a cumplir funciones de automatismos lógicos y control de procesos de manufactura, en ambiente industrial y tiempo real, tanto sean de tipo combinacional o secuencial.

Esta definición no debe interpretarse en forma rigurosa ya que los controladores lógicos programables modernos incorporan funciones especiales no solo de tratamiento lógico sino también de calculo numérico, regulación de PID y de servocontrol.

1.2 RESEÑA HISTORICA

A finales de los 60's aparecieron los primeros Controladores Lógicos Programables, PLC. La razón primaria para diseñar semejante dispositivo era eliminar el alto costo involucrado en los complicados sistemas de maquinas de control basados en reles. *Bedford Associates (Bedford, MA)* propuso algo llamado, *Director Digital Modular (MODICON)* a uno de los mas grandes fabricantes de automóviles en Estados Unidos. Al mismo tiempo

[1] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001. Pág.16

otras compañías propusieron esquemas basados en computadores. *El MODICON 084* comercializo el primer Controlador Lógico Programable, PLC, del mundo[2].

Cuando los requisitos de la producción cambian también lo hace el sistema de control, esto llega a ser muy costoso cuando el cambio es frecuente. Los reles eran dispositivos mecánicos que tenían una vida útil limitada y requerían un estricto mantenimiento. Arreglarlo era también bastante tedioso cuando habían muchos reles envueltos. Ahora imagínese un panel de control que incluía ciento o miles de reles. El tamaño era exagerado. Complicado como el alambrado eléctrico de muchos dispositivos individuales, estos reles se alambraban interconectándolos de manera que rendiría el resultado deseado.

Estos *"nuevos controladores"* tenían que ser fáciles de programar por los ingenieros de planta. La vida útil tenía que ser larga y los cambios de programación fácilmente realizados. Estos también tenían que sobrevivir al duro ambiente industrial. La solución era usar una técnica de programación con la cual la mayoría de las personas estuvieran familiarizadas y pudieran reemplazar las partes mecánicas con las transistorizadas.

En 1970 con la evolución del microprocesador aparecen en los Controladores Lógicos Programables, PLC, las funciones de calculo, las funciones secuenciales, funciones de manipulación de datos. El *AMD 2901 y 2903* eran bastante populares en *Modicon y el* Controlador Lógico Programable *Allan Bradley.*

Los desarrollos en las comunicaciones aparecieron aproximadamente en 1973. El primer sistema era el Modbus de Modicon. El Controlador Lógico

[2] http://www.plcs.net/01What is a PLC.htm

Programable, PLC, podía comunicarse con otro PLC y además podían estar lejos de la máquina que estaban controlando. También podían usarse para enviar y recibir señales de voltajes variantes que les permitían entrar en el mundo análogo. Desgraciadamente, la falta de estandarización acompañada con los cambios continuos en la tecnología ha hecho de la comunicación del Controlador Lógico Programable, PLC, una pesadilla de incompatibles protocolos y redes físicas.

En los 80's se vio un esfuerzo por estandarizar las comunicaciones con *General Motor's* fabricando el protocolo de automatización (*MAP*). También era tiempo de reducir el tamaño y hacerles software programable a través de la programación simbólica en las computadoras personales en lugar de comandos de programación especializados o programadores manejables.

Los 90 han visto una reducción gradual en la introducción de nuevos protocolos, y la modernización de las capas físicas de algunos de los protocolos más populares que sobreviven de los 80's. La norma (*IEC 1131-3*) ha intentado unir los lenguajes de programación del Controlador Lógico Programable, bajo un estándar internacional. Ahora tenemos Controladores Lógico Programables, que son programable con diagramas de Bloque de Funciones, Listas de Instrucciones, y Texto Estructurado todos al mismo tiempo. Los PC también está usándose para reemplazar los Controladores Lógicos Programables en algunas aplicaciones.

1.3 CAMPOS DE APLICACIÓN

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

- Espacio reducido.

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

Aplicaciones generales:

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

Tal y como dijimos anteriormente, esto se refiere a los autómatas programables industriales, dejando de lado los pequeños autómatas para uso más personal (que se pueden emplear, incluso, para automatizar procesos en el hogar, como la puerta de una cochera o las luces de la casa).

1.4 VENTAJAS E INCONVENIENTES DE SU UTÍLIZACIÓN

- Menor tiempo de elaboración de proyectos.
- Posibilidad de añadir modificaciones sin costo añadido en otros componentes.
- Menor espacio de ocupación.
- Menor costo de mano de obra.
- Mantenimiento más económico.
- Posibilidad de gobernar varias máquinas con el mismo Controlador Lógico Programable.
- Menor tiempo de puesta en funcionamiento.

- Si el Controlador Lógico Programable, queda pequeño para el proceso industrial puede re-programarse para seguir siendo de utilidad en otras máquinas o sistemas de producción.
- Son útiles para poner las salidas en on/off basándose en el estado de entradas. (control)
- Buenos en reunir y concentrar datos y estados que son cargados en un computador de forma compacta.
- Son más robustos que los computadores y en los últimos cinco, siete, diez años sin necesitar reemplazo.
- Adiestramiento de personal.
- Son eficientes manejando grandes cantidades de datos, complejos, o funciones matemáticas avanzadas.
- Buenos en leer y escribir bancos de datos.
- Buenos para generar informes.
- Buenos en desplegar datos e información al operador.

1.5 FABRICANTES

Entre los principales productores de Controladores Autómatas programables tenemos a los siguientes:

- AB Allan Bradley.
- Mitsubishi.
- Schneider.
- Siemens.

De los cuales presentaremos enlaces a diferentes manuales que se presentaran en el Tutorial.

Capitulo 2. ESTRUCTURA DE LOS CONTROLADORES LÓGICOS PROGRAMABLES

Básicamente un Autómata Programable se divide en dos sistemas funcionales principales uno es la unidad central, y el sistema de entradas y salidas. La estructura de los autómatas pueden clasificarse atendiendo a los conceptos de:

- ***Estructura Modular***
- ***Estructura Compacta***.

La estructura modular divide en distintos módulos dedicados las estructuras funcionales anteriormente referidas, de tal forma, físicamente existen módulos tanto para la Unidad Central De Procesos CPU como para los distintos módulos de entradas / salidas[3]. La principal ventaja de esta disposición radica en la posibilidad de adecuar la arquitectura del sistema a las necesidades estrictas de diseño y funcionamiento. También permite el funcionamiento parcial del sistema en caso de averías al tiempo que hace posible la reducción de los tiempos de reparación de forma notable. La estructura compacta resulta adecuada para pequeñas aplicaciones con un numero prefijado de entradas/salidas.

[3] SIMONS, Andrés. Autómatas Programables. Paraninfo, Madrid, 1988.

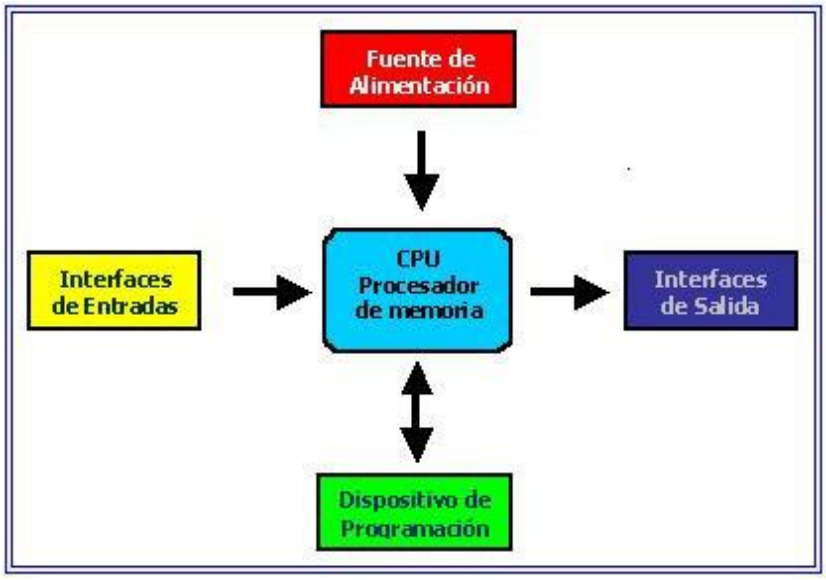


Figura 1. Estructura de un PLC

2.1 FUENTES DE ALIMENTACIÓN

Es la encargada de convertir la tensión de la red (120 - 220) VCA, a baja tensión de C.C, normalmente 24Vcc. Siendo esta la tensión de trabajo en los circuitos electrónicos que forman el Controlador Lógico Programable. La fuente de alimentación puede ser externa (modulo adicional a un lado o cerca de la Unidad Central de Procesos CPU) o interna (incluida dentro del equipo).

2.2 DISPOSITIVOS DE PROGRAMACIÓN

El terminal o consola de programación es el que permite comunicar al operario con el sistema. Las funciones básicas de éste son las siguientes:

- Transferencia y modificación de programas.
- Verificación de la programación.
- Información del funcionamiento de los procesos.

Como Consolas de Programación pueden ser utilizadas las construidas específicamente para el Controlador Lógico Programable, tipo dispositivos de mano o bien un PC, que soporte un software especialmente diseñado para resolver los problemas de programación y control.

2.3 UNIDAD CENTRAL DE PROCESOS CPU

La Unidad Central de Procesos CPU es el auténtico cerebro del sistema. Se encarga de recibir las ordenes, del operario por medio de la consola de programación y el modulo de entradas. Posteriormente son procesadas para enviar respuestas al módulo de salidas. En su memoria se encuentra residente el programa destinado a controlar el proceso.

2.3.1 Estructura Interna

Son las partes en que se ordena su conjunto físico o hardware y las funciones y funcionamiento de cada una de ellas.

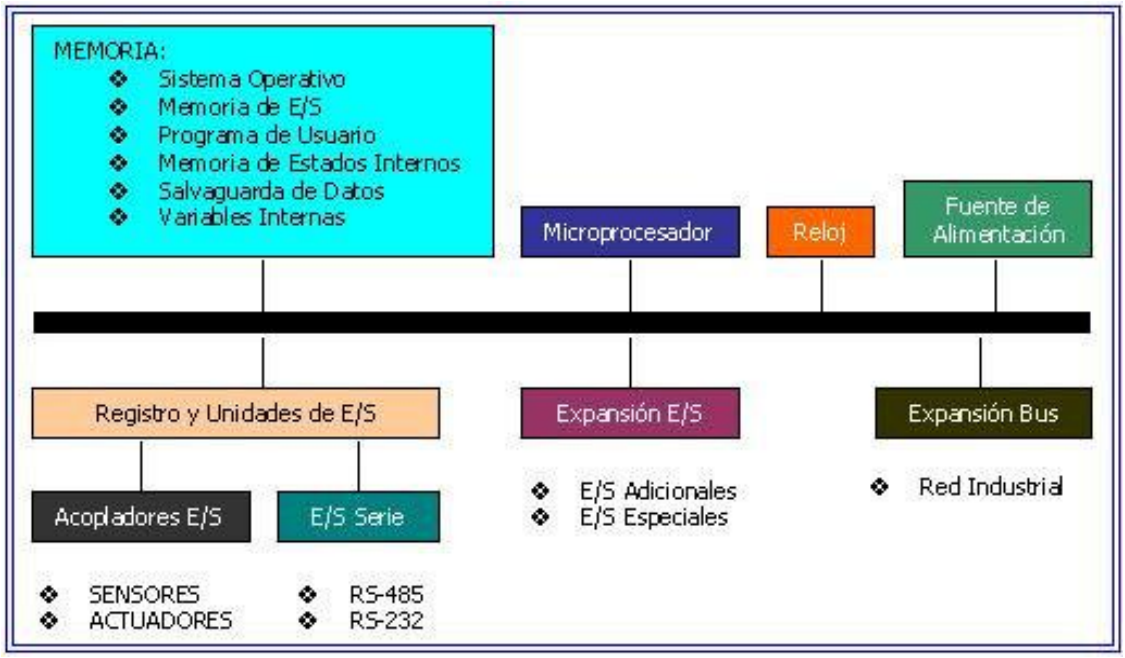


Figura 2. Estructura Interna

2.3.1.1 Memoria

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

- Memoria del Programa de Usuario: aquí introduciremos el programa que el Controlador Lógico Programable, va a ejecutar cíclicamente.
- Memoria de la tabla de datos: se suele subdividir en zonas según el tipo de datos (marcas de memoria, temporizadores, contadores, etc.).
- Memoria del sistema: aquí se encuentra el programa en código máquina que monitoriza el sistema (programa del sistema o firmware). Este programa es ejecutado directamente por el microprocesador que posea el Controlador Lógico Programable.
- 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 Controlador Lógico Programable, divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante.

2.3.1.2 Microprocesador

El microprocesador es el corazón del Controlador Lógico Programable. Es el encargado 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 excede 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.
- Chequeo del sistema.

Figura 3. Ciclo de trabajo del Microprocesador

2.3.1.3 Reloj

Marca el funcionamiento de todo el sistema, comúnmente a una frecuencia de 20MHz. No existe físicamente. Vienen en muchas variedades e incrementos. El más común es con retraso. Otros incluyen sin retraso y los tipos retentivos y no retentivos. Los incrementos varían de 1ms hasta 1s[4].

[4] MURILLO, Luis. Curso Básico de Controladores Lógicos Programables (PLCS)

2.3.2 Estructura Externa

Todos los Controladores Lógicos Programables, poseen una de las siguientes estructuras:

- Compacta: En un solo bloque están todos lo elementos. Este tipo de autómatas se distingue por presentar en un solo bloque todos sus elementos, esto es, fuente de alimentación, CPU, memorias, entradas/salidas, etc.

Son los autómatas de gama baja o nanoautómatas los que suelen tener una estructura compacta. Su potencia de proceso suele ser muy limitada dedicándose a controlar máquinas muy pequeñas o cuadros de mando.

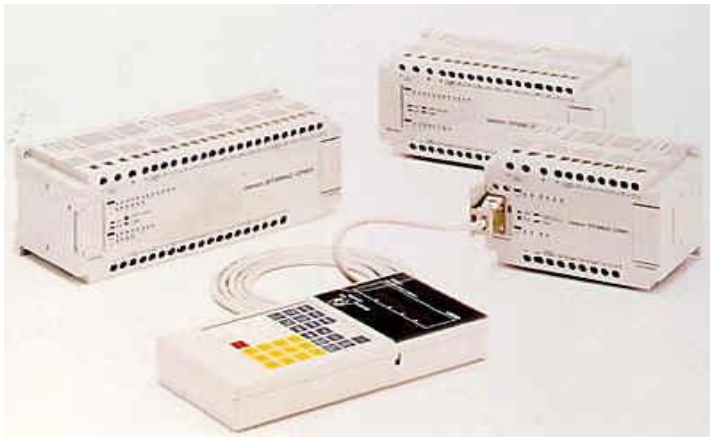


Figura 4. Estructura Compacta

- Modular: estructura americana, separa las E/S del resto del autómata. Estructura europea, cada módulo es una función (fuente de alimentación, Unidad Central De Procesos CPU, E/S, etc.).

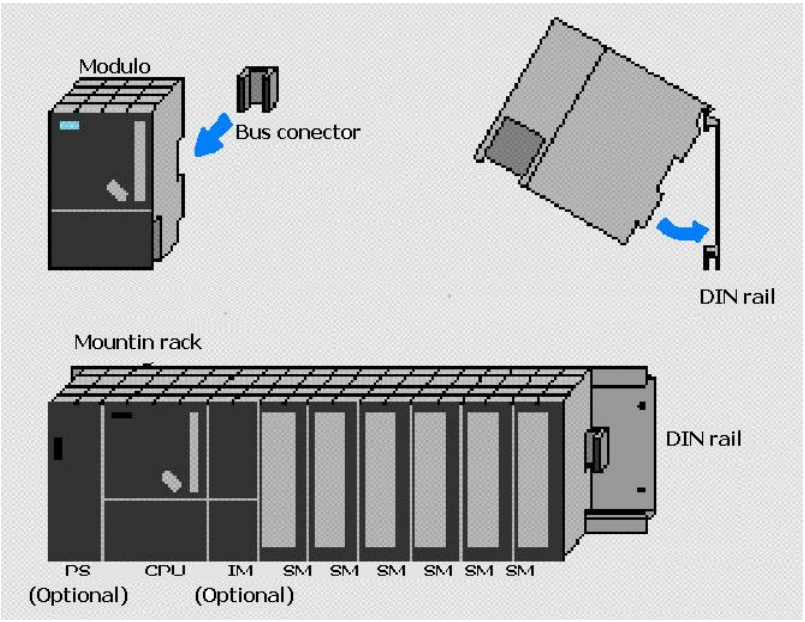


Figura 5. Estructura Modular

Exteriormente nos encontraremos 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 o racks normalizados, para que el conjunto sea compacto y resistente.

2.4 UNIDADES DE ENTRADA Y SALIDA DE DATOS

Las E/S son leídas y escritas dependiendo del modelo y del fabricante, es decir pueden estar incluidas sus particularidades dentro del área de memoria o ser manejadas a través de instrucciones específicas de E/S. Estas entradas y salidas se manejan a nivel de bit, Byte, Palabra, dentro del programa de usuario.

Las entradas se unen eléctricamente a través de captadores y sensores (interruptores, finales de carrera, pulsadores,...). Estos son de diverso tipo de acuerdo a la señal eléctrica que reciben (voltajes, corrientes, resistencia, etc.). La información recibida en él, es enviada a la Unidad Central De Procesos CPU para ser procesada de acuerdo a la programación residente.

Generalmente vamos a disponer de dos tipos de entradas: Digital y Análoga. Las entradas digitales se basan en el principio de todo o nada, es decir o no conducen señal alguna o poseen un nivel de tensión. Estas entradas se manejan a nivel de bit dentro del programa de usuario. Las entradas digitales generalmente son de los siguientes tipos:

- Corriente Alterna (CA) de 120/220 V
- Corriente Directa (CD) de 12/24V

Son muchos los circuitos de entrada digitales mas comunes son los siguientes que nombramos a continuación:

Entrada digital tipo "Source", 110/220 Vca

Las entradas vienen comúnmente en grupos de cuatro u ocho y cada grupo tiene un punto comun (C) que permite manejar diferentes voltajes y disminuir el cableado de campo.

A).

