



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

' ' A R A G O N ' '

**“CONTROLADORES LOGICO PROGRAMABLES,
IMPORTANTE ALTERNATIVA EN LA
AUTOMATIZACION INDUSTRIAL”**

T E S I S

Que para obtener el Titulo de:

INGENIERO MECANICO ELECTRICISTA

(AREA ELECTRONICA)

Presenta:

SANDRA TERESA ROMERO AVILA

Asesor: Ing. Oscar E. Álvarez Melendez

**TESIS CON
FALLA DE ORIGEN**

México, D. F. 1993



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINAS
FALTANTES

33-84-99-100

120-121-124-125

C O N T E N I D O

CAPITULO I

ANTECEDENTES DE LOS CONTROLADORES LOGICO PROGRAMABLES.	1
INTRODUCCION	2
ANTECEDENTES	2
DISEÑO CONCEPTUAL DEL PLC	3
PRIMEROS ALCANCES	4
EL PLC EN LA ACTUALIDAD	9
PANORAMA DE LOS CONTROLADORES LOGICO PROGRAMABLES	11

CAPITULO II

FUNCIONES LOGICAS ELEMENTALES.	13
LA FUNCION AND	15
LA FUNCION OR	17
LA FUNCION NOT	19
LOGICA Y ALGEBRA BOOLEANA COMO HERRAMIENTA DE DISEÑO	24
CIRCUITOS DEL PLC Y SIMBOLOGIA DE CONTACTOS LOGICOS	24
CONCEPTO DE DIRECCION EN EL PLC	26

CAPITULO III

COMPONENTES PRINCIPALES DE UN CONTROLADOR LOGICO PROGRAMABLE. FUNCIONAMIENTO	29
PRINCIPIOS DE OPERACION DE UN PLC	30
UNIDAD CENTRAL DE PROCESO	32
MONITOREO	35
VERIFICACION DEL PROCESO	37
VERIFICACION GLOBAL	38
VERIFICACION REITERATIVA CICLICA	39
VERIFICACION LONGITUDINAL REITERATIVA	40
VERIFICACION CICLO-OR-EXCLUSIVA	40

DIAGNOSTICO DEL SISTEMA	40
SISTEMA DE ALIMENTACION	41
ENTRADA DE VOLTAJE	41
AISLAMIENTO POR TRANSFORMADOR	43
MEMORIA	44
TIPOS DE MEMORIA	46
CAPITULO IV	
DISPOSITIVOS PERIFERICOS Y APARATOS DE PROGRAMACION	59
EL CONTROLADOR PROGRAMABLE COMO UN COMPUTADOR INDUSTRIAL	60
MODULOS E/S DIGITALES.	63
MODULOS DE ENTRADAS DISCRETAS.	66
MODULOS DE SALIDAS DISCRETAS	69
MODULOS DE ENTRADA NO VOLTAJE.	71
ENTRADAS/SALIDA TTL	71
OTROS DISPOSITIVOS DE SEÑALES DISCRETAS.	72
MODULOS E/S ANALOGICAS	72
MODULOS DE COMUNICACIONES Y FUNCIONES ESPECIALES	75
MODULOS DE COMUNICACIONES.	75
MODULOS DE FUNCIONES ESPECIALES.	79
APARATOS DE PROGRAMACION	81
CAPITULO V	
LENGUAJES DE PROGRAMACION	85
CONTROLADORES PROGRAMABLES Y EVOLUCION DEL SOFTWARE	86
CONTACTOS NA Y NC	87
INSTRUCCION DE SALIDA	90
TRASLACION A LA SIMBOLOGIA DE CONTACTOS	90
DIAGRAMA DE CONTACTOS	93

INSTRUCCIONES BASICAS	101
INSTRUCCIONES DE TEMPORIZACION Y CONTEO	114
TEMPORIZADORES	115
CONTADORES	122
INSTRUCCIONES DE CONTROL DE PROGRAMA Y FLUJO DE OPERACIONES	123
INSTRUCCIONES ARITMETICAS	130
INSTRUCCIONES PARA EL MANEJO DE DATOS	136
INSTRUCCIONES DE TRANSFERENCIA DE DATOS	142
REDES DE COMUNICACION	152
OTROS LENGUAJES DE PROGRAMACION	159
LISTA DE INSTRUCCIONES	159
DIAGRAMA DE FUNCIONES	169
GRAFICET	173
CAPITULO VI	
DISEÑO E IMPLEMENTACION DE UN SISTEMA BASICO	177
DOCUMENTACION	178
IMPLEMENTACION	180
DEFINICION DE CONTROL	181
ESTRATEGIA DE CONTROL	181
DESARROLLO DEL SOFTWARE	187
GUIA DE IMPLEMENTACION	189
EL DIAGRAMA DE FLUJO Y LA GENERACION SECUENCIAL DE SALIDAS	190
CONFIGURACION DEL SISTEMA DE PLC	191
PROBLEMA DE APLICACION	192
CONCLUSIONES FINALES	205
BIBLIOGRAFIA	207

CAPITULO I

ANTECEDENTES
DE LOS
CONTROLADORES
LOGICO
PROGRAMABLES.

I N T R O D U C C I O N .

El controlador lógico programable o PLC, puede considerarse como un miembro más de la industria de las computadoras dado que siendo un dispositivo capaz de almacenar instrucciones que establecen funciones de control; coordina la secuencia de operación de acuerdo con los tiempos, contadores, aritmética y datos involucrados es además un computador que realiza funciones especiales dentro del control industrial de máquinas y procesos.

Muchas definiciones describen al PLC únicamente como un controlador programable, sin embargo un controlador lógico programable puede concebirse en forma más clara y precisa como un computador industrial, cuya arquitectura amplía sus límites más allá de la unidad de proceso, estableciendo un medio de comunicación entre el proceso real y el controlador programable mediante módulos de interfases entrada/salida (I/O).

El diseño del controlador lógico programable adopta como base la simplicidad de uso y aplicación práctica a sistemas industriales sin importar la magnitud del proceso.

ANTECEDENTES.

El criterio de diseño para el primer controlador programable fue especificado en 1968 por la división hidramatic de la compañía General Motors. El objetivo primario fue la eliminación del alto costo de los sistemas controlados por tiempo.

Los propósitos requerían de un dispositivo que contara con un computador, que fuese capaz de sobrevivir en el difícil medio industrial, de fácil programación y mantenimiento para ingenieros y técnicos, reutilizable en otros sistemas cuando ya no se requiere en el actual; tales características reducirían el tiempo de los procesos, optimizando el uso de la máquina con lo que abrirían su desarrollo hacia el futuro.

DISEÑO CONCEPTUAL DEL PLC.

Los primeros controladores programables tendían a sustituir el uso de cableado de relevadores con lo que se ejecutaba la secuencia de operaciones lógicas en el proceso, estas operaciones frecuentemente eran repetitivas, lo que encarecía su realización con estos mecanismos que normalmente requerían de un cableado distinto por cada etapa del proceso.

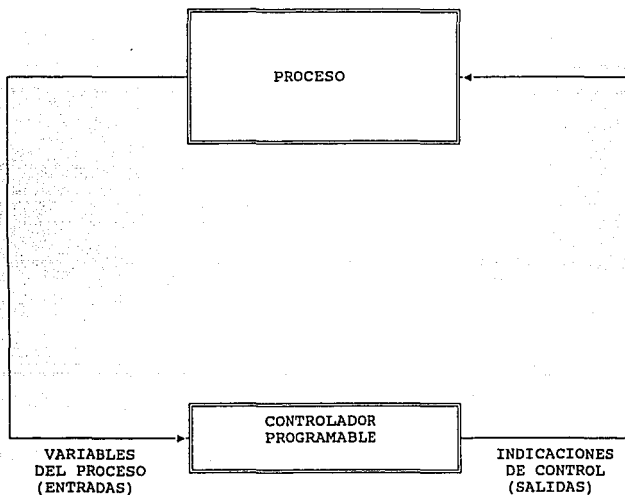


FIGURA 1.1. DIAGRAMA CONCEPTUAL DE LA APLICACION DE UN PLC

El controlador programable con otras posibilidades ofrece más ventajas que el cableado de relevadores: fácil instalación, utilización de menor espacio y energía, por ser un dispositivo "inteligente" puede proporcionar alternativas que faciliten la solución de problemas, en función de las condiciones que rodeen la falla, lo cual no es posible con un sistema de relevadores.

La arquitectura original del PLC superó los objetivos con que fue concebido el controlador por la General Motors, con lo que su proliferación en el medio industrial ha sido inminente.

El controlador lógico programable se puede considerar una novedad dentro del control tradicional, cuando se compara con los sistemas de relevadores con cableado fijo, instrumentación analógica y otros dispositivos con lógica de estado sólido con alta velocidad de respuesta.

El PLC ha mejorado a través de los años con el desarrollo de los semiconductores, que generó un considerable aumento en la velocidad de respuesta en las operaciones lógicas, el desarrollo de interfases y optimización en el procesamiento de datos; de cualquier forma estos versátiles dispositivos superaron la facilidad de uso y mantenimiento, razón de su origen.

PRIMEROS ALCANCES.

La presencia del controlador programable y sus primeras características de diseño, dieron fuertes razones a la industria automotriz para sustituir la tarea realizada por el cableado de relevadores con un sistema de control con PLC, con lo que se observaron las siguientes ventajas:

- * Precios competitivos con los sistemas tradicionales de control con cableado de relevadores.
- * Capacidad de mantenerse en el medio industrial sin que se puedan calificar de obsoletos.
- * Fácil montaje y desmontaje de sus partes, gracias a su diseño arquitectura modular.
- * Su diseño modular facilita el reemplazo de interfases I/O.
- * Comunicación entre el controlador y el sistema controlado lo que actualiza las variables de control y permite la manipulación de datos.

- * Métodos de programación que por su simplicidad son comprendidos fácilmente por el personal de planta.
- * Diagrama de escalera, método estándar de la presentación a un programa que contrarresta lo tedioso que pudiera resultar un programa por lo pequeño de sus instrucciones.
- * Capacidad de reutilización en otros proyectos.

Sin embargo, el producto que satisfizo las necesidades que impulsaron su creación permaneció en un camino oculto hasta que en 1969, se logró el primer controlador que superó los objetivos originales abriendo las puertas al desarrollo de una tecnología de control que en poco tiempo se ha venido extendiendo hacia otras áreas de la industria como la alimentaria, de metales, manufactura, de papel, etc.

El avance tecnológico, mejoró los procesos de fabricación de los semiconductores y con ello el desarrollo de circuitos integrados de alta escala de integración (VLSI) lo cual provoca la aparición del primer microprocesador industrial: el INTEL 8008 que con el éxito de sus características impulsó a los fabricantes de semiconductores a incluir entre sus productos microprocesadores con cada vez mejores características.

La llegada del microprocesador provocó que para los años setenta, los controladores programables sufrieran un cambio radical que agregó a sus cualidades mayor flexibilidad e "inteligencia" lo que optimizó la realización de las operaciones aritméticas y lógicas, manipulando funciones de datos mediante una comunicación entre el operador y el computador.

El tubo de rayos catódicos (TRC) utilizado en la pantalla de las computadoras para su programación viene a ser ahora el medio de comunicación entre el programador del PLC y el propio controlador.

El empleo del TRC ofreció una alternativa en la monotonía que presenta la introducción de un programa utilizando procedimientos de carga que varían en complejidad dependiendo del controlador. La utilización de la pantalla tiene la ventaja de permitir observar desde el diagrama de escalera de un proceso hasta la más alejada de las fallas en el sistema.

La "inteligencia" agregada al controlador programable permitió el desarrollo de lenguajes de programación como el Diagrama de Escalera o Diagrama de Contactos entre otros.

El conjunto de símbolos del diagrama de escalera puede crear la más difícil de las instrucciones, estas instrucciones activan la función programada. Otros símbolos se siguen sumando a los ya existentes, representando funciones de comparación, transferencia de datos y funciones aritméticas.

La suma de funciones aritméticas y mejores instrucciones aumentaron la aplicación de los controladores programables, permitiendo utilizarlos con aparatos de instrumentación para introducir datos numéricos de las variables de control.

La utilización de instrumentación entre el controlador programable y el proceso facilitó la lógica y secuencia de tareas, mejorando la exactitud de los cálculos, basada en la constante actualización de los datos; estos procedimientos son fundamentales en las operaciones del controlador programable.

El concepto de "inteligencia artificial" se generó cuando el avance en la tecnología de los computadores logró el desarrollo de sistemas capaces de tomar decisiones, lo que fue el principio de un sin fin de nuevos adelantos tecnológicos.

La evolución del hardware y software entre 1975 y 1979 agregó al PLC mayor capacidad de memoria, manejo de entradas y salidas remotas, control analógico, posicional, comunicación con el operador, detección de falla en la máquina y aumento en el software de aplicación.

Tales logros han hecho del controlador lógico programable una útil herramienta para un gran número de aplicaciones, contribuyendo en gran forma a la reducción de costos de implantación e instalación eléctrica.

Los sistemas con expansión de memoria permitieron almacenar largos programas de aplicación y grandes bloques de datos. El aumento de memoria no sólo facilitó las tareas de lógica y secuenciación, sino que mejoró la adquisición y manipulación de datos.

La habilidad de "memorizar" más datos implica que los pasos e información residente pueden estarse guardando y recuperando automáticamente durante el proceso.

Esta flexibilidad evita la necesidad de parar un proceso para restablecer o cambiar los parámetros. Es decir, que si algo altera el proceso, todos los valores presentes en el momento del suceso podrían ser cambiados y los rangos establecidos para las variables de control, modificados. El costo de instalación es significativamente más bajo, dado que el diseño del PLC adquiere características mejores.

Se obtienen ventajas en los medios con que el PLC se comunica con el proceso, optimizando la localización de señales E/S de sistemas remotos, distantes de la unidad central de proceso (CPU) y cercanos al proceso controlado.

Se introducen técnicas como el multiplexaje de señales, pudiendo ser transportadas todas señales del proceso hasta la CPU, utilizando tan sólo un par de cables en lugar de manejar un cable por cada subsistema.

Este procedimiento facilita el mantenimiento, permitiendo arrancar gradualmente cada uno de los sistemas que puede englobar un macrosistema.

Con el desarrollo del control analógico, el controlador programable, puede manipular el encendido y apagado del sistema en función del control de instrumentos, habilitando un puente en el manejo de variables analógicas y variables discretas.

Sin embargo, la activación del encendido en forma automática, no es conveniente para sistemas que utilizan procesos químicos como el tratamiento de aguas residuales, procesamiento de minerales, etc., aplicaciones de este tipo requieren la combinación del sistema encendido/apagado con el control de sus variables analógicas.

Durante este período, otros adelantos en hardware, permitieron al controlador programable la ejecución de operaciones de control de posición; mediante el uso de indicadores de paso (STEPPER) y la codificación de información, en los diferentes módulos o interfaces.

La interfase de entrada cuenta con un tren de impulsos que suministra un valor numérico al PLC para la verificación de un movimiento. Un dato enviado por el controlador habilita la interfase de salida, produciendo esta señal que es interpretada por el indicador de paso para realizar la operación correspondiente.

Las primeras aplicaciones de este tipo de interfaces, se realizaron en sistemas con procesos trituradores, líneas de transferencia y pinturas con atomizador.

La evolución en los sistemas de comunicación permitió a los controladores lógico programables comunicarse con otros dispositivos que mejoren la operación de interfaces.

El Hardware y software del TRC tuvieron un desarrollo más lento, la producción de tablas, manejo de reportes y mantenimiento de datos, dieron la posibilidad de generar una copia en firme de la información mediante medios como la impresora.

La rapidez con que se desarrollaron las redes de comunicación local, facilitó la tarea de control en todo el ámbito de una planta, pudiendo distribuirse esta tarea entre distintos controles, enlazando todas las señales indicadoras de cada control en una sola red de comunicaciones manejada por el control programable.

Con la perfección de los mecanismos y procesos de comunicación, como la transferencia en línea, que optimiza la localización y manejo de la información; la comunicación humana con el sistema puede reducirse o eliminarse, frente a los nuevos esquemas del control moderno.

El desarrollo de hardware generó un aumento de software de aplicación lo que ofrece a las computadoras el manejo de nuevas instrucciones, que dieron al controlador programable el mejor manejo y manipulación de la información, mejor comunicación con los dispositivos periféricos, dejando atrás instrucciones del tipo relevador tediosas o imposibles.

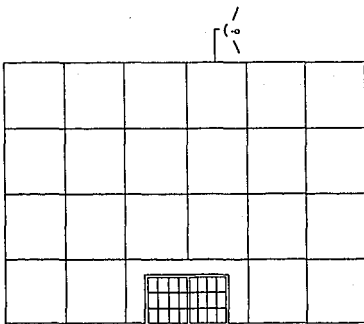
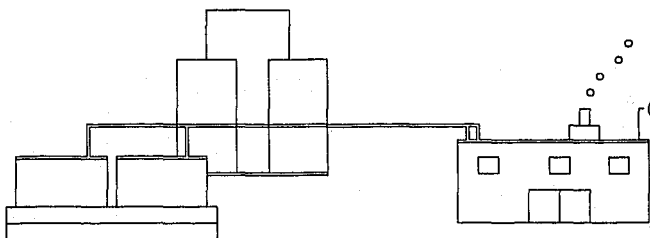


FIGURA 1.2.- LA DISTANCIA QUE DEBE HABER ENTRE EL CONTROLADOR Y EL PROCESO, DEPENDE DE LA CALIDAD EN LA RED DE COMUNICACION Y EL EQUIPO INSTALADO EN LA PLANTA.

Otros avances en el software, mejoraron los procedimientos de monitoreo del proceso, los menús de instrucciones simplificaron la operación de interfaces y periféricos con la facilidad que resulta elegir el número de la función que se desea ejecutar.

Con el desarrollo tecnológico durante la década de los setenta, el controlador lógico programable dio el primer paso para sustituir el uso de la minicomputadora en muchas aplicaciones industriales.

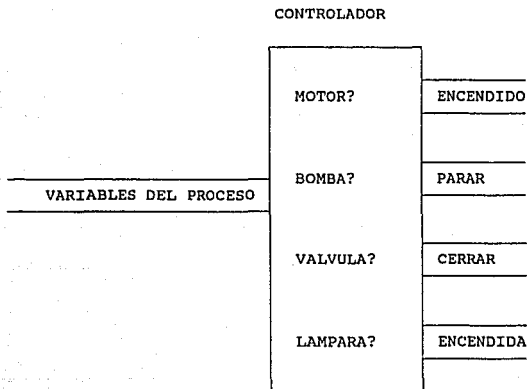


FIGURA 1.3: LAS VARIABLES DEL PROCESO, CONTIENEN LA INFORMACION NECESARIA PARA QUE EL CONTROLADOR "DECIDA" LAS CONDICIONES EN QUE DEBEN ESTAR LOS DISPOSITIVOS EN EL SIGUIENTE PASO.

EL PLC EN LA ACTUALIDAD.

El paso de la última década trajo un gran desarrollo técnico en la industria del PLC que continúa actualmente. Los cambios no sólo han afectado su diseño, sino que se han aproximado incluso a la filosofía de diseño de los sistemas de control. A continuación se enumeran algunos de los cambios sufridos por el PLC en los últimos años.

En Hardware:

- * Optimización de tiempo en los procesos, gracias al desarrollo del microprocesador.
- * Reducción de costos y espacio requeridos, pudiendo reemplazar de 4 a 10 cableados de relevadores, dependiendo del dispositivo.
- * La "inteligencia" del microprocesador, que coordina el proceso auxiliándose de las interfases de entrada/salida, siendo las más comunes las de Control Proporcional, Integral y Derivativo (PID); Comunicación ASCII, computador administrador y módulo de lenguajes.
- * Interfases especiales que permiten conectar directamente al controlador ciertos dispositivos. Interfases típicas que incluyen termoacopladores, calibrador de esfuerzos y mejor respuesta a las entradas.
- * Diseño mecánico que integra en una unidad terminal el sistema de entrada/salida, mejorando incluso el ciclo de intercambio de información entre el sistema y su controlador.
- * Equipo periférico que facilita la operación de interfases y mejora los métodos de manejo de información.

Los significativos adelantos de hardware provocaron el desarrollo de la familia de los controladores programables. Esta familia consistió en una línea de productos que alcanzaron la denominación de microcontroladores, debido a su tamaño.

Mediante el uso del multiplexaje de señales, su capacidad puede extenderse desde 10 puntos hasta 8000 puntos de entrada/salida, dependiendo del equipo utilizado; algunos grandes y sofisticados sistemas de PLCs pueden llegar a tener 128,000 o más palabras de memoria.

La familia de controladores lógico programable utilizan generalmente, sistemas de entrada/salida y programación de periféricos que normalmente pueden ser enlazados en una red local de comunicación.

El concepto que conserva la familia de PLCs, es el de mantener en las modificaciones a su diseño, la importancia de disminuir costos, simplificando su uso y mantenimiento.

En Software:

- * Lenguajes de alto nivel, que se han venido instaurando en algunos módulos del controlador proporcionando gran flexibilidad de programación en la comunicación de dispositivos periféricos.
- * Funcionales avances en bloques de instrucciones que el software adelantado ofrece al PLC para manejar el Diagrama de contactos, utilizando comandos sencillos de programación.
- * Diagnóstico y detección de fallas, tales como mal funcionamiento de algunos controles, fallas en máquinas o procesos de control.
- * Utilización del punto flotante que hace posible la ejecución de cálculos complejos requeridos en tareas de control como son calibración, balanceo, estadísticas de control, etc.
- * Instrucciones que facilitan el manejo de datos pudiendo enlazar los procedimientos de control con los datos que involucran toda la información almacenada y la generada por el proceso, con lo que el controlador programable "decide" la acción consecuente.

El controlador lógico programable es hoy un reconocido sistema de control que ofrece mucho más otros de sus precursores. El PLC entre sus ventajas cuenta con la capacidad de comunicarse con otros sistemas de control, proporcionando reportes de producción, diagnóstico de fallas propias, del proceso o maquinaria.

Estos adelantos hacen de los controladores lógico programables una importante aportación de la ciencia moderna que contribuye a la actual demanda de mayor calidad en la productividad, conservando los objetivos primarios de su concepción: sencillez de uso y mantenimiento.

PANORAMA DE LOS CONTROLADORES LOGICO PROGRAMABLES.

El futuro del controlador lógico programable no depende sólo del desarrollo de nuevos productos, sino también de la relación del PLC con otro tipo de controladores y fabricantes de administradores electrónicos. Sin lugar a dudas el resultado de esta relación determinará el sustancial papel que el PLC juega en el desarrollo del control automático de la industria del futuro.

El controlador lógico programable sigue incorporándose rápidamente a sistemas como el de Manufactura Integrada por Computadora (Computer Integrated Manufacturing: CIM), combinando su poder y capacidad de resolución con controles numéricos, robots, sistemas CAD/CAM, sistemas administradores de información y sistemas de computo que establecen la jerarquía de sistemas.

Los nuevos avances en la tecnología del PLC incluirán mejor operación de interfaces, orientación del diseño de interfaces con el concepto de hombre-máquina, base del desarrollo de otras interfaces que faciliten la comunicación con otros equipos; en general adelantos en hardware y software que den soporte al desarrollo de sistemas de inteligencia artificial.

El desarrollo de software tiende a proporcionar mejores interconexiones entre los diferentes tipos de equipos utilizando un medio de comunicación estándar como la Manufactura de Protocolo Automático (Manufacturing Automation Protocol: MAP). El desarrollo de nuevas instrucciones de PLC buscarán agregar inteligencia a los sistemas basada en la introducción de instrucciones que permitan "reconocer" o "aprender" las diferentes etapas del proceso.

La filosofía del control del futuro será determinada por la utilización del concepto de "sistemas de manufactura flexible" (Flexible Manufacturing System: FMS). Es prácticamente un hecho que el desarrollo actual, hará del controlador programable un importante jugador en la industria del mañana

Las estrategias de control serán divididas con "inteligencia" en lugar de centralizarse. Los super PLCs que se están concibiendo hoy, serán usados en aplicaciones que requieran de cálculos complejos, redes de comunicación y supervisión de otros PLCs más pequeños y máquinas controladas.

CAPITULO II

FUNCIONES LOGICAS ELEMENTALES.

Un requisito fundamental para la comprensión de los controles programables y sus aplicaciones, es el manejo de los conceptos de las operaciones lógicas básicas: AND, OR y NOT cuya combinación simplifica la toma de decisiones en las diferentes etapas del proceso.

LOGICA BINARIA Y COMPUERTAS LOGICAS ELEMENTALES.

El concepto de lógica binaria, se basa en los valores de verdad que puede tener una proposición simple (condición) lo que nos lleva a un estado "1" cuando es verdadera (se da la condición) o a un estado "0" cuando es falsa (no se da la condición) estos valores lógicos los podemos identificar en una lámpara "encendida" o "apagada", un interruptor "abierto" o "cerrado".

En sistemas digitales los estados lógicos pueden ser tomados como una señal PRESENTE o NO PRESENTE, ACTIVADA o NO ACTIVADA, ALTA o BAJA, etc. La toma de decisiones en forma electrónica se basa en la concepción de estos dos estados, adaptándose al sistema de números binarios en el cual se basa toda la lógica de control de las computadoras digitales y el controlador lógico programable.

Para el PLC el estado "1" representa la presencia de una señal o la ocurrencia de algún evento, mientras que el estado "0" representa la ausencia de la señal o la no ocurrencia del evento.

En los sistemas digitales estos dos estados son en realidad representados por dos niveles de voltaje distintos, de los cuales un voltaje es más positivo (a una referencia más alta) que otro, por tanto un "1" lógico será reconocido como verdadero, encendido, alto, etc., mientras que un "0" lógico representará términos como falso, apagado, bajo, etc. [tabla 2.1].

DISPOSITIVO	1 (+ V)	0 (0 V)
Interruptor	Cerrado	Abierto
Campana	Repica	No repica
Bocina	Suena	No suena
Válvula	Cerrada	Abierta
Motor	Encendido	Parado
Alarma	activada	Desactivada

TABLA 2.1: EJEMPLO DE LOS ESTADOS LOGICOS DE ALGUNOS DISPOSITIVOS UTILIZANDO LOGICA POSITIVA.

Nombre	Forma diferenciada	Forma Rectangular	Ecuación Algebraica	Tabla de Verdad															
AND			$F=XY$	<table border="1"> <thead> <tr><th>X</th><th>Y</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	X	Y	F	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	F																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR			$F=X+Y$	<table border="1"> <thead> <tr><th>X</th><th>Y</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	X	Y	F	0	0	0	0	1	1	1	0	1	1	1	1
X	Y	F																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
Inversor			$F=\bar{X}$	<table border="1"> <thead> <tr><th>X</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	X	F	0	1	1	0									
X	F																		
0	1																		
1	0																		
Buffer			$F=X$	<table border="1"> <thead> <tr><th>X</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	X	F	0	0	1	1									
X	F																		
0	0																		
1	1																		
NAND			$F=\overline{X \cdot Y}$	<table border="1"> <thead> <tr><th>X</th><th>Y</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X	Y	F	0	0	1	0	1	1	1	0	1	1	1	0
X	Y	F																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
NOR			$F=\overline{X+Y}$	<table border="1"> <thead> <tr><th>X</th><th>Y</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X	Y	F	0	0	1	0	1	0	1	0	0	1	1	0
X	Y	F																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
OR excluyente (XOR)			$F=XY+\bar{X}\bar{Y}$ $=X \oplus Y$	<table border="1"> <thead> <tr><th>X</th><th>Y</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X	Y	F	0	0	0	0	1	1	1	0	1	1	1	0
X	Y	F																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
NOR excluyente (XNOR)			$F=XY+\bar{X}\bar{Y}$ $=X \oplus Y$	<table border="1"> <thead> <tr><th>X</th><th>Y</th><th>F</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	X	Y	F	0	0	1	0	1	0	1	0	0	1	1	1
X	Y	F																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

FIGURA 2.1.- COMPUERTAS LOGICAS.

Quando la ocurrencia del evento o el nivel de voltaje más alto se representa por un "1" lógico, se trabaja con una lógica denominada Lógica Positiva (Tabla 2.1). Por otro lado si la ocurrencia del evento o el nivel de voltaje más alto se representa por un "0" lógico, se trabaja con una lógica denominada Lógica Negativa [Tabla 2.2].

DISPOSITIVO	1 (+ V)	0 (0 V)
Interruptor	Abierto	Cerrado
Campana	No repica	Repica
Bocina	No suena	Suena
Válvula	Abierta	Cerrada
Motor	Parado	Encendido
Alarma	Desactivada	Activada

TABLA 2.2: EJEMPLO DE LOS ESTADOS LOGICOS DE ALGUNOS DISPOSITIVOS UTILIZANDO LOGICA NEGATIVA.

Las variables binarias podemos concebirlas como cantidades o variables físicas que pueden ser representadas por un dígito binario o nivel lógico ("1" o "0"). Ahora bien, si se combinan dos o más de estas variables binarias se puede formar una condición que si se cumple o no, puede ser representada por "1" o "0", dependiendo de la lógica a utilizar.

El controlador lógico programable se basa en las condiciones de las variables del proceso para determinar las acciones consecuentes en la secuencia del mismo.

Las operaciones lógicas ejecutadas en equipos digitales como los PLCs se basan en las funciones elementales AND, OR y NOT. Cada función lógica tiene una regla que determina una respuesta o salida a una o más condiciones de entrada, pudiendo ser representada gráficamente por un símbolo que representa la operación [Figura 2.1].

LA FUNCION AND.

El número de entradas de la compuerta AND es limitada únicamente por el número de condiciones o variables que involucra una etapa o paso del proceso, mientras que tiene una salida única. La salida de la función AND será verdadera únicamente si todas sus entradas son verdaderas.

Un ejemplo representativo de una aplicación de la función AND es el siguiente.

Ejemplo:

Una lámpara enciende únicamente cuando el interruptor PB1 y el interruptor PB2 están cerrados simultáneamente (se encuentran en un nivel lógico alto: "1").

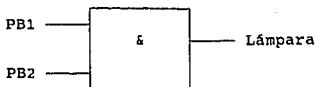
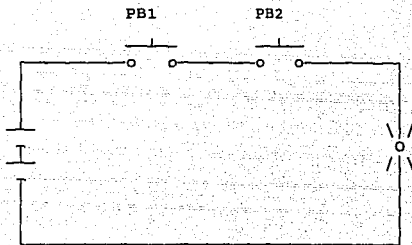


FIGURA 2.2: REPRESENTACION COMO UNA FUNCION AND.

PB1	PB2	LAMPARA
ABIERTO	ABIERTO	APAGADA
ABIERTO	CERRADO	APAGADA
CERRADO	ABIERTO	APAGADA
CERRADO	CERRADO	ENCENDIDA

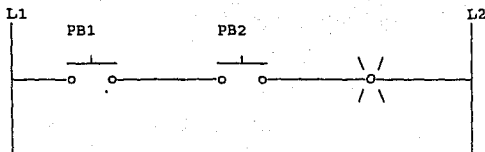
TABLA 2.3: TABLA DE VERDAD CORRESPONDIENTE A LA FUNCION AND.



(a)

Línea de voltaje

Común



(b)

FIGURA 2.3: (A) REPRESENTACION ELECTRONICA.
(B) DIAGRAMA ELECTRICO.

LA FUNCION OR.

El número de entradas de la compuerta OR es limitada únicamente por el número de condiciones o variables que involucra una etapa o paso del proceso, mientras que tiene una salida única. La salida de la función OR será verdadera si una o más de sus entradas son verdaderas.

Un ejemplo representativo de una aplicación de la función OR es el siguiente.

Ejemplo:

Una lámpara enciende si de dos interruptores PB1 y PB2 se cierran (se ponen en "1") cualquiera de los dos o ambos.

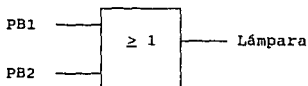
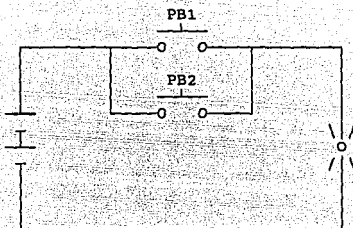


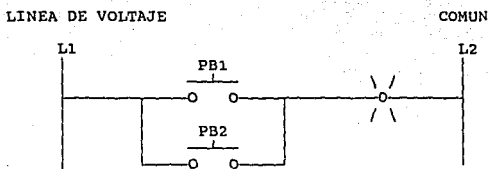
FIGURA 2.4: REPRESENTACION COMO FUNCION OR.

PB1	PB2	LAMPARA
ABIERTO	ABIERTO	APAGADA
ABIERTO	CERRADO	ENCENDIDA
CERRADO	ABIERTO	ENCENDIDA
CERRADO	CERRADO	ENCENDIDA

TABLA 2.4: TABLA DE VERDAD CORRESPONDIENTE A LA FUNCION OR.



(A)



(B)

FIGURA 2.5: (A) REPRESENTACION ELECTRONICA. (B) DIAGRAMA ELECTRICO.

LA FUNCION NOT.

La función NOT puede tener sólo una entrada y sólo una salida, se caracteriza como inversor por mostrar a su salida el complemento binario de su entrada ($A = A'$).

La barra sobre la variable equivale al símbolo de complemento del algebra de conjuntos, por lo que si estamos hablando de un sistema binario, una variable con estado lógico "1", tendrá un complemento "0" y una variable con estado lógico "0", tendrá un complemento "1".

Normalmente la función NOT se utiliza en conjunto con las funciones OR y AND, a primera vista no es fácil localizarla dado que generalmente es identificada en los diagramas como un pequeño círculo a la entrada o salida que se quiere invertir [Figura 2.6].

Una función NOT es útil cuando con un "0" lógico (nivel bajo) se desea encender o activar un dispositivo y cuando con un "1" lógico (nivel alto) se desea apagar o desactivar un dispositivo.

SE INVIERTE LA
ENTRADA A.

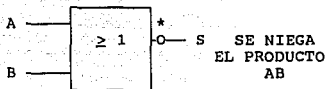
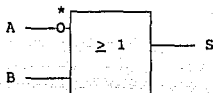


FIGURA 2.6: LA FUNCION NOT SE REPRESENTA MEDIANTE UN PEQUEÑO CIRCULO (*) A LA ENTRADA O SALIDA DE UNA TERMINAL.

Los siguientes ejemplos muestran aplicaciones de la función NOT.

Ejemplo:

La válvula de selenoide V1 [Figura 2-7] abrirá (se pondrá en "1"), si el interruptor selector esta cerrado (en "1") y el interruptor de nivel L1 esta abierto (el líquido no alcanza el nivel).

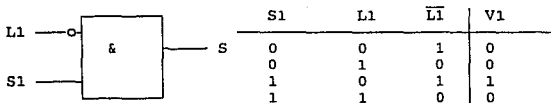


FIGURA 2.7: LA FUNCION LOGICA REPRESENTATIVA Y SU TABLA DE VERDAD.

El interruptor de nivel L1 es normalmente abierto, cerrándose cuando el líquido alcanza un nivel determinado. El circuito eléctrico necesita un relevador CR1, para implementar la función NOT (NO-normalmente abierto-).

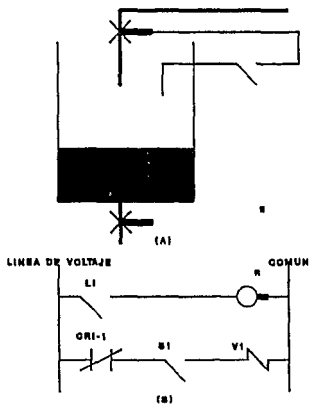
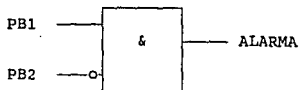


FIGURA 2.8: (A) ESQUEMA DEL TANQUE.
(B) DIAGRAMA ELECTRICO.

La función NOT puede utilizarse para negar la salida de la función AND o la función OR formándose respectivamente las funciones NAND y NOR.

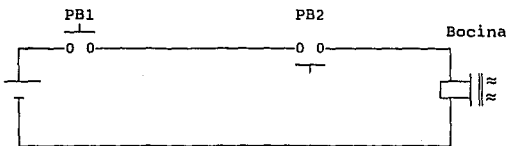
Ejemplo:

Una alarma sonará sí el interruptor PB1 está en "1" (oprimido) y el interruptor PB2* no está en "0" (no oprimido).

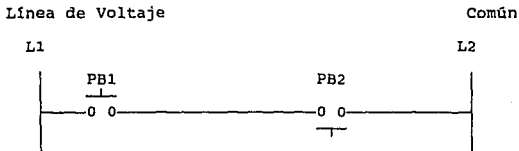


PB1	PB2	$\overline{PB2}$	ALARMA
NO OPRIM (0)	NO OPRIM (0)	1	NO SUENA
NO OPRIM (0)	NO OPRIM (1)	0	NO SUENA
OPRIMIDO (1)	OPRIMIDO (0)	1	SUENA
OPRIMIDO (1)	NO OPRIM (1)	0	NO SUENA

FIGURA 2.9: COMPUERTA LOGICA Y SU TABLA DE VERDAD.



(a)

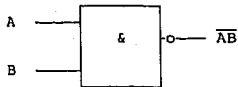


(b)

FIGURA 2.10: (A) ESQUEMA DEL TANQUE. (B) DIAGRAMA ELECTRICICO.

NOTA: Para este ejemplo la representación física de la función esta representada por un interruptor normalmente cerrado.

COMPUERTA NAND DE DOS ENTRADAS



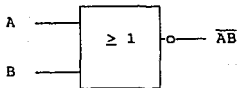
SIMBOLO

A	B	\overline{AB}
0	0	1
0	1	1
1	0	1
1	1	0

TABLA DE VERDAD

FIGURA 2.11: SIMBOLO Y TABLA DE VERDAD DE LA FUNCION NAND.

COMPUERTA NOR DE DOS ENTRADAS



SIMBOLO

A	B	$\overline{A+B}$
0	0	0
0	1	1
1	0	1
1	1	1

TABLA DE VERDAD

FIGURA 2.12: SIMBOLO Y TABLA DE VERDAD DE LA FUNCION NOR.

LOGICA Y ALGEBRA BOOLEANA COMO HERRAMIENTA DE DISEÑO.

La comprensión de los principios del algebra de Boole, facilita la simplificación de las complejas expresiones de los circuitos digitales, por lo que son una útil herramienta en la creación de programas de control basándose en los diagramas eléctricos convencionales y la lógica elemental.

El algebra de Boole fue desarrollada en 1849 por el inglés George Boole, con el fin de facilitar la lógica del razonamiento humano, objeto de estudio de la filosofía; con el tiempo sirvió para implementar la denominada lógica digital, que basa sus principios en los valores lógicos falso y verdadero.

