



T
629.895
FAR



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**"DISEÑO DE UNA MAQUINA MEZCLADORA
UTILIZANDO EL PLC Y UN SOFTWARE DE
VISUALIZACION DE PROCESOS"**

TOPICO DE GRADUACION

**Previo a la Obtención del Título de:
INGENIERO EN ELECTRICIDAD
Especialización: INDUSTRIAL**

**Presentada por:
Freddy Geovanny Fares Vargas**

**Guayaquil - Ecuador
1999**

AGRADECIMIENTO

A Dios y a mi familia, por todo el apoyo recibido durante mi carrera universitaria.

A mis maestros, por sus enseñanzas impartidas.

Y a todas las personas que contribuyeron para la culminación del presente trabajo.

DEDICATORIA

A MIS PADRES
A MIS HERMANOS
AL M.J.E.



.....
ING. ARMANDO ALTAMIRANO

Presidente del tribunal

.....
Alberto Larco

ING. ALBERTO LARCO

Director de Tópico

.....
Norman Chou Tong

ING. NORMAN CHOUTONG

Miembro del Tribunal

.....
Nelson Layedra

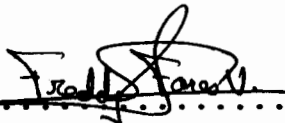
ING. NELSON LAYEDRA

Miembro del Tribunal

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, me 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).


.....

FREDDY GEOVANNY FARES VARGAS

RESUMEN

El trabajo presente muestra los conocimientos adquiridos en el transcurso de los tópicos de graduación y de la carrera en general. El sistema diseñado constituye básicamente una aplicación de los controladores lógicos programables (PLC), de un software de visualización de procesos y de electrónica básica. Se trata de un sistema mezclador en el que se utiliza un tambor con ocho orificios; estos orificios sirven de recipientes para contener y mezclar dos sustancias.

El programa de control de todo el proceso es basado en la CPU 212 de Siemens, para el proceso en modo automático se han utilizado 5 entradas digitales y 6 salidas tipo relé y para el proceso en modo manual se necesitaron adicionalmente 7 entradas digitales.

Finalmente con el software de visualización de procesos "In Touch" se diseñó la pantalla que visualiza la forma de operación del sistema. Esta pantalla diseñada sirve para visualizar como funciona el sistema.

INDICE GENERAL

	Pág.
RESUMEN	VI
INDICE GENERAL	VII
INDICE DE FIGURAS	X
INTRODUCCION	XII
CAPITULO I	
PRINCIPIOS BASICOS DE CONTROLADORES PROGRAMABLES	14
1.1 INTRODUCCION	14
1.2 ESTRUCTURA INTERNA DE UN PLC	18
1.2.1 UNIDAD CENTRAL DE PROCESO	19
1.2.2 UNIDAD DE MEMORIA	22
1.2.3 UNIDAD DE ENTRADA Y SALIDA	23
1.3 OPERACIÓN INTERNA Y PROCESAMIENTO DE LAS SEÑALES EN LOS PLCs	25
1.3.1 PROCESAMIENTO DE ENTRADAS Y SALIDAS	26
CAPITULO I I	
DISEÑO DEL PROGRAMA DE CONTROL	29
2.1 INTRODUCCION	29
2.2 ESPECIFICACIONES DE FUNCIONAMIENTO DEL SISTEMA	30
2.2.1 DIAGRAMA DE BLOQUES	32
2.2.2 DIAGRAMA DE FLUJO DEL PROGRAMA	33

2.3	DISPOSITIVOS MECANICOS QUE CONSTITUYEN LA MEZCLADORA	35
2.4	TIPO DE PLC A USARSE	36
2.5	ENTRADAS Y SALIDAS	36
2.5.1	ENTRADAS DIGITALES	37
2.5.2	SALIDAS DIGITALES	38
2.6	DESCRIPCION Y ANALISIS DE LOS SEGEMENTOS DEL PROGRAMA	39
2.7	CODIGO DEL PROGRAMA DE CONTROL	43
CAPITULO I I I		
	CIRCUITOS CONTROLADORES	56
3.1	CIRCUITO CONTROLADOR DE UN MOTOR DE PASOS	56
3.2	CIRCUITO DE POSICION DEL TAMBOR	63
3.3	CIRCUITO DE CONTROL DE VELOCIDAD DEL BATIDOR	67
CAPITULO I V		
	SIMULACION GRAFICA DEL PROCESO	72
4.1	INTRODUCCION	72
4.2	DOCUMENTACION	73
4.2.1	CREACION DE LOS OBJETOS GRAFICOS	74
4.2.2	ENLACES DE ANIMACION	76
4.3	APLICACIÓN SCRIPT	80
4.4	COMUNICACIÓN DE IN TOUCH CON LA CPU 212	85
	CONCLUSIONES	88

RECOMENDACIONES	89
APENDICE	91
APENDICE A	91
SOFTWARE DE PROGRAMACION DE LAS CPUs S7-200	91
A.1 INTRODUCCION	91
A.2 LENGUAJES DE PROGRAMACION	92
A.3 EJECUCION DEL PROGRAMA EDITADO	93
A.4 REALIZACION DE UN PROGRAMA	94
A.5 COMPILACION DEL PROGRAMA	95
A.6 DIVISION DE MEMORIA DE LA CPU S7-200	96
A.6.1 MEMORIA DE DATOS	96
A.6.2 MEMORIA DE PARAMETROS	97
A.6.3 MEMORIA DE PROGRAMA	97
A.6.4 MEMORIA DE VARIABLES	97
A.7 EDICION DE LOS PROGRAMAS DE USUARIO EN KOP	98
APENDICE B	99
BREVE ESTUDIO DE SOFTWARE DE VISUALIZACION DE PROCESOS INDUSTRIALES "IN TOUCH".	99
B.1 INTRODUCCION	99
B.2 ANIMACION Y ENLACES DE GRAFICOS	100
B.2.1 CREANDO ENLACES DE ANIMACION	101
B.2.2 ENLACES DE ENTRADA	102

B.3	CONJUNTO BASICO DE INSTRUCCIONES Y FUNCIONES	103
	PARA LA EDICION DE LOGIN SCRIPTS	
B.3.1	INTRODUCCION A LOS SCRIPTS	103
B.3.2	APLICACIÓN SCRIPTS	103
B.4	SERVIDOR DDE	104
B.4.1	INTRODUCCION A DDE Y SERVIDORES DDE	104
B.5	COMUNICACIÓN ENTRE EL SERVIDOR DDE E IN TOUCH	108
	BIBLIOGRAFIA	120

INDICE DE FIGURAS

FIG.		Pág.
1.1	Estructura del PLC	20
1.2	Arquitectura del PLC	21
1.3	Circuito Opto aislador	24
2.1	Diagrama de bloques del sistema	32
2.2	Diagramas de flujo del programa	33
2.3	Dispositivos mecánicos que constituyen la mezcladora	35
2.4	Entradas digitales	37
2.5	Salidas digitales	38
3.1	Diagrama de bloques para el control de un motor de pasos	57
3.2	Forma de onda del voltaje para manejar dos fases	59
3.4	Circuito controlador de un motor de pasos	60
3.4	Carga paralela del registro	61
3.5	Contador de anillo	62
3.6	Circuito de ubicación del tambor	65
3.7	Circuito troceador básico	69
3.8	Circuito variador de velocidad	70
4.1	WindowsView	84

INTRODUCCION

Se trata de automatizar y diseñar un sistema mezclador, el mezclador cuenta con un tambor rotatorio el cual tiene 8 orificios y los cuales sirven de recipientes para el mezclado; un émbolo giratorio es desplazado hacia cada uno de los recipientes para realizar la mezcla y obtener la homogeneización de la sustancia líquida y la sustancia sólida.

Los principales objetivos que se pretenden alcanzar en el desarrollo de este proyecto son los siguientes:

- Profundizar en el estudio de los controladores lógicos programables, especialmente la CPU 212 de Siemens.
- Usar el software de visualización de procesos "In Touch" de Wonderware.

Para empezar a diseñar el control del sistema mezclador fue necesario estudiar el Software de programación Step 7-Micro/Win que asiste las CPUs S7-200. Este software permite realizar diversas funciones tales como introducir, editar, depurar e imprimir el programa de usuario, etc.

Después de estudiar la CPU 212 se procedió a estudiar el software de visualización de procesos "In Touch", el cual nos permite ver en la pantalla de la computadora mediante representaciones gráficas los procesos en tiempo real.

El diseño del programa de control que se implanta en la memoria de programa del PLC, está diseñado de tal manera que se utiliza un mínimo de circuitos externos.

La posición del tambor es determinada por un sensor óptico; el cual genera una entrada para el programa cargado en el PLC. La posición de reposo y final del émbolo batidor son detectados por dos conmutadores que indican al programa su posición. Los circuitos que controlan la posición del tambor, los motores y las válvulas solenoides son sencillos; ya que solo se necesitaron conocimientos de electrónica básica para su diseño. Como se puede ver, se ha disminuido el uso de hardware externo al utilizarse un controlador lógico programable.

CAPITULO I

PRINCIPIOS BÁSICOS DE CONTROLADORES PROGRAMABLES

1.1. INTRODUCCION

Los primeros PLCs fueron desarrollados a inicios del año de 1970 y fueron usados principalmente para controlar motores industriales, donde se reemplazó a inmensos paneles de relés. Estos PLCs fueron capaces de proveer el control requerido y con menos espacio que el sistema de cableado de relés equivalentes y fueron también más exactos en sus largos periodos de operación.

Los PLCs son extremadamente flexibles en los términos

de modificar de manera fácil la secuencia de control si es necesario. Es posible alterar un sistema de control sin tener que recurrir a cambios de conexiones físicas. Solo es necesario cambiar la lógica de un programa residente, usando un pequeño teclado que va conectado con el controlador programable.

El incremento de las aplicaciones de los controladores lógicos en los procesos industriales ha fomentado la manufactura y el desarrollo de las familias de sistemas basadas en microprocesadores, teniendo varios niveles de desarrollo. El rango de PLCs disponibles se extiende ahora desde pequeñas unidades con 20 entradas y salidas digitales y con 500 pasos de programa hasta sofisticados sistemas modulares, así mismo, con un amplio rango de funciones modulares para tareas de entradas y salidas análogas y comunicaciones. Estos métodos modulares permiten la expansión o actualización de un sistema de control con mínimos costos y disturbios.

Con los avances tecnológicos del último siglo, se han logrado crear los circuitos integrados de muy alta escala, por medio de los cuales se han podido crear

escala, por medio de los cuales se han podido crear máquinas lógicas programables, como el PLC, que está formado por una Unidad Central de Proceso (CPU), y es la que controla las operaciones de la máquina y realiza las funciones de procesamiento de datos. Estas máquinas programables incorporan una variedad de instrucciones y funciones de control y que son ejecutadas cíclicamente por el programa creado por el usuario.

Los controladores lógicos programables (PLC) ofrecen a los sistemas de control, amplia seguridad de sus operaciones y flexibilidad en el desarrollo de los sistemas, ya que nos permite realizar cambios en la lógica de control de una manera sencilla, es decir, solo se requiere cambiar la lógica del programa de usuario, y no necesariamente recurrir a cambios físicos.

Los PLCs proveen facilidad y flexibilidad de control basadas sobre programas y ejecución simple de instrucciones. Los PLCs tienen funciones internas

tales como temporizadores, contadores y registros de desplazamientos.

Un controlador programable opera examinando las señales de entrada de un proceso y llevando la lógica de instrucciones (con las cuales ha sido programada en la memoria) sobre esas señales de entrada, produciendo señales de salida para manejar procesos o maquinarias.

Interfaces estándar propias de los PLCs permiten conectarse directamente a los actuadores de los procesos y transductores (ejemplo, bombas y válvulas) sin la necesidad de circuitos intermedios o relés.

A través del uso del PLC se hace posible modificar un sistema de control sin tener que desconectar o cambiar la ruta de un cableado; solo es necesario cambiar el programa de control. Los PLCs son similares a las computadoras "convencionales" en términos de tecnología de hardware, ellos tienen características específicas acompañadas del control industrial:

- Inmunidad al ruido

- Construcciones modulares, permitiendo fácil reemplazo y adición de unidades;
- Conexión estándar de Entradas / Salidas y niveles de señal;
- Fácil comprensión del lenguaje de programación

Estas características hacen de un controlador programable altamente deseable en una amplia variedad de plantas industriales y situaciones de control de procesos.

1.2. ESTRUCTURA INTERNA DE UN CONTROLADOR LÓGICO PROGRAMABLE

Se define como estructura de un sistema, la forma en como los componentes están relacionados entre sí. Los controladores programables están formados por tres áreas funcionales: procesador, memoria, y unidad de Entradas / Salidas. Las condiciones de entrada al PLC son leídas y almacenadas en memoria donde el PLC ejecuta el programa de instrucciones lógicas sobre estas entradas. Las condiciones de salida son entonces generadas al controlador del equipo asociado. La acción tomada depende totalmente del programa que se

mantiene en memoria. En la figura 1.1 se aprecia en bloques la estructura básica de un controlador lógico programable.

1.2.1. UNIDAD CENTRAL DE PROCESO

La CPU controla y supervisa todas las operaciones del PLC, llevando hacia él las instrucciones del programa almacenadas en memoria. Una vía de comunicación interna rápida, o sistema de bus, lleva información hacia y desde la CPU. Las unidades de memoria y unidades de entrada y salida están bajo el control de la CPU. La CPU cuenta con una frecuencia de reloj generada por un cristal de cuarzo externo o por un circuito oscilador RC, típicamente entre 1 y 8 Megahertz dependiendo del microprocesador usado y del área de aplicación. El reloj determina la velocidad de operación del PLC y provee temporización y sincronización para todos los elementos del sistema (ver figura 1.2).

Virtualmente todos los controladores programables modernos están basados en un microprocesador.