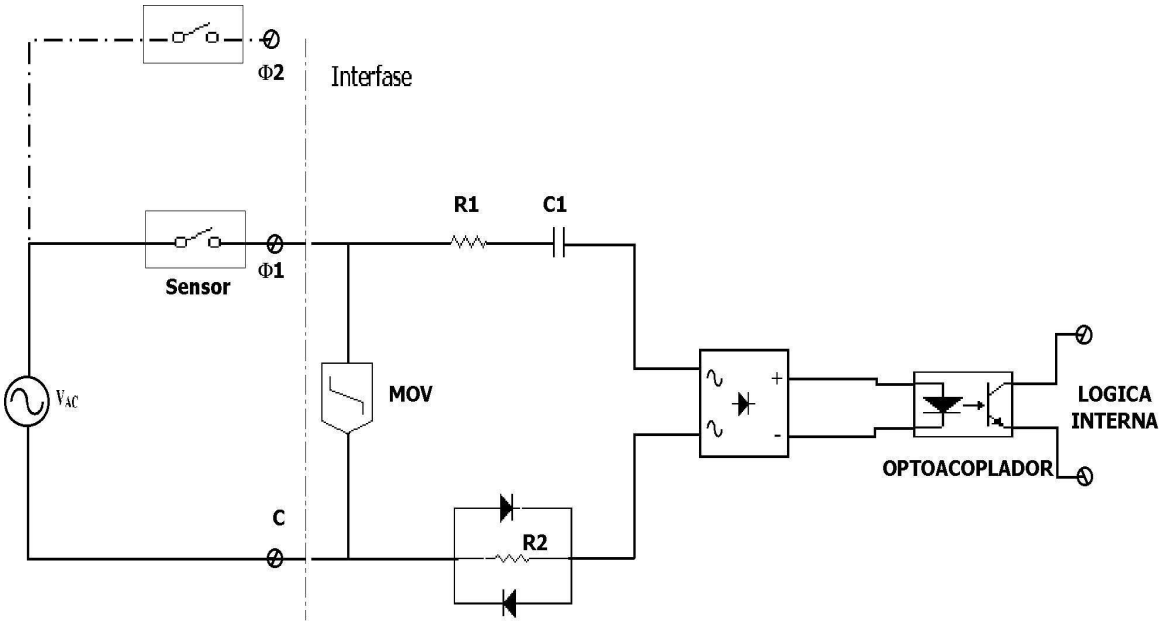


Figura 6. Entrada digital tipo "Source"

B)

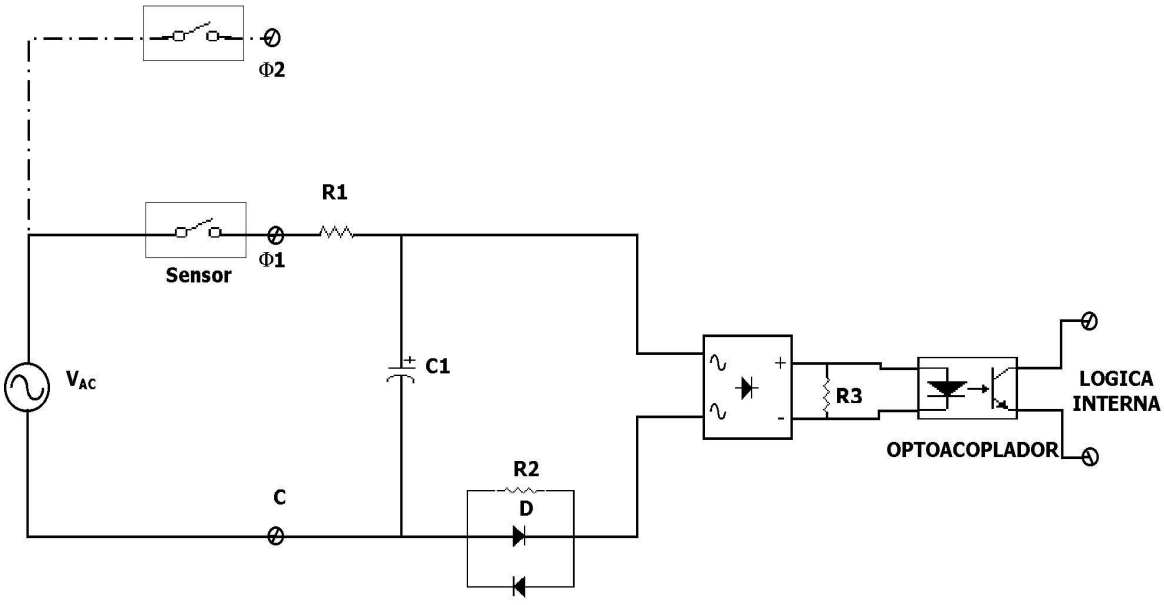


Figura 7. Entrada digital tipo "Source"

Entrada digital tipo "Sink", 24 Vcd

A)

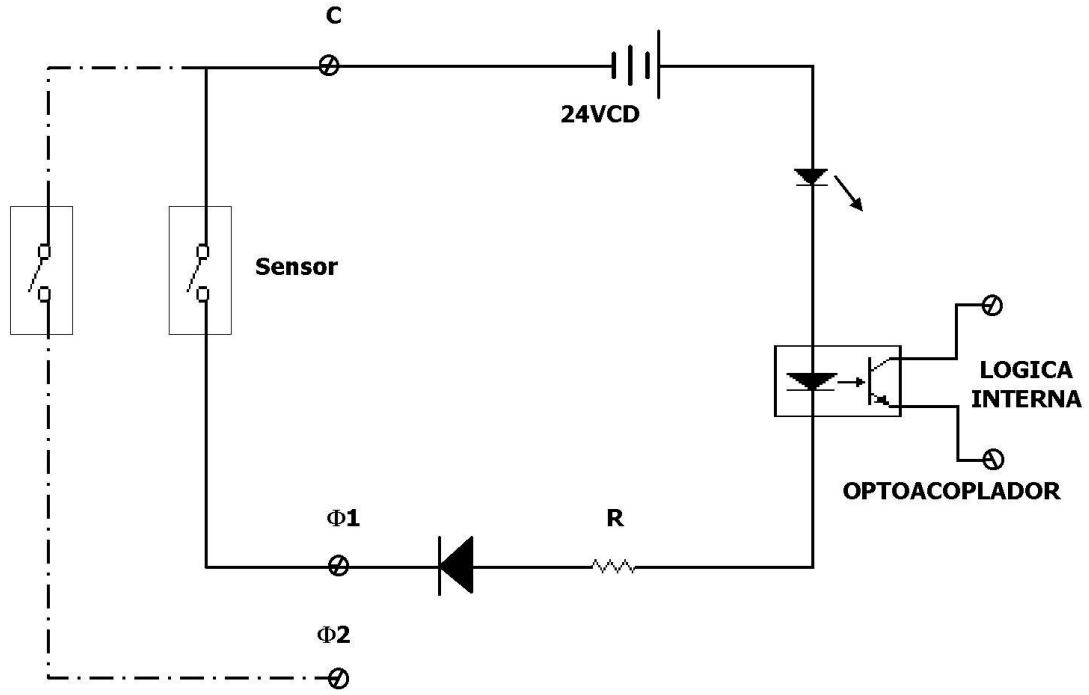


Figura 8. Entrada digital tipo "Sink"

B)

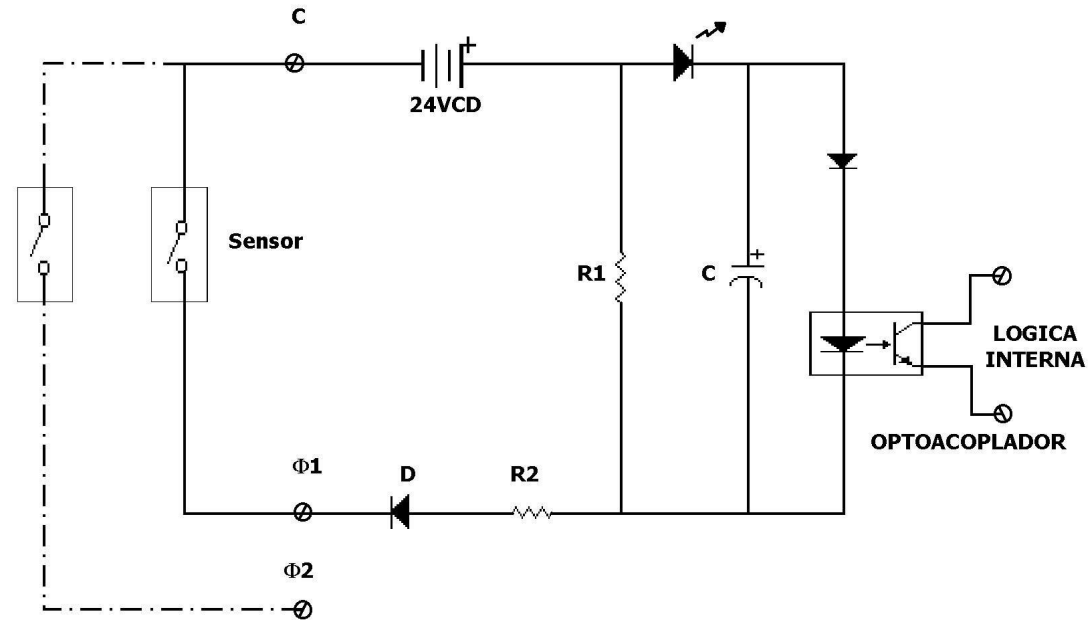


Figura 9. Entrada digital tipo "Sink"

Las entradas análogas pueden poseer cualquier valor dentro de un rango determinado especificado por el fabricante. Se basan en conversores Análogo a Digital (ADC) aislados de la Unidad Central de Procesos (ópticamente).

Las Entradas análogas son de los siguientes tipos:

- Voltaje DC = 0 a 10V, -10 a +10V.
- Corriente DC = 4 a 20 mA, 0 a 20 mA.

Se emplean en la implementación de controladores análogos, los cuales vienen normalmente ya configurados en el PLC.

Igual que con las entradas vamos a disponer de dos tipos de salidas: Digital y Análoga. El modulo de salidas del Controlador Lógico Programable, es el encargado de activar y desactivar los actuadores (bobinas de contactores, lámparas, motores pequeños, etc.).La información enviada por las entradas a la Unidad Central de Procesos, una vez procesada, se envía al módulo de salidas para que estas sean activadas y a la vez los actuadores que en ellas están conectados. Son de muy diverso tipo dependiendo de la señal eléctrica que entregan.

Las salidas digitales al igual que las entradas digitales se basan en el principio de todo o nada, es decir o no conducen señal alguna o poseen un nivel de tensión. Estas salidas se manejan a nivel de bit dentro del programa de usuario. Las salidas digitales generalmente son de los siguientes tipos:

- Corriente Alterna (CA) de 120/220 V (un Triac)
- Corriente Directa (CD) de 12/24V (un Transistor)

- Contacto sin tensión (un Relé)

Salida Digita Tipo "Source"

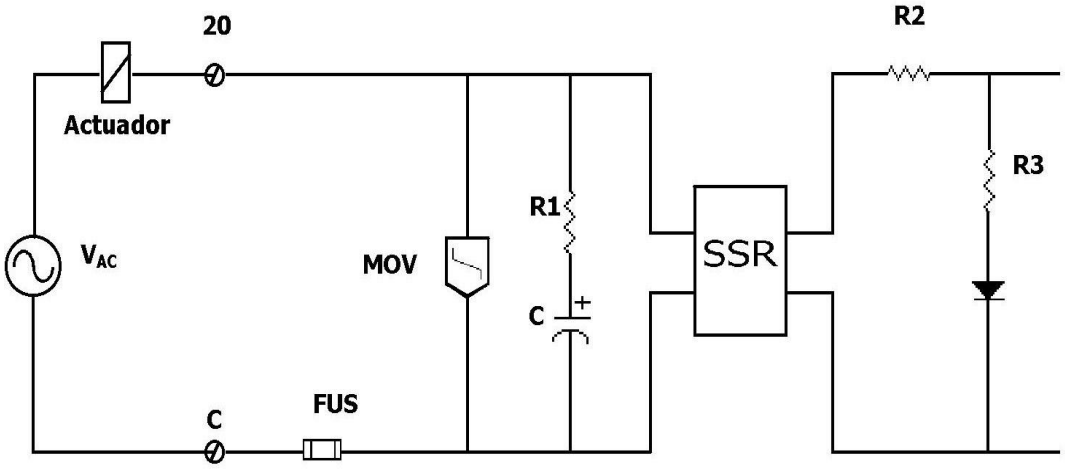


Figura 10. Salida digital tipo "Source"

Salida Digital Tipo "Sink"

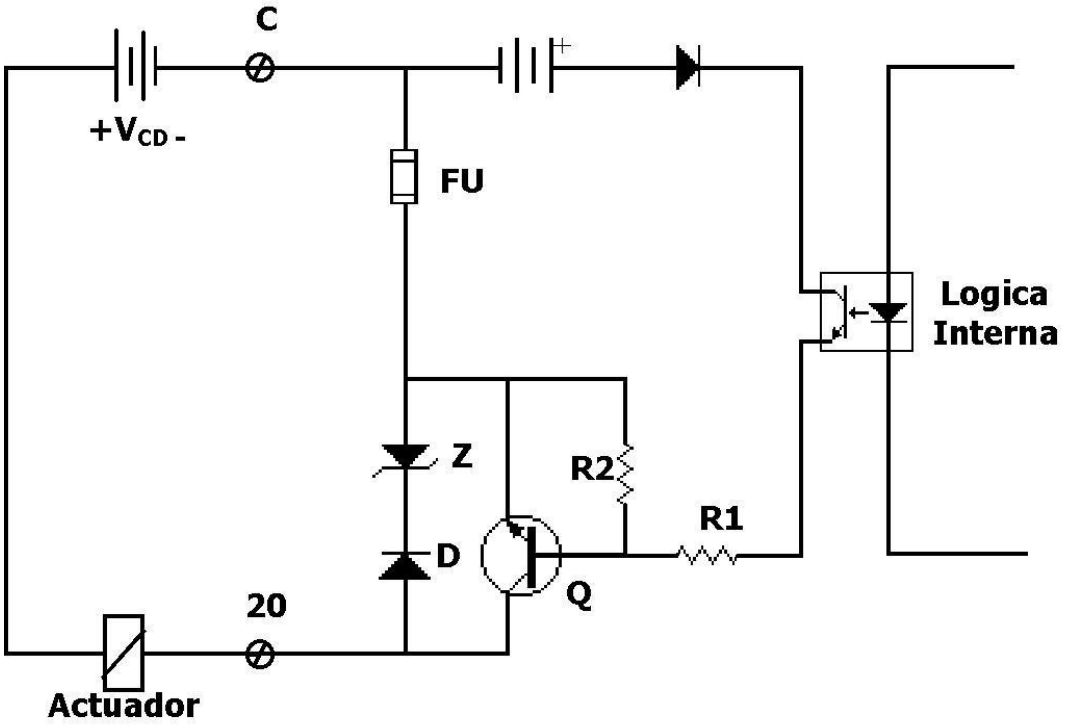


Figura 11. Salida digital tipo "Sink"

Salida Digital por rele tipo "Source"

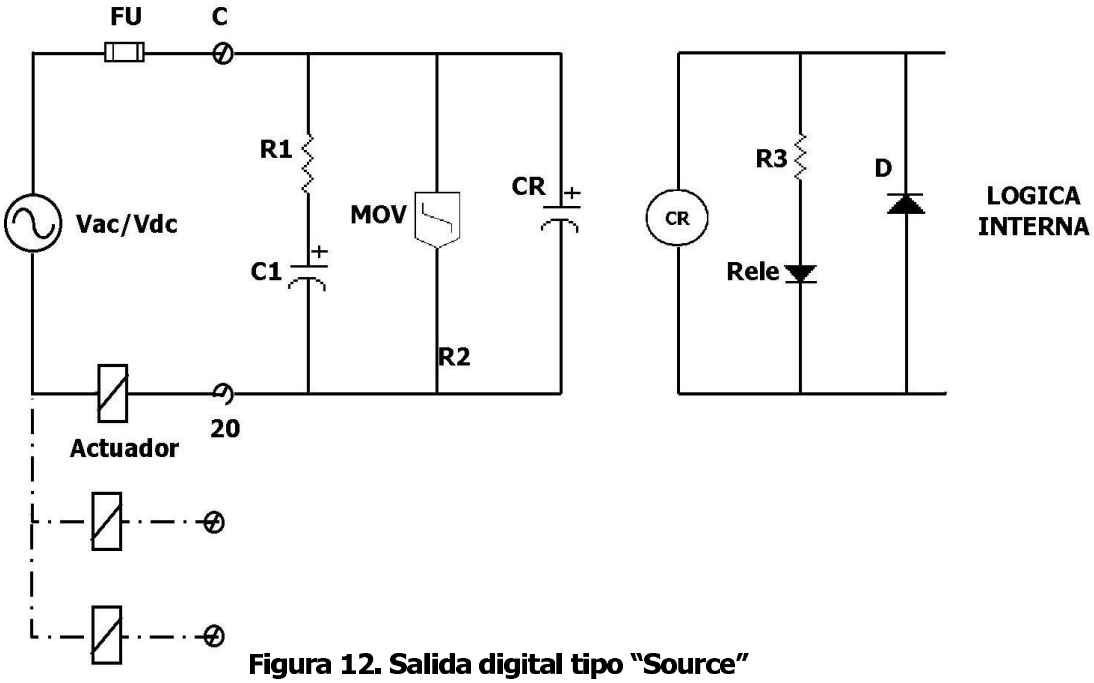


Figura 12. Salida digital tipo "Source"

Las salidas análogas pueden poseer cualquier valor dentro de un rango determinado especificado por el fabricante. Se basan en Conversores de Digital a Análogo (DAC) aislados de la Unidad Central de Procesos (por etapa de potencia). Las salidas análogas generalmente son de los siguientes tipos:

- Voltaje DC = 0 a 10V,-10 a +10V.
- Corriente DC = 4 a 20 mA, 0 a 20 mA.

2.5 INTERFACES

2.5.1 Interfaces de Comunicación

Las comunicaciones con RS-232 son el método más popular para las comunicaciones con dispositivos externos del Controlador Lógico

Programable. El RS-232 especifica los detalles de la conexión física y los detalles eléctricos. Debido a que está diseñada para usarse con dispositivos como módems y terminales especifica la transmisión de caracteres. Cada carácter consta de 7 bits de datos. El RS-232 define una comunicación serial asíncrona, es decir que los bits viajan por el alambre uno detrás otro. Permite al transmisor el envío de un carácter en cualquier momento y un retardo arbitrariamente antes del envío de otro. Cuando se transmite un carácter, el transmisor y el receptor no coordinan antes de la transmisión. Sin embargo una vez que comienza un carácter el hardware transmisor envía todos los bits en secuencia, sin retardo.

El hardware RS-232 del receptor no puede usar la falta de voltaje para marcar el fin de un bit y el comienzo del siguiente. Tanto el transmisor como el receptor deben acordar el tiempo que se mantendrá el voltaje para cada bit El bit extra se conoce como bit de inicio. La norma RS-232 especifica que el transmisor debe dejar inactiva la línea durante un tiempo mínimo, el requerido para transmitir un bit, se puede pensar En la terminología de RS-232, un voltaje negativo pone en el estado de MARCA, y el voltaje positivo pone al alambre en estado de ESPACIO.

El RS-232 utiliza un cable de par trenzado que contiene dos alambres, en donde uno de los alambres conduce la señal y el otro es a tierra que da la trayectoria de regreso, ya que los datos deben fluir simultáneamente en dos direcciones.

La transferencia simultánea en dos direcciones se llama transmisión dúplex integral, y esta requiere un alambre para los datos que viajan en una dirección y otro para la dirección opuesta y un alambre de tierra para completar el circuito en ambas direcciones.