En el desarrollo de la lógica digital, el algebra de Boole vino a ser un camino para el análisis y simplificación de los circuitos lógicos. Todos los sistemas digitales están basados en los valores lógicos verdadero/falso en donde verdadero toma un valor lógico de "1" y falso el valor lógico de "0".

Esta interrelación entre la lógica booleana y la lógica digital induce al uso indistinto del termino "compuerta lógica" o "compuerta booleana", una gran conexión de circuitos con estas compuertas se denomina Red Booleana, al igual que en el PLC el lenguaje se conoce como Lenguaje Booleano.

CIRCUITOS DEL PLC Y SIMBOLOGIA DE CONTACTOS LOGICOS.

El control clásico creaba las funciones lógicas de control (secuenciación, temporizadores y control) basándose únicamente en el cableado de relevadores lo cual dificultaba cualquier cambio.

En contraste en los controladores programables las funciones lógicas son fácilmente cambiadas, la instalación eléctrica es actualizada alterando únicamente parte de la trayectoria hacia los dispositivos.

Una primera tarea del controlador lógico programable es la de sustituir los sistemas basados en el cableado de relevadores montando las funciones correspondientes al nuevo sistema de control.

La figura 2.13 muestra el circuito eléctrico de un sistema por relevadores y el diagrama correspondiente para su montaje con PLC.

Las conexiones de la circuitería son determinadas mediante instrucciones de software. Estas conexiones pueden ser consideradas como parte de la CPU del controlador programable.

El sistema de relevadores se construye en el PLC con base en las funciones lógicas elementales AND, OR y NOT; las cuales pueden ser utilizadas independientes o combinadas entre sí, formando las instrucciones que controlan los dispositivos.

Las instrucciones así formadas engloban los comandos que se transmiten al PLC creando un lenguaje de comunicación con el controlador.

Existen varios lenguajes de programación para el PLC, tales como la representación mediante de funciones lógicas y el diagrama de escalera, este último es el más utilizado por su gran similitud con el diagrama eléctrico común.

El diagrama de escalera es también conocido como simbología de contactos, cuenta con instrucciones que representan gráficamente los contactos -normalmente abierto- y -normalmente cerrado- de los sistemas por cableado fijo.

El diagrama de escalera es una forma muy simple de representar el control lógico en una simbología que es utilizada en los esquemas de control actual, que permite observar fácilmente la información basada en los diagramas eléctricos.

Sí se utiliza el diagrama de escalera como lenguaje de programación, la traslación de un sistema con lógica por relevadores a un sistema por lógica programada se reduce a un sólo paso. Cuando se utiliza como lenguaje de programación la representación como funciones lógicas, la conversión a la simbología de contactos es innecesaria.

El diagrama de escalera de la figura 2.13 puede ser diseñado mediante circuitos individuales teniendo una salida única. Cada circuito es conocido como una etapa o red del sistema (rung), las etapas forman lazos de control por lo que son interdependientes para generar una salida en el PLC.

Algunos controladores permiten etapas con salidas múltiples, sin embargo lo convencional son las etapas con una sola salida.

El programa completo de un PLC consiste de etapas fijas que son controladas mediante interfases E/S conectadas a los dispositivos de campo.

Cada etapa o red es una combinación de las condiciones de entrada (símbolos), conectándose de izquierda a derecha entre dos líneas verticales, el símbolo que representa la salida es el último símbolo hacia la derecha.

Los símbolos que representan las entradas son conectadas en serie, paralelo o combinaciones de ambas con las que se obtienen la función lógica deseada; estos símbolos de entrada representan la entrada a los dispositivos conectadas a las interfases de entrada del PLC.

El concepto de programación por etapas "rung", es remanente de los sistemas por relevadores, en los cuales los dispositivos de entrada son conectados en serie o paralelo para controlar las diferentes salidas.

Cuando se activan estos dispositivos de entrada, permiten la alimentación de corriente al circuito o la suspensión de la misma encendiendo o apagando de esta manera algún dispositivo.

Los símbolos de entrada en las etapas seriadas pueden representar señales generadas por los dispositivos E/S [tabla 2-4] o por señales internas del controlador.

DISPOSITIVOS DE ENTRADA	DISPOSITIVOS DE SALIDA
Botón	Lámpara piloto
Interrupción selector	Válvula selenoide
Interrupción limitador	Bocina
Interrupción de proximidad	Relevador
Contacto temporizador	Temporizador

TABLA 2.4: ENTRADAS DE ENCENDIDO/APAGADO Y DISPOSITIVOS DE SALIDA.

CONCEPTO DE DIRECCION EN EL PLC.

Cada símbolo representativo de una etapa o paso, esta representado por un número de referencia, que indica la dirección de memoria en donde se almacena su estado actual.

Cuando las señales de un dispositivo son conectadas a las interfases I/O, la dirección representa la ubicación de la terminal o hilo donde se dirige o es originada la señal.

La dirección de una entrada o salida establecidas en el sistema pueden ser utilizadas en el programa tantas veces como sea requerido por la lógica del proceso.

Esta característica del PLC sin duda supera al sistema por cableado fijo en donde una etapa repetitiva generalmente tiene que ser agregada cada vez a la instalación del sistema.

La figura 2.13 muestra relevadores y la implementación equivalente con PLC. Cada dispositivo de campo es activado por los interruptores de presión PB1 y PB2, un interruptor de límite es conectado al módulo de entrada del PLC el cual tiene el número que identifica su posición o dirección en memoria.

La alimentación puede ser dirigida a través de los dispositivos 10 y 12 u 11 y 12 activando la salida identificada en la interfase con el número 15, la cual encenderá la lámpara PL1. La lógica que se aplica en un circuito electromecánico es la misma que se aplica en el circuito del controlador programable.

El orden en que el PLC suministra la alimentación al sistema en las direcciones 10, 11 o 12 es determinada por la conexión de los dispositivos a la interfase de entrada que identifica las etiquetas o números de referencia 10, 11 y 12 siendo activadas al oprimirse el interruptor de presión correspondiente o cerrando el interruptor de límite.

Los símbolos característicos de un lenguaje, generalmente mantienen una relación dependiente con el programador, el cual utiliza su número de referencia o dirección, identificándolo en sistema octal o decimal.

La forma como el controlador programable, almacena cada símbolo depende de las características particulares de cada equipo, sin embargo el procedimiento se realiza similarmente.

CAPITULO III

COMPONENTES PRINCIPALES

DE UN

CONTROLADOR LOGICO

PROGRAMABLE.

FUNCIONAMIENTO.

PRINCIPIOS DE OPERACION DEL PLC.

Un controlador programable esta formado de dos partes fundamentales: La unidad central de procesos y el sistema de interfases entrada/salida (I/O). [Figura 3.1].

La unidad central de proceso CPU, esta compuesta por un microprocesador, una unidad de memoria y una fuente de alimentación. [Figura 3.2].

LA OPERACION DE UN PLC SE RESUME DE LA SIGUIENTE FORMA:

El sistema entrada/salida es físicamente conectado a los dispositivos de campo englobados, en el proceso, tales como diversos tipos de detectores, válvulas, interruptores, transductores, arrancadores de motor, etc.

Las interfases E/S comunican al controlador lógico programable con el proceso controlado. Durante la operación, la unidad central de proceso manipula toda la información de los módulos E/S.

La CPU, lee o capta los datos de entrada o estados de los dispositivos por medio de la interfase de entrada, ejecutando el programa de control almacenado en la memoria del PLC y enviando la orden consecuente a los dispositivos de salida mediante la interfase de salida. Este proceso de intercambio de información es conocido como monitoreo.

El propósito del sistema E/S formado básicamente por interfases es direccionar las variables emanadas desde el controlador o desde el sistema por los diversos dispositivos de control (selectores analógicos, interruptores, detectores, etc.) conectados electricamente a la interfase correspondiente.

La fuente de alimentación, suministra el voltaje requerido por los diferentes componentes del PLC. Normalmente se conecta con una separación galvánica entre el controlador programable y la línea de alimentación general.

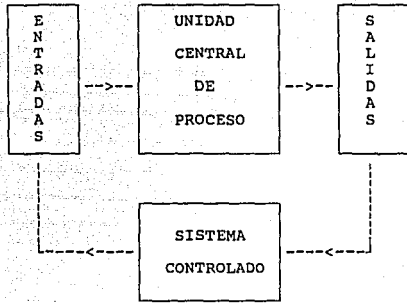


FIGURA 3.1: DIAGRAMA A BLOQUES DE UN PLC.

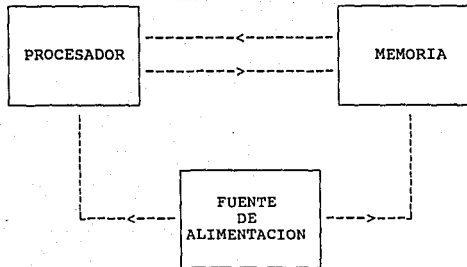


FIGURA 3.2: DIAGRAMA A BLOQUES DE LA CPU.

UNIDAD CENTRAL DE PROCESO.

En forma errónea se utilizan indistintamente los términos de CPU y microprocesador, como se mencionó antes, las partes de la CPU son el procesador, la unidad de memoria y la fuente de poder.

El microprocesador es aquel procesador que los avances de la tecnología VLSI logro encapsular en una pastilla o chip, por tanto podemos considerarlo como parte de la CPU.

El termino CPU engloba generalmente, a aquellos elementos que conforman la inteligencia del PLC, definiendo su interacción con el sistema controlado:

El microprocesador ejecuta el programa de control almacenado en memoria en un lenguaje estándar, mientras que la fuente de poder suministra aquellos niveles de voltaje requeridos por las operaciones propias del microprocesador y los dispositivos de memoria.

En la figura 3.3 se ilustra la interacción del controlador lógico programable con el sistema.

La interfase de entrada recibe las señales de campo mediante interruptores o detectores, enviándolas hacia el microprocesador de la CPU. En la memoria se almacena el programa de control en diagrama de escalera.

La CPU sigue la secuencia de los pasos programados de acuerdo a la información recibida y envía ordenes al proceso a través de la interfase de salida que normalmente esta conectada a arrancadores de motores, lámparas, variadores de temperatura o presión, etc. [Figura 3.3].

Hoy en día, la inteligencia asociada con los controles programables, se debe a las gigantescas capacidades de cálculo y control lógico de los microprocesadores.

Los microprocesadores desempeñan operaciones matemáticas, manejo de datos y rutinas de diagnóstico que no serían posibles con el control por cableado de relevadores.

La principal función del microprocesador es el mando y determinación de las actividades del sistema completo. Las actividades ejecutadas son tomadas de un programa sistemático conocido como ejecutivo. El programa ejecutivo es una colección de programas supervisores que están constantemente almacenando y verificando parte de la información del sistema y del mismo controlador.

Por acción del programa ejecutivo, el microprocesador puede desempeñar todas sus funciones de control, comunicación y otras funciones propias del procesador.

El programa ejecutivo posee software que establece la comunicación entre el PLC, el sistema y el usuario o programador mediante los dispositivos de programación. Otros tipos de comunicación en la periferia del sistema es soportada también por el ejecutivo.

Tales comunicaciones periféricas incluyen el monitoreo constante de las variables de control, sistema de entrada/salida, dispositivos de memoria, diagnóstico de la condición de la fuente de alimentación, etc.

La CPU de un PLC puede contener más de un microprocesador⁽³⁾ para ejecutar las tareas de comunicación y coordinación de mandatos. La razón de esta duplicidad de elementos procesadores es agilizar las operaciones del controlador aumentando la rapidez con que se van llevando a cabo.

El uso de los microprocesadores se ha extendido a las interfases, aumentando las capacidades del PLC. La "inteligencia" del PLC es proporcionada por estos arreglos de microprocesadores que no solo se ubican en la CPU del controlador, sino que son utilizados además en las interfases E/S convirtiendo estos módulos en tarjetas inteligentes con memoria propia, capaces de mantener por sí solas un ciclo de control, en caso de alguna falla o desconexión de la CPU.

Una tarjeta de este tipo es el módulo de control Proporcional-Integral-Derivativo (PID), el cual puede ejecutar un lazo cerrado de control en forma independiente de la CPU.

Los microprocesadores utilizados en los PLC son clasificados de acuerdo a el tamaño de palabra o número de bits que pueden utilizar para ejecutar una operación. El tamaño estándar de una palabra es de 4, 8, 16 y 32 bits.⁽⁴⁾

La longitud de la palabra, afecta la rapidez de respuesta del PLC a las variables del proceso, de tal forma que un microprocesador de 16 bits puede manejar dos palabras en el tiempo que un microprocesador de 8 bits maneja una palabra.

La diferencia de la longitud de palabra que maneja el microprocesador de un PLC es asociada a la capacidad y grado de sofisticación del mismo.

(3) Algunos PLC como el S 155 de Siemens manejan incluso más de una CPU con el fin de aumentar la rapidez de respuesta del controlador a las variables del proceso.

(4) El tamaño de palabra más utilizado es el 16 bits.

MONITOREO.

La función elemental de cualquier controlador es la de leer o "percibir" las señales o indicaciones del objeto de control y en base a estas generar una respuesta.

El controlador lógico programable recibe la información de un proceso a través de los dispositivos sensores que lo conectan a la interfase o tarjeta de entrada.

Esta información es procesada por la CPU en base al programa de control almacenado en memoria, enviando una señal eléctrica a la interfase o tarjeta de salida, encendiendo o apagando el dispositivo de campo indicado en el programa en ejecución.

Esta tarea es realizada por el controlador mediante señales internas que determinan el tiempo exacto de ejecución en que debe activarse o desactivarse un dispositivo, además de las condiciones que deben cumplir en un paso específico, aquellos dispositivos que generan variables en el proceso.

Cuando el procesador ha concluido la evaluación de los dispositivos en un paso determinado, corroborando que su estado presente sea el especificado en el programa, pasa al paso consecuente, enviando una señal a las interfaces de salida involucradas en el control de los dispositivos indicados en la etapa del proceso

Este proceso compuesto por la lectura de las entradas al controlador, la ejecución del programa almacenado en memoria y la activación de las salidas correspondientes, es conocido como monitoreo.

El monitoreo básicamente consiste en un examen secuencial que continuamente realiza el controlador de todos los dispositivos detectores (entradas), ejecutando las ordenes indicadas en el programa de control, enviando las señales eléctricas (salidas) que encienden o apagan los dispositivos involucrados en la secuencia.

La etapa de monitoreo la podemos representar básicamente en la figura 3.4 Este proceso es repetido continuamente por el PLC que recorre en secuencia las instrucciones contenidas en el programa; el tiempo de monitoreo, es el tiempo que tarda el controlador en recorrer el programa en cada paso.

Algunos fabricantes de PLCs llaman "monitoreo" únicamente al proceso de leer las entradas y activar las salidas adecuadas.

El monitoreo contempla también el proceso de ejecutar el programa que "decide" cuales son las salidas adecuadas conforme a los estados indicados en los sensores del controlador.

Es debido a la naturaleza del monitoreo que el tiempo en que se realiza depende de la longitud de las rutinas de programa que condicionan la continuación de la secuencia.

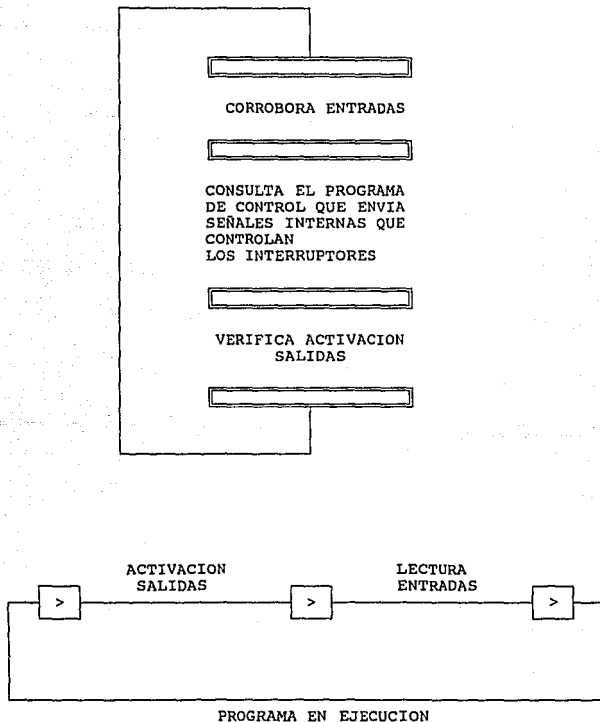


FIGURA 3.4.- REPRESENTACION GENERAL DEL MONITOREO EN EL PLC.

Dado que en cada rutina de programa se esta verificando la ejecución de instrucciones, consultando los indicadores de estado de los dispositivos incluidos en el paso; el tiempo de monitoreo es la suma del tiempo de respuesta del equipo utilizado en el sistema de control.

El tiempo en que se realiza la comunicación entre sistema y controlador, el tiempo de ejecución por instrucción, el tiempo que tarda en encender un motor, abrir una válvula, etc. La suma total será el tiempo que dure cada ciclo de monitoreo.

Es común que en algunos sistemas grandes que utilizan el control por PLC, se requiera calcular el tiempo exacto de monitoreo y la programación de una rutina de control que verifique que el tiempo de ejecución del programa de control sea en los rangos establecidos para cada etapa.

El sistema de comunicación del PLC determina el fin de monitoreo, con lo que en ese momento se actualizan en la CPU los estados en las diferentes entradas, de acuerdo a las lecturas obtenidas en los indicadores del proceso.

VERIFICACION DEL PROCESO.

El Procesador cuenta con métodos de detección de errores, los cuales incluyen rutinas que constantemente examinan las condiciones actuales del equipo, dispositivos indicadores, lazos de comunicación (entre proceso, subsistemas y controlador) e incluso de los módulos que conforman todo el controlador lógico programable.

Los métodos de detección de errores utilizados por los PLC comerciales, varían de acuerdo al fabricante. En forma general se utilizan principalmente dos técnicas: Paridad y Verificación Global (checksum).

Paridad.

El método de detección de error más utilizado es el de paridad, es utilizado principalmente en el enlace de comunicaciones localizando fallas en el tipo de información a lo largo de la línea de transmisión. El método de paridad, es también conocido como Examen Vertical Iterativo (VRC: Vertical Redundancy Check).

Un ejemplo de la aplicación de esta técnica es en la comunicación entre la CPU y los subsistemas, en la cual la paridad encuentra fallas cuando se verifica la comunicación durante la transmisión de datos.

Quando se aplica el método de paridad, la transmisión de datos es verificada por un número par o non de unos. Los datos deben ser enviados con un número determinado de unos o ceros.

El método de paridad consiste en agregar a cada palabra de la información un bit más, generalmente en la posición menos significativa; el cual nos indica la clase del número de unos que contiene la palabra que se esta trasmitiendo de la información en binario.

El error en el dato enviado es detectado cuando el número de unos, indica que el número par o non de los unos que contiene una palabra de la información en binario, no es el esperado; es decir sí se espera un número par de unos y el recibido es non, o sí se espera un número non de unos y el recibido es par.

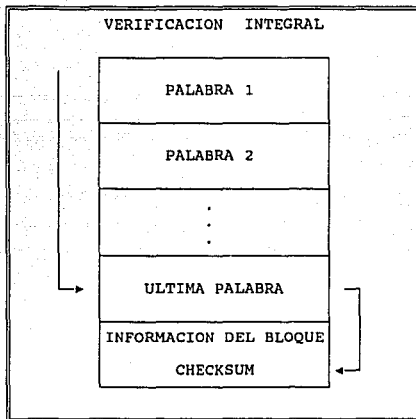
De lo anterior podemos decir, que el método de paridad tiene dos modalidades: paridad par, que consiste en asegurar que la palabra trasmitida contenga un número par de unos y paridad non que consiste en asegurar que la palabra trasmitida contenga un numero non de unos.

Quando un subsistema envía información al controlador, el tipo de paridad utilizada debe ser prefijada por el PLC, la paridad utilizada puede variar, sin embargo, regularmente en la red de comunicación con el proceso, el PLC utiliza el mismo tipo de paridad para todos los dispositivos.

VERIFICACION GLOBAL.

El método de paridad tiene la desventaja de desperdiciar aproximadamente un 12.5 % de la capacidad de memoria de datos, al tener que agregar un bit por palabra trasmitida, por este motivo se utiliza también un método de detección de error denominado Verificación Global (checksum) [figura 3.5].

La verificación global detecta los errores en un bloque de palabras a diferencia del análisis individual que se hace en el método de paridad. Con la técnica de verificación global, el análisis de la información se realiza evaluando y agregando a un bloque de palabras, una palabra que refleje las características del mismo. La palabra agregada se conoce como **Característica de Comprobación De Bloque (BCC: Block Check Character)**.



**FIGURA 3.5.-CARACTERISTICA DE COMPROBACION DE BLOQUE
Y FIN DEL BLOQUE DE DATOS.**

VERIFICACION REITERATIVA CICLICA.

(CRC: Cyclic Redundancy Check):

Técnica que consiste en realizar una suma de todas las palabras en el bloque de datos y el resultado total es almacenado en el último registro asignado al BCC.

Esta suma cíclica puede alcanzar un límite preestablecido. Una variación del CRC consiste en almacenar únicamente el exceso de esta suma. Comúnmente la palabra resultante de la aplicación de esta técnica se complementa antes de almacenarla.

Durante el ciclo de verificación por CRC todas las palabras del bloque se suman juntas, la suma que es la última palabra del BCC se convertirá a 0.

Existen algunas técnicas auxiliares en la realización de una verificación global como las siguientes.

Un bloque de palabras inválido puede ser detectado verificando únicamente en la suma un resultado final de 0. Otro tipo de CRC utiliza el exceso de la suma dividiendo el total entre un número binario fijo.

VERIFICACION LONGITUDINAL REITERATIVA.

(LRC: Longitudinal Redundancy Check).

Este procedimiento consiste en una serie de resultados obtenidos de una operación OR-exclusiva: la primera y segunda palabras del bloque se relacionan mediante una función OR-exclusiva, se realiza la misma operación con el resultado y la tercera palabra, se hace lo mismo con el segundo resultado y la cuarta palabra y así sucesivamente; el último resultado se almacena en la última localidad de la memoria asignada al BCC.

VERIFICACION CICLO-OR-EXCLUSIVA.

El procedimiento es similar al LRC, la verificación inicia con la operación OR-exclusiva a una primera palabra del verificador global conteniendo ceros y la primera palabra del bloque de información en análisis.

El seguimiento se hace alternando los bits a la izquierda de la palabra en análisis. El resultado es relacionado con la siguiente palabra del bloque con la función OR-exclusiva, siguiendo el mismo procedimiento hasta analizar todo el bloque de datos, agregándose el resultado final al BCC.

DIAGNOSTICO DEL SISTEMA.

El procesador es responsable de la detección y la comunicación de errores y fallas durante la operación del sistema. El procesador cuenta con rutinas capaces de alertar al operador de un posible mal funcionamiento en el equipo y algunos errores que no pueden autocorregirse.

En cada ciclo del programa el procesador del PLC monitorea las condiciones en que deben estar todos los indicadores para realizar la instrucción siguiente.

Estas condiciones comúnmente son llamadas en forma incorrecta el "status" del sistema, representando en realidad el estado lógico en que se encuentran los dispositivos en el paso que se está monitoreando.

La información es llevada desde el proceso hasta el PLC que generalmente cuenta con indicaciones luminosas en la tarjeta en que se esta procesando la señal de salida o entrada.

Los dispositivos del PLC que normalmente son examinados por el procesador son la memoria, la batería de la RAM, la fuente de poder y el mismo procesador; en algunos PLC más sofisticados el procesador detecta fallas en contactos, interruptores y bajo condiciones específicas del controlador incluso, temporizadores.

SISTEMA DE ALIMENTACION.

En el funcionamiento del sistema, uno de los factores de mayor importancia es el suministro de energía durante toda la operación de la secuencia del proceso, tanto para el controlador como para el equipo del proceso y dispositivos detectores que integran el sistema automatizado.

La tarea del sistema de alimentación es administrar el voltaje DC requerido por los componentes centrales del PLC como el procesador, dispositivos de memoria e interfases E/S.

Es responsabilidad del sistema de alimentación la regulación del voltaje administrado. La fuente de poder debe tener una buena y una protección contra posibles descargas que pudieran generarse en dispositivos que manejen tensiones mayores en el proceso.

Dentro del sistema de alimentación se cuenta con señales indicadoras que advierten al operador o usuario cualquier anomalía que afecte un suministro de energía apropiado para cada dispositivo, ya sea en forma directa mediante indicaciones luminosas o por medio de mensajes que el procesador edita cuando el monitoreo detecta alguna irregularidad.

ENTRADA DE VOLTAJE.

El suministro de energía a el PLC normalmente es a través de una línea de corriente alterna, algunos PLC tienen una entrada que permite sean alimentados en forma inmediata desde una fuente de corriente directa.

Este tipo de controladores programables son requeridos con mayor frecuencia en sistemas cuyo proceso requiere equipo que consume en su mayoría corriente directa. Los PLC más comunes son los de 120 Vac y 220 Vac, sin embargo algunos de estos controladores aceptan 24 Vdc.

Normalmente las industrias viven diariamente la experiencia de fluctuaciones en los rangos de voltaje y frecuencia en la línea de tensión, por este motivo es muy importante que el fabricante especifique claramente los rangos de trabajo para los que sus equipos trabajan en forma óptima.

Un buen equipo de PLC debe trabajar con un rango de tolerancia entre el 10 y 15 % en las variaciones de la línea. Algunos PLC cuentan con rutinas en su programa de control que hacen que el procesador inhabilite aquellos módulos que pueden ser dañados cuando las fluctuaciones exceden sus rangos de trabajo. Algunas plantas cuya producción puede ser afectada por las variaciones, tienen sistemas de regulación que estabilizan las condiciones de la línea de alimentación.

Aún cuando la fuente de energía es diseñada para tolerar las fluctuaciones de la tensión de línea, es conveniente considerar que algunas industrias por el tipo de equipo que utilizan en sus procesos, pueden provocar inestabilidad en la línea de alimentación; en estos casos es recomendable prevenir de alguna forma los efectos que se producirían en el controlador.

La compensación para las variaciones provocadas por el funcionamiento de la planta en la línea de voltaje, dependen de los niveles mínimo y máximo que pueden alcanzar, así como la localización del PLC dentro de la misma.

Algunos efectos causados por las fluctuaciones en la alimentación pueden ser los siguientes:

- * Arranque o Paro de equipo cercano como motores, bombas, compresores, acondicionadores de aire, etc.
- * Perdidas en la línea proveniente de una posible subestación.
- * Perdidas en la línea interna de la planta debido a conexiones de baja calidad.
- * Situaciones que provoquen la interrupción accidental o necesaria del suministro de energía, lo que reduce la actividad normal de la planta.

Una solución es la instalación de un transformador en donde se logran compensar las variaciones de voltaje con un circuito que entregue un nivel de tensión más estable.

Un transformador a voltaje fijo se obtiene alimentando la tensión inestable o con fluctuaciones en el primario obteniéndose de esta forma mayor estabilidad en el voltaje de salida del secundario.

Cuando se opera con una carga baja, el transformador puede mantener aproximadamente $\pm 1\%$ de regulación con fluctuaciones a la entrada de cuando mucho un 15 %. El porcentaje de regulación depende directamente de la carga que representa el consumo de todos los módulos del PLC [figura 3.6].

El rango de voltaje que se obtenga en la compensación mediante transformador, debe cubrir la amplitud de requerimientos del sistema en operación tanto a mínima como a máxima carga, considerando el límite de carga máxima en unidades de Volt-Amperes (VA) para la selección del transformador.

Es aconsejable que este tipo de compensación se lleve a cabo desde la fabricación del PLC, sin embargo, son muchas las industrias que su mismo proceso provocan fluctuaciones; razón por la cual cada empresa debe considerar una compensación externa de acuerdo a las especificaciones del fabricante del PLC que va a instalar y a las condiciones propias de su planta.

AISLAMIENTO POR TRANSFORMADOR.

Considerando que el PLC es un computador industrial, al igual que cualquier equipo de computo requiere estar aislado de ambientes que generen interferencias electromagnéticas (EMI), es común que el fabricante recomiende se instale el controlador en gabinetes que los aislen de cualquier interferencia que pudiera alterar el funcionamiento correcto de la circuitería.

Por seguridad, es conveniente agregar al PLC un circuito que lo separe eléctricamente de la fuente de corriente alterna que alimenta a los equipos que generan interferencias electromagnéticas.

Un tipo de aislamiento que asegura el buen funcionamiento del controlador programable, es conectar un transformador entre el controlador y la fuente de alterna.

Se debe tomar en cuenta que este aislamiento por transformador no cubre necesariamente la compensación por transformador que requiere un PLC para evitar las fluctuaciones provenientes de la línea.

M E M O R I A

Una característica primordial en el PLC es la velocidad y grado de sencillez que tenga para intercambiar las señales E/S con el proceso.

La unidad de memoria es el lugar en donde se almacenan todas la instrucciones y secuencias del programa de control que deben ser ejecutadas por el procesador.

Existen diferentes tipos de dispositivos de memoria, que utiliza el PLC. El programa de control se graba regularmente en aquellos tipos de memoria que permiten modificación en el contenido de sus registros. Es decir, pueden ser reprogramados en caso necesario de correcciones o cambios.

La función que tiene el almacenamiento y extracción de información del módulo de memoria del controlador programable esta íntimamente relacionada con las consignas que seguirá el mismo en su interacción con el proceso. Sistema de Memoria Integral.

La memoria del PLC, básicamente es dividida en dos memorias virtuales, la primera almacena al Programa Ejecutivo y la segunda lo que se denomina Area de Aplicación [Figura 3.7].

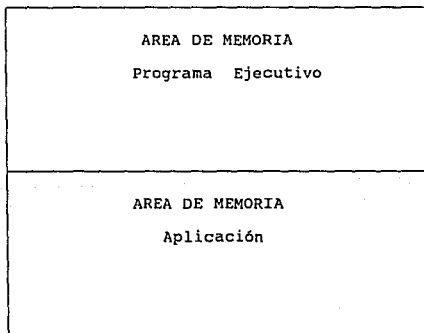


FIGURA 3.7: DIAGRAMA SIMPLIFICADO DEL SISTEMA DE MEMORIA INTEGRAL DE UN CONTROLADOR PROGRAMABLE.

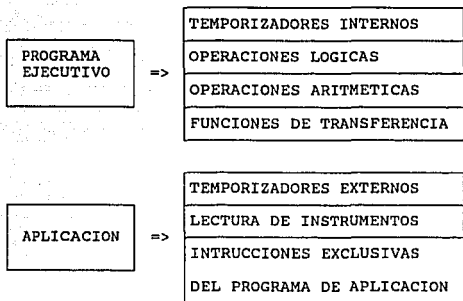


FIGURA 3.8: FUNCIONES CARACTERISTICAS DE QUE SE REALIZAN EN LAS DIVISIONES DE LA MEMORIA VIRTUAL

El Ejecutivo, esta compuesto de una colección de programas que son considerados parte del PLC, por lo que son almacenados en forma permanente sin que se pueda actuar sobre ellos.

Como se menciona antes el programa ejecutivo realiza una función de supervisión en todo el sistema de control, influyendo directamente en todas las actividades del mismo.

El área del ejecutivo es la parte de la memoria en donde se encuentran todos los registros que contienen las instrucciones o software que definen el comportamiento del controlador, como son los temporizadores, bloques de transferencia, instrucciones de cálculo, etc. La influencia del software que alberga esta sección se observa desde el momento en que se energiza el PLC.

El área designada como área de aplicación, es la parte de la memoria que el PLC, reserva para almacenar las instrucciones particulares del programa introducido por el usuario, en esta parte se guarda también la información que genera el proceso y que es determinante en las decisiones tomadas por el controlador.

TIPOS DE MEMORIA.

El tipo de memoria adecuada para almacenar el programa ejecutivo es aquella en la que el usuario no pueda modificar su contenido, esto es para proteger la información contenida en los registros la cual debe ser permanente, dado que es la que rige el comportamiento del PLC independientemente del programa de aplicación.

En cambio la memoria que se requiere para almacenar toda la información que genera una aplicación debe de almacenar el programa de control del proceso en particular con la flexibilidad de hacer modificaciones en el mismo para incrementar o cambiar el sistema controlado. Además permitir la lectura y escritura de las variables del proceso.

Técnicamente sólo se pueden manejar dos tipos de memorias volátiles y no volátiles, las no volátiles son las que en ausencia de tensión mantienen la información que almacenan, mientras que las volátiles pierden la información que contienen cuando carecen de alimentación.

Las características que cubren las necesidades del programa ejecutivo son las memorias no volátiles, en cambio el área en que se guarda la información general de la aplicación, requiere de ambos tipos de memoria.

En muchas ocasiones la memoria volátil es suficiente para que el controlador programable trabaje, sin embargo tiene la desventaja de tener que cargar el programa de control cada vez que se arranca el sistema.

Las memorias no volátiles se conocen también como memorias de sólo lectura (ROM: Read Only Memory), existiendo varios tipos de estas. Las memorias volátiles, generalmente forman parte de las memorias de acceso aleatorio o de lectura/escritura (RAM: Random Access Memory), se les da este nombre por permitir el acceso a cualquier dirección para leer o escribir, sin tener que recorrer los demás registros.

MEMORIA DE SOLO LECTURA, (ROM).

El diseño de una memoria ROM, utilizada en la arquitectura de un PLC, es concebido de forma que en circunstancias normales su contenido no pueda ser alterado y el programa interno que contiene permanezca fijo.

Este tipo de memoria se designa Memoria de Solo Lectura (ROM: Read Only Memory) debido a la característica de sólo permitir el examen y lectura de su contenido.

Una vez almacenado el programa, los registros de la memoria no pueden ser alterados; es debido a esta ventaja que el programa ejecutivo es grabado en una memoria ROM.

CARACTERISTICA	ROM	PROM	EPROM	SRAM
1.-SE REQUIERE DE UNA MARCAJE ESPECIAL PARA LA REPROGRAMACION	SI	NO	NO	NO
2.-ES NECESARIO EQUIPO ESPECIAL PARA LA PROGRAMACION DEL USUARIO?	NO	SI	SI	SI
3.-SE PUEDE REPROGRAMAR?	NO	SOLO UNA VEZ	SI	SI
4.-SE DEBE RETIRAR EL CIRCUITO PARA LA REPROGRAMACION	N/A	SI	SI	NO
5.-ES POSIBLE UNA REPROGRAMACION SELECTIVA?	N/A	NO	NO	NO
6.-TIENE CAMBIOS RÁPIDOS	SI	SI	SI	SI
7.-ES NO VOLÁTIL	SI	SI	SI	SI

TABLA 3.1: RESUMEN DE CARACTERISTICAS DE LAS MEMORIAS ROM SEMICONDUCTORAS.

El empleo de memorias ROM para el área de aplicación no es recomendable, puesto que la flexibilidad del PLC pierde una ventaja importante: la capacidad de reutilización. Sin, embargo, en algunos procesos que manejan muchos parámetros fijos, el uso de la ROM puede agregar rapidez al controlador.

MEMORIAS DE LECTURA ESCRITURA (RAM).

Las memorias de lectura/escritura, también llamadas memorias de acceso aleatorio (RAM: Random Access Memory) tienen en su diseño la ventaja de poder acceder directamente cualquier localidad de memoria para leer o grabar un dato.

Existen dos tipos de memorias RAM. Las RAM volátiles, pierden la información cuando dejan de ser alimentadas y normalmente requieren de una batería de respaldo. Las RAM no volátiles (NOVRAM), conservan su contenido si la energía es suspendida.

Actualmente la mayoría de los controladores programables utilizan bancos de memorias RAM, que dan soporte a toda la información que se cambia continuamente en un programa y en muchos casos almacenan el programa de aplicación, la desventaja del uso de una batería para protección de la información conlleva muchas veces a combinar el uso de la RAM con otros tipos de memoria.

MEMORIA DE SOLO LECTURA PROGRAMABLE (PROM).

La Memoria de sólo Lectura Programable (PROM: Programmable Read Only Memory) es una variedad de la memoria ROM que difiere en su construcción.

La ROM es grabada desde su manufactura, con un programa o tabla de verdad que se proporciona al fabricante (Figura 3.9), mientras la PROM puede ser grabada por el usuario con las desventajas de requerir un equipo especial y de poderse grabar una sola vez, por lo que si se requiere un nuevo programa es necesario sustituir el circuito integrado.

Actualmente el porcentaje en que utiliza un controlador programable memorias PROM es menor de 1%. Las memorias PROM son utilizadas para almacenar programas de aplicación que no necesitan cambios, en muchas ocasiones almacenan el programa con que se carga la RAM de un PLC en donde la memoria reservada para aplicaciones esta constituida por memorias volátiles, por lo que debe cargarse el programa cuando se arranca el sistema de control.

MEMORIA BORRABLE DE SOLO LECTURA (EPROM).

Este tipo de Memoria es conocido como Memoria PROM Borrable o EPROM (Erase Programmable Read Only Memory), tiene la capacidad de poder borrarse con luz ultravioleta y después ser reprogramada indefinidas veces, siempre que se respete el tiempo que el fabricante indica para borrar su contenido.

La EPROM es considerada una memoria semi-permanente debido a que en condiciones normales su información no puede ser alterada, si se requiere algún cambio en el contenido, la memoria debe ser borrada totalmente antes de efectuar cualquier grabación.

La memoria reservada en el PLC para aplicaciones, comúnmente utiliza una EPROM para guardar el programa diseñado para el control particular del proceso.

Las características de la EPROM no permiten la lectura y escritura de las variables del proceso, algunas veces aleatorias; debido a esto es que se utiliza en combinación con una memoria de lectura/escritura o memoria RAM.

Las Industrias en las cuales sus productos no requieren cambios, regularmente tienen procesos en los que las variables de control son prácticamente nulas, es por ello que la memoria EPROM puede constituir toda la sección que el controlador programable reserva para la aplicación.

ROM ELECTRICAMENTE ALTERABLE (EAROM).

La memoria EAROM (Electrically Alterable Read Only Memory) es una variedad de memoria ROM parecida a la EPROM con algunas diferencias.

La EAPROM no requiere de rayos ultravioleta para ser borrada, pues su reprogramación por ser eléctrica es prácticamente instantánea (del orden de milisegundos).

La memoria EPROM por necesitar de una fuente de rayos ultravioleta, para ser borrada debe retirarse del circuito impreso, mientras que la memoria EAROM puede borrarse en el mismo lugar donde esta colocada. Es debido a esta característica que la memoria EPROM es más solicitada por la industria, para protección de sus programas de control.

MEMORIA ELECTRICAMENTE BORRABLE (EEPROM).

Las memorias EEPROM, fueron desarrolladas a mediados de los 70s, con las bondades de las memorias EPROM y la flexibilidad de programación que tienen las memorias RAM.

Hoy en día pequeños y medianos controladores utilizan la memoria EEPROM como único medio de almacenar información debido a las ventajas que ofrece.

