

156
2ej.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERÍA

**PROGRAMACIÓN DE UN ROBOT INDUSTRIAL
DE 6 EJES PARA LA APLICACIÓN
DE SELLADOR EN EL PARABRISAS DE UNA
CAMIONETA PICK UP**

**TESIS PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO ELECTRICISTA**

**P R E S E N T A
RAÚL SÁNCHEZ SOTO**

DIRECTOR: INGENIERO EDUARDO RAMÍREZ SÁNCHEZ

CIUDAD UNIVERSITARIA D.F.

DICIEMBRE 1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

*...y estaba soñando con un mundo gobernado
por androides y desperté y me lleno de
gozo al saber que ninguna máquina
podrá razonar ni pensar de la
forma como lo hace el
ser humano.*

A mis padres :

*por el mejor legado que pudieron darme
por los preceptos y creencias
por mi vida
gracias.*

A mis hermanos :

*Por todo el apoyo que me dieron durante
mi formación profesional.*

A Ernesto, Luis y Mauricio.

*Que con su apoyo fue posible
la realización de esta tesis .*

A mis compañeros y amigos.

*A quienes me impulsaron a terminar
gracias.*

*A:
Raquel mi esposa y
Oscar y Diana mis hijos.*

*Por todo el tiempo que estuve separado de
ellos mientras realizaba esta tesis.*

Por su comprensión y cariño.

Por que los amo.

ÍNDICE

INTRODUCCIÓN

1.- DESCRIPCIÓN GENERAL DEL SISTEMA

- 1.1 Área de trabajo
- 1.2 Sistema neumático
- 1.3 Sistema eléctrico
- 1.4 Sistema de control
- 1.5 Área de aplicación

2.- EL CONTROLADOR LÓGICO PROGRAMABLE (PLC)

- 2.1 Hardware
- 2.2 Software

3.- EL CONTROLADOR DEL ROBOT

- 3.1 Sistema de control
- 3.2 Los controles
- 3.3 Movimientos del Robot
- 3.4 Operación
- 3.5 El menú de instrucciones

4.- EL PANEL DE CONTROL

- 4.1 Entradas
- 4.2 Salidas
- 4.3 Interfase con el robot
- 4.4 Secuencia de el programa de trabajo

5.- PROGRAMACIÓN DEL ROBOT

- 5.1 Diseño y descripción del Programa Administrador
- 5.2 Diseño y descripción del Programa de Aplicación

CONCLUSIONES

BIBLIOGRAFÍA

INTRODUCCIÓN

Actualmente la industria automotriz ha tenido un crecimiento acelerado debido a la constante demanda de unidades por parte de los consumidores. Aunado a ésta se exige también una mejor calidad en el producto, ya que al aumentar la cantidad de producción se corre el riesgo de perder calidad, que actualmente es una de las armas más poderosas que utilizan los competidores de este ramo.

Como solución a estos problemas la industria automotriz ha tenido que hacer uso de los avances tecnológicos. Así, de esta manera, es muy común encontrarse con controladores automatizados que tienen el mando de procesos tan sofisticados como pueden ser: la aplicación de color primario, color base y barniz a las carrocerías utilizando sistemas automatizados que controlan el flujo y la calidad de aplicación, sistemas automáticos utilizados para el inflado de llantas manteniendo en estas una presión constante y con un margen de +/- 5% del valor estándar, alineación y balanceo computarizado, control automatizado de temperatura a cassetas de horneado de pintura, aplicación robotizada de primer a las carrocerías, ensamble de carrocerías utilizando un sistema robotizado, etc.

Con todos estos procesos se puede garantizar un mayor volumen de producción de unidades cuidando la calidad y además reducir los costos de producción.

Todo esto repercute en la integración de nuevos equipos que faciliten las operaciones que intervienen en la producción en serie de unidades automotrices. En otros casos es necesario hacer uso de los que ya existen.

Para algunas personas la palabra robot todavía sigue evocando a un androide con habilidades entre técnicas y mágicas. También suele representarse como un ente peligroso y malvado, producto de la ciencia ficción. Para la gente de la industria, en cambio, un robot es un artefacto familiar que le apoya en los trabajos más sofisticados.

La instalación de robots no debe verse como una forma de sustitución de mano de obra, sino de aumentar la calidad y volverse más competitivos en los mercados mundiales y ninguno como la industria automotriz demanda tantos y tan disímiles productos.

Comunmente el trabajo manual registra una serie de variaciones que pueden ir desde modestas hasta considerables, según sea el caso. Si un ser humano realiza la aplicación de sellador a parabrisas, la calidad del trabajo necesariamente será irregular, de buena a mediocre. Esta variación, que cualquier persona razonable considerará aceptable, tiene sin cuidado al anónimo comprador de un automóvil, a quien lo que le importa es que el producto terminado cumpla con sus expectativas. Para el caso de la aplicación de sellador, el robot repite exactamente la trayectoria de acuerdo a su programa, asegurando la calidad, precisión y uniformidad del material.

Los modernos robots industriales trabajan con una precisión de una décima de milímetro, otro aspecto llamativo es la facilidad operativa y la "capacidad de aprendizaje" lograda por los controles electrónicos y el software, esto quiere decir, para crear una nueva rutina de trabajo, el programador sólo debe mostrar al sistema central de cómputo a qué puntos en el espacio quiere que se desplace el robot y el resto correrá por cuenta del sistema computarizado.

El objetivo de esta tesis es programar un robot industrial que inicialmente fue diseñado para aplicar líquido primario a parabrisas, para que realice la aplicación de sellador a los parabrisas de camiones modelo T-300 1997.

CAPITULO

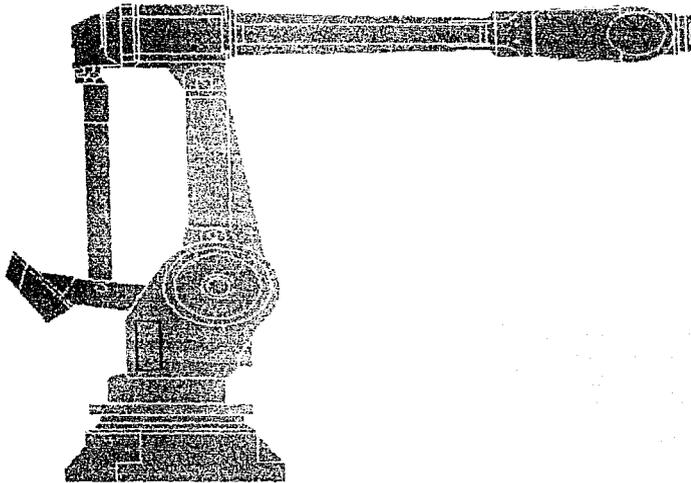
1

1.- DESCRIPCIÓN GENERAL DEL SISTEMA

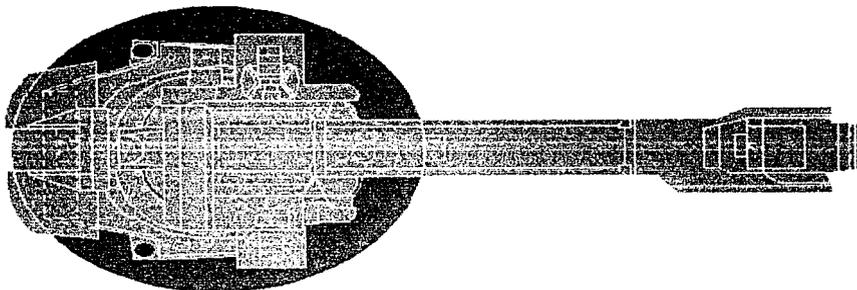
1.1 ÁREA DE TRABAJO

El sistema para la aplicación de sellador (uretano) a parabrisas es un sistema modular que consta de las siguientes partes fundamentales:

- **El cuerpo del robot.**- Consiste de un brazo que tiene 5 articulaciones, los cuales le permiten tener una área de trabajo extensa que es de aproximadamente 30 Mts. cuadrados, teniendo cada una de sus articulaciones movimientos circulares que van desde 70° hasta 300°. Colocado en uno de sus extremos se encuentra la herramienta que aplicará el sellador a los parabrisas. Véase figura 1.1 y figura 1.1.1.

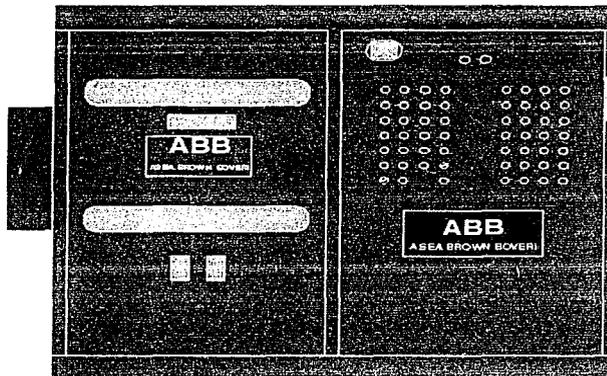


1.1. VISTA LATERAL DEL BRAZO ROBOT



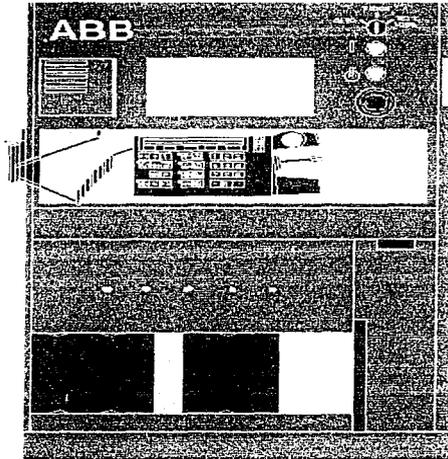
1.1.1. VISTA SUPERIOR DEL BRAZO ROBOT

- **Panel de control.**- Es un gabinete dentro del cual se encuentra el sistema de relevadores para controlar la etapa de potencia, la fuente de alimentación de c.c. para los dispositivos de entradas y salidas, y el rack o soporte que contiene el Controlador Lógico Programable (PLC por sus siglas en inglés) los módulos de entradas y salidas, y la fuente de alimentación para el rack y los módulos de entradas y salidas. En la parte externa del gabinete, se tienen los botones pulsadores de arranque y paro de proceso así mismo los selectores de ciclo de proceso manual o automático, todas estas son señales que están conectadas a los módulos de entradas del sistema controlador. Véase figura 1.1.2.



1.1.2. PANEL DE CONTROL PRINCIPAL

- **Controlador del robot.**- Es un gabinete diseñado para soportar en su interior todo el sistema de control para el brazo robot, mencionando entre otros los servoamplificadores, la computadora central del sistema, la fuente de alimentación, etc., además de los dispositivos de control de entradas y salidas que interactúan con el sistema del controlador lógico. En la parte exterior del panel, se tienen los botones de control de encendido y apagado, la unidad de programación, la unidad de disco flexible, el monitor que nos sirve para visualizar las instrucciones y comandos que se están realizando dentro del sistema de control. Véase figura 1.1.3.



1.1.3. PANEL DE CONTROL DEL ROBOT

1.2 EL SISTEMA NEUMÁTICO

Para abastecer del fluido neumático al sistema, se cuenta con una toma de alimentación de la línea de distribución de la planta. La presión de aire adecuada para lograr un buen funcionamiento del sistema es de 70 PSI., el aire de alimentación se hace pasar a través de una válvula de globo, posteriormente se tiene una válvula de desfogue rápido la cual cuenta con un silenciador por donde se hace pasar el aire despresurizado, a continuación se cuenta con un filtro neumático para eliminar el agua que pueda contener el aire ocasionado por la condensación que se lleva a cabo dentro de los compresores, a continuación se tiene un regulador para mantener una presión constante en el sistema, bajo estas condiciones se reparte el aire a toda el área de trabajo.

El sistema neumático esta formado de nueve cilindros¹, cada cilindro realiza una función específica dentro del sistema, de esta manera el primer cilindro en la secuencia de trabajo, le da un giro de 70 grados a la herramienta (cargador de cristales) encargada de colocar los cristales en la mesa de trabajo , forzando a que el cristal quede ubicado sobre ocho soportes distribuidos en el área, los siguientes dos cilindros colocados sobre el cargador le van a dar una posición al cristal obligándolo a que éste se localice dentro de una área determinada, otros dos cilindros colocados a 90 grados uno respecto del otro se van a encargar de centrar el cristal en el área de trabajo. Los últimos cuatro cilindros se utilizan para desplazar los dispositivos encargados en detener el cristal a los soportes (copas de vacío) . La velocidad y sentido de desplazamiento de cada cilindro esta comandado por dos reguladores de aire y de una válvula de dos vías y dos posiciones.

1.3 EL SISTEMA ELÉCTRICO

La alimentación del sistema es tomada de la red general de la planta a un voltaje de 440 V de a.c. mismo que es conectado a dos transformadores, uno de 3KVA y otro de 1.5KVA. El primero reduce el voltaje de línea de 440 V a.c. a 110 V. a.c. . A esta línea se conectan las fuentes de alimentación para cada uno de los rack's del PLC², además de el aire acondicionado y las lámparas de iluminación del interior de la consola de control. El segundo transformador reduce su voltaje de 440 V. a. c. a 110 V. a. c. conectando a esta línea una fuente de alimentación de 24 V. c.c. que alimenta a todos los relevadores que controlan los sistemas de seguridad del sistema, entradas y salidas del sistema de control del robot, al sistema de vacío utilizado para las ventosas que sostienen el cristal , y a los módulos de entradas y salidas del PLC.

1.4 EL SISTEMA DE CONTROL

El sistema de control, esta diseñado a base de un controlador lógico programable que es quien se encarga de dar la secuencia lógica al proceso. Conectado a él, están los módulos de entradas y salidas , quiénes son los encargados de recibir las señales de control que llegan desde los diferentes sensores del sistema y enviar las señales de habilitación a todos los dispositivos a controlar como son solenoides, relevadores, lámparas indicadoras , etc. Además de los módulos de entradas y salidas , se conectan también el PLC las líneas de comunicación con los controladores de cada robot, la comunicación entre el PLC y los controladores de los robots se lleva a cabo mediante un puerto serie y línea de comunicación RS232.

¹CONJUNTO EMBOLO PISTON

²Controlador Logico Programable
POR SUS SIGLAS EN INGLES

1.5 EL ÁREA DE APLICACIÓN DEL SELLADOR

El área donde se realiza la aplicación del sellador, consiste de dos partes fundamentales, el cargador de/o posicionador de cristales y la mesa de aplicación :

El posicionador de cristales:

Sostenido de dos soportes verticales se encuentra una estructura giratoria, en la parte inferior esta unida mediante articulaciones a un cilindro, el cilindro al ser desplazado provoca que la estructura gire 70 grados en forma vertical respecto a su posición anterior. Sobre esta misma estructura se han colocado dos cilindros más, los cuales en uno de sus extremos han sido colocadas dos estructuras en forma de escuadras, mismas que van a soportar el cristal cuando éste se encuentre en posición de reposo, cuando el cristal se coloca en la posición de trabajo ocasionado por el giro de el cargador, los cilindros deslizan las escuadras para que den paso a la siguiente secuencia.

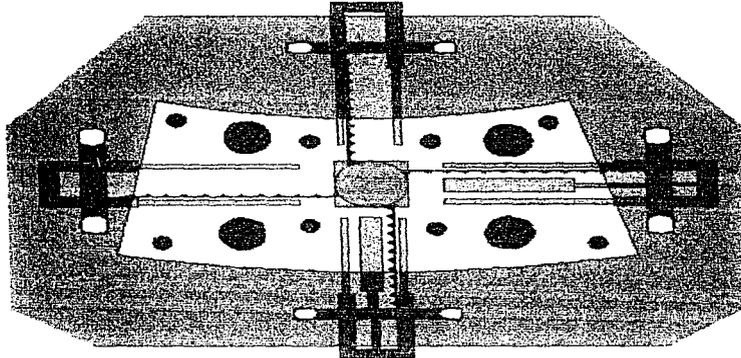


FIGURA 1.1.4 MESA DE APLICACION DE URETANO.

La mesa de aplicación:

Es el área donde es colocado el cristal para aplicarle el sellador, así de esta manera se cuenta en esta área con ocho soportes verticales donde será colocado el cristal, en el centro de la mesa de trabajo, se tiene un juego de dos engranes y cuatro cremalleras, cada engrane se une a dos cremalleras una en sentido opuesto a la otra como se muestra en la figura 1.1.4. Los juegos de engranes y cremalleras se encuentran colocados uno encima del otro y a 90 grados uno con respecto al otro de tal manera que se mueven en forma independiente. En uno de los extremos

de cada una de las cremalleras se ha colocado una estructura que se desliza sobre un par de rieles, en la parte superior de cada una de las estructuras se han colocado en forma vertical dos postes, así cuando las cremalleras se deslizan, obligan al cristal a ser colocado exactamente en el centro de la mesa de trabajo. A el juego de cremalleras, engranes y estructuras se les conoce como ecualizador, así que por la ubicación dentro de la mesa se les conoce como ecualizador horizontal y ecualizador vertical. Para mover los ecualizadores se cuenta con dos cilindros, uno para el ecualizador horizontal y uno para el ecualizador vertical .

Ubicados dentro de la mesa de trabajo, se encuentran también cuatro cilindros perpendiculares a la mesa, cada cilindro cuenta con un émbolo hueco, y en el extremo superior del émbolo se tiene colocada una ventosa, en el extremo inferior se tiene un conexión que comunica el émbolo con un dispositivo generador de vacío, de tal forma que cuando el cristal ha sido debidamente ecualizado las ventosas se adhieren a el cristal impidiéndole cualquier movimiento.

CAPITULO

2

**2.- EL CONTROLADOR LÓGICO
PROGRAMABLE (PLC)**

Este capítulo da una breve descripción del hardware y el software que rige a un PLC de la familia 5 de Allen Bradley.



FIGURA 2.1 VISTA FRONTAL DE UN PLC DE LA FAMILIA ALLEN BRADLEY

2.1 HARDWARE

El hardware para un sistema con PLC-5 consiste básicamente de los siguientes componentes:

- MÓDULO CONTROLADOR PLC-5.
- FUENTE DE PODER
- HARDWARE DE ENTRADAS Y SALIDAS.