El RS-232 puede configurarse para ignorar los alambres de control y suponer que el otro lado está trabajando. Las conexiones que ignoran las señales de control se llaman circuitos de tres alambres, ya que necesitan los tres para conducir datos[5].

El módem transmite en el pin 2 y recibe en el pin 3, el computador transmite en el pin 3 y recibe en el pin 2 (el alambre de tierra usa el pin7). El cable para conectar una computadora a un módem tiene un alambre del pin 2 al pin 2 y otro del pin 3 al pin 3. Por lo tanto el cable usado para conectar dos computadores debe tener un alambre del pin 2 al pin 3 y otro del 3 al pin 2. El intercambio se llama por lo común intercambio 2-3.

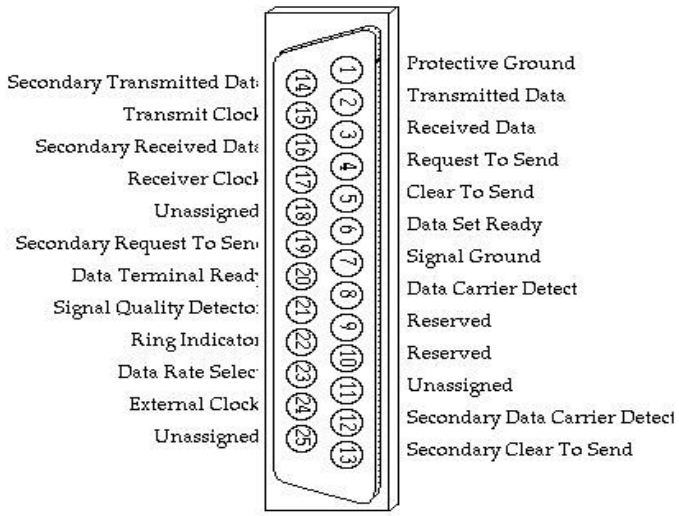


Figura 13. Conector DB25

El estándar especifica 25 pines de señal, y que el conector de DTE (*Equipo Terminal De Datos),* debe ser macho y el conector de DCE (*Equipo De Comunicación De Datos*) hembra. Los conectores mas usados son el DB-25 macho, pero muchos de los 25 pines no son necesarios. Por esta razón en muchos PC modernos se utilizan los DB-9 macho.

[5] Curso de LABVIEW, Minor de Automatización, CUTB, 2003

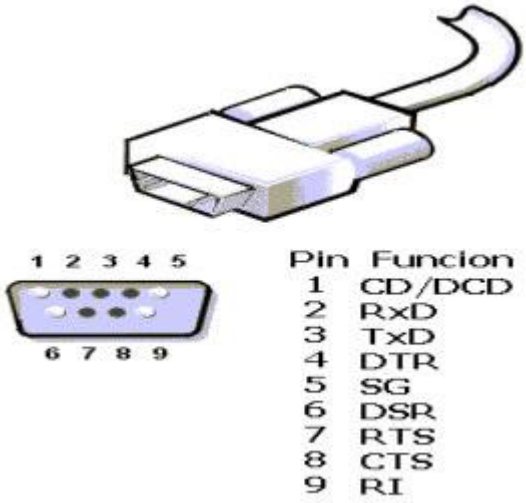


Figura 14. Conector DB9

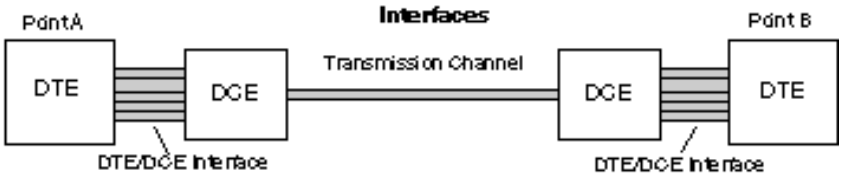


Figura 15. Transmisión con RS-232

Se encuentran algunos de estos conectores en el panel trasero del PC. Los voltajes para un nivel Lógico alto están entre -3V y -15V. Un nivel Lógico bajo tendrá un voltaje entre +3V y +15V. Los voltajes más usados son +12V y -12V. Las señales más utilizadas se listan a continuación:

- DTR (Data-Terminal-Ready): El PC indica al módem que esta encendido y listo para enviar datos.
- DSR (Data-Set-Ready): El módem indica al PC que esta encendido y listo para transmitir o recibir datos.
- RTS (Request-To-Send): El PC pone esta señal a 1 cuando tiene un carácter listo para ser enviado.
- CD (Carrier-Detect): El módem pone esta señal a 1 cuando ha detectado el ordenador.

- CTS (Clear-To-Send): El módem esta preparado para transmitir datos. El ordenador empezara a enviar datos al módem.
- TxD: El módem recibe datos desde el PC.
- RxD: El módem transmite datos al PC.

Numero	de Pin	Señal	Descripción	E/S
En DB-25	En DB-9			
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

Figura 16. Pines del Conector

2.5.2 Interfase MPI

Además de la interfase RS-232 muchos PLC'S utiliza otro tipo de interfases como por ejemplos fabricantes como SIEMENS que utiliza comúnmente la MPI(*Message Passing Interface*). Esta es una Interfase estandarizada para la realización de aplicaciones paralelas basadas en paso de mensajes. El modelo de programación que subyace tras MPI es MIMD (*Multiple Instruction streams, Multiple Data streams*) aunque se dan especiales facilidades para la utilización del modelo SPMD (*Single Program Multiple Data*).

MPI es como su nombre indica, una interface, lo que quiere decir que el estándar no exige una determinada implementación del mismo.

Todas las CPU´s (312, 313, 314, 315 y 315 -2DP) lo incorporan desde fábrica. Con éste puerto se puede comunicar fácilmente a distancias reducidas sin requerir módulos adicionales, por ejemplo hacia equipos de M+V (manejo + visualización), unidades de programación y otros autómatas S7-300 o S7- 400 para probar programas o consultar valores de estado.

Se pueden enviar datos a 4 distintos aparatos al mismo tiempo y utilizando siempre el mismo puerto a una velocidad de 187,5 Kbits / seg o 187,5 Kbaudios. Para pequeñas redes de comunicación o pequeños volúmenes de datos la CPU ofrece el servicio de Datos Globales, que permite intercambiar cíclicamente cantidades de datos en paquetes de hasta 22 bytes como máximo.

Distancia máxima entre dos estaciones o nudos de red de MPI adyacentes: 50 metros (sin repetidores); 1100 metros (con dos repetidores); 9100 metros (con más de 10 repetidores en serie); por encima de los 500 Klm. (cable de fibra óptica, con módulos de conexión ópticas)

Capacidad de expansión: los componentes comprobadores de campo son usados para configurar la comunicación de interfase multipunto: cables LAN, conectores LAN y repetidores RS-485, desde el PROFIBUS y la línea de productos de entradas/salidas distribuidas. Estos componentes permiten una óptima utilización de la configuración.

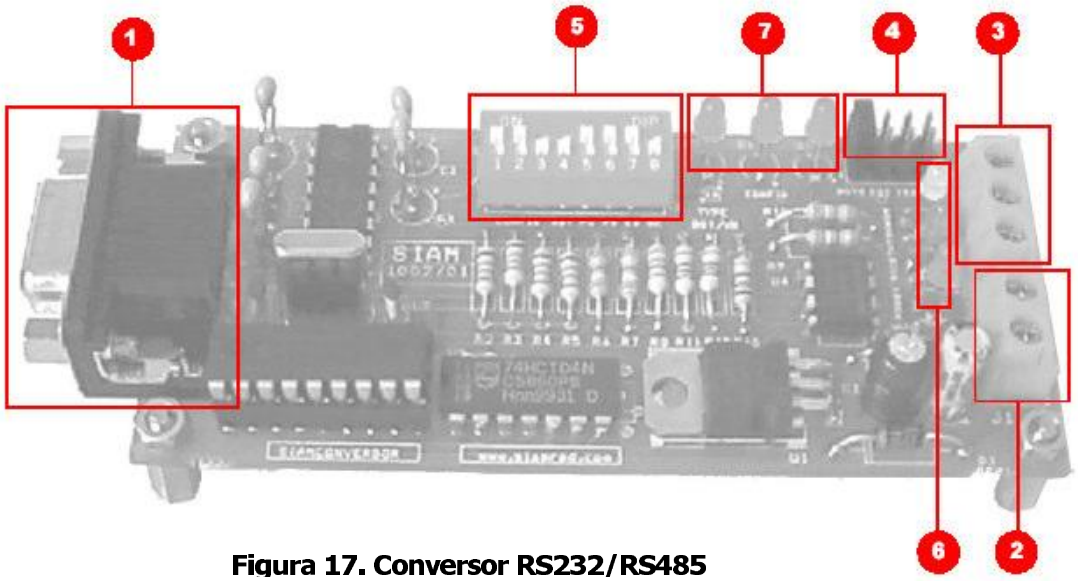


Figura 17. Conversor RS232/RS485

1. Conector DB9 para RS-232
2. Conectores de Alimentación
3. Conectores para bus RS-485
4. Jumpers de configuración del modo de funcionamiento
5. Interruptores de Configuración
6. Leds de monitorización de la comunicación
7. Leds de monitorización de la velocidad

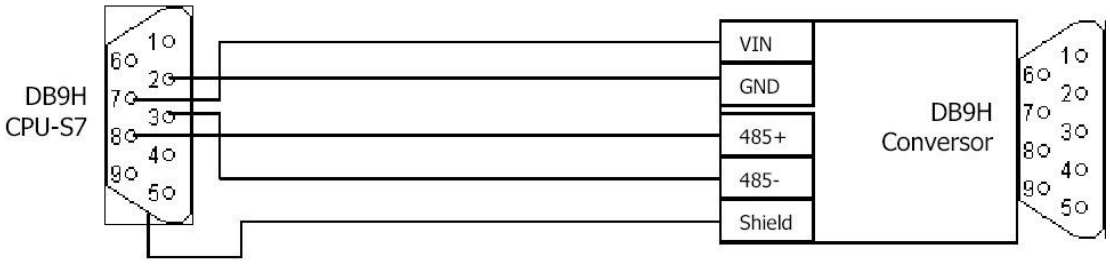


Figura 18. Conexión con PLC Siemens S7-200

2.5.3 Interfaces de Operador

Las más comunes para el control y de las plantas y los Controladores Lógicos Programables, son los siguientes.

- Paneles de operador básicos con pulsadores, interruptores, indicadores luminosos.
- Paneles de visualización de texto.
- Paneles de operador con texto y teclados.
- Paneles de operador gráficos.
- Paneles de operador gráficos de pantalla de toque, puertos, teclados, etc.
- Interfaces basadas en PC (Software de supervisión).

Panel de Operador:

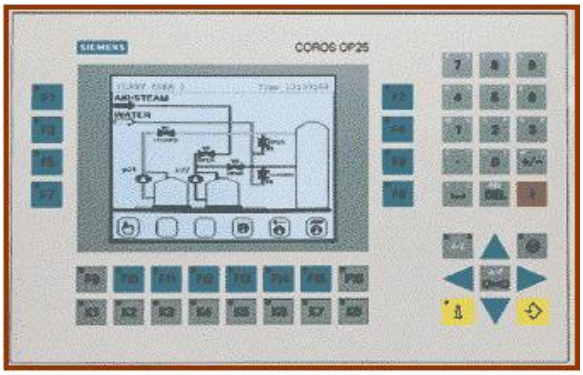


Figura 19. Esquema de panel de control

Los OP (Operator Panel) son paneles de operación que facilitan el acceso visual del operario al sistema de automatización, englobándose dentro del famoso "Human Machine Interface" (Interfaz Humano con la Maquinaria). Algunas posibilidades de los OP son:

- Acceso rápido y sencillo a los datos del sistema.

- Supervisión y control del proceso.
- Visualización del proceso (sólo en OP gráficos).
- Modificación de parámetros y órdenes (sólo en algunos casos).

2.6 PERIFÉRICOS

Como elementos auxiliares y físicamente independientes del autómata, los equipos periféricos realizan funciones concretas de gran importancia. El incremento que experimentan las prestaciones del autómata hace que el numero de periféricos aumenten día a día para equipos de la misma gama. Los equipos que se utilizan son:

- Impresoras.
- Unidades de cinta o memoria.
- Monitores.
- Lectores de código de barras.
- Displays y teclados.
- Unidades de teclado y tests.
- Servidores.

2.7 TAMAÑO DE LOS AUTÓMATAS

La clasificación de los Controladores Lógicos Programables, se realizan en función del numero de entradas/salidas, son admitidos los tres grupos siguientes:

- Gama baja. Hasta un máximo de 128 entradas / salidas
- Gama Media. De 128 a 512 entradas / salidas.
- Gama Alta. Mas de 512 entradas / salidas.

Capitulo 3. CONCEPTOS GENERALES DE PROGRAMACIÓN.

Para realizar una correcta programación se debe tener en cuenta diferentes pasos que son de gran importancia durante el desarrollo.

- Especificación de Control: Determinar que debe hacer el sistema de Control y en que orden.
- Identificar las señales de entrada / salida de la planta a controlar
- Representar mediante un modelo el sistema de control, indicando todas las funciones que intervienen, las relaciones entre ellas, y la secuencia que deben seguir:
    - Modelo Algebraico: Boole
    - Modelo Gráfico: Grafcet, máquinas de estados.
- Asignar direcciones de entrada / salida o internas a cada uno de los componentes que aparecen en el modelo.
- Codificar la representación anterior según la herramienta de programación del Controlador Lógico Programable.
- Cargar programa en el Controlador Lógico Programable y verificar funcionamiento.
- Puede haber una fase de simulación sin la planta real.

3.1 EQUIPOS Y UNIDADES DE PROGRAMACIÓN

La unidad de programación es el medio material del que se auxilia el programador para grabar o introducir en la memoria de usuario las instrucciones del programa. Pero esta unidad realiza otras tareas fundamentales.

3.1.1 Funciones principales

La gama de funciones que son capaces de ejecutar los equipos de programación son múltiples y variadas, aumentando el tipo de estas en razón directa a la complejidad del equipo.

Programación:

- Introducción de instrucciones
- Búsqueda de instrucciones
- Modificación del programa
- Detención de errores de sintaxis o formato
- Visualización del programa del usuario o parte del mismo, contenido en la memoria de usuario.
- Forzamiento del estado de marcas, registros, contadores.

Grabación de Programas:

- En cinta casete.
- En chips de memoria EPROM o EEPROM
- En papel mediante impresora
- En disquete mediante PC.

Visualización y Verificación Dinámica del Programa:

- Del programa o parte de el.
- De entradas y Salidas
- De temporizadores, contadores, etc.

Modos de Servicios:

- Stop
- Run

- Otros modos intermedios como monitor.

3.1.2 Tipos de Unidades de Programación

Desde el punto constructivo podemos distinguir tres tipos principales:

Unidades Tipos Calculadores: Constan de un correspondiente teclado, conmutador de modos, display de siete segmentos o cristal liquido, así como de las entradas para la grabación del programa del usuario. Puede ser totalmente independiente, ser enchufada directamente en la Unidad Central de Procesos, o con ambas posibilidades[6].

Consola de Programación: Es una función intermedia entre el tipo Calculadora y el PC. Consta de pantalla de plasma o tipo similar y tamaño suficiente para 20-30 líneas y 60 a 80 caracteres por líneas, así como teclado. Utiliza el software de programación preciso para los lenguajes utilizados en el Controlador Lógico Programable[7].

Unidad con PC: Unidad que se adapta al autómata mediante la interfase correspondiente. Lleva incorporado un monitor y realiza la misma función que la unidad de programación normal, pero con mayores prestaciones permitiendo visualizar los esquemas o diagramas completos. Este equipo incorpora el software necesario para poder trabajar en mas de un lenguaje de programación.

---

[6] MANDADO PEREZ, Enrique. Controladores Lógicos y Autómatas Programables. Alfaomega, 2ª ed. México, 1999.

[7] MANDADO PEREZ, Enrique. Controladores Lógicos y Autómatas Programables. Alfaomega, 2ª ed. México, 1999.

3.1.3 Funcionamiento

Las instrucciones que se introducen en la unidad de programación no son directamente interpretables por el procesador, que se ha de auxiliar de un circuito intermedio llamado *Compiler.* Es por tanto, el Compiler el elemento de unión entre el autómata y la unidad de programación. Su misión es la de traducir la información textual de la unidad de programación a lenguaje maquina y viceversa mediante unos códigos intermedios que son interpretados por un programa residente en firmware.

3.2 CICLO DE TRABAJO DE UN AUTÓMATA

Figura 20. Ciclo de Trabajo de un Autómata

La ejecución de un Autómata Programable se produce en forma de ciclos repetitivos del mismo conjunto de instrucciones almacenadas en memoria, cada vez que se ejecutan las instrucciones se definen que se ha ejecutado un ciclo de programa. Bajo la denominación ciclo de funcionamiento, se refieren a conceptos acerca de cómo, cuando, y en que frecuencia dentro de un mismo ciclo, se realizan las adquisiciones de entrada y se proceden al envío de las salidas, cuando se realiza la evaluación de instrucciones y

ecuaciones lógicas en el tramo comprendido entre la primera y la ultima instrucción de programas. □

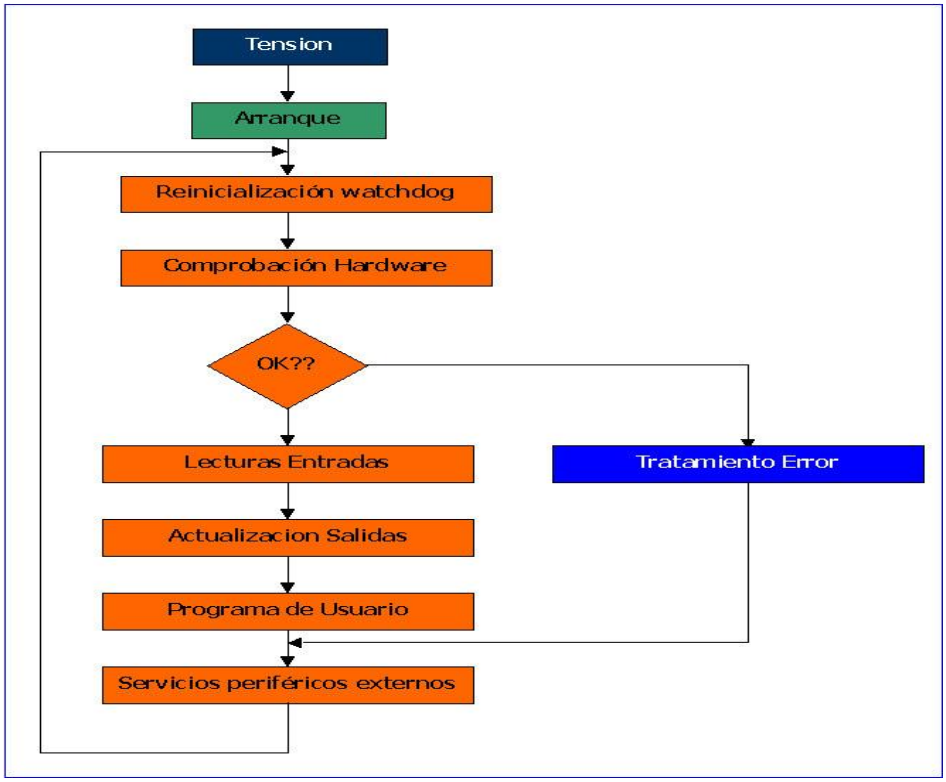


Figura 21. Ciclo de Funcionamiento de un Autómata

3.3 ESTÁNDAR IEC 1131-3

Debido a que los Controladores Lógicos Programables, se utilizan hoy en día en sistemas de control mucho más complejos que los diseñados con relés, se hizo necesaria la aparición de lenguajes de programación mucho más amigables y potentes. Esto dio lugar a que cada fabricante comenzara a utilizar nuevos tipos de lenguajes de programación completamente diferentes a los de sus competidores. Por esto la programación comenzó a ser mas complicada, al tener que aprender lenguajes muy diferentes entre sí cada vez

que se cambiaba de marca o familia de Controladores Lógicos Programables[8].

