

TUES
1504
E32d
1994
Ej.2

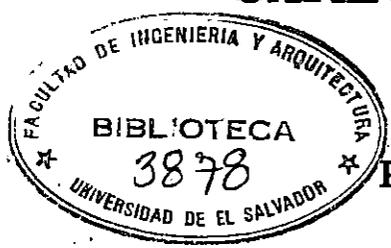
UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA



**" DISEÑO Y CONSTRUCCION DE UN SISTEMA
CONTROLADOR PROGRAMABLE DE
PROCESOS INDUSTRIALES "**

TRABAJO DE GRADUACION PRESENTADO POR :

JORGE HORACIO EGUIZABAL CASTANEDA
CARLOS ALFREDO GOMEZ ZELAYA



15101218
15101218

PARA OPTAR AL TITULO DE :

INGENIERO ELECTRICISTA

JUNIO 1994

SAN SALVADOR, EL SALVADOR, CENTRO AMERICA

R/21/07/94



UNIVERSIDAD DE EL SALVADOR

RECTOR:

DR. FABIO CASTILLO FIGUEROA

SECRETARIO GENERAL:

LIC. MIRNA ANTONIETA PERLA DE ANAYA

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO:

ING. JOAQUIN ALBERTO VANEGAS AGUILAR

SECRETARIO:

ING. JOSE RIGOBERTO MURILLO CAMPOS

ESCUELA DE INGENIERIA ELECTRICA

DIRECTOR:

ING. SALVADOR DE JESUS GERMAN



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA

Trabajo de graduacion previo al grado de:
Ingeniero Electricista

TITULO: "DISEÑO Y CONSTRUCCION DE UN SISTEMA
CONTROLADOR PROGRAMABLE DE PROCESOS INDUSTRIALES"

Presentado por:

JORGE HORACIO EGUIZABAL CASTANEDA
CARLOS ALFREDO GOMEZ ZELAYA

Trabajo de graduacion aprobado por:

Coordinador :

Ing. Ricardo Ernesto Cortéz

ESCUELA DE INGENIERIA ELECTRICA
FACULTAD DE INGENIERIA
Y ARQUITECTURA
Universidad de El Salvador

Asesor:

Ing. Ricardo Arnaldo Cruz Parada

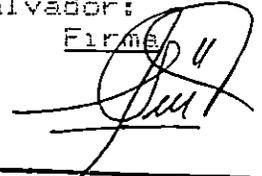
San Salvador, Junio de 1994

ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, 11 de Junio de 1994,
en el local de Sala de Lectura de la Escuela de Ing. Eléctrica,
a las 10:00 horas, con la presencia de las siguientes autoridades de
Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

- 1- Ing. Salvador de J. German
Director
- 2- _____
- 3- _____

Firma



Y con el Honorable Jurado de evaluación integrado por las personas
siguientes:

- 1- Ing. Carlos Mauricio Erroa Colato
- 2- Ing. Gerardo Marvin Jorge Hernández
- 3- Ing. René Naúm Clímaco Cortez
- 4- _____
- 5- _____
- 6- _____



Se efectuó la defensa final reglamentaria del Trabajo de
Graduación: "DISEÑO Y CONSTRUCCION DE UN SISTEMA CONTROLADOR PROGRAMABLE DE PROCESOS
INDUSTRIALES"

a cargo del (los) Br(es): Jorge Horacio Eguizábal Castaneda
Carlos Alfredo Gómez Zelaya

Habiendo obtenido el presente trabajo una nota final, global de 8.4
(Ocho punto cuatro.)

TRABAJO DEDICADO A:

El presente Trabajo de Graduación lo dedico a todas las personas que me dieron el apoyo moral y el estímulo necesario para llevarlo a un feliz término, en especial:

A DIOS TODOPODEROSO:

Por darme la vida y guiarme por un buen camino, por darme tranquilidad en los momentos de flaqueza, por darme la oportunidad de darle a mis seres queridos la satisfacción de ver terminados mis estudios superiores.

A MIS PADRES :

Jorge Horacio Eguizábal Aguirre y Silvia Concepción Castaneda Cardona, por confiar en mí; por darme la oportunidad de poder superarme y porque me siento orgulloso de ellos.

A MIS HERMANDOS:

Humberto Horacio, Alba Yesenia y Juan Francisco Eguizábal Padilla, por darme un constante apoyo fraternal, a quienes deseo con gran sinceridad alcancen sus metas propuestas y que DIOS les bendiga.

A MIS FAMILIARES :

En especial a mis abuelas Clara Aguirre (Q,D,D,G) e Irene Cardona, por mantener una constante preocupación por mi bienestar en toda mi vida. A mis tias Dolores y Ana del Carmen Eguizábal Aguirre, por proporcionarme un enorme apoyo, cariño y comprensión en todos los sentidos. A mi primo Fidel Arturo López Eguizábal por compartir momentos de alegría y preocupación, por darme siempre un gran apoyo.

A MIS COMPAÑEROS Y AMIGOS:

Por el apoyo y ayuda desinteresada en todo momento de nuestra carrera, en especial en el transcurso de nuestro Trabajo de Graduación.

JORGE HORACIO

TRABAJO DEDICADO A:

A DIOS OMNIPOTENTE: Sin quien no hubiera podido obtener este triunfo académico.

A MIS PADRES: Carlos Alfredo Gómez Castro y Elsy Zelaya de Gómez.

Con reconocimiento y gratitud profunda, por su desinteresado amor, esfuerzo y sacrificio;

A MIS HERMANOS Yolanda Josefina y Victor René.

En fraternal afecto e incomparable reconocimiento, por el oportuno y constante apoyo; para cristalizar en realidad mi gran anhelo.

A MIS FAMILIARES,
COMPAÑEROS Y AMIGOS: Por el oportuno estímulo, a efecto de que no desfalleciera en mi noble empresa, de obtener mi investidura profesional. A todos ellos dedico este triunfo.

C A GZ

AGRADECIMIENTOS

Presentamos nuestros agradecimientos a las siguientes personas por su desinteresada colaboración en el desarrollo de este trabajo.

Agradecemos sinceramente al Ingeniero Ricardo Cortéz, al Señor Salvador Posada, por su valiosa ayuda, aportes desinteresados y asesoría durante el tiempo que transcurrió la realización del trabajo.

Agradecemos mucho el apoyo y ayuda invaluable y oportuna de todos nuestros compañeros de estudio durante el transcurso de nuestra carrera; y muy especialmente en el desarrollo del presente trabajo a: José Edgardo Mauricio Gutiérrez Rivas, Mario Alberto Monge Ramos, Gerbert Orlando Mancía Pérez, Oscar René López Fuentes, ya que su aporte fué dado en los momentos más críticos; demostrando nuestros compañeros un verdadero espíritu universitario, tan característico de los estudiantes de ingeniería eléctrica.

PREFACIO

Los PLCs fueron desarrollados a principios de la decada de los 70s, siendo utilizados tempránamente para la industria automotriz y químicas, donde reemplazaron grandes paneles de control constituidos por relés; que rapidamente fueron removidos por los nacientes autómatas programables debido a su flexibilidad en términos de modificación de secuencia de control. Con lo cual se logró alcanzar una de las metas de la automatización que es reducir costos de producción y mejorar la calidad del producto.

En nuestro país el desarrollo de la automatización por medio de los controladores programables, ha sido bastante lento, siendo su precio una de las principales razones de ello; a la par de que se cuenta con poco personal capacitado para su uso; ademas de contar con una escasa, practicamente nula, documentación sobre el tema, limitándose a manuales del tipo técnico que solo ofrecen una vaga idea de la filosofía de los mismos.

El objetivo de este trabajo es diseñar y construir un controlador lógico programable sencillo, que pueda ser utilizado para pequeñas tareas de control tanto combinacional como secuencial, que suprima la presencia humana en tareas tediosas o peligrosas, a la vez que se presenta como una alternativa funcional que sirva para el conocimiento de estos sistemas, por parte de los estudiantes de la EIE, ya que ademas de sistema construido, se presenta este documento como una base bibliográfica suficiente para quienes deseen conocer sobre automatismos programables.

Los alcances logrados cumplen con los objetivos planteados, finalizando el trabajo con la construcción de el PLC y desarrollo de algunos pequeños programas de ejemplo.

RESUMEN DEL TRABAJO

El presente trabajo se desarrolla a partir de una explicación y concepción de lo que es la participación de los autómatas lógicos programables en los procesos de control industriales.

El capítulo I se presenta como una introducción de lo que es la filosofía de los autómatas programables y de los sistemas de control en general, dando un marco de referencia en base al cual se presentan todo el trabajo.

El segundo capítulo se describe como está constituido físicamente el PLC, específicamente los mini autómatas, partiendo de ello para fundamentar la filosofía de el diseño propio.

En el tercer capítulo se plantean criterios sobre la programación y herramientas para para el diseño de programas de aplicación presentándose el criterio fundamental de el porque se programa en el lenguaje del Z80.

Por último se proporciona un pequeño manual de usuario para que describe funcionalmente como trabaja el sistema ya montado en su caja un ejemplo de aplicación.

INDICE

CAPITULO	PAGINA
I INTRODUCCION A LOS SISTEMAS DE CONTROL	
INTRODUCCION	1
1.1 ANTECEDENTES SOBRE-AUTOMATIZACION Y CONTROL	1
1.1.1 ENTRADAS	3
1.1.2 SALIDAS	3
1.1.3 SECCION DE PROCESAMIENTO	4
1.1.3.1 MAQUINA CABLEADA.....	5
1.1.3.2 CONTROL PROGRAMABLE.....	6
1.1.4 SISTEMAS DIGITALES Y ANALOGOS	7
1.2 TIPOS DE PROCESO INDUSTRIAL	7
1.2.1 PROCESO CONTINUO	8
1.2.2 PROCESO CONJUNTO	8
1.2.3 PRODUCCION DE MODULAR	9
1.3 CONTROL DE PROCESOS POR COMPUTADORA	10
1.4 CONTROLADORES LOGICOS PROGRAMABLES (PLC)	11
1.5 CRITERIOS DE SELECCION DE PLC SOBRE OTRAS TECNOLOGIAS	15
CONCLUSIONES	17
BIBLIOGRAFIA	18
II DISEÑO DEL HARDWARE DEL PLC	
INTRODUCCION	19
2.1 DISEÑO DEL HARDWARE	20
2.1.1 UNIDAD CENTRAL DE PROCESOS	20
2.1.2 MEMORIA	21
2.1.3 UNIDADES DE ENTRADA SALIDA	22
2.2 TIPOS DE SISTEMAS PLC	22
2.2.1 PEQUEÑOS PLCs	23
2.2.2 MEDIOS Y GRANDES PLCs	24
2.3. ARQUITECTURA DE LA CPU	25
2.3.1 SISTEMA MINIMO BASADO EN EL Z-80	28
2.3.2 CIRCUITO DE RELOJ	29
2.3.3 CIRCUITO DE PASO UNICO DE INSTRUCCION	30
2.3.4 CIRCUITO DE RESET	30
2.3.5. BUFFERING DE SALIDA DE CONTROL, BUSES DE DATOS Y DE DIRECCIONES	32
2.3.6 DECODIFICACION DE MEMORIAS Y PUERTOS DE ENTRADA/SALIDA	33
2.3.7 PUERTOS DE ENTRADA Y SALIDA	36
2.3.8 MEMORIA	36
2.4 MODULOS DE ENTRADA Y SALIDA	38
2.4.1 MODULO DE ENTRADA	39
2.4.2 MODULO DE SALIDA	41
2.4.2 LOS CAPTADORES	43
2.4.3 ACCIONADORES Y PREACCIONADORES	44
CONCLUSIONES	46
BIBLIOGRAFIA	47

III EL SOFTWARE DEL PLC	
INTRODUCCION	48
3.1 LENGUAJES EN CONTROL DE PROCESOS	48
3.2 OPERACION INTERNA Y PROCESAMIENTO DE SEÑAL DEL PLC	50
3.2.1 PROCESAMIENTO DE ENTRADA/SALIDA	51
3.3 INSTRUCCIONES LOGICAS	53
3.4 DISEÑO UTILIZANDO EL SOFTWARE	54
3.5. FUNCIONAMIENTO DE LOS SISTEMAS	56
3.5.1. REPRESENTACION GRAFICA	57
3.5.1.1 ESQUEMA DE CONTACTOS O DIAGRAMA EN ESCALA	58
3.5.1.2 LOGIGRAMA.....	59
3.5.1.3 FLUJOGRAMAS.....	60
3.5.1.3 EL GRAFCET.....	61
3.6 ESTRUCTURA DE LOS PROGRAMAS EN EL PLC	63
CONCLUSIONES	66
BIBLIOGRAFIA	67
IV ESPECIFICACIONES TECNICAS Y APLICACIONES DEL PLC	
INTRODUCCION	68
4.1 CAMPO DE APLICACION	68
4.2 DESCRIPCION	69
4.3 PROGRAMACION DE LA MEMORIA EPROM	72
4.3.1 UNIDAD DE PROGRAMACION DEL PLC	73
4.3.2 LENGUAJE DE PROGRAMACION UTILIZADO	73
4.3.3 PROCEDIMIENTO DE PROGRAMACION	76
4.4 EJEMPLO DE APLICACION DEL PLC	77
CONCLUSIONES	84
BIBLIOGRAFIA	85
CONCLUSIONES GENERALES	86
RECOMENDACIONES GENERALES	87
ANEXO A	
diagrama de los circuitos diseñados	88
ANEXO b	
Elaboración de programas y quemado de memoria	91

INDICE DE FIGURAS

DESCRIPCION	PAGINA
1.1 Elementos de un sistema de control	2
1.2. Ejemplo de la máquina cableada	5
1.3. Representación de un proceso continuo	8
1.4 Un proceso conjunto en una planta química	9
1.5 Ejemplo de producción de partes discretas	10
1.6. Disposición de un PLC	13
2.1. Arquitectura interna de un PLC	21
2.2 Circuito aislado ópticamente	22
2.3. Diagrama de bloques de la arquitectura Interna del procesador central Z-80	25
2.4. Configuración de registros del Z-80	26
2.5. Circuito generador de reloj	29
2.6 Circuito de paso unico	30
2.7 Circuito de reset	31
2.8. Z-80 con buses de direcciones y datos provistos de buffers	32
2.9. Decodificación de puertos de E / S	33
2.10 Decodificación de memoria de lectura / escritura	34
2.11 Decodificación de E/S expandida	34
2.13 Diagrama de bloques de decodificación de 64 puntos	35
2.14 Decodificación de memoria	35
2.15. Pin out del IC 74LS373	36
2.16. Memoria RAM y EPROM	37
2.17. Captadores "todo o nada"	38
2.18. diagrama de módulo de entrada TON	40
2.19. Diagrama del módulo de salida TON	42
2.20. Captador de posición	44
2.21 Ejemplo de interrelacion pre-accionador y accionador	44
3.1. Sistema MDS	50
3.2. Tratamiento de I/O por actualizado continuo	51
3.3. Copiado masivo de entradas/salidas	52
3.4. Simbolos para gráficos de escalera	53
3.5. Diseño de un procedimiento para PLC	56
3.6. representación del diagrama de escala	58
3.7 simbolo de diagrama de escalera de un circuito de un motor eléctrico	58
3.8. Diagrama de escala	59
3.9.a) representación de funciones lógicas	59
b) logigráma de la relación combinatoria S = a.b.(c+d)	60

3.10. Representación de control lógico por medio	
de flujograma	61
3.11. Ejemplo de representación GRAFCET	62
4.1. Ejemplo de aplicación	77
4.2. Gráfico del ejemplo	79
4.3. Flujograma de la tarea de control	80
4.4. Sistema de contactos utilizados	81
4.5. Estado del programa en ensamblador Z80	83

LISTA DE TABLAS

DESCRIPCION	PAGINA
1.1 Tipos de transductor de entrada	1
1.2 Tipos de dispositivos de salida	2
1.3 Sistemas de control y la forma de control provisto	7
1.4 Desarrollo cronológico de los PLC's	14
1.5 Comparación de sistemas de control	15
2.1. Clasificación de los PLC	23
2.2 Mapa de memoria	37
3.1 Secciones de un programa de PLC	64
4.1 (a), Abreviaturas utilizadas en la descripción de las instrucciones del Z80	74
(b) Listado de las instrucciones mas empleadas para el autómata programable	75

CAPITULO I

INTRODUCCION A LOS SISTEMAS DE CONTROL

INTRODUCCION

Aunque en el área industrial las posibilidades de la aplicación de los microprocesadores son inmensas, y de hecho, ya se siente la invasión de estos dispositivos; nuestro país tiene un notable grado de atraso en este campo, las razones son principalmente económicas ya que los precios de algunos equipos o sistemas que los utilizan es en verdad bastante elevado; debiéndose tomar en cuenta la escasa preparación técnica con que se cuenta en las industrias, donde solo pocas personas saben utilizar correctamente los equipos, y debido a la introducción de estos sistemas es urgente también la necesidad de técnicos que desde el nivel más superior hasta el profesional, reúnan conocimientos teóricos y prácticos para su buen uso.

Debido a que estos componentes y equipos son importados en su totalidad, se genera una notable dependencia y hasta cierto punto un comodismo por parte de los ingenieros en el área lo cual no hace sino que se continúe en ese círculo vicioso de la dependencia; por lo cual es necesario que se generen mecanismos el desarrollo y aplicación de sistemas que utilizan microprocesadores los cuales sean capaces de solucionar tareas más o menos sencillas con lo cual se ahorre tiempo y dinero que de otra forma se invertiría en costosos equipos que en alguna medida serán sub-utilizados.

En vista de esto se pretende realizar un Controlador Lógico Programable (PLC) utilizando dispositivos que se puedan encontrar en el mercado y darle características de fiabilidad, flexibilidad y menor costo el cual consistiría en una estructura principal para todas las aplicaciones a las que se le destine y para su posterior utilización como una alternativa a sistemas extranjeros partiendo de que este sistema será mucho más sencillo.

1.1 ANTECEDENTES SOBRE AUTOMATIZACION Y CONTROL

Actualmente en casi todos los tipos de industria, muestran la tendencia de incrementar la productividad a través de un incremento de la automatización de procesos y de nuevas máquinas, lo cual conduce a una mejoría en la calidad de los productos. De cualquier manera, la automatización siempre cuenta con injerencia humana la cual

es la que dispone de los criterios básicos para su desempeño en una planta. Ya que en lugar de requerir obreros para tareas peligrosas o repetitivas; se destinan, ya capacitados, para operar la máquina que controla el proceso, y darle su respectivo mantenimiento.

Para la ejecución de el proceso de automatización, el operador debe de ser reemplazado de alguna manera de el sistema; dejando el sistema con una poca o nula intervención humana.

Cualquier sistema de control puede ser dividido en sus tres secciones constituyentes: entradas, procesamiento y salidas. La cual es modelado en la figura 1.1, pudiendo ser descrito en terminos de acciones, que consisten en mediciones de entrada, procesamiento de señales de control, produciendo acciones las cuales son enviadas a la salida del sistema. Las tareas de la sección de procesamiento son para producir respuestas pre-determinadas (en forma de salidas) como resultado de la información provista por medio de las señales medidas. Hay muchos métodos diferentes disponibles para la implementación de procesos funcionales, pero todos ellos parten de la misma filosofía.

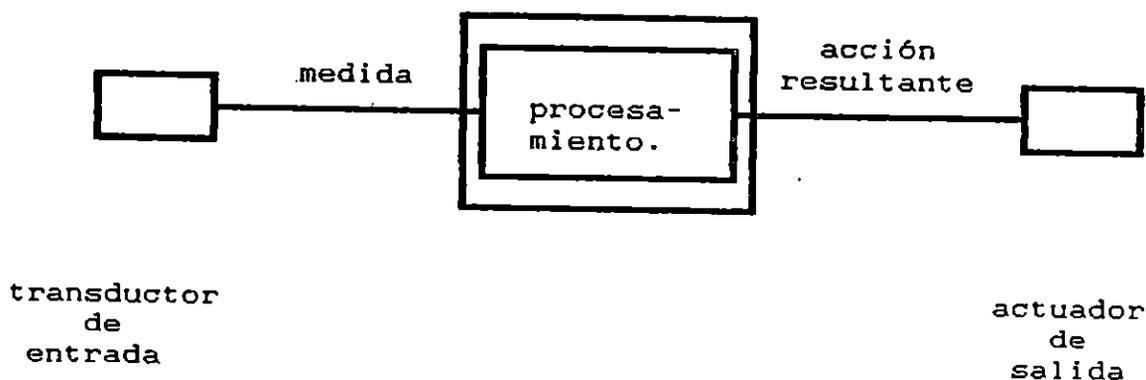


Figura 1.1 Elementos de un sistema de control.

Este modelo representa tambien el control de un operador humano actuando como la "sección de procesamiento". En esta sección el operador sabe que salida del proceso es la necesaria, para así monitorear las variables relevantes de la entrada (usando captadores como interruptores de posición, termistores, fotoceldas, etc.). En respuesta a estas lecturas, el operador alterará el estado del sistema (por medio de válvulas, motores, calentadores, etc.) para obtener a la salida el proceso de control deseado.

1.1.1 ENTRADAS

Las señales de entrada son normalmente provistas por varios transductores que convierten cantidades físicas en señales eléctricas. Estos transductores pueden ser desde simples push-buttons, switches o termostatos hasta convertidores o circuitos mas complejos, los cuales transmiten información de la cantidad que está siendo medidas. Dependiendo de el tipo de transductor utilizado, esta información de entrada puede ser representada como discontinua (binaria) o continua (análoga). En la tabla 1.1 se hace una pequeña clasificación de algunas transductores de entrada que son de los mas comunmente utilizados en procesos de control.

TABLA 1.1 Tipos de transductor de entrada

transductor	cantidad medida	cantidad a la salida
interruptor	movimiento/posición	voltaje binario
interruptor de limite	movimiento/posición	voltaje binario
termostato	temperatura	voltaje binario
termistor	temperatura	voltaje variante
fotocelda	luz	voltaje variante
celda de proximidad	presencia de objetos	voltaje variante

1.1.2 SALIDAS

Los sistemas de control deben de ser capaces de alterar elementos o cantidades dentro de un proceso. En un proceso de control, la forma de ejecución del mismo es desarrollado por medio de los dispositivos de salida; tales como bombas, relés, sistemas neumáticos, etc. los cuales alteran la magnitud u otra característica de las señales del sistema de control de manera que pueda ser utilizada. En otras palabras, los dispositivos de salida son tambien

transductores pero en otra dirección. Como los transductores de entradas, los dispositivos de salida pueden ser simples unidades binarias o de variable continua según el tipo de respuesta que se desee obtener, en la tabla 1.2 se presenta algunos tipos de dispositivos de salida, para tener una idea más o menos clara de estos.

También es de tener en cuenta que de acuerdo a las exigencias del sistema a controlar, es como se aplica al sistema un dispositivo de salida que ejecute algún cambio en las tareas a desempeñar. Tal como la variación de velocidad, temperatura o cualquier otra que sea de interés para el proceso en particular.

TABLA 1.2 Tipos de dispositivos de salida

Dispositivo de salida	Cantidad producida	entrada
motor	movimiento rotacional	eléctrica
bomba	movimiento rotacional mas desplazamiento de productos	eléctrica
pistones	movimiento lineal/presion	hidráulico y neumático
solenoides	movimiento lineal/presion	eléctrico
calentador	calor	eléctrico
válvula	variación de un orificio	eléctrica, hidráulica o neumática
relé	interruptor eléctrico/ limitador de movimiento	eléctrico

1.1.3 SECCION DE PROCESAMIENTO.

Esta corresponde a los "conocimientos del operador" sobre la operación que es requerida para mantener un proceso "en control". El operador utiliza estos conocimientos en conjunción con la información obtenida de las lecturas tomadas por las entradas, produciendo como resultado una acción a la salida.

A partir de la información obtenida por las entradas el sistema de control automático tiene que producir las señales de salida necesarias en respuesta a el control de planta construido en la sección de procesamiento. Esta planta de control puede ser implementado de dos diferentes maneras, usando "control de máquina cableada" o bien "control programable".

1.1.3.1 MAQUINA CABLEADA

La máquina cableada fué el primer tipo de control, fué uno de los primeros productos de aplicación industrial de la naciente rama de la electrónica, funcionando en grandes industrias hasta mediados del siglo XX; y cuyo principio de funcionamiento se basaba en la utilización de un circuito específico para cada aplicación y con lo cual se había resuelto la mayor parte de los problemas existentes. Con una tecnología cableada el automatismo se realiza por módulos conectados entre si; el funcionamiento obtenido resulta de la elección de estos módulos y del cableado que los conecta. Estas máquinas, cuyo diseño era exclusivo en cada caso, requerían importantes cambios en sus circuitos físicos cuando se intentaba adaptarla a otras circunstancias diferentes a las previstas, o bien cuando se pretendía alterar alguna de sus características fundamentales.

Tres tecnologías permiten realizar los automatismos cableados y estos son: Relés electromagnéticos, Módulos lógico-neumáticos y tarjetas o módulos electrónicos. En la figura 1.2 se ofrece un concepto gráfico de la máquina cableada.

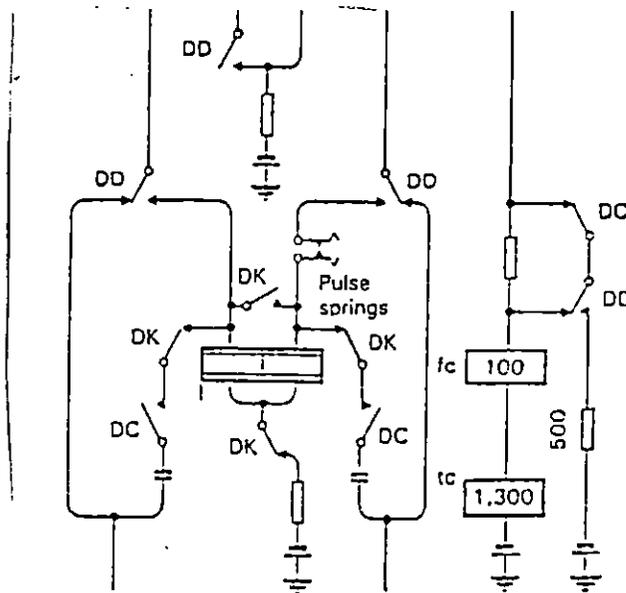


Figura 1.2. Ejemplo de la máquina cableada

Como se puede observar estos sistemas poseen funciones de control permanentemente fijadas lo cual se puede tomar como uno de sus mas graves defectos ademas de su escasa fiabilidad, consecuencia del elevado número de componentes que las constituyen, y su alto costo debido a conceptos de materiales y mano de obra. En resumen, el diseño de este tipo de controladores lleva consigo un alto costo y un grave problema en el caso que se tenga que modificar una estructura principal para cada aplicación.

1.1.3.2 CONTROL PROGRAMABLE

Un sistema programable es mucho mas versátil, ya que la función de control es programada y guardada adentro de un circuito integrado de memoria, y que puede ser alterada y reprogramado si es necesario. Los motivos fundamentales que propiciaron la aparición y desarrollo de las máquinas programadas, referidas en nuestro caso a las implementadas con microprocesadores, fueron tres:

- i El incremento de la complejidad en las funciones y en el número de elementos que conforman las máquinas, las cuales tienen que resolver trabajos mas importantes cada vez. En este aspecto se pontenció la posibilidad de facultar a los dispositivos no solo de elementos de regulación y control de las acciones físicas, sino tambien de otros destinados al cálculo y a la toma de decisiones.
- ii Los avances tecnológicos en la fabricación de circuitos integrados de alta escala de integración (VLSI), que permitieron poner en una sola pastilla de numerosos componentes activos y pasivos, formaron circuitos muy complejos y potentes.
- iii La casualidad, que tanto ha acompañado a los descubrimientos en electrónica y en general en todas las ciencias.

Solamente las tecnologías electrónicas de alta integración permiten la concentración de los componentes y la obtención de tiempos de respuestas necesarios para la realización de los constituyentes programables. Estos constituyentes pueden tomar formas diferentes: Tarjetas electrónicas estandares y específicas, micro y mini-ordenadores y autómatas programables. Las cuales poseen la poderas ayuda de un soporte lógico el cual es el lenguaje de programación que es mas o menos accesibles, segun el tipo de los constituyentes seleccionados.

1.1.4 SISTEMAS DIGITALES Y ANALOGOS

El gran desarrollo de la electrónica la ha hecho apoderarse de diversos sectores en los cuales ha venido a mejorar condiciones pre-existentes, La técnica de mediciones, control y regulación es el campo de aplicaciones mas alto que ha encontrado los microprocesadores. En la industria estos controlan líneas de producción, no solo para grandes fábricas sino tambien en forma rentable en las pequeñas.

TABLA 1.3 Sistemas de control y la forma de control provisto

Máquina cableada	tipo	Sistema programable	tipo
relés	digital	computadoras	digital análogo
lógica elec- trónica	digital	microcomputadora	digital análogo
lógica neumática	digital	PLC	digital análogo

En la tabla 1.3 se listan uno ejemplos de sistemas donde comunmente se utilizan tecnologías programables o cableadas, y junto con el tipo de control que ellas pueden desarrollar - digital (interruptor) o análogo (continuo) dependiendo de el tipo de señal medida o de el tipo de acción que se pretende ejecutar en la planta industrial

El control continuo o análogo utilizado directamente señales representativas de sensores o manejar dispositivos tales como bombas o calentadores, en donde este tipo de señales depende de un tratamiento tal como una señal amplificada o una forma de función matemática, tal como una suma o integral. En cambio el control binario o discontinuo es común en otro tipo de máquinas o proceso que consisten en controlar un gran numero de operaciones simples donde siempre se tienen una de las dos condiciones,

1.2 TIPOS DE PROCESO INDUSTRIAL

Existe una gran cantidad de diferentes procesos de manufacturación; pero todos esos procesos pueden ser agrupados dentro de tres categorías principales en relación a el tipo de operaciones que tienen lugar en el proceso.:

- i - producción continua
- ii - producción en conjunto
- iii - producción discreta

Cada uno de ellos presenta características individuales y requerimientos que deben de ser considerados cuando se diseña un control industrial.

1.2.1 PROCESO CONTINUO

Un proceso continuo (como el representado en la figura 1.3), se necesitan de materiales en bruto en la entrada, o sea el inicio y corre continuamente, produciendo un material o producto terminado a la salida, este proceso puede trabajar por grandes periodos, que pueden comprender desde horas hasta semanas completas en algunos casos.



