

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

**“DISEÑO DEL SISTEMA AUTOMATICO PARA LA
DISTRIBUCION DE CAJAS UTILIZANDO EL MICRO-PLC
S7-200 Y EL GRAFICADOR DE PROCESOS INTOUCH”**

TOPICO DE GRADUACION

PREVIA A LA OBTENCION DEL TITULO DE:

INGENIERO EN ELECTRICIDAD

ESPECIALIZACION: INDUSTRIAL

PRESENTADA POR:

JULIO ERNESTO GOMEZ ASSAN

WILMER ABELARDO NARANJO ROSALES

GUAYAQUIL – ECUADOR

1.999

AGRADECIMIENTO

A Dios por darnos la fuerza
espiritual para seguir adelante y
permitirnos ser cada día mejores.

DEDICATORIA

A nuestros padres que con su educación
y esfuerzo, nos han sabido guiar por el
camino de la vida

DECLARACION EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestos en este proyecto, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”.

(Reglamento de Exámenes y Títulos Profesionales de la ESPOL).

Sr. Julio Gómez Assán

Sr. Wilmer Naranjo Rosales

Ing. Armando Altamirano
Presidente del Tribunal

Ing. Alberto Larco G.
Director del Tópico

Ing. Nelson Layedra
Miembro del Jurado

Ing. Norman Chootong
Miembro del Jurado

RESUMEN

En este trabajo, proporcionamos los conceptos e ideas básicas más importantes de los robots industriales, para que se tenga una idea de lo complejo que resulta diseñar sistemas robóticos. Estos sistemas son el resultado de la correcta relación de los sistemas mecánico, eléctrico, electrónico, y de control. Se ha puesto especial énfasis en los sistemas electrónico y de control.

Se ha diseñado el sistema de control que servirá para manejar y monitorear un sistema mecánico, el cual podría ser utilizado para realizar la distribución o el ordenamiento de cajas que serán colocadas en un estante. El sistema controlará dos motores de paso y un motor trifásico, que representan los motores utilizados en el proceso.

Se creó el circuito de control para un motor de paso, que servirá para producir la secuencia correcta de pulsos, los cuales serán aplicados a los transistores, éstos manejarán directamente las bobinas del motor. Se realizaron varias pruebas para observar el comportamiento del motor de paso.

Siguiendo el mismo modelo de control, también se maneja a un pequeño motor de corriente alterna trifásico; para esto realizamos el estudio de los inversores trifásicos. Se hicieron pruebas al motor trifásico variando la frecuencia de operación.

Para la programación se ha recurrido al software Step 7-Micro/Win 2.0, el cual permitirá programar al Micro-PLC S7-200 CPU 212 de Siemens, el cual es el controlador programable lógico utilizado.

El movimiento del sistema será simulado y visualizado en el monitor de una computadora, utilizando el Graficador de Procesos InTouch 5.6b de Wonderware versión Evaluación; este software industrial servirá para controlar y monitorear cada movimiento del sistema mecánico.

INDICE GENERAL

	Pág.
RESUMEN	VI
INDICE GENERAL	VIII
INDICE DE ABREVIATURAS	XVI
INDICE DE FIGURAS	XIX
INDICE DE TABLAS	XXIV
LISTA DE SIMBOLOS	XXV
INTRODUCCION	27
I. ROBOTS INDUSTRIALES	28
1.1. Introducción	28
1.2. Concepto De Robot Industrial	29
1.2.1. Características De Los Robots Industriales	32
1.3. Clasificación De Los Robots Industriales	36
1.3.1. Según La Configuración	36
1.3.1.1. Configuración Rectangular	37
1.3.1.2. Configuración Cilíndrica	38

1.3.1.3. Configuración Articulada	40
1.3.1.4. Configuración Esférica	42
1.3.2. Según El Sistema De Control	44
1.3.2.1. Sistema De Control De Lazo Abierto	45
1.3.2.2. Sistema De Control De Lazo Cerrado	46
1.3.3. Según La Potencia Utilizada	51
1.3.4. Según El Sistema De Programación	52
1.3.4.1. Programas De Detención Positiva	53
1.3.4.2. Programas De Trayectorias Punto a Punto	54
1.3.4.3. Programas De Trayectorias Continua	55
1.4. Elementos De Un Sistema Robótico	56
1.4.1. Actuador	60
1.4.1.1. Actuador Eléctrico	60
1.4.1.2. Actuador Neumático	62
1.4.1.3. Actuador Hidráulico	63
1.4.2. Manipulador	65
1.4.3. Herramienta Final	65
1.4.4. Interface	66
1.4.5. Sensores Y Transductores	67
1.4.5.1. Codificador Optico De Posición	68
1.4.5.2. Resolver	70
1.5. Tipos De Control Posicional	72
1.5.1. Control Punto A Punto	73

1.5.1.1. Método A	73
1.5.1.2. Método B	73
1.5.1.3. Método C	74
1.5.2. Control De Trayectoria Continua	74
II. DISPOSITIVOS Y CIRCUITOS ELECTRICOS	76
2.1. Introducción	76
2.2. Motor De Paso	77
2.2.1. Características	78
2.2.2. Tipos De Control	80
2.2.3. Circuito Manejador (Driver)	81
2.2.4. Circuito De Reloj Con LM555	85
2.3. Motor Trifásico	86
2.3.1. Características	88
2.3.2. Inversor Trifásico	92
2.3.3. Inversor Trifásico Conducción A 120°	94
2.3.4. Circuito De Control	99
2.4. Interruptor De Límite	104
2.5. Dispositivo Para Medir La Posición Angular	104
2.6. Fuentes De Poder	110
2.7. Pruebas De Laboratorio	112
2.7.1. Pruebas Del Circuito Manejador Del Motor De Paso	112
2.7.2. Pruebas Del Circuito Manejador Del Motor Trifásico	115

III. PROGRAMACION DEL CONTROLADOR LOGICO PROGRAMABLE	121
3.1. Introducción	121
3.2. Introducción Al Simatic S7-200 CPU 212	122
3.2.1. Equipos Necesarios	123
3.2.2. Capacidad De La CPU 212 Del S7-200	124
3.2.3. Principales Componentes Del SIMATIC S7-200	124
3.2.3.1. El Módulo CPU S7-200	125
3.2.3.2. Módulo De Ampliación	127
3.2.4. Sección De Entradas/Salidas	129
3.2.4.1. E/S Digitales	129
3.2.4.2. E/S Análogas	130
3.2.5. Partes De La CPU	132
3.2.5.1. Memoria Del Programa	132
3.2.5.2. Memoria De Datos	133
3.2.5.3. Memoria De Parámetros	134
3.2.6. Ciclo De la CPU	134
3.2.6.1. Leer Las Entradas Digitales	135
3.2.6.2. Ejecutar El Programa De Usuario	136
3.2.6.3. Procesar Las Peticiones De Comunicación Y Ejecutar El Autodiagnóstico De La CPU	136
3.2.6.4. Escribir Las Salidas Digitales	137

3.2.6.5. Interrumpir El Ciclo	138
3.2.6.6. Imagen Del Proceso De Las Entradas Y Salidas	138
3.2.6.7. Control Directo De Las Entradas Y Salidas	139
3.2.7. Establecer Comunicación Con Una CPU S7-200	140
3.2.8. Programación Del S7-200	142
3.2.8.1. Editor AWL (Lista de Instrucciones)	143
3.2.8.2. Editor KOP (Esquema de Contactos)	144
3.2.9. Crear Y Guardar Un Programa	145
3.2.8.1 Crear Un Programa	149
3.2.8.2 Guardar Un Programa Y Cargarlo En La CPU	150
3.2.10. Crear Un Bloque De Datos (DB)	151
3.2.11. Utilizar La Tabla De Estado Forzado	154
3.2.12. Utilizar El Direccionamiento Simbólico	155
3.3. Nuestra Solución de Automatización Con El SIMATIC S7-200 CPU 212	157
3.3.1. El Proceso A Automatizar	157
3.3.2. Funciones Del Proceso	158
3.3.2.1. Entradas	158
3.3.2.2. Salidas	160
3.3.2.3. Descripción Del Funcionamiento	161

3.3.3.	Lista De Nombres Simbólicos	167
3.3.4.	Programa General	168
IV.	EL VISUALIZADOR DE PROCESOS INTOUCH	193
4.1.	Introducción	193
4.2.	Principales Aspectos De InTouch	194
4.3.	Funciones De InTouch	195
4.3.1.	Gráfico Orientado A Los Objetos	195
4.3.2.	Características De Visualización	198
4.3.3.	Descripción De Texto	201
4.3.4.	Diccionario De Datos (Tagname Dictionary)	202
4.3.5.	Tipos De Tagname	205
4.3.6.	Enlaces De Animación	209
4.3.7.	Asistentes	214
4.3.8.	Funciones De Comando/Control	215
4.3.9.	Control Lógico De La Aplicación	216
4.3.9.1.	Sistema Lógico	216
4.3.9.2.	Editor Lógico	218
4.3.9.3.	Lógica De Control Condicional	218
4.3.9.4.	Lógica De Control De Teclado	219
4.3.10.	Escritura Lógica (Logic Script)	219
4.3.11.	Alarmas	222

4.3.11.1. Capacidad De Visualización De Alarmas	225
4.3.11.2. Capacidad De Archivo Y De Impresión De Las Alarmas	228
4.3.12. Eventos	229
4.3.13. Tendencia En Tiempo Real Y Recolección de Datos Históricos Distribuidos	229
4.3.14. Impresión Y Agenda De Reportes	231
4.3.15. Referencia Dinámica	231
4.3.16. Seguridad Del Sistema	232
4.3.17. Factory Focus	233
4.4. Módulos De InTouch	234
4.5. Desarrollo De La Aplicación	235
4.5.1. Diseño De La Pantalla De Control	235
4.5.2. Escritura Lógica De La Aplicación	236
CONCLUSIONES	248
ANEXO A	
Teoria Básica De Motores De Paso	250
ANEXO B	
Teoría Básica De Inversores Trifásicos	267

ANEXO C

Datos de elementos **273**

BIBLIOGRAFIA **283**

INDICE DE ABREVIATURAS

A	Imagen del Proceso de las Salidas
ACU	Acumuladores
BMP	Mapa de bits
BJT	Transistor de Unión Bipolar (Bipolar Junction Transistor)
CA o ca	Corriente Alterna
CAD	Convertidor Analógico a Digital
CD o cd	Corriente Directa
CLK	Señal de reloj (CLOCK)
CPU	Unidad Central de Procesamiento (Central Processing Unit)
DAC	Convertidor Digital a Analógico
DAQ	Adquisición De Datos (Data Acquisition)
DDE	Intercambio Dinámico de Datos (Dynamic Data Exchange)
E	Imagen del Proceso de las Entradas
EEPROM	Memoria ROM Programable, Eléctricamente Borrable (Electricity Erasure Programmable ROM)
E/S	Entrada/Salida
GTO	Interruptor Controlador por Compuerta

	(Gate Turn-Off)
HC	Contadores Rápidos
HMI	Interface Humano-Máquina (Human-Machine Interface)
Hz	Hertz
IGBT	Transistores Bipolares de Compuerta Aislada (Isolated Gate Bipolar Transistor)
M	Marcas Internas
MCT	Tiristor Controlado por Semiconductor de Oxido de Metal (MOS Control Tyristor)
MOSFET	Transistor de Efecto de Campo de Semiconductor de Oxido de Metal (Metal Oxide Semiconductor Field Effect Transistor)
MP	Imán Permanente
MPI	Interface Múltiple Punto (Multiple Point Interface)
OIS	Sistema de Interface con el Operador (Operator Interface System)
OLE	Enlace de Objetos y Empotrado (Object Linking and Embedding)
PC	Computadora Personal (Personal Computer)
PLC	Controlador Lógico Programable (Programmable Logic Controllers)

PPI	Interface Punto a Punto (Point-Point Interface)
PWM	Modulación por Ancho de Pulso (Pulse Width Modulation)
RAM	Memoria de Acceso Aleatorio (Random Access Memory)
RI	Robot Industrial
ROM	Memoria de Sólo Lectura (Read Only Memory)
RV	Reluctancia Variable
SCADA	Sistema de Adquisición de Datos y Control Supervisorio (System Control And Data Acquisition)
SIT	Transistores de Inducción Estática (Static Induction Transistor)
SM	Marcas Especiales
SPC	Control De Procesos Estadístico (Statistical Process Control)
SQL	Lenguaje Estructurado Requerido (Structured Query Language)
T	Temporizadores
TTL	Lógica de Transistor-Transistor (Transistor Transistor Logical)
V	Memoria de Variables
VCD, VCC, VS	Voltaje de Corriente Directa

INDICE DE FIGURAS

Figura		Pág.
1.1	Los seis grados de libertad para el movimiento de un robot	34
1.2	Añadiendo más grados de movimiento, no necesariamente se incrementarán los grados de libertad	35
1.3	Sistema de coordenadas rectangulares	37
1.4	Dos implementaciones de robot cartesiano	38
1.5	Sistema de coordenadas cilíndricas	39
1.6	Típica configuración de robot cilíndrico	40
1.7	Dos configuraciones de robot articulado	41
1.8	Implementación de robot articulado	42
1.9	Sistema de coordenadas esféricas	43
1.10	Típica configuración de robot esférico	43
1.11	Sistema de control de lazo abierto	46
1.12	Diagrama de bloques de un sistema de lazo cerrado	47
1.13	Diagrama de bloques de sistemas de control de lazo abierto y Lazo cerrado para actuadores de robot	50
1.14	Práctico sistema industrial de robot	58
1.15	Diagrama de bloques de un típico sistema robótico industrial	59
1.16	Tipos de pinzas con diferentes sistemas mecánicos	66
1.17	Sistema captador de imágenes para robot	67

Figura	Pág.
1.18 Disposición de un codificador óptico de posición	69
1.19 Estructura de un resolver de cuatro devanados	71
1.20 Tres métodos para desplazarse entre dos puntos	74
2.1 Foto del motor paso a paso utilizado	78
2.2 Secuencia de pulsos de control para conducción de paso completo	80
2.3 Diagrama esquemático del circuito manejador del motor de paso	83
2.4 (a) Vista superior de la tarjeta controladora del motor de paso	
(b) Circuito impreso	84
2.5 Circuito de reloj para el controlador del motor de paso	85
2.6 Foto del motor trifásico utilizado	87
2.7 Circuito de inversor trifásico, configuración de seis transistores y seis diodos	94
2.8 (a) Señales de base de los transistores para conducción a 120 grados	
(b) Formas de onda de voltaje de línea a neutro idealizadas	
(c) Formas de onda de voltaje de línea a línea idealizadas	98
2.9 Componentes de los campos magnéticos y del campo magnético neto durante cada uno de los seis intervalos de tiempo	99
2.10 Diagrama esquemático del inversor trifásico junto con el circuito controlador de encendido/apagado de los transistores	102
2.11 (a) Vista superior de la tarjeta controladora del motor trifásico	
(b) Circuito impreso	103

2.12	Potenciómetro de precisión (a) Potenciómetro de giro simple	
	(b) Diagrama esquemático del potenciómetro sin carga	105
2.13	Potenciómetro con carga	107
2.14	Dos clases de potenciómetros (a) Potenciómetro de alambre bobinado (b) Potenciómetro de película	108
2.15	Pasos de salida de voltaje de un potenciómetro de alambre bobinado	110
2.16	Señal de salida del manejador del motor de paso (A1 y B1) sin carga	113
2.17	Señal de salida del manejador del motor de paso (A1 y B1) con carga	114
2.18	Señal de salida del manejador del motor de paso (A1 y A2) con carga	114
2.19	Señal de salida de reloj del manejador del motor de paso	115
2.20	Señal de control aplicada a la base de los transistores Q1 y Q2	116
2.21	Señales de voltaje trifásicos VAN y VBN con carga resistiva	117
2.22	Señales de voltaje trifásicos VAB y VBC con carga resistiva	117
2.23	Señal de voltaje trifásico VAB con carga resistiva	118
2.24	Señales de voltaje trifásico VAB y VBC conectado al motor trifásico	118
2.25	Señales de voltaje trifásico VAB y VBC conectado al motor trifásico Con banco de capacitores	119
2.26	Señal de corriente trifásica IA y señal de voltaje trifásico VAB del motor trifásico con banco de capacitores	119
3.1	Micro-PLC Simatic S7-200 CPU 212	123
3.2	Componentes de un Micro-PLC S7-200	124
3.3	Módulo de ampliación	127
3.4	Ciclo de la CPU S7-200	134

3.5	Comunicación PPI con una CPU	141
3.6	Comunicación MPI en una red con maestro y esclavos	142
3.7	Ventana del Editor AWL con un programa ejemplo	143
3.8	Pila lógica de la CPU S7-200	144
3.9	Elementos básicos de KOP	145
3.10	Crear un nuevo proyecto	146
3.11	Ventana del editor KOP	150
3.12	Cargar los componentes del proyecto en la CPU	150
3.13	Ejemplo de bloque de datos	152
3.14	Ejemplo de una tabla de estado forzado	154
3.15	Ejemplo de una tabla de símbolos	157
3.16	Contacto NA del interruptor de límite 1 conectado en paralelo con el contacto NA del interruptor de límite 2	159
3.17	Proceso que se automatizará	162
4.1	Caja de herramientas (ToolBox)	197
4.2	Caja de diálogo para definir al diccionario de tagname	204
4.3	Ejemplo de pantalla que aparece cuando un tagname no ha sido	204
4.4	Diferentes tipos de tagnames	205
4.5	Caja de diálogo para definir los enlaces de animación	211
4.6	Ventana para la selección del Wizard	214
4.7	Ventana del menú para encontrar los Scripts	220
4.8	Ventana de diálogo del editor de Script	221
4.9	Ejemplo de Alarmas	224

4.10	Aplicación de la pantalla de Tendencia de Datos Históricos Distribuidos	230
4.11	Diseño de la pantalla de visualización en el Desarrollador (Window Maker)	237
4.12	Diseño de la pantalla de visualización en modo Runtime (WindowViewer)	238

INDICE DE TABLAS

Tabla		Pág.
I	Características de los robots según el tipo de programación	57
II	Ventajas y desventajas de los diferentes tipos actuadores	64
III	Datos de placa del motor de paso	79
IV	Secuencia de los pulsos de voltaje de cd en los respectivos devanados del motor de paso	81
V	Modo de operación del registro de desplazamiento bidireccional 74LS194	82
VI	Características principales del motor trifásico	92
VII	Condiciones de cada uno de los seis intervalos de tiempo que comprenden un ciclo completo de salida del inversor	95
VIII	Terminales de voltaje de salida de la fuente	112
IX	Notación para introducir valores en un bloque de datos	152
X	Identificadores válidos de tamaños	153
XI	Lista de elementos utilizados con su nombre simbólico y dirección	168

LISTA DE SIMBOLOS

θ	Angulo del eje del potenciómetro
θ_T	Angulo de barrido de rotación total
α	Factor de rotación desde la referencia del potenciómetro
Φ_{1-3}	Flujo magnético
DA - DF	Datos de entrada de los registros universales
DB	Bloque de Datos (Date Block)
E	Voltaje de excitación
ΔE	Incremento de voltaje
e_o	Voltaje de salida del potenciómetro sin carga
$e'o$	Voltaje de salida del potenciómetro con carga
g1 - g6	Señales de base de los transistores del inversor
Kp	Constante de voltios/radianes
n	Número de vueltas de alambre para el potenciómetro
OB1	Bloque Lógico 1
QA - QF	Datos de salida de los registros universales
Rp	Resistencia total del potenciómetro
R1-3, R2-4	Devanados del rotor del resolver
S1-3, S2-4	Devanados del estator del resolver

VA, VB, VC	Voltaje de línea
VAB, VBC, VCA	Voltaje de línea a línea
Vpc	Señal de pulso de la fotocelda del codificador óptico
VR1-3, VR2-4	Voltajes de excitación de los devanados del rotor del resolver
VS1-3, VS2-4	Voltajes de excitación de los devanados del estator del resolver

INTRODUCCION

Con la llegada de las potentes computadoras diseñadas para el control de las máquinas industriales, éstas han aumentado su versatilidad de desarrollar varios ciclos de trabajo. Esta relación que surgió entre las computadoras y las máquinas industriales es lo que empujó a los robots industriales a entrar en escena.

Nuestro proyecto se basa en diseñar un programa que automatizará el sistema mecánico que servirá para realizar el llenado de un estante por medio de cajas, las cuales serán colocadas de una manera ordenada. Para esto, utilizamos un controlador programable lógico (PLC), que utilizará las instrucciones del programa diseñado para crear las señales de salidas del PLC, dichas señales de salida servirán de entrada para los circuitos de control de estado sólido. Estos circuitos de control de estado sólido que también han sido diseñados manejarán directamente los motores del sistema.

Para poder supervisar, controlar y monitorear toda una línea de producción de una industria desde el monitor de una computadora, se creó la interface humano-máquina (HMI). Por medio de ésta, podemos mandar a prender un motor, una bomba o una válvula, simplemente pulsando la representación de una botonera que se encuentra en pantalla por medio del ratón.

CAPITULO 1

ROBOTS INDUSTRIALES

1. INTRODUCCION

La llegada de la microcomputadora de bajo costo, ha aumentado la adaptación de la maquinaria industrial, debido a que su ciclo de manipulación puede ser reorganizado completamente, por lo tanto, consiguiendo que la máquina pueda realizar varias tareas. Esta es la idea que empujó a los Robots Industriales (RI) a entrar a escena.

En este capítulo analizaremos de manera general, los conceptos básicos de las partes principales que forman los robots (hardware y software).

Vamos a exponer las clasificaciones más importantes de robots industriales.

Describiremos las características de cada una de las tres categorías de programación para robots.

Explicaremos la operación del codificador óptico y del resolver, los cuales sirven como codificadores de posición para controlar el movimiento de los robots. Describiremos los diferentes sistemas de control utilizados para guiar a un robot a través de su secuencia de trabajo.

1.2. CONCEPTO DE ROBOT INDUSTRIAL

La automatización, el precursor de los dispositivos robóticos modernos, es ampliamente utilizado en los ambientes industriales. Pero, qué es un robot y qué lo hace diferente de la automatización industrial?

La *automatización* consiste principalmente, en diseñar sistemas capaces de ejecutar tareas repetitivas hechas por los hombres y de controlar operaciones sin la intervención humana. El término automatización también se utiliza para describir a los sistemas de no manufactura, en los cuales los dispositivos automáticos pueden operar casi independientemente del mando humano. Por ejemplo, en la industria de las comunicaciones, la aviación y astronáutica, se usan tales dispositivos como el teléfono automático y los pilotos automáticos (5).

La mayoría de las industrias han sido automatizadas o utilizan tecnología para automatizar alguna de sus operaciones. Los dispositivos automáticos han estado

en la escena industrial por más de 100 años; ellos pueden ser dispositivos mecánicos diseñados para ejecutar una específica tarea, lo cual sería un limitante debido a que tales dispositivos no pueden ser fácilmente cambiados. A este aspecto se conoce como automatización fija (no flexible).

La automatización fija se caracteriza por los siguientes aspectos (2):

- a. Relativamente barato y simple de implementar.
- b. Eficiente y confiable en operación, ya que todos los movimientos son fijados dentro de la estructura del dispositivo.
- c. Fácil de controlar ya que realiza una tarea fija y no existen movimientos redundantes.
- d. Por lo limitado de su alcance y poca flexibilidad, y si el modelo de producción cambia puede convertirse en obsoleto de manera rápida.
- e. Requiere poco mantenimiento o intervención humana, debido a su propio diseño de construcción.

Como vemos, uno de los principales motivos por el cual la automatización fija ya no se utiliza, es su poca flexibilidad para realizar diferentes tareas, por lo tanto, es comprensible que a finales de los 60' se empezara con el estudio para diseñar robots, en el Instituto de Investigación de Stanford, aquí se diseñó al robot llamado *Shakey*, este robot fue capaz de tomar bloques en una pila utilizando una cámara de video como sensor visual y procesando esta información en una pequeña computadora. Tiempo después, a mediados de 1970, la General Motors financió un programa de desarrollo en el que el investigador Victor Scheinman del Instituto de Massachusetts inventó un brazo mecánico para producir el llamado "manipulador universal programable para

ensamblaje" (programmable universal manipulator for assembly, *PUMA*), marcándose así el inicio de la era de los robots.

Los robots son una subcategoría de la automatización, tienden a ser más versátiles y adaptables que los dispositivos automáticos dedicados, por lo que las raíces de su desarrollo se debe al esfuerzo para automatizar algunas o todas las operaciones requeridas en una fábrica.

La definición precisa y detallada de los robots aún es tema de debate en los círculos de robótica. Sin embargo, todo el mundo está de acuerdo en que, cuando menos un robot es: *"Un manipulador mecánico, reprogramable y multifuncional, que puede fácilmente estar dirigido a realizar una variedad de tareas, de manera automática, a través de diversos movimientos, controlados por una computadora programable"*.

1.2.1. CARACTERISTICAS DE LOS ROBOTS INDUSTRIALES

Los dispositivos automáticos dedicados y los RI, presentan características comunes como la operación automática sin la intervención humana. Pero los RI exhiben ciertas características que los hace diferentes. Estas son las siguientes (2):

1. Los robots industriales son controlados por computador

La mayoría de los robots industriales modernos son controlados por medio de sistemas de computadoras dedicadas digitales. El control por

computadora es necesario por las siguientes razones:

- a) Para llevar a cabo complejos cálculos matemáticos y coordinar transformaciones para manipular al robot en tres dimensiones.
- a. Para coordinar el movimiento de los ejes, logrando exactitud sin importar las veces que se repita los ciclos de trabajo.
- a. Para simplificar el diseño, desarrollo, programación y control de los robots.
- b. Para recibir y procesar las múltiples entradas que provienen de una variedad de sensores externos.
- c. Para obtener una operación rápida y confiable.

1. Los robots industriales son re-programables

Los movimientos y acciones de un robot industrial, pueden ser determinados inicialmente por un operador humano. La secuencia de movimientos requeridos podrían ser guardados en una computadora de control, la cual conducirá al robot a ejecutar dichos movimientos, repetidamente y con gran exactitud. Si la tarea cambia, el robot puede fácilmente adaptarse a las nuevas condiciones; desde que la operación del robot está bajo control por software, el robot puede ser considerado como un tipo de automatización suave.

2. Los robots industriales tienen múltiples movimientos

Los robots industriales pueden efectuar maniobras de tres dimensiones, fuera de su forma estructural. Seis grados de libertad se encuentran presentes en la mayoría de los RI, esto lo hace adaptable para realizar una gran cantidad de operaciones. Adicionales grados de libertad pueden ofrecer un incremento en la accesibilidad para ciertas

aplicaciones. Más adelante, explicamos los términos grados de libertad y libertad de movimiento.

Para entender mejor las características de los robots, vamos a explicar brevemente algunos términos, entre ellos los más importantes son:

Grados de Libertad es un término que se refiere al movimiento que puede realizar un eje independiente del movimiento de cualquier otro eje. Cualquier cuerpo en el espacio tiene seis grados de libertad debido a que puede tener movimientos lineales a través de los tres ejes perpendiculares entre sí (ver figura 1.1), y movimiento rotacional alrededor de dichos ejes.

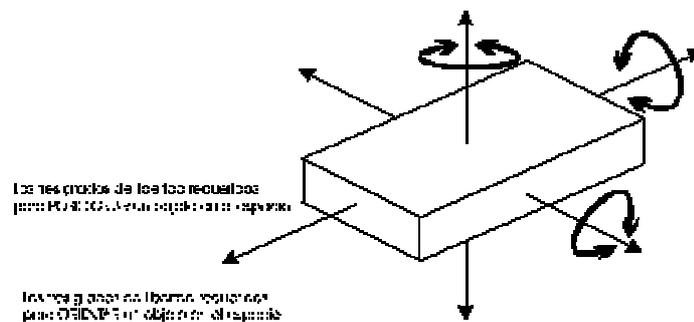
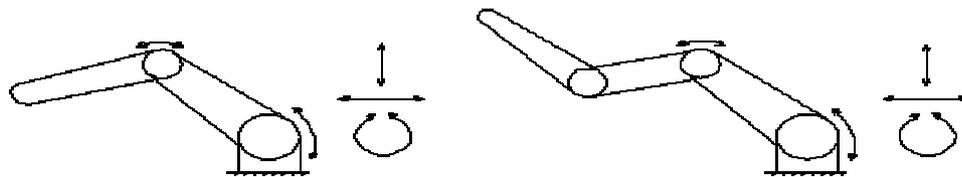


FIGURA 1.1. Los seis grados de libertad para el movimiento de un robot

Como se indica en la figura 1.2, hay que diferenciar entre grados de libertad y grados de movimiento. En la figura 1.2 (a), el brazo tiene dos ejes de

movimiento representado por las juntas A y B. Las juntas, cuando actúan separadamente o juntas, son capaces de mover al brazo a través de tres grados de libertad. Estos son: dos movimiento lineales y un movimiento rotacional; ahora bien, añadiendo más juntas en el brazo, como se ve en la figura 1.2 (b), no necesariamente se incrementa los grados de libertad (2).



a. (b)

FIGURA 1.2. Añadiendo más grados de movimiento, no necesariamente se incrementarán los grados de libertad (a) Brazo con dos juntas (b) Brazo con tres juntas

Juntura rotatoria o prismática es la parte del robot en la cual se unen dos enlaces. Por medio de ellas se logra el movimiento deseado, el cual puede ser conseguido por deslizamiento o rotación de los enlaces, aunque también se puede obtener por medio de la combinación de estos dos tipos de movimientos. Juntura prismática es aquella que produce movimiento por deslizamiento; mientras que la juntura rotatoria lo hace por rotación.

1.3. CLASIFICACION DE LOS ROBOTS INDUSTRIALES

Así como existen varias definiciones de robot, son muchas las formas de clasificar a los diferentes tipos de robots. Entre las varias clasificaciones que existen, algunas están relacionadas con otras, por lo tanto, es posible que un robot esté dentro de varias clasificaciones. Una de las principales clasificaciones se ha hecho de acuerdo a los niveles de inteligencia, o al tipo de control por

computadora que el robot muestra. Las más comunes formas de clasificación son las siguientes:

1.3.1. SEGÚN LA CONFIGURACION

Los robots industriales pueden ser agrupados de acuerdo a su diseño físico (estructura geométrica). Por ejemplo, algunos robots tienen brazos rotatorios mientras otros pueden moverse solamente en línea recta. El término grados de libertad llega a ser muy importante cuando consideramos las configuraciones de robot, ya que puede influir significativamente en el espacio de trabajo en el cual va a desarrollarse.

Cuatro son los diseños físicos básico para los RI, que a continuación explicamos.

1.3.1.1. CONFIGURACION RECTANGULAR

Esta es la más simple de todas las configuraciones y está representado en la figura 1.3. Los tres ejes forman el sistema tridimensional, con cada eje formando 90° (ortogonal) con respecto a los otros. A veces, también lleva el nombre de configuración rectangular, debido a que su área de trabajo genera un volumen rectangular (3).

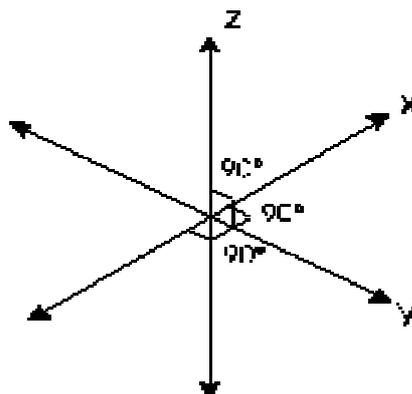


FIGURA 1.3. Sistema de Coordenadas Rectangulares

A continuación incluiremos algunas de las ventajas particulares de esta configuración (2):

- a. Movimientos fácilmente controlados y programados.
- b. Alta exactitud.
- c. Precisión, velocidad y capacidad de carga.
- d. Sencillo sistema de control.
- e. Las coordenadas X, Y y Z son fácilmente entendibles.
- f. Gran área de alcance.
- g. Amplia capacidad de carga.
- h) Estructura simple, ofrece confiabilidad.
- i) Fácil para expandir debido a su diseño modular.

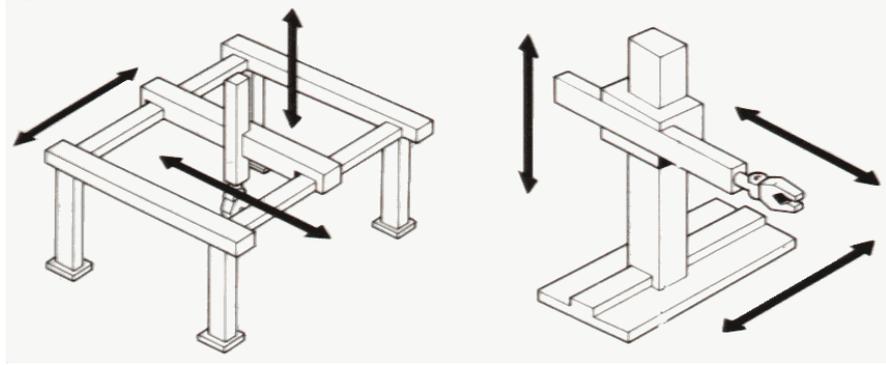


FIGURA 1.4. Dos implementaciones de robot cartesiano

Esta configuración encuentra aplicación en áreas donde se precisan movimientos lineales y exactos, por ejemplo en aplicaciones de *recoge y coloca*. Dos implementaciones de este tipo de configuración son ilustrados en la figura 1.4 (3).

1.3.1.2. CONFIGURACION CILINDRICA

El siguiente nivel de complejidad geométrica es el sistema de coordenadas que describen la posición en términos de un cilindro; es llamada de esta forma debido a que su movimiento engendra un volumen cilíndrico de trabajo.

Las ventajas de esta configuración incluyen (2):

- Movimientos fácilmente controlados y programados.
- Simple sistema de control
- Gran exactitud.

- Rápida operación.
- Fácil acceso al frente y a los lados.
- Estructura simple, ofrece alta confiabilidad.

La figura 1.5 ilustra este tipo de sistema. Es posible moverse de arriba hacia abajo, desde O_1 hasta O_2 ; de adentro hacia fuera, desde X_2 hasta X_1 y viceversa; y rotar alrededor del eje OY , un ángulo ϕ (3).

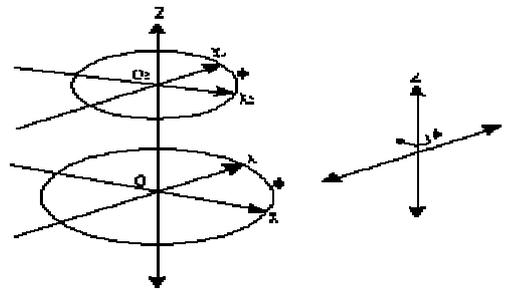


FIGURA 1.5. Sistema de Coordenadas Cilíndricas

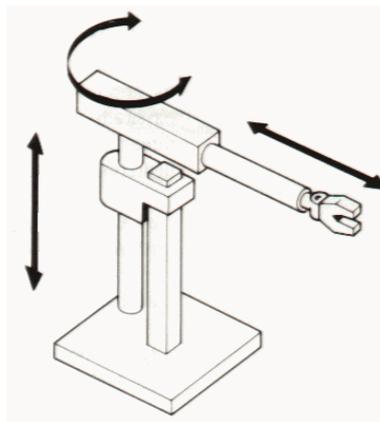


FIGURA 1.6. Típica configuración de robot cilíndrico

La mayoría de los robots de este tipo son operados neumáticamente.

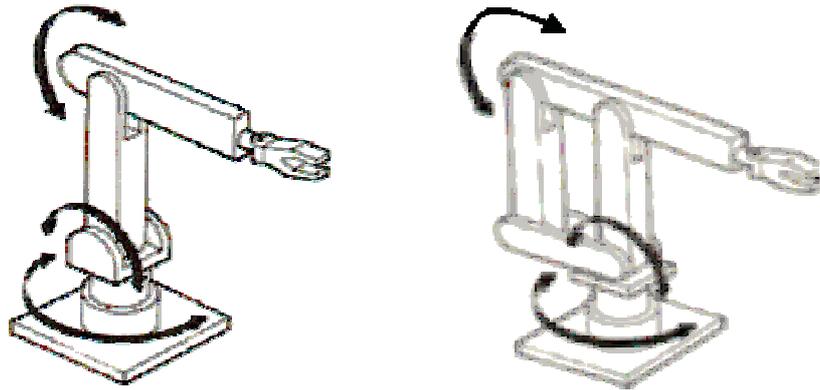
Una típica configuración cilíndrica es ilustrada en la figura 1.6.

1.3.1.3. CONFIGURACION ARTICULADA

La configuración articulada comprende un número de rígidos enlaces conectados en juntas rotatorias. Este tipo de robot, a veces llamado *configuración de brazo articulado*, debido a que el movimiento que realiza se parece a los movimientos del cuerpo humano (configuración antropomórfica). La estructura cinemática abierta y cerrada son dos tipos de configuraciones articuladas. Estas son ilustradas en la figura 1.7 (a) y (b).

Típicas ventajas de esta configuración incluyen (2):

- Facilidad de manejo.
- Fácil acceso hacia el frente, a los lados y en la parte posterior.
- Rápida operación debido a las articulaciones rotatorias.
- Habilidad para ejecutar complejas trayectorias continuas.
- Gran alcance para pequeñas áreas.
- Diseño que permite fácil integración con los esquemas del área de trabajo.



a. (b)

FIGURA 1.7. Dos configuraciones de robot articulado (a) Estructura cinemática abierta (b) Estructura cinemática cerrada

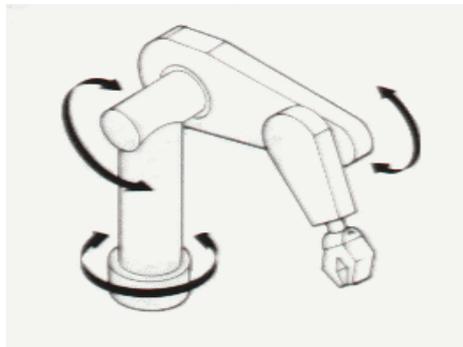


FIGURA 1.8. Implementación de robot articulado

Debido a que todos los movimientos son producidos por la rotación angular de las articulaciones, se necesitan complejos cálculos y transformaciones para mover el brazo a través de movimientos lineales. Para programar este tipo de brazo puede ser difícil debido a que no es fácil visualizar los movimientos tridimensionales (3).

1.3.1.4. CONFIGURACION ESFERICA

La herramienta de esta clase de robot puede moverse un ángulo azimut ϕ con respecto al eje OZ y al eje OX, también puede moverse un ángulo de elevación θ con respecto al eje OY (ver figura 1.9). Este tipo de robot puede emular un considerable número de operadores humanos (3).

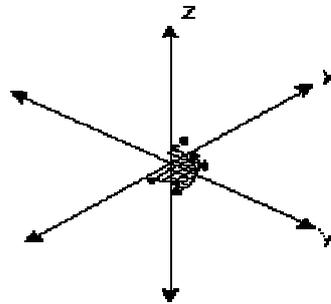


FIGURA 1.9. Sistema de coordenadas esféricas

Algunas ventajas de esta configuración son (2):

- a. Movimientos fáciles de programar y controlar.
- b. Coordenadas polares de fácil comprensión.
- c. Gran capacidad de carga
- d. Rápida operación.
- e. Exactitud y repetibilidad.

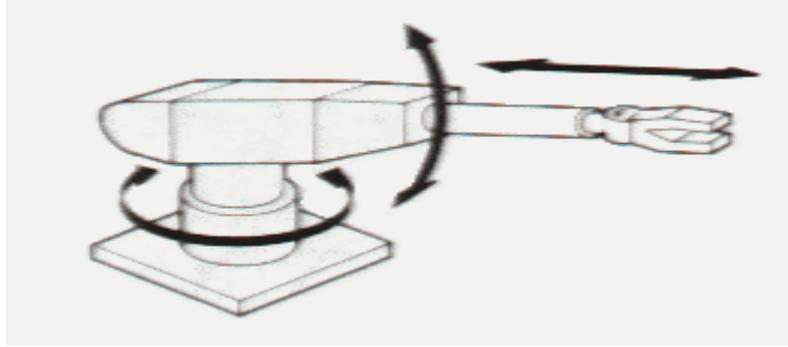


FIGURA 1.10. Típica configuración del robot esférico

Una de sus desventajas sería, no poder utilizarlo donde se necesite complicadas trayectorias de movimiento. Esta configuración es ilustrada en la figura 1.10.

2. SEGÚN EL SISTEMA DE CONTROL

Los RI también pueden ser clasificados según el sistema de control utilizado para gobernar su movimiento. Los sistemas de control se emplean para conseguir un incremento de la productividad y un mejor rendimiento de un aparato o sistema. Tal sistema puede ser mecánico, eléctrico, electrónico, hidráulico, neumático, o una combinación de ellos. Es muy importante elegir el sistema de control que se va a utilizar para gobernar el movimiento de un robot.