A raíz de esto se presenta un estándar para tratar de resolver este inconveniente, este estándar es el IEC 1131, el cual aun no es utilizado completamente por los fabricantes. La creciente complejidad en la programación de los Controladores Lógicos Programables, requiere más que nunca de la estandarización de la misma. Bajo la dirección del IEC (Comisión Electrotécnica Internacional) el estándar IEC 1131-3 para la programación de Controladores Lógicos Programables 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. Con la idea de hacer el estándar adecuado para un gran abanico 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.

[8] http://www.plcs.net/01What is a PLC.htm

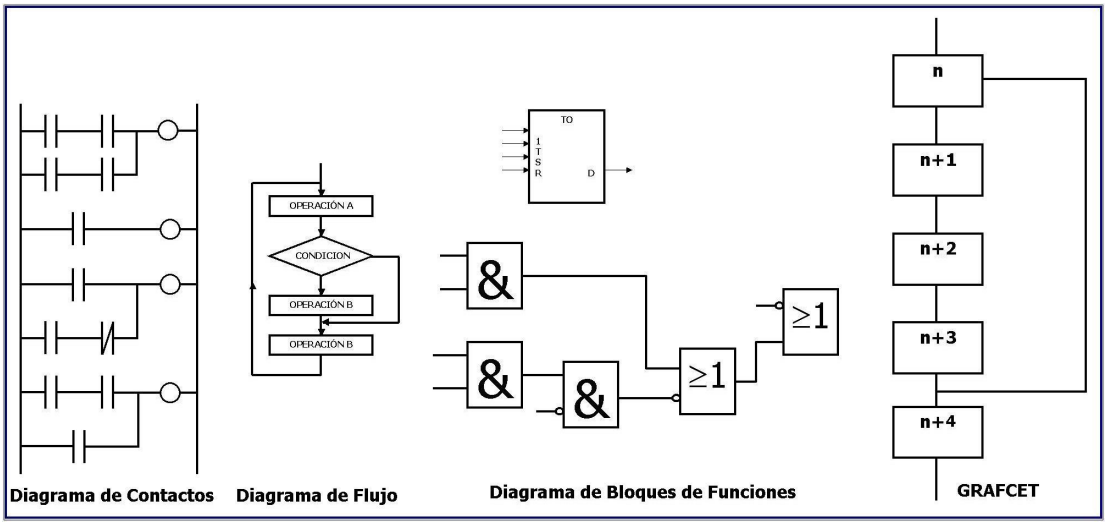


Figura 22. Tipos de Lenguajes de Programación.

Ventajas de usar IEC 1131-3

- Estándar internacional aceptado.
  - Paso a paso todos los proveedores lo tendrán en el futuro.
  - Estructuras uniformes, idiomas y forma de manejar todos los proveedores.
- Ahorra su tiempo
  - El único con el concepto de modelo software.
  - Usted sólo tiene que aprender un lenguaje para diferentes tipos de Controladores Lógicos Programables.
  - Reduce las equivocaciones y errores.
  - Funciones estándar.

- Seguridad de apoyos y programación de calidad.

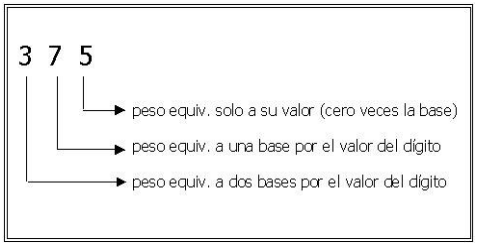
    - Fácil y cómoda estructuración.
    - Indicación de errores durante la programación.
- Provee el mejor lenguaje para cada problema.

    - Especificaciones en 5 lenguajes de programación.
    - Lenguajes textuales y gráficos.
    - Disponibilidad de alto nivel de lenguaje.
    - Posibilidad de mezclar los diferentes lenguajes.

Figura 23 Estandarización Lenguajes de Programación

3.4 SISTEMA DE NUMERACIÓN Y MANIPULACIÓN DE DATOS

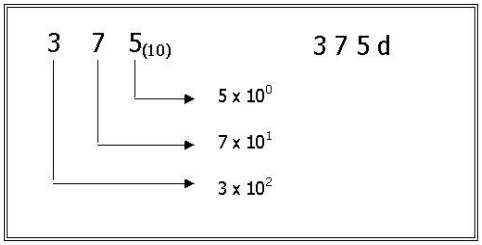
En todo sistema base de numeración, el valor de un numero se basa en el de cada dígito individual y el peso de su posición dentro del numero.



Cada dígito tiene su propio valor propio, por ejemplo el 3 indica que su valor son tres unidades, que es una unidad mas que dos, la cual a su vez es una unidad mas que uno, y este a su vez una unidad mas que nada (cero, 0).

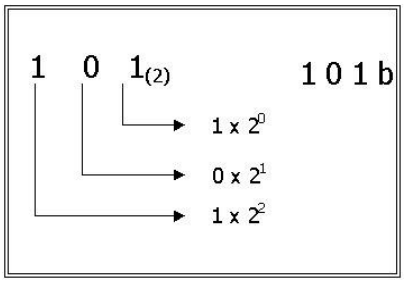
Sistema decimal

Es el sistema de numeración normal que utilizamos en nuestra vida diaria. Tiene base 10 (cada posición pesa diez veces mas que la anterior), y por tanto maneja 10 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).



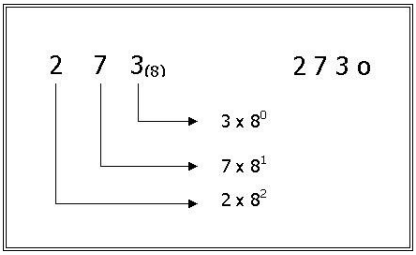
Sistema Binario

Es el sistema de numeración en el cual trabajan los sistemas digitales en su interior. Tiene base 2 (cada posición pesa dos veces mas que la anterior), y por tanto maneja solo 2 dígitos (0, 1).



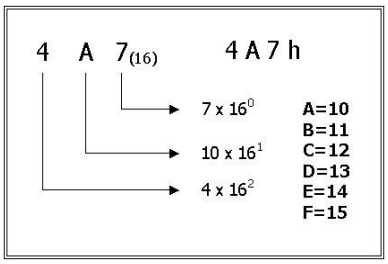
Sistema Octal

Es un sistema de numeración que se utiliza para representar números binarios debido a su facilidad de conversión desde y a binario. Tiene base 8 (cada posición pesa ocho veces mas que la anterior), y por tanto maneja 8 dígitos (0, 1, 2, 3, 4, 5, 6, 7).



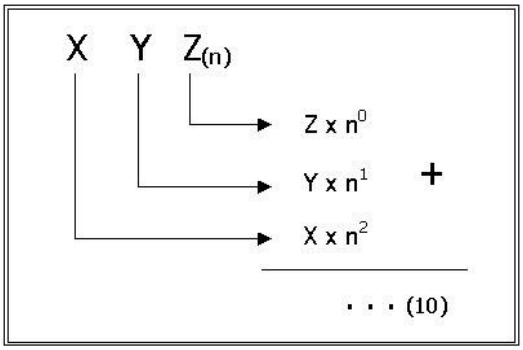
Sistema Hexadecimal

Es un sistema de numeración que se utiliza para representar números binarios debido a su facilidad de conversión desde y a binario. Tiene base 16 (cada posición pesa dieciséis veces mas que la anterior), y por tanto maneja 16 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).



Conversión de Base n a Base 10 (Decimal)

Se multiplica cada digito por el peso de su posición (base elevada a su posición, comenzando a contar desde cero), posteriormente se suma todo y de esta forma se obtiene el numero equivalente en base 10.



Conversión de Base 10 (Decimal) a Base n

Se divide el numero entre la nueva base en forma sucesiva hasta obtener un cociente menor a la base. Posteriormente se forma el numero en la nueva base tomando el ultimo cociente, posteriormente el ultimo residuo y todos los residuos de ahí en adelante hasta el primer residuo para formar el numero de derecha a izquierda.

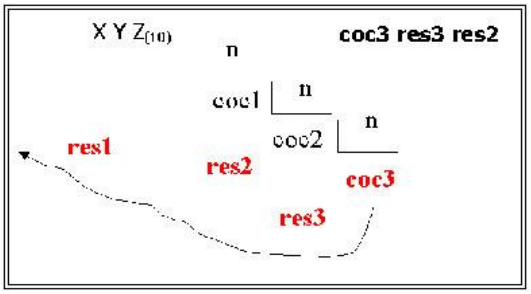


Tabla de conversión de sistemas numéricos

Decimal	Binary	BCD	Hexadecimal
0	0	0000	0
1	1	0001	1
2	1	0010	2
3	11	0011	3
4	100	0100	4
5	101	0101	5
6	110	0110	6
7	111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	0001 0000	A
11	1011	0001 0001	B
12	1100	0001 0010	C
13	1101	0001 0011	D
14	1110	0001 0100	E
15	1111	0001 0101	F
16	10000	0001 0110	10
17	10001	0001 0111	11
18	10010	0001 1000	12
19	10011	0001 1001	13
20	10100	0010 0000	14
.	.	.	.
.	.	.	.
126	111 1110	0001 0010 0110	7E
127	111 1111	0001 0010 0111	7F
128	1000 0000	0001 0010 1000	80
.	.	.	.
.	.	.	.
510	1 1111 1110	0101 0001 0000	1FE
511	1 1111 1111	0101 0001 0001	1FF
512	10 0000 0000	0101 0001 0010	200

3.5 MARCAS DE MEMORIA

La memoria esta organizada como una matriz, cada fila tiene un numero asociado, es decir, la dirección que se corresponde con la secuencia de la fila .Cada columna esta dividida en bits. El tamaño típico son 8 bits (1 Byte)[9].

Existen varias formas para acceder a la información en la memoria como lo son:

- Bit (M #C.#F)
- Byte (M B #F)
- Word (M W Fi)
- Long Word (M D Fi)

También son denominadas como variables de memoria. Son de propósito general, es decir, podremos emplearlas en lo que deseemos. Se distinguen dos tipos de marcas de memoria:

- Remanentes: Estas marcas permanecerán en memoria aunque apaguemos el autómata. En total hay 64 bytes de memoria para estas marcas, por lo que tendremos 512 marcas remanentes de 1 bit cada una: M0.0 ... M63.7.

- No remanentes: Estas marcas se borrarán en cuanto apaguemos el autómata. También tenemos 64 bytes destinados a estas marcas, por lo que tendremos 512 marcas no remanentes de 1 bit cada una: M64.0 ... M127.7.

[9] Curso de PLC, Minor de Automatización industrial, CUTB, 2003

Hay que destacar que las marcas se ponen a cero cada vez que reseteamos el autómata. Esta característica nos puede ser de mucha utilidad en algunos casos.

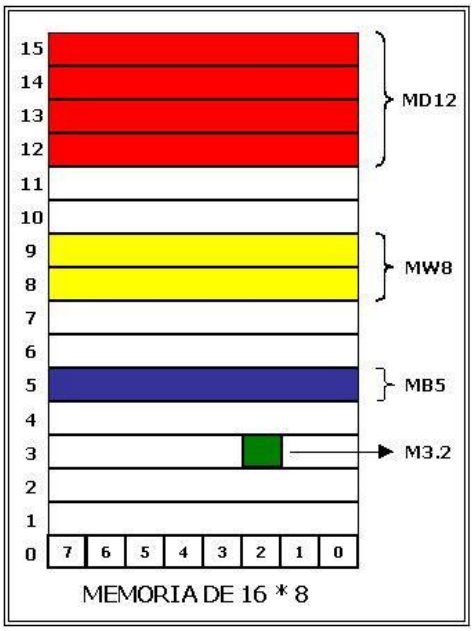


Figura 24. Estructura interna de la memoria

3.6 ENTRADAS Y SALIDAS

Salvo excepciones y ampliaciones, los autómatas presentan 8 entradas (E) normales de 1bit: E32.0 ... E32.7 y 2 entradas especiales de 1bit: E33.0 y E33.1. Estas últimas tienen la peculiaridad de funcionar como entradas digitales o como entrada de alarma (E33.0) y entrada rápida (E33.1).

Hay 6 salidas (A), de 1bit cada una: A32.0 ... A32.5

3.7 PROGRAMACIÓN

La programación en un autómata consiste en el establecimiento de una sucesión ordenada de instrucciones (disponibles en el equipo) que resuelven

una determinada tarea de control. La secuencia, que establece la relación entre las distintas variables lógicas, es lo que constituye el programa del autómata.

Es posible programar un autómata mediante el establecimiento directo de una secuencia de instrucciones en lenguaje maquina, pero es indudable que este lenguaje esta bastante alejado del utilizado por el técnico especializado para especificar el funcionamiento de un sistema de control. Por ello los fabricantes utilizan distintos lenguajes de programación que serán mencionados en los siguientes apartados[10].

Lenguajes de Programación: En una primera clasificación los lenguajes pueden clasificarse en:

- INFORMATICOS:
  - Bajo Nivel: Lista de Instrucciones
  - Alto Nivel
- GRAFICOS:
  - Diagramas de escalera (Ladder Diagrams).
  - Flujo gramas.
  - Diagramas de funciones lógicas.
  - Grafcet.

De forma casi omnipresente se dispone del lenguaje de contactos o diagramas en escalera, cuya introducción formo parte de las especificaciones de diseño originales con objeto de que el personal de mantenimiento de los antiguos armarios de reles pudiesen operar con estos sin excesiva dificultad.

[10] SIMONS, Andrés. Autómatas Programables. Paraninfo, Madrid, 1988.

Este lenguaje es una simulación grafica de los circuitos eléctricos basados en elementos eléctricos normales abiertos, cerrados y bobinas de accionamientos.

Otra forma de lenguaje de programación, tales como los diagramas de funciones lógicas están influenciadas y basadas por las técnicas de electrónica digital. En estos podemos encontrar bloques programables de puerta lógicas, temporizadores, contadores, biestables, comparadores etc., disponibles en librerías de objetos prediseñados. Otro tipo de lenguaje son de carácter eminentemente informaticos estos están constituidos por lenguajes de instrucciones, basados en nemotécnicos, Basic, C, flujo gramas etc. En la actualidad se están incorporando lenguajes derivados de grafos, basados en estados y eventos, tales como, Grafcet.

La programación de los autómatas programables puede llevarse acabo de forma off-line y on-line. La primera modalidad suele utilizarse en la fase de diseño e implementación del algoritmo de control. En la mayoría de los casos, se ha generalizado el uso de los computadores personales como soporte del entorno software de programación suministrado por los fabricantes.

La segunda modalidad para la programación en planta, cuando por aplicación de las tareas de mantenimiento software se llevan a cabo las modificaciones del programa inicialmente implantadas. Estas modificaciones suelen realizarse incluso sin detener el funcionamiento del API, mediante terminales específicos de programación de pequeños tamaños adaptados al efecto.

3.7.1 Lenguaje de Contactos

La mayoría de los fabricantes incorporan este lenguaje, ello es debido a la semejanza con los esquemas de reles utilizados en los automatismos eléctricos de lógica cableada, lo que facilita la labor de los técnicos para trabajar con dichos automatismo. En este lenguaje, la tarea que debe el autómata se representa gráficamente mediante un esquema de contacto. Para programar en este lenguaje se necesita una unidad de programación que posea un terminal con una pantalla semigráfica que permite visualizar el esquema de contactos[11].

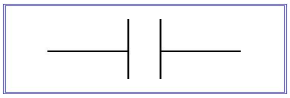


Figura 25. Variable de entrada, salida de contactos normalizados según norma NEMA.

- Identificación de las Variables

Las variables binarias se representan mediante contactos, a cada uno de los cuales se asigna una identificación. Los contactos normalmente abiertos se representan como la figura mostrada anteriormente y representan variables directas. Las variables inversas se representan mediante contactos normalmente cerrados como lo indica la siguiente figura:

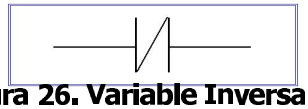


Figura 26. Variable Inversa

[11] MANDADO PEREZ, Enrique. Controladores Lógicos y Autómatas Programables. Alfaomega, 2ª ed. México, 1999.

- Secuencias Lógicas

Las distintas funciones lógicas se pueden representar en este lenguaje a continuación se analizan cada una de ellas.

- Función de una variable de entrada directa:

Esta función se representa mediante un contacto normalmente abierto que, en general, activa una variable de salida.

Figura 27. Variable directa

- Función de Selección de una variable invertida

Esta función se representa mediante un contacto normalmente cerrado que activa una variable de salida.

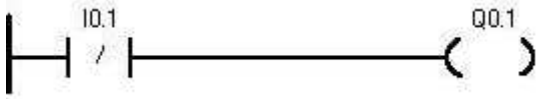


Figura 28 Variable Invertida

- Función OR

Esa función se representa mediante un montaje paralelo de contactos, ya sean abiertos o cerrados.

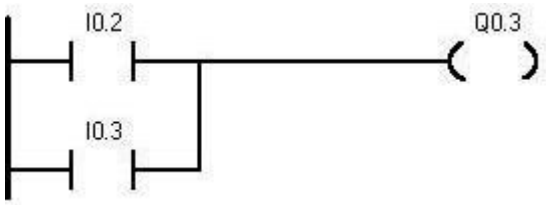


Figura 29 Lógica OR

- Función AND

Esa función se representa mediante un montaje serie de contactos, ya sean abiertos o cerrados.

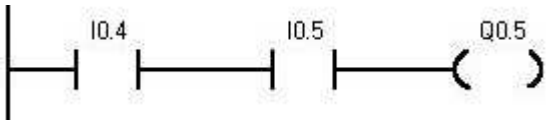


Figura 30. Lógica AND

- Función OR – AND

Esta función se representa mediante la combinación en paralelo de contactos conectado en serie.