Entre las ventajas que la EEPROM agregó al PLC se encuentra la reducción de los retardos asociados a los cambios en la programación.

Una desventaja de la EEPROM es que para poder escribir un byte de información debe haberse borrado antes el byte que ocupa la dirección en que va a grabarse. debido a esto el procedimiento que se aplica en el empleo de la EEPROM es el borrado/escritura proceso que dura aproximadamente entre 10 y 15 ms.

MEMORIA MAGNETICA, CORE.

Estas memorias basan su construcción en un pequeño toroide magnético cuyo arreglo permite almacenar un "uno" o "cero" lógicos, son consideradas no volátiles por mantener almacenados sus datos aun cuando la energía es suspendida.

Los estados lógicos de la memoria Core pueden ser alterados electricamente, por lo que se considera una RAM no volátil.

Este tipo de memoria fue muy utilizada en los primeros controladores programables, siendo hoy pocos los que la siguen utilizando debido a tiene una baja rapidez de respuesta y un costo relativamente alto, además de requerir un gran espacio debido a su estructura magnética.

RAM NO VOLATIL (NOVRAM)

Las memorias NOVRAM y EEPROM son las más recientes en la industria de los computadores, la NOVRAM por su características de no volátil y fácil reprogramación cumple con los requerimientos del PLC para la memoria reservada para aplicación y su uso viene extendiéndose en los controladores programables modernos.

ESTRUCTURA BASICA DE LA MEMORIA.

La unidad básica de memoria es una celda que almacena un 1 lógico o 0 lógico, el 1 es reconocido por la CPU cuando la celda tiene un nivel de voltaje alto, y el 0 cuando tiene un voltaje bajo o nulo.

El 1 lógico representa una condición presente, encendido, de arranque, de activación, etc. El 0 lógico en cambio una condición no presente, de apagado, de paro, etc.

Una celda de almacenamiento indica la afirmación o negación de la situación que tienen o deben tener los dispositivos, sensores, indicadores, etc., en un ciclo de monitoreo.

Cada bit representa la unidad más pequeña de información, los bits son agrupados en grupos de 4, 8, 16 y 32. Los términos más comúnmente utilizados para su identificación en la memoria son los de Byte y Palabra.

Un Byte es la agrupación de 8 bits, la Palabra puede agrupar 4, 8, 16, y 32 bits. La unidad más pequeña de información es conocido como Nibble y agrupa 4 bits [Figura 3.10].

El tamaño de la palabra es determinado por la cantidad de bits que contiene, comúnmente una palabra agrupa cantidades de bits en múltiplos de 8.

Los bits que almacena una palabra pueden representar un número, una instrucción o cualquier información codificada en binario para ser interpretada por la CPU del controlador programable.

La palabra almacenada en memoria es localizada mediante un número denominado dirección. Las direcciones van de $2^n - 1$, donde n es el número de líneas de dirección [Figura 3.9].

DIRECCION	CONTENIDO
0 3 0 4	2 7 8 A
0 3 0 5	1 4 B 4
0 3 0 6	C F C 6
0 3 0 7	8 9 A C
0 3 0 8	8 D 8 0
0 3 0 9	1 B 7 B
0 3 0 A	D F 3 1
0 3 0 B	E 3 F B
.	.
.	.
.	.

FIGURA 3.9: EJEMPLO DE UN LISTADO SUMINISTRADO POR UN USUARIO AL FABRICANTE DE CI ROM PROGRAMADA POR MASCARA.

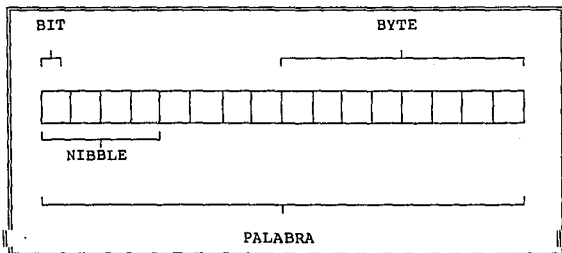


FIGURA 3.10: ESTRUCTURA BASICA DE LA MEMORIA DEL PLC.

Generalmente la capacidad de la memoria la determina la cantidad de unidades de palabra que puede almacenar aunque puede determinarse por unidades de información ya sean bits, bytes o palabras que son los términos más usuales.

Una memoria de 1K puede almacenar 1024 palabras, una de 2K tiene capacidad para 2048 palabras. El prefijo K= 2^{10} y puede sustituirse también por los prefijos M= 2^{20} y G= 2^{30} , cuando se manejan memorias muy grandes [Figura 3.11].

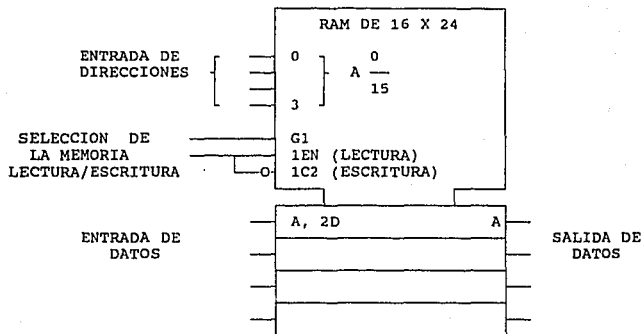


FIGURA 3.11: SIMBOLO GRÁFICO ESTANDAR DE UNA RAM DE 16 X 4.

El tamaño de la memoria que necesita un PLC es seleccionado de acuerdo al número de palabras que requiere para almacenar el programa de aplicación y total de variables generadas por el proceso.

En la tabla 3.2 se muestra un ejemplo de algunas instrucciones y la cantidad de palabras que implica su almacenamiento, la cantidad de palabras necesarias por instrucción es variable de acuerdo a la marca del equipo y a la sofisticación del mismo.

El área que utiliza cada instrucción del PLC se denomina **Área de Utilización** o **Área de Ocupación**. Cada instrucción del PLC debe tener un número de palabras predeterminado por el fabricante.

INSTRUCCION	PALABRAS REQUERIDAS
EXAMEN DE CONTACTO ABIERTO/CERRADO	1
SEÑAL DE SALIDA	1
OPERACION: SUMA	1
RESTA	1
COMPARACION	1
TEMPORIZADOR	3
CONTADOR	3

TABLA 3.2 LA OCUPACION DE LA MEMORIA ES DE ACUERDO A LA INSTRUCCION.

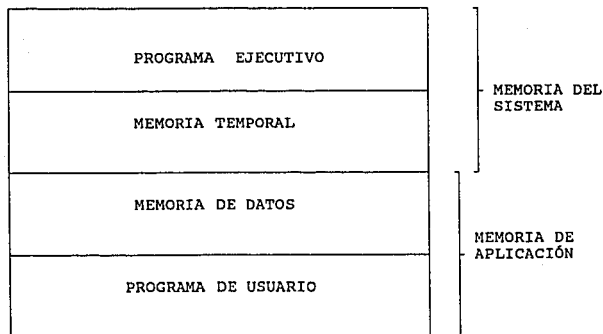


FIGURA 3.12: MAPA SIMPLIFICADO DE LA MEMORIA DE UN PLC.

Una cualidad muy apreciada en la industria de los controladores programables es la de utilizar un mínimo de palabras por instrucción.

El empleo de palabras pequeñas reduce el tiempo de respuesta del controlador programable, pues lee más rápido las instrucciones contenidas en el programa de control.

ORGANIZACION DE LA MEMORIA Y SU INTERACCION CON EL SISTEMA E/S.

La organización de la memoria es llamada también **Mapa de la Memoria**. En el mapa de la memoria se muestran todas las divisiones que tiene tanto para la sección del programa ejecutivo y la sección de aplicación (Figura 3.12).

Como se mencionó antes el **ejecutivo** es una colección programas que son parte del PLC y es quien determina el comportamiento del controlador respecto a cualquier programa introducido por el usuario.

La sección de memoria destinada al sistema del controlador incluye una sección de **memoria temporal**, en los que el controlador graba parcialmente los datos necesarios para ejecutar sus operaciones de cálculo y control.

El área designada como **memoria de datos** en la sección de aplicación, guarda cualquier dato asociado con el programa de control como los que involucran temporizadores y contadores, cualquier constante o variable utilizada por programa de control o la CPU.

En la memoria de datos se retiene el estado lógico de las entradas al sistema una vez leídas y el de las salidas del sistema una vez establecidas por el programa de control.

La memoria de datos se compone de tres partes: **memoria de entradas**, **memoria de salidas** y **área de almacenamiento**. Estas secciones de la memoria contienen información en binario que representa un estado de encendido o apagado, números o códigos útiles en el desarrollo del proceso (Figura 3.13).

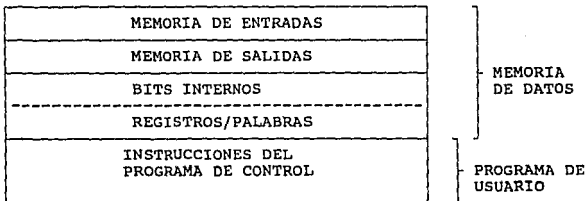


FIGURA 3.13: MAPA SIMPLIFICADO DE LA MEMORIA DE DATOS Y DE USUARIO

MANEJO DE LAS ENTRADAS.

La memoria de entradas es una disposición de bits que guardan el estado y localización de las entradas digitales o discretas, que se conectan al controlador programable por medio de módulos o interfaces de entrada.

Cada entrada tiene un bit que corresponde exactamente a la terminal en que se conecta, la dirección del dispositivo de entrada es interpretado como una palabra o bit que determina su localización dentro de la memoria de entradas (Figura 3.14).

En la figura 3.14 se mapea una señal de entrada con dirección 13010₅, la palabra que contiene la información se ubica en la localidad 130₅ e indica el bastidor en donde se genera la señal, el número 10₅ indica el bit que representa la terminal del bastidor que envía la señal de entrada al controlador.

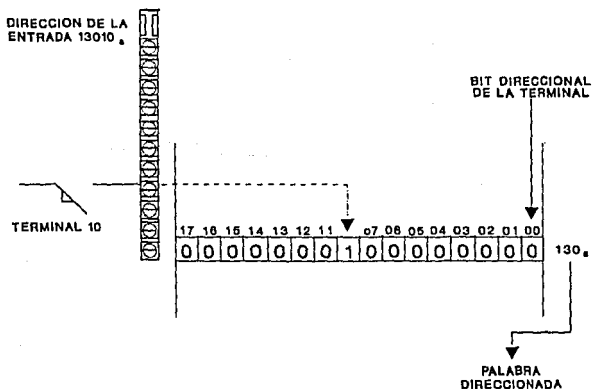


FIGURA 3.14.-MAPEO DE UN DISPOSITIVO DE ENTRADA A LA MEMORIA DE ENTRADAS.

En el ejemplo presentado en la figura 3.14 la dirección es codificada en sistema octal, esta forma no es general en todos los controladores programables.

Los códigos numéricos utilizados por los controladores programables generalmente son en los sistemas Binario, Octal y Hexadecimal.

El número máximo de bits que debe almacenar la memoria de entradas es establecido por el número de entradas de campo que pueden ser conectadas al PLC.

La mayor parte de los PLC modernos pueden ampliar su capacidad agregando módulos de entrada, el número que se pueden agregar es determinado por el fabricante.

MANEJO DE LAS SALIDAS.

La memoria de salida es una disposición de los bits con que se controla el estado y localización de las salidas digitales transmitidas a los dispositivos por el controlador programable, mediante módulos o interfaces de salida.

Cada salida tiene un bit que corresponde exactamente a la terminal en que es destinada, este bit es almacenado en la memoria de salidas por el procesador, indicando una orden de encendido o apagado de acuerdo a la información proporcionada por el monitoreo

El bit representativo de la salida es interpretado y de acuerdo a las instrucciones del programa de control es enviado al módulo de salida correspondiente.

Cuando el bit es 1 un interruptor es puesto en la posición de encendido desde la interfase de salida, si el bit almacenado es 0, entonces el interruptor es puesto por la interfase en una posición de apagado.

La interfase de salida ejecuta las ordenes del controlador al finalizar cada etapa de monitoreo, esto es debido a que en cada exploración se lleva al procesador las condiciones actuales de las variables involucradas en cada paso del proceso para ser comparadas con los parámetros fijados en el programa de control.

La dirección de la señal, es interpretada similarmente a la dirección de una señal de entrada, indica al módulo de salida la ubicación de la terminal a la que debe suministrarse o suprimir la energía [Figura 3.15].

Al igual que en las entradas, la capacidad de almacenamiento de la memoria de salidas esta determinada por el número de salidas que pueden conectarse al PLC dependiendo de la cantidad de módulos que puedan agregarse de acuerdo al fabricante.

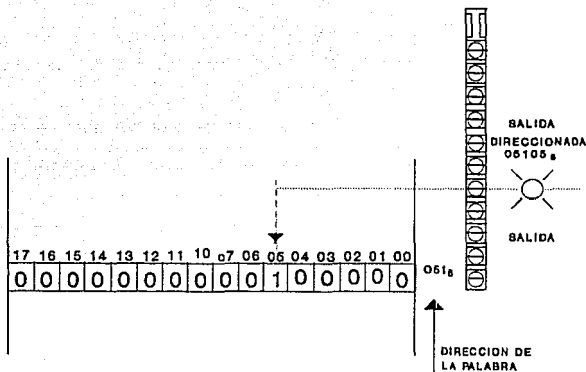


FIGURA 3.15.-MAPEO DE UN DISPOSITIVO DE SALIDA PARA LA MEMORIA DE SALIDAS.

El área de almacenamiento tiene el fin de guardar los datos que parcialmente son requeridos por el controlador para ejecutar alguna acción y que en cada monitoreo son actualizados.

El área de almacenamiento esta formada por dos subdivisiones [figura 3.13]:

La de bits internos que representan señales internas como las de control interno (p. ej. temporizaciones) y señales que son utilizadas como banderas de encendido o apagado, reconocidas por las módulos de salida.

El área señalada como registro/palabra es encargada de almacenar cantidades o códigos necesarios para el funcionamiento del controlador [Tabla 3.3].

Los códigos son entre otros el binario, BCD y el ASCII, las cantidades decimales son representadas en coligo binario, los controladores modernos almacenan incluso cantidades con punto flotante, cualidad que no tenían los primeros controladores programables.

Los registros son clasificados en tres tipos: Registros de entrada, registros remanentes y registros de salida. Los registros pueden guardar la información en código BCD o binario.

El **área de usuario**, es la parte de la memoria del PLC que se asigna a las instrucciones que forman el programa de aplicación para la que se destine el controlador programable.

Todas las instrucciones con que son controlados máquinas y proceso son almacenadas en esta sección del área de aplicación. Las instrucciones son escritas en un lenguaje interpretado por el programa ejecutivo.

Cuando el controlador esta en el modo RUN y el programa del usuario esta siendo ejecutado el procesador interpreta las localidades de memoria y controla los bits en la memoria de datos según corresponda.

CAPITULO IV.

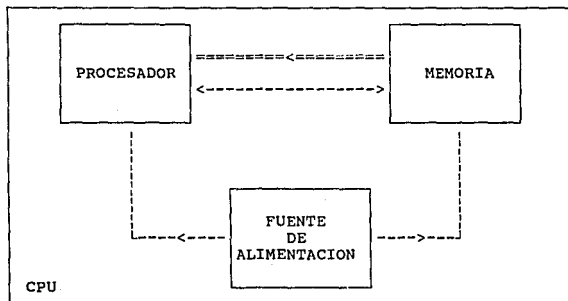
DISPOSITIVOS
PERIFERICOS
Y
APARATOS DE
PROGRAMACION.

EL CONTROLADOR PROGRAMABLE COMO UN COMPUTADOR INDUSTRIAL.

No se debe olvidar que el Controlador Lógico Programable es uno de los miembros más jóvenes de la gran familia de las computadoras y como tal sus principios de funcionamiento son similares aunque con características propias que lo hacen un excelente Computador Industrial.

Analizado como un computador, la unidad central de procesamiento del PLC como se vio en el capítulo anterior cuenta con una memoria de instrucciones, una unidad lógico aritmética y una unidad procesadora de control capaz de realizar ciclos de búsqueda y ejecución de instrucciones [figura 4.1].

Las instrucciones del controlador programable, son una combinación de bits que indican al procesador las acciones que debe realizar, estas instrucciones son leídas por el procesador en las unidades de memoria del PLC [Figura 4.3]



-----INFORMACION: DATOS O INSTRUCCIONES
 -----SEÑALES DE CONTROL.

FIGURA 4.1: DIAGRAMA A BLOQUES DE LA CPU DE UN PLC.

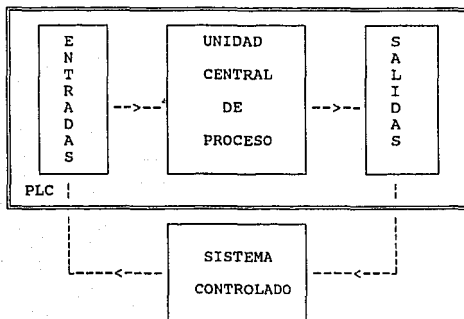


FIGURA 4.2: DIAGRAMA SIMPLIFICADO DE UN SISTEMA CONTROLADO POR PLC.

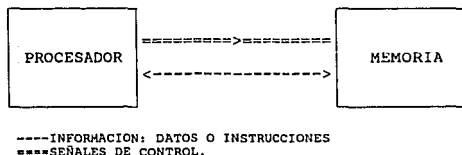


FIGURA 4.3: INTERACCION DEL PROCESADOR Y LA UNIDAD DE MEMORIA.

El enlace entre el procesador y la memoria se constituye por un conjunto de señales de control que envía el procesador a la memoria y un segundo conjunto de señales que consisten en un intercambio de información entre memoria y procesador.

La información de campo es transferida a la CPU desde el exterior, mientras que las señales de respuesta son enviadas por el procesador desde la CPU hacia el exterior.

Como se mencionó en el capítulo 1, el PLC es un computador industrial "inteligente" formado por módulos. La CPU es el cerebro del equipo, los módulos periféricos forman el "cuerpo" del controlador programable a excepción del de comunicaciones que vendría siendo un conjunto de "terminales nerviosas".

Todos los módulos que son conectados a la CPU son conocidos como **Dispositivos Periféricos** [Figura 4.4], generalmente son sistemas digitales y en la mayoría de los casos secuenciales síncronos.

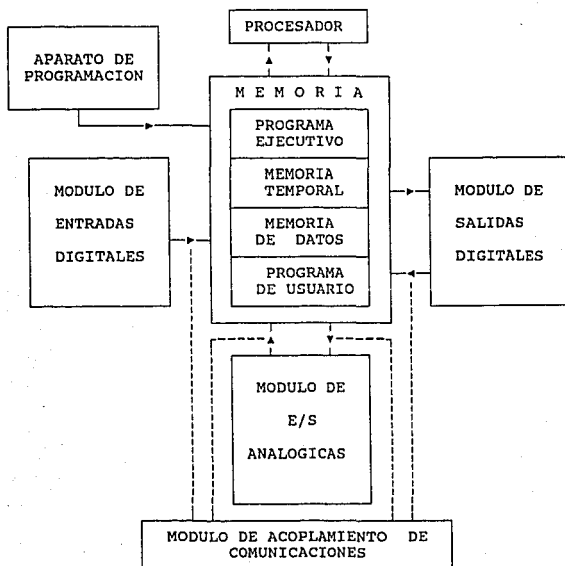


FIGURA 4.4: DIAGRAMA A BLOQUES DE PLC Y SUS PERIFERICOS.

Entre los dispositivos periféricos se tienen todos los módulos E/S de señales discretas y analógicas, sincronía de comunicaciones y un módulo generalmente portátil, conocido como **Aparato de Programación**.

Los módulos E/S son unidades de acoplamiento o interfaces entre el procesador y sus , sus propósitos son de aplicación general y su diseño debe contemplar las características particulares del uso que se dará al módulo.

MODULOS E/S DIGITALES.

Los módulos E/S son fijados en unos bastidores (racks), a los cuales se asigna una dirección de acuerdo a su localización y a la terminal fuente o destino [Figura 4.5]. Se utilizan más comúnmente los módulos E/S digitales que los módulos E/S analógicas.

En el bastidor maestro se instala el módulo de la fuente de poder, el módulo que contiene la CPU, el módulo de entradas digitales, el módulo de salidas digitales y el módulo de comunicaciones, regularmente en ese orden, cuando el sistema es ampliado, los módulos agregados, suelen instalarse en otro orden para no cambiar toda la instalación.

Cuando la capacidad del programador necesita de varios bastidores deben colocarse en forma matricial respetando el orden de los puestos como se indica en la figura 4.6, esto es con el fin de facilitar la localización de fallas.

SISTEMAS REMOTOS DE E/S.

Se pueden manejar subsistemas remotos, en la instalación del sistema, los módulos E/S conectan físicamente la unidad central de proceso con los dispositivos de campo ya sea en un sistema local o remoto.

Un sistema remoto habitualmente incluye los mismos elementos que contiene el sistema maestro, los sistemas muy grandes controlados por PLC requieren de tarjetas de comunicaciones con un grado de sofisticación elevado [Figura 4.7a, 4.7b y 4.7c].

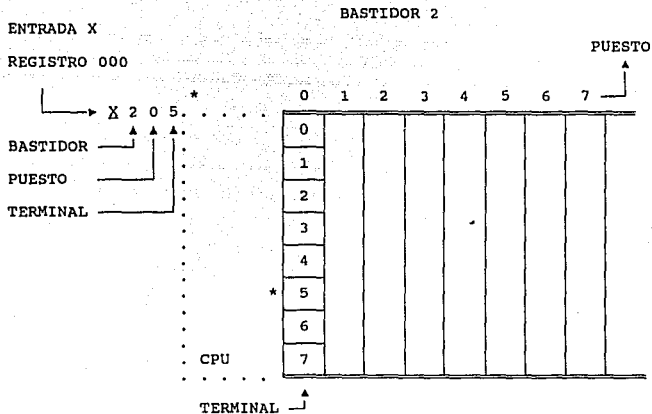
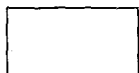
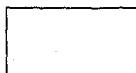


FIGURA 4.5.- INFORMACION QUE CONTIENE LA DIRECCION ASIGNADA VARIABLE X.



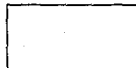
GABINETE 0



GABINETE 1



GABINETE 2



GABINETE 3

FIGURA 4.6.- CADA GABINETE ESTA COMPUESTO POR UN JUEGO DE BASTIDORES. LA UBICACION DE CADA GABINETE SE AGREGA A LA IZQUIERDA DE LA DIRECCION QUE UBICA UNA SEÑAL EN UN JUEGO DE BASTIDORES.

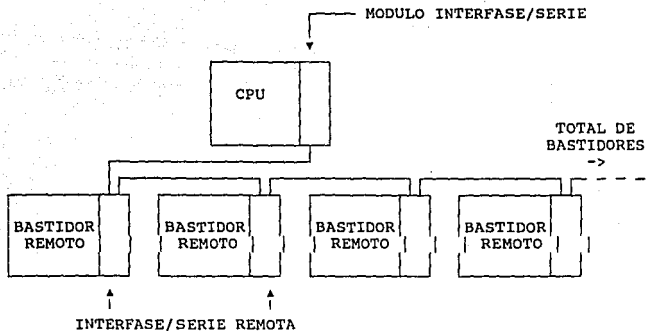


FIGURA 4.7A: CONFIGURACION DE E/S REMOTAS: CADENA DE MARGARITAS

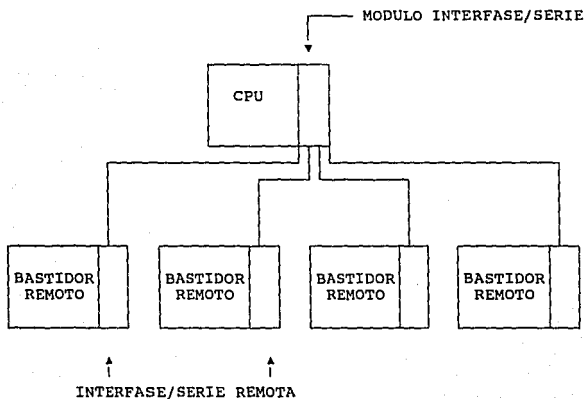


FIGURA 4.7B: CONFIGURACION DE E/S REMOTAS: ESTRELLA

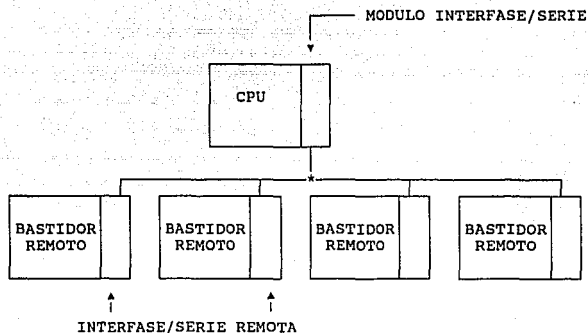


FIGURA 4.7C: CONFIGURACION DE E/S REMOTAS: ESTRELLA

Los sistemas remotos no sólo amplían una planta en otro lugar, sino que simplifican el control al reducir el acumulamiento de circuitos eléctricos en la zona del proceso.

ENTRADAS DISCRETAS.

* Entrada AC/DC.

El circuito de entrada a los módulos de entrada AC/DC normalmente cuentan con un circuito de aislamiento entre la parte digital y la entrada de la línea [Figura 4.8].

El módulo de entrada AC/DC tiene un circuito puente rectificador que convierte las señales AC a señales DC antes de alimentar el circuito lógico conectado al procesador.

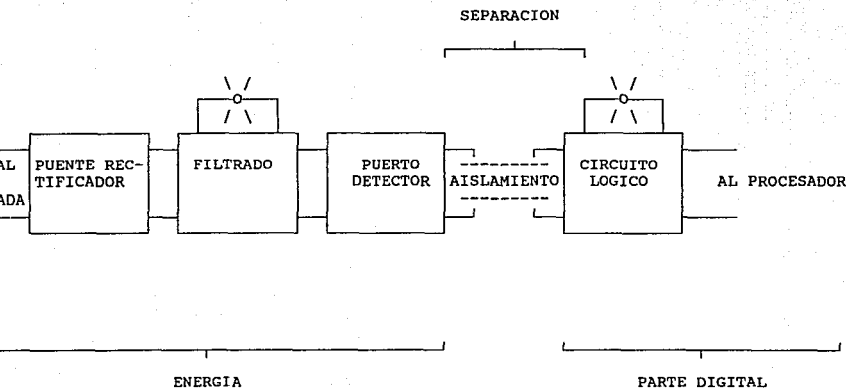


FIGURA 4.8: DIAGRAMA A BLOQUES DE UNA ENTRADA TIPICA AC/DC.

La diferencia entre un módulo de entrada DC y un módulo de entrada AC/DC, es que el primero no necesita de un circuito convertidor.

La sección de energía de un módulo de entradas tiene la tarea de convertir las señales de dispositivos sensores [Tabla 4.2], de 230 V AC, 115 V AC, etc., a señales DC que son filtradas y protegidas de fluctuaciones sobre la línea.

* Entradas DC.

Los módulos de entradas DC acoplan los dispositivos de campo que envían señales DC a la CPU, los rangos de voltaje varían entre 5 y 30 Volts DC, el módulo reconoce una señal de encendido, si el nivel de voltaje que detecta es del 40 % con respecto al de referencia u otro nivel especificado por el fabricante.

Los módulos de entrada DC reciben corriente desde un dispositivo de entrada de alimentación, a su vez suministran la corriente que demandan los dispositivos de entrada [Tabla 4.2].

DISPOSITIVOS DE CAMPO: EMISORES/GENERADORES

ENTRADAS DISCRETAS	SALIDAS DISCRETAS
INTERRUPTOR SELECTOR	ALARMA
BOTON PULSADOR	LAMPARAS
OJO ELECTRONICO	BOCINAS
INTERRUPTOR LIMITADOR	VALVULAS
INTERRUPTOR AUTOMATICO	ARRANCADORES DE MOTOR
INTERRUPTOR DE PROXIMIDAD	SELENOIDES
CONTACTOS DE ARRANCADOR DE MOTOR	VENTILADORES

TABLA 4.2: ALGUNOS TIPOS DE DISPOSITIVOS ENTRADAS O SALIDAS DIGITALES

SALIDAS DISCRETAS.

Las interfases que manejan la información en forma digital son las más solicitadas a los fabricantes de controladores programables.

Los módulos de salida digitales son la conexión entre el controlador programable y los dispositivos de campo que son controlados a través del PLC.

Los dispositivos controlados regularmente son digitales de fábrica y sólo requieren de un bit indicador de encendido/apagado o abierto/cerrado [Tabla 4.2], generalmente los módulos de salida digital deben manejar varios niveles de voltajes [Tabla 4.1].

En general los módulos de salida DC usan un transistor de potencia como interruptor de voltaje DC. El diseño se realiza habitualmente con dispositivos optoelectrónicos que proporcionan una protección contra las posibles corrientes en la línea que alimenta la planta [Figura 4.9].

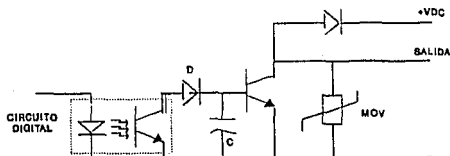


FIGURA 4.9.- CIRCUITO TÍPICO DE SALIDA DE SALIDA DC.

Salidas AC.

La mayoría de los circuitos de entrada a los dispositivos de la planta requieren de uno o varios niveles de voltaje AC, estos niveles de voltaje son suministrados por los módulos de salida A/C.

Los módulos de salidas AC, son ensamblados con un circuito aislador que separa la parte digital y la sección donde se genera la energía con que se alimentan los dispositivos [Figura 4.10].

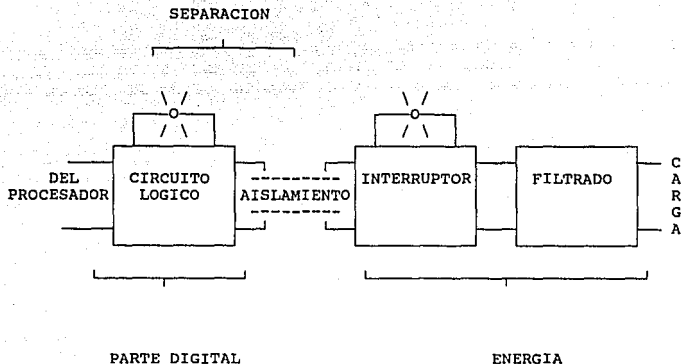


FIGURA 4.10: DIAGRAMA A BLOQUES DE UNA SALIDA AC

Durante la operación del sistema el PLC puede controlar un dispositivo manipulando un interruptor de alimentación a través del módulo de salida AC.

El procesador es el encargado de llevar a la interfase, la instrucción del programa de control, que ordena la apertura o cierre del interruptor.

Los módulos de las salidas AC cuentan con un circuito alternador construido básicamente por TRIACS y SCRs, pertenecientes a la familia de semiconductores conocida como tiristores.

Los tiristores son conocidos como elementos electrónicos de potencia. La protección de las señales de salida comúnmente utilizan circuitos RC de paro repentino (RC snubber) y varistores de metal óxido.

MODULOS DE ENTRADA NO-VOLTAJE.

Los módulos de entrada No-Voltaje, facilitan la detección de la condición de apertura y cierre de contactos, acepta señales de dispositivos de entrada discretos como los mencionados en la tabla 4.2.

Los módulos de entrada No Voltaje se diferencian de otros módulos de entrada por no requerir que los dispositivos de su dependencia sean alimentados por fuentes de energía externa.

ENTRADA/SALIDAS TTL

Los módulos E/S TTL, permiten el acoplamiento de la CPU con la variedad de dispositivos que pueden constituir un proceso industrial como pueden ser controles de estado sólido e instrumentos sensores.

* Entradas TTL.

Las entradas TTL Permiten al controlador programable una mejor interacción con señales TTL generadas por dispositivos compatibles, facilitan el acoplamiento de dispositivos sensores a 5 V DC y algunos tipos de sensores fotoeléctricos. La configuración de estos módulos tienen una configuración similar a la de los módulos de entradas AC/DC aunque en los módulos de entrada TTL, el filtrado es de mejor calidad. Generalmente los módulos de entrada TTL son alimentados por la fuente de poder del bastidor donde están instalados, algunos requieren de una fuente de energía externa.

* Salidas TTL.

Las interfases de salida TTL facilitan al controlador el manejo de dispositivos activados electrónicamente como son los circuitos integrados y en general instrumentos digitales auxiliares en los procesos de producción. Los módulos de salida TTL, requieren normalmente de una fuente externa de 5 V DC, con especificaciones de corriente adecuadas a los dispositivos de campo controlados por medio del módulo. Generalmente los módulos de salida TTL se encuentran con ocho terminales de acoplamiento para los dispositivos digitales compatibles, algunos permiten la conexión simultánea de varios dispositivos.

OTROS DISPOSITIVOS DE SEÑALES DISCRETAS.

Los dispositivos manuales de respaldo o bypass pueden incorporarse al sistema de PLC, para dar flexibilidad a las pruebas de operación o para forzar el encendido o apagado de dispositivos, en caso de que se presente alguna falla en las interfases de activación.

Los módulos de registro E/S, proporcionan una comunicación paralela entre los dispositivos E/S y el procesador, manejan dispositivos de campo como los indicadores de siete-segmentos.

Los módulos de contactos de salida, son utilizados como interruptores de señales AC o DC a la carga. Este tipo de interfase aplican el multiplexado de señales analógicas para control de interrupciones cuando se presentan casos de bajos voltajes y corrientes pequeñas o muy grandes.

ENTRADAS	SALIDAS
TRANSDUCTORES DE TEMPERATURA	VALVULAS ANALOGICAS
TRANSDUCTORES DE PRESION	ACTUADORES
CELDAS TRANSDUCTORAS DE CARGA	GRAFICADORES
TRANSDUCTORES DE HUMEDAD	MEDIDORES ANALOGICOS
TRANSDUCTORES DE FLUJO	ARRANCADORES ELECTRICOS DE MOTOR
POTENCIOMETROS	TRANSDUCTORES DE PRESION

TABLA 4.3: ALGUNOS TIPOS DE DISPOSITIVOS ANALÓGICOS DE ENTRADA O SALIDA.

MODULOS E/S ANALOGICAS.

Es conocido el hecho que en realidad no existen señales de tipo discreto, sino que más bien las señales en su forma más pura son señales analógicas, es por ello que en los procesos industriales ha sido necesario la concepción de equipo que facilite el manejo de estas señales para los diferentes formas de controlar los procesos.

En los continuos avances de la ciencia, las aplicaciones del Control Lógico Programable son innumerables, en la manufactura del PLC se contempla la necesidad de incluir módulos E/S analógicos que le permitan el manejo de las variables analógicas que un proceso pueda generar.

Los módulos E/S analógicos son utilizados en aplicaciones que requieren el monitoreo de variables continuas como la temperatura, presión etc.

Los módulos E/S analógicas tienen circuitos convertidores A/D o D/A, que transforman una señal de tipo analógico a una señal digital o viceversa según si es módulo de entrada o salida.

Los módulos E/S analógicas permiten el examen de voltajes o corrientes analógicos de dispositivos sensores, instrumentos, etc., [Tabla 4.3].

ENTRADAS	SALIDAS
4-20 MA	4-20 MA
0 A +1 VOLTS DC	10-50 MA
0 A +5 VOLTS DC	0 A +5 VOLTS DC
0 A +10 VOLTS DC	0 A +10 VOLTS DC
1 A +5 VOLTS DC	2.5 VOLTS DC
5 VOLTS DC	5 VOLTS DC
± 10 VOLTS DC	+10 VOLTS DC

TABLA 4.4: RANGOS DE TRABAJO MAS COMUNES EN LOS MODULOS E/S ANALOGICOS.

* Entradas Analógicas.

Las entradas analógicas pueden ser unipolares (positivas) o bipolares (positivas y negativas). Los rangos de trabajo más usuales se listan en la tabla 4.4.

Los módulos de entrada analógicas convierten las señales continuas dentro de cálculos establecidos para el proceso de conversión A/D, dando como resultado señales digitales proporcionales a los voltajes o corrientes medidos por los módulos de entrada.

Los módulos de entrada analógicas tienen una alta impedancia de entrada, lo que permite su acoplamiento con dispositivos con impedancias grandes de salida.

Los datos analógicos son transferidos por el procesador desde el módulo en que son generados hasta la localidad de memoria en que son almacenados. Generalmente son manejados en canales de señales analógicas y transferidos mediante instrucciones especiales del PLC para su manejo.

Las entradas analógicas son conectadas a la interfase, mediante módulos diferenciales u otros tipos de módulos apropiados a las características del PLC y las señales del proceso.

* Salidas Analógicas.

Los módulos de salida analógica son utilizados en el control de dispositivos de salida que necesitan corrientes o voltajes analógicos para su operación [Tabla 4.3]. Requieren de convertidores D/A para transformar los valores digitales del equipo a un valor analógico equivalente.

Al igual que las entradas del mismo tipo, las salidas analógicas pueden ser unipolares o bipolares, con rangos de trabajo típicos como los mostrados en la tabla 4.4.

Los módulos de salida analógicos convierten un valor numérico almacenado en la memoria del sistema a una señal analógica proporcional.

Los datos de salida analógicos, son manejados por el procesador mediante instrucciones características del PLC para el manejo de señales analógicas.

Los módulos de salida analógicos manejan de cuatro a ocho canales, a través de los cuales pueden actualizarse los datos con una instrucción del procesador.

Las líneas transmisoras de señales analógicas deben ser tendidas con cables que no permitan fluctuaciones causadas por ruido eléctrico o cualquier otro factor que pueda alterar la información contenida en la señal analógica.

MODULOS DE COMUNICACIONES Y FUNCIONES ESPECIALES.

MODULOS DE COMUNICACIONES.

El continuo desarrollo del control industrial y la sofisticación de los procesos productivos en la actualidad creó dentro de la manufactura de los PLC, el desarrollo de sistemas electrónicos que ampliaran sus perspectivas de aplicación.

La instalación de sistemas complejos presenta la necesidad de una red de comunicaciones que satisfaga los requerimientos de una buena comunicación entre los subsistemas y el sistema maestro.

Un módulo de comunicaciones facilita el manejo de las diferentes señales que contienen la información que emanan controlador y proceso, de forma que se optimice la interacción entre ambos.

La función de un módulo de comunicaciones es la manipulación simultánea de las señales que contienen la información de las variables del proceso, además de las señales que envía el PLC para el control del equipo.

Un módulo de comunicaciones es diseñado de acuerdo al número de autómatas (PLC) y dispositivos inteligentes que generan información en un sistema automatizado mediante controladores programables, la complejidad de la red de comunicaciones es dictada por la sofisticación del sistema de control.