Un sistema de control podría ser definido como: *"Uno o más dispositivos interconectados, los cuales al trabajar de manera sincronizada, automáticamente mantendrán o alterarán la respuesta deseada del sistema"* (2).

Los principales aspectos de control aplicado a los RI son los siguientes (2):

- a. Control de velocidad, posición y movimiento de los ejes.
- b. Programa de control y coordinación de los ejes.
- c. Control de los dispositivos de entrada y salida.
- d. Control global de todo el sistema robótico.

Para lograr un exacto control del movimiento se necesita constantemente información de estas dos variables: velocidad y posición. Dos son los sistemas de control más utilizados, éstos son: los de lazo abierto y los de lazo cerrado.

1.3.2.1. SISTEMAS DE CONTROL DE LAZO ABIERTO

Los sistemas de control de lazo abierto no emplean retroalimentación, esto implica que ni la velocidad ni la posición del eje están siendo medidos. Para lograr un control exacto de posición y velocidad en sistemas de control de lazo abierto, se emplea un motor especial conocido como motor de paso, el cual es controlado por medio de una secuencia de pulsos. El motor de paso es el único que no necesita la aplicación de voltaje variable, diferente situación a la de los motores convencionales de corriente alterna y corriente directa.

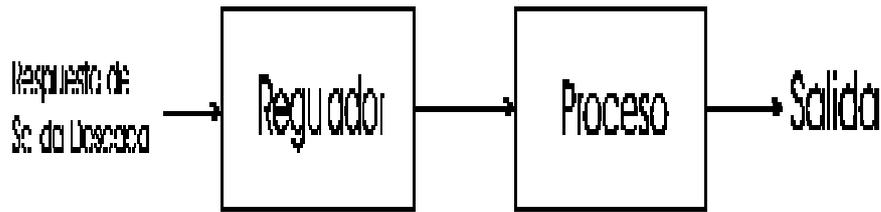


FIGURA 1.11. Sistema de control de lazo abierto

Como observamos en la figura 1.11, la relación entrada-salida representa la relación de causa y efecto del proceso, la cual a su vez representa el procesamiento de la señal de entrada para proporcionar una variable de señal de salida, frecuentemente con una amplificación de potencia. Un sistema de control de lazo abierto utiliza un actuador de control, a fin de obtener la respuesta deseada. Este tipo de control es el método más sencillo para controlar a un robot. Una manera de utilizar este tipo de sistema de control es empleando motores de paso, para mover cada parte del robot.

1.3.2.2. SISTEMAS DE CONTROL DE LAZO CERRADO

En contraste con un sistema de lazo abierto, el de lazo cerrado utiliza una medición adicional de la salida real, para compararla con la respuesta de salida deseada.

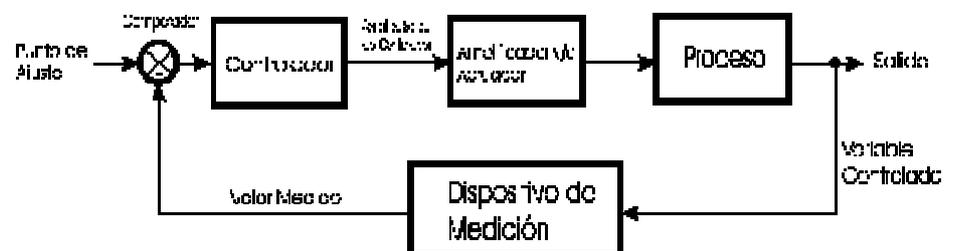


FIGURA 1.12. Diagrama de bloques de un sistema de lazo cerrado

En la figura 1.12, se muestra el diagrama de bloques de un sistema de control simple de lazo cerrado con retroalimentación. Las ideas principales de este diagrama de bloques son las siguientes: se mide una *variable de proceso* que está siendo controlada (temperatura, presión, humedad, velocidad mecánica o, en nuestro caso, posición mecánica), y se alimenta a un *comparador*. El comparador que puede ser mecánico, eléctrico o neumático, lleva a cabo una comparación entre el valor medido de la variable y el *punto de ajuste*, que representa el valor deseado de la variable. El comparador genera una *señal de error*, que representa la diferencia entre el valor medido y el deseado. Se considera que la señal de error es igual al valor medido menos el valor deseado, por lo que, si el valor medido es demasiado grande, la señal de error es positiva, y si el valor medido es demasiado pequeño, la señal de error es de polaridad negativa. Esto se expresa en la ecuación

$$\text{Error} = \text{valor medido} - \text{punto de ajuste} \quad (1-1)$$

El controlador, que también puede ser eléctrico, mecánico o neumático, recibe la señal de error y genera una señal de salida. La relación entre la señal de salida del controlador y la señal de error depende del diseño y ajuste del controlador.

Todos los controladores en lazo cerrado pueden clasificarse en cinco clases, o modos de control, los cuales son:

1. Encendido – apagado (on-off)

2. Proporcional
3. Proporcional más integral
4. Proporcional más derivativo
5. Proporcional más integral más derivativo

La lista anterior de modos de control, está clasificada por orden de complejidad de los mecanismos y los circuitos involucrados. Esto es, el modo de encendido-apagado, es el más sencillo de poner en práctica. A medida que se baja por la lista, la construcción de los controladores se vuelve más compleja.

En general, entre más difícil sea el problema de control, debemos irnos más abajo en la lista para encontrar el modo de control adecuado. Sin embargo, en muchos procesos industriales no se requiere un control muy preciso; o la naturaleza del proceso puede ser tal que es sencillo conseguir un control preciso.

Dentro de los modos de control, hay ciertas variaciones, pero estas variaciones no constituyen una diferencia de control esencial. El modo de control no tiene nada que ver con la naturaleza eléctrica, mecánico o neumática del controlador. Depende sólo de cuán drásticamente y de qué manera reacciona el controlador a una señal de error. Siendo más precisos, depende de la relación matemática entre la salida del controlador y su entrada (su entrada es la señal de error) (4).

Ahora podremos entender mejor la definición de un sistema de

control de lazo cerrado. Esta es la siguiente: "Un sistema de lazo cerrado es aquel que tiende a mantener una relación sugerida de una variable del sistema con otra, comparando funciones de estas variables y usando diferencias como medio de control".

Estos sistemas son aplicados para controlar el movimiento, la posición y la velocidad de cada eje del robot; una instrucción del programa de control llega a ser la señal de entrada, el eje de motor sería el dispositivo controlado y la posición del eje sería la variable a controlar. La señal de entrada es una señal de baja potencia que no podrá manejar directamente al actuador (motor), por lo que es indispensable introducir una etapa amplificadora entre ellos; el actuador operará sobre el elemento del robot que se desea mover, el cual tendrá acoplado algún dispositivo medidor de posición, cuya función será la de producir enviar una señal que será retroalimentada y comparada con la señal de entrada. El comparador emitirá una señal proporcional a la diferencia entre las dos señales (señal de entrada y la señal del dispositivo de medición).

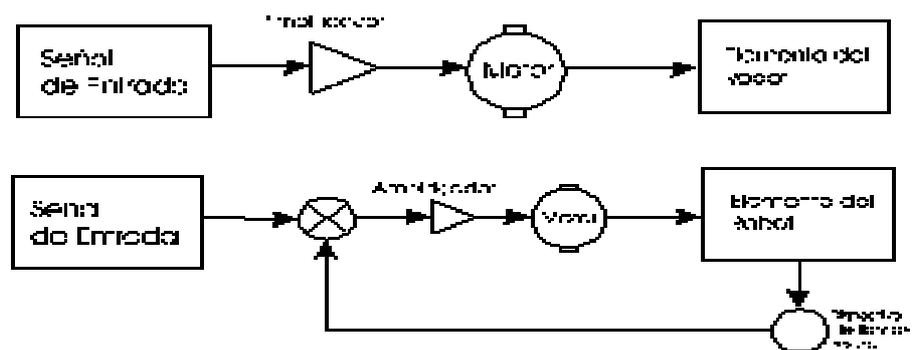


FIGURA 1.13. Diagrama de bloques de sistemas de control de lazo abierto y lazo cerrado para actuadores de robot

En la figura 1.13, se muestra el diagrama de bloques de los sistemas de lazo abierto y cerrado, aplicados para el control del eje de un robot industrial.

Los sistemas de lazo cerrado son la base para el diseño de sistemas de control automático aplicados al manejo de robots. Cualquier sistema de lazo cerrado que sirva para controlar la posición mecánica de cualquier dispositivo, es más conocido como un *servomecanismo* (2).

1.3.3. SEGÚN LA POTENCIA UTILIZADA

Los robots industriales pueden ser alimentados por medio de dispositivos eléctricos, hidráulicos, neumáticos y mecánicos. Los requerimientos necesarios para una aplicación influirán en el tipo de potencia requerida; por ejemplo, aplicaciones donde se requiera el uso de una gran potencia para levantar cargas, es necesario utilizar los actuadores hidráulicos. En muchos casos es indispensable manejar más de un sistema de potencia. Nosotros nos referiremos exclusivamente a la potencia eléctrica.

Entonces, los RI también pueden ser clasificados según la potencia que utilicen. Hemos guiado nuestra atención hacia los actuadores eléctricos (motor de paso, motor trifásico) con el correspondiente diseño del circuito electrónico de control. Más adelante hablaremos brevemente de cada tipo de actuador y luego en una tabla daremos a conocer las ventajas y desventajas de cada tipo.

1.3.4. SEGÚN EL SISTEMA DE PROGRAMACION

El tipo de programación utilizado para el control del robot, está

ligado con la complejidad de movimiento que tiene que realizar el robot y con el tipo de control, es decir, mientras más complejo sea el movimiento, más complicada será la programación a usar. El rango de dificultad, va desde la colocación física de un interruptor (switch) de límite para determinar una serie fija de desplazamientos, hasta escribir programas que serán utilizados en computadoras dedicadas para ser usados exclusivamente en robots.

Hay que entender bien un concepto: *"la categorización común de los robots se hace en base a la complejidad del programa, no en la complejidad mecánica del robot"*; por supuesto, es un desperdicio tener un robot de alto nivel desempeñando manipulaciones sencillas. Por lo tanto, cuando seleccionamos el tipo de robot a utilizar, tratamos de igualar las capacidades físicas del robot con el tipo de programación a utilizar.

Los tres tipos de programación son (4):

1.3.4.1. PROGRAMAS DE DETENCIÓN POSITIVA

Un programa de detención positiva produce un movimiento solamente entre dos puntos fijos en cualquiera de los ejes del robot. Con esto queremos decir que sólo hay dos posiciones posibles en las que puede detenerse cada eje o articulación.

Dado que los ejes son mecánicamente independientes, no existe ninguna regla que diga que puede moverse un eje a la vez. De hecho la mayoría de los programas producen movimiento en varios ejes a la vez. Tal movimiento se llama *movimiento compuesto*.

Generalmente, la posición más lejana a la que es capaz de moverse un eje del robot, se encuentra determinada por la posición de un interruptor (switch) de límite ajustable. Se deben proporcionar dos interruptores de límite por cada eje. Cada interruptor señala a la microcomputadora o PLC que el eje ha alcanzado su límite en una dirección. La parte en movimiento no podrá detenerse en una posición intermedia entre los dos puntos, pues las instrucciones del programa provocan que continúe el desplazamiento hasta que reciba una señal de detención, por parte de uno de los interruptores de límite. Esta incapacidad de detener el eje del robot en cualquier punto, distinto de las dos posiciones límite, es la característica de los programas de detención positiva.

Se necesitan topes mecánicos ajustables en este tipo de programas, para evitar que la parte mecánica en movimiento del robot, sobrepase los interruptores de límite. Esta precaución es necesaria, porque si un eje pasara de largo a través de su posición límite, no habría ninguna previsión en el programa para permitirle recuperarse y regresar a su posición deseada.

Este tipo de programa se adapta mejor en aplicaciones donde el robot deba levantar una carga y depositarla en una ubicación distinta.

1.3.4.2. PROGRAMAS DE TRAYECTORIA PUNTO A PUNTO.

La característica esencial que distingue a un programa *punto a punto* es la capacidad que posee un eje de robot de ubicarse en cualquier punto dentro de sus dos posiciones límite. Por ejemplo, si la mecánica de nuestro robot le diera un rango inherente de movimiento de 0° a 128° en el hombro, un *programa punto a punto con resolución de 8 bits* (1 parte en 256) podría posicionar el hombro a 0.0° , a 1.0° , a 1.5° y así hasta 127.5° ; por consiguiente, el eje podrá ubicarse en cualquiera de las dos posiciones.

El programa punto a punto debe especificar digitalmente una posición destino para cada eje, de manera que, cuando cada eje alcance dicha posición destino, la herramienta estará en una nueva posición espacial (4).

1.3.4.3 PROGRAMAS DE TRAYECTORIA CONTINUA.

Un programa de trayectoria continua es como un programa punto a punto, pero con las posiciones destino muy cercanas. Sin embargo, el programa de trayectoria continua es capaz de mover el dispositivo herramienta a la posición destino con mucha

rapidez, también desplazarlo a través de una trayectoria virtualmente invariable, por la sencilla razón de que la posición inicial actual de cada eje es muy cercana a la posición destino.

El proceso de instrucción que usa un robot de trayectoria continua es éste: La secuencia de manipulación o trayectoria, que tiene que desarrollar la herramienta final, es partida en una gran cantidad de pequeños movimientos compuestos, haciendo que el programa monitor tome muestras rápidamente de los diferentes codificadores de posición, a medida que el dispositivo herramienta es maniobrado por un operador a través de una trayectoria deseada. Se podría ver al operador como al maestro que guía al alumno (robot) en la realización de una tarea (secuencia de movimientos).

En la tabla I, se muestran las características de cada tipo de programación.

1.4 ELEMENTOS DE UN SISTEMA ROBOTICO

Una instalación de un robot industrial puede visualizarse como se sugiere en la figura 1-14. Un operador dispone que un programa sea ingresado en una microcomputadora (computadora dedicada). Una vez que se ha ingresado un programa correcto, la supervisión por parte del operador ya no es requerida.

<i>TIPO DE ROBOT SEGÚN SU PROGRAMACION</i>	<i>CARACTERISTICAS</i>
---	-------------------------------

<p style="text-align: center;">DETENCION POSITIVA</p>	<ul style="list-style-type: none"> • Bajo Costo • Fácil mantenimiento • Rápido funcionamiento • No realimentados • Usualmente utilizan actuadores neumáticos • Generalmente se controlan con PLC • En sus programas se hace un control para la secuencia de trabajo y el tiempo • Aplicable a tareas no muy complicadas • Limitada cantidad de información puede ser guardada en su memoria • Tres o Cuatro grados de libertad • Flexibilidad pobre • Difícil de programar
<p style="text-align: center;">PUNTO A PUNTO</p>	<ul style="list-style-type: none"> • Más fácil de manejar • De cuatro a seis grados de libertad • Control por retroalimentación • Usado para carga y descarga • No pueden usarse en aplicaciones

En muchos robots, el mecanismo suministra señales de regreso (sistemas retroalimentados) a la microcomputadora, lo que permite al programa seguir la pista de las direcciones de los diferentes miembros del mecanismo. Una vez que se ha completado la manipulación, el mecanismo regresa a su posición inicial y el programa se detiene. Cuando las condiciones del entorno industrial nuevamente son las adecuadas, se repite el proceso completo (4).

Si bien, un robot industrial moderno es visto como una máquina inteligente, esto se debe a la combinación de vieja y nueva tecnología, la cual ha seguido evolucionando para el bien del desarrollo industrial

Un sistema de robot consta de varios subsistemas. En la figura 1.15 observamos el diagrama de bloques de un sistema típico robótico.

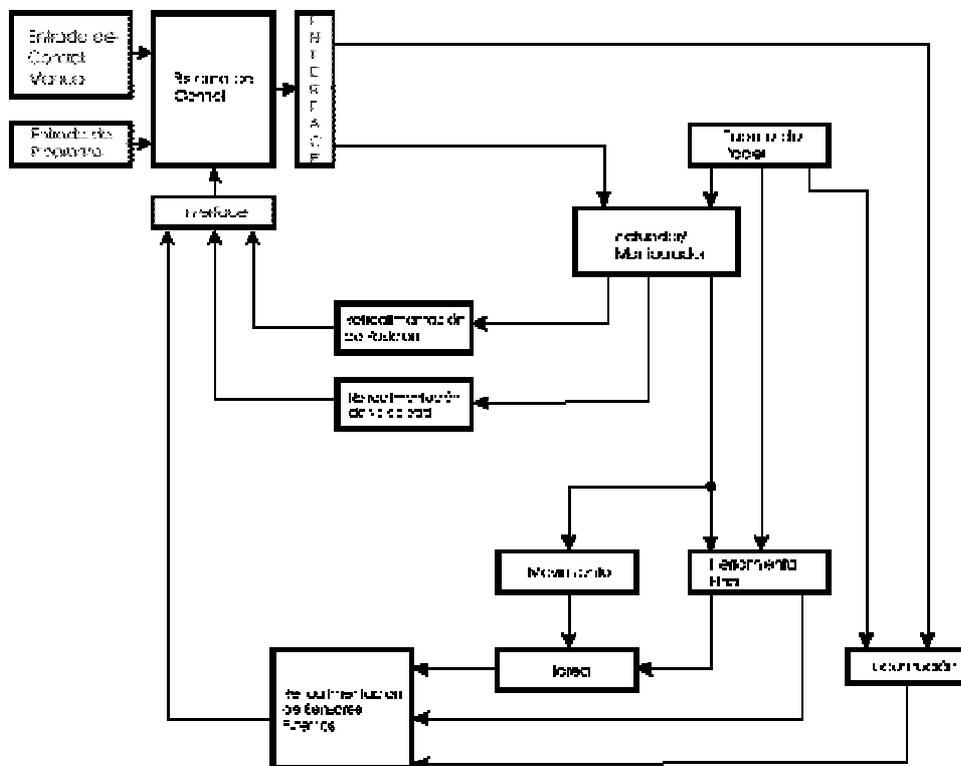


FIGURA 1.15. Diagrama de bloques de un típico sistema robótico

industrial

La figura muestra cada una de estas tecnologías que integradas de manera apropiada, servirán para un robot particular. Ciertos elementos físicos pueden ser comunes para todos los RI; por ejemplo, cilindros de aire y servo motores de corriente directa nunca estarán juntos para operar un robot. Un típico robot

industrial incluye los siguientes elementos (2):

1.4.1. ACTUADOR

Con este término se conoce al dispositivo que provee de movimiento al robot. En general, habrá un actuador por cada eje de movimiento. Los actuadores pueden ser eléctricos, como los servomotores de corriente alterna y de corriente directa, y el motor de paso; hidráulicos y neumáticos con sus respectivos cilindros, pistones y actuadores rotatorios. O una combinación de éstos con elementos de transmisión mecánica. Si se emplean actuadores hidráulicos o neumáticos, entonces es necesario equipo adicional como bombas, compresores, acumuladores, filtros, etc.

Los actuadores mecánicos también pueden ser empleados como dispositivos intermediarios para la conversión de movimiento, tales dispositivos pueden ser los engranes, correas, cadenas, cables y otros, para producir un sistema práctico de trabajo.

1.4.1.1. ACTUADOR ELECTRICO

Existen varias clases de actuadores eléctrico como son:

- Servomotor de corriente directa;
- Servomotor de corriente alterna;
- Motor de corriente alterna; y
- Motor de paso.

El más utilizado es el servomotor de corriente directa, aunque un limitado número de fabricantes emplean servomotores de corriente alterna. Los motores de paso se utilizan muy poco debido a su poca potencia, generalmente empleados para fines didácticos.

Como se observa en la tabla II, la principal desventaja de usar un motor eléctrico para manejar el movimiento del robot, es que no tiene la capacidad para lograr un posicionamiento preciso. Cuando se requiere un posicionamiento preciso, muchos fabricantes de robots instalan frenos mecánicos para hacer frente al problema. Los frenos detienen el eje del motor eléctrico tan pronto deja de girar, esto evita que el contrapar derivado de la carga gire con el eje del motor, y por tanto evita una caída.

Otro inconveniente de los motores eléctricos es que ellos giran muy rápido, por lo que se necesitan reductores mecánicos como

engranes para reducir la velocidad. Para nuestro caso de estudio, se ha elegido el motor de paso y el motor trifásico de corriente alterna. El actuador eléctrico es probablemente el más usado de entre los tipos de actuadores.

1.4.1.2. ACTUADOR NEUMATICO

Los robots simples y sencillos, también llamados como *robots recoge y coloca*, son la mayor parte de las veces operados con aire, lo cual permite movimientos repetitivos a un mínimo costo.

Los actuadores neumáticos tienen una capacidad de potencia menor, en comparación a la potencia desarrollada por los actuadores eléctricos o hidráulicos, debido a las desventajas inherentes asociadas con la compresión del aire. Un cilindro neumático servirá para un desplazamiento *todo o nada* (on-off). La extensión y el retorno de un miembro o la rotación de una junta, se lleva a cabo por medio de cilindros de aire de acción simple o doble. Cilindros de acción simple obviamente es el método más barato y son diseñados para operar en ambas direcciones por medio del retorno de un resorte.

1.4.1.3. ACTUADOR HIDRAULICO

Este tipo de actuador es el que posee más potencia debido a los cilindros hidráulicos, siendo por lo tanto esta su principal ventaja. Entre las desventajas de los sistemas hidráulicos son su alto costo

de compra, de operación y las fugas constantes de aceite.

Naturalmente, un motor eléctrico, generalmente de inducción trifásico es utilizado para operar la bomba hidráulica, la cual tiene que estar constantemente en operación para mantener la presión adecuada del aceite, inclusive durante los períodos en que el robot no está en movimiento. Por consiguiente, posee un alto costo de operación, en contraste con un motor eléctrico que consume energía sólo cuando está produciendo movimiento. La fuerza motriz proviene enteramente de la acción del aceite sobre pistones o cilindros hidráulicos para dar movimiento lineal o angular.

En la tabla II se muestra las ventajas y desventajas de cada tipo de actuador.

<i>TIPO DE ACTUADOR</i>	<i>VENTAJAS</i>	<i>DESVENTAJAS</i>
	<ul style="list-style-type: none">• Costo inicial menor que un sistema hidráulico• Costo de operación menor que un sistema hidráulico• Limpio debido a	<ul style="list-style-type: none">• Relación de potencia-peso inferior al del sistema hidráulico. El motor paso a paso es el peor• Poca fuerza de retención al estar

<p>ELECTRICO</p>	<p>que no hay fugas de aceite por limpiar</p> <ul style="list-style-type: none"> • Se puede lograr un posicionamiento y control preciso de velocidad tipo seguimiento • Fácil de usar en ambientes frío o calientes • Funcionamiento rápido 	<p>detenido</p> <ul style="list-style-type: none"> • Requiere de frenos mecánicos • El servomotor ca es el más complicado drive eléctrico • Respuesta lenta para todos los drives • Requieren de engranes para mover la carga
<p>NEUMÁTICO</p>	<ul style="list-style-type: none"> • Costo inicial menor que un sistema hidráulico • Menor costo de operación que un sistema hidráulico • Limpio porque no existen fugas de aceite • Rápida respuesta • Puede ser muy exacto con paros mecánico 	<ul style="list-style-type: none"> • Imposible la programación de posicionamiento y control de velocidad • Movimiento limitado • Ruidoso • Se requieren paradas mecánicas • Poca capacidad de fuerza • Mediana fuerza de retención al detenerse

HIDRÁULICO	<ul style="list-style-type: none"> • Gran capacidad de fuerza necesario para manejar pesadas cargas • Gran fuerza de retención cuando se detiene • Se puede lograr un preciso control de velocidad y posición tipo servo • Rápida respuesta 	<ul style="list-style-type: none"> • Costo inicial alto • Alto costo de operación • Sucio debido a las fugas de aceite • Al arranque en frío, el sistema requiere un periodo de calentamiento • Exactitud es afectada en ambientes de calor o frío extremo

TABLA II. Ventajas y desventajas de los diferentes tipos de actuadores

1.4.2. MANIPULADOR

Se conoce como manipulador a toda la estructura del robot. Está hecho de una serie de arreglos de juntas, uniones, engranes, bandas, cadenas, pistones, cilindros, y demás elementos utilizados para satisfacer alguna configuración particular de robot. El manipulador es responsable de la capacidad de carga del robot, por lo que sus características de peso, fuerza y rigidez son factores importantes.

1.4.3. HERRAMIENTA FINAL

La herramienta final es la parte del robot que realiza el trabajo, para el cual fue diseñado. La mayoría de las tareas industriales requieren la transportación y la manipulación de herramientas para la realización de diversas operaciones. Estas tareas podrían requerir el levantamiento o agarre de objetos de diferentes formas y tamaños. Como es imposible diseñar un dispositivo herramienta que sea práctico y universal, por lo que, para mantener la flexibilidad del robot, éste podría ser ajustado con una variedad de *manos o pinzas*, cada una de ellas ajustable para el tipo de tarea que vaya a realizar en ese momento. La herramienta final puede ser diseñada para soldar, pintar y enganchar dependiendo de la tarea. La herramienta puede ser un dispositivo mecánico, electro-magnético, neumático o una combinación de ellos. En la figura 1.16, observamos diferentes tipos de pinzas, de acuerdo a su estructura mecánica (2).

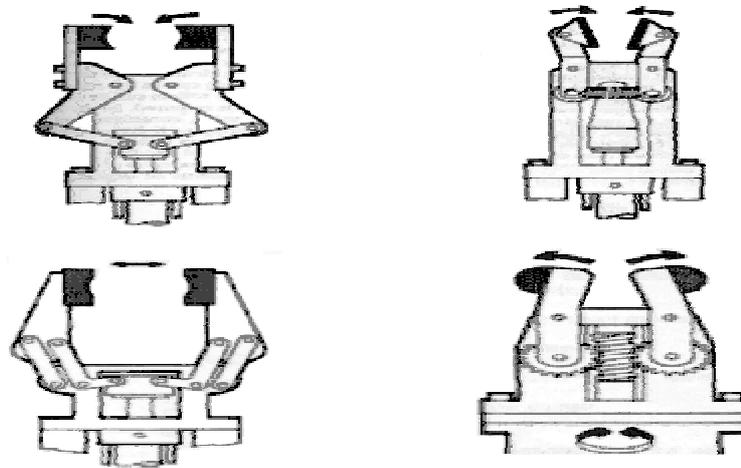


FIGURA 1.16. Tipos de pinzas con diferentes sistemas mecánicos

4. INTERFACE

La información que se transfiere entre dispositivos necesita ser compatible con las señales que el sistema de control puede aceptar, entender y transmitir. Los sistemas de control de robot son predominantemente digitales por lo que recibirá, procesará y transmitirá información en forma digital. Una colección de dispositivos llamados interface, necesitan ser facilitados para hacer la respectiva conversión y hacerlos compatibles. Generalmente, toda señal entrante que proviene de algún dispositivo externo o sensor, pasa a través de una interface.

1.4.5. SENSORES Y TRANSDUCTORES

Para controlar un actuador, la computadora necesita información sobre la posición y la velocidad del actuador, por lo que es necesario el uso de sensores y transductores para la medición de dicha posición y velocidad. Los sensores pueden ser simples como los sensores de proximidad, que sirven para detectar la presencia de un componente, hasta los más sofisticados sistemas captadores de imágenes (ver figura 1.17), capaces de identificar la forma y la orientación de varios objetos, lo que hacen del robot un mecanismo que sea capaz de tomar sus propias decisiones, de acuerdo a las condiciones en el ambiente de trabajo.

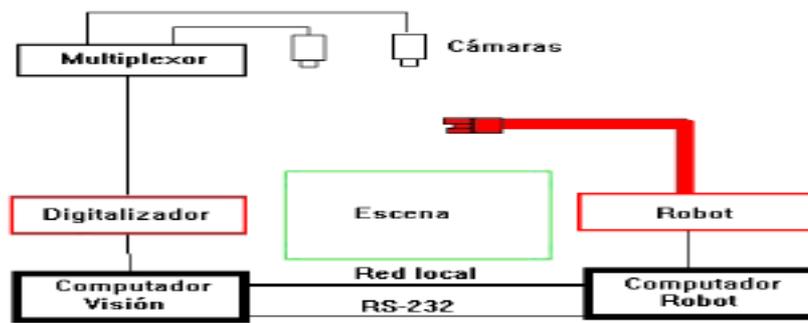


FIGURA 1.17. Sistema captador de imágenes para robot

La *posición* es una de las variables más importantes de controlar en los robots. Básicamente son dos los tipos de transductores para medir la posición en los RI. Tales transductores son:

- a. *Transductores angulares.*
- b. *Transductores lineales.*

Dichos transductores deben ser capaces de medir la posición o la distancia que se ha movido un objeto desde un punto de referencia, por lo que es necesario un transductor de posición en cada eje. Más adelante nos referiremos a los transductores de posición.

En sistemas de control de lazo cerrado (ver figura 1.11), los transductores facilitan la retroalimentación a los sistemas de control. Los transductores para medir velocidad y posición colocados en cada eje de movimiento, proporcionan un completo control de movimiento.

1. CODIFICADOR OPTICO DE POSICION

La idea de un disco giratorio para pasar y bloquear alternativamente un haz de luz a una celda fotovoltaica puede ser utilizado para medir la cantidad que ha girado un eje. En la figura 1.18, se muestra un engrane de diámetro grande en el eje medido, acoplado a un engrane de diámetro pequeño en el eje del disco. El disco óptico tiene muchas ranuras. Una fuente de luz estacionaria está montada a un lado del disco, colocándose una celda fotovoltaica directamente enfrente de ella al otro lado del disco, como está claro en la figura 1.18 (a). A medida que gira el disco, alternativamente pasa y bloquea la luz a la fotocelda debido a las ranuras, se produce un pulso de voltaje de fotocelda, como se indica en la figura 1.18 (b).

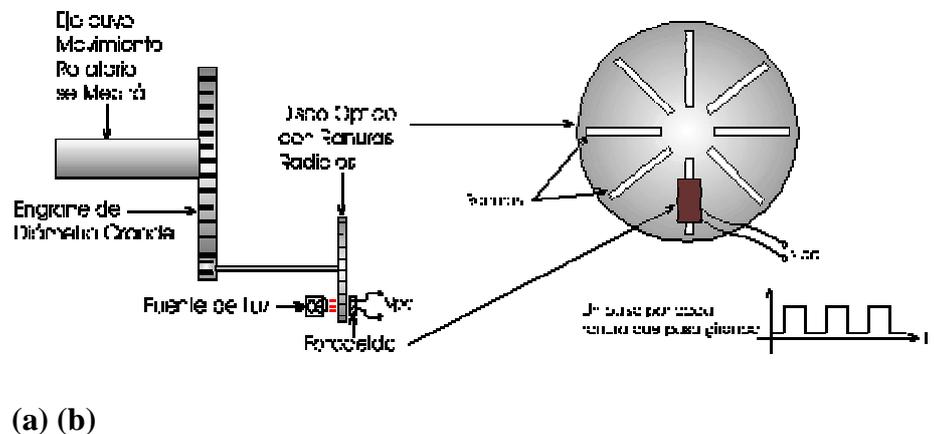


FIGURA 1.18. Disposición de un codificador óptico de posición a) Vista lateral b) Vista frontal

La forma de onda de V_{pc} es procesada por un circuito de acondicionamiento de señal, para hacerlo compatible a la

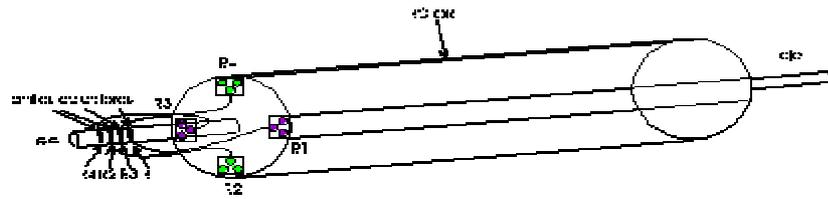
lógica TTL por ejemplo. Los pulsos son pasados a un contador binario. La lógica de control del sistema restaura a cero el contador cada vez que se inicie el movimiento del eje medido. Al comenzar a alejarse el eje de su posición anterior, el contenido del contador binario representará el movimiento angular que ha realizado (4).

2. RESOLVER

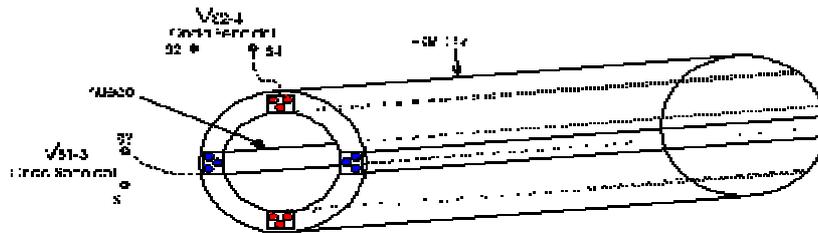
Un resolver es un dispositivo semejante a un generador, que puede utilizarse para la medición precisa de la posición angular de un eje. La estructura física de un resolver se muestra en la figura 1.19 de manera simplificada.

El estator se muestra en la figura 1.19 (a), es un cilindro hueco con dos devanados de múltiples vueltas. El devanado del estator 1-3 simbolizado como S_{1-3} al ser excitado por una onda senoidal crea un flujo magnético Φ_{1-3} también senoidal, el cual es perpendicular al plano del devanado y cambia de dirección de acuerdo al semiciclo positivo o negativo (4).

La estructura del rotor se muestra en la figura 1.19 (b). Existen dos devanados del rotor, R_{1-3} y R_{2-4} . Salen de la máquina a través de anillos colectores y las escobillas de contacto, por las terminales etiquetadas como R_1 , R_2 , R_3 y R_4 . Las escobillas de carbón no se muestran en la figura.



(a)



(b)

FIGURA 1.19. Estructura de un resolver de cuatro devanados

(a) Devanados del estator excitados por voltajes senoidales externos VS1-3 y VS2-4 (b) Devanados del rotor R1-3 y R2-4 desfasados 90°

Los devanados del estator actúan como el primario de un transformador, produciendo un flujo magnético que pasa a través de bobinas del rotor, las cuales actúan como el secundario del transformador.

Los voltajes de excitación de los devanados del estator del resolver son VS1-3 y VS2-4, y los voltajes inducidos en los devanados del rotor por la acción de transformación son simbolizados por VR1-3 y VR2-4. "Estos voltajes dependen de una manera muy precisa de la posición angular del rotor en relación al estator estacionario". Por tanto, midiendo VR1-3 y/o VR2-4 se puede obtener una indicación precisa de la posición del eje. Con el eje del rotor conectado al eje que está siendo medido en el sistema industrial, el resolver se vuelve un transductor preciso del movimiento angular de ese eje (4).

Si revisamos un libro de teoría electromagnética, encontraremos que:

$$V_{R1-3} = V_{S1-3} \times \cos \theta \quad (1-2)$$

$$V_{R2-4} = V_{S2-4} \times \sin \theta \quad (1-3)$$

suponiendo una relación de vueltas de rotor-estator de $n = 1.0$.

1.5. TIPOS DE CONTROL POSICIONAL

Otra manera de clasificar a los RI, es por el tipo de control posicional aplicado a cada movimiento de eje. Existen dos tipos de control posicional, los cuales son suficientes para clasificar a todos los equipos robóticos.

1.5.1. CONTROL PUNTO A PUNTO

Este tipo de control es empleado donde se necesite que la herramienta alcance puntos fijos en el menor tiempo posible. Para entender la operación, vamos a considerar las variadas formas para moverse entre dos puntos A y B. Lo explicaremos en dos dimensiones para una mejor visualización, pero los mismos argumentos pueden ser extendidos a tres dimensiones. En la figura 1.20 se muestra los tres posibles métodos (2).

1. METODO A

Quizás es el método más lento debido a que cada eje es energizado separadamente. Esto tiene su ventaja ya que el sistema de control no será muy complicado debido a que no se requiere el movimiento coordinado de los ejes; figura 1.20 (a).

2. METODO B

Definitivamente es el más rápido camino entre puntos porque

se elige la más corta trayectoria. Este método implica el uso de sofisticados sistemas de control para coordinar la velocidad de cada eje a fin de mantener la línea recta. El software de control aplicado a este método comprende subrutinas de interpolación; ver figura 1.20 (b).

3. METODO C

Es el método más comúnmente utilizado. Ambos ejes al principio se mueven simultáneamente a alta velocidad formando una trayectoria de 45° y al final sólo se mueve un eje. Los requerimientos de control son relativamente simples sin sacrificar demasiado la velocidad; ver figura 1.20 (c).

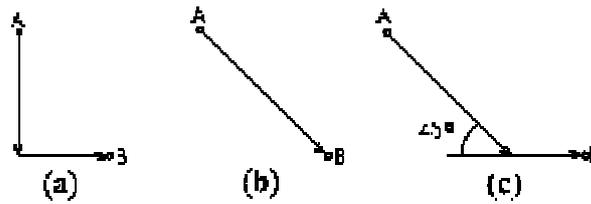


Figura 1.20. Tres métodos para desplazarse entre dos puntos

1.5.2. CONTROL DE TRAYECTORIA CONTINUA

El método por el cual el sistema de control se mueve desde un punto a otro cuando se combina con el movimiento de ejes, se conoce como *interpolación*, la cual sirve para calcular puntos intermedios entre los puntos inicial y final. Cuando el sistema de control produce que los ejes se muevan a través de estos puntos intermedios, se logrará un movimiento mucho más natural, hecho importante para realizar tareas como el pintado de piezas, por ejemplo. Mientras más puntos intermedios son programados, se obtendrá un movimiento más suave. Existen tres tipos de interpolación: lineal, circular y parabólica. La mayoría de los sistemas de control de robot utilizan interpolación lineal y circular, pocos controles usan interpolación parabólica (2).

CAPITULO II

DISPOSITIVOS Y CIRCUITOS ELECTRICOS

1. INTRODUCCION

Los motores eléctricos son, probablemente, el método más universal para la conversión de energía de una fuente a otra. Los sistemas hidráulicos y neumáticos simplemente describen el uso de un motor eléctrico para manejar compresores y las bombas hidráulicas hacen que estos sistemas entren en operación.

Los motores descritos en este capítulo son algunos de los motores que serían encontrados en robots de tipo industrial. Un tipo de motor es el motor de paso, los cuales son usados por robots para fines didácticos que demuestran los conceptos básicos de robótica.

El otro motor es un pequeño motor trifásico de corriente alterna (CA),

que aunque no se encuentra en forma directa en el control del brazo de robot, si está presente para el manejo de los sistemas hidráulicos y neumáticos.

En este capítulo nos referiremos a estos dos tipos de motor. Nombraremos sus principales características y explicaremos los diferentes circuitos de control aplicados a dichos motores.

Analizaremos el circuito temporizador utilizando el 555 y su configuración estable.

Además haremos un pequeño estudio de interruptores de límite; del potenciómetro, como transductor para sensar la posición y de las fuentes de poder utilizadas.

2. MOTOR DE PASO

El motor de paso, llamado también *motor de conmutación electrónica*, es apropiado para mover gradualmente el eje del motor un ángulo previamente determinado.

Un motor de paso es un dispositivo electromecánico, que convierte pulsos eléctricos en movimientos mecánicos discretos. La secuencia de pulsos aplicados está directamente relacionado con la dirección de rotación del eje del motor.

La velocidad del eje de rotación del motor está directamente relacionado con la frecuencia de los pulsos de entrada, y la longitud de rotación está directamente

relacionada al número de pulsos aplicados a la entrada del motor.

En la figura 2.1, se puede observar el motor de paso que se utilizó para diseñar el sistema de control.



FIGURA 2.1. Motor de paso utilizado

2.2.1. CARACTERISTICAS

Los motores de paso son diferentes a los demás motores de corriente directa (CD), porque no tienen escobillas ni conmutador mecánico. El rotor no tiene devanado de armadura, en su lugar existe una colección de imanes permanentes salientes. Las principales clases de motores de paso, los conceptos básicos, su estructura interna, y las más esenciales características, se podrán apreciar en el Anexo A.

En la tabla III se encuentran las características principales de los datos de placa del motor de paso utilizado.

Motor	Step-Syn Stepping Motor
Tipo	103G770-2511 IBM PN 6328842
Fabricante	Sanyo Denki Co. Ltd.
Voltaje (V)	4.1
Corriente (A)	1.1
Impedancia por fase (ohm)	7.5
Grados/Pasos	1.8
Clase	B
Diámetro (mm)	56.5
Diámetro del eje (mm)	6.5
Número de Fases	2
Longitud (mm)	50

Tabla III. Datos de placa del motor de paso

2.2.2. TIPOS DE CONTROL

Los modos de control más comunes son los siguientes:

- **Conducción a un paso** (solamente una fase está energizada). Ver Anexo A.

- **Conducción de paso completo** (2 fases están energizadas).
- **Conducción de medio paso** (1 y 2 fases están alternadamente energizadas). Ver Anexo A.
- **Micropasos** (Variando continuamente la corriente del motor).