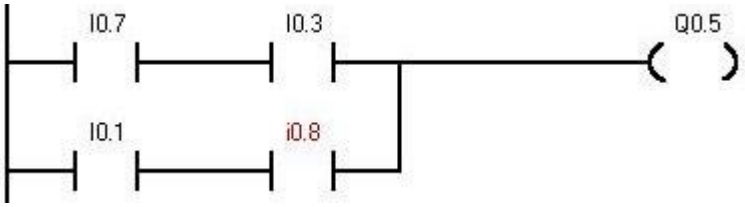


Figura 31. Función OR-AND

- Función AND-OR

Esta función se representa mediante la conexión en serie de contactos conectado en paralelo.

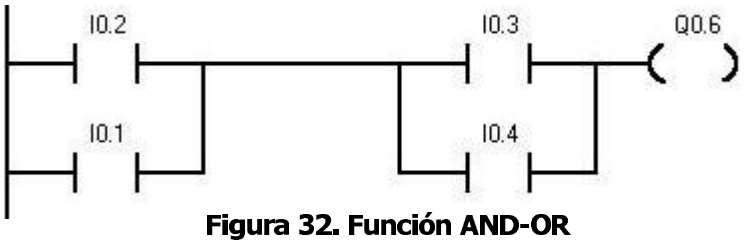


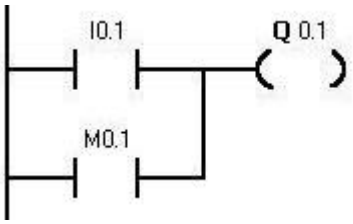
Figura 32. Función AND-OR

Ejemplo: *Con la activación del pulsador conectado a la entrada I0.1, las tres luces de un semáforo deben encenderse una tras otra, una a cada segundo. Al cabo de un segundo del encendido completo, las luces deberán apagarse.*

Para programar la solución a este problema se ha hecho uso de símbolos del S7200 y además fue simulado en el MicroWin 32. Su correspondencia con los operandos absolutos se ha establecido según la tabla siguiente:

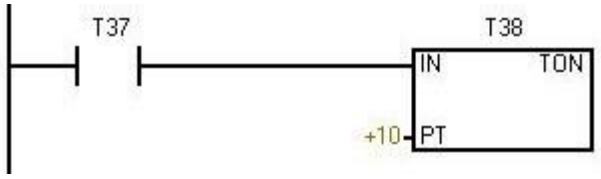
Direccion	Comentario
I0.1	Start.
Q0.1	Luz 1
Q0.2	Luz 2
Q0.3	Luz 3

Comentario 1: Activa I0.1 (start) y enciende la primera luz.



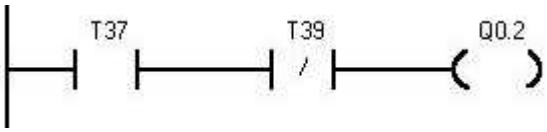
Comentario 2: Se activa el temporizador por 1 seg. (Para la primera luz)

Comentario 3: Se activa el segundo temporizado por 1 seg, para la segunda luz.

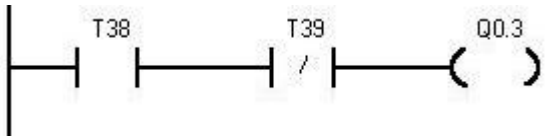


Comentario 4: Se activa el temporizado por 1 seg. Para la tercera luz.

Comentario 5: Se enciende la segunda luz mientras T39 este apagada.



Comentario 6: Se enciende la Tercera luz mientras T39 este apagada.



3.7.2 Diagrama de Funciones

Constituye un lenguaje simbólico en el que las distintas combinaciones entre variables se representan mediante símbolos lógicos normalizados. Este lenguaje de programación esta especialmente indicado para los usuarios familiarizados con la electrónica digital y al igual que el esquema de contactos, se necesita una unidad de programación dotada de pantalla[12].

- Identificación de las Variables

En función del tipo de variable se realiza tal como se indica a continuación:

[12] MANDADO PEREZ, Enrique. Controladores Lógicos y Autómatas Programables. Alfaomega, 2ª ed. México, 1999.

a. Variables de Entrada: El termino X representa una variable binaria de entrada y lleva asociada un numero que corresponde a su situación en el conector de entrada.
b. Variables de Salida: El termino Y representa una variable de salida y lleva asociada un numero que corresponde a su situación en el conector de salida.
c. Variables de Salidas Internas: El termino IR representa una variable binaria interna (elemento de memoria) y lleva asociado un numero el cual corresponde al orden.

- Operaciones Lógicas

- Función de selección de una variable de entrada directa

Esta función se representa mediante el símbolo normalizado de la siguiente figura:

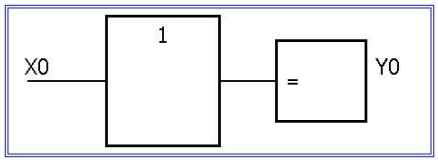


Figura 33. Entrada directa.

- Función de Selección de una variable de entrada Invertida

Esta función se representa mediante el símbolo normalizado de la siguiente figura:

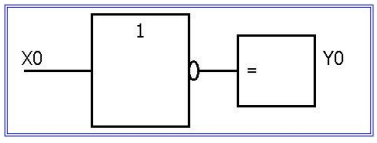


Figura 34. Entrada inversa.

❖ Función OR

Esta función se puede realizar tanto por variables directas, invertidas o combinación de ambas.

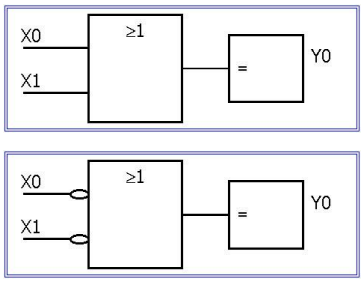


Figura 35. Función OR

❖ Función AND

Esta función se puede realizar tanto con variables directas como invertidas o combinación de ambas:

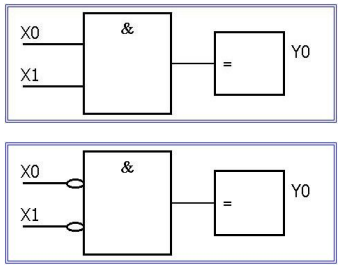


Figura 36. Función AND

❖ Función OR-AND

Esta función se representa mediante la combinación de símbolos de puertas AND y OR como se indica en la figura:

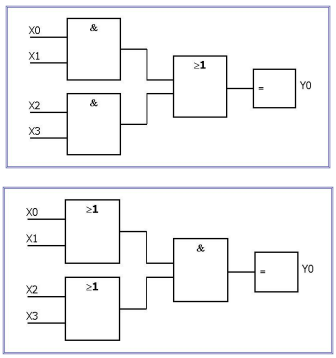


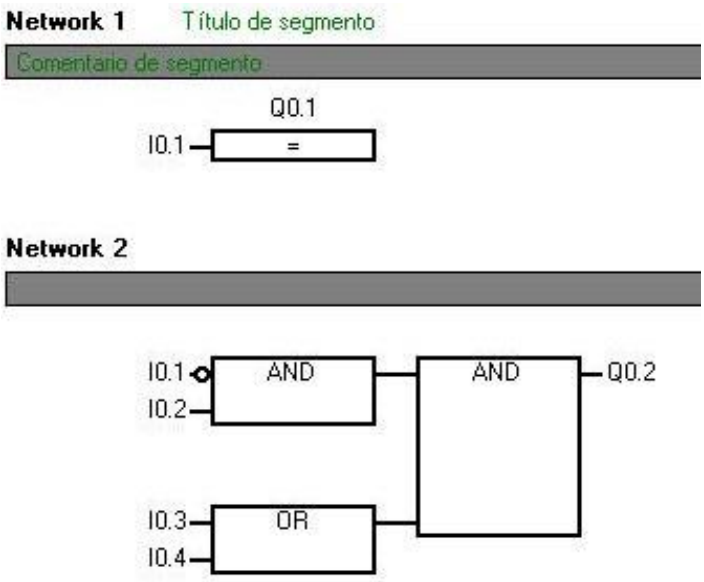
Figura 37. Función OR-AND

Ejemplo:

Poner en marcha un motor, el cual controla el aire central de un almacén, se activara una sirena cuando una entrada de seguridad se active, al ocurrir esto significara que existe alguna anomalía; cada vez que se active la sirena se para el motor. Dos interruptores colocados en lugares diferentes del almacén encienden el motor, además existe un sistema on / off.

Este ejemplo se realizo utilizando el S7200 de Siemens como PLC a usar, simulando con el MicroWin/32.

Direcciones	Comentarios
I0.1	Deteccion de anomalia.
I0.2	Pone en marcha el Sistema ON/OFF
I0.3	Interruptores
I0.4	Interruptores
Q0.1	Sirena
Q0.2	Motor



Si I0.1 es 1, se activa la alarma, si es 0 y cualquier otra entrada es 1 se prende el cero.

3.7.3 Lista de Instrucciones

Este lenguaje consiste en un conjunto de códigos simbólicos, cada uno de los cuales corresponde a una instrucción en lenguaje maquina.

Dado que los lenguajes maquinas de los autómatas son diferentes también lo son los lenguajes de lista de instrucciones. Por ser la programación mediante códigos simbólicos la que mas se aproxima al lenguaje maquina, esta especialmente indicada para usuarios familiarizados con la electrónica digital y la informática. Por otra parte, este lenguaje es el único utilizable con las unidades de programación sencillas, que solamente visualizan una o varias líneas de programación simultáneamente[13].

[13] MANDADO PEREZ, Enrique. Controladores Lógicos y Autómatas Programables. Alfaomega, 2ª ed. México, 1999.

- Identificación de las Variables

Se utilizan la misma nomenclatura utilizada en el apartado anterior.

- Instrucciones

Se supone que el autómata dispone de las siguientes instrucciones, clasificadas en tres grupos.

- Instrucciones de Selección, de entrada, salida o de operación.

Estas instrucciones realizan alguna de las siguientes acciones:

  - Seleccionan una determinada variable para utilizarla como operando o como objeto de una acción de lectura de una entrada o activación/desactivacion de una salida.

  - Realizan una acción de entrada o de salida.

  - Realizan una cierta operación con una variable.

Se supone que el autómata posee dentro de este grupo las siguientes instrucciones, las cuales se encuentran escritas en codigo internacional, estas instrucciones varian un poco dependiendo del fabricante pero en general se emplean de la misma forma:

STR, Se utiliza para seleccionar la primera variable que se va utilizar en una secuencia de instrucciones.

STR NOT, Se utiliza para seleccionar la primera variable invertida que se va utilizar en una secuencia de instrucciones.

OUT, Actúa sobre la variable de salida asociada a ella.

Ejemplo:	STR	X0
	OUT	Y0

Selecciona la variable de entrada X0 y hace la variable de salida externa Y0 igual a la entrada de X0.
OUT NOT, Actúa sobre la variable invertida de salida asociada a ella.

OR, Realiza la función lógica OR entre una variable o combinación de variables y la variable especificada en ella.

Ejemplo:	STR	Y5
	OR	X3

Realiza la función lógica OR entre Y5 y X3

OR NOT, Realiza la función lógica OR entre una variable o combinación de variables y la variable especificada en ella invertida.

Ejemplo:	STR	Y5
	OR NOT	X3

Realiza la función lógica OR NOT entre Y5 y X3.

AND, Realiza la función lógica AND entre una variable o combinación de variables y la variable especificada en ella.

Ejemplo:	STR	X5
	AND	X3

Realiza la función lógica AND entre X5 y X3.

AND NOT, Realiza la función lógica AND entre una variable o combinación de variables y la variable especificada en ella invertida.

Ejemplo:	STR	Y5
	AND NOT	X3

Realiza la función lógica OR NOT entre Y5 y X3.

OR STR, Realiza la función OR entre las dos secuencias anteriores a ella iniciadas por STR o STR NOT.

STR	X7
OR	X9
AND NOT	Y5
STR NOT	IR3
AND	X6
OR NOT	Y6
OR STR	
OUT	Y8

Este realiza las siguientes acciones selecciona la variable de entrada X7, realiza la OR entre X7 y X9, realiza la lógica AND entre Y5 negado y X7+X9, inicia otra secuencia seleccionando la variable IR3 invertida, realiza una AND entre IR3 negado y X6, realiza una OR entre Y6 negado y IR3 negado por

X6, realiza una OR entre las dos expresiones, es decir (X7+X9)Y5 negado e IR3 negado por X6+Y6 negado, hace la variable de salida externa Y8 igual a la expresión anterior[14].

AND STR, Realiza la función AND entre las dos secuencias anteriores a ellas iniciadas por STR o STR NOT.

- Instrucciones de Temporización y Conteo

Como su nombre indica, generan variables cuya activación, duración o desactivación es una función del tiempo o del numero de impulso aplicados a una variable de entrada.

Para que un autómata posea estas instrucciones es necesario que incluya en su sistema físico temporizadores y contadores. Se supone que el autómata posee las siguientes instrucciones de este tipo:

TMR, Realiza la función de Temporización, utiliza para ello dos variables.

Xi = Variable de puesta a cero
Xj = Variable Temporizada

La salida del temporizador se realiza a través de una variable de salida externa o interna. La programación del temporizador necesita cuatro instrucciones en secuencia una instrucción de selección de la variable de puesta, una instrucción de la variable temporizada, la instrucción TMRn que

[14] MANDADO PEREZ, Enrique. Controladores Lógicos y Autómatas Programables. Alfaomega, 2ª ed. México, 1999.

elige el temporizador. TMRn inicia la Temporización si Xi esta en 1 y Xj pasa a ser 1 (Se activa la variable cuyo cambio se temporiza) una disposición de memoria de programa que almacene el valor del tiempo preseleccionado.

Ejemplo:

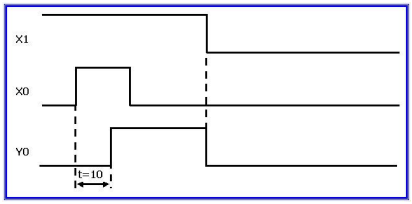


Figura 38. Diagrama de secuencia temporal de un temporizador

STR	X1
STR	X0
TMR	0
	10
OUT	Y0

En la figura se representa el diagrama temporal de las señales XO y X1 de entrada y Y0 de salida. Para que la señal Y0 pase al nivel 1 es necesario que X1 este en 1 y X0 pase al nivel 1 y que transcurran 10 unidades de tiempo.

Las instrucción TMR puede ser utilizada para generar un retardo a la activación de una variable. Para ello se utiliza la misma variable como variable de puesta a cero y temporizada.

Ejemplo:

STR	X5
STR	X5
TMR	2
	10

OUT Y5

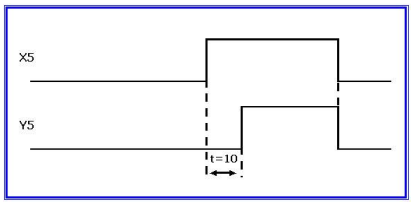


Figura 39. Diagrama de secuencia temporal de un retardo a la conexión

Igualmente TMR puede ser utilizado para generar un retardo a la desactivación de una variable si selecciona invertida como puesta a cero o como variable temporizada.

Ejemplo:

STR NOT	X4
STR NOT	X4
TMR	6
	10
OUT NOT	Y9

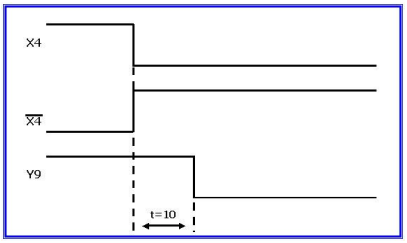


Figura 40. Diagrama de secuencia temporal de un retardo a la desconexión

CTR, Realiza la función de Contaje para ello se puede utilizar dos o tres variables. En el primer caso funciona como contadores ascendentes en el que la primera variable es la de puesta a cero y la segunda es la de Contaje. En el segundo caso funciona como contador reversible, y en el, la primera y tercera variable actúan igual que en el caso anterior y la segunda selecciona

el modo de Contaje ascendente o descendente, según se encuentre en nivel cero o uno respectivamente.

Por lo tanto, la programación de un contador necesita cuatro o cinco instrucciones en secuencia respectivamente que actúan de forma parecida a la instrucción TMR.

Ejemplo:

```
STR   X1
STR   X0
CTR   3
      10
OUT   Y2
```

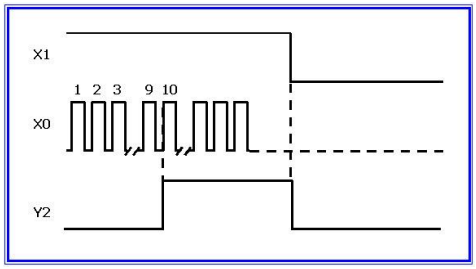


Figura 41. Diagrama de secuencia temporal de un Contador ascendente

- Instrucciones de control

Influyen en la ejecución de las demás instrucciones, mediante tomas de decisión. Aunque no son imprescindibles en un autómata, su existencia facilita la programación.

Admiten múltiples variantes alternativas, de las que el diseñador del autómata elige las que le parecen más interesantes.

Se supone que el autómata posee las siguientes instrucciones asociadas:

JMP-JME: Hacen que las instrucciones del programa situadas entre ellas se ejecuten o no en función del resultado de la operación lógica inmediatamente anterior a JMP. Si el resultado de dicha operación lógica es uno, las citadas instrucciones se ejecutan normalmente y, por lo tanto, se actualizan las salidas (externas o internas) seleccionadas entre las instrucciones JMP y JME. Si el resultado de la operación lógica es cero, la porción de programa comprendida entre JMP y JME no se ejecuta y, por lo tanto, no se modifica el estado de ninguna de la salida.

IL-ILC: este par de instrucciones hacen que todas las salidas seleccionadas entre ellas se actualicen normalmente o sean puestas a cero, dependiendo de que el resultado de la operación lógica inmediatamente anterior a IL, sea uno o cero respectivamente.

3.7.4 Grafcet

Es un método grafico de modelado de sistemas basado en automatismo de carácter secuencial. Surge como una secuencia de la voluntad de unificar y racionalizar los lenguajes de descripción relativos a los sistemas lógicos en general, y de los automatismos de carácter secuencial en particular. Los trabajos de investigación que han precedido y originado este método, con total implantación en el mundo industrial de los automatismos, fueron los trabajos de P.GIRAUD que introdujo los conceptos de receptividad y etapa, y en la misma época en EE.UU. Los de Karl Petri que, a partir de su tesis doctoral, introduce las denominadas redes de Petri[15].

[15] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

El Grafcet no se aplica buscando la minimización de las funciones lógicas que modelan la dinámica del sistema. La mejor cualidad del Grafcet radica en que es una herramienta poderosa y sobretodo metodológica, para la implementación de los automatismos de carácter secuencial. Mediante una aplicación adecuada presenta ciertas cualidades en los modelos que pueden implementarse, tales como:

- Claridad
- Legibilidad
- Presentación Sintética

Las principales características del Grafcet como herramienta de modelado de sistemas de eventos discretos son:

- Ofrecen una metodología de programación estructurada TOP DOWN (en forma descendente) que permite el desarrollo conceptual, de lo general a lo particular, descendiendo a niveles muy precisos de descripción y descomposición en las diversas tareas a llevar a cabo por el automatismo en sus distintas fases de ejecución y funcionamiento.