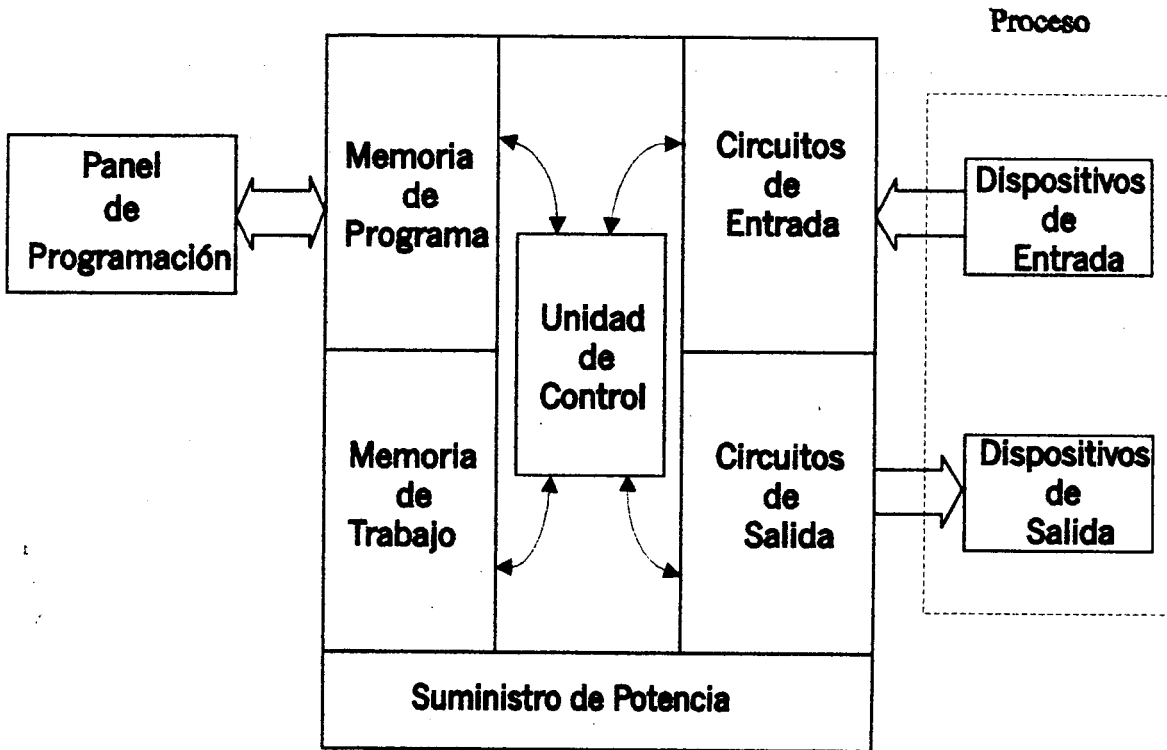


Figura 1.1 Estructura del PLC

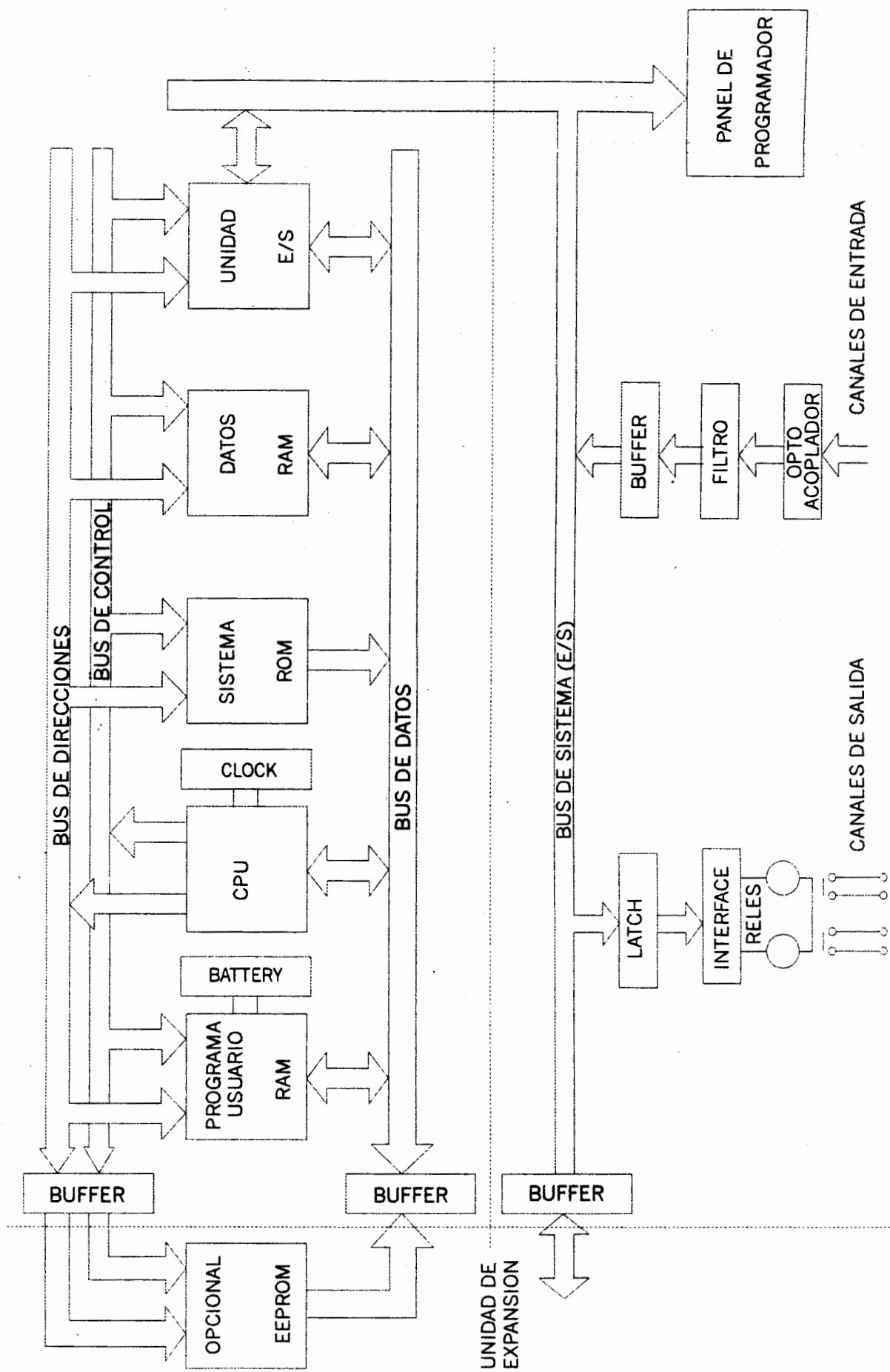


Figura 1.2 Arquitectura del PIC

Algunos controladores programables grandes también emplean microprocesadores adicionales y que trabajan en forma paralela para controles más complejos, tales como funciones de tiempo, procesamiento matemático y controladores proporcionales Integrales diferenciales, etc.

1.2.2. UNIDAD DE MEMORIA

Para almacenar el programa de control todos los controladores lógicos programables modernos usan dispositivos de memoria de material semiconductor, tales como la memoria RAM (memoria de acceso aleatorio), que es memoria de lectura/escritura o memoria programable de solamente lectura, tales como son las familias EPROM o EEPROM.

- Virtualmente la mayoría de las memorias RAM son utilizadas para desarrollo y prueba de programas iniciales, ya que este tipo de memoria permite realizar cambios en el programa de una manera rápida y sencilla, al contrario de las EPROM y de las EEPROM.

1.2.3. UNIDAD DE ENTRADA Y SALIDA

La mayoría de los PLCs operan internamente entre 5 y 15 V d.c. mientras que las señales de los procesos pueden ser más grandes, típicamente de 24 V d.c a 240 V a.c. con varios amperios de corriente.

La unidad de Entrada/Salida constituye la interface entre la microelectrónica del controlador programable y el mundo real externo, y debe así mismo proveer todas las funciones de aislamiento y acondicionamiento necesario de las señales. A menudo se permite que un PLC sea conectado directamente a los actuadores y transductores del proceso (ejemp. Bombas y válvulas) sin la necesidad de circuitos intermedios o relés.

Los canales de Entrada/Salida son eléctricamente aislados del proceso de control, usando circuitos opto-aisladores sobre los módulos de entrada y salida. Un circuito opto-aislador consiste de un diodo de emisión de luz y un foto transistor tal como se lo puede ver en la figura 1.3, formando un

par opto acoplador que permite pasar pequeñas señales a través de él.

Las unidades de Entradas y Salidas son diseñadas con el objetivo de simplificar la conexión de los transductores del proceso con el controlador programable.

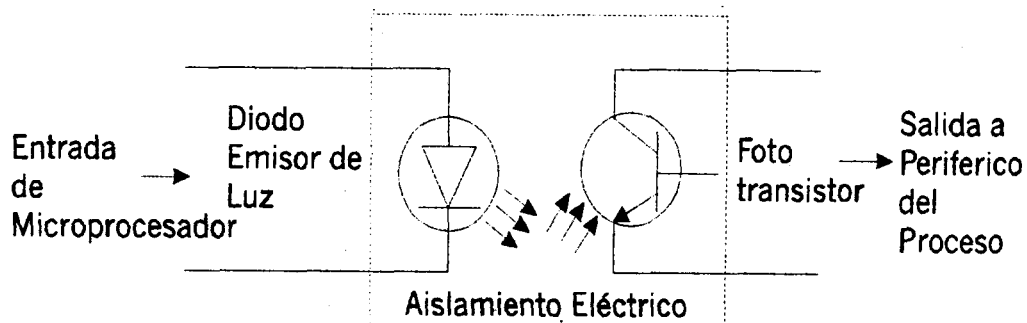


Figura 1.3 Circuito opto aislador

Para este propósito todos los PLCs son equipados con terminales estándar atornillados o conectados sobre algún punto de la Entrada/Salida, permitiendo reemplazar y remover rápidamente las tarjetas de E/S cuando estén defectuosas.

Todas las Entradas y Salidas tienen una única dirección o un número de canal el cual es usado durante el desarrollo del programa para especificar el monitoreo de una entrada o la activación de una salida particular dentro del programa.

1.3. OPERACIÓN INTERNA Y PROCESAMIENTO DE LAS SEÑALES EN LOS PLCS

Cuando un programa es cargado en el PLC, las instrucciones son ubicadas en localizaciones de memoria individual (dirección).

La CPU contiene un registro contador de programa el cual apunta a la próxima instrucción a ser buscada en la memoria. Cuando una instrucción es recibida por la CPU esta es ubicada en el registro de instrucciones para decodificarlas en operaciones internas (micro instrucciones) requeridas por la instrucción particular. Por ejemplo, puede resultar de instrucciones que son leídas de memoria, o en un dispositivo físico que esta siendo manejado por la CPU.

Cuando el controlador programable está inicialmente seteado en Run, el contador de programa apunta la dirección 0000, que es la localización de la primera instrucción. La CPU entonces busca, decodifica y ejecuta esta instrucción del programa

1.3.1. PROCESAMIENTO DE ENTRADAS Y SALIDAS.

Existen dos métodos diferentes para el procesamiento de Entradas/Salidas en controladores lógicos programables: (a) actualización continua, y (b) copia conjunta de Entradas/Salidas.

(a) Actualización continua

La CPU rastrea los canales de entrada tal como ocurren en las instrucciones del programa, con retardos propios medidos, se validan solamente las señales de entrada que son leídas en el procesador. (El retardo es típicamente de 3 ms el cual previene los pulsos de rebotes y otros ruidos de entrada al PLC). Los canales de salida son manejados cuando las instrucciones de salida son ejecutadas siguiendo una operación lógica.

Las salidas son sostenidas en una unidad de E/S de tal manera que ellos retengan sus estados hasta la próxima operación.

(b) Copia conjunta de entradas y salidas

En los PLCs más grandes se pueden tener cientos de puntos de entradas y salidas. Además la CPU solamente puede tratar con una instrucción a la vez durante la ejecución del programa, el estado de cada punto de entrada debe ser examinado individualmente para determinar el efecto de esta entrada sobre el programa. Se requieren por lo menos 3 ms de retardo para cada entrada, el tiempo total del ciclo para un sistema que funcione continuamente aumentará a medida que aumente la razón de entradas.

Para una rápida ejecución del programa, las entradas y salidas actualizadas pueden ser llevadas a un punto particular del programa. Una área específica de memoria RAM dentro del PLC es usada como un buffer de almacenamiento entre la lógica de control y la unidad de E/S, cada entrada y cada salida tiene una celda en esta RAM

de E/S. Durante la copia de E/S, la CPU busca todas las Entradas en la unidad de E/S y copia su estado en las celdas de RAM de E/S. Esto mejora el inicio y fin de cada ciclo del programa.

Cuando el programa es ejecutado, los datos de Entradas almacenados son leídos una a la vez en la localización de E/S de la RAM. Las operaciones lógicas son ejecutadas sobre los datos de entrada, y resultando señales de salidas que son guardadas en la sección de E/S de la RAM. Entonces al finalizar cada ciclo del programa todas las señales de salida de la RAM se transfieren a los correspondientes canales de salida. Las etapas de salida son sostenidas, es decir retienen sus valores en sus últimos estados hasta que ellos sean actualizados por la próxima rutina de E/S.

Esta tarea es llevada automáticamente por la CPU como una subrutina al programa normal. La copia de las E/S toma lugar entre el fin de un ciclo de programa y el inicio del próximo.

CAPITULO II

DISEÑO DEL PROGRAMA DE CONTROL

2.1. INTRODUCCION

En este capítulo consta el diseño del programa que controlará el proceso de mezclado. El control es de lazo abierto, ya que no se cuenta con realimentación. Se ha utilizado la CPU 212 de Siemens con su respectivo software de programación. Este software permite editar y cargar el programa en la memoria de la CPU. El tipo de lenguaje empleado es el KOP (esquema de contactos o diagrama de escalera), por ser muy familiar con lo aprendido en controles industriales eléctricos.

La confección del programa de control consta de algunas fases y consisten en comprender claramente el funcionamiento del sistema, en realizar diagramas de bloques, diagramas de flujo del proceso y la resolución del diagrama de flujo con las instrucciones que posee el software de programación de la CPU.

2.2. ESPECIFICACIONES DE FUNCIONAMIENTO DEL SISTEMA.

Se trata de automatizar el mezclado de dos sustancias, una sustancia líquida y una sustancia sólida, para lo cual existe un tambor rotatorio, que es movido por un motor de pasos, en el tambor existen ocho orificios que sirven de recipientes y en los que se realiza la mezcla de las dos sustancias para posteriormente batirla, dando lugar a la homogeneización.

En un principio, considerando que el émbolo batidor se encuentra en reposo, empieza el ciclo de trabajo activándose la señal que controla al motor y que da movimiento al tambor hasta que llegue a la posición de llenado y mezclado. La entrada "Lib_Sust_Sol"

(liberación de sustancia sólida) permite la activación de la electroválvula que controla la salida de la sustancia sólida hacia el recipiente por un tiempo determinado, seguidamente se activa la señal "Lib_Sust_liq" (liberación de sustancia líquida), la cual activa otra electroválvula, permitiendo el paso de la sustancia líquida por un corto tiempo, una vez que un orificio posee las dos sustancias, se activa el motor 1 y permite girar el tambor, el cual se mueve hasta que se activa el detector de posición, luego se activa la señal que baja el émbolo con el batidor, hasta que llega al tope inferior y que es señalado por la entrada "Detect_Emb_Baj" (Detector de émbolo en posición baja), donde permanece activado un tiempo con el propósito de mezclar y homogeneizar la mezcla. Finalmente, el émbolo sube, activándose "Sub_Emb_Bat" (subida de émbolo con batidor) hasta que el émbolo se ubica en la posición de reposo, con lo cual finaliza el ciclo.

Mientras en un recipiente se está llenando con las dos sustancias, en el anterior, es decir en el que ya fue llenado, se está realizando el mezclado, y así

sucesivamente hasta completar el llenado y mezclado de los ocho recipientes con lo cuál concluye el proceso.

