

ESCUELA POLITÉCNICA DEL EJÉRCITO

**DEPARTAMENTO DE ELÉCTRICA Y
ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN
DEL TÍTULO DE INGENIERÍA**

**DISEÑO Y CONSTRUCCIÓN DE UN ROBOT
PARA IMPRESIÓN DE CIRCUITOS ELECTRÓNICOS**

BLADIMIR ROBERTO HARNISTH PINOS

SANGOLQUÍ – ECUADOR

2007

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado en su totalidad por el Sr. Roberto Bladimir Harnisth Pinos bajo nuestra dirección.

Ing. Hugo Ortiz
DIRECTOR

Ing. Evelio Granizo
CO DIRECTOR

RESUMEN

El presente proyecto tiene por objetivo construir un robot que imprima sobre baquelita circuitos de electricidad y electrónica básica, para lo cual utiliza como herramienta un dispensador de tinta indeleble. El proceso puede ser realizado manualmente por medio de los botones de su panel de control, o desde un computador mediante transmisión serial. Por último, la placa impresa debe ser sometida a un proceso químico conocido como “atacado con ácido”, en el que se elimina todo el cobre que no esté protegido por la tinta, dejando como resultado únicamente las pistas del circuito. Su mayor ventaja es guardar en memoria el diseño realizado, lo que permite repetir la misma impresión las veces que se desee.

El robot construido posee dos motores que permiten los desplazamientos a lo largo y ancho sobre el plano horizontal y uno que eleva o desciende la herramienta. Está conformado por un panel de control, una pantalla LCD que despliega opciones o el estado del robot, sensores fin de carrera que definen el área de trabajo, puerto de comunicación serial y alarmas auditivas y visuales. Para el control computarizado utiliza un programa gráfico, desarrollado en Visual Basic, que permite realizar el diseño de los circuitos en pantalla antes de enviarlos a imprimir. Su cerebro es un microcontrolador PIC16F887A programado en PICBASIC Pro, este controla todo en cualquiera de los modos que se encuentre.

DEDICATORIA

A Nelly Beatriz, mi madre, que siempre ha estado a mi lado, a mis tres hermanos: Antonio, Dayanara y Paúl, y a Carolina, mi novia y compañera incondicional. Ellos han sido mi apoyo, no solo durante la realización de este proyecto, sino durante toda mi carrera profesional y mi vida en general.

AGRADECIMIENTO

A Dios, por haberme llenado de herramientas para caminar en su mundo, a mi madre, por no fallarme con su apoyo físico y moral desde el día en que me dio la vida hasta este mismo instante, y a la Escuela Politécnica del Ejército, institución que junto a todos los profesores que tuve durante mi vida universitaria, me permitieron adquirir los conocimientos necesarios para culminar la carrera de Ingeniería Electrónica, especialmente los Ingenieros: Hugo Ortiz y Evelio Granizo, que con gran acierto me supieron guiar en la realización de este proyecto.

PROLOGO

El presente proyecto tiene como objetivo construir un robot cartesiano electromecánico de tres ejes (X,Y,Z). Poseerá tres motores de pasos que se encargarán de movilizar una herramienta de trabajo sobre dichos ejes. En vista de que se ha planteado como aplicación imprimir circuitos básicos de electricidad y electrónica en baquelita, se instalará como herramienta un dispensador de tinta indeleble y la superficie de trabajo será una baquelita virgen. Los motores se encargarán de elevar o descender el dispensador de tinta y llevarlo a lo largo y ancho del área de trabajo, dejando plasmados los circuitos requeridos. La baquelita impresa quedará lista para ser utilizada después de haber sido sometida a un proceso químico conocido como “atacado con acido” en donde se desvanece el cobre que no este marcado con tinta, dejando únicamente el circuito diseñado.

El robot podrá ser utilizado de forma manual o por medio de un computador. Para el modo manual se utilizarán los botones de dirección instalados en su panel de control y para el modo computarizado se recurrirá a un software específico, creado para diseñar circuitos en pantalla y enviarlos a imprimir en el robot por medio de comunicación serial.

El control del robot en cualquiera de sus modos estará a cargo de un microcontrolador PIC16F877A.

A continuación se detallan los siete capítulos que conforman el presente proyecto:

En el **CAPÍTULO 1** se presenta la justificación e importancia del proyecto y se explica sintéticamente el funcionamiento y alcance que tiene el robot.

En el **CAPÍTULO 2** se detallan los aspectos teóricos de los dispositivos y materiales que utiliza el robot, así por ejemplo para los motores paso a paso se describe el

funcionamiento, tipos, especificaciones, etc., y para los microcontroladores se describen modelos, características, configuraciones, principio de funcionamiento, etc.

En el **CAPÍTULO 3** se detalla toda la información referente al diseño del proyecto: técnica que se utilizó, materiales empleados, las fuentes de energía que alimentan el proyecto, configuración mecánica y electrónica, flexibilidad que brinda el proyecto e incluso las dimensiones del robot.

En el **CAPÍTULO 4** se describe el desarrollo del software que utiliza el robot, se detalla el diseño y desarrollo del programa del microcontrolador así como también detalla el programa gráfico que permite el control del robot desde un computador, el cual es sencillo y amigable.

En el **CAPÍTULO 5** se presenta la implementación del robot con el ensamblaje de cada parte, tanto en lo mecánico como en lo electrónico.

En el **CAPÍTULO 6** se presenta los resultados de las pruebas a las que se sometió el robot para llegar a un óptimo funcionamiento.

En el **CAPÍTULO 7** se expone las conclusiones sobre el desarrollo e implementación del proyecto y varias recomendaciones sobre las limitaciones, ventajas y desventajas de este proyecto.

ÍNDICE DE CONTENIDO

CONTENIDO	PÁG.
CAPÍTULO 1	
INTRODUCCIÓN	
1.1. ANTECEDENTES	1
1.2. JUSTIFICACIÓN E IMPORTANCIA.....	2
1.3. DESCRIPCIÓN GENERAL DEL PROYECTO	3
CAPÍTULO 2	
FUNDAMENTOS	
2.1. INTRODUCCIÓN A LA ROBÓTICA INDUSTRIAL	5
2.1.1. Robot Industrial	6
2.1.2. Robot Cartesiano	7
2.2. MOTORES PASO A PASO.....	8
2.2.1. Descripción general de los tipos de motores paso a paso.....	10
2.2.2. Motores paso a paso de magneto permanente	12
2.2.2.1. Principio de funcionamiento.....	12
2.2.2.2. Tipos de Motor PAP magneto permanente	13
2.2.2.3. Manejo de los Motores PAP magneto permanente	16
2.2.2.4. Secuencias para manejar motores paso a paso de magneto perramente.....	17
2.3. MICROCONTROLADORES	20
2.3.1. Microcontrolador PIC 16F877A.....	20
2.3.1.1. Configuración y descripción de los pines.....	22
2.3.1.2. Comparación con otros modelos de PIC	26

2.4. LCD	28
2.4.1. Principio de funcionamiento del LCD.....	29
2.4.2. Modos de visualización de un LCD	30
2.4.3. Características principales del LCD	30
2.4.4. Tipos de LCD	31
2.5. SENSORES FIN DE CARRERA.....	33

CAPÍTULO 3

DISEÑO DEL ROBOT

3.1. ESTRUCTURA Y DISEÑO MECÁNICO	35
3.1.1. Plataforma de trabajo.....	35
3.1.2. El eje Y	35
3.1.3. El eje X.....	37
3.1.4. El eje Z	38
3.2. DISEÑO DEL HARDWARE DEL ROBOT	40
3.2.1. Etapa de Control	41
3.2.1.1. Conexión de los dispositivos electrónicos al microcontrolador	43
3.2.2. Etapa de potencia y alimentación del robot.....	48

CAPÍTULO 4

PROGRAMACIÓN DEL MICROCONTROLADOR DEL ROBOT

4.1. INTRODUCCIÓN.....	49
4.2. PROGRAMA DEL MICROCONTROLADOR	49
4.2.1. Programa Principal	51
4.2.2. Programa para el Control Manual	52
4.2.3. Programa para el Control Computarizado	54
4.2.3.1. Programa diseñado para el uso del robot mediante computadora	54
4.2.3.2. Funcionamiento del Programa para computador.....	55
4.2.4. Programa para la opción IMPRIMIR MEMORIA	56
4.2.5. Programa para la opción FIGURAS	56

CAPÍTULO 5

IMPLEMENTACIÓN DEL ROBOT

5.1. MESA DE TRABAJO.....	58
---------------------------	----

5.2. EJE Y	58
5.3. EJE X.....	65
5.4. EJE Z	67
5.5. SENSORES	70
5.6. SUJECIÓN DE LA BAQUELITA.....	71
5.7. CAJA PRINCIPAL	71
5.8. PLACA ELECTRÓNICA DEL ROBOT	72
5.9. PANEL DE CONTROL Y PANTALLA LCD	74
5.10. PUERTO DE COMUNICACIÓN SERIAL.....	75
5.11. SWITCH DE ENCENDIDO (ON / OFF)	76

CAPÍTULO 6

PRUEBAS Y RESULTADOS

6.1. PRUEBAS EN MODO MANUAL	77
6.1.1. Desplazamiento sobre el eje x	77
6.1.2. Desplazamiento sobre el eje y	78
6.1.3. Desplazamiento sobre el eje z	78
6.2. PRUEBAS EN MODO COMPUTARIZADO	79
6.3. PRUEBAS EN MODO IMPRIMIR MEMORIA.....	81
6.4. PRUEBAS EN MODO FIGURAS	82

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES.....	83
7.2. RECOMENDACIONES	84

ÍNDICE DE TABLAS

Tabla. 2.1. Secuencia para controlar Motor PAP Bipolar	17
Tabla. 2.2. Secuencia para controla Motor PAP Unipolar – Normal	17
Tabla. 2.3. Secuencia para controlar Motor PAP Unipolar – Wave Drive.....	18
Tabla. 2.4. Secuencia para controlar Motor PAP Unipolar – Medio Paso	18
Tabla. 2.5. Características del Microcontrolador PIC 16F877A	21
Tabla. 2.6. Descripción de pines del PIC 16F877A	23
Tabla. 2.7. Comparación de modelos PIC de la familia 16F8XX.....	27
Tabla. 2.8. Descripción de los pines del LCD	32
Tabla. 3.1. Configuración de pines del microcontrolador PIC 16F877A para el proyecto	41
Tabla. 3.2. Botones del Panel de control	47
Tabla. 4.1. Botones del Panel de Control dentro del menú Control Manual.....	53

ÍNDICE DE FIGURAS

Figura. 1.1. Diagrama General del Prototipo	4
Figura. 2.1. Motores paso a paso	9
Figura. 2.2. Vista de sección de un motor por pasos de reluctancia variable.....	10
Figura. 2.3. Vista en sección de un magneto permanente	11
Figura. 2.4. Vista expandida ilustrativa del desplazamiento de dientes	11
Figura. 2.5. Imagen del rotor	12
Figura. 2.6. Imagen de un estator de 4 bobinas	12
Figura. 2.7. Motor PAP Bipolar	13
Figura. 2.8. Puente en H (H-Bridge) para control de cada bobina del motor.....	14
Figura. 2.9. Circuito controlador mediante H-Bridge integrados L293	14
Figura. 2.10. Motor PAP Unipolar	15
Figura. 2.11. Circuito controlador de Motor PAP unipolar mediante un ULN2803.....	16
Figura. 2.12. Diagrama de bloques de un sistema con motor paso a paso	16
Figura. 2.13. PIC 16F877A	21
Figura. 2.14. Configuración de pines del PIC 16F877A	22
Figura. 2.15. LCD de dos filas y treinta y dos segmentos.....	28
Figura. 2.16. Modos de Visualización de un LCD	30
Figura. 2.17. Sensor fin de carrera	33
Figura. 2.18. Sensor fin de Carrera Normalmente Abierto (NO).....	34
Figura. 3.1. Esquema de la plataforma de trabajo	36
Figura. 3.2. Esquema de los ejes Y	36
Figura. 3.3. Esquema de los ejes Y con las piezas móviles (soportes).....	37
Figura. 3.4. Esquema de eje X (Plataforma)	38
Figura. 3.5. Esquema del eje Z	38

Figura. 3.6. Abrazadera que sostendrá la herramienta de trabajo.....	39
Figura. 3.7. Esquema General del Robot.....	39
Figura. 3.8. Esquema del Hardware Electrónico del Robot	40
Figura. 3.9. Configuración del microcontrolador PIC16F877A.....	43
Figura. 3.10. Conexión del LCD	44
Figura. 3.11. Conexión de la alarma auditiva y visual	45
Figura. 3.12. Conexión de los sensores fin de carrera.....	45
Figura. 3.13. Conexión del microcontrolador al puerto serial de un computador.....	46
Figura. 3.14. Conexión de los botones del panel de control.....	47
Figura. 3.15. Configuración de los motores del robot.....	48
Figura. 4.1. Diagrama de flujo del Programa Principal.....	51
Figura. 4.2. Pantalla principal del programa para Control Computarizado.....	54
Figura. 5.1. Ejes Y con respectivas piezas plásticas y sus bocines	59
Figura. 5.2. Eje horizontal con piñones	59
Figura. 5.3. Eje horizontal desmontado de la pieza base que lo permitirá girar.....	60
Figura. 5.4. Eje Auxiliar Horizontal	60
Figura. 5.5. Ejes Y paralelos y eje Auxiliar horizontal que gira	61
Figura. 5.6. Bandas Transportadoras	61
Figura. 5.7. Motor Y engranado al eje horizontal	62
Figura. 5.8. Eje Y con motor	63
Figura. 5.9. Eje Y sin motor	63
Figura. 5.10. Ejes Y (Vista Lateral).....	64
Figura. 5.11. Ejes Y (Vista Posterior)	64
Figura. 5.12. Ejes Y (Vista Frontal)	64
Figura. 5.13. Eje X (plataforma).....	65
Figura. 5.14. Vista frontal de la plataforma (Vista del Motor X).....	66
Figura. 5.15. Banda templada en el Eje X.....	66
Figura. 5.16. Eje Z.....	67
Figura. 5.17. Motor Z	68
Figura. 5.18. Abrazadera metálica.....	68
Figura. 5.19. Eje Z (Vista lateral).....	69
Figura. 5.20. Eje Z (Vista Superior)	69

Figura. 5.21. Sensores fin de carrera	70
Figura. 5.22. Marco que sujeta la baquelita.....	71
Figura. 5.23. Caja Principal	72
Figura. 5.24. Circuito Impreso	73
Figura. 5.25. Placa electrónica (Vista frontal).....	73
Figura. 5.26. Placa electrónica (Vista trasera).....	74
Figura. 5.27. Placa electrónica dentro de la Caja Principal.....	74
Figura. 5.28. LCD y Panel de Botones (Vista interna).....	75
Figura. 5.29. Panel de Botones	75
Figura. 6.1. Trabajo del Robot en Modo Manual	79
Figura. 6.2. Circuito Diseñado en el Computador.....	80
Figura. 6.3. Circuito Impreso desde el Computador	81
Figura. 6.4. Impresión en Modo Figuras	82

CAPÍTULO 1

INTRODUCCIÓN

1.1. ANTECEDENTES

La tecnología avanza de manera sorprendente, al punto que las personas están rodeadas de aparatos electrónicos: en el hogar, dentro de los autos, en los juguetes de los niños, etc. Todos estos aparatos poseen una placa electrónica que a su vez han sido creadas por otros equipos electrónicos de la industria, generalmente robots industriales, en los que se destacan los robots cartesianos.