Un módulo controlador PLC-5 es un sistema digital basado en microprocesador diseñado para aplicaciones generales. El procesador esta encapsulado en un modulo de tal manera que el fabricante solo provee al exterior las siguientes partes fundamentales :

- Indicador de estados .- El módulo procesador cuenta con LED's indicadores de estado que van ha mostrar la operación normal y condiciones de error del sistema procesador.
- Selector de modo de operación.- una llave selectora ubicada en la parte frontal del módulo procesador selecciona cada uno de los tres modos de operación :

* **REMoto.-** En modo remoto se puede seleccionar cualesquiera de los tres modos de trabajo. (programación remota, prueba remota y operación remota) utilizando la terminal de programación , el cuadro siguiente muestra una explicación de cada uno de los modos.

Modo de trabajo	Medio de selección del modo	Información del modo seleccionado
Operación remota	* Mediante la llave selectora. * A través del software de programación.	Las salidas son activadas en tiempo real. Se pueden guardar o restablecer archivos, así como editar programas en línea.
Programa remoto		Este modo funciona de la misma manera que el modo programa.
Prueba remota	Solo mediante el software de programación.	* Se pueden ejecutar los programas de escalera manteniendo las salidas desactivadas. * No se puede crear o borrar programas o archivos de datos.

* **RUN.-** El modo RUN es el modo donde se puede correr los programas, forzar las entradas y salidas, salvar los programas en un diskette y mantener las salidas de tiempo real activas.

No se pueden crear o borrar archivos de escalera, archivos de datos, o cambiar los modos de operación mediante la terminal de programación, mientras se está en modo RUN.

* **PROGRama.-** En el modo de programa las salidas son deshabilitadas y el procesador deja de monitorear el programa.

En este modo , se puede modificar o borrar programas de escalera o archivos de datos, así mismo cargarlos en el módulo de memoria EEPROM, además de salvar y cargar programas de escalera. Estando en este modo no se puede cambiar de modo de operación desde la terminal de programación.

- **Conectores del panel frontal.-** En la parte frontal del procesador se tienen tres conectores, mismos que son utilizados para la comunicación con dispositivos periféricos, a continuación se menciona la aplicación de cada conector.
 - Conector para la terminal de programación; es un conector de 9 pins carcasa-D, y esta etiquetado como PEER COMM INTFC, este es un puerto de comunicación entre el procesador y la terminal de programación.
 - El segundo conector etiquetado PEER COMM INTFC, es la conexión de el data highway plus y es una conexión paralela al conector de 9 pins carcasa-D PEER COMM INTFC.
 - El conector etiquetado REM I/O, se utiliza para las entradas y salidas remotas.
- **Batería.-** El procesador cuenta con una batería de litio, esta mantiene alimentada la memoria del procesador cuando éste se encuentra apagado o por alguna razón se pierde la energía de alimentación al procesador.

- **Grupo de interruptores.-** El procesador cuenta con tres grupos de interruptores de configuración que de acuerdo a su colocación van a determinar el funcionamiento del mismo, así el primer grupo va a determinar:
 - El número de estación del procesador cuando este en la red del data highway plus.
 - El modo de operación del procesador ya sea scanner o adaptador .

El segundo grupo determina:

- El número de palabras reservadas de la tabla de datos para la comunicación entre el procesador PLC-5 y un procesador huésped, cuando el procesador se encuentra configurado en modo adaptador.
- El número asignado al grupo de entradas y salidas a el módulo procesador cuando éste se encuentra en modo adaptador.
- El número de rack de entradas y salidas asignado a el módulo procesador cuando éste está en modo adaptador.

El tercer grupo se utiliza para conectar un terminador a través de la línea al procesador cuando el procesador es un dispositivo final en la línea de data highway plus o en el enlace de E/S remotas.

FUENTES DE ALIMENTACIÓN.

Para que un sistema a base de PLC pueda trabajar se requiere de alimentación a el módulo procesador, y los módulos de entradas y salidas. Existen dos tipos de fuentes de alimentación :

- Módulos de fuentes de alimentación, localizados dentro del mismo chasis del procesador.
- Fuentes de poder localizadas fuera del chasis.

Estas fuentes suministran el nivel de voltaje adecuado a todos los módulos de entradas y salidas dentro del rack para poder establecer un protocolo entre las señales del procesador y los dispositivos a controlar.

HARDWARE DE ENTRADAS Y SALIDAS (E/S).

Para conectar el procesador con la máquina o proceso en cuestión, se requiere de un chasis y varios módulos de entradas y salidas como interfase, además si el proceso requiere de más entradas y salidas, es necesario instalar un módulo adaptador de entradas y salidas o un PLC - 5 configurado como módulo adaptador para cada chasis remoto utilizado, a continuación se describen las partes que integran el software de entradas y salidas.

Chasis de entradas y salidas.- Un chasis de entradas y salidas es un gabinete compacto dentro del cual se coloca el procesador, los módulos de E/S y los módulos adaptadores de E/S remotas. La ranura del extremo izquierdo del chasis de E/S esta reservada para el procesador o para un módulo adaptador.

- **Módulos de entradas y salidas.-** Los módulos de E/S están agrupados en tres diferentes tipos: discretos , analógicos y de comunicación.
Los módulos de entradas y salidas discretos interactúan con los dispositivos on/off como son interruptores de límite o carrera, botones pulsadores y arrancadores de motores. Los módulos de E/S discretos están disponibles con terminales de 8 , 16, o 32 puntos de conexión de entradas o salidas por módulo. Los módulos de E/S analógicos realizan las conversiones necesarias analógico - digital, digital-analógico para interconectar las señales analógicas con que trabajan algunos dispositivos, al procesador. Los módulos de E/S de comunicación,

enlazan al procesador con otros dispositivos como son chasis de E/S remoto, controladores de circuitos, y cables de fibra óptica.

Módulo adaptador remoto de entradas y salidas.- Un módulo adaptador de E/S remoto permite la comunicación entre el procesador y los módulos de E/S contenidos en un chasis de E/S remoto.

MODOS DE DIRECCIONAMIENTO PARA UN RACK DE ENTRADAS Y SALIDAS.

Para direccionar un rack, es necesario conocer las diferencias que existen entre grupos de E/S y racks de E/S.

Un grupo de entradas y salidas es una unidad de direccionamiento que puede contener hasta 16 terminales de entrada y 16 terminales de salida vease figura 2.2

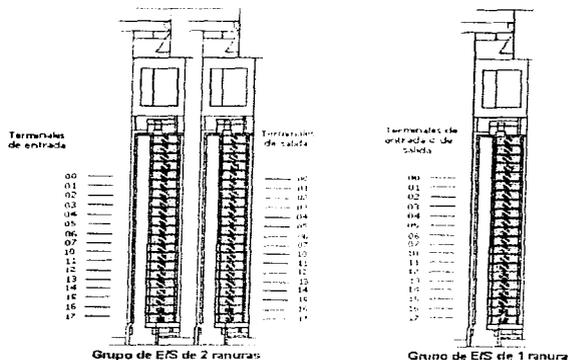


FIGURA 2.1.1 GRUPOS DE ENTRADAS Y SALIDAS

Un rack de E/S es una unidad de direccionamiento que puede contener hasta 8 grupos de E/S

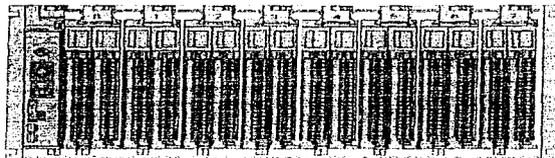


FIGURA 2.1.2 RACK DE ENTRADAS Y SALIDAS

El procesador PLC-5 puede direccionar sus entradas y salidas en grupos de 2 ranuras, 1 ranura o 1/2 ranura.

Se puede seleccionar el método de direccionamiento para el chasis en el que reside el procesador, utilizando los interruptores 4 y 5 del ensamblaje de interruptores que se encuentra en el chasis de E/S. los diferentes modos de direccionamiento se mencionan a continuación:

Direccionamiento a dos ranuras.-

Cuando se selecciona el direccionamiento de dos ranuras, el procesador direcciona dos ranuras de E/S como un grupo de E/S.

Cada grupo físico de E/S de dos ranuras corresponde a una palabra en la tabla de imagen¹ de entrada y una palabra en la tabla de imagen de salida. Cada terminal de entrada corresponde a un bit en la palabra de la tabla de imagen de entrada y cada terminal de salida corresponde a un bit en la palabra de la tabla de imagen de salida.

De la tabla de imagen de E/S, 32 bits están disponibles para cada grupo de E/S de 2 ranuras: 16 bits en la tabla de imagen de entrada y 16 bits en la tabla de imagen de salida. El número de bits por palabra estará determinado por el tipo de módulo que se instale.

Los módulos de E/S de 16 puntos, tienen 16 terminales de entrada y 16 terminales de salida. Un módulo de E/S de 16 puntos utiliza una palabra completa en la tabla de imagen de entrada o salida como se muestra en la figura. Este grupo de E/S usa 16 bits de la tabla de imagen de entrada y 16 bits en la tabla de imagen de salida.

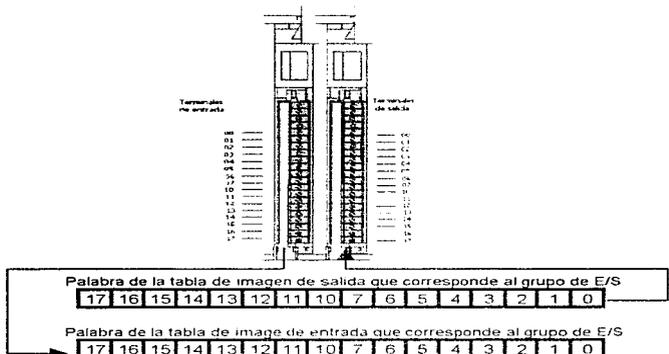


FIGURA 2.1.3 TABLA DE IMAGEN DE ENTRADAS Y SALIDAS A DOS RANURAS

¹Localidades del mapa de memoria.

Cuando se utiliza direccionamiento de dos ranuras, a cada par de ranuras se le asigna un correspondiente par de palabras en la tabla de imagen de entrada y salida.

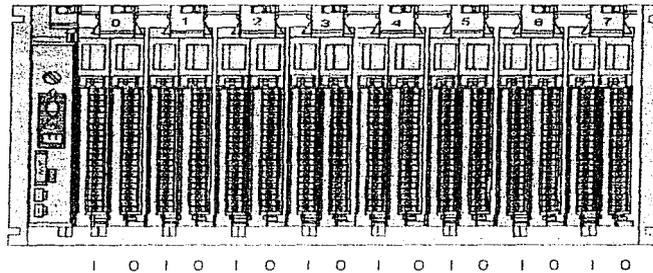


FIGURA 2.1.4 RACK DE ENTRADAS Y SALIDAS CONFIGURADO A DOS RANURAS

Direccionamiento a una ranura.-

Cuando se selecciona direccionamiento de una ranura, el procesador direcciona una ranura de un módulo de E/S como un grupo de E/S. Cada ranura físico en el chasis corresponde a una palabra en la tabla de imagen de entradas y salidas. El tipo de módulo que se utilice determina el número de bits y el tipo de palabra que se utiliza.

Con el método de direccionamiento de una ranura, 16 bits de entrada y 16 bits de salida estarán disponibles en la tabla de imagen del procesador para cada grupo de E/S. Sin embargo se pueden utilizar módulos de 8 puntos, 16 puntos, o una combinación de ambos en cualquier orden.

Cuando se utilizan módulos de entrada y salida de 8 puntos y direccionamiento de un ranura, solo son utilizados 8 bits de la palabra de la tabla de imagen de E/S por cada grupo de E/S que se utilice :

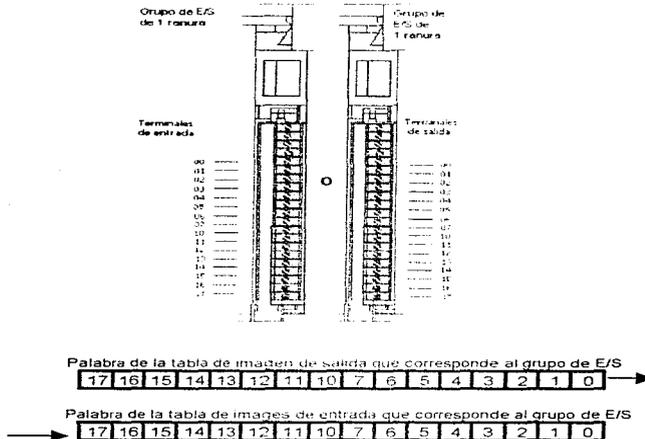


FIGURA 2.1.5 TABLA DE IMÁGENES DE ENTRADAS Y SALIDAS A UNA RANURA

La figura 2.1.5 muestra un grupo de E/S de una ranura con un módulo de E/S de 16 puntos. Un módulo de 16 puntos en un grupo de E/S de una ranura utiliza una palabra completa en la tabla de imagen del procesador. Los módulos de E/S de 32 puntos necesitan 32 bits de entrada y 32 bits de salida en la tabla de imagen del procesador. Debido a que solo se tienen disponibles 16 bits de entrada y 16 de salida por cada grupo de E/S de una ranura, el procesador utiliza las direcciones de la palabra asociada con la ranura de E/S adjunto para poder direccionar un módulo de E/S de 32 puntos.

Quando se utiliza direccionamiento de una ranura, a cada ranura se le asigna una palabra correspondiente en la tabla de imagen de entrada o salida :

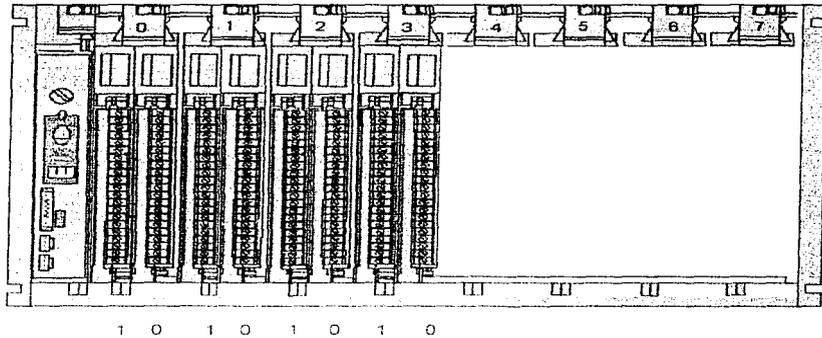


FIGURA 2.1.6 RACK DE ENTRADAS Y SALIDAS CONFIGURADO A UNA RANURA.

2.2 SOFTWARE

Las instrucciones de programación de un PLC-5 se encuentran desglosadas en diferentes grupos, como son: instrucciones de tipo relé, de temporizador, de contadores, de comparación, instrucciones de secuenciador etcétera. Debido a que la finalidad de esta tesis es la de programación de un robot, y no el análisis de un PLC, solo se darán a conocer las instrucciones que intervienen en el proceso, y el resto de las instrucciones solo se darán a conocer literalmente.

El lenguaje de usuario para la programación de un PLC es conocido como diagrama de escalera, o escalera lógica. Se escribe o diseña una escalera lógica, para examinar el estado de encendido o apagado del dispositivo de una máquina mediante la lectura del bit de dato en el archivo de imagen de entrada y salida; tomar decisiones basadas en las condiciones de entradas y salidas o bien controlar el estado de un bit de datos en el archivo de imagen de salida que controla el estado de encendido o apagado de los dispositivos de salida.

Para diseñar una escalera lógica, es necesario entender las siguientes definiciones:

- Renglón .- es el estado lógico que controla uno o más bits basado en el estado de otros bits previamente examinados . Una escalera lógica esta compuesta de una lista de renglones.
- Instrucciones de entrada .- (condiciones) examina el estado de los bits de entradas y salidas. Estos comandos deben ser utilizados al lado izquierdo del renglón.
- Instrucciones de salida .- controlan el estado del bit o bits dependiendo de las instrucciones de condicionamiento. Estos comandos son usualmente programados en el extremo derecho del renglón.
- Renglón lógico .- la siguiente figura 2.2.1 muestra un ejemplo de renglones lógicos.

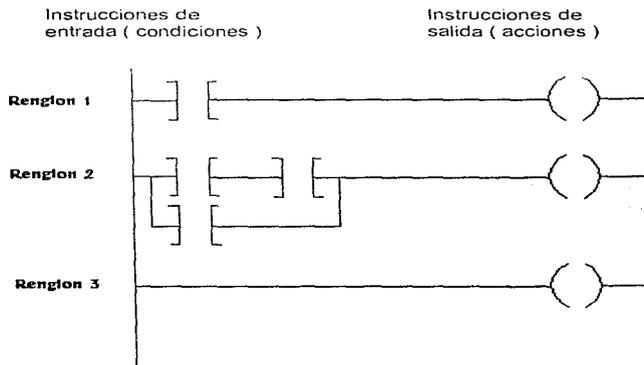


Figura 2.2.1

Cada instrucción de entrada e instrucción de salida programadas, generan un elemento de renglón. Sin embargo, cuando cada instrucción de entrada es ejecutada, el bit direccionado es examinado para verificar si este coincide con una cierta condición (encendido o apagado) . si la condición de entrada es verdadera (en el renglón 1) , el

renglón conmuta a verdadero. Todas las instrucciones de entrada deben seguir una trayectoria de elementos verdaderos desde el inicio del renglón hasta la instrucción de salida, para que la instrucción de salida sea habilitada, de tal forma que el renglón dos cuenta con dos trayectorias condicionadas para que la instrucción de salida sea habilitada. Si un renglón no cuenta con condiciones de entrada (renglón 3) la señal de salida permanecerá siempre activa.

La figura anterior utiliza las siguientes instrucciones de entrada y salida:

NOMBRE	FORMATO	DESCRIPCION				
EXAMINE ON		<p>Es una instrucción de entrada que examina el estado del bit dependiendo de las siguientes condiciones:</p> <p>si el bit es: entonces la instrucción es</p> <table border="0"> <tr> <td>ON (1)</td> <td>VERDADERA</td> </tr> <tr> <td>OFF (0)</td> <td>FALSA</td> </tr> </table>	ON (1)	VERDADERA	OFF (0)	FALSA
ON (1)	VERDADERA					
OFF (0)	FALSA					
OUTPUT ENERGIZE		<p>Es una instrucción de salida que controla el estado de un bit que a la vez esta controlando un dispositivo de salida, dependiendo de :</p> <p>si la instrucción es : entonces el bit es :</p> <table border="0"> <tr> <td>VERDADERA</td> <td>ON (1)</td> </tr> <tr> <td>FALSO</td> <td>OFF (0)</td> </tr> </table>	VERDADERA	ON (1)	FALSO	OFF (0)
VERDADERA	ON (1)					
FALSO	OFF (0)					

Como se mostró en la tabla anterior, por su modo de respuesta a las instrucciones se les conoce como instrucciones del tipo relay. Se utilizan las instrucciones de tipo relay, para monitorear y controlar el estado de los bits en la tabla de datos.