- Permite la introducción del concepto de diseño estructurado, de forma que las diversas tareas del automatismo se estructuran de forma jerarquizada, mediante el forzado de eventos de modelos Grafcet jerárquicamente superiores.

Definición de Conceptos y Elementos Gráficos Asociados:

- Etapa

Se define como la situación del sistema en la cual todo o una parte del órgano de mando es invariante con respecto a las entradas y salidas del sistema automatizado.

La característica fundamental de la etapa es la inclusión intrínseca del concepto de activación y la de acción o acciones asociadas. La etapa puede estar activada o no activada (marcada o no-marcada)[16].

Gráficamente la etapa normal se representa por un rectángulo que se enumera en su interior y en la parte superior, dando de esta manera un sentido de secuencialidad a las etapas.

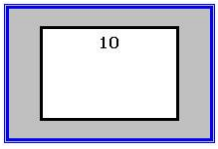


Figura 42. Etapa

Las etapas pueden ser:

- Etapas Normal: esta ligada a una transición de entrada y otra de salida.

[16] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

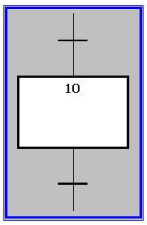


Figura 43. Tipos de Etapa

- Etapa de inicialización: Son aquellas que deberán quedar activadas al comienzo de la ejecución del algoritmo de control. Cuando la etapa es de inicialización el rectángulo se representa con doble recuadro.

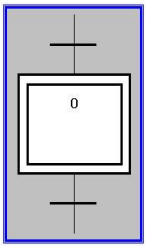


Figura 44. Tipos de Etapa

- Etapa Fuente: Es la etapa que no posee transición de entrada.

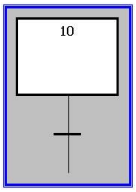


Figura 45. Tipos de Etapa

- Etapa Sumidero: es la etapa que no posee trancision de salida y por lo tanto no esta conectada con ningun elemento de salida.

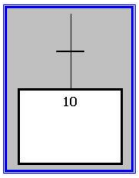


Figura 46. Tipos de Etapa

- Acción Asociada

Se trata de una o más posibles operaciones a realizar sobre el sistema, cuando la etapa de la cual depende dichas operaciones se encuentra activada. La situación de etapa activa se indica mediante la colocación de una marca en el interior del gráfico representativo de la etapa[17].

La acción o acciones elementales a realizar durante la etapa en el sistema, vienen indicadas mediante las etiquetas, que son rectángulos conectados a las etapas correspondientes y situados a la derecha de las mismas. En su interior se indica, bien de forma literal, bien de forma simbólica lo que debe realizarse.

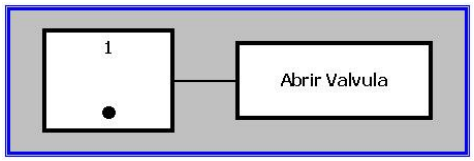


Figura 47. Acción Asociada

Cuando la etapa esta activa, se ejecutan sobre el proceso las acciones elementales referenciadas. Dichas acciones se pueden clasificar como siguen:

- Reales: Se trata de acciones concretas, que se producen en el automatismo, tales como, abrir / cerrar una válvula, arrancar / parar un motor, etc. A su vez se clasifican en:

[17] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

- Internas: Son acciones que se producen en el propio dispositivo de control, tales como temporizaciones, operaciones de cuenta, cálculos numéricos, etc.
- Externas: Se producen sobre el proceso en sí, externamente respecto al dispositivo lógico de control.
- Incondicionales: Son acciones que se ejecutan con solo quedar activadas las etapas correspondientes.
- Condicionales: Son aquellas que requieren el cumplimiento de una condición adicional a la propia activación de la etapa correspondiente.

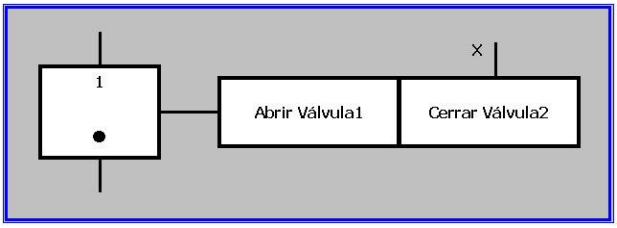


Figura 48. Acción Condicional

- Virtuales: No se realizan en una acción sobre el sistema; suelen utilizarse como situaciones de espera a que se produzcan determinados eventos o se activen determinadas entradas al sistema, que permitan la evolución del proceso. En estas etapas la etiqueta esta vacía o sin etiqueta. El estado de activación de una etapa se indica gráficamente, mediante la colocación de la señal de destino (Token) en el interior de la etapa[18].

Una etapa no activa puede ser, a su vez, inactiva o activable. La diferencia entre ambas viene dada por debido a que en la etapa activable la transición

[18] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

inmediatamente anterior esta validada; y lo esta por que la etapa inmediatamente anterior esta activa. Por todo ello una etapa es activable cuando esta activa la etapa precedente.

- Transición y Receptividad

El concepto de transición se asocia a la barrera existente entre dos etapas consecutivas y cuyo franqueamiento hace posible la evolución del sistema. A toda transición le corresponde una condición de transición o función lógica booleana que se denomina receptividad, y que puede ser verdadera o falsa[19].

Se dice que la transición esta validada, cuando la etapa o etapas inmediatamente precedentes están activadas. El franqueamiento de la transición se producirá si, y solo si, la transición esta validada y la receptividad es verdadera. Gráficamente se representa mediante un segmento de recta dispuesto ortogonalmente al arco, como se indica en la figura.

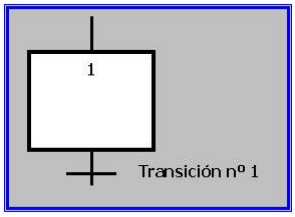


Figura 49. Transición

[19] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

La receptividad puede escribirse de forma literal, o de forma simbólica y debe situarse a la derecha del símbolo gráfico de la transición, en la siguiente se disponen diversas maneras de escritura de la receptividad.

- Caso a: Forma literal
- Caso b: Forma simbólica
- Caso c: Forma especial de indicar receptividad siempre verdadera.
- Caso d: En la receptividad se toma en cuenta el flanco de subida asociado a la variable b
- Caso e: En la receptividad se toma en cuenta la activación simultanea de ambos flancos descendentes asociados a las variables a y c.
- Caso f: En este caso se toman en cuenta el flanco ascendente coincidente con el final de la Temporización.

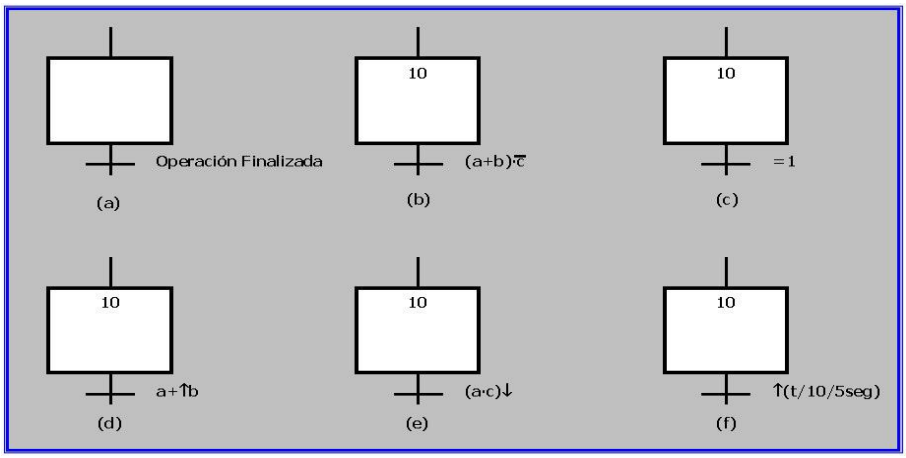


Figura 50. Forma de representar la receptividad

Por otra parte, las transiciones pueden clasificarse en:

- Transición Fuente: Sino esta ligada a una etapa anterior.
- Transición Sumidero: Si no esta ligada a una etapa posterior.

- Arco

Segmento de recta que une una transición con una etapa o viceversa pero nunca elementos homónimos entre si, obligatoriamente se deben cumplir las alternativas entre etapas y transiciones.

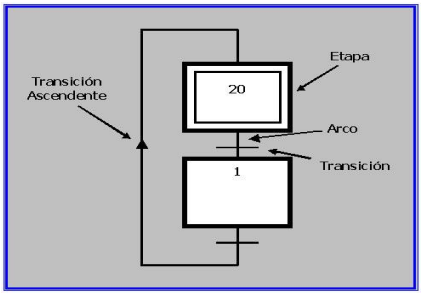


Figura 51. Etapas y Transiciones

Los arcos pueden representarse en vertical y horizontal al construir los diagramas, si bien los arcos verticales quedaran orientados mediante flechas en e caso que su sentido sea ascendente.

- Trazos Paralelos

Se utilizan para representar varias etapas cuya evolución esta condicionada con una misma transición. Los trazos paralelos se utilizan para la implementación del concepto de concurrencia entre subprocesos.

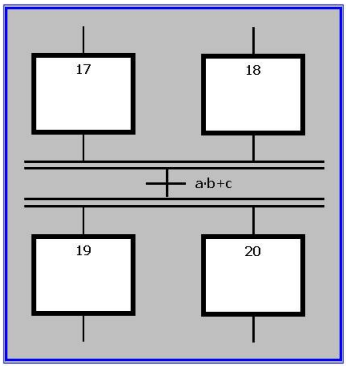


Figura 52. Trazos Paralelos

3.7.4.1 Reglas de Evolución

La dinámica evolutiva del Grafcet viene dada por un conjunto de reglas que pasamos anunciar seguidamente. También realizaremos una serie de definiciones que nos ayudaran a realizar el comportamiento del sistema. Todo ello nos va a permitir hacer un seguimiento de las marcas, a través del diagrama funcional[20].

a) No se produce franqueamiento por no estar validada la transición..

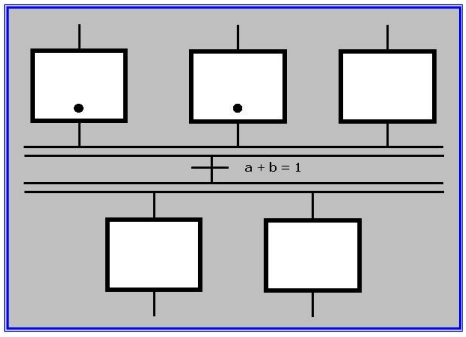


Figura 53. Reglas de Evolución

b) No se produce franqueamiento por no ser cierta la condición

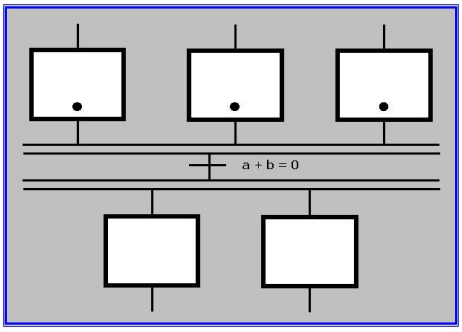


Figura 54. Reglas de Evolución

[20] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

c) Regla de Evolución

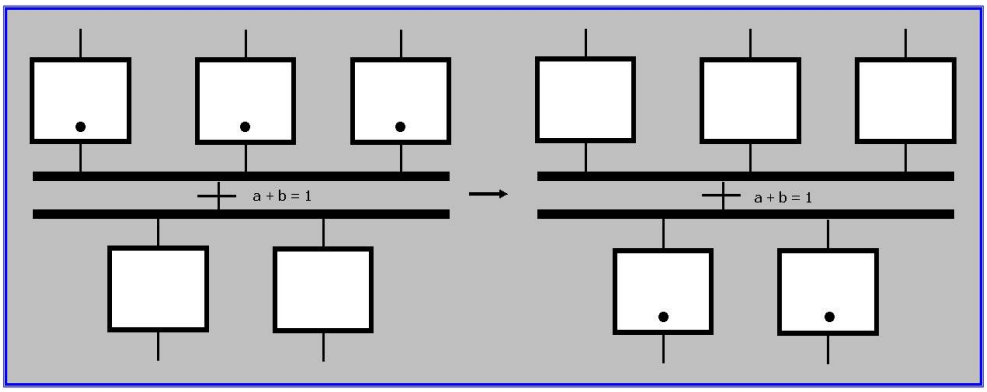


Figura 55. Reglas de Evolución

- Condiciones Evolutivas

Las cinco reglas que se describen seguidamente, definen la dinámica evolutiva de los sistemas modelados mediante esta herramienta:

- Regla Nº 1

La situación inicial de un Grafcet, caracteriza el comportamiento inicial de la parte de control frente a la parte operativa, el operador o los elementos exteriores. Se corresponde con las etapas activadas al principio de funcionamiento y se corresponde en general con una situación de reposo.

- Regla Nº 2

Una transición se dice valida cuando todas las etapas inmediatamente precedentes, unidas a dicha transición, están activadas.
El franqueamiento de una transición se produce:

- Cuando la transición esta validada.
- Cuando la receptividad a dicha transición es verdadera

Una transición puede estar:

- Validada.
- No validada.
- Liberada (Franqueada).

Una etapa se define como activable, si la transición precedente esta validada.

❖ Regla Nº 3

El franqueamiento de una transición tiene como consecuencia la activación de todas las etapas siguientes inmediatas, y la desactivación de las precedentes.

❖ Regla Nº 4

Transiciones conectadas en paralelos franqueables, se franquean de forma simultanea si se cumplen las condiciones para ello.

La regla de franqueamiento simultaneo permite la descomposición de un Grafcet en varios diagramas, de forma que asegura rigurosamente su sincronización. En este caso, es indispensable hacer intervenir en las receptividades los estados activos de las etapas.

❖ Regla Nº 5

Si durante su funcionamiento una misma etapa es simultáneamente activada y desactivada, deberá mantenerse activada.

3.7.4.2 Estructuras en El Grafcet

Consisten en una serie de estructuras que dotan al Grafcet de una gran capacidad de representación grafica de los automatismos. A grandes rasgos pueden ser clasificadas en estructuras básicas y lógicas. Las básicas atienden a conceptos tales como secuencialidad y concurrencia, y permiten realizar el análisis del sistema mediante su descomposición en subprocesos. Las estructuras lógicas atienden a conceptos de concatenación entre si de las anteriores estructuras[21].

- Estructuras Básicas

- Secuencia Única

Esta compuesta de un conjunto de etapas que van siendo activadas, una tras otra, sin interacción con ninguna otra estructura. En la secuencia única, a cada etapa le sigue una sola transición y cada transición es validada por una sola etapa. La secuencia se dice que está activa, sí una de sus etapas lo esta. Se dice inactiva, si todas sus etapas lo están.

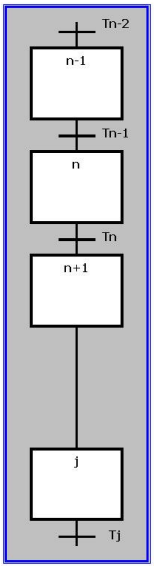


Figura 56. Secuencia única

- Secuencias Paralelas

Se denominan así al conjunto de secuencias únicas que son activadas de forma simultanea por una misma transición. Después de la activación de las distintas secuencias su evolución se produce de forma independiente.

---

[21] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

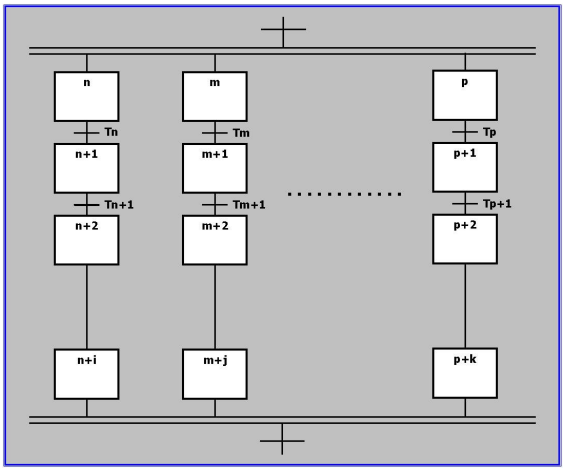


Figura 57. Secuencia paralelas

- Estructuras Lógicas en el Grafcet

Las estructuras lógicas OR Y AND son utilizadas para realizar el modelado de los conceptos de secuencias exclusivas y secuencias concurrentes respectivamente. Mediante la utilización de dichas estructuras se dota al modelo de un aspecto legible, ya que estos conceptos pueden ser implementados e identificados de inmediato.

- Divergencia en OR

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

Esta estructura lógica debe utilizarse cuando se trata de modelar la posibilidad de tomar dos o mas secuencias alternativas a partir de una etapa común. El Grafcet permite el franqueamiento simultaneo de las transiciones participantes, de modo que puedan dispararse de forma concurrente. No

obstante esta estructura no es la mas adecuada para la implementación de la concurrencia, debido a los problemas de sincronismo que introduce.

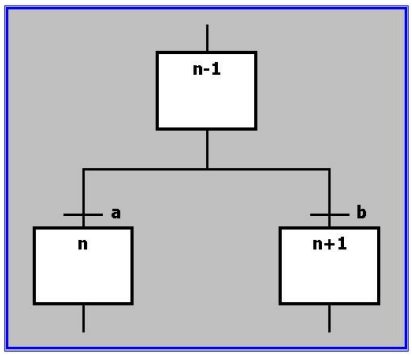


Figura 58. Divergencia en OR

- Convergencia en OR

La etapa n pasa a ser activa si estando 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.

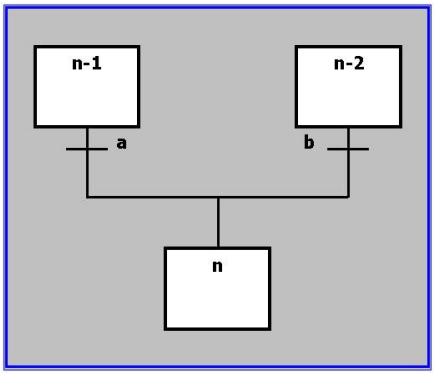


Figura 59. Convergencia en OR

- Divergencia en AND

Las etapas n+1 y n+2 pasan a ser activas si estando activa la etapa n, se satisface la transición f cuya receptividad es d+c.

Mediante esta estructura lógica se implementa el concepto de concurrencia y sincronismo, de forma que dos o mas subprocesos del sistema, representados por las secuencias paralelas, pueden activarse de forma sincronizada; y después de esto evolucionar concurrentemente de forma independiente.

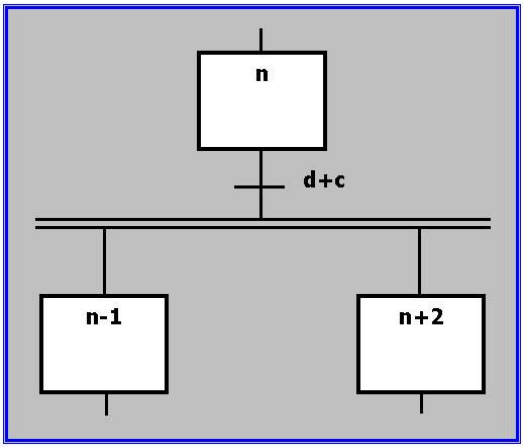


Figura 60. Divergencia en AND

- Convergencia en AND

La etapa n pasa al estado activo, si las etapas n-1 y n-2 están activas, se satisface la receptividad f.