2.2.1. DIAGRAMAS DE BLOQUES DEL SISTEMA

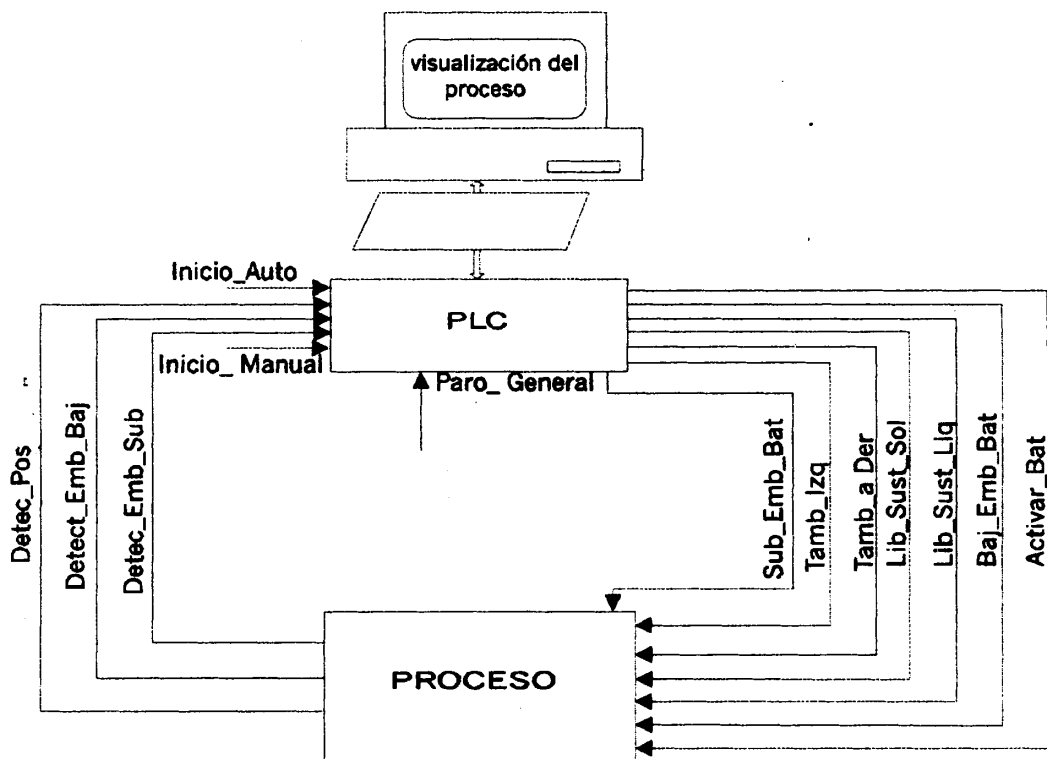


Figura 2.1 Diagrama de bloques del sistema

2.2.2. DIAGRAMAS DE FLUJO DEL PROGRAMA

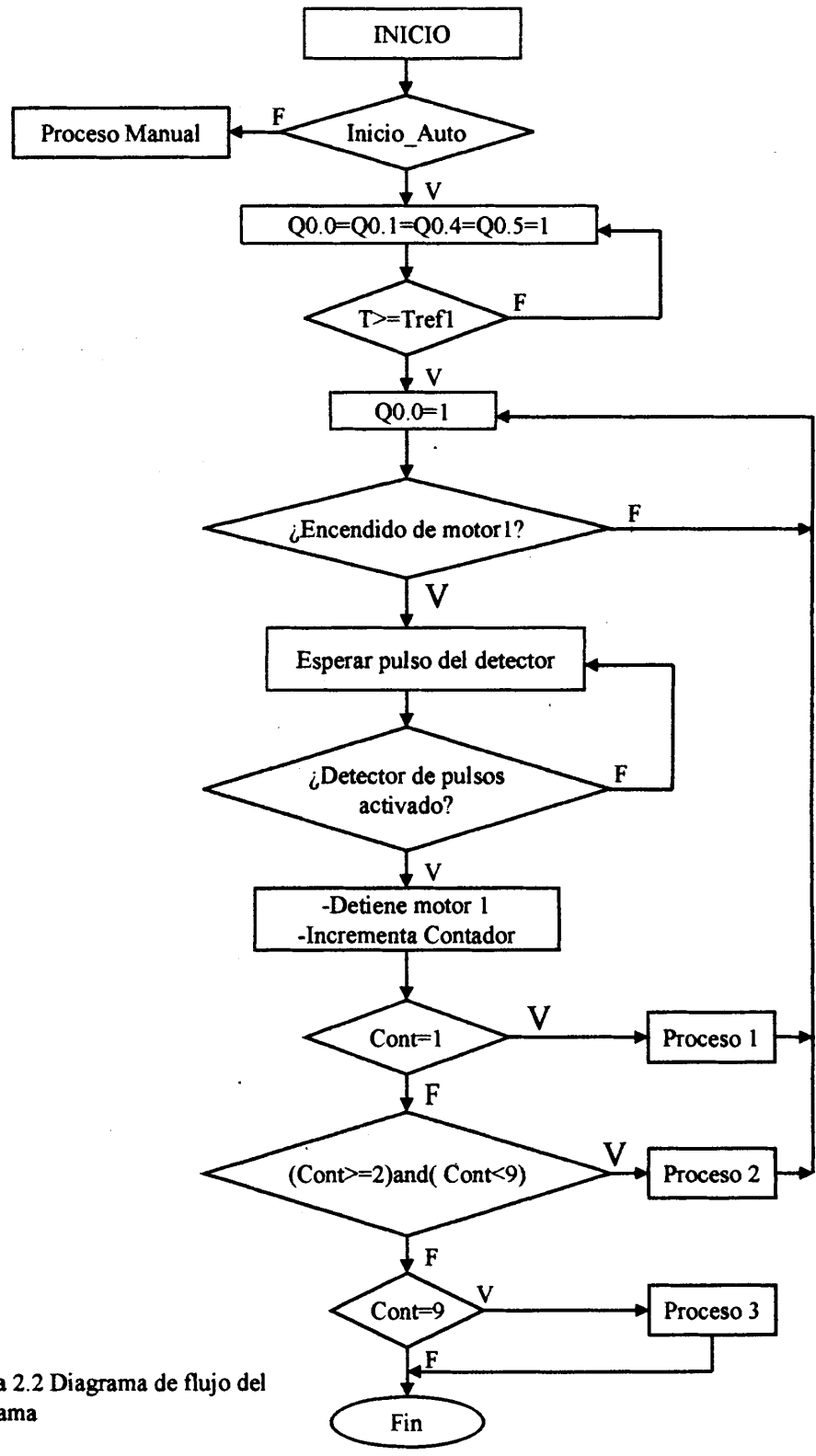
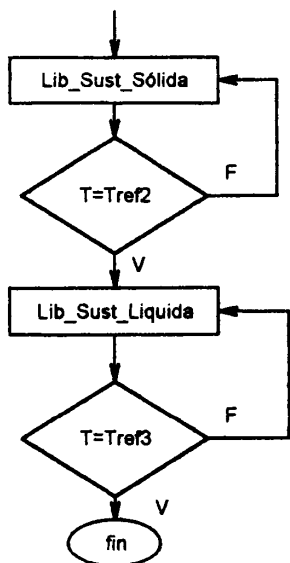
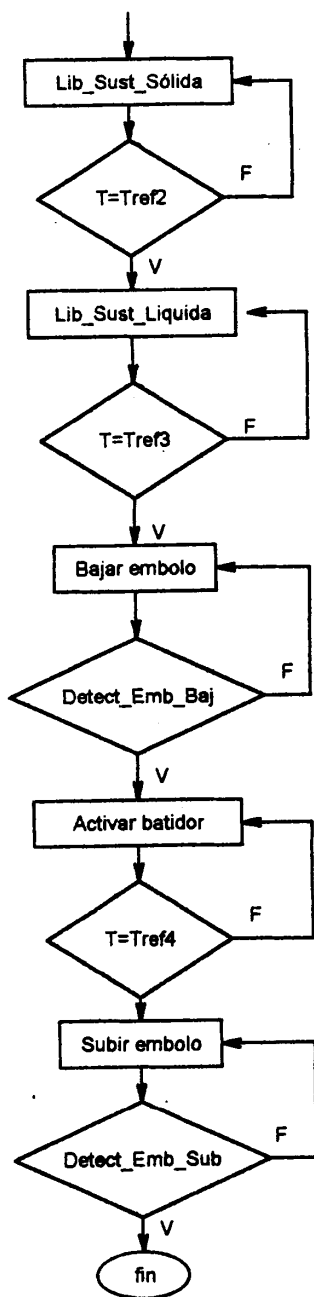


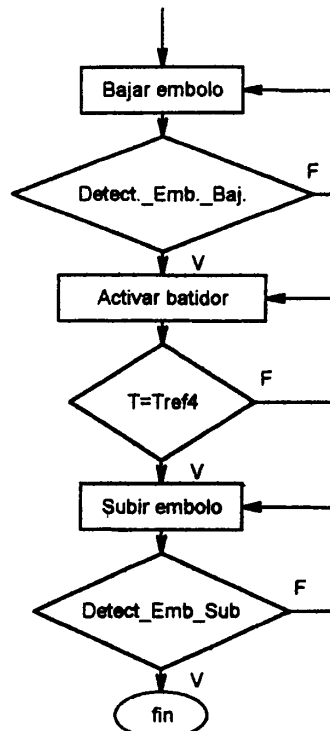
Figura 2.2 Diagrama de flujo del programa



Proceso 1



Proceso 2



Proceso 3

Figura 2.2. (Continuación)

2.3. DISPOSITIVOS MECANICOS QUE CONSTITUYEN LA MEZCLADORA

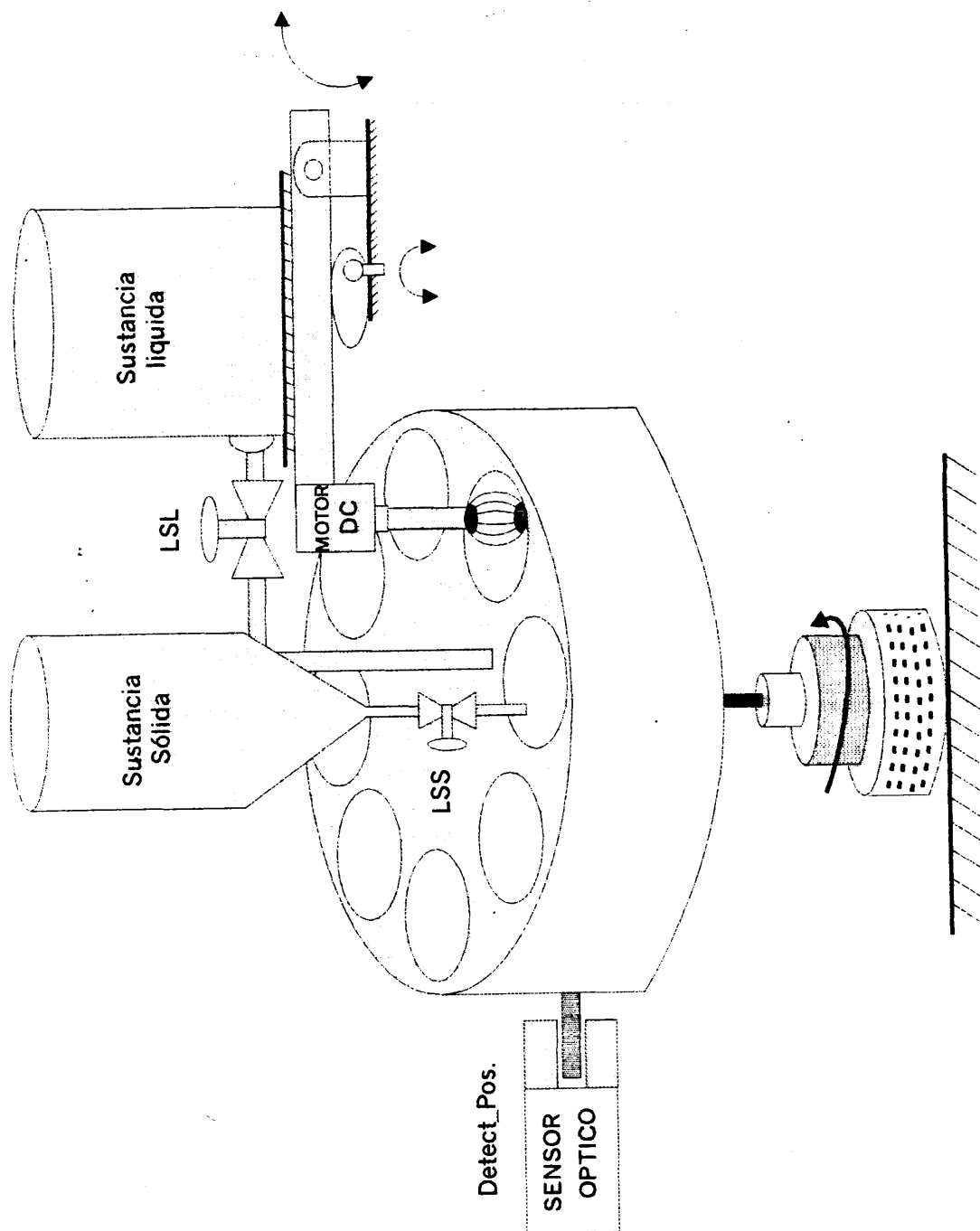


Figura 2.3. Dispositivos mecánicos que constituyen la mezcladora

2.4 TIPO DE PLC A USARSE

El controlador programable a usarse es la CPU 212 de Siemens, con alimentación AC, entradas DC tipo fuente y salidas de relé.

Un modulo de ampliación de entradas digitales 8 x DC 24 V y un modulo de 8 salidas de relé, son necesarios para cumplir todas las condiciones del proceso.

2.5 ENTRADAS Y SALIDAS

La CPU 212 de Siemens con 2 módulos de ampliación satisfacen todas las necesidades de entradas y salidas para automatizar el sistema, en el sistema automatizado se emplean entradas digitales y salidas de relé. Las entradas digitales son de tipo fuente con un rango de voltaje directo de DC 15 V a 30 V.

Las salidas digitales son de tipo relé con un margen de tensión de DC 5 V a 30 V/AC 250 V, y con una corriente de carga máxima de 2 Amp.

2.5.1. ENTRADAS DIGITALES

Inicio_auto		10.0	Inicio en modo automático
Paro_General		10.1	Detiene el proceso
Detect_pos		10.2	Detecta la pos. del recipiente
Detect_Emb_Baj		10.3	Detecta la pos. final del émbolo
Detect_Emb_Sub		10.4	Detecta la Pos. inicial del émbolo
Inicio_Manual		10.5	Inicio en modo manual
Tamb_Izq		10.6	Giro del tambor hacia la izquierda
Tamb_a Der		10.7	Giro del tambor hacia la derecha
Lib_Sust_Sol		11.0	Libera sustancia sólida
Lib_Sust_Liq		11.1	Libera sustancia líquida
Bajar_Emb_Bat		11.2	Baja émbolo batidor
Activar_Bat		11.3	Activa batidor
Sub_Emb_Bat		11.4	Sube émbolo batidor

Figura 2.4 Entradas digitales

2.5.2. SALIDAS DIGITALES

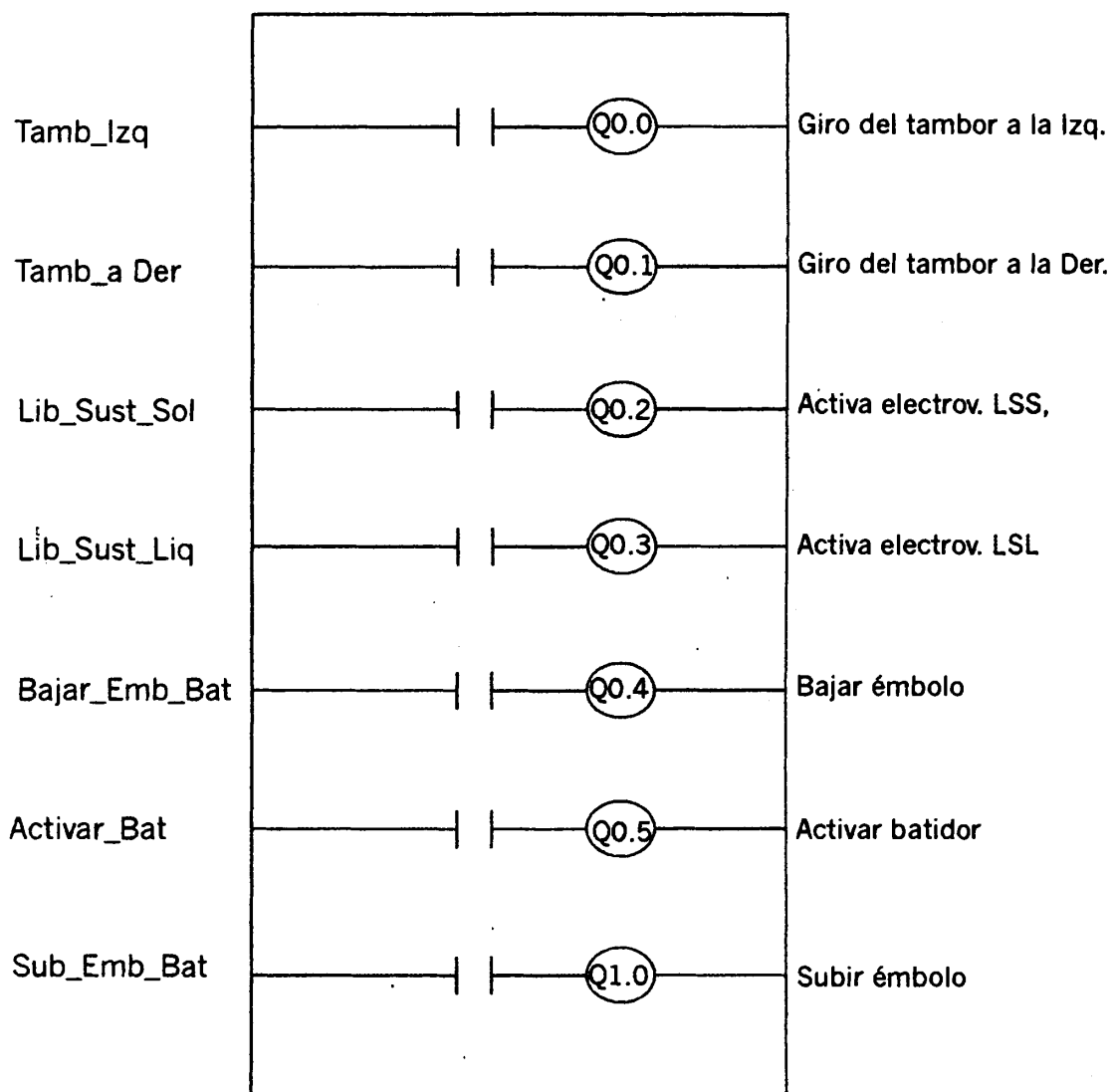


Figura 2.5 Salidas digitales

2.6. DESCRIPCIÓN DEL PROGRAMA Y ANÁLISIS DE LOS SEGMENTOS.

El programa a sido editado en lenguaje KOP, o conocido como lenguaje de esquema de contactos; por ser muy similar a la forma aprendida en controles industriales eléctricos.