Hablando específicamente de las placas electrónicas, en nuestro país, debido a la escasa tecnología, no se ha podido reproducir ni crear alguno de estos mencionados aparatos, limitándose simplemente a manejarlos, configurarlos y programarlos, y en el mejor de los casos darles mantenimiento o cambiar una pieza dañada. Sin embargo, los estudiantes y aficionados a la electrónica crean sus pequeños aparatos electrónicos, con un diseño no muy sofisticado pero si muy respetable en nuestro medio.

Por lo que en el presente proyecto de grado, se propone un robot cartesiano (X,Y,Z), que con la ayuda de un dispensador de tinta indeleble instalado en su eje Z, permitirá dejar plasmados circuitos básicos de electricidad y electrónica en una placa o baquelita, la cual después de ser sometida a un proceso químico conocido, estará lista para ser utilizada.

1.2. JUSTIFICACIÓN E IMPORTANCIA

La principal justificación para el desarrollo de este proyecto radica en que hoy en día todo proceso ya sea doméstico, profesional o industrial está tendiendo a ser automatizado por máquinas electrónicas, razón por la cual la industria electrónica esta creciendo en todos los rincones del mundo y nosotros no podemos ser la excepción. El primer paso para la creación de cualquier aparato electrónico es el diseño del circuito y la implementación del mismo en una placa, lamentablemente la principal desventaja en nuestro medio es que no existe la tecnología para que los diseños de circuitos electrónicos y eléctricos sean impresos directamente en una placa, motivo por el cual, antes que recurrir a una empresa extranjera se opta por realizar las placas manualmente; cabe recalcar que a pesar de ser lo más meticuloso las placas hechas a mano no siempre quedan bien. Además, si se desea realizar varias placas de las mismas, se tiene que partir todo el proceso desde cero para cada placa, lo que no garantiza que todas ellas se obtengan idénticas. Por lo tanto, con la ayuda de un robot que realice circuitos impresos se obtendrá las siguientes ventajas: reducción de tiempo, ahorro de dinero y un importante aumento de calidad, porque al ingresar el diseño del circuito a la memoria de la máquina se podrá imprimir las veces que se desee, realizar cambios en los diseños, y además se podrá definir tareas para que trabaje sola.

La importancia del proyecto radica en que se está logrando introducir una nueva herramienta tecnológica para nuestro medio, porque una herramienta como ésta permitirá mejorar notablemente la calidad de los pequeños proyectos de electrónica, y además permitirá la proyección a la producción en serie.

Por último, con el desarrollo de este proyecto se determinará que podemos diseñar y construir robots con similares características o funciones parecidas a los existentes fuera del país, estableciéndose de esta manera un avance tecnológico muy notable en nuestra industria electrónica.

1.3. DESCRIPCIÓN GENERAL DEL PROYECTO

En el presente proyecto se desarrolla un robot cartesiano electromecánico que permite la impresión de circuitos de electricidad y electrónica básica, el cual está compuesto de una herramienta de impresión, tres ejes, sensores de fin de carrera, motores paso a paso, etapa electrónica, panel de control y etapa de potencia, todo instalado en un tablero. Además, posee un puerto COM de comunicación y un programa específico para su utilización mediante computadora. A continuación se describen dichas partes:

- **La herramienta de impresión** que se utiliza en este proyecto es un marcador de tinta indeleble, el cual podrá ser removido de forma fácil las veces que así se requiera.
- **Los tres ejes** permiten cumplir con el objetivo de desplazar la herramienta de impresión sobre todo el plano, de la siguiente manera: el dispensador de tinta sujeto al eje Z, describe un movimiento vertical, de tal manera que se eleva o desciende sobre el plano, a su vez, el dispensador de tinta se desplaza de derecha a izquierda en el plano XY.
- **Los sensores de fin de carrera** son dispositivos que se activan por medio de contacto, que emiten una señal programable a conveniencia, en este caso los sensores son colocados en los extremos de cada eje y programados para que los motores paren al sobrepasar los límites de los ejes X e Y, con esto se establece el área de trabajo del robot.
- **Los motores paso a paso** son los encargados de lograr el movimiento del dispensador de impresión sobre el plano, para lo cual se utiliza uno en cada eje, es decir, un motor hace que se eleve o descienda el dispensador y otros dos permiten el desplazamiento de derecha a izquierda, respectivamente. Se elige este tipo de motores por dos razones principales: brindan una alta precisión y es fácil su control y programación.
- **La etapa electrónica** es sin duda la parte mas importante del robot pues aquí se halla su cerebro, el microcontrolador PIC16F877A, éste se encuentra interconectado a los motores, los sensores, los botones del panel de control, el puerto de comunicación con la computadora y el LCD que despliega información. El microcontrolador mediante un programa previamente diseñado y cargado en su

memoria, procesa todas las señales recibidas y ejecuta o no los movimientos del robot.

- **El panel de control** está formado por varios botones con los que se maneja el robot y un LCD que despliega las distintas opciones de trabajo.
- **La etapa de potencia** es indispensable, pues la débil señal que envía el microcontrolador no basta para que trabajen los motores, de modo que se hace necesario procesar dicha señal en esta etapa, logrando que se amplifique en voltaje y corriente suficientes para que funciones los motores.
- **El programa específico** que se menciona como parte del proyecto, es un software creado bajo Visual Basic, que permite por medio de un puerto de comunicación serial, controlar el robot desde un computador que tenga cargado dicho programa.

A continuación se presenta la Figura. 1.1. que muestra un diagrama general de cómo estará estructurado el prototipo.

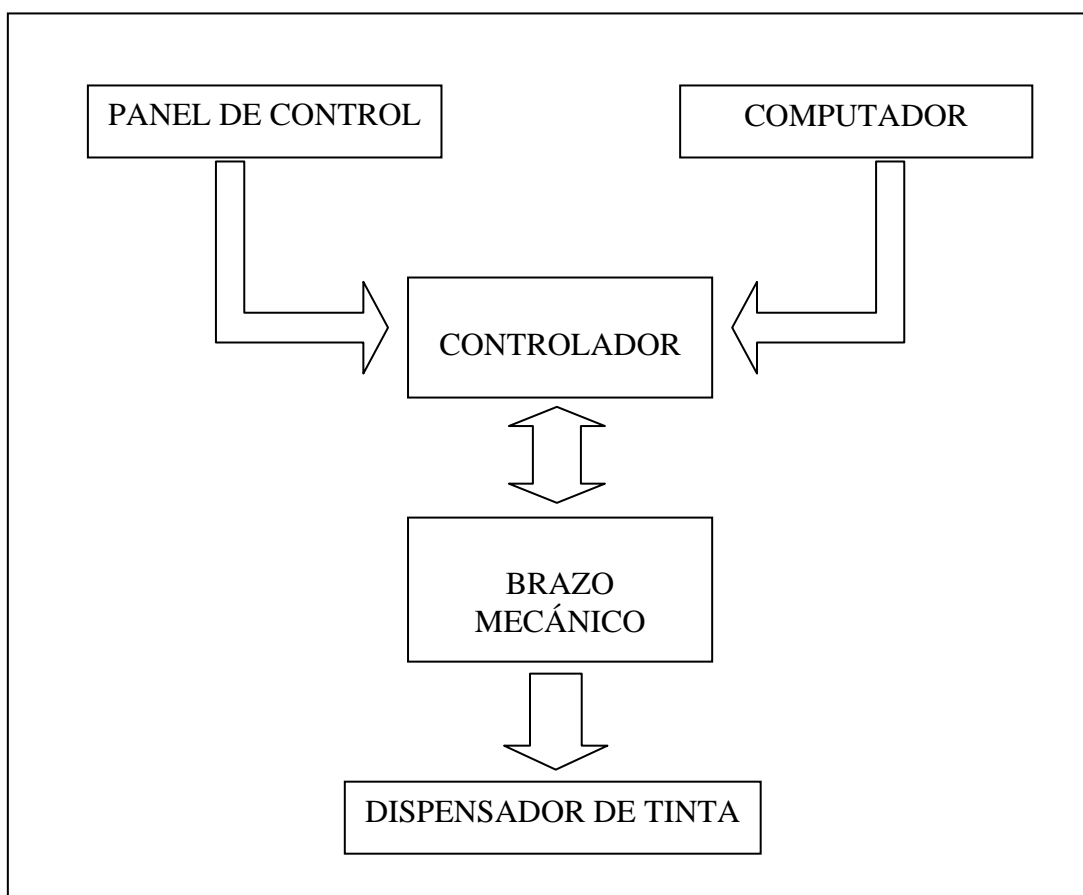


Figura. 1.1. Diagrama General del Prototipo

CAPÍTULO 2

FUNDAMENTOS

En este capítulo se expondrá de manera general los fundamentos básicos de los dispositivos y elementos electrónicos más importantes que se utilizan en el ensamblaje del robot.

2.1. INTRODUCCIÓN A LA ROBÓTICA INDUSTRIAL

Los robots han dejado de ser máquinas misteriosas propias de la ciencia ficción, para constituirse en la actualidad elementos indispensables en una gran parte de los procesos de manufactura.

Existen una amplia variedad de tipos de robots, que cumplen con el concepto general de ser manipuladores programables que pueden detenerse con precisión y repetibilidad en cualquier punto de su área de trabajo en base a un programa.

Las principales características en base a las cuales se clasifican los robots son las siguientes:

- Propósito o función
- Sistema de coordenadas empleado
- Número de grados de libertad
- Sistema de control

Los robots de acuerdo a su aplicación se clasifican en:

- Impulsados neumáticamente
- Equipados con servomecanismos
- Punto a punto
- Controlados por computadora
- Con capacidades sensoriales
- Industriales

2.1.1. Robot Industrial

Sin duda el Robot Industrial brinda las mayores aplicaciones, puesto que es un dispositivo multifuncional, al contrario de la máquina automática clásica que es fabricada para realizar de forma repetitiva un tipo determinado de operaciones, por lo tanto, el robot industrial es diseñado en función de diversos movimientos que debe ejecutar; es decir, sus grados de libertad, su campo de trabajo y su comportamiento estático y dinámico.

La capacidad del robot industrial para reconfigurar su ciclo de trabajo, unida a la versatilidad y variedad de sus elementos terminales (pinzas, garras, herramientas, etc.), le permite adaptarse fácilmente a la evolución o cambio de los procesos de producción.

Los robots industriales están disponibles en una amplia gama de tamaños, formas y configuraciones físicas, por lo que la gran mayoría de los robots comercialmente disponibles en la actualidad tienen una de las siguientes cuatro configuraciones básicas:

- **La configuración polar.** Utiliza coordenadas polares para especificar la posición del robot en términos de una rotación sobre su base, un ángulo de elevación y una extensión lineal del brazo.
- **La configuración cilíndrica.** Sustituye el movimiento lineal por el rotacional sobre su base, obteniendo un medio de trabajo en forma de cilindro.

- **La configuración de brazo articulado.** Utiliza únicamente articulaciones rotacionales para conseguir cualquier posición del robot, por lo que es más versátil que las anteriores configuraciones.
- **La configuración de coordenadas cartesianas.** Posee tres movimientos lineales de acuerdo con las coordenadas cartesianas, de donde proviene su nombre, las cuales son más adecuadas para describir la posición y movimiento del brazo. Los robots cartesianos a veces reciben el nombre de XYZ, donde las letras representan a los tres ejes del movimiento.

2.1.2. Robot Cartesiano

El robot cartesiano, dentro de la clasificación general de los robots, es un robot industrial, que nace de la unión de una estructura mecánica articulada y de un sistema electrónico de control en el que se integra una computadora. Esto permite la programación y control de los movimientos a efectuar por el robot y la memorización de las diversas secuencias de trabajo, por lo que le da al robot una gran flexibilidad y posibilita su adaptación a muy diversas tareas y medios de trabajo.

Los robots cartesianos o XYZ se aplican en un número ilimitado de funciones, no obstante, la práctica ha demostrado que su adaptación es óptima en determinados procesos tales como: soldadura, paletización, impresión, corte y fresado.