El archivo de imagen de entrada en el procesador almacena el estado de los dispositivos conectados en las terminales de los módulos de entrada. Las instrucciones en la lógica de escalera se programan para controlar los bits en el archivo de imagen de entrada.

De la misma manera, el archivo de imágenes de salida controla el estado de los dispositivos conectados a los módulos de salida.

Las siguientes son instrucciones de tipo relay.

Examine On (XIC)

Cuando un dispositivo cierra su circuito , el módulo al cual esta conectado el circuito lo detecta como circuito cerrado, el procesador refleja este estado en la tabla de datos. Cuando el procesador encuentra una instrucción XIC que direcciona al bit que corresponde a la terminal de entrada, el procesador determina si el dispositivo está cerrado (ON), si esta condición se da, la instrucción se hace verdadera, si el procesador encuentra un estado de circuito abierto (OFF) , la instrucción es falsa. De esta manera una instrucción XIC es verdadera o falsa dependiendo de la condición en que se encuentra el bit direccionado por el procesador.

Examine off (XIO)

Cuando un dispositivo abre su circuito, el módulo al cual fue conectado lo detecta como un circuito abierto y el procesador lo refleja como un estado OFF en la tabla de datos. Cuando el procesador encuentra una instrucción XIO que direcciona al bit correspondiente a la terminal al cual fue conectado el dispositivo, el procesador determina si el dispositivo está abierto (OFF). Si el procesador detecta un estado OFF designa un valor lógico para esta instrucción como verdadero. Si el procesador encuentra un estado ON, le da un valor de falso a la instrucción XIO.

Output energize (OTE)

La instrucción OTE se utiliza para controlar un bit en memoria. Si este bit corresponde a una terminal de un módulo de salida, el dispositivo conectado a este punto se energizará cuando la instrucción es habilitada y se desenergizará cuando la instrucción es deshabilitada. Si las instrucciones que preceden a la instrucción OTE son falsas, el procesador deshabilita la instrucción, y el dispositivo correspondiente se desenergiza.

La instrucción OTE es similar a una bobina de relé, en el que la bobina es controlada por los contactos en el renglón alambreado.

Output latch (OTL)

La instrucción OTL es una instrucción de salida retentiva que solamente puede activar un bit (no lo puede desactivar). Esta instrucción usualmente se utiliza con la instrucción unlatch (OTU) con las dos instrucciones direccionando el mismo bit.

Cuando se le asigna una dirección a la instrucción OTL que corresponde a un módulo de salida, el dispositivo conectado a ésta dirección se energizará cuando todas las instrucciones de entrada que preceden a la instrucción OTL son verdaderas. Cuando las condiciones de entrada se hacen falsas, el procesador mantiene activado el bit de salida correspondiente a la instrucción OTL. Para desactivar el bit activado, se utiliza la instrucción OTU.

Output unlatch (OTU)

Inversa a la instrucción OTL, la instrucción OTU desactiva el bit direccionado. Las dos instrucciones se utilizan en forma pareada apuntando al mismo bit, de manera tal que cuando la instrucción OTL activa un bit, la instrucción OTU que direcciona al mismo bit lo desactiva.

Las instrucciones del tipo contadores y de timers permiten el control de una operación que está basada en tiempo o en número de eventos, las siguientes son instrucciones de timers y contadores.

Timer on delay (TON)

Se utiliza la instrucción TON para activar o desactivar una salida después de un intervalo de tiempo preestablecido. La instrucción TON comienza a acumular tiempo desde el momento que el renglón se vuelve verdadero, y continúa así hasta que cualesquiera de las siguientes condiciones se cumple:

- El valor acumulado es igual al valor preestablecido.
- El renglón conmuta a falso.
- Una instrucción de reset restablece el temporizador.

El procesador restablece el valor acumulado cuando las condiciones del renglón se hacen falsas, independientemente de que el temporizador se haya sobrepasado de el tiempo permitido o no.

Para controlar eventos dependientes del tiempo, el TON cuentan con tres bits de estado.

1. El bit de temporizador habilitado (.EN) es activado cuando el renglón es verdadero y permanece así hasta que el renglón conmuta a falso, o bien una instrucción de reset restablece el temporizador. Una vez activado, el bit .EN indica que el temporizador esta habilitado.
2. El bit de temporizador contando (.TT) es activado cuando el renglón es verdadero y permanece así hasta que el renglón es falso, el bit .DN es activado, esto es el valor acumulado es igual al preestablecido o bien una instrucción de reset restablece el temporizador.
3. El bit de tiempo terminado (.DN) es habilitado cuando el valor acumulado es igual al valor preestablecido hasta que el renglón sea falso o bien hasta que una instrucción de reset restablece el temporizador. Cuando se activa, el bit .DN indica que la operación de temporización ha sido terminada.

El diagrama siguiente muestra un ejemplo de aplicación de la instrucción TON.

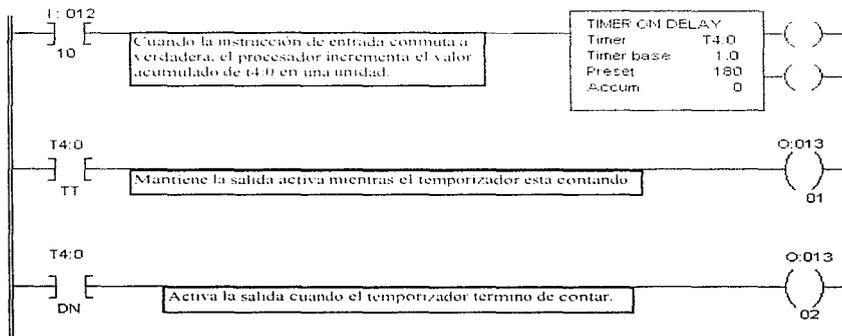


FIGURA 2.2.2 DIGRAMA DE ESCALERA QUE MUESTRA LA INSTRUCCIÓN TON

Timer off delay (TOF)

Esta instrucción se utiliza para activar o desactivar una salida después de que su renglón ha sido desactivado por un intervalo de tiempo. La instrucción TOF comienza a acumular tiempo cuando el renglón conmuta a falso y continua así hasta que cualesquiera de las siguientes condiciones se cumple:

- El valor acumulado es igual al valor preestablecido.
- El renglón se vuelve verdadero.
- Una instrucción de reset restablece el temporizador.

El TOF cuenta con las siguientes condiciones para sus bits de estado.

1. Bit de temporizador habilitado (.EN) es activado cuando el renglón es verdadero. El bit .EN es restablecido cuando el renglón es falso o cuando una instrucción de reset restablece el temporizador.
2. El bit de temporizador contando (.TT) es activado cuando el renglón se vuelve falso y el valor acumulado es menor que el valor preestablecido. El bit .TT es restablecido cuando el renglón es verdadero o cuando el bit .DN es restablecido, es decir el valor acumulado es igual al valor preestablecido.
3. El bit de tiempo terminado .DN es restablecido cuando el valor acumulado es igual al valor preestablecido. El bit .DN es activado cuando el renglón se vuelve verdadero o cuando una instrucción de reset restablece el temporizador

El diagrama siguiente muestra un ejemplo de la instrucción TOF.

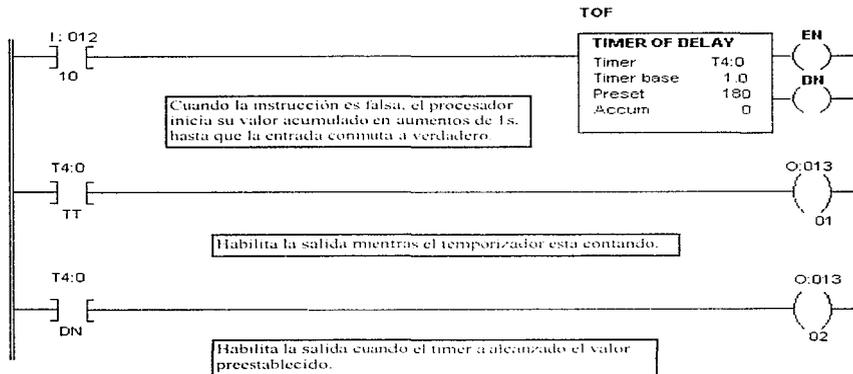


FIGURA 2.2.3 DIAGRAMA DE ESCALERA QUE MUESTRA LA INSTRUCCIÓN TOF

Count Up (CTU)

La instrucción CTU realiza un conteo ascendente cada vez que su renglón transita de falso a verdadero esta cuenta va de -32768 a 32767. Cuando el valor acumulado es igual al valor o mayor al preestablecido, la instrucción CTU habilita el bit .DN el cual puede ser utilizado dentro del programa para realizar una acción .

El valor acumulado de la instrucción es retentivo y la cuenta se mantiene hasta que una instrucción RESET que apunta a la misma dirección que el contador la restablece.

Los bits de estado de la instrucción CTU se utilizan para controlar eventos y las características de éstos son :

- El bit .CU es habilitado cuando el renglón conmuta de falso a verdadero, indicando que se esta realizando una cuenta ascendente. El bit .CU es restablecido cuando el renglón conmuta a falso o bien cuando una instrucción RES que apunta a la misma dirección lo restablece.
- El bit .DN es habilitado por el procesador cuando el valor acumulado alcanza al valor preestablecido, y permanece en este estado mientras el valor acumulado sea igual o mayor que el preestablecido, la instrucción RES restablece al bit .DN .
- El bit .OVL es habilitado por el procesador para indicar que la cuenta ha sobrepasado el valor limite superior (32767) a su vez el contador reinicia su cuenta a partir de -32768. el bit .OVL es restablecido cuando una instrucción RES que apunta a la misma dirección de la instrucción CTU es habilitada.

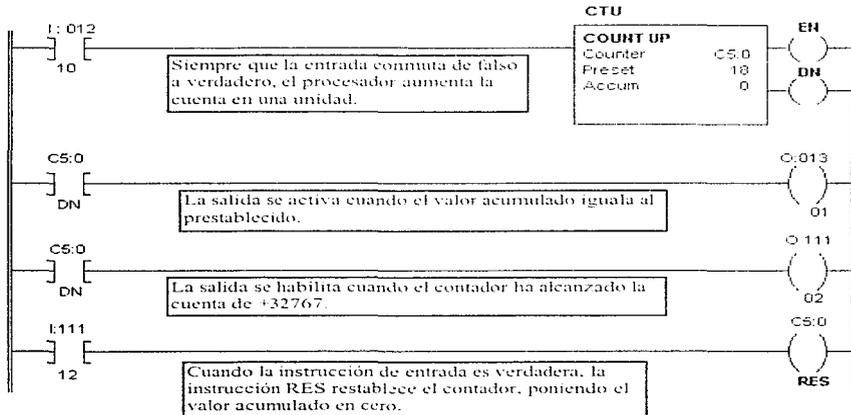


FIGURA 2.2.4 DIAGRAMA DE ESCALERA QUE MUESTRA LA INSTRUCCIÓN CTU

Count Down (CTD)

Contraria a la instrucción CTU la instrucción CTD realiza una cuenta regresiva cada vez que el renglón conmuta de falso a verdadero. La cuenta puede realizarse desde 32767 a -32768. El bit .DN permanece activo en todo momento mientras que el valor acumulado no sea menor al valor preestablecido, cuando esto sucede el bit .DN se deshabilita.

Los bits de estado para la instrucción CTD tienen las siguientes características:

- El bit .CD se habilita cuando el renglón conmuta de falso a verdadero para indicar que el contador esta realizando una cuenta descendente. El bit .CD es restablecido cuando el renglón conmuta de verdadero a falso o bien cuando una instrucción RES que apunta a la misma dirección que el contador es habilitado.
- El bit .DN es habilitado por el procesador cuando el valor acumulado es mayor o igual que el valor preestablecido. El bit .DN es restablecido cuando el valor acumulado es menor que el valor preestablecido, o bien cuando una instrucción RES que apunta a la misma dirección de la instrucción CTD es habilitado.
- El bit .UN es habilitado por el procesador para indicar que el valor acumulado es menor al límite inferior (-32768) a su vez el contador continua con su cuenta descendente a partir de 32767. El bit .UN es restablecido con una instrucción RES apuntando a la misma dirección del contador.

Como se ha visto los contadores y temporizadores se restablecen cuando una instrucción RES apuntando a la misma dirección de cualesquiera de los contadores en cuestión se habilita. Así, de esta manera, la instrucción RES sirve para restablecer cualquier contador o temporizador, lo que realiza la instrucción RES es poner a cero el valor acumulado.

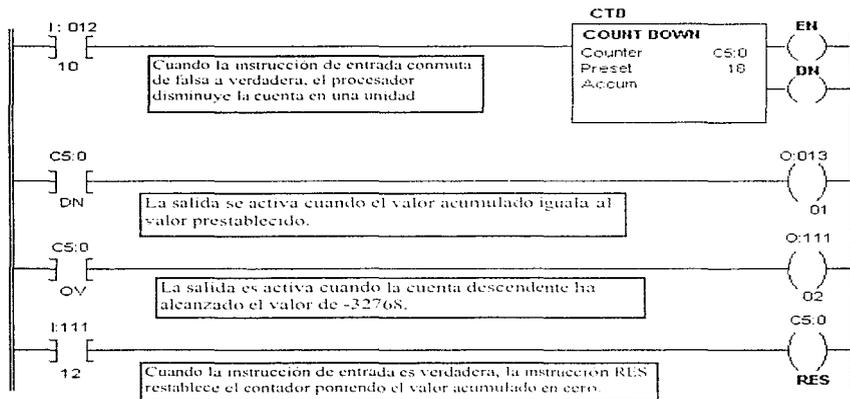
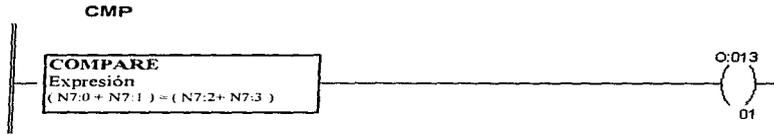


FIGURA 2.2.5 DIAGRAMA DE ESCALERA QUE MUESTRA LA INSTRUCCIÓN CTD

Las instrucciones de comparación pertenecen a otro bloque de instrucciones con las que cuenta el procesador, estas instrucciones se utilizan para realizar comparaciones entre dos valores utilizando expresiones o valores específicos en la comparación, las siguientes son instrucciones de comparación.

Compare (CMP)



La instrucción CMP realiza la comparación dada por una expresión aritmética, si esta comparación es verdadera el renglón conmuta a verdadero en cualquier otro caso el renglón es falso.

La estructura de la expresión sobre la cual se va a realizar la comparación, se define con operadores, direcciones o constantes, el direccionamiento de las expresiones puede ser directo, indirecto o indexado. Los operadores para la comparación pueden ser las siguientes: =, <>, <, <=, >, >=, +, -, *, sqrt etc.

Equal to (EQU)



La instrucción EQU se utiliza para comparar si los valores entre dos fuentes son iguales, los valores de las fuentes A y B pueden ser valores directos o direcciones que contienen valores. Si los valores de la fuente A y la fuente B son iguales, el renglón conmuta a verdadero.

Greater than or equal to (GEQ)



Se utiliza la instrucción GEQ para comparar si el valor de una fuente (A), es mayor o igual a el valor de otra fuente (B). Los valores de las fuentes A y B pueden ser valores directos o direcciones que contienen valores. Cuando esto sucede, el renglón conmuta a verdadero.

Greater than (GRT)



Se utiliza la instrucción GRT para comparar si el valor de una fuente (A), es mayor a el valor de otra fuente (B). Los valores de las fuentes A y B pueden ser valores directos o direcciones que contienen valores. Cuando esto sucede, el renglón conmuta a verdadero.

Less than or Equal to (LEQ)



Se utiliza la instrucción LEQ para comparar si el valor de una fuente (A), es menor o igual a el valor de otra fuente (B) . Los valores de las fuentes A y B pueden ser valores directos o direcciones que contienen valores. Cuando esto sucede, el renglón conmuta a verdadero.

Less than (LES)



Se utiliza la instrucción LES para comparar si el valor de una fuente (A), es menor a el valor de otra fuente (B) . los valores de las fuentes A y B pueden ser valores directos o direcciones que contienen valores. Cuando esto sucede, el renglón conmuta a verdadero .

Not Equal to (NEQ)



Se utiliza esta instrucción para comprobar si los valores entre dos fuentes son diferentes, cuando se da esta condición el renglón conmuta a verdadero.

Otro grupo de instrucciones son las de movimiento y modificación de bits, estas instrucciones operan con 16 bits de números enteros o binarios. Las instrucciones utilizan una máscara para bloquear o dejar pasar bits de datos, la máscara debe de tener una palabra del mismo tamaño que la fuente y el destino.

Move (MOV)

La instrucción MOV es una instrucción de salida que copia el valor de una dirección fuente a una dirección destino. Todo el tiempo que el renglón permanezca verdadero la instrucción mueve los datos en cada exploración.

Para programar esta instrucción se debe de dar la siguiente información al procesador:

- Fuente: es una constante de programa o direcciones de datos de donde la instrucción lee una imagen del valor.
- Destino: es una dirección de datos en donde la instrucción escribe el resultado de la operación, la instrucción escribe sobre cualquier dato almacenado.

Masked Move (MVM)

La instrucción MVM es una instrucción de salida que copia el valor de una dirección fuente a una dirección destino, enmascarando parte de los datos. Cuando el renglón que contiene esta instrucción es verdadero, los datos contenidos en la dirección fuente pasan a través de una máscara depositándolos en una dirección destino.

Los datos de la fuente y el destino deben de ser del mismo tipo, de manera contraria, la instrucción no realizará la operación. Se puede utilizar la instrucción MVM para copiar imagen de entrada o salida, valores enteros o binarios.

Los siguientes son otros grupos de instrucciones de los cuales solo se hará mención de las instrucciones que los componen.

Instrucciones de computo.

Las instrucciones de computo permiten evaluar operaciones aritméticas utilizando una instrucción o expresión aritmética específica. Se pueden realizar operaciones con datos de diferentes tipos, como son de punto flotante y enteros. Las siguientes son instrucciones de computo que el procesador puede realizar:

Aritméticas (-, +, *, /, SQR, LN, LOG)

Trigonométricas (ACS, ASN, ATN, COS, SIN, TAN)

Lógicas (AND, OR, XOR, NOT)

De conversión (BINARIO A BCD, BCD A BINARIO, DEG, RAD)

Instrucciones de archivo.