La codificación del programa se basa en instrucciones sencillas. Se utilizan marcas internas como relés de control, contadores, temporizadores y la función **MOV** que es la principal en el desarrollo del programa de control. Como se han empleado Entradas y Salidas estándar, estas se actualizarán al finalizar cada ciclo;

Se utiliza la entrada con la dirección I0.0 para dar inicio al proceso en modo automático y la entrada I0.1 para detener el proceso. La entrada que indica la posición correcta del tambor es indicada por la entrada I0.2. se utiliza las direcciones de entrada I0.3 y I0.4 para indicarle al controlador programable las posiciones iniciales y finales del émbolo batidor. Las salidas Q0.0 y Q0.1 son utilizadas para controlar

S1 y S0 del registro de desplazamiento y que a la vez controlan la dirección de giro del tambor; la salida Q0.2 gobierna el encendido y apagado de la electroválvula que libera la sustancia sólida; Q0.3 es utilizada para gobernar el encendido o apagado de la electroválvula que libera la sustancia líquida; La salida Q0.4 sirve para activar el motor que hace bajar al embolo batidor; Q0.5 activa al batidor y por último Q1.0 se utiliza para regresar el émbolo a la posición de reposo.

Un temporizador se utiliza para mantener por un cierto tiempo las entradas del registro de desplazamiento activadas; dos temporizadores controlan el tiempo de activación de las electroválvulas las cuales controlan la salida de las sustancias líquidas y sólidas, por medio de estos tiempos se determina la cantidad de sustancia requerida para un recipiente; otro temporizador controla el tiempo de mezclado de las sustancias.

en la figura 2.2 se presentaron los diagramas de flujo del programa, los mismos que son después codificados

codificados con las instrucciones del software de programación de la CPU 212.

A continuación se explican cada uno de los segmentos que conforman el programa.

- Segmento 1** La Entrada "Inicio_Auto" da inicio a la secuencia de funcionamiento del sistema en modo automático, habilitando la función MOV se activan las salidas Q0.0, Q0.1, Q0.4 y Q1.0. Estas salidas son las que controlan S1 y S0 de los registros de desplazamiento y que a la vez controlan a los motores de pasos
- Segmento 2** Temporiza la carga de los 2 registros de desplazamiento
- Segmento 3** Se activa la salida Q0.0, que es la que controla a S1 del registro de desplazamiento e indica el giro del tambor hacia la izquierda.
- Segmento 4** Con la pulsación de la Entrada "Paro_general" se detiene y se encera todo el proceso de manera general.

- Segmento 5** La Entrada "Detect_Pos" de manera indirecta incrementa al contador de ciclos del proceso.
- Segmento 6** El contador C0 controla el número de ciclos del proceso.
- Segmento 7** Se hace la comparación de C0 con 2 para ver si se ejecuta el lazo en el que solo se realiza el llenado de las dos sustancias.
- Segmento 8** La activación de la marca M0.2 activa la bobina M0.3.
- Segmento 9** Se compara C0 con 9, si es igual se activa la marca M1.1 y se ingresa al lazo en el que solo realiza el mezclado del último recipiente.
- Segmento 10** La marca M1.1 activa la bobina M1.2 y permite ingresar al lazo de solo mezclado.
- Segmento 11** Se realiza el llenado del primer recipiente con la sustancia líquida y con la sustancia sólida.
- Segmento 12** Si C0 es mayor o igual que nueve se activa la bobina M0.4, la cual es usada

como condición para ingresar al lazo en el cual solo se realiza el mezclado.

Segmento 13 Al activarse M0.4 se activa M0.5 como un relé auxiliar.

Segmento 14 Se realiza el llenado del segundo al octavo recipiente, y se realiza la mezcla de primero hasta el séptimo recipiente.

Segmento 15 En este segmento se activan las salidas necesarias para solo realizar el mezclado.

Segmento 16-17 en estos dos segmentos manteniendo activada la entrada I0.5 se puede realizar el proceso en modo manual.

Segmento 18 El programa termina con la bobina absoluta, Finalizar programa principal (END)..

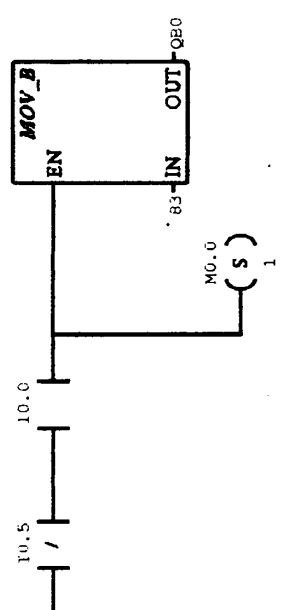
2.7. CÓDIGO DEL PROGRAMA DE CONTROL

De la página 44 a la 55 se presenta la codificación del programa, la misma que está dividida en 18 segmentos.

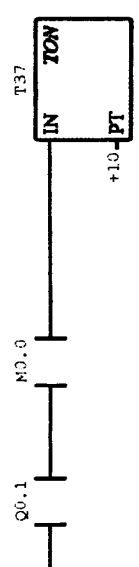
COMENTARIOS SOBRE EL TÍTULO DEL PROGRAMA

Pulse F1 para obtener ayuda y un programa de ejemplo ..

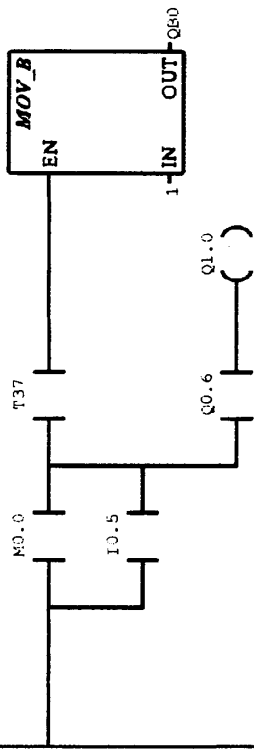
Network 1 Inicio del proceso en modo automatico



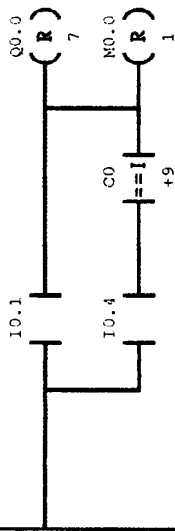
Network 2 Temporiza la carga del registro de control de los dos motores



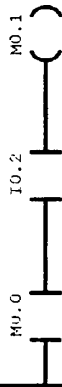
Network 3 Giro del tambor



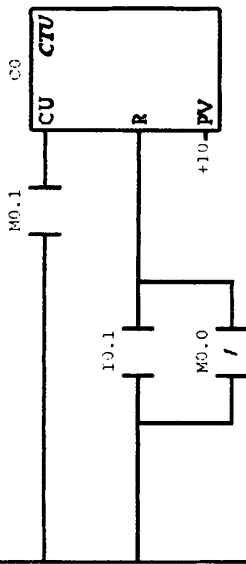
Network 4 Detiene el proceso



Network 5 Incremento de contador



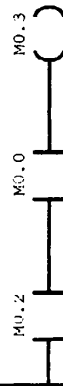
Network 6 Contador de recipientes llenados y mezclados



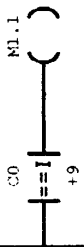
Network 7



Network 8



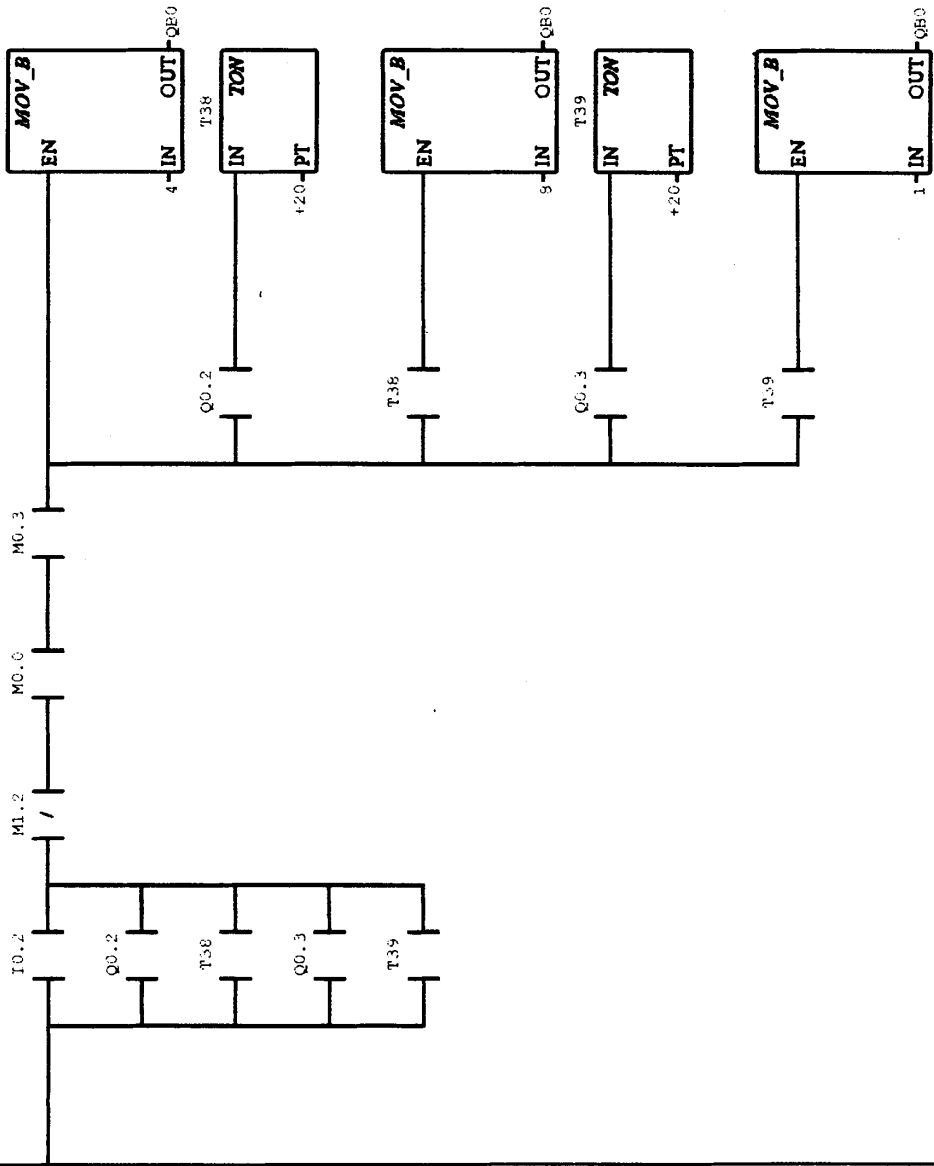
Network 9



Network 10



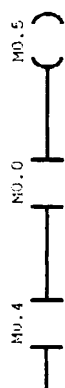
Network 11 Llena el primer recipiente con las dos sustancias