La Asociación de Industrias Robóticas (RIA) de los Estados Unidos, conceptualiza un robot industrial como: “Manipulador programable multifuncional, diseñado para mover piezas, herramientas, dispositivos especiales mediante movimientos variados, programados para la ejecución de diversas tareas”¹; la Organización Internacional de Normas (ISO), define a los robots industriales de manera similar, esta dice: “Un robot industrial es un manipulador automático reprogramable y multifuncional, que posee ejes capaces de agarrar materiales, objetos, herramientas mecanismos especializados a través de operaciones programadas para la ejecución de una variedad de tareas”²; como se puede apreciar, estas definiciones se ajustan a la mayoría de las aplicaciones industriales de

¹ Tomado de la Página de Internet: <http://www.une.edu.ve/~cmarcano/ERR/clase2.htm>

² Tomado de la Página de Internet: <http://www.une.edu.ve/~cmarcano/ERR/clase2.htm>

robots cartesianos, salvo para las aplicaciones de inspección y para los robots móviles (autónomos) o robots personales.

Los robots cartesianos, deben su nombre a que combinan movimientos rectilíneos en el espacio (X, Y, Z y angular), logrando cualquier recorrido, y abarcan áreas de la ingeniería mecánica, electrónica e informática.

Los robots cartesianos, al igual que cualquier otro robot industrial, son dispositivos multifuncionales, es decir, no se encierra en un determinado trabajo, se lo puede reconfigurar y de este modo brinda una diversidad de aplicaciones.

Un robot cartesiano está configurado por tres movimientos lineales, es decir se desplaza sobre tres ejes, el X el Y y el Z, que son las coordenadas cartesianas, es por esto que también se los conoce como robots XYZ.

2.2. MOTORES PASO A PASO

Los motores de pasos, al igual que todo motor, cumplen con la principal función de ser un conversor electromecánico, que transforma la energía eléctrica en mecánica, para lo cual emplea un método tan peculiar que le permite tener su propia categoría. Específicamente hablando, se los puede definir como dispositivos electromagnéticos, rotativos, incrementales que convierten pulsos digitales en rotación mecánica.

Este tipo de motores son ideales en la construcción de fresadoras, bordadoras, tornos, y otros tipos de mecanismos que requieren movimientos exactos, es así que su mayor ventaja ante otros tipos de motores es su alta precisión en cuanto a velocidad, desplazamiento, enclavamiento y giros.

En la Figura. 2.1. se puede ver algunos modelos de motores paso a paso, utilizados en: impresoras, discos duros, juguetes, y otros dispositivos pequeños.

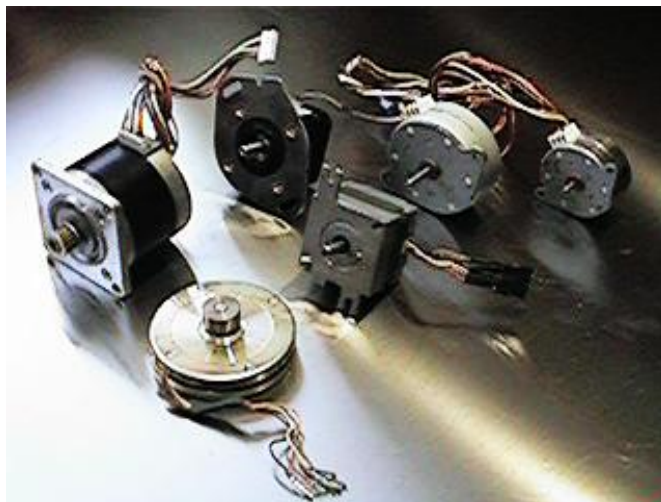


Figura. 2.1. Motores paso a paso

Al aplicar a sus bobinas un conjunto adecuado de impulsos eléctricos éstos giran sobre su eje un ángulo fijo, este ángulo recorrido, que depende de las características del motor, se le llama paso, de forma que se puede controlar mediante un circuito electrónico, la cantidad, velocidad y sentido de los pasos. Este paso puede variar desde 90° hasta pequeños movimientos de tan solo 1.8° , es decir que se necesitarán 4 pasos en el primer caso (90°) y 200 para el segundo caso (1.8°) para completar un giro completo de 360° .

Estos motores poseen la habilidad de quedar enclavados si una o más de sus bobinas están energizadas o completamente libres si no circula corriente por ninguna de sus bobinas.

La cantidad de rotación es directamente proporcional al número de pulsos y la velocidad de rotación es relativa a la frecuencia de dichos pulsos. Los motores por pasos son simples de operar en una configuración de lazo cerrado y debido a su tamaño proporcionan un excelente torque a baja velocidad. Los beneficios ofrecidos por estos motores incluyen: diseño efectivo y bajo costo, alta confiabilidad, alta precisión, no necesitan mantenimiento (no disponen de escobillas), trabajan en lazo abierto (no requieren dispositivos de realimentación).

2.2.1. Descripción general de los tipos de motores paso a paso ³

A pesar de que varios tipos de motores por pasos han sido desarrollados, todos los mismos caen dentro de tres categorías básicas:

1. de reluctancia variable (V.R.)
2. de magneto permanente (armazón metálica)
3. híbridos

1. Motores PAP de reluctancia variable (V.R.). El tipo de motor de reluctancia variable o V.R. (Figura. 2.2.) consiste en un rotor y un estator cada uno con un número diferente de dientes. Ya que el rotor no dispone de un magneto permanente el mismo gira libremente, o sea que no tiene torque de detención. A pesar de que la relación del torque a la inercia es buena, el torque dado para un tamaño de armazón dado es restringido.

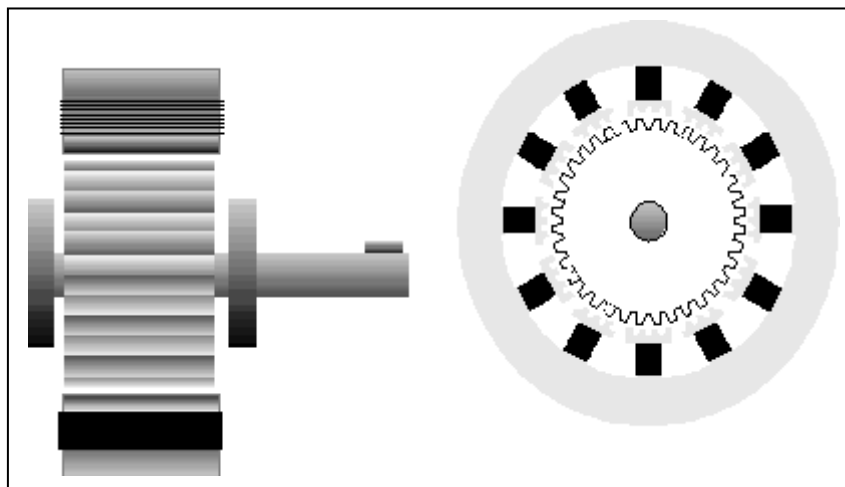


Figura. 2.2. Vista de sección de un motor por pasos de reluctancia variable

2. Motor paso a paso de magneto permanente (PM). Este es quizá el motor por pasos mas ampliamente usado para aplicaciones no industriales. En su forma más simple, el motor consiste en un rotor magneto permanentemente magnetizado radial y en un estator similar al motor V.R. como se muestra en la Figura. 2.3. Debido a las técnicas de manufactura usadas en la construcción del estator, los mismos se conocen a veces como

³ Tomado de la Página de Internet: <http://descargas.abcdatos.com/tutorial/accederL6279.htm>

motores de polo de uñas o “claw pole” en Inglés.

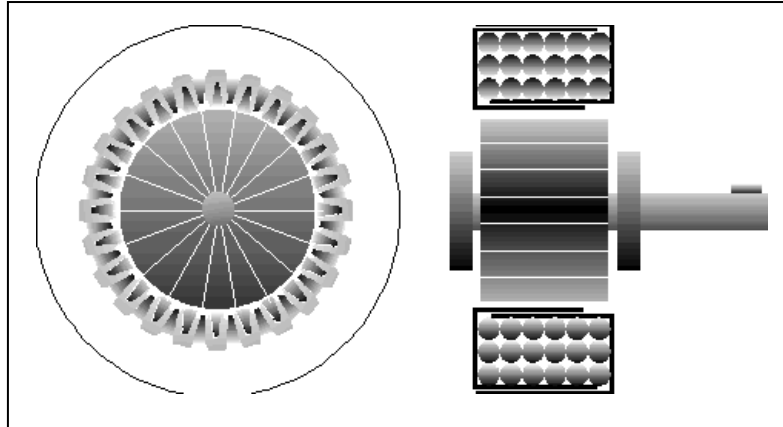


Figura. 2.3. Vista en sección de un magneto permanente

3. Motor paso a paso de tipo Híbrido. Es probablemente el más usado de todos los motores por pasos. Originalmente desarrollado como un motor PM sincrónico de baja velocidad su construcción es una combinación de los diseños V.R. y P.M. El motor Híbrido consiste en un estator dentado y un rotor de tres partes (apilado simple). El rotor de apilado simple contiene dos piezas de polos separados por un magneto permanente magnetizado (Figura. 2.4.), con los dientes opuestos desplazados en una mitad de un salto de diente para permitir una alta resolución de pasos.

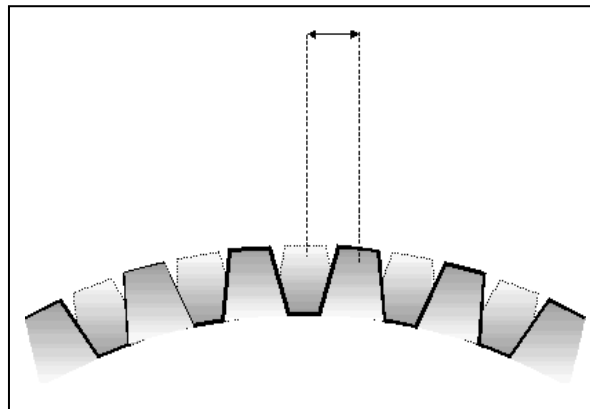


Figura. 2.4. Vista expandida ilustrativa del desplazamiento de dientes

2.2.2. Motores paso a paso de magneto permanente ⁴

Este tipo de motores PAP son los que se utilizan en el desarrollo de este proyecto y de hecho son los más utilizados dentro del campo de la robótica.

2.2.2.1. Principio de funcionamiento

Básicamente, estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes (Figura. 2.5.) y por un cierto número de bobinas excitadoras bobinadas en su estator (Figura. 2.6.).

Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador.



Figura. 2.5. Imagen del rotor

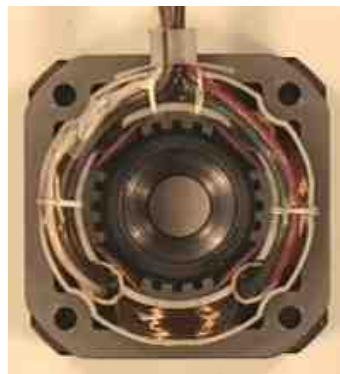


Figura. 2.6. Imagen de un estator de 4 bobinas

⁴ Tomado de la Página de Internet: <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>

2.2.2.2. Tipos de Motor PAP magneto permanente

Hay dos tipos básicos, los BIPOLARES que se componen de dos bobinas y los UNIPOLARES que tienen cuatro bobinas. Externamente se diferencian entre sí por el número de cables:

Bipolar. Estos tienen generalmente cuatro cables de salida (Figura. 2.7.). Necesitan una configuración especial para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

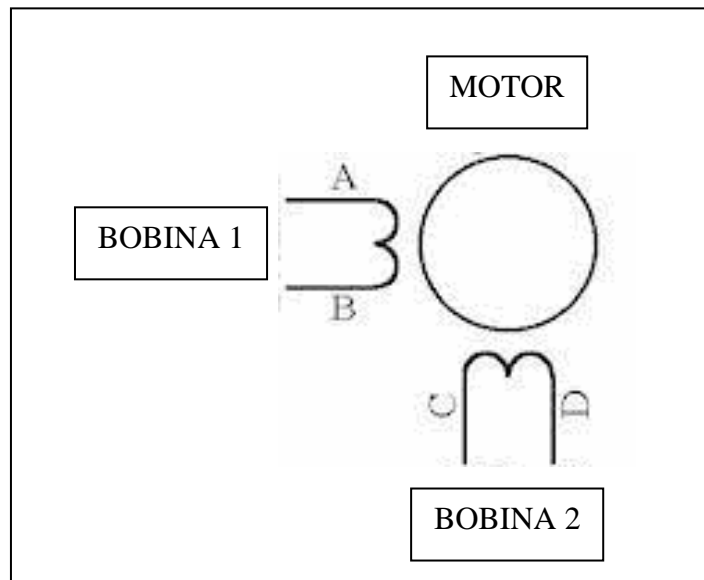


Figura. 2.7. Motor PAP Bipolar

En Figura. 2.8. se puede observar un ejemplo de control de estos motores mediante el uso de un puente en H (H-Bridge). Como se aprecia, será necesario un H-Bridge por cada bobina del motor, es decir que para controlar un motor Paso a Paso de 4 cables (dos bobinas), se necesita usar dos H-Bridges.

En general es recomendable el uso de H-Bridge integrados como son los casos del L293 (Figura. 2.9.).