Las instrucciones de archivo permiten realizar operaciones con archivos de datos. Las instrucciones de archivos son:

File Arithmetic and Logic (FAL), File Search and compare (FSC), File Copy (COP), File Fill (FLL).

Instrucciones de Registros de Corrimiento.

Un registro de corrimiento es usualmente utilizado para simular un movimiento o flujo de partes e información. Las instrucciones de registro de corrimiento son:

Bit Shift Left (BSL), Bit Shift Right (BSR), Fifo Load (FFL) Fifo Unload (FFU), Lifo Load (LFL), Lifo Unload (LFU)

CAPITULO

3

3.- EL CONTROLADOR DEL ROBOT

El sistema del robot , es un sistema basado en computadora de aplicaciones generales. El sistema básico consiste de :

- Unidad mecánica.
- Sistema de control.

El sistema de control y el mecánico son dos unidades separadas interconectadas por un cable. El sistema de control contiene un sistema electrónico, y ofrece una diversidad de oportunidades para la comunicación externa y control de equipos periféricos .

La comunicación con el operador, la programación y la puesta en marcha en forma manual es realizada mediante una unidad de programación portátil y el panel del operador o el gabinete de control.(véase figura 3.1.1)

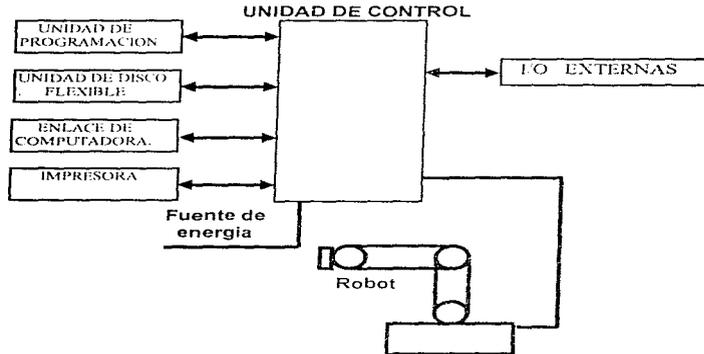


FIGURA 3.1.1 ESQUEMA GENERAL DEL SISTEMA ROBOT

3.1 SISTEMA DE CONTROL

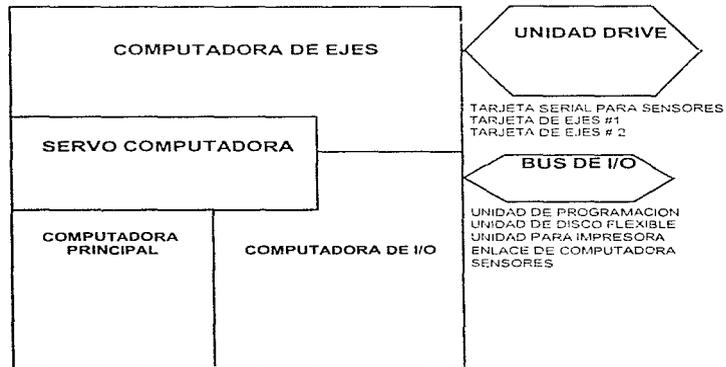
El sistema es un equipo computarizado diseñado para controlar el brazo del robot. La consola del controlador es un gabinete metálico dentro del cual se encuentran los componentes de control, las unidades externas, control de los ejes externos, periféricos, etc. El sistema de control consiste básicamente de las siguientes unidades:

Unidad electrónica.

Todo el control electrónico se encuentra localizado dentro de un rack el cual contiene las siguientes subunidades o subensambles:

- Computadora del robot, que esta compuesta por una computadora principal, una computadora de entradas y salidas, una computadora para los servos, una computadora para los ejes, memorias de lectura y escritura para los programas de los robots.
- Tarjeta de seguridad, que contiene la lógica para los paros de emergencia.
- Tarjeta de entradas y salidas digitales.
- Unidad de alimentación de voltaje.
- Unidades para los drives.

La computadora del robot esta construida en una tarjeta en la que se distribuye todo el sistema de control del robot, esta consiste de cinco bloques funcionales que son:



1.- La computadora principal, es un sistema que controla la servo computadora y la computadora de entradas y salidas. esta computadora es un sistema digital basado en el microprocesador 68000, con sus EPROM para el programa monitor, memoria RAM para los datos y programas de usuario y un canal serie que es utilizado para los sensores conectados al sistema.

2.- La computadora de entradas y salidas, tiene como función la comunicación entre la computadora principal y el mundo exterior, es un sistema digital basado en microprocesador cuya arquitectura tiene como CPU el integrado 68008, además de memorias RAM y un bus de interfase para el bus de entradas y salidas, tiene también un canal serie para conectar a una impresora.

3.- La servo computadora trabaja entre la computadora principal y la computadora de los ejes, es un sistema digital que tiene en su estructura al microprocesador 68000, memoria RAM y lógica digital para adaptar las señales a la computadora de los ejes.

4.- La computadora de los ejes trabaja sobre la servo computadora para controlar el sistema de medición, para calcular la posición de los ejes y controlar la velocidad. la computadora de ejes consiste de un CPU TMS 32010, memoria RAM y lógica digital para adaptar la tarjeta de ejes, tarjeta serial de sensores y unidad de drives.

5.- Los puertos de comunicación mediante los cuales se tiene acceso a las computadoras del robot.

Todas las señales que afectan la operación y seguridad de personal son ensambladas y coordinadas por la tarjeta de seguridad, la tarjeta es la conexión entre la sección de la computadora y las funciones de seguridad y supervisión que son independientes de la computadora.

La tarjeta de entradas y salidas es utilizada para la comunicación con el equipo de proceso asociado al robot. Cada tarjeta digital tiene 16 salidas y 16 entradas, en grupos de ocho canales. los diferentes grupos están aislados entre si y con el sistema de control.

La unidad de fuente de voltaje es alimentada por la unidad de potencia y provee cuatro voltajes regulados que tienen los siguientes valores:

- + 5 V d.c.
- +/- 15 V d.c.
- + 24 V d.c.

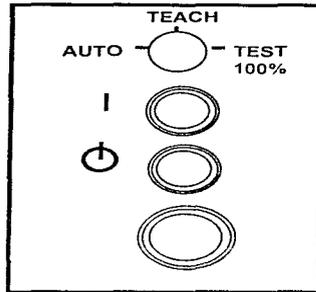
La unidad de drive es un convertidor de frecuencia utilizado para el control de torque de motores asíncronos o motores síncronos permanentemente magnetizados.

Una corriente de referencia de dos fases es utilizada para la función del control. El convertidor de frecuencia es alimentado por una fuente de c.c. de 340 V.

Unidad de alimentación.

La unidad de alimentación se encuentra localizada dentro del gabinete y provee la alimentación de voltaje necesaria para soportar a todo el sistema.

Unidad del operador.



La unidad del operador está montada al frente del gabinete de control. Los botones pulsadores de la unidad del operador están contruidos con la misma función de lámparas indicadores, las siguientes funciones están localizadas dentro de la unidad del operador :

- Selector de modo de operación; con este se pueden seleccionar tres modos diferentes de operación como a continuación se menciona:

Modo AUTO.- El modo AUTO es utilizado para correr los programas previamente almacenados, durante el proceso en cuestión. Durante este modo se presentan las siguientes condiciones:

- La programación y la edición es deshabilitada.
- El uso del joystick es deshabilitado.
- No se puede reducir la velocidad del robot.
- La corrección a los programas es deshabilitado.
- Se habilita el control externo mediante una computadora remota.

Modo TEACH.- El modo TEACH es utilizado cuando se trabaja dentro del área de trabajo del robot y cuando se está programando. Cuando este modo es seleccionado, se tiene lo siguiente :

- La programación y la edición es habilitada.
- El uso del joystick es permitido.
- La velocidad del robot es reducida a un máximo de 250 mm/s.
- El control externo es deshabilitado.

Modo TEST 100.- Este modo es utilizado para correr los programas a la velocidad real programada, pudiendo variar algunos parámetros al mismo tiempo. Las características de este modo son:

- La programación y la edición es permitida.
- El uso del joystick es permitido.
- La velocidad del robot es la real.
- El control externo es deshabilitado.

- **Botón de operación** ; es utilizado para arrancar el sistema cuando es seleccionado el modo AUTO. El voltaje es conectado a todos los motores del robot cuando el sistema esta en modo AUTO.
- **Botón de standby** ; es utilizado para poner en standby el robot sin importar el modo de operación previo. El voltaje de los motores es desconectado cuando el modo standby es seleccionado.
- **Paro de emergencia**; cuando se acciona el paro de emergencia, el robot se detiene inmediatamente sin importar el estado previo o el modo de operación del robot. El voltaje de los motores es desconectado después de que el botón de emergencia fue accionado.

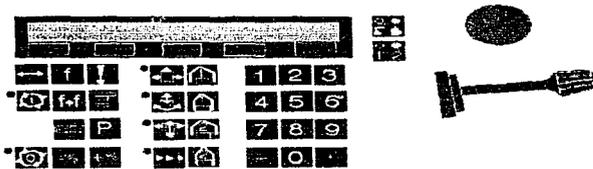


FIGURA 3.1. UNIDAD DE PROGRAMACIÓN DEL ROBOT.

La unidad de programación es una terminal portátil que se utiliza para la programación y operación del robot.

Un teclado es utilizado para insertar instrucciones de programa y otros datos de entrada. los programas del robot, menús de funciones, mensajes de error y del sistema son mostrados en un display de dos líneas. los movimientos de los ejes del robot son controlados con un joystick.

La unidad es conectada mediante un cable a el gabinete de control del robot. La unidad de programación esta compuesta de las siguientes partes fundamentales :

1. Teclado; el teclado es del tipo de membrana, todas las teclas están conectadas en un arreglo matricial de 8 x 7 . Diez LED'S son asociados a cada tecla para indicar el estado de la función en cuestión.
2. Display; el display es del tipo de cristal liquido y tiene dos líneas con 40 caracteres cada una.
3. Joystick ; el joystick tiene tres ejes, con cada movimiento se hace variar el valor de un potenciómetro. El potenciómetro esta conectado a un voltaje de 0 a 5 volts. El voltaje del potenciómetro pasa a un convertidor A/D para de aquí ser procesada la información.
4. Sección de la computadora; la sección consiste de un procesador 6803, una memoria RAM con 2KB de capacidad , una EPROM con 9KB de capacidad. La parte de la computadora maneja una comunicación serie con la computadora principal, lee el teclado, verifica e interpreta la deflexión del joystick y controla el display.
5. El bus de interfase serial; un canal serie RS-422 , es utilizado para la comunicación entre la unidad de programación y la computadora del robot, la comunicación es bidireccional.

La programación y otros trabajos con el sistema robot, mediante la unidad de programación, es realizada como un dialogo entre el operador y el sistema robot. El dialogo esta basado en un conjunto de menús desde el cual pueden ser seleccionadas diferentes alternativas.

Los menús son presentados en el display de la unidad de programación. El operador selecciona un menú mediante la presión de una tecla ubicada bajo la alternativa deseada.

El sistema robot responde presentando preguntas secuenciales , presentando información o textos de error o presentando una instrucción de programa en el display.

Los menús principales son obtenidos mediante la presión de las teclas INSTRUCTION, POSITIONING, AUTOMATIC, EDITING, MANUAL, VISION y PROCESS que se encuentran dentro de la unidad de programación.

Los cinco botones localizados abajo del display son designados como botones multifunción, y el trabajo que realizan dependerá de la función desplegada en el menú de alternativas al momento de ser oprimidas. La función SCAN es utilizada para desplegar otras alternativas dentro del display.

Cuando una alternativa ha sido seleccionada un nuevo menú o una pregunta con alternativas es presentada. Si un valor es requerido, éste es alimentado mediante las teclas numéricas. Instrucciones completas del programa son presentadas en la línea superior del display cuando el dialogo termina.

Un joystick es utilizado para posicionar el robot en un punto determinado cuando se esta programando. El joystick puede ser movido hacia delante o atrás, hacia la izquierda o derecha , girarlo en sentido directo o inverso a las manecillas del reloj o bien una combinación de estas. La velocidad correspondiente a los movimientos del robot es proporcional a la deflección del joystick.

Dos botones están asociados con el joystick. El control de los ejes del robot o el control de los ejes externos puede ser seleccionados mediante el botón superior. La función del botón inferior dependerá de el estado de el botón superior teniendo las siguientes funciones:

- Si se seleccionan los ejes del robot con el botón superior, con el botón inferior se puede seleccionar movimientos para los ejes 1-3 en una posición , y los ejes 4-6 en la otra.
- Si se seleccionan los ejes externos con el botón superior, con el botón inferior se pueden seleccionar los ejes 7-9 en una posición y 10-12 en la otra.

En cada ocasión que se presionan los botones la selección cambia y se ilumina un LED indicando la selección correspondiente.

La tarjeta de entradas y salidas

La tarjeta de entradas y salidas digitales es una tarjeta que acopla ópticamente el proceso para detectar señales de 24 V d.c. provenientes de contactos de relays o sensores y provee un voltaje de 24 V d.c. a cargas como relays, pequeños solenoides y lámparas. La tarjeta tiene los siguientes bloques:

1. Bus de interfase; el bus de interfase contiene circuitos que adaptan las señales del bus de entradas y salidas y la tarjeta.
2. Canal de entrada; los 16 canales de entrada están divididos en dos grupos de 8 canales cada grupo, aislados uno del otro, el voltaje de alimentación es común entre cada grupo. Las entradas son aisladas de la lógica interna mediante opto acopladores. La entrada de cada canal esta fabricada con una protección de transitorios y un filtro RC colocado después del opto acoplador.
3. Canal de salidas; el canal de salidas esta constituido por dos grupos de 8 canales aislados uno del otro. cada grupo tiene un voltaje común de alimentación. Las salidas están aisladas de la lógica interna mediante opto acopladores. Las salidas tienen además una protección contra corto circuito. LED's al frente de la tarjeta indican el estado de cada una de las salidas.

Entradas y salidas remotas (RIO)

La tarjeta RIO tiene un conector que permite la comunicación con un módulo remoto de PLC. Al frente de la tarjeta se cuenta con 32 LED's que indican el estado de las entradas y salidas. La tarjeta RIO contiene los siguientes bloques:

1. Bus de interfase; El bus contiene circuitos para conectar las señales del bus de E/S con la tarjeta y las direcciones internas decodificadas.
2. Indicador de estados; 32 LED's ubicados al frente de la tarjeta son utilizadas para indicar el estado de los 16 canales de entrada y los 16 canales de salida. La información del estado es tomada del bus de entradas y salidas.

Monitor.

Un monitor es utilizado para desplegar la información del estado del control, como es la parte del programa dentro del programa activo cuando la ejecución programada es detenida, cualquier comentario en el programa durante la ejecución programada y los mensajes de error ocurridos durante la ejecución del programa entre otros.

3.2 MOVIMIENTOS DEL ROBOT.

La unidad mecánica

La figura 3.2.1 muestra un patrón con los movimientos de cada uno de los ejes del brazo robot.

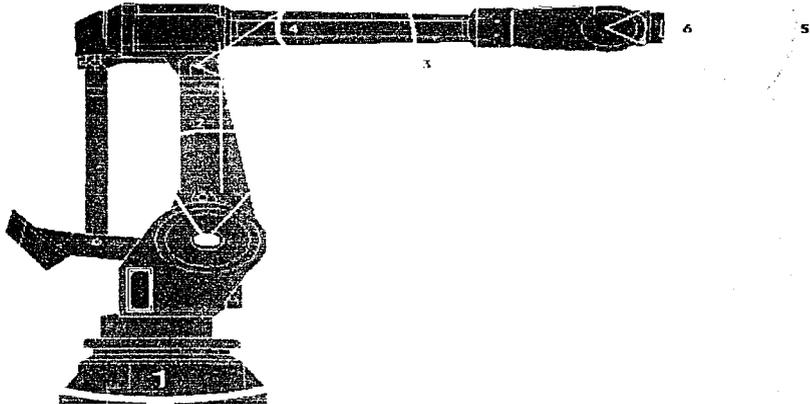


FIGURA 3.2.1 MOVIMIENTOS DEL ROBOT

- Eje 1 , Realiza una rotación completa del cuerpo del robot.
- Eje 2 , Realiza un movimiento hacia delante y hacia atrás en el brazo corto.
- Eje 3 , Genera un movimiento vertical en el brazo largo.
- Eje 4 , Realiza una rotación completa de la muñeca.
- Eje 5 , Inclina la muñeca alrededor del centro de la misma.
- Eje 6 , Rota la parte final del brazo.

Todos los movimientos del robot son servo controlados mediante motores de a.c., especialmente adaptados a cada eje.

El robot cuenta con un sistema de frenos electromecánicos en cada uno de los ejes . Los frenos son activados cuando éstos son desenergizados .

Sistemas coordenados.

Sistemas coordenados utilizados para programar el robot.

Todas las posiciones del robot son expresadas con la ayuda de los valores coordenados. Estos valores describen las posiciones del robot en el espacio. Para saber como interpretar las coordenadas, por ejemplo, para saber de que referencia y en que direccion se medirá la posición, una referencia, un marco o un sistema coordenado debe ser definido.

Los cálculos internos en el controlador del robot pueden ser expresados en el Sistema Coordenado de Base (SCB) como se observa en la figura siguiente.



FIGURA 3.2.2 SISTEMAS COORDENADOS

El SCB siempre es tomado en la base del robot con el plano x-y en el piso, y el eje z apuntando hacia arriba y concéntrico a el primer eje de rotación.

Sin embargo, no es adecuado expresar las posiciones programadas en SCB ya que es muy difícil cambiar las posiciones si el robot se mueve de posición. En ocasiones cuando el robot se mueve de posición, los movimientos relativos son los mismos; sin embargo las posiciones se encontrarán cambiadas. Por lo tanto no es necesario reprogramar todas las posiciones, basta solo redefinir el sistema coordenado utilizado. Para esta aplicación, el Sistema Coordenado Mundial (SCM) es utilizado para definir las posiciones almacenadas. La proyección del SCM respecto al SCB es definido como valor - FRAME, que mantiene los valores de x, y, z definiendo

el desplazamiento, y la reorientación. Otra de las coordenadas mostradas en la figura es el sistema coordenado de meta (SCM) este define la posición y orientación de meta, es expresado en coordenadas rectangulares y almacenado como un valor de posición en cada instrucción de posición.

Coordenadas utilizadas para definir movimientos y trayectorias en modo automático.