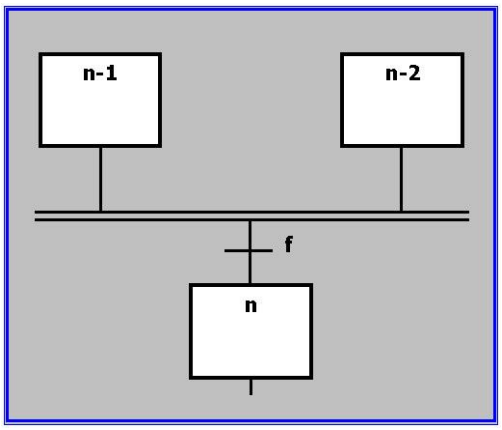


Figura 61. Convergencia en AND

- Saltos Condicionales

En el Grafcet de la figura (a) se produce un salto de la etapa n a la etapa n+i+1, si la receptividad representada por la variable A es cero. Si A es uno se prosigue la secuencia n, n+1, n+2, etc.

En la siguiente figura (b) se produce una repetición de la secuencia de etapas n, n+1, n+2, ..., n+i, en forma de bucle, mientras se mantenga el valor de D en cero.

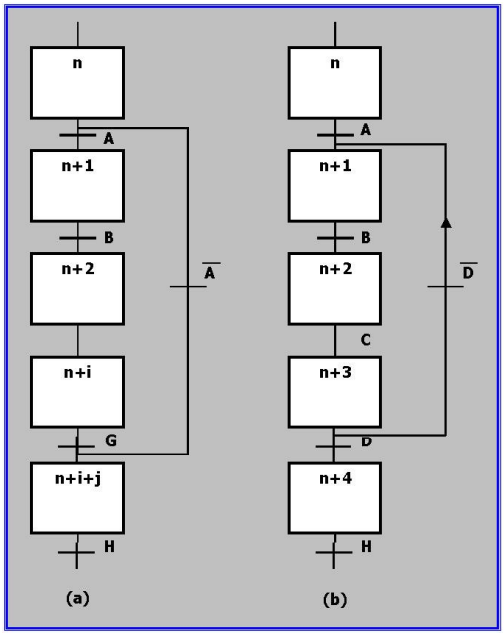


Figura 62. (a) Salto Condicional, (b) Bucle.

3.7.4.3 Macro Representaciones

El objeto de estos es la simplicidad en los diagramas, esta idea persigue que la complejidad en los modernos sistemas automatizados puedan reducirse, evitando que conceptos de tipo general puedan ser enmascarados por la profusión de análisis exhaustivos de mayor detalle, cuya especificación pueda quedar reseñada en documentación a parte[22].

- Repetición de Secuencias

Un conjunto de etapas que suele repetirse en varias ocasiones a lo largo del diagrama Grafcet se puede representar de forma compacta, esta forma de representación recibe el nombre de macro-etapa. Su uso persigue el objetivo de simplificar los modelos a ser representados que contengan secuencias repetidas de etapas y transiciones. Asimismo, su utilización permite el ahorro de códigos en los programas de los Controladores Lógicos Programables[23].

La macro-etapa es una representación simbólica de un conjunto de etapas y transiciones, este conjunto recibe el nombre de expansión de la macro-etapa. La expansión de la macro-etapa posee una única etapa de entrada y una única etapa de salida y constituyen los únicos lazos de unión con el Grafcet al que pertenecen.

La macro-etapa responde a las reglas siguientes:

---

[22] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

[23] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

- La expansión de la macro-etapa comporta una etapa de entrada, E, y una etapa de salida S.

- El franqueamiento de una transición inmediatamente precedente a la macro-etapa activa la etapa de entrada (E) de su expansión.

- La etapa de salida S participa en la validación de las transiciones inmediatas y posteriores a la macro-etapa.

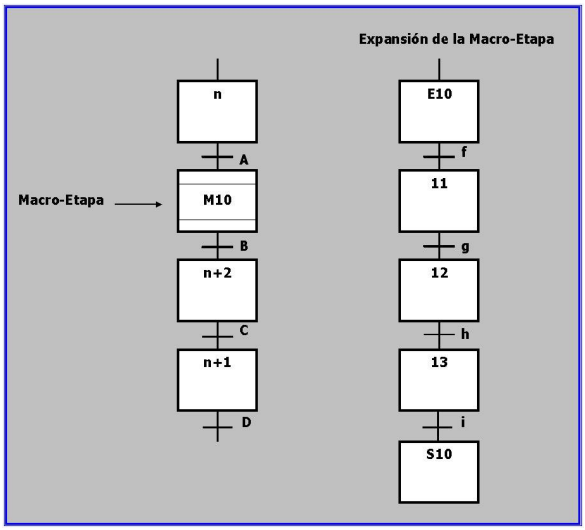


Figura 63. Macro Etapa

Ejemplo:

Controlar el paso de vehículos no autorizados por medio de un brazo de bloque. Con el interruptor S4 activado por el conductor se habilita el paso del vehículo poniendo en funcionamiento el motor que sube el brazo.

Los interruptores de posición S1 y S2 detectan las posiciones extremas del brazo y se encargan de desactivar el motor al final de su recorrido. Con el interruptor S3 se inhibe la actuación del brazo para el cierre, si el vehículo se

encuentra aún estacionado. Una vez que el vehículo pasa, el motor se conecta automáticamente después de 10 seg y baja el brazo.

Al final de su recorrido, en la posición de reposos, se activa S1 el cual se encarga de desconectar el motor definitivamente hasta la aparicion de un nuevo vehículo.

Ver grafica:

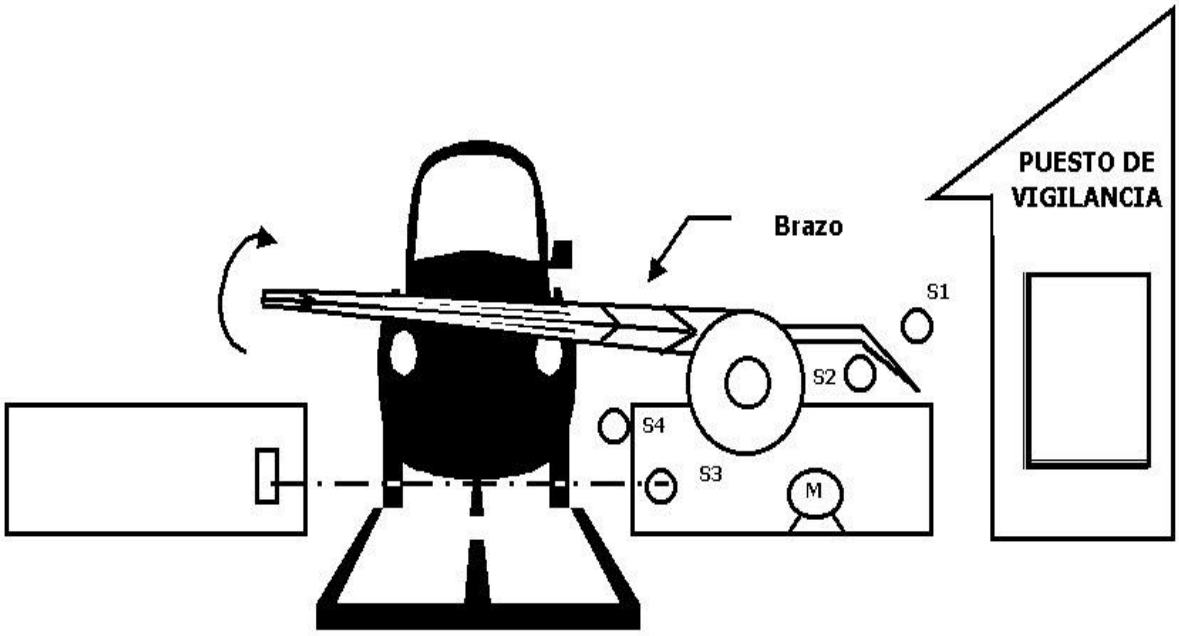


Figura 64.Grafico para Grafcet a realizar

Solución Grafcet:

3.8 REGISTROS Y ACUMULADORES

Son memorias en la que se almacena temporalmente datos, instrucciones o direcciones mientras necesitan ser usados por el microprocesador. Los registros mas importantes del microprocesador son los de instrucciones, datos, direcciones, acumulador, contador de programas, de trabajo y de bandera[24].

[24] GARCIA MORENO, Emilio. Control de Procesos Industriales. Alfaomega, México, 2001.

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

- Registro de estado (VKE): Su tamaño es de 1 bit. Aquí es donde efectuaremos las instrucciones combinacionales, la carga de entradas y la asignación de salidas a nivel de bit.
- Acumuladores (AKKU1 y AKKU2): Sus tamaños son de 16 bits cada uno. Cada vez que carguemos un dato en los acumuladores se seguirá la siguiente secuencia:

  Contenido de AKKU2 → Se pierde el contenido

  Contenido de AKKU1 ===> AKKU2

  DATO ===> AKKU1

A su vez, cuando realicemos una operación entre AKKU' s (como suma o resta) el resultado se introducirá en el AKKU1, perdiéndose el valor antes allí contenido.

3.9 TEMPORIZADORES Y CONTADORES

Para la generación de intervalos de tiempos se utilizan los temporizadores. Fundamentalmente existen los temporizadores con retardo a la conexión y a la desconexión. El dispositivo en cuestión posee una señal de inicio donde es aplicada un flanco de subida de una variable. Igualmente posee una señal de salida cuyo estado lógico se sitúa de la siguiente manera:

Si el temporizador es con retardo a la desconexión después de emitida la señal de inicio y transcurrido el periodo de tiempo determinado, la señal de salida se coloca en estado alto. Por el contrario, si es orientado a la

conexión, la salida permanece en estado lógico alto durante el periodo de Temporización ajustado.

Por otra parte, los dispositivos contadores se utilizan para realizar operaciones de cuenta, de tipo ascendente, descendente o incluso ascendente / descendente. El valor de la cuenta se actualiza cada vez que un flanco de subida se aplica en una determinada entrada. La salida del dispositivo se sitúa en estado lógico alto cuando se alcanza el valor de cuenta predefinido.

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 Z7. Para consultar el estado de cada uno de ellos podremos usarlos como si fueran entradas (mediante operaciones combinacionales) o introduciendo su valor en los AKKU[25].

3.10 CONSTANTES

A la hora de cargar 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 (Nº en coma fija, +32768 a -32768).
- KH: 16 bits (nº hexadecimal, 0000 a FFFF).
- KM: 16 bits (binario natural).
- KY: 16 bits (2 bytes, 0 a 255 en decimal cada uno).

[25] http://www.femz.es/cursos/Automatas.htm

- KT: 16 bits (valor de preselección de temporizadores, 0.0 a 999.3 en decimal).
- KZ: 16 bits (valor de preselección de contadores, 0 a 999 en decimal).

3.11 ESTRUCTURA DEL PROGRAMA

Vamos a tener 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 adquirimos la posibilidad de poder llamar a un módulo desde distintas partes del programa (lo que evita repetir código).

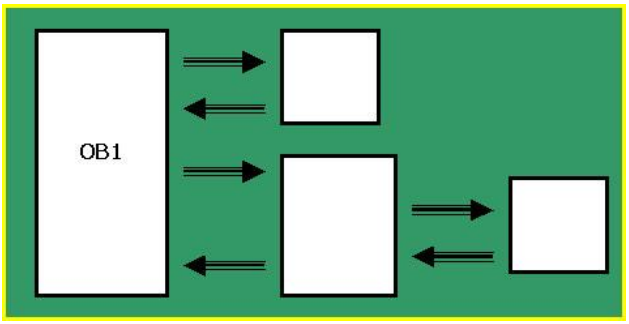


Figura 65. Forma de estructura de programa

- 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 no superemos los 16 niveles de salto que permite como máximo el autómata. Otras limitaciones son:

- El salto de un módulo a otro debe ser siempre hacia adelante (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 estructurada los módulos acabarán mediante la instrucción BE.
- La memoria del autómata S5-90U está limitada a 2K bytes. Cada instrucción ocupa generalmente 2 bytes, por lo que dispondremos de 1000 líneas de programa aproximadamente.

3.12 TIPOS DE MÓDULOS

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

- Módulos de organización (OB): Son los que gestionan el programa de usuario. Numerados OB1, OB3, OB21 y OB22.

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. El OB22 es empleado por el sistema operativo.

- Módulos de programa (PB): Son los que incluyen el programa de usuario dividido, normalmente, según aspectos funcionales o tecnológicos. PB0 ... PB63
- Módulos funcionales (FB): Son módulos de programa especiales. Aquí se introducen las partes de programa que aparecen con frecuencia o

poseen gran complejidad. Poseen un juego de instrucciones ampliado. FB0 ... FB63

- Módulos de datos (DB): En ellos se almacenan datos para la ejecución del programa, como valores reales, textos, etc. Adoptan los valores: DB0 ... DB63

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

- 256 palabras de datos. Para emplear un módulo de datos es necesario activarlo previamente (como se verá más adelante).

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 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.

Capitulo 4. EJEMPLOS Y PRACTICAS DE LABORATORIO

4.1 PRACTICA 1. INVERSOR DE SENTIDO DE GIRO PARA UN MOTOR TRIFÁSICO DE INDUCCIÓN.

Diseñar un programa para el control de un motor trifásico de inducción con dos posibilidades en el sentido de giro. Un pulsador arrancara el motor y éste girará en sentido Horario, mientras que otro pulsador hará que gire en sentido antihorario. Los prerrequisitos son que el pulsador de protección de entrada y el de parada no estén activados (Abiertos). La conmutación de los pulsadores para hacer el cambio de rotación no se podrá hacer sino hasta después de que el pulsador de parada es presionado y un tiempo de 5 segundos ha transcurrido.

De esta manera el motor se puede detener y puede arrancar en sentido opuesto, si es necesario. Si ambos pulsadores son presionados al mismo tiempo, el motor se debe detener y no arrancar.

F1: Fusible Principal
F2: Relé Térmico
K1: Contactor Derecha
K2: Contactor Izquierda

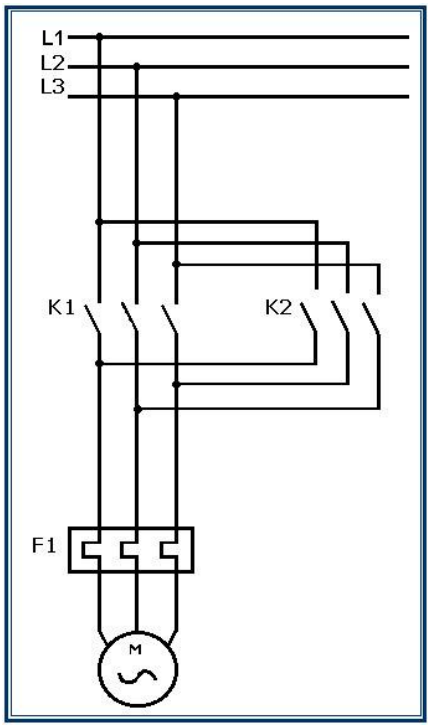


Figura 66. Esquema Eléctrico de Potencia de Inversión del Sentido de Giro de un motor trifásico.

Se pide:

a) Programa en Formato AWL
b) Programación en Formato KOP
c) Programa en Grafcet

SOLUCION

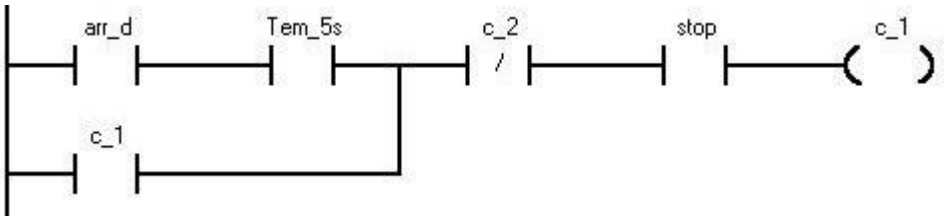
a) programa en formato AWL , elaborado para trabajar en PLC Siemens S7-200.

```
NETWORK 1           //INVERSOR DE GIRO. ARRANQUE INICIAL
DERECHA
LD      "ARR_D"
U       "TEM_5seg"
O       "C_1"
UN      "C_2"
U       "STOP"
```

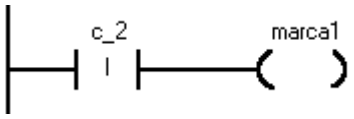
```
=       "C_1"
NETWORK 2       //HABILITA MARCA ARRANQUE A LA DERECHA
LDN     "C_2"
=       "MARCA1"

NETWORK 3       //HABILITA TEMPORIZADO
LD      "MARCA1"
TON     "TEM_5seg", +50

NETWORK 4       //ARRANQUE INICIAL IZQUIERDA
LD      "ARR_I"
U       "TEM5seg"
O       "C_2"
U       "STOP"
UN      "C_1"
=       "C_2"

NETWORK 5       //HABILITA MARCA ARRANQUE IZQUIERDA
LDN     "C_1"
=       "MARCA2"

NETWORK 6       //HABILITA TEMPORIZADO
LD      "MARCA2"
TON     "TEM5seg", +50

NETWORK 7       //FIN
MEND
```

b) programa en formato KOP , elaborado para trabajar en PLC Siemens S7-200.

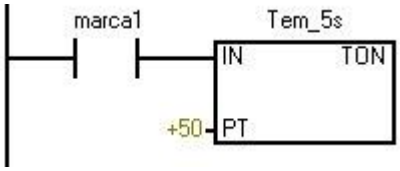
Seg1. Con este segmento se pone en funcionamiento el motor con giro hacia la derecha, activando el Contactor uno.



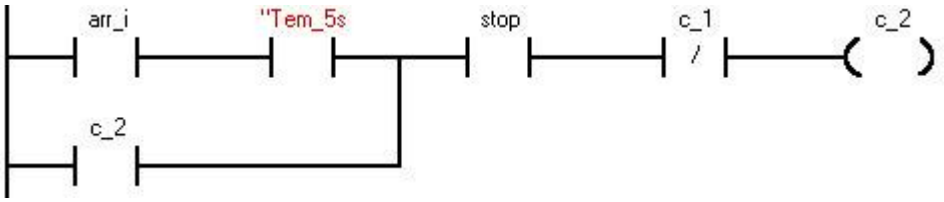
Seg2. Se carga la variable Contactor dos en una marca de memoria.