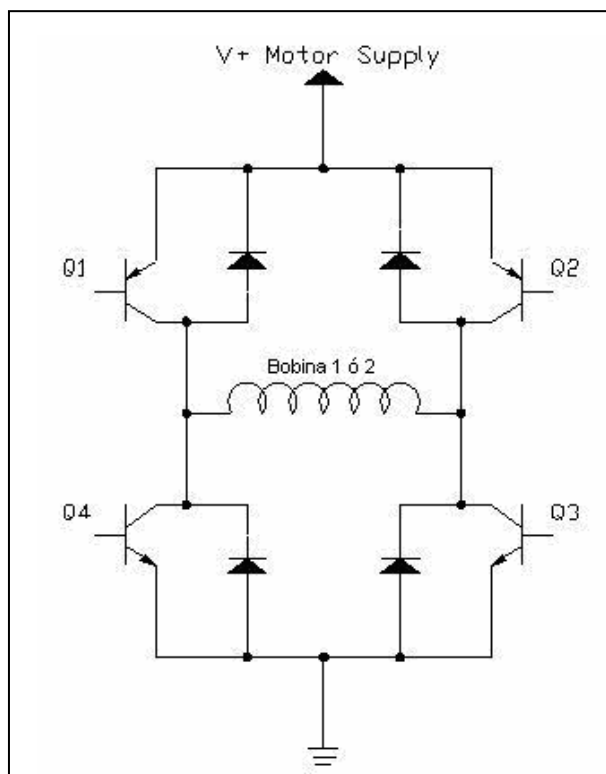


Figura. 2.8. Puente en H (H-Bridge) para control de cada bobina del motor

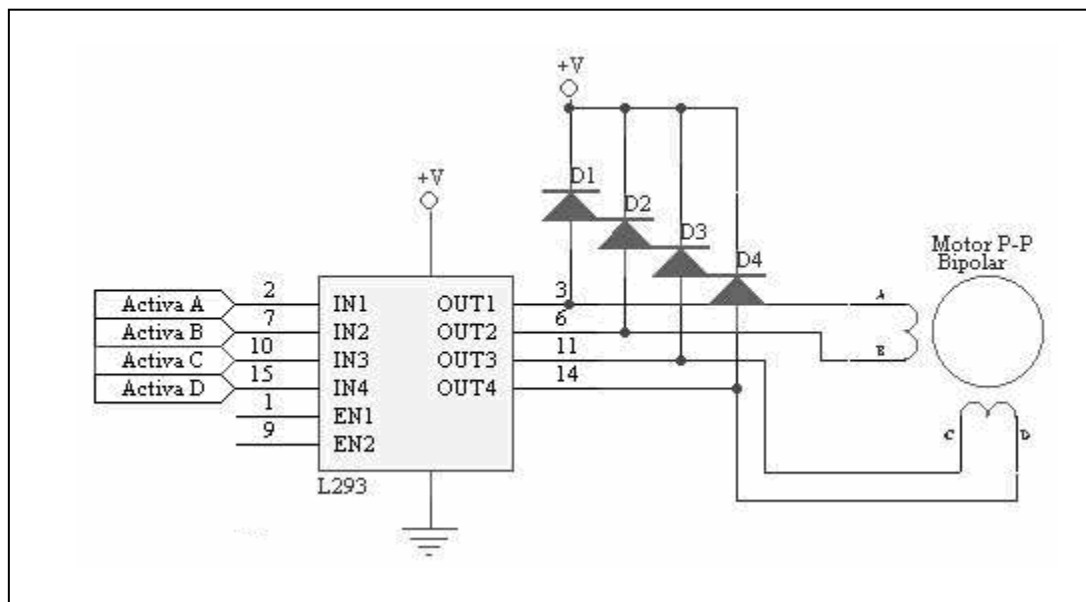


Figura. 2.9. Circuito controlador mediante H-Bridge integrados L293

Unipolar. Estos motores suelen tener 6 o 5 cables de salida, dependiendo de su conexionado interno (Figura. 2.10.). Este tipo se caracteriza por ser más simple de controlar.

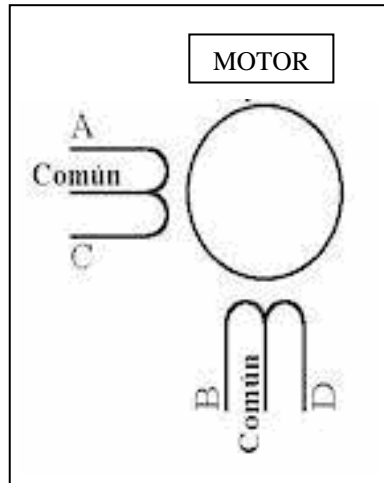


Figura. 2.10. Motor PAP Unipolar

Para identificar cada polo de la bobina, con la ayuda de un multímetro, se mide el valor de resistencia con respecto al común (alimentación). Cada polo de bobina, leerá la mitad de lo que lee entre polos de la misma bobina; por eliminación es fácil encontrar los polos de las bobinas. De haber un error no sucede nada, solo que el motor no girará. Cambiando el orden de dos de los polos de una bobina se cambia el sentido de giro. Si el motor solo tiene cinco cables, el común de alimentación se puede conectar a cualquiera de los lados.

En la Figura. 2.11. se apreciar un ejemplo de conexionado para controlar un motor paso a paso unipolar mediante el uso de un ULN2803, el cual es una array de 8 transistores tipo Darlington capaces de manejar cargas de hasta 500mA. Las entradas de activación (Activa A, B, C y D) pueden ser directamente activadas por un microcontrolador.

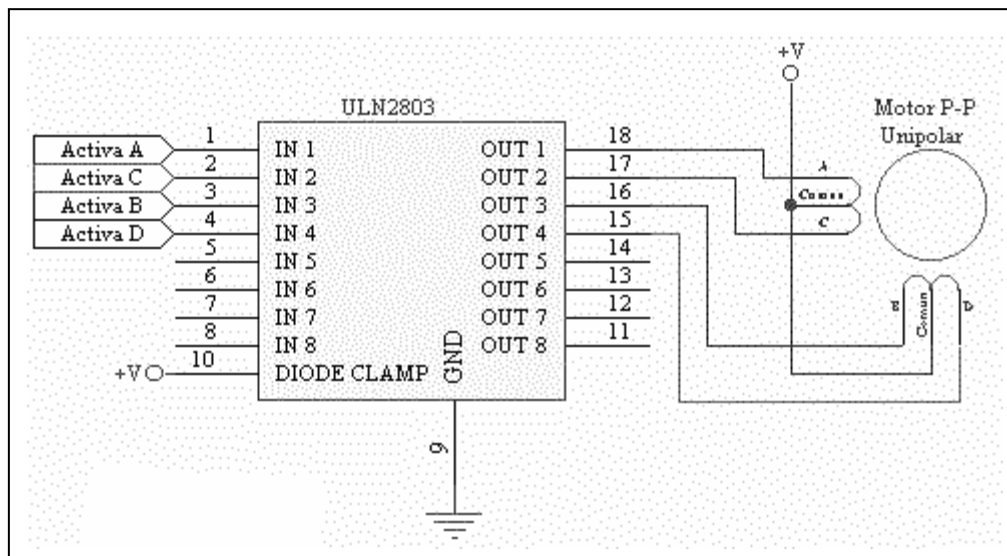


Figura. 2.11. Circuito controlador de Motor PAP unipolar mediante un ULN2803

2.2.2.3. Manejo de los Motores PAP magneto permanente

Para realizar el control de los motores paso a paso es necesario, como se ha visto generar una secuencia determinada de impulsos. Además es necesario que estos impulsos sean capaces de entregar la corriente necesaria para que las bobinas del motor se exciten, por lo general, el diagrama de bloques de un sistema con motores paso a paso es el que se muestra en la Figura. 2.12.

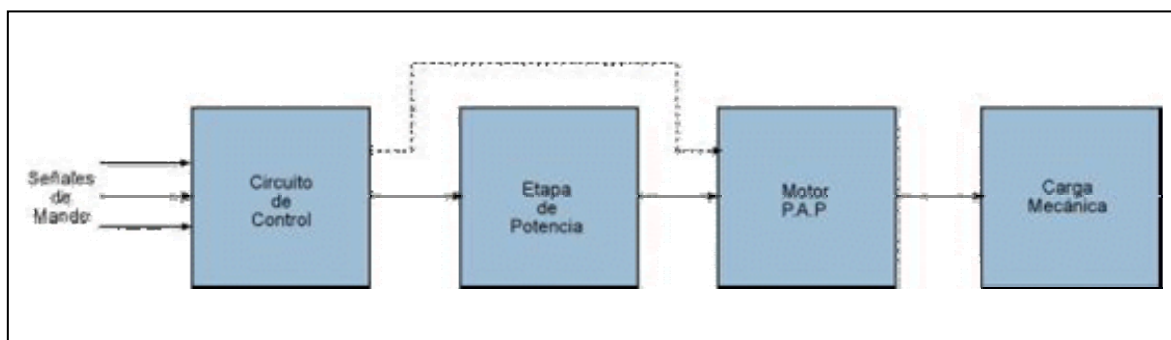


Figura. 2.12. Diagrama de bloques de un sistema con motor paso a paso

2.2.2.4. Secuencias para manejar motores paso a paso de magneto permanente

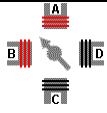
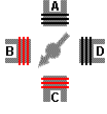
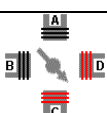
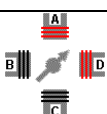
Como se dijo anteriormente, estos motores necesitan la inversión de la corriente que circula en sus bobinas en una secuencia determinada. Cada inversión de la polaridad provoca el movimiento del eje en un paso, cuyo sentido de giro está determinado por la secuencia seguida.

Las siguientes Tablas presentan las secuencias validas para el control de los motores PAP de magneto permanente; una sola secuencia para los Bipolares y tres secuencias para los Unipolares:

Tabla. 2.1. Secuencia para controlar Motor PAP Bipolar

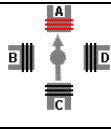
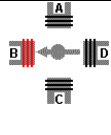
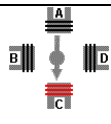
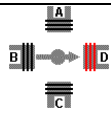
PASO	TERMINALES			
	Bobina A	Bobina B	Bobina C	Bobina D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

Tabla. 2.2. Secuencia para controla Motor PAP Unipolar - Normal

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

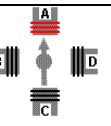
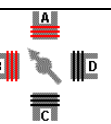
La secuencia Normal es la secuencia más usada y la que generalmente recomienda el fabricante, con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un alto torque de paso y de retención.

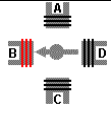
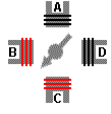
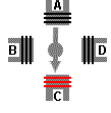
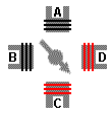
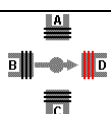
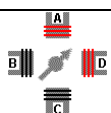
Tabla. 2.3. Secuencia para controlar Motor PAP Unipolar – Wave Drive

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

En la secuencia Wave Drive se activa solo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. La contrapartida es que al estar solo una bobina activada, el torque de paso y retención es menor.

Tabla. 2.4. Secuencia para controlar Motor PAP Unipolar – Medio Paso

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	

3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	
7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

En esta secuencia de Medio Paso se activan las bobinas de tal forma de brindar un movimiento igual a la mitad del paso real. Para ello se activan primero 2 bobinas y luego solo 1 y así sucesivamente. Como se ve, la secuencia completa consta de 8 movimientos en lugar de 4.

Cabe destacar que debido a que los motores paso a paso son dispositivos mecánicos y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los pulsos aplicados es un punto muy importante a tener en cuenta. En tal sentido el motor debe alcanzar el paso antes que la próxima secuencia de pulsos comience. Si la frecuencia de pulsos es muy elevada, el motor no trabajará correctamente.

Para obtener un arranque suave y preciso, es recomendable comenzar con una frecuencia de pulso baja y gradualmente ir aumentándola hasta la velocidad deseada sin superar la máxima tolerada. El giro en reversa debería también ser realizado previamente bajando la velocidad de giro y luego cambiar el sentido de rotación.

2.3. MICROCONTROLADORES

Los microcontroladores son dispositivos que permiten realizar el control automático de procesos. Estos son circuitos integrados conformados por un CPU, memoria RAM, ROM y puertos de entrada y salida, es decir, poseen toda arquitectura interna de un computador, además de ciertos dispositivos específicos, dependiendo del modelo, como conversores análogo/digital, temporizadores, contadores, decodificadores, entre otros.

Ningún microcontrolador puede trabajar sin antes ser debidamente programado, lo cual se realiza en lenguaje Assembler. Cada microcontrolador varía su conjunto de instrucciones de acuerdo a su fabricante y modelo, y dependiendo del número de dichas instrucciones que este maneje, se los clasifica en microcontroladores de arquitectura RISC (Computador de Juego de Instrucciones Reducido) o CISC (Computador de Juego de Instrucciones Complejo).

El microcontrolador se comunica con el medio externo mediante sus pines, cada uno de ellos tiene una función específica. Depende del usuario sacarle el mayor provecho mediante la programación que se le aplique.

MicroChip es uno de los más grandes fabricantes de microcontroladores en el mundo y es de esta casa de donde sale el microcontrolador que se utiliza en este proyecto, el PIC 16F877a, escogido por ser el más versátil y eficiente para realizar el control de nuestro robot, ya que soporta modo de comunicación serial, tiene una amplia memoria para datos y programa y algo muy importante, es de tipo RISC. Además, este PIC (Peripheral Interface Controller) posee memoria FLASH, que nos permite borrar electrónicamente para reprogramar.

2.3.1. Microcontrolador PIC 16F877A

Como se mencionó antes, se utiliza este modelo, debido a que brinda las herramientas suficientes para el desarrollo del proyecto. La Figura. 2.13. muestra físicamente un microcontrolador PIC16F877A.



Figura. 2.13. PIC 16F877A

La Tabla. 2.5. presenta las principales características de forma más específica de un microcontrolador de la serie antes mencionada.