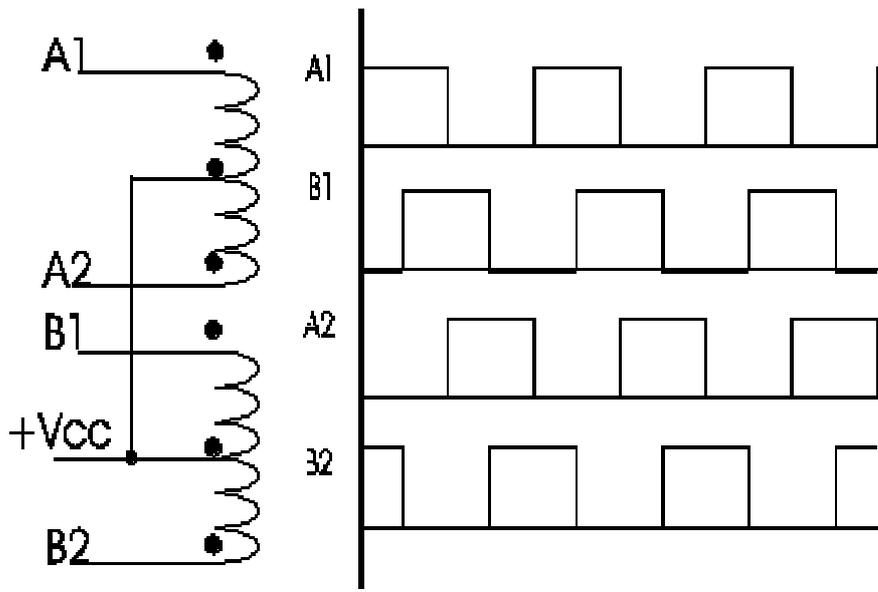


FIGURA 2.2. Secuencia de los pulsos de control para conducción de paso completo.

Para nuestro proyecto nos referiremos solamente al *modo de conducción de paso completo*, el cual se basa en la alimentación de dos bobinados a la vez. Observar la figura 2.2.

En la Tabla IV se encuentra la secuencia de salida de los pulsos de control para manejar el motor de paso.

A1	1	0	0	1	1	0	0	1	1
A2	1	1	0	0	1	1	0	0	1
B1	0	1	1	0	0	1	1	0	0
B2	0	0	1	1	0	0	1	1	0

TABLA IV. Secuencia de los pulsos de voltaje de CD en los respectivos devanados del motor de paso

2.2.3. CIRCUITO MANEJADOR (DRIVER)

Para diseñar el circuito de control, que va a generar la secuencia correcta de pulsos para excitar las bases de los transistores, los cuales van a actuar sobre cada fase del motor de paso, se recurrió a la lógica TTL. Utilizamos el 74LS194, el cual es un registro universal bidireccional de 4 bits. En la figura 2.3 se encuentra el diagrama esquemático completo del circuito manejador del motor de paso.

Para obtener el modo de control de paso completo (dos fases energizadas) para el motor de paso, nos basamos en este registro de desplazamiento universal (74LS194), del cual se usó los 4 bit de salida, para el primer instante de tiempo. Al principio, se hace cargar los 4 bit de datos (DA y DB

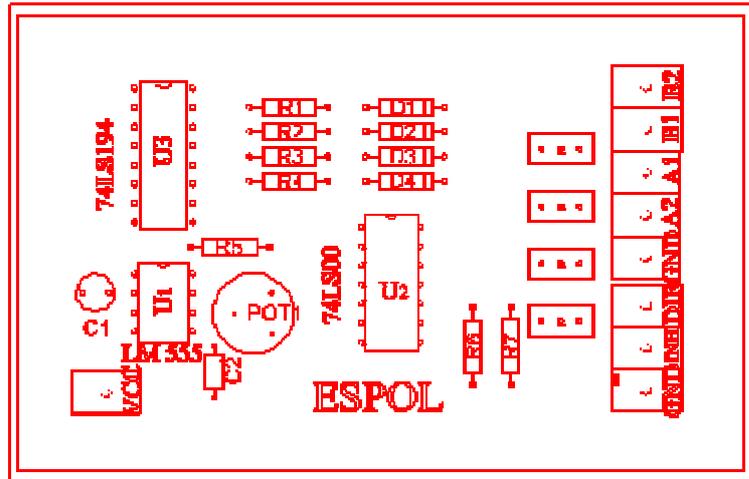
están en alto, DC y DD están en bajo) en paralelo, con S0 y S1 en alto, obteniendo que las salidas QA y QB pasen a un alto, y a la vez, las salidas QC y QD pasen a un bajo.

En la Tabla V se puede observar el modo de operación del registro.

S0	S1	MODO DE OPERACIÓN
0	0	Inhibido
0	1	Giro a la derecha
1	0	Giro a la izquierda
1	1	Carga datos en paralelo

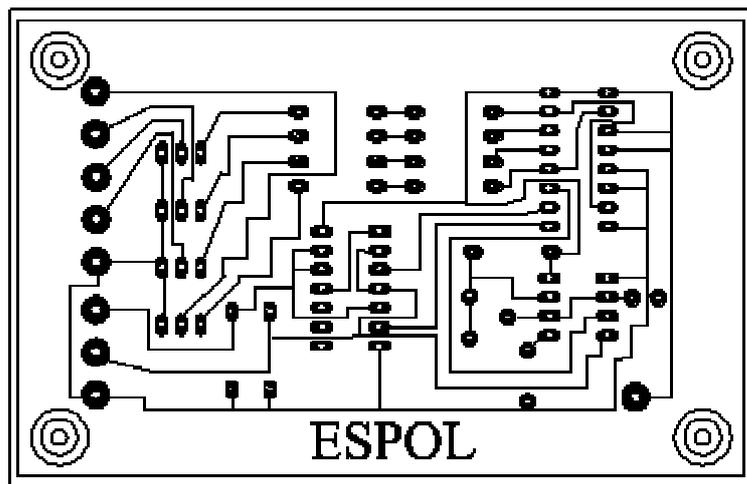
TABLA V. Modo de operación del registro de desplazamiento bidireccional 74LS194

La siguiente operación depende de la decodificación del modo de S0 y S1, los que comandarán el desplazamiento hacia la derecha, o hacia la izquierda. Si S0 se encuentra en alto, y S1 está en bajo, se obtendrá un desplazamiento hacia la derecha de dos bits en alto, y cada bit tiene una duración de 90 grados. Por lo contrario, si S0 es un bajo, y S1 es un alto, los bits tendrán un desplazamiento a la izquierda.



La figura 2.4 (a)

muestra la vista superior de la tarjeta que se utiliza para controlar al motor de paso; y en la figura 2.4 (b) se muestra las pistas que conforman el circuito impreso de dicha tarjeta.



(a)

(b)

FIGURA 2.4. (a) Vista superior de la tarjeta controladora del motor de paso

(b) Circuito impreso

1. MOTOR TRIFASICO

Los motores de corriente alterna no son usados directamente para el control de un brazo de robot, pero todos los demás sistemas tienen como fuente de poder primaria un motor de este tipo. Esto es verdadero porque los robots no eléctricos, tienen sus sistemas hidráulicos y neumáticos manejados por motores de corriente alterna (CA).

Los motores de inducción también llamados motores de corriente alterna, tienen corrientes de este tipo tanto en los devanados del estator como del rotor. En este tipo de motor se suministra la corriente alterna directamente al estator; y por inducción, es decir, por acción-transformador, al rotor. Se puede considerar a la máquina de inducción como un transformador generalizado en el cual se transforma la energía eléctrica entre el estator y el rotor, al mismo tiempo que ocurre un cambio de frecuencia y un flujo de energía mecánica.

En el motor de inducción, el devanado del rotor está en cortocircuito eléctrico y con frecuencia no tiene conexiones al exterior; las corrientes se inducen en él por la acción-transformador procedente del devanado del estator. El flujo de armadura en el motor de inducción adelanta al del rotor y produce un par electromagnético. Es esencial reconocer que los flujos del rotor y del estator giran en sincronismo entre sí, y que este se relaciona con el desplazamiento relativo entre ellos. Pero el rotor no gira en forma sincrónica; es el desplazamiento del rotor a través del flujo sincrónico de la armadura lo que da lugar a las corrientes inducidas en el rotor, y por lo tanto al par. Un motor de inducción trabaja a una velocidad poco menor que la velocidad mecánica sincrónica.

Los motores industriales de corriente alterna generalmente son trifásicos, cuyos voltajes más comunes son 208, 220 y 440 voltios.



FIGURA 2.6. Motor trifásico utilizado

En la figura 2.6 se puede observar el motor trifásico utilizado.

2.3.1. CARACTERISTICAS

Los motores de corriente alterna no son tan adecuados para las aplicaciones donde se requiera un control de velocidad, como los motores de corriente directa, debido a que su velocidad no puede controlarse mediante la variación del voltaje de alimentación. La disminución del voltaje de alimentación para un motor de inducción de 60 Hz trifásico, lógicamente disminuirá su velocidad, pero será incapaz de mantener una velocidad de eje estable ante la demanda de par impuesta por la carga mecánica.

Para controlar la velocidad del motor de ca, se varía simultáneamente la frecuencia y el voltaje de alimentación, de tal forma que la relación, V/f , sea constante. La razón por la cual esta razón debe permanecer constante, se debe a que la intensidad del campo magnético del rotor debe permanecer

también constante, bajo todas las condiciones de operación. Si la intensidad del campo del estator aumenta muy por encima del valor de diseño, el material del núcleo del motor entrará en saturación magnética. Esto disminuirá la permeabilidad del núcleo, inhibiendo, por tanto, la inducción adecuada de voltaje y corriente en las mallas del rotor, denigrando la capacidad de producción de par del motor. Por otra parte, si la intensidad del campo del estator cayera muy por debajo del valor de diseño, el campo magnético disminuido inducirá valores menores de voltaje y corriente en las mallas del motor, de acuerdo con la ley de Faraday. Esto igualmente disminuye la capacidad de producción de par del motor (4).

Luego, el campo magnético senoidal producido por los devanados de estator, debe mantener un valor rms constante, sin importar la frecuencia. Para lograr esto, debemos de tener muy en cuenta a la corriente de magnetización del estator. La corriente de magnetización de un motor de inducción, es la corriente que fluye a través de los devanados del estator, cuando el rotor está girando a la velocidad de estado estable sin par de carga. La corriente de magnetización de un motor de inducción está dada por la ley de Ohm,

$$I_{mag} = V / X_L \quad (2-1)$$

Donde:

V = valor rms del voltaje aplicado del estator,

X_L = reactancia inductiva del devanado del estator.

En la ecuación (2-1), X_L no permanece constante a medida que se ajusta la

frecuencia de la alimentación. Varía en proporción con la frecuencia, según la ecuación (2-2).

$$X_L = 2\pi f L \quad (2-2)$$

Por tanto, V también debe variarse en proporción con la frecuencia, para que la operación de división de la ley de Ohm arroje un valor invariable de corriente de magnetización.

Alternativamente, usando $X_L = 2\pi f L$, podemos reescribir la ecuación (2-1) como

$$I_{\text{mag}} = V/X_L = V/2\pi f L = 1/2\pi L (V/f) \quad (2-3)$$

En la ecuación (2-3), el factor $1/2\pi L$ es una constante, determinada por los detalles de construcción del motor que tienen una relación con la inductancia del devanado de estator. En la ecuación (2-3), es claro que puede lograrse una I_{mag} constante sólo manteniendo una razón V/f constante.

Existen dos métodos básicos para producir una fuente trifásica de frecuencia variable y alta potencia, para el ajuste de velocidad de un motor de inducción de corriente alterna.

Dichos métodos son:

- Convertir una fuente de corriente directa en corriente alterna y obtener

una frecuencia deseada. Un circuito que realiza esto se denomina *inversor*.

- Convertir una fuente trifásica de corriente alterna de 60 Hz, en una fuente trifásica de menor frecuencia. Tal circuito se denomina *cicloconvertidor*, incluyendo los convertidores que proveen de una frecuencia de salida que un múltiplo entero de la frecuencia de la fuente. El múltiplo entero está relacionado con el número de fases de la fuente, empleándose la conmutación por línea.
- Si en el cicloconvertidor se emplea la conmutación por carga, el convertidor puede suministrar una frecuencia de salida que puede ser un múltiplo no entero de la frecuencia de la fuente. Este convertidor también es llamado *cicloinversor*.

Motor	CA
Tipo	1BS 1845-4TA V22411-S-D2 Nr 23987733
Número de Fases	3
Fabricante	Siemens
Voltaje (V)	15 Max.
Corriente (A)	1.2 Max.
Frecuencia (Hz)	200 Max.
Impedancia por fase (ohm)	6.6

TABLA VI. Características principales del motor trifásico

En la Tabla VI se muestran los parámetros del motor trifásico que utilizamos, los mismos que fueron determinados en el laboratorio debido a

que no teníamos los datos de placa del fabricante.

2.3.2 INVERSOR TRIFASICO

Los convertidores de voltaje de entrada de corriente directa a voltajes simétrico de corriente alterna se conocen como *inversores*. Los inversores tienen la facilidad de variar, o fijar, el voltaje de salida, o la frecuencia. Al aumentar el voltaje de entrada de corriente directa, y si la ganancia del inversor (relación entre el voltaje de salida de corriente alterna y el voltaje de entrada corriente directa) se mantiene constante, es posible también aumentar el voltaje de salida de corriente alterna. Pero, si el voltaje de entrada es fijo, se puede obtener un voltaje de salida variable, si se modifica la ganancia del inversor, generalmente esto se hace posible mediante la modulación de ancho de pulso (PWM).

Para aplicaciones de media y baja potencia se puede aceptar voltajes de onda cuadrada. En cambio, para aplicaciones de alta potencia son necesarios los voltajes en forma de onda senoidales de baja distorsión.

Los inversores se pueden clasificar en dos tipos:

- Inversores monofásicos.
- Inversores trifásicos.

Cualquiera de los dos, pueden utilizar dispositivos con activación y desactivación controlada (BJT, MOSFET, IGBT, MCT, SIT, GTO), o

tiristores de conmutación forzada.

Para nuestro caso se optará por el inversor trifásico debido a las condiciones propias del motor, y utilizaremos transistores.

El inversor trifásico tiene dos modos de conducción de los pulsos de control que son:

- Conducción a 120°
- Conducción a 180°

3. INVERSOR TRIFASICO CONDUCCION A 120°

La operación cíclica del circuito en puente de transistores, como observamos en la figura 2.7, se encuentra dividida en seis intervalos de igual duración, como vemos en la figura 2.8 (a). Definamos primero nuestro primer intervalo de tiempo, para que sea el momento en que el circuito digital y electrónico de control, haga que los transistores T1 y T2 se enciendan, mientras los demás están apagados. Esta condición se indica en la primera columna de la figura 2.8 (b).

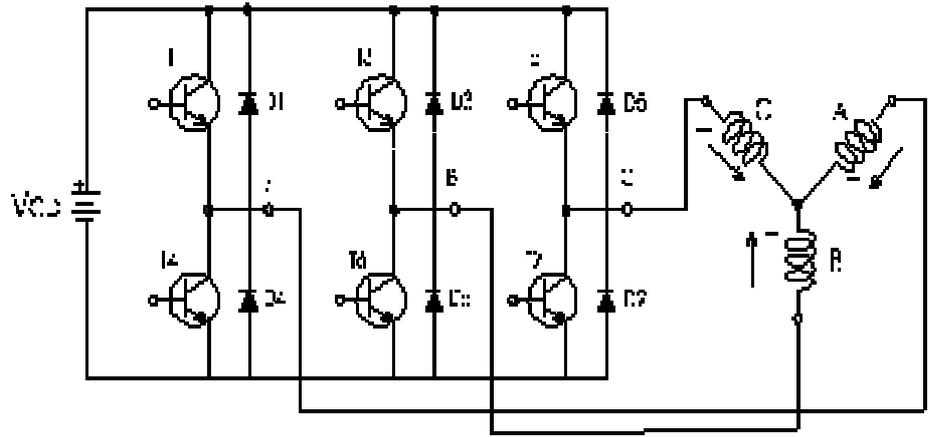


FIGURA 2.7. Circuito de inversor trifásico, configuración de seis transistores y seis diodos.

Con T1 y T2 encendidos, hay una trayectoria de flujo de corriente desde la terminal positiva de la fuente de corriente directa, a través de T1, a través del devanado A del estator en la dirección positiva, por medio del devanado B del estator en la dirección negativa, a través de T2 y por la línea de terminal negativa de fuente de corriente directa. No existe corriente en el devanado C del estator al momento, porque T3 o T6 están apagados. El detalle de los seis intervalos que corresponden a un ciclo completo del inversor se encuentra tabulado en la tabla VII.

Número del Intervalo	1	2	3	4	5	6
Devanados manejados y direcciones	+A, -C	+B, -C	+B, -A	+C, -A	+C, -B	+A, -B
Transistores encendidos	T1, T2	T2, T3	T3, T4	T4, T5	T5, T6	T6, T1

TABLA VII. Condiciones de cada uno de los seis intervalos de tiempo que comprenden un ciclo completo de salida del inversor.

En la figura 2.9, se ve que los componentes de campo magnético vienen de 60° debido a la corriente positiva de A, y de 120° debido a la corriente negativa de C. Estos componentes se combinan para producir un campo magnético neto desde la posición mecánica de 90° en el estator. Entonces, la conmutación de los transistores a medida que procedemos del primero al segundo intervalo de tiempo, ha producido un giro de 60° del campo neto del estator (de la posición de 30° a la posición de 90°). Se puede seguir la trayectoria de corriente en los devanados del motor en la figura 2.7. Los nuevos componentes de campo magnético resultantes de estas corrientes de devanado provocan otro giro de 60° del campo neto del estator, a la posición de 150° como se muestra en la figura 2.9.

Continuando con este proceso por el tercer, cuarto, quinto y el sexto intervalo de tiempo, los transistores se activan y se apagan de acuerdo al programa indicado en la tabla VII, y como se observa en la figura 2.8 (a). Esta secuencia de combinaciones de los transistores produce las formas de onda de corriente en el devanado de estator, dibujadas en la figura 2.8, que hace que el campo magnético neto del estator continúe avanzando en saltos de 60° . De esta manera se produce el efecto de campo giratorio de una línea de campo de corriente alterna trifásica.

El par del motor no es instantáneamente constante, como lo sería si el motor fuera manejado por una onda senoidal trifásica. Pero, tampoco es tan abrupto, como pueden sugerir las formas de onda de la figura 2.8, ya que la inductancia de los devanados del motor tiende a suavizar los ángulos rectos de las formas de onda de corriente.

Los voltajes de línea a línea V_{AB} , V_{BC} y V_{CA} también están desfasados 120° entre ellos, como se observa claramente en las tres formas de onda en la figura 2.8 (c). También se puede establecer que los voltajes de línea adelantan a los voltajes de fase en 30° , igual que para la de corriente alterna de onda senoidal trifásica.

Las formas de onda de voltaje de línea pueden derivarse restando una forma de onda de fase de otra forma de onda de fase. Por ejemplo, V_{AB} es derivado de restando la forma de onda V_B de la forma de onda V_A .

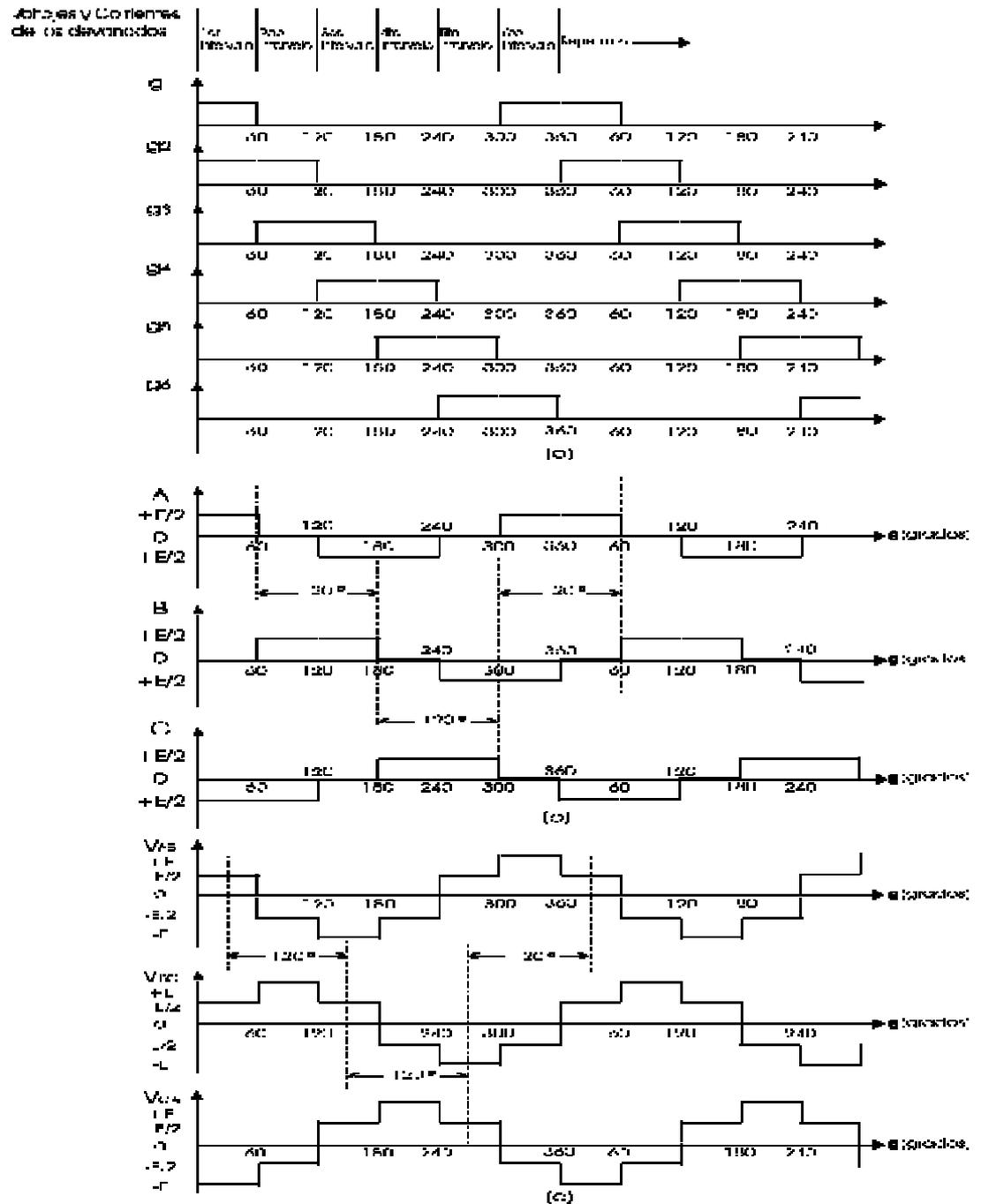


Figura 2.8. (a) Señales de base de los transistores para conducción a 120 grados
(b) Formas de onda de voltaje de línea a neutro idealizadas
(c) Formas de onda de voltaje de línea a línea idealizadas

Como siempre, la resta es equivalente a la inversión de signo y la suma. Por tanto, durante el primer intervalo de tiempo, el valor instantáneo de V_{AB} se obtiene cambiando el signo del valor instantáneo de V_B (que es $-E/2$),

arrojando $+E/2$, sumando entonces al valor instantáneo de V_A ($+E/2$). El resultado es $+E/2 + E/2 = +E$, que está de acuerdo con el primer intervalo de la forma de onda V_{AB} . Los demás intervalos de los demás voltajes de línea pueden derivarse de la misma manera.

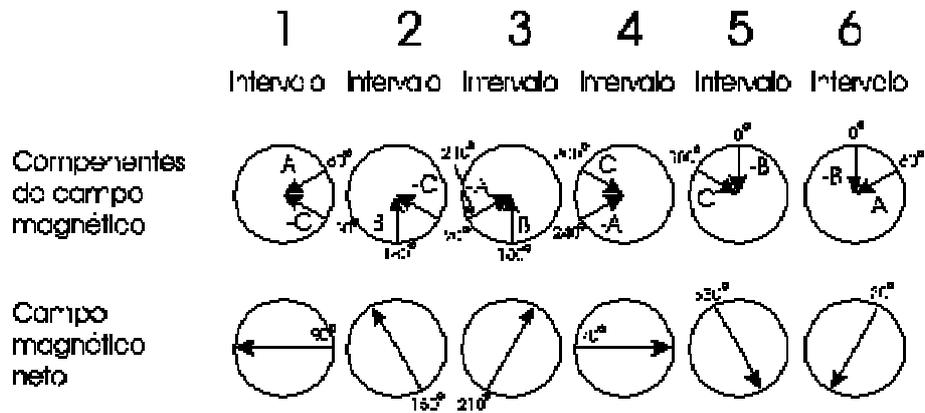


FIGURA 2.9. Componentes de los campos magnéticos y del campo magnético neto durante cada uno de los seis intervalos de tiempo.

2.3.4. CIRCUITO DE CONTROL

Para el diseño del circuito de los pulsos de control se utilizó la lógica TTL, fundamentado específicamente en el registro universal de 8 bits, el 74LS198.

Las características del circuito del registro son:

- Entradas en paralelo
- Entradas en serie
- Salidas con desplazamiento a la derecha o a la izquierda.
- Modo de selección de entradas.

- Entrada directa de apagado.

El registro tiene 4 modos de operación:

- Carga paralelo
- Desplazamiento a la derecha (en la dirección QA hacia QF)
- Desplazamiento a la izquierda (en la dirección QF hacia QA)
- Reloj inhibido (No realiza acción alguna)

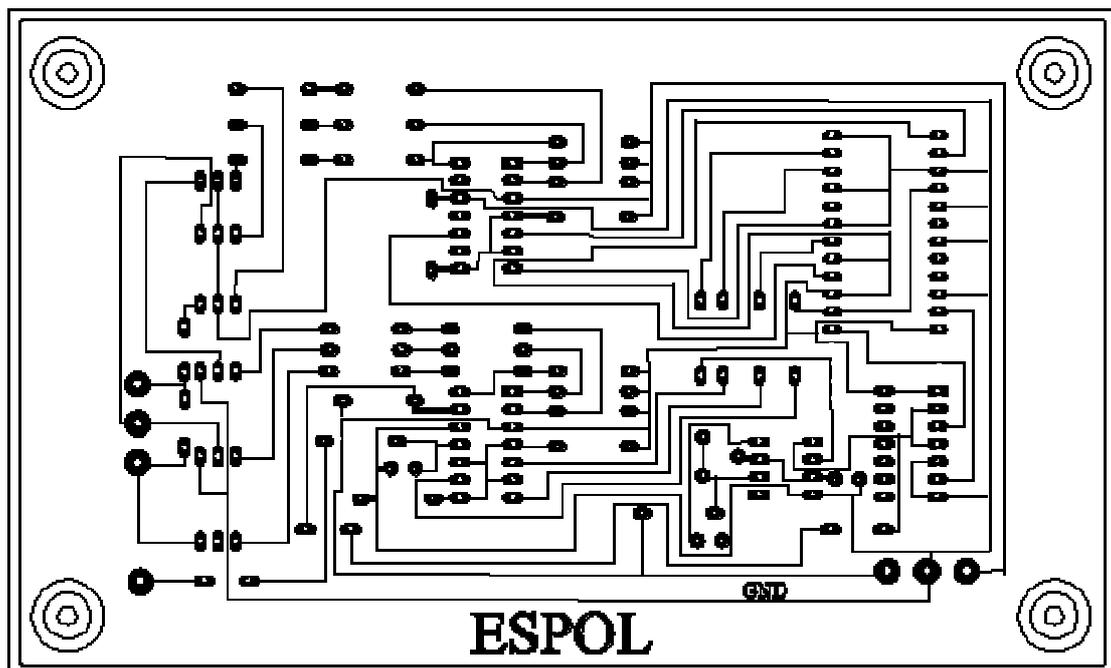
La carga sincrónica en paralelo del registro está acompañada por la aplicación de los 8 bits de datos y tomando los modos de control S0 y S1 en alto. Los datos son cargados en un grupo de flip-flop y aparecen a la salida después de la transición positiva de un flanco de reloj. Durante la carga, el flujo de datos seriales está inhibida.

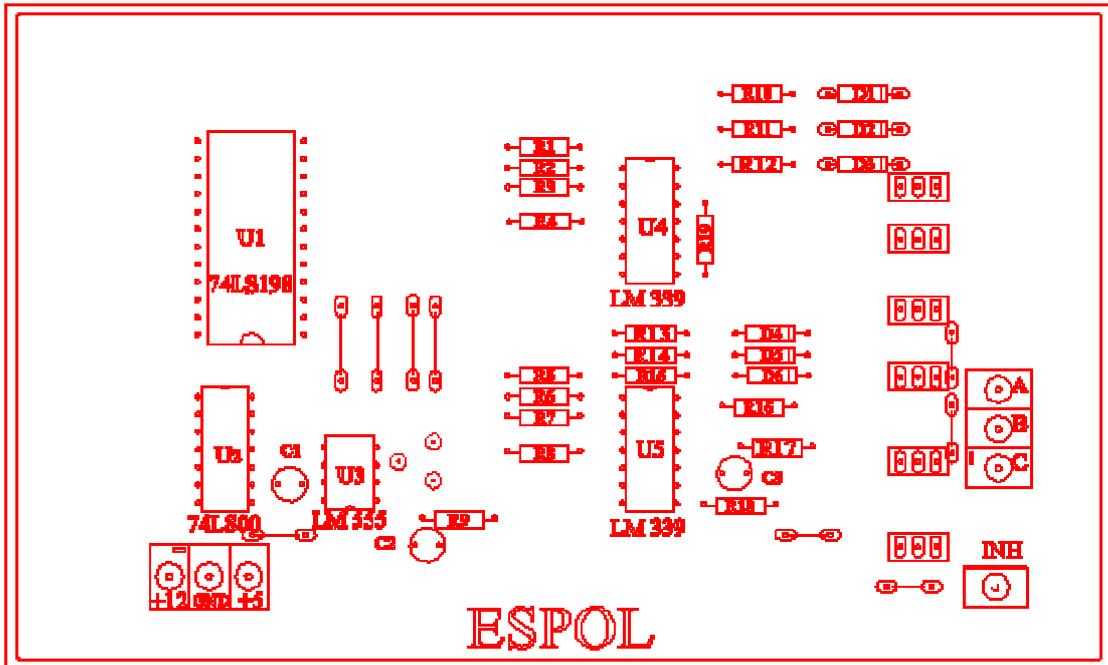
El desplazamiento a la derecha está acompañado sincrónicamente con el flanco del pulso de reloj, cuando S0 es un alto y S1 es un bajo. Los datos seriales para este modo están ingresando por la entrada de dato de desplazamiento hacia la derecha. Cuando S0 es bajo y S1 es alto, y con un flanco de reloj positivo entonces ocurre un desplazamiento hacia la izquierda, y los datos nuevos están ingresando por la entrada de desplazamiento hacia la izquierda.

El reloj de los flip-flop es inhibido cuando ambos modos S0 y S1 son bajos.

Debido a la facilidad del registro de desplazamiento universal, sólo necesitamos 6 bits de salida de los 8 que posee (ver figura 2.10), para el primer intervalo de tiempo hacemos que se cargue la entrada de bit de datos (DA y DB están en alto y DC, DD, DE y DF están en bajo) en paralelo, con S0 y S1 en alto, logrando de esta manera que las salidas QA y QB permanezcan en alto y QC, QD, QE y QF estén en bajo. Los modos de operación varía, con S0 en un alto y S1 en un bajo, obteniendo de esta forma un desplazamiento hacia la derecha de dos bits en alto, cada bit tiene una duración de 60 grados. Toda esta operación está diseñada en forma secuencial automática.

En la figura 2.11 (a), se muestra la vista superior de la tarjeta controladora del motor trifásico. En la figura 2.11 (b), se puede observar el circuito impreso de esta tarjeta.





(a)

(b)

**FIGURA 2.11. (a) Vista superior de la tarjeta controladora del motor trifásico
(b) Circuito impreso**

3. INTERRUPTOR DE LIMITE

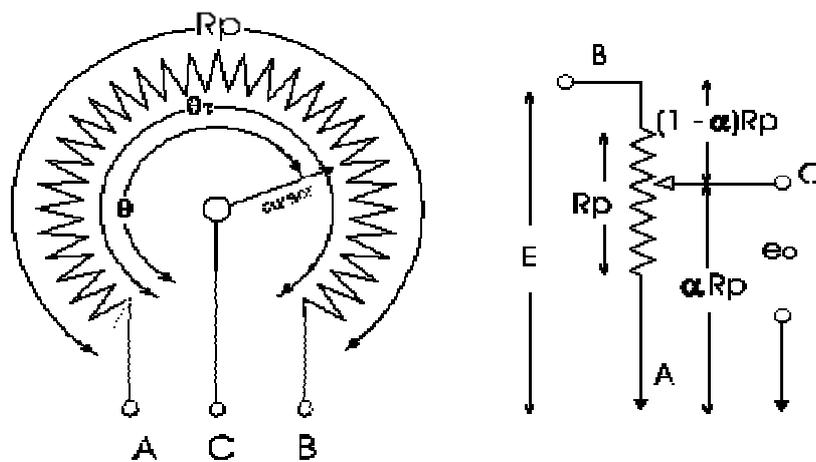
El interruptor límite, como su nombre lo indica, tiene como misión limitar, alguna operación de la máquina, o de un motor impulsor. El tipo más conocido es el interruptor fin de carrera, utilizado para limitar el desplazamiento de una herramienta de corte, o mesa, u otra parte de la máquina. Cuando la herramienta llega a una posición extrema predeterminada, activa un interruptor fin de carrera, haciendo que se detenga el motor y la máquina hasta que hayan sido efectuadas las correcciones necesarias por el operador.

Para evitar que el eje se sobrepase de su posición límite, el programa tiene una previsión para esta operación en la secuencia. Se deben colocar topes mecánicos, cuya función será la de actuar sobre los interruptores límites, los cuales enviarán una señal eléctrica discreta de entrada al PLC, cuando en algún movimiento de la articulación del brazo del robot llegue a la posición extrema de su desplazamiento.

2.5. DISPOSITIVO PARA MEDIR LA POSICION ANGULAR

Potenciómetros de Precisión

Un potenciómetro de precisión es un transductor electromecánico cuya salida



es una función

del ángulo de eje θ , y el voltaje de excitación E . Un diagrama esquemático está mostrado en la figura 2.12.

a. (b)

FIGURA 2.12. Potenciómetro de precisión (a) Potenciómetro de giro

simple (b) Diagrama esquemático del potenciómetro sin carga

La resistencia total del potenciómetro, R_p , es medido entre las terminales del voltaje de excitación A y B. El barrido de la rotación total es θ_T radianes, y es un poco menor que 2π radianes (360 grados). El brazo rota con el eje, por lo tanto, el ángulo de rotación θ radianes está variando entre 0 y θ_T . El voltaje de salida, e_o , es medido entre los terminales A y C. La función de transferencia para un potenciómetro sin carga es:

$$e_o = E \frac{\theta}{\theta_T} = K_p \theta \quad (2-4)$$

A no ser que el voltaje de excitación, E , esté variando, la función de transferencia, K_p , tiene un valor constante medidos en voltio por radianes.

La figura 2.12 (b), muestra el diagrama esquemático para un potenciómetro. Simbolizado en el diagrama está el factor de rotación total, α , que el barrido ha movido desde la tierra de referencia en A hacia B. El voltaje de referencia puede ser expresado como

$$e_o = E \alpha \quad (2-5)$$

R_p

o

$$e_o = E\alpha \quad (2-6)$$

El factor α puede ser declarado como

$$\alpha = \theta / \theta_T \quad (2-7)$$

Si en las terminales de salida se coloca una carga resistiva, R , los flujos de corriente entran a la carga. El voltaje de salida del potenciómetro con carga, e'_o , es menor que el potenciómetro sin carga, para el mismo ángulo de rotación.

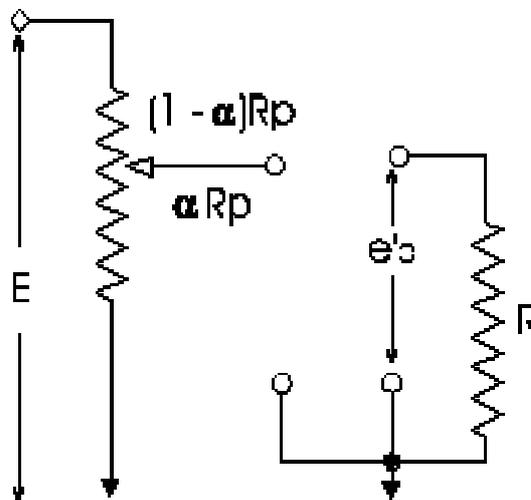


FIGURA 2.13. Potenciómetro con carga

Como observamos en la figura 2.13, se demuestra que el voltaje de salida, e_o , está determinado por la regla del divisor de voltaje. De lo que se obtiene que la resistencia de carga R está en paralelo con R_p . Por lo tanto,

$$R_p // R$$

$$e'_{o} = E \quad (2-8)$$

$$\alpha R_p // R + (1 - \alpha) R_p$$

o

$$E$$

$$e'_{o} = \quad (2-9)$$

$$1 + (R_p // R) \alpha (1 - \alpha)$$

Ahora existe un error que es máximo para un valor de $\alpha = 2/3$. Para una operación razonable, la resistencia de carga efectiva debería estar alrededor de $100 R_p$.

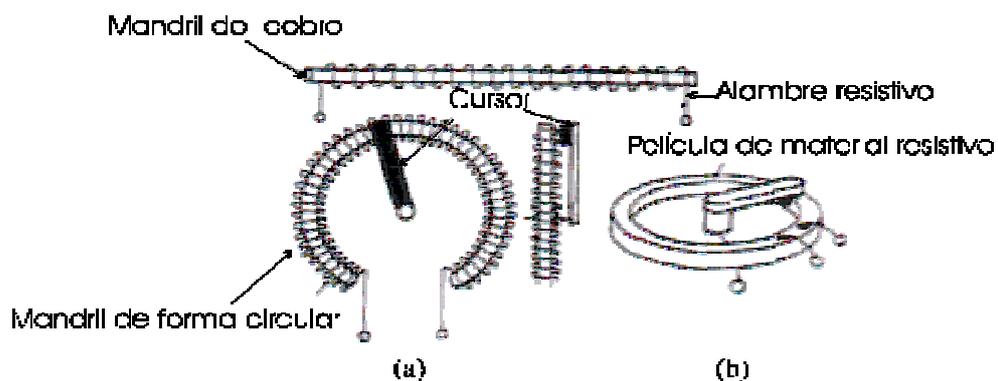


FIGURA 2.14. Dos clases de potenciómetro (a) Potenciómetro de alambre bobinado (b) Potenciómetro de película

Aquí están dos técnicas usadas para crear físicamente resistencias entre los terminales de excitación. Estos métodos son ilustrados en la figura 2.14. El mandril de cobre provee un soporte para las vueltas de alambre resistivo, y un aislamiento de laca que previene cortos en la resistencia. El cursor se desliza por el lado superior del bobinado de resistencia.

Debido a que el cursor del potenciómetro no se desliza a lo largo de un alambre continuo. El voltaje de salida puede estar representado como una sucesión de pasos, como mostramos en la figura 2.15. La resolución está definida como un pequeño incremento de voltaje que puede ser sentido y expresado como un factor de voltaje de excitación, así:

$$\Delta E$$

$$\text{resolución} = \frac{\Delta E}{E} \quad (2-10)$$

$$E$$

Donde, ΔE , es un pequeño incremento de voltaje. Teóricamente, ΔE , depende del número de vueltas (n) del alambre de resistencia en el mandril, así que

$$E / n$$

$$\text{resolución} = \frac{E}{n} \quad (2-11)$$

En

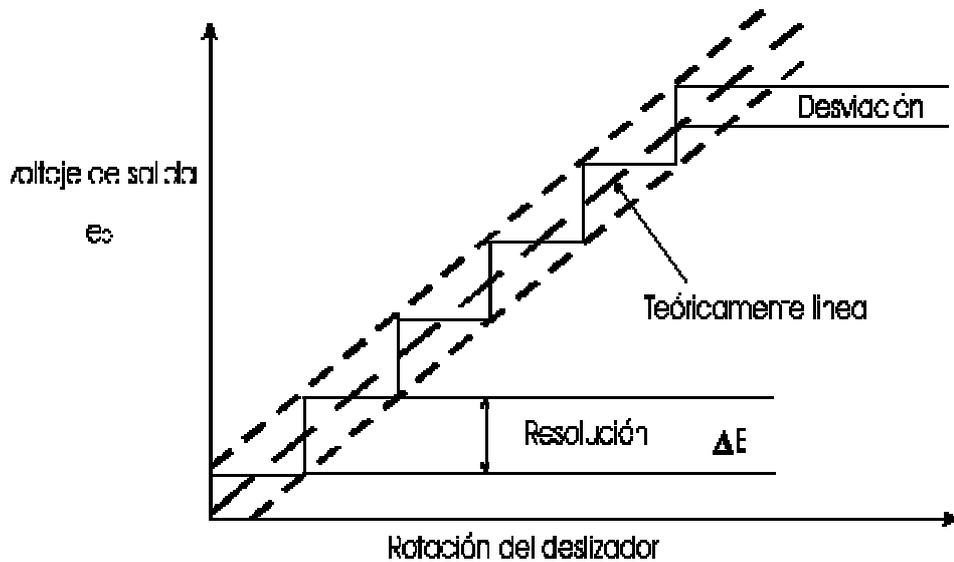


FIGURA 2.15. Pasos de salida de voltaje de un potenciómetro de alambre bobinado

Un potenciómetro lineal se lo considera así, debido a que la resistencia lineal por unidad de longitud, o por unidad angular es constante. Para vencer el problema de los pasos de voltaje, fueron desarrollados potenciómetros con película de material resistivo. Estos, teóricamente, tienen infinita resolución porque el cursor no hace saltos de una vuelta a la siguiente vuelta. Sin embargo, puede haber problemas de depósitos de película no lineal. Estos potenciómetros tienen una limitada disipación de calor, contrario al tipo de alambre devanado, donde el mandril de cobre actúa como sumidero de calor y radiador.