Figura 1.3. Representación de un proceso continuo

Para ilustrar aún mas esta definición se puede tomar el caso de la producción de hojas de acero, en donde grandes bloques de acero al rojo vivo son transportados hacia una serie de rodillos de diferente diámetro, que van reduciendo el grosor del acero para finalmente producir una hoja de acero al final de la línea de rodillos. El tiempo de duración de todo este proceso depende de la longitud del bloque de acero que se le suministra; El resultado final debe de cumplir con estrictas disposiciones lo que hace que el tipo de control deba de ser rápido en su respuesta ante el aviso de cambios de estado o situaciones no deseadas, por parte de los sensores de entrada.

1.2.2 PROCESO CONJUNTO

Un proceso conjunto utiliza una cierta cantidad de un material fuente (entrada) y ejecuta un proceso de operaciones en este material. produciendo una cantidad específica de producto terminado (salida) o bien, este producto será luego descargado en contenedores

posteriormente que ejecutan diferentes sub-procesos hasta obtener el producto final deseado. Un ejemplo típico de este tipo de proceso son los procesos químicos en donde se descarga cierta cantidad de un producto A del tanque 1 y de un producto B contenido en el tanque 2, dentro de un depósito de mezcla y calentamiento donde se mantiene por un periodo de tiempo mezclándose; y luego dicha mezcla puede pasar a otros estados que continúan el proceso o bien detenerse ahí con la mezcla deseada. Ver figura 1.4

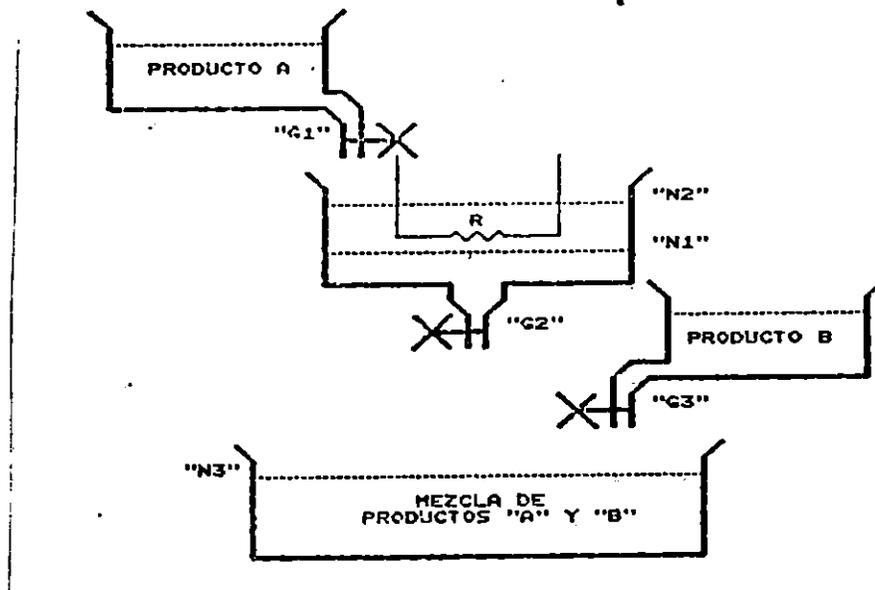


Figura 1.4 Un proceso conjunto en una planta química

1.2.3 PRODUCCION DE MODULAR

En este tipo de procesos un producto individual sufre varias operaciones antes de llegar a su acabado final. Alternativamente, muchos componentes pueden ser combinados o ensamblados dentro del proceso para salir como un producto individual o conformado por partes.

Un ejemplo de producción de partes discretas o modulares es ilustrado en la figura 1.5. Aquí cierta cantidad de metal es cargado en una mesa de taladro por medio de un brazo robot, que lo coloca en una posición exacta y es taladrado convenientemente siguiendo un sólo patrón o variando el diámetro de los agujeros utilizando diferentes brocas de su "cabeza selectora" rotatoria. La pieza terminada es luego liberada del banco y colocada en una banda transportadora adyacente por medio del mismo brazo robot.

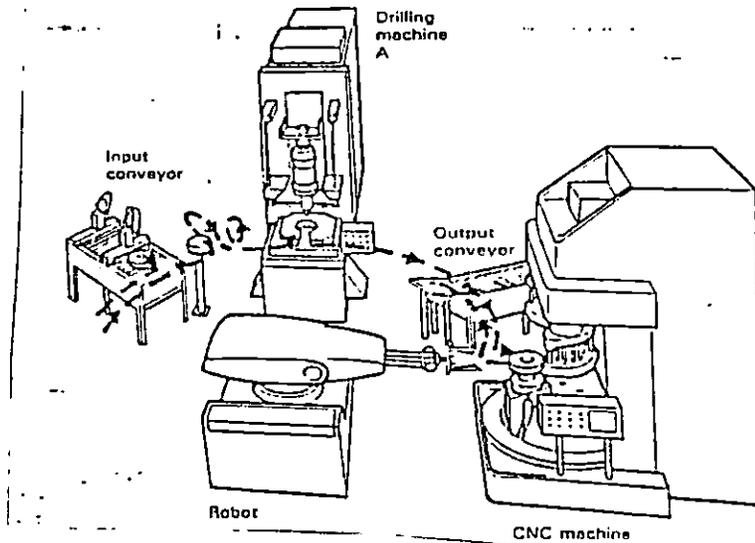


Figura 1.5 Ejemplo de producción de partes discretas.

En este ejemplo se tienen varias secuencias de operación, donde la mayoría de actividades es controlada por medio de estados binarios, incluyendo la intercomunicación entre máquinas cuando ha finalizado una etapa en una de ellas y continúa la siguiente máquina. Esto comúnmente es llamado "encadenamiento".

1.3 CONTROL DE PROCESOS POR COMPUTADORA

Las computadoras han sido utilizadas en procesos de control desde mediados de los años 50's, cuando se realizaban pruebas del desarrollo de la electrónica en la post-guerra. Las computadoras realizaban operaciones tales como transporte de datos y alarmas para ayudar en varias estrategias de control de sistemas con funciones complejas que pueden (en teoría) ser rápida y fácilmente alterados.

Muchas de esas aplicaciones fueron en la industria química, resultando exitosos el control computarizado de procesos tales como la producción de amonio, plásticos y resinas. En esa época el hardware de las computadoras era excesivamente caro y lento (debido esto básicamente por el tipo de memoria utilizado), y físicamente grande (en comparación con los sistemas recientes) puesto que toda la circuitería estaba basada en tubos al vacío.

Desafortunadamente desde la década de los 60's el desarrollo de aplicaciones de control por computadora fué poco exitoso, por varias razones:

- 1- el personal encargado de programar las máquinas raramente tenía algún conocimiento del proceso a ser controlado, y muchos ingenieros tenían poca o nula experiencia en computadoras.
- 2- Los fabricantes de computadoras trataron de hacer la aplicación industrial "idónea" al hardware disponible (usualmente basado en una computadora principal localizada en un cuarto de control remoto y requiriendo de grandes cables para la conexión de los dispositivos de entrada y salida).
- 3- La instalación y mantenimiento de esos sistemas fué usualmente difícil y consumía demasiado tiempo.

Estos y otros factores volvieron inadecuado (aún incorrecto) la utilización de sistemas computarizados, debido a las deficiencias en la implementación física y la ejecución del sistema de control. Estas fallas de dicho sistemas los volvieron comercialmente inaceptables, por lo que la gran mayoría de compañías fabricantes de computadoras desviaron su atención hacia áreas mas rentables.

Como resultado de esto, no fué sino hasta mediados de la década de los 70's, con el posterior desarrollo de la electrónica, que se desarrollaron computadoras pequeñas, rápidas y bastante mas baratas, volviendolas mas atractivas a la industria; pero siempre se requiere de un personal especializado en programación para su uso y aplicación, lo cual sigue siendo una gran desventaja, ya que es necesario la desviación de recursos para el mantenimiento de dicho equipo; lo cual afecta el rendimiento de la empresa.

1.4 CONTROLADORES LOGICOS PROGRAMABLES (PLC)

La NEMA define el controlador programable como un aparato que utiliza la electrónica digital con una memoria programable para el almacenamiento de instrucciones que implementen funciones específicas tales como secuenciales, lógicas, de tiempo, de conteo, para el control de máquinas y procesos. La razón básica del uso de los controladores programables en el control automático de procesos en la industria, radica en su versatilidad y programabilidad. Esta versatilidad se traduce en incrementos de productividad, mejoras en la calidad del proyecto y en una reducción de costos y por lo tanto en mayor competitividad del producto dentro del mercado.

Entre las muchas ventajas que el controlador programable ofrece al compararlo con los controles electromecánicos, se pueden destacar:

i. Disminuyen el costo de materiales, mano de obra y mantenimiento.

ii. Permiten realizar fácil y rápidamente cambios y adaptaciones sin afectar por lo general al estructura del sistema, sino únicamente el programa de instrucciones.

iii. Reducen el hardware y la estructura física ya que utilizan circuitos poderosos basados en tecnología LSI y VLSI.

iv. consumen menor cantidad de energía.

v. Incrementan la fiabilidad del sistema al disminuir considerablemente el número de elementos utilizados.

vi. Si el mismo control y/o secuencia de operación se necesita en gran número de lugares, bastará con cargar el mismo programa de instrucciones en cada uno de los controladores.

Dichos sistemas operan por medio del examen de señales de entrada de un proceso y enviando a la salida instrucciones lógicas (las cuales han sido programadas dentro de su memoria), y que controlarán procesos, equipo ó maquinaria por medio de interfaces que permiten la conexión directa entre el PLC y los actuadores del proceso sin la necesidad de circuitos intermedios. Los controladores programables fueron inicialmente concebidos por un grupo de ingenieros de la General Motors en 1968, proveyéndolo de las siguientes características:

- I) Fácil programación y reprogramación, realizados en la planta, para la alteración de su secuencia de programación.
- II) Fácil mantenimiento y reparación.
- III) Más pequeño que se equivalente de relés.
- IV) Bajo costo

En la figura 1.6 podemos observar gráficamente esta clasificación por secciones de un controlador programable

TABLA 1.4 Desarrollo cronológico de los PLC's

AÑO	DESARROLLO
1968	Se desarrolla el concepto de los PLC.
1969	Hardware del CPU, con instrucciones lógicas, 1K de memoria y 128 puertos de E/S.
1974	Uso de multiprocesadores en un PLC, timers y contadores, operaciones aritméticas, 12 K de memoria y 1024 puntos de E/S.
1976	Sistemas de E/S remotos.
1977	se introduce el uso de PLC's basados en microprocesadores (basado en el 8080)
1980	Desarrollo de módulos inteligentes de E/S desarrollo de facilidades en la comunicación Mayor capacidad del software y utilización de micro-computadoras como auxiliares para la programación.
1983	Se introduce el PLC de bajo costo.
1985	Desarrollo de redes de todo nivel para PLC desarrollo del lenguaje Grafset, y distribución jerárquica del control industrial.

Cabe resaltar el hecho de que desde 1980 el mercado de los pequeños controladores (con un promedio de 20 puntos para E/S digitales, 500 pasos de programación y modulares) ha crecido debido al número de compañías japonesas que han introducido unidades muy pequeñas y de bajo costo, lo que los ha vuelto muy competitivos en el mercado de los autómatas programables. Al comparar los autómatas con los otros sistemas de control, tal y como se ve en la tabla 1.5, estos emergen como la mejor elección, a menos que los requerimientos propios hagan necesario otro tipo de sistemas. Los controladores programables poseen un hardware y un software diseñado de tal manera que los vuelve atractivos como controladores para un ancho rango de equipos industriales.

TABLA 1.5 Comparación de sistemas de control

Característica	Sistema de Relé	Lógica Digital	Computadores	Sistemas PLC
Precio por funciones	tolerable	bajo	alto	bajo
Tamaño físico	grande	muy compacto	tolerable compacto	muy compacto
Velocidad de operación	baja	muy rápida	bastante rápida	rápida
Inmunidad al ruido	excelente	buena	regular	buena
Capacidad de operaciones complejas	no	si	si	si
Fácilidad para el cambio de función	muy difícil	difícil	muy simple	bastante simple
Fácilidad de mantenimiento	difícil por sus contactos	difícil por el No de IC's	difícil	buena
Instalación	consume tiempo en diseñarlo e instl..	consume tiempo en diseñarlo	consume tiempo en programarlo	simple en programarlo e instl.

1.5 CRITERIOS DE SELECCION DE PLC SOBRE OTRAS TECNOLOGIAS

En los automatismos programables, la estructura principal común debe poder realizar un gran conjunto de operaciones básicas, cada una de las cuales es ejecutada cuando la máquina recibe el código de la instrucción correspondiente. En cada aplicación, las instrucciones deben ordenarse adecuadamente para formar un programa específico para el caso. Por lo anterior podemos concluir que la arquitectura de un controlador programable es básicamente la misma de las computadoras de propósitos generales, pero tienen algunas importantes características que los distingue:

Primero, el controlador programable esta diseñado para sobrevivir en las no siempre estables condiciones del medio ambiente de las fábricas. Un PLC bien diseñado puede ser colocado en áreas con una sustancial cantidad de ruido eléctrico, interferencia electromagnética, vibración mecánica o temperatura un poco arriba de lo normal.

Una segunda distinción de los PLC es que el hardware y software son diseñados para un fácil uso por técnicos y electricistas de planta. Las interfaces para conectar dispositivos externos son fácilmente conectados y su construcción modular hace que su reparación sea mas sencilla.

Los PLC son máquinas destinadas a la ejecución de programas sencillos en forma ordenada y secuencial

Los PLC poseen control flexible; la característica de programable provee al controlador de grandes beneficios, especialmente en cuanto a su instalación, siendo este aspecto tambien muy importante ya que son de facil instalación la cual es sencilla y hasta cierto punto rutinaria, lo cual conlleva hacia otra característica importante y es su mantenimiento, ya que desde su comienzo estos se han diseñado teniendo en mente la facilidad del mantenimiento. Con todos los componentes virtualmente de estado sólido, el mantenimiento es reducido a cambios de algún módulo o de algunos componentes alojados en base. Con la ayuda de dispositivos programables, cualquier lógica esperada puede ser observada para comprobar si las entradas o salidas están activadas o no.

CONCLUSIONES

- Es necesario que la Escuela de Ingeniería Eléctrica se actualice con material bibliográfico, ya que este es muy importante para desarrollar una buena investigación de campo sobre los nuevos temas a desarrollar en proyecto de ingeniería y trabajo de graduación; pues hasta ahora existe un marcado desfase entre los temas propuestos y la información que la Escuela pueda brindar.
- La gran versatilidad de los controladores lógicos programables para aplicaciones de control y de procesos industriales le permite el desarrollar aún tareas complejas con una gran precisión lo que da como resultado que esta filosofía de control se imponga en el sector industrial por sobre sus antecesores, la clave de esto es el uso y soporte que le dan los microprocesadores.
- El PLC a desarrollar será un prototipo básico, con lo cual se podrán realizar tareas sencillas de mando y regulación, ofreciendo prestaciones específicas destinadas a la mediana industria, orientado principalmente a sustituir tecnologías cableadas dominadas por relés y con la cual obtendrá la función programada que realmente desee.
- Los controladores lógicos programables ofrecen una gran ventaja sobre las antiguas máquinas cableadas, debido a que este emplea una estructura principal para todas las aplicaciones y los cambios realizados se harán principalmente en programas y no físicos.
- Debido a su estructura de estos PLC's, se tiene que lo más complejo es su soporte lógico, ya que toda la versatilidad y funcionalidad que se tenga depende en gran medida de lo bien diseñado que se tenga dicho software.
- La utilización de los controladores en los procesos industriales, conlleva al incrementos en la productividad, mejoras en la calidad de un producto y en una reducción de costos lo cual evita también que se tenga personal empleado en tareas agobiantes y repetitivas.

BIBLIOGRAFIA

Angulo Usategui, José Maria.

"MICROPROCESADORES, CURSO SOBRE APLICACIONES A SISTEMAS INDUSTRIALES".

Segunda edición, 1980

Paraninfo, Madrid.

Keil, Heinrich.

"MICROCOMPUTADORAS"

Siemens aktiengesellschaft

Marcombo S.A.

Dora Deysi Melendez Castro

Nelson Ovidio Castaneda Vidal

"APLICACIONES DE LOS MICROPROCESADORES A SISTEMAS DE CONTROL"

Tesis optar al grado de ingeniero electricista

Biblioteca de la Escuela de Ingenieria Eléctrica.

UES, 1987.

Angulo Usategui, José Maria

"MICROPROCESADORES"

Cuarta edición, 1985

Paraninfo, Madrid.

Gustavo Sebastian Francés

Francisco Roberto Castaneda Vidal

"APLICACIONES DE LOS MICROPROCESADORES A SISTEMAS DE CONTROL"

Tesis para optar al grado de ingeniero eléctrico

Biblioteca de la Escuela de ingenieria Eléctrica

UCA, 1980.

Simatic S5

"AUTOMATIZACION CON EL SIMATIC S5-101U"

Geratehadbuch

Siemens aktiengesellschaft.

CAPITULO II

DISEÑO DEL HARDWARE DEL PLC

INTRODUCCION

En esta etapa corresponde al diseño del hardware del sistema, haciendo énfasis en la CPU del controlador lógico programable, la cual se basa en un sistema mínimo del Z-80, el cual ha sido elegido por su versatilidad y sencillez, a la vez por la gran cantidad de información bibliográfica con que se cuenta referente a él y a sus circuitería de soporte.

El presente capítulo se hace un breve enfoque del sobre la filosofía, desarrollo y aplicaciones de los autómatas ya que en nuestro medio es prácticamente nula la información de la que se dispone, limitándose la existente a catálogos y manuales. Posteriormente se especifican muchos de los criterios de diseño que fueron tomados en todas las etapas del CPU, llegándose al diseño de un prototipo de la CPU, y de los módulos de entrada salida que conectan al sistema con el mundo real, además teniendo la opción de programación con las PC AT LEMMON de la EIE, y utilizando el software disponible para la programación de la EPROM y para vaciar los programas en ellas.

Al integrarse la CPU con los módulos E/S, se está orientando el sistema hacia tareas de automatización de pequeña escala; puesto que el diseño del autómata va orientado hacia la filosofía de un "mini" PLC, y tal como estos, su uso va dirigido al reemplazo de arreglos alambrados de relés, para el control individual de maquinaria. Por lo tanto los módulos E/S son del tipo "Todo o Nada", siendo las cantidades de salida valores de voltaje binario (ON/OFF). Y las entradas son provenientes de transductores, que de ahora en adelante se han definir como captadores, los cuales pueden ser desde simples interruptores hasta termocuplas; en nuestro caso las cantidades medidas serán de movimiento y posición pues estas son las mas utilizadas en automatización, y para las salida se verá que estan van dirigidas hacia los preaccionadores que son los que conectan al PLC con el proceso

Lo que en esta sección se pretende es brindar toda la información posible sobre la filosofía del aspecto físico de los automatismos programables pequeños además de ofrecer información sobre los criterios de diseño que nos han llevado a escoger al Z-80 como base del CPU del autómata, y el porqué se han escogido modulos de E/S del tipo binario "todo o nada".

2.1 DISEÑO DEL HARDWARE

Los controladores lógicos programables son computadores construidas con un propósito específico y consistentes de tres áreas funcionales: procesamiento, memoria y entradas/salidas. Las condiciones a la entrada para el PLC son sensadas y luego guardadas en memoria, donde el autómata ejecuta las instrucciones lógicas programadas sobre dichos estados a al entrada. Las condiciones para la salida son luego generadas para manejar el equipo asociado. Las acciones tomadas dependen totalmente de el programa de control almacenado en la memoria

En los PLC's pequeños, siendo el diseñado uno de ellos, estas funciones son ejecutadas por circuitos impresos individuales empaquetados en una solo unidad compacta, presentando la ventaja de que cada tarjeta de circuito impreso es facilmente removida y reemplazada, lo que facilita una rápida reparación del sistema si se presentan fallas.

Tambien es necesario el considerar a la "unidad de programación" en donde se le almacenan programas de control a la memoria del PLC

2.1.1 UNIDAD CENTRAL DE PROCESOS

La Unidad Central de Procesos controla y supervisa todas las operaciones del autómata transportando las instrucciones del programa guardadas en la memoria, existe tambien una red interna de comunicaciones llamada bus de datos que carga la información desde y hacia el CPU, la memoria y las unidades de entrada y salida, todo bajo control de la misma central. La CPU es suplida por un reloj externo, el cual puede ser un cristal de cuarzo o un oscilador RC, que genera la frecuencia de trabajo, típicamente varía entre 1 y 8 Mhz dependiendo del microprocesador utilizado y del área de aplicación. En la figura 2.1 presentamos la arquitectura interna de los autómatas programables.

Practicamente todos los modernos controladores lógicos programables poseen una CPU que está basada en microprocesadores los cuales determinan su velocidad y su tamaño. Aunque el los autómatas grandes tambien se emplean microprocesadores adicionales para controles complejos y funciones que consumen mucho tiempo tales como procesamiento matemático y control PID

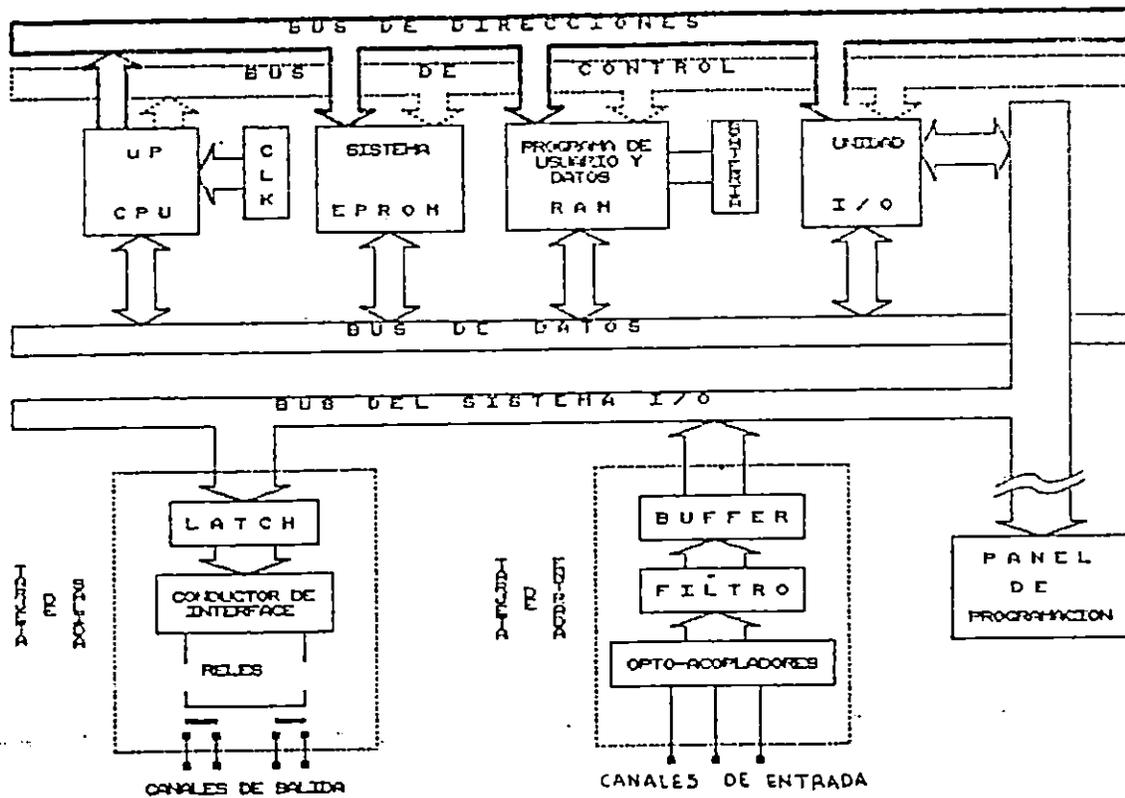


Figura 2.1. Arquitectura interna de un PLC

2.1.2 MEMORIA

Para almacenar los programas todos los controladores programables se utilizan dispositivos de memoria semiconductora tal como la RAM, EPROM o EEPROM.

En este caso se tendrá una EPROM para guardar los programas, de tal manera que los cambios sean hechos fácilmente por el programador. Y si se desea alterar su contenido entonces será fácilmente removible; ya que para ejecutar futuras alteraciones de el programa es necesario borrar dicha memoria y cargar el nuevo programa con todas las alteraciones; o bien copiar el programa ya corregido en otra dirección de memoria que este limpia.

También el controlador puede requerir memoria para otras funciones tales como guardar temporalmente el valor contenido en los puertos de E/S, o para guardar temporalmente el estatus de funciones internas.

Los pequeños controladores normalmente poseen un tamaño de memoria fijo, relacionado estrechamente con su tamaño y aplicaciones, esta capacidad varía entre 300 y 1000 instrucciones; lo cual es espacio suficiente para las necesidades de la mayoría de usuarios. En cambio en los grandes PLCs utilizan módulos de memoria de entre 1K hasta 64K con capacidad de expansión.

2.1.3 UNIDADES DE ENTRADA SALIDA

La mayoría de PLCs operan intermente entre valores de 5 a 15 Vcc (comunmente voltajes TTL y CMOS), con los cuales procesan señales que pueden ser mucho mas grandes, de entre 24 Vcc a 240 Vca y de bastantes amperios.

Los módulos de I/O forman la interface entre el PLC y el mundo real externo, y por lo tanto deben de proveer todas las señales necesariamente acondicionadas y aisladas. El PLC es conectado directamente con los actuadores y transductores sin la necesidad de circuiteria intermedia en algunos casos; pero en el caso de los pequeños autómatas se es necesario el emplear relés entre el y las bombas o válvulas.

Una consideración estandarizada es sobre el tipo de aislamiento eléctrico de los módulos y el proceso a controlar, el cual se hace por medio de circuitos opto-acoplados, (figura 2.2) en dichos módulos. Los cuales consisten en un led y un foto-transistor formando el par acoplado ópticamente que permite el paso solamente a pequeñas señales, pero protegiendo la CPU de picos de voltaje o trasientes que podrían llegar hasta los 1500 V.

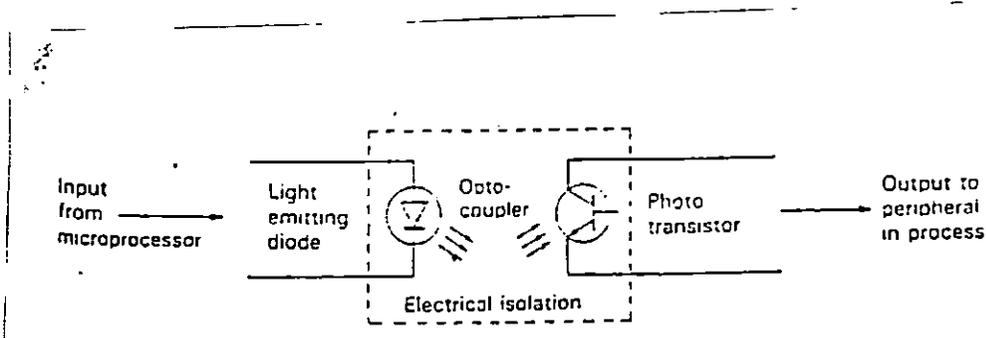


Fig. 2.6 Opto-isolator circuit.

Figura 2.2 Circuito aislado ópticamente

2.2 TIPOS DE SISTEMAS PLC

El incremento de la demanda por parte de la industria por los controladores programables ha hecho que estos se clasifiquen dependiendo de su forma y su tamaño para desarrollar tareas de control; lo que conlleva a que se fabriquen un amplio rango de PLCs con varios niveles de facilidad de ejecución y otras facilidades.

Típicamente estas definiciones son dadas en términos del tamaño de la memoria y de el número máximo de puntos de entrada/salida que es sistema puede soportar, la tabla 2.1 nos muestra estas categorías

TABLA 2.1. Clasificación de los PLC

TAMAÑO DEL PLC	NUMERO DE PUERTOS I/O	TAMAÑO DE LA MEMORIA
PEQUEÑO	<40/<40	1 K
MEDIANO	128/128	4 K
GRANDE	>128/>128	>4 K

Sin embargo, para evaluar propiamente cada controlador programable se debe de considerar otras características adicionales, tales como su procesador, facilidades de programación, capacidad de expansión, etc.

2.2.1 PEQUEÑOS PLCs

En general los pequeños y "minis" PLCs son diseñados como unidades robustas y compactas las cuales pueden ser montadas cerca o sobre es sistema a ser controlado. Son principalmente utilizados para reemplazo de controles por medio de máquinas cableadas para el manejo de timers o de lógica de relés; para el control individual de elementos de una planta o de maquinaria, pero los mas avanzados pueden utilizarse para coordinar varias máquinas trabajando en conjunto con otras.

En el caso concerniente, se ha diseñado un "mini" autómata programable siguiendo esa filosofía, cuya utilidad es la de reemplazar lógica de relés y para el control individual de maquinaria; con una moderada capacidad de expansión de su puntos de E/S, por lo que es posible el añadir otros módulos de E/S con otro tipo de características que los propios que el sistema trae incorporado; se han tomado en cuenta criterios tales como el tipo de usuarios que tendran, tipo de programación, versatilidad de aplicaciones para sencillas y comunes aplicaciones industriales. Dirigidas principalmente a usuarios finales que no necesariamente sean técnicos especializados, ya que se deben de adaptar a las realidades tecnológicas y de preparación de nuestro país.

El "mini" PLC esta basado simplemente en un microprocesador individual, y las facilidades de

programación ofrecidas parten de las que el micro ofrece en base a su lenguaje ensamblador, siendo programado de manera de lista de instrucciones lógicas (esto será enfocado en el próximo capítulo); y dichos programas serán guardados en la EPROM.

2.2.2 MEDIOS Y GRANDES PLCs

Los autómatas programables de tamaño medio son aplicados en tareas de control lógico que no pueden ser desarrolladas por los pequeños controladores por no contar con un suficiente número de E/S; debido a situaciones en las cuales se han expandido las tareas de control. Los sistemas medios ofrecen entre otras las siguientes características:

- Capacidad de expansión en base a sistemas modulares
- Facilidades de comunicación
- Tarjetas de E/S diseñadas para funciones extra
- CPU compuesto de varios microprocesadores
- Amplias opciones de programación.