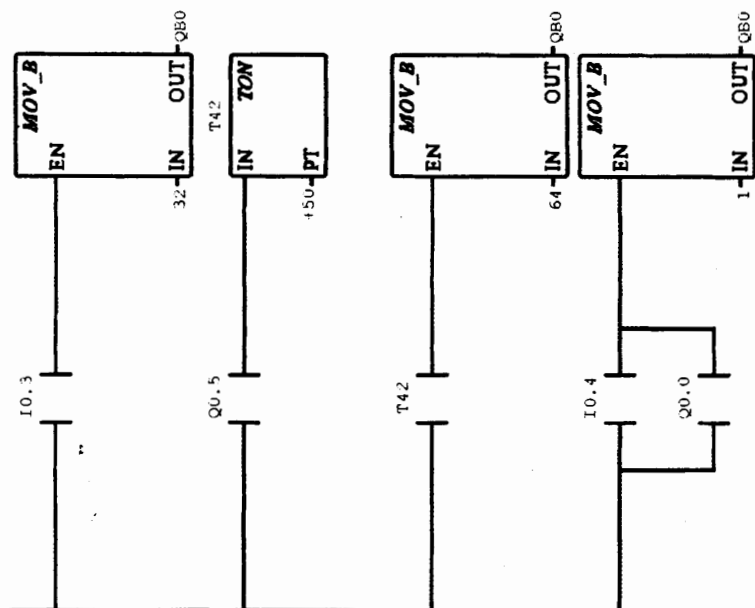


Network 12

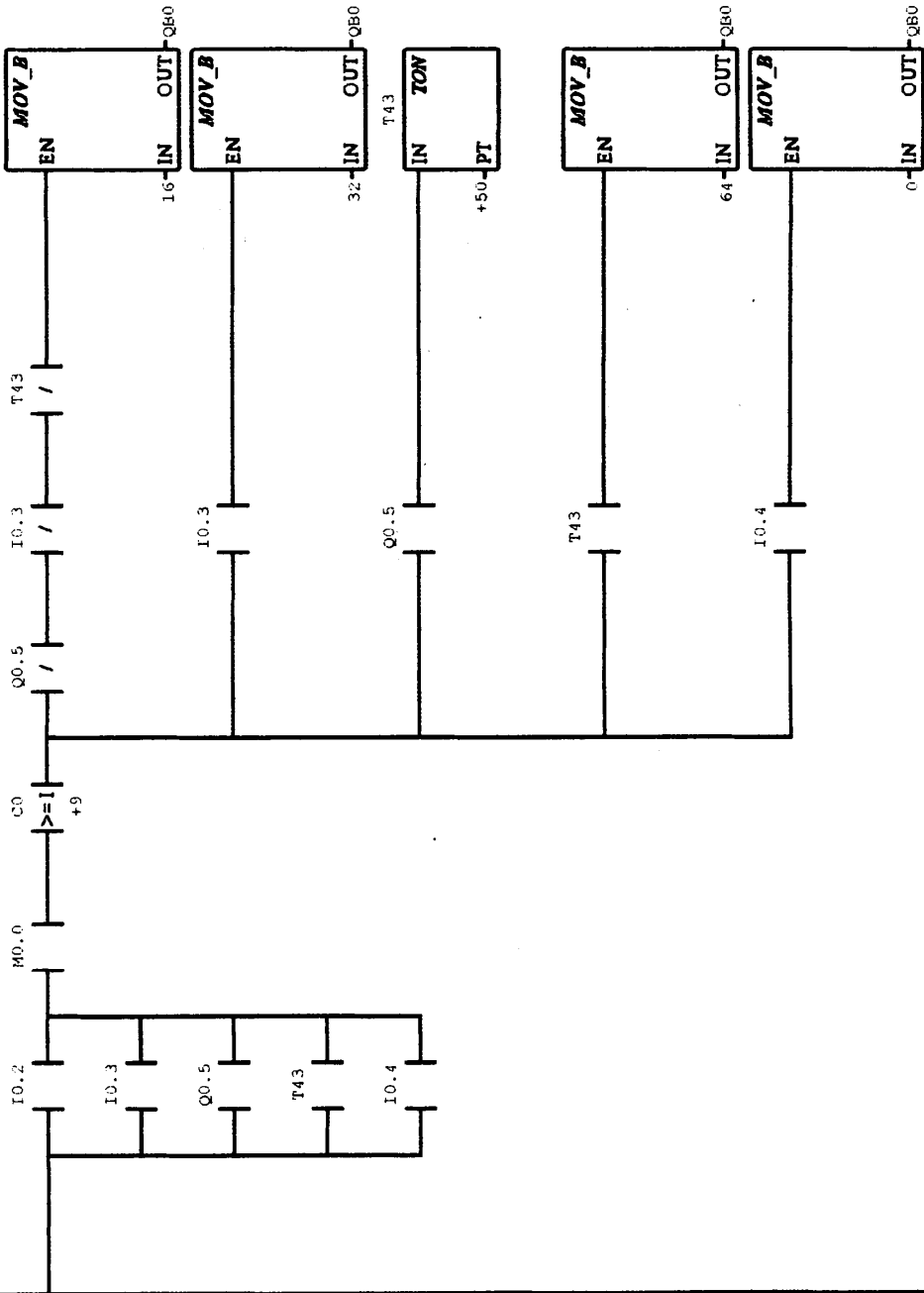


Network 13

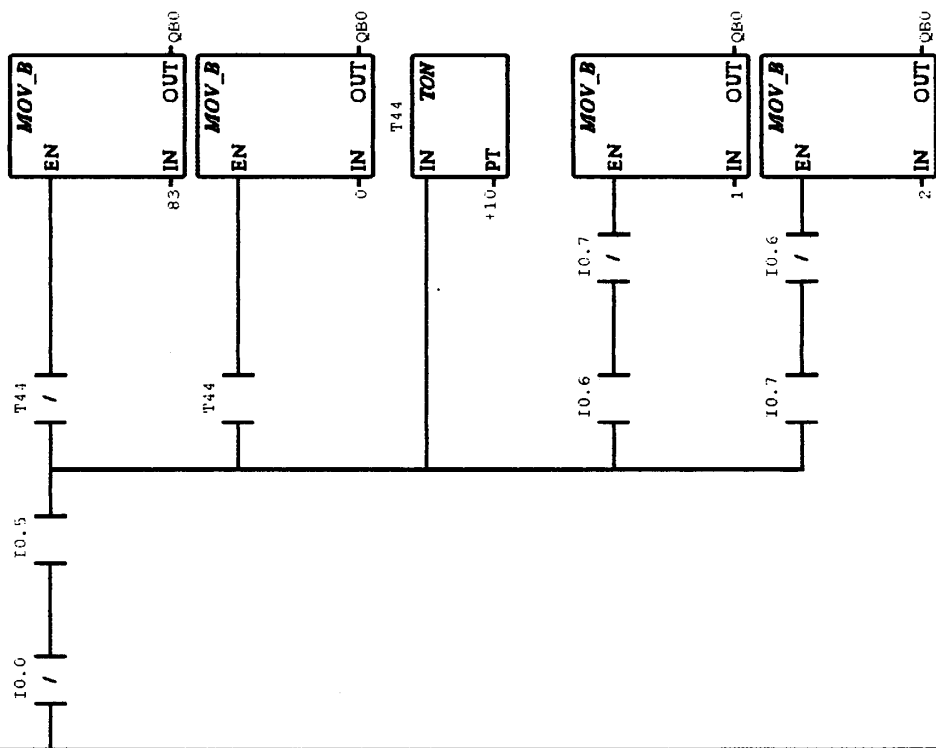




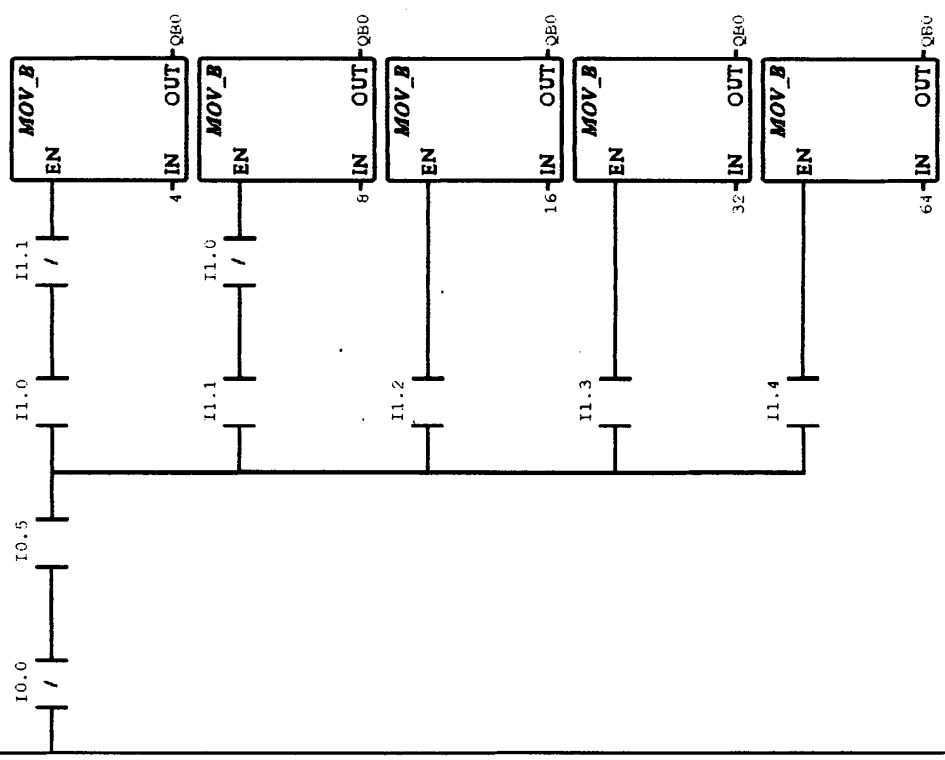
Network 15 Mezcla el ultimo recipiente



Network 16 Operacion en modo manual.



Network 17



Network 18 Fin del programa

(END)

CAPITULO III

CIRCUITOS CONTROLADORES

3.1. CONTROLADOR DE UN MOTOR DE PASOS.

Un motor de pasos es utilizado en el diseño para girar y ubicar el tambor en la posición de llenado y en la posición de mezclado de las dos sustancias, y otro motor de las mismas características es requerido para subir y bajar el émbolo batidor.

Los motores de pasos son controlados por una secuencia de pulsos digitales; estos pulsos alimentan indirectamente las fases del estator del motor. La razón principal de utilizar estos motores es de que

permiten variar su velocidad en función de la frecuencia de los pulsos que alimentan las fases del estator del motor.

El diagrama de bloques de la figura 3.1 muestra los elementos básicos para el control de un motor de pasos, el sistema consiste de un bloque de secuencia lógica, un bloque manejador de potencia y un bloque de suministro de potencia. La función del manejador de potencia consiste en aceptar señales lógicas de nivel bajo en forma de tren de pulsos digitales; esta secuencia de control con la respectiva etapa de amplificación de corriente alimenta las fases del motor, produciendo un movimiento angular discreto.

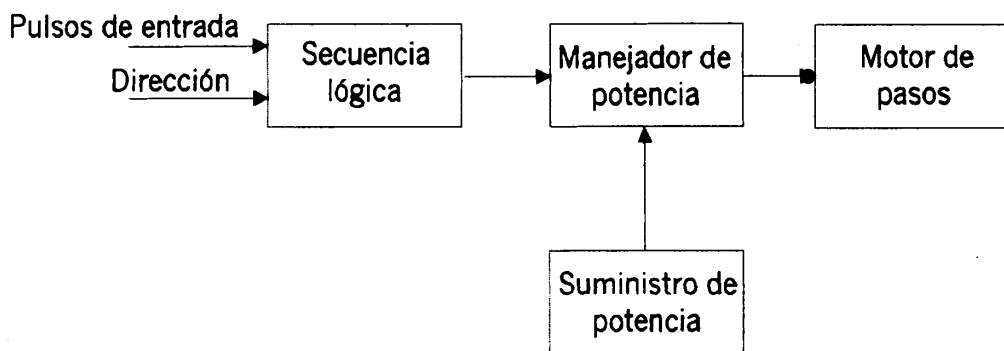


Figura 3.1 Diagrama de bloques de control de un motor de pasos

La sección de secuencia lógica acepta pulsos de entrada con el respectivo comando de dirección, además este bloque suministra una señal de nivel bajo a cada uno de los circuitos de conmutación de potencia, que no son más que transistores tipo Darlington. El bloque de secuencia lógica consiste de un contador de anillo de 4 etapas y que es igual al número de fases del motor.

Los pulsos de entradas en el contador de anillo hacen que se desplacen los bits ya sea a la derecha o a la izquierda, dependiendo de la dirección de giro que se quiera.

El registro de desplazamiento de 4 bits (74LS194) y un circuito temporizador 555 son los elementos principales utilizados en el diseño del circuito de control del motor.

En el diseño del circuito de control se alimentan dos fases del motor a la vez, y las cuales se van alternando secuencialmente de dos en dos, tal como aparece en la figura 3.2.

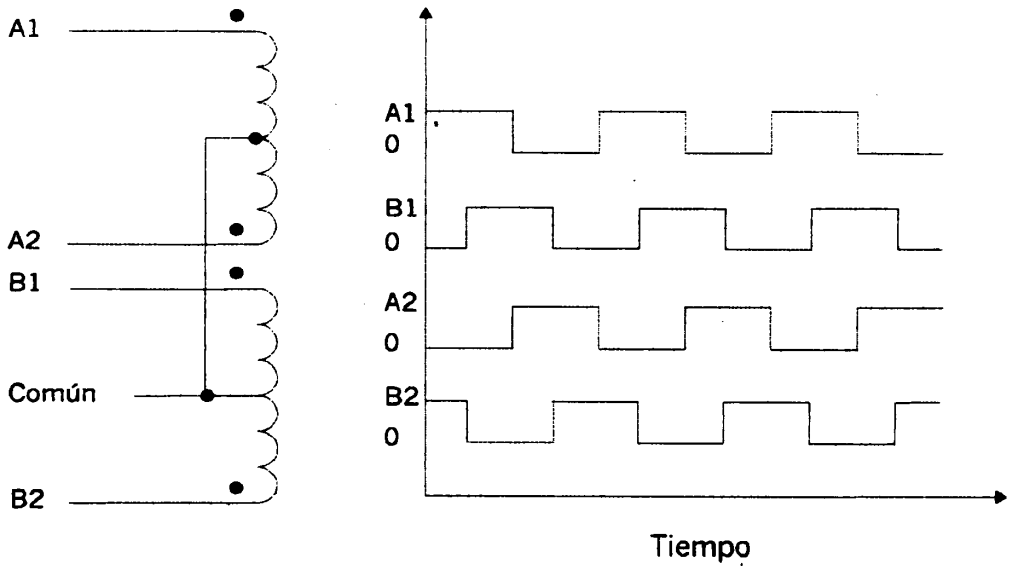


Figura 3.2 Forma de onda del voltaje para manejar dos fases

El circuito que genera la secuencia de pulsos y el circuito de fuerza que alimenta al motor es presentado en la figura 3.3, en la figura se puede apreciar de manera general el bloque de generación de pulsos, este bloque controla la alimentación de las fases del motor polarizando o no directamente las uniones P-N de los transistores de configuración "Darlington".

Para obtener la rotación o desplazamiento de los bits, primero se carga en la salida del registro los valores

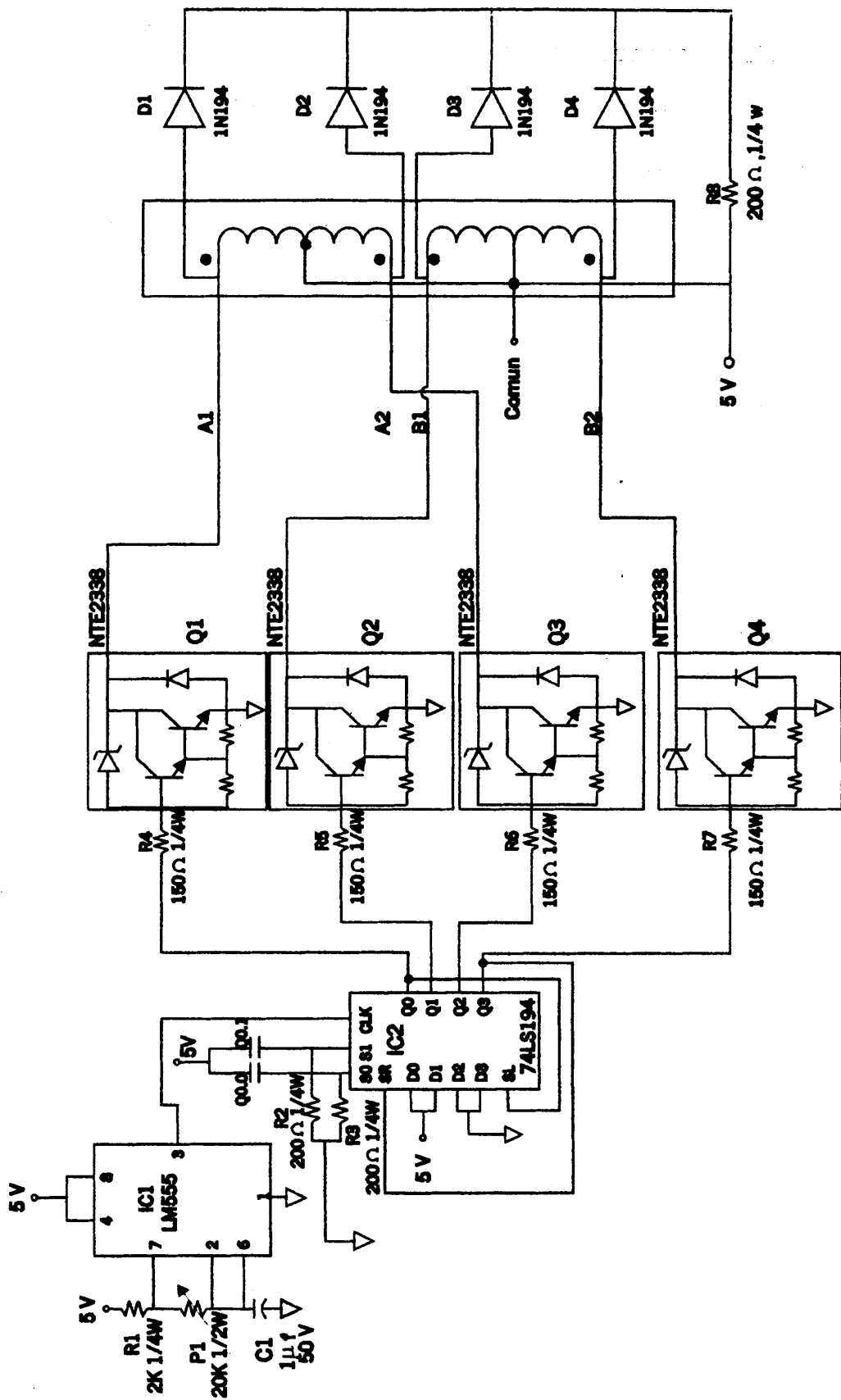


Figura 3.3 Circuito controlador de un motor de pasos

que se tienen en la entrada paralela, esto se lo hace aplicando un pulso de voltaje en las entradas S1 y S0 del registro de desplazamiento, tal como se lo puede ver en la figura 3.4.