Tabla. 2.5. Características del Microcontrolador PIC 16F877A

CARACTERÍSTICAS	16F877A
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8

Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Canales Pwm	2
Pila Harware	-
Ejecución En 1 Ciclo Máquina	-

2.3.1.1. Configuración y descripción de los pines

En la Figura. 2.14. se puede apreciar como están distribuidos sus pines y a continuación se muestra la Tabla. 2.6. en donde se especifica cada uno de ellos:

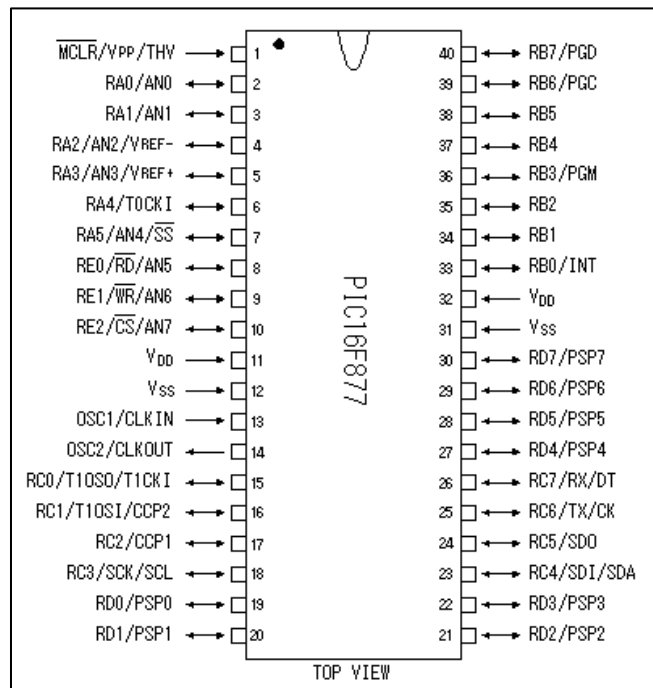


Figura. 2.14. Configuración de pines del PIC 16F877A

Tabla. 2.6. Descripción de pines del PIC 16F877A

NOMBRE DEL PIN	# PIN	TIPO	TIPO DE BUFFER	DESCRIPCIÓN
OSC1/CLKIN	13	I	ST/MOS	Entrada del oscilador de cristal / Entrada de señal de reloj externa
OSC2/CLKOUT	14	O	-	Salida del oscilador de cristal
MCLR/Vpp/THV	1	I/P	ST	Entrada del Master clear (Reset) o entrada de voltaje de programación o modo de control high voltaje test
RA0/AN0	2	I/O	TTL	<p>PORTA es puerto I/O bidireccional</p> <p>RA0: puede ser salida analógica 0</p> <p>RA1: puede ser salida analógica 1</p> <p>RA2: puede ser salida analógica 2 o referencia negativa de voltaje</p> <p>RA3: puede ser salida analógica 3 o referencia positiva de voltaje</p> <p>RA4: puede ser entrada de reloj el timer0.</p> <p>RA5: puede ser salida analógica 4 o el esclavo seleccionado por el puerto serial síncrono.</p>
RA1/AN1	3	I/O	TTL	
RA2/AN2/ Vref-	4	I/O	TTL	
RA3/AN3/Vref+	5	I/O	TTL	
RA4/T0CKI	6	I/O	ST	
RA5/SS/AN4	7	I/O	TTL	

RB0/INT	33	I/O	TTL/ST	<p>PORTB es un puerto I/O bidireccional. Puede ser programado todo como entradas RB0 puede ser pin de interrupción externo. RB3: puede ser la entrada de programación de bajo voltaje Pin de interrupción Pin de interrupción Pin de interrupción. Reloj de programación serial</p>
RB1	34	I/O	TTL	
RB2	35	I/O	TTL	
RB3/PGM	36	I/O	TTL	
RB4	37	I/O	TTL	
RB5	38	I/O	TTL	
RB6/PGC	39	I/O	TTL/ST	
RB7/PGD	40	I/O	TTL/ST	
RCO/T1OSO/ T1CKI	15	I/O	ST	<p>PORTC es un puerto I/O bidireccional RCO puede ser la salida del oscilador timer1 o la entrada de reloj del timer1 RC1 puede ser la entrada del oscilador timer1 o salida PMW 2 RC2 puede ser una entrada de captura y comparación o salida PWN RC3 puede ser la entrada o salida serial de reloj síncrono para</p>
RC1/T1OS1/ CCP2	16	I/O	ST	
RC2/CCP1	17	I/O	ST	
RC3/SCK/SCL	18	I/O	ST	
RC4/SD1/SDA	23	I/O	ST	

RC5/SD0	24	I/O	ST	modos SPI e I2C RC4 puede ser la entrada de datos SPI y modo I2C RC5 puede ser la salida de datos SPI RC6 puede ser el transmisor asíncrono USART o el reloj síncrono. RC7 puede ser el receptor asíncrono USART o datos síncronos
RC6/Tx/CK	25	I/O	ST	
RC7/RX/DT	26	I/O	ST	
RD0/PSP0	19	I/O	ST/TTL	PORTD es un puerto bidireccional paralelo
RD1/PSP1	20	I/O	ST/TTL	
RD2/PSP2	21	I/O	ST/TTL	
RD3/PSP3	22	I/O	ST/TTL	
RD4/PSP4	27	I/O	ST/TTL	
RD5/PSP5	28	I/O	ST/TTL	
RD6/PSP6	29	I/O	ST/TTL	
RD7/PSP7	30	I/O	ST/TTL	
REO/RD/AN5	8	I/O	ST/TTL	PORTE es un puerto I/O bidireccional REO: puede ser control de lectura para el puerto esclavo paralelo o entrada analógica 5 RE1: puede ser escritura de control para el puerto paralelo esclavo o entrada analógica 6
RE1/WR/AN	9	I/O	ST/TTL	

RE2/CS/AN7	10	I/O	ST/TTL	RE2: puede ser el selector de control para el puerto paralelo esclavo o la entrada analógica 7.
Vss	12.31	P	-	Referencia de tierra para los pines lógicos y de I/O
Vdd	11.32	P	-	Fuente positiva para los pines lógicos y de I/O
NC	-	-	-	No está conectado internamente

2.3.1.2. Comparación con otros modelos de PIC

A fin de comprender mejor las ventajas que brinda este modelo, se puede observar la Tabla. 2.7., en donde se muestra una comparación con otros modelos de su familia.

Tabla. 2.7. Comparación de modelos PIC de la familia 16F8XX

Características	PIC16F873	PIC16F874	PIC16F876	PIC16F877A
Frecuencia Máxima de operación	20MHz	20MHz	20MHz	20MHz
Memoria FLASH (14-bit words)	4K	4K	8K	8K
Memoria para Datos (bytes)	192	192	368	368
Memoria datos EEPROM (bytes)	128	128	256	256
Puertos de entrada y salida (I/O)	RA0-5 (6) RB0-7 (8) RC0-7 (8)	RA0-5 (6) RB0-7 (8) RC0-7 (8) RD0-7 (8) RE0-2 (3)	RA0-5 (6) RB0-7 (8) RC0-7 (8)	RA0-5 (6) RB0-7 (8) RC0-7 (8) RD0-7 (8) RE0-2 (3)
Timers	3	3	3	3
CCP	2	2	2	2
Comunicación Serial	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Comunicación Paralelo	-	PSP	-	PSP
10-bit Conversor Análogo/ Digital	5 Canales	8 Canales	5 Canales	8 Canales
Set de instrucciones	35 Instrucciones	35 Instrucciones	35 Instrucciones	35 Instrucciones
Pines (DIP)	28 Pines	40 Pines	28 Pines	40 Pines

2.4. LCD

LCD son las siglas en inglés de "Pantalla de Cristal Líquido" ("Liquid Crystal Display"). La existencia de cristales líquidos fue descubierta en 1888 por el austriaco Fredeich Rheinizer. Más adelante, hacia mediados de 1960, los científicos descubrieron que al someter el cristal líquido a cargas externas, éste modificaba sus propiedades en relación con el paso de la corriente. Este dispositivo fue inventado por Jack Janning, quien era empleado de NCR Corporation. Desde principios de los 70 las pantallas de cristal líquido se han ido extendiendo a calculadoras, relojes, televisores, portátiles, cámaras digitales, entre otros.

La siguiente figura (Figura. 2.15.) presenta un LCD de dos filas y treinta y dos segmentos que es uno de los más comunes y que ha sido utilizado en el desarrollo de este proyecto.

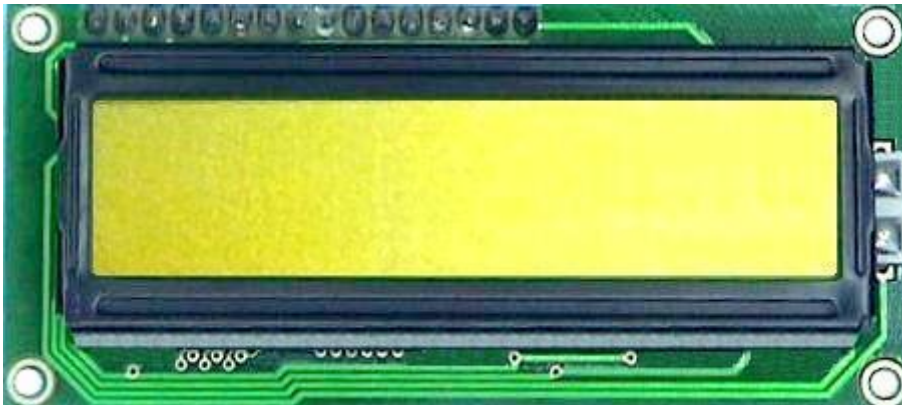


Figura. 2.15. LCD de dos filas y treinta y dos segmentos

Los LCD son visualizadores pasivos, esto significa que no emiten luz como el visualizador o display alfanumérico hecho a base de un arreglo de LEDs. Es por esa razón que, algunas veces, cuando se intenta ver la hora en un reloj que utiliza esta tecnología, es necesario una fuente de luz adicional.

Tiene una vida aproximada de 50,000 horas. Hay diferentes tipos de presentaciones y son muy fáciles de configurar. Hay desde visualizadores comunes de 7 segmentos, hasta una matriz de puntos, todos ellos muy delgados.

2.4.1. Principio de funcionamiento del LCD

El LCD utiliza una tecnología de sustancias (moléculas de cristal líquido), que se encuentran entre dos placas de vidrio paralelas con una separación de unos micrones. Estas placas de vidrio tienen unos electrodos especiales que definen, con su forma, los símbolos, caracteres, etc. que se visualizarán. Cuando la corriente circula entre los electrodos transparentes con la forma a representar (por ejemplo, un segmento de un número) el material cristalino se reorienta alterando su transparencia, es decir, hace que aparezca una zona oscura sobre un fondo claro (contraste positivo). De esta manera aparece la información que se desea mostrar.

Dicho de una forma más simple, el LCD modifica la luz que lo incide. Dependiendo de la polarización que se esté aplicando, el LCD reflejará o absorberá más o menos luz. Cuando un segmento recibe la tensión de polarización adecuada no reflejará la luz y aparecerá en la pantalla del dispositivo como un segmento oscuro.

El LCD se basa en la luz que en el se refleja, ya sea la luz del sol o la de las bombillas, es decir, deja que la luz del sol o artificial sea reflejada en el y nosotros lo veamos de un color distinto al de su fondo para así poder distinguir los números o texto que en el se muestran, por eso el gran inconveniente de un LCD es que de noche no se ve si no está retroiluminado, y que además no es visible en un gran ángulo de visión. Su principal ventaja, además de su reducido tamaño, es el ahorro de energía.

Dependiendo de sus características ópticas y el modo de visualización que brindan, los LCD pueden clasificarse como:

- reflectivos
- transmisivos
- transreflectivos

2.4.2. Modos de visualización de un LCD

Como ya se menciona, existen tres modos para visualizar un LCD: El modo reflector, que refleja la luz ambiente a través del visualizador, el modo transmisor, que es iluminado desde atrás en forma artificial y el modo translector, que es un híbrido de los dos modos antes mencionados y se utiliza para desplegar la información bajo cualquier condición de iluminación, es el único que refleja tanto la luz ambiente como la luz artificial de fondo difusa para uso nocturno. Véase la Figura. 2.16.

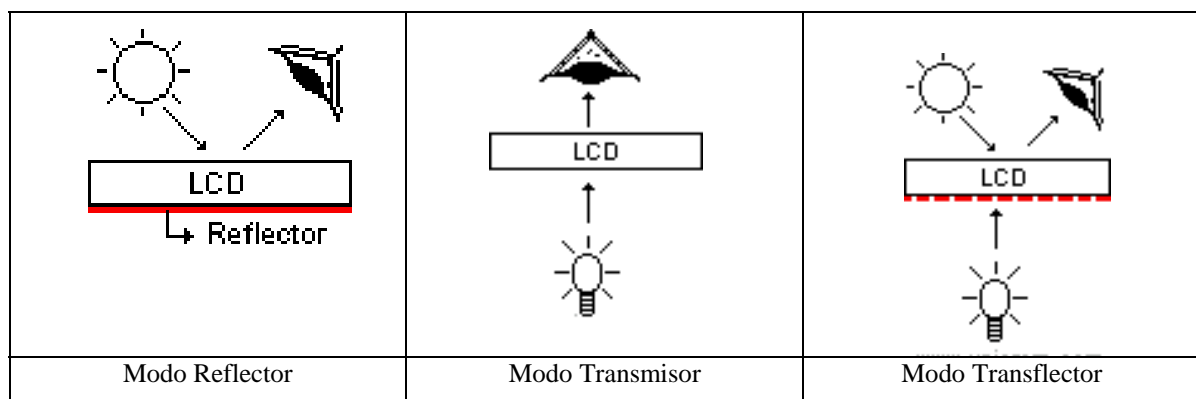


Figura. 2.16. Modos de Visualización de un LCD

2.4.3. Características principales del LCD

Las pantallas LCD se encuentran en multitud de dispositivos industriales y de consumo: máquinas expendedoras, electrodomésticos, equipos de telecomunicaciones, computadoras, etc. Todos estos dispositivos utilizan pantallas fabricadas por terceros de una manera más o menos estandarizada. Cada LCD se compone de una pequeña placa integrada que consta de:

- La propia pantalla LCD.
- Un microchip controlador.
- Una pequeña memoria que contiene una tabla de caracteres.
- Un interfaz de contactos eléctricos, para conexión externa.
- Opcionalmente, una luz trasera para iluminar la pantalla.

El controlador simplifica el uso del LCD proporcionando una serie de funciones básicas que se invocan mediante el interfaz eléctrico, destacando:

- La escritura de caracteres en la pantalla.
- El posicionado de un cursor parpadeante, si se desea.
- El desplazamiento horizontal de los caracteres de la pantalla (*scrolling*).
- Etc.

La memoria implementa un mapa de bits para cada carácter de un juego de caracteres, es decir, cada octeto de esta memoria describe los puntitos o pixels que deben iluminarse para representar un carácter en la pantalla. Generalmente, se pueden definir caracteres a medida modificando el contenido de esta memoria. Así, es posible mostrar símbolos que no están originalmente contemplados en el juego de caracteres.