2.6. FUENTE DE PODER

Fueron dos las fuentes de poder que se necesitó, para manejar el normal

funcionamiento de las tarjetas de circuitos de control y fuerza, tanto para el motor de paso, como para el motor trifásico. Una de las cuales, es una fuente de computadora, la que trae salidas de voltaje con un rizado mínimo y protección contra cortocircuito. Esta tiene un suministro de varias terminales de las que se puede detallar en la tabla VIII.

Los datos de placa de la fuente de poder son:

Entrada 100 - 125 V CA

2.0 A 50 / 60 Hz.

Salida CD 63.5 W

Esta fuente suministra la energía necesaria al motor de paso, y también alimenta a la tarjeta de control del inversor. La otra fuente de poder es utilizada para la parte fuerza del inversor trifásico, que alimenta un motor trifásico con un voltaje de 12 V, y una capacidad de corriente de 1.2 A.

P8-1	— Power Good	
P8-2	— Key	
P8-3	+12 V	0.20 A
P8-4	-12 V	0.25 A

P8-5	GND	
P8-6	GND	
P9-1	GND	
P9-2	GND	
P9-3	-5 V	0.30 A
P9-4	+5 V	1.93 A
P9-5	+5 V	1.93 A
P9-6	+5 V	1.93 A
P10-1	+12 V	0.90 A
P10-2	GND	
P10-3	GND	
P10-4	+5 V	0.60 A

TABLA VIII. Terminales de voltaje de salida de la fuente

7. PRUEBAS DE LABORATORIO

Para las pruebas de laboratorio se utilizó el osciloscopio Wave Tek 600, del que

se aprovechó los dos canales (A y B), para poder realizar un mejor análisis del desfaseamiento entre las diferentes formas de onda producida por los circuitos.

1. PRUEBAS DEL CIRCUITO MANEJADOR (DRIVER) DEL MOTOR DE PASO

Para las pruebas del motor de paso, se obtuvo la gráfica de la señal de salida de voltaje del circuito manejador sin carga, ver figura 2.16. En cambio en la figura 2.17, se observa la señal de salida del circuito con carga; ambas figuras representan las señales de las terminales A1 y B1 (Figura 2.2). Se puede comparar que la diferencia es mínima y la señal se deforma (Figura 2.17) debido al efecto inductivo del motor de paso.

En la figura 2.17, se denota el desfase de 90° entre las señales de salida A1 y B1. En la figura 2.18, se observa el desfaseamiento de 180° entre las señales de salida A1 y A2 del circuito manejador del motor de paso.

En la figura 2.19, se muestra la señal de salida de reloj del manejador del motor de paso y los tiempos de carga y descarga del capacitor C1 (Ver figura 2.5).

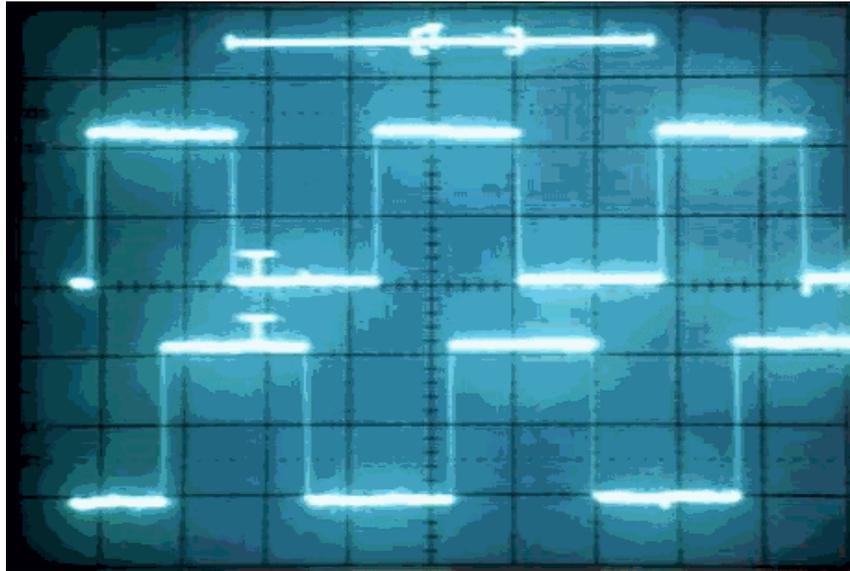


FIGURA 2.16. Señal de salida del manejador del motor de paso (A1 y B1) Sin carga.

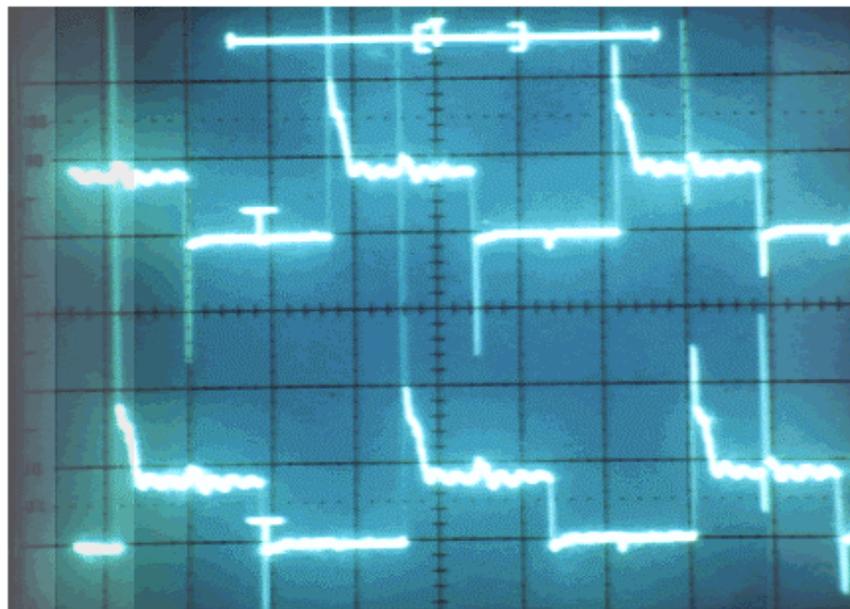


FIGURA 2.17. Señal de salida del manejador del motor de paso (A1 y B1) Con carga.

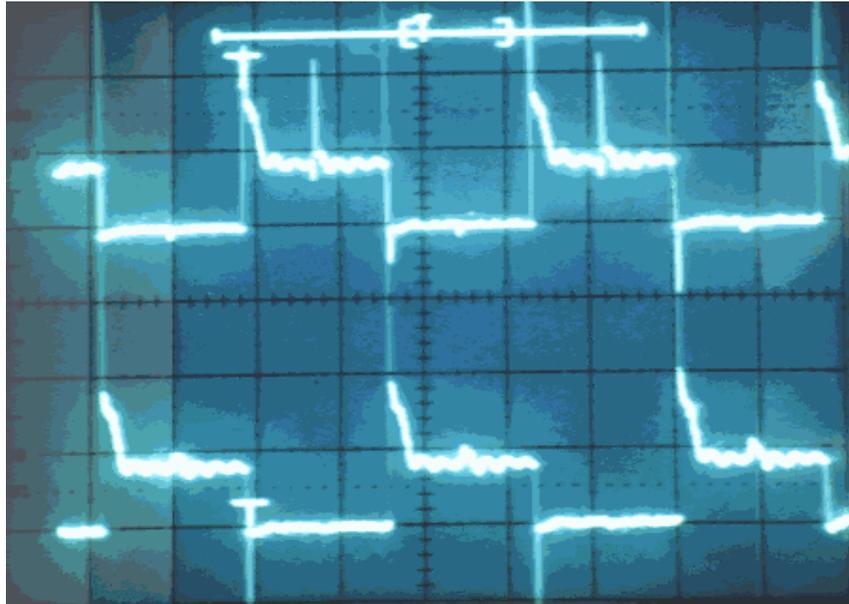


FIGURA 2.18. Señal de salida del manejador del motor de paso (A1 y A2) Con carga.

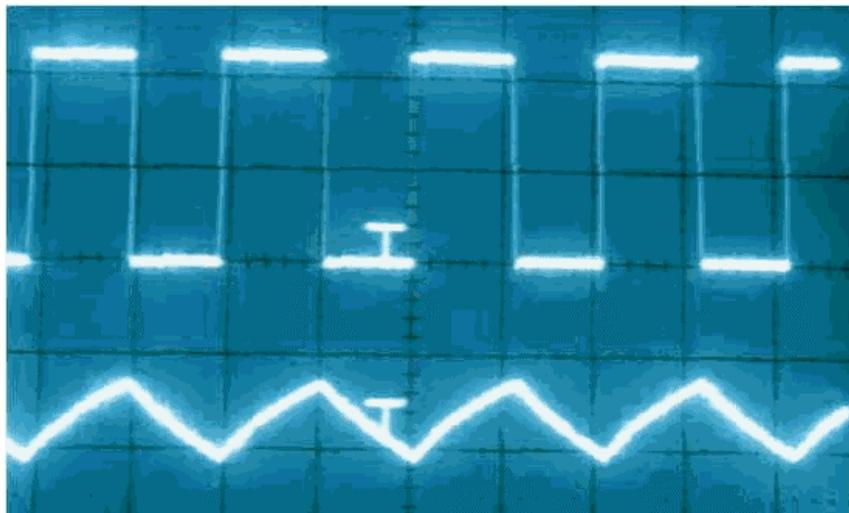


FIGURA 2.19. Señal de salida de reloj del manejador del motor de paso.

2. PRUEBAS DEL CIRCUITO MANEJADOR (DRIVER) DEL MOTOR TRIFASICO (INVERSOR).

Para las pruebas del inversor se tomaron varias muestras en el osciloscopio. Una de ellas, la figura 2.20, muestra las señales de control que se aplican a las bases de los transistores Q1 y Q2 del puente inversor trifásico (Ver figura 2.8), cabe indicar que cada señal tiene un desfase de 60° .

En la figura 2.21, se observan las señales trifásicas de los voltajes de línea a neutro V_{AN} y V_{BN} . Como podemos ver, están desfasadas 120° , con una carga resistiva en estrella. Pero en la figura 2.22 observamos los voltajes de línea a línea, V_{AB} y V_{BC} , con el mismo tipo de carga; desde luego, existe el mismo desfase entre voltaje de línea a línea y el voltaje de línea a neutro.

En la figura 2.23 se muestra la señal trifásica de voltaje V_{AB}

Podemos observar en la figura 2.24 las señales de voltajes trifásicas V_{AB} y V_{BC} , del manejador pero conectado al motor trifásico, se pueden notar en las señales de onda cambios cada 60° , debido a las señales de control del puente inversor.

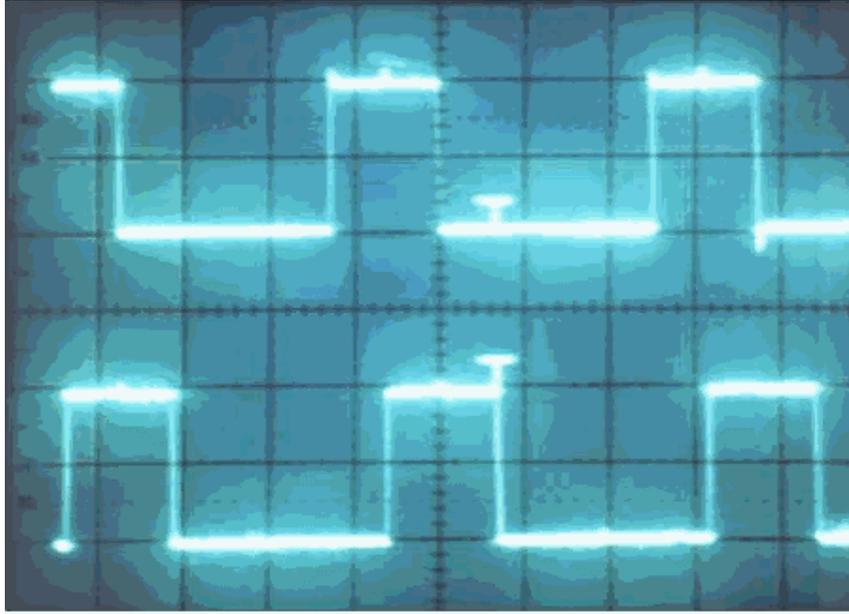


FIGURA 2.20. Señal de control aplicada a la base de los transistores Q1 y Q2.

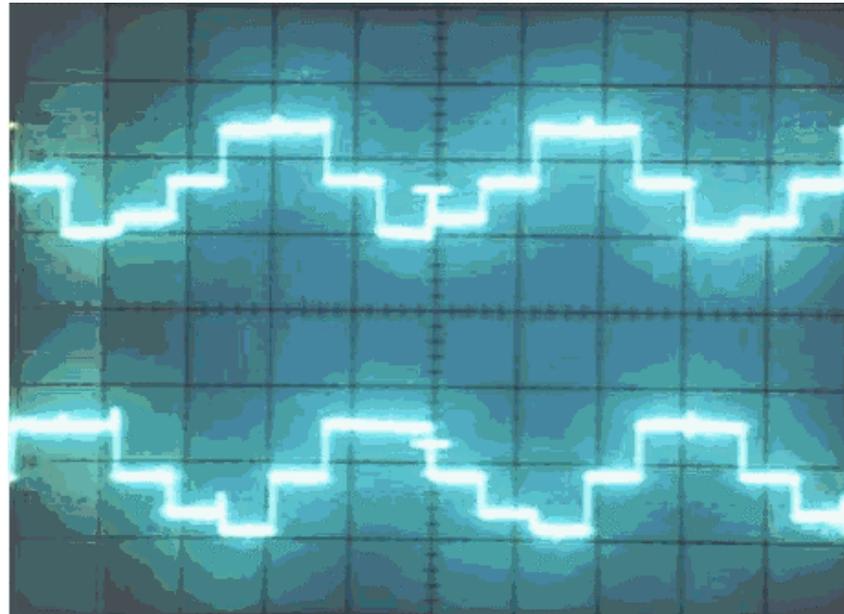


FIGURA 2.21. Señales de voltajes trifásicas V_{AN} y V_{BN} con carga resistiva.

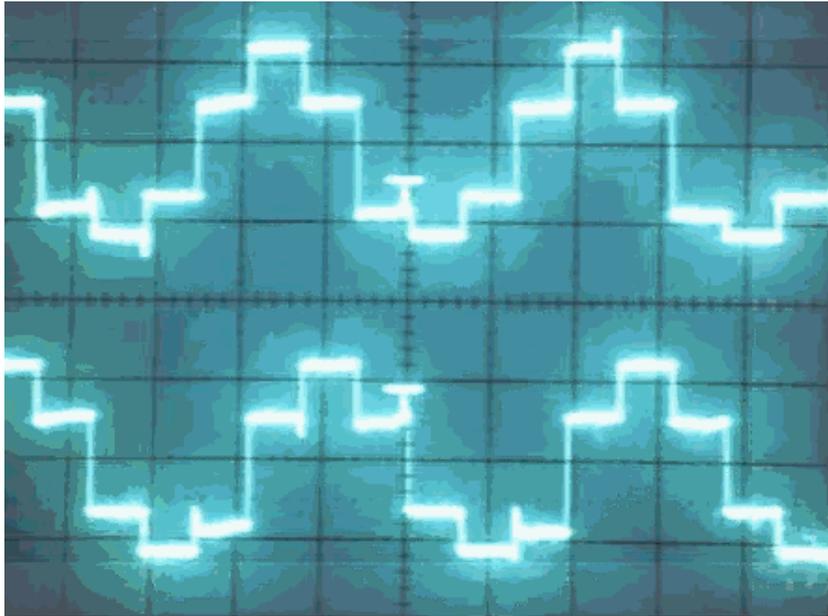


FIGURA 2.22. Señales de voltajes trifásicas V_{AB} y V_{BC} con carga resistiva.

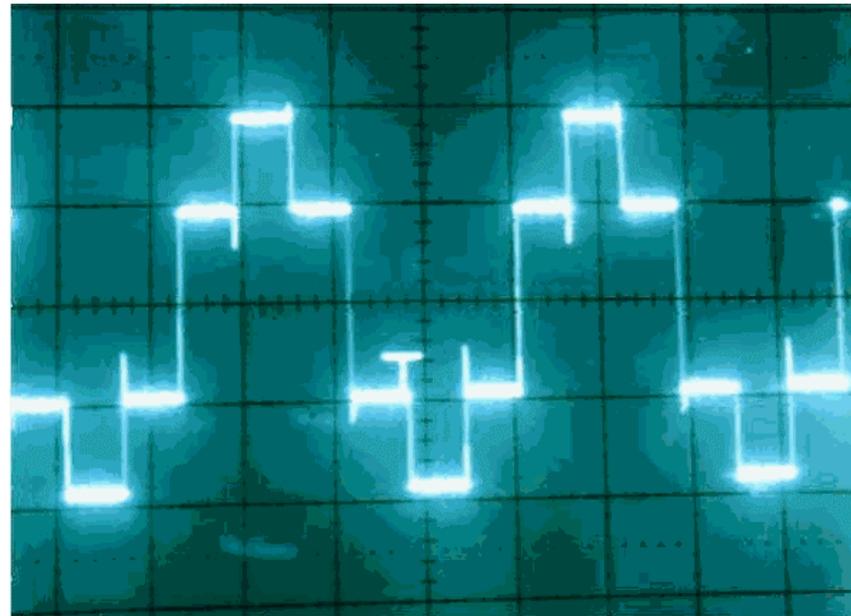


FIGURA 2.23. Señal de voltaje trifásica V_{AB} con carga resistiva.

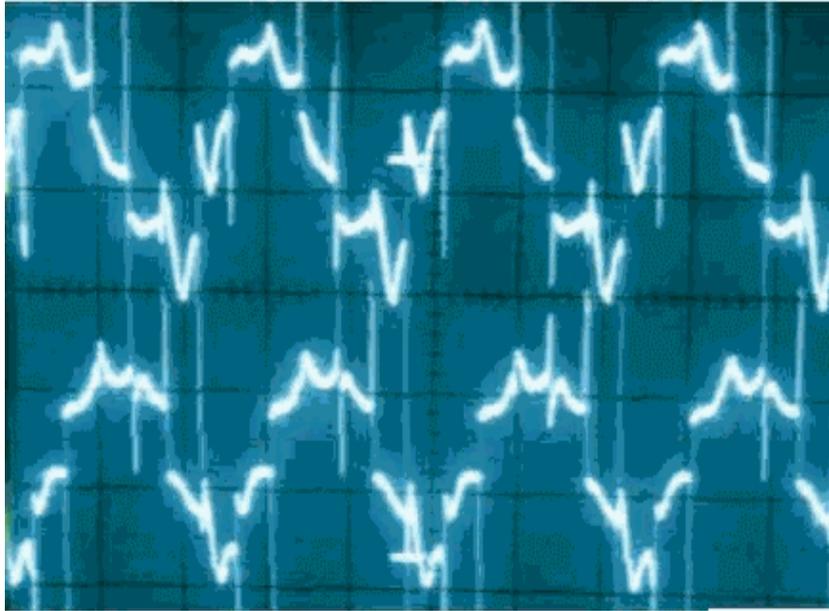
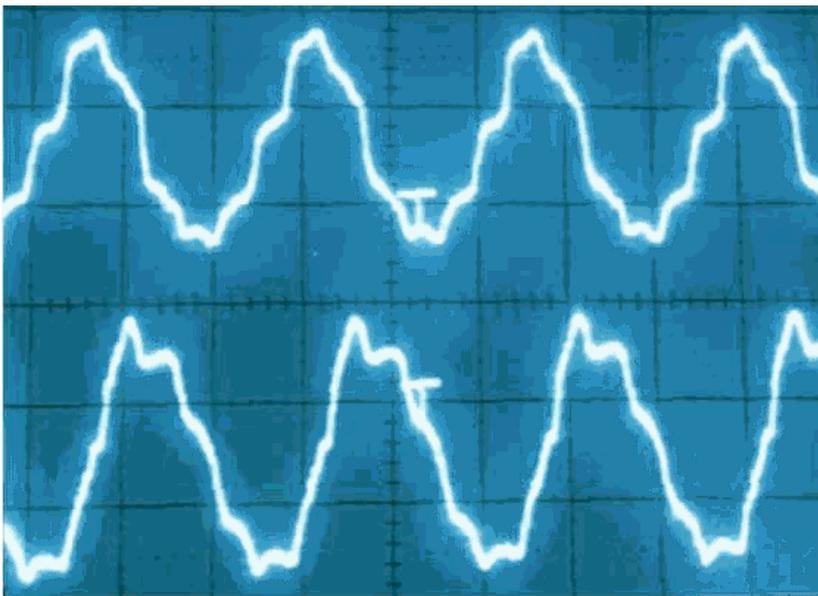


FIGURA 2.24. Señales de voltajes trifásicas VAB y VBC conectado



al motor trifásico.

**FIGURA 2.25. Señales de voltajes trifásicas VAB y VBC conectadas
al motor trifásico, con banco de capacitores.**

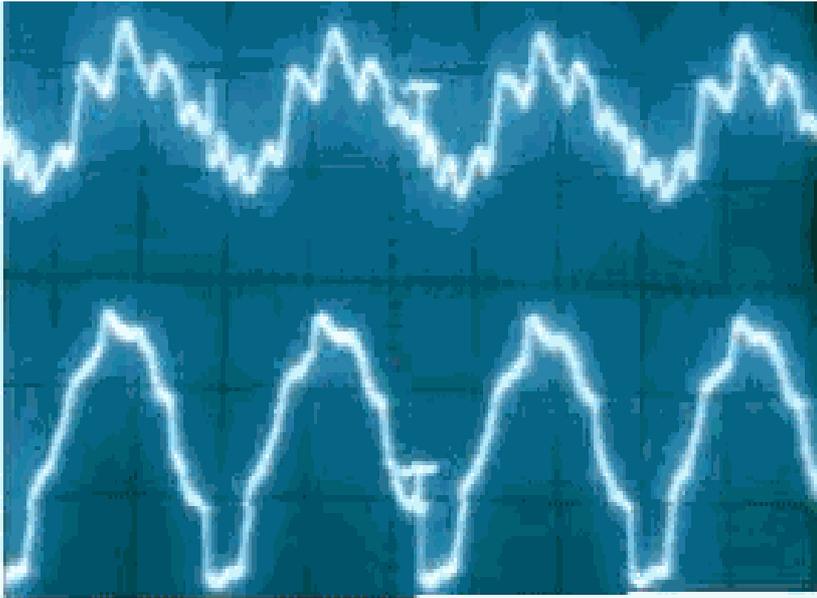


FIGURA 2.26. Señal de corriente trifásica I_A y señal de voltaje trifásica V_{AB} , del motor trifásico, con banco de capacitores.

Vamos a realizar una comparación entre las señales trifásicas producidas en el motor trifásico (Figura 2.24) y las generadas con la misma carga motor pero con un banco de capacitores conectados en estrella, los mismos que servirán para filtrar dichas señales y reducir los picos producidos por efecto inductivo, debido a la transición en la etapa de control entre el apagado de un transistor y el encendido del siguiente.

En la figura 2.26, observamos la señal de corriente I_A junto con la señal de voltaje V_{AB} . Para obtener la corriente I_A se midió la caída de tensión en una resistencia de alta potencia y de baja impedancia, conectada en serie con el motor.

CAPITULO III

PROGRAMACION DEL CONTROLADOR LOGICO PROGRAMABLE

3.1. INTRODUCCION

El controlador lógico programable (PLC) es básicamente un sistema de control industrial basado en una computadora, que utiliza instrucciones de programación para tomar decisiones de encendido y apagado, que de otra manera tendrían que ser realizadas por medio de lógica alambrada, o compuertas lógicas (1). Este sistema contiene un microprocesador que puede ser programado desde un panel programador, a partir de un teclado mediante un cable de comunicación, o por medio de una memoria EEPROM. El PLC está diseñado para recibir señales de entradas y enviar señales de salidas en respuesta a la lógica del programa. El programa generalmente consiste de contactos, salidas, temporizadores, contadores y funciones matemáticas.

El PLC fue desarrollado en el año 1969, para facilitar el problema causado por los cambios periódicos en los sistemas de control. Una firma consultora de ingeniería llamada Bedford Associates, que después cambió su nombre a

Modicon, diseñó el primer controlador programable como un sistema de control de computador dedicado, construido específicamente para la General Motors Hidramatic Division. La industria del automóvil se encontraba interesada en el desarrollo de este tipo de sistema. A esta propuesta también respondió Allen-Bradley con su propio sistema.

3.2. INTRODUCCION AL SIMATIC S7-200 CPU 212

El SIMATIC S7-200 es un Micro-PLC versátil, que puede ser utilizado ampliamente en numerosas tareas. En la figura 3.1, se muestra el micro-PLC Simatic S7-200 CPU-212 que utilizamos. Debido a su diseño compacto, su capacidad de ampliación, su bajo costo y, su desarrollado juego de operaciones, hacen que este Micro-PLC se ajuste a numerosas aplicaciones, como por ejemplo:

- Control de máquinas pequeñas como bobinadoras y máquinas de embalaje.
- Mezcladores de yeso y cemento
- Maquinaria para trabajar la madera
- Industria alimenticia: amasadoras, cortadoras, granuladoras y mezcladoras.
- Sistema de bandas
- Control de compuertas, bombas, compresores.
- Alzamiento Hidráulico
- Instalaciones eléctricas
- Laboratorio: mezcladoras, armarios climáticos, bombas de vacío, extractores.



FIGURA 3.1. Micro-PLC Simatic S7-200 CPU 212

3.2.1. EQUIPOS NECESARIOS

En la figura 3.2, se muestra la estructura básica de un PLC S7-200 en la cual se incluyen: la CPU S7-200, un PC, el software de programación STEP 7- MicroWIN y el cable de comunicación .

Para poder utilizar un PC, se requiere un cable PC/PPI o una tarjeta MPI. El cable de comunicación se suministra junto con la tarjeta MPI.

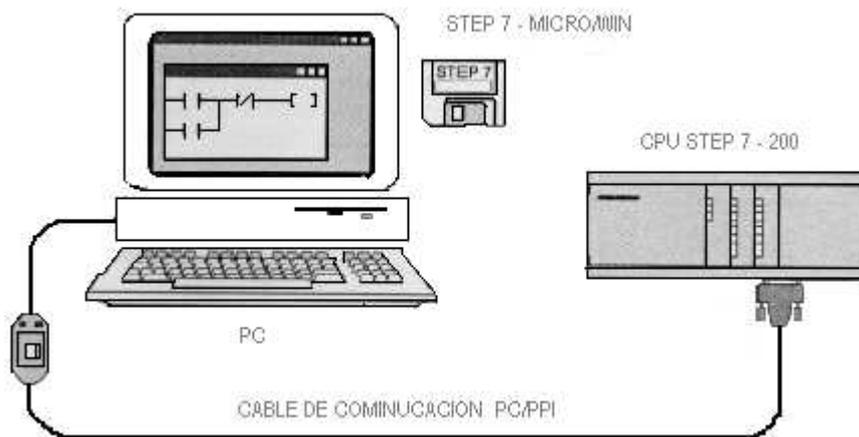


FIGURA 3.2. Componentes de un Micro-PLC S7-200

2. CAPACIDAD DE LA CPU 212 DEL S7-200

La CPU 212 tiene una amplia gama de funciones que permiten diseñar soluciones de automatización complejas. Información adicional de datos técnicos de esta CPU, podrán ser vistos en el Anexo D.

3.2.3. PRINCIPALES COMPONENTES DEL SIMATIC S7-200

El Micro-PLC SIMATIC S7-200 consiste en un módulo, el cual es capaz de procesar las señales de entrada, controlar las salidas y suministrar energía a los sensores como también a sí mismo. Existen módulos de ampliación de entradas y salidas, analógicas o digitales y, además el módulo de fuente de alimentación. Dos son los equipos básicos que se ofrecen:

- CPU 212
- CPU 214

3.2.3.1. EL MÓDULO CPU S7-200

Nos referiremos específicamente a la CPU 212 (Ver figura 3.1), el cual es un aparato autónomo y compacto, que está compuesto de tres partes: la unidad central de procesamiento (CPU), la fuente de alimentación, así como la sección de entradas y salidas digitales.

La función de cada una de estas partes es:

- La CPU ejecuta el programa y almacena los datos para la tarea de automatización.
- La fuente de alimentación le proporciona corriente a la unidad central y a los módulos de ampliación conectados, por medio de 24 VCD. Cada módulo de CPU dispone de una fuente de alimentación para sensores de 24 VCD, puede suministrar esta tensión para puntos de entradas locales, o para bobinas de relés en los módulos de ampliación. Si el consumo de 24 VCD supera a la corriente que es capaz de aportar el módulo CPU, entonces puede añadirse una fuente de alimentación externa para abastecer con 24 VCD a los módulos de ampliación. Cuando deban alimentar diferentes puntos, la fuente de alimentación para sensores del S7-200 y la fuente alimentación externa deberán tener como mínimo

una conexión común. El módulo CPU alimenta también con 5 VCD a los módulos de ampliación cuando se conectan al módulo base. Si el consumo de 5 VCD de los módulos de ampliación supera la capacidad de corriente del módulo CPU, entonces es necesario desconectar tantos módulos de ampliación como sean necesarios para no superar la capacidad de corriente del módulo CPU.

- La sección de entrada vigila las señales de los aparatos de campo (por ejemplo, sensores e interruptores); y la de salida controla el sistema de automatización, supervisando las bombas, motores, válvulas, u otros dispositivos del proceso.

También posee una interface de comunicación que permite conectar la CPU a una unidad de programación, o a otros dispositivos. Unos diodos luminosos indican el modo de operación de la CPU (RUN o STOP), el estado de las entradas y salidas integradas, así como los posibles fallos del sistema que se hayan detectado.

3.2.3.2. MODULO DE AMPLIACION

Los módulos de ampliación se necesitan en aplicaciones donde se requiera más de 8 entradas y 6 salidas (digitales), que son las que se encuentran integradas al CPU 212; también se puede aumentar módulos para manejar variables analógicas. En la figura 3.3, se muestra un ejemplo de módulo de ampliación de 8 salidas de relé.

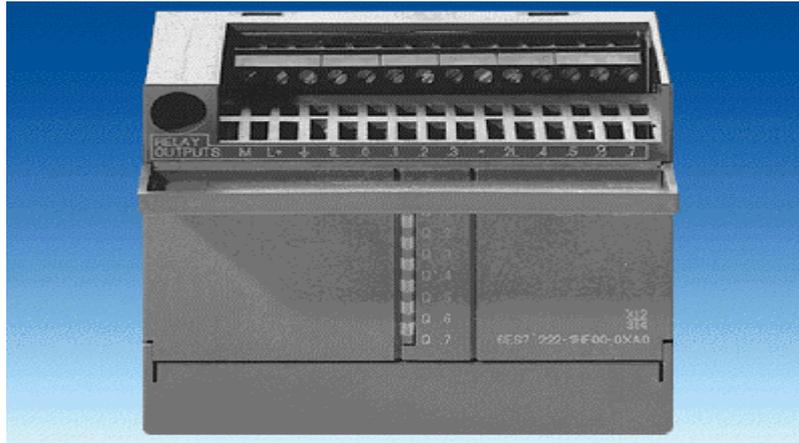


FIGURA 3.3. Módulo de ampliación

Cada módulo de ampliación dispone de una fuente de alimentación, tanto de 5 VCD como de 24 VCD.

Los *módulos de ampliación digitales* reservan siempre un espacio de la imagen del proceso en incrementos de ocho bits (un byte). Si un módulo no dispone de un punto físico para cada bit de cada byte reservado, se pierden estos bits no utilizados y no se pueden asignar a los módulos subsiguientes en la cadena de E/S. Refiriéndonos a los módulos de salidas, los bits no utilizados en los bytes reservados pueden servir de marcas internas (bits M). En cuanto a los módulos de entrada, los bits no utilizados en los bytes reservados se ponen a cero cada vez que se actualizan las entradas, por lo que no pueden servir de marcas internas.

Los *módulos de ampliación analógicos* se prevén siempre en incrementos de dos puntos. Si un módulo no ofrece E/S físicas para cada uno de dichos puntos, se pierden los mismos y no se pueden asignar a los módulos subsiguientes en la cadena de E/S. Puesto que para las E/S analógicas no se prevé ningún espacio en

la imagen del proceso, no hay ninguna manera de aprovechar estas E/S no utilizadas. Todos los accesos de E/S analógicas se establecen en el mismo instante de ejecutarse la operación.

3.2.4. SECCION DE ENTRADA/SALIDA

En un PLC, la sección de entrada/salida (E/S) se encarga de vigilar y supervisar la intercomunicación entre los dispositivos industriales y, los circuitos electrónicos de baja potencia, que almacenan y ejecutan el programa de control.

3.2.4.1. ENTRADA/SALIDA DIGITALES

La sección de entrada puede recibir señales desde dispositivos, tales como: botoneras, pulsadores, selectores, sensores, termostatos, e interruptores de cualquier tipo. Tales dispositivos envían señales de voltaje que pueden ser de corriente directa o de corriente alterna.

Las señales de salida pueden ser: de tipo transistorizados, de relé con contacto de baja potencia, o por triac.

Todos los módulos de entrada de los PLC modernos poseen convertidores ópticos de señal (optoacopladores), los cuales sirven para llevar a cabo el aislamiento eléctrico entre los

circuitos de entrada (los cuales reciben señales de alta potencia, de 24 VCD, o de 120 VCA) y los elementos electrónicos del procesador, que aceptan señales digitales de baja potencia.

Las entradas y salidas que están integradas en la CPU tienen direcciones fijas; en cambio, en los módulos de ampliación de E/S, las direcciones vienen determinadas por el tipo de E/S y la posición del módulo de ampliación a la derecha de la CPU con respecto al anterior módulo de entradas o salidas del mismo tipo. Por ejemplo, un módulo de salidas no afecta las direcciones de un módulo de entradas y viceversa. Igualmente, los módulos analógicos no afectan el direccionamiento de los módulos digitales y viceversa.

3.2.4.2. ENTRADA/SALIDA ANALOGAS

Los procesos industriales producen señales analógicas, como por ejemplo, presión, fuerza, velocidad, caudal, aceleración, desplazamiento (angular, lineal), temperatura, nivel de fluidos, y demás variables de proceso, que varían constantemente con el tiempo. La velocidad de cambio puede ser muy lenta, como la variación de temperatura ambiente, o muy rápida, como sucede en los sistemas de audio. El Micro-PLC para poder manejar dichas señales, se han diseñado unos módulos separados de E/S analógicas.

Una señal análoga incluye todos los valores de la señal, desde un valor mínimo hasta un máximo, de voltaje o corriente.

Para interrelacionar las señales analógicas con la lógica digital del CPU, se necesitan los *convertidores analógico a digital* (CAD), el cual sirve para convertir las señales de entrada análogas de voltaje (0V - 10V), o de corriente (4mA - 20mA), a una señal digital de 16 bits. Las señales de entrada que se encuentran entre 4mA - 20mA, o entre 0V - 10V, pueden tener un número infinito de incrementos entre el valor mínimo y máximo.

Para transformar las señales digitales del CPU a señales análogas de salida del PLC, se necesita el *convertidor digital a analógico* (DAC). La señal de salida análoga varía de aplicación en aplicación. El rango más utilizado de las señales de corriente está entre 4mA - 20mA; y los más usados para las señales de voltaje son 0 - 5V, y 0 - 10V. Para las señales de voltaje, 0V es el valor más bajo; para el rango de corriente, 4mA es el valor más bajo, o completamente apagado.

El número de incrementos entre el valor mínimo y el valor máximo es conocido como resolución. El número de incrementos o, resolución en la señal de salida análoga depende de la señal de entrada digital. Si la entrada digital es de 16 bits, habrán 32.768 incrementos.

5. PARTES DE LA CPU

La memoria de la CPU está dividida en tres partes:

3.2.5.1. MEMORIA DEL PROGRAMA

Contiene las instrucciones que ejecuta el autómata programable para la aplicación deseada. El programa está almacenado en memoria no volátil, es decir, que el programa no se pierde, ni en caso de interrumpirse la alimentación del autómata programable. Antes de que un PLC pueda comenzar a controlar un sistema industrial, deben ser ingresados las instrucciones codificadas que constituyen el programa de usuario. El programa de usuario se puede crear con los lenguajes de programación KOP o AWL. La memoria de programa se denomina bloque de organización (OB).

3.2.5.2. MEMORIA DE DATOS

Es el área de trabajo a la que accede el programa de automatización. La *memoria de datos* contiene direcciones para cálculos, almacenamiento temporal de resultados intermedios, constantes utilizadas para recetas y, otros parámetros de control fijos.

La memoria de datos se compone de:

- **Area De Datos**, la cual se divide en una memoria de variables (V), una imagen de proceso de las entradas (E), una imagen de proceso de las salidas (A), marcas internas (M) y, marcas especiales (SM).

- **Objetos** son direcciones asignadas a elementos. Los objetos abarcan temporizadores (T), contadores (Z), contadores rápidos (HC), entradas/salidas analógicas y acumuladores (ACU).

Una parte de esta área está almacenada en una memoria no volátil y, otra parte en la memoria RAM. El contenido de la RAM se mantiene por un tiempo limitado, incluso, después de desconectar el autómata programable gracias a un condensador de alta potencia.

3.2.5.3. MEMORIA DE PARÁMETROS

Permite almacenar determinados parámetros configurables, tales como contraseñas, direcciones de estaciones e informaciones sobre las áreas remanentes. El contenido de esta área está almacenada en una memoria no volátil.

5. CICLO DE LA CPU S7 - 200

La CPU S7 – 200 ejecuta el programa en un ciclo completo continuo. Como se muestra en la figura 3.4, el ciclo completo de la CPU abarca las siguientes tareas:

FIGURA 3.4. Ciclo de la CPU S7 - 200

1. LEER LAS ENTRADAS DIGITALES

Al principio de cada ciclo se leen los valores actuales de las entradas digitales, escribiéndose luego en la imagen del proceso de las entradas. La CPU reserva un espacio de la imagen del proceso de las entradas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proporcionan una entrada física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos subsiguientes en la cadena de E/S, o no podrán ser utilizados en la imagen del proceso.

Sin embargo, si la CPU asiste a varios módulos de ampliación y no se está utilizando su capacidad de E/S (porque no se han instalado los módulos de ampliación), los bits de entrada de ampliación no utilizados se pueden usar como marcas internas adicionales.

La CPU no actualiza automáticamente las entradas analógicas como parte del ciclo y no reserva un espacio de la imagen del proceso para las mismas. A las entradas analógicas se debe acceder directamente desde el programa de usuario.

2. EJECUTAR EL PROGRAMA DE USUARIO

Durante esta fase del ciclo, la CPU ejecuta el programa

desde la primera operación hasta la última (fin del programa). El programa almacenado en la CPU utiliza el estado de las entradas para evaluar la lógica. A medida que se ejecuta el programa, la CPU actualiza los datos. El control directo de las entradas y salidas permite acceder directamente a las mismas, mientras se ejecuta el programa o una rutina de interrupción.

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan como parte del programa. Las rutinas de interrupción no se ejecutan como parte del ciclo, sino sólo cuando ocurre el evento, esto puede ser en cualquier momento del ciclo.

3. PROCESAR LAS PETICIONES DE COMUNICACIÓN Y EJECUTAR EL AUTODIAGNÓSTICO DE LA CPU

Durante la fase de procesar peticiones de comunicación, la CPU procesa los mensajes que haya recibido por el interface de comunicación.

Durante la fase de autodiagnóstico, se comprueba el firmware de la CPU y la memoria del programa, así como el estado de los módulos de ampliación.