Mientras la posición final de un movimiento siempre es expresada en coordenadas rectangulares como se han descrito, la trayectoria entre las posiciones puede ser ejecutada en tres diferentes sistemas coordenados. Estas son coordenadas rectangulares, coordenadas de robot y coordenadas rectangulares modificadas que pueden todas ser activadas mediante instrucciones en un programa del robot. Cada uno de estos sistemas coordenados puede dar una diferente trayectoria y puede ser utilizada dependiendo de la velocidad, precisión y orientación.

- **Coordenadas rectangulares;** Estas coordenadas son dadas por default y son utilizadas en muchos movimientos. la trayectoria que sigue el TCP¹, puede ser circular o lineal dependiendo de la instrucción de posición utilizada. La velocidad puede ser controlada con el valor programado, pero la precisión de la trayectoria debe ser dada con prioridad. La velocidad sin embargo puede ser modificada cuando sea necesario.
- **Coordenadas de robot;** Cuando se mueve el robot en coordenadas de robot, todos los ejes son movidos a una velocidad constante desde la posición de inicio hasta la posición de fin. La trayectoria que describe el robot puede ser una curva en el espacio. Las coordenadas de robot son utilizadas cuando un movimiento rápido se da con mayor prioridad que una trayectoria de precisión lineal.
- **Coordenadas rectangulares modificadas;** Todos los ejes del robot pueden ser llamados "puntos singulares" en el área de trabajo. Un punto singular es un punto donde dos o más ejes del robot pueden ser paralelos

¹Punto Central de la Herramienta por sus siglas en ingles.

Sistemas coordenados utilizados en modo manual.

Tres sistemas coordenados son usualmente definidos para movimiento manual del robot. Estos son Sistema Coordinado de Base, Sistema Coordinado de la Herramienta y Sistema Coordinado del Robot, observar la figura.

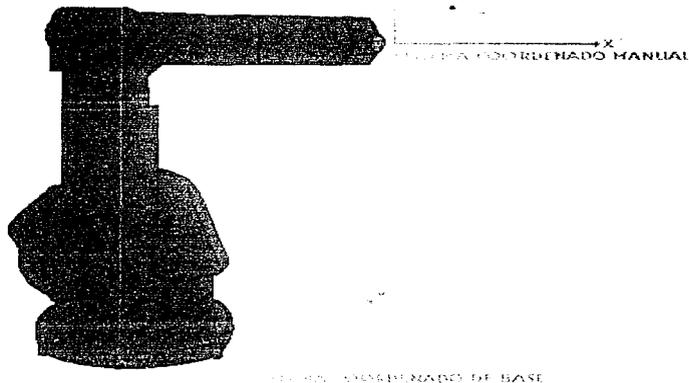


FIGURA 3.2.3 SISTEMAS COORDENADOS EN MODO MANUAL

El sistema coordenado de base siempre es colocado en la base del robot con el plano x-y en el piso y el eje z apuntando hacia arriba y concéntrico al eje principal de rotación. Cuando se utiliza el Sistema Coordinado de Base, los movimientos del joystick pueden corresponder a los movimientos del TCP paralelos a los ejes de base.

El sistema coordenado manual es utilizado para definir la herramienta y el sistema coordenado de la herramienta. El sistema coordenado manual es siempre colocado en el extremo final del robot, con el origen en el centro de la base, el eje x apuntando hacia afuera, y el eje z direccionado hacia una marca en la base.

El sistema coordenado de la herramienta, tiene su origen en el punto central de la herramienta (TCP) y es definido relativamente por el Sistema Coordinado Manual del robot. El Sistema Coordinado de la Herramienta es utilizado para definir las posiciones de mola del robot, de esta manera el Sistema coordenado de la herramienta es movido para coincidirlo con el sistema coordenado manual descrito anteriormente.

Movimientos del robot con el joystick

Para poder programar posiciones del robot, es necesario mover el robot a la posición deseada, y posteriormente guardar esta posición y así continuamente hasta completar la trayectoria deseada de la operación a realizar, para esto se requiere seleccionar cualesquiera de los sistemas coordenados. Las figuras muestran cada uno de los movimientos del robot correspondientes a los movimientos del joystick dependiendo del sistema coordenado seleccionado.

SISTEMA COORDENADO RECTANGULAR.

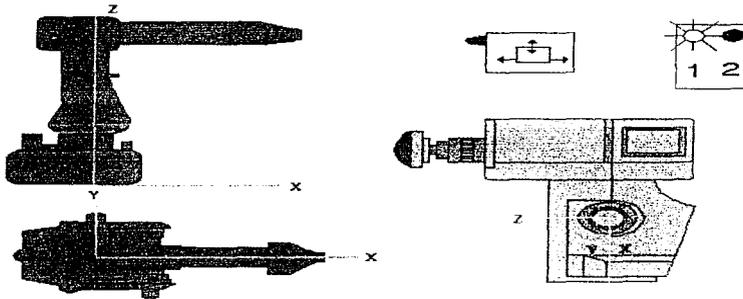


FIGURA 3.2.4 MOVIMIENTOS EN COORDENADAS RECTANGULARES

El sistema coordenado es permanentemente localizado con el origen en la base del robot como se muestra en la figura anterior. Cuando el joystick es manipulado en la dirección que muestran las flechas, el extremo donde esta instalada la herramienta se mueve trazando una línea recta en la dirección X, Y, y Z respectivamente.

SISTEMA COORDENADO DE EJES PRINCIPALES .

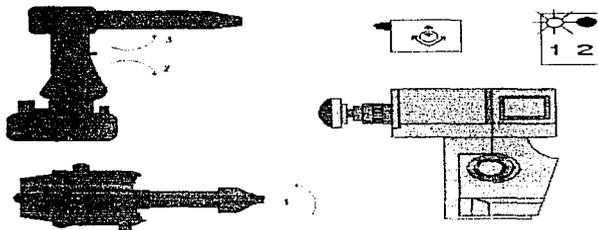


FIGURA 3.2.5 MOVIMIENTOS DE COORDENADAS DE EJES PRINCIPALES

Cuando el sistema coordinado de robot es seleccionado, cada uno de los ejes se mueve en la dirección señalada por las flechas de acuerdo a el movimiento del joystick , describiendo trayectorias circulares, el TCP no se mueve en línea recta como sucede en el sistema coordinado rectangular.

SISTEMA COORDENADO DE EJES DE LA MUÑECA.

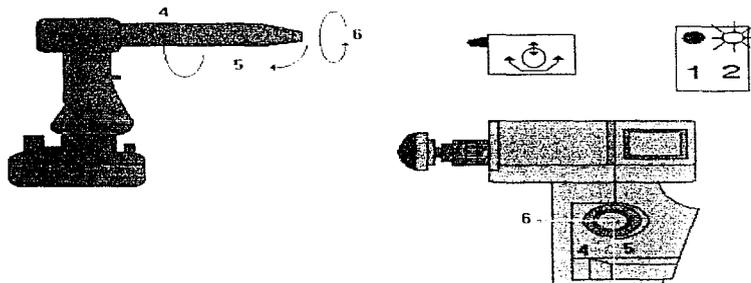


FIGURA 3.2.6 MOVIMIENTOS COORDENADOS DE EJES DE MUÑECA

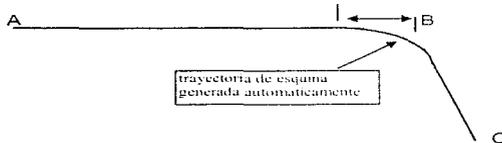
Cuando el sistema coordinado de robot es seleccionado, la deflexión del joystick de acuerdo a las flechas, corresponde a los movimientos de los ejes 4, 5 y 6 del robot respectivamente.

Programación de las trayectorias del robot.

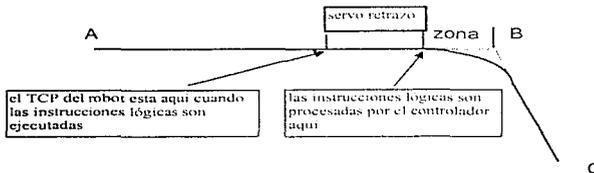
El sistema servo control mantiene el TCP del robot en una trayectoria que une dos posiciones programadas formando una línea continua imaginaria compensando los cambios de momentos de inercia, gravedad y dinámica del robot sin cambios bruscos.

La trayectoria seguida en coordenadas rectangulares es dirigida mediante la coordinación de movimientos continuos de cada eje individual hacia la velocidad de los ejes que limitan la realización por el momento. Cuando un cambio de posición en el movimiento del TCP es programado, el procesador principal automáticamente genera una trayectoria parabólica en la esquina.

La trayectoria de esquina se inicia a una distancia definida desde una posición programada. Esta distancia es llamada zona de esquina o simplemente zona.



Si la velocidad del TCP del robot es demasiada para realizar la esquina, la velocidad de entrada es reducida a un valor adecuado. Una cierta reducción de velocidad puede tomar también dentro de la zona de esquina.



Zonas

Una zona está definida como la distancia desde una posición programada hasta el punto en que el TCP del robot deja la trayectoria continua. Una zona es seleccionada mediante las instrucciones de posición.

El tamaño de la zona es definido en milímetros o en pulgadas, sin embargo el valor es aplicado solo cuando se utilizan coordenadas rectangulares. Cuando se utilizan coordenadas de robot el tamaño de la zona puede diferir levemente de la programada y es generalmente más pequeña que la definida.

Velocidad a lo largo de una trayectoria

Si la velocidad del TCP del robot es muy alta para ser capaz de tomar una curva, la velocidad cuando entra a una zona es reducida automáticamente al máximo permitido para realizar la trayectoria programada sin ninguna variación drástica. La máxima velocidad esta en función del tamaño de la zona y el ángulo de giro.

3.3 OPERACIÓN

Puesta en marcha

Cuando el sistema se coloca en modo RUN , cualquier programa puede ser ejecutado provocando que el robot ejecute los movimientos programados.

Las instrucciones de posición pueden ser ejecutadas en cualquiera de los siguientes modos:

- Instrucción por instrucción hacia adelante.
- Continuamente hacia adelante.
- Instrucción por instrucción hacia atrás.

Otras instrucciones solo pueden ser ejecutadas hacia delante, ya sea instrucción por instrucción o continuamente. Cuando se ejecuta hacia tras, algunas instrucciones pueden ser omitidas utilizando la instrucción SKIP.

Cuando un movimiento programado es ejecutado, el robot describe siempre una línea recta hacia la posición programada independientemente de donde comienza.

La ejecución del ciclo del robot programada, puede ser detenida de diferentes maneras, algunas de éstas son: manualmente , utilizando el boton de paro de la unidad de programación, manualmente utilizando el boton de paro de la consola de control y automáticamente utilizando un paro programado actuado por una señal de entrada digital.

Paros

Cualquier ejecución programada puede ser detenida en diferentes maneras.

- Manualmente, utilizando el boton de paro de la unidad de programación.
- Manualmente, utilizando el boton de paro del control remoto.
- Automáticamente, utilizando una señal de paro de las entradas digitales mediante el programa.
- Automáticamente, utilizando un paro programado dentro del programa del robot.

Registros

El control del programa o equipos periféricos es realizado en base a un valor de un registro. El sistema contiene 120 registros, donde un valor entero puede ser almacenado. Cuando un nuevo valor es almacenado el valor anterior es borrado del registro. Los valores permitidos están en el rango de - 32768 a + 32767.

Durante la ejecución de un programa, un valor numérico puede ser:

- Almacenado directamente
- Almacenado indirectamente desde una entrada digital o análoga
- Puede ser enviado hacia un equipo periférico mediante una salida digital o análoga.
- Puede ser comparado con otro valor designado en una instrucción de salto.
- Se puede sumar, restar, multiplicar o dividir con otro valor de registro
- Define un valor para una posición de robot.

La instrucción de registro es utilizada para:

- Leer un valor dentro de un número de registro directamente o mediante el contenido en otro registro.
- Accesar un valor desde un puerto
- Transmitir un valor de un registro a un puerto.

Entradas y salidas

El sistema del robot y los equipos periféricos pueden comunicarse mediante un sistema de entradas y salidas del tipo digital o analógicas.

Una entrada o salida digital puede tener solo uno de los dos diferentes estados, "1" (presencia de voltaje) o "0" (ausencia de voltaje). El sistema del robot transmite comandos de información a los equipos periféricos mediante las salidas y la correspondiente información proveniente de los mismos, es recibida por las entradas. La habilitación de salidas a 1 o 0, es manejado mediante una instrucción especial cuando se está programando el robot.

Cuando se tiene el sistema en modo manual, sucede cualesquiera de las siguientes condiciones:

- Se puede verificar el estado de cada una de las entradas digitales.
- Se puede verificar el estado de cada una de las salidas, al mismo tiempo se puede modificar el estado digital poniendo en 0 una salida que tenía 1 y viceversa.

Puertos

Un puerto es una conexión entre un número de entradas y salidas digitales y un número de registro dentro de la memoria del programa. Esto es posible en el programa de usuario mediante el uso de la instrucción REGISTER para conectar registros del 0 al 199 con entradas y salidas digitales. La función permite:

- El acceso a el estado de un grupo de entradas digitales para almacenarlo en un número de registro. El estado de cada entrada es dado por un dígito binario 1 o 0 y el número formado es almacenado en un número de registro.
- La transmisión de números de un número de registro a un grupo de salidas digitales poniendo a cada una de estas 0 o 1. Esto es codificado en forma binaria "1" (salida activa) o "0" (inactiva) después de que cada dígito binario es puesto a cero o a uno.

Entradas y salidas remotas RIO².

El sistema de control puede ser equipado con una tarjeta de entradas y salidas (RIO). Una tarjeta RIO puede ser programada a 32, 64, 96 o 128 señales de entradas y salidas, dependiendo de cuantas tarjetas de entradas y salidas sean instaladas en el sistema. El máximo número de entradas y salidas digitales que el sistema puede controlar es de 128. La tarjeta debe ser colocada a la izquierda de la última tarjeta de entradas y salidas del sistema, esto permite mayor espacio para elegir la configuración de la tarjeta RIO.

En la parte frontal, la tarjeta cuenta con 32 led's, los primeros 16 son utilizados para indicar el estado de las primeras 16 entradas de la tarjeta RIO, los siguientes 16 son para indicar el estado de las 16 salidas.

²Entradas y Salidas Remotas por sus siglas en inglés

3.4 EL MENÚ DE INSTRUCCIONES.

Existen cinco menús de instrucciones que sirven para programar y poner en marcha el sistema del robot. Debido a que el propósito de esta tesis es de aplicación particular de un robot, solo se hará hincapié en aquellas instrucciones que se utilizaron para la programación del mismo, haciendo solo mención de las restantes.

- **Menú de instrucciones lógicas.**

Este menú lo constituyen instrucciones que dan el control de secuencia al programa entre otros.

WAIT

Esta instrucción interrumpe la ejecución del programa :

- Hasta que transcurre un tiempo predefinido
- Hasta que una de las ocho entradas ha asumido el estado requerido.

El tiempo de espera puede ser determinado en segundos hasta con dos decimales en un rango de 0 a 99.99 seg.

El robot se mantendrá en estado de espera hasta que la señal de entrada programada , adquiera el valor condicionado 1 o 0 .

Esta instrucción se utiliza cuando se requiere interrumpir la ejecución del programa en una posición específica mientras otra máquina o un operador, realiza otro trabajo.

Un ejemplo de la aplicación de esta instrucción es el siguiente:

```
100 POS V = 60 %  
110 WAIT UNTIL INP 2 = 1 MAX 30 S  
120 POS V = 70 %
```

El robot se detiene en la posición definida por la instrucción de posición 100, hasta que la entrada 2 es igual a 1, si han transcurrido 30 s. y la entrada 2 sigue con el valor de 0 el robot se desplaza a la posición definida por la instrucción 120 y continua con la ejecución del programa.

OUTPUT

Esta instrucción conmuta el estado lógico de una salida durante la ejecución del programa, de esta manera cuando el programa pasa por esta instrucción la salida habilitada conmuta de estado.

```
50 POS V = 100 %  
60 SET OUTPUT 2  
70 RESET OUTPUT 4
```

Cuando el robot se encuentra en la posición definida por la instrucción 50, habilita la salida 2 con un valor lógico de 1, y la salida 4 con un valor lógico de 0.

JUMP

La instrucción JUMP cambia la secuencia del programa dependiendo de la estructura de la instrucción. El salto del programa puede ser de igual forma hacia adelante o hacia atrás dependiendo de la estructura de la instrucción. Esta instrucción es utilizada para cuando se desea controlar la secuencia del programa dependiendo de condiciones externas o internas al sistema.

Existen saltos condicionados o incondicionados; En un salto condicionado la secuencia del programa es desviada dependiendo de una condición en las señales de entrada, por ejemplo:

```
50   POS V = 100 %
60   JUMP TO 40 IF INP 4 = 0
70   POS V = 100%
```

Cuando la secuencia del programa llega a la instrucción 60, si la señal conectada a la entrada 4 es cero, la secuencia regresa a la instrucción 40 de otra manera continua con la instrucción 70. En un salto incondicional, cuando el apuntador del programa pasa por un salto incondicional, la secuencia del programa es dirigida a la instrucción indicada por el salto :

```
100  POS V = 90 %
110  JUM TO 150
120  POS V = 90%
```

CALL

El programa ejecutado se interrumpe para continuar en un subprograma o grupo de programas realizándolo completamente para regresar al programa interrumpido y seguir ejecutándolo. El programa puede ser cambiado mediante la instrucción CALL. El número de programa a ser llamado puede ser especificado de las siguientes maneras:

- Directo .
- Indirecto, mediante un valor numérico dentro de cualesquiera de los 200 registros.

Cuando esta instrucción se ejecuta dentro de un programa sucede cualquiera de las siguientes condiciones:

1. El programa que contiene la instrucción es interrumpido.
2. Se ejecuta el programa que ha sido llamado.
3. El programa principal continua su ejecución a continuación de la instrucción CALL.

RETURN

Esta es la última instrucción de un programa, o de un subprograma el cual fue llamado con la instrucción call. La instrucción regresa el programa ejecutado a:

- La primera instrucción dentro del mismo programa.
- El programa principal el cual fue detenido por una instrucción CALL, y continua su ejecución en la primer instrucción después de esta instrucción .

TCP

Cuando la instrucción TCP es ejecutada durante la secuencia programada, la representación interna de la posición del TCP puede cambiar pero la instrucción TCP por si misma no puede causar ningún movimiento en el robot, se cuenta con un total de 20 archivos para almacenar los datos de el TCP, de esta manera se puede trabajar en un mismo programa con dos o más herramientas instaladas en el mismo robot.

INTER

Esta instrucción interrumpe la ejecución del programa, de esta manera un programa puede ser interrumpido con la ayuda de una señal digital proveniente de un equipo periférico mediante la activación directa de entradas. Estas interrupciones pueden ser controladas para afectar o no la ejecución del programa durante los diferentes estados del ciclo de trabajo.

COORD

Cambia el sistema de coordenadas de la ejecución programada en cualesquiera de las opciones RECT, MODRECT O ROBOT.

- **Menú de instrucciones de posición.**

Esta constituido por instrucciones que determinan la posición, velocidad, trayectoria, etc. el menú de instrucciones de posición es el siguiente:

Cuando se oprime la tecla POSITIONING, se programa la posición de la herramienta a la posición que se movió previamente con el joystick. la velocidad programada adquiere un valor determinado, que será programada con la instrucción V%.

V %

La instrucción V% cambia el porcentaje de velocidad de una posición determinada. El porcentaje de velocidad va a variar en un rango de 0 a 799.9 %.

SAME

Si la posición a programar tiene los mismos argumentos que la anterior, se utiliza la instrucción SAME para repetir esta instrucción, de otra manera se seleccionan los nuevos argumentos que ha de adquirir la nueva posición.

ZONES

Existen cuatro alternativas para programar una zona. La zona dada por default para una instrucción es PATH. las otras alternativas son programadas mediante el uso de la instrucción ZONE y estas son :

FINE, CORNER1, CORNER2. De esta manera una posición puede adquirir los siguientes configuraciones :

100 POS = 100 % PATH
100 POS = 100 % FINE
100 POS = 100 % C1
100 POS = 100 % C2

La primera opción se utiliza para realizar una trayectoria, la segunda se utiliza para una posición seguida de una pausa, c1 se utiliza para trayectorias en una esquina corta, y c2 para curvas rápidas.

- **Menú de automático.**

Este menú permite la ejecución de los programas en modo automático, las siguientes son las instrucciones que componen a este menú.

PROGR ST

Ejecuta el programa cargado en memoria en forma progresiva y ascendente hasta que la última instrucción es ejecutada.

INST ST

Ejecuta el programa en forma ascendente y paso por paso, el robot se detiene después de que finaliza cada instrucción .

BWD

Ejecuta los movimientos de la herramienta instrucción por instrucción y en forma regresiva. Las instrucciones lógicas deben ser saltadas mediante la instrucción SKIP BW para poder continuar con la ejecución regresiva.

SIM

Simula las condiciones de entrada para la instrucción WAIT . Si el botón es oprimido cuando una instrucción WAIT es ejecutada, todas las condiciones se asumen como satisfactorias.

DISPL

Define una posición dentro de un programa ejecutado. En otras palabras, cuando un programa se está ejecutando, una posición determinada puede ser movida en cualesquiera de los tres ejes coordenados de base (X,Y,Z), el desplazamiento que se puede obtener es de 10 mm del punto original en cada ejecución del programa.

- **Menú de edición**

El menú de edición consta de las siguientes instrucciones.

V %

El porcentaje de velocidad en una instrucción puede ser cambiado mediante la instrucción V % cuando se esta editando un programa.

INST NO

Permite llamar un número de instrucción específico, sin tener que recorrer todo el programa hasta llegar a la instrucción deseada, con solo oprimir la tecla INST NO el sistema pregunta por el número de instrucción que se desea y esta será desplegada.

STEP

La función STEP es utilizada cuando por ejemplo, se requiere que el programa sea presentado instrucción por instrucción. Cuando STEP es presionado, da paso a otro menú en el cual aparecen las instrucciones FORW o BWD con las cuales las instrucciones serán desplegadas dependiendo de la selección.

MODPOS

Con esta instrucción se puede modificar una posición previamente programada, con solo llevar la herramienta a la nueva posición y en ésta, oprimir la instrucción MODPOS y el programa almacena la nueva posición.

MODIFY

Cuando se quiere cambiar una instrucción parcial o totalmente, es posible realizarlo mediante la instrucción MODIFY. Al oprimir esta tecla aparece otro menú con las siguientes instrucciones:

MODINST que modifica completamente la instrucción.

MODARG se modifica algún argumento de la instrucción excepto la posición y la velocidad.

DISPL La posición de una instrucción de movimiento puede ser cambiada mediante esta instrucción.

DELETE

La instrucción DELETE permite borrar una o más instrucciones de un programa. La secuencia de borrado es la siguiente:

Una vez llamado el programa, se ubica el cursor en la instrucción a partir de la cual se va a borrar, después de esto se oprime la instrucción DELETE indicando a continuación el número de instrucciones que serán borradas a partir de la posición del cursor.

INSERT

Esta instrucción permite insertar una instrucción dentro de un programa en el lugar deseado. Cuando una instrucción es insertada una nueva instrucción será numerada de acuerdo a lo siguiente:

- La instrucción insertada adquiere el siguiente múltiplo de diez después de la última instrucción donde se inserto la nueva.
- Si el siguiente múltiplo de diez esta ocupado, la instrucción o instrucciones insertadas son numeradas con los números vacantes un ejemplo se muestra a continuación:

PROGRAMA EXISTENTE	INSTRUCCIONES SERÁN INSERTADAS	PROGRAMA DESPUES DE LA INSERCIÓN
110 SET OUTP 10		110 SET OUTP 10
	POS	120 POS
	POS	130 POS
	POS	131 POS
140 WAIT		140 WAIT

PROG

Para escribir un programa, se inicia seleccionando el número de programa que se va a realizar utilizando la tecla PROG en el menú de edición después que el sistema del robot pregunta por el número de programa requerido. Cuando el número es especificado, la línea 10 es desplegada sin instrucciones si se trata de un nuevo programa, pero si el programa existe, aparece la primera instrucción del programa en cuestión.

RESEQ

Cuando esta opción es seleccionada dentro de un programa, todas las instrucciones que lo componen, adquieren una secuencia consecutiva de 10 en 10 para cada instrucción. Supóngase el programa del inciso 5.5.4.7 .

```

120 POS V = %
130 POS V = %
131 POS V = %
132 POS V = %
140 POS V = %

```

Después de la instrucción RESEQ el programa toma la siguiente estructura:

```

120 POS V = %
130 POS V = %
140 POS V = %
150 POS V = %
160 POS V = %

```

Cuando la instrucción RESEQ es ejecutada, las direcciones de saltos son también cambiadas para mantener la secuencia del programa.

COPY

Esta función copia un programa existente en la memoria de usuario con otro número de programa. Esta función se utiliza cuando dos o mas programas similares son utilizados en un mismo proceso, con solo programar uno y realizando las copias requeridas, se puede realizar las modificaciones necesarias sin tener que volver a programar.

ERASE

Con esta función, uno o mas programas dentro de la memoria de usuario pueden ser borrados. Cuando la función es seleccionada, los programas dentro de la memoria de usuario son presentados en el display para poder seleccionar el programa que se desea eliminar.

PROG NO

Todos los números de programas almacenados en memoria pueden ser desplegados mediante la selección de la función **PROG NO**. todos los programas utilizados son presentados en el display. El programa que se encuentra activo es marcado con un asterisco después del número de programa.

- **Menú de manual**

Este menú contiene instrucciones de manejo de diskettes, instrucciones para verificar el estado de entradas y salidas y definición de parámetros entre otros.

DISK

Esta instrucción se utiliza para el manejo de archivos entre la memoria del sistema y la unidad de disco.

DISK (FR DISK)

La transferencia de información de un diskette a la memoria de usuario del sistema del robot es designada como carga. Un bloque completo de programas o un programa individual puede ser cargado con la ayuda de la unidad de programación y con el uso de la instrucción **DISK (FR DISK)**

DISK (TO DISK)

Al igual que la instrucción anterior, la instrucción **DISK (TO DISK)** sirve para la transferencia de información pero en este caso la transferencia se realiza de la memoria de usuario del sistema del robot , hacia el diskette.

DISK (INIT)

La función **INIT** es utilizada para inicializar un diskette antes de ser utilizado en el sistema del robot, de otra manera el diskette no será reconocido por el sistema.

DISK (DELETE)

Esta función es utilizada para borrar un bloque de programas de un diskette.

IN / OUT

Las funciones IN / OUT son utilizadas para monitorear el estado de las entradas y salidas del sistema. Con la instrucción OUT se puede además modificar el estado de cualesquiera de las salidas.

LIST (PROG)

Una cola de impresiones de mas de cinco programas pueden ser requeridos y las impresiones son realizadas en el orden en que fueron demandados mediante la instrucción LIST.

LANG

Un cambio de lenguaje en el manejo de información puede ser seleccionado mediante la función LANG.

ERRORS

Mediante esta función se pueden desplegar mensajes de error ocurridos en el sistema del robot, de esta manera es mas fácil detectar fallas ocurridas en el sistema .

3.5 LA PROGRAMACIÓN.

Inicialización de una secuencia para un programa.

Es recomendable siempre que se va ha realizar un nuevo programa, se tenga el hábito para insertar la siguiente secuencia de instrucciones con los valores requeridos de inicialización en el comienzo de todo programa cuando se esta programando el robot. Esto asegura que las operaciones programadas tendrán los valores correctos desde el inicio.

10	V = (valor), MAX = (valor)	Velocidades básica y máxima
20	TCP (número)	Número de TCP requerido
30	RECT COORD OR ROBOT COORD	Selección del sistema de coord.

Si estos valores no son especificados, el sistema del robot utiliza los valores mas recientemente utilizados.

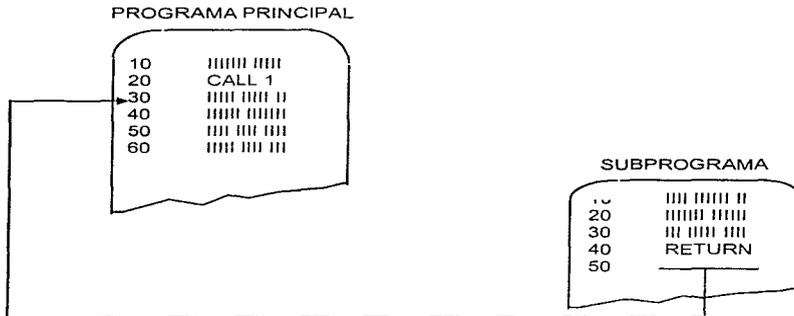
Estructura de un programa

Cada programa de robot tiene su propio número de programa y consiste de una serie de instrucciones que son numeradas dentro del programa. Los programas pueden incluir instrucciones de posición, instrucciones lógicas como son saltos, comparaciones, llamado de subprogramas, etc.

Los programas pueden ser numerados opcionalmente en un rango de 0 a 9999. Las instrucciones dentro del programa pueden ser numeradas en un rango de 1 a 65535 . El tipo de instrucciones determina cuantas pueden ser acomodadas dentro de la memoria en cualquier tiempo. De esta manera las instrucciones de posición ocupan un espacio considerablemente mayor que las instrucciones lógicas.

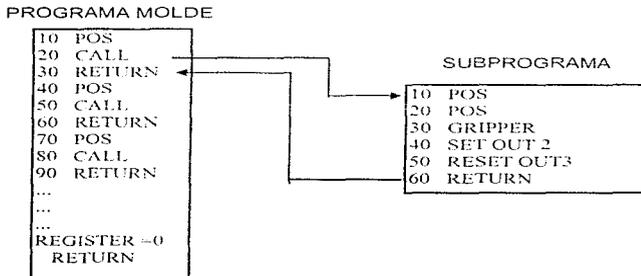
La construcción de un programa apropiado incluye un programa administrador que controla el trabajo del robot y llama los subprogramas para los diferentes subprocesos que son obtenidos repetitivamente durante un ciclo de trabajo.

Un programa principal con todos sus subprogramas asociados son designados como un bloque de programa. El programa número 0 debe ser utilizado como el programa principal si es posible.



Programa molde

Un programa molde es un subprograma que esta dividido en un número de módulos separados por la instrucción **RETURN**, cada módulo puede contener un número de posiciones para una tarea repetitiva donde la misma tarea es almacenada en un subprograma que es llamado por cada módulo del programa como se muestra en el ejemplo:



CAPITULO

4

4.- EL PANEL DE CONTROL

Para tener un control en la secuencia del proceso de aplicación de uretano, se utiliza un controlador lógico programable (PLC) al cual están conectadas todas las señales de entradas y salidas que intervienen en el sistema, el rack que contiene al PLC, la fuente de alimentación y los módulos de entradas y salidas, se encuentra dentro de un gabinete hermético para prevenir al sistema de cualquier daño. En la puerta del gabinete se encuentran instalados los botones de control y las lámparas indicadoras de secuencia; a continuación se muestra la figura del panel de control.

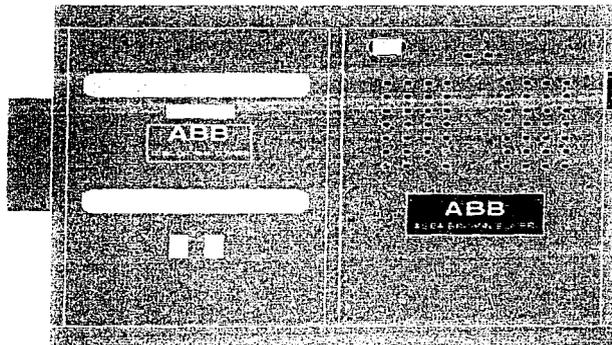


FIGURA 4.1. PANEL DE CONTROL

4.1 ENTRADAS

Todas las señales provenientes de botones pulsadores, fotoceldas y sensores de posición, son conectados a los módulos de entrada del PLC para ser procesados de acuerdo al programa de escalera dentro del procesador. Cada uno de los elementos de entrada están conectados a un nivel de voltaje de 24 V d.c. Las siguientes son las señales de entrada que intervienen en el proceso:

Descripción de la señal de entrada	Dirección correspondiente en el rack de entradas y salidas
Botón pulsador para selección de modo automático.	I : 000/01
Botón pulsador para selección de modo manual.	I : 000/02
Botón de retorno del soporte de cristales.	I : 000/03
Botón de avance del soporte de cristales.	I : 000/04
Botón de retorno del cargador de cristales.	I : 000/05
Botón de avance del cargador de cristales.	I : 000/06
Botón de retorno del ecualizador vertical .	I : 000/07
Botón de avance del ecualizador vertical.	I : 000/10
Botón de retorno del ecualizador horizontal.	I : 000/11
Botón de avance del ecualizador horizontal.	I : 000/12
Botón de energizado de copas de vacío.	I : 000/13
Botón de desenergizado de las copas de vacío.	I : 000/14
Botón para elevar las copas de vacío.	I : 000/15
Botón para bajar las copas de vacío.	I : 000/16
Botón de reset de modo manual / automático.	I : 000/17
Botón de puesta en marcha.	I : 001/00
Botón de restablecer fallas.	I : 001/01
Botón para silenciar la alarma	I : 001/02
Botones de paro de emergencia	I : 001/03
Selector de flujo o no flujo.	I : 001/04
Detector de retorno del cargador de cristales.	I : 001/05
Detector de avance del cargador de cristales.	I : 001/06
Detector de retorno del soporte de cristales.	I : 001/07
Detector de avance del soporte de cristales.	I : 001/10
Detector de retorno del ecualizador vertical.	I : 001/11
Detector de avance del ecualizador vertical.	I : 001/12
Detector de retorno del ecualizador horizontal.	I : 001/13
Detector de avance del ecualizador horizontal.	I : 001/14
Detector de cristales en la mesa de trabajo.	I : 001/15

4.2 SALIDAS

Cuando todas las instrucciones de entrada en un renglón del diagrama de escalera son verdaderas, se activa una señal de salida, dependiendo del componente que este conectado a el módulo de salida correspondiente, será la acción a realizarse, esto es activar un relevador, energizar una lámpara, energizar un solenoide de una válvula, etc.
A continuación se mencionan los componentes que intervienen en el proceso y que están conectados a los módulos de salida.

Descripción de la señal de salida	Dirección correspondiente en el rack de entradas y salidas
Lámpara indicadora de modo automático.	O : 000/01
Lámpara indicadora de modo manual.	O : 000/02
Lámpara indicadora de retorno del soporte de cristales.	O : 000/03
Lámpara indicadora de avance del soporte de cristales.	O : 000/04
Lámpara indicadora de retorno del cargador de cristales.	O : 000/05
Lámpara indicadora de avance del cargador de cristales.	O : 000/06
Lámpara indicadora de retorno del equalizador vertical.	O : 000/07
Lámpara indicadora de avance del equalizador vertical.	O : 000/10
Lámpara indicadora de retorno del equalizador horizontal.	O : 000/11
Lámpara indicadora de avance del equalizador horizontal.	O : 000/12
Lámpara indicadora de copas de vacío energizadas.	O : 000/13
Lámpara indicadora de copas de vacío desenergizadas.	O : 000/14
Lámpara indicadora de copas de vacío arriba.	O : 000/15
Lámpara indicadora de copas de vacío abajo.	O : 000/16
Lámpara indicadora de cristal en la mesa de trabajo.	O : 000/17
Bocina indicadora de falla.	O : 001/00
Solenoide de válvula de avance de soporte de cristales.	O : 001/01
Solenoide de válvula de retorno de soporte de cristales.	O : 001/02
Solenoide de válvula de avance de cargador de cristales.	O : 001/03
Solenoide de válvula de retorno de cargador de cristales.	O : 001/04
Solenoide de válvula de elevar copas de vacío.	O : 001/05
Solenoide de válvula de bajar copas de vacío.	O : 001/06
Solenoide de válvula de avance de equalizador vertical.	O : 001/07
Solenoide de válvula de retorno de equalizador vertical.	O : 001/10
Solenoide de válvula de avance de equalizador horizontal.	O : 001/11
Solenoide de válvula de retorno de equalizador horizontal.	O : 001/12

4.3 INTERFASE CON EL ROBOT

Como se mencionó en el capítulo 2 el PLC cuenta con un conector para ser utilizado como línea de comunicación y que esta etiquetado como REM I/O, este puerto es utilizado para la comunicación con otros controladores o computadoras de uso particular. El PLC reserva un espacio en el mapa de memoria para los dispositivos conectados a este puerto de tal forma que el procesador direcciona a los dispositivos conectados a este puerto como un rack de entradas y salidas local. De igual manera se mencionó en el capítulo 3 que el sistema de control del robot, cuenta con una tarjeta de comunicación serial denominada RIO que se utiliza para enlazar una comunicación con dispositivos externos, de esta manera para tener un control en la secuencia del proceso el sistema del robot es conectado a el PLC via serial utilizando la tarjeta de comunicación RIO conectada al REM I/O del PLC.

La tabla mostrada a continuación es el mapa de memoria para todas las señales de entradas y salidas que interactúan entre el controlador del robot y el PLC, la leyenda en la parte superior de cada uno de los recuadros indica la dirección correspondiente en el mapa de memoria del procesador PLC. Los números de la esquina derecha de los recuadros mayores indica la dirección de salida o entrada correspondiente al mapa de memoria en la tarjeta RIO del controlador del robot.

SALIDAS

I:041/00	I:041/01	I:041/02	I:041/03	I:041/04
BIT 1 DE ESTILO 1	BIT 2 DE ESTILO 2	BIT 4 DE ESTILO 3	BIT 8 DE ESTILO 4	RECONOCIMIENTO 5
RECONOCIDO	RECONOCIDO	RECONOCIDO	RECONOCIDO	DE PROGRAMA
I:042/00	I:042/01	I:042/02	I:042/03	I:042/04
ROBOT 30	PROGRAMA 31	CICLO 32	ROBOT EN 33	
PREPARADO PARA CICLO	EN PROCESO	TERMINADO	POSICION DE INICIO	34

ENTRADAS

O:041/00	O:041/01	O:041/02	O:041/03	O:041/04
BIT 1 DE 1	BIT 2 DE 2	BIT 4 DE 3	BIT 8 DE 4	
ESTILO	ESTILO	ESTILO	ESTILO	5
O:042/00	O:042/01	O:042/02	O:042/03	O:042/04
LECTURA 30	INICIO DE 31	ACTIVAR 32		34
DE ESTILO	CICLO	FLUJO		

MAPA DE MEMORIA DE ENTRADAS Y SALIDAS PARA LA TARJETA DIGITAL RIO

Las primeras cuatro señales de entrada, le indican al controlador el tipo de cristal que el PLC detecto mediante los sensores de cristal dependiendo de los bits activados, es el número de programa que el robot realizará, la señal de entrada 30 es una señal que activa el PLC para indicarle al controlador que esta colocando la información del tipo de programa que debe de

realizar, la señal 31 la activa el PLC cuando todas las condiciones del proceso están listas para la aplicación y le indica al controlador que de inicio al ciclo de trabajo, la señal 32 es activada cuando el operario selecciona si debe de existir flujo de sellador o no .

Las señales de salida 1,2,3 y 4 son activadas por el controlador como reflejo de la información de las entradas 1,2,3 y 4 y son enviadas al PLC para ser procesadas dentro del diagrama de escalera. La señal 5 es activada cuando el controlador ha detectado que existe la información necesaria del tipo de programa que se realizará. La señal 30 la activa el controlador cuando esta listo para iniciar un ciclo de trabajo, la señal 31 es activada cuando el robot esta realizando un ciclo, esta señal la procesa el PLC para impedir cualquier otra operación . La señal de ciclo terminado (32) es activada por el controlador para indicarle al PLC que el programa seleccionado se ha realizado . La señal 33 es activada para indicar al PLC que el robot se encuentra en posición de inicio.

Como se puede observar, las señales de salida del controlador del robot son señales de retroalimentación hacia el PLC dentro del protocolo de comunicación que existe entre el robot y el PLC, y que son utilizadas por este último para tener un control preciso en el proceso.

4.4 SECUENCIA DEL PROCESO DE APLICACIÓN.

La secuencia de trabajo en modo automático inicia cuando el operario coloca el cristal en el posicionador, al oprimir el botón de inicio de ciclo, el posicionador gira sobre su eje de tal forma que el cristal queda colocado en forma horizontal en la mesa de trabajo actuando el sensor de cristal indicando que se encuentra un cristal en la mesa para de esta manera continuar con el siguiente paso. Una vez detectado el cristal, se activa la válvula de avance del equalizador vertical, que tiene como finalidad centrar el cristal en su parte transversal. Después de que el cristal fue centrado por el equalizador vertical, se actúa la válvula de avance del equalizador horizontal, que va a centrar el cristal en forma longitudinal, de esta manera el cristal queda perfectamente centrado en sus dos ejes. En este momento se actúan las válvulas que controlan el avance de las copas de vacío, presionando el cristal para después accionar el vacío que se encargará de detener firmemente el cristal impidiéndole cualquier movimiento lateral o longitudinal, al mismo tiempo mediante unos sensores localizados en el recorrido de los equalizadores se detecta el tamaño del cristal para determinar si existe un parabrisas o no dentro del área de trabajo, si es detectado cristal, se procede con el siguiente paso. Una vez sujetado el cristal, se actúan las válvulas de retroceso de los equalizadores vertical y horizontal respectivamente. En este momento el PLC transfiere vía RIO la señal de inicio de programa al controlador del robot.

Una vez recibida la señal proveniente del PLC, el controlador del robot procesa las señales para identificar el programa que realizará, que es la trayectoria de aplicación de sellador a lo largo del perímetro del cristal, una vez realizada la aplicación el brazo robot regresa a su posición de inicio y el controlador envía una señal hacia el PLC indicándole que se ha realizado la aplicación.

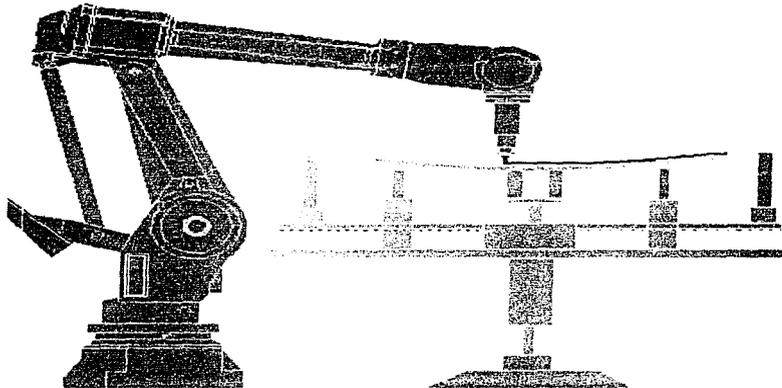


FIGURA 4.4.1.VISTA LATERAL DEL BRAZO DEL ROBOT Y LA MESA DE APLICACIÓN

CAPITULO

5

5.- PROGRAMACIÓN DEL ROBOT

En la sección 3.5, se explicó que la estructura de un programa de robot, es una serie de pasos consecutivos en forma ascendente y que el flujo de la trayectoria que sigue el programa puede ser truncada ó desviada por una instrucción de salto ó de llamado de subrutina. Al final de cada programa ó subrutina se debe de poner la instrucción RETURN para indicar al procesador que el programa se ha realizado y mandar al apuntador del programa al inicio del mismo, ó bien para indicar que la subrutina ha terminado, de esta manera un programa es ejecutado por el controlador como una secuencia ciclica, esto es, cuando el apuntador esta al final del programa, la instrucción RETURN envía al apuntador al inicio del programa y de esta manera se inicia un nuevo ciclo.

La memoria del controlador tiene la capacidad de almacenar hasta 64 kbytes de información, que corresponde a dos mil pasos aproximadamente que pueden ser distribuidos en diferentes números de programas. Debido a que la ejecución de un programa es ciclico, el diseño apropiado de un proceso debe de incluir un programa principal administrador que controle el trabajo del robot e interrumpa su secuencia ciclica condicionando la continuación de la secuencia a instrucciones específicas y realice el llamado de los subprogramas para cada uno de los diferentes subprocessos que pueden estar continuamente repitiendose durante el ciclo de trabajo, de tal manera que si no se cumple la condición, el robot se mantendrá siempre en una posición estática .

5.1 DISEÑO Y DESCRIPCIÓN DEL PROGRAMA ADMINISTRADOR.

En la sección 3.5 se habló de que es recomendable siempre iniciar un programa con las instrucciones que definen los valores de la velocidades básica y máxima, el número del TCP asignado a la herramienta que será utilizada en el programa en cuestión, y el tipo de coordenadas en las que se moverá el robot ya sea rectangulares ó de robot.

La velocidad del robot durante la ejecución del programa depende de la velocidad básica y la velocidad máxima, de la velocidad programada en la instrucción y el porcentaje de velocidad especificado en cada instrucción de posición. La velocidad de posición es especificada en cada instrucción como un porcentaje de la velocidad base.

De esta manera, las tres primeras instrucciones para el programa monitor son las siguientes:

```
10      V = 300 MM/S  MAX = 1000 MM/S
20      TCP 1
30      ROBOT COORD
```

En la primera instrucción, estamos designando la velocidad base de 300 MM/s y una velocidad máxima de 1000 MM/s, de tal manera que la velocidad máxima a la que trabajará el robot es del 333 % de la velocidad base.

La instrucción 20 designa el número de la herramienta que será utilizada, para este caso será la herramienta # 1, si el proceso requiriera del uso de más de una herramienta colocada en la muñeca del robot, se debe de definir el número de herramienta que se utilizará.

La instrucción 30 nos indica que la siguiente posición a la que el robot se moverá lo hará manipulando sólo un eje a la vez, esto es por que sólo se requiere desplazar el robot a la zona de aplicación utilizando sólo uno de los ejes, manteniendo todos los restantes en la misma posición.

Una vez que se programaron estos tres parámetros, se desplaza el robot utilizando el joystick a la posición que será denominada como posición de inicio, cuando el robot se encuentra en posición de inicio, se oprime el botón de instrucción de posición en la unidad de programación, en este momento el controlador almacena en memoria la posición que se le asigno, al entrar al menú de posición se le debe de asignar el porcentaje de la velocidad a la que se desplazará el robot , todas y cada una de las posiciones programadas son programadas como un porcentaje de la velocidad base, para esta instrucción se programa una velocidad del 100%, y la trayectoria que seguirá el robot es del tipo path o trayectoria.

```
40      POS V = 100 % PATH
50      SET OUTP 33
60      SET OUTP 30
```

Estando el robot en esta posición se tiene al robot en la posición de inicio, esto indica que esta listo para iniciar un ciclo de trabajo. En este momento es necesario indicarle al PLC la condición en la que se encuentra el robot, por lo tanto se activa la señal de salida No. 33 del mapa de memoria, esto lo realizamos en la instrucción 50 activando la instrucción SET del menú funciones. Como se observa en el renglón del diagrama de escalera del PLC, esta instrucción junto con las instrucciones de retorno del cargador, retorno del equalizador vertical, retorno del cargador horizontal y robot en posición de inicio, habilitan el bit de salida b3 / 10 que indica que todas las condiciones se encuentran en estado de inicio de ciclo y de esta manera el controlador del PLC esta listo para iniciar un nuevo ciclo.

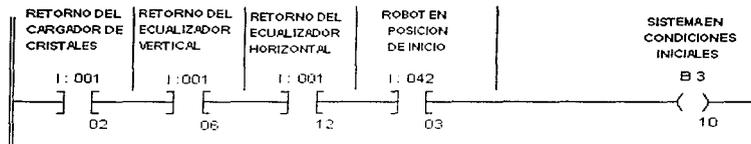


FIGURA 5.1.1. ESCALÓN DE PLC DE CONDICIONES INICIALES

En la instrucción 60, estamos activando la salida 30 (ROBOT PREPARADO PARA CICLO), esta señal nos permite, como vemos en el escalón del PLC, habilitar bits que den acceso a movimientos en forma automática, si esta instrucción no es habilitada, ningún movimiento en forma automática ó manual podrá realizarse en el caso específico del renglón mostrado, para poder activar la válvula del ecualizador vertical es necesario que la señal de salida 30 del robot este activada, el bit del sistema en modo automático este activado, el bit de las condiciones de avance ecualizador este activado y el bit de las condiciones para retroceso del ecualizador no esten dadas, esta última condición como se puede observar esta negada, y es requerida de esta manera para que el bit de retroceso de ecualizador pueda ser activado.



FIGURA 5.1.2. ESCALÓN DE PLC PARA RETROCESO DE ECUALIZADOR

Mediante la instrucción RESET en el paso número 70, es desactivada la señal de salida 32 que es activada para indicar que se ha terminado el ciclo de trabajo del robot y que será activada como se verá más adelante en el subprograma No 1, esta instrucción la recibe el PLC y es una indicación para que permita realizarse un nuevo ciclo. De la misma manera, en el paso número 80 desactivamos la señal 31 de ciclo en proceso que será activada por el robot cuando se encuentra en movimiento realizando el proceso de aplicación como medida de seguridad, esto es si el robot se encuentra en movimiento la señal esta presente dentro del programa de escalera y si mediante los sensores que se encuentran en el área, se detecta que una persona ha invadido la zona delimitada como área de trabajo, el programa del PLC ordena al robot detener su movimiento. Por lo anterior, estas últimas instrucciones quedan programadas de la siguiente manera:

70 RESET OUTP 32
80 RESET OUTP 31

La posición que fué programada en la instrucción 40, como se mencionó previamente es la posición de inicio, y el controlador esta esperando la señal de inicio de ciclo que será activada por el PLC cuando las condiciones sean las adecuadas, por lo tanto la siguiente instrucción será una instrucción de comentario que es la siguiente:

90 (* EN ESPERA DE CICLO *)

La instrucción 100, es una instrucción de salto condicionado, esto es, el programa desviará su trayectoria dependiendo de la instrucción de entrada

100 JUMP TO 120 IF INP 30 = 1

En esta instrucción el apuntador del programa saltará a la instrucción 120 si la instrucción de entrada 30 (LECTURA DE ESTILO) es activada por el PLC. Como se observa en el renglón la señal de salida es activada cuando el bit de condiciones de aplicación listas y la señal de inicio de ciclo son activadas, en este momento el PLC activa la señal de salida (LECTURA DE ESTILO) le esta indicando al controlador del robot que le va ha dejar la información del tipo de cristal que debe de ejecutar.



FIGURA 5.1.3. RENGLÓN DE ESCALERA LECTURA DE ESTILO

Si alguna de las dos condiciones de entrada no se cumple, la señal INP 30 proveniente del PLC con la dirección O:042/10 no se activará, y en consecuencia el apuntador del programa continuará con la instrucción 110.

110 JUMP TO 50

Esta instrucción regrasa el apuntador a las condiciones iniciales, ya que las condiciones para leer el tipo de cristal no se han dado, la instrucción es de salto del tipo incondicional.

120 FETCH R1 FROM PORT NO 11

Cuando la instrucción de entrada INP 30 ha sido activada, debido a que el ciclo de trabajo ha activado esta señal, el apuntador salta de la instrucción 100 a la instrucción 120. Esta instrucción le indica al controlador del robot que lea el valor del puerto 11 y que lo deposite en el registro R1, el puerto número 11 esta compuesto por las cuatro primeras señales de entrada de mapa de memoria y están configuradas en forma hexadecimal, este valor es decodificado por el controlador y su valor decimal es el almacenado en memoria del procesador para ser utilizado posteriormente. El valor del puerto 11 es el resultado de la instrucción de salida del PLC que se muestra a continuación:

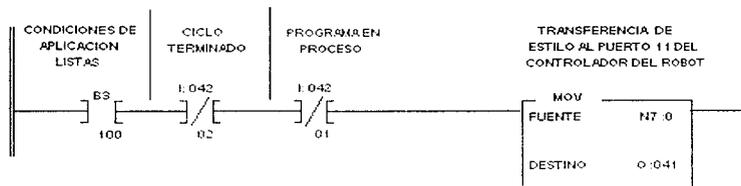


FIGURA 5.1.4. RENGLÓN PARA TRANSFERENCIA DE ESTILO

Como se puede observar la instrucción bit B3 / 100 además de que sirve para habilitar la señal de entrada INP 30 en los renglones anteriores, en conjunción con las señales OUTF 32 y OUTF 31 provenientes del controlador del robot y que fueron previamente reseteadas en las instrucciones 70 y 80, reflejadas como I:042/02 y I:042/01 respectivamente, activan la instrucción MOV, esta instrucción lo que realiza, como se explicó en el capítulo 2, es depositar el valor de la fuente N7:0 a un destino O:041 que no es otro que los cuatro primeros bits de entrada del mapa de memoria. El archivo N7:0 del programa del PLC fue previamente cargado con el valor prestablecido correspondiente al tipo de cristal que fue detectado por los sensores de posición, y que activan las salidas correspondientes. Al mismo tiempo que estas salidas son activadas, las señales de entrada activan la instrucción de movimiento seleccionada.

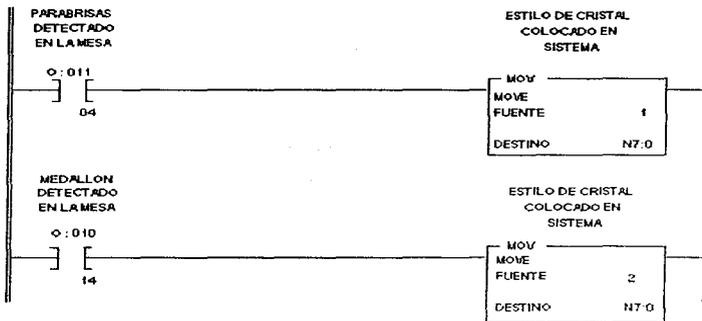


FIGURA 5.1.5. ESCALONES DE DETECCIÓN DE TIPO DE CRISTAL

Como se mencionó en el capítulo 3, el controlador del robot tiene la capacidad de almacenar en memoria más de un programa, para la aplicación en cuestión se cuenta con la opción de poder programar el robot para realizar la aplicación a medallones, por lo tanto los sensores de posición en conjunto con las instrucciones anteriores almacenan el valor de 1 en el registro N7:0 cuando los sensores detectaron un parabrisas y el valor de 2 cuando un medallón ha sido detectado, lo anterior es mostrado en los escalones de la figura 5.1.5.

Las siguientes dos instrucciones realizan un salto condicionado a la instrucción 120, en la primera, si el valor del registro R1 es igual a cero, y en la segunda si el valor del registro R1 es mayor de dos:

```

130   JUMP TO 120 IF R1 = 0
140   JUMP TO 120 IF R1 > 2

```

Cuando cualesquiera de las dos condiciones se da, se esta mandando a leer nuevamente del puerto 11 el valor correcto, ya que cualquier valor diferente de uno ó dos detectado previamente por los sensores en el programa del PLC, es un valor erroneo.

```

150   TRANSF R1 TO PORT NO 1
160   SET OUTP 25

```

En los pasos 150 y 160, en la primera instrucción, el controlador esta colocando el valor leído del puerto 11 al puerto 1, el puerto 1 esta compuesto por los primeros cuatro bits de salida del mapa de memoria, mismos que se ven reflejados en la palabra de entrada al PLC direccionada como I:041, la instrucción número 160 es habilitada para indicarle al PLC que el estilo de cristal ha sido reconocido. Estas dos instrucciones son utilizadas por el programa del PLC como lo muestra el siguiente renglón, para almacenar el valor leído por el puerto 11 del controlador del robot, en el archivo n7:1.



FIGURA 5.1.6. ESCALÓN PARA ALMACENAR ESTILO RECONOCIDO

La instrucción 170, es una instrucción de espera condicionada, en este momento el apuntador se detiene hasta que la instrucción de entrada número 31 (INICIO DE CICLO) es activada por el PLC.

170 WAIT UNTIL INP 31 = 1



FIGURA 5.1.7. ESCALÓN PARA INICIAR CICLO

La entrada 31 pertenece a la salida O: 042/01 en el mapa de memoria, en la figura anterior se observa el renglón correspondiente.

En este renglón se tiene como condiciones de entrada a la señal I : 041/04 que es la señal de reconocimiento de programa y que fué activada previamente por el controlador del robot en la instrucción número 160, la siguiente instrucción es una instrucción de comparación, en las que las fuentes A y B son los valores de los registros N7:0 Y N7:1. El registro N7:0 contiene el valor del tipo de programa que fué seleccionado por los sensores de tipo de cristal, y que fué colocado en el puerto de entrada número 11, el valor del registro n7:1 es el valor del puerto 11 y que mediante el registro R1 fué transferido al puerto 1 en la instrucción 150. Esto quiere decir que el PLC esta comparando si el valor que le envió al controlador del robot es el mismo que este recibio. Si esta condición se cumple y junto con la instrucción de reconocimiento de programa I:041/04, se activa la señal de salida O:042/01 .

Las instrucciones 180 y 190 , una vez que fué activada la señal de inicio de ciclo, son utilizadas para resetear las señales de salida OUDP 5 y OUDP 30. La primera desactiva la señal de reconocimiento de programa para permitir al PLC detectar un nuevo tipo de cristal en el siguiente ciclo, la segunda desactiva la señal de robot preparado indicando que ha sido activado para iniciar un nuevo ciclo.

180 **RESET OUDP 5**
190 **RESET OUDP 30**

Una vez que la señal de inicio de ciclo fué activada, indica que todas las condiciones para realizar la aplicación estan dadas, por lo tanto la siguiente instrucción es la encargada de llamar al subprograma que realizará la aplicación a los cristales .

200 **CALL PROG (R1)**

Esta instrucción hace un llamado de programa el cual estará determinado por el valor del registro R1 que previamente fué cargado con el valor leído del puerto 11, cuando el programa del registro R1 ha sido realizado, el apuntador regresa a la instrucción inmediata siguiente de donde fué llamado el subprograma, y la siguiente instrucción es la instrucción de retorno, esta instrucción regresa al apuntador al inicio del programa, terminando de esta manera el ciclo de trabajo.

210 **RETURN.**

Resumiendo, el programa monitor ó supervisor queda estructurado de la siguiente manera:

10	V = 300 MM/S MAX = 1000 MM/S	" Condiciones iniciales "
20	TCP 1	" Herramienta número 1 "
30	ROBOT COORD	" Coordenadas de Robot "
40	POS V = 100 % PATH	" Posición de inicio "
50	SET OUTP 33	" Señal posición de inicio on "
60	SET OUTP 30	" Señal robot preparado on "
70	RESET OUTP 32	" Señal ciclo terminado off "
80	RESET OUTP 31	" Señal programa en proceso off "
90	(* EN ESPERA DE CICLO *)	" Comentario "
100	JUMP TO 120 IF INP 30 = 1	" Lee estilo "
110	JUMP TO 50	" Retorno a leer estilo "
120	FETCH R1 FROM PORT NO 11	" Bits de estilo en puerto 11 "
130	JUMP TO 120 IF R1 = 0	" Error de estilo "
140	JUMP TO 120 IF R1 > 2	" Error de estilo "
150	TRANSF R1 TO PORT NO 1	" Bits de estilo en puerto 1 "
160	SET OUTP 5	" Programa reconocido on "
170	WAIT UNTIL INP 31 = 1	" Inicio de programa "
180	RESET OUTP 5	" Programa reconocido off "
190	RESET OUTP 30	" Robot preparado off "
200	CALL PROG (R1)	" Llamado de programa "
210	RETURN.	" Retorno a inicio de prog. "

5.2 DISEÑO Y DESCRIPCIÓN DEL PROGRAMA DE APLICACIÓN

En la sección anterior se describió el programa monitor, en este se mencionó que el robot se mantendrá estático hasta que las condiciones de inicio de ciclo sean activadas mediante las señales provenientes de PLC. Cuando ha detectado estas condiciones, el programa monitor en la instrucción 200 hace un llamado al subprograma 1 que es el encargado de realizar la aplicación de uretano a los cristales. A continuación se describe el programa 1 .

Como se mencionó en la sección 3.5 , y de la misma manera que se realizó en el programa cero, el programa 1 se debe de iniciar con los parámetros que definan a la herramienta, las velocidades básica y máxima, y la manera en la que el robot realizará sus movimientos ya sea en coordenadas de robot o coordenadas rectangulares, de tal forma que las tres primeras instrucciones serán como se muestra.

```
10      V = 300 MM/S  MAX = 1000 MM/S
20      TCP 1
30      ROBOT COORD
```

Además de estas tres instrucciones existe otra mas que debe de ser definida, tal instrucción esta relacionada con la tarjeta analógica que esta conectada en el rack del controlador del robot y su función es determinar el valor del voltaje analógico que pondrá en sus terminales de salida.

```
40      GLSP : 10V = 400 MM/S
```

Esta instrucción lo que determina es el valor de voltaje correspondiente a la velocidad prefijada, esto es el voltaje máximo en la salida de la tarjeta analógica será de 10Va.c. cuando la velocidad de la herramienta sea igual a 400 MM/S.

La posición de inicio del robot es una posición estática en un costado de la mesa de aplicación, siempre que se inicia un ciclo el robot se trasladará a el centro de la mesa para iniciar la trayectoria de aplicación y cuando esta termina, se desplazará a el punto de inicio, el punto de inicio es el que muestra la figura:

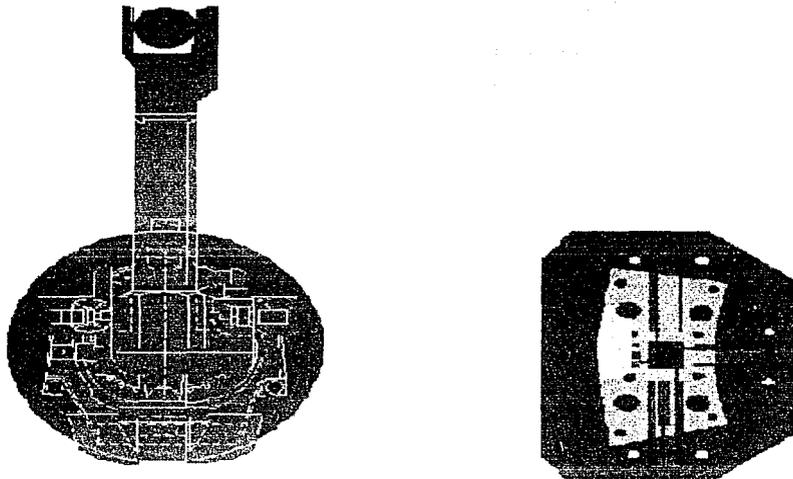


FIGURA 5.2.1. ROBOT EN POSICIÓN DE INICIO

Esta posición es definida por la instrucción 50, cuando el robot va a iniciar la aplicación se va a trasladar de la posición A a la posición B, en este momento activamos la señal de salida número 31 que corresponde a la instrucción PROGRAMA EN PROCESO, esta señal indica que el robot está a punto de realizar movimientos y se debe de contar con las precauciones necesarias para evitar un accidente, esta salida la activamos en la instrucción 60.

```
50   POS V = 100 % PATH
60   SET OUTP 31
```

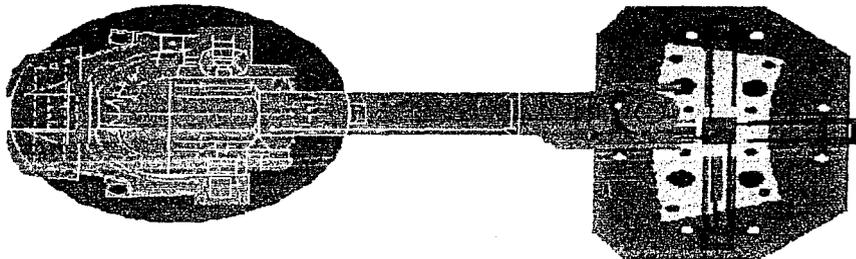


FIGURA 5.2.2. ROBOT SOBRE EL CRISTAL

La instrucción 70 es otra instrucción de posición en este caso esta instrucción es utilizada para desplazar el robot a la posición de aplicación, como lo muestra la figura 5.2.2. En este momento el robot esta listo para iniciar la trayectoria de aplicación a lo largo de todo el contorno del cristal.

70 POS V = 100 % PATH

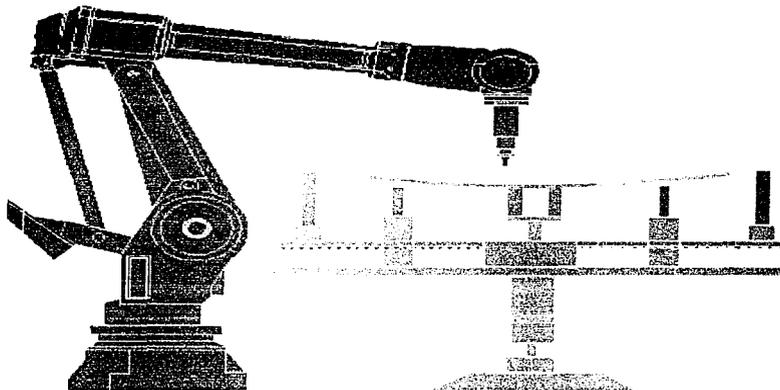


FIGURA 5.2.3. ROBOT SOBRE EL CRISTAL (VISTA LATERAL)

La posición que muestra la figura 5.2.3 anterior vista desde un plano lateral, mostraría que el robot esta posicionado a una distancia del cristal de aproximadamente 30 cm., para que se pueda llevar a cabo la aplicación del sellador es necesario que el robot describa la trayectoria a lo largo del cristal a una distancia de separación del mismo de aproximadamente 1 cm., por lo tanto la instrucción 80 es para desplazar el robot en el eje Z, manteniendo la misma posición en el plano X,Y.

80 POS V = 50 % FINE

Podemos ver que en este caso la velocidad se disminuyó al 50%, y el tipo de trayectoria es fine, esto es debido a que se requiere que el robot se desplace mas lento para evitar un daño en el cristal. Una vez que el robot se encuentra en esta posición, se modifica el modo de movimiento de coordenadas de robot a coordenadas rectangulares, ya se explicó en el capítulo 3 que los movimientos en coordenadas rectangulares dan una mayor precisión y velocidad al robot en trayectorias críticas como por ejemplo en curvas, el cambio de tipo de movimiento se realiza con la instrucción 90.

90 RECT COORD

La siguiente instrucción es un salto condicionado, en esta instrucción se esta condicionando la continuidad del programa a la señal de entrada número 32, esto es si se ha seleccionado que el ciclo se debe de realizar con flujo, la señal de entrada INP 32 es válida por lo tanto el apuntador continúa con la siguiente instrucción, si el valor de la señal INP 32 fuera cero , el apuntador del programa saltaría a la instrucción 120.

100 JUMP TO 120 IF INP 32 = 0

Si el operario selecciona que el ciclo debe de realizarse con aplicación de sellador, se debe de activar la señal de salida OUTP 9, esta salida pertenece al bloque de señales digitales alambrada, esto es la salida OUTP 9 esta conectada directamente al dispositivo que controla el flujo de sellador, cuando el controlador de sellador detecta esta señal, manda a abrir la válvula de control y de esta manera se inicia la aplicación de sellador.

110 SET OUTP 9

A partir de este momento se procede a programar cada uno de los puntos que describirán la trayectoria de aplicación de sellador al cristal, cada uno de estos puntos son programados llevando el robot a la posición deseada mediante el joystick, una vez en posición el robot, se oprime el botón de posición y los argumentos de velocidad y tipo de trayectoria quedan definidos por default y sólo cuando en algún punto específico se desea modificar estos argumentos, se realiza cambios a tales argumentos, en caso contrario el controlador queda listo para continuar con la siguiente instrucción. Todo el resto de las instrucciones y hasta la instrucción número 540 son instrucciones de posición y que fuerón programadas a distancias cortas para dar una mayor definición en el recorrido de la herramienta de aplicación, por lo tanto sólo se mostrará estas instrucciones mediante el diagrama siguiente.

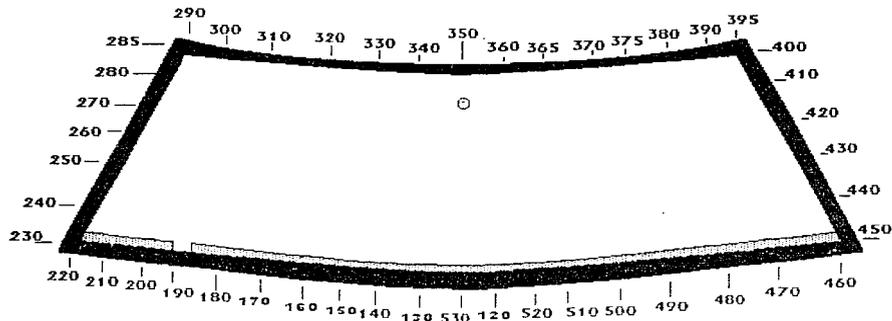


FIGURA 5.2.4. DIAGRAMA DE PUNTOS DE TRAYECTORIA DEL ROBOT

Cuando la aplicación de sellador se ha realizado en todo el contorno del cristal, es necesario desactivar la válvula para que se deje de proveer sellador, esto lo realizamos mediante la instrucción RESET OUTP 9, al mismo tiempo despegamos la punta del aplicador respecto al

crystal, esto lo hacemos elevando el robot en el eje Z, esta posición le corresponde a la instrucción 560

550 **RESET OUTP 9**
560 **POS V = 100 % PATH**

El siguiente punto a donde se desplazará el robot es el punto de inicio, despues de programar este punto, es necesario activar la señal OUTP 32 para indicar a el controlador del robot al PLC que el ciclo ha concluido

570 **POS V = 100 % PATH**
580 **SET OUTP 32**

Por último se programa la instrucción de RETURN, en este momento el apuntador regresa a la instrucción siguiente de donde fué llamado y queda listo para reiniciar un nuevo ciclo.

590 **RETURN.**

CONCLUSIONES

Contrario a lo que pudiera pensarse, la automatización en las industrias, no llegó acompañada de reajustes de personal o de cambios organizacionales, sino de nuevas contrataciones pues es necesario incorporar ingenieros y técnicos con conocimientos en el área electrónica y de sistemas para poder manipular y programar estos robots, igualmente para poder realizar los mantenimientos preventivos y correctivos necesarios para asegurar la continuidad en los procesos.

En nuestro país existe poco desarrollo en esta área, en consecuencia de esto, compañías extranjeras son las solicitadas por la industria para la capacitación de su personal, o lo que es peor, se contratan sus servicios para realizar la programación de los robots para nuevos procesos.

El costo para programar un robot que realice la aplicación de sellador a parabrisas, diseñado por una compañía local, es de \$ 12,000 dólares aproximadamente. (ver cotización anexa) mientras que el costo de diseño de un programa para la misma aplicación realizado por un empleado de la compañía que cuente con estos robots, asciende a el total de una semana de salario del empleado, ya que el tiempo necesario para diseñar y realizar este programa es de cinco días, si tomamos en cuenta que el salario de este empleado es de \$ 8,000 pesos por mes, el costo total del diseño es de \$2,000 pesos que equivale a \$ 250 dólares al tipo de cambio al mes de noviembre de 1997. Esto trae un ahorro de \$ 9750 dólares para la empresa.

Cabe hacer mención que el ahorro efectuado por la programación fue considerable y, demuestra la capacidad de nuestra gente para realizar trabajos característicos de países desarrollados; además de las divisas que son retenidas en nuestro país que tan importantes resultan en la economía que actualmente tenemos.

Por último, tenemos que quitarnos la idea de que el robot es una máquina que desplaza al ser humano, esto no es posible simple y sencillamente por dos razones, no piensa y por tanto no razona, características de la mente humana la cual es tan compleja que ni la máquina más avanzada podría igualarlas.

CHYRSLER DE MEXICO, S.A. DE C.V.
LAGO ALBERTO 320
COL. ANAHUAC
11320 MEXICO, D.F.

ATENCION: SR. ING. HUMBERTO MONTAYA

ESTIMADOS SEÑORES:

EN RELACION A NUESTRA COTIZACION SC97169 DE 7 DE OCTUBRE DE 1997 POR LA FABRICACION DE 2 MESAS PARA ROBOTS DE APLICACIÓN DE URETANO, DESGLOSAMOS EL PRECIO DE LA PROGRAMACION DE LOS ROBOTS POR SEPARADO.

PRECIO DE LAS MESAS: S 46,812.00 DLLS C/U.

PRECIO DE LA PROGRAMACION: S 12,173.00 DLLS C/U

EN CASO DE QUE LA PROGRAMACION LA REALIZE EL PERSONAL DE CHRYSER O ALGUNA OTRA EMPRESA, DESCONTAR LA CANTIDAD INDICADA.

SIN MAS POR EL MOMENTO, ENVIAMOS A USTED UN CORDIAL SALUDO.

ATENTAMENTE,

P.A. 

ING. ARMANDO ROMERO MARISCAL RENSTROM
DIRECTOR GENERAL.

BIBLIOGRAFIA

- 1.- - Manual de programación para controladores lógicos programables.
- Allen Bradley
1995.
- 2.- - Service Manual for Robot Control System S3.
- Asea Brown Boveri.
1994.
- 3.- - S3 M92 Training Manual
- Asea Brown Boveri.
1994.
- 4.- - Service Manual
-Asea Brown Boveri.
1994.
- 5.- -Pro-Meter Urethane System Manual.
-Nordson Corporation.
-1996.