La elección de un PLC grande parte de el mismo principio, la capacidad de gran expansión de puntos de entrada y salida; estos están diseñados para su uso en grandes plantas o en máquinas muy complejas, que requieren un control continuo, además de supervisión y monitoreo de otros PLCs de menor tamaño o de máquinas inteligentes. Ofreciendo entre otras las características más importantes presentadas a continuación:

- Microprocesadores de 16 bits como procesador principal, para aritmética digital y manejo de textos.
- Procesadores más pequeños que sirven como coprocesadores o procesadores paralelos para conteos rápidos, almacenamiento de información, comunicación. etc
- Procesadores periféricos para el manejo de tareas adicionales que son dependientes del tiempo o de tiempo crítico, tales como:

- Control de lazo cerrado (PID)
- Control de posición
- Cálculos con punto flotante
- Diagnóstico y monitoreo
- decentralización de I/O y su intercomunicación
- manejo de I/O remotos

2.3. ARQUITECTURA DE LA CPU

El microprocesador es el componente sobre el cual recaerá toda las decisiones de control, sean estas internas o bién de el proceso que se encuentre monitoreando, por lo tanto la elección del mismo es vital para la posterior configuración de todo el sistema; en nuestro caso se han tomado los siguientes criterios que condujesen a su elección:

- Documentación bibliográfica suficiente
- existencia en la EIE de este IC y demás circuitería de soporte.
- Conocimiento, por parte nuestra, sobre su funcionamiento
- Disponibilidad de software
- Circuitería de apoyo reducida

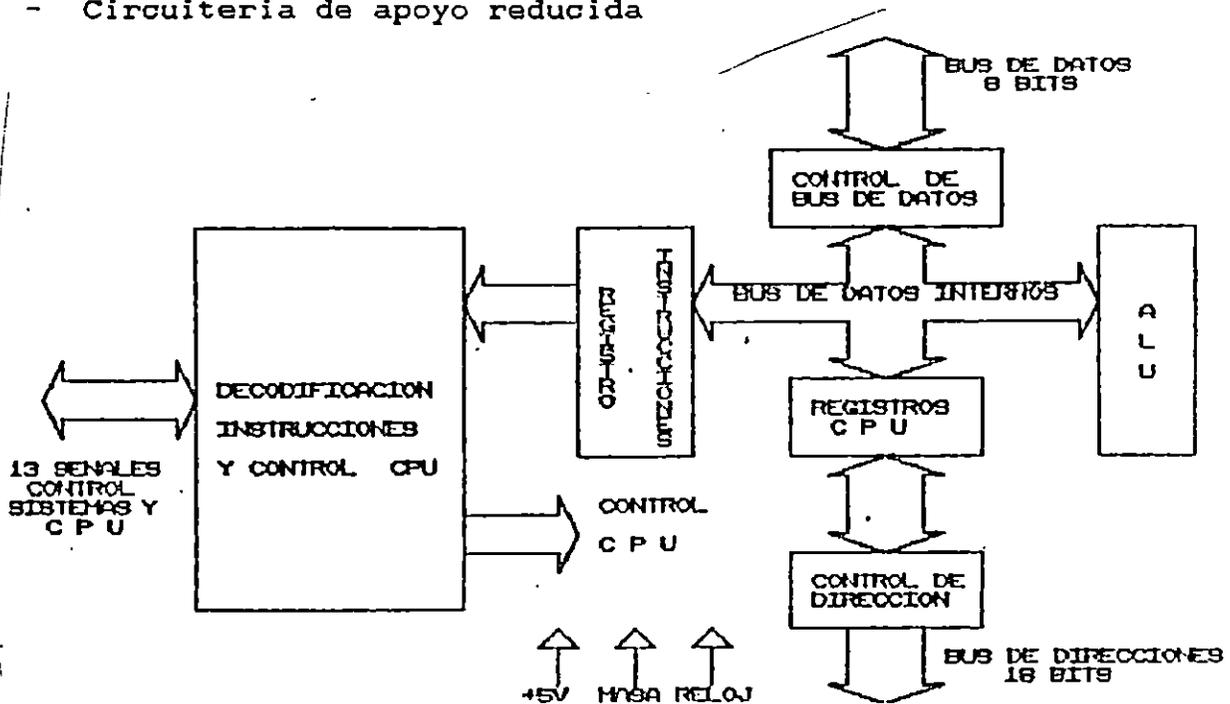


Figura 2.3. Diagrama de bloques de la arquitectura interna del procesador central Z-80

En base a estos criterios es que nos hemos decidido por utilizar al Z-80 de Zilog, pues los cumple a cabalidad. Los componentes internos del Z-80 en diagrama de bloques se pueden apreciar en la figura 2.5, puede observarse que es un microprocesador orientado hacia los registros. Consta de 18

registros de 8 bits y cuatro registros de 16 bits los cuales son fácilmente accesibles por el programador, pudiendo funcionar también como una memoria programable estática. Dichos registros se dividen en dos que son el principal y el alternativo, cada uno de los cuales contiene 6 registros de 8 bits de uso general que pueden utilizarse ya sea individualmente o como pares. También incluyen dos conjuntos de acumuladores y de registro de estado.

En la figura 2.4 se muestra lo dicho anteriormente de los acumuladores y registros de estado junto con registros de uso general y especial.

CONJUNTO DE REGISTROS
PRINCIPALES

ACUMULADOR	INDICADORES DE ESTADO
A	F
B	C
D	E
H	L

CONJUNTO DE REGISTROS
ALTERNATIVOS

ACUMULADOR	INDICADORES DE ESTADO
A'	F'
B'	C'
D'	E'
H'	L'

VECTOR DE INTERRUPCION I	REGENERACION DE MEMORIA R
REGISTRO INDICE	IX
REGISTRO INDICE	IY
PUNTERO DE PILA	SP
CONTADOR DE PROGRAMA	PC

FIGURA 2.4. Configuración de registros del Z-80

A continuación se da una descripción de la función y de la estructura de los componentes del procesador central Z-80:

I REGISTROS

A. Acumuladores y registro de estado

El acumulador recibe los resultados de todas las operaciones lógicas y aritméticas de 8 bits, y el registro de estado indica la ocurrencia de condiciones específicas tales como paridad, cero, signo, acarreo, etc.

B. Registros de uso general

Hay dos conjuntos de registros de uso general el principal y el alternativo, los cuales pueden servir tanto para operaciones de 8 ó de 16 bits, ya que pueden agruparse en pares.

C. Registros de uso especial

El contador de programa contiene una dirección de 16 bits en memoria a partir de la cual se busca la instrucción en curso. Después de ejecutar dicha instrucción, el contador se incrementa en el siguiente octeto de memoria a menos que se ejecuten saltos o una instrucción de llamada.

El puntero de pila es una zona reservada de memoria para almacenar datos temporalmente, y en cuya parte superior se indica por el contenido del puntero de pila, y las posiciones están almacenadas de forma LIFO (last in, first out o sea "último en entrar, primero en salir").

D. Registros índices IX y IY

Estos registros facilitan la manipulación de datos de tablas, son independientes de 16 bits, que retienen las direcciones de base utilizadas en modos de direccionamiento indexado.

E. El Vector de interrupción es un registro de 8 bits que puede cargarse con una dirección de una rutina de servicio de interrupción.

F. El registro de refresco de memoria se utilizará para validar memorias dinámicas.

II Unidad lógica y aritmética

Las manipulaciones aritméticas y operaciones lógicas se tratan como 8 bit simultáneos en esta sección, la cual también se comunica internamente con los registros del procesador central; no es directamente accesible por el programador.

III Registro de instrucciones y control del procesador central. El registro de instrucciones es el que retiene el contenido de memoria direccionada por el contador del programa y se carga durante un ciclo de búsqueda de cada instrucción. La unidad de control de procesador ejecuta las funciones definidas por la instrucción en el registro de instrucciones y genera todas las señales de control necesarias para transmitir los resultados a los registros adecuados.

IV Tipos de instrucciones del Z-80.
El Z-80 puede ejecutar instrucciones independientes que se pueden agrupar en:

- Carga e intercambio
- Aritméticas y lógicas
- Búsqueda y transferencia de bloques
- Rotación y desplazamiento
- Manipulación de Bits
- Salto, llamada y retorno
- Entrada y salida
- Control CPU

2.3.1 SISTEMA MINIMO BASADO EN EL Z-80

Un sistema mínimo es considerado al compuesto por un puerto de entrada, un puerto de salida, memoria ROM y RAM (aunque en algunos casos la RAM no es utilizada por lo que existen requerimientos en los que no es incluida).

Es evidente que un sistema alrededor del Z-80 resulta sencillo y un tanto económico, por lo que presenta ventajas en comparación con sistemas basados en otros microprocesadores de su general. Ya que aparte de requerir pocos componentes adicionales para configurar un sistema mínimo, el Z-80 permite que éste pueda ser ampliado de acuerdo a las necesidades existentes; siendo esto último una de las razones más importantes que nos inclinó por su elección, ya que nuestro diseño requiere esto, un sistema que pueda ser fácilmente reconfigurable desde el hardware y el software.

En esta parte se tratará el diseño de un sistema mínimo funcional, que constituirá el núcleo alrededor del cual se irán agregando las otras partes que complementarán el sistema final.

Un criterio que se ha tomado en cuenta, es el hecho de obtener resultados a partir de cosas sencillas, para así llegar a las más complejas. Así nuestro primer objetivo es la obtención de un circuito que funcionó, y una vez que trabaje correctamente puede comenzarse a realizar mejoras. Tal y como se dijo anteriormente el sistema mínimo basado en el Z-80 requiere de los siguientes componentes:

- CPU Z-80
- Memoria ROM
- Memoria RAM
- 1 puerto de entrada
- 1 puerto de salida
- fuente de alimentación
- Generador de señales de reloj
- Generador de señal de reset.

2.3.2 CIRCUITO DE RELOJ

La señal de reloj en un sistema basado en microprocesadores es una forma de onda que se utiliza para sincronizar la operación del sistema. Dado que el microprocesador Z-80 no posee oscilador interno para generarla (a diferencia de otros microprocesadores, como el 8080 que necesita de dos señales de reloj generada con componentes discretos), necesita de una señal de fase generada externamente que este alternando entre 0 y 5 voltios, cuya frecuencia de oscilación puede variar desde 500 KHz hasta 6 MHz dependiendo de la versión del microprocesador. Por lo que es de gran importancia el implementar un circuito que la genere.

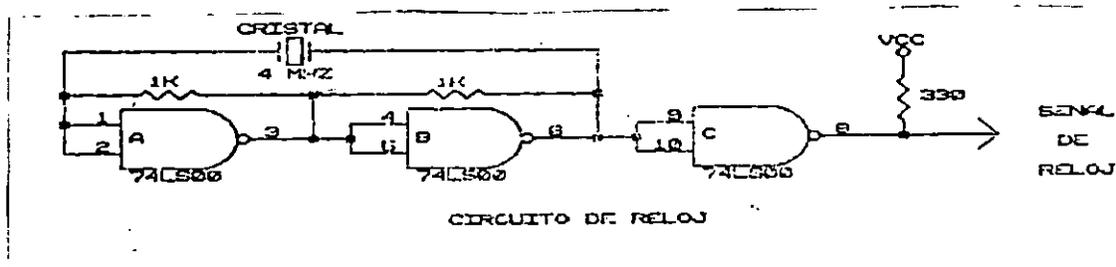


Figura 2.5. Circuito generador de reloj

La señal de reloj será generada mediante el control de un cristal con una frecuencia máxima de 4 MHz para la versión del microprocesador utilizado (Z80 A CPU), tal y como se muestra en la figura 2.5, y como puede observarse es

un circuito bastante simple. La realización de pruebas se hace con frecuencias menores de los 4 MHz para lo cual se necesita variar dicha frecuencia; así que en esta etapa es necesario el utilizar el generador de frecuencia proporcionado por el protoboard del laboratorio de la Escuela de Ingeniería Eléctrica, el cual es compatible con TTL. Terminadas las pruebas se fija la frecuencia seleccionada para el sistema.

2.3.3 CIRCUITO DE PASO UNICO DE INSTRUCCION

Este circuito es hecho con el objetivo de poder depurar el programa del usuario y poder mostrarle al mismo el buen funcionamiento del sistema, el circuito es mostrado en la figura 2.6. El circuito realiza la función de paso unico mediante la utilización de señales de control generadas por el microprocesador Z-80 durante la ejecución de un programa. Las señales de control de interes son M1 y WAIT en donde la primera es una salida y la segunda es una entrada, con esto se logra que el usuario pueda observar la dirección de cada instrucción.

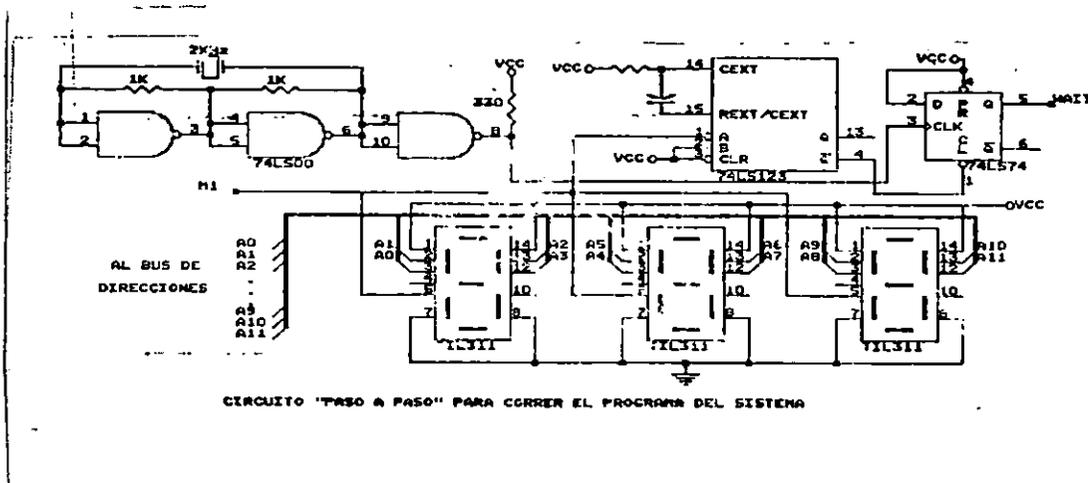


Figura 2.6 Circuito de paso unico.

2.3.4 CIRCUITO DE RESET

La función que realiza un circuito de reset es de gran importancia, ya que permite interrumpir la ejecución del Z-80, y con ello poder detener y reiniciar en cualquier momento dicha ejecución. Por lo tanto, es una señal de gran importancia, la cual se manifiesta al momento de energizar al sistema y cuando los programas se esten ejecutando indebidamente.

El circuito de reset puede ser manual o automático, o bien una combinación de las dos funciones. El circuito de reset manual es de gran utilidad, ya que permite al usuario interrumpir al sistema cuando así lo desee. Mientras que el circuito de reset de encendido automático permite a la CPU comenzar la ejecución del programa inmediatamente después del encendido del sistema.

Dado que ambas funciones son importantes en la implementación de la CPU, se realiza un circuito que las combine. Siendo el circuito implementado una combinación de ambos. Ver figura 2.7.

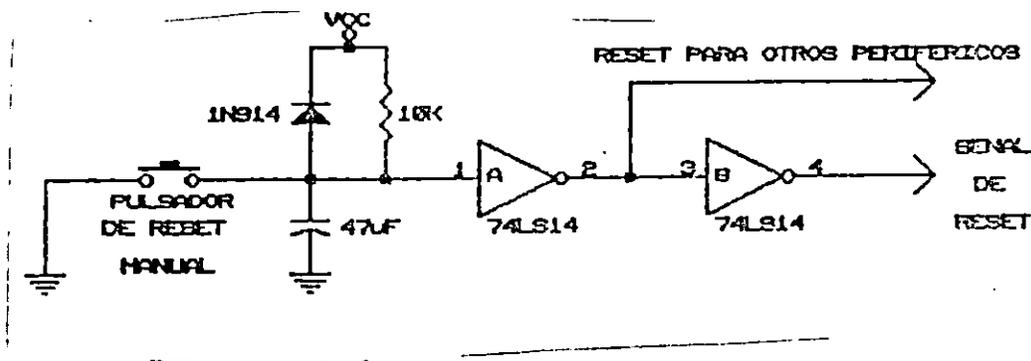


Figura 2.7 Circuito de reset

El reset manual lo proporciona el pushbutton, el cual al ser presionado aterriza el pin reset del Z-80. La función de reinicialización es indispensable, ya que el microprocesador exige que el terminal de reset se mantenga en nivel bajo por lo menos durante 50 microsegundos después de que se haya alcanzado los +5 voltios. en los terminales de alimentación del sistema.

Es por eso que el circuito de reset a utilizar en este sistema tendrá un pulso con una longitud de 50ms, el cual se determina por un producto RC, con valores de $R=10K$ y $C=4.7\mu F$, Dicho producto nos proporciona un valor mucho mas largo que el valor necesario para que el sistema alcance los +5 voltios. El diodo del circuito funciona como seguridad que permite la descarga rápida del capacitor al desconectar la alimentación. Mientras que los inversores con trigger Schmitt aumentan la confiabilidad del diseño.

2.3.5. BUFFERING DE SALIDA DE CONTROL, BUSES DE DATOS Y DE DIRECCIONES

Cuando el usuario, pretende conectar una serie de dispositivos a la salida de una compuerta o en el caso específico a los buses de datos y direcciones del Z-80. Esto puede ocasionar serios problemas ya que la carga presentada por todos los dispositivos puede ser mayor que la capacidad de excitación de dicha salida. Por lo tanto, es de gran importancia considerar el fan-out de las salidas del Z-80; en este caso es indispensable la utilización de Buffer's ya que básicamente estos permiten aumentar la capacidad de manejo de las señales provenientes del microprocesador, además de conseguir un alto grado de aislamiento entre el Z-80 y otras secciones del sistema.

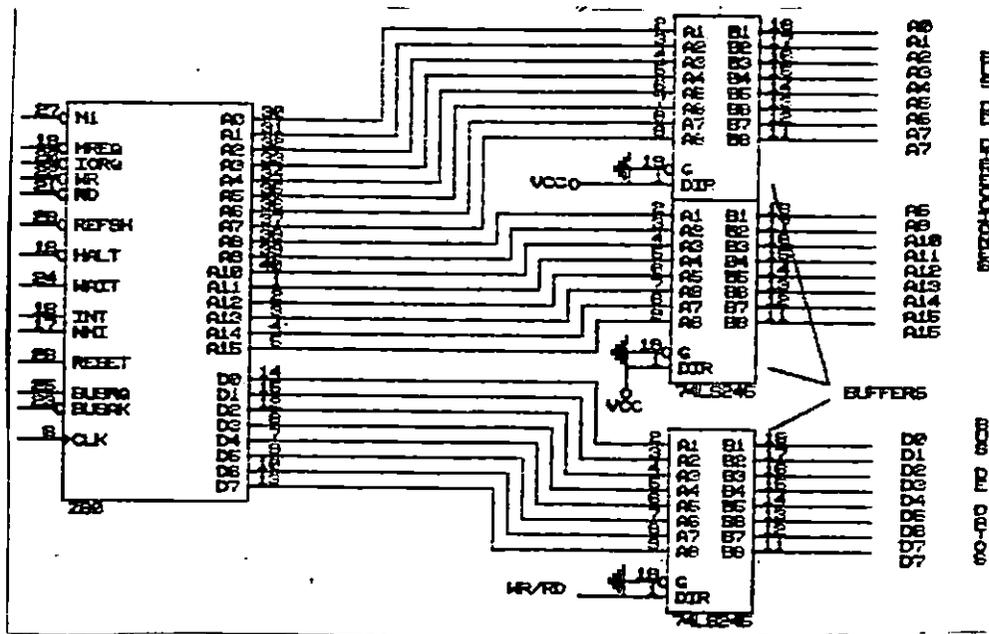


Figura 2.8. Z-80 con buses de direcciones y datos provistos de buffers

En nuestro caso para compensar este problema se utiliza el IC 74LS245 que es un transceptor octal triestado, el cual a la vez es un dispositivo de amplificación o separación que presenta la ventaja de ser bi-direccional y no-inversor. Además de poseer las entradas a un lado del IC y las salidas al otro del mismo, lo que es de gran ayuda al realizar los impresos del circuito. En la figura 2.8 se muestra la configuración del microprocesador provisto de buffers en los buses de datos y direcciones.

Tanto el bus de datos como el de direcciones y las salidas de control pasan al estado lógico de alta impedancia al pasar el BUSACK al estado de 0 lógico, y en este caso los

IC 74LS245 pasan también a este estado, siendo esto de gran importancia ya que permite que otro dispositivo pueda tomar el control del bus de direcciones.

2.3.6 DECODIFICACION DE MEMORIAS Y PUERTOS DE E/S

Para realizar esta decodificación es necesario que la CPU reconozca la selección ya sea de memoria, puerto de entrada ó de salida. Para lo cual se realiza un combinacional con las señales MREQ, IORQ, RD y WR del Z-80, las cuales son de gran importancia para ejecutar cada una de las funciones donde:

MREQ: petición de memoria

IORQ: petición de entrada/salida

RD: petición de lectura

WR: petición de escritura

Para la decodificación de puertos de E/S las señales IORQ, WR y RD se hacen pasar por las mismas compuertas tal y como se muestra en la figura 2.9. Y así obtener las señales de strobe de lectura de un puerto de entrada (IORD) y strobe de escritura de un puerto de salida (IOWR).

De igual forma se realiza la decodificación de lectura ó escritura de memoria, cambiando únicamente la señal MREQ por la señal IORQ.

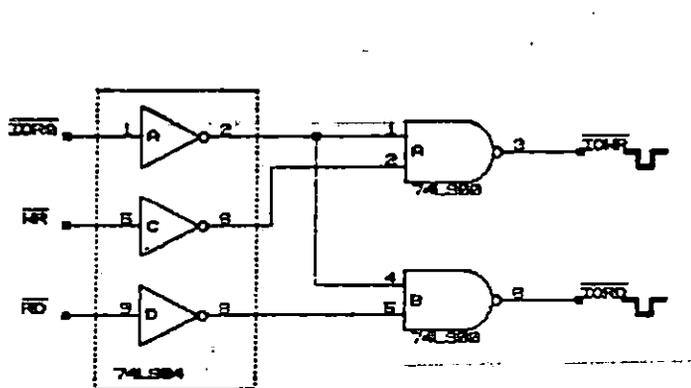


Figura 2.9. Decodificación de puertos de E / S .

Obteniéndose así, las señales strobe de lectura de memoria (MRD) y strobe de escritura de memoria (MWR) cuya configuración se muestra en la figura 2.10.

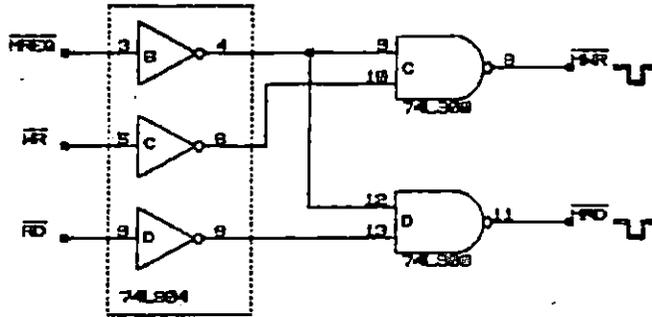


Figura 2.10 Decodificación de memoria de lectura / escritura.

En la decodificación de puertos de Entrada/Salida se requiere implementar un sistema que seleccione más señales de petición de puertos de E/S, por lo que es necesario cambiar el sistema mostrado anteriormente por el de la figura 11, en el que se involucra una decodificación con un número mayor de señales de petición de puertos de E/S. Utilizando una cantidad de chips lo más reducido posible, incluyendo en el mismo sistema un circuito combinacional con compuertas lógicas que eliminen el riesgo de equivocarse al seleccionar un puerto de E/S. Para el caso se utilizan decodificadores de 3-8 líneas de petición de puertos de E/S en el sistema; tomando en cuenta que cada señal de petición habilita un módulo de Entrada ó de Salida, con 8 puntos de conexión, de modo que el sistema se pueda expandir a 64 puntos de E/S. En la fig.13 se muestra el diagrama de bloques para una decodificación de los 64 puntos de conexión de E/S.

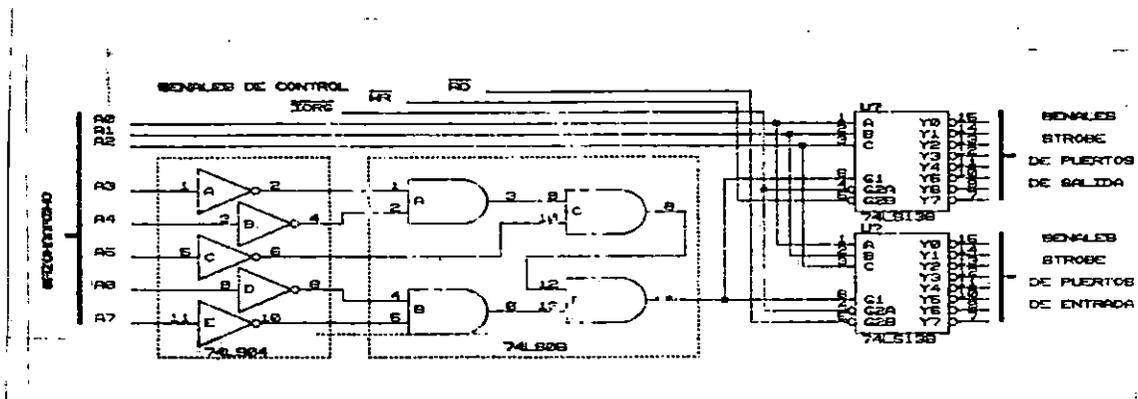


Figura 2.11 Decodificación de E/S expandida

Si se necesitan más de 8 strobus, el decodificador 74LS138 se intercambia por otro decodificador que proporcione las señales de petición necesarias, por ejemplo: el 74LS154 (decodificador de 4-16) que selecciona 16 líneas de petición de puertos de E/S.

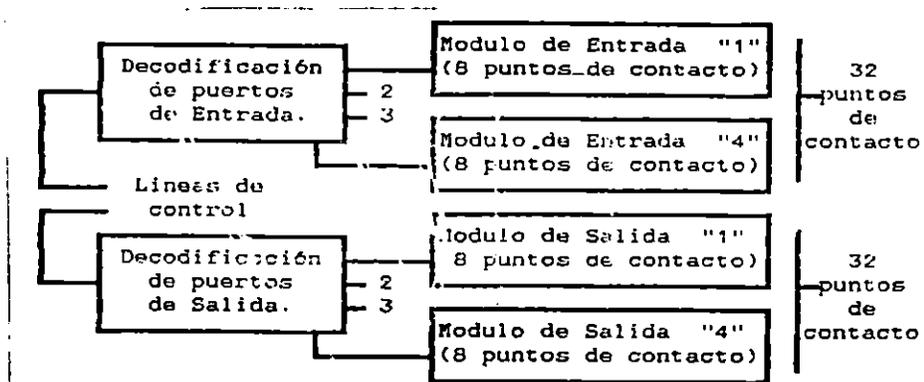


Figura 2.13 Diagrama de bloques de decodificación de 64 puntos

Para la decodificación de memoria no se necesita el utilizar un decodificador de muchas líneas de petición, ya que en el banco de memoria solo se necesita una memoria RAM y una ROM. Por lo tanto, solo se intercambia el circuito de la figura anterior, por el mostrado en la fig.14 para realizar la misma función con un solo integrado

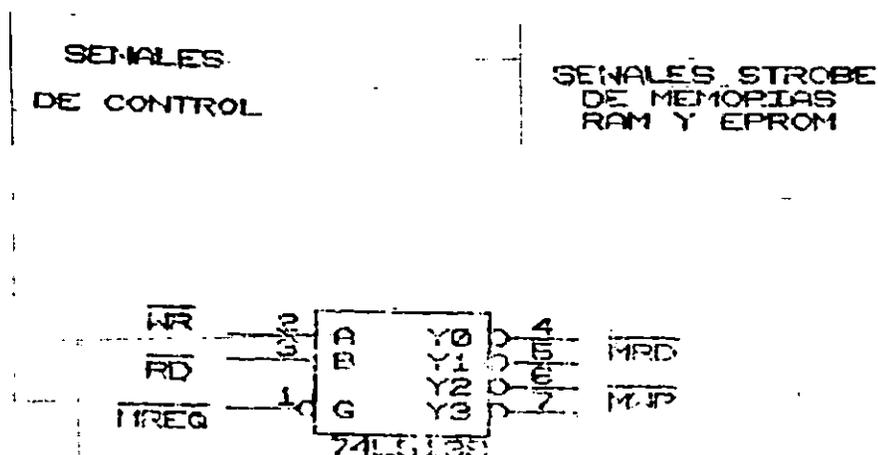


Figura 2.14 Decodificación de memoria

2.3.7 PUERTOS DE ENTRADA Y SALIDA

Dado que en los sistemas con microprocesador se necesita retener la información que aparece en el bus del sistema para ser usada posteriormente, es de gran importancia el utilizar latches para dicha función. El dispositivo utilizado en el sistema mínimo a realizar es el latch octal transparente 74LS373 el cual en modo transparente se comporta como 8 buffers, con los niveles lógicos de entrada (D) transmitidos hacia las salidas (Q). Al encontrarse en modo de detención, las salidas permanecen en los estados lógicos que tenían cuando se recibió la señal de retención. En la figura 2.15 se muestra el pin out del CI 74LS373.

Al utilizar este dispositivo como puerto de salida, el pin 1 (OE) se aterriza mientras que el pulso de control IOWR se conecta al pin 11 (G). Las entradas se pasan hacia las salidas cuando el pin 11 está en alto activa y los retiene cuando esta en estado bajo.

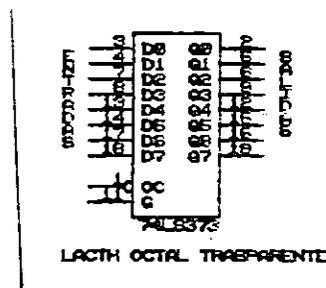


Figura 2.15. Pin out del IC 74LS373.

Al ser utilizado como puerto de entrada el pin 11 (G) se conecta a +Vcc, mientras que el pulso de control IORD se conecta al pin 1 (OE). Las entradas pasan a las salidas cuando el pin 1 está en baja activa y lo retiene cuando está en estado alto.