Normalmente los PLC que pueden ser acoplados al módulo que proporciona la red de comunicaciones, están restringidos a utilizar únicamente módulos manufacturados por el mismo fabricante, limitando así el tipo y número de dispositivos que pueden ser conectados a la red.

Generalmente, cuando un mensaje se emite por el procesador u otro dispositivo conectado a la red, es retransmitido por el módulo de comunicaciones a través de la red en un rango de baudios en la transmisión determinado por la capacidad de respuesta de los dispositivos y el módulo de comunicaciones.

La comunicación entre periféricos más típica es en forma serie a un rango de velocidad de transmisión de 110 a 19200 baudios (bits por segundo), dependiendo del módulo de comunicaciones y la velocidad de respuesta del equipo conectado a la red.

La transmisión contempla la paridad o no paridad, asincronía, etc., características supeditadas al uso de los diferentes tipos de acoplamiento que proporcionan los fabricantes de módulos de comunicaciones.

La red recibe y envía el mensaje a un dispositivo predeterminado en las acciones de control, el tipo de protocolo que se utiliza para el enlace de la red depende de la arquitectura y alcances del módulo de comunicaciones seleccionado.

La problemática de las redes de comunicaciones dentro de una planta industrial, presenta tantos aspectos, que organizaciones internacionales se han ocupado de normalizar paulatinamente el tipo de conexiones en las redes de comunicación dentro de la automatización industrial.

COMUNICACION ESTANDAR.

Básicamente el tipo de comunicación más comúnmente utilizada es dictada por el Instituto de Ingeniería Eléctrica y Electrónica [IEEE: Institute of Electrical and Electronics Engineers] y la Asociación de la Industria Electrónica [EIA: Electronic Industries association].

La IEEE y la EIA norman las características que debe contemplar la fabricación de los módulos de comunicación para tener una red de comunicaciones con un esquema de estabilidad y compatibilidad con los diferentes productos de manufactura estándar. La comunicación estándar la engloban en dos categorías oficiales: Normas de facto y Normas proclamadas.

Las normas de facto son aquellas normas que debe seguir todo fabricante en la manufactura del módulo de comunicaciones para que tenga una forma de comunicación estandarizada.

Las normas proclamadas son las normas que utiliza el fabricante dando características particulares al diseño de sus equipos.

En las figuras 4.11, 4.12 y 4.13 se representan algunas conexiones de los módulos de comunicación y el controlador programable.

La figura 4.11 muestra la conexión a un PLC, de una terminal de datos por medio de una conexión serie normalizada (interfase de acoplamiento IEEE-488, EIA RS-232C, EIA RS-422, RS-232, etc.), esta conexión se hace a través de un módulo de comunicación serie.

CONTROLADOR PROGRAMABLE

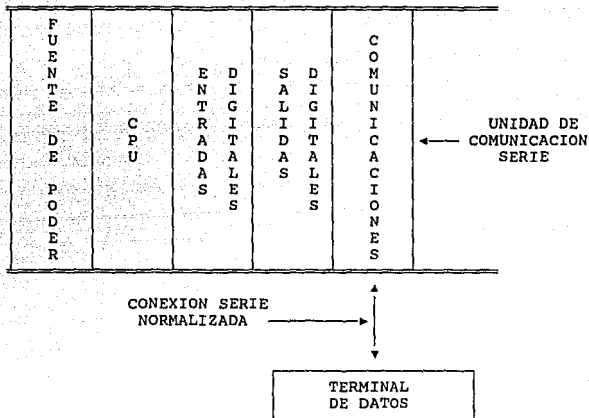


FIGURA 4.11: ENLACE DE UNA TERMINAL DE DATOS Y DE UN PLC A TRAVES DE UNA CONEXION SERIE NORMALIZADA.

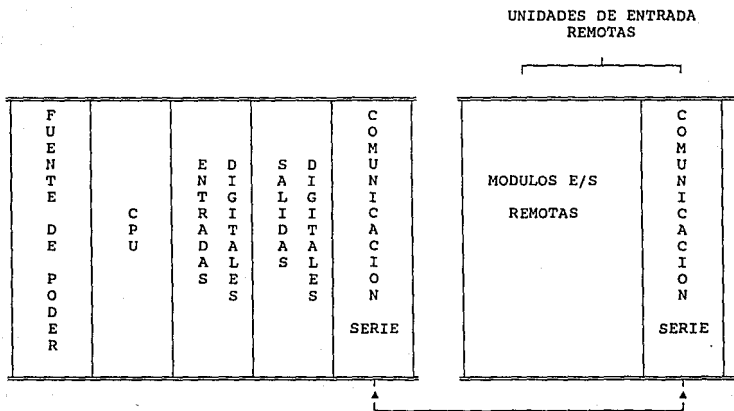


FIGURA 4.12: CONEXION DE E/S REMOTAS A UN PLC.

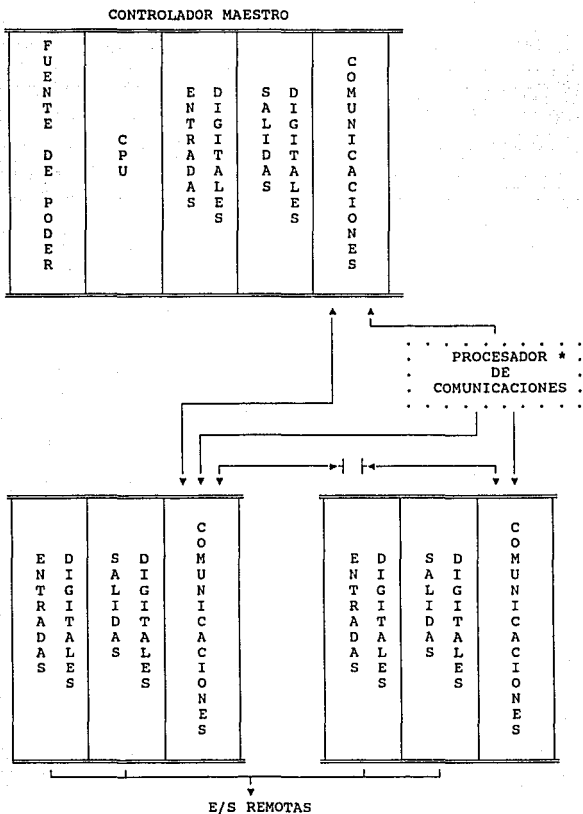


FIGURA 4.13: CONEXION DE UN CONTROLADOR PROGRAMABLE A LOS MODULOS DE E/S REMOTOS. (* GENERALMENTE EL PROCESADOR DE COMUNICACIONES FORMA PARTE DEL HARDWARE DEL MODULO DE COMUNICACIONES).

Las interfaces IEEE-488, EIA RS-232C y EIA RS-422 se basan en la categoría proclamadas. Ejemplos de la categoría de facto son la interfase unicable PDP-11 y la interfase de corriente cíclica 20 mA (Loop Current).

MODULOS DE FUNCIONES ESPECIALES.

Los módulos de funciones especiales son utilizados cuando señales específicas requieren de un preprocesamiento o una función especial de operación que necesita ser ejecutada en el módulo.

Los módulos de preprocesamiento son interfaces que acondicionan las señales de bajo nivel. Cuando se tienen módulos E/S inteligentes son incorporados dentro del sistema de PLC, es decir cuando el sistema automatizado cuenta con una distribución de E/S preprocesadas.

Los módulos especiales de E/S discretas incluyen el manejo de Señales de entrada rápidas, fallas en el alambrado de entrada, alta velocidad de respuesta, salidas redundantes, etc.

El módulo de entradas rápidas es elementalmente un acoplador de pulsos ya que agranda o reduce la señal entrada de acuerdo al tiempo de monitoreo. El período normalmente está entre 50 y 100 microsegundos y el rango de voltaje entre 10 y 24 Volts DC.

Los módulos de falla en el alambrado de entrada son diseñados con el fin de detectar fallas de continuidad y corto circuito en las conexiones mediante las variaciones de la corriente suministrada al módulo.

Los módulos para alta velocidad de respuesta tienen la función de habilitar señales E/S que requieren una rapidez de respuesta alrededor de un microsegundo.

Los módulos de salidas redundantes proporcionan un circuito reservado para el manejo de la carga de salida. Cuando en el módulo se detecta una falla en el primario del circuito de uso normal, es accionado automáticamente el secundario o en su defecto, el circuito de respaldo.

Los módulos especiales E/S incluyen dispositivos con técnicas de control proporcional, integral y derivativo (PID) y control analógico.

El módulo de entradas por termocopladores, acepta directamente señales del orden de milivolts, desde los transductores termocoplados al módulo.

Este tipo de módulo tiene un funcionamiento similar al de los módulos de entrada analógica con la diferencia que este módulo almacena directamente en un registro el valor de temperatura obtenido en la lectura del termocoplador, ya sea en bits o en código BCD.

Usualmente el módulo de entrada por termocopladores provee una compensación de frío en la juntura dado que la señal sufre pérdidas debido a la resistencia de la conexión propia del módulo.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

El módulo RTD (Resistance Temperature Detector) es un detector de variaciones de resistencia debido a cambios en la temperatura, que permite el acoplamiento directo a los dispositivos. Los rangos de trabajo se más comunes se listan en la tabla 4.5.

TIPO RTD	RANGO DE RESISTENCIA (OHMS)	RANGO DE TEMPERATURA (O C)	RANGO DE TEMPERATURA (O F)
PLATINO	100	-200 A 850	-328 A 1562
NIQUEL	120	-80 A 300	-112 A 572
COBRE	10	-200 A 260	-328 A 500

TABLA 4.5: RANGOS DE TRABAJO EN LOS MÓDULOS RTD MAS USUALES.

Los módulos de control proporcional, integral y derivativo (PID) son requeridos por sistemas que requieren lazos de control cerrados como puede ser la regulación de una variable específica del proceso.

Básicamente los módulos PID son módulos (generalmente inteligentes) que tienen la función de controlar variables analógicas del proceso cuyas lecturas son proporcionadas por el monitoreo, manteniendo un rango fijo de trabajo (Setpoint).

Normalmente los módulos PID guardan una comunicación bidireccional con el proceso vía instrucciones de transferencia.

La industria de los PLC contempla módulos que satisfacen las necesidades de dispositivos que requieren controlar la posición de equipo aislado como pueden ser motores de paso, servomotores, etc.

Se cuenta también con módulos que toman las señales de dispositivos como interruptores de límite, marcadores, acumuladores, canales de comunicación etc.

Generalmente un módulo de este tipo es conocido como módulo contador/decodificador, dado que puede manejar dispositivos acumuladores en general y decodificar señales como la de un interruptor de límite y otras que requieren ser traducidas al lenguaje del controlador programable.

APARATOS DE PROGRAMACION.

Se ha mencionado reiteradamente que el PLC requiere de un tipo de lenguaje específico, determinado por el fabricante; existen diversos lenguajes que permiten la escritura básica de un programa de control, sin embargo debe recordarse que estos lenguajes con codificados en una composición fundamentalmente binaria.

Un aparato de programación es un sistema electrónico que tiene asignadas las siguientes funciones:

- * Funciona como interprete entre el controlador programable y el usuario.
- * Su arquitectura contempla el diseño de programas de control en un lenguaje claro para el usuario y un código que optimice los comandos internos de la CPU.
- * Transferencia del programa de control al PLC que lo almacena en forma binaria dentro de la memoria de aplicación en el área reservada para el usuario.
- * Simulación del comportamiento del programa de control.

La diversidad de aparatos de programación ofrecen características que varían de acuerdo al fabricante, los más comunes incluyen: el tubo de rayos catódicos (TRC), miniprogramadores y computadoras personales. El tipo de lenguaje que maneja el PLC, determina el tipo de aparato de programación que requiere.

PANTALLA DE TUBO DE RAYOS CATODICOS (TRC).

La pantalla de TRC es la más utilizada para la programación de los controles programables, ofrecen principalmente la ventaja de ver la lógica de control sobre la pantalla desde el inicio hasta el final de su creación dentro del programa de aplicación.

El TRC, es clasificado en dos categorías: como unidad tonta y unidad inteligente, la primera se comunica con el PLC para una programación en línea únicamente, mientras que las unidades inteligentes pueden programar en línea con el autómata o fuera de línea, es decir; no requieren conectarse al PLC para diseñar o capturar el programa de control.

MINIPROGRAMADORES.

Los miniprogramadores, son conocidos también como programadores manuales y portátiles, comúnmente su uso se restringe a PLC de poca capacidad (128 E/S físicas).

Algunos controladores programables incluyen un miniprogramador y en algunos casos es desmontable. El diseño de una unidad debe contemplar su acoplamiento con dos o más PLC de la familia de productos. Los miniprogramadores son utilizados como una herramienta de cambios, arranque y monitoreo de las funciones lógicas de control.

COMPUTADOR PERSONAL.

El uso común en la vida diaria de la computadora personal (PC), provocó la concepción de la misma como un posible aparato de programación, obligando a los fabricantes de PLC a suministrar una fuente de acoplamiento entre sus controladores programables y las denominadas PC compatibles.

Los fabricantes de controles programables modernos cuentan con un dispositivo de acoplamiento que se agrega a una computadora personal que cumpla con ciertas características de memoria y compatibilidad.

Este dispositivo habilita a la computadora personal para recibir el software necesario para la captura, prueba y edición del programa de control; agregando una herramienta en funciones importantes del autómeta como es el monitoreo.

Los alcances de estos dispositivos de acoplamiento en cuanto a funciones y estructura dependen exclusivamente del fabricante de controladores programables, basándose en las habilidades que quiera darle a la computadora personal.

La capacidad de representación y visualización que requiere un PLC del aparato de programación depende de los tipos de lenguaje de programación que manejen.

Los autómetas más sencillos manejan básicamente el diagrama de escalera, también conocido como diagrama de contactos y la lista de instrucciones, este último lenguaje, por ejemplo puede representarse en microprogramadores sencillos que cuenten con una pantalla en base a LEDs (diodos emisores de luz).

Sin embargo existen programadores que manejan un buen número de representaciones de los programas de control en lenguajes sofisticados que proporcionan al usuario un sencillo panorama de las funciones de control.

Sin duda son este tipo de controladores los que requieren de aparatos de programación con pantallas con una alta resolución como son las de TRC, panel de plasma o cristal líquido.

Existen infinidad de módulos de periféricos con funciones específicas que han venido facilitando el control dentro de la automatización basada en el controlador lógico programable.

La complejidad de los sistemas de control industrial es prácticamente infinita dado el campo de aplicaciones que proporcionan los avances tecnológicos.

La información anterior pretende explicar en forma global las herramientas con que se cuenta actualmente en los sistemas automatizados por medio de PLC.

CAPITULO V.

LENGUAJES
DE
PROGRAMACION.

CONTROLADORES PROGRAMABLES Y EVOLUCION DEL SOFTWARE.

El software utilizado en los controladores programables ha venido evolucionando desde su aparición en los años 60 's, al ritmo del desarrollo de las ciencias computacionales y su aplicación a nivel industrial.

Con la proliferación de los nuevos productos, la versatilidad de las instrucciones hizo una necesidad, el desarrollo de lenguajes de programación del PLC con instrucciones para mejorar las tareas del controlador programable.

El monitoreo de los resultados de un paso en particular, la transferencia de bloques de datos, consulta directa de la memoria, consulta de un tiempo particular de ejecución, el manejo de datos en general, son algunas de las facilidades que los autómatas modernos ofrecen al usuario mediante lenguajes de programación sofisticados.

La adición de nuevas instrucciones vienen originándose con el desarrollo de los módulos E/S que día con día requieren de mejores formas de envío y recepción de las señales de información del sistema automatizado.

La programación del PLC requiere de una secuencia de instrucciones que constituyen el programa de control, cada autómata responde a un lenguaje de máquina de acuerdo a su hardware de construcción.

El programa de control bien podría capturarse en el lenguaje de máquina del controlador programable, sin embargo esta forma de escritura dificultaría la comprensión de la lógica de control al usuario. De esta manera es como los fabricantes de controladores lógico programables han venido generando lenguajes de programación que guardan a la vez un alto grado de sofisticación y sencillez.

La industria del PLC por tanto, ha venido conceptualizando la facilidad de programación con la representación efectiva del programa de control, aspectos fundamentales en el control de máquinas o procesos.

Esta representación de la lógica de control, se basa en el uso de los diagramas lógicos a base de relevadores, surgiendo como resultado un primer lenguaje de programación denominado **Diagrama de Escalera** o **Diagrama de Contactos**, de estructura similar a la de un diagrama eléctrico.

Las operaciones elementales representadas por relevadores y las operaciones del diagrama de escalera son prácticamente las mismas. como se menciona las funciones en el diagrama de escalera son de mayor poder en cuanto a manipulación de la información y los comandos de control.


Un ejemplo de la similitud entre el diagrama de contactos del PLC y operación por relevadores es mostrado en la figura 5.1, en donde un relevador A tiene puestos dos contactos, uno normalmente abierto (A-1) y uno normalmente cerrado (A-2).

Si en el relevador no se alimenta la señal A, es decir permanece apagado: El contacto A-1 permanecerá abierto, mientras que el contacto A-2 permanecerá cerrado. Si por el contrario la terminal correspondiente a la señal A es alimentada: El contacto A-1 se cerrará, mientras que el contacto A-2 se abrirá.

Hay que recordar, que cuando un puesto de contactos es cerrado se activa un flujo de energía o continuidad en el circuito donde es utilizado el puesto.


Cada puesto de activación de señal en el PLC cuentan en sus respectivos contactos con una dirección asignada por contacto para su identificación en la memoria de datos.

Los siguientes símbolos son utilizados en la traslación de la simbología por relevadores a la simbología de contactos lógicos. Estos símbolos son también las instrucciones básicas establecidas para el diagrama de escalera.

CONTACTOS NORMALMENTE ABIERTOS  :

Representa cualquier entrada para el control lógico. Una entrada puede ser una conexión a un interruptor cerrado, un sensor, un contacto desde una conexión de salida o un contacto desde una salida interna¹⁾.

Quando es interpretado, la entrada o salida direccionada es examinada para una condición de encendido. Si su estado es "1", el contacto será cerrado permitiendo el flujo de corriente a través del mismo. Si el estado de la entrada o salida direccionada es "0", el contacto estará abierto y no permitirá el paso de la corriente.

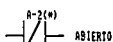
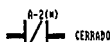
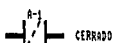
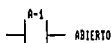
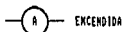
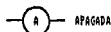
CONTACTOS NORMALMENTE CERRADOS  :

Representa cualquier entrada para el control lógico. Una entrada puede ser una conexión a un interruptor cerrado, un sensor, un contacto desde una conexión de salida o un contacto desde una salida interna.

(*) Una señal interna se utiliza en funciones de control no visibles para el usuario.



a) CONFIGURACION ESTANDAR PARA RELEVADORES NA Y NC.

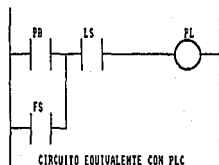
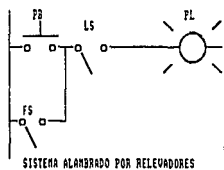


b) SEÑAL A ENERGIZADA

c) SEÑAL A DES-ENERGIZADA

(*) LA LINEA GUESA INDICA CONDICION DE CERRADO EN LOS CONTACTOS (INTERRUPCION DEL FLUJO DE ENERGIA)

FIGURA 5.1.- SIMBOLOGIA BASICA DEL LENGUAJE DE ESCALERA REPRESENTANTE DE SISTEMAS POR RELEVADORES.



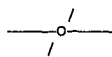
5.2.- SISTEMA POR RELEVADORES Y SU EQUIVALENTE EQUIVALENTE EN DIAGRAMA DE CONTACTOS.

Quando es interpretado, la entrada o salida direccionada es examinada para una condición de apagado. Si su estado es "0", el contacto permanecerá cerrado permitiendo el flujo de corriente a través del mismo. Si el estado de la entrada o salida direccionada es "1", el contacto se abrirá y no permitirá el paso de la corriente.

SALIDA  :

Representa cualquier salida que sea manejada por una combinación de entradas lógicas. Una salida puede ser la conexión a un dispositivo o una salida interna.

Quando todas las condiciones de entrada a la izquierda o derecha son verdaderas (todos los contactos cerrados), la terminal de salida con la dirección indicada será alimentada (posición de encendido).

SALIDA NEGADA  :

Representa cualquier salida que sea manejada por una combinación de entradas lógicas. Una salida puede ser la conexión a un dispositivo o una salida interna.

Quando todas las condiciones de entrada a la izquierda o derecha son verdaderas (todos los contactos cerrados), se suspende la alimentación a la terminal de salida con la dirección indicada (posición de apagado).

El procedimiento que debe seguirse para trasladar un sistema clásico por relevadores a una programación en simbología de contactos lógicos, se indica a continuación.

PROCEDIMIENTO DE TRASLACION A LA SIMBOLOGIA DE CONTACTOS.

Contactos normalmente abiertos: Cuando este tipo es evaluado por el programa de control, este símbolo es examinado como "1", para cerrar el contacto abierto; de tal forma que para cambiar de condición, en la dirección asignada al contacto se debe tener una señal que indique la orden de encender, cerrar, activar, etc.

Contactos normalmente cerrados: Cuando este tipo es evaluado por el programa de control, este símbolo es examinado como "0", guardando la condición de contacto cerrado; de manera que en la dirección referenciada se tenga un indicación de apagar, abrir, desactivar, etc.

Diagramas

Diagramas

Diagramas

Diagramas

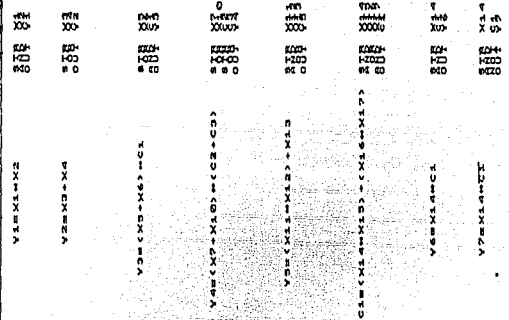
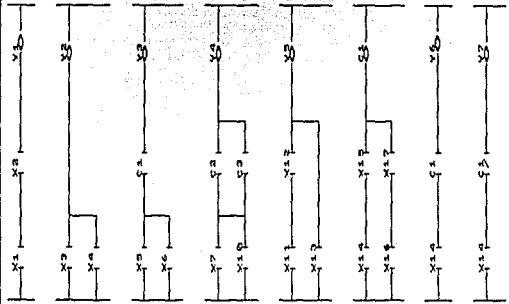
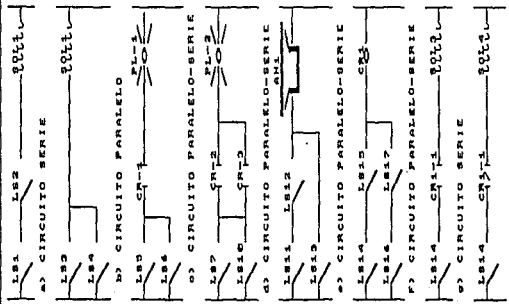


Diagrama serie
EJEMPLO DE EQUIVALENCIAS ENTRE EL CABLEADO DE RELEVADORES Y EL DIAGRAMA DE CONTACTOS.

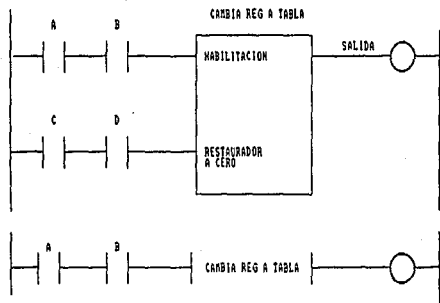


FIGURA 5.3.- EJEMPLO DE INSTRUCCION AGREGADA AL CONJUNTO BASICO DE SIMBOLOS DEL DIAGRAMA DE ESCALERA.

Salidas: una salida de encendido dada en una etapa o "rung" es alimentada si cualquier circuito a la izquierda o derecha tiene continuidad, es decir los contactos que contiene están cerrados, una salida puede activar un dispositivo si se direcciona la ubicación del mismo, siempre y cuando la referencia indique también un punto terminal o una señal interna.

Entradas: un símbolo de contactos puede representar una señal de entrada enviada desde las entradas conectadas, contactos desde salidas internas o contactos desde salidas conectadas.

Dirección de Contactos: Cada símbolo del programa es reconocido a través de una dirección que se le asigna en memoria, cuando esta dirección indica una conexión a un dispositivo E/S, es definida por el punto al cual el dispositivo esta físicamente conectado.

Formato de la lógica de control: Los contactos son programados en serie o paralelo, dependiendo de la lógica requerida por la aplicación. El número máximo de lazos en serie o paralelo es determinado por el fabricante del autómata.

Contactos de uso iterativo: Una señal puede ser utilizada reiteradamente dentro del programa, ya sea de entrada, salida o una señal interna del controlador.

La tabla 5.1 muestra la traslación de circuitos serie y paralelo a programación en contactos lógicos o diagrama de escalera, agregando la ecuación booleana y su equivalencia en listado de instrucciones.

Los circuitos serie son equivalentes a una operación booleana AND de tal forma que todas las entradas deben estar en posición de encendido para activar una salida. Los circuitos paralelo son equivalentes a una operación booleana OR, de tal forma que si por lo menos una entrada es encendida se activa la salida.

La evolución de lenguaje para la programación de circuitos en base al diagrama de escalera, introdujo el uso de más instrucciones con un mayor poder de alcances en los comandos de control. Con la evolución del software para controladores programables, nuevas instrucciones fueron agregándose para implementar instrucciones avanzadas como operaciones de tiempo y conteo en la programación en diagrama de escalera.

El nombre de "función" es utilizado para describir las operaciones que se ejecutan, como implica el termino que identifica la instrucción, tal sería el caso del manejo y transferencia de funciones de datos dentro de la memoria del controlador programable.

Estas instrucciones también están basadas en el principio de construcción de operaciones a base de relevadores, sin embargo las nuevas instrucciones tienen la capacidad de ejecutar funciones con un mayor grado de complejidad.

La adición de este tipo de instrucciones a la lógica elemental de escalera se clasifican en bloques y se conocen como bloques funcionales, pueden considerarse con un lenguaje de escalera aumentado.

En la figura 5.3 se muestra el ejemplo de una instrucción avanzada adicionada a las operaciones elementales de la programación en lenguaje de escalera. Como puede observarse la representación puede ser a través de un bloque funcional o una instrucción operativa encerrada entre símbolos de contactos.

El formato de las instrucciones depende exclusivamente de la arquitectura del autómatas, sin embargo la naturaleza de las instrucciones es similar en todos los controladores programables.

El lenguaje programación estándar, como se menciona antes es el de simbología de contactos, conocido también como diagrama de escalera o de contactos, prácticamente todos los PLC lo manejan por lo que se analizará con mayor profundidad que otros, como el Diagrama de Funciones, Grafcat, etc.

El diagrama de contactos o escalera clasifica sus instrucciones en dos tipos:

- * Instrucciones Básicas
- * Instrucciones Avanzadas.

Sin embargo, la forma en que se agrupan las instrucciones en estas categorías difiere de acuerdo al fabricante y al usuario, por lo que los límites entre una categoría y otra prácticamente no existen.

Existen normas de facto que marcan un estándar que busca asegurar a través del tiempo, el empleo de instrucciones clasificadas en un grupo determinado. La tabla 5.2 muestra clasificaciones típicas sobre un grupo de instrucciones específicas.

El conjunto de instrucciones básicas también se considera un lenguaje de bajo nivel y el conjunto de instrucciones avanzadas como un lenguaje de alto nivel. La definición de las instrucciones para programar un PLC se hace en base a las funciones del comando en cuestión, agrupándose como se muestra a continuación.

- * Relevadores básicos
- * Tiempo y Conteo
- * Programa/Flujo de control
- * Aritmética
- * Datos Manipulación/Manejo
- * Transferencia de Datos
- * Funciones especiales

INSTRUCCIONES BASICAS	INSTRUCCIONES AVANZADAS
CONTACTOS RELEVADORES	ARITMETICA DE DOBLE PRECISION
SALIDAS POR RELEVADORES	RAIZ CUADRADA
TEMPORIZADORES	CLASIFICADORES
CONTADORES	MOVIMIENTO DE REGISTROS
SEGUROS	MOVIMIENTO DE REGISTROS POR TABLA
OPERACIONES DE SALTO (GO TO)	FIFO
MCR	REGISTROS DE CORRIMIENTO
END (MCR)	REGISTROS ROTATIVOS
SUMA	BLOQUE DE DIAGNOSTICO
RESTA	BLOQUES DE TRANSFERENCIA (E/S)
MULTIPLICACION	ORDENADORES
DIVISION	PID
COMPARACIONES	REDES
MANEJO DE SUBRUTINAS (GO SUB)	MATRICES LOGICAS

TABLA 5.2: CLASIFICACION DE INSTRUCCIONES DEL PLC.

DIAGRAMA DE CONTACTOS.

El diagrama de contactos es un conjunto de símbolos que engloba las instrucciones para crear un programa de control en el autómatas. Las instrucciones del diagrama de escalera pueden ser enlazadas de forma que se obtenga la lógica de control que se desea grabar en memoria.

La función principal de un programa en diagrama de contactos es el control de señales de salida y la ejecución de operaciones funcionales basadas en las condiciones de entrada.

El control se realiza a través de las funciones requeridas en la secuencia de las etapas del proceso y las instrucciones que van conformando las acciones ejecutadas por el programa dadas las condiciones en cada etapa. Una etapa consiste en instrucciones condicionales representadas por un conjunto de contactos combinados en serie y/o paralelo.

Al cumplirse las condiciones contempladas en las instrucciones de entrada se origina la ejecución de una instrucción de salida. La figura 5.4 muestra la estructura elemental de una etapa en diagrama de escalera. Las condiciones se pueden dividir en condiciones de entrada, condiciones de etapa y condiciones de control lógico.

Una etapa del diagrama de contacto es activada mediante una salida o una instrucción funcional cuando la secuencia lógica es autorizada por una etapa previa.

La secuencia lógica se da cuando a un circuito se le suministra energía. La secuencia de etapas la determina la lógica de control establecida por el programa de aplicación para los posibles eventos contemplados en el proceso.

Normalmente el flujo de corriente indica un sentido de izquierda a derecha en el diagrama de contactos. en la figura 5.4, la línea L1 representa la fuente de energía y la línea L2 la terminal de activación conectada al sistema electromecánico, esto indica que el flujo de energía será de izquierda a derecha una vez que se den las condiciones de entrada de la etapa.

Los contactos serán abiertos o permanecerán cerrados de acuerdo al estado de las entradas de referencia (cada contacto tiene una dirección asignada en memoria en donde el autómata puede consultar en cualquier momento su estado lógico).

Cuando un bloque funcional es creado, las condiciones de entrada también son representadas por instrucciones de contactos manejando la lógica del bloque. El formato del bloque puede tener una o varias señales de salida que guardan el estado de la función que esta siendo ejecutada. El bloque funcional puede manejar una o varias entradas de habilitación que controlan las acciones del mismo.

En la figura 5.6 se muestra un bloque funcional con líneas de habilitación, cuando hay continuidad será activado el bloque para realizar su operación.

Si en el programa mostrado en la figura 5.6, la línea de habilitación es alimentada (encendida) por la lógica de control, se permite el flujo de corriente debido que se cumplen las condiciones de continuidad.

Dependiendo de las instrucciones del bloque funcional [figura 5.6] para las posibles condiciones, otras líneas de habilitación pueden regresar la salida a su condición original o activar otras salidas simultáneas.

En cambio el bloque funcional mostrado en la figura 5.7 crea una señal de salida activa durante la ejecución del programa, dado que no tiene condiciones para la alimentación de su línea de habilitación.

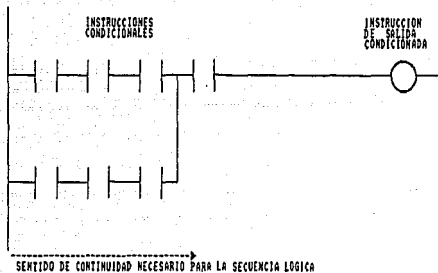


FIGURA 5.4.- ETAPA DE UN PROGRAMA EN DIAGRAMA DE CONTACTOS.

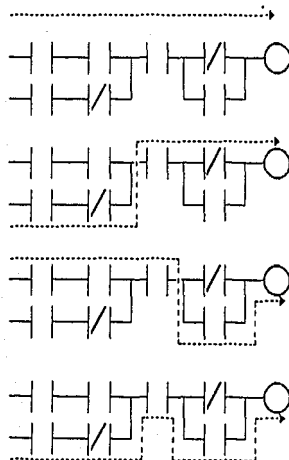


FIGURA 5.5.- POSIBLES LAZOS DE CONTINUIDAD QUE PUEDEN ACTIVAR UNA SALIDA ENERGIZADORA.

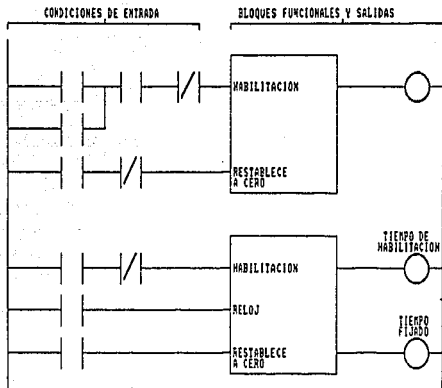


FIGURA 5.6.- INSTRUCCIONES EN FORMATO DE BLOQUES FUNCIONALES CON LÍNEAS DE HABILITACION.

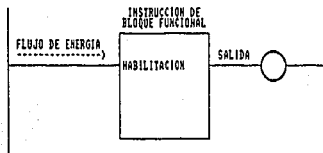


FIGURA 5.7.- BLOQUE FUNCIONAL INCONDICIONAL

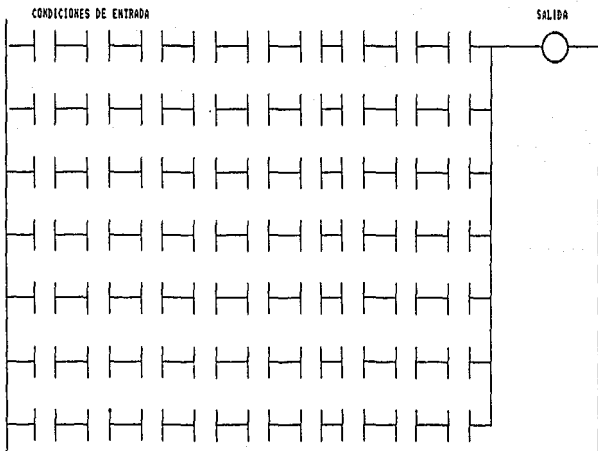


FIGURA 5.8.- MATRIZ DE ETAPAS ESCALONADAS.

El número máximo de contactos, normalmente es representado como se muestra en la figura 5.8 y su estructura se conoce como **matriz de etapas escalonadas**, el tamaño de esta matriz es restringido por el fabricante del PLC y por el aparato de programación utilizado.

El uso de una matriz de etapas escalonadas en bloques funcionales disminuye el número de contactos que pueden utilizarse [figura 5.9 (a)], de acuerdo al tamaño de la pantalla del aparato de programación. Si se utilizan funciones avanzadas.

Si se utilizan instrucciones avanzadas en un PLC en particular, las instrucciones pueden tomar varios espacios de los símbolos de contactos para mostrar la instrucción en pantalla [figura 5.9 (b)].

El dispositivo de programación generalmente muestra sobre pantalla todos los posibles lugares en donde se pueden colocar símbolos, permitiendo colocar los símbolos de la instrucción correspondiente en el lugar deseado.

Cada fabricante establece las reglas particulares de su producto de acuerdo a las capacidades que ofrece y las normas de facto establecidas por las instituciones encargadas de la normalización de los equipos industriales.

Una regla que todos los autómatas deben tener es el no permitir el flujo de corriente de inversa [figura 5.10] con el fin de prevenir pérdidas a través de circuitos que no deben ser alimentados, pudiendo alterar la función del alambrado dentro del sistema electromecánico.

Si la lógica de control provoca en algún punto un flujo inverso, se debe reprogramar el sistema. La figura 5.10 muestra una solución para una corriente inversa.

Los contactos y señales son los símbolos básicos de las instrucciones del diagrama de escalera. La programación de una etapa dada representa las condiciones que van a ser evaluadas en el orden establecido, para el control de la salida correspondiente a la etapa.

Cuando se programa, cada contacto o señal debe ser llamado con un número de dirección con el que es identificado para su evaluación y control. el llamado a través de la dirección localiza en la memoria de datos E/S el registro correspondiente al número asignado al contacto o señal.

En general un contacto representa una conexión E/S o una señal interna de salida, pudiendo ser utilizada durante la ejecución del programa cuando la lógica requiera la evaluación del estado del contacto almacenado en la memoria de datos.

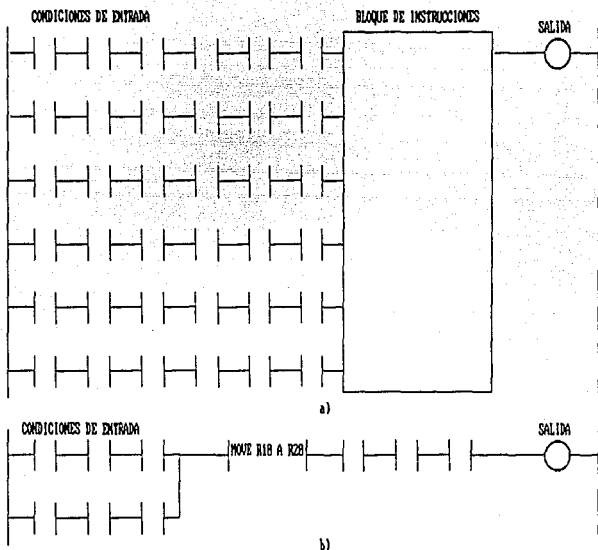


FIGURA 5.9.- USO DE LA MATRIZ ETAPAS ESCALONADAS CON INSTRUCCIONES AVANZADAS.

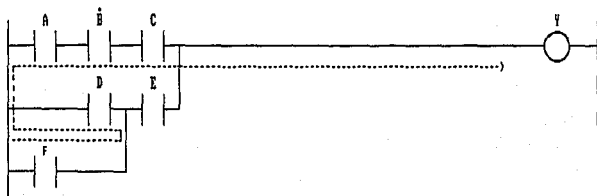


FIGURA 5.10.- NO SE PERMITE EL FLUJO DE CORRIENTE INVERSA.

La forma en que se distribuyen las etapas la dicta la lógica requerida por el proceso. Los contactos pueden conformarse en configuración serie, paralelo o paralelo/serie.

Cuando se cumple la secuencia lógica previa al contacto, se consideran presentes (verdaderas) las condiciones para el desarrollo de una etapa, si por el contrario no se cumple la secuencia la etapa no se activa por no cumplir con las condiciones establecidas en el programa (condiciones falsas).