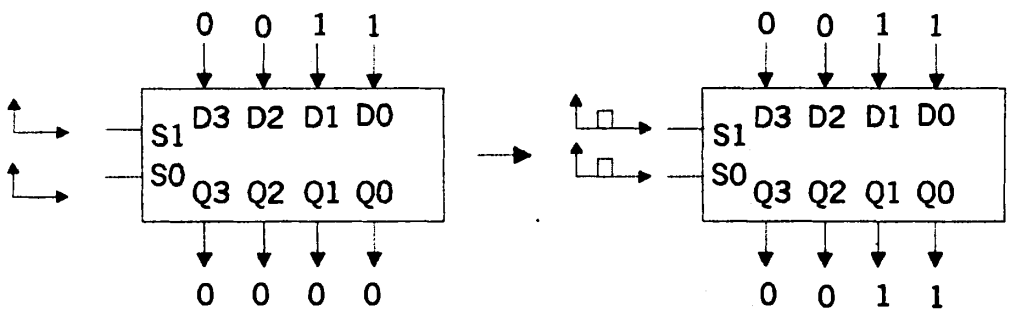


Figura 3.4 Carga paralela del registro

Después de que se encuentren en la salida los valores 1100, se activa una de las dos señales, S1 o S0, para dar inicio al desplazamiento de los bits y por ende el giro del motor ya sea a la izquierda o hacia la derecha. Como en un contador de anillo una vez que el último bit llega a la posición final este es ingresado por la entrada serial derecha, si es que se trata de un desplazamiento hacia la derecha, o por la entrada serial izquierda, si es un desplazamiento hacia la izquierda tal como se lo aprecia en la figura 3.5.

Desplazamiento hacia la izquierda				Desplazamiento hacia la derecha			
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0
0	0	1	1	0	0	1	1
0	1	1	0	1	0	0	1
1	1	0	0	1	1	0	0
1	0	0	1	0	1	1	0

Figura 3.5 Contador de anillo

La parte amplificadora de corriente esta formada por transistores Darlington NTE 2338, estos transistores contienen internamente un diodo de protección y un diodo zener para limitar el voltaje que se produce en las bobinas cuando se les suprime bruscamente la corriente. Para que el transistor Darlington no trabaje de manera forzada se a agregado un diodo en paralelo con cada bobina del motor, con el objetivo de limitar externamente el alto voltaje que se genera en las bobinas.

A continuación se presentan las características del transistor NTE 2338 y del motor de pasos usado.

CARACTERÍSTICAS DEL TRANSISTOR DARLINGTON NTE 2338

Máx. Corriente De colector	BVCBO	BVCEO	BVEBO	$\beta_{\text{máx}}$
1.5 A	60±10	60±10	7	3000

CARACTERÍSTICAS ELÉCTRICAS DEL MOTOR

VOLTAJE	CORRIENTE	PRECISIÓN
4.0 DC	0.9 A	1.8 GRAD./PASO

La posición inicial y final del émbolo es indicada por dos conmutadores, uno indica la posición en la cual el émbolo esta arriba o en reposo y el otro indica la posición baja del émbolo. Estos Conmutadores, conmutan una señal constante de 24 voltios y que son leídas como entradas por la CPU.

3.2. CIRCUITO DE POSICIÓN DEL TAMBOR

Un acoplador óptico es utilizado para ubicar el tambor en la posición de llenado y en la posición que deben tener los recipientes para el respectivo mezclado de las sustancias. Este sensor óptico esta formado por un

diodo "Emisor" y por un transistor "Detector", cada posición correcta del recipiente hará que se interrumpa el paso de la luz en el optoacoplador y por lo tanto mantendrá en estado de corte al transistor detector. Esto es aprovechado para energizar y desenergizar un relé de 12 V DC, el cual conmuta los 24 V DC de entrada para el PLC.

El foto transistor posee una unión P-N colector-base fotosensible. La corriente inducida por el efecto foto Eléctrico es la corriente de base del transistor. Si asignamos la notación I_{λ} para la corriente de base foto inducida, la corriente de colector resultante, de forma aproximada, es:

$$I_c = \beta I_{\lambda} \quad (3.1)$$

Como se puede ver en la ecuación 3.1, un incremento en la intensidad de luz corresponde a un incremento en la corriente del colector.

El circuito que se presenta en la figura 3.6 sirve para polarizar el foto transistor y es utilizado como transductor de entrada para el controlador lógico programable. Como se puede ver, el circuito consta de dos subcircuitos, uno para el diodo emisor y otro para el transistor receptor.

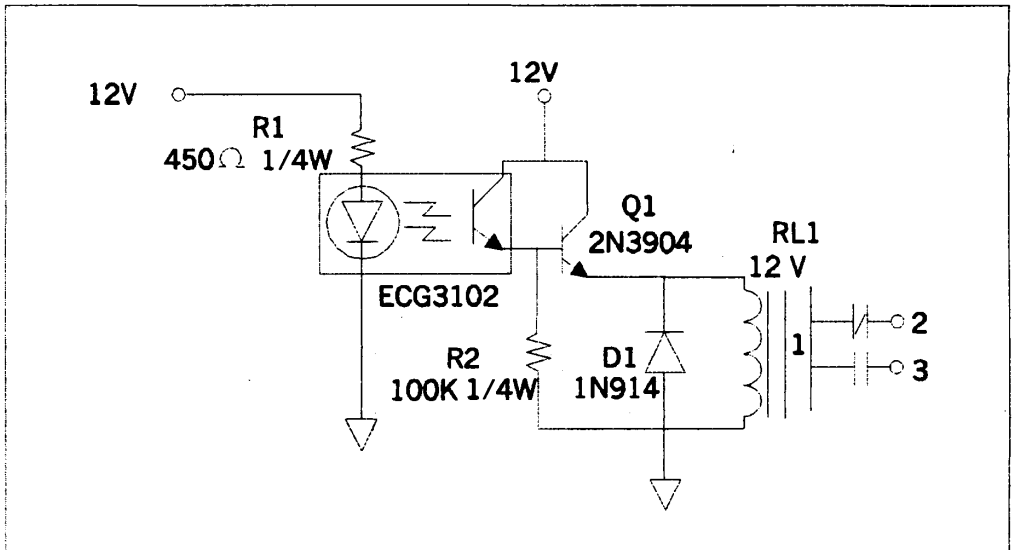


Figura 3.6 Circuito de ubicación del tambor

Al polarizarse directamente el diodo emisor, este se enciende, cayendo a través de sus terminales un voltaje alrededor de 1.2 V y pasando una corriente

cuyo valor se lo halla por medio de la siguiente ecuación.

$$I_d = \frac{(V_{cc} - V_d)}{R_a} \quad (3.2)$$

$$I_d = \frac{(12 - 1.2)V}{450\Omega}$$

$$I_d = 26.2\text{ma.}$$

El valor de corriente obtenido es un valor menor del máximo permitido por el foto transistor, esta corriente sirve para excitar el transistor de salida, obteniéndose la saturación del mismo. Este transistor fotosensible sirve para excitar a otro transistor NPN, obteniéndose una configuración de transistor "Darlington" y que sirve para amplificar corriente, y con esto alimentar un relé de 12 Voltios. Los contactos de este relé sirven para conmutar los 24 V que son leídos por la CPU.

Al quitarse la alimentación del relé se produce un voltaje grande en los terminales de la bobina, debido

a que la bobina no permite un cambio brusco de corriente, por esta razón se ha conectado en paralelo un diodo, el cual sirve para descargar la corriente de la bobina y con lo cual se eliminan los voltajes picos.

Las características del opto acoplador ECG3102 son las siguientes:

Especificaciones del emisor

Máxima corriente	$I_f = 60 \text{ ma}$
Voltaje de caída	$V = 1.7 \text{ V}$

Especificaciones del detector

Voltaje Máximo colector-emisor	$V = 55 \text{ V}$
---	--------------------------------------

maxima corriente colector	$I_c = 100 \text{ ma}$
--------------------------------------	--

3.3. CIRCUITO DE CONTROL DE VELOCIDAD DEL BATIDOR

El motor empleado para el batidor es de 12 voltios de corriente continua, como se necesita inicialmente establecer la velocidad de giro del motor de acuerdo a

las necesidades, se ha diseñado un circuito variador de velocidad.

El motor de corriente continua es el indicado para este tipo de aplicación en el que se requiere cambio de velocidad, por arriba y por debajo de sus valores nominales.

La velocidad de un motor de DC está definida por:

$$\omega = \frac{(V_t - R \cdot I_a)}{K \cdot \phi} \quad (3.3)$$

existen dos métodos generales para gobernar la velocidad de un motor DC:

- 1.- variando la tensión en los terminales de la armadura del motor.
- 2.- variando el flujo magnético.

La segunda opción no es posible aplicarla en el tipo de motor escogido, por lo tanto, la única opción aplicable es la primera.

La variación del voltaje es obtenida para este caso por medio de un convertidor DC a DC, conocido como troceador. Un circuito básico de un troceador es mostrado en la figura 3.7.

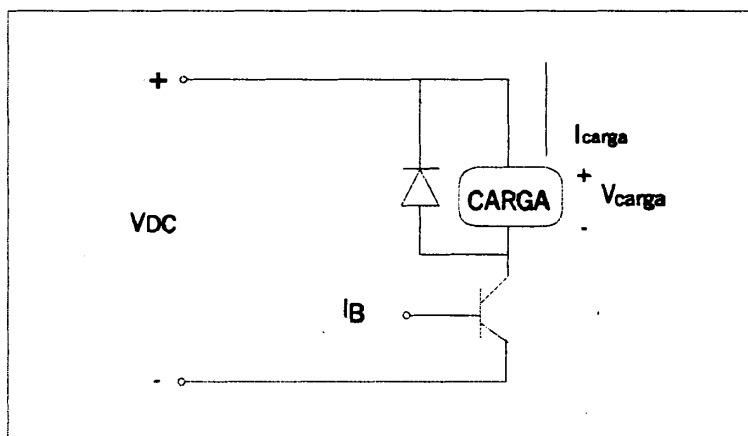


Figura 3.7 Circuito troceador básico

El circuito troceador diseñado es hecho a base de un circuito integrado LM555, con el cual se genera una onda cuadrada y con ancho de pulso variable, este pulso es aplicado a la base de un transistor el cual actúa como un interruptor, determinando el voltaje promedio aplicado a los terminales del motor. La figura 3.8 muestra el circuito general que permite variar manualmente la velocidad del motor DC. La

variación de la velocidad se la consigue variando la frecuencia de salida del circuito oscilador, circuito que es implementado con el circuito integrado LM555.

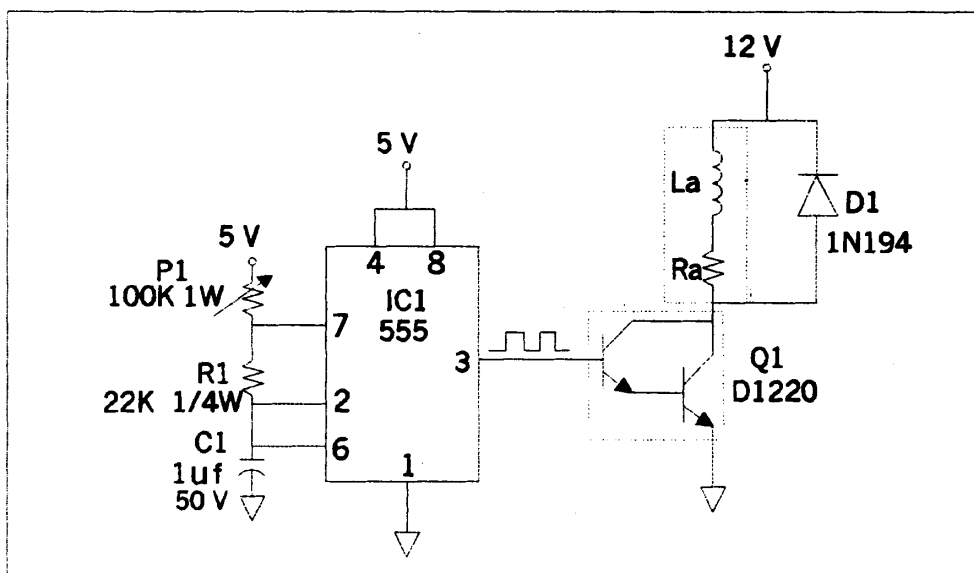


Figura 3.8 Circuito variador de velocidad.

Para obtener una frecuencia de 100 milisegundos a la salida del 555 se seleccionó un capacitor polarizado con un valor de $1\mu\text{F}$, y la resistencia R1 con un valor de 22K. P1 es en realidad un potenciómetro de 100K.

de 22K. P1 es en realidad un potenciómetro de 100K. Los valores son obtenidos por medio de las siguientes ecuaciones

$$T_{\text{alto}} \approx 0.7(P_1 + R_1)C \quad (3.4)$$

$$T_{\text{bajo}} \approx 0.7R_1 * C \quad (3.5)$$

Por lo tanto y por característica propia del motor DC, se obtiene la variación de velocidad del motor al variarse el voltaje de entrada promedio.

CAPITULO IV

SIMULACION GRAFICA DEL PROCESO

4.1. INTRODUCCION

En este capítulo se presenta la pantalla de simulación del proceso, la pantalla ha sido elaborada con el software de visualización de procesos, In Touch, de Wonderware.

Este software trabaja con dos tipos básicos de datos, tales como los datos tipo Memoria y los datos tipo DDE. Se han usado los datos tipo memoria para ejecutar una demostración del proceso sin la necesidad de conectarse a un servidor DDE real.

Para que el proceso dibujado en la pantalla principal de In Touch, pueda comunicarse con el proceso real externo, se debe utilizar los tags o variables tipo DDE para adquirir datos continuamente del medio exterior.

En el gráfico elaborado en **WindowsMaker** (ventana de elaboración de gráficos de In Touch), se han utilizado objetos simples tales como líneas, curvas y textos teniendo cada uno de estos sus propios atributos, los cuales afectan su apariencia.

La aplicación script contiene el programa que gobierna y manipula las variables y que a su vez controlan los objetos o símbolos creados.

La aplicación Script es ejecutada cíclicamente por períodos que se pueden escoger dentro de un determinado rango.

4.2. DOCUMENTACIÓN

Se detalla la forma en que son creados los objetos con

Sus respectivos enlaces de animación, las variables o tagnames utilizados y la creación de la aplicación Scripts, que es la que maneja la simulación del proceso de manera general.

4.2.1. Creación de los objetos gráficos

Todos los objetos gráficos son creados en la pantalla **WindowsMaker** de In Touch. Existen objetos simples y compuestos. Los objetos simples son las líneas, las curvas, textos y que son de base para la creación de objetos compuestos. Los objetos compuestos son la unión de objetos simples en un símbolo.