2.3.8 MEMORIA

Tanto las instrucciones de programación como los datos deben almacenarse en un tiempo adecuado, por lo que es indispensable el tener en un sistema de computadora un banco de memoria.

En el sistema a realizar se utilizan dos tipos de memoria, una memoria RAM de 2k y una EPROM de 4 K (ver figura 2.16).

La EPROM es una memoria sólo de lectura que puede ser programada por el usuario y también puede borrarse y reprogramarse tantas veces como se desee; Además de ser una

memoria no volátil, o sea que la información no se pierde al suspender el suministro de energía. Esta memoria es empleada para almacenar programas y datos en equipos controlados por microprocesadores.

La RAM es una memoria de acceso aleatorio, lo cual quiere decir que cualquier localidad de memoria se puede acceder fácilmente como cualquier otra. Esta memoria es utilizada para almacenar información temporal de programas y datos. Su mayor desventaja es que es volátil por lo que al suspender el suministro de energía pierde la información almacenada en ella.

El motivo de utilizar las memorias EPROM y RAM de 4 y 2k respectivamente es simplemente para tener suficiente cantidad de bytes disponibles y hacer más versátil el diseño. Aunque en los mini-PLC la cantidad de memoria disponible aproximadamente de 1K, con lo cual se tiene la capacidad de almacenar unos 500 pasos de programación, suficiente para una aplicación sencilla.

El hecho de utilizar una memoria EPROM es por que en su mayor parte, es usada como una ROM para periodos prolongados de tiempo, borrada cuando se tenga la necesidad de cambiar la información que contenga y luego reprogramada para su respectivo uso.

La distribución del mapa de memoria es mostrada en la tabla 2.2, la cual muestra la cantidad de memoria RAM a utilizar.

Tabla 2.2 Mapa de memoria.

Memoria	Dirección de localización
ROM (4K)	DE 0000 A 0FFF HEX.
RAM (2K)	DE 1000 A 17FF HEX.
ROM (2K)	DE 000 A 7FF HEX.
ROM (2K)	DE 800 A FFF HEX.

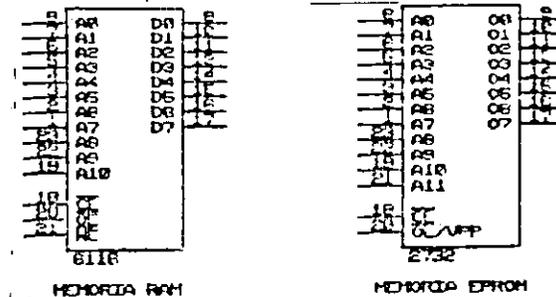


Figura 2.16. Memoria RAM y EPROM.

2.4 MODULOS DE ENTRADA Y SALIDA

Los autómatas pequeños o "minis" tienen un total de puntos para E/S bastante limitado, y normalmente pueden tener su total de puertos expandidos al añadirse uno o dos módulos extra; y si el grado de complejidad de alguna planta exige mayor atención, entonces se requiere el reemplazo de la unidad completa. En cambio los PLC's mayores tienen un rango de soporte de puntos de E/S mucho mayor.

Como se ha dicho las funciones lógicas que se han de desarrollar son las básicas del álgebra de Boole, ya que esta permite traducir y manipular las combinaciones de variables sencillas tales como AND, OR y NOT. Para estas aplicaciones los automatismos "todo o nada" (TON) son suficientes, esto es debido al hecho de que su comportamiento se describe fácilmente mediante las variables booleanas.

Los módulos de entrada o salida puede tomar dos estados: estado de reposo y estado accionado, a cada uno de estos estados corresponde una señal de salida (eléctrica, neumática,...) de nivel determinado o bien 1 o bien 0 por convención, tal y como se muestra en la figura 2.17. Si la magnitud medida es analógica, es decir que esta puede tomar una sucesión continua de valores, un captador de umbral traduce este valor en señal "todo o nada", según los umbrales alcanzado.

*: magnitud física portadora de una señal, tensión, presión

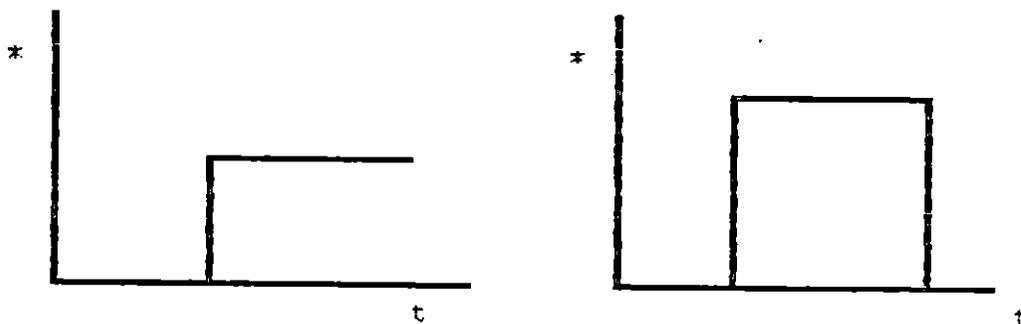


Figura 2.17. Captadores "todo o nada".

En este caso el sistema cuenta con 16 puntos I/O, dividiéndose en 8 puertos de entrada y 8 de salida; con una capacidad de expansión con la cual se podrán acceder como un máximo de 32 puntos de entrada y 32 de salida; pero para trabajar con este sistema expandido será necesario el acoplarle nuevos módulos de E/S, y estarán conectados con el CPU por medio de uniones alámbricas "hilo a hilo". El módulo

de entrada debe permitir a la unidad central del autómata efectuar una "lectura" del estado lógico de los captadores asociados al mismo (Módulo de 4,8 o más entradas).

2.4.1 MODULO DE ENTRADA

Un módulo de entradas debe de permitir a la unidad central de procesos del autómata efectuar una "lectura" del estado lógico de los captadores que están asociados al mismo. A cada entrada corresponde una vía que trata la señal eléctrica para elaborar una información de bits, la cual es almacenada en los latches. Periodicamente el procesador pregunta (direccionando) a través del módulo, el contenido de la palabra de entrada del módulo, dicho contenido de palabra se copia entonces en la memoria de datos del autómata; pero, en este caso se almacenan bits independiente, informando cada uno de ellos de una etapa en particular del proceso, por lo que no se ocupa siempre la palabra completa ya que en muchas ocasiones no es necesario para el control de un proceso en particular; pero esto no es limitante sustancial ya que el sistema, por medio de software puede configurarse para que el conjunto de bits de entrada forme "la palabra" de entrada;

El módulo de entrada opera internamente con niveles de alimentación TTL, procesando señales de 24 Vcc y manejando una corriente de entre 300 a 500 mA, se compone de 8 canales de entrada, donde cada uno de ellos es una vía o un bit proveniente del mundo real. Cada vía se y se aísla eléctricamente del exterior (debido a rebotes de contactos, señales parásitas, etc) por razones de fiabilidad y seguridad (aislamiento galvánico) lo cual se hace por medio de opto-acopladores 4N25 que consisten en un LED y un foto transistor, que forman el par opto-acoplado que permite que una pequeña señal pase a travez de el, pero que lo protegerá de cualquier pico de alto voltajeo trasientes que se produzca proporcionan un aislamiento de hasta 2.5 KV, obteniendo una señal de el colector de nivel TTL, la cual por razones practicas es invertida y puesta en forma haciéndola pasar por un inversos con trigger smith y posteriormente llega a un "one-shot" no redisparable (74LS221) con lo cual obtenemos un pulso digital de 10 ms, debido a la fórmula de duración del pulso que es de $1.1RcC$ y donde $R=1\text{ K}$ y $C=10\text{ uF}$, teniéndose un pulso que es suficiente para que pueda ser leído por el microprocesador. (ver figura 2.18) Debido a las condiciones de aislamiento galvánico obtenidas se hace posible que el PLC sea directamente conectado a los captadores asociados sin necesidad de otro tipo de circuitería intermedia o tratamiento de las señales; este módulo de entrada TON esta diseñado de manera tal que simplifique al máximo la función

de interconectar la CPU con el mundo exterior por medio de ocho terminales exteriores, una para cada punto de entrada, además de 8 LEDs indicadores del status de cada señal de entrada, lo que facilita el chequeo del sistema en plena operación, puesto que indica que punto de entrada ha recibido una señal desde los captadores del proceso externo, y su secuencia de aparición está relacionada con la parte del proceso que se encuentra desarrollando.

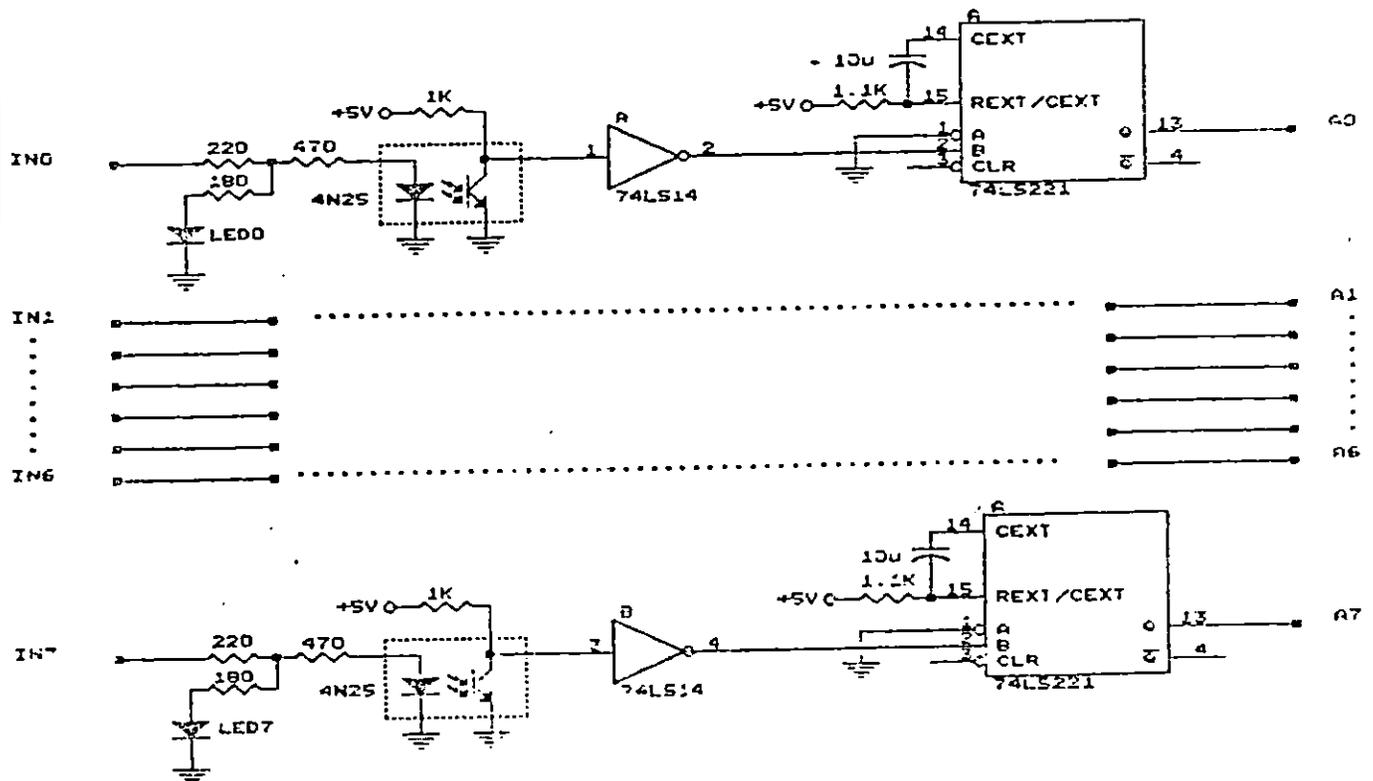


Figura 2.18. diagrama de módulo de entrada TON.

2.4.2 MODULO DE SALIDA

El módulo de salida permite al autómata programable actuar sobre los accionadores. Dicho módulo realiza la correspondencia estado lógico — señal eléctrica. Periodicamente el procesador direcciona el módulo y provoca la escritura del bit de una palabra de memoria en los puertos de salida del módulo.

Al igual que el módulo de entrada, el de salida consta de 8 etapas o vías iguales que sirven para enviar las señales hacia los pre-accionadores; desde el CPU le llegan señales digitales, pulsos de una duración en el orden de los micro segundos, dicho pulso es enviado a un buffer que pone en forma la señal ya que consta de trigger smith, de ahí se tiene dos opciones con esta señal, la primera es enviarla directamente hacia los opto-acopladores y de ahí hacia los circuitos de mando que van hacia afuera del PLC, y la otra es enviarla hacia un "one-shot" para generar un pulso de duración fija que varía entre 0.5 y 3.5 segundos, con lo que obtenemos tiempos que pueden servir para fines específicos por su duración.

Esta división de las señales entre las de tiempo fijo y las de tiempo programable se realizó con el objeto de optimizar el funcionamiento del autómata programable pues se tiene capacidad de enviar pulsos de duración específicos que ahorren tiempo en la duración del programa, ya que con una instrucción del PLC nos genera un pulso de varios microsegundos, la cual es adaptada para que a la salida del autómata se tenga un pulso de varios segundos que pueda activar un relé y accionar o apagar un motor, por ejemplo, sin necesidad de perder tiempo en el desarrollo del programa el cual sigue adelante en posteriores etapas de lectura de otros puntos de salida. O también el caso de los pulsos programados que sirven para pre-accionadores tales como calentadores o electro-válvulas que permanecen activadas durante el tiempo que dure el pulso, no teniendo una duración específica; si no hasta que por medio del proceso y del software sea desactivada dicha salida.

Los etapa de aislamiento galvánico se divide en dos tipos puesto que utilizamos dos clases de opto-aisladores, los de led-fototransistor utilizados en la tarjeta de entrada, y los de un par darlington de foto-transistores que nos proporciona una alta ganancia para manejar una amplia gama de pre-accionadores; puesto que se consta de 4 TIL 44 y los demás son 4N25, ver figura 2.19, teniendo capacidad de manejo de corriente amplia. Obteniendo módulos de gran flexibilidad y sumamente económicos y simplificados, lo que facilita su mantenimiento y conexión con los actuadores externos que controlan el proceso.

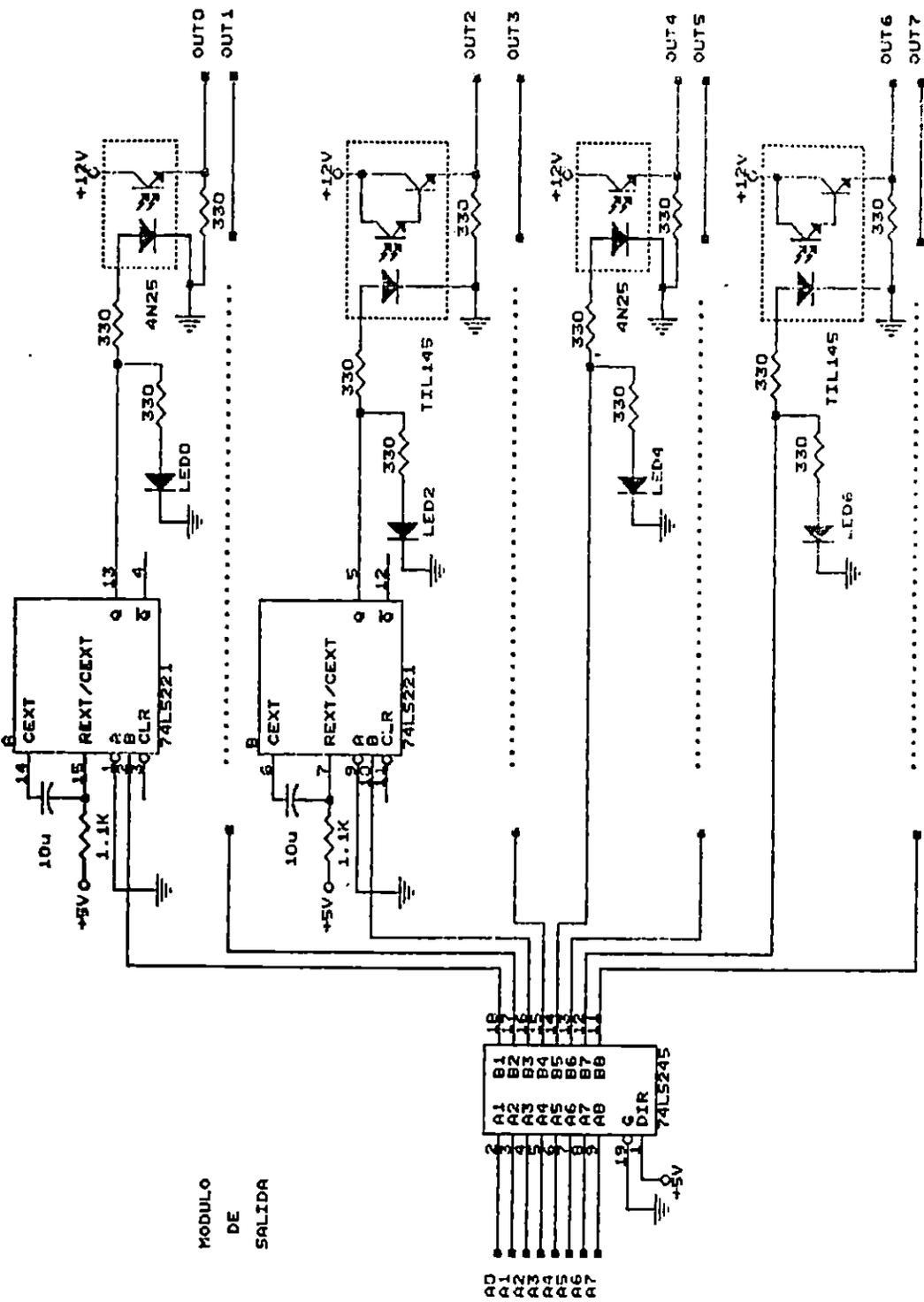


Figura 2.19. Diagrama del módulo de salida TON

2.4.2 LOS CAPTADORES

Los captadores son los componentes externos que envían las señales que provienen del proceso controlado hacia el PLC, ellos suministran la información en retorno necesaria para la conducción del proceso. Pueden detectar posiciones, presiones, temperaturas, caudales, etc. esquematiza una unión hilo a hilo entre un captador y el autómata programable; cuando el captador cierra el circuito, los bornes del módulo de entrada se activan, de esta forma la señal se toma en consideración por el autómata. La conexión de los captadores se realiza directamente en el PLC sin requerir interfaces, gracias a los módulos de entrada que incluyen las adaptaciones y protecciones necesarias.

Es de hacer notar que los diferentes tipos de captadores se enriquecen continuamente por constituyentes con cada vez mejores características técnicas, más especializadas y más fáciles de emplear. Se pueden distinguir las grandes familias de captadores por los tipos de señales que transmiten

- Señales "Todo o Nada"
- Señales numéricas
- Señales analógicas

Un captador "todo o nada" solamente puede tomar dos estados: estado de reposo y estado accionado, los cuales dicho sea de paso, son los más empleados en la automatización corriente, ya que suministran una señal binaria, lo cual los hace propios para trabajar con variables booleanas, pues solo toman dos valores 0 y 1; aún si la magnitud medida es analógica, se trabaja con captadores TON de umbral que traduce dicha entrada a "todo o nada" según los umbrales alcanzados. En nuestro caso nos limitaremos a los captadores TON que son los más utilizados para los mini PLCs.

Los captadores numéricos transmiten los valores numéricos precisando posiciones, presiones, temperatura, etc. En forma de combinaciones de señales binarias que se pueden leer ya sea en paralelo en varios conductores o bien en serie en un solo conductor.

Los captadores analógicos, como su nombre lo indica, traduce los valores de presiones, de temperaturas, etc. en forma de una señal que evoluciona entre dos valores límites

Volviendo a lo anterior, se ha dicho que los captadores TON son los más utilizados, y de estos los captadores de

posición tienen bastantes aplicaciones en las máquinas de producción; dividiéndose estos en: captador de contacto, detector de proximidad y detector de distancia. Siendo el más sencillo y empleado el detector de contacto, el cual conmuta cuando el objeto a detectar acciona físicamente el elemento portátil del captador. Su conmutación se realiza por cierre o apertura de un contacto electromecánico. Desde los minirruptores hasta los grandes interruptores de posición las gamas son muy variadas; pero su principio fundamental se esquematiza en la figura 2.20.

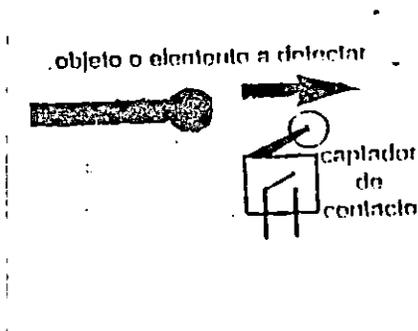


Figura 2.20. Captador de posición

2.4.3 ACCIONADORES Y PREACCIONADORES

Los preaccionadores son quienes reciben la señal procedente de el módulo de salida, que se limita a enviar la señal de mando que recibe el tratamiento especial o la puesta en forma necesaria para que sea recibida por el accionador quien si es el que controla o altera el proceso, el accionador es el destinado a mover el proceso automatizado; o sea que a cada accionador en la máquina (motor, cilindro hidráulico, etc) se encuentra asociado un preaccionador (contactor, variador de velocidad, etc), tal y como se muestra en la figura 2.21

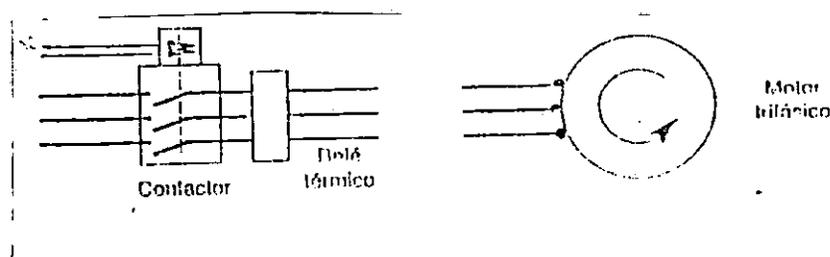


Figura 2.21 Ejemplo de interrelación pre-accionador y accionador

Para responder a las variadas necesidades de las máquinas hay tres tecnologías de accionadores que se complementan:

La primera es la de los accionadores eléctricos. Estos utilizan directamente la energía eléctrica distribuida en las máquinas y toman diferentes formas: motores de velocidad constante ó variable, válvulas eléctricas de flujo, resistencias de calentamiento, electroimanes, cabezas de soldadura por resistencia.

Los preaccionadores asociados a estos accionadores eléctricos son principalmente los contactores y variadores de velocidad, equipados con los sistemas de seguridad necesarias.

También se tienen a las accionadores neumáticos, que son los que utilizan directamente aire comprimido distribuido en las máquinas sencillas, y presentados bajo formas muy diversas, los cilindros neumáticos se utilizan para numerosos movimientos: transferencias, aprietes, marcadores, ensamblajes, moldeadores, etc. Los distribuidores son los preaccionadores que están asociados a los mismos. Reciben una señal neumática o una señal eléctrica cuando están equipados con una válvula eléctrica.

Y por último los accionadores hidráulicos, que son cilindros o motores que sólo se utilizan cuando los accionadores eléctricos y neumáticos no dan satisfacción ya que exigen que se instalen en la máquina un grupo generador de presión. Son necesarios cuando los esfuerzos a desarrollar son muy importantes, tal y como los que se ejercen en las prensas, o cuando las velocidades lentas se deben controlar con una gran precisión.

CONCLUSIONES

- En la industria del primer mundo, los controladores lógicos programables se ha vuelto una herramienta indispensable en el área de la automatización, por lo cual esta ha avanzado aceleradamente en muy pocos años, ofreciendo actualmente una diversidad de opciones ha elegir satisfaciendo cualquier demanda en cualquier rama industrial que lo exija, teniendo autómatas desde los mas sencillos hasta los mas complejos que pueden intercambiar información entre si por medio de redes que pueden ser desde locales hasta WAN.

- En nuestro País la utilización de los PLC's no ha desarrollado tanto en la industria, debido a su alto costo y al poco personal capacitado para su atención, además de la restringida oferta existente, ya que una sola empresa acapara el mercado, ofreciendo su PLC el cual para aplicaciones no muy complejas resulta subutilizado y demasiado caro; por lo que es conveniente presentar una alternativa mas sencilla y económica.

- Nuestro diseño está destinado a sencillas aplicaciones industriales en donde sustituirá arreglos de relés, y por su versatilidad podrá ser instalado y configurado de bastante manera, además de tener la capacidad de ampliación hacia situaciones mas complejas.

- La reducida circuiteria de apoyo con que se puede construir un sistema mínimo con el Z-80 resulta ventajoso ya que reduce los costos en la construcción del prototipo, y puede obtenerse un rendimiento aceptable.

- Las señales del tipo TON para captadores de posición o movimiento, y sus respectivos pre-accionadores, son las mas utilizadas industrialmente sobre todo en aplicaciones no muy complejas, y su diseño es bastante sencillo por lo que partiremos desde la realización de módulos de este tipo.

BIBLIOGRAFIA

- Telemecanique, "LOS AUTOMATISMOS PROGRAMABLES".
Madrid, Editions Citef, 1991.

- Steve Ciarcia, "CONSTRUYA UNA MICRO-COMPUTADORA BASADO EN EL Z-80".
EE UU, Mc Graw-Hill, 1981.

- Joseph C. Nichols, "MICROPROCESADOR Z-80".
EE UU, Marcombo editores, 1984.

- Ian G. Warnock, "PROGRAMMABLE CONTROLLERS".
United Kingdom, Prentice Hall, 1988.

- Ronald J. Tocci, "SISTEMAS DIGITALES".
México, Prentice Hall Hispanoamericana, 1987.

CAPITULO III

EL SOFTWARE DEL PLC

INTRODUCCION

Los sistemas basados en microprocesadores, como es sabido, necesitan para su funcionamiento de la existencia de un listado de instrucciones o programa que dirija y coordine las operaciones y manipulaciones de datos necesarias para la obtención de resultados esperados. En nuestro caso para un autómata programable los resultados deseados son el fiel control de el proceso para el cual se ha elaborada una o varias rutinas de software. Este programa o tipos de programa deben de poseer las características necesarias para el óptimo funcionamiento del PLC, ya que de el depende que los actuadores cumplan las funciones para las cuales han sido diseñadas.

En el PLC el programa a utilizar, programa monitor, esta constituido con las instrucciones almacenadas necesarias, en memoria ROM, las cuales dirigen y determinan las funciones y la capacidad de el sistema, teniendo la opción de adaptarse en una amplia gama de aplicaciones de control industrial; desarrollándose en esta etapa de la siguiente manera:

- a) Capacidad de procesamiento de datos e instrucciones proporcionadas al sistema para habilitar al autómata en la ejecución del programa que el usuario a vaciado en la EPROM para un proceso industrial en particular.
- b) Capacidad de ejecutar el programa del usuario, que rige al sistema industrial.
- c) Capacidad de interacción con el sistema industrial, una vez ejecutado el programa del usuario, el PLC es capaz de enviar señales eléctricas necesarias para que el estado del sistema bajo control corresponda a aquel de las variables que intervienen en le proceso.

3.1 LENGUAJES EN CONTROL DE PROCESOS

Los fabricantes de sistemas para control de procesos han desarrollado lenguajes para el control de dichos procesos, de alto nivel, el cual está basado en un estilo de "conversación de entradas". Existen diseños específicamente desarrollados para personal poco capacitado en la estructura

de operación del sistema, usando una forma de presentación de pregunta y respuesta. Aquí el usuario o programador puede introducir las estrategias requeridas como respuestas a las interrogantes sobre el tipo de control deseado, junto a los parámetros relevantes. El usuario debe de construir un "diagrama mímico del proceso", o sea una descripción detallada del tipo y forma de control a desarrollar.

A este software se le ha denominado como lenguaje de control de procesos (PCL), el cual ha sido ampliamente desarrollado por las industrias constructoras y afines de los autómatas, tenemos como ejemplo los lenguajes comercialmente extendidos tales como:

- PROTEUS (Westinghouse),
- BICEPS (Honeywell),
- STEP-5 (Siemens),
- PCL (Hitachi),
- GRAFNET (Telemecanique)

Estos PCL's son implementados solamente en sistemas de unidad central (al mando de una PC), no se encuentran disponibles para sistemas de microprocesadores; por lo que es necesario implementar otro tipo de software que ayude para sistemas basados en una microcomputadora, el cual consiste en una microcomputadora con disco flexible, programador de memorias EPROM's con su software asociado (usualmente un editor, assembler o compiler, linker) esto compone al circuito emulador que se emplea a algunos PLC's para introducir la información al microprocesador objetivo, ver figura 3.1. Todo esto es muy importante ya que sirve para probar el software de operación antes de su instalación. Esto es necesario para producir el código objeto que se vacía en la EPROM para ejecutar la tarea requerida. Para sistemas donde se involucren mini-PLC's la programación es realizada de una manera aún mas sencilla ya que por su nivel no es necesario la disposición de un lenguaje de alto nivel, ya que su utilización es para control de máquinas pequeñas; por lo que su programación debe de efectuarse de una manera mas sencilla o bien disponer del software comercialmente disponible; también se presenta como solución la alternativa de modificar el contenido de la memoria por medio de una PC, teniendo así la opción de desarrollar un software sencillo para cada aplicación contando con las ventajas del sistema MDS (Microcomputer software development). Para lo cual es necesario desmontar la EPROM para modificación; esta es una alternativa que ha sido elegida por muchos diseñadores y se ha utilizado en esta etapa del trabajo, ya que el por hecho de probar el estado de los dispositivos de Entrada/Salida lo que se recomienda es guardar el programa en esta memoria. por lo que se ha desarrollado esta opción en el sistema ya en circuito impreso.

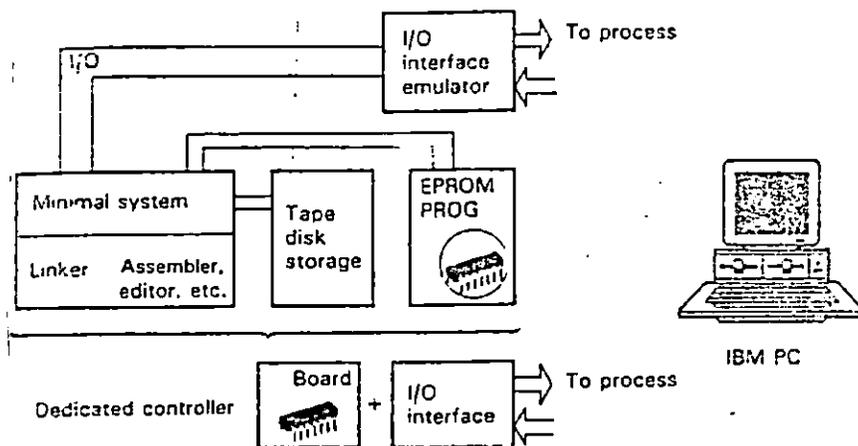


FIGURA 3.1. Sistema MDS

3.2 OPERACION INTERNA Y PROCESAMIENTO DE SEÑAL DEL PLC

Quando se carga un programa en la memoria de el autómata que se diseña, cada instrucción es colocada en una localidad de memoria particular llamada dirección, y también nuestro caso la forma de tratamiento que se le dará a los programas es la misma de operación del Z-80; ya que para la programación nos valemos del lenguaje ensamblador propio de este microprocesador.

El CPU contiene un contador del programa PC, el cual apunta a la siguiente instrucción, ya que contiene la dirección de la siguiente instrucción a ser ejecutada. La presencia de el contador del programa es indispensable y fundamental para la ejecución del programa. Cuando una instrucción es recibida por el CPU esta es puesto en el registro de instrucciones para su decodificación.

Al igual que el microprocesador, cuando el controlador programable es puesto en marcha, el contador del programa apunta a la dirección 0000, y luego atrapa, decodifica y ejecuta las instrucciones programadas; como un ejemplo de esta filosofía de trabajo, primero examina los puertos para evaluar su estado, esto desde una localidad temporal de memoria o un registro interno. Luego la CPU espera por las siguientes instrucciones para completar la información sobre el circuito lógico, la secuencia continua hasta que la última localidad de memoria es alcanzada por el contador de el programa, o hasta la instrucción END; en ambos casos el contador de el programa es reseteado hasta la localidad 0000, ocasionando que el programa comience otra vez; con lo que estamos obteniendo un ciclo continuo de búsqueda, en donde el autómata estará permanentemente a la espera de la activación o desactivación de los captadores de entrada, para poder así ejecutar el programa de control lógico que le ha sido cargado en la memoria EPROM, y de acuerdo a esto enviar las señales pertinentes a los preaccionadores.

3.2.1 PROCESAMIENTO DE ENTRADA/SALIDA

Existen dos métodos diferentes utilizados para el procesamiento de entrada/salida en los controladores programables siendo estos; a) actualizado continuo, y b) copiado I/O masivo.

a) ACTUALIZADO CONTINUO

Esto comprende lo siguiente, que el CPU escudriñe los canales o puntos de entrada tal y como ocurre al ejecutar las instrucciones de un programa, con un retraso interno creado por software para atrapar y reconocer solamente los señales de datos válidos para ser leídos en el microprocesador (en autómatas comerciales se ha estandarizado un retraso típico de 3 ms -previando pulsos debido a rebote de contactos o algún otro ruido del entorno del PLC). Los canales o puntos de salida son manejados directamente cuando las instrucciones OUT son ejecutadas siguiendo los pasos de la operación lógica planteada. Las salidas son atrapadas en la unidad I/O para retener el status hasta que se le envíe una señal de reset o hasta el siguiente actualizado. Ver la figura 3.2.

Este tipo de tratamiento de señal es el más sencillo y es el empleado en los PLC's pequeños, como en nuestro caso, ya que está diseñado para trabajar con pocos puntos de entrada/salida, sin afectar por lo tanto la velocidad de trabajo del sistema ante cualquier eventualidad; por lo tanto el actualizado continuo es el tipo de tratamiento de la señal que haremos nosotros en nuestro prototipo de autómata programable, ya que sus retrasos de ejecución debido al tratamiento que se le da, que es muy parecido al tipo de tratamiento que los microprocesadores dan a su lenguaje ensamblador, no nos afecta de una manera considerable; a la vez que ya estamos acostumbrados a este tipo de tratamiento por tener experiencia con el ensamblador del 2-80.

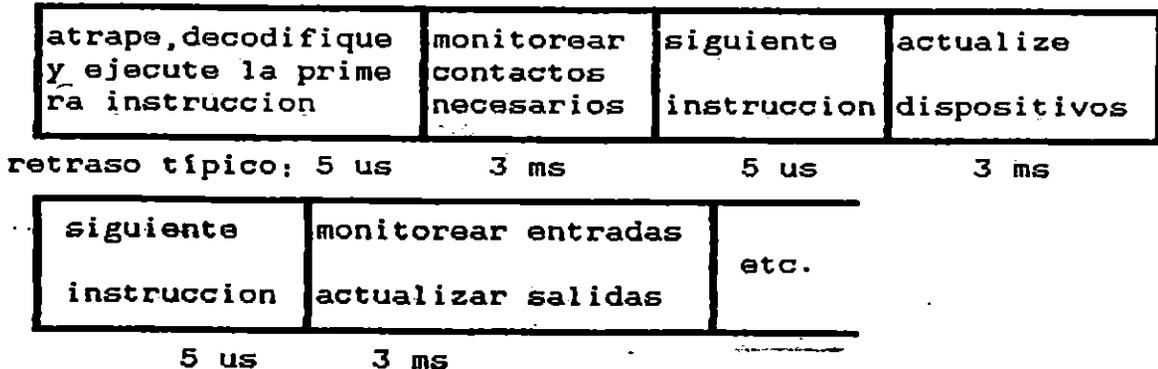


Figura 3.2. Tratamiento de I/O por actualizado continuo

b) COPIADO DE I/O MASIVO

En los autómatas grandes, que tienen la capacidad de manejo de varios cientos de puntos de entrada/salida, se tiene la desventaja de que el CPU puede trabajar con una sola instrucción a la vez, durante el tiempo de ejecución de el programa, el status de cada punto de entrada debe de ser examinado individualmente para determinar su efecto en el programa, por ejemplo si se requiere un retraso de 3 ms para cada entrada, El ciclo total de tiempo para un muestreo continuo se volverá progresivamente mas grande segun el número de entradas.

Para obtener una ejecución mas rápida de los programas, todos los valores de entradas y salidas deben de ser cargadas en un punto particular de el programa, teniéndose una RAM específica para guardar estos valores entre el control lógico y la unidad de I/O, cada entrada y cada salida tiene una celda en esta RAM de I/O. Durante el copiado de I/O, la CPU monitorea todas las entradas en la unidad de I/O y copia sus estatus en esta celdas.

Cuando el programa es ejecutado, los datos guardados son leídos de sus celdas a la vez, y operados lógicamente tal y como han sido programado el PLC, resultando las señales de salida las cuales serán guardadas en la sección de salida de la I/O RAM. Luego al fin de cada ciclo de programa una rutina de copiado de I/O transfiere todas las señales de salida desde la I/O RAM a los correspondientes canales de salida, para que estos manejen los preaccionadores. Una vez hecho esto se vuelve al estado inicial (ver figura - 3.3) - Y - con lo cual obtenemos un retardo de aproximadamente unos 20 ms para un programa de 1 K, lo cual es satisfactorio para autómatas bastante grandes; y aún se puede ahorrar un poco mas de tiempo si se ejecuta una subrutina solamente encargada para el copiado de I/O.

COPIADO DE I/O	COPIA TODAS LA ENTRADAS EN RAM
EJECUCION DEL PROGRAMA	ATRAPA, DECODIFICA Y EJECUTA TODAS LAS INSTRUCCIONES EN SECUENCIA
FIN DE COPIADO DE I/O	COPIA TODAS LA SALIDAS DE LA RAM HACIA LA UNIDAD DE SALIDA, Y TODAS LAS ENTRADAS EN RAM

figura 3.3. Copiado masivo de entradas/salidas

3.3 INSTRUCCIONES LOGICAS

Hasta esta etapa del diseño hemos podido ver que el lenguaje que se emplea para programar a un autómata es bastante simple y su presentación es bastante parecida lo que es el lenguaje ensamblador de los microprocesadores, siendo esto válido aun para los lenguajes comerciales anteriormente citados, pero no solamente este tipo de lenguaje es utilizado, aunque el uso de esas instrucciones lógicas es relativamente facil de aprenderlo y utilizarlo, ya que tambien existen otro tipo de presentaciones que de alguna manera sirve para expresar mas claramente el tipo de control que se efectúa; en este tipo de herramientas se tienen los diagramas de relés o de escalera los cuales son representaciones bastante comunes para describir control de procesos utilizando la lógica de de circuitos con relés ya que este es le lenguaje natural de los PLC's, y en nuestro caso nos servirá para definir la lógica a emplear, esta representación está standarizada para símbolos de gráficos de escalera tal y como se muestran en la figura 3.4.

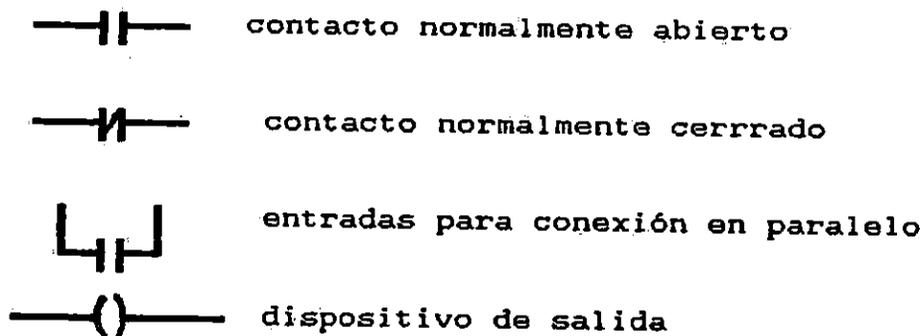


Figura 3.4. Simbolos para gráficos de escalera

Así que se utilizará la lógica siguiente, el circuito de control que se desea emplear, se dibuja el circuito de relés y luego se convierte a mnenónicos de instrucciones que serán vaciadas en la EPROM del sistema, en nuestro caso estos nemónicos serán la representación de la acción que ejecutará el controlador. Tambien es necesario tener en cuenta que estas instrucciones han sido diseñadas tomando en cuenta la forma de los diagramas de escalera, asignando a las entradas y salidas físicas con un operando. La mayoría de conexiones son asociados a relés y contactores, ya que la mayoría de funciones de control envuelven estos servicios así como tambien retrasos de tiempo, conteo de eventos y almacenar algun valor en memoria de un status determinado, estas son funciones internas y no salidas o entradas físicas, pero cada una de estas funciones puede ser relacionada con un contacto el cual es utilizado para controlar diferentes elementos en el programa.

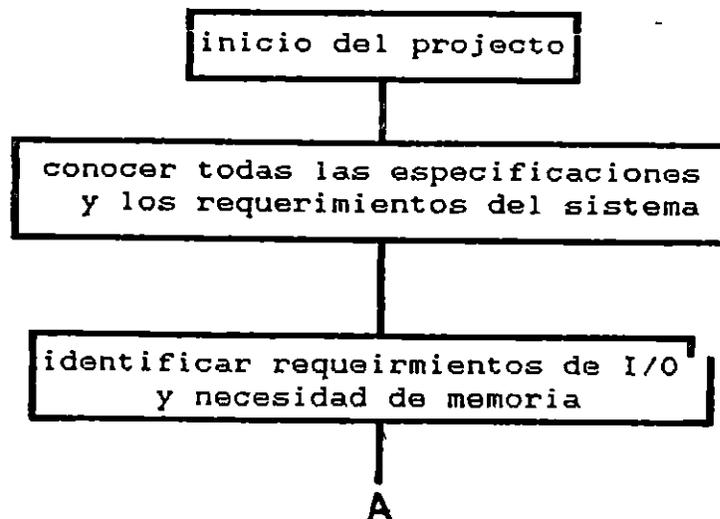
3.4 DISEÑO UTILIZANDO EL SOFTWARE

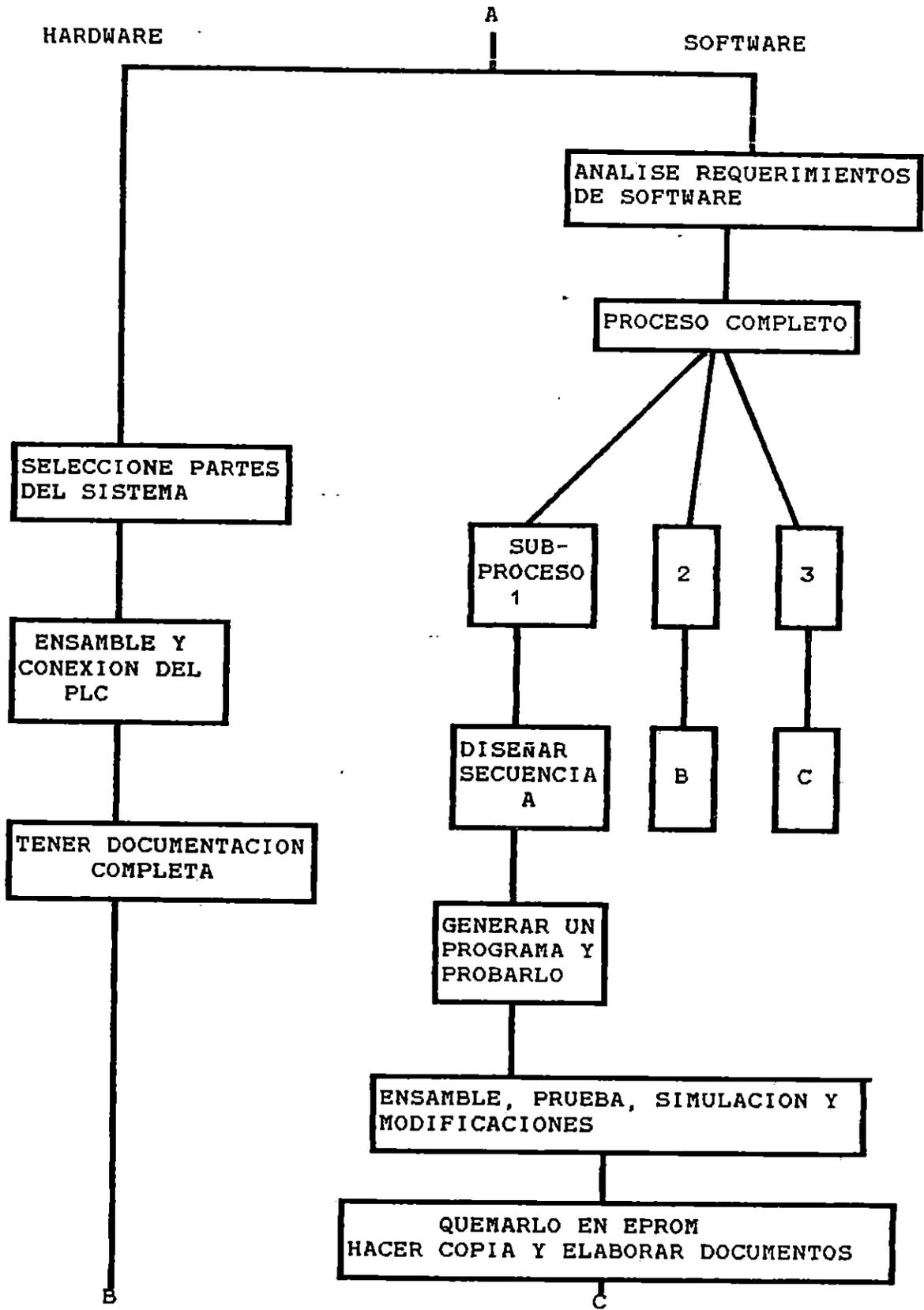
Cuando un diagrama de escalera está siendo desarrollado para una simple acción de control, o se está planeando el diseño para un trabajo determinado, la importancia de este diagrama se vuelve vital y es una etapa que físicamente, en terminos de trabajo, se encuentra aislada de las demas acciones; y esta representación debe de ser lo suficientemente sencilla para entender los términos del circuito que representa y sobre su operación.

Se debe tener un algoritmo generalizado que oriente sobre esta etapa y que sirva para el desarrollo tanto de sistemas sencillos como de complejos, en términos de esta metodología para el diseño, la programación de un autómata no es muy diferente de la programación convencional de una computadora. Así debemos de centrar la atención en los aspectos siguientes:

- Definición de la tarea/especificaciones
- Técnicas para el desarrollo del software
- documentación
- prueba del programa

El procedimiento general debe de ser desarrollado teniendo en cuenta todo tipo de consideraciones, el siguiente diagrama representado en la figura 3.5 nos muestra la estrategia desarrollada para el diseño de una aplicación para un controlador programable. Tomando en cuenta todos los aspectos relacionados como son desarrollo e instalación tanto del hardware como del software, eso si en rutas separadas pero paralelas, proporcionándole al diseñador del programa el máximo de tiempo posible para considerar los requerimientos del control y así generar la solución de software que mejor satisfaga. En el esquema se presentan las estructuras de este proceso que será utilizado para orientarnos en las aplicaciones que desarrollaremos posteriormente.





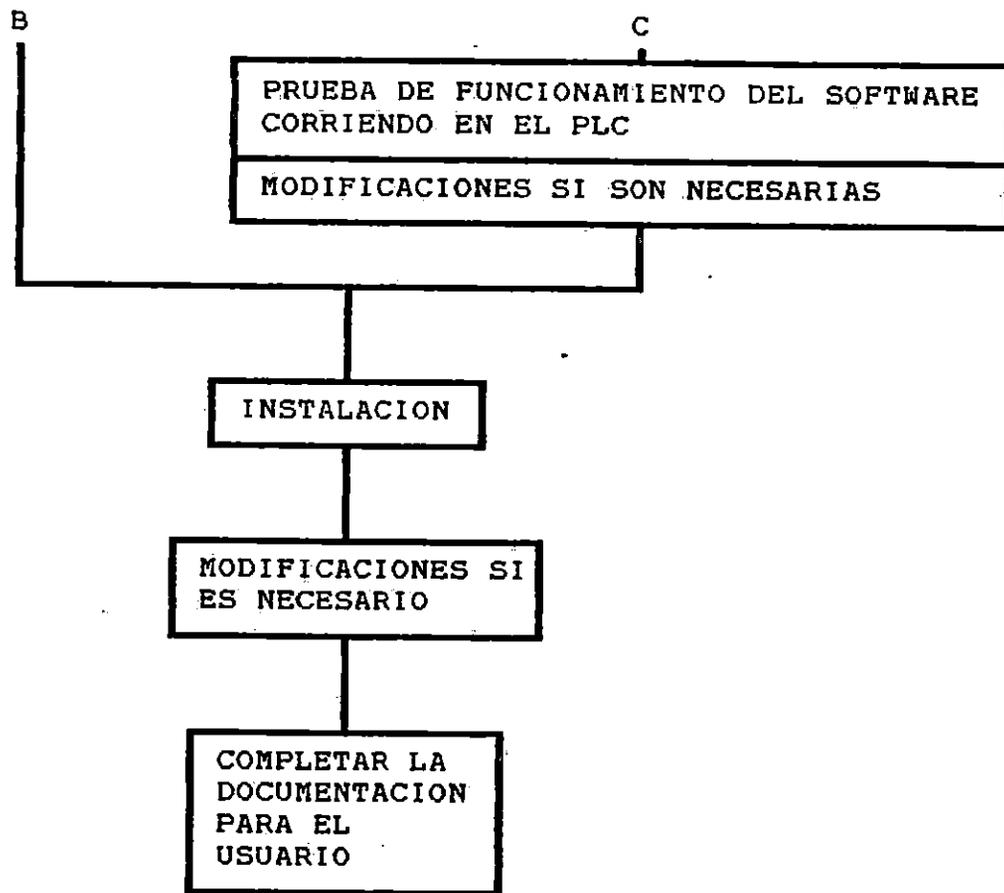


Figura 3.5. Diseño de un procedimiento para PLC.

3.5. FUNCIONAMIENTO DE LOS SISTEMAS

La mayoría de sistemas de control industriales deben de ser considerados como un conjunto de áreas funcionales o bloques, todo esto para ayudar al entendimiento de como el sistema total opera, se considera, por ejemplo, que cada máquina en la planta es un sub-proceso separado; sobre el cual puede comandar un mini-PLC como el que se está diseñando, entonces cada proceso se puede figurar como un bloque el cual es descrito en términos de sus secuencias básicas. Todo esto es muy importante para el diseño de software.

De acuerdo a la lógica de programación existen dos diferentes tipos de proceso que deben de ser usados para implementar lo anteriormente expuesto:

Lógica combinacional, donde las salidas son puramente dependientes de la combinación de las entradas en cualquier instante de tiempo, como por ejemplo la expresión booleana $S = a.b.c (c+d)$. Se puede apreciar que las instrucciones lógicas standar con las cuales puede trabajar cualquier controlador programable, son las expresiones booleanas típicas AND, OR, NOT, etc. las cuales se pueden combinar para aplicaciones simples donde a cada combinación de variable de entrada solo corresponde a un estado de una variable de salida ya que cualquier circuito lógico o programa puede ser convertido a una ecuación booleana la cual ha de mostrar como las entradas están relacionadas o derivan en la salida del circuito.

Redes secuenciales. donde la salida es dependiente no solamente de las entradas actuales sino también de la secuencia de las entradas y salidas previas (involucrando a la memoria en estos eventos).

Los problemas secuenciales durante mucho tiempo han sido resueltos utilizando compuertas lógicas convencionales como bloques, pero tratándolas de manera que expresen e identifiquen las secuencia de las ecuaciones lógicas que controlan el sistema de salidas; este criterio es utilizado en la actualidad, sin embargo, se han desarrollado expresiones simbólicas útiles que proveen especificaciones que simplifican el diseño e implementación de los sistemas secuenciales

3.5.1. REPRESENTACION GRAFICA

La descripción mas detallada de como se soluciona el problema es el trasladar la descripción anterior a una descripción gráfica, ya que este además de ser el más común de los métodos de representación nos da una idea mas clara de como se pretende resolver, a la par de ser el complemento de las expresiones simbólicas, a la vez que son muy apreciados debido a lo siguiente:

- ya sea porque éstos son parecidos a ciertas tecnologías de realización, tales como los contactos electromecánicos y los módulos lógicos.
- o, por el contrario, como no se parecen a la tecnología que se aplicará, permiten una descripción funcional de los problemas secuenciales.

Los principales tipos de representación para automatismos son presentados en las etapas siguientes

3.5.1.1 ESQUEMA DE CONTACTOS O DIAGRAMA EN ESCALA

Este lenguaje gráfico de descripción de los automatismos nació en la época en que únicamente eran disponibles, las tecnologías de contactos para resolver los problemas de mando. Los diagramas de escala consisten en dos líneas verticales que representan la alimentación mas los simbolos de los circuitos que componen el sistema de escalas.

La figura 3.6 muestra como, situando en serie o en derivación los contactos, ya sean estos normalmente abiertos o normalmente cerrados, se puede reproducir cada una de las funciones lógicas Y, O, NO

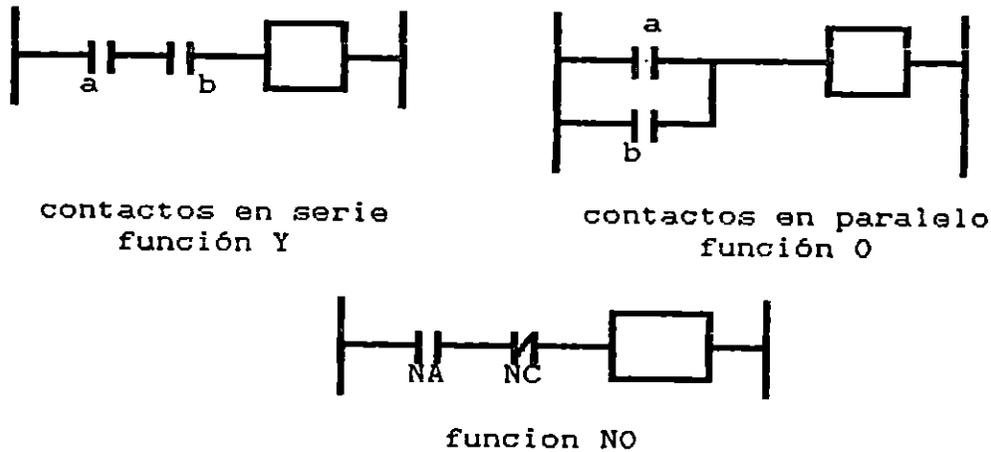


Figura 3.6. representacion del diagrama de escala

Los símbolos de los diagramas de escala son usados para construir cualquier tipo de sistema de control de interruptores lógicos y los diagramas producidos pueden ser tan complejos como sea necesario para una aplicación en particular, la cual debe de ser bien documentada para poder ser mejor interpretada. Esto es sencillo al describir así las expresiones combinatorias como la presentada en la figura 3.7; donde el motor es conectado a una fuente de poder por medio de tres interruptores en serie. El motor se encenderá si todos los interruptores estan cerrados

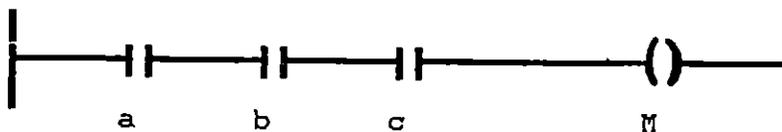


Figura 3.7 simbolo de diagrama de escalera de un circuito de un motor eléctrico

Por lo expuesto se puede resaltar que con este método es bastante sencillo describir así las expresiones combinatorias. Por el contrario, para los problemas secuenciales, es necesario construir una sucesión de circuitos de autoalimentación, lo que resulta difícil de establecer y de leer.

Los esquemas de contactos son normalizados. Porque son muy familiares a los electricistas, se utilizan frecuentemente par expresar o visualizar los automatismos programados. Para ello se emplea su versión Norteamericana, la cual es el diagrama en escala o escalera de mano (ladder diagram) que se puede apreciar en la figura 3.8 en la cual representa otro tipo de función el cual es la función memoria que se puede describir de la siguiente manera: el cierre del contacto ha activado la bobina M que cierra el contacto m asociando garantizando así la autoalimentación de la bobina que mantiene este estado hasta la apertura de b. Este principio de representación es idéntico al de los esquemas de contacto. Solamente cambia la forma de los símbolos elementales, .

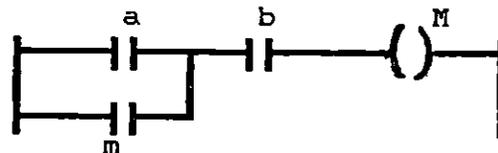
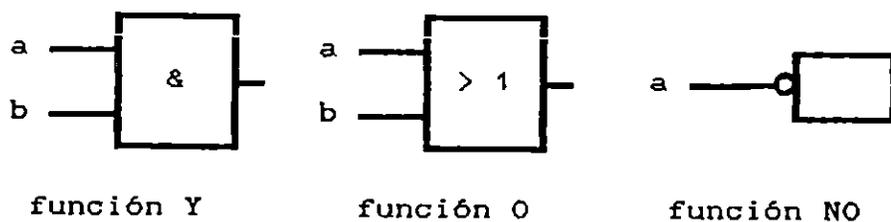


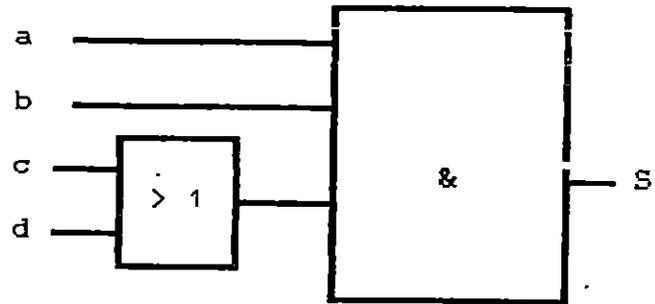
Figura 3.8. Diagrama de escala

3.5.1.2 LOGIGRAMA

Es la representación gráfica de las relaciones lógicas cuyas funciones de base Y, O, NO, MEMORIA, se han representado en la parte anterior. Este tipo de representación está normalizado en un plano internacional y conduce a resultados claros y compactos por medio del agrupamiento gráfico de las funciones de base representada como diagramas de bloque de cada función lógica, ver figura 3.9, Es necesario el hacer la siguiente aclaración, que solamente los técnicos norteamericanos en electrónica utilizan otras variantes de los símbolos en semi-luna, pero no son eficientes en la representación de automatismo.



a)



b)

Figura 3.9.a) representación de funciones lógicas
 b) logigráma de la relación combinatoria
 $S = a.b.(c+d)$

Ambas representaciones, el logigrama y diagramas en escala, están relacionados directamente a los circuitos físicos controlados por el PLC, y por lo tanto son ideales para aplicaciones en las cuales el controlador programable está reemplazando un sistema convencional de relés. El funcionamiento secuencial de los procesos de producción no se puede describir claramente por medio de un esquema de contactos o un logigrama. Esta es la razón por la que se han desarrollado diferentes tipos de representaciones y lenguajes gráficos, que tienen por objetivo esencial una expresión clara de los problemas secuenciales. También es conveniente hacer mención de que para este tipo de representación hacemos uso del concepto y esquema de los Flip-Flop's, aún que también se pueden utilizar para sistemas combinatoriales, lo que observaremos más adelante.

3.5.1.3 FLUJOGRAMAS

Estos están más a menudo relacionados o asociados con programación de computadoras, pero son un método común de representación para operaciones secuenciales de un sistema de control. Los flujogramas están directamente relacionados con las descripciones verbales de la secuencia de control, mostrando cada prueba y acción de los resultados dentro de una serie de símbolos interconectados (figura 3.10). Las acciones en los flujogramas están contenidas en símbolos rectangulares, mientras que las condiciones o pruebas son encerradas en símbolos en forma de diamante. Desafortunadamente, cuando se utiliza para la descripción de grandes sistemas de control, el flujograma puede volverse difícil de interpretar, ya que resulta muy largo y enredado.

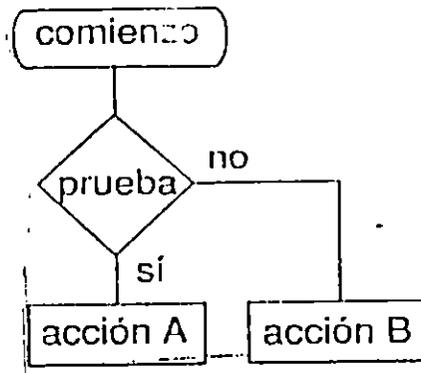


Figura 3.10. Representación de control Lógico por medio de flujograma

3.5.1.3 EL GRAFCET

El cronograma, diagrama de fase, redes de Petri entre otros han sido lenguajes gráficos utilizados en el automatismo. Cada uno de ellos ha ayudado a obtener experiencia para que el GRAFCET constituya la síntesis; el cual en pocos años se ha vuelto una de las más populares formas de representación de tareas secuenciales utilizando una forma condensada de descripción simbólica.

De origen francés, el lenguaje GRAFCET se impone rápidamente en la práctica de los automatismos. Publicada en 1988, la norma internacional IEC 848 (International Electrotechnical Commission) precisa las reglas y amplía su empleo bajo la denominación FUNCTION CHART. Ahora normalizado, el GRAFCET se reconoce como el lenguaje gráfico mejor adaptado a la expresión de la parte secuencial de los automatismos de producción. El GRAFCET es también una valiosa ayuda para prueba y diagnóstico de fallas del sistema automatizado.

Representando la sucesión de las etapas del ciclo. La evolución del ciclo etapa por etapa se controla con una "transición" dispuesta entre cada etapa. A cada una de las etapas puede corresponder una o varias acciones. A cada transicióncorraesponde una "receptividad", condición que debe cumplirse para poder franquear la transición, lo que permite la evolución de una etapa a la siguiente.

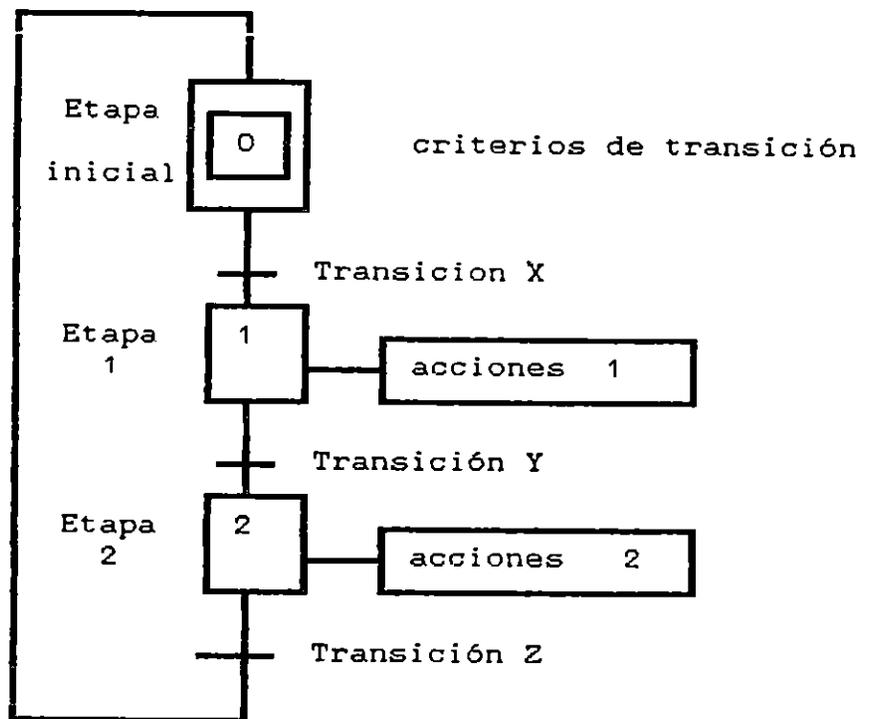


Figura 3.1. Ejemplo de representación GRAFCET.

El ciclo se desarrolla etapa por etapa: La etapa inicial (etapa 0, según se aprecia en la figura 3.11) que se activa al comienzo del funcionamiento, válida la transición que la sigue; la misma se franquea si se cumple la receptividad X. Entonces se activa al etapa 1 y se desactiva la etapa 0. Las acciones asociadas a la etapa 1 se desarrollan hasta que la receptividad Y de la transición que sigue se cumpla, y así sucesivamente hasta llegar a la transición Z la cual después de cumplirse regresa al estado inicial

El ejemplo anterior es conocido como una secuencia lineal, pero también se tienen opciones para escribir las secuencias opcionales o simultáneas. Las secuencias opcionales son aquellas en donde se desarrolla una condición dependiendo de la cual se ramifica el programa en dos ramificaciones, y las simultáneas, tal y como su nombre lo indica ocurren en un mismo tiempo y se indican con un doble trazo en las barras de condiciones. Tal y como si fuera trabajar con flujogramas; recordando siempre que las condiciones representan entradas del sistema y que las acciones asociadas a cada etapa representan salidas del sistema

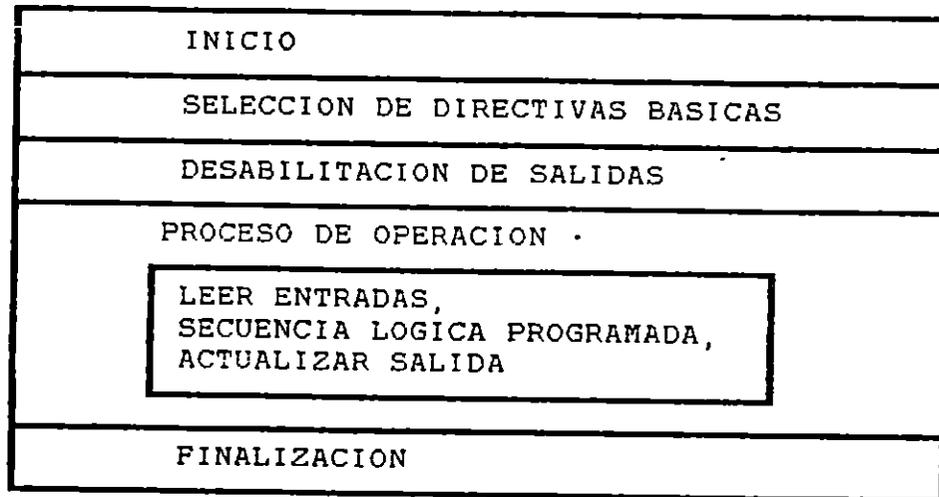
3.6 ESTRUCTURA DE LOS PROGRAMAS EN EL PLC

Todo el contenido que hasta ahora a sido enfocado en este capítulo es, si se quiere, un preámbulo sobre el desarrollo del software y herramientas de diseño de programas en un PLC; ya que esto servirá para el tratamiento que se le dará a los programas que se han de desarrollar en el protipo diseñado y sobre el cual se deben de aclarar lo siguiente:

El hecho de seguir la filosofía de los automatismos programables que han llegado a desarrollar, en base a varios años-hombre, módulos autoprogramados que disponen de un juego de operaciones y de funciones básicas ampliamente extendido, coloca en clara desventaja este diseño, de manera tal que si se pretendía crear un lenguaje especialmente desarrollado con sus propias instrucciones de control las cuales serían traducidas a ensamblador del 280 para su aplicación, en un tiempo tan breve, los resultados difícilmente hubiesen sido funcionales; ya que el hecho de elaborar un procesador de lenguaje que lo interprete el microprocesador, podría ser fácilmente tema para un nuevo trabajo de graduación enfocado solamente a software. Por lo tanto se ha tomado un criterio de simplificación y de funcionalidad; resolviendo en el hecho de utilizar las instrucciones propias del microprocesador para la elaboración de los programas, aprovechando totalmente las posibilidades de este lenguaje, por medio del juego de instrucciones básicas. Por lo tanto se parte de su filosofía y se crean los programas de control específicos siguiendo este diseño. Además se sigue plantea una estructura de programas basada en la de los PLC, la cual a su vez se origina de la estructura general de operación de todos los sistemas de control de proceso. Esto significa que se tienen secciones bien definidas de acuerdo a su operación, y funciones básicas, el cual es un proceso encadenado siguiendo una secuencia rígida que tiene como principal característica el ahorrar tiempo al revisar las condiciones de las entradas y de acuerdo a esto actualizar las salidas.

Basándose en nemónicos, ya que así se hace en los lenguajes ensambladores, se diseña el programa creándolo primero uno que es llamado código fuente, el cual se debe de convertir en un código objeto que es el que ejecuta directamente el micro; pero tomando en cuenta que esta ejecución debe de ahorrar el mayor tiempo posible; siendo esta una exigencia vital, por lo tanto la estructura sugerida debe de ser fielmente seguida, pues su finalidad es el ahorro de tiempo tal y como se indica en la tabla 3.1.

TABLA 3.1 Secciones de un programa de PLC



a) **SELECCION DE DIRECTIVAS BASICAS.** Las directivas son órdenes especialmente dadas por el programador donde se guardan valores dentro de símbolos en la memoria que han de ser invocados constantemente y ayudan para hacer mas rápida y entendible la ejecución del programa; el ensamblador del z80 tiene capacidad para desarrollar 11 directivas, pero se utilizan principalmente dos de ellas, las cuales son:

-ORG, la cual se utiliza para colocar la dirección de memoria donde iniciliza el programa, a la vez que sirve para colocar diferentes segmentos de programa en localidades de memoria diferentes.

-EQU, esta directiva es utilizada para asignar un valor a una etiqueta que será utilizada muchas veces, o para darle claridad al programa.

b) **DESHABILITACION DE SALIDAS.** Esta sección sirve para resetear contactos al arrancar el proceso, para que el sistema inicie sin estados no precisados; puesto que envia señales de deshabilitación por medio del módulo de salida, tambien sirve para limpiar los registros del ensamblador que serán utilizados ya sea para guardar datos o para contadores internos, lo que tambien evita el tener estados no deseados al inicio.

C) PROCESO DE OPERACION. Esta es la parte principal del programas, ya que en ella encierra todos los pasos necesarios para controlar el proceso al cual se ha dirigido, constando a su vez de las siguientes secciones:

i. leer entradas, carga el valor de los bits de entradas en el acumulador o cualquier otro registro,

ii. secuencia lógica programada, dependiendo de el valor guardado en el acumulador y de el programa elaborado se ejecutan los pasos necesarios para el control,

iii. actualizar salida, se envia hacia el módulo de salida TON, los estados actualizados dependiendo de el valor leído por las entradas TON del sistema.

Esto es a grandes rasgos la función que desempeñan cada una de las etapas de la seccion de procesamiento; pero como se está trabajando segun la filosofía de actualizado continuo, esta parte es repetitiva ya que se lee el valor de determinado bit y se opera segun este valor, procediendo luego a leer otro bit y operar y así sucesivamente hasta terminar con los bits que segun el programa se deben de leer y así mandar pulsos hacia los preaccionadores para controlar el proceso; como se puede apreciar esto es repetitivo y manejado por el programador.

Con esta estructura de programas adoptada, se puede crear facilmente software entendible para cada aplicación, lo que nos permite localizar rapidamente fallas y ver los resultados obtenidos en poco tiempo; ya que con el listado del programa y el circuito paso a paso con que consta el sistema es facil simular el sistema trabajando, y en caso de encontrar alguna falla se corrige mas rapidamente, ya que se sabe exactamente en que parte de la sección de procesamiento se encuentra

CONCLUSIONES

- Evaluando el nivel de penetración que han tenido los automatismos programables en El Salvador, es de hacer notar el atraso que se tiene, en términos generales, aunque ya varias empresas los han adoptado; y dependiendo de la casa que los distribuya así son los servicios técnicos que se prestan, pero se basan en términos de dependencia ya que el usuario requiere de sus servicios para cualquier modificación en lo que a programación respecta, lo cual genera mayores beneficios económicos para los distribuidores. Tomando esto en cuenta se pretende suministrar al usuario toda la información disponible para que el modifique sus propios programas, siempre que cuente con el equipo indispensable.
- En la etapa de diseño de algún programa para aplicaciones particulares, sobresale al representación GRAFCET para ordenar y entender la estrategia o enfoque que se le hace al problema, ya que para le automatismo es la mas apropiada debido al planteamiento sintético que este hace, y por estar diseñado específicamente para esta aplicación.
- Una gran mayoría de las aplicaciones de los mini-PLCs se da en el área de transporte de paletas de productos , siendo a la vez una de las aplicaciones más sencillas de lógica combinatorial.
- Las tarjetas de entrada/salida "todo o nada" para el prototipo de autómata programable han sido diseñadas para simplificar el trabajo de tratamiento de las señales tanto de las que entran como de las que salen, logrando con un ingenioso diseño el evitar problemas de aislamiento, rebote y comunicación con la CPU; siendo las señales que recibe como las que emite hacia los preaccionadores, valores flexibles para una mejor adaptación a diferentes usos.
- El tratamiento de los problemas que han de emplear controladores programables, han sido simplificados al utilizar algunas instrucciones del lenguaje ensamblador del Z-80 para programar la memoria EPROM del prototipo, creando un programa sencillo y rápido para cada aplicación en particular; eso si, debiendo guardar una estructura que facilite su elaboración.

BIBLIOGRAFIA

Telemecanique. "LOS AUTOMATISMOS PROGRAMABLES"
Madrid, España. Editions Citef.

Steve Ciarcia. "CONSTRUYA UNA MICROCOMPUTADORA BASADO EN EL
Z80".
Estados Unidos. Byte Books/Mc Graw-Hill.

Rodnay zacks. "PROGRAMING THE Z-80"
Estados Unidos. Ediciones Sybex.

Joseph Nichols. "MICROPROCESADOR Z-80"
Estados Unidos. Ediciones Alfaomega/Marcombo.

CAPITULO IV

ESPECIFICACIONES TECNICAS Y APLICACIONES DEL PLC

INTRODUCCION

En esta sección se proporcionan detalles o especificaciones ya funcionales del sistema del PLC, con el objeto de hacerlo una referencia breve y sencilla donde se especifiquen detalles tanto de su puesta y marcha y funcionamiento físico, como sobre su programación y como se desarrollan los programas para el control, para su uso, algo así, como su nombre lo indica un manual de usuario.

El prototipo de controlador lógico programable diseñado en la Escuela de Ingeniería Eléctrica, se ha fabricado partiendo de los criterios principales que rigen a este tipo de sistemas; pero adaptándolo a nuestras posibilidades reales, lo cual no significa que el sistema no sea funcional, solamente es un prototipo de un sistema básico que es uno de los primeros intentos de trabajar en el área de la automatización, específicamente el la de los PLC.

El objeto de esto es dejar ciertas referencias que simplifiquen su uso para obtener de él un buen rendimiento en cualquier trabajo o aplicación que permita demostrar su capacidad, a la vez que se emplea lo anteriormente expuesto que es el uso de las herramientas de diseño de software, y las de desarrollo o representación de los programas, por medio de representación en logigrama, flujograma o GRAFCET.

4.1 CAMPO DE APLICACION

Tomando como base la exitosa filosofía de diseño de los automatismos programables, se presenta el mini-PLC, que puede integrarse fácilmente en un ambiente de producción para la sustitución de tecnologías cableadas en base a relés, con una pequeña cantidad de entradas-salidas; en tareas donde dichos relés eran empleados para tareas combinacionales o sencillas tareas secuenciales de control; pero siempre tomando en cuenta que existen límites para el control por medio de PLC que todavía es cubierto por relés; sobre todo en situaciones donde se efectúen conmutaciones con corrientes de alto nivel. Por otra parte los relés eléctricos continúan, siendo necesarios alrededor del los autómatas programables.

Tambien sustituyen a las tarjetas electrónicas estándares, en algunas situaciones; debido principalmente a la universalidad que proporciona la programabilidad de estos sistemas, que requiere mucha menor preparación de su ambiente, en contra de la especificidad de aplicaciones de dichas tarjetas, ya que estas se diseñan para solucionar especialmente el problema planteado; pero es frecuente complementar ambas tecnologías para la obtención de mejores resultados.

Evaluando algunos de los criterios técnicos que colocan a el mini-autómata por sobre las tecnologías cableadas a base de relés y tarjetas electrónicas dedicadas; resulta ya una idea clara de su aplicación en el área industrial, resultando un sistema que puede ser utilizado como sustitución de arreglos sencillos que anteriormente involucraran las tecnologías cableadas descritas, en donde se necesiten pocos puntos de entrada y salida; aplicaciones en que puedan desarrollarse por medio de programas pequeños, facilmente entendibles, y aún si se cuenta con el soporte necesario, desarrollados por el usuario. Así que entre las categorías de las aplicaciones se tienen las siguientes:

- líneas de ensamblaje
- paletizadores
- sobreembalaje
- módulos de carga y descarga
- mando sobre transfer-switch
- bandas transportadoras

Siendo las anteriores solamente un ejemplo de aplicaciones, dado con el objeto de proporcionar una idea de su campo de aplicaciones, pero esto es bastante amplio y puede desempeñar variados procesos, dependiendo de la lógica de programación empleada.

4.2 DESCRIPCION

Se ha desarrollado un sistema pequeño, de bajo costo y bastante compacto, diseñado para aplicaciones sistemas industriales controlando y supervisando plantas y procesos que emplean relés; desarrollado en base al microprocesador z80, que proporciona un filosofía de 16 bits que puede manejar 16 puntos de I/O.

Se ha instalado todo el sistema en una caja pequeña y robusta, teniendo en su interior los impresos que han sido separados por módulos para un mejor mantenimiento o detección de errores consistiendo estos módulos de los siguiente:

- fuente de alimentación.
- CPU.
- módulo de la interface de entrada TON.
- módulo de la interface de salida TON.
- módulo de memoria EPROM de 4K (removible).
- módulo de manejo de I/O, con capacidad de expansión I/O.
- presentador del programa "paso a paso" .
- indicador estado de los 16 puntos de I/O.

a) **Fuente de alimentación.** Esta esta diseñada para alimentar al sistema completo y con una capacidad sobrada para futuras ampliaciones, a su entrada el voltaje de alimentación es de 10 Vac, 60 Hz, 5 A. Con el cual se alimenta al sistema proporcionando los voltajes a todas las tarjetas de la siguiente manera, niveles TTL Todos los modulos del sistema y proporcionando 12 Vcc para el módulo de salida, con el cual se alimentan los transistores que envian las señales hacia el exterior.

b) **CPU.** En el CPU del sistema se encuentra el microprocesador Z80 que es el "alma" del sistema, y toda su circuiteria de apoyo, previamente descrita, ademas la memoria RAM que es la encargada de almacenar algunas subrutinas y valores de los puntos de I/O, La frecuencia del reloj que alimenta al microprocesador esta proporcionada por un cristal de cuarzo de 3.8 MHz, que es un valor apropiado para los pequeños autómatas; este módulo esta conectado por medio de buses con los módulos siguientes: manejador de I/O, de memoria EPROM.

c) **Módulo de interface de entrada TON.** El módulo de entrada recibe las señales TON provenientes de los captadores en el exterior y la procesa para ponerla en forma de que puedan ser utilizadas por la CPU, consta de 8 entradas digitales "todo o nada" con un rango de voltaje de entre 12 a 24 Vcc y 60 mA, aislada galvánicamente por medio de opto-acopladores que le dan una protección de aislamiento de de picos de voltaje de hasta 2.5 KV, enviando señales strobe con un periodo de duración de 10 mS lo que las hace óptimas para evitar rebotes y poder ser utilizadas.

d) **Módulo de interface de salida TON.** Estos módulos envian las ordenes hacia los preaccionadores, reciben señales desde el CPU, de una duración en el orden de los micro-segundos, pero por medio de software o de circuiteria digital la transforman en pulsos strobe de entre 0.5 hasta 3.5 segundo, con un rango de entre 6 y 12 Vcc y entre 30 y 50 mA; a la vez que aíslan estas señales hasta 2 KV, teniendo capacidad

de manejo de 8 señales de strobe, para 8 pre-accionadores, sus niveles de alimentación son de 5 y 12 Vcc proporcionados por la fuente del sistema.

e) Memoria EPROM. Esto no es nada más que la memoria EPROM de 4 KB, 2732A, con que cuenta la CPU, que ha sido retirada de dicho módulo, con el objeto de que sea removible externamente para cargar o modificar los programas de control por el usuario, sin necesidad de que se abra toda la caja, todo esto por que el diseño se ha hecho considerando que se trabajará con la lógica de programas en EPROM donde se guardan permanentemente; y están conectados a la CPU de modo que sus conexiones provienen del bus de direcciones y datos de la CPU.

f) Módulo de manejo de I/O. En este impreso es donde se decodifican los puertos de entrada y salida, por medio de arreglos con las señales MREQ, IORQ, RD Y WR, se obtiene dicha decodificación, con la cual se puede tener la capacidad de manejo de hasta 32 puntos de entrada y 32 de salida; pero que en este caso solamente se trabaja con 8 y 8, quedando para posteriores ampliaciones dicha capacidad, teniendo en cuenta que para ello será necesario la incorporación de otros módulos de entradas y salidas; ya que el sistema solamente posee dos con capacidad para 8 bits cada uno.

g) presentador de el programa "paso a paso" Y el indicador de programas "paso a paso" son presentadores alfanuméricos que por medio de una circuitería conectada al bus de direcciones de la CPU, indica los códigos objeto de el programa que se encuentra corriendo el PLC, sólo que a una frecuencia muchísimo mas reducida (de 2 KHz) para que pueda ser claramente visible por cualquier persona, y con el listado del programa con el código hexadecimal, pueda depurar o revisar el programa para posteriores depuraciones o modificaciones; todo esto con el PLC, desconectado del proceso industrial, ya que dicho indicador esta diseñado para eso.

h) indicador de estado de los 16 puntos de I/O . El indicador de estado de puerto no es mas que uno leds que proceden de los modulos de entrada y salida e indican cual de los 8 bits, tanto de entrada como de salida, se encuentra recibiendo o enviando un pulso; lo cual sirve para evaluar el estado actual de el proceso de control de el autómata. estos indicadores para las salidas pueden operar de dos maneras dependiendo del tipo de señal que proviene del módulo de salida, o sea si es por medio de los one-shot tendrá valores fijos de algunas fracciones de segundo, y si es por medio de software puede duran mucho mas.

4.3 PROGRAMACION DE LA MEMORIA EPROM

Una de las principales características de cualquier autómata programable es su capacidad de almacenamiento de programas; una de las alternativas que se tienen industrialmente es la de utilizar memoria volátil RAM ya que es mas barata y sencilla de utilizar en un sistema donde se tiene desarrollado toda una lógica de apoyo tanto para programación como para depuración de los programas; pero en este caso es una limitante ya que como se mencionó anteriormente no se cuenta con un software capaz de todo eso ya para el desarrollo de este sistema se ha partido de cero, por ser nula la información que en medio se tiene sobre la filosofía de diseño tanto de hardware como de software; habiendo consumido mucho tiempo al inicio, solamente dedicado a la recabación de información especialmente de el extranjero, para poder obtener un sistema funcional como el desarrollado ha sido necesaria la inversión de mucho tiempo en detalles que parecen simples en los sistemas ya construidos; pero en el caso presente ello a consumido una gran cantidad de tiempo y esfuerzo.

En vista de lo anterior el hecho de desarrollar un sistema a base de memoria RAM y que pudiera desempeñar las funciones que se desarrollan en un PLC comercial, es en si, un proyecto aparte propio de otro tema de trabajo de graduación. Debiendo de limitarnos a la utilización de memoria EPROM para el almacenamiento de los programas, los cuales han de ser depurados previamente antes de quemar la memoria; aunque esto parezca muy sencillo es una alternativa funcional y viable que es tomada en cuenta en no pocos, mini-PLCs industriales, en donde se carece de facilidades de programación internas, (tal es el caso de los mini autómatas Mitsubishi y General Electric), en donde los programas almacenados son de propósito específico, careciendo de un software monitor residente en el PLC, y donde las alteración no son frecuentes y cualquier alteración ha de hacerse vaciando el programa ya corregido en otra posición, o si ya esta llena la memoria pues será necesario borrar dicha memoria, o bien utilizar otra; pero como en nuestro caso que tenemos una memoria de 4K, se dispone de bastante memoria para guardar programas.

Debido tambien a que estos por la misma filosofía que se ha tomado, que siguiendo una estructura ya establecida, los programas de control específicos resultan bastante cortos, sencillos y claros; pudiendo guardarse bastantes programas de aplicaciones específicas, y como es externamente removible no se tiene problemas para su manejo.

4.3.1 UNIDAD DE PROGRAMACION DEL PLC

Uno de los principales requerimientos de los lenguajes de programación es que deben de ser fácilmente entendibles y utilizables en situaciones de control. Por lo tanto se necesita el soporte de un lenguaje de alto nivel que proporcione facilidades de programación y de depuración para que los usuarios lo utilicen, sin la necesidad de consumir bastante tiempo profundizando en aprender comando complejos.

Por lo tanto para hacer mucho mas competitivo el autómeta, siguiendo lo anteriormente expuesto, es que nos hemos decidido por la utilización de un sistema MDS, que fué descrito anteriormente y del cual se cuenta en la EIE, que nos proporciona excelentes herramientas para la programación de los programas en la memoria EPROM del sistema y su depuración siendo muy confiable al momento de quemar al memoria, pues se ha simulado y comprobado la eficacia del sistema sin necesidad de desperdiciar memoria, todo ello se ha desarrollado en la computadora.

Si bien esta es una ventaja para nuestra situación, crea cierta situación de dependencian por parte de algún hipotético usuario, pero es de tomar en cuenta que los niveles de automatización con que se cuenta en el país, este procedimiento es el comunmente empleado, donde los usuarios solamente exponen sus necesidades y la empresa distribuidora es la encargada de todo lo demas, desde retirar la EPROM del PLC, programarla, hasta ir a colocar nuevamente la memoria.

4.3.2 LENGUAJE DE PROGRAMACION UTILIZADO

Las instrucciones lógicas de los programas de los PLCs deben de ser cortas, sencillas y pocas para que los usuarios no se les dificulte su operación. Esto es básico y partiendo de ello es que vemos las posibilidades que como tal ofrecen el lenguaje ensamblador del z80, donde sus nemónicos representes acciones que han de ser ejecutadas por el controlador.

Cada instrucción esta constituida por dos partes : un opcode o nemónico de operación, y un componente denominado operando que identifica a un elemento den particular (por ejemplo un punto de salida) que esta interconectado con el PLC; por ejemplo:

opcode	operando
OUT	(POTO),A

En esta instrucción no indica que el contenido de el acumulador es enviado hacia el puerto de salida 0, o sea hacia los preaccionadores, como se ve es una operación bastante sencilla y fundamental ya que es muy común en un proceso de el autómata. principalmente se desarrollan las operaciones de lectura de puertos, comparación, prueba de bits y escritura en puertos, las cuales son instrucciones sencillas, donde se utilizan pocos nemónicos, lo cual es otra ventaja pues se obtienen programas cortos y sencillos, que en la mayoría de los casos no toman mas de 150 líneas de programación; realizando esto a partir de los siguiente nemónicos del z80 presentados en la tabla 4.1

TABLA 4.1,

a), Abreviaturas utilizadas en la descripción de las instrucciones del Z80

SIMBOLO	SIGNIFICADO
Acumulador	Registro A
POUTn	puerto de salida "n"
PINn	puerto de entrada "n"
r,r'	Uno de los registros A,B,C,D,E,H o
n	Una expresión de 1byte en 1 gama de 0 a 255 inclusive.
(HL)	Especifica el contenido de memoria en la posición direccionada por el contenido del par de registro HL.
SP	Puntero de pila
C	bandera de acarreo
N	Bandera de suma/resta
Z	Bandera de cero
S	Bandera de Signo
P/V	Paridad, desbordamiento
d	Una expresión de 1 byte entre -128 127

(b) Listado de las instrucciones mas empleadas para el autómata programable.

INSTRUCCION	DESCRIPCION
AND s	ejecuta operación lógica AND con el operando s(s puede ser r,n,(HL))
BIT b,r	prueba el bit b del registro r
CALL cc,qq	llamado de subrutina con condiciónl
CALL pq	llamado de subrutina a localidad pq
CP s	compara el operando s con el acumulador
DEC m	decrementa operando m
IN A,(n)	carga en el acumulador el contenido de puerto periférico n
JP cc,pq	salto con condición a localidad pq
JP pq	salto a localidad pq
JR cc,e	salto si condición a localidad e
JR e	salto relativo a e
LD r,r'	carga el contenido de r' en r
LD r,n	carga el registro r con el dato n
LD dd,nn	carga registro par dd con el dato nn
NOP	no operación
OUT (n),A	Envía el contenido del acumulador al puerto periférico n
RET	retorno de subrutina
RETN	retorno de interrupción no enmascarable
XORs	Or exclusiva con el operando s

4.3.3. PROCEDIMIENTO DE PROGRAMACION

A continuación se han detallado los pasos a seguir para la programación de la memoria, partiendo desde el desarrollo del programa en papel hasta quemar la memoria, aunque esto se detalla de manera escueta, en los anexos se proporcionan todos los pasos necesarios para desarrollar lo anterior.

1. En primer lugar el proceso es descrito en un algoritmo, donde se han de detallar todas las posibilidades con las que han de contarse, el cual ha de ser lo suficientemente completo, identificando de una vez todas las entradas y salidas del sistema.
2. La descripción realizada por medio del algoritmo es trasladada o cualquier método de representación gráfico de los requerimientos la lógica de control, tal como flujograma o GRAFCET, dependiendo del tipo con el que este mas familiarizado el usuario.
3. De la representación anterior resulta mas facil el trasladarlo a formato de esquema de contactos, de donde se procede a reconocer los captadores y principalmente los preaccionadores y accionadores del sistema.
4. Utilizando tambien la representación en flujograma o GRAFCET, se desarrolla el programa ahora ya en nemónicos del ensamblador del z-80.
5. A partir del programa escrito, que tambien se conoce como programa fuente, en papel se genera un archivo en ASCII puro, por medio de el editor de texto SIDEKICK.
6. Se crea el programa objeto, a partir de el programa fuente, en nemónicos, esto es por medio del archivo XASM80. El cual tambien nos indicará si existen errores que deben de ser corregidos volviendo al programa fuente.
7. El próximo paso es convertir este programa objeto en un archivo con formato hexadecimal de intel, con el archivo XLINK.
8. Se crea un archivo binario a partir del archivo en formato hexadecimal, por medio del archivo HEXBIN, y siempre manteniendo el formato intel.
9. Con la memoria EPROM ya colocada en la interface, se procede a entrar en el program EPPGM4, y cargar en el buffer el programa en código binario, especificando la dirección de inicio y finalización del buffer y luego se procede a quemar el programa en la memoria.

10. Se procede a instalar la memoria en el controlador programable y se realizan una última revisión conectando el circuito paso a paso, corroborando el desarrollo del listado del programa con lo que indican los presentadores.

4.4 EJEMPLO DE APLICACION DEL PLC

Entre las múltiples aplicaciones de los automatismos programables, se ha seleccionado uno sencillo y claro que puede ser fácilmente entendido, y es el de una línea de panificación industrial, que como la mayoría de unidades de producción alimentarias es modular, para responder a las necesidades de flexibilidad y de evolución, (ver figura 1.1) el cual es desarrollado siguiendo los pasos anteriormente descritos; empezando por una descripción del sistema;

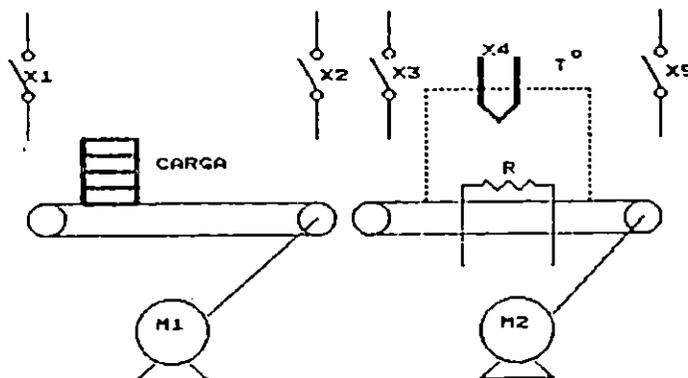


Figura 1.1 Ejemplo de aplicación

El producto es cargado en el extremo inicial de la banda transportadora, en donde existe un primer captador, el cual detecta que el producto está sobre la banda 1, entonces la señal que envía hacia el módulo de entrada inicia el proceso de control, procediendo el autómata a enviar una señal hacia el preaccionador que hace arrancar el motor 1, enviando el producto hasta el final de la banda 1, en donde existe un segundo captador, de fin de carrera, que envía otra señal al PLC, ordenando este, que apaga el motor 1,

simultaneamente arranca el motor 2, que mueve la segunda banda transportadora, por lo que el producto pasa de una banda a otra; posteriormente el producto llega al tercer captador del autómata, procediendo este a detener el segundo motor, ya que el producto ha llegado al horno para proceder a su cocción. Se calienta este hasta determinada temperatura, donde un cuarto captador, de temperatura, indica al autómata que dicha temperatura ha sido alcanzada, ordenado que se apague el calentador, esperando un determinado tiempo para que se enfríe y procede a reactivar el motor 2, con lo el producto ya cocido sigue su trayecto, hasta llegar al último captador, que es un indicador de final de carrera; cuya señal al llegar a la CPU del sistema ejecuta las últimas instrucciones que ordenan detener el motor 2 definitivamente, esperando que se retire la carga, y luego al llegar otro producto se inicia de nuevo el proceso.

Con esta descripción definimos las entradas y salidas del sistema, quedando distribuido de la siguiente manera:

4 captadores

- captador de posición 1
- captador de fin de carrera 1
- captador de posición 2
- captador de temperatura máxima
- captador de fin de carrera 2

4 preaccionadores

- arranque de motor 1
- parada de motor 1
- arranque de motor 2
- activar/desactivar calentador
- parada de motor 2

Sobre la elección de estos captadores y accionadores es una cuestión que practicamente se sale de los alcances del trabajo, por lo que se harán simulaciones de todo este proceso, con relés, contactores y opto-acopladores para simular el proceso.

El proceso se describe gráficamente mediante flujograma y GRAFCET, lo cual nos dá una idea mucho mas clara para la elaboración del programa en ensamblador del Z-80, tambien se elabora el diagrama de contactos que nos indica como irán conectados los preaccionadores para activar/desactivar los motores y el calentador.

conectados los preaccionadores para activar/desactivar los motores y el calentador.

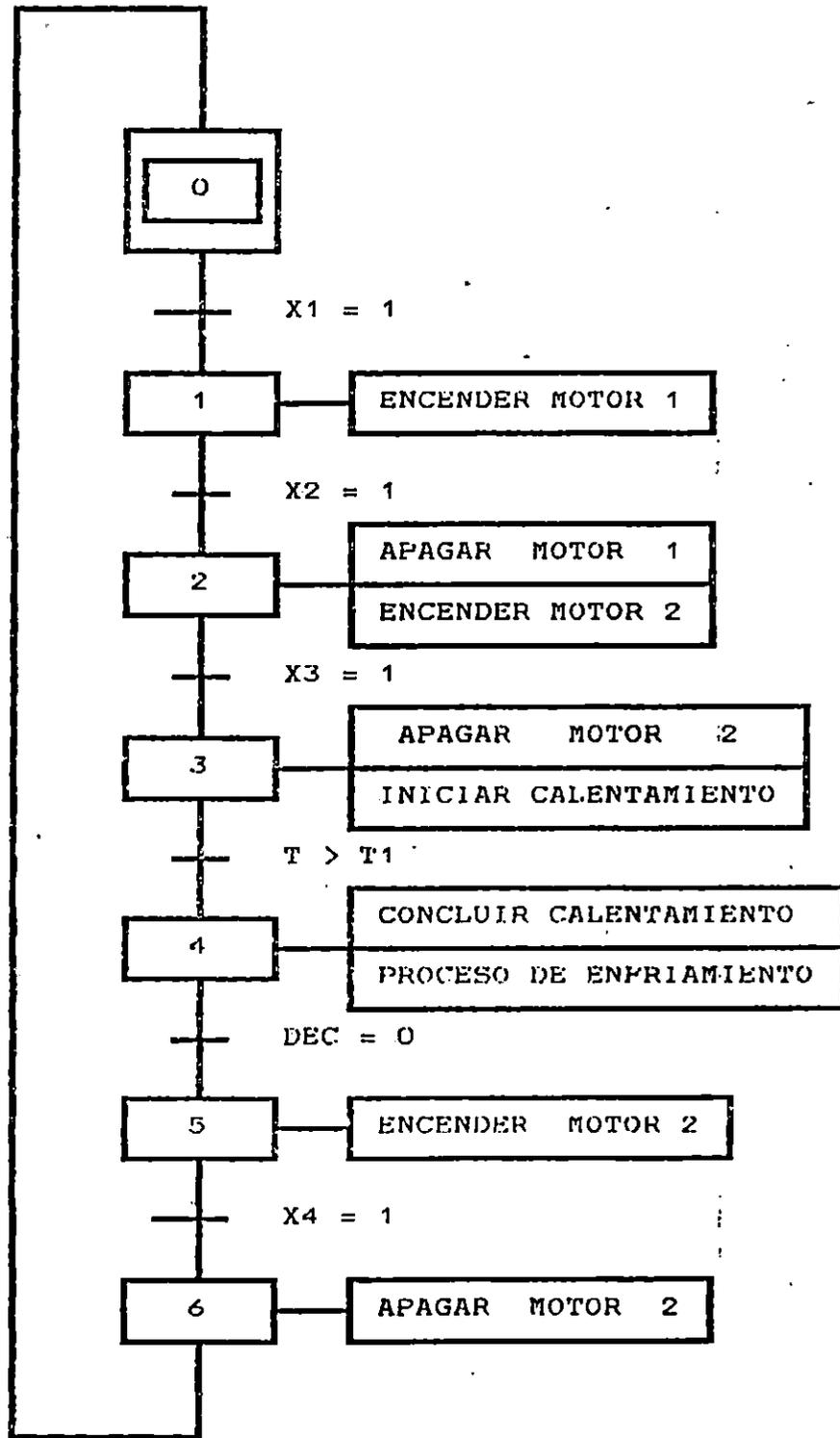


Figura 1.2 Grafcet del ejemplo

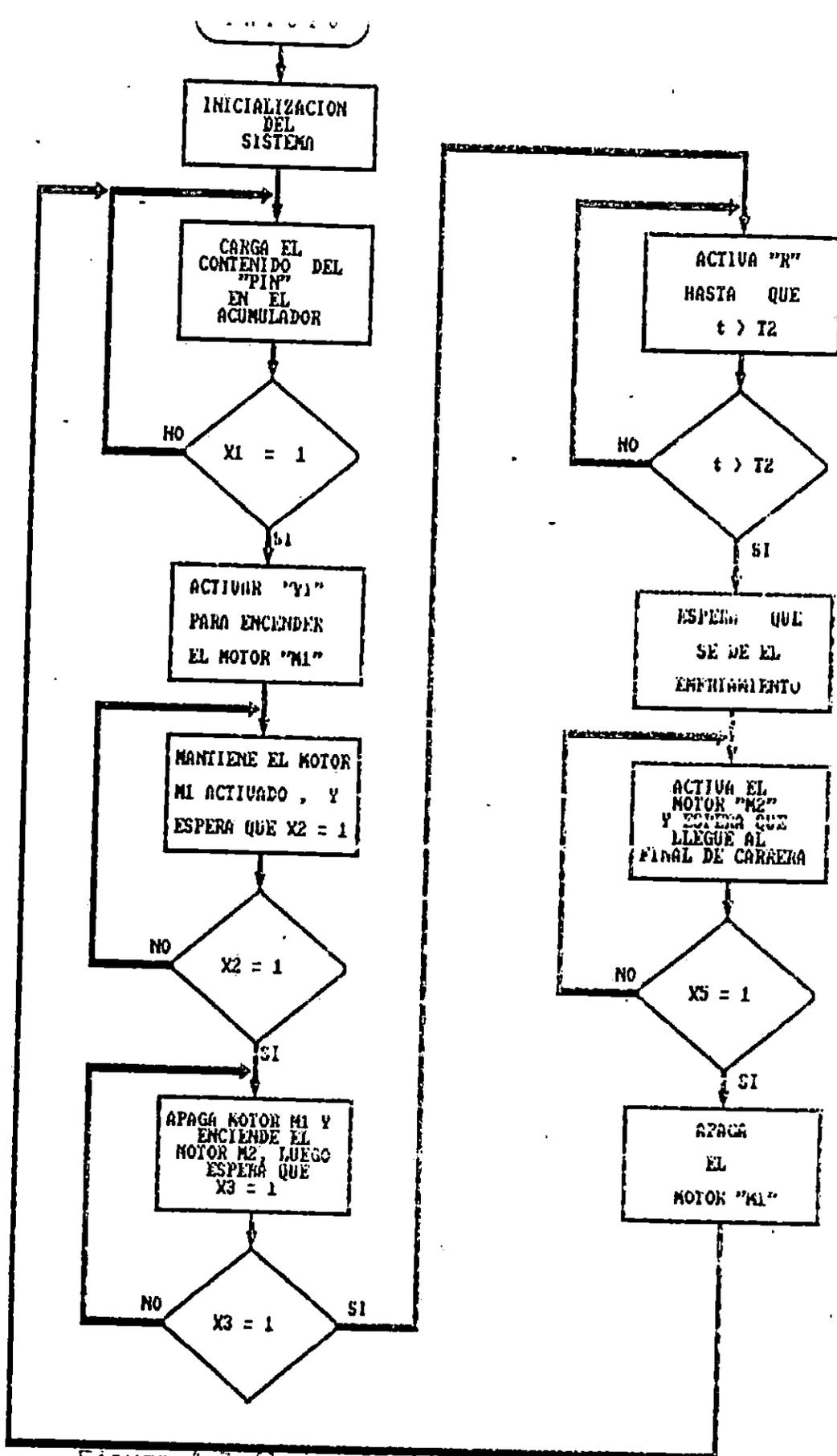


Figura 4.3 flujograma de la tarea de control

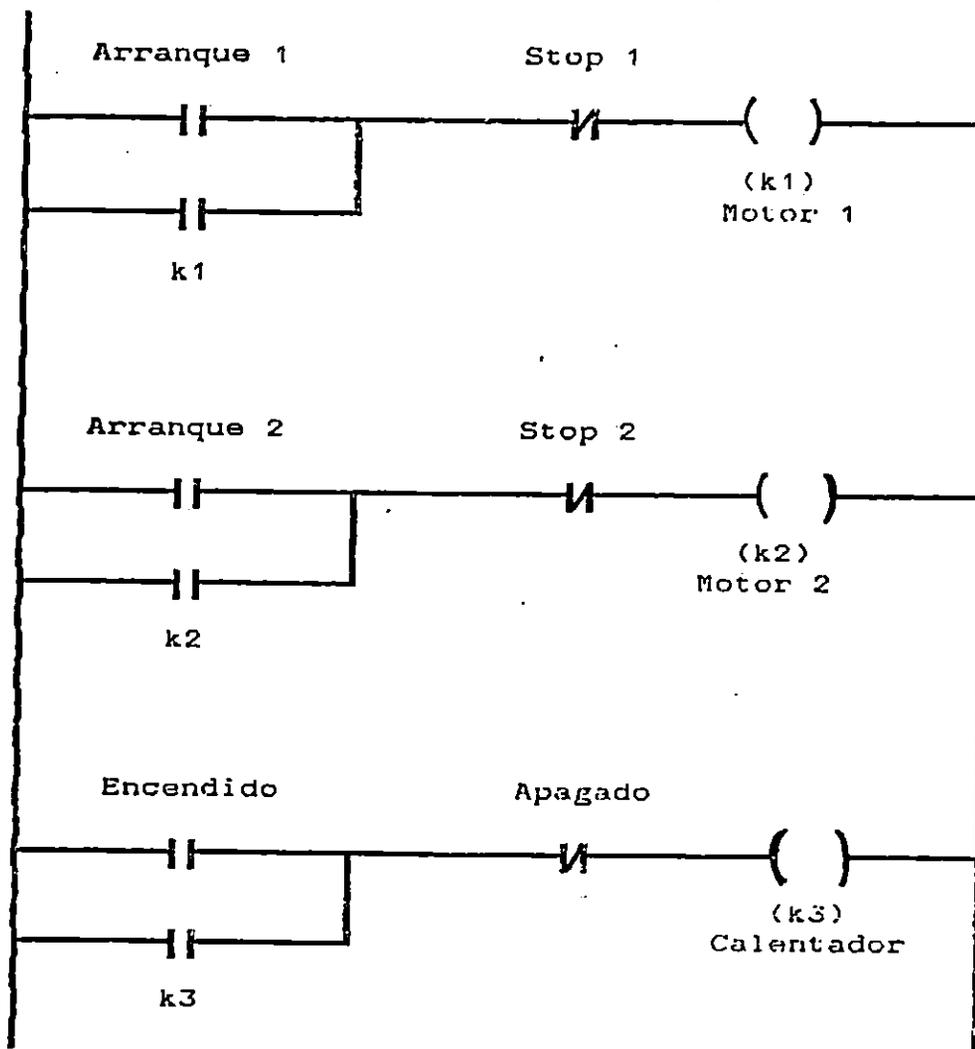


Figura 4.4 Esquema de contactos utilizados

Input Filename : TRES.asm
Output Filename : TRES.obj

```

1 0100
2      0000          FOUT EQU      0
3      0000          PIN  EQU      0
4      0001          Y1  EQU      00000001B
5      0002          Y2  EQU      00000010B
6      0004          Y3  EQU      00000100B
7      0008          Y4  EQU      00001000B
8      0010          Y5  EQU      00010000B
9      0020          Y6  EQU      00100000B
10     0040          Y7  EQU      01000000B
11     0080          Y8  EQU      10000000B
12
13
14     ; ***** RUTINA DE CONTROL DEL PLC *****
15
16
17 0100 31 00 10    ACT:  LD  SP,1000H    ;MANDA EL SP A LA DIRECCION 1000 DE RAM
18 0103 AF          XOR  A                    ;LIMPIA EL REGISTRO A
19 0104 D3 00      OUT  (P0UT),A            ;LIMPIA EL PUERTO DE SALIDA
20 0106 06 00      LD  B,0H
21 0108 0E 00      LD  C,0H
22 010A 16 00      LD  D,0H                ;LIMPIA LOS REGISTROS PARA EL RETARDO
23
24
25 010C 0B 00      ;
26 010E E6 01      INICIO: IN  A,(PIN)
27 0110 CB 47      AND  00000001B
28 0112 2B FB      BIT  0,A
29 0114 00         JR   Z,INICIO           ;EVALUA SI SE INICIA LA SECUENCIA DE CONTROL
30 0115 3E 40      NOP                                     ;SI ES ASI QUE COMIENZE EL PROCESO
31 0117 D3 00      LD  A,Y7
32 0118 D3 00      OUT  (P0UT),A            ;PONE EN MARCHA EL MOTOR 1
33 011D 00         LD  A,0H
34 011E DB 00      OUT  (P0UT),A
35 0120 E6 02      AND  0000010B
36 0122 CB 4F      BIT  1,A
37 0124 2B FB      JR   Z,POLL
38 0126 00         NOP                                     ;EVALUA EL ESTADO DEL INTERRUPTOR 2
39 0127 3E 80      LD  A,Y8
40 0129 D3 00      OUT  (P0UT),A            ;SI ES 1, ENTONCES ACTIVA EL SEGUNDO MOTOR
41 012B 3E 00      LD  A,0H
42 012D D3 00      OUT  (P0UT),A
43 012F 00         NOP

```

44	0130	DB 00	STOP:	IN	A, (PIN)	
45	0132	E6 04		AND	00000100B	
46	0134	CB 57		BIT	2, A	
47	0136	2B FB		JR	2, STOP	; EVALUA SI EL INTERRUPTOR 3 ES 1
48	0138	00		NOP		
49	0139	3E 20		LD	A, Y6	
50	013B	D3 00		OUT	(P0UT), A	; SI ES 1, QUE APAGUE AMBOS MOTORES
51	013D	3E 00		LD	A, 0H	
52	013F	D3 00		OUT	(P0UT), A	
53	0141	00		NOP		
54	0142	16 02		LD	D, 2H	; EL CALENTAMIENTO SE HACE DURANTE UN TIEMPO
55	0144	3E 02	LAZO:	LD	A, Y2	; DETERMINADO DE ENCENDIDO/APAGADO
56	0146	D3 00		OUT	(P0UT), A	; ACTIVA EL PROCESO DE CALENTAMIENTO
57	0148	DB 00	CALENT:	IN	A, (PIN)	
58	014A	E6 08		AND	00001000B	; LIMPIA LOS BITS NO UTILIZADOS
59	014C	CB 5F		BIT	3, A	; HASTA QUE ALCANZE LA TEMPERATURA MAXIMA
60	014E	2B FB		JR	2, CALENT	
61	0150	3E 00		LD	A, 0H	
62	0152	D3 00		OUT	(P0UT), A	; DESACTIVA EL CALENTADOR
63	0154	00		NOP		
64	0155	0E A0		LD	C, A0H	
65	0157	06 FF	DEL1:	LD	D, FFH	
66	0159	05	DEL2:	DEC	B	
67	015A	20 FD		JR	NZ, DEL2	
68	015C	0D		DEC	C	
69	015D	20 FB		JR	NZ, DEL1	
70	015F	00		NOP		; SE ENFRIA EL PRODUCTO
71	0160	00		NOP		
72	0161	15		DEC	D	
73	0162	20 E0		JR	NZ, LAZO	; EJECUTA EL CALENTAMIENTO Y ENFRIAMIENTO 3 VECES
74						
75	0164	3E 80		LD	A, Y8	
76	0166	D3 00		OUT	(P0UT), A	; VUELVE A ACTIVAR EL MOTOR 2
77	0168	3E 00		LD	A, 0H	
78	016A	D3 00		OUT	(P0UT), A	
79	016C	00		NOP		
80	016D	DB 00	FIN:	IN	A, (PIN)	
81	016F	E6 10		AND	00010000B	
82	0171	CB 67		BIT	4, A	
83	0173	2B FB		JR	2, FIN	; EVALUA ESTADO DEL INTERRUPTOR DE FINAL DE
84	0175	3E 20		LD	A, Y6	; CARRERA DEL MOTOR 2
85	0177	D3 00		OUT	(P0UT), A	
86	0179	3E 00		LD	A, 0H	
87	017B	D3 00		OUT	(P0UT), A	; SI ES 1, APAGA EL SEGUNDO MOTOR
88	017D	00		NOP		
89	017E	00		NOP		
90	017F	C3 0C 01		JF	INICIO	

Figura 4.5. Listado del programa en ensamblador Z80

CONCLUSIONES

- Si bien el hecho de utilizar una memoria EPROM, pareciera limitante del sistema, en realidad su función es similar a la realizada por una memoria RAM, ya que en los sistemas que la utilizan, los programas guardados en ella son permanentes debido a la incorporación de una pila de respaldo; y por ser programas pequeños los utilizados se puede utilizar la memoria, direccionándola en las localidades que se encuentren limpias.

- El hecho de utilizar un desarrollo de software ayudado por computadora (MDS), que ya se encuentra instalado en la EIE; proporciona una gran versatilidad al sistema, ya que los programas pueden ser depurados, utilizando todas las herramientas que proporciona la computadora, otorgándole una alta fiabilidad a los programas desarrollados, aún antes de su instalación.

- El ejemplo que en este capítulo se presenta es para tener una noción más clara de cómo utilizar las diferentes herramientas de diseño de software para PLC, tales como el flujograma, esquema de contactos y el GRAFCET, que en nuestro país no es muy conocido y en el extranjero es el principal soporte para el diseño de programas de control de lógica combinatorial.

- El hecho de utilizar pocas instrucciones del ensamblador del 280 presenta una ventaja para los usuarios que conocen poco sobre lenguaje ensamblador; ya que no es necesario que se compenetre en aprender sobre todas las instrucciones disponibles del microprocesador, ya que para nuestros fines solamente son algunos comandos los necesarios.

BIBLIOGRAFIA

Rodnay Zacks, "PROGRAMMING THE Z-80",
Third edition, Sybex copyright, 1980

Steve ciarcia, "CONSTRUYA UNA MICROCOMPUTADORA BASADO EN EL Z80".
Estados Unidos, Byte books/Mc Graw-Hill.

Joseph Nichols, "MICROPROCESADOR Z-80".
Estados Unidos, Ediciones Alfaomega/Marcombo

Simatic S5
"AUTOMATIZACION CON EL SIMATIC S5-101U"
Geratehadbuch
Siemens aktiengesellschaft.

Dora Deysi Melendez Castro
Nelson Ovidio Castaneda Vidal
"APLICACIONES DE LOS MICROPROCESADORES A SISTEMAS DE CONTROL"
Tesis optar al grado de ingeniero electricista
Biblioteca de la Escuela de Ingenieria Eléctrica.
UES, 1987.

CONCLUSIONES GENERALES

El controlador programable diseñado se presenta como una alternativa sencilla y barata de control; que si bien es cierto, parte de los lineamientos básicos de los mini-PLC, presente variantes fundamentales, debido principalmente a que se ha debido de adaptar detalles técnicos a las condiciones que se encuentran en la EIE y en el mercado nacional; pero aún con eso, el sistema obtenido resulta económico y confiable.

El sistema diseñado es en sí el "cerebro" que rige y ordena las tareas de automatización, el controlador programable se limita a recibir señales de los captadores, y en base al programa almacenado en memoria, envía las señales pertinentes hacia los preaccionadores; que son los que influyen en el proceso directamente; por lo que es necesario que saber diferenciar entre estos elementos, y su papel en el control de procesos.

Los programas que se desarrollan en este autómata, poseen una estructura principal; pero es necesario que desarrolle completamente en nemónicos de acuerdo a la función deseada, por lo que se ha ideado el empleo de las instrucciones necesarias, para que sea más fácilmente entendibles y realizables, por un usuario hipotético.

El estudio de los PLC no forma parte de los programas de las asignaturas de la carrera, además la Escuela no cuenta con la documentación sobre el funcionamiento, características y aplicaciones de estos; por lo que el presente documento y el circuito diseñado pretende proveer tanto a docentes como a estudiantes la información básica necesaria.

RECOMENDACIONES

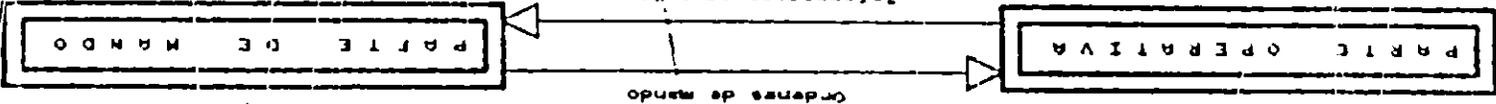
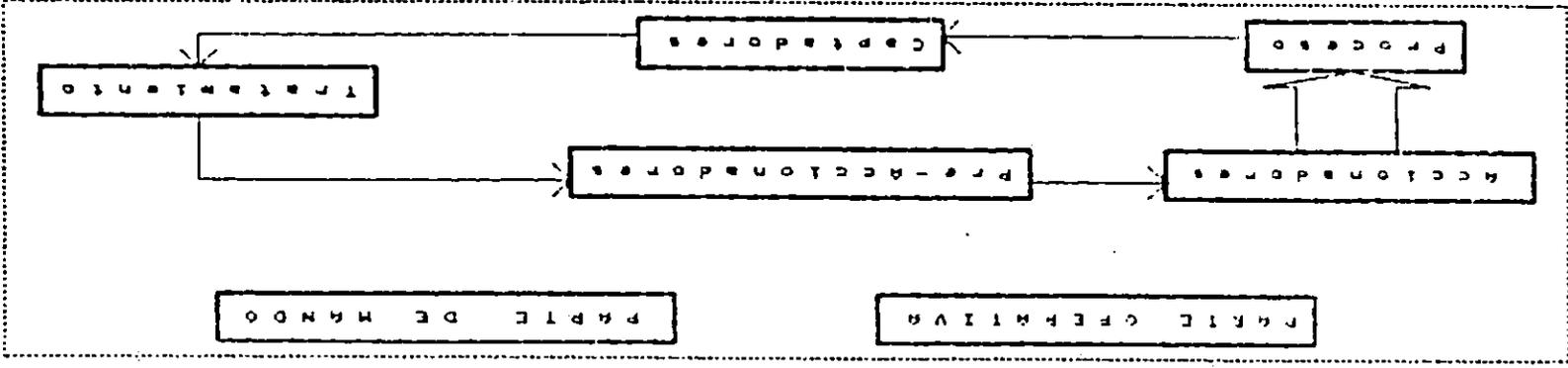
Se recomienda que la EIE estudie la posibilidad de poner en funcionamiento el autómata SIMATIC, que posee, integrándole los módulos necesarios para ello, o que adquiriera uno de los pequeños módulos didácticos que distribuye la empresa Telemecanique, con el objeto de cubrir esa área que hasta ahora no se estudia.

Se recomienda que la Escuela trabaje en la recopilación bibliográfica sobre la información concerniente a automatismos programables, y se actualize sobre los avances tecnológicos en este campo, no solo en la relacionado al hardware, sino también al software.

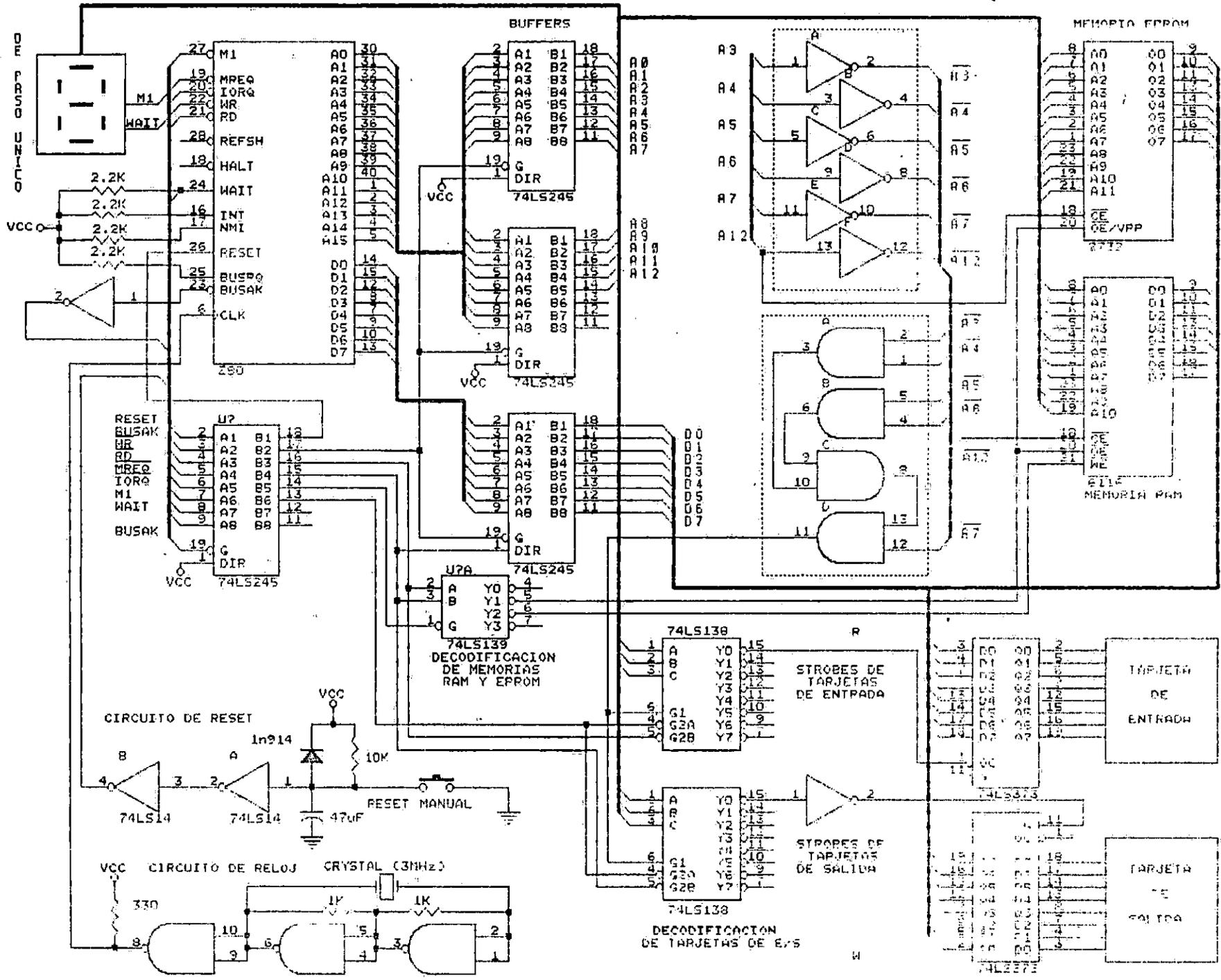
Se recomienda el incorporar el estudio de los PLC en el plan de estudios de la carrera, incluyéndolo en la materia relacionada o bien desarrollarla como técnica electiva, no sólo como parte teórica, sino también con prácticas de laboratorio; ya sea con el SIMATIC ó con el prototipo diseñado.

ANEXO A
diagramas de los circuitos diseñados

ESTRUCTURA DE LOS SISTEMAS AUTOMATIZADOS



DE CIRCUITO
 DE CARRILLO
 DE CARRILLO
 DE CARRILLO



HARDWARE COMPLETO DEL SISTEMA " MINI PLC "

ANEXO B

ELABORACION DE PROGRAMAS Y QUEMADO DE MEMORIAS

2.4. CONVENCIONES DEL ASSEMBLER Z80.

Una característica de éste Assembler es que permite letras mayúsculas y minúsculas indistintamente.

2.4.1. Delimitadores: entre los campos se utilizan los siguiente delimitadores:

- 1) Dos puntos (':') después de un rótulo, excepto para las pseudo-operaciones EQU, que requieren un espacio.
- 2) Un espacio después del codop.
- 3) Una coma entre los operandos en el campo de operandos.
- 4) Un punto y coma(';') antes de un comentario.
- 5) Paréntesis alrededor de las referencias de memoria.

Instrucciones típicas del lenguaje ensamblador son:

```
START:  LD      A,(100)      ;OBTIENE LONGITUD
        ADD    HL, DE
        HALT
```

2.4.2. Rótulos: el assembler permite seis caracteres en los rótulos, el primero debe ser una letra y los siguiente pueden ser letras, números o el guión bajo ('_'). Se utilizan para guardar un valor determinado en tiempo de edición y de ensamble, también son conocidos como identificadores o símbolos.

2.4.3. Nombres reservados: son los nombres de los registros (A, B, C, D, E, H, L, I, R, AF, BC, DE, HL, AF', BC', DE', HL'), de los índices (IX, IY) y las banderas (C, NC, Z, ZN, M, P, PE, PO). Ya que son reservados no se pueden utilizar para definir rótulos.

2.4.4. Pseudo-Operaciones: las mas ocupadas por nosotros serán las siguientes: EQU, DEFL, DEFB, DEFW, BYTE, WORD, BLKB, BLKW y END..

EQU: es usada para asignarle valores a algún identificador (rótulo). Hay que hacer notar que esta definición sólo tiene valor cuando se está editando el programa y en tiempo de ensamble, luego su existencia llega a su fin.

Ejemplo:

```
DIRECC EQU 40 ;le asigna a DIRECC = 40
```

DEFL: posee la misma función que EQU, con la diferencia que el identificador que definen pueden tomar otro valor en otra parte del programa. Es decir, el valor del identificador puede ser redefinido, algo que no permite EQU.

Ejemplo:

```
DIRREC DEFL 04
... ;cuando se use DIRECC, este equivaldra a 04
...
DIRECC DEFL 05 ; a partir de acá DIRECC = 05
```

DEFB: esta es una directiva para localizar un byte en RAM. También se utiliza para colocar cadenas de caracteres.
Ejemplo:

```
INICIO: DEFB 8 ;coloca el valor 8 en la posición
; de memoria INICIO

ERROR: DEFB 'Existe error' ; coloca en memoria el mensaje
;en ASCII a partir de la
;posición de memoria ERROR.
```

DEFW: esta es una directiva para localizar una palabra (2 bytes) en RAM. Coloca el byte LSB en la posición de memoria más baja y el MSB en la siguiente posición de memoria.

BYTE: se utiliza para definir una tabla de bytes en memoria.
Ejemplo:

```
DAT01: BYTE 1,2,3,4,5 ;define la tabla de números dados a
;partir de posición DAT01 en memoria.
```

WORD: se utiliza para definir una tabla de palabras en memoria.

BLKB: reserva un area de memoria definida en bytes.

Ejemplo:

```
BYTE1: BLKB 5 ;el micro reserva 5 bytes en memoria
;a partir de BYTE1.
```

BLKW: reserva un area de memoria definida en palabras.

END: le indica al ensamblador que esta es la última línea a ensamblar, lo demás será ignorado.

Ejemplo de Pseudo-operaciones:

Memor.	Valor asignado	Pseudo-operación
	0010	DAT01 EQU 10H
	0015	DAT02: DEFL 15H
	0020	DAT02: DEFL 20H
0000	20	BYTE1: DEFB 20H
0001	40	DEFB 40H

0002	CDAB	WORD1:	DEFW 0ABCDH
0004	2301		DEFW 0123H
0006	10 20 30	TABLA1:	BYTE 10H, 20H, 30H
0009	40 50 60		BYTE 40H, 50H, 60H
000C	2301 CDAB	TABLA2:	WORD 0123H, 0ABCDH
0010		BLKB 2
0012		BLKW 2

TABLA DE SIMBOLOS.

Nombre del Símbolo	Valor en Hexadecimal
BYTE1	0000
DAT01	= 0010
DAT02	= 0020
TABLA1	0006
TABLA2	000C
WORD1	0002

2.4.5. Direcciones: Este ensamblador permite entradas en el campo de direcciones en cualquier de las siguiente formas:

- 1) Decimal (el caso por defecto).
Ejemplo: 1247
- 2) Hexadecimal (debe comenzar con un dígito y terminar con cero).
Ejemplo: 142CH, 0E7H
- 3) Octal (debe finalizar con Q).
Ejemplo: 1247Q
- 4) Binario (debe finalizar con B).
Ejemplo: .10101010B
- 5) ASCII (encerrado en apóstrofes).
Ejemplo: 'AQUI'
- 6) Como un desplazamiento del contador de programa.
Ejemplo: \$+12H, 34H-\$.
El signo '\$' es el equivalente al valor del contador en ese instante.

Además se permiten operandos aritméticos en el campo de dirección, la tabla de estos se muestra a continuación:

OPERATOR	FUNCTION	PRIORITY
+	UNARY PLUS	1
-	UNARY MINUS	1
.NOT. or \	LOGICAL NOT	1
.RES.	RESULT	1
**	EXPONENTIATION	2
*	MULTIPLICATION	3
/	DIVISION	3
.MOD.	MODULO	3
.SHR.	LOGICAL SHIFT RIGHT	3
.SHL.	LOGICAL SHIFT LEFT	3
+	ADDITION	4
-	SUBTRACTION	4
.AND. or &	LOGICAL AND	5
.OR. or	LOGICAL OR	6

3. EDICION DE PROGRAMAS FUENTE.

La creación de programas en Asembler Z80 requiere de generar un archivo en ASCII puro, es decir, que no introducirá caracteres de control³, los cuales confundirían al Ensamblador.

Para nuestro caso utilizaremos el editor de texto SIDEKICK el cual presenta muchas ventajas, una de ellas es que se mantiene residente en memoria junto con el archivo que se está editando, de tal forma que no es necesario cargar el editor y el archivo de trabajo cada vez que se realiza una corrección en éste último.

El software básico de éste esta constituido por el archivo ejecutable SK.COM. Para ejecutarlo se tecléa lo siguiente:

```
A:\>SK <←
```

El computador carga el editor de texto en memoria y regresa el control al DOS (aparece el prompt nuevamente). Para entrar al editor de texto tecléamos (!!de dónde sea que estemos!!) la secuencia <shift izquierda><shift derecha> simultáneamente o <ctrl izquierda><alt izquierda>; al hacer esto, el microprocesador conmutará la tarea que se encuentre realizando y pasará el control al SIDEKICK.

Lo primero que observaremos es un menú, del cual nos interesará la segunda opción F2 Notepad; presionamos F2 y luego ESC una vez, con lo cual nos pedirá el nombre de un archivo. Introducimos un nombre (con extensión .ASM) y en este punto puede suceder dos cosas: si el archivo no existe creará un archivo en el disco con dicho nombre y si ya existe cargará el contenido de tal archivo para su inmediata edición.

El proceso de carga del archivo a editar únicamente sucederá cuando ejecutemos esta operación la primera vez, la segunda vez el contenido del archivo ya no se cargará porque se encuentra residente en memoria de la vez anterior.

Para salirnos del SIDEKICK únicamente presionamos la tecla ESC, y el microprocesador continuará con la tarea que se encontraba realizando anteriormente.

Para grabar el archivo editado se utiliza la función F2, y este se graba automáticamente con el nombre que se le dio al inicio. Para cargar y editar un nuevo archivo, se hace con la tecla F3, e introduciendo el nombre del archivo.

Otros comandos podrán ser encontrados en el ANEXO II de esta guía, consúltelos cuando sea necesario.

^{3/} Caracteres de control son los que utiliza el computador para comunicarse con los periféricos, y lo forman los caracteres ASCII del 1 hasta el 30.

4. ENSAMBLE Y LINKEO.

Una vez terminado la edición del código fuente (en mnemónicos) se realiza la tarea de ensamble y por último el linkeo.

ENSAMBLE:

El programa ensamblador se llama XASMB0.EXE, y debe estar contenido en el directorio a partir del cual ensamblaremos los archivos de código fuente. Para ensamblar un archivo denominado Z80-1.ASM ubicado en el drive A, se ejecuta lo siguiente (se supone que nos encontramos en el drive A y que el ensamblador se encuentra en dicho drive): la letra pequeña es la forma en que responde el ensamblador:

```
A:\XASMB0<_
```

El computador presenta lo siguiente:

```
      Z80 Macro Ensamblador - Version 4.00g  
      Copyright (C) 1985 by 2500 A.D. Software, Inc.
```

Listing Destination (N, T, D, E, L, P, <CR> = N) :

El computador nos pide hacia donde queremos que direccionemos el archivo .LST, que es el listado del código fuente con su traducción a hexadecimal y con la tabla de símbolos (rótulos). Si tecleamos N, no genera ningún listado; al teclear T nos presenta el listado en pantalla a medida que se va generando y la opción D -la que más utilizaremos- graba este archivo en el disco A con el nombre que tiene el código fuente pero con una extensión .LST.

Cuando digitamos la opción D el computador nos responde con lo siguiente:

Generate Cross Reference ? (Y/N <CR> = No) : Y

Debemos de contestar Y y luego ENTER, caso contrario no generará el listado. A continuación nos pide el nombre del archivo de entrada -programa fuente- y el nombre del archivo de salida -programa objeto-, como se ve a continuación:

```
Input Filename : Z80-1
```

```
Output Filename : Z80-1
```

Se introduce el nombre de nuestro archivo, si posee extensión .ASM, no es necesario ponerla como en el ejemplo. El archivo de salida se le ha puesto el mismo nombre, pero el ensamblador le coloca la extensión .OBJ cuando no se especifica lo contrario. El Ensamblador responde con:

```
Lines Assembled : 23
```

```
Assembly Errors : 0
```

Si el programa no posee errores -indicada en la última línea-, procedemos al paso siguiente, caso contrario el Ensamblador nos dirá que hay error y nos dará un indicador que nos dirá en que lugar está el error y la posible causa de éste; esta información la tenemos que buscar en el archivo de listado y luego de saber los errores cometidos regresamos al editor y lo corregimos. Recuerda, si esta usando SIDEKICK solo presione <shift><shift> y luego F2.

Se observa la utilidad de generar el listado en un archivo .LST, ya que podemos imprimirlo y tener a mano los errores hallados.

LINKEADO:

El próximo paso a realizar es convertir el programa objeto en un archivo con formato Hexadecimal Intel, para que pueda ser usado por el simulador.

Para ello, se requiere del archivo XLINK.EXE en nuestro disco de trabajo. El primer paso es ejecutarlo:

```
A:\>XLINK<_
```

El micro responderá de la siguiente forma:

```
2500 A.D. Linker Copyright (C) 1985 - Version 4.00i
```

```
Input Filename : Z80-1<_
```

```
Enter Offset For 'CODE' : 40<_
```

```
Input Filename : <_
```

```
Output Filename : Z80-1<_
```

```
Options (D, S, A, N, X, H, E, T, 1, 2, 3, (CR) = Default) : D<_
```

En el anterior desarrollo lo que se encuentra en **negrita** es la respuesta del usuario. La primera pregunta que le hace el computador al usuario es el nombre del archivo de entrada, el usuario entra un nombre con una extensión -la extensión que el linkeador toma por defecto es .OBJ-. En el caso presentado el linkeador busca un archivo llamado Z80-1.OBJ.

A continuación, se le pregunta sobre el desplazamiento del código en lenguaje máquina a partir de la dirección cero. En el caso presentado, el programa se colocará a partir de la posición 40H memoria RAM; hay que hacer notar que este valor es tomado como el hexadecimal 4016.

Posteriormente, el linkeador nos pregunta por otro archivo de entrada, lo cual es útil cuando hemos ensamblado subrutinas independientes unas de otras. Si existe otro archivo, se introduce el nombre de éste y la posición de memoria a partir

del cual se introducirá. Si no existe otro archivo, como en el caso presentado, únicamente presionamos ENTER.

Hay que tener especial cuidado cuando son varios programas objetos los que linkearán, ya que estos se pueden traslapar el final de uno con el comienzo del otro, lo cual redundaría en un resultado erróneo. !!Cuidado!! , el linkeador no muestra que haya error si lo anterior sucede.

Después de digitar todos los archivos de entradas y sus desplazamientos, se nos pide el nombre del archivo de salida. La última sentencia, Options, se refiere al tipo de formato que tendrá el archivo de salida, los cuales están de acuerdo a la siguiente tabla:

<u>Opción</u>	<u>Formato</u>	<u>Extensión</u>
D	Intel Hex	.HEX
S	Intel Hex	.HEX
A	Intel Hex	.HEX
M	Corre con la directiva SYMBOLS	
X	Ejecutable	.TSK
H	Intel Hex	.HEX
E	Extended Intel Hex	.HEX
T	Tektronix Hex	.TEK
1	Motorola S19	.S19
2	Motorola S28	.S28
3	Motorola S37	.S37

Estos diferentes formatos del archivo de salida corresponden a ciertas normas que tiene cada fabricante para sus dispositivos, por ejemplo si deseamos programar una EPROM Intel, el formato del archivo que se vaciará a través del programador debe tener el formato Intel Hex.

En nuestro caso, el formato que nos interesará es el Intel Hex, que es el que acepta el simulador, para ello utilizaremos exclusivamente la opción D.

En este punto ya se han introducido todos los datos requeridos por el linkeador, y el computador nos presenta lo siguiente para el caso específico:

```

#####
#                               L O A D   M A P                               #
#####
# Section Name      Starting Address  Ending Address  Size  #
#####
# z80-1.obj
# CODE              0040              0055          0016 #
#####

```

Link Errors : 0

Output Format : Intel Hex

La primera parte es el mapa de carga, donde nos detalla la dirección de inicio y final en memoria de cada código correspondiente a los módulos de entrada, agregando además el tamaño de cada módulo.

Para el caso específico, el código ocupa de la dirección 0040₁₆ a la 0055₁₆ y tiene un tamaño de 16₁₆.

Hay que observar, que si existe un traslape de un módulo con otro, en este punto se determinará y hay que ejecutar el linkeo proceso evitando que se cometa el mismo error.

En este punto hemos obtenido un archivo en lenguaje máquina que puede ser ejecutado en forma directa por el microprocesador Z80, es entonces que surge la pregunta ¿En dónde correremos el programa, sino tenemos un sistema basado en el Z80? Y la respuesta es en una AT LEMON basada en el Z86 - también en sistemas con microprocesador 8088 y 8086- con un software que simula al microprocesador Z80, y que por ello se llama SIMULADOR.

5. SIMULACION

Como se dijo anteriormente, el simulador para el Z80 es un software que nos sirve como herramienta para ejecutar y depurar programas objetos para sistemas diseñados en base al Z80.

El simulador está constituido por un único archivo, el SIMBO.EXE. La forma de ejecutarlo es digitando desde el sistema operativo lo siguiente:

```
A:\>SIMBO<_
```

Luego estaremos en el ambiente del simulador, para una mejor explicación de sus características, comandos y utilidad remitase al ANEXO I de esta guía.

Resumen: hasta este momento, han sido detalladas cada una de las herramientas que se utilizarán para la programación del microprocesador Z80. Los archivos que se han de tener en el disco de trabajo son los siguientes:

Archivo	Tamaño
SIMBO	EXE 60501
XASMBO	EXE 122621
XLINK	EXE 141317
SK	COM 39520

El tamaño global de los 4 archivos suma aproximadamente 364 Kbytes, por lo cual se requiere de un disco de alta densidad para trabajar y poder tener los datos y programas en un sólo disco.

ANEXO I I

COMANDOS BASICO DEL SIDEKICK

CURSOR MOVEMENTS:	COMMAND	KEY
Character left	Ctrl-S	<-
Character right	Ctrl-D	->
Word left	Ctrl-A	Ctrl-(-
Word right	Ctrl-F	Ctrl--)
Line up	Ctrl-E	^
Line down	Ctrl-I	
Scroll up	Ctrl-W	
Scroll down	Ctrl-Z	
Page up	Ctrl-R	PgUp
Page down	Ctrl-C	PgDn
CURSOR MOVEMENTS:		
To left on line	Ctrl-Q-S	None
To right on line	Ctrl-Q-D	End
To top of page	Ctrl-Q-E	Ctrl-Home
To bottom of page	Ctrl-Q-X	Ctrl-End
To top of file	Ctrl-Q-R	Ctrl-PgUp
To end of file	Ctrl-Q-C	Ctrl-PgDn
To end of file & insert date/time	Ctrl-Q-D	
To beginning of block	Ctrl-Q-B	
To end of block	Ctrl-Q-K	
To last cursor position	Ctrl-Q-P	
INSERT & DELETE		
Insert mode on/off	Ctrl-V	Ins
Insert line	Ctrl-II	
Delete line	Ctrl-Y	
Delete to end of line	Ctrl-Q-Y	
Delete right word	Ctrl-T	
Delete char under cursor	Ctrl-B	Del
Delete left character	Ctrl-H	<--
BLOCK COMMANDS		
Mark block begin	Ctrl-K-B	F7
Mark block end	Ctrl-K-K	F8
Mark single word	Ctrl-K-T	
Hide/display block	Ctrl-K-H	
Copy block	Ctrl-K-C	
Move block	Ctrl-K-V	
Delete block	Ctrl-K-Y	
Read block from disk	Ctrl-K-R	
Write block to disk	Ctrl-K-W	
Sort block	Ctrl-K-S	
MISC. COMMANDS		
Save note file	Ctrl-K-D	F2
Paste block	Ctrl-K-E	
Cut and paste		F4

Paste block lets you paste a block of text from the Notepad to any other program.

Cut and paste lets you cut data from the screen and paste it into the Notepad.

MISC. COMMANDS	COMMAND	KEY
Tab		Ctrl-I
Repeat last find		Ctrl-L
Control character prefix		Ctrl-P
Re-format line		Ctrl-B
Set right margin		Ctrl-Q-R
Find		Ctrl-Q-F
Find & replace		Ctrl-Q-A
Auto tab on/off		Ctrl-Q-I
Restore line		Ctrl-Q-L
Read date and time		Ctrl-Q-T

When off, you may edit text produced by editors using the 8th bit of the characters (like Word-Star). When on, you can see the entire 256-character IBM PC character set, including the semi-graphics.

The right margin is set to column 65 by default (unless changed during installation). You may change this value with the Ctrl-Q-R command.

When you reach the right margin, a line break will automatically be inserted, and the last word is moved to the new line if it does not fit within the margins.

The current right margin is saved permanently when text. A paragraph of text ends with a blank line. For example, this is the first paragraph on this page.

This is the second paragraph. The first paragraph ends in line 4 because of the blank line following it.

This is still the second paragraph. An indent does not start a new paragraph.

You can disable word wrap by setting the right margin to 250.

FUNCTION KEYS

F1 Help. The help key will display detailed help about Notepad. Depending on which features you are using you will get different help texts.

F2 Save. Saves the contents of your note file on disk. In order to let you switch disks this is never done automatically, and you must therefore remember to save for instance before you shut down the computer. You may also use the WordStar command **Ctrl-F 2** to save.

F3 New note file. Defines the name of a file to be used as note file. The default file name is **NOTES** (unless changed in Setup).

F4 Cut and paste. The screen as it were before Notepad was opened will appear. You may then mark a block of text on the screen with the normal block markers and paste it into the Notepad with the **Ctrl-K-C** command.

F9 Expand window. Pressed once, this key causes the arrow keys to move the borders of the Notepad window outwards, expanding its size. When pressed again, **F9** returns the arrows to their normal use.

F10 Contract window. Pressed once, this key causes the arrow keys to move the borders of the Notepad window inwards, contracting its size. When pressed again, **F10** returns the arrows to their normal use.

Marcaado de bloques:

Hold down the **Ctrl**-key and type **K** and **B** to mark the block beginning (upper left corner). Use the **^** and **->** keys to move the cursor to the end of the block, the lower right corner. The block is marked on the screen as the cursor moves along.

3.0 CREACION DEL FIRMWARE

3.1 MENU PRINCIPAL

En este paso, el archivo binario debe estar presente en el mismo directorio donde reside el software de la interfase. Para programar la EPROM en este momento se debe utilizar el programa EPPGM4.EXE. Como su extensión lo indica, este programa es ejecutable directamente desde el MS-DOS y para correrlo no hace falta más que digitar en el prompt del sistema operativo:

```
EPPGM4 <ENTER>
```

Inmediatamente después de darle entrada, se presenta el menú principal del programa de la interfase.

Básicamente, al desplegar el menú principal, la pantalla se divide en dos partes: la mitad izquierda, que contiene las opciones que podemos elegir. La mitad derecha muestra una ventana en su parte superior que muestra un resumen de informaciones relacionados con los parámetros de creación del firmware. Estos parámetros serán descritos más adelante.

En la parte inferior izquierda de la pantalla, un mensaje nos indica que debemos digitar el número o la letra que identifica la opción que deseamos se ejecute.

La descripción de las opciones del menú principal son las siguientes:

- 1) DIR: Esta opción es idéntica al DIR del MS-DOS, y para poder ejecutarla es necesario que el archivo COMMAND.COM del sistema operativo esté copiado en el mismo disco que el EPPGM4.EXE.

- 2) **LOAD BIN FILE TO MEMORY BUFFER:** Esta opción cargará el archivo binario creado mediante HEXDIN.EXE en el buffer de la memoria RAM de la computadora, la cual sirve de interfase (o buffer) entre el disco en que se está trabajando y la EPROM. Se observará una nota bajo la opción QUIT del menú que advierte la cantidad de RAM destinada a servir como buffer (128 Kb). Se llamará solamente "buffer" en el resto de este manual a la porción de RAM destinada para alojar el programa binario que maneja la interfase.

- 4) **EDIT MEMORY BUFFER:** Esta opción sirve para modificar el contenido del archivo binario alojado en el buffer una vez haya sido cargado en RAM. Para esto se utiliza el archivo DEBUG.COM del sistema operativo y todos los comandos relacionados con él.

SAVE MEMORY BUFFER TO DISK: Esta opción permite guardar en el disco el contenido del buffer que haya sido modificado mediante la opción anterior o lo que se halla leído de la EPROM. Al ejecutarla aparece una nueva ventana en donde se pregunta el nombre con el cual se almacenará en forma de archivo binario, así como las direcciones inicial y final del buffer que se quiere almacenar.

- 5) **TYPE SELECT 27/07/28 SERIES:** Mediante esta opción es posible comunicar a la interfase el tipo de dispositivo que se desea programar, así, la interfase seleccionará el voltaje y la velocidad de trabajo en el chip, así como su pin-out. Al seleccionar esta opción aparece una nueva ventana de trabajo, en la cual se tiene un sub-menú de selección para todos los tipos de

EPROM soportados por la interfase programadora . Estos tipos aparecen ennumerados desde 0 hasta E (hexadecimal) . Los dos números que aparecen en el menú corresponden a las dos últimas cifras del número de modelo del dispositivo. Por default, la interfase sostiene la opción 0 de dicho sub-menú.

MODIFY TARGET ZONE: Esta opción permite seleccionar la dirección de comienzo y finalización del buffer en donde se alojó el archivo binario. Todas las direcciones están en hexadecimal y se recomienda introducir como dirección de comienzo del buffer la dirección 00000. En la parte inferior izquierda del menú principal está señalada esta dirección generalmente como 029AB de la RAM. La dirección final se recomienda que sea la dirección inicial del buffer más la longitud en bytes del archivo binario a cargar, si es que solo se pretende programar un archivo por vez.

También en esta opción se selecciona la dirección de comienzo de "quemar" en la EPROM, esta dirección está en base del dispositivo, o sea que si se desea empezar a programar desde el principio, se debe digitar 00000. La dirección de finalización del programa en la EPROM será la suma de la dirección inicial más la longitud en bytes dimensionada en el buffer.

COMPARE AND DISPLAY MASTER GANG: Mediante esta función se puede comparar los bytes cargados en el buffer y los bytes contenidos en la EPROM de acuerdo a como se hayan configurado las direcciones en la opción anterior.

La interfase muestra un reporte llamado MASTER GANG y en él se muestra la diferencia entre las direcciones del buffer y las del

dispositivo, existen cuatro de ellos, dependiendo en el receptáculo de los cuatro existentes se encuentra instalada la EPROM a comparar, cada MASTER GANG está numerado de acuerdo al número de receptáculo.

Existen tres tipos de comparación: normal, par e impar.

En la comparación par se comparan las direcciones pares del buffer con las direcciones consecutivas de la EPROM, así, si el buffer se hace comenzar en 00000 y el dispositivo en 00000, se comparará la dirección 00000 con la 00000, la 00002 del buffer con la 00001 de la EPROM, la 00004 del buffer con la 00002 de la EPROM, etc.

En la comparación impar se comparan las direcciones impares del buffer con las direcciones consecutivas de la EPROM, así, si el buffer comienza en 00000 y el dispositivo en 00000, se comparará la dirección 00001 del buffer con la 00000 de la EPROM, la 00003 del buffer con la 00001 de la EPROM, la 00005 del buffer con la 00002 de la EPROM, etc.

En la comparación normal se comparan las direcciones consecutivas del buffer con las direcciones consecutivas de la EPROM, así, si el buffer comienza en 00000 y el dispositivo en 00000, se comparará la dirección 00000 con la 00000, la 00001 con la 00001, la 00002 con la 00002, etc.

Si la interfase programadora encuentra diferencias entre el contenido del buffer y el contenido de la EPROM, en comparación normal, el Master Gang aparecerá en la pantalla de la siguiente manera:

Error at :

Press <ESC> to terminate display of

Memoria *buffer*
 00013:BB-(00013:A7) 00019:00-(00019:02)

Press any key to back to main menu

El significado del Master Gang es el siguiente:

Los primeros cinco dígitos representan la dirección del dispositivo que se está programando, en el ejemplo 00013; la cual contiene el byte BB, mientras que, dentro de los paréntesis se encuentran los dígitos 00013 que resulta ser a la dirección del buffer que corresponde a la dirección 00013 de la EPROM. La interfase detecta que en esta dirección del buffer se encuentra almacenado el byte A7 proveniente del programa binario cargado anteriormente. De igual manera, en la dirección 00019 del dispositivo se encuentra el byte 00 que no concuerda con el byte 02 de la dirección 00019 del buffer.

Esta comparación se recomienda hacerla antes y después de programar la EPROM para chequear los lugares de memoria a grabar o utilizados. Así se puede comprobar si la EPROM está virgen (todas sus localizaciones de memoria con FF) o si la transferencia de datos se realizó correctamente (no presentación del Master Gang).

READ MASTER GANG: Con esta opción se carga en el buffer, con las direcciones de inicio y final previamente definidas, el contenido de la EPROM del receptáculo señalado por el MASTER GANG. Así es posible "quemar" con el contenido de una ROM fuente a otras

5) CHANGE I/O BASE-ADR OF HARDWARE: La interfase programadora para una tarjeta con componentes electrónicos, la cual debe estar colocada en uno de los compartimentos de I/O de la computadora, cada uno de estos puertos I/O es identificado por la máquina con una dirección. Se debe asegurar que la computadora "encuentre" la tarjeta de la interfase, y esto se logra mediante esta opción del menú. En esta opción se presenta un menú con varias direcciones y un mensaje de diagnóstico acerca de la tarjeta. Debe activarse una de las opciones del sub-menú que indique que la tarjeta ha sido reconocida, lo cual se mostrará con el mensaje: "card exists".

DISPLAY LOADED FILE HISTORY: Mediante esta opción se puede verificar la carga de los archivos binarios en el buffer actual en su dirección de inicio y fin. Se mostrará en pantalla un listado con las direcciones del buffer y los archivos que se han cargado. CHANGE VPP: Mediante esta opción se puede cambiar el voltaje de polarización de la EPROM que se utiliza sin cambiar su tipo. El voltaje que se usa se presenta en la parte superior derecha de la pantalla y maneja los voltajes de 25, 21, 12.5 y 12.9 voltios.

CHANGE MASTER GANG: Mediante esta opción se puede escoger el receptor para lectura y comparación en donde se introduce la EPROM. Existen 4 opciones que corresponden a los cuatro receptáculos disponibles. El reporte de errores (en MASTER GANG

estará de acuerdo al número de receptáculo escogido. Esto se podrá visualizar en la pantalla en la parte superior derecha.

BLANK CHECK: Con esta opción se verifica el estado de la EPROM, mostrando si la zona escogida para programación esta "en blanco". Esta zona se establece con las direcciones inicial y final del buffer, las que deben coincidir con la del dispositivo. Se mostrará un mensaje en pantalla indicando si está en "blanco" (con OK) o si hay error, mostrando a la vez su dirección. Esta función la realiza para los cuatro receptáculos.

VERIFY: En esta función se procede a verificar si concuerdan los datos programados con los datos en el buffer de acuerdo a las direcciones especificadas en la opción (Z.). En caso de no coincidir se enviará un mensaje de error.

PROGRAM: Mediante esta opción se procede a la "quemada" de la EPROM. Un contador especifica las direcciones que se van programando. Es necesario verificar que las direcciones y carga del archivo esten correctas ya que este paso es irreversible.

Se pueden programar hasta 4 EPROMs simultáneamente con el mismo programa ya que la señal enviada es igual para todos los receptáculos.

AUTO: En esta opción se ejecutan las funciones de BLANK CHECK, PROGRAM y VERIFY automáticamente y se deben tomar las precauciones especificadas anteriormente.