Las instrucciones tipo relevador conforma las instrucciones más básicas en un controlador programable. Agregan algunas herramientas como son el alambrado de la lógica de relevadores, pero con una gran flexibilidad.

Estas instrucciones elementales habilitan el examen del estado falso/verdadero de un bit específico almacenado en memoria, mediante el resultado del examen se controla una salida externa o interna.

INSTRUCCIONES BASICAS.

ANÁLISIS DE ENCENDIDO-NORMALMENTE ABIERTO.

Generalmente se representa con el símbolo de un contacto normalmente abierto (NA), esta instrucción es programada para verificar la condición de encendido en la dirección que identifica el contacto. El llamado puede hacerse desde un dispositivo de entrada o desde un bit de salida en la sección de bits internos del área de almacenamiento o en la sección de memoria de salidas (ver Cap III, distribución de la memoria).

Si en la dirección mencionada, encuentra un "1" lógico (condición de encendido) el procesador, enviara la orden de cambiar la condición normal de "abierto" a la condición de "cerrado" permitiendo el flujo de energía a través del contacto.

Si el procesador se encuentra con un "0" lógico permitirá al contacto permanecer en su estado original (normalmente abierto) y seguirá suspendido el flujo de energía a través del contacto.

ANÁLISIS DE APAGADO, CONTACTO NORMALMENTE CERRADO —|/|— .

El análisis de la instrucción en condición de apagado, también es conocida como una instrucción que identifica un contacto normalmente cerrado., esta instrucción es programada cuando se requiere probar una condición de apagado en una dirección específica indicada en el programa de control.

La dirección puede ser indicada desde la memoria de entradas (dispositivos de entrada como detectores de posición, sensores, etc.) o desde la memoria de salidas.

Durante el monitoreo, el procesador examina en la dirección referenciada una condición de apagado. Si el contacto normalmente cerrado tiene un estado lógico "0" (condición de apagado) la instrucción generará las condiciones de continuidad en el circuito para la conexión o suministro de energía de la etapa consecuente del proceso.

Si por el contrario el estado lógico almacenado es "1" (condición de encendido), la instrucción indicará la apertura del contacto normalmente cerrado identificado por la dirección mencionada y el suministro de energía se suspenderá es decir, se desactiva la etapa consecuente del programa de control.

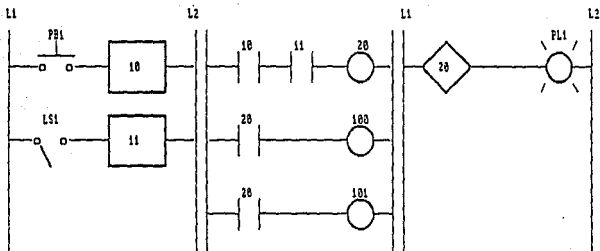
SEÑAL DE SALIDA ACTIVADA —O—, —()—.

La instrucción indica una señal de salida la cual es programada para controlar cualquier salida real conectada al PLC, mediante módulos interfase entre el equipo utilizado en el proceso y el controlador programable, también se utiliza en el control de salidas internas como las que manipulan los temporizadores.

La dirección utilizada por la señal de salida se encuentra en la memoria de almacenamiento (ejecutivo). Durante el monitoreo, el procesador evalúa todas las condiciones de entrada en el paso actual.

Si en cualquier rama, las condiciones cierran un circuito; el procesador dirigirá un uno lógico en la señal de salida que se indica en la dirección de referencia en el diagrama de escalera para el cierre del circuito el estado lógico "1" indica una condición de encendido para la instrucción de salida indicada al final de una etapa o paso de programa.

Si el número de identificación (referencia) corresponde al bit almacenado en el área de salidas reservada en la memoria de aplicación, el procesador dirigirá el bit obtenido en la dirección en una señal de encendido o apagado que controlará un dispositivo mediante un módulo interfase que manipula la alimentación de energía al equipo de proceso con la referencia.



NOTE: 20 ES UNA SALIDA REAL 100 Y 101 SON SALIDAS INTERNAS.

FIGURA 5.11.- MANEJO DE SEÑALES DE SALIDA MEDIANTE CONTACTOS NORMALMENTE ABIERTOS Y CONTACTOS NORMALMENTE CERRADOS.



Todas las señales de salida serán actualizadas al final de un ciclo de monitoreo, es decir; las condiciones indicadas en la etapa en análisis, indicará para la siguiente etapa acciones de encendido, apagado, cambio de posición, cierre o apertura de válvulas, etc.

Cuando la señal de salida condicionada por los lazos de control de una etapa, debe ser activada, por lo menos un lazo debe tener todos sus contactos cerrados al inicio del siguiente monitoreo para que se active la salida.

La actualización de la memoria de entradas y salidas se realiza al término de cada monitoreo, continua la secuencia de programa activándose el paso siguiente con las condiciones presentes generadas en la etapa anterior.

Cuando la dirección que identifica la señal de salida identifica un bit interno (señal interna) y la orden de ejecución es de encendido, en el registro de memoria correspondiente a la dirección se almacenará un uno lógico.

Este tipo de salidas es denominado como internas, siendo utilizadas para la intercomunicación de etapas y la secuencia de las mismas, además se utilizan cuando una salida real (rombo) no es necesaria físicamente y su indicación es fundamental para el control interno de programa.

Cuando la salida depende de contactos normalmente abiertos (NA)  o contactos normalmente cerrados (NC) , la apertura o cierre de los contactos dependerá exclusivamente de las condiciones de activación de la salida referenciada.

La figura 5.11 muestra un ejemplo sencillo de manejo de las señales de salida mediante contactos NA y NC, para que la salida etiquetada con el número 20 sea puesta en la condición de encendido, deben presentarse dos eventos: el interruptor PB1 etiquetado con el número 10 debe cambiar a la posición de encendido y el interruptor de límite LS1 identificado con la dirección de entrada 11, no debe ser activado.

Los bits que debe contener en las direcciones mencionadas la memoria de entrada son respectivamente 1 y 0, dado que el procesador analiza en la dirección de entrada 100 una condición de encendido y en la dirección de entrada 11 una condición de apagado. Cuando ambas condiciones se encuentren presentes en el paso analizado, la salida 20 será energizada, utilizándose como bandera de activación en los lazos siguientes.

El contacto NA referenciado con la etiqueta 20 se cerrará, encendiendo la salida interna con etiqueta 100, el contacto NC con la referencia 20 será abierto dado que al analizarse para una condición de apagado, la condición buscada no estará presente (la salida 20 esta encendida) por lo que la salida interna en la dirección 101 se apagará.

El control de la lámpara PL1 es una consecuencia de de la etapa anterior, cuando la salida 20 es alimentada, el procesador envía un uno lógico al módulo actuador que encenderá la lámpara piloto PL1.

Cuando la continuidad es suspendida por no encontrarse en las direcciones 10 y 11 las condiciones esperadas, la salida 100 no afecta ningún dispositivo real por ser una señal interna de control, es decir los bits de control que pueden afectar a los dispositivos de campo (reales) son los almacenados en la memoria de E/S en el área de aplicación.

SEÑAL DE SALIDA NEGADA —(/)— .

La instrucción que indica la negación (función lógica NOT) de una salida, esencialmente es lo contrario de una instrucción de una salida activada. si en la etapa analizada no existe continuidad, es decir, en todos los lazos existe por lo menos un contacto abierto, el bit que almacena el estado lógico de la salida referenciada deberá indicar una condición de encendido ("1" lógico).

Si existe por lo menos un lazo que tenga todos los contactos cerrados, es decir; existe continuidad en el paso, el bit que contiene la información de la dirección referenciada, deberá guardar una condición de apagado ("0" lógico).

Cuando se activa una instrucción de salida negada, el estado de los contactos de referencia debe ser cambiados: contactos NA cerrados y contactos NC abiertos, por el contrario cuando se desactiva una instrucción de salida negada, los contactos NA se abrirán y el estado de los contactos NC se cerraran.

Normalmente el uso de esta instrucción implica algunas artimañas de programación que facilitan la implementación de las expresiones lógicas que condicionan una salida, aplicando las reglas usuales del algebra de Boole, simplificando los lazos de control de la etapa programada.

RETENCION DE UNA SEÑAL DE SALIDA (LATCH) —(L)— .

La instrucción de retención de una señal de salida es programada cuando se requiere mantener una señal, aún cuando han desaparecido las que condiciones que generaron su encendido, es decir, la salida permanece encendida aunque los contactos que forman el lazo hayan vuelto a la condición que tenían antes del encendido perdiéndose la continuidad.

La salida permanecerá encendida mientras que el procesador no encuentre en la secuencia programada la instrucción que libera la señal de salida retenida, la función primordial de esta instrucción complementaria es la de apagar o desconectar la salida atrapada con la instrucción de retención de una señal de salida.

LIBERACION DE UNA SEÑAL DE SALIDA (UNLATCH).

La instrucción que libera una señal de salida se utiliza únicamente después de que se ha utilizado una instrucción de retención de señal, es decir, las instrucciones de retención y liberación de una señal de salida se usan normalmente en conjunto con la misma dirección.

La supresión del uso de la instrucción de liberación en el programa de control, implica que la señal retenida permanecerá encendida hasta que se suspenda el total de la energía al sistema.

El uso de la instrucción de liberación de una señal de salida normalmente genera un error de programación en un programa en el que previamente no se ha utilizado la instrucción de retención, la figura 5.12 ilustra el uso en conjunto de las instrucciones de retención y liberación de una señal de salida.

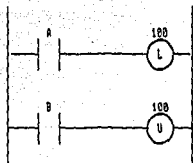
Las instrucciones de retención y liberación de una señal de salida pueden ser englobadas en forma de bloque como se muestra en la figura 5.13, la única diferencia es que la señal de liberación es ejecutada en la misma instrucción, cuando el lazo que activa la liberación tiene cerrados todos sus contactos, la señal de salida mantenida en posición de encendido es apagada.

SALIDA DE DISPARO UNICO (ONE SHOT) —(OS)—.

El uso de la instrucción de señal de salida de disparo único es similar al manejo de la salida común entre ramas, si en una etapa, algún lazo tiene cerrados todos los contactos que lo forman, hay continuidad por lo que la señal de disparo único es activada; no obstante, el tiempo de duración del flanco o pulso activado es menor o igual que el tiempo de monitoreo, dependiendo del lugar que la instrucción ocupa en la secuencia de programa.

Esta función de disparo único, normalmente es utilizada para regresar una condición a su estado original, se debe cuidar las etapas o bloques funcionales que pueden ser afectados por la acción de la instrucción de disparo único, dado que la programación de la instrucción implica la previa programación de la etapa que se pretende inhabilitar con el disparo único, la figura 5.14 muestra una salida de disparo único y su correspondiente diagrama de tiempos.

Dependiendo del autómata utilizado, puede manejarse el flanco positivo o el flanco negativo de la entrada A como marca de disparo. El primer pulso de la entrada A, pone la salida Yos en "1" durante un tiempo de monitoreo en la transición de apagado a encendido (flanco positivo). el segundo pulso de la señal de entrada A pone la salida Yos en "1" durante el tiempo de un monitoreo en la señal de transición de encendido a apagado (flanco negativo).



SI EL CONTACTO A CIERRA, ACTIVA LA SALIDA 100 HASTA QUE EL CONTACTO B CIERRE Y DESCONECTE LA SENAL 100.

FIGURA 5.12.- ETAPA DE UN PROGRAMA QUE UTILIZA LAS INSTRUCCIONES DE MANTENCIÓN Y LIBERACION DE UNA SENAL DE SALIDA.

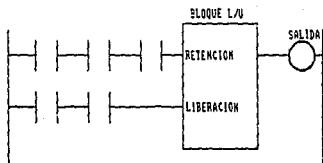
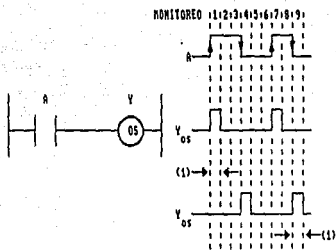
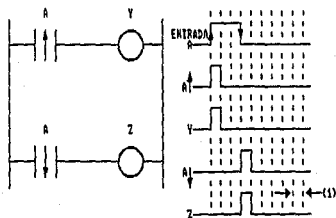


FIGURA 5.13.- BLOQUE DE INSTRUCCIONES PARA MANTENIR Y LIBERAR UNA SENAL DE SALIDA.



NOTA: TIEMPO DE DURACION DE UN MONITOREO.

FIGURA 5.14 IMPLEMENTACION DE UN DISPARO UNICO Y SU DIAGRAMA DE TIEMPO.



NOTA: TIEMPO DE DURACION DE UN MONITOREO.

FIGURA 5.15.- INSTRUCCION CONTACTO DE TRANSICION Y SU DIAGRAMA DE TIEMPO.

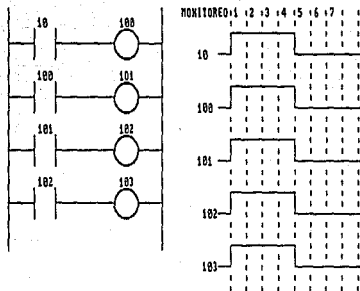


FIGURA 5.16

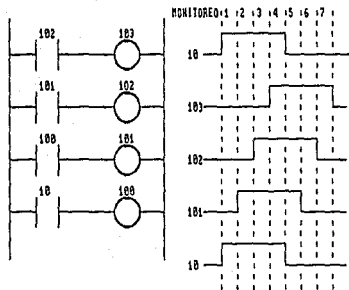


FIGURA 5.17

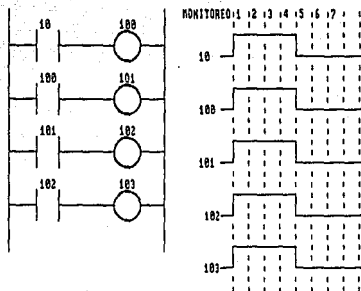


FIGURA 5.16

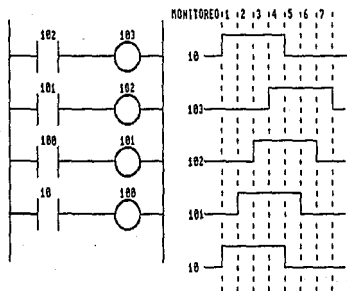


FIGURA 5.17

CONTACTO DE TRANSICION



La instrucción de un contacto de transición es programada para suministrar un pulso de disparo, tomando como marca de disparo el flanco positivo o negativo, es decir una transición de encendido a apagado y viceversa.

Este contacto cerrara exactamente lo que dure el ciclo de monitoreo, la señal de disparo la dará la transición permitiendo la secuencia lógica durante un ciclo de monitoreo, tiempo en el cual la señal de disparo esta encendida.

La señal de disparo se activara nuevamente con un flanco positivo cerrando una vez más el contacto de transición. La dirección asignada al contacto puede representar E/S externas o internas.

Normalmente la instrucción de disparo mediante un contacto de transición es suministrada por el software de un PLC, ya que el controlador por sí mismo no proporciona ningún pulso similar de disparo a través de su hardware.

El contacto de transición se diferencia con la instrucción de salida de disparo único en que esta última se utiliza para regresar a su posición original contactos controlados por las instrucciones como retención y liberación de una señal de salida. La figura 5.15 muestra instrucciones programadas para los flancos positivo y negativo.

EVALUACION POR MONITOREO CONTINUO.

La evaluación en el programa de control es un concepto fundamental, el cual se aplica en el orden en que se ejecutan las instrucciones programadas. Como se menciona con anterioridad el programa inicia su ejecución siempre con una primera evaluación del estado lógico (status) de los dispositivos que condicionan el arranque del sistema.

Dicha tarea es ejecutada por el procesador que hace la evaluación mediante los datos que almacenan su estado actual en la memoria de entradas comparándolos con las condiciones de arranque que al cumplirse dan inicio a la secuencia programada.

La resolución tomada por el procesador parte de la primera etapa o "rung", en la cual el procesador examina en la dirección programada el dato o condición que determina la secuencia lógica en los lazos de control de la etapa analizada.

Es importante enfatizar el hecho de que el procesador no puede por sí mismo regresar a una etapa anterior para modificar estados lógicos que afectan una etapa actualmente en análisis.

En el diagrama mostrado en la figura 5.16 el contacto NA de referencia 10 el cual representa un interruptor de botón, al ser encendido, se activará o encenderá la salida con etiqueta 100 la cual dará vía a la segunda etapa encendiendo la salida 101, la cual cerrará el contacto con la misma referencia encendiendo la salida con dirección 102 la cual al cerrar el contacto con la misma etiqueta encenderá la salida 103.

Todas estas salidas son encendidas durante el mismo monitoreo y cuando el ciclo de este termina estas condiciones son "trasladadas" a los dispositivos reales de salida conectados a los módulos interfase. En el caso del circuito de control mostrado en la figura 5.16 si los dispositivos reales se trataran de lámparas piloto, el diagrama de tiempo representativo del estado lógico de las mismas estaría representado a la derecha del diagrama de contactos.

Otro ejemplo de la respuesta del procesador a la posición de las etiquetas, se muestra en la figura 5.17 en donde el diagrama de contactos es básicamente el mismo con la diferencia que las etiquetas asignadas se han invertido.

Para el caso de la figura 5.17 se supone que si se oprime el botón con referencia 10 en un primer proceso de monitoreo, el procesador analizará en el mismo ciclo las referencias 102, 101 y 100 dejando abiertos los interruptores y manteniendo desactivadas las salidas 103, 102 y 101 correspondientes a los interruptores.

En el segundo monitoreo en la memoria de entradas en la dirección 10 ya se encuentra un uno lógico por lo que el procesador en las referencias 102 y 101 encuentra y mantiene la condición de apagado manteniendo desactivadas las salidas 103 y 102, el uno lógico de la dirección 10 indica el encendido de la salida 100 y consecuentemente se cierra el contacto correspondiente encendiendo la salida 101.

En el tercer monitoreo se encontrarán las salidas 100 y 101 encendidas en la memoria de entradas así que el contacto 102 y la salida 103 se mantendrán en la misma situación mientras que el contacto con etiqueta 101 se cierra y se enciende la salida 102.

En el cuarto monitoreo el procesador encuentra en la memoria de entradas un uno lógico en la referencia 102 por lo que activará la salida 103.

La operación física del circuito de la figura 5.18 es imposible de observar dado que la duración de un monitoreo es del orden de milisegundos, por lo que las lámparas piloto parecerán encendidas al mismo tiempo; por ello para tener la seguridad del comportamiento de los lazos de control, se debe analizar el diagrama de tiempo de la respuesta del sistema.

En un "pre-monitoreo" se "leen" las entradas, ejecutándose la lógica programada que activa las salidas que arrancan, paran o inhiben el sistema de control hasta el siguiente monitoreo. En general este "pre-monitoreo" es utilizado en las pruebas del programa de control.

Es importante recordar que si se desea una salida que repercute sobre una etapa consecutiva, la repercusión se dará si al finalizar el monitoreo de la etapa presente y el levantamiento de los datos de entrada generados en la misma, el monitoreo siguiente los lee y ejecuta la acción correspondiente en aquellos contactos con el mismo número de referencia de la salida generada.

La orden de la ejecución puede darse en forma particular mediante contactos de transición o salidas de disparo único para liberar una señal retenida y permitir su retorno a otra etapa.

PROGRAMACION DE UNA ENTRADA NORMALMENTE CERRADA.

Existen varias formas de manejar señales externas desde los dispositivos conectados al autómata, entre estas se encuentra la condición normalmente-cerrada, la cual tiene un buen fundamento.

La programación de una entrada normalmente-cerrada para determinado dispositivo puede resultar obvia para el personal acostumbrado al manejo de este tipo de señales, sin embargo el la forma correcta en que debe utilizarse en un sistema de control presenta cierta dificultad en su concepción inicial dentro de la secuencia lógica del programa de control, puesto que el equipo debe acoplarse a la señal normalmente-cerrada de acuerdo al tiempo de reacción y comportamiento que presenta a las señales de excitación.

El siguiente ejemplo pretende ilustrar este problema. El diagrama eléctrico mostrado en la figura 5.18 se trasladará a un diagrama de contactos que realice la misma función, al implementar la misma lógica, la lámpara piloto PL1 debe comportarse en forma idéntica.

Cuando el interruptor PB1 no esta oprimido, PL1 debe estar encendida, en cambio si PB1 es oprimido la lámpara PL1 debe apagarse. Las figuras 5.19 y 5.20 muestran dos posibles métodos de programar PL1 e implementar su lógica de control.

A primera vista podría parecer la solución el diagrama mostrado en la figura 5.19, si se observa con cuidado se encuentra programada con la referencia del interruptor PB1 (10) un contacto normalmente-cerrado, el cual maneja la señal de salida real 100 conectada a la lámpara piloto PL1.

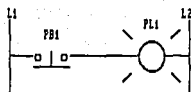


FIGURA 5.18.- DIAGRAMA ELECTRICO

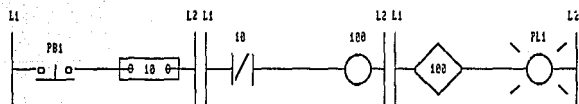


FIGURA 5.19 PROGRAMACION UTILIZANDO UN CONTACTO NC.

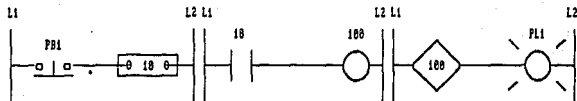


FIGURA 5.20.- PROGRAMACION UTILIZANDO UN CONTACTO NA.

Cuando el autómata arranca, el procesador lee el estado de los dispositivos de entrada conectados a la entrada 10 y almacena la información encontrada en la memoria de entradas.

Si en un diagrama de contactos como el de la figura 5.19 PBI no se oprime, en la dirección 10 se leerá un uno lógico (flujo de corriente al módulo).

Durante la ejecución de la secuencia lógica se evaluará la condición de apagado y en ese momento la dirección 10 será puesta en uno, el contacto normalmente cerrado será abierto y la continuidad cesará.

La salida 100 es apagada al igual que la lámpara PL1. Si PBI es oprimido en la memoria de entradas se almacenará un cero lógico en la referencia 10 (se corta la corriente al módulo). La condición de apagado en la referencia 10 será lo esperado, por lo que una instrucción indica el suministro de energía a la salida 100 y PL1.

En la figura 5.20, la condición de entrada debe ser programada en base a una condición de encendido. Durante la operación, si PBI no se oprime la dirección 10 de entrada al módulo será leída con un estado de encendido.

Cuando el lazo es evaluado se encontrara verdadero el dato almacenado en la memoria de entradas, por lo que el suministro de energía se realizará al abrirse el contacto con la referencia 10, se encenderán la salida 100 y la lámpara piloto PL1.

Si en la figura 5.20 PBI es oprimido el módulo tendrá un estado de apagado y el procesador almacenará un cero lógico en la dirección 10 de la memoria de entradas.

Durante la evaluación de la etapa, el análisis de la dirección 10 para una condición de encendido no será verdadera y no habrá continuidad por que los contactos referenciados estarán abiertos, por lo que la salida 100 y PL1 estarán apagadas.

De la solución programada para la conexión de una entrada normalmente-cerrada en la figura 5.20 se concluye:

Si se desea alambrear la entrada a un dispositivo, normalmente-cerrada, cuando conecte, se comportará como un lazo continuo o normalmente-cerrado conectado al dispositivo de entrada, se debe programar con un análisis de encendido o instrucción de contacto normalmente-abierto.

Las entradas discretas al PLC se estructuraran como normalmente-abiertas o normalmente-cerradas a pesar de su configuración original.

Esta capacidad para analizar un único dispositivo para cualquier estado (NA o NC) es la clave para la flexibilidad del autómata programable, sin embargo se hace la configuración del alambrado de forma que el programa de control manipule la acción deseada con los cambios de estado de los contactos.

En la mayoría de los casos una entrada a un dispositivo alambrada en forma NC es programada como NA en el contacto del PLC. Hay que recordar que la forma de programación o alambrado de una entrada no sólo depende de su estado normal sino también de la acción que se desea del autómeta.

INSTRUCCIONES DE TEMPORIZACION Y CONTEO.

Las funciones de temporización y conteo son instrucciones internas que se controlan temporizadores y/o contadores que forman parte del hardware del PLC, la temporización y conteo se utiliza en el encendido o apagado de dispositivos después de un tiempo programado o al alcanzar un valor establecido (cuenta).

La temporización y conteo son funciones que en el análisis del programa de control, normalmente se consideran como señales de salida internas que se utilizan fundamentalmente para establecer la secuencia del diagrama de escalera.

Los temporizadores pueden tener uno o más tiempos base para controlar el tiempo de un evento. El rango de trabajo puede variar entre un centésimo de segundo y un segundo, los tiempos bases típicos son de 0.01, 0.1 y 1 segundo aunque pueden variar de acuerdo al fabricante, pudiéndose encontrar tiempos bases de hasta 1 minuto.

La aplicación de los temporizadores pueden considerarse prácticamente infinita dada la variedad de todos los sistemas que se podrían controlar mediante un PLC, de cualquier forma estos dispositivos pueden ser agregados en forma externa al autómeta para el control específico de un proceso que requiere de alguna función de retardo o control de tiempo.

Los contadores generalmente son utilizados para la cuenta de eventos iterativos como puede ser un producto terminado transportado en una banda para su empaque. Cuando se alcanza el valor fijado en el programa dispositivos como un selenoide pueden ser encendidos para que se lleve a cabo el paro del dispositivo de transporte.

Los valores de temporizadores y contadores (T/C) son dos: un valor fijo y un valor acumulado, estos valores son almacenados en la localidad correspondiente en la memoria de entradas.

Los temporizadores y contadores pueden ser implementados a través de las funciones elementales (AND, OR Y NOT) que forman parte del bloque de instrucciones básicas de la simbología de contactos.

TEMPORIZADORES.

La función de temporización puede ser fijada desde el circuito del sistema de control o establecerse por instrucciones de temporización programables desde el PLC, que pueden ofrecer temporizaciones con diferentes características de acuerdo a las posibilidades del fabricante y las necesidades del industrial.

A continuación se mencionan algunos tipos de temporizadores disponibles en los PLC comerciales.

- * Temporizadores de encendido con retardo a la conexión.
- * Temporizadores de encendido con retardo de la desconexión.
- * Temporizadores de apagado con retardo a la conexión.
- * Temporizadores de apagado con retardo de la desconexión.
- * Temporizadores retentivos o inhibidores.

La funcionalidad de un tipo determinado de temporizador depende esencialmente de la diferencia que representa el tipo de salida que ofrece para cubrir una necesidad.

La figura 5.21 ilustra dos formas comunes de temporizadores. El formato de bloque puede tener una o dos entradas dependiendo del autómeta. Estas entradas normalmente son niveles lógicos que habilitan o restauran condiciones determinadas en un lazo de control.

Cuando los dispositivos que conforman la línea de continuidad la habilitan y la línea de activación del temporizador es alimentada se activa la función de temporización el formato en serie normalmente trabaja con una línea de habilitación que al generarse la continuidad activa el temporizador.

Comúnmente para ambos formatos de temporizadores es utilizado un registro X, donde X constituye la dirección en donde esta fijo un valor y un registro Y, en donde Y es la dirección en que se almacena el valor del tiempo acumulado, estos valores se estarán comparando, al igualarse se ejecuta la acción programada al finalizar el intervalo de tiempo fijado.

El tiempo base puede ser seleccionado de acuerdo al autómeta que se este utilizando (0.01 seg, 0.1 seg, 1.0 seg, etc.) y a la resolución que tenga el temporizador.

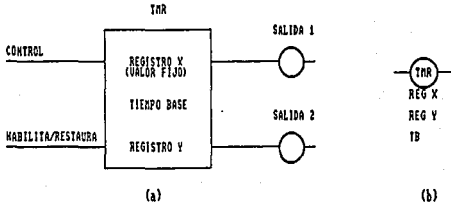
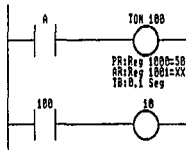


FIGURA 5.21.- FORMATOS DE TEMPORIZACION: (a) BLOQUE FUNCIONAL Y (b) SEÑAL DE SALIDA



LA SALIDA 100 ES ENCENDIDA 5 SEG DESPUES DE QUE EL CONTACTO A CIERRA

FIGURA 5.22.- TEMPORIZADORES DE RETARDO DE ENCENDIDO DEL SUMINISTRO DE ENERGIA.

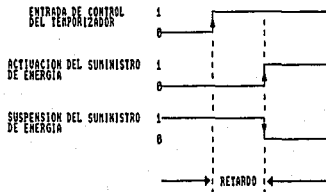
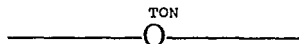


FIGURA 5.23.- DIAGRAMAS DE TIEMPO PARA LOS TEMPORIZADORES DE SUMINISTRO DE LA ENERGIA.

TEMPORIZADORES DE ENCENDIDO CON RETARDO A LA CONEXION.

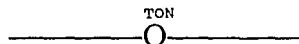


Esta instrucción es programada para dar un tiempo de retardo en la ejecución de una acción o para medir la duración de un evento que esta llevándose a cabo.

Una vez que se da la continuidad, el temporizador inicia la cuenta de los intervalos de tiempo-base (flancos) hasta que la cuenta es igualada con el valor fijado.

Al darse las condiciones de igualdad en el temporizador su salida es conectada y el contacto de duración controlada asociado con la salida es cerrado (ver figura 5.22). Los contactos de duración controlada pueden utilizarse en el programa como NA o NC. Si la secuencia lógica se pierde antes el registro acumulador del contacto de duración controlada es puesto a cero.

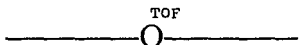
TEMPORIZADORES DE ENCENDIDO CON RETARDO A LA DESCONEXION.



Este tipo de temporizador es similar al anterior, la diferencia consiste en que su salida esta conectada y una vez que la continuidad en el lazo se presenta y transcurre el tiempo programado la salida se desconecta (es desconectada).

Normalmente la manufactura del PLC proporciona un temporizador con retardo a la conexión y desconexión de un circuito, sin embargo en cualquier manufactura utilizando las instrucciones básicas puede implementarse cualquiera de estos tipos. La figura 5.23 muestra los diagramas de tiempo para estos temporizadores

TEMPORIZADORES DE APAGADO CON RETARDO A LA CONEXION.



Sirve para retardar la ejecución de una acción. Si el lazo de control no tiene continuidad, el temporizador inicia el tiempo de retardo hasta que se iguala el valor acumulado con el valor fijado.

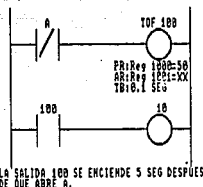


FIGURA 5.24.- TEMPORIZADORES DE RETARDO A LA DESCONEXION DEL SUMINISTRO DE ENERGIA.

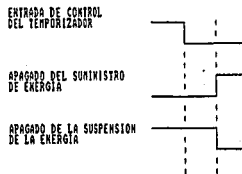
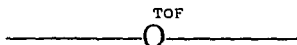


FIGURA 5.25.- DIAGRAMA DE TIEMPO DE LOS TEMPORIZADORES DE APAGADO.

Cuando el valor fijado es igual al valor acumulado, el contacto de duración controlada asociado con la salida es cerrado [figura 5.24]. El contacto de duración controlada puede utilizarse en el programa como un contacto NA o NC. Si la secuencia lógica se presenta antes de que se alcance el valor fijado, en el temporizador el contacto de duración limitada deja de operar y el acumulador es puesto a cero. La figura 5.25 muestra el diagrama de tiempo representativo de este tipo temporización.

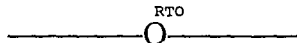
TEMPORIZADORES DE APAGADO CON RETARDO A LA DESCONEXION.



La diferencia de este temporizador con el anterior es que la desconexión se realiza cuando en la línea se pierde la continuidad, el intervalo de tiempo ha transcurrido o los valores del acumulador y el valor fijado son idénticos.

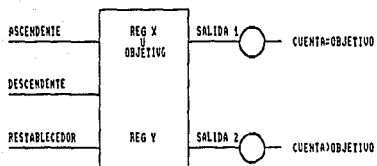
Al igual que los temporizadores de encendido el fabricante comúnmente proporciona sólo un tipo de este temporizador de retardo, ya sea el de conexión o desconexión, pudiéndose implementar el complemento con el bloque de instrucciones básicas.

TEMPORIZADORES RETENTIVOS O INHIBIDORES.

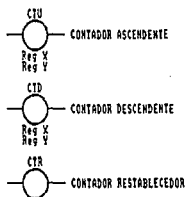


Estos temporizadores se programan si se necesita mantener un valor acumulado, si se pierden la continuidad o el sistema de alimentación. Si en cualquier lazo se genera continuidad el temporizador inicia la cuenta de los intervalos de tiempo base hasta que el acumulador iguale el tiempo fijado.

Cuando se igualan los tiempos acumulado y fijado, la salida del temporizador es activada y el contacto de duración controlada asociado con la salida es puesto en posición de encendido. La programación del contacto se puede hacer considerándolo NA o NC y se pone a cero con la siguiente instrucción.



a) FORMATO DE BLOQUE FUNCIONAL



b) FORMATO DE SEÑAL DE SALIDA

FIGURA 5.26.- FORMATOS TÍPICOS DE CONTADORES.

RESTAURACION DE UN TEMPORIZADOR RETENTIVO O INHIBIDOR.

RTR



Esta instrucción limpia el valor del acumulador de un temporizador inhibidor si en cualquier lazo en la etapa se genera continuidad el acumulador del temporizador con la misma referencia será puesto a cero.

CONTADORES.

Básicamente se fabrican dos tipos de contadores, uno que cuenta en forma ascendente y otro que cuenta en forma descendente. El formato varía con el fabricante, algunos se manejan como un elemento serie más en el lazo (como una señal de salida) y otros presentan un formato de bloque, ambos tipos se muestran en la figura 5.26.

CONTADOR ASCENDENTE.

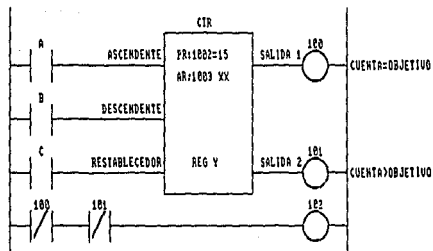
CTU



El contador ascendente incrementa en uno el acumulador en cada intervalo de tiempo contando generalmente algún evento, se utiliza para encender o apagar algún dispositivo cuando se alcanza una cuenta determinada, al igual que en los temporizadores el contador tiene dos registros uno X en el que se almacena un valor prefijado y un registro Y en el cual se acumula el último valor contado cuando se pasa del estado cero al estado uno (flanco positivo).

Cuando los dos registros del contador tienen el mismo valor la salida del contador es puesta en 1, el proceso de conteo termina y el contacto asociado con la salida referenciada es cerrado.

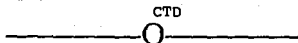
Dependiendo del fabricante se tienen contadores con la opción de restaurar automáticamente el valor de cero o continuar con la cuenta caso en el cual se debe utilizar la entrada de restauración de cero en el acumulador.



EL CONTADOR CUENTA EN FORMA ASCENDENTE CUANDO A CIERRA,
 CUENTA DESCENDIENTEMENTE CUANDO B CIERRA Y RESTABLECE EL
 CERO CUANDO CIERRA C.

FIGURA 5.27.- INSTRUCCION DE CUENTA ASCENDENTE/DESCENDENTE EN FORMATO DE BLOQUE.

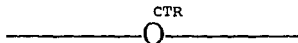
CONTADOR DESCENDENTE.



El contador descendente decrementa en una unidad el acumulador en cada intervalo de tiempo cuando ocurre cierto evento.

Normalmente la utilización de los contadores ascendente y descendente se combinan formando uno ascendente/descendente, dado que normalmente se utilizan con el mismo número de referencia. En forma común el contador descendente se utiliza para corrección de un dato de más por el contador ascendente.

CONTADOR RESTAURADOR.



La función principal de esta instrucción es la de restaurar el valor inicial de los contadores ascendente y descendente [figura 5.27].

INSTRUCCIONES DE CONTROL DE PROGRAMA Y FLUJO DE OPERACIONES.

Las instrucciones de control de programa son utilizadas directamente con el flujo de operaciones y ejecución de instrucciones dentro de la secuencia de un programa.

Estas operaciones se realizan, agrupando las instrucciones de control de programa formando un bloque de instrucciones (rutina) dentro del programa .

Estos bloques de instrucciones engloban zonas de instrucciones en serie que son utilizadas para implementar una función necesaria para el desarrollo del proceso controlado, estas zonas se pueden clasificar como rutinas (ver figura 5.29) y pueden ser llamadas por un programa maestro para su ejecución dentro del programa que opera mediante una secuencia lógica establecida en el diagrama de escalera.

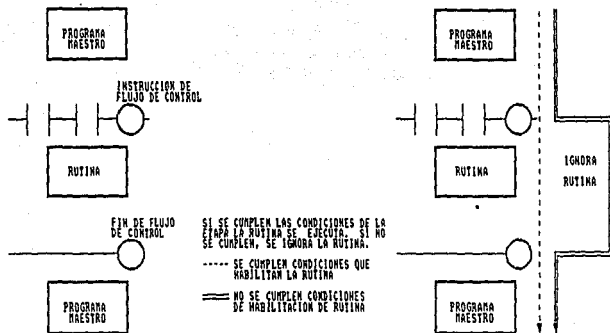


FIGURA 5.28.- EJEMPLO DEL USO DE UNA ROUTINA EN UN PROGRAMA PRINCIPAL.

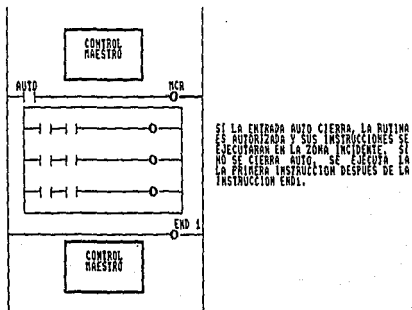


FIGURA 5.29.- APLICACION DE LA INSTRUCCION MCR: RELEVADOR DE CONTROL MAESTRO.

La tabla 5.3 muestra las instrucciones más comunes en el control de flujo de operaciones, estas normalmente se utilizan en parejas.

INSTRUCCION	NOMBRE
MCR	RELEVADOR DE CONTROL MAESTRO
ZCL	CONTROL DE ZONA (ULTIMO ESTADO)
JMP (GO TO)	SALTO
GOSUB (JSB)	SALTO A RUTINA
LBL	NUMERO DE RUTINA
RET	FIN DE RUTINA-RETORNO
END	FIN (MCR O ZCL)

Tabla 5.3: Tipos más comunes de instrucciones de control de flujo.

RELEVADOR DE CONTROL MAESTRO (MCR).