El tambor que sirve para el mezclado esta formado por varios objetos simples, tales como: las elipses, que representan la vista superior y frontal de los recipientes del tambor.

Los recipientes grandes que contienen las sustancias son rectángulos simples y forman parte del menú de la caja de herramientas. Para insertarlos en la

pantalla, se ubica el puntero del mouse sobre el rectángulo y se hace clic sobre el sitio donde se lo desee pegar.

El dispositivo mezclador esta formado por varios cuadros, dos circunferencias simulan los visores para los niveles del émbolo batidor. Cuando el émbolo se encuentra en posición baja cambian de apariencia una de ellas indicando la posición, Cuando el émbolo esta en la posición de reposo se activa el otro visor.

La paleta mezcladora esta formado por tres rectángulos que son las que ingresan en el recipiente para realizar la mezcla.

Se ha dibujado una válvula, por medio de la herramienta que permite crear los símbolos, que no es otra cosa que la unión de varios objetos simples.

Entre los objetos propios de In Touch se ha utilizado botoneras y cuadros.

4.2.2. Enlaces de animación

Los efectos de animación son obtenidos por la definición de enlaces de animación para un objeto o símbolo seleccionado.

In Touch soporta dos tipos básicos de enlaces: enlaces de entrada y enlaces de salida o de visualización.

Es importante comprender que un objeto o símbolo puede tener múltiples enlaces, la habilidad para combinar los enlaces de animación permiten crear en la pantalla efectos imaginables. Los objetos pueden cambiar de color, tamaño, localización, visibilidad, etc. Tal Como se lo requiera en la aplicación.

A continuación se presentan los pasos básicos que se necesitan para crear un enlace de animación.

1. - Crear y seleccionar un objeto (línea, curva, texto) al cual va a ser unido el enlace.

2. - A paso siguiente se invoca el comando **Special/Animation/links** o se hace doble clic sobre el objeto para acceder a la caja de diálogo de selección de enlace.

3. - Se selecciona los enlaces deseados para el objeto haciendo clic sobre el nombre del enlace. Haciendo clic sobre el botón del nombre del enlace se selecciona el enlace y se produce la aparición de la caja de dialogo para definir los detalles específicos para el tipo de enlace seleccionado.

4. - ingresar los detalles para la definición del enlace y clic sobre **OK** para regresar a la caja de dialogo de selección de enlace.

Los pasos presentados son aplicados a todos los objetos simples y compuestos que conforman la pantalla de visualización del proceso.

Las elipses que representan los recipientes tienen dos enlaces de salidas, enlaces de visibilidad y enlace de llenado con color. El enlace de visibilidad maneja las 24 elipses y son controladas por una expresión en la que es comparada una

variable **n** y la cual es incrementada cíclicamente en la aplicación Script. A medida que se incrementa **n** en cinco unidades se hace valida una comparación y se hace visible un recipiente y desaparece otro de tal manera que esto simula el movimiento de los recipientes de una posición a otra.

El enlace de llenado con color de los recipientes es obtenido también por la comparación de la variable **n**. Las elipses y cuadros que visualizan el llenado con las sustancias son controladas por las variables "Refpolvo" y "Reflíquido" en la aplicación Scripts. "Refpolvo" controla el llenado de la sustancia sólida y "Reflíquido" controla el llenado con la sustancia líquida.

Las válvulas que controlan la salida de las sustancias tienen enlaces de visualización, la cual cambia de color cuando cambian del estado abierto al cerrado, la variable que maneja el cambio de color de la válvula es controlada por el Tag de memoria analógico "líquido".

El gráfico de las tuberías que conducen las sustancias, de los recipientes grandes a los recipientes pequeños tienen el enlace de visualización, particularmente el enlace de llenado y es controlada por el Tag de memoria analógico "liquido".

El eje que se desplaza con el batidor hacia el recipiente tiene los enlaces de localización y es controlada por el Tag analógico "embolo" en la aplicación Scripts.

El batidor que simula la mezcla de las sustancias en el recipiente del tambor tiene los enlaces de localización, de tamaño y el enlace misceláneo de blinqueo. El enlace de localización que hace desplazar el batidor hacia el recipiente es controlado por el Tag analógico "émbolo" en la aplicación script. El tamaño es controlado por el Tag "paleta" y el blinqueo es controlado en función de la comparación con la variable "delay". La botonera que controla el inicio y el final del

proceso esta enlazada con el Tag de memoria discreto "inicio".

Los botones de entrada que permiten abrir y cerrar la compuerta principal están controlados por los tags analógicos "abrir_la" y "cerrar_la" respectivamente.

4.3. APLICACIÓN SCRIPTS

La aplicación Scripts sirve para enlazar aplicaciones, para activar otras aplicaciones, crear simulaciones de procesos, cálculo de variables, etc. Para crear una aplicación Scripts se debe invocar el comando **Special/Scripts/Application Scripts.**

Los Scripts en In Touch permiten ejecutar comandos y operaciones lógicas basadas en criterios específicos.

La sintaxis usada en Scripts y en las expresiones de cajas de diálogos es similar a la sintaxis algebraica de las calculadoras.

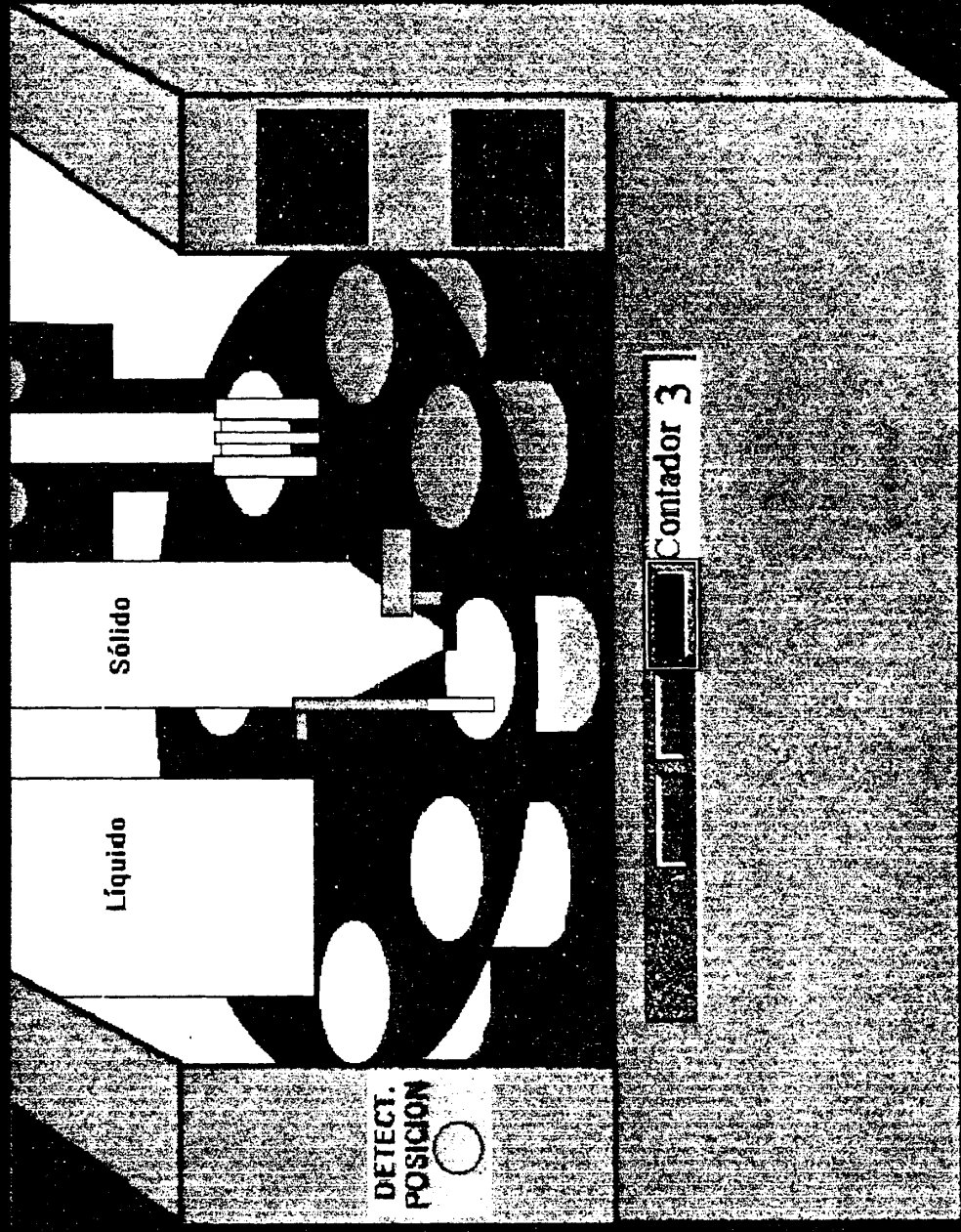
El Scripts realizado para simular el proceso de mezcla consta de varias estructuras, estas estructuras son conocidas en los lenguajes de programación de computadoras. El programa se basa principalmente en la estructura de comparación IF-THEN-ELSE. El programa esta seccionado en varias partes a manera de cuerpo de instrucciones. El programa editado en la Aplicación Scripts es ejecutado cíclicamente por un periodo de 1msec.

A continuación se presenta la Aplicación Script y en la figura 4.1. se puede apreciar el WindowsView de In Touch.

```
IF (abrir_la==1) THEN
  cerrar_la=0;
  ventana=ventana+2;
  IF ventana>=100 THEN
    ventana=100;
    abrir_la=0;
  ENDIF;
ENDIF;
IF( cerrar_la==1) THEN
  abrir_la=0;
  ventana=ventana - 2;
  IF ventana<=0 THEN
    ventana=0;
    CERRAR_LA=0;
  ENDIF;
ENDIF;
IF( (star==1) AND (contador<=9)) THEN
  IF ((contador==9) AND (ciclo==5)) THEN
    n=0;
    star=0;
  ENDIF;
  IF (ref==0) THEN
    n=n+1;
    IF((n>=20)OR((n==1)AND(contador==0))) THEN
      Refpolvo=0;
      refliquido=0;
      contador=contador+1;
      IF contador==9 THEN
        ciclo=6;
        ref=1;
      ELSE
        ciclo=1;
        ref=1;
      ENDIF;
    ENDIF;
  ENDIF;
  IF ciclo==1 THEN
    polvo=polvo+2;
    refpolvo=refpolvo+2;
    IF polvo==30 THEN
      polvo=0;
      ciclo=2;
    ENDIF;
  ENDIF;
  IF ciclo==2 THEN
    liquido=liquido+2;
    refliquido=refliquido+2;
```

```
IF liquido>=30 THEN
    liquido=0;
    ciclo=6;
ENDIF;
ENDIF;
IF ciclo==6 THEN
    IF contador==1 THEN
        ref=0;
        ciclo=5;
        n=0;
    ENDIF;
ENDIF;
IF ((contador>=2) AND (ciclo==6)) THEN
    ref=1;
    embolo=embolo+5;
    IF embolo>=80 THEN
        paleta=paleta+9;
    ENDIF;
    IF embolo>=125 THEN
        embolo=125;
        ciclo=3;
    ENDIF;
ENDIF;
IF ciclo==3 THEN
    ref=1;
    delay=delay+1;
    IF delay==100 THEN
        ciclo=4;
        delay=0;
    ENDIF;
ENDIF;
IF ciclo==4 THEN
    ref=1;
    embolo=embolo - 5;
    IF embolo>=80 THEN
        paleta=paleta - 9;
    ELSE
        paleta=0;
    ENDIF;
    IF embolo<=1 THEN
        embolo=0;
        n=0;
        ref=0;
        ciclo=5;
    ENDIF;
ENDIF;
ENDIF;
```

In Touch - Window Viewer



DETECT.
POSICION

Contador 3

4.4. COMUNICACIÓN DE IN TOUCH CON LA CPU 212

In Touch usa el Intercambio Dinámico de Datos (DDE) para comunicarse con otros programas de Windows y con servidores DDE para conectarse con dispositivos externos, tales como el PLC. DDE es un protocolo que requiere tres piezas de información para enlazarse y transferir datos hacia otros programas. Estas piezas de información son: el nombre de la aplicación, el nombre del tópico y el nombre del ítem.

In Touch para adquirir datos de otras aplicaciones, debe conocer el nombre de la aplicación que provee los datos, el nombre del tópico dentro de la aplicación que contiene los datos, y el nombre del ítem específico dentro del tópico. Adicionalmente In Touch necesita conocer los tipos de datos que se van a adquirir, los cuales pueden ser: discreto, entero, real (punto flotante), o datos tipo mensaje (cadenas de texto). Toda esta información determina el tipo de Tag DDE a ser definido en la base de datos de In Touch.

Cuando se ejecuta **WindowViewer** (ejecución o corrida de la pantalla de In Touch), esta realiza automáticamente todas las acciones requeridas para comunicarse con el PLC, y poder adquirir y mantener los valores de sus Tags DDE definidos previamente.

La computadora necesita de un programa para reconocer al PLC y poder interactuar con él. Es decir, el programa que habilita el uso del PLC con la computadora. Uno de los programas utilizados y conocido es el **KEPSERVER**, el cual permite la comunicación entre el PLC y el **WindowsViewWer** de In Touch.

En el programa **KEPSERVER** se crea un canal de comunicación para enlazarse con In Touch, y otro canal para enlazarse con el PLC. Al crearse el canal de enlace con el PLC se deben especificar el puerto de comunicación serial al cual va conectado el PLC, la velocidad de transferencia de los datos, el formato de los datos, etc. En el **KEPSERVER** se pueden crear varios canales de comunicación dependiendo de los puertos y de las interrupciones disponibles en la computadora.

Cuando In Touch requiere datos del PLC, se comunica con el servidor **KEPDDE** especificando las tres piezas de información, las cuales son: el nombre de la aplicación que en este caso sería **KEPDDE**, el nombre del tópicos el cual es creado al momento de crearse un canal de comunicación en el **KEPSERVER** y que puede ser un nombre cualquiera, y por último se debe especificar el ítem, que no es otra cosa que una localidad de memoria del PLC.

Una vez especificada la ruta de la aplicación que contiene los datos, In Touch está listo para leer y escribir valores en el PLC.

CONCLUSIONES

- 1.- El presente trabajo me ha permitido combinar dos aplicaciones con la electrónica básica estas son la programación de la CPU 212 de Siemens y el software de visualización de procesos "In-Touch", relacionando esto al Ingeniero Eléctrico y el control de los procesos industriales con las computadoras.
- 2.- El control de los procesos Industriales se hace más dependiente de las computadoras y de los avances tecnológicos, dejando atrás la electrónica clásica y el cableado fijo.
- 3.- El sistema diseñado combina la programación de la CPU y el Software de visualización de procesos, la aplicación es bastante similar a la utilizada en la elaboración de helados o en los diversos procesos químicos.
- 4.- El principal inconveniente del diseño fué la falta de pruebas con la CPU 212 de Siemens, por no estar la misma al alcance económico personal ni de la FIEC.

RECOMENDACIONES

- 1.- Es casi obligación de la **FIEC** para la formación de los futuros Ingenieros Eléctricos Industriales la adquisición y actualización de materiales de laboratorio, como son estos los PLCs y las computadoras, los cuales son necesarios para el desarrollo de las tesis y de los tópicos de graduación.
- 2.- Es necesario un nuevo estructuramiento del flujograma de materias de la carrera, de tal manera que se enfoquen de cerca los avances tecnológicos relacionados con el control de los procesos industriales y de esta manera y con la debida preparación de los profesores obtener verdaderos profesionales en el área Industrial.
- 3.- Debe existir mayor apoyo tanto logístico como económico para los estudiantes que se encuentran en la etapa de graduación, para así de esta manera conseguir el desarrollo de verdaderos proyectos y que estén a la altura de las necesidades del medio.