El interfaz de contactos eléctricos suele ser de tipo paralelo, donde varias señales eléctricas simultáneas indican la función que debe ejecutar el controlador junto con sus parámetros. Por tanto, se requiere cierta sincronización entre estas señales eléctricas.

La luz trasera facilita la lectura de la pantalla LCD en cualquier condición de iluminación ambiental.

2.4.4. Tipos de LCD

Existen dos tipos de pantallas LCD en el mercado: pantallas de texto y pantallas gráficas.

LCD de gráficos. Las pantallas LCD gráficas permiten encender y apagar individualmente pixels de la pantalla. De esta manera es posible mostrar gráficos en blanco y negro. No solamente texto. Los controladores más populares son el Hitachi HD61202 y el Samsung KS0108. Los tamaños también están estandarizados y se miden en filas y columnas de pixels. Algunos tamaños típicos son 128x64 y 96x60. Naturalmente, algunos controladores también permiten la escritura de texto de manera sencilla.

Estas pantallas son más caras y complejas de utilizar. Existen pocas aplicaciones donde no baste con un LCD de texto. Se suelen utilizar, por ejemplo, en ecualizadores gráficos.

LCD de texto. Los LCD de texto son los más baratos y simples de utilizar. Solamente permiten visualizar mensajes cortos de texto. Existen algunos modelos estandarizados en la industria, en función de su tamaño medido en número de líneas y columnas de texto. Existen modelos de una, dos y cuatro filas únicamente. El número de columnas típico es de ocho, dieciséis, veinte y cuarenta caracteres.

Para el caso específico de este proyecto, se utiliza un LCD de Texto de dos filas, de dieciséis caracteres cada una, con Backlight (luz propia) y su controlador es un Hitachi 44780.

Debido a la funcionalidad que brinda el controlador **Hitachi HD44780**, se ha convertido en un estándar para muchos fabricantes, quienes imitan sus características. A continuación se presenta la Tabla. 2.8. que contiene la descripción de cada pin de un LCD con este modelo de controlador:

Tabla. 2.8. Descripción de los pines del LCD

# Pin	Nombre	Descripción
1	Vss	Tierra
2	Vdd	Alimentación 5 voltios DC
3	Vo	Ajuste contraste – 0~5 voltios DC
4	RS	Selección del Registro (control / datos)
5	R/W	Lectura / Escritura
6	E	Habilitación
7~ 13	D0 ~ D7	Ocho señales eléctricas – Bus de datos bidireccional
15	A	Alimentación del backlight 5 voltios DC
16	K	Tierra del backlight

2.5. SENSORES FIN DE CARRERA

Dentro de los componentes electrónicos, el **final de carrera** o sensor de contacto (Véase la Figura. 2.17.) son dispositivos eléctricos, neumáticos o mecánicos situados al final del recorrido de un elemento móvil, como por ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito.

Internamente pueden contener interruptores normalmente abiertos, cerrados (NA o NO en inglés) o conmutadores dependiendo de la operación que cumplan al ser accionados, de ahí la gran variedad de finales de carrera que existen en mercado.



Figura. 2.17. Sensor fin de carrera

Generalmente estos sensores están compuestos por dos partes: un cuerpo donde se encuentran los contactos y una cabeza que detecta el movimiento. Su uso es muy diverso, empleándose, en general, en todas las máquinas que tengan un movimiento rectilíneo de ida y vuelta o sigan una trayectoria fija, es decir, aquellas que realicen una carrera o recorrido fijo, como por ejemplo ascensores, montacargas, robots, etc.

Los finales de carrera están fabricados en diferentes materiales tales como metal, plástico o fibra de vidrio.

Para la estructura del robot propuesto en este proyecto se utiliza sensores fin de carrera de contacto, configurados como normalmente abiertos (Figura. 2.18.), esto quiere decir que su salida entregara 1 lógico (5V) todo el tiempo y cuando son alcanzados, se

activan cerrando el circuito interno y su salida entrega un 0 lógico (tierra). Este cambio es el que identifica el microcontrolador y ejecuta una acción.

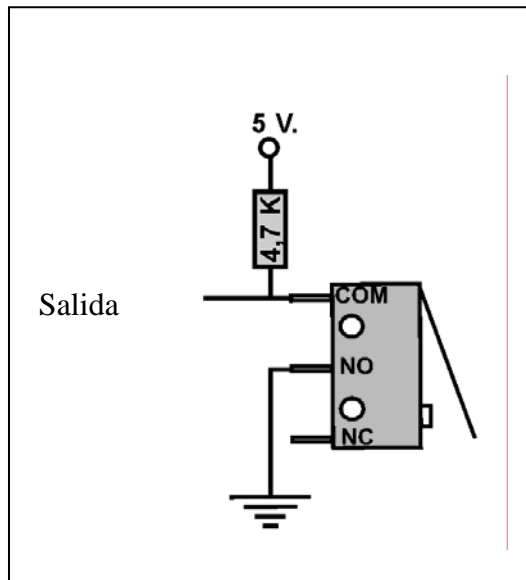


Figura. 2.18. Sensor fin de Carrera Normalmente Abierto (NO)

CAPÍTULO 3

DISEÑO DEL ROBOT

En este capítulo se detalla el diseño de cada una de las partes que componen al robot y como están estructuradas y ensambladas entre sí. Básicamente, al igual que todo robot industrial cartesiano, posee una herramienta de trabajo que se moviliza sobre ejes determinados utilizando motores, el área de trabajo lo definen la longitud de dichos ejes. Puede verse ciertas similitudes con el diseño de otros robots cartesianos, ya que la estructura expuesta es generalmente la misma para todo robot de esta categoría.

3.1. ESTRUCTURA Y DISEÑO MECÁNICO

3.1.1. Plataforma de trabajo

Lo primero que se necesita es una mesa o plataforma en donde se va a montar el robot; es importante escoger bien su material ya que debe permitir perforar, clavar e instalar tornillos sobre ella. Tendrá 80 centímetros de largo y 70 de ancho, con un espesor de 1.5 centímetros, como se muestra en la Figura. 3.1.

3.1.2. El eje Y

El eje Y estará conformado por dos ejes paralelos fijos al tablero, cada uno de 40 centímetros de largo y 1 de diámetro, tal como se muestra en la Figura. 3.2.; sobre ellos se

desplazarán piezas mediante bocines, de modo que el material que se escoja para los mismos, debe ser extremadamente liso.

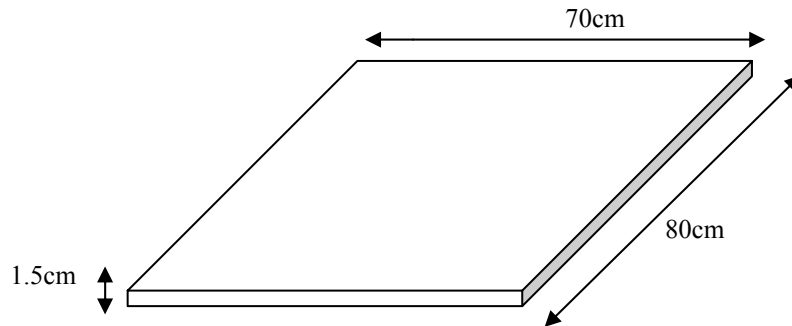


Figura. 3.1. Esquema de la plataforma de trabajo

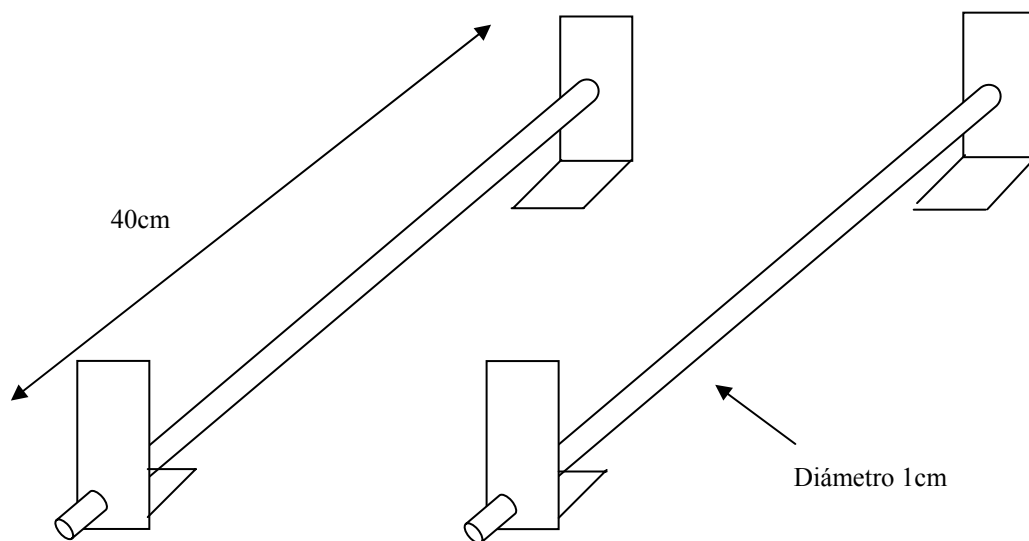


Figura. 3.2. Esquema de los ejes Y

En la siguiente figura (Figura. 3.3.) se puede apreciar como quedarían los ejes ya con las piezas móviles instaladas, sobre las cuales se atornillará una plataforma que será el Eje X.

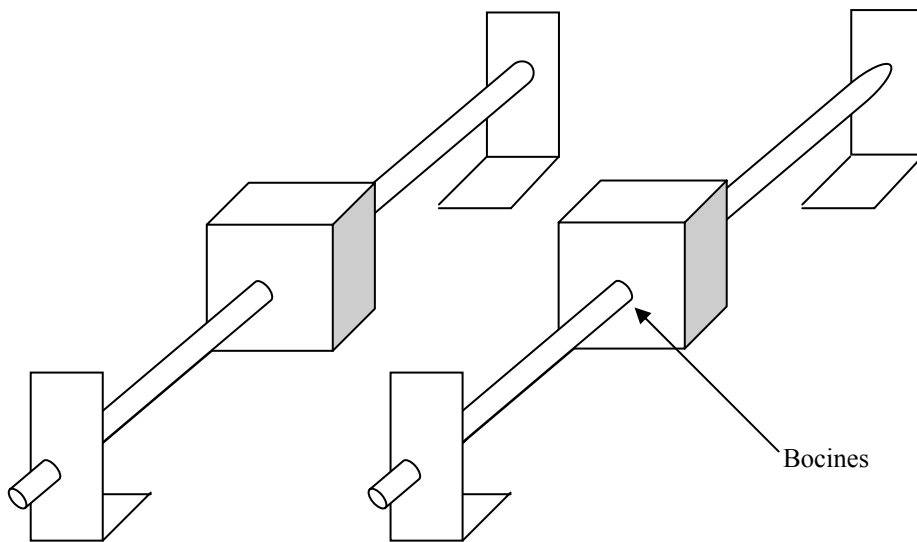


Figura. 3.3. Esquema de los ejes Y con las piezas móviles (soportes)

3.1.3. El eje X

El eje X como ya se mencionó será una plataforma, irá atornillada a las piezas móviles del los ejes Y, el material a escoger para este deberá tener la característica de ser rígido pero a su vez lo mas liviano posible, lo mas recomendado es plástico. Tendrá 44 centímetros de largo por 12 de ancho. Su forma, como se muestra en la Figura. 3.4., será algo especial debido a que en ella ira montado el eje Z.

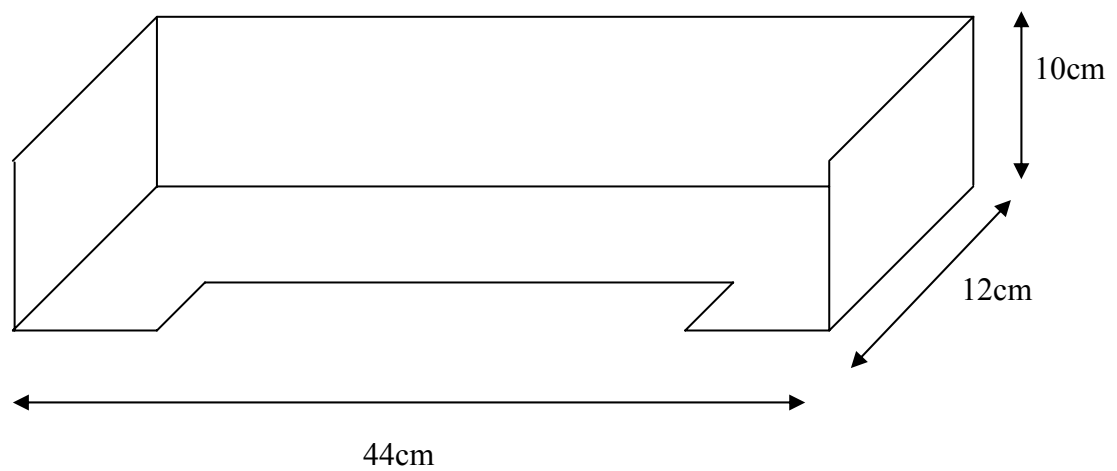


Figura. 3.4. Esquema de eje X (Plataforma)

3.1.4. El eje Z

Como se muestra en la Figura. 3.5., será una pieza móvil, que se desplazará a lo largo del eje X sobre dos ejes auxiliares.

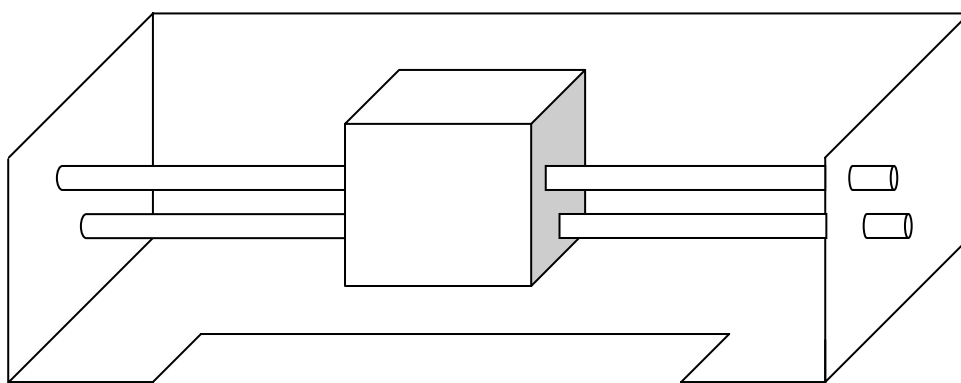


Figura. 3.5. Esquema del eje Z

Esta pieza tendrá una abrazadera metálica de 8 por 6 centímetros, donde se instalará la herramienta de trabajo. La siguiente figura (Figura. 3.6.) permite apreciar mejor lo mencionado.

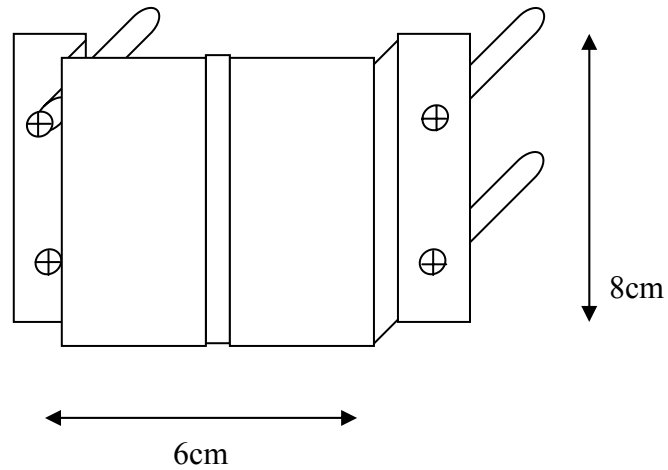


Figura. 3.6. Abrazadera que sostendrá la herramienta de trabajo

La siguiente figura (Figura. 3.7.), muestra cada una de las partes del robot acopladas entre si y montadas en la mesa de trabajo.

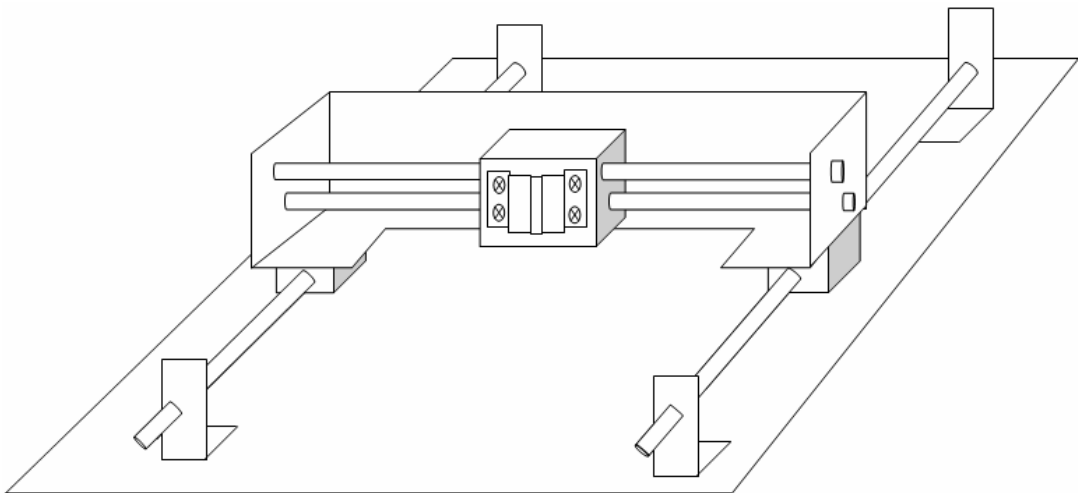


Figura. 3.7. Esquema General del Robot

3.2. DISEÑO DEL HARDWARE DEL ROBOT

El hardware del robot está conformado por los motores paso a paso, el microcontrolador que los controla, el LCD que despliega en pantalla las opciones de trabajo, el panel de control con el que se ejecuta los movimientos y los sensores fin de carrera instalados al principio y fin de cada eje que definen el área de trabajo, todo alimentado por una fuente de voltaje DC.

Para facilitar el entendimiento, se presenta a continuación un diagrama de bloques (Figura. 3.8.) que muestra las distintas etapas que componen el hardware electrónico del robot.

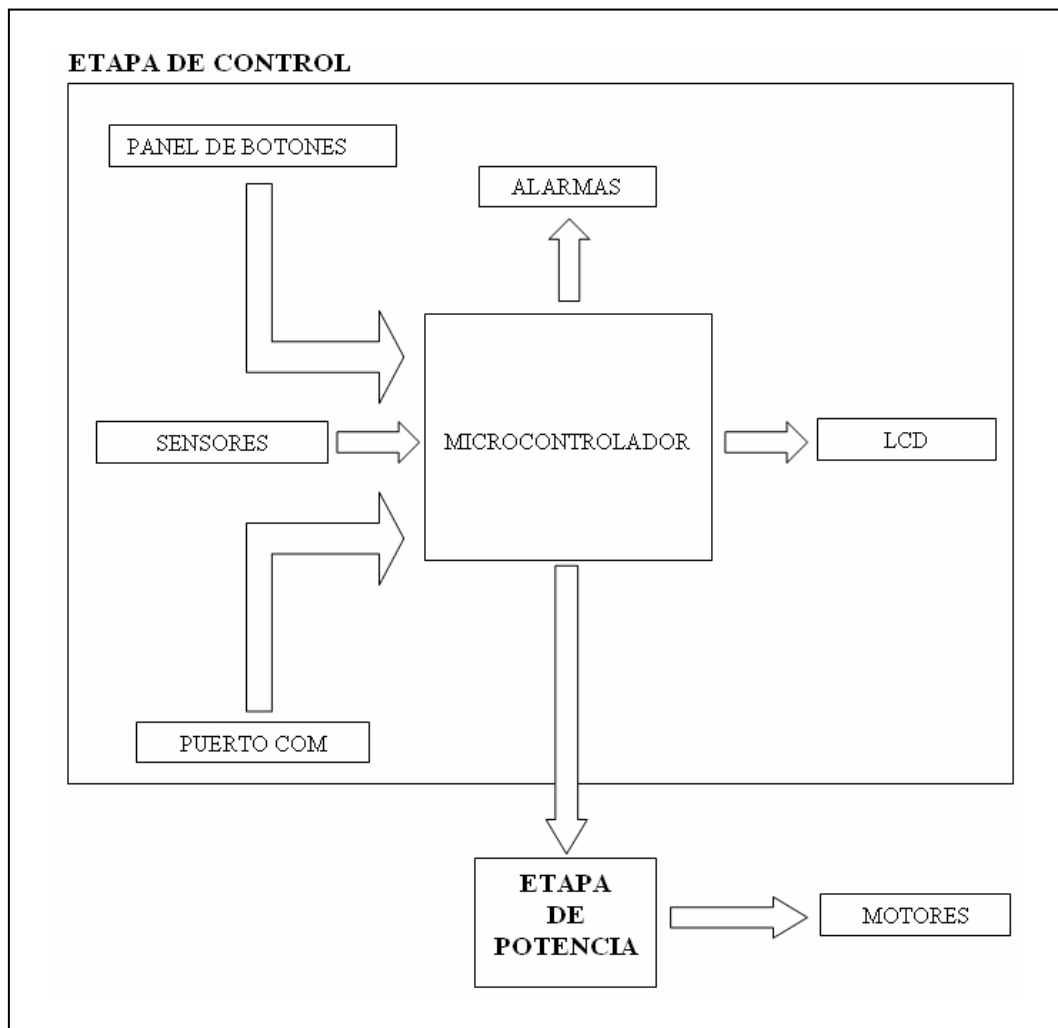


Figura. 3.8. Esquema del Hardware Electrónico del Robot

Como se puede apreciar en la Figura. 3.8. el hardware del robot está comprendido de dos etapas: de control y de potencia. Estas etapas se enlazan por medio del microcontrolador, el mismo que interactúa con todos los elementos que conforman al robot; a él llega toda señal ya sea de los sensores, el puerto de comunicación serial o de los botones del panel de control. Las señales son procesadas y de acuerdo al programa que tiene cargado el microcontrolador, ejecuta acciones como son: presentar un texto en el LCD, encender una alarma, mover un motor, entre otros.

3.2.1. Etapa de Control

En esta parte del proyecto, se detalla como están configurados e interconectados los dispositivos electrónicos que conforman la etapa de control. Básicamente, todo elemento electrónico que interviene en el funcionamiento del robot esta controlado por el microcontrolador PIC16F877A, del cual se habló en el capítulo de fundamentos, es decir, tanto motores como sensores, el LCD, leds, alarmas auditivas y todos los botones del panel de control se conectan a una de las entradas o salidas del microcontrolador, inclusive el puerto de comunicación serial que permite el funcionamiento del robot por medio de una computadora.

En la Tabla. 3.1. se muestra los cuarenta pines que posee el microcontrolador y que elemento está conectado a cada uno de ellos.

Tabla. 3.1. Configuración de pines del microcontrolador PIC 16F877A para el proyecto

# PIN	Nombre / Descripción	Elemento que está conectado
1	MCLR (reset externo)	5 V DC
2	Puerto RA0	LCD
3	Puerto RA1	LCD
4	Puerto RA2	LCD
5	Puerto RA3	LCD
6	Puerto RA4	LCD
7	Puerto RA5	Luz del LCD
8	Puerto RE0	LCD

9	Puerto RE1	Alarma visual y auditiva (Chicharra y Led Verde)
10	Puerto RE2	Puerto de comunicación serial
11	VDD	5 V DC
12	VSS	Libre
13	oscilador OSC1 / CLK IN	Cristal oscilador 4Mhz
14	oscilador OSC2 / CLK OUT	Cristal oscilador 4Mhz
15	Puerto RC0	Fila A del panel de control (botones)
16	Puerto RC1	Fila B del panel de control (botones)
17	Puerto RC2	Fila C del panel de control (botones)
18	Puerto RC3	Fila D del panel de control (botones)
19	Puerto RD0	Sensor fin de carrera, inicio eje X
20	Puerto RD1	Sensor fin de carrera, fin eje X
21	Puerto RD2	Sensor fin de carrera, fin eje Y
22	Puerto RD3	Sensor fin de carrera, inicio eje Y
23	Puerto RC4	Columna UNO del panel de control (botones)
24	Puerto RC5	Columna DOS del panel de control (botones)
25	Puerto RC6	Columna TRES del panel de control (botones)
26	Puerto RC7	Columna CUATRO del panel de control (botones)
27	Puerto RD4	Bobina D del motor Y
28	Puerto RD5	Bobina C del motor Y
29	Puerto RD6	Bobina B del motor Y
30	Puerto RD7	Bobina A del motor Y
31	VSS	Tierra
32	VDD	Libre
33	Puerto RB0	Bobina D del motor Z
34	Puerto RB1	Bobina C del motor Z
35	Puerto RB2	Bobina B del motor Z
36	Puerto RB3	Bobina A del motor Z
37	Puerto RB4	Bobina D del motor X
38	Puerto RB5	Bobina C del motor X
39	Puerto RB6	Bobina B del motor X
40	Puerto RB7	Bobina A del motor X

3.2.1.1. Conexión de los dispositivos electrónicos al microcontrolador

Como ya se dijo, todos los dispositivos electrónicos que intervienen en el funcionamiento del robot se encuentran controlados por el microcontrolador PIC16F877A e interconectados por el mismo.

A continuación se detalla la conexión de cada dispositivo electrónico al microcontrolador. Para facilitar el entendimiento se recomienda tener presente la Tabla. 3.1.

Alimentación y configuración del Microcontrolador. Este es el primer paso que se debe dar para que el microcontrolador empiece a trabajar. Se conecta 5 V DC al pin VDD y tierra al pin VSS. A demás se requiere activar el pin MCLR conectándolo a 5 V DC con una resistencia de por medio. Finalmente se debe colocar un cristal oscilador entre los pines OSC1 y OSC2 y a su vez llevarlos a tierra pasando por un capacitor respectivamente. Observe la Figura. 3.9.

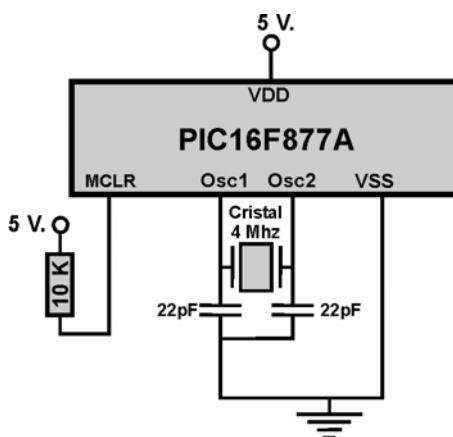


Figura. 3.9. Configuración del microcontrolador PIC16F877A

Conexión del LCD. Para hacer funcionar el LCD existen configuraciones predeterminadas, es decir, cada pin del LCD tiene un puerto designado en el microcontrolador, lo cual se define al inicio del programa grabado en el PIC. Los pines V+ y A, van conectados a 5 V DC, los pines R/W, Gnd y Vo se los conecta a tierra, sin olvidar una resistencia de protección por cada uno, el pin K se lo conecta en el puerto A5 con un

transistor de por medio, el pin E está conectado al puerto E0, y por último, los pines D4, D5, D6, y D7 se conectan a los puertos A0, A1, A2, A3, respectivamente. Observe la Figura. 3.10.