4. ESCRIBIR LAS SALIDAS DIGITALES

Al final de cada ciclo, la CPU escribe los valores de la imagen del proceso de las salidas en las salidas digitales. La CPU reserva un espacio de la imagen del proceso de las salidas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proveen una salida física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos subsiguientes en la cadena de E/S.

No obstante, los bits no utilizados de la imagen del proceso de las salidas se pueden usar como marcas internas adicionales.

La CPU no actualiza automáticamente las salidas analógicas como parte del ciclo y no reserva un espacio de la imagen del proceso para las mismas. A las salidas analógicas se debe acceder directamente desde el programa de usuario.

5. INTERRUMPIR EL CICLO

Si se utilizan *interrupciones*, las rutinas asociadas a los eventos de interrupción se almacenan como parte del programa. Las rutinas de interrupción no se ejecutan como parte del ciclo, sino sólo cuando ocurre un evento. La CPU procesa las interrupciones habilitadas de forma asincrónica al ciclo y ejecuta las rutinas de interrupción

cuando se presente el correspondiente evento, de acuerdo a su orden de aparición y a su prioridad.

6. IMAGEN DEL PROCESO DE LAS ENTRADAS Y SALIDAS

Al ejecutarse el programa, los accesos a las entradas y salidas se efectúan generalmente a través de la respectiva *imagen del proceso*, y no de forma directa. Las imágenes del proceso existen por tres razones principales:

- El sistema comprueba todas las entradas al comenzar el ciclo. De este modo se sincronizan y congelan los valores de estas entradas durante la ejecución del programa. La imagen del proceso actualiza las salidas cuando termina de ejecutarse el programa.
- El programa de usuario puede acceder a la imagen del proceso mucho más rápido de lo que podría acceder directamente a las entradas y salidas físicas, con lo cual se acelera su tiempo de ejecución.
- Las entradas y salidas son unidades de bit a las que se debe acceder en formato de bit. Sin embargo, la imagen del proceso permite acceder a ellas en formato de bits, bytes, palabras y palabras dobles, lo que ofrece flexibilidad adicional.

Otra ventaja es que las imágenes del proceso son lo suficientemente grandes para poder procesar el número máximo de entradas y salidas. Puesto que un sistema real comprende tanto entradas como salidas, en la imagen del proceso existe siempre un número de direcciones que no se utilizan. Estas direcciones libres pueden utilizarse como marcas internas adicionales.

1. CONTROL DIRECTO DE LAS ENTRADAS Y SALIDAS

El acceso a las entradas y salidas se efectúa generalmente a través de las imágenes del proceso. Sin embargo, el control directo de las mismas permite acceder inmediatamente a una entrada o salidas físicas. El acceso directo a una entrada no modifica la dirección correspondiente en la imagen del proceso de las entradas. En cambio, el acceso directo a una salida, actualiza simultáneamente la dirección correspondiente en la imagen del proceso de las salidas.

3.2.7. ESTABLECER COMUNICACIÓN CON UNA CPU S7-200

El *interface punto a punto* (PPI) integrado ofrece al usuario del SIMATIC S7-200, numerosas posibilidades de comunicación. En este modo de comunicación se pueden programar, manejar y vigilar varios SIMATIC S7-200 a través de una línea bifilar común. La velocidad de transmisión de los datos es de 9.6 kbits/s.

La figura 3.5, muestra una configuración típica para conectar el ordenador personal a la CPU mediante el cable PC/PPI. Para establecer un enlace correcto entre los componentes, necesitamos que se cumplan los siguientes requisitos:

1. Ajuste los interruptores del cable PC/PPI para una velocidad de transferencia.

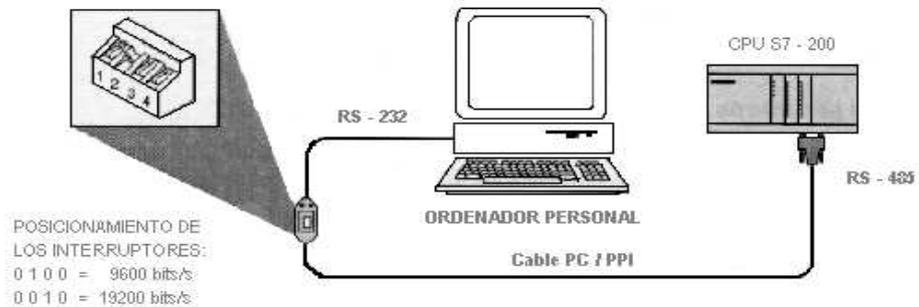


FIGURA 3.5. Comunicación PPI con una CPU

2. Conecte el extremo RS-232 ("PC") del cable PC/PPI al interface de comunicación de su ordenador (COM1 o COM2) y apriete los tornillos de conexión.
3. Conecte el otro extremo (RS-485) del cable PC/PPI al interface de comunicación de la CPU y apriete los tornillos de conexión.

También se puede establecer la comunicación con *interface multipunto* (MPI), conectando el ordenador a la CPU S7-200. El STEP- 7 Micro/WIN también se puede utilizar con una tarjeta de interface

multipunto (MPI). La tarjeta MPI incorpora una sola interface RS-485 para su conexión a la red mediante un cable directo. Una vez establecido un enlace MPI es posible conectar STEP 7-Micro/WIN a una red que contenga otros maestros, tal como se aprecia en al figura 3.6. Cada maestro debe tener una dirección unívoca.

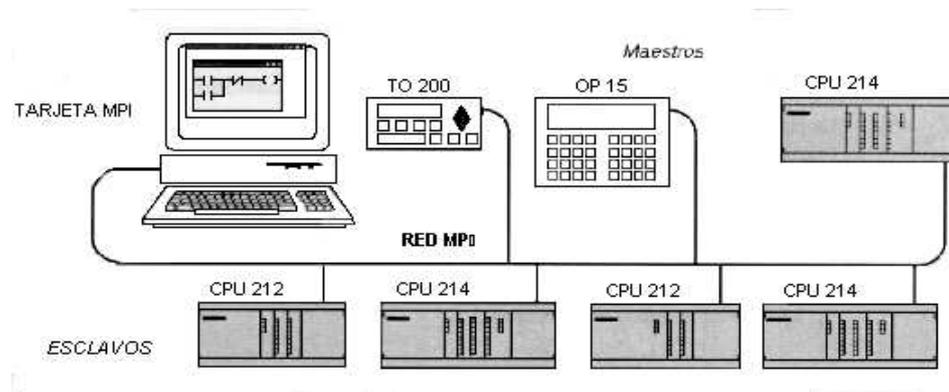


FIGURA 3.6. Comunicación MPI en una red con maestro y esclavos

3.2.8. PROGRAMACION DE S7-200

Antes de crear un proyecto, indique las preferencias para su entorno de programación. Para seleccionar las preferencias, el STEP 7 – Micro/WIN ofrece dos representaciones nemotécnicas para visualizar las direcciones y las operaciones del programa:

- Norma Internacional
- Norma Simatic

Tanto la nemotécnica internacional como la SIMATIC, se refieren al mismo juego de operaciones para el S7 – 200, existiendo una

correspondencia directa entre ambas. Las funciones de las dos representaciones son idénticas. En mayor medida se utiliza la nemotécnica internacional. También determinará el modo de editar el programa:

1. EDITOR AWL (LISTA DE INSTRUCCIONES)

La lista de instrucciones (AWL) es un lenguaje de programación en el que cada línea del programa contiene una operación que utiliza una abreviatura nemotécnica, para representar una función de la CPU.

En la figura 3.7, se muestran los elementos básicos de un programa AWL.

```
//
//Programa para tren transportador
//
//Pulse F1 para obtener ayuda y un programa de ejemplo
//
NETWORK 1 //Marcha del motor:
LD "MARCHA1" //Si I0.0 está activada (on)
AN "PARO_Em1" //y I0-1 no está activada,
= Q0.0 //poner en marcha el motor del transportador.

NETWORK 2 //Paro de emergencia transportador.
LD I0.1 //Si Paro_Em1 está activada
O I0.3 //o si Paro_Em2 está activada,
R Q0.0, 1 //parar el motor del transportador.

NETWORK 3 //Fin del programa
MEND
```

FIGURA 3.7. Ventana del Editor AWL con un programa de ejemplo

Las operaciones AWL utilizan una pila lógica en la CPU para resolver la lógica. Como se muestra en la figura 3.8, dicha pila tiene nueve bits de profundidad y uno de ancho. La mayoría de las operaciones AWL trabajan con el primer bit, o bien con el primero y el segundo bit de la pila. En la pila se pueden cargar nuevos valores. Combinando los primeros dos bits de la pila, se pierde el valor del nivel superior y en su lugar se coloca el valor del nivel siguiente, es decir que a la pila se le resta un bit.

Bits de la pila lógica	S0	Pila 0	Primer nivel de la pila o Primer valor de la pila
	S1	Pila 1	Segundo nivel de la pila
	S2	Pila 2	Tercer nivel de la pila
	S3	Pila 3	Cuarto nivel de la pila
	S4	Pila 4	Quinto nivel de la pila
	S5	Pila 5	Sexto nivel de la pila
	S6	Pila 6	Séptimo nivel de la pila
	S7	Pila 7	Octavo nivel de la pila
	S8	Pila 8	Noveno nivel de la pila

FIGURA 3.8. Pila lógica de la CPU S7 – 200

2. EDITOR KOP (ESQUEMA DE CONTACTOS)

Al programar con KOP, se crean y se disponen componentes gráficos que conforman un segmento de operaciones lógicas. Como se muestra en la figura 3.9, se ofrecen los siguientes elementos básicos para crear programas:

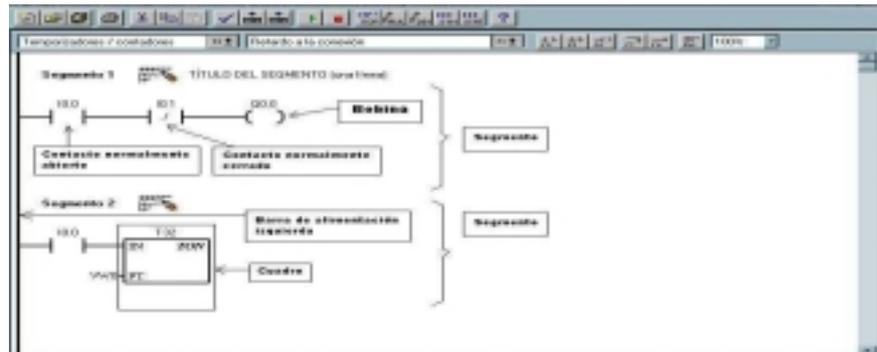


FIGURA 3.9. Elementos básicos de KOP

- **Contactos:** un contacto representa un interruptor por el que fluye la corriente cuando está cerrado.
- **Bobinas:** una bobina representa un relé que se excita cuando se aplica tensión.
- **Cuadros:** un cuadro representa una función que se ejecuta cuando la corriente fluye por él.
- **Segmentos:** un segmento constituye un circuito completo. La corriente fluye desde la barra de alimentación izquierda pasando por los contactos cerrados para excitar las bobinas o cuadros.

8. CREAR Y GUARDAR UN PROGRAMA

Antes de poder generar un programa, es necesario, crear o abrir un proyecto. Al crear un proyecto, STEP 7-Micro/WIN abre los siguientes editores:

- Editor KOP o AWL (dependiendo del editor predeterminado)
- Editor de bloque de datos.
- Editor de tablas de estado / de forzado.
- Editor de tabla de símbolos.

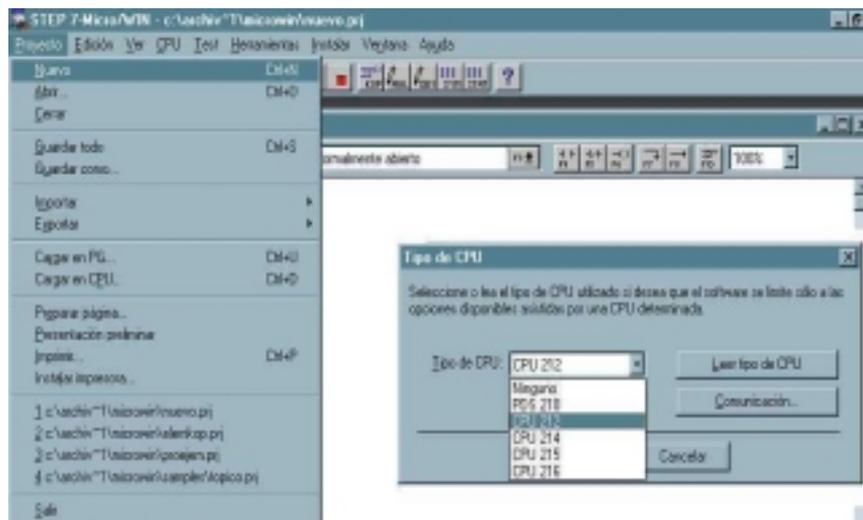


FIGURA 3.10. Crear un nuevo proyecto

Para crear un proyecto (ver figura 3.10), elija el comando **Proyecto > Nuevo....**Entonces se abre el cuadro de diálogo "CPU".

Al crear un proyecto nuevo, se tendrá que elegir el tipo de CPU del PLC

S7-200. Una vez cargado, el programa verificará la utilización de funciones que sólo estén disponibles en la CPU.

En STEP 7, todos los programas de usuario y los datos necesarios para ellos, están depositados en bloques, es decir en módulos de software. La posibilidad de llamar un bloque desde otro (como si fueran subprogramas) facilita la estructuración del programa y aumenta notablemente la claridad, comprensibilidad y calidad de los programas de PLC.

Los bloques disponibles en el STEP 7 son:

- ***Bloques de organización (OB)***: sirve para regular los recursos del programa. Cuando se arranca un OB se entrega una información detallada sobre el evento desencadenante, información que puede ser evaluada en el programa de usuario. Los OB se dividen en clases de acuerdo con el evento desencadenante, a cada una de las cuales se le asigna una prioridad y en función de la misma, se pueden interrumpir unos a otros.
- ***Bloque de función (FB)***: Contiene el programa de usuario. Los FB pueden ser alimentados con datos diferentes, cada vez que se los llame (instancia). Estos datos juntos con las variables internas y los resultados, se depositan en el DB de instancia asignado y el sistema los gestiona automáticamente.
- ***Bloques de datos de instancia (DB de instancia)***: se asignan al bloque cuando se llama a un bloque de función y a un bloque de función del sistema. Se generan automáticamente durante la compilación. El usuario

puede acceder a estos datos de instancia (también simbólicamente) desde cualquier punto de su programa de usuario o incluso desde un sistema de manejo y visualización.

- **Funciones (FC):** contienen rutinas de programa para funciones utilizadas frecuentemente. Cada función tiene un valor de función fijo. Todos los parámetros de salida tienen que procesarse inmediatamente después de la llamada, por esto, las funciones no necesitan de DB de instancia.
- **Bloques de datos (DB):** son las áreas de memoria para almacenar los datos de usuario. Además de los datos asignados en cada caso a un FB (datos de instancia), pueden definirse datos globales para su utilización por parte de los bloques que lo precisen, por ejemplo, para recetas. A los componentes de un DB se les puede asignar datos simples o compuestos. Los datos simples son: BOOL, REAL o INTEGER. Los datos compuestos se componen a su vez de datos simples. Los datos del DB pueden direccionarse simbólicamente, lo cual simplifica la programación de legibilidad del programa.
- **Bloques de función del sistema (SFB):** son bloques de función integrados en el sistema operativo de la CPU. Las variables de estos SFB están depositadas también en los bloques de datos de instancia.
- **Funciones de sistema (SFC):** son funciones integradas en el sistema operativo de la CPU.
- **Bloques de datos del sistema (SDB):** son datos para el sistema operativo

de la CPU con sus respectivos ajustes, por ejemplo, en los parámetros de módulos.

3.2.9.1. CREAR UN PROGRAMA

Se puede crear un programa de usuario (OB1) con el editor KOP, o el editor AWL. El editor KOP sirve para escribir programas en lenguaje de contactos. El editor AWL es un editor de texto, que brinda cierta flexibilidad en cuanto a la forma de introducir las operaciones del programa.

En la figura 3.11, se puede observar la ventana del editor KOP. Una vez completada la serie de segmentos puede comprobar la sintaxis lógica eligiendo el comando **CPU > Compilar**.

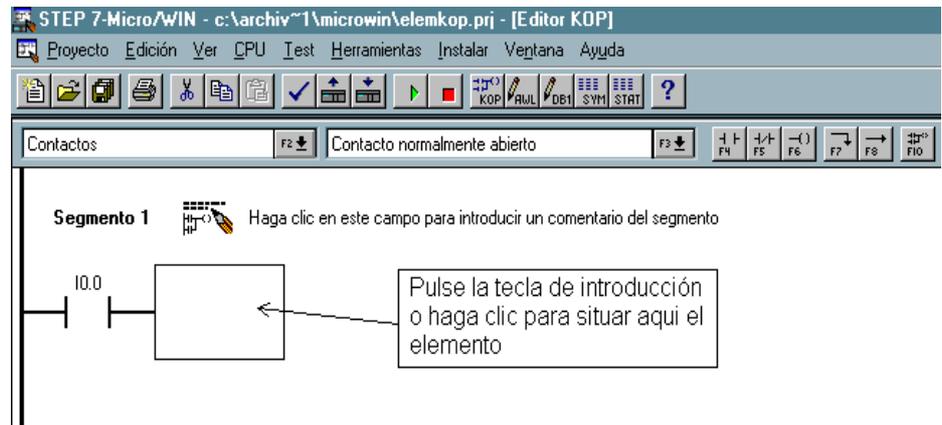


FIGURA 3.11 Ventana del editor KOP

3.2.9.2. GUARDAR UN PROGRAMA Y CARGARLO EN LA CPU

Para guardar los componentes de un proyecto, elija el comando **Proyecto > Guardar**.

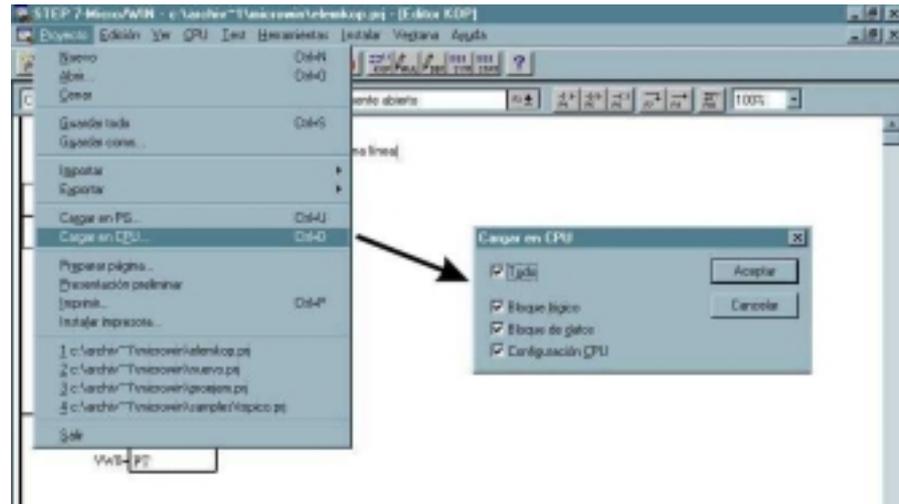


FIGURA 3.12. Cargar los componentes del proyecto en la CPU

Después de introducir completamente el programa, el proyecto se puede cargar en la CPU. Entonces se abrirá un cuadro de diálogo (ver figura 3.12) en el que se puede especificar las componentes del proyecto que desea cargar en la CPU.

- El bloque de organización (OB) contiene el programa que debe ser ejecutado por la CPU.
- El bloque de datos (DB) contiene los valores de inicialización utilizados en el programa.

- La configuración de la CPU (CFG), contiene la información de instalación del sistema incluyendo los parámetros de comunicación, los márgenes remanentes, los ajustes de los filtros de entrada, de contraseña y los ajustes de salidas.

3.2.10. CREAR BLOQUE DE DATOS (DB)

Con el editor de bloque de datos se puede predefinir e inicializar las variables utilizadas en el programa. El uso del bloque de datos es opcional.

El editor de bloque de datos (ver figura 3.13) es un texto que brinda cierta flexibilidad en cuanto a la forma de introducir los valores de datos.

Al crear un bloque de datos tome en cuenta lo siguiente:

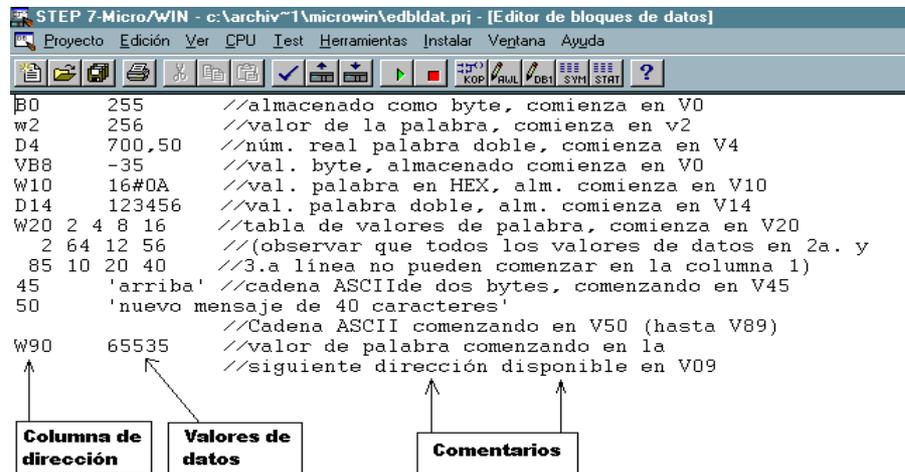


FIGURA 3.13. Ejemplo de bloque de datos

- 1.- Utilice la primera columna de cada línea para indicar el tamaño de

los datos y, la dirección inicial de cada valor al almacenarlo en la memoria V.

TIPO DE DATOS	EJEMPLO
Hexadecimal	16#AB
Entero (decimal)	10 o 20
Entero con signo (decimal)	-10 o +50
Real (coma flotante): utilice (".") y no una coma (",")	10.57
Texto (ASCII): texto de la cadena, contenido entre apóstrofe (Nota "\$" es un carácter especial que sirve de comodín de un apóstrofe o de un signo de dólar dentro de una cadena.)	'Siemens' 'Que \$ es esto' 'Sólo \$\$25'

TABLA IX. Notación para introducir valores en un bloque de datos.

TAMAÑO DE DATOS	ejemplo	descripcion
Byte	B0 VB10	Almacena los valores siguientes como bytes de datos, comenzando en la dirección indicada
Palabra	W0 VW22	Almacena los valores siguientes como palabras de datos, comenzando en la dirección indicada
Palabra doble	D0 VD100	Almacena los valores siguientes como palabras dobles de datos, comenzando en la dirección indicada
Autotamaño	10 V10	Almacena los datos de tamaño mínimo requerido (byte, palabra o palabra doble) requerido para almacenar los valores. Los valores introducidos en esta línea se almacenan comenzando en la dirección de memoria V que

		se haya indicado.
Conservar el Tamaño anterior	(Columna de dirección vacía)	Almacena los datos en bytes palabras o palabras dobles, dependiendo del tamaño indicado en la línea superior.

TABLA X. Identificadores válidos de tamaños.

2.- La dirección inicial y los valores de datos se deberán separar con un espacio en blanco o, un tabulador.

En la tabla IX se indican ejemplos de la notación a utilizar al introducir valores para un bloque de datos.

En la tabla X se muestra los identificadores válidos para introducir el tamaño de los datos y la dirección inicial.

3.2.11. UTILIZAR LA TABLA DE ESTADO FORZADO

Con la *tabla de estado forzado* se puede leer, escribir o forzar variables en su programa. Las entradas y salidas pueden forzarse independientemente del ciclo y con ello se puede probar de forma permanente el programa de usuario.

Dirección	Formato	Valor actual	Cambiar valor por
I0.0	Bit	2#0	2#1
I0.2	Bit	2#0	
I0.3	Bit	2#0	
Q1.2	Bit	2#1	
V0	Con signo	84	
VW2	Sin signo	4400	
VW6	Binario	2#00000010001100	
VD10	Hexadecimal	16#0064	16#65
VD14	Coma flotante	0.0000	10.0
VW20	Hexadecimal	16#0027	16#28
VW24	ASCII	AB	BA

FIGURA 3.14. Ejemplo de una tabla de estado forzado

En la figura 3.14, se muestra un ejemplo de la tabla de estado/forzado. Para leer y escribir variables con la tabla de estado forzado se debe seguir los siguientes pasos:

1. En la primera celda de la columna de direcciones, introduzca la dirección o el nombre simbólico del elemento de su programa cuyo valor desee leer o escribir. Repita este procedimiento para todos los elementos que desee insertar en la tabla.
2. Si el elemento es un bit (I, Q o M, por ejemplo), se ajusta en la segunda columna el formato binario. Dicho formato no se puede cambiar. Si el elemento es un byte, una palabra o, una doble palabra, puede seleccionar la celda en la columna "formato" para observar los formatos válidos.

3.2.12. UTILIZAR EL DIRECCIONAMIENTO SIMBOLICO

La *tabla de símbolos* permite adjudicar nombres simbólicos a las entradas, salidas y direcciones en la memoria interna (Ver figura 3.15). Los símbolos que se hayan asignado a dichas direcciones se pueden utilizar en los editores KOP, AWL, así como los editores de tabla de

estado y forzado de STEP/Micro-WIN. El editor de bloque de datos no asiste la utilización de nombres simbólicos.

La primera columna de la tabla de símbolos se utiliza para descartar una fila. Las demás columnas son para el nombre simbólico, la dirección y el comentario. En cada fila se asigna un nombre simbólico a la dirección absoluta de una entrada o salida digitales, de una dirección en la memoria, de una marca especial o, de otro segmento. Opcionalmente se puede agregar un comentario a cada símbolo asignado. Tenga en cuenta las siguientes directrices al crear una tabla de símbolos:

1. Puede introducir los nombres simbólicos y las direcciones absolutas en cualquier orden.
2. Puede utilizar hasta 23 caracteres en el campo del nombre simbólico.
3. Puede definir hasta un total de 500 símbolos.
4. En la tabla de símbolos se distingue entre mayúsculas y minúsculas. Ejemplo: "Brazo1" se considera como un símbolo diferente a "brazo1".
5. Los nombres simbólicos no pueden incluir espacios en blanco. Utilice caracteres de subrayado en vez de dichos espacios. Ejemplo : "Motor_brazo_1".

6. Los nombres y /o las direcciones simbólicas dobles se marcarán con letra cursiva color azul, no se compilarán y no se podrán utilizar en el programa. Las direcciones que se solapan no se señalan como si fueran dobles. Ejemplo: VB0 Y VW0 se solapan en esta memoria pero no son dobles.

Nombre simbólico	Dirección	Comentario
<i>Marcha1</i>	I0.0	Para borrar el contenido de una celda, pulse la tecla DEL (Supr) o la barra espaciadora
<i>Paro_Em1</i>	I0.1	para línea ensamble 1
<i>Luz_Verde1</i>	Q1.0	ra línea ensamble 1
<i>MotorMarcha1</i>	Q1.1	Motor línea ensamble 1
<i>Mezc1_Tempor</i>	T0	
<i>Mezc2_Tempor</i>	T37	
<i>Línea1_Contad</i>	C1	
<i>Relé_1</i>	M0.0	
<i>Relé_1</i>	M0.1	

FIGURA 3.15. Ejemplo de una tabla de símbolos

3.3. NUESTRA SOLUCIÓN DE AUTOMATIZACIÓN CON EL SIMATIC S7-200 CPU 212

Existen diversos métodos para crear una solución de automatización con el Micro-PLC. A continuación indicaremos algunas de las reglas generales aplicables a nuestro proyecto.

3.3.1. EL PROCESO A AUTOMATIZAR

El proceso que vamos a automatizar, podría ser parte de una gran instalación en una fábrica, cuyo producto final requiera ser embalado en cajas de madera. Estas cajas se desplazarán por medio de una banda

transportadora y, llegarán a la sección bodega de la fábrica. Las cajas necesitan ser ubicadas ordenadamente en un armario, el tiempo necesario, hasta que se solicite que sean colocadas en el camión repartidor.

3.3.2. FUNCIONES DEL PROCESO

Para poder describir las funciones del proceso, tenemos que incorporar los siguientes aspectos:

3.3.2.1. ENTRADAS

Se requieren de seis interruptores de límite, que servirán como dispositivos piloto de control y, se emplearán como entradas discretas al Micro-PLC. Utilizando una configuración mecánica de colocar dos contactos de los interruptores de límite NA correspondientes a un mismo movimiento en paralelo, se ha optimizado el número de entradas discretas usadas a la mitad. Por ejemplo, el contacto del interruptor de límite 1 se encuentra conectado en paralelo con el contacto del interruptor de límite 2 (Ver figura 3.16), de aquí sale una señal que será entrada al PLC. Los siguientes son los interruptores de límite utilizados:

1) Interruptor de Límite 1 (SW1) sirve para restringir el movimiento del carro, cuando sobrepase la posición de la columna 1.

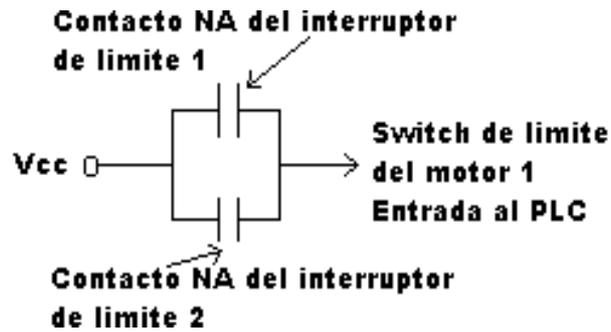


FIGURA 3.16. Contacto NA del interruptor de límite 1 conectado en paralelo con el contacto NA del interruptor de límite 2

1. **Interruptor de Límite 2 (SW1)** se emplea para limitar el desplazamiento del carro, cuando exceda la ubicación donde se recogen las cajas desde la banda.
2. **Interruptor de Límite 3 (SW2)** actúa de tal manera que prohíbe la traslación extrema del elevador hacia arriba.
3. **Interruptor de Límite 4 (SW2)** entra en operación para restringir el movimiento límite del elevador hacia abajo.
4. **Interruptor de Límite 5 (SW3)** sirve para impedir el desplazamiento de las paletas, más allá de la posición extrema que el sistema mecánico pueda ofrecer.
5. **Interruptor de Límite 6 (SW3)** actúa para limitar la traslación de las paletas, cuando éstas recojan la caja y, la contraigan hacia el carro del montacarga.

Nuestro sistema también requiere el manejo de entradas analógicas para poder controlar del movimiento de las paletas, del

elevador y del carro montacarga, por lo tanto, necesitamos manejar tres señales analógicas, que servirán de entrada al módulo analógico. Dichas señales analógicas serán enviadas por los siguientes potenciómetros:

1. **Potenciómetro 1 (POT_1)**, acoplado mecánicamente para sensar el movimiento de las paletas.
2. **Potenciómetro 2 (POT_2)**, enlazado para medir el movimiento del elevador.
3. **Potenciómetro 3 (POT_3)**, sirve para medir el desplazamiento de la posición del carro del montacarga.

2. SALIDAS

Se necesitan de seis señales de salida, para habilitar a los tres actuadores (motores). Cada uno de los actuadores, necesitan de dos señales, una para seleccionar dirección y la otra para energizarlo.

1. **Motor 1** es el que actúa sobre el sistema mecánico (*paletas*) que funciona para recoger las cajas de la banda y, también depositarlas en el estante.

2. **Motor 2** sirve para activar al *elevador*, el cual se emplea para hacer subir o bajar las cajas.

3. **Motor 3** trabaja para desplazar al *carro del montacarga* a través de los rieles desde la posición donde la banda suministra las cajas hasta el lugar correcto de la columna del estante donde las cajas serán distribuidas.

3. DESCRIPCION DEL FUNCIONAMIENTO

En la figura 3.17, se presenta el sistema mecánico que hemos elegido para automatizar. Dicho sistema debe ser capaz de ordenar (posicionar) cajas de madera en un estante, mediante el diseño de un programa. Cada una de las cajas está sobre un palette. Esta es la secuencia de eventos que el programa debe realizar:

1. El estante se encuentra dividido en tres filas (A, B, C) y tres columnas (1, 2, 3); la primera columna 1 será llenada, por lo que el programa manejará inicialmente el posicionamiento de las filas.

2. Al inicio, el sistema mecánico será posicionado justo donde pueda recoger las cajas. El programa se iniciará cuando se pulse "START", esto provocará el funcionamiento del motor de la banda, empezando así, el desplazamiento de la primera caja.

3. La primera caja se desplaza hasta que actúa sobre un interruptor de límite, ubicado donde la caja debe colocarse, para que pueda ser recogida por las paletas del montacarga. Por lo tanto, cuando el interruptor de límite sea habilitado por la caja, mandará una señal de control, la cual provocará que el motor de la banda se apague.
4. Cumplido con lo anterior, el motor de la articulación 1 (paletas), recibirá una señal que le indicará la dirección y el arranque. Esto impulsará a las paletas a recoger la primera caja.
5. Luego se ordena a que actúe el motor 2 (elevador), controlando la dirección y arranque de dicho motor, con el fin de elevar la caja unos pocos centímetros de la banda, para que pueda ser recogida. Una vez que la caja es recogida, ésta ya no habilita al interruptor de límite, por lo que el motor de la banda arrancará para desplazar a la siguiente caja.
6. Ahora debe actuar el motor de la articulación 1 (paletas), para recoger la caja y llevarla hacia el montacarga.
7. Una vez realizado el proceso anterior, la caja se desplaza junto con el montacarga hasta llegar a la posición de la columna 1. Para esto, se debe controlar el posicionamiento, la dirección y arranque del motor 3 (motor del carro).

8. El siguiente paso es colocar la caja en la posición A1 (primera fila, primera columna). Por consiguiente, el motor de la articulación 2 (elevador), debe recibir la orden de arranque y dirección, hasta ubicarse correctamente en la posición A1.
9. Una vez que el elevador está correctamente ubicado, se activa el motor que actúa sobre las paletas, para colocar la caja en la posición A1. El motor que actúa sobre el elevador bajará la caja hasta que es finalmente situada en la posición antes mencionada.
10. Posteriormente, la paleta es recogida, el elevador baja y, el montacarga es situado en la posición donde pueda recoger la siguiente caja.
11. Si existe en la banda la segunda caja, el programa repite los pasos del 4 al 7, y la caja ahora será depositada en la posición A2 del estante, es decir, cambiamos de fila pero no de columna. Los pasos 8 y 9 se realizan ahora para la posición A2. El paso 10 es igual. Cuando la tercera caja es detectada, se reproducen los pasos del 4 al 7 y, los pasos 8 y 9 se ejecutan para situar la caja en la posición A3. Por lo tanto, se ha completado el llenado de la primera columna del estante.
12. Con la siguiente caja , los pasos del 4 al 6 se repiten.

Ahora, el programa que sabe que se ha llenado la primera columna, provocará la ubicación del carro del montacarga en la columna 2. La secuencia del llenado de esta columna será igual al de la columna 1.

13. Esta parte corresponde a la ubicación de las cajas en la columna 3, cumpliéndose con el llenado de todo el estante.

14. Una vez terminada la secuencia, el programa acepta "STOP" para finalizar el proceso.

15. A pocos centímetros del extremo derecho de la columna 1 y, del extremo izquierdo de la columna 3, han sido ubicado unos topes mecánicos. En la parte superior e inferior del elevador también existen topes mecánicos. Estos sirven para evitar situaciones peligrosas. Si en algún momento del proceso, alguna articulación no se sitúa en la posición determinada por el programa y, su desplazamiento llega hasta un tope límite, éste accionará al correspondiente interruptor límite de fin de carrera, el cual mandará una señal al PLC de detener el proceso en ese instante. El operario corregirá la falla del sistema y, luego enviará una señal al PLC para que continúe el proceso.

3.3.3. LISTA DE NOMBRES SIMBÓLICOS

Para un mejor entendimiento del programa a continuación en la tabla XI

se detallará todos los elementos utilizados en la programación con sus respectivos nombres simbólicos y direcciones, con un breve comentario.