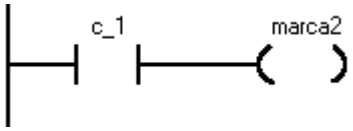
Seg3. Temporización de seguridad para la inversión de giro



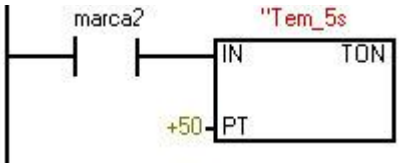
Seg4. Segmento que activa la inversión de giro hacia la izquierda



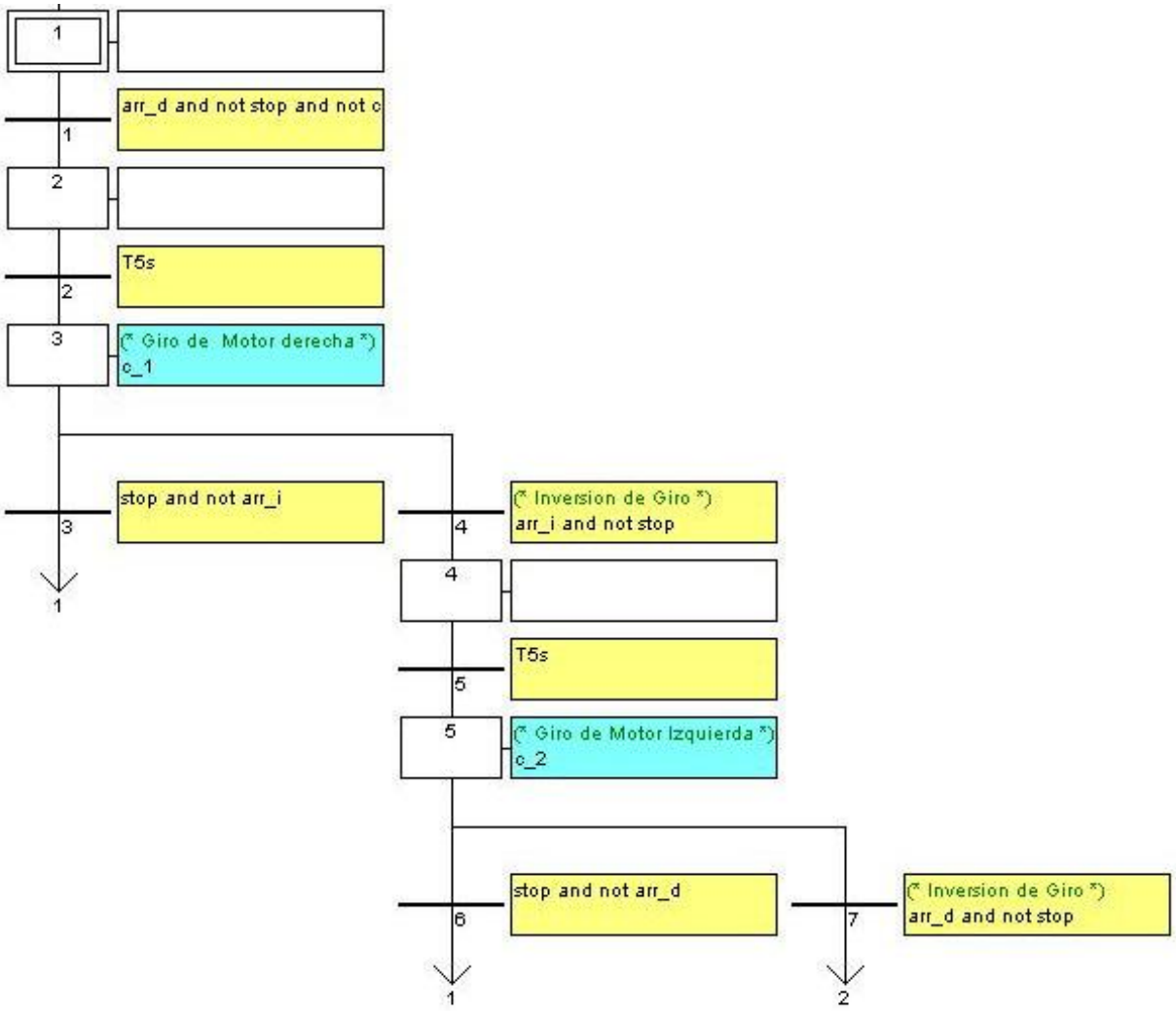
Seg5. cargar la variable Contactor dos en una marca de memoria.



Seg6. Temporización de seguridad para la inversión de giro.



c) Diagrama de Grafcet, realizado en ISAGRAF 3.4



4.2 PRACTICA 2. SISTEMA DE DOS BANDAS TRANSPORTADORAS BANDA_A Y BANDA_B

Deben cumplir con el siguiente ciclo de trabajo: Al activar el pulsador START comenzará a funcionar la Banda A, que transporta piezas sobre ella hasta el comienzo de la resbaladera. Al llegar a este punto, las piezas caen por gravedad por dicha resbaladera y al pasar por el sensor luminoso f1 lo activan, produciendo la parada de la Banda A, 5 segundos después y la puesta en marcha de la Banda B. Las piezas caídas en la Banda B se desplazan por ella hasta llegar al final de esta, donde esta ubicado otro censor luminoso f2. Al pasar las piezas por el sensor f2 se para la Banda B y se termina el ciclo de trabajo.

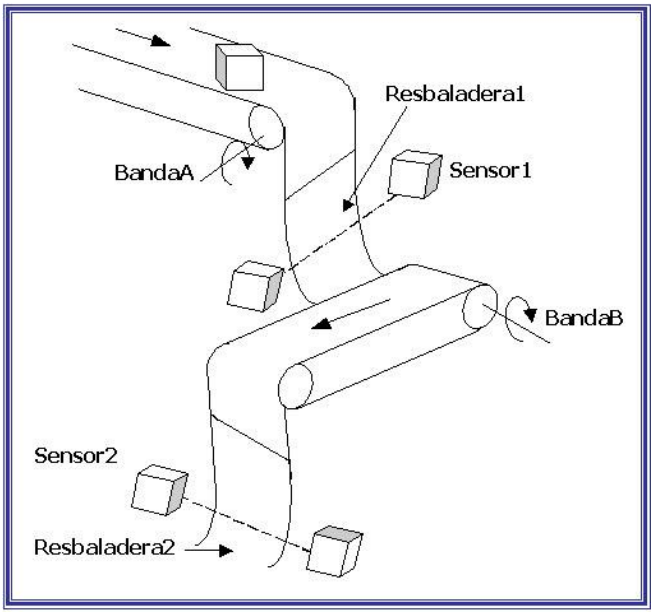


Figura 67. Esquema Bandas Transportadoras

Se pide:

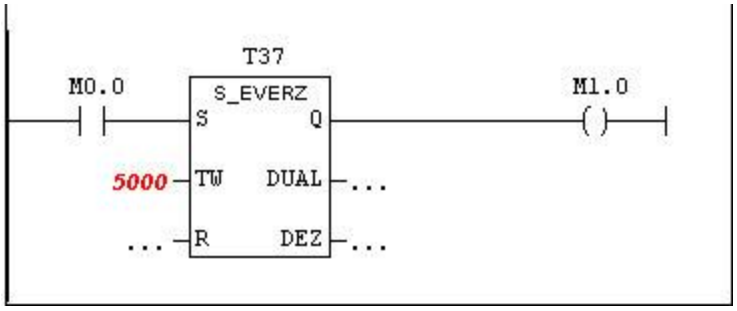
a) Programa en Formato KOP.
b) Programa en formato AWL.
c) Programa en Grafcet.

SOLUCION

a) programa en formato KOP, elaborado para ejecutar en PLC Siemens S7-200

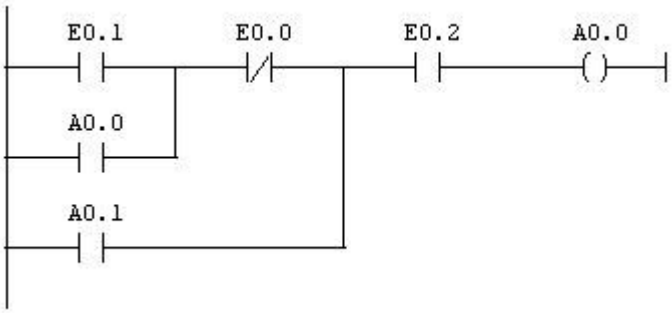
Seg1: **Temporización**

Este es un temporizador orientado a la conexión con un retardo de 5seg, es decir a los 5 segundos activa la salida.



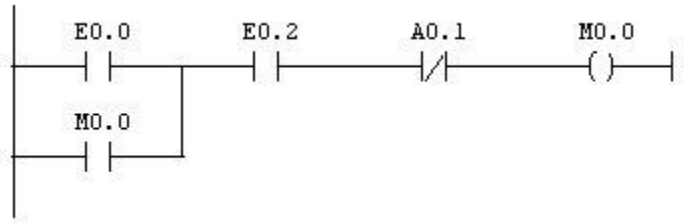
Seg2: **PUESTA EN MARCHA DE LA BANDA # 1**

En este segmento se activa la banda uno, con el pulsador de marcha(E0.1) o con la banda numera 2, también se puede parar con el pulsador de parada(E0.2) o con el sensor que es la entrada(E0.0).



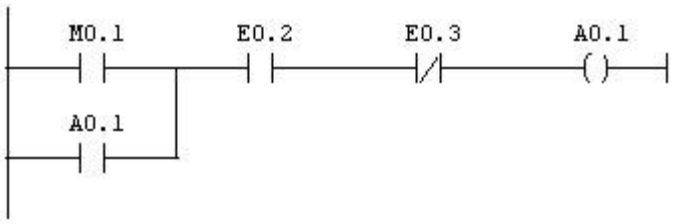
Seg3: **ACTIVACION DEL TEMPORIZADOR**

El temporizador se activa inmediatamente se ponga en uno el sensor(E0.0), y sé Desactiva ya sea cuando la banda dos este andando o cuando se pulse parada.



Seg4: **PUESTA EN MARCHA DE LA BANDA # 2**

La banda dos se activara cuando el temporizador después de contar los 5 segundos active la posición de memoria(M0.1) y se parara cuando el sensor dos(E0.3) mande la señal.

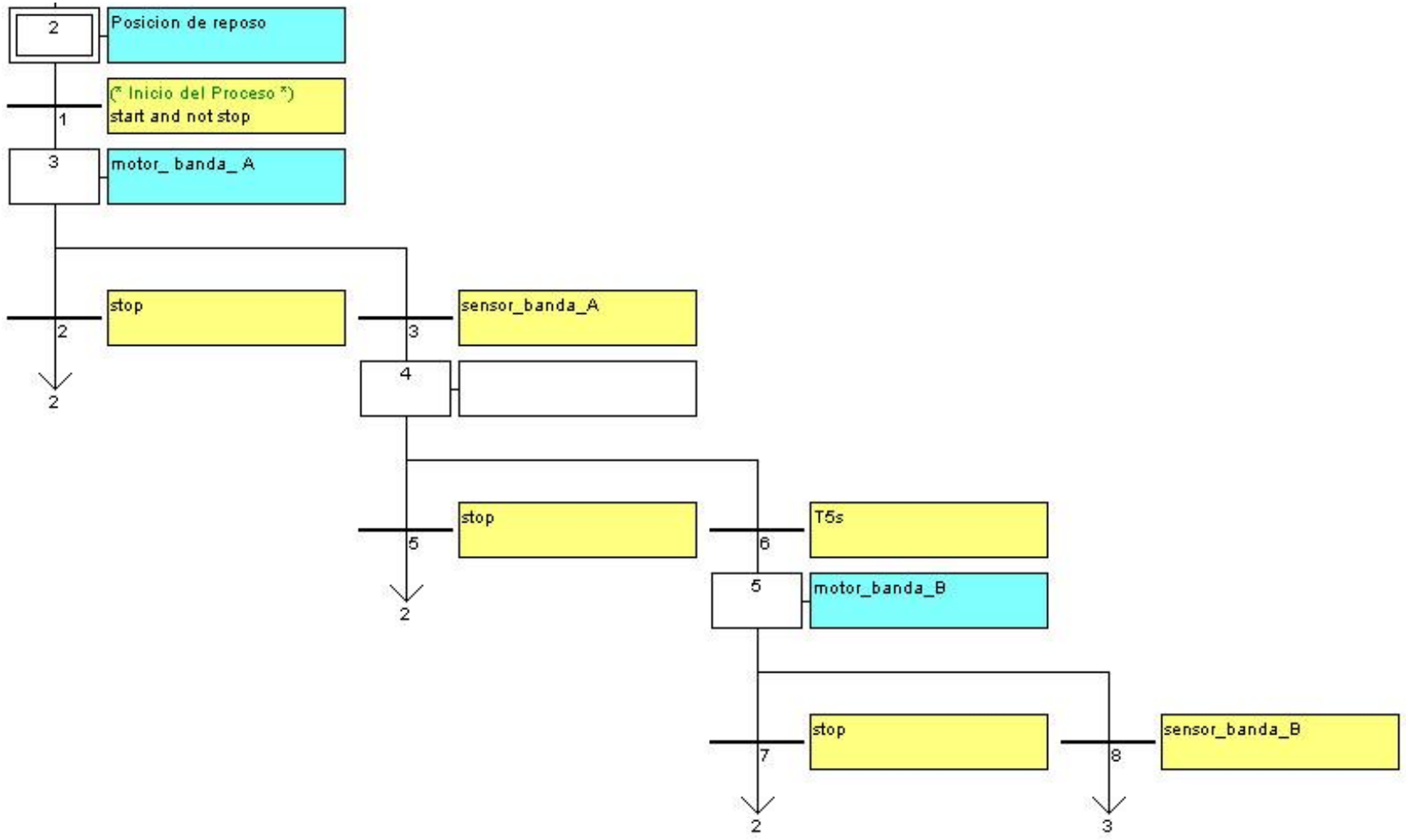


b) programa en formato AWL, elaborado para trabajar en PLC S7-200.

```
NETWORK 1       //ARRANQUE MOTOR BANDA A
LD      "start"
O       "Motor_Banda_A"
UN      "Tiempo_Stop_B_A"
U       "stop"
=       "Motor_Banda_A"

NETWORK 2       //SENSOR EN LA BANDA A, HABILITA BANDA B
LD      "Sensor_Banda_A"
ED
S       "Hab_Banda_B", 1
```

```
NETWORK 3
LD     "Sensor_Banda_B"
ED
S      "Hab_temp_B", 1

NETWORK 4          //SE HABILITA TEMPORIZADO PARA LUEGO
ARRANCAR MOTOR BANDA B
LD     "Hab_Banda_B"
TON    "Tiempo_Stop_B_A", +50

NETWORK 5       //SE DESABILITA TEMPORIZADO
LD     "Tiempo_Stop_B_A"
R      "Hab_Banda_B", 1

NETWORK 6
LD     "Tiempo_Stop_b_B"
R      "Hab_temp_B", 1

NETWORK 7
LD     "Hab_Banda_B"
O      "Motor_Banda_B"
UN     "Tiempo_Stop_b_B"
U      "stop"
=      "Motor_Banda_B"

NETWORK 8
LD     "Hab_temp_B"
TON    "Tiempo_Stop_b_B", +10

NETWORK 9       //FIN
```

c) Grafcet, elaborado en ISAGRAF 3.4



4.3 PRACTICA 3. SEMÁFORO PARA VEHICULOS Y PEATONES EN EL CRUCE DE DOS CALLES

El funcionamiento del semáforo será el siguiente: Cuando se activa el pulsador de inicialización del sistema se pasará al estado inicial, en el cual se encenderán todas las luces de todos los semáforos. El sistema permanecerá en este estado mientras dicho pulsador este activo. Después de la inicialización se pondrán los semáforos de coches en verde y los peatones en rojo. Cuando algún peatón pulse alguno de los botones de solicitud de paso se esperaran quince segundos y se pondrá el semáforo de coches en ámbar. Cinco segundos después se pondrá el semáforo de coches en rojo y tres segundo mas tarde se pondrá en el semáforo de peatones en verde (para evitar que los listillos atropellen a los peatones).

El semáforo de peatones se mantendrá en verde durante diez segundos. Una vez transcurridos los diez segundos, la luz verde del semáforo de peatones parpadeara con una frecuencia de un segundo durante seis segundos. A continuación se pondrá el semáforo de peatones en rojo y pasados tres segundos más (por si hay algún viejecito que aún no terminado de cruzar) se pondrá el semáforo de coches en verde. A partir de este instante una nueva pulsación en alguno de los botones de solicitud de paso provocara otro ciclo completo del semáforo. Si se pulsa algún botón de solicitud de paso en mitad del ciclo no deber ocurrir nada.

a. Solución en Grafcet.
b. Solución en KOP
c. Solución en AWL

a) Solución en Grafcet, elaborada en ISAGRAF 3.4.

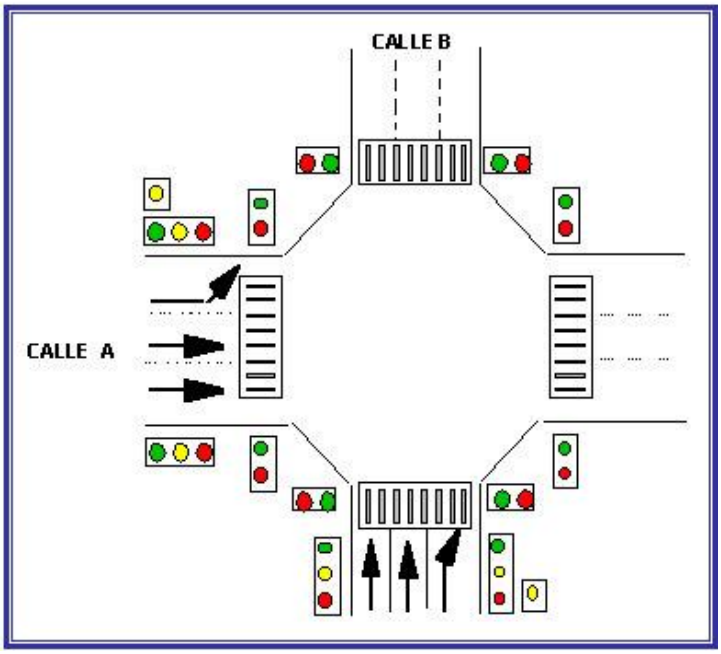


Figura 68 Señalización de las vías y disposición del conmutador para el control de semáforo

Entradas:

Símbolo	Descripción	Variable
RESET	Pulsador inicialización del sistema	**E0.1**
ESPD	Pulsador de solicitud de paso dcha.	**E0.2**
ESPI	Pulsador de solicitud de paso izda.	**E0.3**

Salidas:

Símbolo	Descripción	Variable
SLVC	Luz verde coches	**A0.0**
SLAC	Luz ámbar coches	**A0.1**
SLRC	Luz roja coches	**A0.2**
SLVP	Luz verde peatones	**A0.3**
SLRC	Luz roja peatones	**A0.4**

Estados:

Símbolo	Descripción	**Variable estado**	**Variable copia**
E0	Inicialización	M0.0	M10.0
E1	Luz verde coches	M0.1	M10.1
E2	Esperar 15s para cambiar	M0.2	M10.2
E3	Luz ámbar coches	M0.3	M10.3
E4	Luces rojas 3s	M0.4	M10.4
E5	Luz verde peatones	M0.5	M10.5
E6	Intermitencia luz verde	M0.6	M10.6
E7	Rojo peatones 3s	M0.7	M10.7

Grafcet: