



“CONTROL POR COMPUTADORA PARA BRAZO ROBOT”

Trabajo de Graduación Preparado Para la Facultad de Ingeniería

Escuela de Estudios Tecnológicos

Para Optar al Grado de

TÉCNICO EN INGENIERÍA ELECTRÓNICA

Por

CARLOS ERNESTO CABRERA GONZÁLEZ 9501034

EDUARDO RIVERA 9401062

Marzo 1998

Soyapango- El Salvador- Centroamérica



“CONTROL POR COMPUTADORA PARA BRAZO ROBOT”

Trabajo de Graduación Preparado Para la Facultad de Ingeniería

Escuela de Estudios Tecnológicos



Para Optar al Grado de

TÉCNICO EN INGENIERÍA ELECTRÓNICA

Por

CARLOS ERNESTO CABRERA GONZÁLEZ 9501034

EDUARDO RIVERA 9401062

Marzo 1998

Soyapango- El Salvador- Centroamérica

UNIVERSIDAD DON BOSCO

RECTOR

ING. FEDERICO HUGUET RIVERA

VICE-RECTOR ACADÉMICO

LIC. BALTAZAR DÍAZ MUÑOZ

SECRETARIO GENERAL

PBRO. PEDRO JOSÉ GARCÍA CASTRO

ASESOR DEL TRABAJO DE GRADUACIÓN

ING. OSCAR REINALDO VILLALTA LARA

JURADO EXAMINADOR

ING. HERBERT ISRAEL CARDONA

ING. FEDERICO ANTONIO LAÍNEZ

INDICE

	Pag.
Introducción	7
Objetivos	8
1. Marco Teorico	9
1.1 Motores paso a paso	9
1.1.1. Principio de funcionamiento	10
1.1.2. Clasificación de los motores paso a paso	11
1.2. Control por computadora	17
1.4. Modos de operación	17
1.3. Control de la trayectoria	19
1.5. Control punto a punto	19
1.6. Programación textual Explícita	26
1.7. Programación textual Específica	29
1.8. Clases de robot.....	30
1.9. Elementos que componen un robot industrial	31
2. Software	33
2.1. Descripción del software	33
2.1.1. Archivo	34
2.1.1.1. Abrir	35
2.1.1.2. Guardar	36
2.1.1.3. Nva. Sec.	36
2.1.1.4. Guardar como	38
2.1.1.5. Modificar	39

2.1.2 Ejecutar	40
2.1.2.1. Mov. Por Eje	40
2.1.2.2. Mov. Simultáneo.....	41
2.1.2.3. Ejecución de Sec.....	42
2.1.2.4. Mov. cíclico	43
2.1.3. DOS prompt	43
2.1.4. Salir	44
2.1.5. Ayuda	44
2.1.5.1. Tutorial	44
2.1.5.2. Generalidades	45
2.1.5.3. Glosario	46
2.2. Listado de comandos del robot	
2.2.1. Zero	46
2.2.2. Move	47
2.2.3. Here	47
2.2.4. Position	47
2.2.5. Goto	48
2.2.6. Nest	48
2.2.7. Grip Close	48
2.2.8. Grip Open	48
2.2.9. Speed	48
2.2.10. Delay	49
2.2.11. Limite.....	49
3. Hardware	50

3.1. Estructura mecánica del brazo.....	50
3.1.1. Base	50
3.1.2. Espalda	51
3.1.3. Antebrazo	51
3.1.4. Elevación de la muñeca.....	51
3.1.5. Rotación de la muñeca	51
3.1.6. Pinzas	52
3.2. Modalidades de utilización	53
3.2.1. Modo self test	53
3.2.1.1. modalidad 1.....	53
3.2.1.2. modalidad 2	55
3.3. Modo ordenador personal	56
3.4 Cables para las conexiones ROBOT- ORDENADOR.....	56
3.5. Diagrama del circuito interno del robot	60
3.6. Mapeo de memoria	63
3.7. Descripción del programa monitor	63
3.7.1. Caso A,B,E, F,I,J,K,Q,R,T,U,V,W,X,Y	65
3.7.2. Caso C	66
3.7.3. Caso D	66
3.7.4. Caso G	66
3.7.5. Caso H	67
3.7.6. Caso L1,L0	67
3.7.7. Caso M	68
3.7.8. Caso N	68

3.7.9. Caso O	68
3.7.10 Caso P.....	68
3.7.11. Caso S	69
3.7.12. Caso Z	69
3.7.13. Caso (+),(-)	69
3.7.14. Caso (,);(,);(espacio en blanco)	70
4. Mejoramiento de la Versatilidad del robot	71
4.1. Características del mecanismo	72
4.2. Descripción del circuito para control de motores y	
Forma de conectarlo	74
4.2.1. Descripción	74
4.2.2. Forma de conexión	76
4.3. Modificación en el software.....	77
CONCLUSIONES	80
BIBLIOGRAFIA	81
ANEXO A	82
ANEXO B	98
ANEXO C	160
ANEXO D	165

INTRODUCCION

El presente documento, incluye una base teórica referente al tema de los robots concierne, motores paso a paso, su clasificación, principio de funcionamiento, control de la trayectoria, etc. y otros aspectos relacionados con el ámbito, esto se ha hecho con dos fines, el primero como introducción a cualquier persona que este interesada en el tema y segundo, incluir algunas de las ideas en la ayuda del programa, específicamente en la opción generalidades del submenú ejecutar.

El segundo capítulo se refiere al software, es decir que en él se explican cada una de las alternativas del menú principal y de los submenús, así como también se agrega un listado de los comandos que utiliza el robot, en lenguaje Qbasic, para interactuar con cualquier computadora personal a través del puerto paralelo de una máquina en la que se pueda correr el programa.

El tercer capítulo trata acerca del hardware, aquí se incluye teoría acerca de la estructura del robot, así como una descripción sobre el funcionamiento de la microcomputadora que controla el autómata.

El cuarto y último capítulo consta de una explicación de cómo evitar el control que ejerce el up Z80 en los motores que logran el movimiento en el brazo robot, con el fin de tener un control más directo de los motores.

También se agregan en forma de anexos el programa monitor en lenguaje ensamblador del brazo robot, además del código fuente del proyecto desarrollado en Qbasic, hasta el momento de la entrega del documento.

OBJETIVOS

- 1.- Exponer el programa de control del brazo robot; haciendo énfasis en las opciones fundamentales de éste, tales como lo son las opciones referentes al movimiento de el autómata, operación, y control.

- 2.- Demostrar de forma práctica el funcionamiento del brazo robot, utilizando las opciones del programa principal creado en QBASIC.

- 3.- Presentar una amplia descripción acerca del software y el hardware que involucra el proyecto, así como también información de cómo estos interactúan entre sí.

1. MARCO TEÓRICO

A continuación, se presenta la teoría básica que se necesita para tener una orientación en cuanto a sistemas robóticos. Se presenta una secuencia progresiva en cuanto a la presentación de la teoría, a fin de introducir al lector poco a poco en el área que concierne al presente anteproyecto.

1.1. MOTORES PASO A PASO

En situaciones en las que se requiere un control preciso de la trayectoria a seguir por la mano o la herramienta de un robot manipulador, es más sencillo y económico usar motores paso a paso que motores de corriente continua con retroalimentación.

El único problema que presentan los motores de paso a paso es una potencia limitada. Sin embargo, en cierto sentido, este problema está siendo resuelto por los nuevos diseños, con los que se ha logrado potencias superiores a 1 CV (caballo de vapor), suficientes en muchos casos para los robots comerciales de tamaño medio.

El motor paso a paso es un elemento capaz de transformar pulsos eléctricos (información digital), en movimientos mecánicos; el eje del motor gira un determinado ángulo por cada impulso de entrada, el resultado de este movimiento, fijo y repetible es un posicionamiento preciso y fiable.

Un motor de paso a paso puede girar, en ambos sentidos, un número exacto de grados con incrementos mínimos determinados por el diseño. Este incremento queda comprendido entre 0.72 grados y 90 grados, correspondiente a 500 y 4 pasos por revolución, respectivamente.

Aunque el motor de paso a paso es de concepción antigua, sólo se ha empleado, en la práctica, a partir de la aparición de los modernos semiconductores electrónicos que implementan los circuitos adecuados para la actuación y el control.

1.1.1. PRINCIPIO DE FUNCIONAMIENTO

El principio básico de funcionamiento, está basado en las fuerzas de atracción y de repulsión ejercidas entre polos magnéticos.

Con referencia a la figura 1, y teniendo en cuenta que los polos magnéticos del mismo signo se repelen, si los bobinados del estator A se alimenta de tal manera que se

comporta como un polo norte, y el estator B se comporta como un polo sur, el rotor imantado(imán permanente), si es giratorio, se mueve hasta una posición de 180 grados de la mostrada.

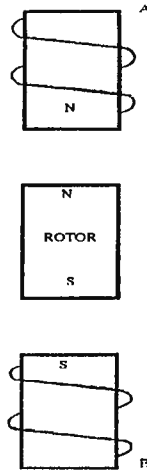


figura 1

Si el estator consta de cuatro polos como se muestra en la figura 2(página 10), formados por electroimanes y un rotor común de imán permanente, al cambiarse la polaridad de los polos del primero, mediante un control externo con la secuencia indicada, el segundo giraría en sentido contrario a las agujas del reloj, con incrementos de 90 grados. Claramente un motor con incrementos de 90 grados no representa un dispositivo práctico de control, por lo que el diseño se ha modificado y afinado para obtener incrementos mucho más pequeños.

Para permitir una mejor resolución por paso, se añaden más polos al estator; además, en dichos polos se mecanizan una serie de dientes al igual que en el rotor. En la figura 3 (página 11), se muestra la configuración de los dientes del rotor y del estator, para alcanzar una resolución de 1.8 grados por paso.

En la figura 4 (página 12), se presentan las distintas partes que componen un motor de paso a paso. Se aprecia, claramente el diseño característico del rotor y del estator.

1.1.2. CLASIFICACION DE LOS MOTORES PASO A PASO

a.- De imán permanente: De los tres tipos de motores paso a paso que existen, el de imán permanente está formado por un estator de forma cilíndrica, con un cierto número de bobinados alimentados en secuencia, que crean un campo magnético giratorio de manera discontinua. El rotor, concéntrico con el estator y situado sobre el eje, contiene un imán permanente magnetizado, que en cada instante tenderá a alinearse con el campo magnético creado.

b.- De reluctancia variable: El estator presenta la forma habitual, con un número determinado de polos electromagnéticos. Sin embargo, el rotor no es de imán permanente, sino que está formado por un núcleo de hierro dulce de estructura cilíndrica, pero con un cierto número de dientes tallados longitudinalmente a lo largo de su superficie lateral.

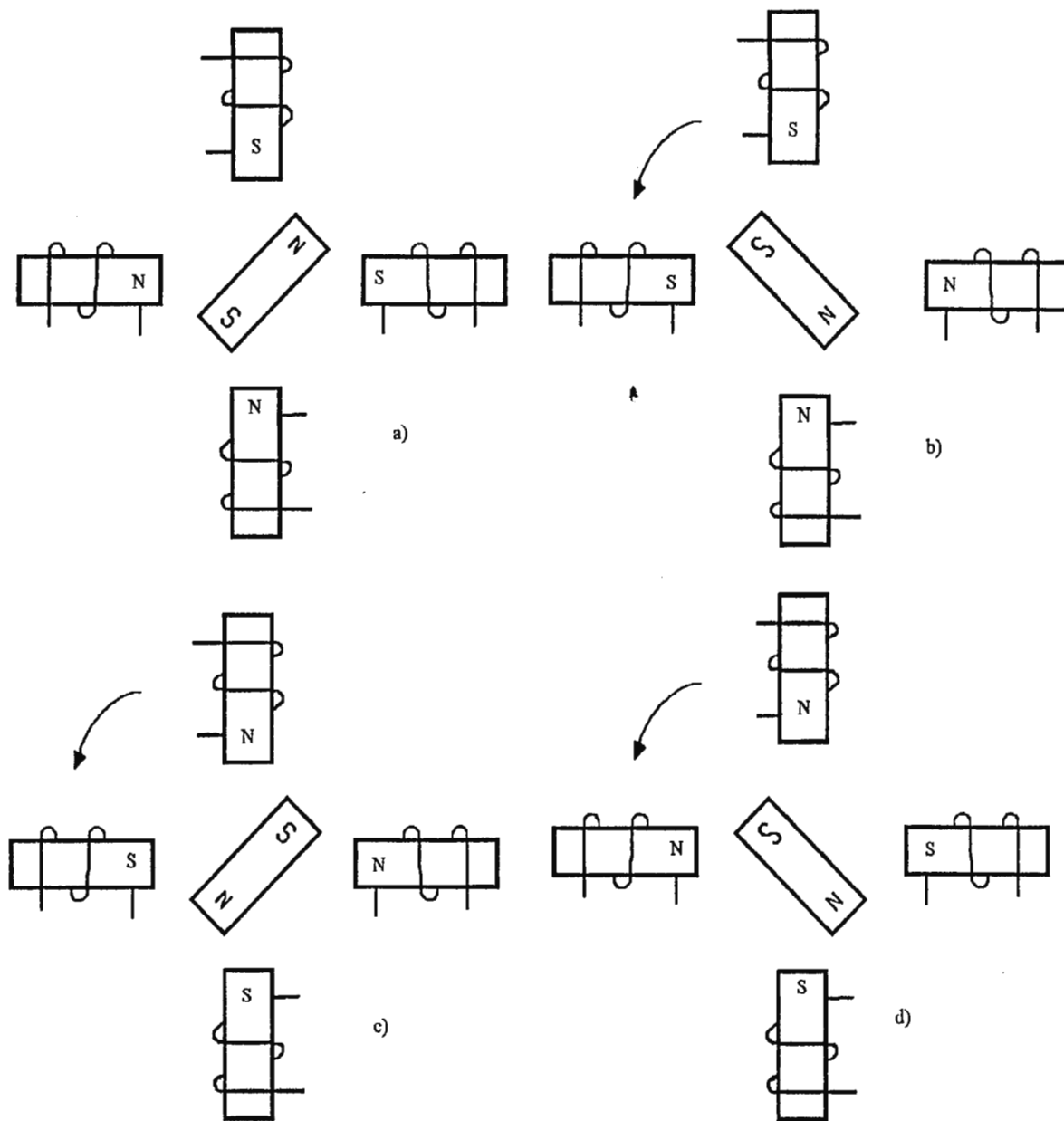
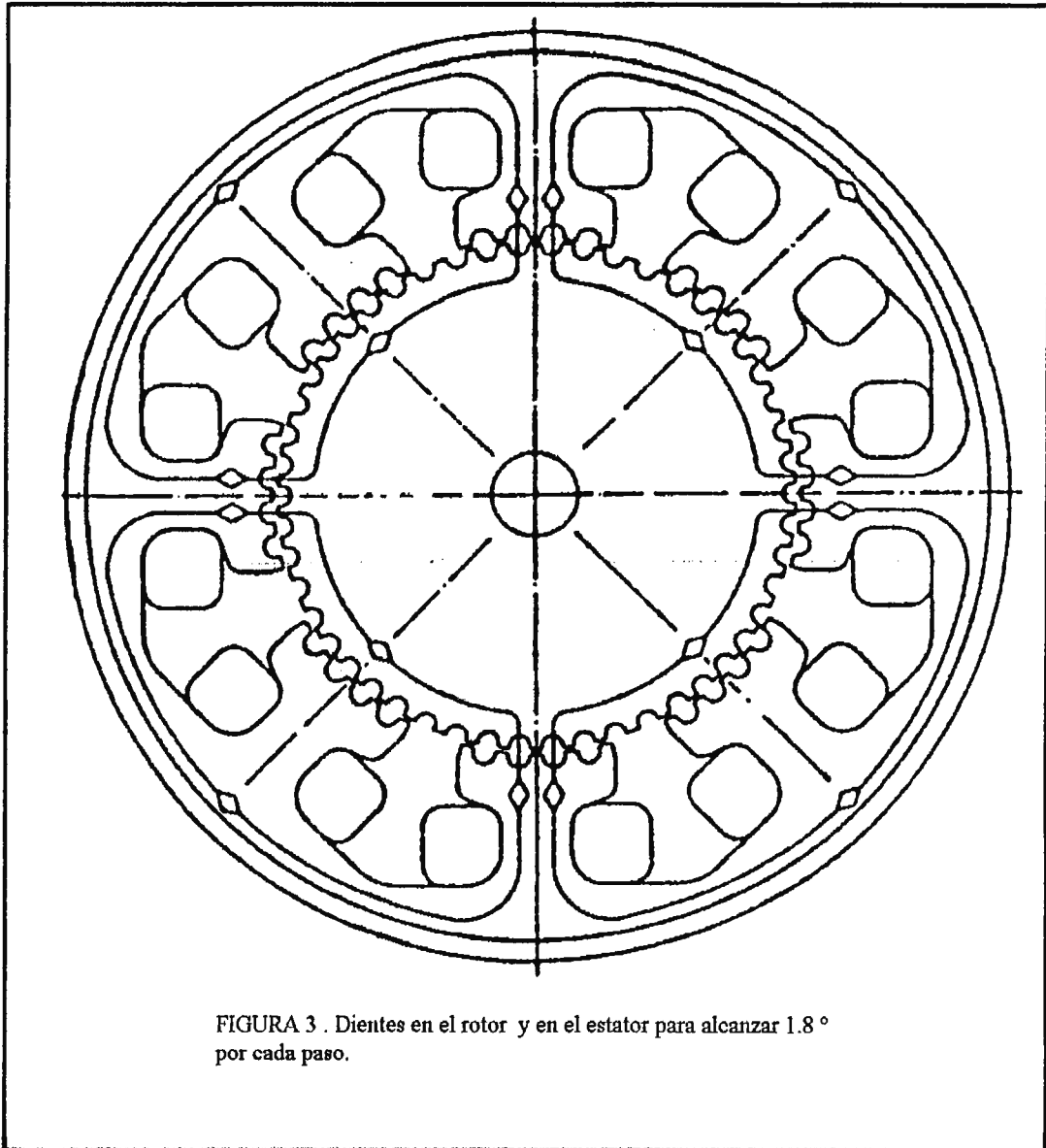


FIGURA 2. Principio de funcionamiento de un motor de paso a paso de imán permanente con dos fases.



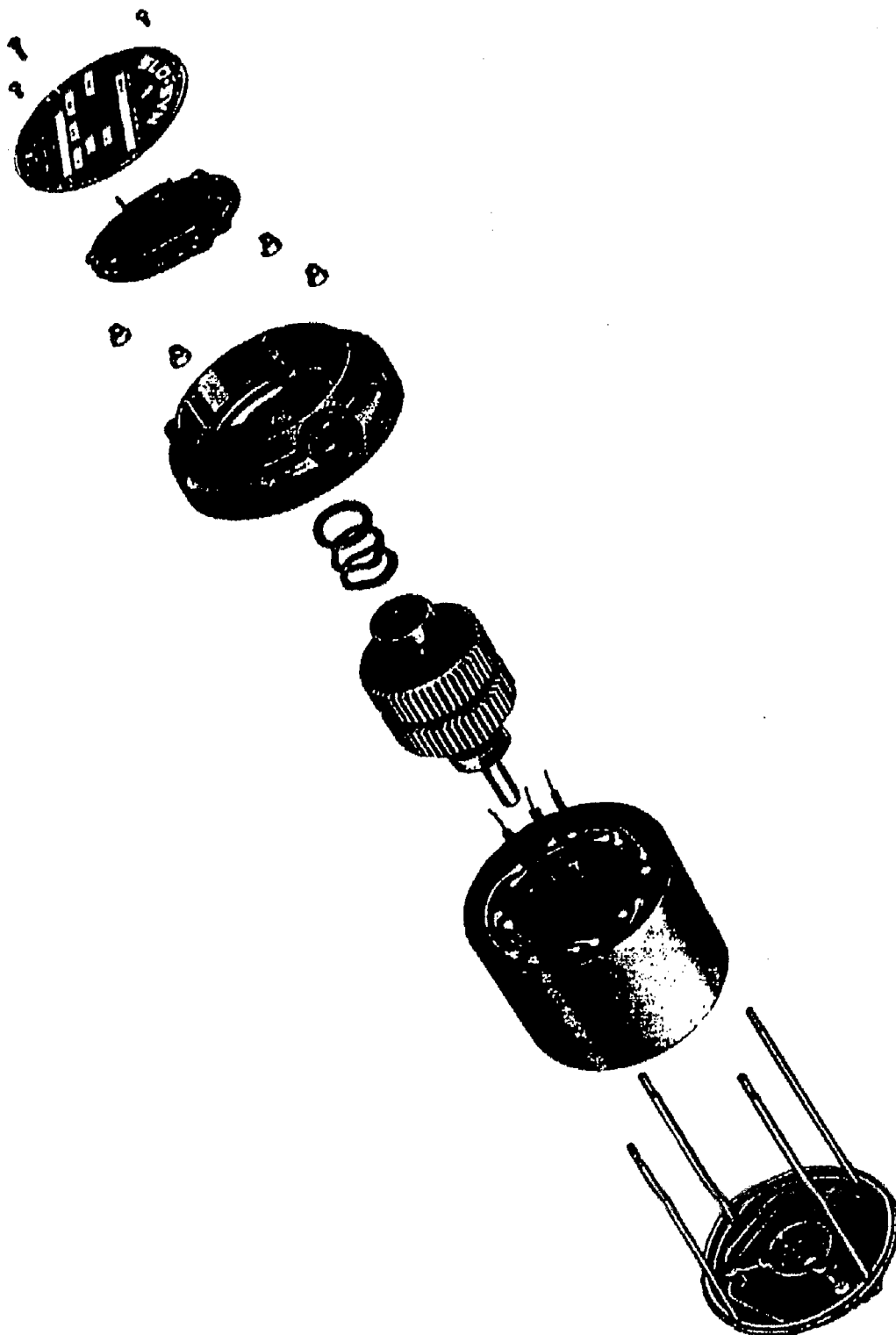


FIGURA 4. Fotografía de las partes que componen un motor de paso a paso. Se aprecian los devanados de las dos secciones del estator y las dos secciones del rotor (imanes permanentes), con las "aletas" que forman los polos.

Cuando una corriente circula a través del bobinado adecuado, se desarrolla un momento que hace girar al rotor hacia la posición en la cual la reluctancia del circuito sea mínima, cuando circula una nueva corriente por otro bobinado, el punto de reluctancia mínima se genera en otra posición, produciendo el giro del rotor a esa nueva posición.

En la figura 5 (ver página 14), se muestra un motor de paso a paso de reluctancia variable, constituido por tres devanados, B_1 , B_2 y B_3 excitados secuencialmente y por un rotor con cuatro dientes, D_1 , D_2 , D_3 , y D_4 . Refiriéndonos a la figura 5, cuando el primer arrollamiento, B_1 , recibe alimentación, atraerá al rotor hasta que el diente más cercano, por ejemplo D_1 , se alinee con el campo. Al llegar la excitación a B_2 , el diente D_2 será el más próximo, con lo que el rotor girará 30 grados. De la misma forma con el siguiente pulso aplicado a B_3 , será el diente D_3 el alineado, girando otros 30 grados. Al restituir la alimentación a B_1 , el atraído será D_4 , con lo que se vuelve a avanzar un ángulo similar a los anteriores.

La principal característica de los motores con reluctancia variable es su elevada velocidad de accionamiento.

c.- Híbridos: Combinan los dos anteriores, obteniendo importantes pares de accionamiento, que giran en gran número de pasos por vuelta y a una frecuencia de trabajo elevada.

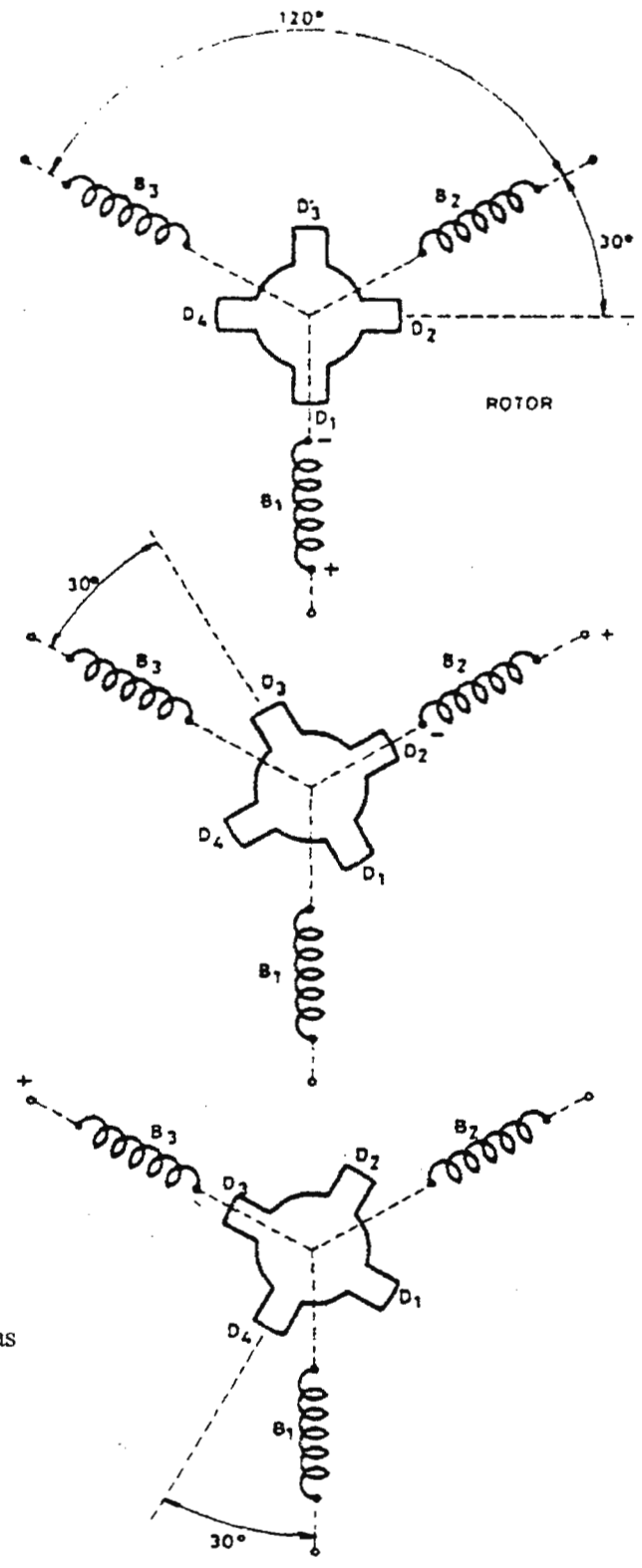


FIGURA 5. Giro de los dientes del rotor al ir activado, secuencialmente, las bobinas del estator.

1.2. CONTROL POR COMPUTADORA

La invención de la computadora ha sido un elemento clave en la evolución del robot industrial moderno. Algunas de las funciones realizadas por el control computarizado de un robot pueden llevarse a cabo con otros dispositivos especializados, pero la flexibilidad global requerida de un robot encierra funciones que sólo puede realizar una computadora.

Algunas de estas funciones se enumeran a continuación:

- Toma de decisiones, en base a parámetros previamente establecidos.
- Almacenamiento y ejecución de programas.
- Aceptar la creación de rutinas de movimiento.
- Mostrar la posición del efector final en el espacio, basándose en el marco de referencia definido.

Una microcomputadora, que es un sistema con un solo microprocesador, no puede llevar a cabo todas estas actividades, cabe mencionar que una computadora no es lo mismo que una microcomputadora. Para controlar un robot, la computadora debe contener suficientes procesadores para controlar aquellas funciones no realizadas por dispositivos especializados.

1.3. MODOS DE OPERACIÓN

Una computadora para el control de un robot puede funcionar en cualesquiera de los modos siguientes: apagado, premanual, manual, enseñanza y automático. Debe hacerse notar que los nombres asignados pueden variar de máquina a máquina.

En un modo *apagado* la computadora está inactiva, pero los datos almacenados en la memoria se preservan si se incluye un suministro de energía de respaldo, como puede ser una batería.

En el modo *premanual* la computadora está funcionando, pero no es necesario suministrar energía al manipulador ya que sólo se lleva a cabo una función de transferencia de datos.

El modo manual se selecciona cuando se requiere un control manual directo de los ejes individuales para las funciones de preparación y mantenimiento.

Como su nombre lo indica el modo de *enseñanza* se utiliza para mostrar a la computadora ciertas secuencias de movimientos y otras operaciones que se han de llevar a cabo, en última instancia en el modo automático. *Enseñanza* es un título generalizado para los métodos de introducción de instrucciones a la computadora de control utilizando el robot.

Durante la enseñanza, el robot no está disponible para una operación automática. Los programas para un robot pueden compilarse utilizando una computadora independiente; esto se conoce como programación fuera de línea. Sin embargo, la programación en línea forma parte de la función de enseñanza, dado que la computadora debe abandonar la operación automática y ubicarse en un modo de enseñanza que acepte información, la cual se suministra, por lo general, desde un teclado.

En el modo de operación *automática*, el robot está disponible ya para ejecutar todas las instrucciones “previamente recibidas” en el modo de enseñanza, de manera secuencial o por un nivel de jerarquía de instrucciones, que pueden involucrar muchas tareas.

1.4. CONTROL DE LA TRAYECTORIA

Se utilizan distintas estrategias para controlar la trayectoria seguida por el efector final durante el movimiento de un punto en un programa al siguiente. Estas estrategias son importantes durante la operación automática y también durante la enseñanza. El control punto a punto requiere que se almacenen sólo las coordenadas de los puntos inicial y de destino.

El control de la velocidad de los actuadores ayuda a suavizar la trayectoria tomada por el efector final durante un control punto a punto, pero la trayectoria real no puede predecirse con facilidad.

Se requiere un control de trayectoria continua cuando la ruta recorrida por el efector final es importante para la tarea. La trayectoria puede definirse en un programa generado fuera de línea o puede enseñarse (entiendase como llevar físicamente al autómatas por los puntos que habrá de recorrer), directamente al robot.

Por ejemplo, el control de trayectoria continua de un robot para una operación de pintura, reproduce la trayectoria trazada por la pistola rociadora durante la etapa de enseñanza, cuando la pistola se dirige en forma manual a través de la operación de pintado.

1.5. CONTROL PUNTO A PUNTO

Muchas veces en una gran cantidad de aplicaciones para robots, sólo importa el movimiento que estos son capaces de efectuar ya que simplifica de gran manera la forma de controlarlos, con esto se refiere a que usualmente se requiere para aplicaciones prácticas únicamente mover alguna cosa de un lugar a otro. tal como puede ser una herramienta, una pieza, etc. En estos casos es relativamente fácil definir la trayectoria que habrá de seguir el

efector final o la herramienta que posea el robot; incluso existen ciertos tipos de éstos, a los cuales se les indican los puntos inicial y final, y de acuerdo a directivas especificadas por el programador, pueden seleccionar la trayectoria más conveniente, ya que lo que interesa es llegar de un punto a otro, para lo cual algunos autómatas (más avanzados), están dotados de sensores ópticos o cámaras de TV para tal propósito (denominado control punto a punto). Pero también existen aquellas aplicaciones, en las cuales se hace necesario asegurar la trayectoria que seguirá el efector, debido a la naturaleza de lo que se esté manipulando o de la disposición física del entorno del robot, en estos casos el movimiento del robot se controla para cada posicionamiento, entre el punto inicial y el final. En estos casos lo usual es que se defina primero una serie de trayectorias óptimas que el efector puede seguir, y las demás se inhiben en su programación básica.

Existen casos en los que se considera crítica la forma de la trayectoria a seguir, ya que puede necesitarse que el efector describa una trayectoria específica, tal como una línea recta, circular, elíptica, etc. En estos casos la forma de control es mucho más compleja y se hace necesario además la utilización de los sensores antes mencionados, para llevar a cabo los movimientos exigidos al robot, a ello se le llama control de trayectoria continua, y se consideran en esta forma todos los puntos intermedios entre una posición y otra (usualmente estos son sistemas de lazo cerrado).

Cuando la trayectoria real entre las posiciones deseadas no tiene mayor importancia, se utiliza un control punto a punto; cuando la trayectoria del centro de la herramienta es importante, se hace necesario un control de trayectoria continua.

Existen cuatro tipos de control punto a punto:

- 1.- Secuencial.
- 2.- No coordinado.

3.- Coordinado en los puntos terminales.

El punto de enseñanza es común a todos los tipos. Sólo los puntos inicial y final se almacenan en la memoria y se ignora la trayectoria recorrida por el punto central de la herramienta durante la enseñanza. La trayectoria real seguida durante la repetición depende del tipo de control que está en operación.

1.- Al usar la forma de control sécuencial se tiene la restricción de que cada eje se impulsa por turnos. Esta forma de control se encuentra comúnmente en robots animados por motores de pasos y controlados por una sola microcomputadora. La misma unidad del microprocesador se usa para generar los pulsos que accionan el motor de pasos, de manera que en un momento dado sólo es posible mover un eje durante la enseñanza y la repetición. Por lo tanto, al moverse de un punto a otro la computadora mueve el primer eje hacia su destino de acuerdo con la secuencia, repite el proceso para el segundo eje y continúa con este proceso hasta que todos los ejes han alcanzado sus nuevas posiciones. La simplicidad de esta forma de control, se paga con el tiempo largo que se requiere para terminar un movimiento completo.

2.- El control punto a punto no coordinado hace más rápido el movimiento al impulsar de manera simultánea todos los ejes. Se requiere un control de posición para cada uno de los ejes, pero no se tiene control de la velocidad de los actuadores, de manera que los actuadores pueden llegar a sus destinos en momentos distintos y, por lo tanto, la trayectoria trazada por el centro de la herramienta es difícil de predecir. La velocidad global con frecuencia la determina el actuador más lento.

3.- Por otra parte se tiene el control punto a punto coordinado en los puntos terminales, el cual es utilizado ampliamente. Los ejes se mueven a velocidades controladas, ya que se

cuenta con un control de la velocidad; de esta manera, todos los ejes llegan a sus destinos en forma simultánea.

Se usa el control punto a punto cuando la trayectoria entre las posiciones deseadas del centro de la herramienta no es de gran importancia. Sin embargo, resulta instructivo hacer notar cómo varían las trayectorias para distintas formas de control punto a punto. Considérese un robot de brazos articulados con cuatro grados de libertad como el mostrado en la figura 6 que mueve el punto central de la herramienta desde las coordenadas de máquina iniciales a la posición destino, que se presenta en dicha figura.

Ambos puntos se han seleccionado en el mismo plano radial vertical, lo que permite que todo el movimiento sea fácil de representar en una página bidimensional. En la figura 6, se muestran las configuraciones inicial (S) y de destino (D) en el plano radial.

Las dimensiones del manipulador $L_1 = L_2 = 1\text{m}$ y $L_3 = 0.25\text{m}$, y las rotaciones requeridas son por lo tanto:

Base: 0°

Hombro: $+28^\circ 10' 22.8''$

Codo: $+46^\circ 20' 6''$

Muñeca: $+15^\circ 29' 31.2''$

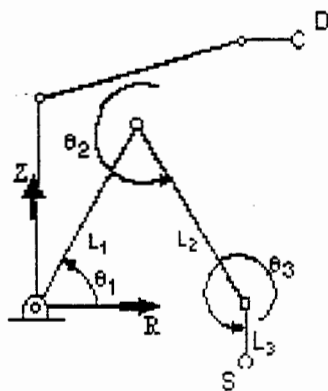


Figura 6. Control punto a punto de un manipulador de brazo articulado; definición de las coordenadas de máquina.

El control secuencial se ilustra en la figura 7; en la figura 7.a. se muestra la posición inicial, en la figura 7.b, la posición a la terminación de la rotación del hombro y en la figura 7.c, la posición después de la rotación del codo. Por último, la muñeca se gira, con lo que se alcanza el destino mostrado en la figura 7.d. Si cada uno de los actuadores tuviese una velocidad máxima de 1.57 rad/seg, el movimiento completo requerirá un segundo.

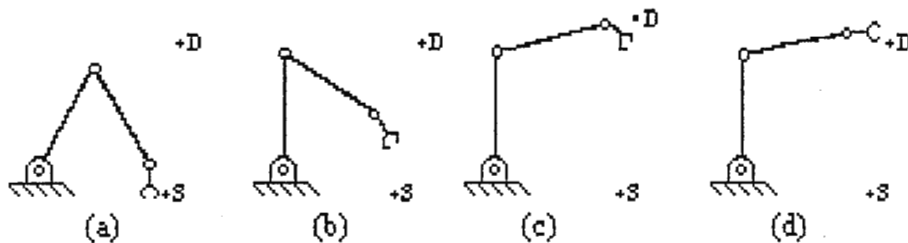


Figura 7. Control punto a punto secuencial: (a) inicio; (b) terminación del movimiento del hombro; (c) determinación del movimiento del codo; (d) determinación del movimiento de inclinación vertical en el destino.

El control punto a punto no coordinado de los mismos actuadores se muestra en la figura 8 (página 24). La posición inicial se muestra en la figura 8.a. Se supone que las velocidades de los actuadores son iguales en la figura 8.b, se muestra la configuración cuando cada actuador ha girado 15.492 grados en éste caso se habrá complementado el movimiento de la muñeca. El hombro debe girar 12.681 grados adicionales, y las configuraciones resultantes se muestran en la figura 8.c. El destino final (figura 8.d) se alcanza cuando el codo ha girado 46.335 grados. Si todos los actuadores se mueven con la

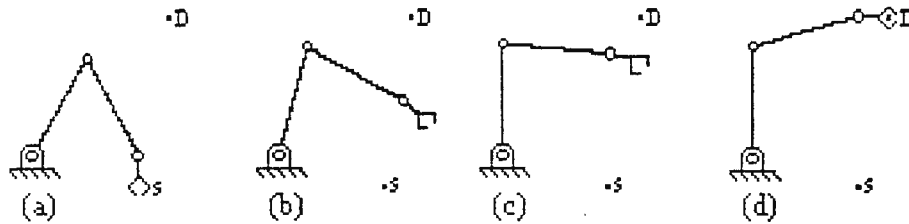


Figura 8. Control punto a punto no coordinado: (a) inicio; (b) terminación del movimiento de inclinación vertical del paso; (c) terminación del movimiento del hombro; (d) terminación del movimiento del codo en el destino.

misma velocidad, el tiempo requerido para completar el movimiento se determina según el eje que requiere mayor rotación; en este caso el codo.

Con los actuadores moviéndose a 1.57 rad/seg, el control punto a punto no coordinado completará este movimiento en 0.51 seg.

El control punto a punto coordinado en los puntos terminales, realiza el movimiento en el mismo tiempo que la forma no coordinada, pero todos los actuadores se mueven a una velocidad calculada para completar sus rotaciones al mismo tiempo. En este ejemplo esto requeriría que el codo se moviera a la máxima velocidad, dado que presenta la mayor rotación; la velocidad de los otros actuadores se calcularía de manera que cada uno completará su rotación en el mismo tiempo que el codo. Las velocidades de rotación requeridas son:

Base: 0 rad/seg

Hombro: 0.96 rad/seg

Codo: 1.57 rad/seg

Muñeca: 0.53 rad/seg

Las posiciones después de 0.17 seg, después de 0.34 seg y en el destino (0.51 seg) se muestran en la figura 9 (página 25).

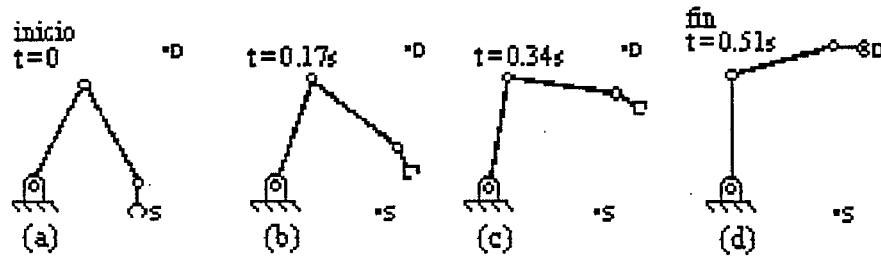


Figura 9. Control punto a punto coordinado: (a) inicio; (b) posición después de 0.17s; (c) posición después de 0.34s; (d) destino.

Las trayectorias del punto central de la herramienta para las distintas formas del control se muestran en la figura 10. Los resultados del control secuencial en la figura 10.a, del control no coordinado en la figura 10.b y del control coordinado en el punto terminal en la figura 10.c.

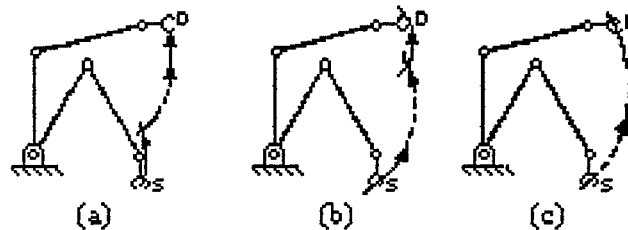


Figura 10. Comparación de las rutas trazadas por el punto central bajo control punto a punto (a) secuencial; (b) no coordinado; (c) coordinado.

Se puede ver que, de las formas de control descritas, el control coordinado en el punto terminal es el único que no presenta discontinuidades en la trayectoria del centro de la herramienta.

1.6. PROGRAMACION TEXTUAL EXPLICITA(PARA AUTOMATAS)

El programa queda constituido por un texto de instrucciones o sentencias, cuya confección no requiere de la intervención del robot; es decir, se efectúan "off-line". Con este tipo de programación el operador no define, prácticamente las acciones del brazo manipulador, sino que se calculan, en el programa, mediante el empleo de las instrucciones textuales adecuadas.

En una aplicación tal como el ensamblaje de piezas, en la que se requiere una gran precisión, los posicionamientos seleccionados mediante la programación gestual(conectar al robot a un humano y que éste imite y guarde los movimientos efectuados), no son suficientes, debiendo ser sustituidos por cálculos más perfectos y por una comunicación con el entorno que rodea al sistema.

En la programación textual, la posibilidad de edición es total. El robot debe intervenir, sólo, en la puesta a punto final.

Según las características del lenguaje, pueden confeccionarse programas de trabajo complejos, con inclusión de datos condicionales, empleo de base de datos, posibilidad de creación de módulos operativos intercambiables, capacidad de adaptación a las condiciones del mundo exterior, etc.

Dentro de la programación textual, existen dos grandes grupos de características netamente diferentes:

- 1- Programación textual explícita.
- 2- Programación textual especificativa.

En la programación textual explícita, el programa consta de una serie de órdenes o instrucciones concretas, que van definiendo con rigor las operaciones necesarias para llevar

a cabo la aplicación. Se puede decir que la programación explícita engloba a los lenguajes que definen los movimientos punto a punto similares a los de programación gestual, pero bajo la forma de un lenguaje formal. Con este tipo de programación, la labor de tratamiento de las situaciones anormales, colisiones, etc., queda a cargo del programador.

Dentro de la programación explícita hay dos niveles:

1- Nivel de movimiento elemental

Comprende los lenguajes dirigidos a controlar los movimientos del brazo manipulador. Existen dos tipos:

a.- Articular, cuando el lenguaje se dirige al control de los movimientos de las diversas articulaciones del brazo.

b.- Cartesiano, cuando el lenguaje define los movimientos relacionados con el sistema de manufactura, es decir, los del punto final de trabajo.

Los lenguajes del tipo cartesiano utilizan transformaciones homogéneas. Este hecho confiere "portabilidad" al programa, independizando a la programación del modelo particular del robot, puesto que un programa confeccionado para uno en coordenadas cartesianas, puede utilizarse en otro, con diferentes coordenadas, mediante el sistema de transformación correspondiente. Son lenguajes que se parecen al BASIC, sin poseer una unidad formal y careciendo de estructuras a nivel de datos y de control.

Por el contrario, los lenguajes del tipo articular indican los incrementos angulares de las articulaciones. Aunque ésta acción es bastante simple para motores de paso a paso y corriente continua, al no tener una referencia general de la posición de las articulaciones con relación al entorno, es difícil relacionar al sistema con piezas móviles, obstáculos, cámaras de TV, etc.

Los lenguajes correspondientes al nivel de movimientos elementales aventajan, principalmente, a los de punto a punto, en la posibilidad de realizar bifurcaciones simples y saltos a subrutinas, así como de tratar informaciones sensoriales.

2-Nivel estructurado

Intenta introducir relaciones entre el objeto y el sistema del robot, para lo que los lenguajes se desarrollan sobre una estructura formal.

Se puede decir que los lenguajes correspondientes a este tipo de programación adoptan la filosofía del pascal. Describen objetos y transformaciones con objetos, disponiendo, muchos de ellos, de una estructura de datos arborescente.

El uso de lenguajes con programación explícita estructurada aumenta la comprensión del programa, reduce el tiempo de edición y simplifica las acciones encaminadas a la consecución de tareas determinadas.

En los lenguajes estructurados, es típico el empleo de las transformaciones de coordenadas, que exigen un cierto nivel de conocimientos. Por este motivo dichos lenguajes no son populares hoy en día.

1.7. PROGRAMACIÓN TEXTUAL ESPECIFICATIVA

Se trata de una programación del tipo no procesal, en la que el usuario describe las especificaciones de los productos mediante una modelización, al igual que las tareas que hay que realizar sobre ellos.

El sistema informático para la programación textual especificativa ha de disponer del modelo del universo o mundo donde se encuentra el robot. Este modelo será, normalmente una base de datos más o menos compleja, según la clase de aplicación pero que requiere, siempre, computadoras potentes para el procesado de una abundante información.

El trabajo de programación consistirá, simplemente, en la descripción de las tareas a realizar, lo que supone poder llevar a cabo trabajos complicados.

En la actualidad los modelos del universo son del tipo geométrico, no físico. Este hecho implica por ejemplo la no-consideración de la fuerza de la gravedad, que ha de ser tenida en cuenta por el programador constantemente, para considerar sus efectos en la manipulación de los objetos.

Dentro de la programación textual especificativa, hay dos clases, según que la orientación del modelo se refiera a los objetos o a los objetivos.

Si el modelo se orienta al nivel de los objetos, el lenguaje trabaja con ellos y establece las relaciones entre ellos. La programación se realiza "off-line".

Dada la inevitable imprecisión de los cálculos de la computadora y de las medidas de las piezas, se precisa de una ejecución previa, para ajustar el programa al entorno del robot.

Los lenguajes con un modelo del universo orientado a los objetos son de alto nivel, permitiendo expresar las sentencias en un lenguaje similar al usado comúnmente.

Por otra parte, cuando el modelo se orienta hacia los objetivos, se define el producto final.

La creación de lenguajes de muy alto nivel transferirá una gran parte del trabajo de programación, desde el usuario hasta el sistema informático; éste resolverá la mayoría de los problemas, combinando la automática y la inteligencia artificial.

La filosofía de los lenguajes de programación en función de los objetivos, consiste en definir la situación final del producto a fabricar, a partir de la cual se generan los planes de acción tendentes a conseguirla, obteniéndose, finalmente, el programa de trabajo.

Estos lenguajes de tipo natural, no eran realistas en 1983, suponiendo una potenciación extraordinaria de la Inteligencia Artificial, para descargar al usuario de las labores de programación. Preveían, incluso, la comunicación hombre-máquina a través de la voz

1.8. CLASES DE ROBOT

Los robots se clasifican de la siguiente manera:

Por el sistema de accionamiento

- De accionamiento eléctrico
- De accionamiento hidráulico

Por el sistema de articulaciones

- De articulaciones polares

- De articulaciones telescópicas

Por el nivel de control de sus trayectorias

- De punto a punto sin servocontrol (motores paso a paso)
- De punto a punto con servocontrol, en lazo cerrado
- De control continuo, para seguimiento de trayectorias
- De control continuo con capacidad de generación de trayectorias

Por la tarea a realizar

- De manipulación en general y montajes
- De soldadura por puntos
- De soldadura por arco
- De pintura

Por la estructura del manipulador

- Verticales
- Horizontales
- De pórtico

1.9 ELEMENTOS QUE COMPONEN UN ROBOT INDUSTRIAL

a.- El manipulador: Es un brazo montado en una base y está compuesto por base, hombro, antebrazo, codo, muñeca (con tres movimientos, rotación, doblado y derrape); y mano, la cual puede adoptar la forma de una pinza, dispositivo de agarre, pistola de pintar, etc.

Por lo general, los robots tienen un captador de posición y uno de velocidad por eje, para informar a la unidad de control constantemente sobre estas variables.

b.- Mano de los robots: Son adecuadas al tipo de trabajo a realizar estas pueden asir objetos por tres métodos: mecánicos, neumáticos o electromagnéticos.

c.- Unidad de mando y control: Permite el funcionamiento automático que caracteriza al robot, puede estar equipada con uno o dos microprocesadores de 16 bits con sus correspondientes memorias EPROM para las funciones de la máquina y RAM para almacenar los programas.

Como entradas y salidas pueden llevar un teclado funcional, un display o pantalla, un telemando manual, una disquetera, etc.

2. SOFTWARE

2.1. DESCRIPCION DEL SOFTWARE

El software está diseñado para trabajar bajo ambiente MS-DOS, y ha sido generado en el lenguaje de programación QBASIC. Debido a las características del lenguaje de programación, se decidió usar el modo texto de este, debido a que presenta muchas más ventajas que las que permite el modo gráfico, pero ello no ha sido una limitante para presentar de una forma más simple y “amigable” para el usuario, ya que se hace uso de ventanas, barras de menú, etc. En sí la presentación del programa de control para el brazo robot, es muy parecida a la usada en los programas generados en Foxpro o Dbase.

En dicho software, se permite al usuario que maneje ciertas opciones usadas en otro tipo de paquetes, tales como archivo, ayuda, guardar archivo, etc., pero todo ello está orientado al control del robot de una forma sencilla.

Al ingresar al programa se presenta una pantalla, la cual en el centro se incluyen los nombres de los creadores del paquete, y el nombre de éste; además se puede ver una barra de menú con las opciones de este, dispuestas como se muestra a continuación:

Archivo Ejecutar DOSprompt Salir Ayuda

Esta barra aparece casi en todo momento mientras se utiliza el programa, las letras son negras en un fondo gris y cada palabra con ciertas letras resaltadas, ello indica que el menú no está activado, esta barra desaparece, cuando se trabaja con las opciones de cambiar, Nueva Sec, del menú de Archivo y en las de Mov. por eje, Mov simultáneo del menú Ejecutar. Para

utilizar el menú es necesario presionar la **Barra espaciadora**, con lo que se activan las opciones, presentando las posibles selecciones del menú con un fondo verde claro la palabra del menú sobre la que se ha posicionado, de las cuales algunas presentan un submenú vertical, el cual puede desplegarse de tres formas, una es ubicándose sobre el elemento deseado y pulsando **enter** sobre él, la segunda forma, es con las flechas cursoras posicionarse en la selección deseada y apretar luego la tecla cursora hacia arriba o hacia abajo. La tercera y última, es desde cualquier lugar del menú, digitar la letra resaltada de cada palabra, con ello se resaltarán la opción seleccionada, se deberá presionar **enter** después y a continuación se procede a describir lo que puede suceder dependiendo de la selección hecha en esta barra de menú.

2.1.1.- ARCHIVO

La característica que indica que se ha seleccionado esta opción del menú la cual es común para las demás, es que aparece bajo otro color de fondo (verde claro), y una letra de la palabra resaltada en color blanco intenso. Para poder acceder a los elementos de este submenú, se puede desplazar a través de este en dos formas, una es presionando las teclas cursoras hacia arriba \uparrow o abajo \downarrow , la otra es presionando la letra resaltada, como ya se había mencionado anteriormente se resalta la selección correspondiente a la letra presionada. En el caso de utilizar las teclas cursoras, se pueden usar las del bloque numérico o las marcadas como cursoras del teclado, que se encuentran entre el bloque numérico y el teclado alfanumérico.

Como su nombre lo indica este submenú es el referente al tratamiento de archivos, los elementos que se verán al seleccionar archivo son los siguientes:

Archivo Ejecutar DOSprompt Salir Ayuda

Abrir

Guardar

Nva. Sec.

Guardar Como

Modificar

2.1.1.1. ABRIR

En caso de escoger Abrir, dentro del submenú archivo, de cualquiera de las formas descritas anteriormente, sucederá lo siguiente:

a.- Sí no existe ningún archivo, que pueda ser utilizado por el programa (con extensión “.seq”), se ignora el comando Abrir archivo, ya que en los archivos “. seq” se guardan los valores de variables para efectuar movimientos, las cuales podrían servir después al usuario y con ello se evita el estar creando secuencias de movimiento cada vez que sean necesarias.

b.- En el caso que ya existan archivos “.seq” (por lo menos uno de ellos), se visualizará en pantalla un listado de los mismos, y debajo de la lista se presenta un cuadro de diálogo, en donde deberá digitarse el nombre deseado. Si no es una entrada válida, el nombre anterior se ignora y se pregunta de nuevo. De no ser así, se abre el archivo, y se presenta una pantalla en la cual se pueden visualizar los valores de las variables contenidas en el archivo abierto, y fueron guardadas de una secuencia previamente generada, si se desea cambiar dichos valores, deberá presionarse Esc., para volver al submenú anterior, y ya en él, seleccionar Modificar, para cambiar dichos valores; la forma de hacerlo, se explicará cuando se defina la opción antes mencionada. Pero en el caso que se necesite ejecutar la secuencia, debe entonces presionar Esc., lo cual regresa al submenú anterior, y una vez allí se deberá cambiar al submenú

Ejecutar(con las teclas cursoras) y se selecciona Ejecutar sec., como hacerlo se explica más adelante en la sección 2.1.2

2.1.1.2. GUARDAR

En el ítem Guardar, se permite actualizar una secuencia que haya sido editada con la opción Modificar, si se desea que la secuencia editada tenga el mismo nombre del archivo actual o en si fuere el caso, del archivo que fué abierto. Si fuese la primera vez que guarda el archivo, significa que no han sido guardados los valores correspondientes a las variables de movimiento que se hayan especificado en Nva. Sec, el programa despliega una ventana, en la cual se pedirá un nombre para el archivo, con ello se está guardando en la memoria de la computadora, una nueva secuencia de movimiento. Si existe un archivo con el nombre especificado, se ignora la entrada y se pide otro nombre, hasta que se de uno diferente a los ya existentes.

2.1.1.3. Nva. Sec.

Para crear una secuencia de movimientos, deberá escogerse Nva. Sec. Aquí se presenta una pantalla, en la cual se muestran los nombres de cada una de las partes del robot en seis filas y cuatro columnas habilitadas para poder crear las sucesiones de movimientos deseados, lo cual indica que se tiene la capacidad de especificar un desplazamiento cualquiera por motor, para un sólo desplazamiento y además cuatro posicionamientos diferentes por secuencia.

Para movilizarse dentro de la pantalla presentada por Nva. Sec. Se dispone de los números de fila (del uno al seis), correspondiente a cada motor del robot, los que van en el siguiente orden desde arriba hacia abajo: Base, Espalda, Antebrazo, Muñeca(arriba/abajo), Muñeca(rotación) y finalmente Pinza; para el desplazamiento entre columnas(el cual se

permite sólo hacia la derecha), se utiliza la tecla TAB, después de presionar esta tecla por cuarta vez, automáticamente se regresa al menú principal. Una vez posicionado en la fila y columna que se desea modificar, se puede incrementar o disminuir el valor del desplazamiento del elemento del robot con las teclas cursoras $\hat{\uparrow}$ o $\hat{\downarrow}$, respectivamente, ello es para permitir al usuario ver de forma física la posición en el espacio, en la cual se encuentra la pinza robot según las coordenadas presentadas en pantalla, ya que sería muy dificultoso el tener que determinar la posición en base a relaciones de ángulos entre los diferentes elementos que componen al robot, y haciendo uso de relaciones de ángulos y pasos para cada pieza del robot, todo ello obligaría al usuario a conocer mucho sobre la forma de posicionar la pinza del robot en un punto del espacio deseado, pero se pretende hacer del manejo del robot algo sencillo.

Debido a lo anterior, se permite el manejo de los motores, directamente desde el teclado, para llevar la pinza hasta el punto deseado, pero además debe tomarse en cuenta que para cada vez que se presiona una tecla cursora, el motor seleccionado se moverá un número determinado de pasos, a lo cual se la ha denominado “paso de incremento o disminución”, el cual se ha dejado por defecto de 10 pasos para cada motor, por cada vez que se presione una de las dos teclas ya mencionadas, es muy posible que algunas veces cuando alguna de las posiciones a la cual deberá llegar la pinza, esté muy cerca de la anterior, se necesitará que el número de pasos sea menor al escogido por defecto; o podría suceder lo contrario, posiblemente los desplazamientos sean muy grandes, en ese caso sería deseable un desplazamiento mayor por cada vez que se presione una tecla, por tal razón para proporcionar más flexibilidad al usuario, se permite el modificar el paso por incremento o disminución, si se desea un paso diferente, se debe presionar la tecla “I”, con lo que se despliega un mensaje preguntando por el nuevo paso deseado (máximo de pasos

999 y mínimo 1), ya que se trata de un sistema discreto en el tiempo, no se aceptan números decimales para la selección anterior. Cabe mencionar que puede usarse la tecla backspace para cambiar el número anterior sólo si no se ha presionado antes la tecla Enter; una vez aceptado el dato, se borra el mensaje en el cual se pide dicho dato, y se retorna el control a la pantalla de creación de secuencias.

Para salir al submenú archivo, puede presionarse la tecla Esc, o presionar TAB cuatro veces, ello le indica al programa que se ha terminado de crear la secuencia, lo anterior se ha hecho de esa forma, debido a que el movimiento del robot de un punto a otro, siempre es relativo, y el poder cambiar de la cuarta columna a la primera, implica que el movimiento que sea especificado por la segunda columna de la opción, estará referido al punto en el cual se deberá quedar el robot, luego de realizar el desplazamiento desde el origen físico, hasta el punto especificado por la columna uno de la pantalla correspondiente a la de creación de secuencias. Entonces como puede verse, se trastornaría todo el movimiento, debido a que se pierde un punto de referencia para el movimiento. Por otro lado, el movimiento absoluto no es conveniente, ya que para cada momento deberá calcularse el cero que será la referencia, pero debido al lenguaje de programación usado, la capacidad del robot, y de las computadoras, se presenta un desbordamiento de memoria en las últimas, debido a la gran cantidad de cálculos que se deben efectuar. Pero, si se evalúan, las ventajas contra las desventajas, puede verse que son mucho más de peso la ventajas, además editar secuencia con Cambiar no es nada difícil.

2.1.1.4. GUARDAR COMO

Con Guardar como, se permite que las modificaciones que no hayan sido guardadas con los valores correspondientes a las variables de movimiento, sean guardadas con otro

nombre distinto al del archivo original, ello permite al usuario crear un archivo en base a otro, pero se desea además tener la fuente y una variante de éste, aunque sea mínima la diferencia entre uno y otro.

Cuando se hace uso de esta opción, aparece una ventana al igual que en Guardar en la cual se pedirá un nombre, para el nuevo archivo a crear, con ello se están guardando modificaciones a una secuencia, con un nombre diferente, sin cambiar la original, y permitir así ejecutar ambas. Si el nombre introducido ya pertenece a otro archivo, también ignorará dicha entrada, y luego esperará por una entrada válida, que no haya sido asignada a otro archivo.

Para salir de cualquiera de las opciones se debe presionar la tecla ESCAPE. Dicha tecla es reconocida en todo el menú y submenús, el objetivo de ésta tecla es desactivar el menú principal.

2.1.1.5 MODIFICAR

Al presionar Enter sobre esta opción, se ingresa a una pantalla, similar a la de el numeral 2.1.1.3.; con la diferencia de que aparecerán los últimos valores que hayan sido cargados con la sección 2.1.1.3. o con la 2.1.1.1. en lugar de solo ceros.

Como su nombre lo indica, su objetivo es cambiar dichos valores, en caso de existir algún error(desplazamiento equivocado), que nos impida ubicar el autómata en una posición deseada.

Todas la explicaciones de movimiento en la pantalla, selección de incremento/decremento, y forma de salir de la sección 2.1.1.3. son válidas para esta alternativa.

2.1.2. EJECUTAR

El submenú denominado ejecutar, es el que trata los diferentes movimientos que pueden especificársele al robot, esta selección de la barra de menú principal, es la que produce un efecto más palpable, tal como es el movimiento del robot, ya que las demás podrían considerarse complemento del programa, y considerar este submenú como el centro y la razón de ser del programa; aunque no se debe descartar el resto del paquete, ya que ha sido incluido, para hacer del interfaz entre el usuario y el robot, más fácil, y bajo un entorno menos rígido.

Al ser seleccionado, se despliega un conjunto de opciones, dispuestas en forma vertical, tal como se puede visualizar a continuación:

Archivo Ejecutar DOSprompt Salir Ayuda

 Mov. por eje

 Mov. simultáneo

 Ejecución de seq.

 Mov. Cíclico.

2.1.2.1. MOV. POR EJE

En esta alternativa es posible escoger cuál de los motores del brazo robot se quiere movilizar, al ser seleccionada aparece una pantalla, en la cual, se pregunta por la velocidad a la que se desea realizar el movimiento(sólo se acepta de 1 a 3); posteriormente se pide el motor que se habrá de mover, para ello se han numerado los motores de uno a seis de la siguiente manera: Base, Espalda, Antebrazo, Muñeca(arriba/abajo), Muñeca(rotación) y finalmente Pinza. Al programa se le introduce el correlativo asignado para cada posibilidad, dependiendo de la selección se presentan los rangos de desplazamiento válidos para el motor escogido; por

último se pide sea introducido el número de pasos que deberá moverse, y presionando Enter, se ejecuta la acción requerida; pero es de mencionar que si el dato introducido está fuera del rango presentado para cada selección en particular, se ignora y se espera por un valor válido, lo mismo sería si no se introduce un carácter numérico, (exceptuando el signo menos).

Al finalizar la acción pedida, el robot regresa al punto de partida y aparece un mensaje en la parte inferior de esta pantalla, preguntando si se desea efectuar otro movimiento, si se presiona la tecla "S", se regresa al inicio de la pantalla, repitiendo todo el procedimiento anterior. En el caso que se presione "N", se regresará al submenú ejecutar.

Para esta sección todos los movimientos que sean realizados, serán con referencia absoluta, es decir, se harán respecto al origen indicado en la estructura del robot. Esto se debe a que sólo se realiza el desplazamiento a una posición, y luego se regresa al punto de partida.

No se hace relativo como en otros casos de este submenú, debido a que en los tales, no se regresa al lugar de partida, sino hasta que se abandona la selección, lo cual agrega un mayor grado de complejidad al movimiento, por que se hace necesario definir de alguna forma la posición actual de la pinza del robot luego de un movimiento.

2.1.2.2. MOV. SIMULTÁNEO

Aquí se muestra una variante a la alternativa anterior, ya que no sólo es un motor el que puede moverse a la vez, sino que se puede escoger los motores que deberán actuar para hacer posible la ubicación del efector en un punto determinado del espacio -en la sección anterior se definió un movimiento restringido a un eje-.

Para tal efecto se hace la presentación de una pantalla, en la cual se muestra un listado de los motores que impulsan al mecanismo del autómatas, a continuación se encuentra una sección en la que se pregunta la velocidad a la que se realizará el movimiento -el rango válido

b) Después de editar una secuencia, ya sea en **Nva. Sec.** ó en **Modificar** del submenú Archivo (numerales 2.1.1.3 y 2.1.1.5 respectivamente).

Asumiendo que ya se tienen valores de desplazamiento, estos se envían al robot, el cual los ejecuta y luego retorna a la posición de la cual partió. Después de lo cual regresa el control al submenú ejecutar.

El objeto de esta opción es poder verificar que los movimientos que sean guardados en un archivo o que se quieran salvar sean los deseados.

2.1.2.4. MOV. CÍCLICO

Básicamente efectúa lo mismo que **ejecución de sec.**, con la diferencia de que aquí se repite un número indefinido de veces, regresando al origen físico indicado en el robot y comenzando de nuevo. La única forma de detener este ciclo es presionando la tecla ESC, luego de esto se devuelve el control al submenú ejecutar.

2.1.3. DOSPROMPT

Al seleccionar DOSprompt del menú principal, nos saca momentáneamente del programa al indicador del sistema operativo, esta opción es útil en el caso que se quiera utilizar en forma temporal alguno de los comandos del DOS. Una vez aparece el indicador del DOS, puede utilizarse como tal. Para regresar al programa, basta con digitar EXIT y presionar la tecla ENTER lo cual nos regresa al menú previo donde se activó la opción DOS prompt.

2.1.4. SALIR

Es la alternativa que permite salir o terminar una sesión en el programa controlador del brazo robot, basta con presionar ENTER, desde cualquiera de las secciones del menú principal o antes de activar cualquier selección válida de los submenús para que se cumpla la acción mencionada anteriormente.

2.1.5. AYUDA

La última de las variantes que presenta el menú principal es la opción ayuda, la cual permite tener acceso a diferentes temas relacionados con la estructura física del robot así como también información referente a los motores paso a paso y a los pines del microprocesador Z80.

A continuación se presenta el submenú y sus correspondientes alternativas:

Archivo	Ejecutar	DOSprompt	Salir	Ayuda
				Tutorial
				Generalidades
				Glosario

2.1.5.1 TUTORIAL

Si se presiona Enter en Tutorial, se activa la modalidad tutora del programa, la cuál difiere con el menú principal en que no se observa la ventana de mensaje que indica el tema del programa. Inmediatamente se despliega una ventana que envía un mensaje para que el

usuario sepa en donde está y pueda obtener información acerca de cómo desplazarse por los menús, seleccionar una alternativa o que hace una opción de los submenús, así como también explicar los pasos a seguir cuando se ha ingresado a una sección específica a manera de introducir al usuario de una manera sencilla al manejo del programa.

Todos los mensajes que son presentados, se borran de manera automática para asegurar que la pantalla este limpia para la siguiente ventana de ayuda.

En caso de querer salir de esta modalidad, basta con presionar ESC para salir a la pantalla de inicio del programa.

2.1.5.2 GENERALIDADES

Al ingresar a esta variante, se obtiene un listado, cuyo contenido es proporcionar al usuario información acerca de temas específicos. Para desplazarnos, se utilizan las teclas cursoras y para seleccionar, basta con dar ENTER en la alternativa deseada.

Una vez escogido el tema se procede a desplegar en pantalla, todos los datos y gráficas referentes a la selección en cada pantalla nueva, aparecerá en la parte inferior derecha un dibujo del ratón utilizado en computadoras con flechas sobre los botones las cuales indicarán la forma para acceder a pantallas posteriores o anteriores. En caso de no disponerse de un ratón en con la máquina, puede usarse las teclas + y - como otra alternativa para realizar lo antes expuesto, e incluso presionando la tecla enter puede realizarse la presentación de la pantallas, únicamente que serán mostradas en orden hacia delante. Para salirse del tema elegido basta con presionar ESC desde cualquier pantalla(desplegadas) y la misma tecla para regresar el control al menú ayuda. A continuación se muestra el listado de temas:

MOJORES PASO A PASO

CONTROL POR COMPUTADORA

CONTROL DE LA TRAYECTORIA

MODOS DE OPERACIÓN

CONTROL PUNTO A PUNTO

PROGRAMACIÓN TEXTUAL EXPLÍCITA

PROGRAMACIÓN TEXTUAL ESPECIFICATIVA

HARDWARE

ACERCA DE Z80

2.1.5.3. GLOSARIO

En esta opción se muestra el glosario incluido en el documento, en forma de listado, para desplazarse por dicha lista, se utilizan las teclas cursoras, una vez posicionado sobre la alternativa deseada basta con presionar enter para desplegar una ventana que contiene el significado de la palabra o frase seleccionada, si ya se ha leído el contenido solo hay que presionar enter, de nuevo, para salir de la ventana y regresar a la lista principal del glosario. Para retornar al menú principal, basta con presionar ESC.

2.2. LISTADO DE COMANDOS DEL ROBOT

2.2.1 ZERO (posicion DE CERO)

Mediante este mando (LPRINT "Z"), el robot toma como posición inicial absoluta de referencia, la posición actual.

Está opción es necesaria debido a que los motores son de avance gradual y están desprovistos de un control automático de posición y, por consiguiente, de toda referencia para su posición inicial absoluta.

2.2.2. MOVE

Con esta instrucción (LPRINT Ma1,a2,a3,a4,a5,a6) cada uno de los 6 ejes del robot toma la posición especificada por los parámetros a1, a2, a3, a4, a5 y a6 , donde :

a1 es la base

a2 es la espalda

a3 es el brazo

a6 son las pinzas

Los motores a4 y a5 determinan juntos el movimiento de la muñeca.

2.2.3. HERE

Mediante esta instrucción (LPRINT "Ha0), en la memoria RAM del sistema de control se memoriza la posición en la que se encuentra el robot y, por consiguiente, el valor de los 6 parámetros relativos de los ejes. Pueden memorizarse como máximo 100 posiciones diferentes.

2.2.4. POSITION

Con este mando (LPRINT "pa0, a1, a2, a3, a4, a5, a6"), se memoriza con un índice "a0" la posición especificada por los parámetros a1...a6 .

Todas las posiciones se refieren al punto cero fijado con este mando. Las posiciones pueden ser, a lo sumo, 100. La diferencia con el comando anterior es el hecho

de que no es necesario desplazar físicamente al robot a la posición deseada para memorizarla, sino, que es suficiente especificar el número de la posición.

2.2.5. GO TO

Mediante esta instrucción (LPRINT "Ga0") el robot se desplaza a la posición a0 correspondiente.

2.2.6. NEST

Con esta instrucción (LPRINT "N") el robot se desplaza a su posición mecánica de arranque, previamente fijada con el mando ZERO o tomada por el robot en el momento del encendido.

2.2.7. GRIP CLOSE

Este mando (LPRINT "C") determina el cierre de las pinzas que alcanzan su límite mínimo de apertura.

2.2.8. GRIP OPEN

Con esta instrucción (LPRINT "O") las pinzas se abren y alcanzan su límite máximo de apertura.

2.2.9. SPEED

Con este mando (LPRINT "Sa0") se selecciona la velocidad de desplazamiento de los ejes. Este valor puede variar entre 1 y 3.

2.2.10 DELAY

Tras recibir este mando (lprint "Da0"), el robot queda a la espera de más instrucciones durante un tiempo correspondiente al número de segundos indicados mediante a0.

2.2.11 LIMITE

Este mando (LPRINT "L0") permite controlar los límites de desplazamiento de los ejes. Los límites por definición pueden ser disminuidos, pero no aumentados.

3. HARDWARE

3.1. ESTRUCTURA MECÁNICA DEL BRAZO ROBOT

El robot tiene cinco grados de libertad, y es del tipo de acoplamiento vertical múltiple; la precisión de posicionamiento es de ± 0.9 mm.

Cabe mencionar que en la base de este, se encuentran dos teclas, las cuales son denominadas TEST y RESET, la función de la primera tecla, se describe adelante con bastante detalle; mientras que la segunda, como su nombre lo indica, sirve para reinicializar el sistema.

La capacidad de carga es de 500 gramos(máximo) mientras que el peso total del robot es de 8 Kg. Las pinzas prevén una apertura máxima de 45mm, mientras que la velocidad máxima puede ser de 300mm/seg.

Se concibió¹ la distribución de las masas para mantener una buena estabilidad, incluso en situaciones de carga más gravosas, es decir cuando las pinzas se encuentran en una posición muy alejada del eje principal del robot; de hecho la base y el soporte son mucho más pesados que el resto de la estructura, para servir de contrapeso a la misma, en los casos ya mencionados.

3.1.1. BASE

Tipo de movimiento: rotación en el plano horizontal. Intervalo de desplazamiento: 240 grados (120 grados en sentido dextrorso y los otros 120 grados en sentido sinistrorso).

¹ Ello se refiere al fabricante del brazo robot, ELECTRÓNICA VENETTA. En ningún momento se ha cambiado el diseño de la estructura del mismo.

Número de pasos realizados: 2000, lo cual indica un ángulo recorrido de 0.12 grados por paso.

3.1.2. ESPALDA

Tipo de movimiento: rotación en el plano vertical. Intervalo de desplazamiento: 144 grados. (72 grados hacia arriba y 72 hacia abajo). Número de pasos realizados: 1200, con un ángulo recorrido de 0.12 grados por cada paso.

3.1.3. ANTEBRAZO

Tipo de movimiento: rotación en el plano vertical. Intervalo de desplazamiento: 100 grados. Número de pasos realizados: 1000 pasos (50 grados hacia arriba y 50 grados hacia abajo). Recorriendo un ángulo de 0.1 grados por pasos.

3.1.4. ELEVACION DE LA MUÑECA

Tipo de movimiento: rotación en el plano vertical. Intervalo de desplazamiento: +90 grados, hasta -90 grados. Número de pasos realizados: 1800, por tanto el paso es de 0.1 grados.

3.1.5. ROTACION DE LA MUÑECA

Tipo de movimiento: rotación alrededor del eje principal de las pinzas; intervalo de desplazamiento: +180 grados, -180 grados. Número de pasos realizados en sentido dextrorso 1800 y en sentido opuesto. El ángulo recorrido es de 0.1 grados por paso.

3.1.6. PINZAS

Tipo de movimiento: rotación alrededor del eje principal de la pinza; el peso máximo que se puede manipular es de 0.5 Kg, incluyendo el peso de la pinza . El motor que se utiliza es de avance gradual, así como también para el movimiento de las articulaciones. Si la pinza está cerrada, se puede efectuar un movimiento de apertura máxima de 1800 pasos, lo que equivale a una apertura de 0.025 mm por paso. En cambio, si la pinza está abierta, se puede efectuar un movimiento análogo de cierre de 1800 pasos.

3.2. MODALIDADES DE UTILIZACION

El robot posee dos modos de trabajo, los cuales son:

- MODO "SELF TEST"
- MODO DE CONTROL POR COMPUTADORA

3.2.1. MODO SELF TEST

Con esta modalidad, es posible verificar el funcionamiento del robot y de su control observando los diferentes movimientos incluidos en la memoria ROM del robot, la cual trabaja en conjunto con un microprocesador, el cual simplifica la forma de control del robot. En este modo, puede efectuarse dos tipos de SELF TEST:

3.2.1.1. MODALIDAD 1

Dicho test, tiene que efectuarse, preferiblemente, sin aplicar ninguna carga a las pinzas del robot; el objetivo que esta fase pretende lograr es el de verificar el movimiento correcto de todos los ejes. Además, este test se puede utilizar como procedimiento sencillo para volver a todos los ejes a su posición inicial de cero². Operaciones a efectuar:

1. Presione la tecla RESET.
2. Medio segundo después, oprímase la tecla TEST. Al mismo tiempo, presiónese también la tecla RESET y suéltese primero ésta última.
3. Manténgase oprimida la tecla TEST durante unos tres segundos y luego se deberá soltar.
4. Si la pinza está ya cerrada, cada vez que se "aprieta" sucesivamente la tecla TEST, el robot efectúa un movimiento relativo a cada eje. hasta que no se suelte dicha tecla.

Al oprimirse la tecla TEST por primera vez, se acciona el motor, que determina la rotación dextrorsa de la base del robot (viéndose el robot desde arriba). El motor se detiene al soltarse la tecla TEST.

Al apretarse la tecla TEST por segunda vez, se acciona siempre el motor de la base, con una rotación siempre dextrorsa. El motor se detiene al soltarse la tecla TEST.

Al presionarse la tecla TEST por tercera vez y cuarta vez, se pone en marcha el motor que hace subir y bajar la espalda del robot.

Al apretarse la tecla TEST por quinta y sexta vez, se pone en acción el motor que hace subir y bajar el brazo del robot.

Cuando se “aprieta” la tecla TEST por séptima vez, se da una vuelta en sentido dextrorso³ a la muñeca del robot. Los dos motores de la muñeca se mueven con una combinación oportuna de pasos, de tal manera que se obtiene un movimiento rotativo global (ambos motores sufren un incremento o una disminución de un número de pasos).

Cuando la tecla TEST se oprime por octava vez, la muñeca del robot da una vuelta en sentido dextrorso.

Al apretarse por novena vez la tecla TEST, la muñeca se levanta, y los motores de ella se mueven con una combinación de pasos tal que producen un movimiento vertical global.

Cuando se aprieta la tecla TEST por décima vez, la muñeca de baja.

Al apretarse por undécima y duodécima vez la tecla TEST, las pinzas se abren y se cierran respectivamente.

Si se vuelve a oprimir la tecla TEST, el ciclo se repite desde el comienzo.

² La estructura del robot. posee marcas, las cuales indican la posición inicial de cada uno de los ejes.

³ Si se mira la muñeca desde la base del robot.

Durante todas estas fases es aconsejable no superar los límites de desplazamiento de cada eje.

3.2.1.2. MODALIDAD 2

Este tipo de test es muy útil para controlar la sincronización de las articulaciones, la dinámica de una secuencia y la angulación de cada eje durante los diferentes movimientos.

Operaciones a efectuar:

1. Colocar el robot en la posición de cero mecánico.
2. Presionar la tecla RESET para salir de la modalidad 1.
3. Oprimir la tecla RESET y luego la tecla TEST. Soltar la tecla RESET, y tras unos cinco segundos también la tecla TEST.

El robot empieza a efectuar una serie de movimientos que implica todos los ejes. Esta secuencia está memorizada en la RAM del sistema de control del microprocesador. Dicha secuencia se repite con tres valores crecientes de velocidad de los motores. Si se vuelve a apretar la tecla test, se sale de esta fase.

3.3. MODO ORDENADOR PERSONAL

La serie de los movimientos efectuados por el robot se puede controlar totalmente mediante un lenguaje intérprete de tipo BASIC, PASCAL, C, etc.

Los mandos correspondientes a cada actividad del robot se transmiten a la unidad central de control a través de instrucciones particulares. Las posibles elecciones para cada tipo de actuación se efectúan directamente desde el teclado y se completan con la gráfica y los recursos software disponibles en el ordenador a utilizar.

3.4. CABLE PARA LAS CONEXIONES ROBOT-ORDENADOR

ROBOT	PC
3 DB0	2
5 DB1	3
7 DB2	4
9 DB3	5
11 DB4	6
13 DB5	7
15 DB6	8
17 DB7	9
1 STB	1
20 BUSY	11
19 ACK	10
18 ERROR	--
2,4,6 GND	19-30 GND
8,10,12 GND	--
14,16 GND	--

Puerto paralelo. conector DB25

FIGURA 11
ROBOT - VISTA FRONTAL

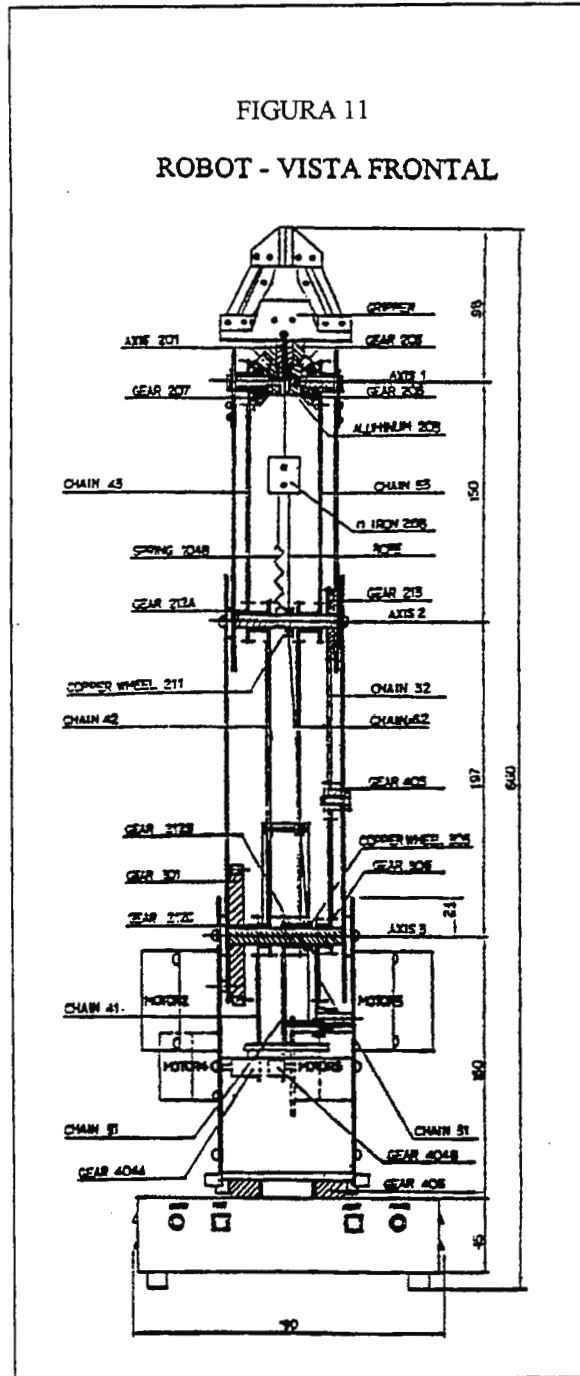


FIGURA 12
ROBOT - VISTA LATERAL DERECHA

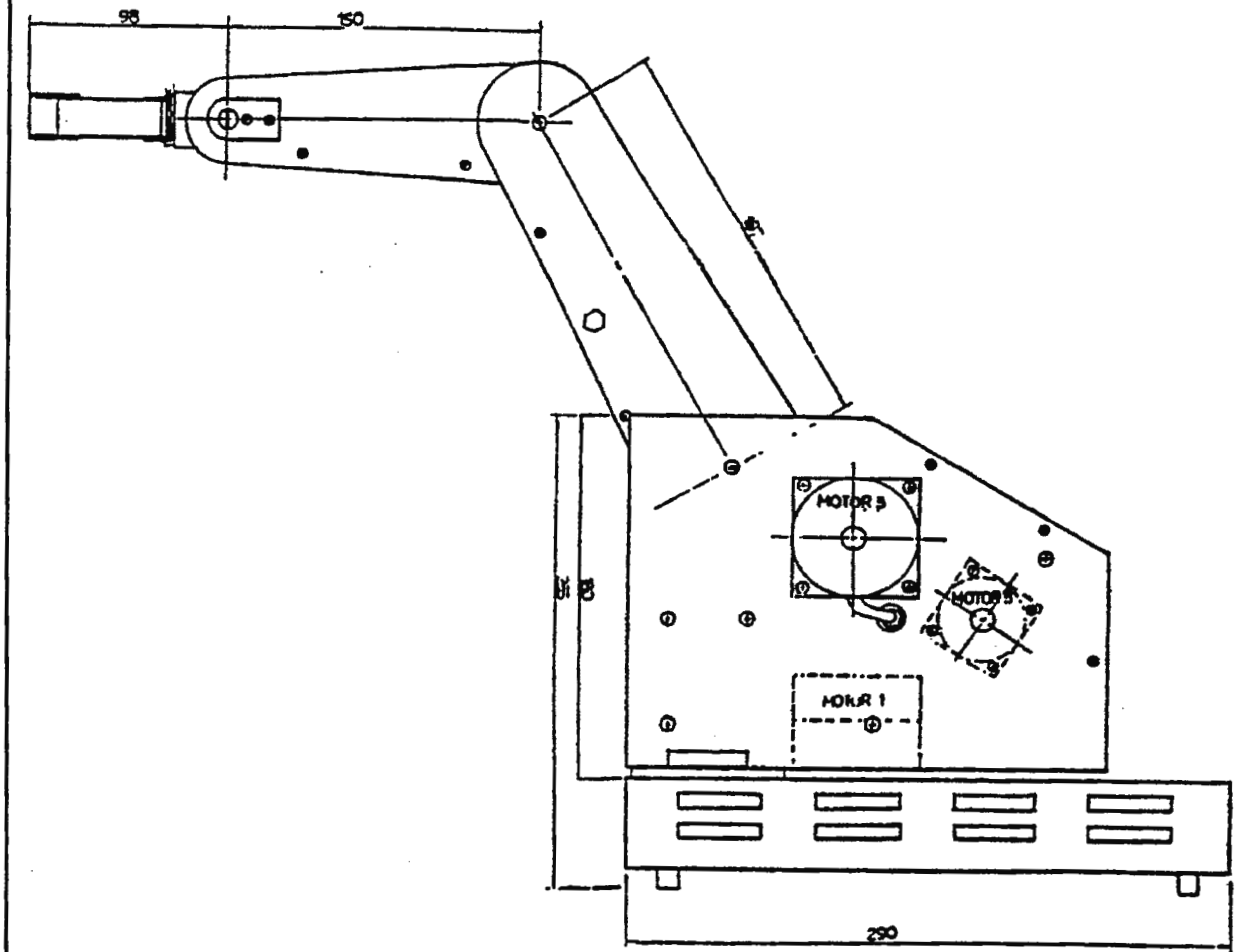
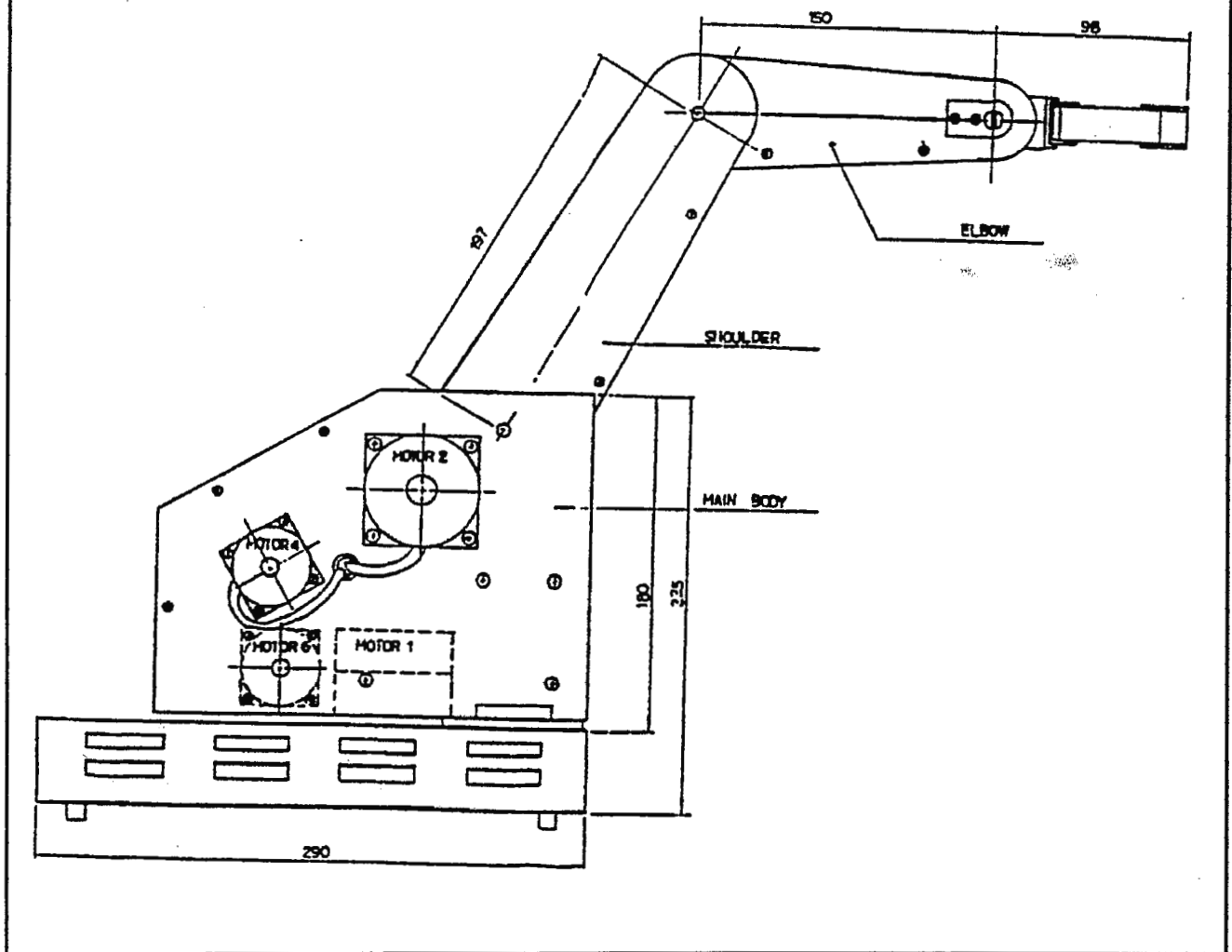


FIGURA 13
ROBOT- VISTA LATERAL IZQUIERDA



3.5. DIAGRAMA DEL CIRCUITO INTERNO DEL ROBOT

El circuito interno del robot, es prácticamente una microcomputadora que recibe ordenes o comandos de una computadora personal a través del puerto paralelo de ésta.

El microprocesador utilizado por el fabricante es el Z80, de 40 pines de los cuales 16 son de direcciones, 8 de datos, los pines de interrupción no son utilizados (MI, RFSH, HALT, WAIT, INT, NMI, BUSRQ, BUSAK, todos ellos activos con bajo) .

El circuito, como era de esperarse, tiene su sistema de memoria, tanto RAM como ROM. La RAM es una HM6116 de 2x8, y la ROM es una 2732 de 4x8, como es de suponer, en la ROM se encuentra el programa monitor que maneja los datos enviados por la computadora personal para los motores que provocan el movimiento del brazo robot.(ver sección 3.7).

En el diagrama que se encuentra en el anexo C, se puede visualizar el circuito interno del robot. Si observamos el circuito, podemos llegar a la conclusión de que es más fácil, tanto para explicar como para entender, llevar dicho circuito a un diagrama de bloques más general, obtenido; claro está, a partir del diagrama circuital.

A continuación se muestra un diagrama de bloques que ayudará a explicar el funcionamiento del circuito mostrado en anexo C.

DIAGRAMA DE BLOQUES

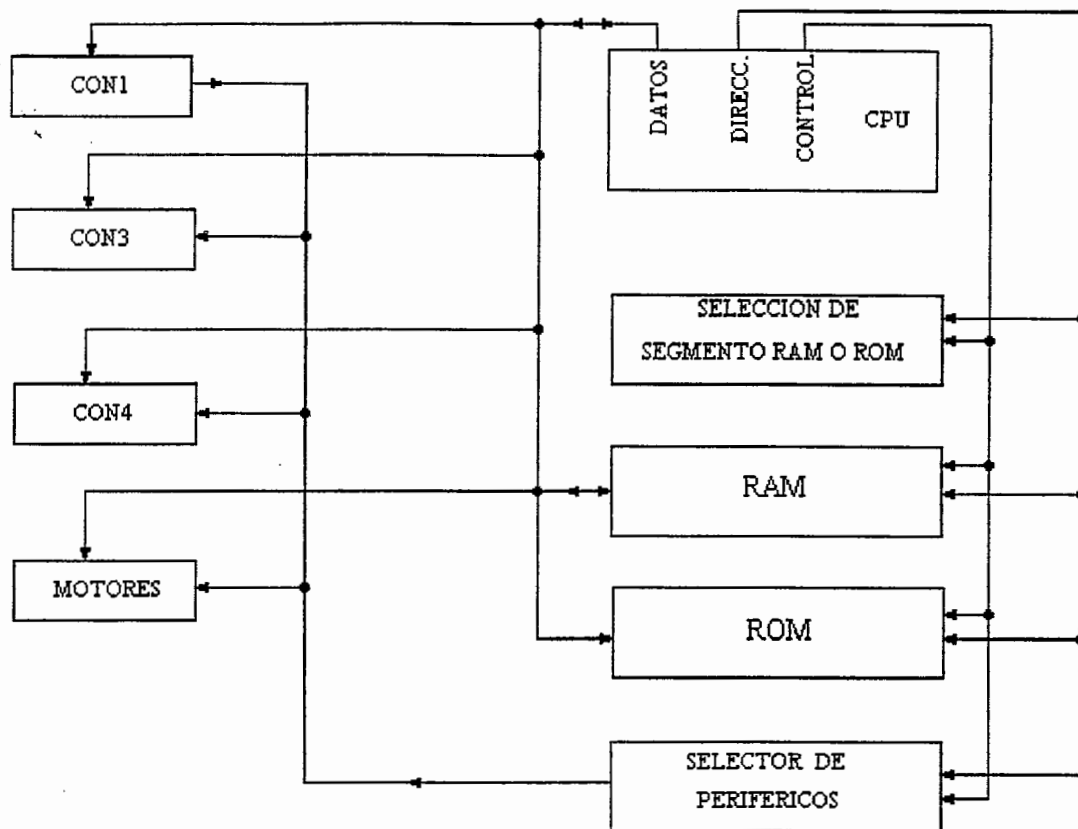


Figura 14 Diagrama a bloques del circuito interno de el CPU Z80 del robot

Basándose en el diagrama de la figura 14, el circuito funciona de la siguiente forma:

Las direcciones provenientes del microprocesador están conectadas al decodificador de memoria, el cual determina si el dispositivo en uso son las memorias o los periféricos. En el caso de seleccionar las memorias, éstas reciben la dirección y al mismo tiempo una habilitación ya sea para escribir (en la RAM) o para leer(ambas memorias) datos.

Si los seleccionados fueron los periféricos , puede ocurrir alguno de los siguientes literales:

- a) Mandar una señal de activación por el puerto (A0), al integrado 74LS273(recibe datos del puerto paralelo de la PC), en cuyo caso se envían datos al puerto paralelo de la computadora personal, aunque ocurra esto; no es de mucho interés ya que el conector (con 3) no está unido al puerto.
- b) Interactuar con la computadora personal, mandado señales a través de CON 1 (algunas de éstas señales son: dato recibido, espere, error y una de las más importantes recibir o leer los datos).
- c) Activar el o los motores, para que se desplacen a una posición determinada por los datos recibidos de la computadora personal. En este caso, también se envían datos al puerto paralelo, pero al igual que en el literal a, no existe la conexión.

3.6 MAPEO DE MEMORIA

MAPA E/S		MAPA E/S	
Desde	Hasta	Desde	Has
0000	0FFF	00	1:
ROM MONITOR		CON 2 (00)	
1000	1FFF	20	3
ROM DE EXPANSIÓN		SEÑAL ACK (20)	
		40	5
		ACTUALIZAR DATO DE ENTRADA (40)	
		60	7
RAM		SEÑAL ERR (60)	
8000	87FF	80	9
RAM DE EXPANSIÓN		SEÑAL TEST (80)	
9000	97FF	A0	B
		CON 3(OUT)(A0) SEÑAL WAIT (IN)(A0)	
		C0	D
		CON 5 - 10	
		E0	F
		MOTORES (E0)	

FIGURA 15: Mapa de memoria y mapa de puertos de entrada/salida

3.7. DESCRIPCION DE EL PROGRAMA MONITOR DEL ROBOT

Para realizar la descripción del programa monitor, se tomará como base el flujograma presentado en el anexo C, el cual se ha derivado del código fuente de dicho programa en lenguaje ensamblador, y que además se presenta en el anexo A.

Al aplicarse la alimentación al robot, comienza el microprocesador interno a ejecutar el programa monitor, comenzando por escribir 00, en todas las localidades de la memoria RAM de la microcomputadora, con el objeto de inicializar la memoria volátil con dicha condición (00). En este punto el sistema no necesita ningún comando o parámetro del ordenador, simplemente es un algoritmo que se ejecuta desde la memoria ROM de la microcomputadora del robot.

Una vez inicializada la RAM, el programa monitor, incluye un conjunto de instrucciones, con los cuales se inicializan los motores del robot, además se toma la posición en la cual se encuentra el mecanismo del autómeta como la posición de inicio (home) o de arranque, a la cual se deberá regresar la estructura del brazo robot, cuando así se lo solicite el programador por medio de los comandos enviados a través de la computadora personal⁴. Para este proceso, al igual que en el anterior, no se necesitan comandos o parámetros externos a la microcomputadora. Si se diera el caso en el que alguno de los motores no responda al proceso de inicialización, en la base del robot se encenderá el indicador de error, y no responderá el sistema a los datos enviados por el ordenador personal o a la tecla TEST. En tal caso se deberá dar RESET al robot y solucionar el problema.

Una vez completado el proceso de inicialización, el programa monitor de la microcomputadora, comienza por leer la tecla de TEST. Si se detecta dicha señal, comienza a ejecutarse una secuencia, de cualquiera de las dos que ya se mencionaron y que el fabricante denomina SELF TEST. El microprocesador del robot, selecciona cual secuencia SELF TEST ejecutará, según el tiempo que se tenga presionada dicha tecla. La diferencia entre ambas secuencias se explicó las secciones 3.2.1.1. y 3.2.1.2.

Al salirse de la secuencia SELF TEST (cualquiera de las dos disponibles), el autómeta vuelve a leer la tecla TEST, y queda esperando por datos provenientes del ordenador que lo controla, al igual que si al leer la tecla antes mencionada, no se detecta la señal que ésta provee, el sistema procede a avisar que está en espera de los datos. Cabe

⁴ Se sugiere que la estructura sea colocada en la posición de arranque denotada por las marcas en el mecanismo.

mencionar que sólo es posible hacer que el robot entre en TEST, si se presiona la tecla correspondiente, inmediatamente después que se ha inicializado el sistema.

Cuando la microcomputadora recibe un dato cualquiera proveniente del ordenador que controla al autómata, el programa monitor, verifica el carácter recibido, el cual puede ser cualquiera de los caracteres ASCII correspondientes al alfabeto inglés en letras mayúsculas⁵, los correspondientes a los números del 0 al 9, o símbolos tales como el punto (.); la coma (,); el signo más (+); el signo menos (-); o el espacio en blanco. La sólo algunos de los caracteres antes mencionados representan entradas válidas para el autómata, cuando el carácter recibido no es un valor válido, se queda en estado de espera por otro dato que sea enviado del ordenador controlador.

A continuación se describe, la acción que ejecutará el autómata, en el caso que reciba cualquiera de los caracteres mencionados:

3.7.1. CASO A, B, E, F, I, J, K, Q, R, T, U, V, W, X, Y,

En el caso que sea recibido el código en hexadecimal correspondiente a las letras anteriores, el programa monitor lo compara con los códigos correspondientes al resto de las letras del alfabeto inglés y con cada uno de los códigos de ellas mismas, el proceso es de comparación y decisión, es decir, se busca tanto datos válidos como no válidos, y al ser encontrado, en el programa se toma la decisión de lo que se debe hacer, lo cual para el presente caso, es el de ignorar los datos y enviar señal de espera por datos al ordenador controlador. Proceso muy diferente al de eliminación. que aunque es más rápido, puede dar

⁵ Necesariamente deben ser letras mayúsculas, ya que las minúsculas son ignoradas, así como los demás caracteres no mencionados.

lugar a errores; el proceso utilizado es factible de usarse en este caso. debido a que por lo general las instrucciones en lenguaje ensamblador son ejecutadas en un tiempo muy corto.

3.7.2. CASO C

En el caso que le sea enviado el código correspondiente a la letra C mayúscula, el programa monitor selecciona un sub-programa, en el cual se le da la orden de cerrar las pinzas del robot⁶.

3.7.3. CASO D

Si el autómata llega a recibir del ordenador el número hexadecimal 44 que es el código correspondiente a la letra D, en el área del programa monitor residente en la memoria ROM, correspondiente a la validación de los datos, selecciona el sub-programa correspondiente a una rutina de retardo de tiempo, mientras se espera por una instrucción; el tiempo que se mantendrá el autómata en este estado, debe ser especificado por una variable, la que indica el número de segundos.

3.7.4. CASO G

Cuando se recibe el código ASCII de la letra G, es seleccionado el sub-programa correspondiente a un comando de movimiento, pero dicho comando además necesita los valores de las variables obtenidas por los comandos H y P, los cuales se explicarán adelante; los valores válidos para este comando, se encuentran dentro del rango de 0-100 en

⁶ En este caso como en los siguientes, los cuales involucren movimiento de cualquier motor, los rangos de posicionamiento máximos y mínimos para cada elemento son verificados antes de efectuar el desplazamiento.

décimal. El valor de esta variable indica que se tiene capacidad para ubicar el efector en un máximo de 100 posiciones diferentes.

3.7.5. CASO H

El código de la letra H, permite seleccionar el comando Here, con el cual se puede guardar el valor actual de las diferentes variables que definen la posición del efector, para que estos puedan ser usados posteriormente por el comando G, el cual lleva al efector a la posición deseada. Cabe mencionar que el número de posiciones seleccionadas, no debe exceder de 100, debido a que se reserva un área de la memoria RAM para almacenar dichos datos.

3.7.6. CASO L1, L0

Si el ASCII de L1 es identificado, es permitido controlar los límites de movimiento con relación a la posición CERO, definida por el usuario, pero siempre se tendrá la misma cantidad de pasos, sólo que se habrá modificado el rango en ambos sentidos, por ejemplo puede reducirse el rango inferior y por consiguiente se habrá aumentado el superior. En caso de que los nuevos límites sean sobrepasados, se emitirá una señal de error para indicarlo y no se permitirá ningún otro ASCII que no sea el correspondiente a N, que devuelve a la posición CERO.

Por otra parte, si el código identificado es el de L0, el control de límites no se efectúa y los límites son los predeterminados por el fabricante, esta es la condición al momento de encender el robot.

momento de ejecutar un comando Go to (G), este toma los valores de las variables, las que le indican la posición a la que deberá ir el efector.

3.7.11. CASO S

Selecciona dentro del programa monitor la velocidad con la que habrá de efectuarse el movimiento de cualquier eje, en movimiento individual; el parámetro de velocidad debe estar en un rango de 1 y 5. De acuerdo con las especificaciones de los dispositivos mecánicos y la resistencia de la estructura, el fabricante sugiere utilizar los valores de 1 y 2 para cuando se desea manejar cargas considerables, y hasta 5 para cargas livianas.

3.7.12. CASO Z

Para el caso que se recibe el código correspondiente a la letra Z, el programa monitor llama el sub-programa, que le permite llevar a toda la estructura del brazo robot hacia la posición actual definida como de inicio. Ello incluye mover los motores los pasos necesarios para lograrlo.

3.7.13. CASO (+); (-)

Los signos más y menos son necesarios para indicarle al sistema si debe mover determinado elemento de la estructura mecánica hacia arriba, abajo, izquierda, derecha, cerrar o abrir. Al igual que lo explicado en la sección anterior el proceso siempre es de comparación y decisión; para el caso que sea detectado cualquiera de los dos signos, el valor de ellos será guardado como una variable temporal de memoria dentro del programa, y como ya se mencionó, el movimiento será regido por ella.

3.7.14. CASO (;); (;); (ESPACIO EN BLANCO)

En cualquiera de los casos que se reciba el código de los caracteres antes mencionados, en el programa monitor se ignoran dichos datos, ya que sólo indican separación entre los diferentes parámetros que le son enviados al programa para ejecutar los diferentes comandos.

4. MEJORAMIENTO DE LA VERSATILIDAD DEL BRAZO ROBOT

De lo expuesto en las secciones anteriores, puede determinarse el tipo de control que se ejerce sobre el brazo robot, el cual es de punto a punto no coordinado, el cual se caracteriza por la forma en la que el autómata alcanza la posición final, partiendo de un punto cualquiera en el espacio de acción para este. En la descripción del software⁷, diseñado para controlar el dispositivo antes mencionado, se hace mención que para el movimiento simultáneo, todos los motores, se desplazan a una misma velocidad angular, lo cual podrá no satisfacer algunas necesidades, en las que se requiere que los motores vayan a una velocidad diferente cada uno, tal es el caso del control punto a punto coordinado, expuesto en la sección 1.5, en el cual se requiere que todos los actuadores del robot completen su rotación en un mismo tiempo, independientemente del ángulo de desplazamiento que deban recorrer.

Lo planteado anteriormente no es posible realizarse, si no se elimina el CPU del robot, ya que es el uso de este que simplifica la forma de control, pero que a su vez restringe la versatilidad del sistema. Y debido a que se planteó originalmente el presente proyecto tal como se expuso en las secciones anteriores, no se concibió la idea de crear un programa capaz de cumplir con la tarea de la microcomputadora interna del autómata, sino más bien se creó uno capaz de interactuar con ésta. Pero a continuación se describe la forma de como puede incrementarse la versatilidad del robot, usando la configuración mecánica de este, sin la necesidad de eliminar el CPU Z80, pudiendo elegirse la forma de control.

⁷ Ver sección 2. Descripción del software

4.1. CARACTERÍSTICAS DEL MECANISMO

El mecanismo del robot, es impulsado por seis motores paso a paso, todos ellos con un ángulo por paso de 15 grados, de lo cual puede deducirse el número de pasos por revolución

$$\text{Total de pasos} = \frac{360 \text{ grados}}{1 \text{ rev.}} \times \frac{1 \text{ paso}}{15 \text{ grados}} = 24 \frac{\text{pasos}}{\text{rev.}}$$

Luego se procedió a determinar la velocidad máxima a la que pueden girar los motores sin ninguna carga en las pinzas, para lo cual de la observación experimental, se obtuvieron los siguientes valores de tiempo promedio y desplazamiento angular, con el fin de poder calcular las respectivas velocidades angulares.

Motor de la base:

$$\text{Pasos} = 1000 \quad \text{pasos/seg.} = \frac{\text{pasos}}{\Delta t_{\text{prom.}}} = \frac{1000 \text{ pasos}}{7.11 \text{ seg.}} = 140.646 \frac{\text{pasos}}{\text{seg.}}$$

$$\Delta t_{\text{prom.}} = 7.11 \text{ seg.} \quad \omega = \frac{\Delta \theta}{\Delta t_{\text{prom.}}} = \frac{120 \text{ grados}}{7.11 \text{ seg.}} \times \frac{2\pi \text{ rad}}{360 \text{ grados}} = 0.09376\pi \frac{\text{rad.}}{\text{seg.}}$$

$$\Delta \theta = 120 \text{ grados}$$

Motor de la espalda:

$$\Delta \theta = 72 \text{ grados} \quad \text{pasos/seg.} = \frac{\text{pasos}}{\Delta t_{\text{prom.}}} = \frac{600 \text{ pasos}}{4.21 \text{ seg.}} = 142.517 \frac{\text{pasos}}{\text{seg.}}$$

$$\Delta t_{\text{prom.}} = 4.21 \text{ seg.} \quad \omega = \frac{\Delta \theta}{\Delta t_{\text{prom.}}} = \frac{72 \text{ grados}}{4.21 \text{ seg.}} \times \frac{2\pi \text{ rad}}{360 \text{ grados}} = 0.0950\pi \frac{\text{rad.}}{\text{seg.}}$$

$$\text{pasos} = 600$$

Motor del antebrazo:

$$\Delta\theta = 50 \text{ grados} \quad \text{pasos/seg.} = \frac{\text{pasos}}{\Delta t_{\text{prom.}}} = \frac{500 \text{pasos}}{3.55 \text{seg.}} = 140.845 \frac{\text{pasos}}{\text{seg.}}$$

$$\Delta t_{\text{prom.}} = 3.55 \text{ seg.} \quad \omega = \frac{\Delta\theta}{\Delta t_{\text{prom.}}} = \frac{50 \text{grados}}{2.93 \text{seg.}} \times \frac{2\pi \text{rad}}{360 \text{grados}} = 0.09480\pi \frac{\text{rad.}}{\text{seg.}}$$

pasos = 500

Elevación de la muñeca:

$$\Delta\theta = 90 \text{ grados} \quad \text{pasos/seg.} = \frac{\text{pasos}}{\Delta t_{\text{prom.}}} = \frac{900 \text{pasos}}{6.37 \text{seg.}} = 141.287 \frac{\text{pasos}}{\text{seg.}}$$

$$\Delta t_{\text{prom.}} = 6.37 \text{ seg.} \quad \omega = \frac{\Delta\theta}{\Delta t_{\text{prom.}}} = \frac{90 \text{grados}}{5.31 \text{seg.}} \times \frac{2\pi \text{rad}}{360 \text{grados}} = 0.09416\pi \frac{\text{rad.}}{\text{seg.}}$$

pasos = 900

Rotación de la muñeca:

$$\Delta\theta = 180 \text{ grados} \quad \text{pasos/seg.} = \frac{\text{pasos}}{\Delta t_{\text{prom.}}} = \frac{1800 \text{pasos}}{12.75 \text{seg.}} = 141.176 \frac{\text{pasos}}{\text{seg.}}$$

$$\Delta t_{\text{prom.}} = 12.75 \text{ seg.} \quad \omega = \frac{\Delta\theta}{\Delta t_{\text{prom.}}} = \frac{180 \text{grados}}{10.56 \text{seg.}} \times \frac{2\pi \text{rad}}{360 \text{grados}} = 0.09469\pi \frac{\text{rad.}}{\text{seg.}}$$

pasos = 1800

Pinzas:

$$\Delta d = 45\text{mm} \quad \text{pasos/seg.} = \frac{\text{pasos}}{\Delta t_{\text{prom.}}} = \frac{1800\text{pasos}}{12.05\text{seg.}} = 14.937 \frac{\text{pasos}}{\text{seg.}}$$

$$\Delta t_{\text{prom.}} = 12.05 \text{ seg.}$$

$$\text{pasos} = 1800$$

Para el caso de la pinza, no fue posible encontrar una relación entre el número de pasos por segundo y la velocidad angular, debido a que dentro del mecanismo, se tiene un resorte, al cual el cable que controla la apertura o cierre de la pinza, debe vencer primero, para ello se hace necesario conocer el torque que hace el motor respectivo y por la ley de hooke podría sacarse la relación, pero no se encontró mayor especificación técnica sobre motores paso a paso.

4.2. DESCRIPCIÓN DEL CIRCUITO PARA CONTROL DE MOTORES Y FORMA DE CONECTARLO.

4.2.1. DESCRIPCIÓN.

El circuito antes mencionado, se muestra en el anexo C, el cual se ha ideado, para que reciba datos desde el puerto paralelo, y al iniciarse el programa controlador, dicho circuito permite el uso del CPU Z80 del robot

Cuando se envíe por el puerto paralelo, el código binario del carácter ASCII "B", cuando se presenta en el bus de datos de dicho puerto, el número 42_{hex} . El cual al pasarlo a su equivalente binario(01000010), puede verse la peculiaridad que en el bit 6 y en el bit 1, se encuentran en alto, con lo cual la compuerta NAND 7430, la cual es de 8 entradas, a través de la negación de las entradas conectadas a los bits respectivos del bus de datos,

tendrá un cero, lo cual hará que ésta proporcione un uno lógico a su salida. con lo cual se producirá una transición de alto a bajo, que será recibida por el flip-flop JK 7476 en la entrada CK, lo que conmutará la salida Q se fija en uno, la entrada 2G de los circuitos 74241, permita que las entradas 2A1, hasta la 2A4, dejen pasar el dato presente en las líneas antes mencionadas a las salidas 2Y1 hasta la 2Y4, con todo ello, cualquier dato presente en el puerto paralelo, pasará a las entradas de los flip-flops tipo D 7475 para el caso de el 74241 situado en la parte superior de la página, y los otros cuatro bits restantes que pasen a través de los buffers 74241 del medio del circuito deberán tomarse 3, los correspondientes a los bits 4,5 y 7, el motivo de hacerlo así, es debido, a que se requiere que en los bits 2 y 6 se tenga un estado lógico uno, para que el circuito haga la conmutación de hacia donde irán los datos, por lo que debe tenerse en cuenta que para seleccionar el motor que habrá de enviársele la secuencia, se hace necesario hacer una cuenta binaria de 0 a 5, y la no utilización del bit 6 impide una conmutación involuntaria, por error del programador a la hora de escribir el programa.

Con el software adecuado los datos son enviados a través de este circuito, y almacenados en los flip-flop tipo D tomando en cuenta que debe mandarse en los primeros cuatro bits, del cero al tres, la combinación necesaria para que cada motor se desplace un paso, y en los bits 4,5 y 7 sea enviado el número correspondiente a que motor se le están enviando datos. Una vez los datos estén cargado en los registros para cada motor, con la etapa de potencia, que es común tanto para este circuito como para el que está dentro de la estructura de robot, será posible que los motores sean excitados debidamente.

Para seleccionar de nuevo el control vía CPU Z80, debe enviársele de nuevo el Número 42_{hex.}, que corresponde al ASCII "B", lo cual de nuevo ocasionará una transición positiva cuando la NAND 7430 cambie su salida de uno a cero, haciendo que el flip-flop

7476 bascule hacia cero, entonces 1G tiene cero, permitiendo que las entradas 1A1 hasta 1A4 hagan pasar los datos a sus respectivas salidas y hacia CON 1 del circuito que compone a la microcomputadora que ayuda al control del brazo robot que también se muestra en el anexo C.

4.2.2. FORMA DE CONEXIÓN

El circuito descrito anteriormente, deberá conectarse entre el puerto paralelo y el brazo robot, teniendo en cuenta, que deberán desconectarse los cables mediante los cuales se alimentan los motores desde la etapa de potencia , y conectar estos a un área que debe ser incluida en la tarjeta de circuito impreso que se elabore para implementar el circuito selector, a continuación en la figura 16 se muestra como deberá conectarse el circuito antes mencionado.

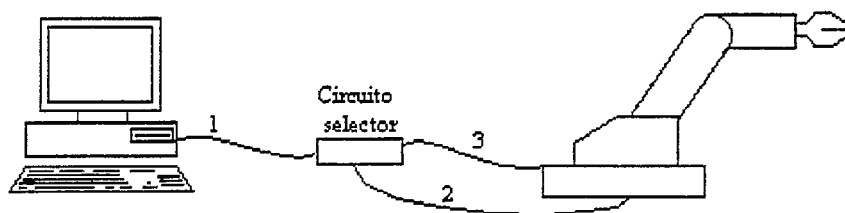


Figura 16: Disposición física del circuito, computadora y robot

En la figura anterior, el cable señalado con el número 1, es el que debe conectarse al puerto paralelo, el número 2, será el que vaya del circuito impreso del CPU interno del robot y el número 3, el que alimente los motores para cualquiera de las etapas de potencia que se encuentre trabajando. Ya que se usan los IC's 1749, y estos al no recibir un voltaje

en su entrada, se desconectan, eliminan la posibilidad de cortocircuito en la conexión mostrada en la figura 17 donde se sugiere una forma de llevar un solo cable hacia los motores, agregando un conector extra para los cables provenientes del brazo robot, pero no por ello, se restringe a que sea esta la forma, ya que puede realizarse de muchas distintas e incluso mejores en cuanto a economía que la presentada, por lo tal ello queda a elección del lector que decida implementar el circuito.

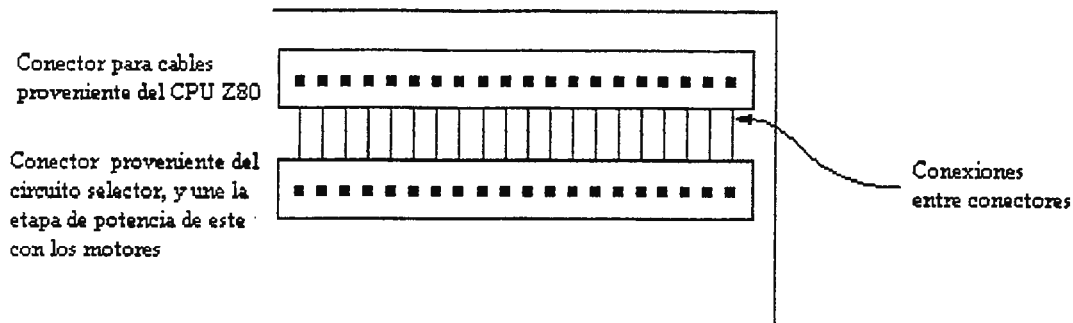


Figura 17: Tarjeta de circuito impreso con las conexiones sugeridas.

4.3. MODIFICACIONES EN EL SOFTWARE

Cuando sea enviado el número 42_{hex} . Después de abandonar el control directo de los motores, el brazo robot continuará siendo controlado por el software descrito anteriormente, con la variante que debe agregársele en el submenú ejecutar, la opción que permita conmutar el control vía CPU Z80 o entre el control directo del programa de los motores del robot, esto en cuanto a presentación, pero, internamente al programa debe especificársele, que cuando se haga dicha selección pidiéndole hacer la conmutación para poder tener acceso directo a los motores, este envíe cada vez el comando $LPRINT\ 66_{dec}$, que es el equivalente en decimal para el código ASCII "B".

Todo lo anterior, implica que el software creado deberá enviar una vez se seleccione el acceso directo a los motores, las secuencias de movimiento, para ello deberá crearse un subprograma que haga el desplazamiento de un bit en un grupo de 4 bits, y además indique a que motor se le enviará determinada combinación, con un retardo de tiempo justo para que los motores sean capaces de reaccionar a las combinaciones presentes en sus bobinas, dicho retardo deberá ser calculado por el programador. El subprograma deberá llamarse cada vez que sea necesario que el robot se mueva, el comando LPRINT con los demás parámetros utilizados en el software que se ha creado, deberá cambiarse por una llamada a este subprograma, pasándole las variables de los desplazamientos como compartidas o de uso exclusivo del mismo.

Todo lo anteriormente expuesto, es posible realizarlo, si se estudia este documento, ya que se ha presentado en secciones anteriores, los pasos que pueden moverse los diferentes motores en función del elemento de la estructura del autómata que manejan; además, se presentan datos de la cinemática del robot, según lo sugerido por el fabricante del mismo, el código fuente programa usado para el presente proyecto, información de cómo se efectúa el control por medio del microprocesador Z80 y la computadora, etc.

Al tener control directo sobre los motores que impulsan al brazo robot, es posible entonces implementar el control punto a punto coordinado, para ello se hizo el estudio de la cinemática del robot, se explicó el funcionamiento del circuito selector para la forma de control y las modificaciones que deberán agregarse al programa presentado.

Para cumplir con la forma de control de la trayectoria antes mencionada, deberá agregarse una opción que permita que todos los motores comiencen un movimiento simultáneamente y lo terminen en el mismo tiempo. para ello se hace necesario especificar los pasos que deberá moverse cada motor y determinar cual de todos tiene el mayor

desplazamiento angular, obtenido este dato debe calcularse el intervalo de tiempo que durará e movimiento, recordando que :

$$\omega = \frac{\Delta\theta}{\Delta t} \quad \text{y} \quad \Delta t = \frac{\Delta\theta}{\omega_{\text{máx.}}}$$

Donde Δt será el intervalo antes mencionado, y deberá ser común para todos los motores, a partir de ahí tomando en cuenta que la velocidad angular máxima es $\omega = 0.095\pi$ rad/seg., luego se debe proceder a calcular la velocidad angular de cada motor, teniendo en cuenta que, el que posea mayor módulo de desplazamiento angular, deberá moverse a la velocidad angular máxima. Una vez determinadas las velocidades particulares, deberá relacionarse está, con la frecuencia a la que deberá enviarse la secuencia de movimiento a cada motor, determinando un retardo de tiempo tal que haga que se envíe la excitación en el tiempo adecuado. El retardo puede hacerse desde QBASIC si se desea, pero tómese en cuenta que la variable TIME\$, sólo permite incrementos de un segundo como mínimo, pero se aconseja utilizar el lenguaje ensamblador para ello, ya que usando la función $AH=2C_{\text{hex}}$. De la interrupción 21h del DOS, la cual da una precisión de centésimas de segundos; en caso de requerirse una mayor precisión deberá usarse un programa de retraso en ensamblador, calculando el número de ciclos que le toma a la máquina el ejecutar el programa, pero sólo puede crearse un programa especialmente para una computadora con una velocidad de reloj específica.

En los pasos anteriores deberá tenerse un gran cuidado con el sentido de rotación de los motores, ya que no todos se mueven en el mismo sentido, debido a la disposición física de cada uno en la estructura de robot.

CONCLUSIONES

De la investigación realizada, y de lo observado por experimentación con el brazo robot de Electrónica Venetta, se ha llegado a la conclusión que por la forma como se realiza el control de dicho dispositivo, a éste, se le ejerce un control punto a punto no coordinado.

También se concluye que para el presente caso, la gran sencillez de la forma de manejar al brazo robot, la cual se ha tipificado anteriormente, tiene como consecuencia una reducción en la versatilidad de trabajo del sistema.

Por otra parte se ha podido observar que el uso de movimientos relativos permite realizar acciones mucho más complejas de una forma aparentemente “más sencilla”. No todas las áreas presentan el desplazamiento relativo.

Se debe permitir, que en las demás secciones, se pueda usar el origen físico de la estructura, por la razón de que todos los movimientos iniciales, y todos los desplazamientos requeridos en el caso de “movimiento por eje”, que se le ordenen al robot, deben tener como referencia de partida dicho punto, porque de lo contrario se ocasionará una incongruencia entre los límites mecánicos inherentes de cada motor y los pre-establecidos por software.

BIBLIOGRAFIA

Angulo, José María; CURSO DE ROBOTICA, Editorial Paraninfo S.A., Madrid, 1984.

Lasheras, José María, INTRODUCCION AL CONTROL NUMERICO Y ROBOTICA,
Editorial CEDEL, Barcelona, 1986.

Nichols, Joshep C., PROGRAMACION E INTERFACE CON EL MICROPROCESADOR
Z-80, Editorial MARCOMBO, Barcelona, 1984.

ANEXO A

Código fuente en lenguaje ensamblador
del programa monitor

0000 C33800 JP 38	0050 D3A0 OUT (A0),A
0003 00 NOP	0052 D360 OUT (60),A
0004 00 NOP	0054 3E03 LD A,3
0005 00 NOP	0056 CD3F0A CALL A3F
0006 00 NOP	0059 0603 LD B,3
0007 00 NOP	005B AF XOR A
0008 00 NOP	005C D3E0 OUT (E0),A
0009 00 NOP	005E C610 ADD A,10
000A 00 NOP	0060 10FA DJNZ 5C
000B 00 NOP	0062 0603 LD B,3
000C 00 NOP	0064 3E30 LD A,30
000D 00 NOP	0066 218980 LD HL,8089
000E 00 NOP	0069 D3E0 OUT (E0),A
000F 00 NOP	006B 3600 LD (HL),0
0010 00 NOP	006D C610 ADD A,10
0011 00 NOP	006F 23 INC HL
0012 00 NOP	0070 10F7 DJNZ 69
0013 00 NOP	0072 0E00 LD C,0
0014 00 NOP	0074 3A1180 LD A,(8011)
0015 00 NOP	0077 6F LD L,A
0016 00 NOP	0078 DB80 IN A,(80)
0017 00 NOP	007A F6FE OR FE
0018 00 NOP	007C 2F CPL
0019 00 NOP	007D 321180 LD (8011),A
001A 00 NOP	0080 BD CP L
001B 00 NOP	0081 C2A400 JP NZ,A4
001C 00 NOP	0084 0C INC C
001D 00 NOP	0085 79 LD A,C
001E 00 NOP	0086 FE04 CP 4
001F 00 NOP	0088 DAA600 JP C,A6
0020 00 NOP	008B 3A1180 LD A,(8011)
0021 00 NOP	008E 1F RRA
0022 00 NOP	008F D2B200 JP NC,B2
0023 00 NOP	0092 3E01 LD A,1
0024 00 NOP	0094 320380 LD (8003),A
0025 00 NOP	0097 210880 LD HL,8008
0026 00 NOP	009A 34 INC (HL)
0027 00 NOP	009B 7E LD A,(HL)
0028 00 NOP	009C FEB8 CP B8
0029 00 NOP	009E D2E502 JP NC,2E5
002A 00 NOP	00A1 C3A400 JP A4
002B 00 NOP	00A4 0E00 LD C,0
002C 00 NOP	00A6 215804 LD HL,458
002D 00 NOP	00A9 2B DEC HL
002E 00 NOP	00AA 7C LD A,H
002F 00 NOP	00AB B5 OR L
0030 00 NOP	00AC C2A900 JP NZ,A9
0031 00 NOP	00AF C37400 JP 74
0032 00 NOP	00B2 3A0380 LD A,(8003)
0033 00 NOP	00B5 FE01 CP 1
0034 00 NOP	00B7 CA3B01 JP Z,13B
0035 00 NOP	00BA CDF809 CALL 9F8
0036 00 NOP	00BD 3A0280 LD A,(8002)
0037 00 NOP	00C0 1F RRA
0038 314E83 LD SP,834E	00C1 DAD903 JP C,3D9
003B 210080 LD HL,8000	00C4 D320 OUT (20),A
003E 010020 LD BC,2000	00C6 DB40 IN A,(40)
0041 1600 LD D,0	00C8 1F RRA
0043 72 LD (HL),D	00C9 D2C600 JP NC,C6
0044 23 INC HL	00CC DB00 IN A,(0)
0045 0B DEC BC	00CE FE41 CP 41
0046 78 LD A,B	00D0 DAC400 JP C,C4
0047 B1 OR C	00D3 FE5B CP 5B
0048 C24300 JP NZ,43	00D5 D2C400 JP NC,C4
004B 3E60 LD A,60	00D8 E61F AND 1F
004D D3E0 OUT (E0),A	00DA 47 LD B,A
004F AF XOR A	00DB 87 ADD A,A

```

00DC 80  ADD A,B
00DD 2600 LD H,0
00DF 6F  LD L,A
00E0 01E700 LD BC,E7
00E3 09  ADD HL,BC
00E4 D320  OUT (20),A
00E6 E9  JP (HL)
00E7 C3D903 JP 3D9
00EA C3D903 JP 3D9
00ED C3D903 JP 3D9
00F0 C35E0A JP A5E
00F3 C3930E JP E93
00F6 C3D903 JP 3D9
00F9 C3D903 JP 3D9
00FC C3590D JP D59
00FF C39D0A JP A9D
0102 C3D903 JP 3D9
0105 C3D903 JP 3D9
0108 C3D903 JP 3D9
010B C3250D JP D25
010E C3570B JP B57
0111 C3BB0D JP DBB
0114 C3D20A JP AD2
0117 C3FE0D JP DFE
011A C33B0E JP E3B
011D C3D90D JP DD9
0120 C3640E JP E64
0123 C3D903 JP 3D9
0126 C3D30E JP ED3
0129 C3DB0E JP EDB
012C C35E0F JP F5E
012F C3D903 JP 3D9
0132 C3D903 JP 3D9
0135 C30E0B JP B0E
0138 C3D903 JP 3D9
013B CDF809 CALL 9F8
013E 3A0280 LD A,(8002)
0141 1F  RRA
0142 D24901 JP NC,149
0145 3EFF  LD A,FF
0147 D3A0  OUT (A0),A
0149 210200 LD HL,2
014C 226C80 LD (806C),HL
014F 29  ADD HL,HL
0150 226A80 LD (806A),HL
0153 0606  LD B,6
0155 218C80 LD HL,808C
0158 3602  LD (HL),2
015A 23  INC HL
015B 3600  LD (HL),0
015D 23  INC HL
015E 10F8  DJNZ 158
0160 3E40  LD A,40
0162 320080 LD (8000),A
0165 AF  XOR A
0166 321580 LD (8015),A
0169 AF  XOR A
016A 321280 LD (8012),A
016D 320A80 LD (800A),A
0170 320680 LD (8006),A
0173 320980 LD (8009),A
0176 320880 LD (8008),A
0179 320780 LD (8007),A
017C 320380 LD (8003),A
017F 320580 LD (8005),A
0182 321180 LD (8011),A
0185 060C  LD B,C
0187 217480 LD HL,8074
018A 77  LD (HL),A
018B 23  INC HL
018C 10FC  DJNZ 18A
018E 3A1580 LD A,(8015)
0191 4F  LD C,A
0192 C661  ADD A,61
0194 D3E0  OUT (E0),A
0196 CB39  SRL C
0198 79  LD A,C
0199 0600  LD B,0
019B 218080 LD HL,8080
019E 09  ADD HL,BC
019F 3A1580 LD A,(8015)
01A2 CB3F  SRL A
01A4 DAC201 JP C,1C2
01A7 36FF  LD (HL),FF
01A9 FE03  CP 3
01AB CAB601 JP Z,1B6
01AE FE04  CP 4
01B0 CAB001 JP Z,1B0
01B3 C3DD01 JP 1DD
01B6 23  INC HL
01B7 36FF  LD (HL),FF
01B9 C3DD01 JP 1DD
01BC 2B  DEC HL
01BD 3601  LD (HL),1
01BF C3DD01 JP 1DD
01C2 3601  LD (HL),1
01C4 FE04  CP 4
01C6 CAD701 JP Z,1D7
01C9 FE03  CP 3
01CB CAD101 JP Z,1D1
01CE C3DD01 JP 1DD
01D1 23  INC HL
01D2 3601  LD (HL),1
01D4 C3DD01 JP 1DD
01D7 2B  DEC HL
01D8 36FF  LD (HL),FF
01DA C3DD01 JP 1DD
01DD 217480 LD HL,8074
01E0 09  ADD HL,BC
01E1 09  ADD HL,BC
01E2 3601  LD (HL),1
01E4 23  INC HL
01E5 3600  LD (HL),0
01E7 FE04  CP 4
01E9 CAF401 JP Z,1F4
01EC FE03  CP 3
01EE CAFE01 JP Z,1FE
01F1 C30402 JP 204
01F4 2B  DEC HL
01F5 2B  DEC HL
01F6 3600  LD (HL),0
01F8 2B  DEC HL
01F9 3601  LD (HL),1
01FB C30402 JP 204
01FE 23  INC HL
01FF 3601  LD (HL),1
0201 23  INC HL
0202 3600  LD (HL),0
0204 3A1180 LD A,(8011)
0207 6F  LD L,A
0208 DB80  IN A,(80)
020A F6FE  OR FE
020C 2F  CPL
020D 321180 LD (8011),A
0210 BD  CPL
0211 C22A02 JP NZ,22A
0214 210780 LD HL,8007
0217 34  INC (HL)
0218 7E  LD A,(HL)
0219 FE03  CP 3

```

021B DA3202 JP C.232
 021E 3600 LD (HL),0
 0220 3A1180 LD A,(8011)
 0223 1F RRA
 0224 DA3C02 JP C.23C
 0227 C39B02 JP 29B
 022A 210780 LD HL,8007
 022D 3600 LD (HL),0
 022F E5 PUSH HL
 0230 00 NOP
 0231 E1 POP HL
 0232 3A0380 LD A,(8003)
 0235 1F RRA
 0236 DA3C02 JP C.23C
 0239 C39B02 JP 29B
 023C 3E01 LD A,1
 023E 320380 LD (8003),A
 0241 3A0580 LD A,(8005)
 0244 1F RRA
 0245 DA7B02 JP C.27B
 0248 3A1580 LD A,(8015)
 024B 1F RRA
 024C 5F LDE,A
 024D 4F LDC,A
 024E 0600 LD B,0
 0250 218680 LD HL,8086
 0253 09 ADD HL,BC
 0254 7E LD A,(HL)
 0255 21800F LD HL,F80
 0258 4F LDC,A
 0259 09 ADD HL,BC
 025A AF XOR A
 025B 7B LD A,E
 025C 07 RLCA
 025D 07 RLCA
 025E 07 RLCA
 025F 07 RLCA
 0260 86 ADD A,(HL)
 0261 D3E0 OUT (E0),A
 0263 210880 LD HL,8008
 0266 34 INC (HL)
 0267 7E LD A,(HL)
 0268 213D09 LD HL,93D
 026B FE07 CP 7
 026D DADC02 JP C,2DC
 0270 3E01 LD A,1
 0272 320580 LD (8005),A
 0275 213C09 LD HL,93C
 0278 C3DC02 JP 2DC
 027B 210000 LD HL,0
 027E 220B80 LD (800B),HL
 0281 2A6E80 LD HL,(806E)
 0284 227280 LD (8072),HL
 0287 210400 LD HL,4
 028A CDB606 CALL 6B6
 028D 3A0280 LD A,(8002)
 0290 1F RRA
 0291 D20402 JP NC,204
 0294 3EFF LD A,FF
 0296 D3A0 OUT (A0),A
 0298 C30402 JP 204
 029B 3A0380 LD A,(8003)
 029E 1F RRA
 029F DAC102 JP C,2C1
 02A2 210980 LD HL,8009
 02A5 34 INC (HL)
 02A6 7E LD A,(HL)
 02A7 214209 LD HL,942
 02AA FE07 CP 7
 02AC DAD602 JP C,2D6
 02AF 0606 LD B,6
 02B1 AF XOR A
 02B2 320980 LD (8009),A
 02B5 D3E0 OUT (E0),A
 02B7 C610 ADD A,10
 02B9 10FA DJNZ 2B5
 02BB 213E09 LD HL,93E
 02BE C3DC02 JP 2DC
 02C1 212409 LD HL,924
 02C4 2B DEC HL
 02C5 7C LD A,H
 02C6 B5 OR L
 02C7 C2C402 JP NZ,2C4
 02CA 3A1580 LD A,(8015)
 02CD 3C INC A
 02CE FE0C CP C
 02D0 D26501 JP NC,165
 02D3 C36601 JP 166
 02D6 DBC0 IN A,(C0)
 02D8 E6FC AND FC
 02DA D360 OUT (60),A
 02DC 2B DEC HL
 02DD 7C LD A,H
 02DE B5 OR L
 02DF C2DC02 JP NZ,2DC
 02E2 C30402 JP 204
 02E5 CDF809 CALL 9F8
 02E8 3A0280 LD A,(8002)
 02EB 1F RRA
 02EC D2F302 JP NC,2F3
 02EF 3EFF LD A,FF
 02F1 D3A0 OUT (A0),A
 02F3 3E80 LD A,80
 02F5 320080 LD (8000),A
 02F8 AF XOR A
 02F9 320480 LD (8004),A
 02FC 320380 LD (8003),A
 02FF 321180 LD (8011),A
 0302 320780 LD (8007),A
 0305 3E01 LD A,1
 0307 321080 LD (8010),A
 030A 21AE0F LD HL,FAE
 030D 221680 LD (8016),HL
 0310 3A1080 LD A,(8010)
 0313 CD3F0A CALL A3F
 0316 AF XOR A
 0317 320F80 LD (800F),A
 031A 2A1680 LD HL,(8016)
 031D 010C00 LD BC,C
 0320 117480 LD DE,8074
 0323 EDB0 LDIR
 0325 CDA705 CALL 5A7
 0328 3A0280 LD A,(8002)
 032B 1F RRA
 032C D23303 JP NC,333
 032F 3EFF LD A,FF
 0331 D3A0 OUT (A0),A
 0333 06FF LD B,FF
 0335 218E02 LD HL,28E
 0338 2B DEC HL
 0339 7C LD A,H
 033A B5 OR L
 033B C23803 JP NZ,338
 033E CD9503 CALL 395
 0341 10F2 DJNZ 335
 0343 3A0F80 LD A,(800F)
 0346 3C INC A
 0347 FE06 CP 6
 0349 D25903 JP NC,359
 034C 2A1680 LD HL,(8016)

```

034F 010C00 LD BC,C
0352 09 ADD HL,BC
0353 221680 LD (8016),HL
0356 C31703 JP 317
0359 3A1080 LD A,(8010)
035C C602 ADD A,2
035E FE07 CP 7
0360 D26903 JP NC,369
0363 321080 LD (8010),A
0366 C30A03 JP 30A
0369 3A0380 LD A,(8003)
036C FE01 CP 1
036E C2F802 JP NZ,2F8
0371 AF XOR A
0372 320480 LD (8004),A
0375 320380 LD (8003),A
0378 321180 LD (8011),A
037B 320780 LD (8007),A
037E CD9503 CALL 395
0381 218E02 LD HL,28E
0384 2B DEC HL
0385 7C LD A,H
0386 B5 OR L
0387 C28403 JP NZ,384
038A 3A0380 LD A,(8003)
038D FE01 CP 1
038F CAF802 JP Z,2F8
0392 C37E03 JP 37E
0395 3A1180 LD A,(8011)
0398 6F LD L,A
0399 DB80 IN A,(80)
039B F6FE OR FE
039D 2F CPL
039E 321180 LD (8011),A
03A1 BD CPL
03A2 C2CA03 JP NZ,3CA
03A5 210780 LD HL,8007
03A8 34 INC (HL)
03A9 7E LD A,(HL)
03AA FE07 CP 7
03AC DAD203 JP C,3D2
03AF 3600 LD (HL),0
03B1 210480 LD HL,8004
03B4 3A1180 LD A,(8011)
03B7 1F RRA
03B8 D2C203 JP NC,3C2
03BB 7E LD A,(HL)
03BC 320380 LD (8003),A
03BF C3D803 JP 3D8
03C2 3601 LD (HL),1
03C4 210000 LD HL,0
03C7 C3D803 JP 3D8
03CA 210780 LD HL,8007
03CD 3600 LD (HL),0
03CF E5 PUSH HL
03D0 E1 POP HL
03D1 00 NOP
03D2 2E05 LD L,5
03D4 2D DEC L
03D5 C2D403 JP NZ,3D4
03D8 C9 RET
03D9 3EFF LD A,FF
03DB D3A0 OUT (A0),A
03DD D320 OUT (20),A
03DF 0601 LD B,1
03E1 10FE DJNZ 3E1
03E3 DB40 IN A,(40)
03E5 1F RRA
03E6 D2E303 JP NC,3E3
03E9 DB00 IN A,(0)
03EB FE4E CP 4E
03ED C2DD03 JP NZ,3DD
03F0 D320 OUT (20),A
03F2 214A80 LD HL,804A
03F5 3600 LD (HL),0
03F7 214B80 LD HL,804B
03FA 3600 LD (HL),0
03FC AF XOR A
03FD 320080 LD (8000),A
0400 CD1404 CALL 414
0403 3A0280 LD A,(8002)
0406 FE01 CP 1
0408 CADD03 JP Z,3DD
040B CD1B0A CALL A1B
040E 00 NOP
040F 00 NOP
0410 00 NOP
0411 00 NOP
0412 00 NOP
0413 76 HALT
0414 AF XOR A
0415 062A LD B,2A
0417 211880 LD HL,8018
041A 77 LD (HL),A
041B 23 INC HL
041C 10FC DJNZ 41A
041E 0607 LD B,7
0420 214280 LD HL,8042
0423 77 LD (HL),A
0424 23 INC HL
0425 10FC DJNZ 423
0427 214980 LD HL,8049
042A 77 LD (HL),A
042B 010000 LD BC,0
042E 110000 LD DE,0
0431 C33604 JP 436
0434 D320 OUT (20),A
0436 DB40 IN A,(40)
0438 1F RRA
0439 D23604 JP NC,436
043C DB00 IN A,(0)
043E FE20 CP 20
0440 CA3404 JP Z,434
0443 FE0E CP E
0445 D26104 JP NC,461
0448 3A0080 LD A,(8000)
044B 214980 LD HL,8049
044E CB47 BIT 0,A
0450 CA5404 JP Z,454
0453 34 INC (HL)
0454 7E LD A,(HL)
0455 214A80 LD HL,804A
0458 BE CP (HL)
0459 C2C104 JP NZ,4C1
045C AF XOR A
045D 320280 LD (8002),A
0460 C9 RET
0461 FE2C CP 2C
0463 C27904 JP NZ,479
0466 214980 LD HL,8049
0469 34 INC (HL)
046A 7E LD A,(HL)
046B 214A80 LD HL,804A
046E BE CP (HL)
046F D2C104 JP NC,4C1
0472 03 INC BC
0473 110000 LD DE,0
0476 C33404 JP 434
0479 FE2B CP 2B
047B CAB204 JP Z,4B2

```

```

047E FE2D CP 2D
0480 CAB204 JP Z,4B2
0483 FE30 CP 30
0485 DAC104 JP C,4C1
0488 FE3A CP 3A
048A DA9F04 JP C,49F
048D FE47 CP 47
048F D2C104 JP NC,4C1
0492 FE41 CP 41
0494 DAC104 JP C,4C1
0497 210080 LD HL,8000
049A CB4E BIT 1,(HL)
049C CAC104 JP Z,4C1
049F 324C80 LD (804C),A
04A2 214B80 LD HL,804B
04A5 7E LD A,(HL)
04A6 214280 LD HL,8042
04A9 09 ADD HL,BC
04AA 34 INC (HL)
04AB BE CP (HL)
04AC DAC104 JP C,4C1
04AF 3A4C80 LD A,(804C)
04B2 211880 LD HL,8018
04B5 09 ADD HL,BC
04B6 09 ADD HL,BC
04B7 09 ADD HL,BC
04B8 09 ADD HL,BC
04B9 09 ADD HL,BC
04BA 09 ADD HL,BC
04BB 19 ADD HL,DE
04BC 77 LD (HL),A
04BD 13 INC DE
04BE C33404 JP 434
04C1 3E01 LD A,1
04C3 320280 LD (8002),A
04C6 C9 RET
04C7 060E LD B,E
04C9 214D80 LD HL,804D
04CC AF XOR A
04CD 77 LD (HL),A
04CE 23 INC HL
04CF 10FC DJNZ 4CD
04D1 3A0080 LD A,(8000)
04D4 CB4F BIT 1,A
04D6 C26205 JP NZ,562
04D9 010000 LD BC,0
04DC 3A4980 LD A,(8049)
04DF FE01 CP 1
04E1 D8 RET C
04E2 3D DEC A
04E3 324980 LD (8049),A
04E6 111880 LD DE,8018
04E9 60 LD H,B
04EA 69 LD L,C
04EB 29 ADD HL,HL
04EC 09 ADD HL,BC
04ED 29 ADD HL,HL
04EE 19 ADD HL,DE
04EF EB EX DE,HL
04F0 1A LD A,(DE)
04F1 210080 LD HL,8000
04F4 FE2D CP 2D
04F6 CA0405 JP Z,504
04F9 FE2B CP 2B
04FB CB9E RES 3,(HL)
04FD C20705 JP NZ,507
0500 13 INC DE
0501 C30705 JP 507
0504 13 INC DE
0505 CBDE SET 3,(HL)
0507 214D80 LD HL,804D
050A 09 ADD HL,BC
050B 09 ADD HL,BC
050C 225F80 LD (805F),HL
050F 214280 LD HL,8042
0512 09 ADD HL,BC
0513 7E LD A,(HL)
0514 FE01 CP 1
0516 DA4805 JP C,548
0519 326180 LD (8061),A
051C 1A LD A,(DE)
051D D630 SUB 30
051F D5 PUSH DE
0520 1600 LD D,0
0522 5F LD E,A
0523 62 LD H,D
0524 6B LD L,E
0525 3A6180 LD A,(8061)
0528 3D DEC A
0529 CA3505 JP Z,535
052C 29 ADD HL,HL
052D 29 ADD HL,HL
052E 19 ADD HL,DE
052F 29 ADD HL,HL
0530 54 LD D,H
0531 5D LD E,L
0532 C32805 JP 528
0535 2A5F80 LD HL,(805F)
0538 7E LD A,(HL)
0539 83 ADD A,E
053A 77 LD (HL),A
053B 23 INC HL
053C 7E LD A,(HL)
053D 8A ADC A,D
053E 77 LD (HL),A
053F D1 POP DE
0540 3A6180 LD A,(8061)
0543 3D DEC A
0544 13 INC DE
0545 C31405 JP 514
0548 3A0080 LD A,(8000)
054B CB5F BIT 3,A
054D CA5E05 JP Z,55E
0550 2A5F80 LD HL,(805F)
0553 7E LD A,(HL)
0554 2F CPL
0555 5F LD E,A
0556 23 INC HL
0557 7E LD A,(HL)
0558 2F CPL
0559 57 LD D,A
055A 13 INC DE
055B 72 LD (HL),D
055C 2B DEC HL
055D 73 LD (HL),E
055E 03 INC BC
055F C3DC04 JP 4DC
0562 215B80 LD HL,805B
0565 111880 LD DE,8018
0568 3A4280 LD A,(8042)
056B 47 LD B,A
056C 3E04 LD A,4
056E FE01 CP 1
0570 DA9005 JP C,590
0573 B8 CP B
0574 CA7C05 JP Z,57C
0577 3600 LD (HL),0
0579 C38B05 JP 58B
057C 4F LD C,A
057D 1A LD A,(DE)

```

057E D630 SUB 30
 0580 FE10 CP 10
 0582 DA8705 JP C,587
 0585 D607 SUB 7
 0587 77 LD (HL),A
 0588 05 DEC B
 0589 13 INC DE
 058A 79 LD A,C
 058B 3D DEC A
 058C 23 INC HL
 058D C36E05 JP 56E
 0590 214E80 LD HL,804E
 0593 115B80 LD DE,805B
 0596 1A LD A,(DE)
 0597 ED6F RLD
 0599 13 INC DE
 059A 1A LD A,(DE)
 059B ED6F RLD
 059D 13 INC DE
 059E 2B DEC HL
 059F 1A LD A,(DE)
 05A0 ED6F RLD
 05A2 13 INC DE
 05A3 1A LD A,(DE)
 05A4 ED6F RLD
 05A6 C9 RET
 05A7 AF XOR A
 05A8 320280 LD (8002),A
 05AB 3A0180 LD A,(8001)
 05AE 1F RRA
 05AF D2E905 JP NC,5E9
 05B2 FD21980F LD IY,F98
 05B6 DD217480 LD IX,8074
 05BA CD8B0B CALL B8B
 05BD FD219C0F LD IY,F9C
 05C1 DD217680 LD IX,8076
 05C5 CD8B0B CALL B8B
 05C8 FD21A00F LD IY,FA0
 05CC DD217880 LD IX,8078
 05D0 CD8B0B CALL B8B
 05D3 FD21A80F LD IY,FA8
 05D7 DD217A80 LD IX,807A
 05DB CDCC0B CALL BCC
 05DE FD21A40F LD IY,FA4
 05E2 DD217E80 LD IX,807E
 05E6 CD8B0B CALL B8B
 05E9 0606 LD B,6
 05EB 217480 LD HL,8074
 05EE 118680 LD DE,8086
 05F1 7E LD A,(HL)
 05F2 23 INC HL
 05F3 B6 OR (HL)
 05F4 CA0E06 JP Z,60E
 05F7 1A LD A,(DE)
 05F8 E5 PUSH HL
 05F9 21800F LD HL,F80
 05FC 85 ADD A,L
 05FD 6F LD L,A
 05FE D20206 JP NC,602
 0601 24 INC H
 0602 78 LD A,B
 0603 2F CPL
 0604 C607 ADD A,7
 0606 87 ADD A,A
 0607 87 ADD A,A
 0608 87 ADD A,A
 0609 87 ADD A,A
 060A 86 ADD A,(HL)
 060B E1 POP HL
 060C D3E0 OUT (E0),A
 060E 23 INC HL
 060F 13 INC DE
 0610 10DF DJNZ 5F1
 0612 21502E LD HL,2E50
 0615 2B DEC HL
 0616 7C LD A,H
 0617 B5 ORL
 0618 C21506 JP NZ,615
 061B AF XOR A
 061C 321280 LD (8012),A
 061F 320A80 LD (800A),A
 0622 3A1280 LD A,(8012)
 0625 6F LD L,A
 0626 DBC0 IN A,(C0)
 0628 E680 AND 80
 062A 07 RLCA
 062B 321280 LD (8012),A
 062E BD CPL
 062F C24506 JP NZ,645
 0632 210A80 LD HL,800A
 0635 34 INC (HL)
 0636 7E LD A,(HL)
 0637 FE04 CP 4
 0639 DA4A06 JP C,64A
 063C 3A1280 LD A,(8012)
 063F 320680 LD (8006),A
 0642 C35606 JP 656
 0645 210A80 LD HL,800A
 0648 3600 LD (HL),0
 064A 215804 LD HL,458
 064D 2B DEC HL
 064E 7C LD A,H
 064F B5 ORL
 0650 C24D06 JP NZ,64D
 0653 C32206 JP 622
 0656 AF XOR A
 0657 320A80 LD (800A),A
 065A 210000 LD HL,0
 065D 226A80 LD (806A),HL
 0660 220B80 LD (800B),HL
 0663 2A6E80 LD HL,(806E)
 0666 227280 LD (8072),HL
 0669 118080 LD DE,8080
 066C 217480 LD HL,8074
 066F 0606 LD B,6
 0671 C5 PUSH BC
 0672 4E LD C,(HL)
 0673 23 INC HL
 0674 46 LD B,(HL)
 0675 78 LD A,B
 0676 07 RLCA
 0677 3E01 LD A,1
 0679 D28506 JP NC,685
 067C 78 LD A,B
 067D 2F CPL
 067E 47 LD B,A
 067F 79 LD A,C
 0680 2F CPL
 0681 4F LD C,A
 0682 03 INC BC
 0683 3EFF LD A,FF
 0685 12 LD (DE),A
 0686 70 LD (HL),B
 0687 2B DEC HL
 0688 71 LD (HL),C
 0689 E5 PUSH HL
 068A 2A6A80 LD HL,(806A)
 068D AF XOR A
 068E ED42 SBC HL,BC
 0690 D29706 JP NC,697

```

0693 ED436A80 LD (806A),BC
0697 E1 POP HL
0698 23 INC HL
0699 23 INC HL
069A 13 INC DE
069B C1 POP BC
069C 10D3 DJNZ 671
069E 2A6A80 LD HL,(806A)
06A1 29 ADD HL,HL
06A2 226C80 LD (806C),HL
06A5 EB EX DE,HL
06A6 218C80 LD HL,808C
06A9 0606 LD B,6
06AB 73 LD (HL),E
06AC 23 INC HL
06AD 72 LD (HL),D
06AE 23 INC HL
06AF 10FA DJNZ 6AB
06B1 EB EX DE,HL
06B2 29 ADD HL,HL
06B3 226A80 LD (806A),HL
06B6 00 NOP
06B7 220D80 LD (800D),HL
06BA 7C LD A,H
06BB B5 OR L
06BC CA7509 JP Z,975
06BF D9 EXX
06C0 3A0080 LD A,(8000)
06C3 CB7F BIT 7,A
06C5 C2D406 JP NZ,6D4
06C8 2E08 LD L,8
06CA 2D DEC L
06CB 00 NOP
06CC 00 NOP
06CD 00 NOP
06CE C2CA06 JP NZ,6CA
06D1 C3DA06 JP 6DA
06D4 CD9503 CALL 395
06D7 C3DA06 JP 6DA
06DA ED5BA60F LD DE,(FA6)
06DE 2AA280 LD HL,(80A2)
06E1 AF XOR A
06E2 ED52 SBC HL,DE
06E4 2E05 LD L,5
06E6 F2FC06 JP P,6FC
06E9 2E02 LD L,2
06EB 3A8580 LD A,(8085)
06EE FE01 CP 1
06F0 CAFC06 JP Z,6FC
06F3 210000 LD HL,0
06F6 227E80 LD (807E),HL
06F9 C30107 JP 701
06FC 2D DEC L
06FD 00 NOP
06FE C2FC06 JP NZ,6FC
0701 3A1280 LD A,(8012)
0704 6F LD L,A
0705 DBC0 IN A,(C0)
0707 E680 AND 80
0709 07 RLCA
070A 321280 LD (8012),A
070D BD CPL
070E C22707 JP NZ,727
0711 210A80 LD HL,800A
0714 34 INC (HL)
0715 7E LD A,(HL)
0716 FE07 CP 7
0718 DA3007 JP C,730
071B 3600 LD (HL),0
071D 3A1280 LD A,(8012)
0720 1F RRA
0721 D23D07 JP NC,73D
0724 C37407 JP 774
0727 210A80 LD HL,800A
072A 3600 LD (HL),0
072C E5 PUSH HL
072D 00 NOP
072E 00 NOP
072F E1 POP HL
0730 210000 LD HL,0
0733 3A0680 LD A,(8006)
0736 1F RRA
0737 D23D07 JP NC,73D
073A C37407 JP 774
073D ED5BA40F LD DE,(FA4)
0741 2AA280 LD HL,(80A2)
0744 AF XOR A
0745 ED52 SBC HL,DE
0747 F25407 JP P,754
074A 2E05 LD L,5
074C 2D DEC L
074D C24C07 JP NZ,74C
0750 00 NOP
0751 CA9C07 JP Z,79C
0754 3A0080 LD A,(8000)
0757 CB6F BIT 5,A
0759 CA6507 JP Z,765
075C 2E03 LD L,3
075E 2D DEC L
075F C25E07 JP NZ,75E
0762 C39C07 JP 79C
0765 210000 LD HL,0
0768 227E80 LD (807E),HL
076B 3E01 LD A,1
076D 320280 LD (8002),A
0770 00 NOP
0771 C39C07 JP 79C
0774 3A0080 LD A,(8000)
0777 CB6F BIT 5,A
0779 C0 RET NZ
077A 3A8580 LD A,(8085)
077D FE01 CP 1
077F CA8707 JP Z,787
0782 2E05 LD L,5
0784 C39307 JP 793
0787 210000 LD HL,0
078A 227E80 LD (807E),HL
078D 2E03 LD L,3
078F 00 NOP
0790 C39307 JP 793
0793 2D DEC L
0794 C29307 JP NZ,793
0797 00 NOP
0798 00 NOP
0799 C39C07 JP 79C
079C D9 EXX
079D 2A8C80 LD HL,(808C)
07A0 ED5B7480 LD DE,(8074)
07A4 AF XOR A
07A5 ED52 SBC HL,DE
07A7 D2E207 JP NC,7E2
07AA ED5B6A80 LD DE,(806A)
07AE 19 ADD HL,DE
07AF 3A8080 LD A,(8080)
07B2 47 LD B,A
07B3 3A8680 LD A,(8086)
07B6 80 ADD A,B
07B7 E603 AND 3
07B9 328680 LD (8086),A
07BC EB EX DE,HL

```

07BD 21800F LD HL,F80
 07C0 85 ADD A,L
 07C1 6F LD L,A
 07C2 D2C607 JP NC,7C6
 07C5 24 INC H
 07C6 7E LD A,(HL)
 07C7 C600 ADD A,0
 07C9 D3E0 OUT (E0),A
 07CB 2A9880 LD HL,(8098)
 07CE CB78 BIT 7,B
 07D0 C2D707 JP NZ,7D7
 07D3 23 INC HL
 07D4 C3DB07 JP 7DB
 07D7 2B DEC HL
 07D8 C3DB07 JP 7DB
 07DB 229880 LD (8098),HL
 07DE EB EX DE,HL
 07DF C3E707 JP 7E7
 07E2 0611 LD B,11
 07E4 10FE DJNZ 7E4
 07E6 00 NOP
 07E7 228C80 LD (808C),HL
 07EA 2A8E80 LD HL,(808E)
 07ED ED5B7680 LD DE,(8076)
 07F1 AF XOR A
 07F2 ED52 SBC HL,DE
 07F4 D22F08 JP NC,82F
 07F7 ED5B6A80 LD DE,(806A)
 07FB 19 ADD HL,DE
 07FC 3A8180 LD A,(8081)
 07FF 47 LD B,A
 0800 3A8780 LD A,(8087)
 0803 80 ADD A,B
 0804 E603 AND 3
 0806 328780 LD (8087),A
 0809 EB EX DE,HL
 080A 21800F LD HL,F80
 080D 85 ADD A,L
 080E 6F LD L,A
 080F D21308 JP NC,813
 0812 24 INC H
 0813 7E LD A,(HL)
 0814 C610 ADD A,10
 0816 D3E0 OUT (E0),A
 0818 2A9A80 LD HL,(809A)
 081B CB78 BIT 7,B
 081D C22408 JP NZ,824
 0820 23 INC HL
 0821 C32808 JP 828
 0824 2B DEC HL
 0825 C32808 JP 828
 0828 229A80 LD (809A),HL
 082B EB EX DE,HL
 082C C33408 JP 834
 082F 0611 LD B,11
 0831 10FE DJNZ 831
 0833 00 NOP
 0834 228E80 LD (808E),HL
 0837 2A9080 LD HL,(8090)
 083A ED5B7880 LD DE,(8078)
 083E AF XOR A
 083F ED52 SBC HL,DE
 0841 D27C08 JP NC,87C
 0844 ED5B6A80 LD DE,(806A)
 0848 19 ADD HL,DE
 0849 3A8280 LD A,(8082)
 084C 47 LD B,A
 084D 3A8880 LD A,(8088)
 0850 80 ADD A,B
 0851 E603 AND 3
 0853 328880 LD (8088),A
 0856 EB EX DE,HL
 0857 21800F LD HL,F80
 085A 85 ADD A,L
 085B 6F LD L,A
 085C D26008 JP NC,860
 085F 24 INC H
 0860 7E LD A,(HL)
 0861 C620 ADD A,20
 0863 D3E0 OUT (E0),A
 0865 2A9C80 LD HL,(809C)
 0868 CB78 BIT 7,B
 086A C27108 JP NZ,871
 086D 23 INC HL
 086E C37508 JP 875
 0871 2B DEC HL
 0872 C37508 JP 875
 0875 229C80 LD (809C),HL
 0878 EB EX DE,HL
 0879 C38108 JP 881
 087C 0611 LD B,11
 087E 10FE DJNZ 87E
 0880 00 NOP
 0881 229080 LD (8090),HL
 0884 2A9280 LD HL,(8092)
 0887 ED5B7A80 LD DE,(807A)
 088B AF XOR A
 088C ED52 SBC HL,DE
 088E D2C908 JP NC,8C9
 0891 ED5B6A80 LD DE,(806A)
 0895 19 ADD HL,DE
 0896 3A8380 LD A,(8083)
 0899 47 LD B,A
 089A 3A8980 LD A,(8089)
 089D 80 ADD A,B
 089E E603 AND 3
 08A0 328980 LD (8089),A
 08A3 EB EX DE,HL
 08A4 21800F LD HL,F80
 08A7 85 ADD A,L
 08A8 6F LD L,A
 08A9 D2AD08 JP NC,8AD
 08AC 24 INC H
 08AD 7E LD A,(HL)
 08AE C630 ADD A,30
 08B0 D3E0 OUT (E0),A
 08B2 2A9E80 LD HL,(809E)
 08B5 CB78 BIT 7,B
 08B7 C2BE08 JP NZ,8BE
 08BA 23 INC HL
 08BB C3C208 JP 8C2
 08BE 2B DEC HL
 08BF C3C208 JP 8C2
 08C2 229E80 LD (809E),HL
 08C5 EB EX DE,HL
 08C6 C3CE08 JP 8CE
 08C9 0611 LD B,11
 08CB 10FE DJNZ 8CB
 08CD 00 NOP
 08CE 229280 LD (8092),HL
 08D1 2A9480 LD HL,(8094)
 08D4 ED5B7C80 LD DE,(807C)
 08D8 AF XOR A
 08D9 ED52 SBC HL,DE
 08DB D21609 JP NC,916
 08DE ED5B6A80 LD DE,(806A)
 08E2 19 ADD HL,DE
 08E3 3A8480 LD A,(8084)
 08E6 47 LD B,A
 08E7 3A8A80 LD A,(808A)

```

08EA 80  ADD A,B
08EB E603 AND 3
08ED 328A80 LD (808A),A
08F0 EB  EX DE,HL
08F1 21800F LD HL,F80
08F4 85  ADD A,L
08F5 6F  LD L,A
08F6 D2FA08 JP NC,8FA
08F9 24  INC H
08FA 7E  LD A,(HL)
08FB C640 ADD A,40
08FD D3E0 OUT (E0),A
08FF 2AA080 LD HL,(80A0)
0902 CB78 BIT 7,B
0904 C20B09 JP NZ,90B
0907 23  INC HL
0908 C30F09 JP 90F
090B 2B  DEC HL
090C C30F09 JP 90F
090F 22A080 LD (80A0),HL
0912 EB  EX DE,HL
0913 C31B09 JP 91B
0916 0611 LD B,11
0918 10FE DJNZ 918
091A 00  NOP
091B 229480 LD (8094),HL
091E 2A9680 LD HL,(8096)
0921 ED5B7E80 LD DE,(807E)
0925 AF  XOR A
0926 ED52  SBC HL,DE
0928 D26309 JP NC,963
092B ED5B6A80 LD DE,(806A)
092F 19  ADD HL,DE
0930 3A8580 LD A,(8085)
0933 47  LD B,A
0934 3A8B80 LD A,(808B)
0937 80  ADD A,B
0938 E603 AND 3
093A 328B80 LD (808B),A
093D EB  EX DE,HL
093E 21800F LD HL,F80
0941 85  ADD A,L
0942 6F  LD L,A
0943 D24709 JP NC,947
0946 24  INC H
0947 7E  LD A,(HL)
0948 C650 ADD A,50
094A D3E0 OUT (E0),A
094C 2AA280 LD HL,(80A2)
094F CB78 BIT 7,B
0951 C25809 JP NZ,958
0954 23  INC HL
0955 C35C09 JP 95C
0958 2B  DEC HL
0959 C35C09 JP 95C
095C 22A280 LD (80A2),HL
095F EB  EX DE,HL
0960 C36809 JP 968
0963 0611 LD B,11
0965 10FE DJNZ 965
0967 00  NOP
0968 229680 LD (8096),HL
096B CD9909 CALL 999
096E 2A0D80 LD HL,(800D)
0971 2B  DEC HL
0972 C3B706 JP 6B7
0975 3A0080 LD A,(8000)
0978 CB6F BIT 5,A
097A CA8309 JP Z,983
097D 3E01 LD A,1
097F 320280 LD (8002),A
0982 C9  RET
0983 CB77 BIT 6,A
0985 C0  RET NZ
0986 214A3B LD HL,3B4A
0989 2B  DEC HL
098A 7C  LD A,H
098B B5  OR L
098C C28909 JP NZ,989
098F 0606 LD B,6
0991 AF  XOR A
0992 D3E0 OUT (E0),A
0994 C610 ADD A,10
0996 10FA DJNZ 992
0998 C9  RET
0999 ED4B0D80 LD BC,(800D)
099D 2A7280 LD HL,(8072)
09A0 7C  LD A,H
09A1 B5  OR L
09A2 CAA909 JP Z,9A9
09A5 2B  DEC HL
09A6 C3A009 JP 9A0
09A9 2A7080 LD HL,(8070)
09AC 7C  LD A,H
09AD B5  OR L
09AE CAB509 JP Z,9B5
09B1 2B  DEC HL
09B2 C3AC09 JP 9AC
09B5 2A6C80 LD HL,(806C)
09B8 AF  XOR A
09B9 ED42 SBC HL,BC
09BB 2A7280 LD HL,(8072)
09BE ED5B0B80 LD DE,(800B)
09C2 D2D709 JP NC,9D7
09C5 7C  LD A,H
09C6 B5  OR L
09C7 CAD309 JP Z,9D3
09CA 010800 LD BC,8
09CD AF  XOR A
09CE ED42 SBC HL,BC
09D0 C3F009 JP 9F0
09D3 13  INC DE
09D4 C3E909 JP 9E9
09D7 7A  LD A,D
09D8 B3  OR E
09D9 CAE009 JP Z,9E0
09DC 1B  DEC DE
09DD C3E909 JP 9E9
09E0 010800 LD BC,8
09E3 09  ADD HL,BC
09E4 00  NOP
09E5 00  NOP
09E6 C3F009 JP 9F0
09E9 00  NOP
09EA 00  NOP
09EB 00  NOP
09EC 00  NOP
09ED C3F009 JP 9F0
09F0 227280 LD (8072),HL
09F3 ED530B80 LD (800B),DE
09F7 C9  RET
09F8 060A LD B,A
09FA 217480 LD HL,8074
09FD AF  XOR A
09FE 77  LD (HL),A
09FF 23  INC HL
0A00 10FC DJNZ 9FE
0A02 016009 LD BC,960
0A05 71  LD (HL),C
0A06 23  INC HL

```

```

0A07 70 LD (HL),B
0A08 3E20 LD A,20
0A0A 320080 LD (8000),A
0A0D CDA705 CALL 5A7
0A10 AF XOR A
0A11 320080 LD (8000),A
0A14 21A280 LD HL,80A2
0A17 77 LD (HL),A
0A18 23 INC HL
0A19 77 LD (HL),A
0A1A C9 RET
0A1B 0606 LD B,6
0A1D 217480 LD HL,8074
0A20 119880 LD DE,8098
0A23 C5 PUSH BC
0A24 1A LD A,(DE)
0A25 2F CPL
0A26 4F LD C,A
0A27 13 INC DE
0A28 1A LD A,(DE)
0A29 2F CPL
0A2A 47 LD B,A
0A2B 03 INC BC
0A2C 71 LD (HL),C
0A2D 23 INC HL
0A2E 70 LD (HL),B
0A2F 23 INC HL
0A30 13 INC DE
0A31 C1 POP BC
0A32 10EF DJNZ A23
0A34 CDA705 CALL 5A7
0A37 3A0280 LD A,(8002)
0A3A 1F RRA
0A3B DAD903 JP C,3D9
0A3E C9 RET
0A3F 3D DEC A
0A40 87 ADD A,A
0A41 0600 LD B,0
0A43 4F LD C,A
0A44 21840F LD HL,F84
0A47 116E80 LD DE,806E
0A4A 09 ADD HL,BC
0A4B EDA0 LDI
0A4D EDA0 LDI
0A4F 0600 LD B,0
0A51 4F LD C,A
0A52 218E0F LD HL,F8E
0A55 117080 LD DE,8070
0A58 09 ADD HL,BC
0A59 EDA0 LDI
0A5B EDA0 LDI
0A5D C9 RET
0A5E 214A80 LD HL,804A
0A61 3600 LD (HL),0
0A63 214B80 LD HL,804B
0A66 3600 LD (HL),0
0A68 3E20 LD A,20
0A6A 320080 LD (8000),A
0A6D CD1404 CALL 414
0A70 3A0280 LD A,(8002)
0A73 1F RRA
0A74 DAD903 JP C,3D9
0A77 060A LD B,A
0A79 217480 LD HL,8074
0A7C AF XOR A
0A7D 77 LD (HL),A
0A7E 23 INC HL
0A7F 10FC DJNZ A7D
0A81 EB EX DE,HL
0A82 ED4BA280 LD BC,(80A2)
0A86 2AA40F LD HL,(FA4)
0A89 AF XOR A
0A8A ED42 SBC HL,BC
0A8C EB EX DE,HL
0A8D 73 LD (HL),E
0A8E 23 INC HL
0A8F 72 LD (HL),D
0A90 CDA705 CALL 5A7
0A93 3A0280 LD A,(8002)
0A96 1F RRA
0A97 DAD903 JP C,3D9
0A9A C3C400 JP C4
0A9D 214A80 LD HL,804A
0AA0 3601 LD (HL),1
0AA2 214B80 LD HL,804B
0AA5 3603 LD (HL),3
0AA7 3E01 LD A,1
0AA9 320080 LD (8000),A
0AAC CD1404 CALL 414
0AAF 3A0280 LD A,(8002)
0AB2 1F RRA
0AB3 DAD903 JP C,3D9
0AB6 CDC704 CALL 4C7
0AB9 2A4D80 LD HL,(804D)
0ABC 44 LD B,H
0ABD 4D LD C,L
0ABE 29 ADD HL,HL
0ABF 09 ADD HL,BC
0AC0 29 ADD HL,HL
0AC1 29 ADD HL,HL
0AC2 014F83 LD BC,834F
0AC5 09 ADD HL,BC
0AC6 119880 LD DE,8098
0AC9 EB EX DE,HL
0ACA 010C00 LD BC,C
0ACD EDB0 LDIR
0ACF C3C400 JP C4
0AD2 214A80 LD HL,804A
0AD5 3600 LD (HL),0
0AD7 214B80 LD HL,804B
0ADA 3600 LD (HL),0
0ADC AF XOR A
0ADD 320080 LD (8000),A
0AE0 CD1404 CALL 414
0AE3 3A0280 LD A,(8002)
0AE6 1F RRA
0AE7 DAD903 JP C,3D9
0AEA 060A LD B,A
0AEC 217480 LD HL,8074
0AEF AF XOR A
0AF0 77 LD (HL),A
0AF1 23 INC HL
0AF2 10FC DJNZ AF0
0AF4 2AA60F LD HL,(FA6)
0AF7 ED5BA280 LD DE,(80A2)
0AFB AF XOR A
0AFC ED52 SBC HL,DE
0AFE 227E80 LD (807E),HL
0B01 CDA705 CALL 5A7
0B04 3A0280 LD A,(8002)
0B07 1F RRA
0B08 DAD903 JP C,3D9 0B0B C3C400 JP C4
0B0E 214A80 LD HL,804A
0B11 3600 LD (HL),0
0B13 214B80 LD HL,804B
0B16 3600 LD (HL),0
0B18 3E20 LD A,20
0B1A 320080 LD (8000),A
0B1D CD1404 CALL 414
0B20 3A0280 LD A,(8002)

```

```

0B23 1F RRA
0B24 DAD903 JP C,3D9
0B27 060A LD B,A
0B29 217480 LD HL,8074
0B2C AF XOR A
0B2D 77 LD (HL),A
0B2E 23 INC HL
0B2F 10FC DJNZ B2D
0B31 EB EX DE,HL
0B32 ED4BA280 LD BC,(80A2)
0B36 215E01 LD HL,15E
0B39 AF XOR A
0B3A ED42 SBC HL,BC
0B3C EB EX DE,HL
0B3D 73 LD (HL),E
0B3E 23 INC HL
0B3F 72 LD (HL),D
0B40 CDA705 CALL 5A7
0B43 3A0280 LD A,(8002)
0B46 1F RRA
0B47 DAD903 JP C,3D9
0B4A 060C LD B,C
0B4C 219880 LD HL,8098
0B4F AF XOR A
0B50 77 LD (HL),A
0B51 23 INC HL
0B52 10FC DJNZ B50
0B54 C3C400 JP C4
0B57 214A80 LD HL,804A
0B5A 3606 LD (HL),6
0B5C 214B80 LD HL,804B
0B5F 3604 LD (HL),4
0B61 3E05 LD A,5
0B63 320080 LD (8000),A
0B66 CD1404 CALL 414
0B69 3A0280 LD A,(8002)
0B6C 1F RRA
0B6D DAD903 JP C,3D9
0B70 CDC704 CALL 4C7
0B73 010C00 LD BC,C
0B76 214D80 LD HL,804D
0B79 117480 LD DE,8074
0B7C EDB0 LDIR
0B7E CDA705 CALL 5A7
0B81 3A0280 LD A,(8002)
0B84 1F RRA
0B85 DAD903 JP C,3D9
0B88 C3C400 JP C4
0B8B DD6601 LD H,(IX+1)
0B8E DD6E00 LD L,(IX+0)
0B91 DD4625 LD B,(IX+25)
0B94 DD4E24 LD C,(IX+24)
0B97 AF XOR A
0B98 ED4A ADC HL,BC
0B9A FAAB0B JP M,BAB
0B9D FD5601 LD D,(IY+1)
0BA0 FD5E00 LD E,(IY+0)
0BA3 AF XOR A
0BA4 ED52 SBC HL,DE
0BA6 F8 RET M
0BA7 C8 RET Z
0BA8 C3B90B JP BB9
0BAB FD5603 LD D,(IY+3)
0BAE FD5E02 LD E,(IY+2)
0BB1 AF XOR A
0BB2 ED52 SBC HL,DE
0BB4 F0 RET P
0BB5 C8 RET Z
0BB6 C3B90B JP BB9
0BB9 EB EX DE,HL

0BBA AF XOR A
0BBB ED42 SBC HL,BC
0BBD C3C00B JP BC0
0BC0 DD7401 LD (IX+1),H
0BC3 DD7500 LD (IX+0),L
0BC6 3E01 LD A,1
0BC8 320280 LD (8002),A
0BCB C9 RET
0BCC DD6625 LD H,(IX+25)
0BCF DD6E24 LD L,(IX+24)
0BD2 DD5601 LD D,(IX+1)
0BD5 DD5E00 LD E,(IX+0)
0BD8 19 ADD HL,DE
0BD9 226280 LD (8062),HL
0BDC DD6627 LD H,(IX+27)
0BDF DD6E26 LD L,(IX+26)
0BE2 DD5603 LD D,(IX+3)
0BE5 DD5E02 LD E,(IX+2)
0BE8 19 ADD HL,DE
0BE9 226480 LD (8064),HL
0BEC 2A6280 LD HL,(8062)
0BEF FD5601 LD D,(IY+1)
0BF2 FD5E00 LD E,(IY+0)
0BF5 AF XOR A
0BF6 ED5A ADC HL,DE
0BF8 FA230C JP M,C23
0BFB 2A6280 LD HL,(8062)
0BFE EB EX DE,HL
0BFF AF XOR A
0C00 ED52 SBC HL,DE
0C02 FA230C JP M,C23
0BCF DD6E24 LD L,(IX+24)
0BD2 DD5601 LD D,(IX+1)
0BD5 DD5E00 LD E,(IX+0)
0BD8 19 ADD HL,DE
0BD9 226280 LD (8062),HL
0BDC DD6627 LD H,(IX+27)
0BDF DD6E26 LD L,(IX+26)
0BE2 DD5603 LD D,(IX+3)
0BE5 DD5E02 LD E,(IX+2)
0BE8 19 ADD HL,DE
0BE9 226480 LD (8064),HL
0BEC 2A6280 LD HL,(8062)
0BEF FD5601 LD D,(IY+1)
0BF2 FD5E00 LD E,(IY+0)
0BF5 AF XOR A
0BF6 ED5A ADC HL,DE
0BF8 FA230C JP M,C23
0BFB 2A6280 LD HL,(8062)
0BFE EB EX DE,HL
0BFF AF XOR A
0C00 ED52 SBC HL,DE
0C02 FA230C JP M,C23
0C05 2A6280 LD HL,(8062)
0C08 226880 LD (8068),HL
0C0B 2A6480 LD HL,(8064)
0C0E 226680 LD (8066),HL
0C11 CD900C CALL C90
0C14 2A6680 LD HL,(8066)
0C17 226480 LD (8064),HL
0C1A 2A6880 LD HL,(8068)
0C1D 226280 LD (8062),HL
0C20 C36B0C JP C6B
0C23 2A6480 LD HL,(8064)
0C26 FD5601 LD D,(IY+1)
0C29 FD5E00 LD E,(IY+0)
0C2C AF XOR A
0C2D ED5A ADC HL,DE
0C2F FA5A0C JP M,C5A
0C32 2A6480 LD HL,(8064)

```

```

0C35 EB   EX DE,HL
0C36 AF   XOR A
0C37 ED52 SBC HL,DE
0C39 FA5A0C JP M,C5A
0C3C 2A6280 LD HL,(8062)
0C3F 226680 LD (8066),HL
0C42 2A6480 LD HL,(8064)
0C45 226880 LD (8068),HL
0C48 CD900C CALL C90
0C4B 2A6680 LD HL,(8066)
0C4E 226280 LD (8062),HL
0C51 2A6880 LD HL,(8068)
0C54 226480 LD (8064),HL
0C57 C36B0C JP C6B
0C5A 3E01 LD A,1
0C5C 320280 LD (8002),A
0C5F 210000 LD HL,0
0C62 226280 LD (8062),HL
0C65 226480 LD (8064),HL
0C68 C36B0C JP C6B
0C6B 2A6280 LD HL,(8062)
0C6E DD5625 LD D,(IX+25)
0C71 DD5E24 LD E,(IX+24)
0C74 AF   XOR A
0C75 ED52 SBC HL,DE
0C77 DD7401 LD (IX+1),H
0C7A DD7500 LD (IX+0),L
0C7D 2A6480 LD HL,(8064)
0C80 DD5627 LD D,(IX+27)
0C83 DD5E26 LD E,(IX+26)
0C86 AF   XOR A
0C87 ED52 SBC HL,DE
0C89 DD7403 LD (IX+3),H
0C8C DD7502 LD (IX+2),L
0C8F C9   RET
0C90 2A6680 LD HL,(8066)
0C93 ED5B6880 LD DE,(8068)
0C97 FD4603 LD B,(IY+3)
0C9A FD4E02 LD C,(IY+2)
0C9D AF   XOR A
0C9E ED52 SBC HL,DE
0CA0 FABB0C JP M,CBB
0CA3 AF   XOR A
0CA4 ED42 SBC HL,BC
0CA6 FAD30C JP M,CD3
0CA9 CAD30C JP Z,CD3
0CAC 3E01 LD A,1
0CAE 320280 LD (8002),A
0CB1 EB   EX DE,HL
0CB2 AF   XOR A
0CB3 ED4A ADC HL,BC
0CB5 226680 LD (8066),HL
0CB8 C3D30C JP CD3
0CBB AF   XOR A
0CBC ED4A ADC HL,BC
0CBE F2D30C JP P,CD3
0CC1 CAD30C JP Z,CD3
0CC4 3E01 LD A,1
0CC6 320280 LD (8002),A
0CC9 EB   EX DE,HL
0CCA AF   XOR A
0CCB ED42 SBC HL,BC
0CCD 226680 LD (8066),HL
0CD0 C3D30C JP CD3
0CD3 2A6680 LD HL,(8066)
0CD6 ED5B6880 LD DE,(8068)
0CDA FD4605 LD B,(IY+5)
0CDD FD4E04 LD C,(IY+4)
0CE0 AF   XOR A
0CE1 ED5A ADC HL,DE
0CE3 FA050D JP M,D05
0CE6 AF   XOR A
0CE7 ED42 SBC HL,BC
0CE9 FA240D JP M,D24
0CEC CA240D JP Z,D24
0CEF 3E01 LD A,1
0CF1 320280 LD (8002),A
0CF4 EB   EX DE,HL
0CF5 AF   XOR A
0CF6 ED42 SBC HL,BC
0CF8 110000 LD DE,0
0CFB EB   EX DE,HL
0CFC AF   XOR A
0CFD ED52 SBC HL,DE
0CFE 226680 LD (8066),HL
0D02 C3240D JP D24
0D05 AF   XOR A
0D06 ED4A ADC HL,BC
0D08 F2240D JP P,D24
0D0B CA240D JP Z,D24
0D0E 3E01 LD A,1
0D10 320280 LD (8002),A
0D13 EB   EX DE,HL
0D14 AF   XOR A
0D15 ED4A ADC HL,BC
0D17 110000 LD DE,0
0D1A EB   EX DE,HL
0D1B AF   XOR A
0D1C ED52 SBC HL,DE
0D1E 226680 LD (8066),HL
0D21 C3240D JP D24
0D24 C9   RET
0D25 214A80 LD HL,804A
0D28 3601 LD (HL),1
0D2A 214B80 LD HL,804B
0D2D 3601 LD (HL),1
0D2F 3E01 LD A,1
0D31 320080 LD (8000),A
0D34 CD1404 CALL 414
0D37 3A0280 LD A,(8002)
0D3A 1F   RRA
0D3B DAD903 JP C,3D9
0D3E CDC704 CALL 4C7
0D41 2A4D80 LD HL,(804D)
0D44 7C   LD A,H
0D45 85   ADD A,L
0D46 FE01 CP 1
0D48 CA530D JP Z,D53
0D4B FE00 CP 0
0D4D CA530D JP Z,D53
0D50 C3D903 JP 3D9
0D53 320180 LD (8001),A
0D56 C3C400 JP C4
0D59 214A80 LD HL,804A
0D5C 3601 LD (HL),1
0D5E 214B80 LD HL,804B
0D61 3603 LD (HL),3
0D63 3E01 LD A,1
0D65 320080 LD (8000),A
0D68 CD1404 CALL 414
0D6B 3A0280 LD A,(8002)
0D6E 1F   RRA
0D6F DAD903 JP C,3D9
0D72 CDC704 CALL 4C7
0D75 2A4D80 LD HL,(804D)
0D78 54   LD D,H
0D79 5D   LD E,L
0D7A 29   ADD HL,HL
0D7B 19   ADD HL,DE
0D7C 29   ADD HL,HL

```

```

0D7D 29  ADD HL,HL
0D7E EB  EX DE,HL
0D7F DD214F83 LD IX,834F
0D83 DD19  ADD IX,DE
0D85 FD219880 LD IY,8098
0D89 217480 LD HL,8074
0D8C 0606  LD B,6
0D8E E5  PUSH HL
0D8F DD6E00 LD L,(IX+0)
0D92 DD6601 LD H,(IX+1)
0D95 FD5E00 LD E,(IY+0)
0D98 FD5601 LD D,(IY+1)
0D9B AF  XOR A
0D9C ED52  SBC HL,DE
0D9E EB  EX DE,HL
0D9F E1  POP HL
0DA0 73  LD (HL),E
0DA1 23  INC HL
0DA2 72  LD (HL),D
0DA3 DD23  INC IX
0DA5 DD23  INC IX
0DA7 FD23  INC IY
0DA9 FD23  INC IY
0DAB 23  INC HL
0DAC 10E0  DJNZ D8E
0DAE CDA705 CALL 5A7
0DB1 3A0280 LD A,(8002)
0DB4 1F  RRA
0DB5 DAD903 JP C,3D9
0DB8 C3C400 JP C4
0DBB 214A80 LD HL,804A
0DBE 3600 LD (HL),0
0DC0 214B80 LD HL,804B
0DC3 3600 LD (HL),0
0DC5 AF  XOR A
0DC6 320080 LD (8000),A
0DC9 CD1404 CALL 414
0DCC 3A0280 LD A,(8002)
0DCF 1F  RRA
0DD0 DAD903 JP C,3D9
0DD3 CD1B0A CALL A1B
0DD6 C3C400 JP C4
0DD9 214A80 LD HL,804A
0DDC 3601 LD (HL),1
0DDE 214B80 LD HL,804B
0DE1 3604 LD (HL),4
0DE3 3E03 LD A,3
0DE5 320080 LD (8000),A
0DE8 CD1404 CALL 414
0DEB 3A0280 LD A,(8002)
0DEE 1F  RRA
0DEF DAD903 JP C,3D9
0DF2 CDC704 CALL 4C7
0DF5 2A4D80 LD HL,(804D)
0DF8 7E  LD A,(HL)
0DF9 D360 OUT (60),A
0DFB C3C400 JP C4
0DFE 214A80 LD HL,804A
0E01 3607 LD (HL),7
0E03 214B80 LD HL,804B
0E06 3604 LD (HL),4
0E08 3E05 LD A,5
0E0A 320080 LD (8000),A
0E0D CD1404 CALL 414
0E10 3A0280 LD A,(8002)
0E13 1F  RRA
0E14 DAD903 JP C,3D9
0E17 3A4280 LD A,(8042)
0E1A FE04 CP 4
0E1C D2D903 JP NC,3D9
0E1F CDC704 CALL 4C7
0E22 2A4D80 LD HL,(804D)
0E25 44  LD B,H
0E26 4D  LD C,L
0E27 29  ADD HL,HL
0E28 09  ADD HL,BC
0E29 29  ADD HL,HL
0E2A 29  ADD HL,HL
0E2B 014F83 LD BC,834F
0E2E 09  ADD HL,BC
0E2F 114F80 LD DE,804F
0E32 EB  EX DE,HL
0E33 010C00 LD BC,C
0E36 EDB0  LDIR
0E38 C3C400 JP C4
0E3B 214A80 LD HL,804A
0E3E 3601 LD (HL),1
0E40 214B80 LD HL,804B
0E43 3602 LD (HL),2
0E45 3E01 LD A,1
0E47 320080 LD (8000),A
0E4A CD1404 CALL 414
0E4D 3A0280 LD A,(8002)
0E50 1F  RRA
0E51 DAD903 JP C,3D9
0E54 CDC704 CALL 4C7
0E57 2A4D80 LD HL,(804D)
0E5A 7D  LD A,L
0E5B E60F AND F
0E5D C660 ADD A,60
0E5F D3E0 OUT (E0),A
0E61 C3C400 JP C4
0E64 214A80 LD HL,804A
0E67 3601 LD (HL),1
0E69 214B80 LD HL,804B
0E6C 3601 LD (HL),1
0E6E 3E01 LD A,1
0E70 320080 LD (8000),A
0E73 CD1404 CALL 414
0E76 3A0280 LD A,(8002)
0E79 1F  RRA
0E7A DAD903 JP C,3D9
0E7D CDC704 CALL 4C7
0E80 2A4D80 LD HL,(804D)
0E83 7C  LD A,H
0E84 B5  ORL
0E85 CAD903 JP Z,3D9
0E88 FE06 CP 6
0E8A D2D903 JP NC,3D9
0E8D CD3F0A CALL A3F
0E90 C3C400 JP C4
0E93 214A80 LD HL,804A
0E96 3601 LD (HL),1
0E98 214B80 LD HL,804B
0E9B 3602 LD (HL),2
0E9D 3E01 LD A,1
0E9F 320080 LD (8000),A
0EA2 CD1404 CALL 414
0EA5 3A0280 LD A,(8002)
0EA8 1F  RRA
0EA9 DAD903 JP C,3D9
0EAC CDC704 CALL 4C7
0EAF 2A4D80 LD HL,(804D)
0EB2 7C  LD A,H
0EB3 B5  ORL
0EB4 CAD00E JP Z,ED0
0EB7 0608 LD B,8
0EB9 CB25 SLA L
0EBB CB14 RL H
0EBD 10FA DJNZ EB9

```

0EBF 018B02 LD BC,28B
 0EC2 54 LD D,H
 0EC3 5D LD E,L
 0EC4 1B DEC DE
 0EC5 7A LD A,D
 0EC6 B3 OR E
 0EC7 C2C40E JP NZ,EC4
 0ECA 0B DEC BC
 0ECB 78 LD A,B
 0ECC B1 OR C
 0ECD C2C20E JP NZ,EC2
 0ED0 C3C400 JP C4
 0ED3 3E01 LD A,1
 0ED5 320080 LD (8000),A
 0ED8 C3E30E JP EE3
 0EDB 3E11 LD A,11
 0EDD 320080 LD (8000),A
 0EE0 C3E30E JP EE3
 0EE3 214A80 LD HL,80A4
 0EE6 3604 LD (HL),4
 0EE8 214B80 LD HL,804B
 0EEB 3601 LD (HL),1
 0EED CD1404 CALL 414
 0EF0 3A0280 LD A,(8002)
 0EF3 1F RRA
 0EF4 DAD903 JP C,3D9
 0EF7 CDC704 CALL 4C7
 0EFA 0604 LD B,4
 0EFC 0E00 LD C,0
 0EFE 214D80 LD HL,804D
 0F01 7E LD A,(HL)
 0F02 FE00 CP 0
 0F04 C20C0F JP NZ,F0C
 0F07 CBB9 RES 7,C
 0F09 C3130F JP F13
 0F0C FE01 CP 1
 0F0E C2D903 JP NZ,3D9
 0F11 CBF9 SET 7,C
 0F13 CB01 RLC C
 0F15 23 INC HL
 0F16 23 INC HL
 0F17 10E8 DJNZ F01
 0F19 79 LD A,C
 0F1A 321480 LD (8014),A
 0F1D AF XOR A
 0F1E 321380 LD (8013),A
 0F21 0E00 LD C,0
 0F23 3A1380 LD A,(8013)
 0F26 47 LD B,A
 0F27 215804 LD HL,458
 0F2A 2B DEC HL
 0F2B 7C LD A,H
 0F2C B5 OR L
 0F2D C22A0F JP NZ,F2A
 0F30 DBC0 IN A,(C0)
 0F32 E63C AND 3C
 0F34 321380 LD (8013),A
 0F37 B8 CP B
 0F38 C2210F JP NZ,F21
 0F3B 0C INC C
 0F3C 79 LD A,C
 0F3D FE04 CP 4
 0F3F DA230F JP C,F23
 0F42 3A1380 LD A,(8013)
 0F45 210080 LD HL,8000
 0F48 CB66 BIT 4,(HL)
 0F4A C24E0F JP NZ,F4E
 0F4D 2F CPL
 0F4E E63C AND 3C
 0F50 0F RRCA

0F51 0F RRCA
 0F52 47 LD B,A
 0F53 3A1480 LD A,(8014)
 0F56 A0 AND B
 0F57 C660 ADD A,60
 0F59 D3E0 OUT (E0),A
 0F5B C3C400 JP C4
 0F5E AF XOR A
 0F5F 0607 LD B,7
 0F61 32A480 LD (80A4),A
 0F64 CD9503 CALL 395
 0F67 DBA0 IN A,(A0)
 0F69 E680 AND 80
 0F6B 21A480 LD HL,80A4
 0F6E BE CP (HL)
 0F6F C25E0F JP NZ,F5E
 0F72 10ED DJNZ F61
 0F74 FE00 CP 0
 0F76 C25E0F JP NZ,F5E
 0F79 C3C400 JP C4
 0F7C 00 NOP
 0F7D 00 NOP
 0F7E 00 NOP
 0F7F 00 NOP
 0F80 05 DEC B
 0F81 060A LD B,A
 0F83 09 ADD HL,BC
 0F84 00 NOP
 0F85 00 NOP
 0F5E AF XOR A
 0F5F 0607 LD B,7
 0F61 32A480 LD (80A4),A
 0F64 CD9503 CALL 395
 0F67 DBA0 IN A,(A0)
 0F69 E680 AND 80
 0F6B 21A480 LD HL,80A4
 0F6E BE CP (HL)
 0F6F C25E0F JP NZ,F5E
 0F72 10ED DJNZ F61
 0F74 FE00 CP 0
 0F76 C25E0F JP NZ,F5E
 0F79 C3C400 JP C4
 0F7C 00 NOP
 0F7D 00 NOP
 0F7E 00 NOP
 0F7F 00 NOP
 0F80 05 DEC B
 0F81 060A LD B,A
 0F83 09 ADD HL,BC
 0F84 00 NOP
 0F85 00 NOP
 0F86 B0 OR B
 0F87 00 NOP
 0F88 F8 RET M
 0F89 00 NOP
 0F8A 2801 JR Z,F8D
 0F8C 3801 JR C,F8F
 0F8E 4B LD C,E
 0F8F 01B700 LD BC,B7
 0F92 6F LD L,A
 0F93 00 NOP
 0F94 3F CCF
 0F95 00 NOP
 0F96 2F CPL
 0F97 00 NOP
 0F98 E8 RET PE
 0F99 03 INC BC
 0F9A 18FC JR F98
 0F9C 58 LD E,B
 0F9D 02 LD (BC),A

0F9E A8	XOR B	0FD5 FE0C	CP C
0F9F FD1C	????	0FD7 FE00	CP 0
0FA1 02	LD (BC),A	0FD9 00	NOP
0FA2 E4FDD0	CALL PO.D0FD	0FDA 00	NOP
0FA5 07	RLCA	0FDB 00	NOP
0FA6 30F8	JR NC,FA0	0FDC 00	NOP
0FA8 2823	JR Z,FCD	0FDD 00	NOP
0FAA 08	EX AF,AF'	0FDE 18FC	JR FDC
0FAB 07	RLCA	0FE0 E8	RET PE
0FAC 40	LD B,B	0FE1 03	INC BC
0FAD 38F4	JR C,FA3	0FE2 E8	RET PE
0FAF 01F401	LD BC,1F4	0FE3 03	INC BC
0FB2 F40100	CALL P,1	0FE4 00	NOP
0FB5 00	NOP	0FE5 00	NOP
0FB6 00	NOP	0FE6 00	NOP
0FB7 00	NOP	0FE7 00	NOP
0FB8 00	NOP	0FE8 00	NOP
0FB9 00	NOP	0FE9 00	NOP
0FBA 18FC	JR FB8	0FEA F4010C	CALL P,C01
0FBC 18FC	JR FBA	0FED FE0C	CP C
0FBE 18FC	JR FBC	0FEF FE00	CP 0
0FC0 00	NOP	0FF1 00	NOP
0FC1 00	NOP	0FF2 00	NOP
0FC2 00	NOP	0FF3 00	NOP
0FC3 00	NOP	0FF4 00	NOP
0FC4 00	NOP	0FF5 00	NOP
0FC5 00	NOP	0FF6 00	NOP
0FC6 F401F4	CALL P,F401	0FF7 00	NOP
0FC9 01F401	LD BC,1F4	0FF8 00	NOP
0FCC 00	NOP	0FF9 00	NOP
0FCD 00	NOP	0FFA 00	NOP
0FCE 00	NOP	0FFB 00	NOP
0FCF 00	NOP	0FFC 00	NOP
0FD0 00	NOP	0FFD 00	NOP
0FD1 00	NOP	0FFE 00	NOP
0FD2 F4010C	CALL P,C01	0FFF 00	NOP

ANEXO B

**Código fuente en Qbasic del programa
controlador**

```
DECLARE SUB tutorial ()
DECLARE SUB mayor ()
DECLARE SUB ciclico ()
DECLARE SUB ejecseq ()
DECLARE SUB resetr ()
DECLARE SUB restear2 ()
DECLARE SUB menger ()
DECLARE SUB resetear2 ()
DECLARE SUB resetear1 ()
DECLARE SUB abrir ()
DECLARE SUB guardar ()
DECLARE SUB gcomo ()
DECLARE SUB sombra ()
DECLARE SUB resetear ()
DECLARE SUB abajol ()
DECLARE SUB resaca ()
DECLARE SUB menarch ()
DECLARE SUB ver ()
DECLARE SUB menejec ()
DECLARE SUB moveje ()
DECLARE SUB marco ()
DECLARE SUB menayuda ()
DECLARE SUB menu ()
DECLARE SUB izquierda ()
DECLARE SUB abajo ()
DECLARE SUB movsimul ()
DECLARE SUB set ()
DECLARE SUB glosariop ()
COMMON SHARED control9
COMMON SHARED nomarch$
COMMON SHARED op
COMMON SHARED limmin1
COMMON SHARED limmax1
COMMON SHARED limmin2
COMMON SHARED limmax2
COMMON SHARED limmin3
COMMON SHARED limmax3
COMMON SHARED limmin4
COMMON SHARED limmax4
COMMON SHARED limmin5
COMMON SHARED limmax5
COMMON SHARED limmin6
COMMON SHARED limmax6
COMMON SHARED bandera
COMMON SHARED desplazar
COMMON SHARED fondo1
COMMON SHARED fondo2
COMMON SHARED fondo3
COMMON SHARED fondo4
COMMON SHARED fondo5
COMMON SHARED fondo6
COMMON SHARED fondo7
```

COMMON SHARED fondo8
COMMON SHARED fondo9
COMMON SHARED fondog1
COMMON SHARED fondog2
COMMON SHARED fondog3
COMMON SHARED fondog4
COMMON SHARED fondog5
COMMON SHARED fondog6
COMMON SHARED fondog7
COMMON SHARED fondog8
COMMON SHARED fondog9
COMMON SHARED fondo10
COMMON SHARED fondog10
COMMON SHARED fondog11
COMMON SHARED fondog12
COMMON SHARED fondog13
COMMON SHARED fondog14
COMMON SHARED fondog15
COMMON SHARED fondog16
COMMON SHARED fondog17
COMMON SHARED fondog18
COMMON SHARED fondog19
COMMON SHARED fondog20
COMMON SHARED fondos
COMMON SHARED fondos1
COMMON SHARED fondos2
COMMON SHARED fondog21
COMMON SHARED fond1
COMMON SHARED fond12
COMMON SHARED fond13
COMMON SHARED fond14
COMMON SHARED fond2
COMMON SHARED fond22
COMMON SHARED fond23
COMMON SHARED fond24
COMMON SHARED fond3
COMMON SHARED fond32
COMMON SHARED fond33
COMMON SHARED fond34
COMMON SHARED fond4
COMMON SHARED fond42
COMMON SHARED fond43
COMMON SHARED fond44
COMMON SHARED fond5
COMMON SHARED fond52
COMMON SHARED fond53
COMMON SHARED fond54
COMMON SHARED fond6
COMMON SHARED fond62
COMMON SHARED fond63
COMMON SHARED fond64
COMMON SHARED q
COMMON SHARED h
COMMON SHARED l

```
COMMON SHARED x
COMMON SHARED y
COMMON SHARED m1
COMMON SHARED m2
COMMON SHARED m3
COMMON SHARED m4
COMMON SHARED m5
COMMON SHARED m6
COMMON SHARED m14
COMMON SHARED m24
COMMON SHARED m34
COMMON SHARED m44
COMMON SHARED m54
COMMON SHARED m64
COMMON SHARED m11
COMMON SHARED m21
COMMON SHARED m31
COMMON SHARED m41
COMMON SHARED m51
COMMON SHARED m61
COMMON SHARED m12
COMMON SHARED m22
COMMON SHARED m32
COMMON SHARED m42
COMMON SHARED m52
COMMON SHARED m62
COMMON SHARED m13
COMMON SHARED m23
COMMON SHARED m33
COMMON SHARED m43
COMMON SHARED m53
COMMON SHARED m63
COMMON SHARED arri
COMMON SHARED arri1
COMMON SHARED glosariol
COMMON SHARED desplazar1
COMMON SHARED control
COMMON SHARED control10
COMMON SHARED control11
COMMON SHARED columna
COMMON SHARED er$
DIM SHARED arch$(60)
control9 = 0
control4 = 0
nomarch$ = ""
c = 1
q = 0
columna = 1
OPEN "dir" FOR INPUT AS #1
INPUT #1, q, arch$(1), arch$(2), arch$(3), arch$(4), arch$(5)
INPUT #1, arch$(6), arch$(7), arch$(8), arch$(9), arch$(10)
INPUT #1, arch$(11), arch$(12), arch$(13), arch$(14), arch$(15)
INPUT #1, arch$(16), arch$(17), arch$(18), arch$(19), arch$(20)
INPUT #1, arch$(21), arch$(22), arch$(23), arch$(24), arch$(25)
```

```

INPUT #1, arch$(26), arch$(27), arch$(28), arch$(29), arch$(30)
CLOSE
COLOR 7, 12
arri1 = 1
resetr
set
INIC:
CLS
COLOR 7, 12
fond1 = 7: fond12 = 12: fond13 = 12: fond14 = 12: fond2 = 12: fond22 = 1
fond23 = 12: fond24 = 12: fond3 = 12: fond32 = 12: fond33 = 12: fond34 =
fond4 = 12: fond42 = 12: fond43 = 12: fond44 = 12: fond5 = 12: fond52 =
fond53 = 12: fond54 = 12: fond6 = 12: fond62 = 12: fond63 = 12: fond64 =
fondos = 8
fondos1 = 1: fondos2 = 1
fondog1 = 10: fondog2 = 8: fondog3 = 8: fondog4 = 8: fondog5 = 8
fondog6 = 8: fondog7 = 8: fondog8 = 8: fondog9 = 8: fondog10 = 8
fondog11 = 8: fondog12 = 8: fondog13 = 8: fondog14 = 8: fondog15 = 8
fondog16 = 8: fondog17 = 8: fondog18 = 8: fondog19 = 8: fondog20 = 8
fondog21 = 8
fondo1 = 7: fondo2 = 7: fondo3 = 7: fondo4 = 7: fondo5 = 7
fondo6 = 7
fondo7 = 7
fondo10 = 7
desplazar1 = 1
glosariol = 0
control = 0
ver
menu
arri = 0
desplazar = 0
control10 = 0
control11 = 0
DO
w$ = INKEY$
IF w$ = CHR$(27) AND control9 = 2 THEN
    CLS
    abajo
    w$ = ""
    bandera = 1
    GOTO f
END IF
i = LEN(w$)
IF i = 1 THEN t = ASC(MID$(w$, 1, 1))
IF i = 2 THEN t = ASC(MID$(w$, 2, 1))
IF t = 77 THEN
    GOSUB der
    t = 2
    i = 0
END IF
IF t = 75 THEN
    GOSUB izq
    t = 2
    i = 0
END IF

```

```

IF t = 72 THEN
    GOSUB mensub
    t = 2
    i = 0
END IF
IF t = 80 THEN
    GOSUB mensub1
    t = 2
    i = 0
END IF
IF w$ = CHR$(32) THEN GOSUB menu1
IF w$ = "" AND control11 = 2 THEN
    SELECT CASE desplazar
    CASE 1
        SHELL "archivo"
    CASE 2
        SHELL "ejecutar"
    CASE 3
        SHELL "dos"
    CASE 4
        SHELL "salir"
    CASE 5
        SHELL "ayuda"
    END SELECT
    GOTO f
END IF
IF w$ = "s" OR w$ = "S" THEN
    salir:
    IF bandera = 1 THEN
        COLOR 7, 0
        CLS
        END
    END IF
END IF
IF w$ = "a" OR w$ = "A" THEN
    IF bandera = 1 THEN
        desplazar = 1
        ver
        izquierda
        IF arri <> 0 THEN
            arri = arri + 1
            GOSUB mensub
        END IF
    END IF
END IF
IF w$ = "e" OR w$ = "E" THEN
    IF bandera = 1 THEN
        desplazar = 2
        ver
        izquierda
        IF arri <> 0 THEN
            arri = arri + 1
            GOSUB mensub
        END IF
    END IF

```

```

        END IF
END IF
IF w$ = "d" OR w$ = "D" THEN
    IF bandera = 1 THEN
        SCREEN 0, 0
        COLOR 7, 0
        CLS
        PRINT "presione 'EXIT' PARA REGRESAR AL PROGRAMA "
        SHELL
        COLOR 7, 12
        CLS
        menu
    END IF
END IF
IF w$ = "y" OR w$ = "Y" THEN
    IF bandera = 1 THEN
        desplazar = 5
        ver
        izquierda
        IF arri <> 0 THEN
            arri = arri + 1
            GOSUB mensub
        END IF
    END IF
END IF
SELECT CASE w$
CASE "o", "O"
    IF desplazar = 1 THEN
        arri = 5
        abajo
    END IF
CASE "M", "m"
    IF desplazar = 2 THEN
        arri = 1
        abajo
    END IF
CASE "v", "V"
    IF desplazar = 2 THEN
        arri = 2
        abajo
    END IF
CASE "j", "J"
    IF desplazar = 2 THEN
        arri = 3
        abajo
    END IF
CASE "i", "I"
    IF desplazar = 2 THEN
        arri = 4
        abajo
    END IF
CASE "l", "L"
    IF desplazar = 5 THEN
        arri = 4
        abajo

```

```

        END IF
CASE "r", "R"
    IF desplazar = 5 THEN
        arri = 2
        abajo
    END IF
CASE "t", "T"
    IF desplazar = 5 THEN
        arri = 1
        abajo
    END IF
CASE "b", "B"
    IF desplazar = 1 THEN
        arri = 1
        abajo
    END IF
CASE "g", "G"
    IF desplazar = 1 THEN
        arri = 2
        abajo
    END IF
CASE "n", "N"
    IF desplazar = 1 THEN
        arri = 3
        abajo
    END IF
CASE "c", "C"
    IF desplazar = 1 THEN
        arri = 4
        abajo
    END IF
CASE CHR$(13)
    IF desplazar = 1 AND arri = 5 THEN
        SHELL "modi"
        arri = 5
    END IF
    IF desplazar = 1 AND arri = 2 THEN
        IF control10 = 1 THEN SHELL "guardar"
        guardar
        arri = 2
    END IF
    IF arri = 1 AND desplazar = 1 THEN
        IF control10 = 1 THEN SHELL "abrir"
        abrir
        IF op = 1 THEN GOTO f
        OPEN "tempo" FOR OUTPUT AS #1
        WRITE #1, control9, er$, gh, columna, fond1, fond12, fond13,
        WRITE #1, arri, desplazar
        WRITE #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
        WRITE #1, m44, m54, m64
        WRITE #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
        WRITE #1, m53, m63, m14, m24
        WRITE #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
        WRITE #1, m62, m13, m23, m33
        WRITE #1, fond62, fond63, fond64, m11, m21, m31, m41, m51, m61.

```

```

WRITE #1, m12, m22, m32
WRITE #1, nomarch$
WRITE #1, limmax11, limmin11, limmax21, limmin21, limmax31,
WRITE #1, limmin31, limmax41
WRITE #1, limmin41, limmin12, limmax12, limmin22, limmax22,
WRITE #1, limmin32, limmax32
WRITE #1, limmax42, limmin42, limmax52, limmin52, limmax62,
WRITE #1, limmin62, limmax13
WRITE #1, limmin13, limmin23, limmax23, limmin33, limmax33,
WRITE #1, limmin43, limmax43
WRITE #1, limmax53, limmin53, limmax63, limmin63, limmax14,
WRITE #1, limmin14, limmax24
WRITE #1, limmin24, limmin34, limmax34, limmin44, limmax44,
WRITE #1, limmin54, limmax54
WRITE #1, limmax64, limmin64, limmax51, limmin51, limmax61,
WRITE #1, limmin61
CLOSE
SHELL "mov"
bandera = 0
OPEN "tempo" FOR INPUT AS #1
INPUT #1, control9, er$, gh, columna, fond1, fond12, fond13,
INPUT #1, arri, desplazar
INPUT #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
INPUT #1, m44, m54, m64
INPUT #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
INPUT #1, m53, m63, m14, m24
INPUT #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
INPUT #1, m62, m13, m23, m33
INPUT #1, fond62, fond63, fond64, m11, m21, m31, m41, m51,
INPUT #1, m61, m12, m22, m32
INPUT #1, nomarch$
INPUT #1, limmax11, limmin11, limmax21, limmin21, limmax31,
INPUT #1, limmin31, limmax41
INPUT #1, limmin41, limmin12, limmax12, limmin22, limmax22,
INPUT #1, limmin32, limmax32
INPUT #1, limmax42, limmin42, limmax52, limmin52, limmax62,
INPUT #1, limmin62, limmax13
INPUT #1, limmin13, limmin23, limmax23, limmin33, limmax33,
INPUT #1, limmin43, limmax43
INPUT #1, limmax53, limmin53, limmax63, limmin63, limmax14,
INPUT #1, limmin14, limmax24
INPUT #1, limmin24, limmin34, limmax34, limmin44, limmax44,
INPUT #1, limmin54, limmax54
INPUT #1, limmax64, limmin64, limmax51, limmin51, limmax61,
INPUT #1, limmin61
CLOSE
GOTO f
END IF
IF desplazar = 1 AND arri = 3 THEN
  m11 = 0: m21 = 0: m31 = 0: m41 = 0: m51 = 0: m61 = 0
  m12 = 0: m22 = 0: m32 = 0: m42 = 0: m52 = 0: m62 = 0
  m13 = 0: m23 = 0: m33 = 0: m43 = 0: m53 = 0: m63 = 0
  m14 = 0: m24 = 0: m34 = 0: m44 = 0: m54 = 0: m64 = 0
  limmin11 = -1000: limmax11 = 1000: limmin21 = -600: limmax21 = 600
  limmin31 = -500: limmax31 = 500: limmin61 = -1800: limmax61 = 0

```

```

limmin41 = -900: limmax41 = 900: limmin51 = -1800: limmax51 = 18
limmin12 = -1000: limmax12 = 1000: limmin22 = -600: limmax22 = 6
limmin32 = -500: limmax32 = 500: limmin62 = -1800: limmax62 = 0
limmin42 = -900: limmax42 = 900: limmin52 = -1800: limmax52 = 18
limmin13 = -1000: limmax13 = 1000: limmin23 = -600: limmax23 = 6
limmin33 = -500: limmax33 = 500: limmin63 = -1800: limmax63 = 0
limmin43 = -900: limmax43 = 900: limmin53 = -1800: limmax53 = 18
limmin14 = -1000: limmax14 = 1000: limmin24 = -600: limmax24 = 6
limmin34 = -500: limmax34 = 500: limmin64 = -1800: limmax64 = 0
limmin44 = -900: limmax44 = 900: limmin54 = -1800: limmax54 = 18
control9 = 1
OPEN "tempo" FOR OUTPUT AS #1
WRITE #1, control9, er$, gh, columna, fond1, fond12, fond13, arr
WRITE #1, desplazar
WRITE #1, fond14, fond2, fond22, fond23, fond24, fond3, m34, m44
WRITE #1, m54, m64
WRITE #1, fond32, fond33, fond34, fond4, fond42, fond43, m43, m5
WRITE #1, m63, m14, m24
WRITE #1, fond44, fond5, fond52, fond53, fond54, fond6, m52, m62
WRITE #1, m13, m23, m33
WRITE #1, fond62, fond63, fond64, m11, m21, m31, m41, m51, m61,
WRITE #1, m12, m22, m32
WRITE #1, nomarch$
WRITE #1, limmax11, limmin11, limmax21, limmin21, limmax31,
WRITE #1, limmin31, limmax41
WRITE #1, limmin41, limmin12, limmax12, limmin22, limmax22,
WRITE #1, limmin32, limmax32
WRITE #1, limmax42, limmin42, limmax52, limmin52, limmax62,
WRITE #1, limmin62, limmax13
WRITE #1, limmin13, limmin23, limmax23, limmin33, limmax33,
WRITE #1, limmin43, limmax43
WRITE #1, limmax53, limmin53, limmax63, limmin63, limmax14,
WRITE #1, limmin14, limmax24
WRITE #1, limmin24, limmin34, limmax34, limmin44, limmax44,
WRITE #1, limmin54, limmax54
WRITE #1, limmax64, limmin64, limmax51, limmin51, limmax61,
WRITE #1, limmin61
CLOSE
SHELL "mov"
CLS
OPEN "tempo" FOR INPUT AS #1
INPUT #1, control9, er$, gh, columna, fond1, fond12, fond13, arr:
INPUT #1, desplazar
INPUT #1, fond14, fond2, fond22, fond23, fond24, fond3, m34, m44.
INPUT #1, m54, m64
INPUT #1, fond32, fond33, fond34, fond4, fond42, fond43, m43, m5:
INPUT #1, m63, m14, m24
INPUT #1, fond44, fond5, fond52, fond53, fond54, fond6, m52, m62.
INPUT #1, m13, m23, m33
INPUT #1, fond62, fond63, fond64, m11, m21, m31, m41, m51, m61.
INPUT #1, m12, m22, m32
INPUT #1, nomarch$
INPUT #1, limmax11, limmin11, limmax21, limmin21, limmax31,
INPUT #1, limmin31, limmax41
INPUT #1, limmin41, limmin12, limmax12, limmin22, limmax22.

```

```

INPUT #1, limmin32, limmax32
INPUT #1, limmax42, limmin42, limmax52, limmin52, limmax62,
INPUT #1, limmin62, limmax13
INPUT #1, limmin13, limmin23, limmax23, limmin33, limmax33,
INPUT #1, limmin43, limmax43
INPUT #1, limmax53, limmin53, limmax63, limmin63, limmax14,
INPUT #1, limmin14, limmax24
INPUT #1, limmin24, limmin34, limmax34, limmin44, limmax44,
INPUT #1, limmin54, limmax54
INPUT #1, limmax64, limmin64, limmax51, limmin51, limmax61,
INPUT #1, limmin61
CLOSE
CLS
END IF
IF arri = 4 AND desplazar = 1 THEN
IF control10 = 1 THEN SHELL "guardarc"
gcomo
arri = 4
END IF
IF arri = 1 AND desplazar = 2 THEN moveje
IF arri = 4 AND desplazar = 2 THEN ciclico
IF arri = 2 AND desplazar = 2 THEN movsimul
IF arri = 2 AND desplazar = 5 THEN
SHELL "slide stepper2.tif stepper3.tif stepper4.tif /m"
IF arri = 2 AND desplazar = 5 THEN
control3 = 1
menger
GOTO f
END IF
IF control3 = 1 AND arri2 = 1 THEN wpg = 0
IF control3 = 1 AND arri2 = 2 THEN wpg = 0
IF control3 = 1 AND arri2 = 3 THEN
menger
GOTO f
END IF
IF arri = 1 AND desplazar = 5 THEN
CLS
tutorial
GOTO f
END IF
IF arri = 3 AND desplazar = 5 THEN
IF control = 2 THEN GOTO x
IF control = 1 AND desplazar1 = 1 THEN
IF arri1 = 1 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12

```

```

LOCATE 8, 25: PRINT "ACCIONAMIENTO"
COLOR 7, 1
LOCATE 10, 12: PRINT "Puesta en marcha."
LOCATE 11, 12
PRINT "Poner a funcionar el sistema."
COLOR 7, 12
h = 0
END IF
IF arri1 = 2 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 24: PRINT "ARTICULACIONES"
COLOR 7, 1
LOCATE 10, 12
PRINT "Uniones entre dos o m s piezas."
COLOR 7, 12
h = 0
END IF
IF arri1 = 3 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 27: PRINT "AUTOMATICO"
COLOR 7, 1
LOCATE 10, 12
PRINT "Ejecuci'n sin intervenci'n humana"
LOCATE 11, 12
PRINT "de ciertas labores industriales"
LOCATE 12, 12
PRINT "administrativas o cientif/cas."
COLOR 7, 12
h = 0
END IF
IF arri1 = 4 THEN
z = 7
v = 1
COLOR z, v
x = 7

```

```

y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 22: PRINT "CAPACIDAD DE CARGA"
COLOR 7, 1
LOCATE 10, 12
PRINT "Es el peso, generalmente"
PRINT "en kg, que el"
LOCATE 11, 12.
PRINT "robot puede manipular,"
PRINT "ello excluye el"
LOCATE 12, 12
PRINT "peso propio del robot."
COLOR 7, 12
h = 0
END IF
IF arril = 5 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 27: PRINT "ESTATOR"
COLOR 7, 1
LOCATE 10, 12
PRINT "Parte fija de una m quina rotatori"
LOCATE 11, 12: PRINT "o motor."
COLOR 7, 12
h = 0
END IF
IF arril = 6 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 27: PRINT "EFECTOR"
COLOR 7, 1
LOCATE 10, 12

```

```

PRINT "Elemento que realiza la acci"n
LOCATE 11, 12
PRINT "sujetar, pintar, soldar o rotar,"
LOCATE 12, 12: PRINT "un robot."
COLOR 7, 12
h = 0
END IF
IF arri1 = 7 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 13
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 22: PRINT "GRADOS DE LIBERTAD"
COLOR 7, 1
LOCATE 10, 12
PRINT "Es el n#mero de "
PRINT "par metros que es"
LOCATE 11, 12
PRINT "preciso conocer para "
PRINT "determinar la"
LOCATE 12, 12
PRINT "la posici"n del robot."
LOCATE 13, 12
PRINT "Movimientos b sicos"
PRINT "independientes que"
LOCATE 14, 12
PRINT "posicionan a los elementos"
PRINT "de un robot "
LOCATE 15, 12
PRINT "en el espacio."
LOCATE 16, 12
PRINT "El n#mero m ximo de"
PRINT "grados de libertad"
LOCATE 17, 12
PRINT "para un robot es de "
PRINT "seis; tres para"
LOCATE 18, 12
PRINT "definir posici"n y tres"
PRINT "para orientar"
LOCATE 19, 12: PRINT "la mano de"
PRINT "sujeci"n."
COLOR 7, 12
h = 0
END IF
IF arri1 = 8 THEN
z = 7
v = 1
COLOR z, v

```

```

x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 24: PRINT "MANIPULADOR"
COLOR 7, 1
LOCATE 10, 12: PRINT "Sistema que"
PRINT "controla o maneja un "
LOCATE 11, 12: PRINT "elemento o part"
COLOR 7, 12
h = 0
END IF
IF arril = 9 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 42
h = 9
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 23: PRINT "MICROPROCESADOR"
COLOR 7, 1
LOCATE 10, 12
PRINT "Dispositivo que puede"
PRINT "buscar instrucciones"
LOCATE 11, 12
PRINT "de una memoria, las "
PRINT "decodifica y las"
LOCATE 12, 12
PRINT "ejecuta, realiza operaciones "
PRINT "aritméticas"
LOCATE 13, 12
PRINT "y lógicas, acepta datos "
PRINT "de dispositivos"
LOCATE 14, 12
PRINT "de entrada y envía "
PRINT "resultados a"
LOCATE 15, 12: PRINT "dispositivos"
PRINT "de salida."
COLOR 7, 12
h = 0
END IF
IF arril = 10 THEN
z = 7
v = 1
COLOR z, v
x = 7

```

```

y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 24: PRINT "MINIATURIZADO"
COLOR 7, 1
LOCATE 10, 12
PRINT "Objeto del tama#o"
PRINT "m s peque#o posible."
COLOR 7, 12
h = 0
END IF
IF arri1 = 11 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 10
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 26: PRINT "PERIFERICOS"
COLOR 7, 1
LOCATE 10, 12
PRINT "Elemento de un"
PRINT "sistema computacional"
LOCATE 11, 12
PRINT "que no es la unidad"
PRINT "central de proceso"
LOCATE 12, 12
PRINT "y que sirve "
PRINT "esencialmente para"
LOCATE 13, 12: PRINT "comunicarse"
PRINT "con el exterior."
COLOR 7, 12
h = 0
END IF
GOTO f
END IF
IF control = 1 AND desplazar1 = 2 THEN
IF arri1 = 1 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 33
l = 40
h = 8
marco

```

```

sombra
control = 2
COLOR 7, 12
LOCATE 8, 49: PRINT "PRECISION"
COLOR 7, 1
LOCATE 10, 35
PRINT "Es la tolerancia de medida"
PRINT "o de"
LOCATE 11, 35
PRINT "transmisi"n del instrumento"
PRINT "y define"
LOCATE 12, 35
PRINT "los limites de los errores"
PRINT "cometidos "
LOCATE 13, 35
PRINT "cuando el instrumento "
PRINT "opera en"
LOCATE 14, 35: PRINT "condiciones"
PRINT "normales. "
COLOR 7, 12
h = 0
END IF
IF arril = 2 THEN
z = 7
v = 1
COLOR z, v
x = 7 ;
y = 33
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 50: PRINT "PUERTO"
COLOR 7, 1
LOCATE 10, 35
PRINT "Se usa para definir un"
PRINT "espacio "
LOCATE 11, 35
PRINT "f/sico y/o l'gico, en el "
PRINT "cual la"
LOCATE 12, 35
PRINT "computadora puede colocar "
PRINT "datos y"
LOCATE 13, 35
PRINT "recibirlos hacia o desde"
PRINT "dispositivos "
LOCATE 14, 35: PRINT "externos."
COLOR 7, 12
h = 0
END IF
IF arril = 3 THEN
z = 7
v = 1

```

```
COLOR z, v
x = 7
y = 33
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 49: PRINT "RESOLUCION"
COLOR 7, 1
LOCATE 10, 35
PRINT "Es la menor variaci"n que i
PRINT "ocurrir "
LOCATE 11, 35
PRINT "en una variable de s
PRINT "como "
LOCATE 12, 35
PRINT "resultado de un camb
PRINT "en una"
LOCATE 13, 35
PRINT "variable de entrada."
COLOR 7, 12
h = 0
```

END IF

IF arril = 4 THEN

```
z = 7
v = 1
COLOR z, v
x = 7
y = 33
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 50: PRINT "ROBOTS"
COLOR 7, 1
LOCATE 10, 35: PRINT "Aparato capa
PRINT "de realizar de manera "
LOCATE 11, 35: PRINT "autom tica"
PRINT "diversas operaciones."
COLOR 7, 12
h = 0
```

END IF

IF arril = 5 THEN

```
z = 7
v = 1
COLOR z, v
x = 7
y = 33
l = 40
h = 8
marco
```

```

sombra
control = 2
COLOR 7, 12
LOCATE 8, 50: PRINT "ROTOR"
COLOR 7, 1
LOCATE 10, 35: PRINT "Parte movil"
PRINT "de un motor, generador "
LOCATE 11, 35: PRINT "elctrico,"
PRINT "turbina, etc."
COLOR 7, 12
h = 0
END IF
IF arril = 6 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 33
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 50: PRINT "RUTINA"
COLOR 7, 1
LOCATE 10, 35: PRINT "Parte de"
PRINT "un programa de inform tica"
LOCATE 11, 35: PRINT "que puede "
PRINT "ser utilizada por otros"
LOCATE 12, 35: PRINT "programas ."
COLOR 7, 12
h = 0
END IF
IF arril = 7 THEN
z = 7
v = 1
COLOR z, v
x = 7
y = 33
l = 40
h = 8
marco
sombra
control = 2
COLOR 7, 12
LOCATE 8, 49: PRINT "SECUENCIAL"
COLOR 7, 1
LOCATE 10, 35
PRINT "Procesamiento en el que"
PRINT "cada operaci'n"
LOCATE 11, 35
PRINT "precede a otra y sigue"
PRINT "à otra, sin"
LOCATE 12, 35

```

```

        PRINT "que nunca dos de "
        PRINT "ellas sean"
        LOCATE 13, 35
        PRINT "simultaneas."
        COLOR 7, 12
        h = 0
    END IF
    IF arri1 = 8 THEN
        z = 7
        v = 1
        COLOR z, v
        x = 7
        y = 33
        l = 40
        h = 8
        marco
        sombra
        control = 2
        COLOR 7, 12
        LOCATE 8, 49: PRINT "SERVOCONTROL"
        COLOR 7, 1
        LOCATE 10, 35
        PRINT "Control automatico de los mandos"
        LOCATE 11, 35: PRINT "un aparato."
        COLOR 7, 12
        h = 0
    END IF
    IF arri1 = 9 THEN
        z = 7
        v = 1
        COLOR z, v
        x = 7
        y = 33
        l = 40
        h = 9
        marco
        sombra
        control = 2
        COLOR 7, 12
        LOCATE 8, 44: PRINT "SISTEMAS DE COORDE"
        COLOR 7, 1
        LOCATE 10, 35
        PRINT "Se refiere a los diferentes"
        PRINT "tipos de"
        LOCATE 11, 35
        PRINT "coordenadas de las cuales"
        PRINT "se valen los"
        LOCATE 12, 35
        PRINT "diseñadores de robots, para"
        PRINT "definir la"
        LOCATE 13, 35
        PRINT "posición del efector en el"
        PRINT "espacio ,"
        LOCATE 14, 35
        PRINT "estas coordenadas pueden ser : "
    
```

```

LOCATE 15, 35
PRINT "Cartesianas, Cilindricas"
PRINT "o Esfericas."
COLOR 7, 12
h = 0
END IF
IF arri1 = 10 THEN
    z = 7
    v = 1
    COLOR z, v
    x = 7
    y = 33
    l = 40
    h = 8
    marco
    sombra
    control = 2
    COLOR 7, 12
    LOCATE 8, 50: PRINT "SUJECION"
    COLOR 7, 1
    LOCATE 10, 35
    PRINT "Accion de sujetar o mantener."
    COLOR 7, 12
    h = 0
    GOTO f
END IF
GOTO f
END IF
END IF
arri1 = 1
desplazar1 = 1
resetear
fondog1 = 10
fondog4 = 8
fondog3 = 8
x:
FOR i = 19 TO 21
    FOR p = 10 TO 60
        LOCATE i, p: PRINT " "
    NEXT p
NEXT i
glosariop
GOTO f
END IF
IF control11 = 2 THEN GOTO f
IF desplazar = 2 AND arri = 3 THEN ejecseq
IF desplazar = 5 AND bandera = 1 THEN
    arri = 1
    abajo
END IF
IF desplazar = 3 AND bandera = 1 THEN
    SCREEN 0, 0
    COLOR 7, 0
    CLS
    PRINT "presione 'EXIT' PARA REGRESAR AL PROGRAMA "
    SHELL

```

```

        COLOR 7, 12
        CLS
        menu
    END IF
    IF desplazar = 4 THEN
        COLOR 7, 0
        CLS
        END
    END IF

END SELECT
f:
LOOP UNTIL w$ = CHR$(27)
bandera = 0
desplazar1 = 0
control = 0
arri1 = 0
control10 = 0
control11 = 0
GOTO INIC
menu1:  bandera = 1
        desplazar = 1
        desplazar1 = 0
        control = 0
        arri1 = 0
        izquierda
        RETURN
der:    IF bandera = 0 THEN GOTO f1
        IF h = 0 THEN GOTO f1
        desplazar = desplazar + 1
        IF control = 1 THEN
            IF arri1 = 0 THEN arri1 = arri1 + 1
            desplazar1 = desplazar1 + 1
            IF desplazar1 > 2 THEN desplazar1 = 1
            desplazar = desplazar - 1
            abajo1
            glosariop
            GOTO f1
        END IF
        izquierda
        IF control11 = 2 THEN
            SELECT CASE desplazar
            CASE 1
                SHELL "archivo"
            CASE 2
                SHELL "ejecutar"
            CASE 5
                SHELL "ayuda"
            END SELECT
        END IF
        IF arri <> 0 THEN abajo
f1:
        RETURN
izq:   IF bandera = 0 THEN GOTO f2
        IF h = 0 THEN GOTO f2

```

```

desplazar = desplazar - 1
IF control = 1 THEN
    IF arri1 = 0 THEN arri1 = arri1 + 1
    desplazar1 = desplazar1 - 1
    IF desplazar1 < 1 THEN desplazar1 = 2
    desplazar = desplazar + 1
    abajol
    glosariop
    GOTO f2
END IF
    izquierda
IF control11 = 2 THEN
SELECT CASE desplazar
CASE 1
    SHELL "archivo"
CASE 2
    SHELL "ejecutar"
CASE 5
    SHELL "ayuda"
END SELECT
END IF
IF arri <> 0 THEN abajo
f2:
RETURN
mensub: IF bandera = 0 THEN GOTO f4
IF control11 = 1 THEN
arri = arri + 1
control11 = 2
SELECT CASE desplazar
CASE 1
    SHELL "archivo"
CASE 2
    SHELL "ejecutar"
CASE 5
    SHELL "ayuda"
END SELECT
arri = arri - 1
END IF
IF h = 0 THEN GOTO f4
arri = arri - 1
IF control = 1 THEN
    arri1 = arri1 - 1
    arri = arri + 1
    abajol
    GOTO f4
END IF
    abajo
f4:
RETURN
mensub1: IF bandera = 0 THEN GOTO f3
IF control11 = 1 THEN
arri = arri + 1
control11 = 2
SELECT CASE desplazar

```

```
CASE 1      SHELL "archivo"
CASE 2      SHELL "ejecutar"
CASE 5      IF arri = 1 THEN CLS
            SHELL "ayuda"
END SELECT
arri = arri - 1
END IF
IF h = 0 THEN GOTO f3
  arri = arri + 1
  IF control = 1 THEN
    arril = arril + 1
    arri = arri - 1
    abajol
    GOTO f3
  END IF
  abajo
f3:
RETURN
movej: IF arri = 1 AND desplazar = 2 THEN moveje
RETURN
```

```

SUB abajo
IF desplazar = 1 THEN GOTO y
IF arri > 4 THEN arri = 1
IF arri = -1 THEN arri = 1
IF arri < 1 THEN
arri = 4
loco = 1
END IF
y:
IF desplazar = 1 THEN
    IF arri > 5 THEN arri = 1
    IF arri = -1 THEN arri = 1
    IF arri < 1 THEN arri = 5
    IF arri = 1 THEN
        fondo6 = 10
        fondo7 = 7
        fondo8 = 7
        fondo9 = 7
        fondo10 = 7
    END IF
    IF arri = 2 THEN
        fondo6 = 7
        fondo7 = 10
        fondo8 = 7
        fondo9 = 7
        fondo10 = 7
    END IF
    IF arri = 3 THEN
        fondo6 = 7
        fondo7 = 7
        fondo8 = 10
        fondo9 = 7
        fondo10 = 7
    END IF
    IF arri = 4 THEN
        fondo6 = 7
        fondo7 = 7
        fondo8 = 7
        fondo9 = 10
        fondo10 = 7
    END IF
    IF arri = 5 THEN
        fondo6 = 7
        fondo7 = 7
        fondo8 = 7
        fondo9 = 7
        fondo10 = 10
    END IF
    menarch
END IF
IF desplazar = 2 THEN
    IF arri = 1 THEN
        fondo6 = 10
        fondo7 = 7
    
```

```

                fondo8 = 7
                fondo9 = 7
        END IF
        IF arri = 2 THEN
                fondo6 = 7
                fondo7 = 10
                fondo8 = 7
                fondo9 = 7
        END IF
        IF arri = 3 THEN
                fondo6 = 7
                fondo7 = 7
                fondo8 = 10
                fondo9 = 7
        END IF
        IF arri = 4 THEN
                fondo6 = 7
                fondo7 = 7
                fondo8 = 7
                fondo9 = 10
        END IF
        END IF
                monejec
END IF
IF desplazar = 3 THEN
        CLS
        menu
END IF
IF desplazar = 4 THEN
        CLS
        menu
END IF
IF desplazar = 5 THEN
        IF arri = 4 AND loco = 1 THEN arri = 3
        IF arri > 3 THEN arri = 1
        IF arri = -1 THEN arri = 1
        IF arri < 1 THEN arri = 3
        IF arri = 1 THEN
                fondo6 = 10
                fondo7 = 7
                fondo8 = 7
                fondo9 = 7
        END IF
        IF arri = 2 THEN
                fondo6 = 7
                fondo7 = 7
                fondo8 = 10
                fondo9 = 7
        END IF
        IF arri = 3 THEN
                fondo6 = 7
                fondo7 = 7
                fondo8 = 7
                fondo9 = 10
        END IF

```

menayuda

g:

END IF

END SUB

```
SUB abajo1
IF desplazar1 = 1 AND arri1 = -1 THEN arri1 = 1
IF desplazar1 = 1 AND arri1 < 1 THEN arri1 = 11
IF desplazar1 = 1 AND arri1 > 11 THEN arri1 = 1
IF desplazar1 = 2 AND arri1 = -1 THEN arri1 = 1
IF desplazar1 = 2 AND arri1 < 1 THEN arri1 = 10
IF desplazar1 = 2 AND arri1 > 10 THEN arri1 = 1
IF desplazar1 = 1 THEN
    IF arri1 = 1 THEN
        fondog1 = 10
        fondog2 = 8
        fondog3 = 8
        fondog4 = 8
        fondog5 = 8
        fondog6 = 8
        fondog7 = 8
        fondog8 = 8
        fondog9 = 8
        fondog10 = 8
        fondog11 = 8
        fondog12 = 8
        fondog13 = 8
        fondog14 = 8
        fondog15 = 8
        fondog16 = 8
        fondog17 = 8
        fondog18 = 8
        fondog19 = 8
        fondog20 = 8
        fondog21 = 8
    END IF
    IF arri1 = 2 THEN
        fondog1 = 8
        fondog2 = 10
        fondog3 = 8
        fondog4 = 8
        fondog5 = 8
        fondog6 = 8
        fondog7 = 8
        fondog8 = 8
        fondog9 = 8
        fondog10 = 8
        fondog11 = 8
        fondog12 = 8
        fondog13 = 8
        fondog14 = 8
        fondog15 = 8
        fondog16 = 8
        fondog17 = 8
        fondog18 = 8
        fondog19 = 8
        fondog20 = 8
        fondog21 = 8
    END IF
END IF
```

```
IF arr11 = 3 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 10
    fondog4 = 8
    fondog5 = 8
    fondog6 = 8
    fondog7 = 8
    fondog8 = 8
    fondog9 = 8
    fondog10 = 8
    fondog11 = 8
    fondog12 = 8
    fondog13 = 8
    fondog14 = 8
    fondog15 = 8
    fondog16 = 8
    fondog17 = 8
    fondog18 = 8
    fondog19 = 8
    fondog20 = 8
    fondog21 = 8
```

```
END IF
```

```
IF arr11 = 4 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 10
    fondog5 = 8
    fondog6 = 8
    fondog7 = 8
    fondog8 = 8
    fondog9 = 8
    fondog10 = 8
    fondog11 = 8
    fondog12 = 8
    fondog13 = 8
    fondog14 = 8
    fondog15 = 8
    fondog16 = 8
    fondog17 = 8
    fondog18 = 8
    fondog19 = 8
    fondog20 = 8
    fondog21 = 8
```

```
END IF
```

```
IF arr11 = 5 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 8
    fondog5 = 10
    fondog6 = 8
    fondog7 = 8
    fondog8 = 8
```

```
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr11 = 6 THEN
  fondog1 = 8
  fondog2 = 8
  fondog3 = 8
  fondog4 = 8
  fondog5 = 8
  fondog6 = 10
  fondog7 = 8
  fondog8 = 8
  fondog9 = 8
  fondog10 = 8
  fondog11 = 8
  fondog12 = 8
  fondog13 = 8
  fondog14 = 8
  fondog15 = 8
  fondog16 = 8
  fondog17 = 8
  fondog18 = 8
  fondog19 = 8
  fondog20 = 8
  fondog21 = 8
```

```
END IF
```

```
IF arr11 = 7 THEN
  fondog1 = 8
  fondog2 = 8
  fondog3 = 8
  fondog4 = 8
  fondog5 = 8
  fondog6 = 8
  fondog7 = 10
  fondog8 = 8
  fondog9 = 8
  fondog10 = 8
  fondog11 = 8
  fondog12 = 8
  fondog13 = 8
  fondog14 = 8
  fondog15 = 8
  fondog16 = 8
  fondog17 = 8
```

```
        fondog18 = 8
        fondog19 = 8
        fondog20 = 8
        fondog21 = 8
END IF
IF arr1 = 8 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 8
    fondog5 = 8
    fondog6 = 8
    fondog7 = 8
    fondog8 = 10
    fondog9 = 8
    fondog10 = 8
    fondog11 = 8
    fondog12 = 8
    fondog13 = 8
    fondog14 = 8
    fondog15 = 8
    fondog16 = 8
    fondog17 = 8
    fondog18 = 8
    fondog19 = 8
    fondog20 = 8
    fondog21 = 8
END IF
IF arr1 = 9 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 8
    fondog5 = 8
    fondog6 = 8
    fondog7 = 8
    fondog8 = 8
    fondog9 = 10
    fondog10 = 8
    fondog11 = 8
    fondog12 = 8
    fondog13 = 8
    fondog14 = 8
    fondog15 = 8
    fondog16 = 8
    fondog17 = 8
    fondog18 = 8
    fondog19 = 8
    fondog20 = 8
    fondog21 = 8
END IF
IF arr1 = 10 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
```

```
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 10
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr11 = 11 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 10
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
END IF
```

```
IF desplazar1 = 2 THEN
```

```
IF arr11 = 1 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
```

```
fondog11 = 8
fondog12 = 10
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr11 = 2 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 10
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr11 = 3 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 10
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
```

```
        fondog20 = 8
        fondog21 = 8
END IF
IF arr1 = 4 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 8
    fondog5 = 8
    fondog6 = 8
    fondog7 = 8
    fondog8 = 8
    fondog9 = 8
    fondog10 = 8
    fondog11 = 8
    fondog12 = 8
    fondog13 = 8
    fondog14 = 8
    fondog15 = 10
    fondog16 = 8
    fondog17 = 8
    fondog18 = 8
    fondog19 = 8
    fondog20 = 8
    fondog21 = 8
END IF
IF arr1 = 5 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 8
    fondog5 = 8
    fondog6 = 8
    fondog7 = 8
    fondog8 = 8
    fondog9 = 8
    fondog10 = 8
    fondog11 = 8
    fondog12 = 8
    fondog13 = 8
    fondog14 = 8
    fondog15 = 8
    fondog16 = 10
    fondog17 = 8
    fondog18 = 8
    fondog19 = 8
    fondog20 = 8
    fondog21 = 8
END IF
IF arr1 = 6 THEN
    fondog1 = 8
    fondog2 = 8
    fondog3 = 8
    fondog4 = 8
    fondog5 = 8
```

```
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 10
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr1 = 7 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 10
fondog19 = 8
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr1 = 8 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
```

```
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 10
fondog20 = 8
fondog21 = 8
```

```
END IF
```

```
IF arr1 = 9 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 10
fondog21 = 8
```

```
END IF
```

```
IF arr1 = 10 THEN
```

```
fondog1 = 8
fondog2 = 8
fondog3 = 8
fondog4 = 8
fondog5 = 8
fondog6 = 8
fondog7 = 8
fondog8 = 8
fondog9 = 8
fondog10 = 8
fondog11 = 8
fondog12 = 8
fondog13 = 8
fondog14 = 8
fondog15 = 8
fondog16 = 8
fondog17 = 8
fondog18 = 8
fondog19 = 8
fondog20 = 8
fondog21 = 10
```

```
END IF
```

```
END IF
```

```
glosariof
```

```
END SUB
```

```

SUB abrir
FOR g = 1 TO 30
IF arch$(g) = "" THEN cv = cv + 1
NEXT g
IF cv = 30 THEN
    op = 1
    GOTO fine
END IF
CLS
control9 = 2
cv = 0
z = 7
v = 1
COLOR z, v
x = 16
y = 23
l = 27
h = 3
marco
sombra
COLOR 7, 1
LOCATE 16, 35: PRINT " ABRIR "
OPEN "dir" FOR INPUT AS #1
INPUT #1, q, arch$(1), arch$(2), arch$(3), arch$(4), arch$(5)
INPUT #1, arch$(6), arch$(7), arch$(8), arch$(9), arch$(10)
INPUT #1, arch$(11), arch$(12), arch$(13), arch$(14), arch$(15)
INPUT #1, arch$(16), arch$(17), arch$(18), arch$(19), arch$(20)
INPUT #1, arch$(21), arch$(22), arch$(23), arch$(24), arch$(25)
INPUT #1, arch$(26), arch$(27), arch$(28), arch$(29), arch$(30)
CLOSE
z = 7
v = 8
COLOR z, v
x = 1
y = 8
l = 40
h = 11
marco
sombra
COLOR 7, 8
FOR x = 6 TO 36
    IF arch$(x - 5) <> "" THEN
        IF x - 5 > 10 THEN
            IF x - 5 > 20 THEN
                LOCATE x - 24, 40: PRINT arch$(x - 5) + ".SEQ"
                GOTO 1
            END IF
            LOCATE x - 14, 25: PRINT arch$(x - 5) + ".SEQ"
            GOTO 1
        END IF
        LOCATE x - 4, 10: PRINT arch$(x - 5) + ".SEQ"
    l:
    END IF
NEXT x

```

```

v:
k = 0
COLOR 7, 1
LOCATE 17, 24: PRINT "nombre :"
LOCATE 17, 31: INPUT nomarch$
nomarch$ = UCASE$(nomarch$)
IF LEN(nomarch$) > 8 THEN
    LOCATE 17, 33: PRINT "
    GOTO v
END IF
u = 1
DO
    k = k + 1
    IF nomarch$ = arch$(k) THEN
        u = 3
    END IF
    IF k = q + 1 THEN u = 2
LOOP UNTIL u > 1
IF u = 2 THEN
    LOCATE 17, 33: PRINT "
    GOTO v
END IF
OPEN nomarch$ + ".SEQ" FOR INPUT AS #1
INPUT #1, s, m14, m24, m34, m44, m54, m64
INPUT #1, m11, m21, m31, m41, m51, m61
INPUT #1, m12, m22, m32, m42, m52, m62
INPUT #1, m13, m23, m33, m43, m53, m63
CLOSE
OPEN "tempo" FOR OUTPUT AS #1
WRITE #1, control9, er$, gh, columna, fond1, fond12, fond13,
WRITE #1, arri, desplazar
WRITE #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
WRITE #1, m44, m54, m64
WRITE #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
WRITE #1, m53, m63, m14, m24
WRITE #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
WRITE #1, m62, m13, m23, m33
WRITE #1, fond62, fond63, fond64, m11, m21, m31, m41, m51,
WRITE #1, m61, m12, m22, m32
WRITE #1, nomarch$
WRITE #1, limmax11, limmin11, limmax21, limmin21, limmax31,
WRITE #1, limmin31, limmax41
WRITE #1, limmin41, limmin12, limmax12, limmin22, limmax22,
WRITE #1, limmin32, limmax32
WRITE #1, limmax42, limmin42, limmax52, limmin52, limmax62,
WRITE #1, limmin62, limmax13
WRITE #1, limmin13, limmin23, limmax23, limmin33, limmax33,
WRITE #1, limmin43, limmax43
WRITE #1, limmax53, limmin53, limmax63, limmin63, limmax14,
WRITE #1, limmin14, limmax24
WRITE #1, limmin24, limmin34, limmax34, limmin44, limmax44,
WRITE #1, limmin54, limmax54
WRITE #1, limmax64, limmin64, limmax51, limmin51, limmax61,
WRITE #1, limmin61
CLOSE
fine:
COLOR 7, 12

```

```

SUB ciclico
DO
p$ = INKEY$
limmin11 = -1000: limmax11 = 1000: limmin21 = -600: limmax21 = 600
limmin31 = -500: limmax31 = 500: limmin61 = -1800: limmax61 = 0
limmin41 = -900: limmax41 = 900: limmin51 = -1800: limmax51 = 1800
limmin12 = -1000: limmax12 = 1000: limmin22 = -600: limmax22 = 600
limmin32 = -500: limmax32 = 500: limmin62 = -1800: limmax62 = 0
limmin42 = -900: limmax42 = 900: limmin52 = -1800: limmax52 = 1800
limmin13 = -1000: limmax13 = 1000: limmin23 = -600: limmax23 = 600
limmin33 = -500: limmax33 = 500: limmin63 = -1800: limmax63 = 0
limmin43 = -900: limmax43 = 900: limmin53 = -1800: limmax53 = 1800
limmin14 = -1000: limmax14 = 1000: limmin24 = -600: limmax24 = 600
limmin34 = -500: limmax34 = 500: limmin64 = -1800: limmax64 = 0
limmin44 = -900: limmax44 = 900: limmin54 = -1800: limmax54 = 1800
OPEN "tempo" FOR INPUT AS #1
INPUT #1, control9, er$, gh, columna, fond1, fond12, fond13,
INPUT #1, arri, desplazar
INPUT #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
INPUT #1, m44, m54, m64
INPUT #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
INPUT #1, m53, m63, m14, m24
INPUT #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
INPUT #1, m62, m13, m23, m33
INPUT #1, fond62, fond63, fond64, m11, m21, m31, m41, m51,
INPUT #1, m61, m12, m22, m32
INPUT #1, nomarch$
CLOSE
m1 = m11: m2 = m21: m3 = m31: m4 = m41: m5 = m51: m6 = m61
LPRINT "M"; m1; ", "; m2; ", "; m3; ", "; m4; ", ";
LPRINT " "; m5; ", "; m6
LPRINT "Z"
m1 = m12: m2 = m22: m3 = m32: m4 = m42: m5 = m52: m6 = m62
LPRINT "M"; m1; ", "; m2; ", "; m3; ", "; m4; ", ";
LPRINT " "; m5; ", "; m6
LPRINT "Z"
m1 = m13: m2 = m23: m3 = m33: m4 = m43: m5 = m53: m6 = m63
LPRINT "M"; m1; ", "; m2; ", "; m3; ", "; m4; ", ";
LPRINT " "; m5; ", "; m6
LPRINT "Z"
m1 = m14: m2 = m24: m3 = m34: m4 = m44: m5 = m54: m6 = m64
LPRINT "M"; m1; ", "; m2; ", "; m3; ", "; m4; ", ";
LPRINT " "; m5; ", "; m6
m1 = -m11 - m12 - m13 - m14: m2 = -m21 - m22 - m23 - m24
m3 = -m31 - m32 - m33 - m34: m4 = -m41 - m42 - m43 - m44
m5 = -m51 - m52 - m53 - m54: m6 = -m61 - m62 - m63 - m64
o = m4
m4 = m4 + m5
m5 = o - m5
LPRINT "M"; m1; ", "; m2; ", "; m3; ", "; m4; ", ";
LPRINT " "; m5; ", "; m6
m1 = 0: m2 = 0: m3 = 0: m4 = 0: m5 = 0: m6 = 0
LPRINT "Z"
p$ = INKEY$
LOOP UNTIL p$ = CHR$(27)
END SUE

```

```

SUB ejecseq
limmin11 = -1000: limmax11 = 1000: limmin21 = -600
limmax21 = 600
limmin31 = -500: limmax31 = 500: limmin61 = -1800
limmax61 = 0
limmin41 = -900: limmax41 = 900: limmin51 = -1800
limmax51 = 1800
limmin12 = -1000: limmax12 = 1000: limmin22 = -600
limmax22 = 600
limmin32 = -500: limmax32 = 500: limmin62 = -1800
limmax62 = 0
limmin42 = -900: limmax42 = 900: limmin52 = -1800
limmax52 = 1800
limmin13 = -1000: limmax13 = 1000: limmin23 = -600
limmax23 = 600
limmin33 = -500: limmax33 = 500: limmin63 = -1800
limmax63 = 0
limmin43 = -900: limmax43 = 900: limmin53 = -1800
limmax53 = 1800
limmin14 = -1000: limmax14 = 1000: limmin24 = -600
limmax24 = 600
limmin34 = -500: limmax34 = 500: limmin64 = -1800
limmax64 = 0
limmin44 = -900: limmax44 = 900: limmin54 = -1800
limmax54 = 1800
OPEN "tempo" FOR INPUT AS #1
INPUT #1, control9, er$, gh, columna, fond1, fond12, fond13,
INPUT #1, arri, desplazar
INPUT #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
INPUT #1, m44, m54, m64
INPUT #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
INPUT #1, m53, m63, m14, m24
INPUT #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
INPUT #1, m62, m13, m23, m33
INPUT #1, fond62, fond63, fond64, m11, m21, m31, m41, m51,
INPUT #1, m61, m12, m22, m32
INPUT #1, nomarch$
CLOSE
m1 = m11: m2 = m21: m3 = m31: m4 = m41: m5 = m51: m6 = m61
LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; ", "
LPRINT " "; m5; ", "; " "; m6
LPRINT "Z"
m1 = m12: m2 = m22: m3 = m32: m4 = m42: m5 = m52: m6 = m62
LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; ", "
LPRINT " "; m5; ", "; " "; m6
LPRINT "Z"
m1 = m13: m2 = m23: m3 = m33: m4 = m43: m5 = m53: m6 = m63
LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; ", "
LPRINT " "; m5; ", "; " "; m6
LPRINT "Z"
m1 = m14: m2 = m24: m3 = m34: m4 = m44: m5 = m54: m6 = m64
LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; ", "
LPRINT " "; m5; ", "; " "; m6
LPRINT "Z"

```

```
m1 = -m11 - m12 - m13 - m14: m2 = -m21 - m22 - m23 - m24
m3 = -m31 - m32 - m33 - m34: m4 = -m41 - m42 - m43 - m44
m5 = -m51 - m52 - m53 - m54: m6 = -m61 - m62 - m63 - m64
o = m4
m4 = m4 + m5
m5 = o - m5
LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; ", "
LPRINT " "; m5; ", "; " "; m6
m1 = 0: m2 = 0: m3 = 0: m4 = 0: m5 = 0: m6 = 0
END SUB
```

```

SUB gcomo
COLOR 15, 12
FOR d = 4 TO 21
  FOR p = 23 TO 60
    LOCATE d, p: PRINT " "
  NEXT p
NEXT d
som:
OPEN "tempo" FOR INPUT AS #1
INPUT #1, control9, er$, gh, columna, fond1, fond12, fond13,
INPUT #1, arri, desplazar
INPUT #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
INPUT #1, m44, m54, m64
INPUT #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
INPUT #1, m53, m63, m14, m24
INPUT #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
INPUT #1, m62, m13, m23, m33
INPUT #1, fond62, fond63, fond64, m11, m21, m31, m41, m51,
INPUT #1, m61, m12, m22, m32
INPUT #1, nomarch$
INPUT #1, limmax11, limmin11, limmax21, limmin21, limmax31,
INPUT #1, limmin31, limmax41
INPUT #1, limmin41, limmin12, limmax12, limmin22, limmax22,
INPUT #1, limmin32, limmax32
INPUT #1, limmax42, limmin42, limmax52, limmin52, limmax62,
INPUT #1, limmin62, limmax13
INPUT #1, limmin13, limmin23, limmax23, limmin33, limmax33,
INPUT #1, limmin43, limmax43
INPUT #1, limmax53, limmin53, limmax63, limmin63, limmax14,
INPUT #1, limmin14, limmax24
INPUT #1, limmin24, limmin34, limmax34, limmin44, limmax44,
INPUT #1, limmin54, limmax54
INPUT #1, limmax64, limmin64, limmax51, limmin51, limmax61,
INPUT #1, limmin61
CLOSE
OPEN "dir" FOR INPUT AS #1
INPUT #1, q, arch$(1), arch$(2), arch$(3), arch$(4), arch$(5)
INPUT #1, arch$(6), arch$(7), arch$(8), arch$(9), arch$(10)
INPUT #1, arch$(11), arch$(12), arch$(13), arch$(14), arch$(15)
INPUT #1, arch$(16), arch$(17), arch$(18), arch$(19), arch$(20)
INPUT #1, arch$(21), arch$(22), arch$(23), arch$(24), arch$(25)
INPUT #1, arch$(26), arch$(27), arch$(28), arch$(29), arch$(30)
CLOSE
z = 7
v = 1
COLOR z, v
x = 10
y = 19
l = 40
h = 8
marco
sombra
COLOR 7, 1
LOCATE 10, 29: PRINT " GUARDAR COMO "

```

```

LOCATE 11, 22: PRINT "nombre del archivo:"
LOCATE 12, 22: INPUT nomarch$
nomarch$ = UCASE$(nomarch$)
IF nomarch$ = "TEMPO" THEN GOTO som
IF nomarch$ = "DIR" THEN GOTO som
IF nomarch$ = "PRE" THEN GOTO som
i = 0
DO
    i = i + 1
    IF arch$(i) = nomarch$ THEN
        respuesta$ = "s"
        i = 59
ELSE
    respuesta$ = "n"
END IF
LOOP UNTIL i > 55
IF respuesta$ = "s" THEN GOTO som
OPEN nomarch$ + ".SEQ" FOR OUTPUT AS #1
WRITE #1, s, m14, m24, m34, m44, m54, m64
WRITE #1, m11, m21, m31, m41, m51, m61
WRITE #1, m12, m22, m32, m42, m52, m62
WRITE #1, m13, m23, m33, m43, m53, m63
CLOSE
q = q + 1
arch$(q) = nomarch$
OPEN "dir" FOR OUTPUT AS #1
WRITE #1, q, arch$(1), arch$(2), arch$(3), arch$(4), arch$(5)
WRITE #1, arch$(6), arch$(7), arch$(8), arch$(9), arch$(10)
WRITE #1, arch$(11), arch$(12), arch$(13), arch$(14), arch$(15)
WRITE #1, arch$(16), arch$(17), arch$(18), arch$(19), arch$(20)
WRITE #1, arch$(21), arch$(22), arch$(23), arch$(24), arch$(25)
WRITE #1, arch$(26), arch$(27), arch$(28), arch$(29), arch$(30)
CLOSE
OPEN "tempo" FOR OUTPUT AS #1
WRITE #1, control9, er$, gh, columna, fond1, fond12, fond13,
WRITE #1, arri, desplazar
WRITE #1, fond14, fond2, fond22, fond23, fond24, fond3, m34,
WRITE #1, m44, m54, m64
WRITE #1, fond32, fond33, fond34, fond4, fond42, fond43, m43,
WRITE #1, m53, m63, m14, m24
WRITE #1, fond44, fond5, fond52, fond53, fond54, fond6, m52,
WRITE #1, m62, m13, m23, m33
WRITE #1, fond62, fond63, fond64, m11, m21, m31, m41, m51,
WRITE #1, m61, m12, m22, m32
WRITE #1, nomarch$
WRITE #1, limmax11, limmin11, limmax21, limmin21, limmax31,
WRITE #1, limmin31, limmax41
WRITE #1, limmin41, limmin12, limmax12, limmin22, limmax22,
WRITE #1, limmin32, limmax32
WRITE #1, limmax42, limmin42, limmax52, limmin52, limmax62,
WRITE #1, limmin62, limmax13
WRITE #1, limmin13, limmin23, limmax23, limmin33, limmax33,
WRITE #1, limmin43, limmax43
WRITE #1, limmax53, limmin53, limmax63, limmin63, limmax14,
WRITE #1, limmin14, limmax24

```

```
WRITE #1, limmin24, limmin34, limmax34, limmin44, limmax44,  
WRITE #1, limmin54, limmax54  
WRITE #1, limmax64, limmin64, limmax51, limmin51, limmax61,  
WRITE #1, limmin61  
CLOSE  
resetear1  
arri = 4  
abajo  
END SUB
```

```

SUB glosariop
IF desplazarl <> 1 AND desplazarl <> 2 THEN fondog4 = 8
control = 1
COLOR 7, 12
z = 15
v = 8
COLOR z, v
x = 5
y = 5
l = 70
h = 13
marco
COLOR z, fondog1: LOCATE 6, 11: PRINT "Accionamiento      "
COLOR z, fondog2: LOCATE 7, 11: PRINT "Articulaciones    "
COLOR z, fondog3: LOCATE 8, 11: PRINT "Autom tico        "
COLOR z, fondog4: LOCATE 9, 11: PRINT "Capacidad de carga"
COLOR z, fondog5: LOCATE 10, 11: PRINT "Estator           "
COLOR z, fondog6: LOCATE 11, 11: PRINT "Efector           "
COLOR z, fondog7: LOCATE 12, 11: PRINT "Grados de libertad"
COLOR z, fondog8: LOCATE 13, 11: PRINT "Manipulador      "
COLOR z, fondog9: LOCATE 14, 11: PRINT "Microprocesador  "
COLOR z, fondog10: LOCATE 15, 11: PRINT "Miniaturizado    "
COLOR z, fondog11: LOCATE 16, 11: PRINT "Perifricos       "
COLOR z, fondog12: LOCATE 6, 44: PRINT "Precisi'n        "
COLOR z, fondog13: LOCATE 7, 44: PRINT "Puerto           "
COLOR z, fondog14: LOCATE 8, 44: PRINT "Resoluci'n      "
COLOR z, fondog15: LOCATE 9, 44: PRINT "Robots           "
COLOR z, fondog16: LOCATE 10, 44: PRINT "Rotor            "
COLOR z, fondog17: LOCATE 11, 44: PRINT "Rutina           "
COLOR z, fondog18: LOCATE 12, 44: PRINT "Secuencial       "
COLOR z, fondog19: LOCATE 13, 44: PRINT "Servocontrol     "
COLOR z, fondog20: LOCATE 14, 44: PRINT "Sistemas de coordenadas"
COLOR z, fondog21: LOCATE 15, 44: PRINT "Sujeci'n        "
COLOR 7, 12
END SUB

```

```
SUB guardar
CLS
```

```
OPEN "tempo" FOR INPUT AS #1
INPUT #1, control9, er$, gh, columna, fond1, fond12,
INPUT #1, fond13, arri, desplazar
INPUT #1, fond14, fond2, fond22, fond23, fond24, fond3.
INPUT #1, m34, m44, m54, m64
INPUT #1, fond32, fond33, fond34, fond4, fond42, fond43,
INPUT #1, m43, m53, m63, m14, m24
INPUT #1, fond44, fond5, fond52, fond53, fond54, fond6,
INPUT #1, m52, m62, m13, m23, m33
INPUT #1, fond62, fond63, fond64, m11, m21, m31, m41,
INPUT #1, m51, m61, m12, m22, m32
INPUT #1, nomarch$
INPUT #1, limmax11, limmin11, limmax21, limmin21,
INPUT #1, limmax31, limmin31, limmax41
INPUT #1, limmin41, limmin12, limmax12, limmin22,
INPUT #1, limmax22, limmin32, limmax32
INPUT #1, limmax42, limmin42, limmax52, limmin52,
INPUT #1, limmax62, limmin62, limmax13
INPUT #1, limmin13, limmin23, limmax23, limmin33,
INPUT #1, limmax33, limmin43, limmax43
INPUT #1, limmax53, limmin53, limmax63, limmin63,
INPUT #1, limmax14, limmin14, limmax24
INPUT #1, limmin24, limmin34, limmax34, limmin44,
INPUT #1, limmax44, limmin54, limmax54
INPUT #1, limmax64, limmin64, limmax51, limmin51,
INPUT #1, limmax61, limmin61
CLOSE
```

```
IF nomarch$ = "" THEN
lsom:
```

```
OPEN "dir" FOR INPUT AS #1
INPUT #1, q, arch$(1), arch$(2), arch$(3), arch$(4), arch$(5)
INPUT #1, arch$(6), arch$(7), arch$(8), arch$(9), arch$(10)
INPUT #1, arch$(11), arch$(12), arch$(13), arch$(14), arch$(15)
INPUT #1, arch$(16), arch$(17), arch$(18), arch$(19), arch$(20)
INPUT #1, arch$(21), arch$(22), arch$(23), arch$(24), arch$(25)
INPUT #1, arch$(26), arch$(27), arch$(28), arch$(29), arch$(30)
CLOSE
z = 7
v = 8
COLOR z, v
x = 1
y = 8
l = 40
h = 11
marco
sombra
COLOR 7, 8
FOR x = 6 TO 36
  IF arch$(x - 5) <> "" THEN
    IF x - 5 > 10 THEN
      IF x - 5 > 20 THEN
        LOCATE x - 24, 40: PRINT arch$(x - 5) + ".SEQ"
```

```

                                GOTO 11
                                END IF
                                LOCATE x - 14, 25: PRINT arch$(x - 5) + ".SEQ"
                                GOTO 11
                                END IF
                                LOCATE x - 4, 10: PRINT arch$(x - 5) + ".SEQ"
11:
                                END IF
                                NEXT x
                                z = 7
                                v = 1
                                COLOR z, v
                                x = 16
                                y = 23
                                l = 27
                                h = 3
                                marco
                                sombra
                                COLOR 7, 1
                                LOCATE 16, 32: PRINT " GUARDAR "
                                LOCATE 17, 25: PRINT "nombre del archivo:"
                                LOCATE 18, 25: INPUT nomarch$
                                nomarch$ = UCASE$(nomarch$)
                                IF nomarch$ = "TEMPO" THEN GOTO 11
                                IF nomarch$ = "DIR" THEN GOTO 11
                                IF nomarch$ = "PRE" THEN GOTO 11
                                i = 0
                                DO
                                    i = i + 1
                                    IF arch$(i) = nomarch$ THEN
                                        respuesta$ = "s"
                                        i = 59
                                    ELSE respuesta$ = "n"
                                    END IF
                                LOOP UNTIL i > 55
                                IF respuesta$ = "s" THEN GOTO lsom
                                OPEN nomarch$ + ".SEQ" FOR OUTPUT AS #1
                                WRITE #1, s, m14, m24, m34, m44, m54, m64
                                WRITE #1, m11, m21, m31, m41, m51, m61
                                WRITE #1, m12, m22, m32, m42, m52, m62
                                WRITE #1, m13, m23, m33, m43, m53, m63
                                CLOSE
                                q = q - 1
                                arch$(q) = nomarch$
                                OPEN "dir" FOR OUTPUT AS #1
                                WRITE #1, q, arch$(1), arch$(2), arch$(3), arch$(4), arch$(5)
                                WRITE #1, arch$(6), arch$(7), arch$(8), arch$(9), arch$(10)
                                WRITE #1, arch$(11), arch$(12), arch$(13), arch$(14), arch$(15)
                                WRITE #1, arch$(16), arch$(17), arch$(18), arch$(19), arch$(20)
                                WRITE #1, arch$(21), arch$(22), arch$(23), arch$(24), arch$(25)
                                WRITE #1, arch$(26), arch$(27), arch$(28), arch$(29), arch$(30)
                                CLOSE
                                arri = 2
                                resetear1
                                CLS

```

```
      abajo
ELSE nomarch$ = UCASE$(nomarch$)
  i = 0
  DO
    i = i + 1
    IF arch$(i) = nomarch$ THEN
      r = i
      i = 59
    END IF
  LOOP UNTIL i > 55
  OPEN nomarch$ + ".SEQ" FOR OUTPUT AS #1
  WRITE #1, s, m14, m24, m34, m44, m54, m64
  WRITE #1, m11, m21, m31, m41, m51, m61
  WRITE #1, m12, m22, m32, m42, m52, m62
  WRITE #1, m13, m23, m33, m43, m53, m63
  CLOSE
  arch$(r) = nomarch$
  arri = 2
  CLS
  abajo
  END IF
abajo
END SUB
```

```
SUB izquierda
COLOR 7, 12
IF desplazar > 5 THEN desplazar = 1
IF desplazar = -1 THEN desplazar = 1
IF desplazar < 1 THEN desplazar = 5
IF desplazar = 1 THEN
    fondo1 = 10
    fondo2 = 7
    fondo3 = 7
    fondo4 = 7
    fondo5 = 7
END IF
IF desplazar = 2 THEN
    fondo1 = 7
    fondo2 = 10
    fondo3 = 7
    fondo4 = 7
    fondo5 = 7
END IF
IF desplazar = 3 THEN
    fondo1 = 7
    fondo2 = 7
    fondo3 = 10
    fondo4 = 7
    fondo5 = 7
END IF
IF desplazar = 4 THEN
    fondo1 = 7
    fondo2 = 7
    fondo3 = 7
    fondo4 = 10
    fondo5 = 7
END IF
IF desplazar = 5 THEN
    fondo1 = 7
    fondo2 = 7
    fondo3 = 7
    fondo4 = 7
    fondo5 = 10
END IF
CLS
menu
END SUB
```

```

SUB marco
LOCATE x, y: PRINT CHR$(218)
LOCATE x, y + 1: PRINT STRING$(1, CHR$(196))
LOCATE x, l + y + 1: PRINT CHR$(191)
LOCATE x + h, y: PRINT CHR$(192)
LOCATE x + h, y + 1: PRINT STRING$(1, CHR$(196))
LOCATE x + h, l + y + 1: PRINT CHR$(217)
FOR i = 1 TO h - 1
  LOCATE x + i, y: PRINT CHR$(179) + STRING$(1, " ")
  LOCATE x + i, l + y + 1: PRINT CHR$(179)
NEXT i
END SUB

```

```

SUB menarch
menu
COLOR 0, 7
h = 6
l = 17
x = 2
y = 1
marco
LOCATE 3, 2: COLOR 0, fondo6: PRINT " A rir ";
LOCATE 3, 5: COLOR 15, fondo6: PRINT "b";
LOCATE 4, 2: COLOR 15, fondo7: PRINT " G";
COLOR 0, fondo7: PRINT "uardar ";
LOCATE 5, 2: COLOR 15, fondo8: PRINT " N";
COLOR 0, fondo8: PRINT "va. seq. ";
LOCATE 6, 2: COLOR 0, fondo9: PRINT " Guardar omo ";
LOCATE 6, 12: COLOR 15, fondo9: PRINT "C";
LOCATE 7, 2: COLOR 0, fondo10: PRINT " M dificar ";
LOCATE 7, 5: COLOR 15, fondo10: PRINT "o"
COLOR 7, 12
END SUB

```

```

SUB menayuda
menu
COLOR 0, 7
h = 4
l = 18
x = 2
y = 51
marco
LOCATE 3, 52:
COLOR 15, fondo6: PRINT " T";
COLOR 0, fondo6: PRINT "utorial ";
LOCATE 4, 52: COLOR 0, fondo8: PRINT " Gene alidades ";
LOCATE 4, 57: COLOR 15, fondo8: PRINT "r";
LOCATE 5, 52: COLOR 0, fondo9: PRINT " G osario ";
LOCATE 5, 54: COLOR 15, fondo9: PRINT "l";
COLOR 7, 12
END SUB

```

```

SUB menejec
menu
COLOR 0, 7
h = 5
l = 21
x = 2
y = 13
marco
LOCATE 3, 14: COLOR 15, fondo6: PRINT " M";
COLOR 0, fondo6: PRINT "ov por eje ";
LOCATE 4, 14: COLOR 0, fondo7: PRINT " Mo Simult neo ";
LOCATE 4, 17: COLOR 15, fondo7: PRINT "v";
LOCATE 5, 14: COLOR 0, fondo8: PRINT " E ecuci'n de seq. ";
LOCATE 5, 16: COLOR 15, fondo8: PRINT "j";
LOCATE 6, 14: COLOR 0, fondo9: PRINT " Mov. C clico ";
LOCATE 6, 21: COLOR 15, fondo9: PRINT " /";
COLOR 7, 12
END SUB

```

```

SUB menger
COLOR 7, 12
FOR fg = 4 TO 20
    FOR hf = 23 TO 52
        LOCATE fg, hf: PRINT " "
    NEXT hf
NEXT fg

```

```

abajo
z = 7
v = 1
COLOR z, v
x = 9
y = 17
l = 40
h = 6
marco
sombra
LOCATE 10, 21: COLOR 7, fondos: PRINT "Estructura del brazo robot "
LOCATE 12, 21: COLOR 7, fondos1: PRINT "Desplazamiento en el programa "
LOCATE 14, 21: COLOR 7, fondos2: PRINT "Microprocesador z-80 (pines) "
control3 = 0
END SUB

```

```

SUB menu
IF control10 = 1 THEN GOTO x1
IF control3 = 1 THEN GOTO x1
ver
x1:
COLOR 7, 12
LOCATE 1, 1:
COLOR 15, fondo1: PRINT " A";
COLOR 0, fondo1: PRINT "rchivo ";

```

```
COLOR 15, fondo2: PRINT " E";
COLOR 0, fondo2: PRINT "jecutar ";
COLOR 15, fondo3: PRINT " D";
COLOR 0, fondo3: PRINT "OSprompt ";
COLOR 15, fondo4: PRINT " S";
COLOR 0, fondo4: PRINT "alir ";
COLOR 0, fondo5: PRINT " A uda ";
LOCATE 1, 55: COLOR 15, fondo5: PRINT "y";
COLOR 12, 7: LOCATE 1, 61: PRINT " ";
COLOR 7, 12
END SUB
```

```

SUB moveje
l = 76
h = 20
x = 2
y = 2
marco
COLOR 10, 12
LOCATE 4, 10: PRINT "m1: motor de la base"
LOCATE 5, 10: PRINT "m2: motor de la espalda"
LOCATE 6, 10: PRINT "m3: motor del antebrazo"
LOCATE 4, 35: PRINT "m4: motor de la mu$eca(arriba/abajo)"
LOCATE 5, 35: PRINT "m2: motor de la mu$eca(rotaci'n)"
LOCATE 6, 35: PRINT "m3: motor de la pinza"
INI:
resaca
s:
COLOR 7, 12
LOCATE 10, 48: PRINT " "
LOCATE 10, 20: PRINT "rango de velocidad (1 a 3)"
LOCATE 11, 49: PRINT " "
LOCATE 11, 20: PRINT "seleccione motor (1 - 6) "
LOCATE 12, 20: PRINT "seleccione el mov. "
LOCATE 10, 48: INPUT s$
IF VAL(s$) < 1 OR VAL(s$) > 3 THEN GOTO s
s = VAL(s$)
m:
LOCATE 11, 48: INPUT m$
IF VAL(m$) < 1 OR VAL(m$) > 6 THEN GOTO m
m = VAL(m$)
IF m = 1 THEN
1 :
LOCATE 12, 55: PRINT " "
LOCATE 12, 40: PRINT "(-1000 a 1000) "
SHELL "pre"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m1 = conta
IF VAL(m1$) < -1000 OR VAL(m1$) > 1000 THEN GOTO 1
m1 = VAL(m1$)
LPRINT "s"; s
LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; ", "
LPRINT " "; m5; ", "; " "; m6
LPRINT "N"
m1 = 0
END IF
IF m = 2 THEN
2 :
LOCATE 12, 55: PRINT " "
LOCATE 12, 40: PRINT "(-600 a 600) "
SHELL "pre"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE

```

```

        m2 = conta
        IF m2 < -600 OR m2 > 600 THEN GOTO 2
        LPRINT "s"; s
        LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; "
        LPRINT " "; m5; ", "; " "; m6
        LPRINT "N"
        m2 = 0
END IF
IF m = 3 THEN
3 :
    LOCATE 12, 55: PRINT "
    LOCATE 12, 40: PRINT "(-500 a 500) "
    SHELL "pre"
    OPEN "pregun" FOR INPUT AS #1
    INPUT #1, conta
    CLOSE
    m3 = conta
    IF m3 < -500 OR m3 > 500 THEN GOTO 3
    LPRINT "s"; s
    LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; "
    LPRINT " "; m5; ", "; " "; m6
    LPRINT "N"
    m3 = 0
END IF
IF m = 4 THEN
4 :
    LPRINT "s"; s
    LOCATE 12, 55: PRINT "
    LOCATE 12, 40: PRINT "("; limmin4; "a"; limmax4; ")"
    SHELL "pre"
    OPEN "pregun" FOR INPUT AS #1
    INPUT #1, conta
    CLOSE
    m4 = conta
    IF m4 < -900 OR m4 > 900 THEN GOTO 4
    m5 = -m4
    LPRINT ; "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4;
    LPRINT " "; m5; ", "; " "; m6
    LPRINT "N"
END IF
IF m = 5 THEN
5 :
    LPRINT "s"; s
    LOCATE 12, 55: PRINT "
    LOCATE 12, 40: PRINT "("; limmin5; "a"; limmax5; ")"
    SHELL "pre"
    OPEN "pregun" FOR INPUT AS #1
    INPUT #1, conta
    CLOSE
    m5 = conta
    IF m5 < -1800 OR m5 > 1800 THEN GOTO 5
    m4 = m5
    LPRINT "M"; m1; ", "; " "; m2; ", "; " "; m3; ", "; " "; m4; "
    LPRINT " "; m5; ", "; " "; m6
    LPRINT "N"

```

```

END IF
IF m = 6 THEN
6 :
    LPRINT "s"; s
    LOCATE 12, 50: PRINT "
    LOCATE 12, 40: PRINT "(-1800 a 0)
    SHELL "pre"
    OPEN "pregun" FOR INPUT AS #1
    INPUT #1, conta
    CLOSE
    m6 = conta
    IF m6 < -1800 OR m6 > 0 THEN GOTO 6
    LPRINT "M"; m1; ","; " "; m2; ","; " "; m3; ","; " "; m4; "
    LPRINT " "; m5; ","; " "; m6
    LPRINT "N"
    m6 = 0
END IF
FOR r = 1 TO 4000
t = t
NEXT r
LOCATE 18, 48: PRINT "
LOCATE 18, 20: PRINT "DESEA OTRO MOVIMIENTO S / N"
LOCATE 18, 48: INPUT r$
IF r$ = "S" OR r$ = "s" THEN GOTO INI
CLS
COLOR 9, 1
menejec
END SUB

```

```

SUB movsimul
l = 76
h = 20
x = 2
y = 2
marco
COLOR 10, 12
LOCATE 3, 10: PRINT "m1: motor de la base"
LOCATE 4, 10: PRINT "m2: motor de la espalda"
LOCATE 5, 10: PRINT "m3: motor del antebrazo"
LOCATE 3, 35: PRINT "m4: motor de la mu$eca(arriba/abajo)"
LOCATE 4, 35: PRINT "m2: motor de la mu$eca(rotaci'n)"
LOCATE 5, 35: PRINT "m3: motor de la pinza"
INICI:
resaca
si:
COLOR 7, 12
LOCATE 7, 48: PRINT "
LOCATE 7, 20: PRINT "rango de velocidad (1 a 3)"
LOCATE 8, 20: PRINT "seleccione el mov. de los ejes "
LOCATE 9, 23: PRINT "
LOCATE 9, 20: PRINT "m1 ( "; limmin1; "a"; limmax1; ")"
LOCATE 10, 23: PRINT "
LOCATE 10, 20: PRINT "m2 ( "; limmin2; "a"; limmax2; ")"
LOCATE 11, 23: PRINT "
LOCATE 11, 20: PRINT "m3 ( "; limmin3; "a"; limmax3; ")"
LOCATE 12, 23: PRINT "
LOCATE 12, 20: PRINT "m4 ( "; limmin4; "a"; limmax4; ")"
LOCATE 13, 23: PRINT "
LOCATE 13, 20: PRINT "m5 ( "; limmin5; "a"; limmax5; ")"
LOCATE 14, 23: PRINT "
LOCATE 14, 20: PRINT "m6 ( "; limmin6; "a"; limmax6; ")"
LOCATE 7, 48: INPUT s$
IF VAL(s$) < 1 THEN s = 1
IF VAL(s$) > 3 THEN s = 3
LPRINT "s"; s
10 :
LOCATE 9, 45: PRINT "
col = 9
OPEN "pregun" FOR OUTPUT AS #1
WRITE #1, conta, col
CLOSE
SHELL "pro"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m1 = conta
IF m1 < limmin1 OR m1 > limmax1 THEN GOTO 10
limmin1 = limmin1 - m1
limmax1 = limmax1 - m1
20 :
LOCATE 10, 45: PRINT "
col = 10
OPEN "pregun" FOR OUTPUT AS #1

```

```

WRITE #1, conta, col
CLOSE
SHELL "pro"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m2 = conta
IF m2 < limmin2 OR m2 > limmax2 THEN GOTO 20
limmin2 = limmin2 - m2
limmax2 = limmax2 - m2
30 :
LOCATE 11, 45: PRINT "
col = 11
OPEN "pregun" FOR OUTPUT AS #1
WRITE #1, conta, col
CLOSE
SHELL "pro"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m3 = conta
IF m3 < limmin3 OR m3 > limmax3 THEN GOTO 30
limmin3 = limmin3 - m3
limmax3 = limmax3 - m3
40 :
LOCATE 12, 45: PRINT "
col = 12
OPEN "pregun" FOR OUTPUT AS #1
WRITE #1, conta, col
CLOSE
SHELL "pro"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m4 = conta
IF m4 < limmin4 OR m4 > limmax4 THEN GOTO 40
limmin4 = limmin4 - m4
limmax4 = limmax4 - m4
50 :
LOCATE 13, 45: PRINT "
col = 13
OPEN "pregun" FOR OUTPUT AS #1
WRITE #1, conta, col
CLOSE
SHELL "pro"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m5 = conta
IF m5 < limmin5 OR m5 > limmax5 THEN GOTO 50
limmin5 = limmin5 - m5
limmax5 = limmax5 - m5
60 :
LOCATE 14, 45: PRINT "
col = 14

```

```

OPEN "pregun" FOR OUTPUT AS #1
WRITE #1, conta, col
CLOSE
SHELL "pro"
OPEN "pregun" FOR INPUT AS #1
INPUT #1, conta
CLOSE
m6 = conta
IF m6 < limmin6 OR m6 > limmax6 THEN GOTO 60
limmin6 = limmin6 - m6
limmax6 = limmax6 - m6
t = m4
m4 = m4 + m5
m5 = t - m5
LPRINT "M"; m1; ", "; m2; ", "; m3; ", "; m4; ", ";
LOCATE 18, 48: PRINT "
LOCATE 18, 20: PRINT "DESEA OTRO MOVIMIENTO S / N"
LOCATE 18, 48: INPUT r$
IF r$ = "S" OR r$ = "s" THEN GOTO INICI
LPRINT "N"
set
CLS
COLOR 9, 1
menejec
END SUB

```

```
SUB resaca
m1 = 0: m2 = 0: m3 = 0: m4 = 0: m5 = 0: m6 = 0
END SUB
```

```
SUB resetear
FOR i = 2 TO 21
FOR r = 1 TO 80
LOCATE i, r: PRINT " "
NEXT r
NEXT i
END SUB
```

```
SUB resetear1
COLOR 7, 12
LOCATE 7, 20: PRINT " "
LOCATE 7, 21: PRINT " "
LOCATE 7, 22: PRINT " "
FOR i = 10 TO 19
FOR r = 10 TO 62
LOCATE i, r: PRINT " "
NEXT r
NEXT i
FOR t = 1 TO 5
arch$(t) = " "
NEXT t
END SUB
```

```
SUB resetear2
COLOR 7, 12
FOR w = 23 TO 52
LOCATE 21, w: PRINT " "
LOCATE 22, w: PRINT " "
NEXT w
COLOR 7, 12
END SUB
```

```
SUB resetr
m11 = 0: m21 = 0: m31 = 0: m41 = 0: m51 = 0: m61 = 0
m12 = 0: m22 = 0: m32 = 0: m42 = 0: m52 = 0: m62 = 0
m13 = 0: m23 = 0: m33 = 0: m43 = 0: m53 = 0: m63 = 0
m14 = 0: m24 = 0: m34 = 0: m44 = 0: m54 = 0: m64 = 0
END SUB
```

```
SUB restear2
COLOR 7, 12
FOR g = 9 TO 21
FOR t = 10 TO 70
LOCATE g, t: PRINT " "
NEXT t
NEXT g
END SUB
```

```
SUB sombra
COLOR 0, 7
FOR i = 1 TO h + 1
  LOCATE x + i, y + 1 + 2: PRINT CHR$(177);
NEXT i
LOCATE x + h + 1, y + 1: PRINT STRING$(1 + 1, CHR$(177));
END SUB
```

```
SUB tutorial
control10 = 1
control11 = 1
menu
desplazar = 5
arri = 0
SHELL "ppal"
END SUB
```

SUB set
limmin1 = 1000
limmax1 = 1000
limmin2 = -600
limmax2 = 600
limmin3 = -500
limmax3 = 500
limmin6 = -1800
limmax6 = 0
limmin4 = -900
limmax4 = 900
limmin5 = -1800
limmax5 = 1800
limmin11 = 1000
limmax11 = 1000
limmin21 = -600
limmax21 = 600
limmin31 = -500
limmax31 = 500
limmin61 = -1800
limmax61 = 0
limmin41 = 900
limmax41 = 900
limmin51 = -1800
limmax51 = 1800
limmin12 = 1000
limmax12 = 1000
limmin22 = -600
limmax22 = 600
limmin32 = -500
limmax32 = 500
limmin62 = -1800
limmax62 = 0
limmin42 = -900
limmax42 = 900
limmin52 = -1800
limmax52 = 1800
limmin13 = 1000
limmax13 = 1000
limmin23 = -600
limmax23 = 600
limmin33 = 500
limmax33 = 500
limmin63 = -1800
limmax63 = 0
limmin43 = -900
limmax43 = 900
limmin53 = -1800
limmax53 = 1800
limmin14 = -1000
limmax14 = 1000
limmin24 = -600
limmax24 = 600
limmin34 = -500

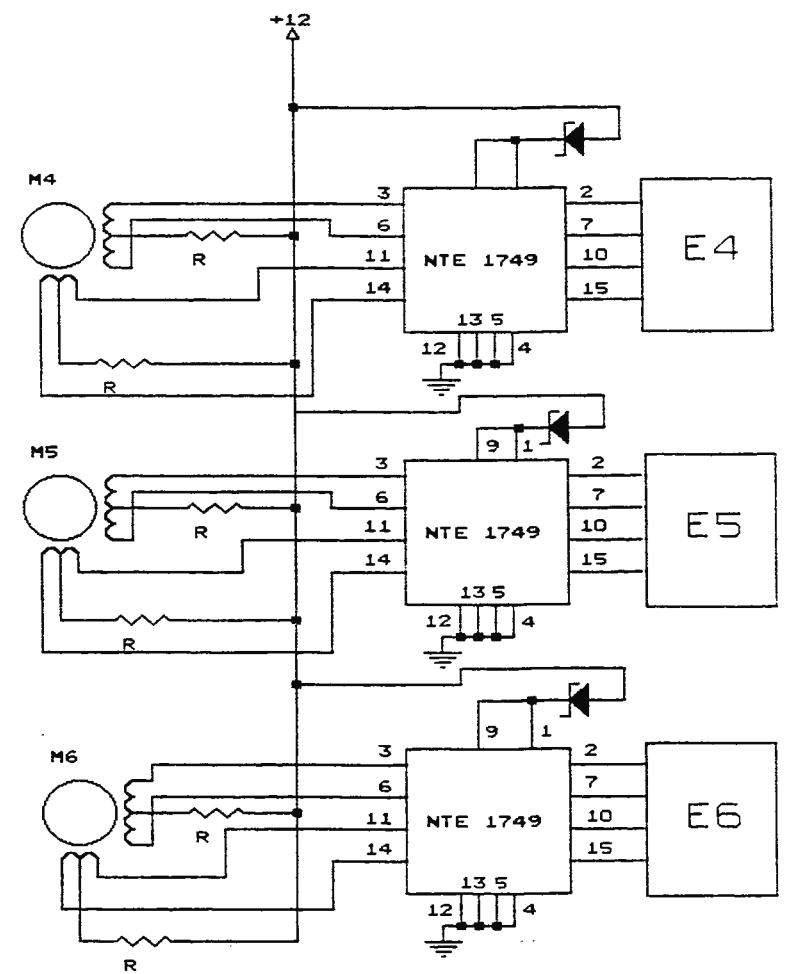
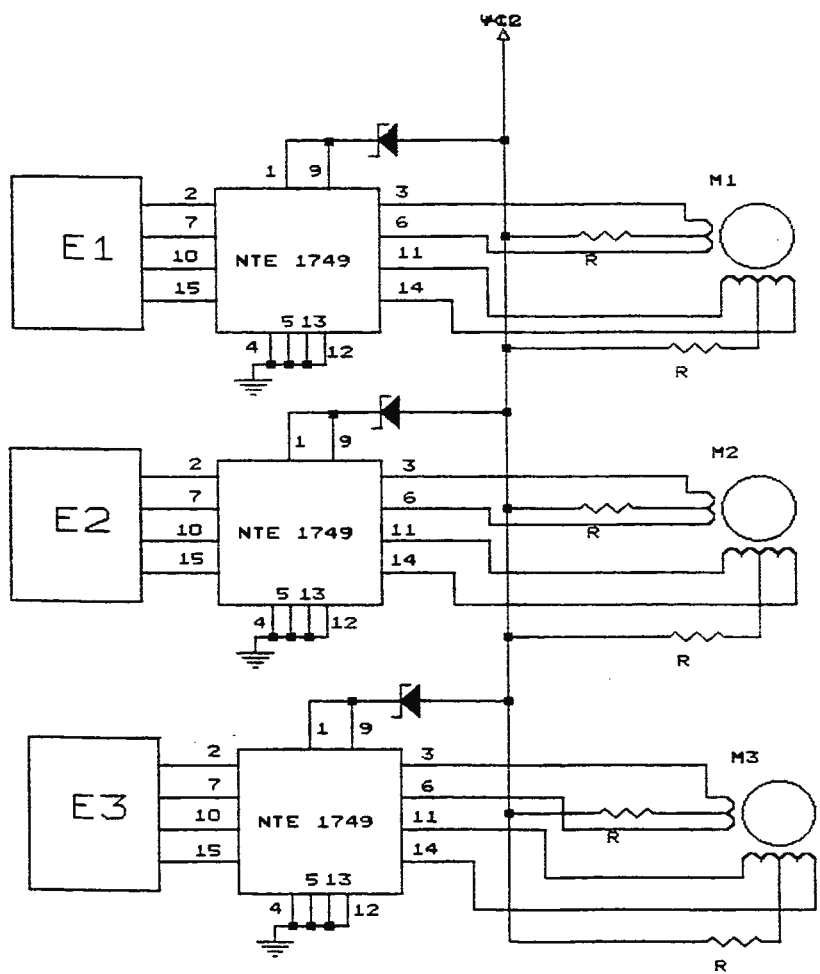
```
limmax34 = 500
limmin64 = 1800
limmax64 = 0
limmin44 = 900
limmax44 = 900
limmin54 = -1800
limmax54 = 1800
END SUB
```

```
SUB ver
COLOR 7, 12
z = 7
v = 8
COLOR z, v
x = 4
y = 23
l = 28
h = 16
marco
COLOR z, v
LOCATE 5, 26: PRINT "      Controlador      "
LOCATE 6, 26: PRINT "      por computadora  "
LOCATE 7, 26: PRINT "      para            "
LOCATE 8, 26: PRINT "      brazo robot     "
LOCATE 11, 26: PRINT STRING$(26, CHR$(178))
x = 12: y = 25: l = 24: h = 7: COLOR z, v: marco
LOCATE 13, 26: PRINT "      Elaborado por:"
LOCATE 14, 26: PRINT "      xxxxxxxxxxxxxx"
LOCATE 15, 26: PRINT "      xxxxxxxxxxxxxx"
LOCATE 16, 26: PRINT "      xxxxxxxxxxxxxx"
LOCATE 17, 26: PRINT "      xxxxxxxxxxxxxx"
LOCATE 18, 26: PRINT "      xxxxxxxxxxxxxx"
LOCATE 10, 25: PRINT "      << 1997 >>      ";
COLOR 7, 12
END SUB
```

ANEXO C

DIAGRAMAS ESQUEMÁTICOS Y DE

FLUJO



ETAPA DE POTENCIA

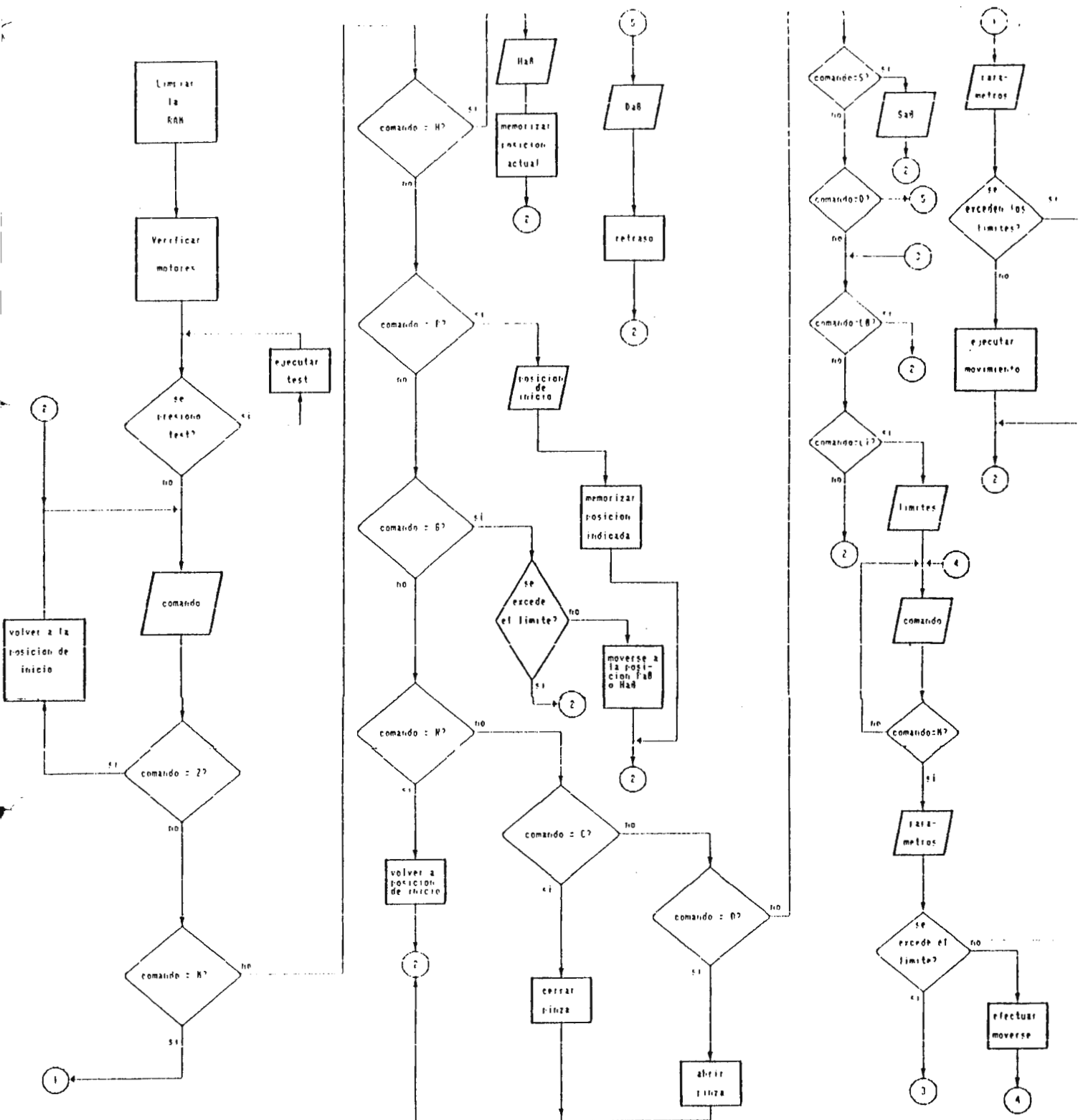


Diagrama de flujo del programa monitor del brazo robot.

ANEXO D

GLOSARIO

GLOSARIO

-Accionamiento:

Puesta en marcha, poner a funcionar el sistema.

-Articulaciones:

Uniones entre dos o más piezas.

-Automático:

Ejecución sin intervención humana de ciertas labores industriales, administrativas o científicas.

-Capacidad de carga

Es el peso, generalmente en kilogramos, que el robot puede manipular, ello excluye el peso propio del robot.

- Efecto:

Elemento que realiza la acción de sujetar, pintar, soldar o rotar en un robot y es este el que debe posicionarse en el espacio.

-Estator:

Parte fija de una máquina rotatoria o motor.

-Grados de libertad

Es el número de parámetros que es preciso conocer para determinar la posición del robot, o dicho de otra forma, los movimientos básicos independientes que posicionan a los elementos de un robot en el espacio. El número máximo de grados de libertad para un robot es de seis; tres para definir posición y tres para orientar la mano de sujeción.

-Manipulador:

Sistema que controla o maneja un elemento o parte.

-Microprocesador:

Dispositivo que puede buscar instrucciones de una memoria, las decodifica y las ejecuta, realiza operaciones aritméticas y lógicas, acepta datos de dispositivos de entrada y envía resultados a dispositivos de salida.

-Miniaturizado:

Objeto del tamaño más pequeño posible.

-Periféricos:

Elemento de un sistema computacional que no es la unidad central de proceso y que sirve esencialmente para comunicarse con el exterior.

-Precisión:

Es la tolerancia de medida o de transmisión del instrumento y define los límites de los errores cometidos cuando el instrumento opera en condiciones normales.

-Puerto:

Palabra usada comúnmente en electrónica, la cual se usa para definir un espacio físico y/o lógico, en el cual la computadora puede colocar datos y recibirlos hacia o desde dispositivos externos.

-Resolución:

Se define como la menor variación que puede ocurrir en una variable de salida, como resultado de un cambio en una variable de entrada.

-Robots:

Aparato capaz de realizar de manera automática diversas operaciones.

-Rotor:

Parte móvil de un motor, generador eléctrico, turbina, etc.

-Rutina:

Parte de un programa de informática que puede ser utilizada por otros programas.

-Secuencial:

Procesamiento en el que cada operación precede a otra y sigue a otra, sin que nunca dos de entre ellas sean simultáneas.

-Servocontrol:

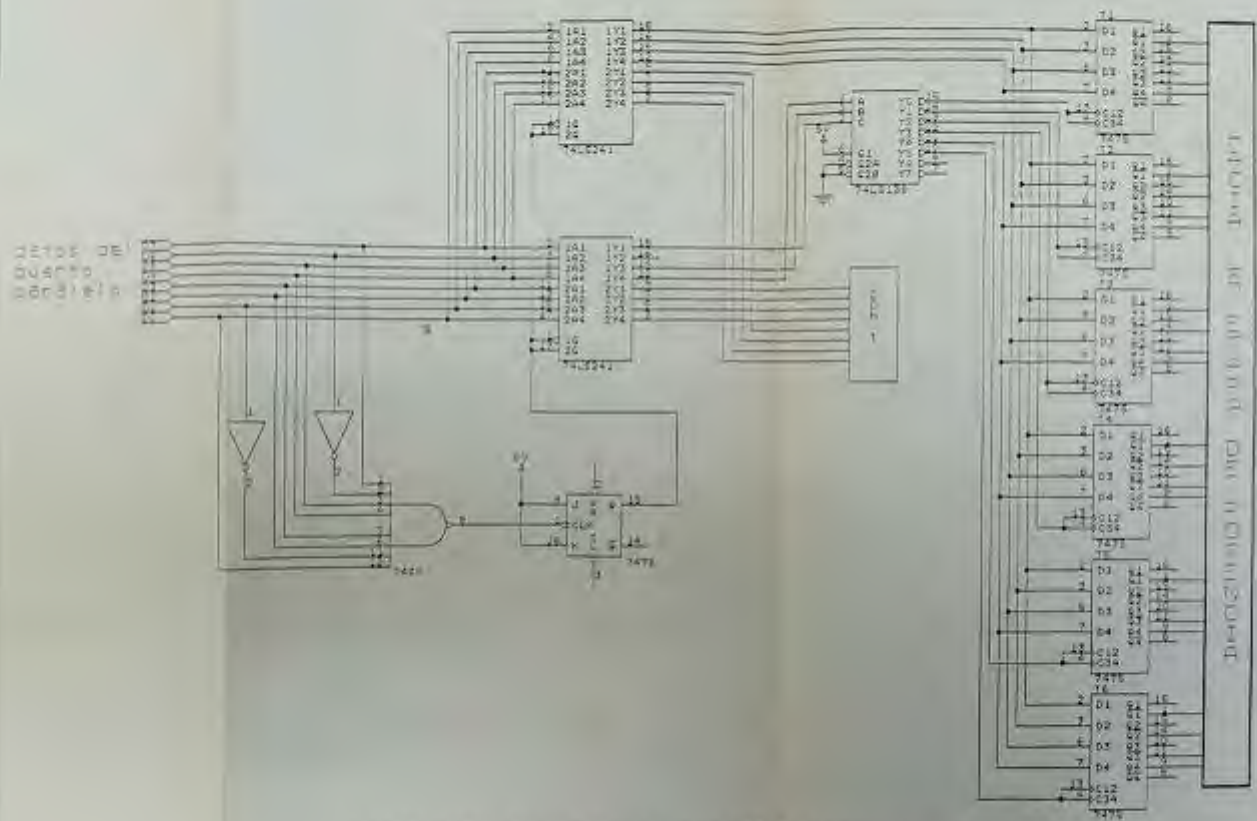
Control automático de los mandos de un aparato.

-Sistemas de coordenadas

Ello se refiere a los diferentes tipos de coordenadas de las cuales se valen los diseñadores de robots, para definir la posición del efector en el espacio, estas coordenadas pueden ser: Cartesianas, Cilíndricas o Esféricas.

-Sujeción:

Acción de sujetar o mantener.



Circuito decodificador para seleccionar control directo de la computadora Z80