NOMBRE SIMBOLICO	DIRECCION	COMENTARIO
START	I0.0	ENCENDIDO DEL SISTEMA
STOP	I0.1	APAGADO DE LA SECUENCIA
S.O.	I0.2	SENSOR OPTICO
SW1	I0.3	SWITCH DE LIMITE DEL MOTOR 1
SW2	I0.4	SWITCH DE LIMITE DEL MOTOR 2
SW3	I0.5	SWITCH DE LIMITE DEL MOTOR 3
RE_INICIO	I0.6	REINICIA EL SISTEMA
DIR_1	Q0.0	DIRECCION DEL MOTOR 1
M1	Q0.1	MANDA ENCENDER AL MOTOR 1
DIR_2	Q0.2	DIRECCION DEL MOTOR 2
M2	Q0.3	MANDA ENCENDER AL MOTOR 2
DIR_3	Q0.4	DIRECCION DEL MOTOR 3
M3	Q0.5	MANDA ENCENDER AL MOTOR 3
POT_1	AIW0	SEÑAL ANALOGICA DEL POT-1
POT_2	AIW1	SEÑAL ANALOGICA DEL POT-2
POT_3	AIW2	SEÑAL ANALOGICA DEL POT-3
	C0	CONTADOR DE FILAS
	C1	CONTADOR DE COLUMNAS

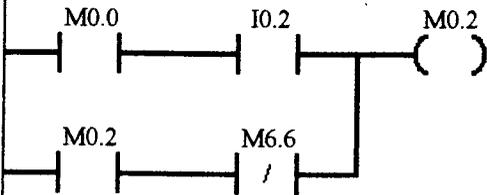
	C2	CONTADOR DE REINICIO
	T32 - T36	TEMPORIZADORES

**TABLA XI. Lista de elementos utilizados con su nombre
simbólico y dirección**

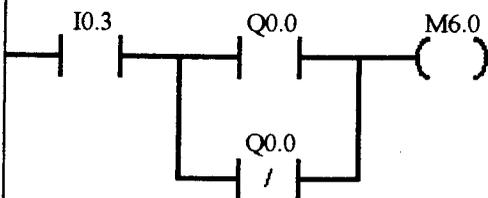
3. PROGRAMA GENERAL

El programa diseñado para controlar el sistema mecánico que va a ordenar las cajas en el estante, fue realizado en el programa Microwin/STEP – 7, el cual sirve para el CPU 212. Este es el siguiente:

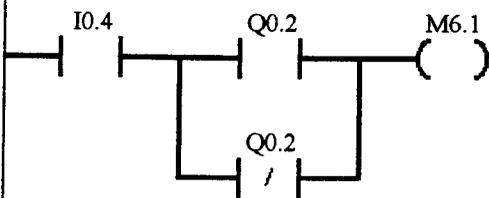
Segmento 3 COMIENZA SECUENCIA, SI DETECTO CAJA EL SENSOR OPTICO



Segmento 4 INTERRUPTOR LIMITE DE ARTICULACION 1



Segmento 5 INTERRUPTOR LIMITE DE ARTICULACION 2

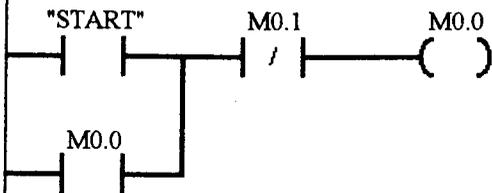


ESCUELA SUPERIOR POLITECNICA DEL LITORAL
TOPICO DE GRADUACION "SISTEMAS ELECTRONICOS INDUSTRIALES"
TEMA "ORDENADOR DE CAJAS"

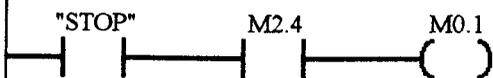
INTEGRANTES:
JULIO GOMEZ ASSAN
WILMER NARANJO ROSALES

Segmento 1 ENCENDIDO DEL PROCESO

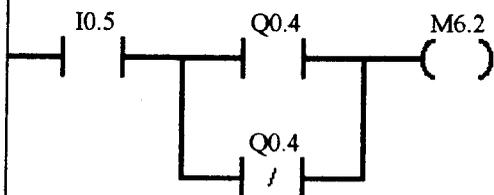
COMENTARIO DEL SEGMENTO



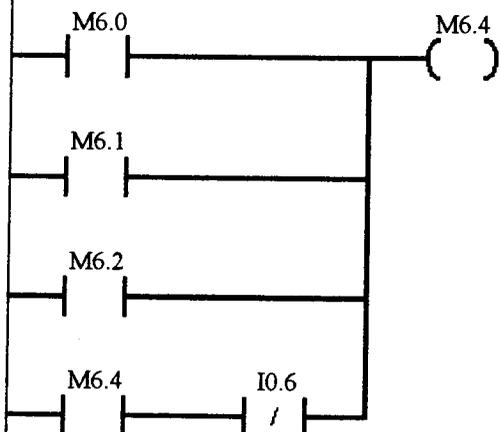
Segmento 2 APAGADO DEL PROCESO SIEMPRE QUE TERMINE LA SECUENCIA DE LLENAR EL ESTANTE



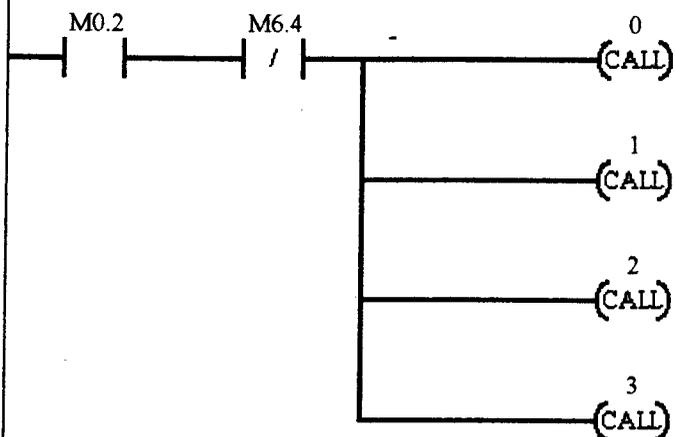
Segmento 6 INTERRUPTOR LIMITE DE ARTICULACION 3



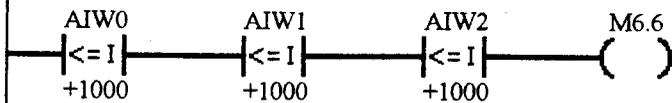
Segmento 7



Segmento 8 PROGRAMA PRINCIPAL



Segmento 9



Segmento 10 FIN DEL PROGRAMA PRINCIPAL



Segmento 11 CUERPO DE SUBROUTINA 0

0

SBR

Segmento 12

M0.2

M1.1

M1.0

/ ()

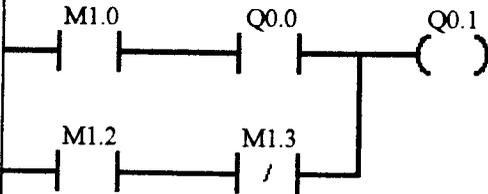
Segmento 13 ENCENDIDO DE ARTICULACION 1 Y HACIA ADELANTE

M1.0

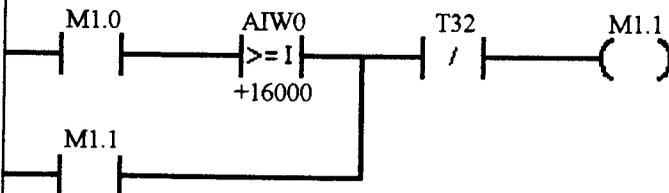
Q0.0

()

Segmento 14



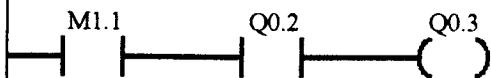
Segmento 15 LA ARTICULACION 1 SE DETENDRA SI AIW0 ES 16000



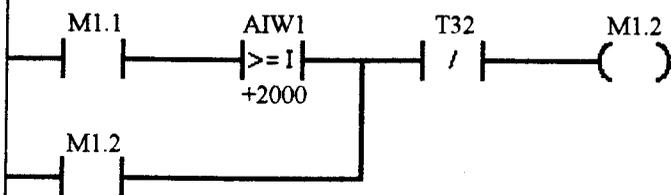
Segmento 16 ENCENDIDO DE LA ARTICULACION 2 Y MOVIMIENTO HACIA ARRIBA



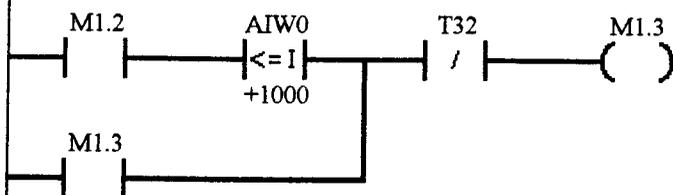
Segmento 17



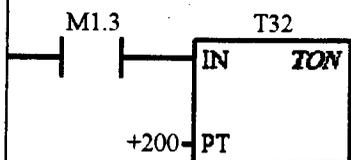
Segmento 18 DESPLAZAMIENTO DE ARTICULACION 2 SE DETENDRA SI AIW1 ES 2000



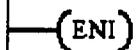
Segmento 19 RECOGIMIENTO DE ARTICULACION 1 HASTA QUE AIW0 SEA 1000



Segmento 20 TEMPORIZACION DE APAGADO DE CONTROL DE LA SUBROUTINA 0



Segmento 21 HABILITAR EVENTOS DE INTERRUPCION



Segmento 22



Segmento 23 TERMINA SUBROUTINA 0

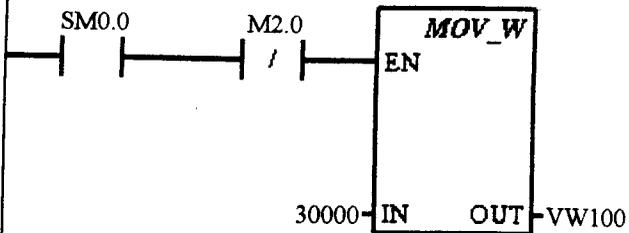
(RET)

Segmento 24 INICIO DE SUBROUTINA 1

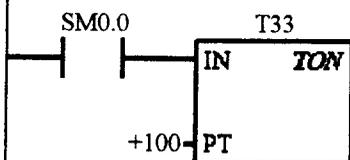
1

SBR

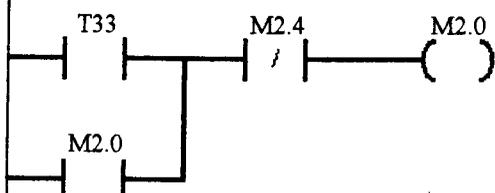
Segmento 25 CARGAR VW100 PARA LA PRIMERA COLUMNA, EN ADELANTE SE
 RESTARAN



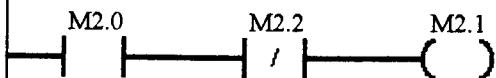
Segmento 26



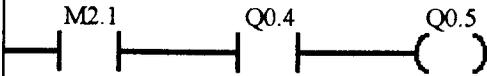
Segmento 27



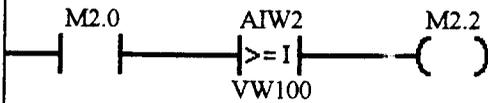
Segmento 28



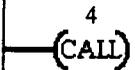
Segmento 30



Segmento 31 LA POSICION 3 SE DETENDRA SI AIW2 ES MAYOR O IGUAL A VW100



Segmento 32 LLAMADA A SUBROUTINA 4 PARA POSICIONAR EN OTRO COLUMNA



Segmento 33 RETORNO DE SUBRITINA

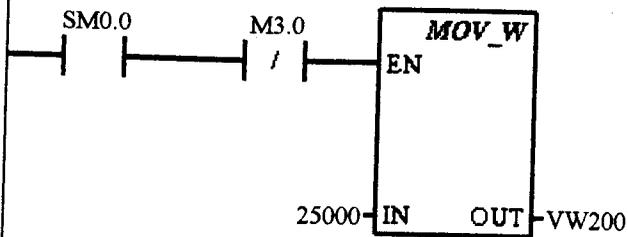
(RET)

Segmento 34 SUBROUTINA 2 "PARA MANEJAR EN QUE FILA ALMACENARA "

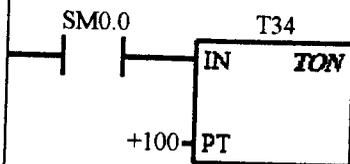
2

SBR

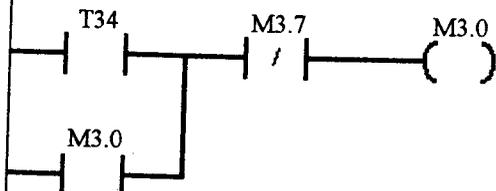
Segmento 35 CARGAR EN EL PRIMER MOVIMIENTO VW200 CON 25000



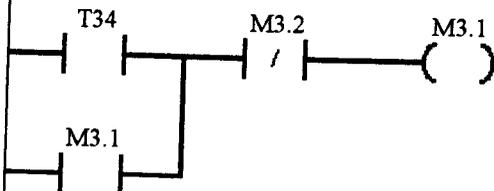
Segmento 36



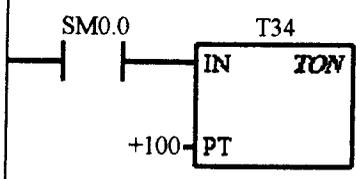
Segmento 37



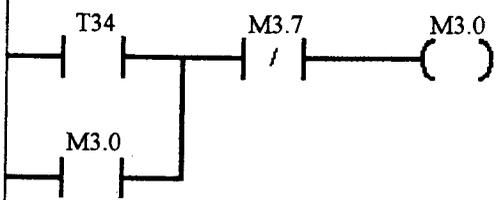
Segmento 38



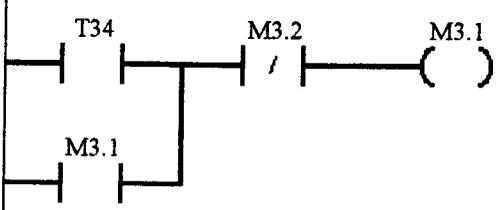
Segmento 36



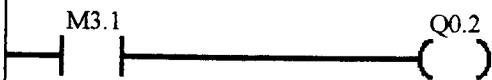
Segmento 37



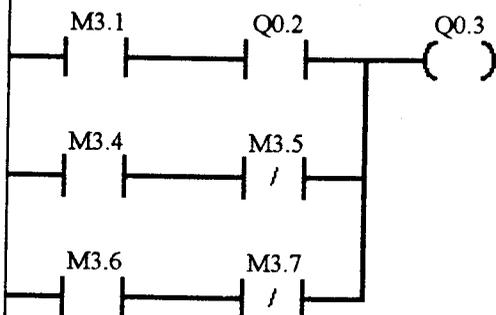
Segmento 38



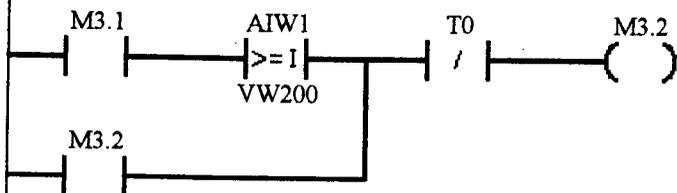
Segmento 39 ENCENDIDO DE LA ARTICULACION 2 Y MOVIMIENTO HASTA VW200



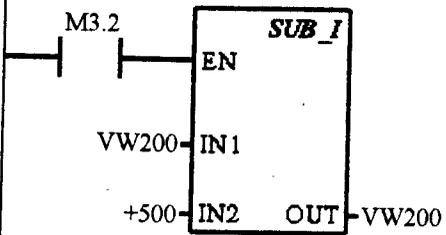
Segmento 40



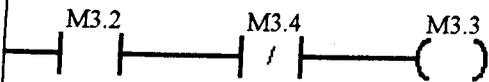
Segmento 41 ARTICULACION 2 PARARA SI AIW2 ES MAYOR O IGUAL A VW200



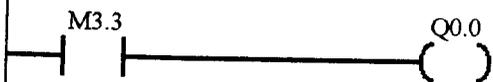
Segmento 42 RESTARA VW200 EN 500 PARA COLOCAR LA CAJA EN EL ESTANTE



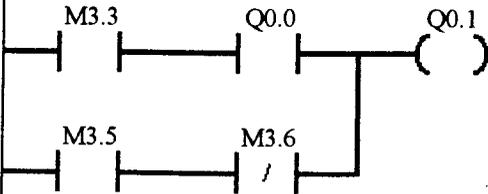
Segmento 43



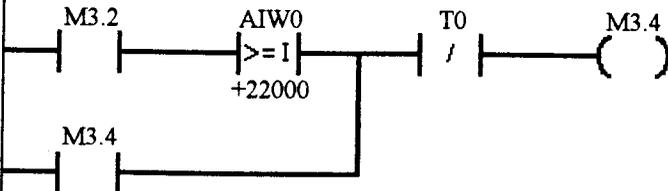
Segmento 44



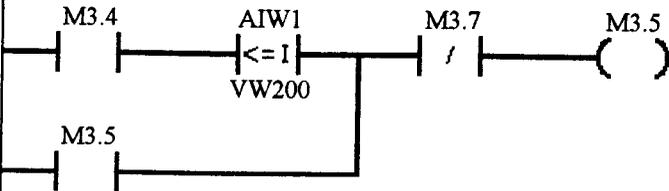
Segmento 45 ENCENDIDO DE ARTICULACION 1 QUE IMPULSARA LA CAJA EN EL ESTANTE



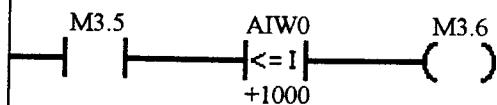
Segmento 46 CUANDO SE INYECTE LA CAJA Y AIW1 SEA MAYOR O IGUAL A 2200 SE PARA



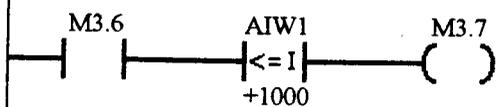
Segmento 47 LA ARTICULACION 2 BAJARA, HASTA EL VALOR VW200 QUE SE RESTO



Segmento 48 UNA VEZ COLOCADA LA CAJA LA ARTICULACION 1 SE RECOGERA



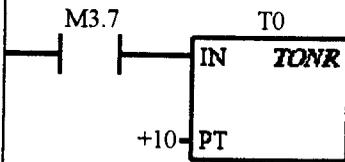
Segmento 49 LA ARTICULACION 2 EMPEZARA A RECOGERSE



Segmento 50 LLAMAR A SUBROUTINA 5 PARA COLOCAR LA SIGUIENTE CAJA A UNA NUEVA FILA



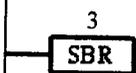
Segmento 51



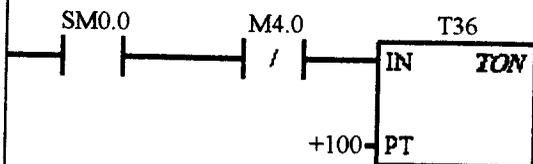
Segmento 52 RETORNO DE SUBROUTINA SI ACTUA CUALQUIER DE ESTOS CONTACTOS



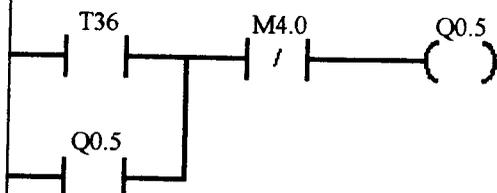
Segmento 53 INICIO DE SUBROUTINA 3



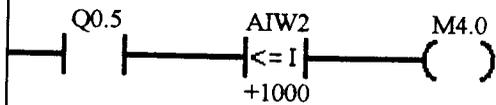
Segmento 54



Segmento 55 RETORNO DE ARTICULACION 3 HASTA LA POSICION INICIAL



Segmento 56 MOVIMIENTO DE ARTICULACION 3 SE DETENDRA EN POSICION INICIAL



Segmento 57

M4.0

(RET)

Segmento 58

(RET)

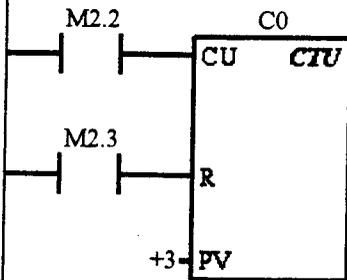
Segmento 59

INICIO DE SUBROUTINA PARA CAMBIAR LA COLUMNA DEL SIGUIENTE MOVIMIENTO

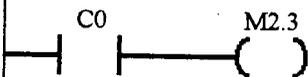
4

SBR

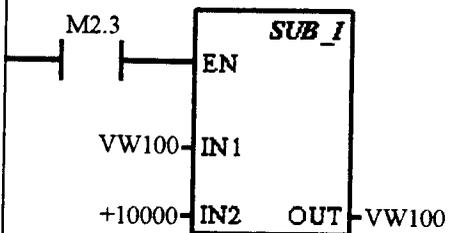
Segmento 60 CUENTA LOS MOVIMIENTOS DE FILA Y SI REALIZA 3 ENTONCES CAMBIARA DE COLUMNA



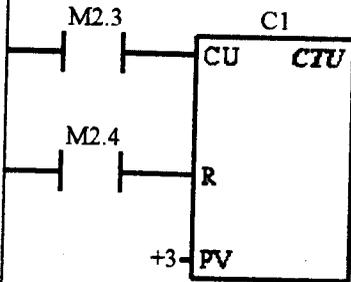
Segmento 61



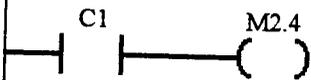
Segmento 62 SI REALIZO LOS 3 MOVIMIENTOS DE FILA ENTONCES RESTARA LA POSICION DE LA COLUMNA



Segmento 63 CONTARA LOS TRES MOVIMIENTOS DE LAS COLUMNAS



Segmento 64 SI CONTO LAS 3 SECUENCIAS ENTONCES HABILITA A LA ENTRADA DE "STOP"



Segmento 65

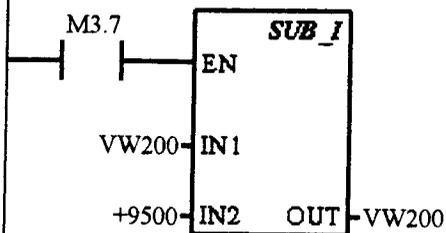


Segmento 66 SUBRTINA PARA CAMBIAR LA FILA DEL ESTANTE DE LA SIGUIENTE SECUENCIA

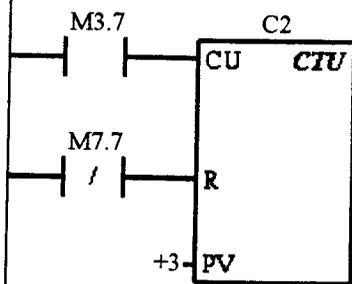
5

SBR

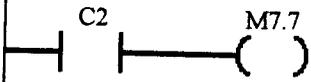
Segmento 67 RESTAR VW200 PARA ELEGIR LA NUEVA FILA



Segmento 68 CONTARA SI SE REALIZO LOS 3 MOVIMIENTOS DE FILA PARA REINICIAR VW200



Segmento 69



Segmento 70



CAPITULO IV

EL VISUALIZADOR DE PROCESOS INTOUCH

4.1. INTRODUCCION

En el capítulo anterior se ha diseñado el programa de automatización para el proceso elegido. Ahora, como otra herramienta de control, requerimos adquirir datos, controlar, visualizar y, supervisar la secuencia de los movimientos de dicho proceso mecánico, desde el monitor de una computadora. Para esto nos ayudaremos del software *InTouch 5.6b* de Wonderware Corporation, versión Evaluación.

Debido a la alta competencia en el mercado mundial, es necesario crear aplicaciones de proceso de tal manera que las fábricas funcionen de forma más eficiente y con mayor productividad. En el mercado mundial existen varias empresas que se han dedicado a desarrollar software especializados en automatización industrial.

En este capítulo, explicaremos el desarrollo de las representaciones gráficas, de los enlaces de animación y, de los *Script* necesarios para lograr la simulación del movimiento y el ordenamiento de las cajas en el estante por medio del carro montacarga. Esto significa que hablaremos de como se van a comportar los diferentes objetos; y de la forma de como le daremos vida eligiendo los adecuados enlaces de animación.

4.2. PRINCIPALES ASPECTOS DE INTOUCH

InTouch es un software utilizado para crear y desarrollar una interface gráfica humano-máquina (HMI) con el operador basado en una PC. Es una herramienta potente y flexible, diseñada para automatizar, monitorear, supervisar y controlar procesos.

Permite a los operadores, ingenieros, administradores y supervisores, observar en pantalla mediante representaciones gráficas de procesos en tiempo real, toda las operaciones de una planta industrial. InTouch sirve como un sistema de interface con el operador (OIS), posee una arquitectura abierta, permitiendo al sistema ejecutarse en un ambiente compartido con soporte en línea.

Usando In Touch, las aplicaciones desarrolladas pueden explotar las principales características del Microsoft Windows, incluyendo el Intercambio Dinámico de

Datos (Dynamic Data Exchange DDE), Enlace de Objetos y Empotrado (Object Linking and Embedding OLE) y gráficos. También se puede lograr un intercambio dinámico de datos (DDE) con otras aplicaciones de control avanzado tales como: Sistemas Expertos, Hojas de Cálculo y, diferentes programas de base de datos.

El DDE en línea puede efectuarse tanto con aplicaciones que estén siendo ejecutadas en el nodo local, como aquellas que se ejecutan en otros nodos, dentro de una red en arquitectura Cliente-Servidor. El OIS tiene la capacidad de interface con una base de datos, que posean una interface de Lenguaje Estructurado Requerido (SQL) y, además, provee un análisis para el Control Estadístico de Procesos (SPC) en línea.

4.3. FUNCIONES DE INTOUCH

InTouch posee un grupo de funciones que aumentan su rendimiento. Estas son:

1. GRÁFICO ORIENTADO A LOS OBJETOS

El sistema de elaboración de gráficos está orientado a objetos, logrando que los objetos y grupos de objetos puedan moverse, redimensionarse y animarse de forma más rápida y sencilla que los gráficos de mapa de bits. Las herramientas de desarrollo de gráficos permite la creación de rectángulos rellenos, círculos, elipses, polígonos, arcos, y demás objetos, cuya forma sólo es limitado por la imaginación del ingeniero. El usuario tiene la posibilidad de ordenar los objetos gráficos basado en los siguientes comandos:

- Alinear arriba
- Alinear abajo

- Alinear a la izquierda
- Alinear puntos centrales
- Espacio vertical
- Enviar adelante
- Enviar atrás
- Espacio horizontal
- Rotar en sentido de las manecillas del reloj
- Rotar en sentido contrario a las manecillas del reloj
- Agrupar objetos dentro de una celda

En la figura 4.1, se aprecia la Caja de herramienta (Toolbox) que es normalmente visible en *WindowMaker*, siempre que una ventana es abierta. Esta Caja de herramienta contiene todos los instrumentos necesarios para crear y manipular objetos, o para realizar otras funciones dentro de una ventana de aplicación.

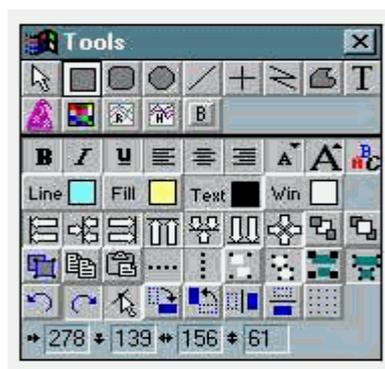


FIGURA 4.1. Caja de herramientas (ToolBox)

Como observamos en la figura 4.1, en la parte de debajo de la Caja de

Herramienta, se muestran unos números, éstos dan información acerca del tamaño de un objeto tanto cuando es dibujado como cuando es seleccionado. Cuando el cursor se encuentra encima de la Caja de herramienta, aparecerá la información del área escogida, y cuando es colocado en un área en blanco de la ventana, mostrará su situación en la ventana.

El *editor de gráficos* permite la estratificación de los objetos con el fin de activarlos basado en ciertas condiciones del proceso. Las herramientas de desarrollo de gráficos permiten la función "*deshacer / rehacer*", con un número configurable de niveles y visualización de los comandos. El sistema es capaz de importar los archivos .DXF con la posibilidad de animar tales objetos. También se pueden importar dibujos e imágenes en el formato de archivo .BMP. Los objetos gráficos animados o los símbolos del proceso pueden copiarse desde una ventana o pantalla a otra reteniendo todas sus características de animación consiguiendo así la eliminación del esfuerzo en la duplicación. Adicionalmente, es posible importar ventanas desde otra aplicación en la misma forma. Todos los objetos tienen la posibilidad de ser ubicados en cualquier ventana y con cualquier configuración.

El usuario tiene la capacidad de efectuar la búsqueda de cualquier nombre de identificación mientras está construyendo una pantalla, y obtener una descripción detallada del ítem (puntos de consigna, dirección de entrada/salida y todos los detalles del diccionario), mientras está construyendo una pantalla sin salir del editor de gráficos. Otro aspecto importante es que se puede configurar pantallas gráficas mientras que el sistema se encuentra monitoreando el proceso.

4.3.2. CARACTERISTICAS DE VISUALIZACION

Las principales facilidades de visualización que posee el operador cuando se encuentra utilizando el sistema HMI son:

1. El operador tiene la posibilidad de acceder a las facilidades de visualización, por medio de un dispositivo apuntador y/o menús por medio del teclado. Los dispositivos apuntadores soportados incluyen un ratón, una pantalla sensible al tacto, o un lápiz luminoso.
2. El sistema brinda el acceso del operador a múltiples pantallas de visualización a la vez, incluyendo partición de pantallas donde se puede visualizar más de una área del proceso al mismo tiempo. Adicionalmente, el sistema soporta el uso ilimitado de pantallas de aparición repentina para ayuda adicional, o información de algún suceso que haya ocurrido.
3. El acceso a todas las pantallas de visualización y a las demás funciones de comando, está dado por un nivel de acceso de seguridad, que protege al sistema contra el uso por parte de operadores no autorizados. El nivel de acceso de seguridad se establece durante el procedimiento de identificación del operador. Un sistema de control supervisorio es utilizado para controlar procesos delicados y equipos de alto costo, es por esto, que la protección es esencial para prevenir acciones no autorizadas, o daños accidentales al sistema.

1. La visibilidad y operación de los botones de comando y símbolos, se pueden controlar de acuerdo al nivel de acceso de seguridad.
2. El sistema permite al usuario ver gráficos animados del proceso, incluyendo tanques, bombas, válvulas y motores.
3. Se puede lograr el llenado del objeto en porcentaje, incluyendo formas irregulares como polígonos y elipses.
4. Se pueden utilizar 32 colores para realizar cualquier gráfico de animación.
5. Los objetos pueden ser animados y visualizados de acuerdo a cualquiera de las ocho diferentes condiciones de alarmas o, basado en cualquier condición del sistema.
6. El sistema permite la animación por medio de un cambio de tamaño, movimiento y/o rotación de los objetos, de acuerdo a la modificación que ocurra en una variable del proceso.
7. Los objetos pueden ser animados basados en cualquier criterio definido por el usuario, compuesto por el estado en que se encuentran otros puntos específicos dentro del sistema. Esto incluye el uso de expresiones que contengan todas las funciones matemáticas y, el estado de los valores discretos o análogos del sistema.

8. Los objetos pueden habilitarse para destellar o, cambiar de color, de acuerdo al resultado de la evaluación de cualquiera de los 32 bits de un registro análogo.

4.3.3. DESCRIPCION DE TEXTO

El sistema permite el uso de estilos y tamaños de caracteres de tipo real, que pueden tener el tamaño deseado. Los principales aspectos para la descripción de textos son:

1. Las fuentes pueden ser cargadas por medio del sistema operativo. El operador puede elegir entre 32 colores diferentes para los textos.
2. El sistema permite el cambio de color del texto, fundamentado en un valor del proceso que se encuentre dentro de cualquiera de los ocho estados diferentes de alarma.
3. El texto puede habilitarse para que destelle basado en cualquier condición que ocurra en el sistema, definida por el usuario; bien sea una alarma, un punto de consigna particular, o basado en el valor actual de una variable del proceso.
4. El sistema puede visualizar valores de proceso, fundamentado en el nivel de acceso de seguridad.

5. El texto puede ser visible o invisible, de acuerdo a una condición de alarma del proceso, o cualquier otro cambio de estado del sistema.

4.3.4. DICCIONARIO DE DATOS (TAGNAME DICTIONARY)

El *diccionario de base de datos* o *diccionario de tagname* (tagname dictionary) es el corazón de InTouch. Para crear la base de datos en runtime, InTouch requiere la información necesaria acerca de todas las entradas y datos (tagname) que están siendo creados. Cada variable necesita ser asignada a un tipo de tagname. Un *tagname* es un nombre simbólico que es ingresado al diccionario de base de datos. Este nombre simbólico puede ser configurado (valor mínimo, máximo, alarmas, etc.) y definido con un tipo específico, por ejemplo, un tagname DDE. El tagname DDE llega a ser el enlace entre InTouch, el servidor de E/S y el *mundo real*. El diccionario de tagname es el mecanismo usado para ingresar la información necesaria acerca de las variables y entradas, dentro de la base de datos.

La creación del diccionario de tagnames puede ser logrado mediante tres diferentes métodos. El primer método es el *método manual*, donde se accede al diccionario de tagnames y se define cada tagnames individualmente, hasta completar la base de datos, de manera similar a la creación de la tabla de E/S en un PLC. El segundo método es el *método automático*, de tal manera que cuando se crea un objeto gráfico, se le asignan los enlaces de animación y el sistema define dicho tagname. El tercer método es el *método externo*, donde se utilizan los utilitarios de Wonderware, el DB Dump (descarga) y el DB Load (carga), para transferir la base de datos desde una aplicación de InTouch a otra aplicación de InTouch.

METODO MANUAL. Para acceder a la caja de diálogo del diccionario de tagname usando el método manual, se selecciona el comando Special/Tagname Dictionary. Cuando realizamos lo anterior, aparecerá la caja de diálogo, igual a la mostrada en la figura 4.2.



FIGURA 4.2. Caja de diálogo para definir al diccionario de tagname

METODO AUTOMATICO. Para acceder a la caja de diálogo usando el método automático, primero se anima al objeto utilizando la caja de diálogo de los enlaces de animación, si al objeto que hemos seleccionado no se le ha asignado tagname alguno, entonces WindowMaker rápidamente hará aparecer el mensaje mostrado en la figura 4.3.

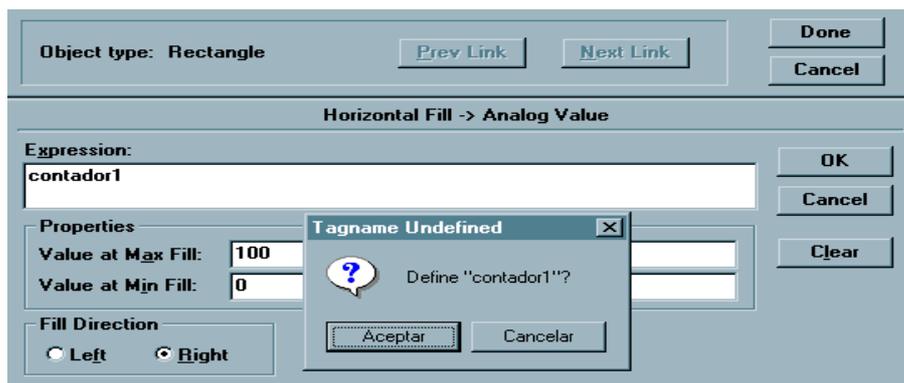


FIGURA 4.3. Ejemplo de pantalla que aparece cuando un tagname no ha sido definido

Los índices de los tagnames son asignados automáticamente. Por

ejemplo, si el primer tagname ingresado es R4001, el siguiente tagname será creado como R4002, y así sucesivamente.

4.3.5. TIPOS DE TAGNAME

Existen diferentes tipos de tagname de acuerdo a su uso. Por ejemplo, si los valores de un tagname son leído o escrito desde otra aplicación de Window, como un servidor DDE, este tagname será un tagname tipo DDE. Debemos también conocer si el tag es discreto, tal como una simple entrada de PLC, o un tag análogo, como un registro de 16, 32 o 64 bits. Los tags análogos son de dos tipos: enteros y reales.



FIGURA 4.4. Diferentes tipos de tagnames

En la figura 4.4, se muestran los diferentes tipos de tagnames. Los tipos de tagnames existen solamente dentro del programa InTouch. Ellos pueden ser usados para crear constantes, demos y simulaciones, también para crear variables y poder acceder a otros programas de Windows. En las simulaciones, los tagnames de memoria pueden ser usados para controlar las acciones de las escrituras lógicas (logic script). Por ejemplo, un tagname de memoria "CONTADOR1" puede ser cambiado por medio de la acción de un pulsador, para causar varios efectos de animación. Existen cuatro tipos de tagname de memoria:

Memoria Discreta. Es un tagname interno discreto que puede tomar el valor de 0 (Falso, Apagado), o 1 (Verdad, Encendido).

Memoria Entera. 32 bits que significan valores enteros entre $-2.147' 483.648$ y $2.147' 483.647$.

Memoria Real. Con un punto flotante decimal. El valor del punto flotante deberá estar entre $\pm 3.4e^{38}$. Todos los cálculos con puntos flotantes son ejecutados con una resolución de 64 bits, pero el resultado es guardado en 32 bits.

Memoria Mensaje. Tagname de texto que puede tener un largo de 131 caracteres.

Todos los tagnames que leen y escriben valores hacia o desde otro programa de Windows son tagnames DDE. Estos incluyen todas las entradas y salidas desde controladores programables PLC, computadoras de procesos, otros programas de Windows y desde nodos de red. Los tagnames DDE son ingresados por medio del protocolo de Microsoft Intercambio Dinámico de Datos. Cuando el valor de lectura/escritura de un tagname tipo DDE cambia, esto es inmediatamente escrito en la aplicación remota vía DDE. El tagname también puede ser actualizado desde la aplicación remota siempre que el dispositivo al cual el tagname se encuentra enlazado, cambie en la aplicación remota. Los datos remotamente adquiridos son guardados automáticamente así como también los cambios de datos que suceden en la fuente. Wonderware

ofrece Servidores DDE para la mayoría de los fabricantes de dispositivos de control y sus variados protocolos. Existen cuatro tipos de tagnames DDE:

DDE Discreto. Tagname discreto de E/S con valores de 0 (Falso, Apagado), o 1 (Verdadero, Encendido).

DDE Entero. 32 bits que significan valores enteros entre $-2.147' 483.648$ y $2.147' 483.647$.

DDE Real. Con un punto flotante decimal. El valor del punto flotante deberá estar entre $\pm 3.4e^{38}$. Todos los cálculos con puntos flotantes son ejecutados con una resolución de 64 bits, pero el resultado es guardado en 32 bits.

DDE Mensaje. Tagname de texto que puede tener un largo de 131 caracteres.

Existen diferentes tipos de tagnames que ejecutan funciones complejas, como por ejemplo, la muestra de alarmas dinámicas y tendencias históricas, monitoreo y/o cambio de tagnames de cada pluma de tendencia histórica, y la consignación de tagnames hacia múltiples fuentes usando tagnames indirectos. Los siguientes describen los diferentes tipos de tagnames:

Group Var. Este tipo de tagname es asignado a los tagnames que tienen el nombre de un Grupo de Alarmas asignados a él. Esto es muy utilizado

en el despliegue de alarmas, discos de registro e impresión dinámica de registro. Las ventanas de alarma o las alarmas de registro pueden ser configuradas para desplegar todas las alarmas asociadas con el Group Var. Un tagname Group Var puede ser usado para crear pulsadores que seleccionarán el despliegue de alarmas para diferentes partes de una planta en la misma ventana de alarmas. Todos los campos asociados con los grupos de alarmas (Alarm Group) pueden ser aplicados al Group Var.

Hist Trend. Este tipo de tagname es asignado a un tagname que será utilizado para un gráfico de Tendencia Histórica. Cuando se configura una carta de Tendencia Histórica, InTouch requiere que el tagname tipo Hist Trend sea asignado a la carta.

Tag ID. Este tipo de tagname es utilizado para conseguir información acerca de los tagnames que están siendo controlados en la carta de Tendencia Histórica. El uso más común es para desplegar el nombre del tagname controlado. Por ejemplo, la Pluma4 de una carta de Tendencia Histórica es desplegada con el tagname de "Análogo1". Una ventana puede ser creada con una salida de campo que muestra el tagname asignado a la Pluma4.

4.3.6. ENLACES DE ANIMACIÓN

Los ***enlaces de animación*** pueden combinarse para ofrecer movimientos, colores, cambios de tamaños y/o de posición a los objetos. Incluyen entradas de contacto discretas y analógicas; deslizadores horizontales y verticales; pulsadores discretos y de acción; pulsadores para mostrar y ocultar ventanas; enlaces de color de línea, relleno y texto para valores y

alarmas discretos y analógicos; enlaces de altura y ancho de objetos; enlaces de posición horizontal y vertical, y demás.

Una vez que un objeto o símbolo ha sido creado, este puede ser animado eligiendo los correctos enlaces de animación. Los enlaces de animación provocan que los objetos, símbolos o celdas cambian su apariencia por medio de la variación del valor de la base de datos de los tagnames.

Cuando un objeto es seleccionado y el comando Special/Animation Links es elegido, entonces aparecerán dos cajas de diálogos, la descripción del Item y la selección de los enlaces.

En la figura 4.5, se muestra la caja de diálogo que es común para crear todos los enlaces. La parte superior de la caja muestra el objeto seleccionado elegido para realizarle los enlaces de animación. Un objeto o símbolo puede tener múltiples enlaces. La habilidad de combinar estos enlaces de animación provee la habilidad de crear cualquier animación de pantalla de efecto inimaginable.

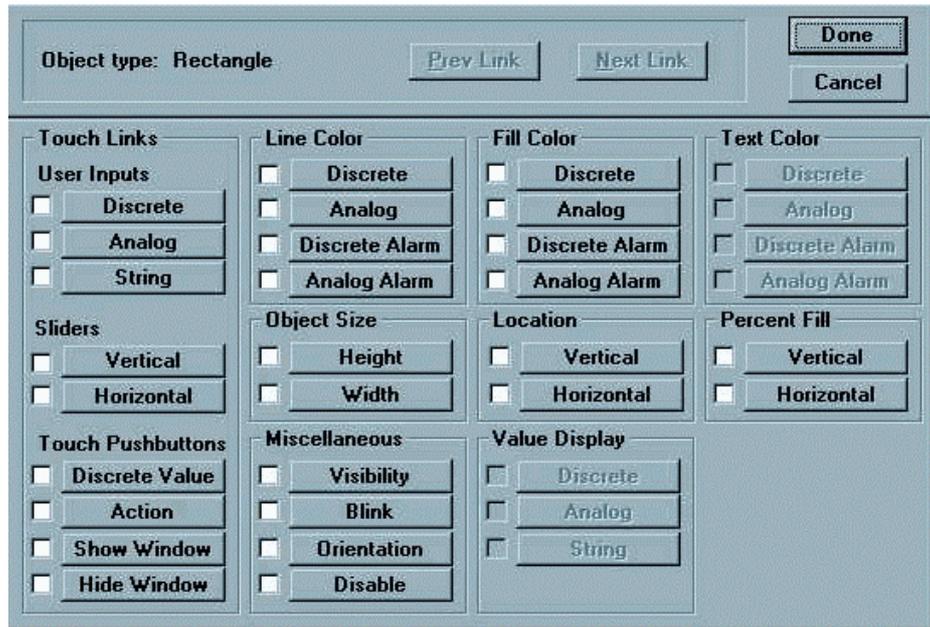


FIGURA 4.5. Caja de diálogo para definir los enlaces de animación

Son varios los Enlaces que disponemos en InTouch, estos son:

Enlaces de Toque (Touch Links). Esta clase de Enlaces hacen que un objeto o símbolo funcione como un botón o pulsador sensible al tacto. Un botón sensible al tacto puede ser activado por medio del ratón, del toque de la imagen en la pantalla, presionando una tecla asignada o la tecla Enter.

Enlaces de Color (Color Links). Con estos Enlaces es posible animar los atributos de las líneas de color, el llenado y/o el color del texto de un objeto. Cada uno de estos atributos de color puede ser realizados de manera dinámica por medio de la definición de los Enlaces de Color. Los atributos de color pueden ser enlazados al valor de una expresión discreta o análoga, o al estado de una alarma discreta o análoga.

Enlaces de Color según el Estado de una Alarma. El color del texto, las líneas y el llenado de un objeto pueden ser enlazados al estado de una alarma de una Tagname, Group Name o un Group Var. Existen dos tipos básicos de Enlaces de Color Orientado a las Alarmas, el primer tipo se define de acuerdo al Estado de la Alarma Discreta, la cual puede ser usada con cualquier tipo de variable. El segundo tipo es según el Estado de la Alarma Análoga, la cual solamente es aplicable a variables análogas.

Enlaces de Tamaño del Objeto. Los enlaces de tamaño son usados para variar el ancho y/o el largo de un objeto de acuerdo al valor de una expresión análoga (entera o real). Los enlaces de ancho y largo pueden ser elegidos a un mismo objeto.

Enlaces de Movimiento. Los Enlaces de Movimiento son utilizados para hacer que un objeto automáticamente se mueva horizontal, vertical o en ambas direcciones, en respuesta a los cambios en el valor de una expresión.

Enlaces de Porcentaje de Llenado. Los Enlaces de Porcentaje de Llenado son usados para variar el nivel de llenado de un símbolo, de acuerdo a los valores de una expresión análoga. Por ejemplo, este enlace puede ser usado para mostrar el nivel de líquido de un recipiente. Un objeto o símbolo puede tener un llenado horizontal, vertical o ambos.