MCR



La instrucción MCR permite el uso de una rutina cuando se programa inmediatamente antes de la misma y las condiciones que autorizan la rutina están presentes. La rutina se ejecutará en forma secuencial hasta encontrar la instrucción END que indica el fin de la zona de instrucciones [figura 5.29].

Sí las condiciones que activan la rutina no están presentes el procesador saltará automáticamente la zona de instrucciones que conforman la rutina y ejecutará la instrucción que sigue a la de END continuando con la secuencia lógica.

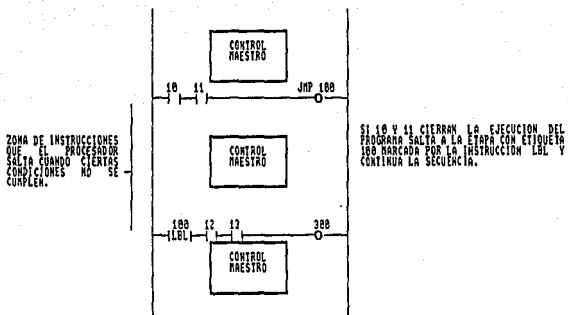
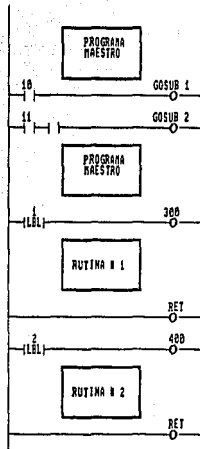
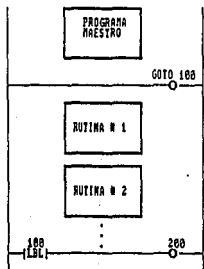


FIGURA 5.30.- APLICACION DE LA INSTRUCCION JMP.



SI EL CONTACTO 10 CIERRA, SE EJECUTA LA RUTINA 1. LA INSTRUCCION RET INDICA AL PROCESADOR EL RETORNO A LA INSTRUCCION SIGUIENTE A LA INSTRUCCION GOSUB 1. SI CIERRA EL CONTACTO 11, LA INSTRUCCION GOSUB 2 ORDENA LA EJECUCION DE LA RUTINA 2.

FIGURA 5.31.- AUTOMATA CON ZONA DE RUTINAS AL FINAL DEL PROGRAMA DE PROGRAMA DE CONTROL.



LA INSTRUCCION INCONDICIONAL GOTO 100 ENVIA AL MICROPROCESADOR A LA INSTRUCCION LBL CON LA MISMA ETIQUETA ULTIMA INSTRUCCION DEL PROGRAMA. CADA RUTINA INICIA Y TERMINA CON LAS INSTRUCCIONES LBL Y RET RESPECTIVAMENTE.

EL FIN DEL MONITOREO OCURRE CUANDO EL PROCESADOR ENCUENTRA LA SALIDA 200.

FIGURA 5.32.- CREACION DE UN AREA DE RUTINAS POR EL USUARIO.

CONTROL DE ZONA DE ULTIMO ESTADO (ZCL).

ZCL



Su funcionamiento es similar a la instrucción MCR, se diferencia por el hecho de que cuando es activada la salida de la zona esta controlada por las propias instrucciones de la rutina.

si no se presentan las condiciones para la activación los parámetros que se generan en la zona mantendrán su último estado lógico.

SALTO (JMP).

JMP



La operación de salto (JUMP TO) es una instrucción que permite cambiar la secuencia del programa de control si ciertas condiciones se presentan, la señal de salto, JMP debe tener una dirección de referencia que indica al procesador la próxima instrucción a ejecutar.

La figura 5.30 muestra el formato con que se programa la operación de salto y su combinación con el uso de la instrucción LBL (Label) que "etiqueta" o direcciona el lugar en donde continuará la secuencia.

La instrucción JMP permite dirigir la secuencia de operaciones hacia una etapa determinada del proceso que requiere de inmediata atención.

SALTO A UNA RUTINA (GOSUB).

GOSUB



La instrucción GOSUB (GO TO SUBROUTINE) permite alterar la secuencia del programa, cuando condiciones presentes requieren de la ejecución de una rutina. La dirección es indicada por la instrucción LBL al procesador que a continuación ejecuta la zona de instrucciones que engloba la rutina. La instrucción GOSUB también puede ser llamada JSB.

El procesador ejecuta las instrucciones hasta que encuentra el límite de la misma que es fijado por la instrucción RET (RETURN) que regresa continuar la secuencia a partir de donde fue requerida la rutina. Cada rutina se inicia con la instrucción LBL y debe finalizar con la instrucción RET (figura 5.31).

La instrucción GOSUB es útil cuando las condiciones presentes en el proceso requieren realizar una rutina, generalmente todas las rutinas son programadas al final del programa de control, es conveniente que se ordenen de acuerdo a las direcciones asignadas.

Algunos autómatas tienen asignada de fabrica un área para rutinas, sin embargo cualquiera permite la creación de un área de rutinas utilizando las instrucciones básicas.

NUMERO DE RUTINA (LBL).

—| LBL |—

Esta instrucción es utilizada para identificar una zona de instrucciones, podríamos decir que es la clave que activa una rutina, indica al procesador el lugar del programa en donde continua la secuencia cuando se requiere de una rutina determinada por las instrucciones JMP y GOSUB.

La dirección que marca una instrucción LBL únicamente puede utilizarse una vez en el programa.

SEÑAL DE RETORNO (RET).

RET
—○—

La instrucción RET es utilizada únicamente para finalizar una rutina. Su programación es con entradas incondicionales, cuando el procesador encuentra una instrucción RET el control es regresado al programa maestro y el procesador ejecuta la instrucción siguiente a la instrucción de salto continuando con la secuencia lógica a partir de ese punto.

FIN (END).

La instrucción END se utiliza para indicar al procesador el fin de un bloque de instrucciones activado por las instrucciones MCR y ZCL, normalmente se utiliza en forma incondicional. La existencia de la instrucción END dentro de las instrucciones de un autómata dependen del fabricante que puede ofrecer otras opciones para el manejo de rutinas en serie con la secuencia de programa.

INSTRUCCIONES ARITMETICAS

Las instrucciones aritméticas implementan operaciones matemáticas básicas cuando algunos procesos requieren de ciertos cálculos utilizando los registros o palabras localizadas en memoria que identifican aquellos valores que se vienen generando en el proceso y con los cuales se deben realizar las operaciones aritméticas.

La representación gráfica de las instrucciones aritméticas y formato numérico varía de acuerdo a la manufactura del PLC.

El formato numérico puede utilizar de 3 a 5 dígitos en código BCD o binario, cuando se necesita el empleo de cantidades muy largas, se cuenta con el manejo de doble precisión cuyo alcance es limitado por el fabricante del equipo.

SEÑAL DE SUMA (ADD).

ADD



Esta instrucción ejecuta la suma de dos valores almacenados en las localidades de memoria direccionadas, la forma como estos valores son accedidos depende del controlador programable, algunos utilizan instrucciones de levantamiento de datos (GET data) para llamar los registros que contienen los operandos, una forma común de programación se muestra en la figura 5.34. El resultado es almacenado en el registro indicado en la señal de suma.

Sí la operación de suma es solicitada cuando se cumplen las condiciones de un lazo de control, se deben tener los valores antes de la instrucción de suma. cuando existe desbordamiento del resultado, usualmente se señala en un bit del registro que almacena el resultado de la suma.

BLOQUE DE SUMA (ADD).

Esta instrucción suma dos valores almacenados y guarda el resultado en un registro específico. Los valores sumados pueden ser constantes de punto fijo, valores contenidos en E/S o variables numéricas almacenadas en cualquier localidad de memoria.

La línea de control habilita la adición en el bloque, cuando se cumplen las condiciones de la etapa.

En este bloque el registro X y el registro Y toman los valores de los operandos indicados por el procesador de acuerdo a las direcciones indicadas en el programa de control, ambos registros son sumados y almacenados en el registro Z (ver figura 5.35).

La salida del bloque se enciende cuando en el resultado se presenta un desbordamiento, algunos PLC limitan el resultado a la capacidad de registro en que almacena el resultado otros almacenan la diferencia del resultado desbordado y el último valor calculado.

SEÑAL DE RESTA O SUBSTRACCION (SUB).

SUB



La resta o substracción realiza la diferencia entre dos registros almacenados, al igual que en la suma sí en una etapa se habilita la operación se ejecuta la operación de resta cargando los datos previamente de los registros referenciados en la instrucción GET.

El registro en donde se almacena el resultado reserva un bit para indicar un resultado negativo. La figura 5.36 muestra un lazo en donde se solicita una substracción.

BLOQUE DE SUBSTRACCION (SUB).

El funcionamiento es similar que en el bloque de suma, cuando se requiere una operación de resta la entrada de control es alimentada con un uno lógico con lo cual la operación de substracción de los registros X y Y ($X - Y$) es ejecutada almacenando el resultado en un registro Z.

Los formatos en que se constituye la información en los registros es variable algunos no reservan un bit para la indicación de signo, en cambio manejan tres salidas que indican " $X > Y$ ", " $X = Y$ " y " $X < Y$ ". Este tipo de bloques esencialmente ejecuta la comparación de dos registros.

Algunas manufacturas permiten la substracción de una constante, incluyéndola en el hardware del bloque de substracción. Se pueden utilizar bloques de substracción que se pueden utilizar para leer E/S analógicas o multibit [figura 5.38].

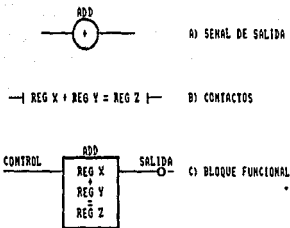
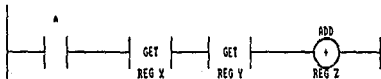


FIGURA 5.33 TIPOS DE FORMATO UTILIZADOS EN LA INSTRUCCION DE SUMA.



SI EL CONTACTO A CIERRA, EL CONTENIDO DE LOS REGISTROS X Y Y SE SUMAN, EL RESULTADO ES ALMACENADO EN EL REGISTRO Z. SI A PERMANECE ABIERTO NO SE EFECTUA LA SUMA.

FIGURA 5.34.- SENAL DE SALIDA PARA SUMAR DOS REGISTROS.

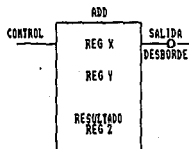


FIGURA 5.35.- BLOQUE FUNCIONAL DE SUMA

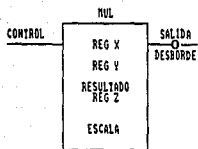


FIGURA 5.39.- BLOQUE DE MULTIPLICACION.

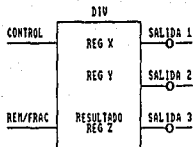


FIGURA 5.40.- BLOQUE DE DIVISION.

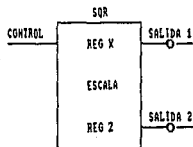


FIGURA 5.41.- FORMATO PARA LA RAIZ CUADRADA

SEÑAL DE MULTIPLICACION (MUL).

MUL
—○—

La instrucción MUL ejecuta la operación de los registros referenciados en la instrucción GET, el resultado se almacena en dos registros referenciados por dos o una señales de salida de acuerdo con el fabricante del autómata.

BLOQUE DE MULTIPLICACION (MUL).

Funciona en forma similar que las demás operaciones por formato de bloque, la operación se habilita por la entrada de control y se almacena el resultado en uno o dos registros dependiendo del fabricante algunos transfieren el resultado a una escala exponencial [figura 5.39].

SEÑAL DE DIVISION (DIV).

DIV
—○—

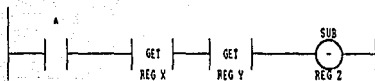
Esta instrucción permite calcular el cociente de dos registros almacenándolo en dos registros de resultado, referenciados por la señal de salida. Normalmente el primer registro de resultado almacena la parte entera y el segundo la parte decimal; al igual que en todas las operaciones mencionadas los operandos son llamados a través de la instrucción GET.

BLOQUE DE DIVISION (DIV).

El bloque de división permite obtener el cociente de dos números que pueden ser almacenados en uno o más registros [figura 5.40]. Algunos PLC manejan tres salidas en sus bloque de división: división iterativa, desbordamiento (X/0) y residuo.

La división se ejecuta una vez que se activa la entrada de control, tomando el registro X como dividendo y el registro Y como divisor, el resultado se guarda en un registro Z.

La entrada REM/FAC (remainder/fracción) permite indicar el registro que guarda el remanente o residuo y el que guarda la fracción decimal.



SI EL CONTACTO A CIERRA, AL CONTENIDO DEL REGISTRO X SE RESTA EL CONTENIDO DE Y, EL RESULTADO ES ALMACENADO EN EL REGISTRO Z. SI A PERMANECE ABIERTO NO SE EFECTUA LA RESTA.

FIGURA 5.36.- FORMATO DE SEÑAL DE SALIDA PARA RESTAR DOS REGISTROS.

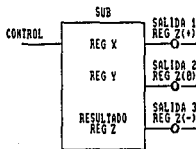
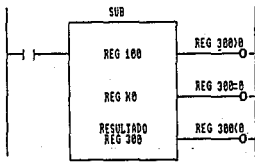


FIGURA 5.37.- BLOQUE DE SUBTRACCION.



SI A CIERRA LA RESTA SE EJECUTA, EL REGISTRO 100 CORRESPONDE A LA LOCALIDAD DE ENTRADA ANALOGICA O MULTIBIT. EL REGISTRO NO CONTIENE CEROS POR LO QUE AL EJECUTARSE LA DIFERENCIA SE GUARDARA EL RESULTADO EN EL REGISTRO 300.

FIGURA 5.38.- BLOQUE DE SUBTRACCION UTILIZADO PARA LEER ENTRADAS ANALOGICAS.

Algunos autómatas manejan factores de escala que se indican en el bloque de funciones, otros almacenan el resultado en dos registros contiguos especificando en cual se guarda la parte entera y en cual la parte decimal.

BLOQUE DE RAIZ CUADRADA (SQR).

Esta instrucción normalmente maneja de dos a tres registros, uno para guardar el valor al que se aplicará el cálculo y uno o dos para el resultado de la operación. Cuando el resultado se guarda en dos registros normalmente uno tiene la parte entera y otro la fracción.

Generalmente el bloque tiene dos salidas de las cuales la primera representa una operación valida y la segunda puede indicar un valor fraccionario. la figura 5.41 muestra un formato de bloque para el cálculo de raíz cuadrada.

INSTRUCCIONES PARA MANEJO DE DATOS.

El manejo datos consiste en la manipulación de la información en cualquier etapa del programa de control que se este ejecutando, por tanto las instrucciones que manipulan los datos permiten enlazar la información en las diferentes etapas del proceso.

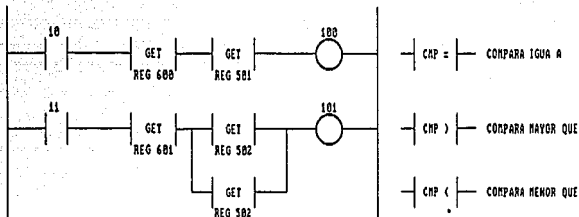
La simbología de contactos permite el manejo de multi-bits en la forma de registros o palabras, las instrucciones más comunes con que se cuenta se detallan a continuación.

COMPARACION DE DATOS (CMP O LIM).

Como su nombre lo indica se utiliza para comparar la información contenida en dos registros, comúnmente es utilizada para mantener una variable dentro de un rango marcado por el proceso.

El formato se encuentra forma de señal básica del diagrama de contactos o en forma de bloque de función, en ambos casos los indicadores mostrarán todas los posibles resultados de la comparación (" $X > Y$ ", " $X = Y$ " y " $X < Y$ ") y en función de esto las salidas pueden ser activadas o desactivadas [figura 5.42].

Al igual que en las operaciones matemáticas se utiliza la instrucción GET para denominar o cargar el primer registro a comparar y la instrucción de comparación CMP ($>$, $=$, $<$) indicara el segundo registro a comparar (ver figura 5.42).



SI A CIERRA EL CONTENIDO DEL REGISTRO 600 SERA COMPARADO CON LA CONDICION DE IGUAL CON EL REGISTRO 501; SI SE CUMPLE LA SALIDA 100 SE ACTIVARA
 SI 11 CIERRA, EL CONTENIDO DEL REGISTRO 601 SERA COMPARADO CON LA CONDICION DE MAYOR O IGUAL QUE CON EL CONTENIDO DEL REGISTRO 502; SI SE CUMPLE SE ENCIENDE LA SENAL DE SALIDA 101.

FIGURA 5.42.- FORMATO DE COMPARACIONES.

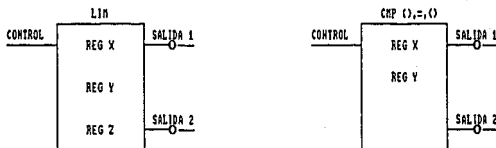


FIGURA 5.43.- FORMATO DE BLOQUE FUNCIONAL PARA COMPARACIONES.

En la figura 5.43 se muestra el formato de bloque para la comparación, en este formato se comparan los datos almacenados en los registros X y Y; cuando se cumplen las condiciones que dan continuidad a la entrada de control la comparación es ejecutada y se activa la salida 1, si la comparación es la esperada, es decir se satisface, se enciende la salida 2.

Algunos controladores tienen instrucciones de comparación basadas en el análisis de una función l1lite (LIM) en las cuales se comparan tres registros. Si se cumple la condición ($X \geq Y \geq Z$) la salida 2 será conectada y la salida 1 encendida, siempre y cuando la instrucción sea habilitada.

Algunos controladores utilizan un bloque de substracción para realizar la comparación manejando tres salidas que indican la relación de los dos registros comparados.

MATRIZ DE FUNCIONES LOGICAS.

Es una instrucción en formato de bloque que permite ejecutar las operaciones lógicas más comunes como son las funciones AND, OR, OR EXCLUSIVA, NAND, NOR Y NOT en dos o más registros.

Una función lógica puede concebirse como una operación matricial de una columna, reservando desde luego un registro para cada operando (ver figura 5.44).

El bloque de la matriz de función lógica es activado mediante la generación de continuidad en la entrada de control en la figura mencionada el registro Z fijará el resultado de la operación lógica.

En el bloque de la figura 5.44 la salida 1 será activada cuando se habilite la función lógica, la salida 2 cuando la operación es dada y la salida 3 cuando se presenta un error.

El número de operandos que puede mantener una matriz lógica depende de la arquitectura del controlador, algunos limitan el número de operandos a dos, guardando en caso necesario el resultado de la operación en un tercer registro.

CONVERSION DE DATOS (BCD-BIN, BIN-BCD, ABS/COMPL/INV).

Esta instrucción permite la conversión del formato de uno o varios registros a un formato apropiado para la ejecución de una operación requerida en la secuencia del programa de control. Las conversiones mas comunes son del código BCD a binario, de binario a BCD, valor absoluto, complemento e inversión.

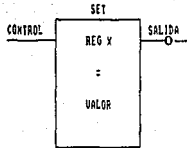


FIGURA 5.47.- INSTRUCCION PARA FIJAR UNA CONSTANTE EN UN REGISTRO.

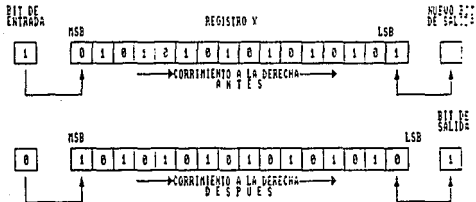


FIGURA 5.48.- FUNCIONAMIENTO DE UN CORRIENTO A LA DERECHA.

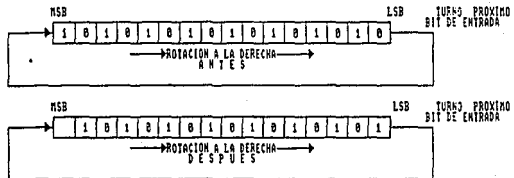


FIGURA 5.49.- FUNCIONAMIENTO DE UN CORRIENTO ROTATIVO.

Un bloque de conversión de datos se muestra en la figura 5.45 en donde se puede convertir el contenido de un registro BCD en binario y viceversa.

Cuando el bloque es activado el contenido del registro X será convertido a BCD o binario, dependiendo del tipo de convertidor, el resultado de la conversión será puesto en el registro Y. La salida del bloque es activada cuando la conversión ha sido terminada.

Las operaciones de valor absoluto, complemento e inversión normalmente se manejan en un mismo bloque, es decir el resultado de la operación seleccionada se almacenara en el mismo registro (ver figura 5.46).

Si se selecciona la operación de valor absoluto del registro X que almacena el número -3773, el contenido del registro X después de la instrucción será +3773.

Si la operación seleccionada es la de complemento, el signo del valor almacenado en el registro X será cambiado después de ejecutada la operación.

Es decir, si antes de la operación se tiene un valor negativo después de ejecutada este será positivo, si por el contrario antes de la operación se tiene un valor positivo, se tendrá un valor negativo después de la ejecución.

Si la función activada del bloque es la de invertir, el contenido del registro X después de la operación será el complemento a unos del contenido antes de habilitar la operación, es decir si se tenía 10101001 se tendrá 01010110.

ALMACENAMIENTO DE PARAMETROS CONSTANTES (SET).

En muchos casos es necesario almacenar parámetros constantes en un registro que después será utilizado en el programa para comparaciones o rangos fijos.

La instrucción SET es una alternativa que dan algunos PLCs para asignar un valor fijo a un registro, aunque otros utilizan formas particulares de su software para el manejo de constantes del programa (figura 5.47).

REGISTROS DE CORRIMIENTO Y ROTACION (ROT/SHIFT).

La instrucción de corrimiento es utilizada para mover de bits de un registros de izquierda a derecha o de derecha a izquierda, la figura 5.48 ilustra la función de corrimiento a la derecha, la instrucción de corrimiento hacia la izquierda funciona de forma similar.

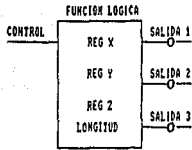


FIGURA 5.44.- FORMATO DE MATRIZ LOGICA.

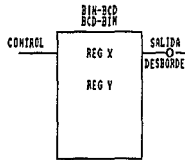


FIGURA 5.45

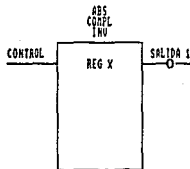


FIGURA 5.46.

La instrucción de rotación es similar a la de corrimiento, la diferencia es que en esta el bit que esta corriendo pasa a ser el último para recorrer ya sea a la izquierda o derecha dependiendo del tipo de corrimiento que se este realizando [figura 5.49].

El formato más común es el de bloque, algunos incluyen tanto la entrada de habilitación a la izquierda como a la derecha o la dirección de la rotación, las variantes que puede manejar un bloque depende del fabricante del PLC. En la figura 5.50 se muestra un registro X, en el cual normalmente se almacena la cadena de bits que van a recorrerse o rotarse.

Las cadenas de bits a los que se aplican estas instrucciones, pueden ser localidades reales de entradas o salidas que pueden ser utilizadas para introducir o extraer datos mediante las instrucciones de corrimiento o rotación.

El número de bits indica la cantidad de bits recorridos o rotados que tiene lugar cuando la entrada de control esta encendida o apagada.

El corrimiento o rotación son requeridas en aplicaciones donde el estado de las entradas debe ser dirigido a través de una línea determinada.

ANALISIS DE BIT (XBON/XBOF).

Esta instrucción permite examinar el estado de un sólo punto o bit, en una localidad de memoria. Este tipo de instrucción generalmente se utiliza cuando las banderas o marcas son puestas durante la ejecución del programa y después de probada y comparada. Un bit puede ser analizado para una estado de apagado o encendido.

La figura 5.51 muestra un bloque de análisis de bit. La función es ejecutada cuando la entrada de control se enciende, la posición del bit especificado en el registro o localidad de memoria X será examinado para una condición de encendido o apagado. La salida será conectada si la instrucción es de encendido.

INSTRUCCIONES DE TRANSFERENCIA DE DATOS.

la función principal de este tipo de instrucciones es la de la comunicación del PLC con todos los dispositivos que tiene conectados. Su función es la de transferir datos numéricos almacenados en memoria entre el autómatas y todas las E/S del sistema.

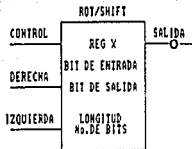


FIGURA 5.50.- INSTRUCCION DE BLOQUE PARA UN CORRIMIENTO DE IZQUIERDA O DERECHA CON OPCION DE ROTACION.



FIGURA 5.51.- INSTRUCCION DE ANALISIS DE ENCENDIDO O APAGADO DE UN BIT.

Como en otras instrucciones el formato puede ser encontrado en formato de señal de escalera (señal) y de bloque, en el formato de señal de escalera incluyen las instrucciones GET y PUT [figura 5.52].

Las funciones GET y PUT son utilizadas generalmente con controladores programables que manejan principalmente la simbología de contactos para la implementación de operaciones aritméticas y comparación.

TRASLADO DE UN BIT O REGISTRO (MOVE).

La función MOVE permite la copia información de una localidad a otra la información puede constituirse de un bit o registro. La figura 5.53 muestra el formato de bloque para esta instrucción cuando el control de entrada es habilitado el contenido del registro X será copiado en el registro Z.

El registro fuente es X y el registro destino es Z, ambos registro pueden formar parte del conjunto de E/S. La salida será encendida como señal de que la instrucción ha sido cumplida.

En algunos autómatas la instrucción se hace a través de palabras o comandos especiales que actualizan la información de la memoria de programa, en estos casos es común que automáticamente se convierta la información del registro fuente al código que opera en el registro destino.

Existe otro tipo de instrucción de traslado que permite la copia de una parte de los bit que componen un registro (figura 5.54) este procedimiento consiste en indicar con un registro máscara los bits que no van ser trasladados, en la figura el registro máscara Y los marca con "0" y en el registro destino se marcan los bits no trasladados con "b".

Algunos controladores comerciales cuentan con una instrucción que permite el traslado del estado general del sistema ó módulos de E/S conectados al PLC: MOVE STATE. la información puede ser enmascarada, comparada o examinada para determinar el mayor o menor número de fallas en el sistema o e módulos E/S que determinan la ejecución de una acción necesaria en el proceso.

TRASLADO DE UN BLOQUE (MOVBK).

Se utiliza cuando se requiere copiar un grupo de registros a otra localidad de memoria. En la figura 5.55 se ejemplifica el procedimiento que se sigue para mover un bloque de registros.

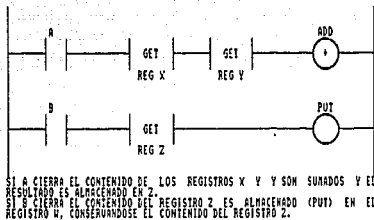


FIGURA 5.52.- FUNCIONAMIENTO DE LAS INSTRUCCIONES GET Y PUT.

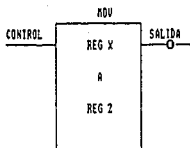


FIGURA 5.53.- INSTRUCCION QUE MUDE UN BIT O REGISTRO DE UNA LOCALIDAD A OTRA.

Cuando se habilita la entrada de control, la información a trasladar parte de la localidad X a la localidad X + L (L= longitud) y es trasladada de la localidad Y a la localidad Y + L. Algunos PLC limitan el número de localidades que pueden trasladarse en un ciclo de monitoreo.

TRASLADO DESDE UNA TABLA O BLOQUE (TABLE MOVE).

Es una instrucción necesaria cuando se requiere extraer un registro de una tabla o bloque de información y trasladarlo a un registro o palabra en memoria.

Se tienen fundamentalmente dos tipos de movimientos: tabla-registro o registro-tabla. La función primordial de la instrucción es la manipulación de un registro indicador, de forma que una palabra o registro pueden ser almacenados en una localidad de una tabla o bloque en particular señalados por el indicador.

Cuando la entrada de control del bloque (figura 5.56) se da un cambio de 0 a 1 lógico, la instrucción TABLE MOVE es ejecutada y el contenido del registro indicador incrementado. La entrada de en medio es utilizada para deshabilitar el registro indicador, la tercera entrada es utilizada para restaurar el indicador a cero (inicializa la tabla).

Sí fuese necesario almacenar o recuperar un dato desde una tabla específica, el registro indicador puede ser cargado con el valor apropiado del punto que específica la localidad. Cuando esta carga es llevada a cabo determina prioridades para la instrucción de movimiento desde un bloque o tabla a través de la instrucción de almacenamiento de un parámetro constante o la de movimiento de un registro.

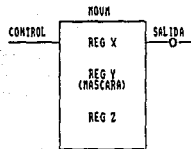
La longitud L determina el número localidades de palabras contenidas en la tabla, contando desde la localidad X hasta la localidad X + L, la salida superior indica la ejecución de la instrucción, la salida inferior se enciende cuando el registro indicador alcanza el fin de la tabla.

La aplicación de esta instrucción es la de actualizar los datos cargando los datos generados en los módulos E/S en el último ciclo de monitoreo cualesquiera que sea la causa de su generación.

TRANSFERENCIA DE BLOQUES E/S.

Este tipo de instrucción es útil en sistemas diseñados en base módulos especiales E/S (señales analógicas, decodificadores, arrancadores de motor de paso, etc.). Se cuenta con dos tipos de transferencia entrada y salida.

La figura 5.57 muestra el formato de la instrucción. La localización de la dirección del módulo se basa en la localización de bastidor en donde la interfase se encuentra físicamente.



REG X 1111000011110101
 REG Y 0000000011111111
 (MASCARA)
 REG Z 0000000011110101

FIGURA 5.54.- EJEMPLO DE UNA INSTRUCCION DE MOVIMIENTO DE INFORMACION CON MASCARA.

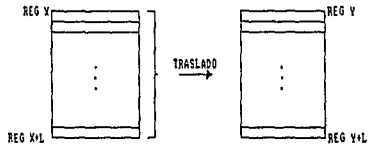
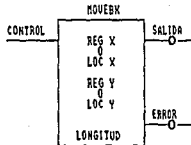


FIGURA 5.55.- FORMATO PARA MOVER UN BLOQUE MEDIANTE LA INSTRUCCION MOVEBK.

Al habilitar la instrucción, el dato del módulo E/S (por un BKXFER IN) es almacenado en la localidad o registro X de la memoria. El destino de la transferencia es indicada por la dirección del módulo.

La longitud del bloque especifica que tantas localidades son necesarias para guardar los datos del módulo de E/S, por ejemplo el dato de un módulo de entradas analógicas de cuatro canales de transferencia pueden ser leídos a la vez, si la longitud especificada en la instrucción es cuatro.

La salida 1 indica que la operación de transferencia ha sido terminada, la salida 2 indica un error en la transmisión como podría ser la velocidad de transmisión o alguna falla en el módulo.

TRANSFERENCIA ASCII.

Esta instrucción permite la distribución de caracteres ASCII desde un controlador programable a un dispositivo periférico. Esta instrucción opera en conjunción con un módulo de comunicación ASCII y su ejecución es sobre el programa de control.

Se manejan dos clase de comunicación: lectura de datos desde el periférico o escritura de datos al periférico. La figura 5.58 muestra el formato para la instrucción.

Cuando la entrada de control habilita la transferencia ASCII (entrada o salida), la instrucción permitirá al módulo especial E/S la lectura.

El dato es leído desde el módulo por el procesador y almacenado en la localidad especificada (X). La localización de la módulo se indica en la entrada de dirección del formato de bloque.

Cuando la operación se trata de una escritura, la información es enviada desde la localidad en donde esta almacenada a la dirección donde el módulo esta localizado.

Algunos autómatas extienden la capacidad de la instrucción de transferencia ASCII al uso de un registro indicador para acceder caracteres específicos en la tabla, otras variantes de la instrucción permiten al programador indicar el número de caracteres o bytes que van a ser transmitidos durante un ciclo de monitoreo.

La rapidez de transmisión (baud rate) depende del tiempo de monitoreo y del número de dispositivos de ASCII activos al mismo tiempo. La instrucción de transferencia ASCII absorbe las características para transmisión: velocidad, bits de paro y arranque, paridad, etc.

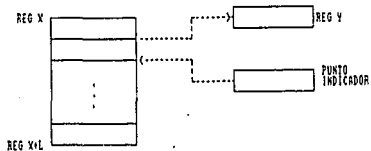
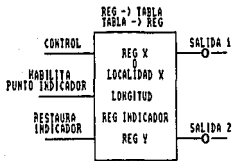


FIGURA 5.56 FUNCIONAMIENTO DE LA INSTRUCCION PARA MOVER UN BLOQUE.

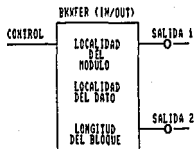


FIGURA 5.57.- INSTRUCCION DE TRANSFERENCIA DE UN BLOQUE E/S

TRANSFERENCIA DE PILA PRIMERA-ENTRADA/PRIMERA-SALIDA (FIFO STACK).

La instrucción FIFO STACK (first-in-first-out) es utilizada para la construcción de una tabla o cola o una cadena de registros o caracteres, donde los datos se almacenan o acumulan (figura 5.59).

La función es similar a la de un registro de corrimiento asíncrono en el cual una palabra (16 bits) es recorrida dentro de la pila cada vez que la instrucción es ejecutada, el dato es recorrido en el mismo orden en el que es recibido: el primer uno de entrada será el primer uno de salida.

La instrucción FIFO consiste de dos modalidades, entrada (FIFO IN) y salida (FIFO OUT) para introducir y extraer una cola de información. La instrucción facilita el almacenamiento y recuperación de largos grupos de datos temporales que se van generando durante la ejecución del programa de control.

Una aplicación de esta instrucción podría ser el control de la posición de una pieza mecánica como un pistón, en donde las señales de entrada informan la posición actual del pistón y las señales de salida activan los mecanismos que corrigen su posición.

La instrucción FIFO es activada cuando la entrada de control tiene un cambio de estado de cero a uno lógico. La señal de restauración limpia los registros de la tabla (figura 5.59).

Para la instrucción de entrada (FIFO IN) el registro fija el dato que será transferido a la cola de información. Este dato ocupa un lugar en la pila cuando el control de entrada es habilitado. En la instrucción de salida (FIFO OUT) es extraído a través del registro Z. la longitud de la pila es especificado por el comando del bloque FIFO LENGHT (del registro X al registro X + L).

ORDENADOR DE BLOQUES O REGISTROS (SORT).

La instrucción ordena en forma ascendente o descendente un bloque o registro de acuerdo a su contenido. Una vez que la instrucción se habilita se analiza el contenido de cada registro para ejecutar el ordenamiento.

En la figura 5.60 si el contacto A cierra la instrucción es habilitada y se procede al análisis de los registros contenidos de la localidad X a la localidad X + L, siendo ordenados en forma ascendente y los registros clasificados y ordenados se almacenan en el bloque limitado por las localidades Y y Y + L.

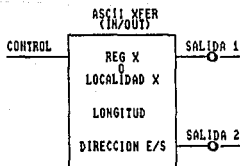


FIGURA 5.58.- INSTRUCCION PARA TRANSFERIR INFORMACION ASCII.

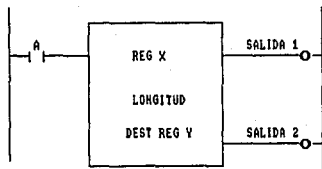


FIGURA 5.59.

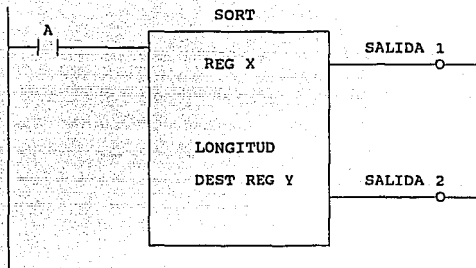


FIGURA 5.60

MANEJO DE FUNCIONES ESPECIALES.

Los módulos de funciones especiales requieren de instrucciones que permiten su enlace con el controlador que requiere de comandos de función especial que facilita la comunicación y funcionamiento de sistemas de mayor complejidad.

SECUENCIADORES.

El bloque secuenciador es una instrucción que permite la simulación de un temporizador de árbol o cilindro. El secuenciador tiene un comportamiento análogo al de una caja de música en la que cada estaca produce un tono al golpear el cilindro rotatorio. En un secuenciador cada estaca (bit) que toca el cilindro puede ser interpretado como un uno lógico, cada estaca que no toca el cilindro produce un cero lógico.

Los secuenciadores se especifican utilizando una tabla como la mostrada en la figura 5.61 que corresponde al comportamiento de un cilindro de salida extendida (sin fin). El número de bits pueden variar desde 8 a 64 o más, dependiendo del fabricante.

En la figura 5.61 se ilustra un cilindro y la tabla de comparación del secuenciador. El ancho de la tabla puede variar al igual que en la caja de música el número de estacas esta limitado por la melodía.

Cada golpe o paso equivale en el secuenciador a un registro de entrada o salida. En la figura 5.62 se muestra el formato de bloque para un secuenciador típico.

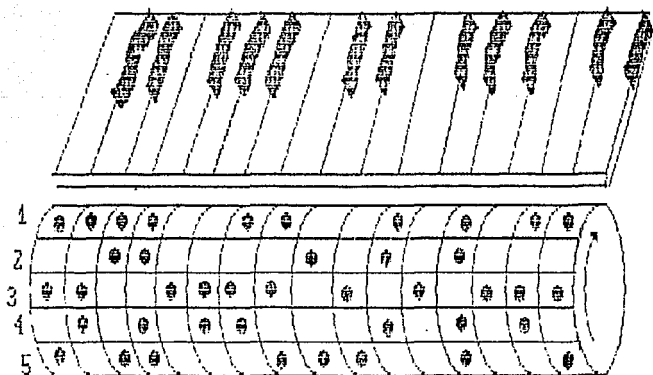


TABLA DEL
CILINDRO DE
PASOS

TABLA DE LOCALIZACION DE BIT

1	1	1	1	0	0	1	1	0	0	1	0	1	1
2	0	0	1	1	0	0	0	0	1	0	1	0	0
3	1	1	0	0	1	1	1	1	0	1	0	1	1
4	0	1	0	1	0	1	1	0	0	0	1	0	1
5	1	0	1	1	0	0	0	1	1	1	0	0	1

"1" = CONEXION "0" = DESCONEXION

FIGURA 5.61.- ANALOGIA DEL COMPORTAMIENTO DE UN SECUENCIADOR.

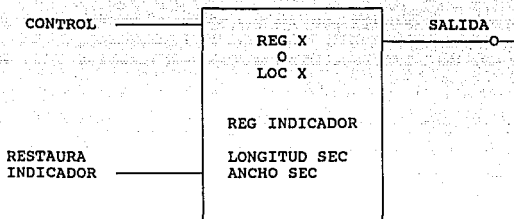


FIGURA 5.62.- BLOQUE SECUENCIADOR

Cuando en la entrada de control se tiene un transición de 0 a 1, el contenido de la tabla del secuenciador será extraída de forma secuencial, cada paso (golpe) es designada por el registro indicador.