APENDICE

APENDICE A

SOFTWARE DE PROGRAMACION DE LAS CPUs S7-200

A.1. INTRODUCCIÓN

Desde el punto de vista de un programador, la mejor forma de comprender la operación de un procesador es aprender y conocer el conjunto de instrucciones de máquina. La CPU puede realizar una variedad de funciones, y esas son reflejadas en la variedad de instrucciones definidas para la CPU. La colección de diferentes instrucciones que la CPU puede ejecutar es referida como el conjunto de instrucciones de la CPU.

"STEP 7-MicroWin" es un programa que nos permite editar y compilar los programas para las CPUs S7-200 de Siemens, nos permite obtener información acerca del tipo de CPU que tenemos conectados al puerto de la computadora, nos permite compilar y depurar los programas editados y tiene dos formas o lenguajes de programación, etc.

A.2. LENGUAJES DE PROGRAMACION

Las CPUs S7-200 se pueden programar de dos maneras, con el lenguaje Esquema de contactos (KOP) y con la Lista de instrucciones (AWL).

PROGRAMAS KOP

En los programas KOP, los elementos básicos se representan con contactos, bobinas y cuadros. Una hilera de elementos interconectados que constituyen un circuito completo se denomina un segmento.

Una bobina es un símbolo que representa una salida cableada, cuando la corriente fluye por la bobina, la salida se activa.

Un cuadro es un símbolo que representa una operación compleja la misma que es ejecutada en la CPU. El cuadro simplifica la programación de dicha operación. Por ejemplo, los temporizadores, los contadores y todas las operaciones aritméticas se representan mediante cuadros.

PROGRAMAS AWL

Los elementos de programas AWL se representan mediante instrucciones que ejecutan las operaciones deseadas. Contrariamente a los programas KOP que se visualizan de forma gráfica, los programas AWL se representan en formato de texto.

A.3. EJECUCION DEL PROGRAMA EDITADO

Una vez cargado el programa en la CPU, se ejecuta de la primera hasta la última operación en un ciclo que se repite permanentemente. Un ciclo de la CPU comprende la lectura de las entradas, la ejecución del programa de usuario, la ejecución de peticiones de comunicación, la ejecución de tareas internas y la escritura en las salidas.

Los tipos básicos de datos con los que trabaja la CPU son datos tipo bit, byte, palabra, doble palabra y reales.

Las instrucciones de control de flujo de **"STEP 7-MicroWin"** son algunas de las estructuras de los

lenguajes de programación conocidos, como el lenguaje C, Basic, etc. Entre esas estructuras tenemos: el lazo For-Next, las llamadas a subrutinas, los saltos del puntero de instrucción, y las solicitudes de interrupción.

"STEP 7-MicroWin" a sido pensado para ser altamente comprendido, ya que es un programa fácil de usar pero al igual que otros programas tienen sus desventajas.

La excesiva libertad en la edición de los programas puede llevar a errores de programación que, por ser correctos sintácticamente no se detectan a simple vista.

A.4. REALIZACION DE UN PROGRAMA

En este apartado se van a exponer los pasos a seguir en la realización de un programa, como en cualquier lenguaje de programación de computadoras, hay que seguir los siguientes pasos:

- Comprender el problema o el proceso que se desea desarrollar.

- Identificar las entradas y salidas del proceso.
- Realizar diagramas de flujo del proceso.
- Codificar las instrucciones
- Depurar el programa
- Pruebas finales

A.5. COMPILACION DEL PROGRAMA

Con la compilación podemos detectar los errores de sintaxis, debido a que el programa escrito no se adapta a la sintaxis y reglas del compilador, estos errores se deben ir corrigiendo hasta obtener una compilación sin errores, con la compilación del programa también podemos darnos cuenta de la cantidad de memoria que ocupa el programa editado. Para compilar un programa tenemos que escoger en el menú principal de STEP 7-MicroWin la opción CPU, y después la opción compilar.

Al ejecutar el programa, pueden ocurrir errores durante la ejecución. Por ejemplo, puede darse una división por cero, estos errores son detectados por la misma CPU con la activación de marcas especiales y que

pueden ser utilizadas en el programa como condicionantes.

A.6. DIVISION DE MEMORIA DE LAS CPUs S7 200.

La memoria de las CPUs S7-200 está dividida en tres áreas: memoria de programa, memoria de datos y memoria de parámetros configurables. Dichas áreas están definidas conforme a su utilización.

En la memoria de programa se encuentra almacenado el programa de usuario.

La memoria de datos incluye un área de trabajo para el programa y un área para almacenar objetos de datos, contiene cálculos, memoria temporal y constantes. Además, allí se almacenan también objetos tales como temporizadores, contadores, contadores rápidos, así como entradas y salidas analógicas.

A.6.1. MEMORIA DE DATOS

La memoria de datos contiene la memoria de variables, la imagen del proceso de las entradas, la imagen de proceso de las salidas, marcas internas y marcas especiales.

A.6.2. MEMORIA DE PARAMETROS

La memoria de parámetros provee espacio de memoria para almacenar los parámetros configurables tales como contraseñas, direcciones de estaciones y áreas remanentes. El contenido de la memoria de parámetros se almacena en la memoria no volátil.

A.6.3. MEMORIA DE PROGRAMA

La memoria de programa contiene la lista de operaciones que ejecuta el PLC para implementar la función de control deseada. La memoria de programa abarca 512 palabras en la CPU 212.

A.6.4. MEMORIA DE VARIABLES

La memoria de variables (memoria V) es una memoria de Lectura/Escritura que se encuentra en la memoria RAM.

A.7. EDICION DE LOS PROGRAMAS DE USUARIO EN KOP

Si se desea introducir un segmento en el editor KOP, utilice la barra de herramientas KOP para insertar las operaciones del segmento.

1. Sitúe el cursor en el segmento que desee editar.
2. Seleccione la operación que desee insertar.
3. Inserte la operación seleccionada pulsando la tecla de introducción o con un doble clic del ratón.
4. Introduzca los parámetros de la operación. Utilice la barra espaciadora o el ratón para desplazarse por los campos.
5. Sitúe el cursor en el segmento siguiente.

APENDICE B

BREVE ESTUDIO DEL SOFTWARE DE VISUALIZACION DE PROCESOS INDUSTRIALES "IN TOUCH"

B.1. INTRODUCCION

In Touch, es una de las herramientas más potentes y flexibles de desarrollo de interfaces de operador para la creación de sistemas personalizados en entornos de fabricación discretos, de proceso, DSC, SCADA y otros.

Permite a ingenieros, supervisores, administradores y operadores, ver en pantalla mediante representaciones gráficas de procesos en tiempo real, los trabajos de una operación completa. Este software incluye varias funciones nuevas de arquitectura distribuida entre las que se encuentran la gestión de alarmas distribuidas, datos históricos distribuidos, conversión de resolución dinámica y desarrollo, y mantenimiento remoto de aplicaciones para su uso en grandes redes basadas en PCs.

B.2. ANIMACION Y ENLACES DE GRAFICOS

Una vez que un objeto gráfico o símbolo a sido creado puede ser animado por medio de enlaces de animación. Los enlaces de animación hacen que los objetos o símbolos cambien en apariencia al reflejar cambios en el valor de una variable conocida como "tagname", o en una expresión donde se usen varios tagnames. Por ejemplo, un gráfico de una bomba puede ser de color rojo cuando está apagado y de color verde cuando está encendida. El gráfico de la bomba puede ser también sensitiva, esto quiere decir que se puede hacer clic sobre el gráfico como si se tratase de una botonera, es decir, puede prenderse o apagarse la bomba simplemente oprimiendo el gráfico de la bomba. Esos y muchos otros efectos de animación son obtenidos definiendo enlaces de animación para un objeto seleccionado o símbolo.

In Touch soporta dos tipos básicos de enlaces: enlaces de entrada y enlaces de visualización. Los deslizadores o botoneras son ejemplos de enlaces de entrada. los colores de llenado de las líneas o los

enlaces de blinqueo son ejemplos de enlaces de visualización

B.2.1. CREANDO ENLACES DE ANIMACION

Si se quiere empezar rápidamente, estos son los pasos básicos que se necesitan para crear un enlace de animación:

1. Crear y seleccionar un objeto (línea, curva, texto, botón o símbolo) al cual el enlace es unido.
2. Invoque los comandos **Special/Animation Links** o doble clic sobre el objeto para acceder a la caja de diálogo de selección de enlace.
3. Seleccione los enlaces deseados para el objeto haciendo clic sobre el botón con el nombre del enlace.
4. Ingrese los detalles para la definición del enlace y haga clic sobre un **OK** para regresar a la caja de diálogo de Selección del Enlace.

Una vez que usted ha ingresado toda la información requerida, haga clic sobre el botón

Done en la parte superior de la caja de diálogo.

El tagname que se usa para el objeto debe ser definido en la base de datos antes que un enlace pueda ser unida a él. Si no está definido, se puede definir una vez que el botón **OK** es presionado.

B.2.2. ENLACES DE ENTRADA

El enlace de entrada permite que algún objeto o símbolo sea una especie de botón sensitivo y que le sirva de entradas al operador. Los enlaces de entrada son identificados en **Runtime** por el "**Touch Frame**" que hace visible la parte que rodea al objeto. Una botonera sensitiva puede ser activada haciendo clic sobre ella con el mouse, tocando la imagen en la pantalla o presionando una tecla equivalente asignada o presionando la tecla **Enter**.

Los enlaces de entrada permiten al operador tener acceso a las entradas del sistema. Estas pueden ser por ejemplo: encender o apagar una válvula,

entrar una nueva referencia de alarmas, correr un complejo login Scripts o acceder en la red ingresando una cadena de textos. Si el objeto o símbolo enlazado contiene objetos de texto el cual es ubicado encima de otro, el tope del objeto texto sería usado para visualizar el valor del dato.

B.3. CONJUNTO BASICO DE INSTRUCCIONES Y FUNCIONES PARA LA EDICION DE LOGIN SCRIPTS

B.3.1. INTRODUCCION A LOS SCRIPTS

Todos los Scripts son manejadores de eventos, los eventos pueden ser un cambio de datos, cambio de condición, clic del mouse, temporizador, etc. El orden de procesamiento es específico a la aplicación

B.3.2. APLICACIÓN SCRIPTS

Las aplicaciones Scripts son enlazadas a aplicaciones enteras y pueden ser usadas para

servir a otras aplicaciones, crear simulación de procesos, calcular variables, etc. Para crear una aplicación script, se debe invocar los siguientes comandos **Special/Scripts/Application**

B.4. SERVIDOR DDE

B.4.1. INTRODUCCION A DDE Y SERVIDORES DDE

In Touch usa el intercambio dinámico de datos (DDE) como un protocolo para comunicarse con otros programas de Windows y con servidores tipo DDE para conectarse con el mundo real. DDE es un protocolo que requiere tres piezas de información para enlazarse satisfactoriamente y transferir datos hacia otros programas. Estas piezas de información son el nombre de la aplicación, el nombre del tópico y el nombre del ítem.

DDE es un protocolo de comunicación diseñado por Microsoft y que permite a las aplicaciones de Windows enviar y recibir datos e instrucciones hacia y desde otras aplicaciones. El servidor de

aplicaciones provee los datos .y acepta requerimientos de otras aplicaciones interesadas en esos datos. Las aplicaciones requeridas son llamadas clientes. Algunas aplicaciones (tales como In Touch y Excel) pueden simultáneamente ser ambos cliente y servidor.

In Touch provee la habilidad para definir bases de datos tipo DDE (tagnames) y que pueden ser adquiridos continuamente desde otras aplicaciones de Windows vía DDE.

Las aplicaciones que proveen los datos a In Touch deben también soportar el protocolo DDE. Una aplicación popular que soporta DDE es el programa Microsoft Excel. In Touch puede leer y escribir valores de Excel, también es posible para Excel leer y escribir valores en la base de datos de In Touch. Este intercambio ocurre en tiempo real, con ambas aplicaciones corriendo simultáneamente. Los valores de los datos adquiridos remotamente son actualizados automáticamente siempre que el valor del ítem cambie en la fuente. Esta

capacidad puede ser usada para configurar una aplicación consistiendo de la interacción de dos o más aplicaciones.

El protocolo normal DDE identifica un elemento de datos usando una convención de nombre con tres partes que incluyen el nombre de la aplicación, el nombre del tópico y el nombre del ítem. Para obtener datos de otras aplicaciones, el programa cliente abre un canal al servidor de la aplicación siempre y cuando se especifiquen estos tres ítems.

In Touch para adquirir datos de otras aplicaciones, debe también conocer el nombre de la aplicación que provee los datos, el nombre del tópico dentro de la aplicación que contiene los datos, y el nombre del ítem específico dentro del tópico. Adicionalmente In Touch necesita conocer los tipos de datos que se van a adquirir como lo son: discreto, entero, real (punto flotante), o datos tipo mensajes(cadenas de texto).

Esta información determina el tipo de DDE para el tagname cuando este es definido en la base de datos de In Touch. Ahora, cuando **WindowsViewer** está ejecutándose, este automáticamente realiza todas las acciones requeridas para adquirir y mantener los valores de estos ítems.

Por ejemplo, en el caso de Excel, el nombre de la aplicación es Excel, el nombre del tópico es el nombre del archivo específico que contiene los datos, y el nombre del ítem es la celda hacia y desde el cual los datos son leídos y escritos.

In Touch puede recibir datos de otras aplicaciones de Windows creando los ítems DDE en el diccionario de tagnames.

Para poder definir un tagname tipo DDE para el ítem en el diccionario de datos se debe hacer lo siguiente:

1. Se escoge el comando **Special/Tagname Dictionary**, con lo cual aparecería una caja de diálogo.

Si esta es la primera vez que la base de datos ha sido accesada, la definición para el sistema interno del tagname \$AccessLevel sería visualizado. Una vez que un tagname ha sido definido y el diccionario de datos es reacesado, los datos sobre el archivo para el tagname previamente accesado automáticamente se visualizaría cuando la caja de diálogo aparece.

2. Clic sobre el botón **New** para definir un nuevo tagname
3. Ingrese el tagname a ser definido para el ítem DDE en la base de datos.

Entonces, se debe hacer clic sobre el botón **Type** para seleccionar el tipo DDE para el tagname y con lo cual aparece la caja de diálogo con todos los tipos para escoger.

B.8. COMUNICACIÓN ENTRE IN TOUCH Y EL SERVIDOR DDE

Todas las entradas o salidas y controladores de PLC usados por In Touch son aplicaciones separadas de

Windows llamadas "servidores DDE." Esos programas responden a requerimientos DDE hechos por otras aplicaciones. Esas aplicaciones son conocidas como clientes. Cuando WindowViewer requiere el status de un ítem DDE, este abre un canal con el servidor DDE e informa a WindowViewer cuando el ítem DDE cambia.

El servidor DDE automáticamente encabeza todas los mensajes a y del PLC. La aplicación cliente simplemente le dice al servidor DDE que registre, el número o el punto de entrada o salida a leer o escribir. El servidor DDE entonces automáticamente actualizaría al cliente.

BIBLIOGRAFIA

1. PROGRAMABLE CONTROLLERS, operation and application, New York, Prentice Hall, first edition 1988
2. THEORY AND APPLICATIONS OF STEP MOTORS.
AUTOR: BENJAMIN C. KUO
3. WONDERWARE CORPORATION "WONDERWARE'S IN TOUCH BASIC TRAINING COURSE MANUAL", 1996, 10-1
4. SIEMENS AG " SISTEMA DE AUTOMATIZACION, 1997, P A-8.
5. MICROPROCESADORES, DISEÑO Y APLICACIONES EN LA INDUSTRIA Y EN LOS MICROCOMPUTADORES.
AUTOR: JOSE M. ANGULO