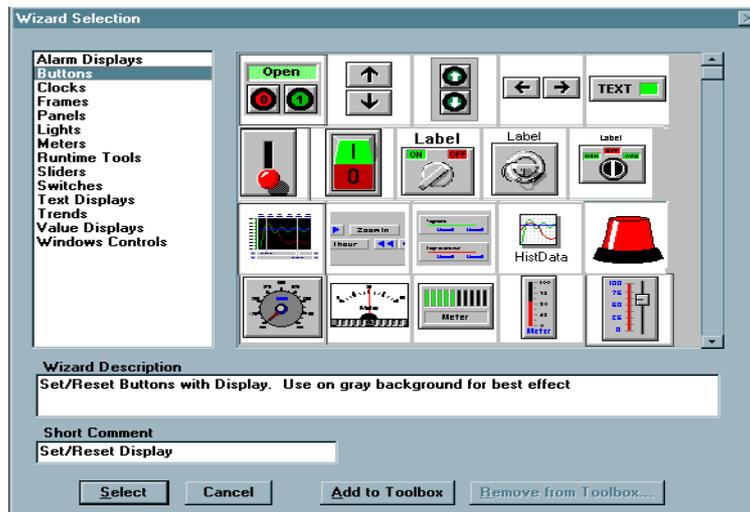
Miscelánea de Enlaces. Existen cuatro Miscelánea de Enlaces: de Visibilidad, Parpadeo, Orientación y No Habilitado. Los ***Enlaces de Visibilidad*** controla la visibilidad de un objeto dependiendo del valor de un tagname o expresión discreto. Este objeto puede ser usado para hacer que los objetos aparezcan o desaparezcan. Los ***Enlaces de Parpadeo***

pueden hacer que un objeto parpadee basado en el valor de un tagname discreto o de una expresión. Los *Enlaces de Orientación* pueden hacer un objeto rote en un eje específico, basado en el valor de un tagname o de una expresión. Los *Enlaces No Habilitado* previenen el uso de un usuario particular o de un grupo de usuarios de tener acceso a un botón si su \$AccessLevel es demasiado bajo. Por ejemplo, si el Enlace No Habilitado es verdad, el texto de un botón tridimensional se encuentra en un color gris.

Enlaces de Despliegue de Valores. Los Enlaces de Despliegue de

Valores nos dan la habilidad de usar objeto texto para mostrar el valor de una variable discreta o análoga.

4.3.7. ASISTENTES



InTouch posee una biblioteca completa de asistentes complejos preconfigurados llamada *wizard selection*. El wizard selection posee una gama de interruptores, deslizadores, alarmas, botoneras, paneles, luces pilotos, switches, tendencias en tiempo real e históricas y medidores, que los usuarios

pueden modificar y duplicar libremente. Es frecuente que los asistentes utilizados puedan añadirse a la barra de herramientas de InTouch para facilitar el acceso durante el desarrollo de aplicaciones.

FIGURA 4.6 Ventana para la Selección del Wizard

En la figura 4.6, se muestra la ventana de Selección del Wizard, la cual sirve para ahorrar tiempo durante el desarrollo de una aplicación. De esta manera, no se invertirá mucho tiempo en dibujar los componentes individuales para los objetos que son requeridos.

4.3.8. FUNCIONES DE COMANDO / CONTROL

Las siguientes son las *funciones de comando / control* más destacadas:

1. Un operador puede controlar un punto discreto utilizando un botón de comando de acción. Este control incluye encendido momentáneo, apagado momentáneo, intercambio de encendido / apagado, ajuste y reposición.
2. El operador puede utilizar los botones de comando para ajustar puntos de consigna en sentido ascendente y descendente, sobre una base porcentual o absoluta. Cada solicitud de incremento o decremento es comparada contra los límites de operación válidos, antes de permitir el ajuste.
3. El control sobre los puntos de consigna individuales, se habilita basado

en el nivel de acceso de seguridad y por medio de una palabra de paso.

4.3.9. CONTROL LOGICO DE LA APLICACIÓN

El sistema tiene la habilidad para ejecutar una serie de comandos lógicos definidos por el usuario. Los comandos lógicos se crean en un ambiente de programación basado en declaraciones, del cual no se requiere de compiladores ni de encadenadores. El control lógico consta de las siguientes partes:

4.3.9.1. SISTEMA LÓGICO

El sistema lógico tiene la capacidad de efectuar funciones automáticamente, tales como:

- Incrementar los puntos de consigna
- Efectuar totalización
- Verificar el estado de los puntos de consigna del proceso para tomar acciones.

El sistema lógico tiene la capacidad de controlar y ejecutar otros programas de aplicación que deban correr en un ambiente compartido. Se puede monitorear el estado de cada variable del proceso dentro del sistema y ejecutar funciones específicas con

base a los siguientes parámetros:

- Estado normal
- Estado de alarma
- Estado de alarma bajo-bajo
- Estado de alarma bajo
- Estado de alarma alto
- Estado de alarma alto-alto
- Resultado de una operación en lógica booleana (<, >, no, y, o, equivalencia, no equivalencia)
- Estado de los bits
- Estado de alarma reconocida
- Estado de alarma no reconocida

El sistema tiene la capacidad de efectuar control de la aplicación para poner o sacar de servicio puntos discretos, mostrar ventanas, descargar recetas. Esta lógica de la aplicación puede también arrancar y detener otros programas de aplicación en el ambiente multitarea, incluyendo hojas electrónicas, programa de base de datos y programas de almacenamiento de recetas. La lógica condicional tiene una capacidad hasta 32.767 bytes de memoria.

4.3.9.2. EDITOR LOGICO

Se dispone de un editor para el desarrollo de un listado de instrucciones lógicas. El usuario puede configurar y editar un listado de instrucciones lógicas mientras que el sistema se encuentre monitoreando el proceso.

4.3.9.3. LÓGICA DE CONTROL CONDICIONAL

El sistema tiene la capacidad de efectuar un control en base al estado de una variable del proceso (temperatura, presión, razón de flujo de fluido, concentración química, humedad, viscosidad, posición mecánica, velocidad mecánica, etc.) o, al resultado de una expresión que encierre múltiples variables del proceso. Esto incluye, el estado Encendido/Apagado de una variable discreta, o a los estados de alarma. Cada listado de instrucciones para la determinación de una condición lógica puede soportar hasta 32.767 bytes de memoria que sostiene la ejecución y el control de la lógica.

4.3.9.4. LÓGICA DE CONTROL DE TECLADO

El sistema se ha diseñado con la capacidad de efectuar un control, con el simple trabajo de presionar una o varias teclas del teclado de la PC. Es decir, cada vez que una tecla sea presionada, liberada, o cuando la tecla se mantenga presionada durante un intervalo preestablecido, provocará una acción en el sistema. El sistema soporta cualquiera de las teclas de un teclado de una PC. La lógica de control del teclado es capaz de soportar hasta 32.767

bytes de memoria y sostener la ejecución y el control de la lógica.

4.3.10. ESCRITURA LOGICA (LOGIC SCRIPT)

El *lenguaje Script* de InTouch es flexible, poderoso y fácil de utilizar, de tal manera que se pueden crear scripts simplemente apuntando con el ratón de la computadora, sin necesidad de tocar el teclado.

El Wonderware scripts incrementa la capacidad de InTouch de proveer la habilidad de ejecutar comandos y operadores lógicos. Ejemplo de ello, puede ser: una tecla que está siendo presionada, una ventana que está siendo abierta, un valor que está cambiando, para nombrar unos pocos. Utilizando los scripts, una gran variedad de sistemas personalizados y automáticos pueden ser creados. Existen seis tipos de

scripts:

- Application Scripts
- Window Scripts
- Key Scripts
- Condition Scripts
- Data Change Scripts

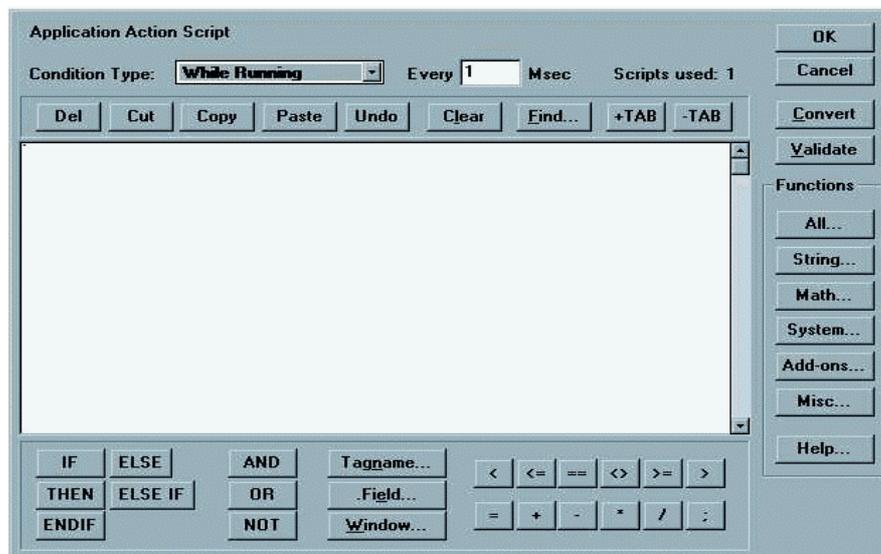


Pushbutton Action Scripts

FIGURA 4.7. Ventana del menú para encontrar los Scripts

Los comandos utilizados para crear estos scripts se encuentran localizados en el menú Special/Scripts, como podemos observar en la figura 4.7.

El *Application Scripts* son enlaces que sirven para desarrollar aplicaciones, crear simulación de procesos, para calcular variables y



mucho más. Esta clase de script es el que utilizamos en nuestro proyecto.

FIGURA 4.8. Ventana de diálogo del editor de Script

En la figura 4.8, mostramos la ventana de diálogo de los script. Aquí es donde se ingresa la escritura lógica para crear simulaciones de proceso, también para el cálculo de variables. Cabe resaltar que el Script posee un abundante número de funciones, los cuales sirven para ser enlazados a objetos o botoneras, o usadas en scripts para realizar una gran cantidad de tareas, tales como: reconocer alarmas, ocultar ventanas, cambiar un tagname, para nombrar unas pocas.

11. ALARMAS

InTouch provee al sistema, los datos necesarios para informar a los operadores las condiciones del sistema y del proceso. El sistema soporta el despliegue, la anotación y la impresión de las alarmas del proceso y de los eventos del sistema. Las *alarmas* nos sirven para advertirnos sobre las condiciones del proceso.

InTouch incluye dos sistemas de alarmas:

- Sistema Estándar
- Sistema Distribuido

El *sistema de alarmas estándar* es usado para mostrar y reconocer eventos y alarmas, generado por la aplicación local de InTouch. El *sistema de alarmas distribuido* va mucho más allá, de tal manera que

nos permite mostrar y reconocer las alarmas generadas por el sistema de alarmas local de otras aplicaciones de InTouch.

El tipo de alarma que será utilizada por un tagname, es seleccionado en el Diccionario de Tagname. Las condiciones para el tipo de alarma seleccionada, también son configuradas en el Diccionario de Tagname.

Esta función admite a varios servidores o suministradores de alarmas simultáneamente, lo que da la capacidad de observar la información de varias alarmas a la vez, desde algunas ubicaciones remotas.

El sistema permite la visualización de alarmas o cualquier sistema de prevención, mediante un objeto cuyo tamaño sea definido por el usuario, el cual puede ser colocado solo o junto con otros objetos dentro de una ventana.

Existen tres tipos de alarmas para los tagnames enteros o reales. Estas son:

1. Valor de la alarma

- Alarma mínima (Low)
- Alarma máxima (Max)
- Alarma mínima-mínima (LoLo)
- Alarma máxima-máxima (HiHi)

1. *Desviación*

- Menor
- Estado normal
- Mayor

3. *Velocidad de cambio*

Los tipos de alarmas Mínimo-Mínimo, Mínimo, Máxima y Máxima-Máxima, son usadas para detectar los valores de un tagname análogo que se disparan más allá del límite absoluto.

Un tagname discreto puede tener su condición de alarma para su estado de encendido (ON) o apagado (OFF). Este tipo de alarma es muy utilizado en controladores (PLC), donde el bit de un registro es utilizado como una bandera de alarma (alarm flag).

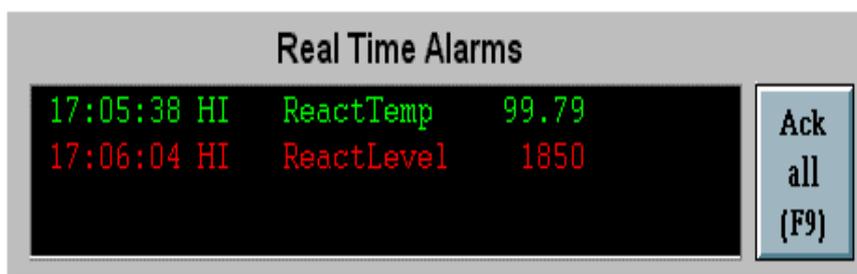


FIGURA 4.9 Ejemplo de Alarmas

En la figura 4.9, se muestra una aplicación de alarmas en tiempo real, en

la cual aparecerá la hora exacta en que apareció el problema en el sistema, clasifica el nivel del problema y también identifica al sensor del dispositivo que presentó la falla.

1. CAPACIDAD DE VISUALIZACION DE ALARMAS

Varios son los aspectos sobre la capacidad de visualización de alarmas. Los más importante son:

1. Es posible un desplazamiento hacia adelante o hacia atrás a través de las pantallas de alarmas, presionando los botones de comando. Las alarmas actuales están disponibles en forma de objeto llamado *resumen de alarmas*, y el resumen cronológico de las alarmas están disponible como un objeto llamado *historia de alarmas*.
2. El operador puede seleccionar las alarmas desplegadas por un objeto, alarmas seleccionadas por grupos y/o prioridad mediante el uso de botones de comando. Se permiten hasta 999 niveles de prioridad.
3. El sistema permite un número ilimitado de pantallas de alarma.
4. Las alarmas pueden codificarse por color, de acuerdo al estado que presenten, pudiéndose escoger entre 32 colores diferentes para la visualización de cada uno de los siguientes estados de alarma:

- Condición de alarma reconocida
- Condición de alarma no reconocida
- Alarma que ha retornado a su condición normal pero que no ha sido aún reconocida

1. La pantalla de alarmas permite la visualización de los siguientes *parámetros de alarma*, los cuales pueden ser elegidos por el usuario en el modo de configuración:

- Fecha
- Hora
- Tipo de alarma (alta-alta, baja-baja)
- Prioridad de la alarma
- Valor de la variable durante la alarma
- Nombre del operador
- Nombre del grupo de alarma
- Límite de alarma
- Comentarios

1. El sistema puede configurarse para que el operador sea notificado de una alarma, sin importar la pantalla que esté observando en esos momentos. La notificación podría ser la visualización de una ventana de alarmas que surge repentinamente, o un símbolo del

proceso que parpadea.

7. El usuario puede visualizar las alarmas en forma individual o por grupo, con capacidad para 16 grupos, pudiendo contener cada uno de ellos hasta 16 sub-grupos. Las alarmas pueden tener una jerarquía hasta de 8 niveles

8. También es posible informar al operador de una condición de alarma mediante un tono audible, una pantalla que aparece repentinamente, o cualquier tipo de animación sobre la pantalla. El reconocimiento de alarma puede ser efectuado sobre todas las alarmas, las alarmas dentro de un solo grupo, alarmas en una colección de grupos dentro de un grupo jerárquico definido, o punto por punto.

Cada tagname se encuentra asociado a un Grupo de Alarma

(Alarm Group). Si el Grupo de alarma no ha sido creada, el tagname es automáticamente asociado al grupo \$System. Los Grupos de Alarmas son organizados de acuerdo a su estructura jerárquica, con el Grupo \$System a la cabeza de dicha estructura, pudiendo tener hasta ocho niveles de jerarquía.

Cada Grupo de Alarma puede tener un máximo de 16 subgrupos, cada subgrupo puede tener un máximo de 16 subgrupos, y así sucesivamente, hasta alcanzar el máximo de 8 niveles.

4.3.11.2. CAPACIDAD DE ARCHIVO Y DE IMPRESIÓN DE LAS ALARMAS

De la capacidad de archivo de alarmas podemos decir que, las alarmas pueden ser anotadas en un archivo para una referencia futura, o para efectuar una revisión de los datos históricos de las alarmas. Para realizar una revisión de las alarmas fuera de la computadora, se podría imprimir dichas alarmas, dirigiéndolas hacia una impresora, usando una interface serial o paralela.

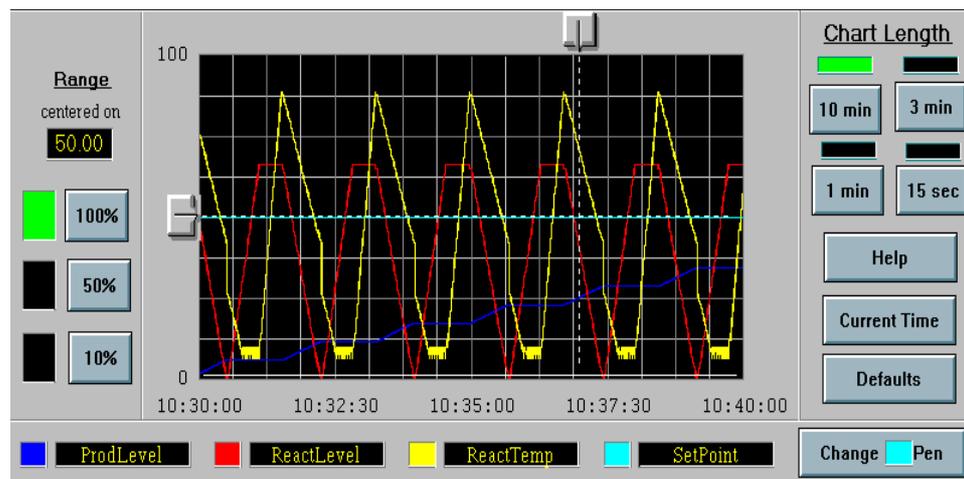
4.3.12. EVENTOS

Los *eventos* representan mensajes sobre las condiciones normales del sistema. Los eventos pueden ser anotados para una revisión posterior por el operador, personal de ingeniería, o directivo. El sistema anotará cualquier nueva sesión iniciada por el operador, cuantas veces un operador cambia un punto de consigna, cuando coloca o retira de servicio un dispositivo. Cada vez que un evento sea archivado, se registra además la sesión durante la cual ocurrió y el tipo de acción tomada (cambio de punto de consigna, cambio de estado con fecha y hora).

4.3.13. TENDENCIA EN TIEMPO REAL Y RECOLECCIÓN DE DATOS HISTÓRICOS DISTRIBUIDOS

La pantalla de *tendencias históricas* le permite al usuario efectuar acercamientos o alejamientos en tiempo, desde un segundo hasta 6

semanas en una pantalla. El operador tiene la posibilidad de efectuar desplazamientos hacia adelante o hacia atrás en el tiempo, para observar los datos históricos recolectados, los cuales tienen la posibilidad de ser impresos en forma de un registro impreso, para propósito de documentación. Los datos históricamente recolectados están disponibles para ser exportados a un formato de hoja electrónica para su análisis, y para la elaboración de reportes adicionales.



Este sistema permite especificar de forma dinámica, diferentes fuentes de datos de archivos históricos, para cada una de las plumas de un gráfico de tendencia. Como InTouch permite usar hasta ocho plumas por gráfico, los usuarios pueden disponer de una cantidad sin precedente de datos históricos en cualquier instante, esenciales para la elaboración de reportes gerenciales.

FIGURA 4.10. Aplicación de la pantalla de Tendencia de Datos Históricos Distribuidos.

En la figura 4.10, se observa la pantalla de tendencias de datos

históricos distribuidos donde aparecen en tiempo real, y se recogen los datos. En el mismo gráfico están leyéndose tres datos.

4.3.14. IMPRESION Y AGENDA DE REPORTES

Los reportes impresos pueden contener información del proceso, incluyendo datos del proceso, variables acumuladas, y estado de las variables. Se dispone de la capacidad de poder incluir "instantáneas" de tendencias, histogramas, y cartas SPC en los reportes impresos.

Los reportes pueden ser incluidos en una agenda por hora del día, día de la semana, o al final de una actividad. Los reportes pueden ser impresos bajo demanda del operador, en base a cualquier cambio de estado dentro del sistema.

4.3.15. REFERENCIA DINÁMICA

Esta función permite a los usuarios cambiar las referencias de bases de datos a etiquetas de entrada/salida durante la ejecución, lo que significa que en cualquier momento se puede cambiar las referencias de datos para direcciones de PLC, celdas de hojas de cálculo de Excel y referencias de intercambio dinámico de datos (DDE).

4.3.16. SEGURIDAD DEL SISTEMA

La protección es esencial para evitar daños al sistema y prevenir acciones no autorizadas. El sistema da la posibilidad de configurar el modo de

ejecución, para limitar de esta manera el acceso directo del operador sobre la consola de presentación y control, mediante la deshabilitación de ciertas teclas en el sistema. De esta manera, sólo la pantalla de presentación y control que se haya habilitado dentro de la aplicación final, estará disponible para el operador del proceso.

Es posible configurar el modo de ejecución del sistema para prevenir un acceso directo del operador al menú de archivos o cualquier otra posibilidad para abrir o cerrar archivos que se encuentren fuera de las facilidades otorgadas por la interface de aplicación al operador final.

Se puede proveer protección mediante palabras de paso (password) que suministra un enmascaramiento móvil, que puede cubrir el sistema completo de interface a nivel gráfico con el usuario, incluyendo las barras de título del sistema operativo y las barras de menú, de modo tal que únicamente personal autorizado pueda tener acceso a este nivel de control. Esta protección es necesaria para prevenir que personal que no esté calificado, pueda ocasionar daños a la aplicación de interface con el operador, borrar accidentalmente registros o archivos, o acceder a otros programas que no se encuentren ligados con la aplicación deseada de monitoreo y control supervisorio de la planta.

Es posible asignar a cada operador una palabra de paso (password), la cual será definida como el único nivel de acceso, lográndose así una entrada, limitada a varias funciones de comando en base al nivel de acceso del operador. Basados en la única palabra de paso del operador, es posible registrar cada una de las acciones para una futura revisión. También es factible definir un rango de tiempo de inactividad entre las acciones del operador sobre el sistema, solicitándole de nuevo su palabra

de paso para reingresar a la sesión. Esta capacidad es útil para prevenir que ocurra un acceso no autorizado al sistema de interface con el operador, mientras el operador autorizado se encuentre retirado de su estación, efectuando otras actividades.

4.3.17. FACTORY FOCUS

Es un potente nodo de sólo visualización que puede ser visto cuando está corriendo el Runtime. Permite a los gerentes y supervisores revisar continuamente las aplicaciones del proceso en tiempo real desde cualquier PC de escritorio en cualquier punto de una red. Posee un sistema de seguridad que permite solamente revisar datos, los cuales no pueden ser cambiados. Aspectos como enlaces de animación, etiquetas (tagnames), tendencia histórica son solamente observados.

4.4. MODULOS DE INTOUCH

InTouch consta de tres módulos, los cuales ayudan a los usuarios a satisfacer los pedidos de reportes industriales. Dichos módulos son:

- ***Lenguaje estructurado requerido (SQL)***

Suministra a los usuarios acceso a los programas de bases de datos más importantes entre los que se incluyen SQL Server de Microsoft, Oracle, Sybase, dBase, y otros que admitan la Conectividad de Base de Datos Abierta (ODBC).

- ***Control estadístico de procesos (SPC)***

Proporciona potentes funciones de análisis y supervisión en tiempo real, mejorando la calidad del proceso, y en última instancia, los productos. Admite histogramas, gráficos de pareto y una gran variedad de gráficos de control.

- ***Administrador de recetas***

Este módulo está diseñado para aplicaciones de proceso y manufactura, en las que se generan productos por lotes utilizando parámetros de configuración de productos.

4.5. DESARROLLO DE LA APLICACIÓN

Explicaremos brevemente las herramientas utilizadas para diseñar la pantalla de control, los enlaces de animación, la escritura lógica. También nos referiremos a los objetos, tagnames y expresiones lógicas utilizadas.

4.5.1. DISEÑO DE LA PANTALLA DE CONTROL

En la figura 4.11, se muestra la ventana en donde se muestra el programa desarrollador de procesos de InTouch, WindowMaker. En esta ventana es en donde se definen todos los tagnames, variables y expresiones lógicas, necesarias para crear la ventana que servirá para visualizar un proceso con control en tiempo real, o como en nuestro caso, para crear la simulación de tal proceso. Para diseñar la simulación del proceso, necesitamos crear algunas cajas, dependiendo del lugar donde ésta se encuentre (banda, paletas, montacarga, elevador o estante) en la ventana. De tal manera, que el movimiento total de una caja sea lo más natural

posible, desde que se mueve por la banda hasta que es colocada finalmente en el estante. También observamos tres medidores, los cuales sirven para controlar el movimiento en metros de las paletas, del montacarga y del elevador. Se ha colocado cuatro luces piloto, las cuales ayudan para saber que motor está funcionando en cualquier instante (banda, paletas, montacarga, elevador).

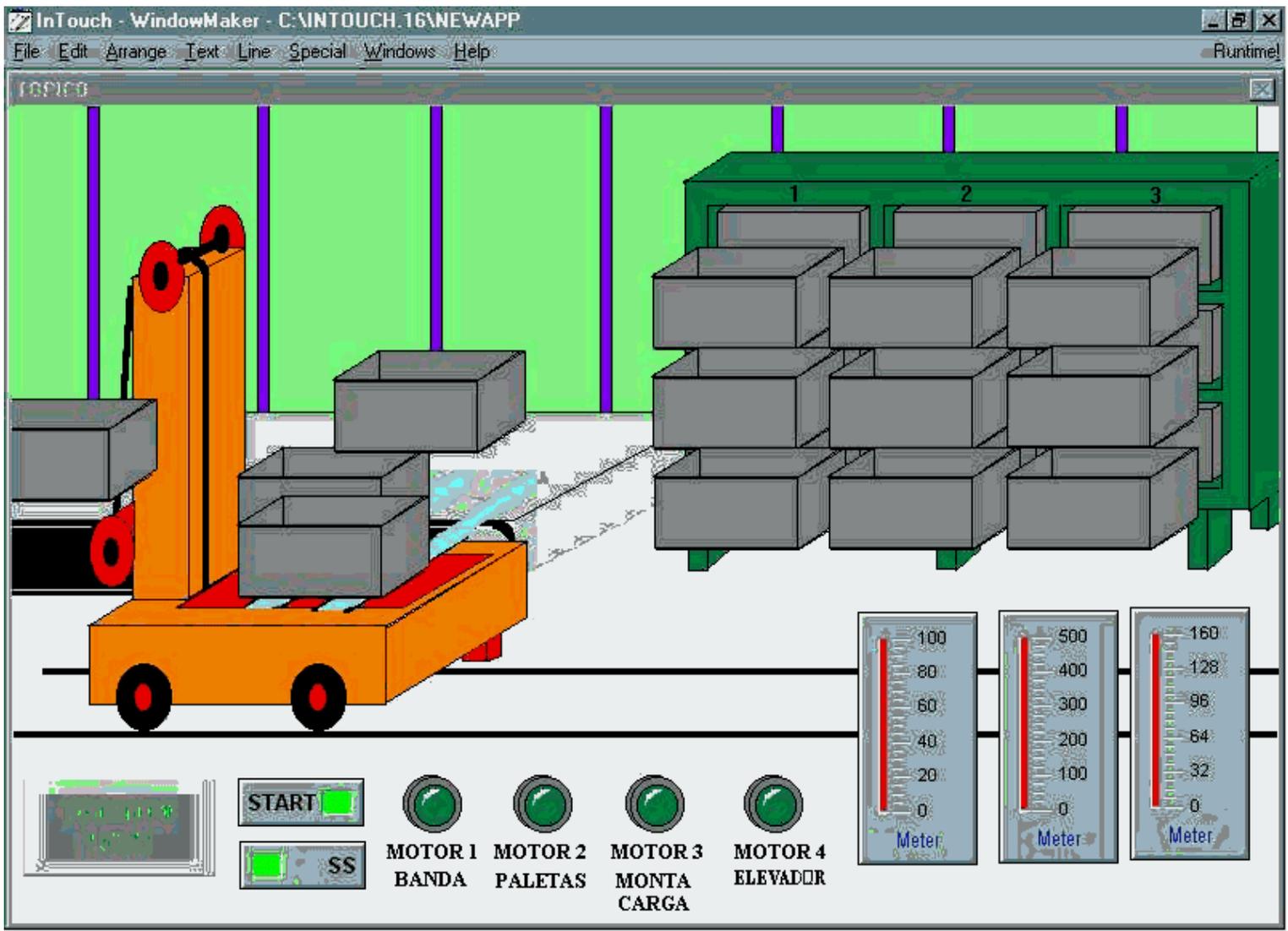
En la figura 4.12, se observa la ventana de ejecución del proceso, llamada WindowViewer. Podemos ver el movimiento del carro montacarga, de las paletas, del elevador y el ordenamiento de las cajas en el estante.

4.5.2. ESCRITURA LOGICA DE LA APLICACION

A continuación se detalla la escritura lógica para crear la simulación del proceso.

```
IF INICIO == 1 THEN  
  
    IF N == 9 AND STOP == 0 THEN  
  
        Show "ALARMA";  
  
    ENDIF;  
  
    IF N <= 8 THEN  
  
        IF STOP == 0 THEN  
  
            IF caja1 < 230 THEN
```

FIGURA 4.11. Diseño de la pantalla de visualización en el Desarrollador
(Window Maker)



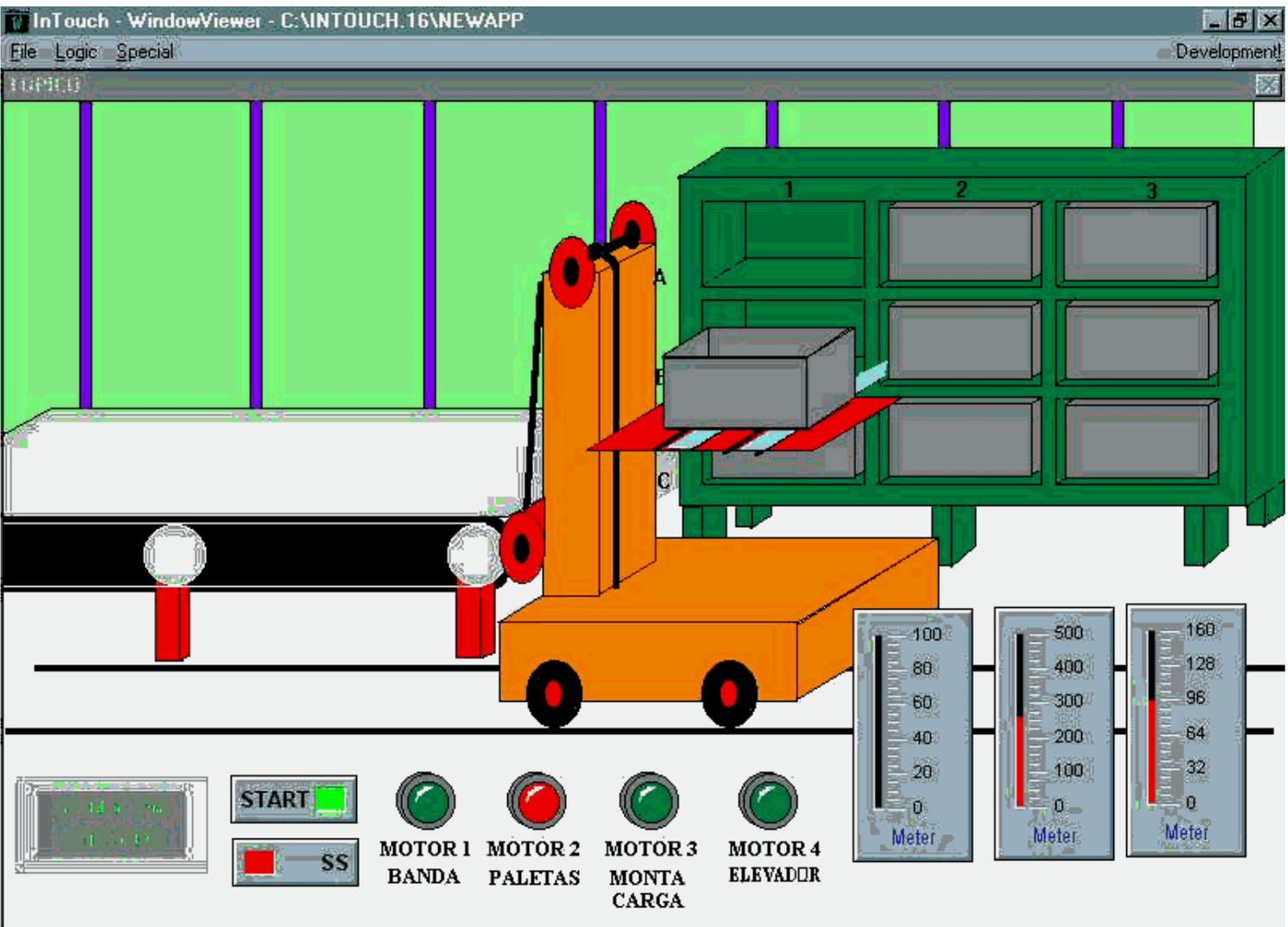


FIGURA 4.12. Diseño de la pantalla de visualización en modo Runtime (WindowViewer)

caja1 = caja1 + 10;

M1 = 1;

ELSE

IF paletas < 60 AND julio == 0 THEN

M1 = 0;

M2 = 1;

paletas = paletas + 6;

ELSE

IF CAJA2 < 30 THEN

M2 = 0;

M4 = 1;

CAJA2 = CAJA2 + 5;

CAJAPRUEBA = CAJAPRUEBA +5;

ELSE

IF caja3 < 60 THEN

M2 = 1;

M4 = 0;

caja3 = caja3 +6;

paletas = paletas - 6;

julio = 1;

ELSE

IF caja4 < 30 THEN

M2 = 0;

M4 = 1;

caja4 = caja4 + 5;

CAJAPRUEBA = CAJAPRUEBA - 5;

ELSE

IF CAJA5 < 480 AND N <= 2 THEN

CAJA5 = CAJA5 + 20;

montacarga = montacarga + 20;

J = 1;

M3 = 1;

M4 = 0;

ELSE

IF CAJA5 < 369 AND (N >2 AND N <= 5)
THEN

CAJA5 = CAJA5 + 20;

montacarga = montacarga + 20;

M3 = 1;

M4 = 0;

J = 1;

ELSE

IF CAJA5 < 258 AND (N > 5 AND N <= 8)
THEN

CAJA5 = CAJA5 + 20;

montacarga = montacarga + 20;

M3 = 1;

M4 = 0;

J = 1;

ELSE

IF CAJA6 < 30 AND (N == 0 OR N == 3 OR N
== 6) THEN

K = 1;

J = 0;

M3 = 0;

M4 = 1;

CAJA6 = CAJA6 + 5;

caja = caja + 5;

ELSE

IF CAJA6 < 92 AND (N == 1 OR N == 4 OR N
== 7) THEN

K = 1;

J = 0;

M3 = 0;

M4 = 1;

CAJA6 = CAJA6 + 5;

caja = caja + 5;

ELSE

IF CAJA6 < 154 AND (N == 2 OR N == 5 OR N

== 8) THEN

K = 1;

J = 0;

M3 = 0;

M4 = 1;

CAJA6 = CAJA6 + 5;

caja = caja + 5;

ELSE

IF CAJA7 < 40 AND (N == 0 OR N == 3 OR N
== 6) THEN

K = 0;

M4 = 0;

M2 = 1;

CAJA7 = CAJA7 + 5;

ELSE

IF CAJA8 < 40 AND (N == 1 OR N == 4 OR N
== 7) THEN

K = 0;

M2 = 1;

M4 = 0;

CAJA8 = CAJA8 + 5;

ELSE

IF CAJA9 < 40 AND (N == 2 OR N == 5 OR N

== 8) THEN

K = 0;

M2 = 1;

M4 = 0;

CAJA9 = CAJA9 + 5;

ELSE

IF (ROTOR == 1) AND ((CAJA7 > 39 AND (N == 0 OR N == 3 OR N == 6)) OR (CAJA8 > 39 AND (N == 1 OR N == 4 OR N == 7)) OR (CAJA9 > 39 AND (N == 2 OR N == 5 OR N == 8))) THEN

F = F + 1;

ROTOR = 0;

M2 = 0;

ELSE

IF caja > 0 THEN

caja = caja - 5;

M4 = 1;

ELSE

IF montacarga > 0 THEN

M4 = 0;

M3 = 1;

montacarga = montacarga - 20;

ELSE

IF montacarga <= 0 THEN

caja1 = 0;

CAJA2 = 0;

caja3 = 0;

caja4 = 0;

CAJA5 = 0;

CAJA6 = 0;

CAJA7 = 0;

CAJA8 = 0;

CAJA9 = 0;

CAJAPRUEBA = 0;

julio = 0;

paletas = 0;

montacarga = 0;

M3 = 0;

ROTOR = 1;

caja = 0;

N = N+1;

ENDIF;

ELSE

caja1 = 0;

CAJA2 = 0;

caja3 = 0;

caja4 = 0;

CAJA5 = 0;

CAJA6 = 0;

CAJA7 = 0;

CAJA8 = 0;

CAJA9 = 0;

F = 0;

K = 0;

N = 0;

paletas = 0;

julio = 0;

CAJAPRUEBA = 0 ;

montacarga = 0;

caja = 0;

ROTOR = 1;

M1 = 0;

M2 = 0;

M3 = 0;

M4 = 0;

ENDIF;

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Controlar la velocidad de rotación y el giro de un motor de paso es sencillo y de bajo costo, debido a que no necesita un control de lazo cerrado, ni variar el voltaje de alimentación. La velocidad de giro depende únicamente de la frecuencia de los pulsos de entrada. El cambio de giro se realiza por medio de la secuencia de alimentación de los devanados
2. Los controladores lógicos programables (PLC) brindan una mayor eficiencia y flexibilidad a la hora de querer modificar el sistema de control, que de otra manera sería un proceso lento y engorroso, debido a que tendríamos que cambiar el cableado y las pistas de cobre de los circuitos electrónicos de estado sólido.
3. Las interfaces humano-máquina (HMI), son sistemas de control que brindan muchas ventajas, como alta eficiencia de producción, control completo de las operaciones y tratamiento preventivo de las acciones a tomar según la variación de una variable de proceso.

ANEXOS

ANEXO A

TEORIA BÁSICA DE MOTORES DE PASO

VENTAJAS DE LOS MOTORES DE PASO

1. El ángulo de giro del motor es proporcional a los pulsos de entrada.
2. El motor tiene un buen torque cuando se encuentra inactivo, pero si los devanados están energizados.
3. Posicionamiento preciso y repetibilidad de movimiento.
4. Excelente respuesta de arranque, parada y reversa.
5. Responden a los pulsos digitales de entrada dado por un control de lazo abierto, por lo que su control es muy simple y de bajo costo.
6. Es posible lograr una muy baja velocidad sincrónica de rotación con carga, la cual es directamente acoplada al eje.
7. Ancho rango de velocidades de rotación, dicha velocidad es proporcional a la frecuencia en los pulsos de entrada.

DESVENTAJAS DE LOS MOTORES DE PASO

1. La resonancia es un gran problema si no es controlado propiamente.
2. No es fácil de operar a muy altas velocidades.

OPERACIÓN EN LAZO ABIERTO

Uno de las más importantes ventajas de los motores de paso es su habilidad de poder ser controlado de manera precisa por medio de un sistema de control de lazo abierto. Un control de lazo abierto significa que no se necesita información de retroalimentación acerca de su posición. Este tipo de control elimina la necesidad de comprar dispositivos de control de alto costo, que funcionan como sensores, tales como los codificadores ópticos. La posición es conocida simplemente por medio de la conservación de los pulsos de entrada.

TIPOS DE MOTOR DE PASO

Hay tres tipos básicos de motores de paso. Estos son:

- De Reluctancia Variable (RV)
- De Imán Permanente (MP)
- Híbrido

Motor de paso de reluctancia variable (RV)

Este tipo de motor de paso, es probablemente el más fácil de comprender desde un punto de vista estructural. La figura 1 muestra la sección transversal de un motor de paso típico de RV. Este tipo de motor consiste de un rotor de hierro dulce multidentado y un estator devanado. Cuando los devanados del estator son energizados con corriente directa (CD), los polos empiezan a magnetizarse. La rotación ocurre cuando los dientes del rotor son atraídos a los polos energizados del estator. Ver figura A1.