Cada vez que la entrada se energiza el registro indicador se incrementa automáticamente e indica la localidad de la próxima tabla.

Dependiendo del PLC, algunas entradas de control permiten la activación por temporizador o por evento, en cada caso el secuenciador puede referirse a la manipulación de un evento o tiempo.

La entrada de restauración regresa el registro indicador a cero en caso necesario (dentro del programa de control lo vuelve al paso 1), El largo y ancho de la secuencia son determinados por la cantidad de pasos o bits utilizados en la tabla correspondiente. La salida se enciende siempre u cuando la instrucción sea habilitada.

DIAGNOSTICADOR.

La instrucción es útil cuando se requiere de la comparación de dos bloques de la memoria de los cuales uno contiene las actuales condiciones y uno las condiciones de referencia o comparación, normalmente las condiciones de referencia se establecen a partir de la máquina controlada, aunque algunos autómatas pueden marcar condiciones de referencia.

Esta comparación se hace bit a bit para diagnosticar si ambos bloques son idénticos, si no sucede así, el número de bits y los estados serán almacenados como los valores actuales.

La figura 5.63 muestra el formato para la instrucción de diagnóstico, cuando la entrada de control se energiza, el contenido de los registros marcados con las localidades de la X a la X + L se comparan con los registros de las localidades de la Y a la Y + L.

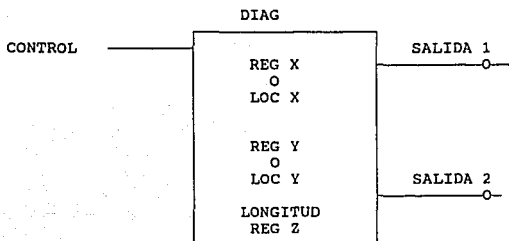


FIGURA 5.63.- BLOQUE DE DIAGNOSTICO.

Si se encuentra diferencia se almacena en el registro Z, el contenido de los valores de referencia se conservan en su forma original en el registro Y. La salida 1 se activa al completarse la operación y la salida 2 se enciende cuando existe diferencia en los registros comparados.

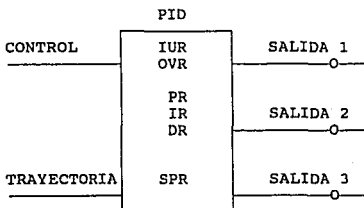


FIGURA 5.64.- INSTRUCCION PID: UTILIZA UN ALGORITMO PROPORCIONAL-INTEGRO-DERIVATIVO.

CONTROL PROPORCIONAL INTEGRO-DERIVATIVO (PID).

Este bloque permite el manejo de variables analógicas utilizando un algoritmo de proporción integro-derivativo (PID). El usuario debe especificar correctamente aquellos parámetros asociados con el algoritmo con el que se pretende controlar el proceso en forma óptima.

La figura 5.64 muestra un bloque PID, este tiene dos formas de operar: manual y auto. En operación auto, cuando se energiza la entrada de trayectoria, determinará si las variables PID están siendo enviadas y no extraídas.

Si el bloque PID no es habilitado, se encuentra en la forma manual. el controlador puede mantener las variables en las mismas condiciones.

El registro de variable de entrada IVR y el registro de variable de salida OVR, relacionan su uso con la localización de los módulos analógicos E/S.

El registro de proporción PR, el registro de integración IR y el registro de derivación DR fijan los valores ganados que necesitan ser especificados para el control del proceso.

El registro de fijación de rango RPS (set point) fija el valor requerido para un punto específico, dependiendo de los controladores programables algunos pueden fijar este punto para variables como tiempo muerto, límites superior e inferior, rangos de levantamiento, etc.

La salida 1 indica un ciclo de control activo, la salida 2 y la 3 usualmente indican límites de alarma superior e inferior.

Comercialmente se tienen PLC que manejan capacidades PID fuera del bloque de instrucción, generalmente en estos casos, el módulo especial PID es utilizado dentro de todos los parámetros que están en el módulo.

El punto fijado en SPR y los parámetros ganados son transferidos durante la inicialización del programa a través de una instrucción de salida como transferencia de bloque (BKXFER), o movimiento de un registro (MOV) a un registro de salida. El módulo de datos puede ser alterado durante la ejecución del programa si cualquiera de los parámetros necesita ser cambiado.

REDES DE COMUNICACION.

Con el desarrollo de las redes locales (LANs: Local Area Networks) se crea una necesidad de comunicación y e intercambio de información entre PLC y redes de trabajo.

El intercambio de información se realiza por medio de instrucciones específicas para cada manufactura comercial. La operación envuelve comunicación de estados de contactos y/o señales de salida así como registros con diversa información.

La siguiente tabla muestra las instrucciones más comunes que pueden encontrarse en la red de comunicación de un controlador programable.

El programador debe proceder conforme a las normas de la red de comunicación del PLC respetándolas íntegramente. También la organización y asignamiento de registros debe hacerse de tal forma que se evite cualquier confusión.

En general una vez que la instrucción es ejecutada y realizada la actualización de los datos en el último monitoreo, la información es pasada a través del alambrado de la red (módulos o tableros internos) para su procesamiento y transmisión.

Dependiendo del controlador utilizado el formato de la instrucción puede ser variable. La instrucción presentada aquí es utilizada como una guía e ilustración de sencilla implementación. Algunos controladores pueden utilizar instrucciones de transferencia de datos para acceder la red y otros tienen instrucciones específicas para el uso de redes locales de comunicación.

La organización de una red depende de su tipo de configuración. Algunos controladores incluyen interfases para la comunicación, otros manejan la comunicación mediante un módulo de interfase, ambos procedimientos manejan la misma tarea.

Sí una interfase para red de comunicación es colocada den los bastidores E/S, implica diferentes caminos o vías que los fabricantes proporcionan al usuario de un PLC en particular para el llamado de un nodo de comunicación.

Varios controladores programables permiten configuraciones sobre la configuración principal o plataforma del PLC, donde la localización del puesto del módulo de la red puede ser especificado. Otros controladores pueden automáticamente reconocer donde se localiza el puesto del módulo de la red de comunicaciones.

En otros métodos pueden ser especificado la localización del puesto mediante una instrucción del software para redes de comunicación de uso similar a la instrucción de transferencia de un bloque E/S (BKXFER).

Las señales de salida y contactos de la red pueden ser citados como salidas de red y red de contactos igual que los registros pueden denominarse registros de red.



INSTRUCCION	DESCRIPCION
NET 	SALIDA DE RED. LA SALIDA ACTIVADA POR LA LOGICA DEL PLC PREPARA UN BIT DE INFORMACION PARA SER TRASMITIDO DEL PLC A LA RED.
NET 	CONTACTOS DE RED. LOS CONTACTOS DE LA SALIDA DE RED PUEDEN SER UTILIZADOS PARA ACTIVAR EL ENVIO O RECEPCION EN LA RED.
NET SEND	RED EMISORA SE ACTIVA POR LA LOGICA DEL PLC QUE PREPARA LA INFORMACION Y LA PONE A DISPOSICION DE LA RED DE RECEPCION.
NET RCV	RED RECEPTORA CAPTURA LOS REGISTROS DE INFORMACION ENVIADOS POR LA RED EMISORA.
SEND NODE	NODO DE ENVIO, ESPECIFICA UN NODO PARTICULAR DE LA RED DEL PLC, ES SIMILAR AL COPIADO DE INFORMACION.
GET NODE	NODO DE CAPTURA, RECUPERA LA INFORMACION EMITIDA MEDIANTE LA INSTRUCCION DE ENVIO ES SIMILAR A LA LECTURA Y COPIADO.

TABLA 5.4.- INSTRUCCIONES TÍPICAS DE UN PLC.

La red de salida son salidas internas que son localizadas en un área especial de la memoria de datos a lo largo de los registros de red. Estos elementos de red pueden ser parte de una área interna de almacenamiento con las capacidades de una red local de comunicaciones (LAN: local area network).

RED DE SALIDA (NET).

La instrucción de red de salida es utilizada en conjunto con la instrucción de red de contactos (ver figura 5.65a) para el traslado del estado de un bit de información desde el PLC a la red.

Sí existe continuidad en la lógica de la red de salida, la dirección indicada encenderá y transmitirá al módulo interfase la transmisión de la red local. La referencia direccionada debe ser una señal válida de la red dependiendo del autómata que se este utilizando y la complejidad del sistema de control. Los estados de las salidas están disponibles para todas las estaciones o nodos de la red de sistema de PLCs.

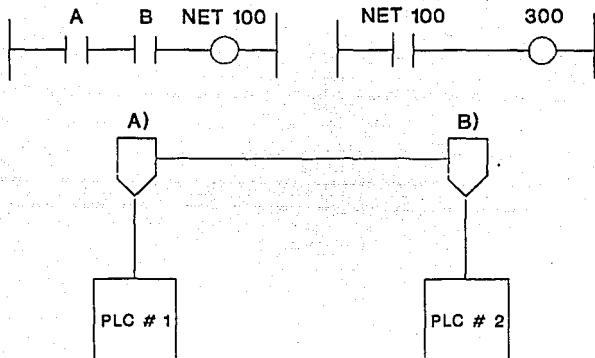


FIGURA 5.65.- EJEMPLO DE LAS INSTRUCCIONES NET PARA SALIDA DE RED Y CONTACTOS DE RED.

RED DE CONTACTOS (NET).

Esta instrucción se utiliza para capturar la información del estado lógico desde una red de salida. La referencia direccionada en la red de contactos debe ser de una red de salida, de otra forma esos contactos (encendidos o apagados) nunca serían evaluados.

La referencia debe también tener una dirección de referencia válida que puede ser diferente entre las diversas manufacturas (la figura 5.65b ejemplifica la operación de la red de contactos).

La información desde la red es generalmente obtenida durante la lectura de las entradas al tiempo que el procesador lee el estado del módulo de la red de comunicaciones proyectándolo como una minimemoria de datos.

Si en la referencia de la red de contactos se encuentra un lógico, la evaluación dará lugar a la apertura o cerradura de sus contactos para dar vía a la continuidad o suspenderla. La evaluación depende de como la red de contactos es programada (—|— o —|/—).

RED EMISORA.

La red emisora es utilizada para el envío de registros de información a la red local en donde es captada por un sistema de recepción.

La instrucción NET SEND es activada como cualquier bloque de funciones, si se cumplen las condiciones para la activación o encendido de la entrada de control, cuando la operación es ejecutada y el contenido de los registros enviados sobre la línea de la red.

La instrucción contiene dos salidas: para indicar que la operación ha sido realizada (salida 1) y que no se detectaron errores en la transmisión (salida 2).

La figura 5.66a muestra la configuración típica de una red emisora. Un registro establecido puede ser transmitido a la red si la longitud L se especifica como mayor que uno; Los registros transmitidos abarcan desde la localidad X a la localidad X + L. La red emisora opera normalmente con la red receptora.

RED RECEPTORA.

La red receptora es normalmente el instrumento que cata los registros disponibles en las líneas de la red y las almacena dentro de la memoria de datos del PLC.

El usuario debe asegurar que las direcciones de referencia de los registros requeridos por el sistema, sean las mismos en la red emisora que en la receptora, dado que serán extraídos de la fuente (normalmente un PLC maestro) y transmitidos al área local en donde son solicitados con la misma dirección. En la figura 5.66b se muestra el uso de la instrucción de red emisora (NET SEND).

Una vez que la instrucción dispone del registro de información, su contenido será almacenado en el registro destino con localidades de la Y a la Y + L si la longitud es mayor que uno. La salida 1 indica el fin de la operación y la 2 un posible error en la recepción.

NODO DE ENVIO.

La instrucción de nodo de envío (SEND NODE) opera en forma más directa que la red emisora, puede preparar el camino o trayectoria del registro de información a la localización específica (nodo) conectada a la red.

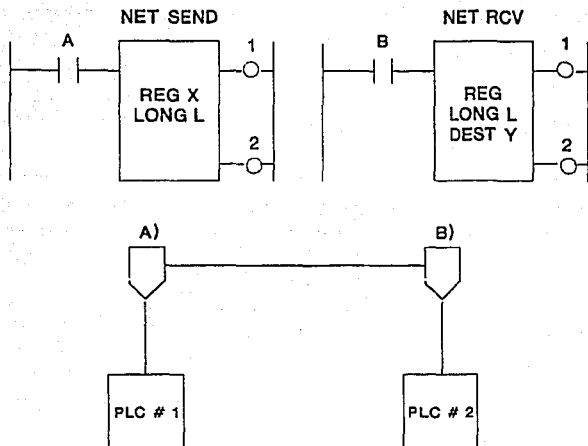


FIGURA 5.66.- INSTRUCCIONES DE ENVIO Y RECEPCION DE INFORMACION.

Básicamente la instrucción SEND NODE copia un registro sobre otro de nodo de comunicación de la red. La figura 5.67 ilustra la instrucción, cuando el bloque es habilitado y el contenido de los registros X a X + L es enviado a los registros Y a Y + L; la salida 1 indica la ejecución de la operación y la salida 2 indica la detección de un error.

NODO DE CAPTURA.

La instrucción GET NODE permite la captura o recuperación de la información transmitida desde un nodo del PLC. Su funcionamiento es similar a la copia de un registro desde un nodo a otro indicado.

La figura 5.68 muestra el formato de la instrucción, cuando se habilita, el contenido de los registros X a X + L de etiqueta N, son almacenados en los registros Y a Y + L del PLC que ejecuta la instrucción GET NODE. La salida 1 indica el fin de operación y la 2 si hubo error en la transmisión.

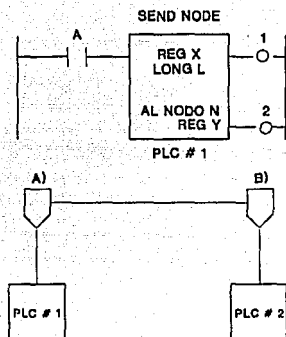


FIGURA 5.67.- INSTRUCCION PARA UN NODO DE ENVIO.

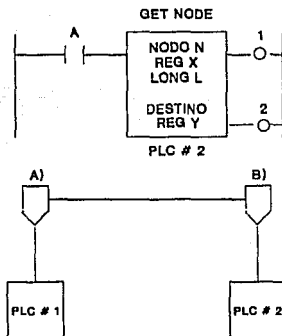


FIGURA 5.68.- INSTRUCCION PARA UN NODO DE CAPTURA.

OTROS LENGUAJES DE PROGRAMACION.

LISTA DE INSTRUCCIONES.

La lista de instrucciones es un código simbólico, en el cual cada elemento del conjunto de comandos corresponde a una instrucción en lenguaje de máquina.

Los fabricantes de PLC en general manejan como lenguaje de programación el diagrama de contactos, incluyen normalmente el listado de instrucciones además de otros lenguajes más avanzados que se han venido agregando en el software para controladores programables.

Las arquitecturas que contemplan la lista de instrucciones en sus lenguajes de programación, tienen características muy particulares de acuerdo a la manufactura del equipo, es decir; el lenguaje de lista de instrucciones es distinto de acuerdo a la marca del equipo.

El listado de instrucciones es el que utilizan las unidades de programación sencillas regularmente portátiles, que manejan unas cuantas líneas en su pantalla y que normalmente son de uso cotidiano en la planta. Este apartado analiza en forma sencilla las variables e instrucciones más comunes que se pueden encontrar.

La programación en código simbólico de este lenguaje acerca su estructura al lenguaje de máquina y esta indicada especialmente para usuarios familiarizados con la programación común y los fundamentos de la electrónica digital.

IDENTIFICACION DE VARIABLES.

Variables de entrada Xn.

El término X representa una variable binaria de entrada y el término n corresponde a su localización en el bastidor o conector de entrada.

Variables de salida externa Yn.

El término Y representa una variable binaria de salida y el término n corresponde a su localización en el bastidor o conector de salida.

VARIABLES DE SALIDA INTERNA: IRn o Cn.

Los términos IR o C pueden representar una variable binaria interna y el término n corresponde a su localización en el programa de control, comúnmente este tipo de variables se utilizan como banderas en diferentes etapas del programa.

COMANDOS DEL LISTADO DE INSTRUCCIONES.

Las instrucciones normalmente se dividen en el controlador de acuerdo a su función, encontrándose tres clasificaciones principales: Instrucciones Básicas, Instrucciones de Tiempo y Conteo e Instrucciones de Control de flujo.

INSTRUCCIONES BASICAS.

Este grupo de instrucciones cubren acciones como:

- * Seleccionar una determinada variable para utilizarla como operando o como objeto de una acción de lectura de una entrada, o activación/desactivación de una salida (acción E/S).
- * Ejecutan acciones de E/S.
- * Realizan operaciones específicas con alguna variable predeterminada en el programa.

Dentro de esta clasificación se tienen las siguientes instrucciones.

STR:

Se utiliza para seleccionar la primera variable que se va a utilizar en una serie o secuencia de instrucciones.

En la tabla 5.1 para la figura a, la variable X1 indica la primera variable que se va a utilizar en la etapa siendo seleccionada con la instrucción STR X1, en la figura b se selecciona la variable X3 con la instrucción STR X3.

STR NOT:

Selecciona la primera variable invertida a utilizar en una secuencia de instrucciones.

Por ejemplo, la instrucción STR NOT X10 toma como primer variable de entrada a X10 y la invierte. La instrucción STR NOT C5 selecciona la variable de salida interna C5 y la invierte.

OUT:

Esta instrucción determina el fin o salida de una etapa o lazo (rung) actuando sobre la variable de salida interna o externa que se le asocia.

En la tabla 5.1 al final de cada figura se pone la rutina o zona de instrucciones que conforman la etapa, obsérvese que al final de cada listado se encuentra la instrucción OUT marcando la señal de salida que se le asocia, por ejemplo en la figura f se tiene al final la instrucción OUT C1.

OUT NOT:

Actúa sobre la variable de salida invertida (interna o externa) asociada a ella.

STR X3
OUT NOT Y3

En la secuencia anterior la instrucción STR X3 selecciona la variable X3 y la instrucción OUT NOT Y3 almacena en la variable Y3 la variable X3 invertida.

OR:

Realiza la función lógica O entre la variable especificada por ella y una o más variables combinadas.

STR Y2
OR X3
OR C9
OUT Y6

Siguiendo la secuencia anterior: Se selecciona la salida externa Y2 (STR Y2). A Y2 y X3 se les aplica la función lógica O (OR X3). Con el resultado de la operación "Y2 O X3" y la variable de salida interna C9 se ejecuta la operación lógica O (OR C9). Finalmente la instrucción OUT Y6 almacena el resultado en la salida externa Y6. En forma algebraica las operaciones indicadas por el programa anterior es

$$Y6 = Y2 + X3 + C9.$$

OR NOT:

Ejecuta la función O entre la negación de la variable especificada por ella y una o más variables combinadas. Los listados siguientes ejemplifican dos formas de expresión.

```
STR C3
OR NOT Y5
  OR NOT X2
  OUT Y10
```

```
STR C3
OR
  NOT Y5
  OR
  NOT X2
  OUT Y10
```

Las operaciones indicadas en orden secuencial son: selecciona la variable de salida interna C3. Aplica la función O a C3 y la variable de salida externa Y5 invertida. El resultado se le aplica la función O junto con la variable de entrada X2 invertida. El resultado final se almacena en la variable de salida externa Y10. La ecuación algebraica es:

$$Y2 = C3 + Y5' + X2'.$$

AND:

Realiza la función lógica Y entre la variable especificada por ella y una o más variables combinadas.

En la tabla 5.1 en el listado de la figura a se tiene la operación Y más sencilla en la que el resultado es almacenado en la variable de salida externa Y1. En el listado de la figura c se observa la operación Y aplicada al resultado de una función lógica O y la salida interna C1.

AND NOT:

Aplica la función Y a una o más variables y a la negación de la variable especificada por ella.

En la tabla 5.1 en la figura h se muestra el listado de la función AND NOT, notese que la salida interna C1 esta testada para indicar su negación. Hay que tener presente que la variable con apóstrofo o la variable testada indican negación.

OR STR:

Aplica la operación lógica O a las dos secuencias anteriores a ella iniciadas por STR o STR NOT.

```

STR X7
OR X10
STR C2
OR C3
OUT Y4
STR X14
AND C1
OUT Y6
OR STR
OUT Y20

```

De la tabla 5.1 si se tuvieran el listado de la figura d y el de la figura g seriados y al final se colocara la instrucción OR STR Y OUT Y20, se ejecutaría la función O a las variables de salida externa Y4 y Y6 guardándose el resultado en la salida externa Y20.

AND STR:

Aplica la operación lógica Y a las dos secuencias anteriores a ella iniciadas por STR o STR NOT.

```

STR X7
OR X10
STR C2
OR C3
OUT Y4
STR X14
AND C1
OUT Y6
OR STR
OUT Y20

```

Si en lugar de la función OR STR en el ejemplo anterior se tuviera la instrucción AND STR, a las variables de salida externa Y4 y Y6 se les aplicaría la operación lógica Y.

INSTRUCCIONES DE TEMPORIZACION Y CONTEO.

Como se menciono antes estas operaciones generan variables que controlan la activación o duración de un evento condicionado a una función de tiempo o número de impulsos de una variable de entrada.

TMR:

Ejecuta una temporización basándose en dos variables:

Xi = variable de puesta a cero.
Xj = variable temporizada.

La señal de salida del temporizador es dada mediante variables internas o externas.

La programación del temporizador requiere de un mínimo de cuatro instrucciones (ver figura siguiente):

- * Selección de variable puesta a cero (Xi).
- * Selección de variable temporizada (Xj).
- * La instrucción TMR n que elige el temporizador (hay que recordar que los contadores y temporizadores del autómata están integrados a su hardware), TMR inicia la temporización si en la variable Xi se encuentra en uno (no hay puesta a cero) y Xj pasa a uno (se activa la variable cuyo cambio marca la temporización).
- * Activación de la posición de memoria que almacena el valor del tiempo preseleccionado (objetivo).

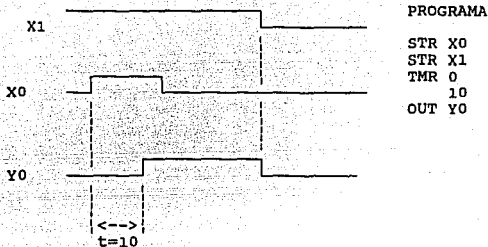


FIGURA 5.69.- DIAGRAMA DE SECUENCIA TEMPORAL DE UN TEMPORIZADOR.

Para que la señal Y0 pase al nivel uno es necesario que X1 este en uno (STR X0) y que transcurran 10 unidades de tiempo.

La instrucción TMR del listado de instrucciones también permite generar retardo para la respuesta de una variable determinada ya sea de conexión o desconexión de la misma. Para ello se utiliza la misma variable como variable de puesta a cero y temporizada, por ejemplo:

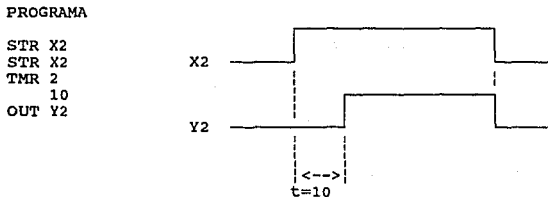


FIGURA 5.70.- DIAGRAMA DE UN TEMPORIZADOR PARA UN RETARDO A LA CONEXION.

La activación de la variable de salida externa Y2 es temporizada a partir de la entrada X2, el diagrama de tiempo correspondiente es ilustrado en la figura 5.70.

El diagrama de tiempo y el programa de una instrucción TMR para un retardo a la desconexión se muestra en la figura 5.71.

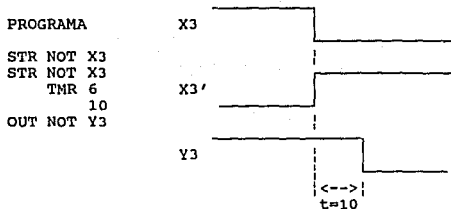


FIGURA 5.71.- DIAGRAMA DE UN TEMPORIZADOR PARA UN RETARDO A LA DESCONEXION.

CTR:

Ejecuta la función de contar, se pueden utilizar dos o tres variables dependiendo del tipo de conteo que se requiera: ascendente o descendente.

S Para una cuenta creciente se requiere de una instrucción que maneja una variable para la puesta a cero o restablecedor y una segunda variable para la cuenta.

Para una cuenta decreciente se requiere además de una variable de entrada que indica el modo de contar (ascendente/descendente).

De acuerdo al programa de la figura 5.72 STR X5 se selecciona la variable que restaura el cero, con STR X4 se selecciona el tren de pulsos que constituyen la señal de cuenta.

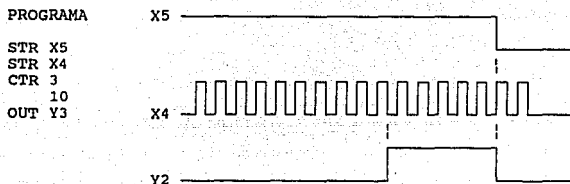


FIGURA 5.72.- DIAGRAMA DE TIEMPOS DE UN CONTADOR ASCENDENTE.

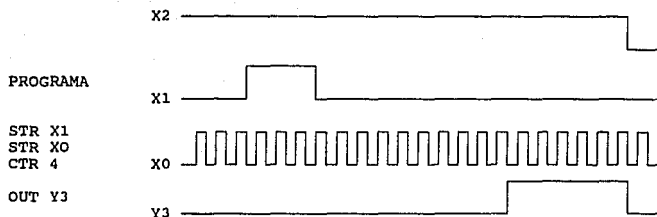


FIGURA 5.73.- DIAGRAMA DE TIEMPOS DE UN CONTADOR ASCENDENTE.

La instrucción CTR indica que la salida Y3 se encenderá cuando pasen 10 impulsos de la señal X4 manteniéndose en uno hasta que la variable X5 pase de uno a cero (restaura el cero a la cuenta). Un ejemplo de contador descendente y su diagrama de tiempo se observa en la figura 5.73.

INSTRUCCIONES DE CONTROL.

Las instrucciones de control tienen la función de facilitar el flujo de señales de información entre sistema controlado y autómeta, los controladores programables elementales carecen de ellas por lo que su eficiencia en cuanto a respuesta del sistema es baja.

Al igual que otras instrucciones cada fabricante agrega instrucciones particulares a sus equipos cuyo grado de auxilio esta en función de la complejidad de su diseño, las funciones más comunes se realizan con instrucciones similares a las siguiente.

JMP-JME:

Estas instrucciones permiten la ejecución de la zona de instrucciones contenida entre ellas, cuyo inicio lo marca la instrucción JMP y el final JME. Cuando de la operación inmediata anterior a JMP resulta un uno, se ejecuta la zona de instrucciones; si el resultado es cero, el procesador ignora la zona de instrucciones y ejecuta la que sigue a la instrucción JME.

```

STR X7
OR X10
STR C2
OR C3
OUT Y4
STR X14
AND C1
OUT Y6
JMP
STR X9
AND X14
OUT C30
JME
OR STR
OUT Y20

```

← zona de instrucciones
condicionada al valor
lógico de Y6.

En el listado anterior si Y6 tiene un valor lógico de uno, la salida Y20 será el resultado de la operación O de Y6 y C30. Si Y6 tiene un valor cero, la operación Y20 será el resultado de la operación O de las variables de salida externa Y4 y Y6.

IL-ILC:

Permiten la actualización de todas las salidas internas y externas sean actualizadas normalmente, dependiendo de que el resultado inmediato anterior a IL sea uno o cero lógico.

SELECCION DE UNA VARIABLE DE ENTRADA INVERTIDA (NEGADA).

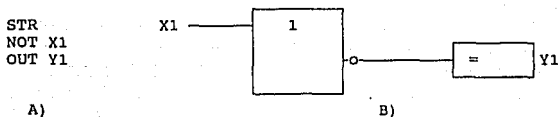


FIGURA 5.75.- SELECCION DE UNA VARIABLE DE ENTRADA NEGADA.

FUNCION LOGICA O.

Esta función se puede aplicar con variables afirmadas o negadas o combinaciones de ambas.

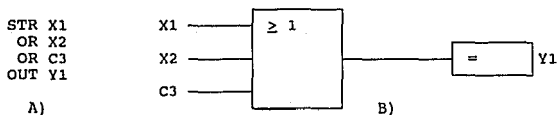


FIGURA 5.76.- SELECCION DE UNA VARIABLE DE ENTRADA NEGADA.

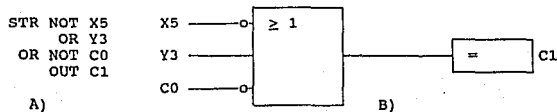


FIGURA 5.77.- SELECCION DE UNA VARIABLE DE ENTRADAS COMBINADAS.

FUNCION LOGICA AND.

Al igual que la función O esta función puede tener señales de entrada directas, negadas o combinación de ambas.

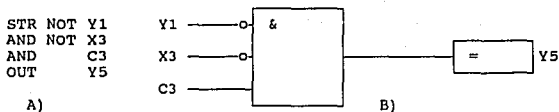


FIGURA 5.79.- SELECCION DE UNA VARIABLE DE ENTRADAS COMBINADAS.

FUNCION LOGICA O DE FUNCIONES Y.

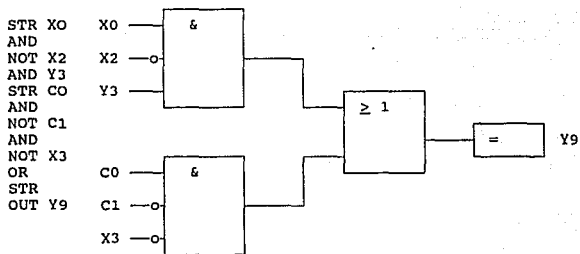


FIGURA 5.80.

FUNCION LOGICA Y DE FUNCIONES O.

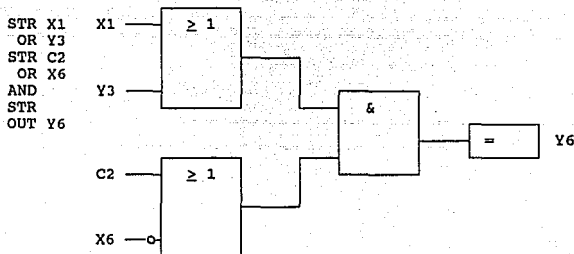


FIGURA 5.81.

TEMPORIZACION.

Al bloque funcional se conectan la variable X1 que temporiza, la variable X0 que restaura el valor a cero y la variable Y0 que transmite la señal de salida
TMR 0

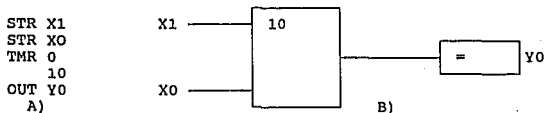


FIGURA 5.82.- FUNCION DE UN TEMPORIZADOR.

FUNCION DE CONTEO.

Opera de forma similar al listado de instrucciones, las variantes de las instrucciones se observan en el programa adjunto al símbolo de función lógica (figuras 5.83 Y 5.84).

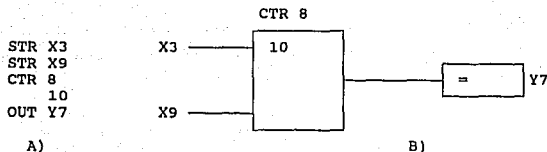


FIGURA 5.83.- CONTADOR ASCENDENTE.

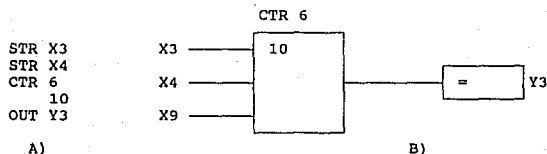


FIGURA 5.83.- CONTADOR DESCENDENTE.

GRAFSET.

El diagrama de flujo como se sabe, es la menor opción que se tiene para especificar un sistema de control, este tipo de diagrama es factible trasladarlo al programa de un PLC en lenguaje de máquina, o cualquier lenguaje de programación que maneje el equipo.

El lenguaje GRAFCET, más que un lenguaje es un método gráfico normalizado capaz de especificar el comportamiento secuencial de un controlador programable.

DEFINICIONES Y SIMBOLOS FUNDAMENTALES DEL GRAFCET.

Fue desarrollado por la Asociación Francesa para la Cibernética Económica y Técnica (AF CET).

SIMBOLOS NORMALIZADOS Y SU COMBINACION.

Permiten representar un diagrama de flujo común a un diagrama de automatización lógica.

El estado interno representado por un círculo se convierte en un cuadrado con un número correspondiente al estado.

En el caso de que se trate de un estado inicial se representa mediante un doble cuadrado al que suele asignarse el número 0.

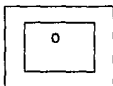


FIGURA 5.90.- REPRESENTACION GRAFICA DEL ESTADO INICIAL.

A cada estado se asignan las acciones que debe realizar el sistema, mediante rectángulos que en el interior especifican cada acción.



5.91.-REPRESENTACION GRAFICA DE LA FORMA DE REPRESENTAR UNA ACCION DE ESTADO INTERNO.

La transición entre estados se representa mediante el enlace de los rectángulos mediante un segmento recto atravesado por una barra a cuyo lado se especifica la capacidad de transición C_t (bandera) que debe tener un uno lógico para autorizar la transición al paso siguiente.

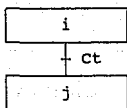


FIGURA 5.92.- TRANSICION DE ESTADOS SECUENCIALES.

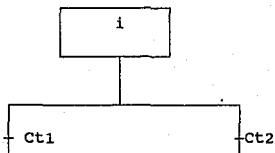


FIGURA 5.93.- TRANSICION DE UN ESTADO A OTROS DOS.

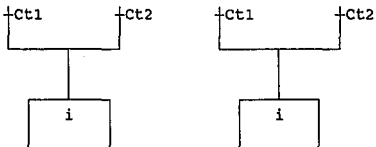


FIGURA 5.94.- TRANSICION DE DOS ESTADOS A OTRO, HACIA ARRIBA.
(CUANDO LA SECUENCIA ES HACIA ARRIBA LA LINEA DE UNION NO LLEVA BARRA ATRAVESADA).

CAPITULO VI

DISEÑO E IMPLANTACION
DE UN
SISTEMA BASICO.

DOCUMENTACION.

La documentación de un sistema de control, se define como la colección ordenada de toda la información concerniente al funcionamiento y operación del equipo y maquinaria del proceso y los componentes de software y hardware del sistema de control.

La realización de los documentos apropiados es generada a través de la recolección de la información de hardware y software con que se cuenta. Una documentación elaborada correctamente debe seguir el siguiente procedimiento:

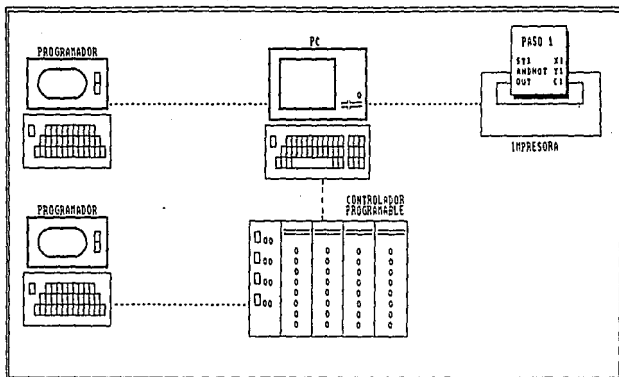
- * Un índice o sumario,
- * La configuración del sistema,
- * Diagrama de conexiones E/S,
- * Asignación de direcciones E/S externas e internas,
- * Registros de asignamiento,
- * Impresión del programa
- * Un módulo de memoria que almacene el programa de control para su carga y reproducción.

El sumario permite visualizar claramente las declaraciones programadas en la tarea de control en cada una de las etapas o pasos secuenciales del programa, la descripción de la filosofía de diseño utilizada para la implementar la solución encontrada al problema de control y los objetivos que deben ser alcanzados por el sistema controlado.

La configuración del sistema es un diagrama de la disposición del sistema que en forma gráfica define la localización, simplificando las conexiones y minimizando los detalles de los componentes del hardware.

El diagrama de conexiones E/S muestra las conexiones actuales de las entradas de campo y dispositivos de salida a los diferentes módulos interfase del autómatas.

La asignación de direcciones E/S externas e internas y los registros de asignamiento son una parte fundamental en la documentación, dado que en ellos se especifica el destino y fuente de las señales internas utilizadas, así como la descripción y uso del registro en que son almacenadas en el controlador programable para su interlocución con el sistema de control.



SALA DE CONTROL

E	I	E	S	S	S	BASTIDOR # 00
N	N	N	A	A	A	64 E/S
T	T	T	L	L	L	PUNTOS: 00-27
X	X	X	I	I	I	CANPO # 4
A	A	A	D	D	D	
D	D	D	A	A	A	
A	A	A				
24	24	24	SIN	SIN	24	24
VAC	VAC	VAC	USO	USO	VDC	VDC

E	E	E	S	S	S	BASTIDOR # 01
N	N	N	A	A	A	64 E/S
T	T	T	L	L	L	PUNTOS: 100-177
X	X	X	I	I	I	CANPO # 5
A	A	A	D	D	D	
D	D	D	A	A	A	
A	A	A				
115	115	115	SIN	SIN	115	115
VAC	VAC	VAC	USO	USO	VDC	VDC

E	E	E	S	S	S	BASTIDOR # 02
N	N	N	A	A	A	64 E/S
T	T	T	L	L	L	PUNTOS: 200-277
X	X	X	I	I	I	CANPO # 24
A	A	A	D	D	D	
D	D	D	A	A	A	
A	A	A				
ANALG	ANALG	ANALG	SIN	SIN	SIN	SIN
8-5	8-20	4-20	USO	USO	USO	USO
VOLTS	MA	MA				

FIGURA 6.1.- ESQUEMA DEL ARREGLO DE UN SISTEMA TÍPICO.

La impresión del programa de control permite la detección de errores en la secuencia de control y la actualización del mismo. Esta impresión es un respaldo por escrito que debe realizarse en cuanto se hace una modificación al programa de control.

El programa de PLC es almacenado generalmente en un dispositivo que permita su reproducción, como puede ser un módulo de memoria EPROM, un cartucho, disco flexible, disco duro, etc.

Este tipo de almacenamiento tiene la función de permitir tener un respaldo del programa y una forma práctica de cargarlo a la CPU del autómata instalado en la planta, dado que normalmente el diseño del programa es desarrollado en un aparato de programación fuera del área de proceso.

La documentación del sistema permite tener información accesible en todo momento permitiendo al personal encargado la fácil comprensión y seguimiento del proceso, así como la reducción del tiempo invertido para encontrar una falla o realizar una modificación en un determinado paso.

Por tanto, la documentación faculta al personal involucrado en el conocimiento del proceso y las herramientas que proporciona el PLC para su control y la solución de problemas de cualquier índole relacionados con el sistema automatizado.