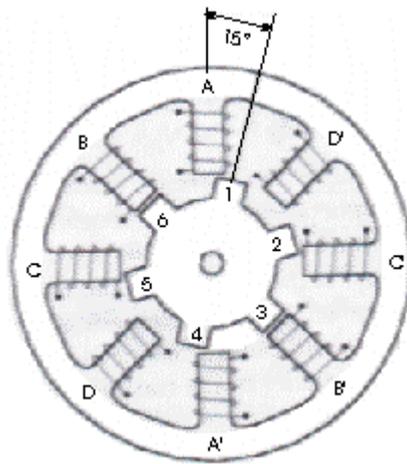


FIGURA A1. Sección transversal de un motor de reluctancia variable (RV)

Motor de paso de imán permanente (MP)

El motor de paso de imán permanente (MP) también es conocido como un motor sincrónico de inducción. Este tipo de motor que fue originalmente diseñado para operar como un motor sincrónico de velocidad baja, tiene la habilidad de dar un paso cuando una fase sea excitada apropiadamente. El estator del motor usualmente contiene un devanado de fase múltiple, y el rotor tiene un imán permanente o un campo unidireccional que es producido por una fuente separada de corriente directa.

Motor de paso Híbrido (HB)

El motor de paso híbrido (HB) es más costoso que el motor de paso de MP, pero posee un mejor funcionamiento con respecto a la resolución de los pasos, al torque y velocidad. Los ángulos de paso típico para un motor HB van desde los 3.6° hasta los 0.9° (100 a 400 pasos por revolución). El motor de paso HB combina las mejores características de los motores de paso de MP y de RV. El rotor es multidentado,

parecido al del motor de RV y contiene un imán concéntrico axialmente magnetizado alrededor del eje. Los dientes del rotor proporcionan un mejor camino para el flujo magnético. Esto provoca un incremento del torque de detención, de mantenimiento y dinámico, comparado con los motores de paso de RV y de MP. Ver figura A2.

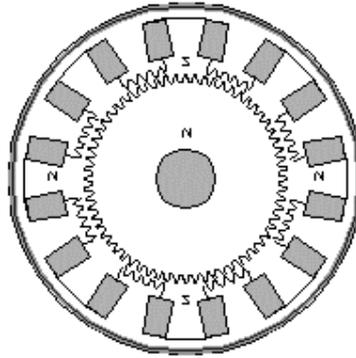


FIGURA A2. Sección transversal de un motor de paso híbrido

Los dos tipos de motores más usados son los de imán permanente e híbridos.

TAMAÑO Y POTENCIA

A parte de poder ser clasificados de acuerdo a su ángulo de paso, los motores de paso también son clasificados de acuerdo a su tamaño correspondiente al diámetro del cuerpo del motor. Como una regla general, el torque de salida de un motor de tamaño determinado, se incrementará con el aumento en la longitud de su estructura.

Los niveles de potencia de los circuitos integrados manejadores de motores de paso vienen dados en un rango que va desde los pocos vatios para manejar motores pequeños, hasta los 10 – 20 vatios, para los más grandes. La máxima potencia de disipación de un motor son establecidas por los datos del fabricante.

¿CUANDO UTILIZAR UN MOTOR DE PASO?

Un motor de paso puede ser una buena elección, si se requiere llevar a cabo el control de movimiento. Ellos son utilizados en aplicaciones donde se necesite el control del ángulo de rotación, velocidad, posición y sincronismo. Estas ventajas lo hacen muy requerido en aplicaciones como: impresoras, plotters, equipos de oficina, manejadores de disco duro, equipos médicos, máquinas fax, automotores y mucho más.

CAMPO MAGNÉTICO ROTACIONAL

Cuando el devanado de una fase de un motor de paso es energizado, un flujo magnético es desarrollado en el estator. La dirección de este flujo es determinado por la Regla de la Mano Derecha.

En la figura A3, se muestra la trayectoria del flujo magnético desarrollada cuando la fase B es energizada con la dirección de corriente mostrada. El rotor se alineará donde la oposición del flujo sea mínima. En este caso el motor girará en el sentido de las manecillas del reloj, el polo sur se alineará con el polo norte del estator B en la posición 2, y el polo norte se alineará con polo sur del estator B en la posición 6. Para lograr la rotación continua del motor, debemos proveer una correcta secuencia de energización a los devanados del estator

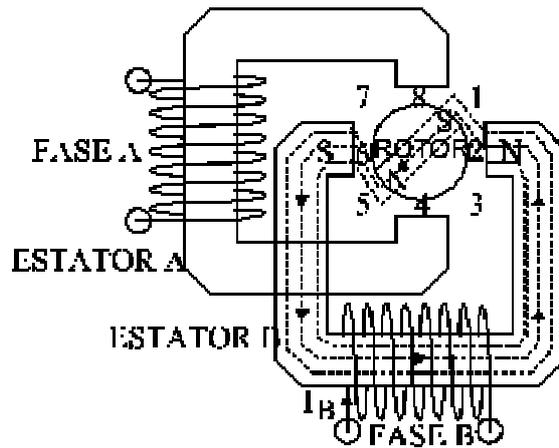


FIGURA A3. Trayectoria del flujo magnético a través de dos polos del motor de paso

GENERACIÓN DEL TORQUE

El torque producido por un motor de paso depende de varios factores, estos son:

- Velocidad de los pasos
- Corriente en los devanados
- Tipo de manejador (driver) utilizado

En un motor de paso, el torque es desarrollado cuando el flujo magnético del rotor y del estator es desplazado mutuamente desde el uno al otro. El estator está hecho de un material magnético de alta permeabilidad. La presencia de esta alta permeabilidad, causa que el flujo magnético esté confinado (encerrado) en la mayor parte de las trayectorias definidas por la estructura del estator, de la misma forma en que la corriente está confinada a los conductores de un circuito electrónico.

El torque de salida producido por un motor es proporcional a la intensidad del flujo magnético generado cuando los devanados son energizados. La relación básica, la cual

define la intensidad del flujo magnético, es la siguiente:

$$H = (N \times i) / l \quad (1)$$

N = Número de vueltas del devanado

I = intensidad de corriente

H = Intensidad del flujo magnético

L = longitud de la trayectoria para las líneas de flujo magnético

La ecuación 1, muestra la intensidad del flujo magnético, consecuentemente, el torque es proporcional al número de vueltas en los devanados y a la corriente, e inversamente proporcional a la longitud de la trayectoria para el flujo magnético. De esta relación básica, podemos concluir que para diferentes motores de paso de igual tamaño, éstos podrían tener diferentes torques de salida, simplemente cambiando los parámetros de los devanados.

FASES, POLOS Y ANGULOS DE FASE

Generalmente los motores de paso tienen dos fases, pero también existen motores de tres y cinco fases.

Un motor bipolar con dos fases, tiene una fase por devanado, mientras que un motor unipolar tiene un devanado, con un tap central por fase. Algunas veces, el motor de pasos unipolar es llamado como "motor de cuatro fases", aunque sólo posee dos fases.

Los motores que tienen dos devanados separados por fase, pueden ser manejados en modo bipolar o unipolar.

Un polo puede ser definido como una de las regiones de un cuerpo magnetizado, donde el flujo magnético está concentrado. Tanto el rotor como el estator de un motor de paso tienen polos. Los polos que son añadidos a la estructura de rotor y del estator sirven para incrementar el número de pasos por revolución del motor, o dicho

de otra manera, para poder definir el ángulo de paso base lo más pequeño posible.

El motor de paso de imán permanente, contiene un número igual de par de polos en el estator y rotor. Típicamente, el motor de imán permanente tiene 12 par de polos. El motor de paso híbrido posee un rotor dentado, el cual es dividido en dos partes, separadas por un imán permanente, haciendo que una mitad sea polo sur y la otra mitad de polo norte. El número de par de polos es igual al número de dientes de una de las mitades del rotor. Usualmente, cuatro polos principales son usados para híbridos de 3.6 grados y ocho polos para los de 1.8 y 0.9 grados.

Es la relación entre el número de polos del rotor (número de polos por fase), los polos del estator equivalente y el número de fases, lo que determina el ángulo de paso de un motor de paso.

$$\text{Angulo de paso} = 360 / (N_{Ph} \times Ph) = 360 / N \quad (2)$$

N_{Ph} = Número de polos por fase = Número de polos del rotor

Ph = Número de fases

N = Número total de polos para todas las fases juntas

MODOS DE CONDUCCION PARA LOS MOTORES DE PASO

Los siguientes son los modos de control de conducción más comunes:

1. Control de encendido de 1 fase. (Wave Drive)
2. Control de encendido de 2 fases. (Full Step Drive)
3. Control de encendido de 1 y 2 fase. (Half Step Drive)

CONTROL DE ENCENDIDO DE 1 FASE

Con este modo de conducción, únicamente un devanado es energizado a la vez. El estator es energizado de acuerdo a la secuencia $A1 \rightarrow B1 \rightarrow A2 \rightarrow B2$, y el paso del rotor será desde la posición $8 \rightarrow 2 \rightarrow 4 \rightarrow 6$. La desventaja de este modo es que, en devanados unipolares solo usan el 25% de los devanado, y en devanados bipolares utilizan el 50% del devanado total del motor cuando son energizados. Por lo tanto, no se obtiene el máximo torque de motor. Ver figura A4.

CONTROL DE ENCENDIDO DE 2 FASES

En este modo de conducción, son energizadas dos fases a la vez. El estator es energizado de acuerdo a la secuencia $A1B1 \rightarrow A2B1 \rightarrow A2B2 \rightarrow A1B2$, obteniendo los pasos del rotor desde la posición $1 \rightarrow 3 \rightarrow 5 \rightarrow 7$. Este modo es parecido al modo de

conducción de una fase a la vez, pero difiere de la posición mecánica. El torque de

Los motores unipolares es menor que los motores bipolares (para motores con el mismo parámetro de devanados). Los motores unipolares utilizan únicamente el 50% de sus devanados, mientras que los motores bipolares aprovechan todo el devanado.

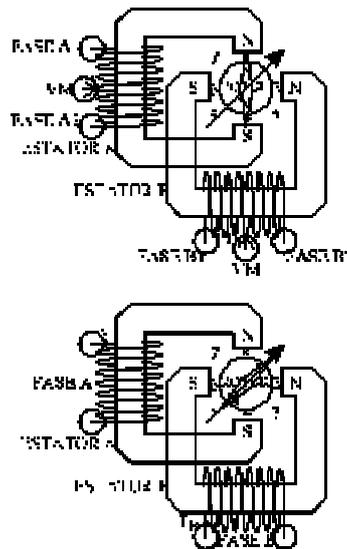


FIGURA A4. Motores de paso unipolar y bipolar

CONTROL DE ENCENDIDO DE 1 Y 2 FASES.

Este modo combina los modos de conducción de 1 y 2 fases anteriormente citados. En todos los segundos pasos únicamente una fase es energizada. El estator es energizado siguiendo la secuencia A1B1 → B1 → A2B1 → A2 → A2B2 → B2 → A1B2 → A1, y los pasos del rotor van desde la posición 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8. Esto resulta en movimientos angulares que son la mitad de aquellos en 1 y

2 fases de encendido.

En la siguiente tabla se observa la secuencia de excitación para los diferentes modos de

control.

FASES	1 FASE	2 FASES	1 - 2 FASES
	1 2 3 4	1 2 3 4	1 2 3 4 5 6 7 8
A1	*	* *	* * *
B1	*	* *	* * *
A2	*	* *	* * *
B2	*	* *	* * *

Tabla A1. Secuencia de excitación para los diferentes modos de conducción

CARACTERISTICAS DE TORQUE VS. ANGULO DE GIRO

Las características de torque vs. ángulo de giro son las relaciones entre el desplazamiento del rotor y el torque aplicado al eje del rotor cuando el motor de paso es energizado. Un motor de paso idealizado tiene la gráfica de torque vs. desplazamiento, en forma sinusoidal, como vemos en la figura A5.

Las posiciones A y C representan los puntos de equilibrio, cuando no existe carga aplicada al eje del motor. Cuando aplicamos una fuerza externa T_a al eje del motor se produce un desplazamiento angular, Θ a. El motor desarrolla un torque, T_a , opuesto a la fuerza externa, a fin de, balancear la carga. Como la carga se está incrementando, también existirá un incremento en el ángulo de desplazamiento, hasta alcanzar el

máximo torque de mantenimiento, T_h , del motor. Una vez que T_h es excedido, el motor ingresará a una región de inestabilidad. En esta región, un torque de dirección opuesta es creado y el rotor salta de un punto inestable a un punto estable.

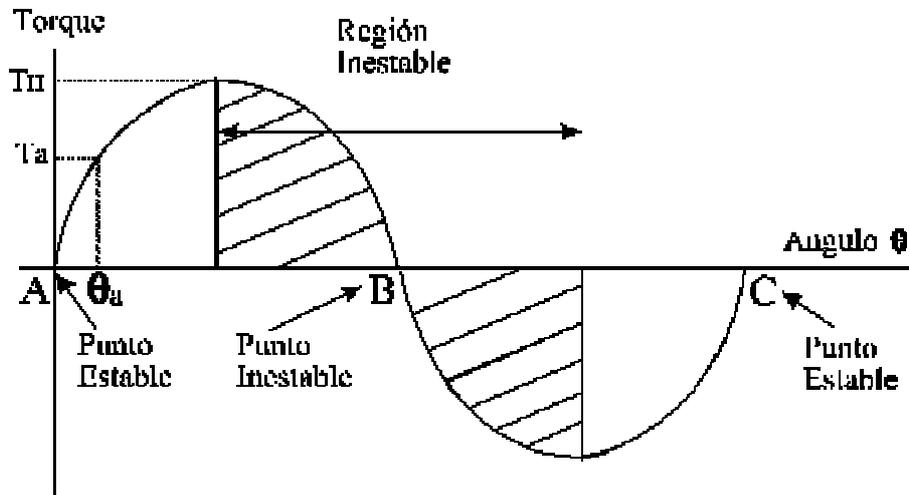


FIGURA 5. Torque vs. Posición angular del rotor

El ángulo de desplazamiento es determinado por la siguiente relación:

$$X = (Z / 2\pi) \times \text{sen} (T_a / T_h) \quad (3)$$

donde:

Z = diente del rotor

T_a = Torque de carga

T_h = Torque de mantenimiento del motor

X = Desplazamiento angular

EXACTITUD EN EL ANGULO DE PASO

Los motivos por el cual, los motores de paso han logrado popularidad como dispositivos de posicionamiento, se debe a sus características propias para conseguir exactitud y repetibilidad. La exactitud de un motor de paso está en función de la precisión de sus partes mecánicas.

Error de Posición en el Paso

Es el máximo error de posición, positivo o negativo, causado cuando el motor ha girado un paso desde su posición previa. De manera abreviada podemos escribir:

$$\text{Error de Posición en el Paso} = \text{Angulo de Paso Medido} - \text{Angulo Teórico} \quad (4)$$

CARACTERISTICAS DE TORQUE vs. VELOCIDAD

Las características de torque vs. velocidad son la llave para elegir el motor correcto y su método de manejo (driver) para una específica aplicación. Estas características son dependientes del motor, del modo de excitación y del tipo de manejo (driver).

Una típica curva de torque vs. velocidad, es representada en la figura A6.

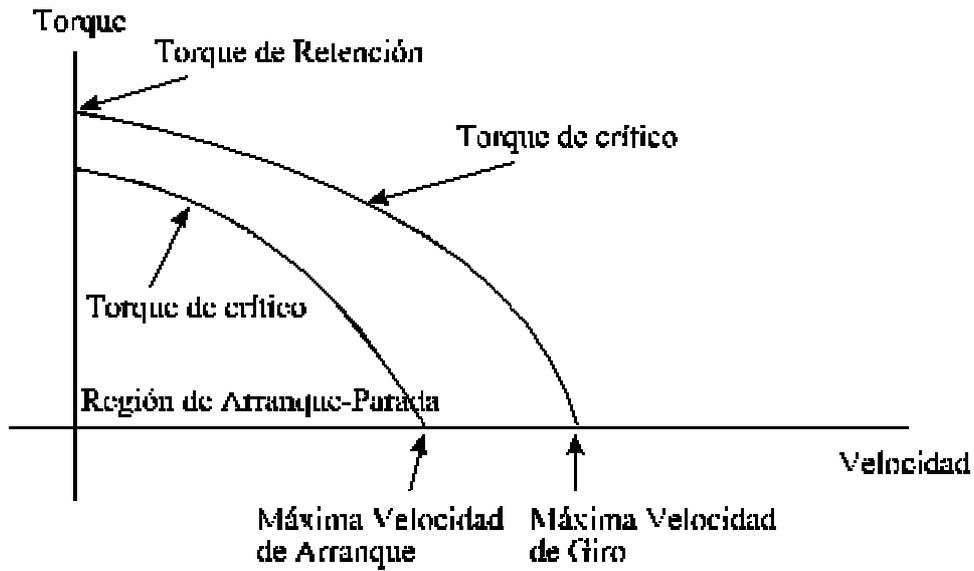


FIGURA A6. Características de Torque vs. Velocidad de un motor de paso

Para lograr un mejor entendimiento de esta curva, es necesario definir algunos conceptos relacionados con ella.

Torque de Mantenimiento

Es el máximo torque producido por el motor cuando está detenido

Curva de Máximo Par

Esta curva define a la región de arranque y parada. Esta es la máxima frecuencia en la cual el motor puede arrancar y parar de forma instantánea, con una carga aplicada, sin pérdida de sincronismo.

Máxima Velocidad de Arranque

Es la máxima frecuencia de pasos en el arranque, aplicado sin carga.

Curva de Máximo Par

Esta curva define un área conocida como la región de exploración. Define la frecuencia máxima a la cual el motor puede operar sin pérdida de sincronismo.

Máxima Velocidad de Giro

Es la máxima frecuencia de operación del motor sin carga. Las características del máximo torque también dependen de la carga. Mientras más grande sea la inercia de la carga, más pequeña es el área del máximo torque. La disminución del torque de salida así como el incremento de la velocidad, es causado por el hecho de que a altas velocidades, la inductancia del motor es el elemento dominante. La forma de la curva de velocidad - torque, puede cambiar dramáticamente dependiendo del tipo de driver utilizado.

RESPUESTA DE PASO SIMPLE Y RESONANCIA

Las características de la respuesta de paso simple de un motor de paso, es mostrado en la figura A7. Cuando un pulso es aplicado al motor de paso, el rotor se comporta de una forma, dada por la curva. El tiempo t , es el tiempo que le toma al eje del motor girar un paso, una vez que el primer pulso es aplicado.

Este tiempo es dependiente de la relación de torque e inercia de la carga, así, como al tipo de driver utilizado. Ahora bien, cuando existe un gran incremento de pasos, un alto torque es desarrollado, y consecuentemente una gran aceleración. Esto puede causar oscilaciones, como se muestra. El tiempo T , es el tiempo que toma a las oscilaciones en desaparecer. Para reducir o eliminar este comportamiento, es necesario utilizar un control por medio de micropasos.

El fenómeno de resonancia del motor de paso, puede ser visto como una caída en el torque, a ciertas velocidades, lo cual puede resultar en pérdida de pasos, o de sincronismo. Esto también ocurre cuando la velocidad de los pulsos de entrada coinciden con la frecuencia de oscilación natural del rotor. Algunas veces, existe un área de resonancia alrededor de los 100 - 200 pulsos por segundo (pps), y una también en la región de alta velocidad de pulsos. Este fenómeno no es posible de eliminar completamente.

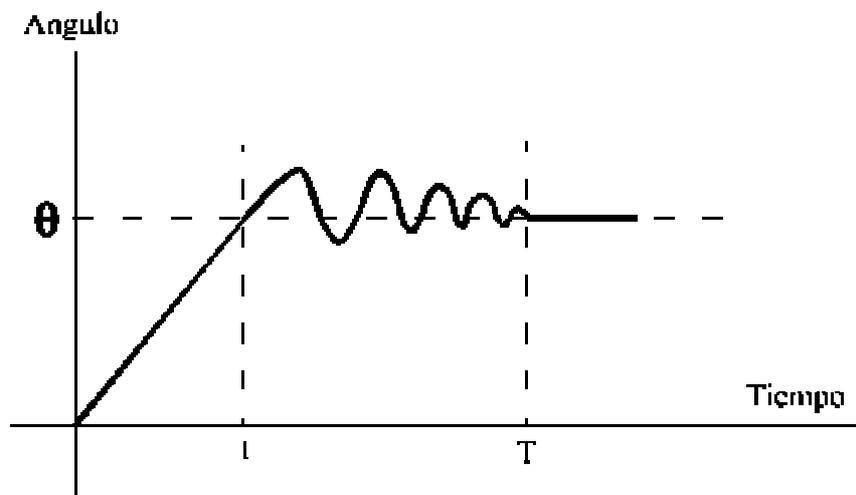


FIGURA A7. Respuesta de paso simple vs. tiempo

ANEXO B

TEORÍA BÁSICA DE INVERSORES TRIFÁSICOS

Los inversores trifásicos se utilizan normalmente en aplicaciones de alta potencia. Se puede obtener una salida trifásica a partir de una configuración de seis transistores y seis diodos, tal como se muestra en la figura B1. A los transistores se les puede aplicar dos tipos de señales de control: conducción a 180° o conducción a 120° .

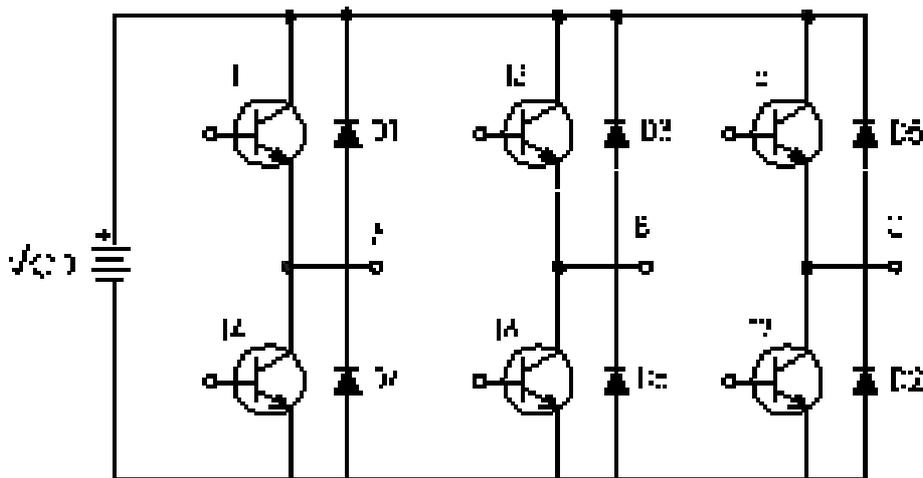


Figura B1. Puente de Inversor Trifásico

Conducción a 120°

Ya lo explicamos en el capítulo 2, numeral 2.3.3.

Conducción a 180 °

En este tipo de conducción, cada transistor conducirá 180° . Tres transistores se mantienen activos durante cada instante de tiempo. Cuando el transistor T1 está activado, la terminal A se conecta con la terminal positiva del voltaje de entrada, como se observa en la figura B1. Cuando se activa el transistor T4, la terminal A se lleva a la terminal negativa de la fuente de corriente directa. En cada ciclo existen seis modos de operación , cuya duración es de 60° . Las señales de excitación mostradas en la figura B2, están desplazadas 60° unas de otras, para obtener voltajes trifásicos balanceados (fundamentales).

Como sabemos, la carga puede conectarse en estrella o en delta. Existen tres modos de operación en un medio ciclo, los circuitos equivalentes aparecen en la figura B3, para el caso de una carga conectada en estrella.

Durante el modo 1, para $0 \leq \omega t \leq \pi /3$,

$$R_{eq.} = R + R/2 = 3R/2 \quad (B-1)$$

$$i_1 = V_s/R_{eq.} = 2V_s/3R \quad (B-2)$$

$$v_{an} = v_{cn} = i_1 R/2 = V_s/3 \quad (B-3)$$

$$v_{bn} = -i_1 R = -2V_s/3 \quad (B-4)$$

Durante el modo 2, para $\pi /3 \leq \omega t \leq 2\pi /3$

$$R_{eq} = R + R/2 = 3R/2$$

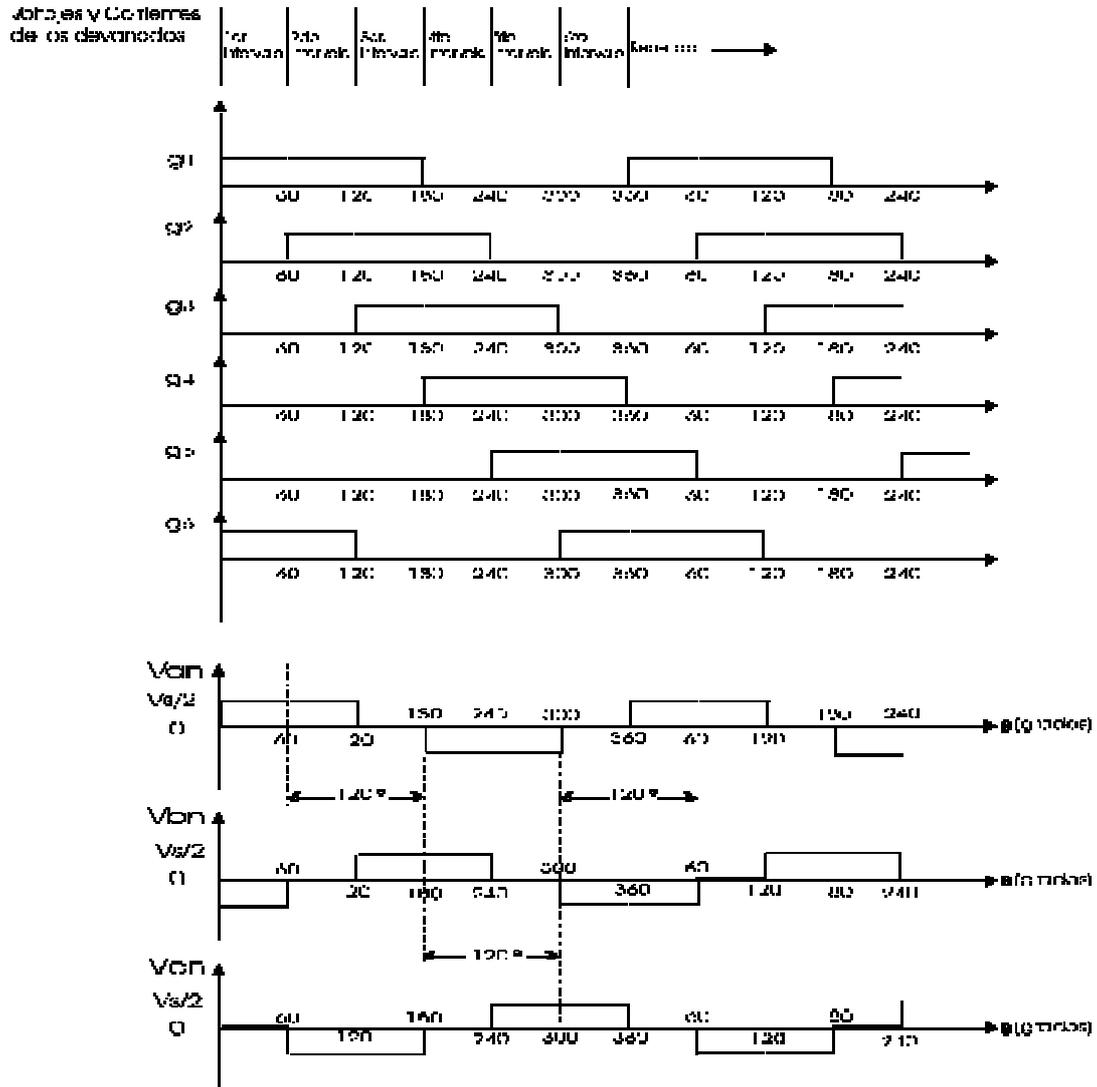


FIGURA B2. Formas de onda para conducción a 180°

$$i_2 = V_s/R_{eq} = 2V_s/3R$$

$$v_{an} = i_2 R = 2V_s/3 \quad (B-5)$$

$$v_{bn} = v_{cn} = -i_2 R/2 = -V_s/3 \quad (B-6)$$

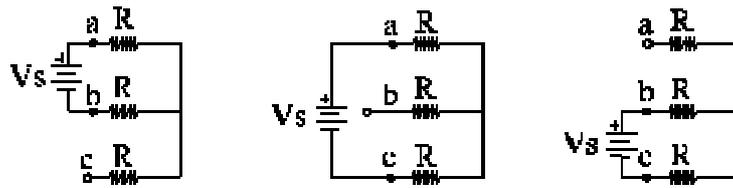


FIGURA B3. Circuitos equivalentes para una carga resistiva conectada en estrella

Durante el modo 3, para $2\pi/3 \leq \omega t \leq \pi$,

$$R_{eq.} = R + R/2 = 3R/2$$

$$i_3 = V_s/R_{eq.} = 2V_s/3R$$

$$v_{an} = v_{bn} = i_3 R/2 = V_s/3 \quad (B-7)$$

$$v_{cn} = -i_3 R = -2V_s/3 \quad (B-8)$$

El voltaje instantáneo de línea a línea v_{ab} , de la figura B2, se puede expresar en una serie de Fourier, reconociendo que v_{ab} está desplazada en $\pi/6$ y las armónicas pares son cero,

$$v_{ab} = \sum_{n=1,3,5..}^{\infty} 4V_s/n\pi \cos n\pi/6 \sin n(\omega t + \pi/6) \quad (B-9)$$

v_{bc} y v_{ca} pueden determinarse a partir de la ecuación (B-9), mediante el desplazamiento de fase de v_{ab} en 120° y 240° , respectivamente,

$$v_{bc} = \sum_{n=1,3,5..}^{\infty} 4V_s/n\pi \cos n\pi/6 \sin n(\omega t + \pi/2) \quad (B-10)$$

$$v_{ca} = \sum_{n=1,3,5..}^{\infty} 4V_s/n\pi \cos n\pi/6 \sin n(\omega t + 7\pi/6) \quad (B-11)$$

Podemos observar de las ecuaciones (B-9), (B-10) y (B-11), que en los voltajes de línea a línea, las armónicas múltiplos de tres ($n = 3, 9, 15, \dots$) son cero.

El voltaje rms de línea a línea se puede determinar a partir de

$$V_L = \left[\frac{2}{2\pi} \int_0^{2\pi/3} V_s^2 d(\omega t) \right]^{1/2} = \sqrt{2/3} V_s = 0.8165 V_s \quad (\text{B-12})$$

De la ecuación (B-9), la n ésima componente rms del voltaje de línea es

$$V_{Ln} = 4V_s / \sqrt{2 n \pi} \cos n\pi / 6 \quad (\text{B-13})$$

Que para $n = 1$, nos da el voltaje fundamental.

$$V_{L1} = 4V_s \cos 30^\circ / \sqrt{2\pi} = 0.7797 V_s \quad (\text{B-14})$$

El valor rms de los voltajes de línea a neutro se pueden determinar a partir del voltaje de línea,

$$V_p = V_L / \sqrt{3} = \sqrt{2} V_s / 3 = 0.4714 V_s \quad (\text{B-15})$$

ANEXO C

DATOS DE ELEMENTOS

CARACTERISTICAS DEL LM555

- Temporizaciones que van desde los microsegundos hasta las horas.
- Opera en ambos modos (Estable y Biestable).
- Ciclo ajustable.
- La salida puede ser fuente de 200mA.
- Salida y fuente compatible con lógica TTL.
- Salidas normalmente encendido y normalmente apagado.
- Empaquetado disponible en 8 pines.

Aplicaciones:

- Tiempos de precisión.
- Generación de pulsos.
- Tiempos secuenciales.
- Generación de retardos de tiempo.
- Modulación con pulsos.
- Generador lineal de rampa.

TABLA DE DATOS TECNICOS

DEL S7-200 CPU 212

Función	CPU 212
Tamaño físico	160 x 80 x 62 mm
Memoria	
Programa (EEPROM)	512 palabras
Datos del usuario	512 palabras
Marcas internas	128
Tiempo de respaldo (Condensador de alto rendimiento)	50 horas (típica) mínimo 8 horas con 40 ° C
Tiempo de carga para condensador de alto rendimiento	3 minutos
Programación	
Lenguaje de programación	STEP 7 Micro/Win
Organización del programa	Un bloque de organización
Ejecución del programa	Cíclica (OB1) Controlada por programa Controlada por tiempo

Niveles de subprogramas	8
JUEGO DE OPRACIONES	
Operaciones básicas	Funcionas lógicas, asignación de resultado, memorizar, contar, cargar, transferir, comparar,desplazar, rotar, formar complementos, llamar subprogramas.
Funciones sofisticadas	Modulación de ancho de impulsos, funciones de operación de trenes de impulso, de salto y de bucle, así como de conversión de códigos y aritméticas (sumador de 32 bits, multiplicador de 16 bits, divisor de 16 bits.
Entrada / salida (E / S)	
E /S integradas	8 DI / 6 DQ
Módulos de ampliación (máx)	2 módulos
Imagen del proceso E /S	30 incluyen (DI/DQ)
E / S Analógica (ampliación)	8
Operaciones	
Velocidad de ejecución booleana	1.2 μ s / operación
Contadores	64
Temporizadores	64
De ellos remanentes	31, ajustables

Margen	2 temporizaciones, 1ms a 30 s 8 temporizaciones, 10 ms a 5 minutos 54 temporizaciones, 100 ms a 54 minutos
Aritmética	Sí
Funciones adicionales	
Contadores	1 Contador hacía adelante o hacía atrás; frecuencia de contaje hasta 2 Khz
Potenciómetros Analógicos	1 resolución de 8 bits
Interruptores de comunicación	1 emisor / 1 receptor
Interrupciones temporizadas	1
Entradas de interrupción de Hardware	1
Comunicación	
Número de interfaces	1 (RS-485)
Protocolos asistidos Interface 0:	Modo PPI, Freeport
Punto a punto	Sólo esclavo
Protección	
Grado de protección	IP 20 según IEC 529
TEMPERATURA AMBIENTE	
Para montaje horizontal	0 a 55 ° C

Para montaje vertical	0 a 45 ° C
Humedad relativa del aire:	5 a 95 % (grado de sollicitación RH 2 según IEC 1131-2)
Presión atmosférica	860 a 1080 hPA
Sollicitación mecánica	
Vibraciones, probadas con	10 a 57 hz (amplitud constante 0,35 mm)
Choques, probadas con	57 a 150 hz (aceleración constante 2g)

TABLA DE DATOS TECNICOS
DEL MODULO DE E/S ANALOGICAS EM235

ENTRADAS	
Al medir la resistencia	-
Tensión de entrada	30 V
Márgenes / resistencia entrada	0...10V / 10 MΩ
Tensión máx. adm. de entrada	30 V
Para entrada de tensión máx.	
Intensidad máx. adm. de entrada para entrada de intensidad máx.	32 mA
Tiempo de reacción(todos los canales)	10 μ s
Tiempo de conversión analógica a digital	25 μ s
Resolución	12 bits
Supresión de interferencia para frecuencia parásita	40 dB 50 / 60 Hz
Tensión de modo común máx.	12 V
Margen de conversión	
Señales unipolares	0 a 32000
Señales bipolares	-32000 a +32000
Linealización de características	

Para termocuplas	Y, K, T
Para termoresistencias	Pt100, Pt1000
Compensación de temperatura	Parametizable
Interna	Posible
Externa con caja de compensación	Posible
SALIDAS	
Márgenes de salida	
Salidas de tensión	-10 a +10 V
Salidas de intensidad	0 a 20 mA
Resistencias de carga	
Con salidas de tensión mín.	5 k Ω
Con salidas de intensidad máx.	0.5 k Ω
Salida de tensión	
Protección contra cortocircuitos	Sí
Intensidad de cortocircuito máx.	11 mA
Salida de intensidad en vacío	15 V
Resolución	
Salida de tensión	12 bits (5mV / contador)
Salida de intensidad	11 bits (10 μ A / contador)
Tiempo de respuesta	
Salida de tensión	

Salida de intensidad	100 μ s 2 ms
Margen de valores de conversión	
Señales unipolares	0 a 32000
Señales bipolares	-32000 a +32000
Límite de error de práctico (0 a 60 ° C, referido al campo de salida)	
Tensión	$\pm 2,0$ %
Intensidad	$\pm 2,0$ %

CARACTERISTICAS DE INTOUCH

Los arreglos de texto se pueden ser de hasta 131 caracteres de longitud.

Minor Deviation Alarm	On-Off
Minor Deviation %	Numérica
Minor Deviation Alarm Priority	Numérica
Major Deviation Alarm Enable	On-Off
Major Deviation %	Numérica
Major Deviation Alarm Priority	Numérica
Deviation Alarm Target	Numérica
Deviation Alarms Deadband	Numérica
Rate of Change Alarm Enable	On-Off
Rate of Change Alarm Percente	Numérica
Rate of Change Alarm Time Base	Sec / Min / Hr
Rate of Change Alarm Time Value	Numérica
Rate of Change Alarm Priority	Numérica
Dispositivos de Entrada / Salida	(variables DDE solamente)

VARIABLES DIGITALES O DISCRETAS

Tagname	Cadena de caracteres
Alarm Group	Cadena de caracteres
Log Events	On / Off
Log Events Priority	Numérica
Retentive Value	On - Off
On Alarm / Event Message	Cadena de Caracteres
Off Alarm / Event Message	Cadena de Caracteres
Initial Value	On - Off
Input Conversion	Directa / Inversa
Acces Name para I/O Divise	Cadena de Caracteres
Item (Registro de dispositivo de entrada / salida y bit, o variable de programa remoto)	
Use Tagname como Item Name	On / Off
Alarm State	On / Off / None
Alarm Priority	Numérica
Dispositivo de Entrada / Salida	(variables DDE solamente)

VARIABLE DE CADENA DE CARACTERES (STRING)

Tagname	Cadena de Caracteres
Alarm Group	Cadena de Caracteres
Log Data	On / Off
Log Events	On / Off
Log Events Priority	Numérica
Retentive Value	On / Off

DICCIONARIO DE DEFINICION DE ENTRADAS

Los siguientes campos están disponibles para el configurador durante el modo de edición:

ANALOGO INCLUYENDO ENTEROS Y NUMEROS REALES

Tagname	Cadena de Caracteres
Alarm Group	Cadena de Caracteres
Log Historical Data	On / Off
Log Events	On / Off
Log Events Priority	Numérica
Retentive Value	On / Off

(El sistema iniciará con el valor antiguo durante una falla en la alimentación de energía)

Retentive Alarm Parameters

(Permite cambios a los valores límites de las alarmas durante la ejecución)

Initial Value Numérica

Deadband Numérica

Min EU Numérica

Max EU Numérica

(Rango de la variable del proceso en unidades de ingeniería)

Min Raw Numérica

Max Raw Numérica

(Rango de la variable del proceso en unidades enteras)

Engineering Units Cadena de Caracteres

Conversion Línea/Raíz Cuadrada

Access Name para I/O Cadena de Caracteres

Item Cadena de Caracteres

(Registro de dispositivo de entrada/salida o variable de programa remoto)

Use Tagname como Item Name On / Off

Historical Logging Deadband Numérica

Lolo Alarm Enable On / Off

Lolo Alarm Value Numérica

Lolo Alarm Priority Numérica

Lolo Alarm Enable On / Off

Lolo Alarm Value Numérica

Lolo Alarm Priority	Numérica
High Alarm Enable	On / Off
High Alarm Value	Numérica
High Alarm Priority	Numérica
High Alarm Enable	On / Off
High Alarm Value	Numérica
High Alarm Priority	Numérica
Alarm Value Deadband	Numérica
Maximun Length	Numérica
Initial Value	Cadena de Caracteres
Access Name for I / O Device*	Cadena de Caracteres
Item*	Cadena de Caracteres
(I / O Device Register o Remote Program Variable)	
Use Tagname como Item Name*	On / Off

*Dispositivo de entrada / salida (variable DDE) solamente

BIBLIOGRAFIA

1. Ericsson, Industrial Circuits Application Note "*Stepper Motor Basics*".
2. Kuo, B. *Theory and Applications of Step Motors*, (USA: West Publishing, 1974), Capítulo 10.
1. Leatham-Jones, B. *Elements of Industrial Robotics*, (Great Britain: Pitman Publishing, 1987), pp. 1-6, 23-27, 31-34, 51-54.
1. Mc Donald, A. *Robot Technology, Theory, Design and Applications*, (New Jersey: Prentice Hall, 1986), pp. 10-19, 124-130, 215-218.
1. Maloney, T. *Electrónica Industrial Moderna*, (3ra. edición, México: Prentice Hall Hispanoamericana, 1997), pp. 283-294, 686 - 694.
1. Rashid, M. *Electrónica de Potencia*, (2da. edición, México: Prentice Hall Hispanoamericana, 1995), pp. 356 - 372.
1. Siemens, *Simatic Sistema de Automatización S7-200 Manual de Sistema*, 1997.
1. Wonderware, *Basic Training Course Manual of InTouch*, (Irvine, USA, Wonderware Corporation, 1996).