La documentación debe contener en forma elemental la información siguiente:

- * Título del programa,
- * Subtítulos (rutinas o ciclos de control),
- * Fecha de elaboración de la última actualización de la documentación,
- * Numeración de páginas,
- * Comentarios detallados antes y después de cada paso o etapa del proceso.
- * Descripción de elementos o contactos.
- * Dirección de cada contacto o señal E/S,
- * Representación gráfica de cada instrucción del PLC,
- * Número de etapas o pasos,

- * Relación de contactos por etapa o paso,
- * Los valores límite fijados en registros como parámetros de comparación en el programa de control,
- * Relación de señales internas y señales E/S reales.

CONSIDERACIONES SOBRE LA DOCUMENTACION.

Existen diversas opiniones en cuanto a la importancia que puede tener la documentación de un sistema de control y su efecto sobre el mismo.

En forma definitiva el completar los procedimientos y contenidos para tener una buena documentación sobre el sistema de control fuera de la planta de proceso, permite analizar en forma objetiva las variantes y posibles problemas que puede tener la automatización del proceso.

Una buena documentación nos definirá el conjunto de actividades asociadas con el desarrollo del proceso y operación de los dispositivos de campo, así como las características de fabricación que permitirán al PLC controlar las variables que involucra cada dispositivo ya sean señales internas o externas.

En función de la dependencia entre variables proporcionada por la documentación, se genera el programa de control este programa es un software de aplicación que al igual que cualquier programa de computo normalmente presenta errores iniciales, el beneficio de tener una documentación estricta; es que estos errores en el arranque del sistema automatizado serán mínimos y además podrán ser corregidos en forma más accesible.

IMPLEMENTACION.

La implementación de un programa de control tiene un procedimiento que va de la organización hasta el orden en la secuencia de control, un buen número de pasos o etapas incluyen instrucciones de salto dentro de las instrucciones programadas lo que permite, salvo restricciones particulares de la aplicación, la modificación de la secuencia principal.

La organización y control cotidianamente se realiza con las características propias del PLC que se va a utilizar y las características del proceso que se desea automatizar, debido a esto cualquier ejemplo debe tomarse como teórico, puesto que no se esta refiriendo a ningún fabricante en particular.

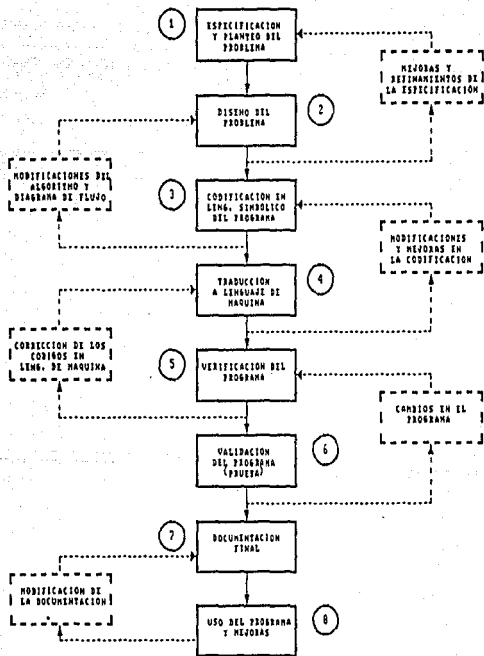


FIGURA 6.2.- CICLO DE DESARROLLO DEL PROGRAMA DE CONTROL.

PROCESO
GRUPO DE UNA O MAS INSTRUCCIONES
QUE EJECUTAN UNA FUNCION DE
PROCESAMIENTO.



ENTRADA/SALIDA
CUALQUIER FUNCION QUE COMPROMETA
LA ENTRADA O SALIDA DE UN
DISPOSITIVO O SEÑALES INTERNAS
DE CONTROL.



DECISION
PUNTO EN EL QUE EL PROCESADOR DEBE
ELIGIR ENTRE DOS ALTERNATIVAS DE
SECUENCIA LOGICA.



PREPARACION
GRUPO DE INSTRUCCIONES QUE PREPARAN
LAS CONDICIONES EXISTENTES PARA
EL PROCESAMIENTO CONSECUTIVO.



PROCESO PREDEFINIDO
GRUPO DE INSTRUCCIONES NO DETALLADAS
EN ESTE DIAGRAMA DE FLUJO COMO LAS
DE UNA SUBROUTINA.



TERMINAL
ORIGEN, FIN O PUNTO DE INTERRUCCION
EN UN PROGRAMA.



CONECTOR
ENTRADA DESDE, SALIDA A: CONECTA DOS
PARTES DE UN DIAGRAMA DE FLUJO.



LINEA DE FLUJO
DIRECCIONA EL FLUJO DEL PROCESO
O FLUJO DE DATOS.



ADNOTACIONES
COMENTARIOS O NOTAS DE EXPLICACION
PARA ACLARAR ALGUNA(S) INSTRUCCIONES.



CONECTOR DE PAGINA
PERMITE CONECTAR LAS PAGINAS DE UN
DIAGRAMA DE FLUJO EXTENSO.



LINEA DE COMUNICACION
INDICA TRANSMISION DE DATOS DE O
INFORMACION DE UN LUGAR A OTRO
MEDIANTE LINEAS DE COMUNICACION.



FIGURA 6.3.- SIMBOLOS COMUNES DEL ALGORITMO DE CONTROL.

DEFINICION DE CONTROL.

El usuario concibe la solución al problema partiendo de la definición que impone el procedimiento que lleva el proceso, el cual marca las variables que se deben controlar, fijando los parámetros de comparación en función de los rangos de trabajo.

La definición clara de la información disponible que involucra el proceso provee la base sobre la cual la operaciones que debe ejecutar el controlador programable son determinadas.

Es importante que las operaciones que requiere la maquinaria y proceso, así como las condiciones previas y posteriores que debe tener la materia prima; sean establecidas por el personal familiarizado con el curso del proceso, con el fin de reducir los errores en el programa de automatización.

La definición de control usualmente se realiza en varios niveles de acuerdo a los departamentos involucrados (producción, electricidad, mecánica, instrumentación, etc.), los cuales deben suministrar aquellas variables del proceso bajo su responsabilidad de forma que la información proporcionada por departamento permita el enlace de las variables del proceso en el proyecto de control.

ESTRATEGIA DE CONTROL.

Una vez definido el problema, se inicia el plan a seguir. El procedimiento comúnmente envuelve determinadas secuencias generadas a partir del curso del proceso. El curso del proceso normalmente se piensa como una secuela de pasos, en la cual cada paso genera una condición que "autoriza" la ejecución del paso siguiente. La parte del desarrollo del programa que se encarga de dictar las condiciones en la secuencia de pasos es un algoritmo de control.

El algoritmo de control debe cumplir con un procedimiento. Como se sabe un algoritmo es uno de los tres tipos de diagrama de flujo que existen: a nivel de concepto, de algoritmo y de instrucción.

Un diagrama de flujo a nivel de concepto, es un gráfico parecido al lenguaje Grafset, dado que utiliza únicamente rectángulos unidos con líneas de dirección de flujo. El diagrama a nivel de instrucciones se construye en base al conjunto de instrucciones básicas y avanzadas que ofrece un fabricante en particular.

El diagrama de flujo a nivel de algoritmo es el más recomendable, dado que involucra a un conjunto de etapas o procesos predefinidos, los cuales son establecidos con el objeto de dar solución a un problema específico.

Un algoritmo es concebido entonces como un proceso lógico que transforma las señales de entrada en señales de salida requeridas por un proceso en particular. El diseño del algoritmo debe contemplar los siguientes aspectos.

El programador debe organizar el control a partir de un conjunto de instrucciones y tratar de producir la respuesta al problema con un número mínimo de las mismas.

En muy pocos casos el desarrollo de un proyecto puede no elaborarse con base en un algoritmo, sin embargo sí con la utilización de esquemas que facilitan su análisis se presentan dificultades; para disminuir los errores en el programa se debe partir de una rígida definición del proceso, para determinar los pasos y la secuencia de los mismos requeridos en el desarrollo del proyecto.

Es terminante que no se puede partir de un patrón común para presuponer un algoritmo, dada la cantidad de ramas y aspectos de la ciencia que pueden ser automatizados, por tanto simplemente se mencionan aquellas instancias que pueden generalizarse.

Una norma fundamental en la estrategia de control es la de efectuar un análisis exhaustivo previo a la programación. Esto permite tener presentes cada una de las alternativas utilizables en la solución del problema, el rango de tiempo de ejecución del algoritmo, adoptando una filosofía de optimización de tiempos y movimientos al diseñar prácticamente sobre la evolución de un proceso para tener un rango de respuesta de las variables del equipo.

Durante la formulación de la estrategia, el programador y/o usuario puede mostrar una nueva aplicación o la modernización del proceso o máquina, dejando muy clara la razón de la elección del procedimiento de programación del sistema de control.

DESARROLLO DEL SOFTWARE.

El desarrollo del software requiere de la consumación de ciertos procedimientos, actividades y tareas, la figura 6.2 intenta esquematizar un método que facilita la comprensión y por tanto la implementación del problema.

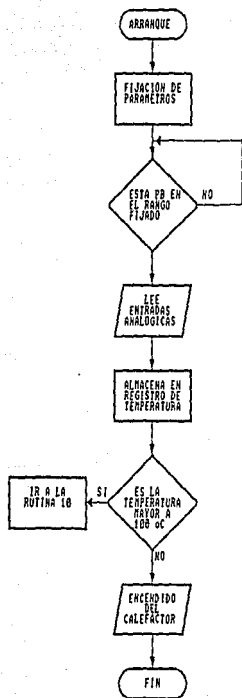


FIGURA 6.4.- EJEMPLO DE UN DIAGRAMA DE FLUJO SENCILLO.

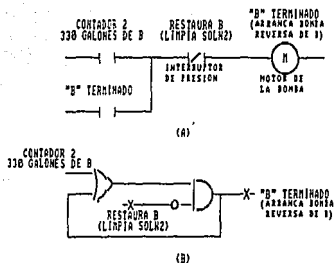


FIGURA 6.5.- REPRESENTACIONES:(A) DIAGRAMA DE ESCALERA DEL PLC (B) CUERPUAS LOGICAS.

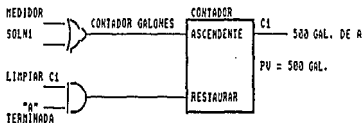


FIGURA 6.6.- COMBINACION DE LOS METODOS DE SIMBOLOGIA DE CONTACTOS Y DIAGRAMA DE FUNCIONES.

Planteo y especificación del problema: la fase más importante es precisamente el planteamiento del problema, es el punto de partida de todas las acciones siguientes. Es en esta etapa donde se define el problema mediante la relación de las distintas funciones que se deben realizar, aclarando las condiciones para su ejecución.

Las condiciones deben ser precisas, explícitas y redactadas de manera que sea fácil su comprensión para el usuario final. Deben especificarse todas las E/S tanto internas como externas dependientes de las condiciones, así como sus respectivas localidades de memoria.

Es conveniente incluir un método que defina el comportamiento del sistema cuando se presentan errores o fallas, ya sea de programación o de respuesta del equipo al mandato del autómata.

Diseño del programa: en esta etapa se diseña el plan para abordar el problema una vez que ha sido definido. Existen diferentes técnicas de programación que pueden facilitar el desarrollo del programa tales como programación estructurada, diseño TOP-DOWN, diseño modular, etc. Durante esta fase el programador puede partir de algún programa ya existente para su modernización o actualización.

Codificación del programa en lenguaje simbólico: Esta fase consiste en la transcripción del programa diseñado por un gráfico como el algoritmo a un lenguaje típico del controlador programable (simbología de contactos, lista de instrucciones, diagrama de funciones, etc).

Se le llama lenguaje simbólico por ser un interprete entre programador y autómata dado que el lenguaje que entiende el procesador es el basado en el sistema binario.

Traducción al lenguaje de máquina: los códigos utilizados en cualquier lenguaje simbólico no son reconocidos por el procesador, esto significa que se debe traducir el programa de control al lenguaje de unos y ceros que es el manejado internamente en el controlador programable.

Cada instrucción conocida también como código mnemónico tiene una representación en binario para el autómata de acuerdo a su manufactura, en algunos casos es necesario traducir el programa de control a su representación en binario y grabarlo en el módulo de memoria en el que se va a guardar el programa, ya sea para su continua lectura durante la ejecución o como parte de la documentación.

Verificación del programa: Como se mencionó antes cualquier programa puede presentar errores aún cuando se haya elaborado con esmero, algunos controladores cuentan con programas editores que permiten la localización sencilla de los errores originados en el programa de control.

Es interesante recordar que si se ha creado la parte de la documentación que contiene las direcciones de E/S se podrá localizar un error y eliminar en forma sencilla su influencia en otras etapas del programa.

Prueba y validación del programa: En esta etapa se localizan los errores que se presentan en el arranque, es conveniente tener presente que como estamos hablando del programa para un controlador industrial, se deben considerar básicamente dos tipos de errores los de programa y los originados por fallas de los dispositivos electromecánicos.

En un monitoreo, se muestra el estado de los dispositivos de campo que involucran el proceso, estos estados son la llave que permite que el programa de control se ejecute secuencialmente, se mantenga en un "loop" o lazo cerrado de control o ejecute cualquier otra acción programada para las condiciones leídas en el último monitoreo.

Lo anterior quiere decir que se puede presentar una acción no deseada en alguna etapa, debido a alguna condición contemplada incorrectamente en la programación o si algún dispositivo no presenta la condiciones requeridas en la etapa en cuestión.

La validación del programa se hace cuando en la prueba del mismo se han obtenido las salidas deseadas en respuesta a las condiciones que se presentan en la evolución del proceso.

Documentación final: En las etapas previas se desarrolla una parte fundamental de la documentación, después de haber verificado y validado el programa viene entonces la preparación de la documentación final en donde se prepara paso a paso toda la información generada en el proyecto desde su concepción como autómatas hasta el momento en que el programa de control satisface los requerimientos del planteamiento del problema.

Esta documentación incluye normalmente en detalle las condiciones de la evolución del proceso, equipo disponible, especificaciones, necesidades, algoritmo de control, memoria de datos E/S (localidades E/S), planos del proceso, etc.

La figura 6.2 resume en bloque el procedimiento que debe seguirse en la elaboración de la documentación.

Es conveniente revisar minuciosamente la secuencia de pasos para abarcar todas aquellas posibilidades que permitan la simplificación del programa de control y las etapas que han de conformarlo.

Con un programa simplificado se optimizan las E/S consideradas y el número de dispositivos de campo que deben ser controlados.

GUIA DE IMPLEMENTACION.

Un PLC es un potente mecanismo que sólo puede automatizar aquello que es concebible, es decir; cualquier evento proceso o procedimiento generado por la ciencia actual puede ser objeto de automatizarse con un controlador lógico programable.

El éxito del programa de control de un autómatas depende de la organización y estructura dada por el programador y/o usuario. Son muchas las formas en que un problema es analizado, pero sí la aplicación se acerca sistemáticamente, la probabilidad de errores al arranque disminuye.

La técnica utilizada para la implementación del programa es subjetiva en cuanto a las características particulares que la aplicación exige de la programación, sin embargo es conveniente basarse en una guía que facilite el desarrollo del mismo (tabla 6.1).

NUEVAS APLICACIONES	MODERNIZACION
<ul style="list-style-type: none"> * COMPRENSION DE LA DESCRIPCION DE LA OPERACION DEL SISTEMA. * REVISION DE LOS METODOS FACTIBLES DE CONTROL Y OPTIMIZACION DE LA OPERACION DEL PROCESO. * DIAGRAMA ESQUEMATICO DEL PROCESO * IMPLEMENTACION DEL DIAGRAMA DEL PROCESO MEDIANTE UN DIAGRAMA LOGICO O ALAMBRADO DE RELEVADORES. * ASIGNACION DE DIRECCIONES A E/S REALES O INTERNAS. * TRASLACION DE LA LOGICA DE CONTROL A UN LENGUAJE PROPIO DEL AUTOMATA. 	<ul style="list-style-type: none"> * COMPRENSION DEL ACTUAL PROCESO Y FUNCION DE LA MAQUINARIA. * REVISION DE LOS MECANISMOS LOGICOS DE OPERACIONES Y LA OPTIMIZACION CUANDO SEA POSIBLE. * ASIGNACION DE DIRECCIONES A E/S REALES E INTERNAS. * TRASLADAR DEL DIAGRAMA DE ALAMBRADO DE RELEVADORES A UN LENGUAJE PROPIO DEL AUTOMATA.

TABLA 6.1.- GUIA DE ACERCAMIENTO.

ORGANIZACION DEL PROGRAMA E IMPLEMENTACION.

Esta sección pretende dar pauta para la organización de los detalles que deben consumarse antes que arranque cualquier programa de control. Una buena organización es la clave de un buen programa, una vez que es tiempo de generar la programación e implementación de la solución dada al sistema de control.

El programador debe conocer perfectamente el proceso que se va a controlar para elegir correctamente maquinaria y equipo complementario (hardware y software).

Una vez que los detalles preliminares son cubiertos, se puede iniciar un esbozo de la solución. Todo el tiempo invertido en este trabajo es muy fructífero si se considera que la minuciosidad de su elaboración reducirá notablemente la aparición de errores al arranque.

EL DIAGRAMA DE FLUJO Y LA GENERACION SECUENCIAL DE SALIDAS.

El diagrama de flujo es una técnica de análisis de un problema que permite su estudio mediante la representación gráfica de las diferentes etapas que componen el o los procesos del sistema su función principal consiste en proporcionar a la documentación un gráfico que contenga las segmentaciones lógicas que dan respuesta al curso del proceso (fig 6.3).

El diagrama de flujo describe el proceso de operación en forma secuencial. Un diagrama de flujo se representa en la figura 6.4, cada paso en el diagrama representa la ejecución de una operación, una entrada/salida, una decisión o procesamiento de datos.

Una vez que el diagrama de flujo es completado, la secuencia lógica puede obtenerse en uno de dos caminos

- (a) mediante compuertas lógicas especificando las condiciones de entrada reales o internas pueden ser usadas para acceder la salida específica de una secuencia.
- (b) por medió del diagrama de contactos del PLC que va a utilizarse directamente para implementar la lógica necesaria para representar la salida de una etapa (la figura 6.5 ilustra los dos métodos).

El programador puede combinar los métodos de diagrama de funciones y simbología de contactos como se muestra en la figura 6.6, sin embargo, en algunos controladores sea conveniente utilizar únicamente el método de diagrama de funciones por mostrar mayor claridad de acuerdo al tipo de programa.

Las entradas y salidas marcadas con una x en la figura 6.5 B se puede utilizar para indicar E/S reales del sistema, si no hay marca significa entonces que se trata de señales internas de control.

La designación para las señales de entrada actuales puede ser el dispositivo actual o caracteres simbólicos que son asociados con cada dispositivo de campo. Es conveniente agregar una breve explicación de la función en el programa relacionando las variables con la operación del proceso.

CONFIGURACION DEL SISTEMA DE PLC.

En una aplicación de PLC, es deseable como en muchas funciones de sistemas de computo, una concepción sólida de las necesidades del sistema, considerándolas al mismo tiempo durante la ejecución. En la configuración del PLC se define cual módulo E/S va a ser utilizado con cada tipo de señal de entrada o salida y donde esta localizado el modulo de referencia en el bastidor.

Recuerde que la localización de un módulo será definida como la dirección o localidad de memoria de aquellas entradas y salidas que son llamadas para utilizarse en el programa de control.

La elaboración de la configuración debe hacerse en forma óptima, considerando los aspectos posibles para futuras expansiones todos los tipos de módulos de uso común o especial, tomando en cuenta las posibles fallas de alambrado que pudieran presentar por el tipo de uso y ubicación en los bastidores. Todos estos detalles ayudarán a llevar a cabo el diseño del sistema.

Las direcciones asignadas incluyen dispositivos de uso interno como registros, contadores y temporizadores, etc. Las direcciones de tipo externo se asocian con la ubicación del módulo interfase conectado al autómata.

Ciertas partes del sistema son de alambrado externo por razones de seguridad. Elementos que producen fluctuaciones de voltaje considerables como paros de emergencia, arranque maestro a través de un interruptor de pulso, estos elementos se conectan al autómata mediante circuitos con separación física para protección contra sobrecorrientes.

La traslación o codificación de un programa es el paso en el que se escribe o reescribe la lógica o diagrama de relevadores en un programa formado por instrucciones de un lenguaje característico del PLC que se va a utilizar.

Hay que recordar que la forma en que se programan los contactos normalmente abiertos o cerrados, depende de la respuesta que se espera de ellos, es decir; la forma como se requiere que operen en la lógica del programa de control.

Casos especiales de la programación de dispositivos de entrada incluyen la programación de dispositivos de señales normalmente cerradas (apagadas), traslación de circuitos maestro de relevadores (MCR), contactos instantáneos de señales de temporización, y dispositivos de entrada que proporciona señales de entrada internas o bien externas al circuito MCR.

Las señales de entrada analógica no lineal, pueden utilizarse en los sistemas de control utilizando rutinas de programación basadas en métodos de interpolación mediante el levantamiento continuo de tablas de datos de valores premedidos.

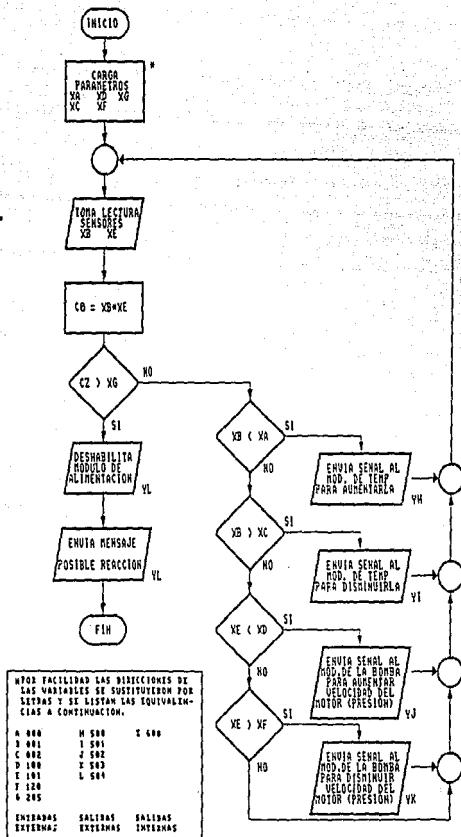
PROBLEMA DE APLICACION.

Un proceso requiere mantener continuamente la temperatura y presión de un fluido circulante bajo un rango de trabajo, dado que si supera al producto de sus valores absolutos a una constante fijada, existe el riesgo de reacción en la sustancia de trabajo, por lo que si se presenta este caso se debe suspenderse inmediatamente la energía.

La presión puede controlarse mediante un módulo interfase con dos entradas de habilitación, capaz de manipular el reostato del motor de una bomba, el módulo reacciona cuando se alimentan una, otra o ambas entradas de control: menor que (<) y mayor que (>).

Cuando se alimenta la entrada menor que se aumenta la velocidad del motor, si la entrada alimentada es la de mayor que entonces la velocidad del motor es disminuida. Cuando ninguna de las entradas es alimentada, entonces la velocidad se mantiene constante.

La temperatura puede controlarse mediante un módulo interfase de dos entradas de habilitación, capaz de manipular un potenciómetro de control de temperatura, el módulo reacciona cuando se alimentan una, otra o ambas entradas de control: menor que (<) y mayor que (>).



NO SE FACILITARON LAS DIRECCIONES DE LAS VARIABLES SE SUSTITUYERON POR LETRAS Y SE LISTAN LAS EQUIVALENCIAS A CONTINUACION.

A 000	M 500	Z 400
B 001	N 501	
C 002	O 502	
D 100	P 503	
E 101	Q 504	
F 120		
G 205		

ENTRADAS SALIDAS SALIDAS
EXTERNAZ EXTERNAS INTERNAS

FIGURA 6.7.- ALGORITMO DE CONTROL DE SECUENCIA DE OPERACIONES CORRESPONDIENTE A LA APLICACION.

Cuando se alimenta la entrada menor que se aumenta la temperatura, si la entrada alimentada es la de mayor que entonces la temperatura es disminuida. Cuando ninguna de las entradas es alimentada, entonces la temperatura se mantiene constante.

El controlador tomará las lecturas de presión y temperatura mediante dos sensores conectados a un módulo de entradas analógicas que las convierte en señales digitales que el procesador reconozca.

DEFINICION DEL PROBLEMA.

Objetivo: mantener la temperatura X001 en el rango $X000 < X001 < X002$ y la presión X101 en el rango $X100 < X101 < X120$, si el producto de la temperatura y la presión $X001 * X101$ excede el parametro X205 entonces se debe suspender la energía.

Señales de entrada: Las señales de entrada al controlador son conectadas desde los módulos interfase y las variables y sus direcciones asignadas son las siguientes.

X000	LIMITE INFERIOR DEL RANGO DE TEMPERATURA
X001	LECTURA ACTUAL DE TEMPERATURA
X002	LIMITE SUPERIOR DEL RANGO DE TEMPERATURA
X100	LIMITE INFERIOR DEL RANGO DE PRESION
X101	LECTURA ACTUAL DE PRESION
X120	LIMITE SUPERIOR DEL RANGO DE PRESION
X205	PARAMETRO DE COMPARACION DEL PRODUCTO $X001 * X101$

Señales de salida: son las señales que se necesita alimentar a los módulos para el control de los dispositivos de campo, las variables y su direcciones asignadas se listan a continuación.

C600	SEÑAL DE SALIDA	INTERNA QUE INDICA EL PRODUCTO $X001 * X101$
Y500	SEÑAL DE SALIDA	EXTERNA QUE INDICA $X001 < X000$
Y501	SEÑAL DE SALIDA	EXTERNA QUE INDICA $X001 > X002$
Y502	SEÑAL DE SALIDA	EXTERNA QUE INDICA $X101 < X100$
Y503	SEÑAL DE SALIDA	EXTERNA QUE INDICA $X101 > X120$
Y504	SEÑAL DE SALIDA	EXTERNA QUE INDICA LA SUSPENSION DE LA ENERGIA Y ACTIVA INDICACION DE POSIBLE REACCION

Requerimientos del proceso: Se refiere a la fijación de las condiciones de trabajo bajo los cuales se requiere mantener la temperatura y la presión y cuyas variables representativas se indican en el apartado para las entradas. Las necesidades generadas por el proceso se listan a continuación.

- * Se debe grabar en la memoria de aplicación los parámetros X000, X002, X100, X120, X205
- * El monitoreo se encarga de actualizar los valores de temperatura y presión mediante la lectura de los sensores correspondientes (X001, X101)
- * Si la temperatura se sale del rango, se envía una orden al módulo de control de temperatura para su restablecimiento.
- * Si la presión se sale del rango, se envía una orden al módulo de control de presión para su restablecimiento.
- * Después de cada actualización se debe efectuar el producto X001*X101 para compararse con el parámetro X205, en caso de superarlo se debe generar una señal de suspensión de la energía por posible reacción.

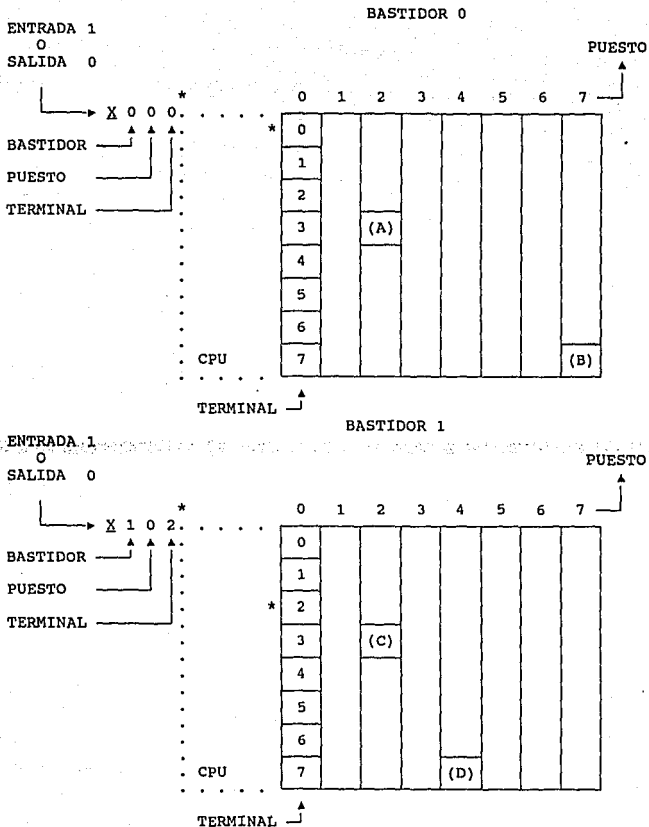


FIGURA 6.8: INFORMACION QUE CONTIENE LA DIRECCION: (A) X023, (B) X077, (C) X123 Y (D) X147. LA X = 0 INDICA SALIDA Y LA X = 1 ENTRADA.

PROGRAMACION.

La programación del problema se hará con las instrucciones del diagrama de escalera vistas en forma general en el capítulo anterior.

Se debe tener presente que en un problema real las instrucciones a utilizar son las del conjunto de instrucciones básicas o avanzadas que maneja el equipo PLC que se va a utilizar.

Las direcciones que se dan a los elementos que forman el diagrama de escalera, se asignan en función de la localización de los registros en la instalación, tomando como puntos de referencia la ubicación del bastidor, el puesto y la terminal correspondiente al registro que contiene la variable que se requiere para la operación, la cual es llamada o enviada de acuerdo a su dirección (ver figura 6.8) a la rutina donde se utiliza su referencia.

El siguiente programa es realizado en forma general, la figura 6.9 muestra la etapa del programa en donde se realiza la el análisis y control de las variables, las figuras 6.10 y 6.11 muestran las subrutinas programadas para los lazos de control de temperatura y presión.

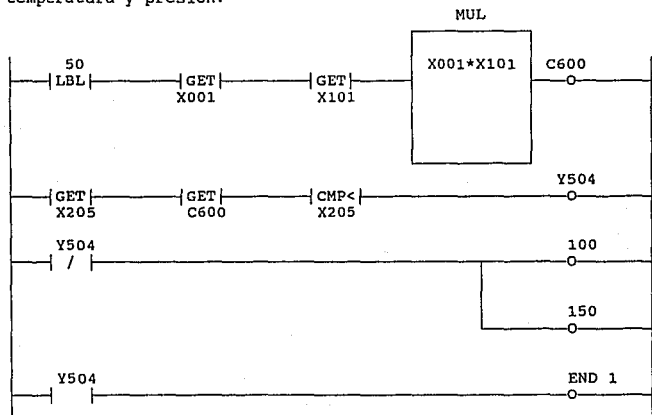


FIGURA 6.9.- Rutina principal del programa regulación T-P que contempla los objetivos del problema de aplicación.

En la figura 6.9, la instrucción LBL 50 sirve para indicar una etiqueta de referencia que permite dirigir el salto de una instrucción a esta rutina de programa.

La instrucción de multiplicación en formato de bloque requiere de una carga previa de los registros que se van a multiplicar, esta carga se realiza mediante la instrucción GET, el bloque funcional ejecutará la multiplicación de los dos registros cargados en forma inmediata anterior a su llamado. generando una salida interna C4 para compararse con la constante de fijada por las condiciones del programa (X205).

La operación de comparación de la figura 6.9 funciona de la siguiente forma: compara si el contenido del registro que guarda la variable C600 es mayor que el contenido del registro que almacena la variable X205, si se cumple la condición la salida Y504 se encenderá.

Para finalizar la rutina, se observa que cuando no esta presente Y504 se ejecutaran simultáneamente las etapa del programa etiquetadas con 100 y 150 como referencia del salto.

En la figura el procedimiento de comparación es similar la instrucción CMP< indica que si la variable X001 es menor que la variable X0 se active la salida Y500, las comparaciones siguientes realizan el mismo procedimiento.

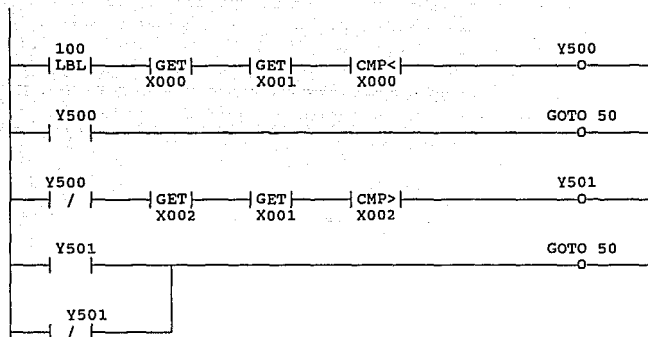


FIGURA 6.10.- RUTINA DEL PROGRAMA EN DONDE SE CONTROLA LA TEMPERATURA.

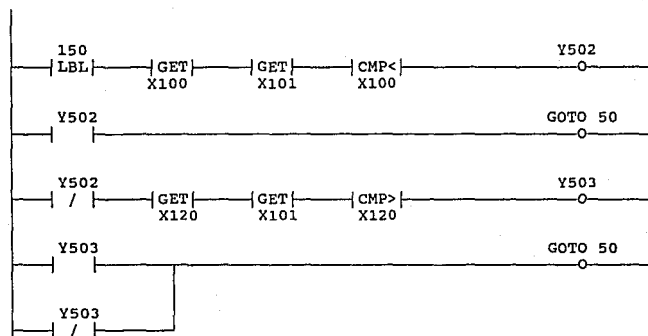


FIGURA 6.11.- RUTINA DEL PROGRAMA EN DONDE SE CONTROLA LA PRESION.

Haciendo un análisis de las direcciones utilizadas que la variable de entrada X000 se ubica dentro del sistema de la siguiente forma:

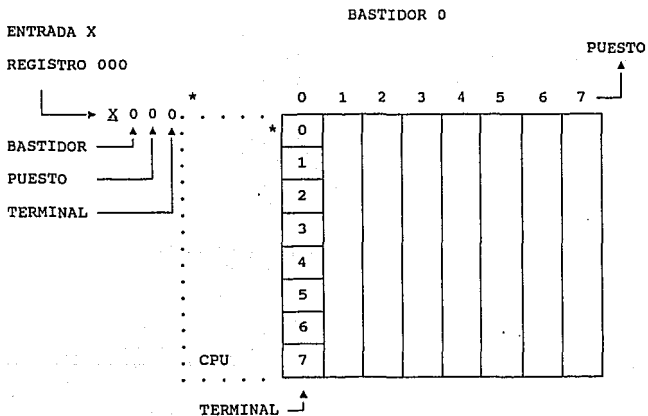


FIGURA 6.12.- INFORMACIÓN QUE CONTIENE LA DIRECCIÓN DE LA variable X000.

De igual forma si se quisiera ubicar la variable X205 en la figura 6.13 se ubicaría en el bastidor 2, puesto 0 y terminal 5.

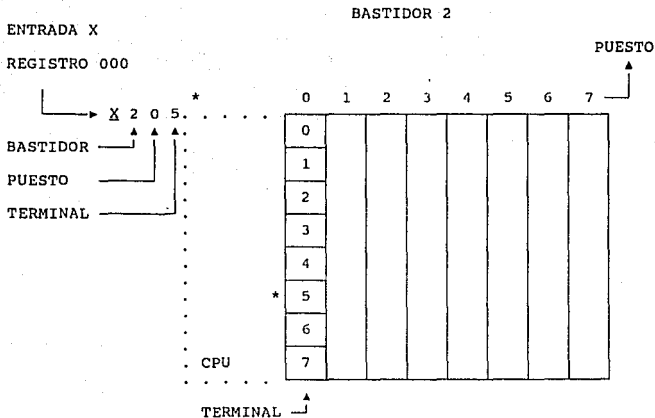


FIGURA 6.13.- INFORMACION QUE CONTIENE LA DIRECCION DE LA VARIABLE X205.

CONCLUSIONES FINALES.

Son muchas las cualidades que se podrían mencionar sobre los controladores programables y prácticamente ningún proceso quedaría fuera de su alcance. El desarrollo de la microelectrónica en todas las áreas promete equipos cada vez más compactos, quizá en un futuro no muy lejano se cuente con PLC portátiles capaces de ser controlados cotidianamente por individuos que bajo la etiqueta de minusválidos carecen de algún miembro.

Actualmente el PLC se introduce rápidamente en la industria del país, sin embargo el conocimiento de su estructura física y cualidades de control no pueda considerarse del dominio popular, de hecho la información básica no se encuentra en una buena parte del acervo que ofrecen las escuelas de nivel superior que imparten carreras del área de ingeniería.

El presente trabajo pretende exponer de una forma sencilla todos aquellos conceptos elementales de los controladores lógico programables, para permitir a toda persona interesada en el tema, su fácil introducción al manejo de los PLC comerciales, bajo la consigna de que para abarcar todos los aspectos de la automatización se debe complementar esta información profundizando en los temas relacionados con los módulos de funciones especiales y manejo de sistemas de instrumentación.

BIBLIOGRAFIA.

APUNTES: SIM-A1 STEP 5 BASICO. México, SIEMENS, Centro de entrenamiento, 1990. Aptdos 1 Y 4.

E.Mandado, J. M. Acevedo, A. Perez López. CONTROLADORES LOGICOS Y AUTOMATAS PROGRAMABLES, España, Marcombo 1992. Parte 3.

SIEMENS S5 STEP 5. Alemania, SIEMENS. Referencia C79000-B8576-C640-01.

CURSO SIG, NOCIONES FUNDAMENTALES DE SIMATIC S5, México, SIEMENS, Centro de Formación de Mando y Accionamientos, s.f. Aptdos 1.5-1.7.

Charles J. Baer. ELECTRICAL AND ELECTRONICS DRAWIND. Estados Unidos. Mc Graw Hill 1973. Pag 342-350.

A.J. Khambata. MICROPROCESADORES/MICROCOMPUTADORES, ARQUITECTURA, SOFTWARE Y SISTEMAS. España, Gustavo Gili 1987. Cpts1,2,3,7,15.

Kenneth L. Short. MICROPROCESADORES Y LOGICA PROGRAMADA. España, Gustavo Gili 1985. Cpts 2 y 3.

M. Morris Mano. INGENIERIA COMPUTACIONAL, DISEÑO DE HARDWARE. México, Prentice-Hall Hispanoamericana 1991. Cpts 2,3 y 6.

L.A. Bryan, E.A. Bryan. PROGRAMMABLE CONTROLLERS, THEORY AND IMPLEMENTATION. Estados Unidos, Industrial Text 1988. Cpts.1,3,4,5,6,7,8,9,10 y 11.

L.A. Bryan, E.A. Bryan. PROGRAMMABLE CONTROLLERS, WORKBOOK AND STUDY GUIDE. Estados Unidos, Industrial Text 1988. Estados Unidos, Industrial Text 1988. Pags 1-113.