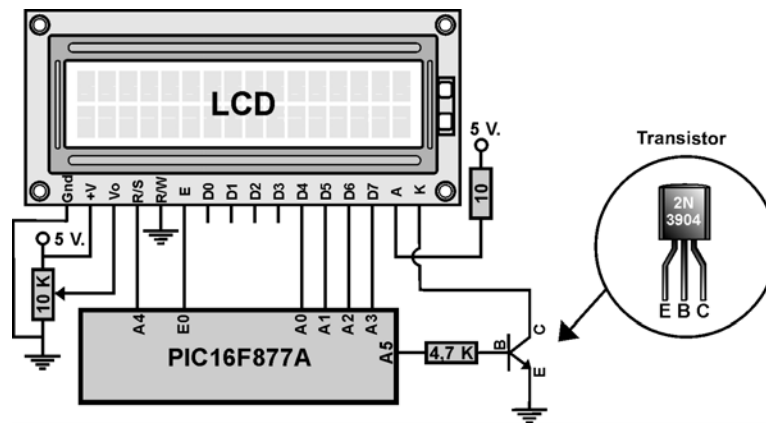


Figura. 3.10. Conexión del LCD

Conexión de alarma auditiva y visual. En la siguiente figura (Figura.3.11), se observa como se debe conectar la Chicharra y el LED Verde (Alarma auditiva y visual). Es fácil ver que los dos dispositivos se encuentran en paralelo entre sí y conectados al puerto E1 del PIC, es decir, cuando el puerto E1 se activa, suena la Chicharra y se enciende el Led al mismo tiempo.

En vista de que la señal que manda el Pic no es lo suficientemente potente para que la chicharra se encienda, esta debe ser amplificada, lo cual se logra haciéndola pasar por un transistor conectado a 12 V, o sea que al llegar la señal del microcontrolador al transistor, este envía los 12 V a la chicharra y logra que esta suene. No se debe olvidar las resistencias de protección.

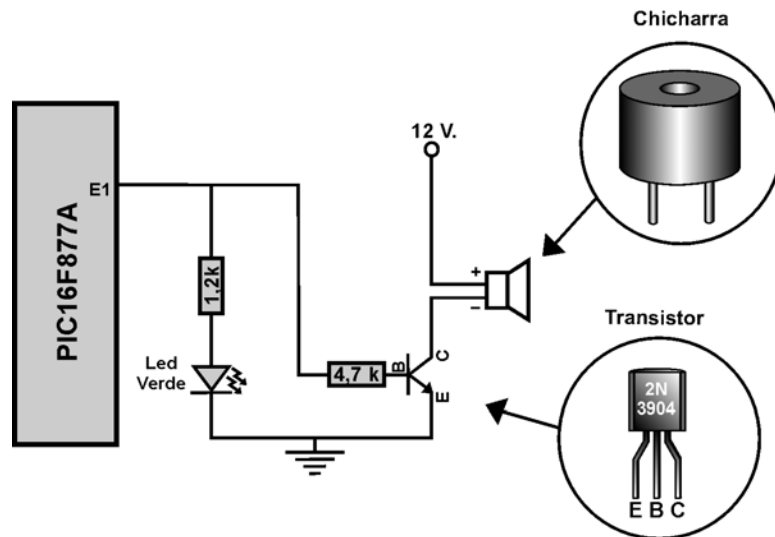


Figura. 3.11. Conexión de la alarma auditiva y visual

Configuración de los sensores fin de carrera. Estos sensores están conectados a los puertos D0, D1, D2 y D3 del microcontrolador, respectivamente. Cada uno está configurado como sensor normalmente abierto, esto quiere decir que todo el tiempo están enviando 5 V DC hacia el PIC; el momento que un sensor es alcanzado, cambia su circuito interno y envía un cero lógico, lo cual, como está establecido en el programa del microcontrolador, hace que se active la alarma y se paren los motores. Cada sensor debe tener conectado una resistencia de protección. Observe la Figura. 3.12.

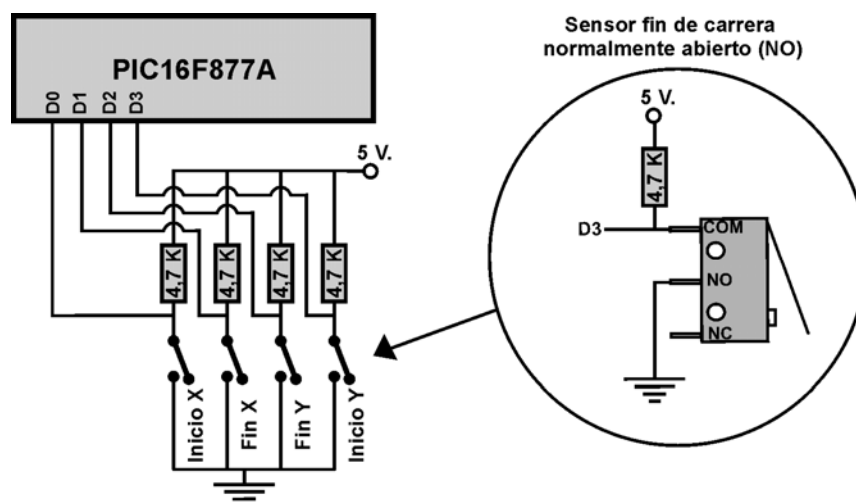


Figura. 3.12. Conexión de los sensores fin de carrera

Interfase Robot – Computador. Para poder controlar el robot desde un computador se realiza una comunicación serial entre ellos. Se conecta el Pin TX del puerto serial Db9 del computador al puerto E2 del microcontrolador, sin olvidar por su puesto, una resistencia de por medio para protección. Observe la Figura. 3.13.

Todo el trabajo realizado en el computador se codifica en datos que son transmitidos al microcontrolador, el cual los decodifica y ejecuta las acciones que correspondan.

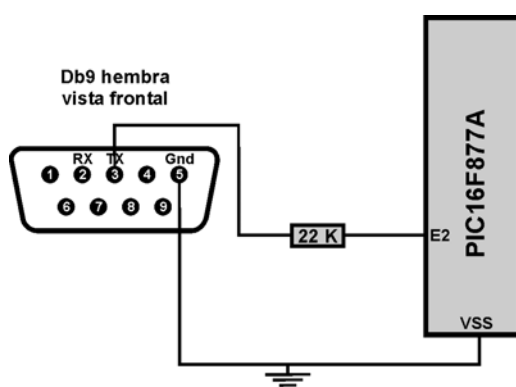


Figura. 3.13. Conexión del microcontrolador al puerto serial de un computador

Configuración del panel de control. Como se puede ver en la Figura. 3.14. y constatar con la Tabla. 3.1., se asignó solo ocho pines del microcontrolador para dieciséis botones del panel de control, un pino por cada columna de botones y un pino por cada fila de botones, de modo que cada botón del panel se identificará por la posición en la que se encuentra con respecto a filas y columnas, y adquirirá como nombre un número del 0 al 15 que será lo que el microcontrolador reconozca. Por ejemplo, si se presiona la tecla Arriba, se está presionando la posición Fila A y Columna 2, a la cual se le asignó el nombre “2”, que es lo que reconocerá el microcontrolador y ejecutará la acción. Esta es una técnica conocida como Barrido de Teclas.

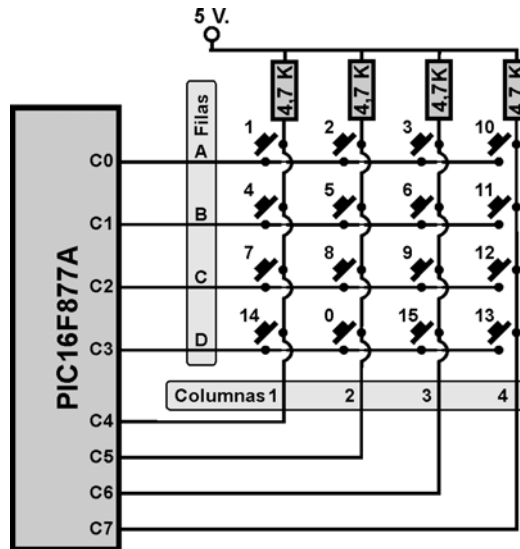


Figura. 3.14. Conexión de los botones del panel de control

La siguiente tabla (Tabla. 3.2.) muestra los botones del panel de control con sus respectivas posiciones y su nombre asignado para ser reconocido por el microcontrolador.

Tabla. 3.2. Botones del Panel de control

Botón	Posición	Nombre asignado
Diagonal Arriba Izquierda	Fila A – Columna 1	1
Arriba	Fila A – Columna 2	2
Diagonal Arriba Derecha	Fila A – Columna 3	3
Menú	Fila A – Columna 4	10
Izquierda	Fila B – Columna 1	4
Home	Fila B – Columna 2	5
Derecha	Fila B – Columna 3	6
Movimiento Paso a Paso	Fila B – Columna 4	11
Diagonal Abajo Izquierda	Fila C – Columna 1	7
Abajo	Fila C – Columna 2	8
Diagonal Abajo Derecha	Fila C – Columna 3	9
Movimiento Corrido	Fila C – Columna 4	12
Star	Fila D – Columna 1	14
Stop	Fila D – Columna 2	0
Enter	Fila D – Columna 3	15
Subir – Bajar Herramienta	Fila D – Columna 4	13

3.2.2. Etapa de potencia y alimentación del robot

Todo el sistema será alimentado por una fuente que brinda 12V y 2A, suficiente para que trabajen los motores y la parte electrónica del robot. A su vez, la fuente mencionada será alimentada por 110 V AC a 60Hz.

Se instalará un switch de encendido (ON / OFF) a la salida de la fuente alimentadora, es decir, así esté alimentada la fuente, si no se pone el switch en la posición de encendido, no pasara energía al robot. Esto, a mas de servir como prevención, permite apagar el sistema en cualquier momento, ya sea por emergencia o porque lo requiera el usuario.

Los tres motores de pasos que utiliza el robot se configura de manera idéntica, se alimentan directamente de la fuente de 12V y sus bobinas trabajan dependiendo de la señal que manda el microcontrolador (Figura. 3.15.). Esta señal es de 5V y pasa por una resistencia de 4,7 k Ω con lo cual limita la demanda de corriente del microcontrolador a 1 mA, esta nueva señal llegará a la base de un transistor TIP110, lo que permitirá el paso de corriente del colector al emisor de modo que cierra el circuito de la bobina correspondiente.

Para proteger del efecto inductivo que genera el motor, se instala un diodo 1N4007 de 2A entre la alimentación del motor y el colector del transistor.

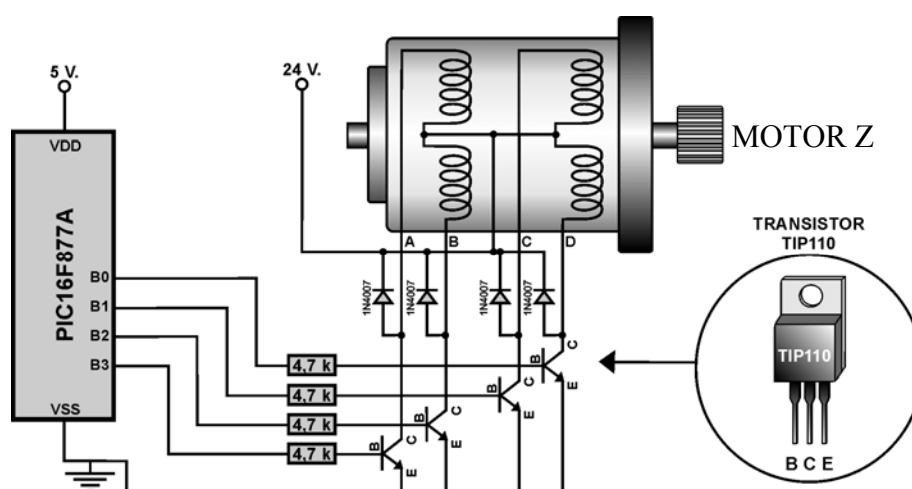


Figura. 3.15. Configuración de los motores del robot

CAPÍTULO 4

PROGRAMACIÓN DEL MICROCONTROLADOR DEL ROBOT

4.1. INTRODUCCIÓN

En este capítulo se diseña el software que será grabado en el microcontrolador, para ello se utiliza el programa de computadora PicBasic Pro, que es uno de los más sencillos programadores de microcontroladores que existen en el mercado.

El microcontrolador escogido para el robot objeto de este proyecto, como ya se habló en el capítulo de diseño, es el PIC 16F877A. Una vez más se dirá, este es el cerebro del robot, de él dependen todos sus movimientos, así como también los despliegues de mensajes en pantalla y la activación o no de alarmas; resumiendo, cualquier pulso o señal se registrará en el microcontrolador y de acuerdo a como se diseño el programa, ejecutará o no las acciones.

El programa diseñado para el robot, tendrá la potestad de permitir manejo manual por medio de los botones del panel de control o manejo automático, desde un computador.

4.2. PROGRAMA DEL MICROCONTROLADOR

Para hablar del programa que estará cargado en el microcontrolador, se debe primeramente entender que este es un conjunto de subprogramas, es decir, se va a desarrollar un programa principal que contiene varios subprogramas que permiten el

control del robot y cada uno de estos subprogramas trabajarán haciendo una llamada a otros pequeños programas como son: el programa que mueve el robot hacia la izquierda, el programa que mueve el robot hacia la derecha, el programa que levanta o baja la herramienta, etc. Incluso habrá un programa para cada sensor y otro para el LCD.

Como ya se mencionó, el desarrollo del programa del microcontrolador se lo hará en PicBasic Pro, software distribuido por MicroCode, en el cual, al igual que en todo programador, el primer paso es definir variables y asignarlas a un pin del microcontrolador si es el caso, además de darles un nombre específico que reconozca el programa, por ejemplo, con la ayuda de la Tabla. 3.1. se sabe que en el pin #40 del PIC (Puerto B7), está conectada la bobina A del Motor X, entonces en el programa se declara: `BXA VAR PORTB.7`. Esto quiere decir que dentro del programa, cada vez que se mencione BXa se hace referencia a una variable, esta es la bobina A del Motor X que está conectada al puerto B7 del PIC, de tal manera que se podrá activar o desactivar dicha bobina con tan solo escribir dentro del programa `Hight BXa` o `Low BXa`, respectivamente.

Otro ejemplo, quizá más fácil de entender sería el caso de la chicharra, a esta se la ha declarado en el programa como `BIP VAR PORTE .1` que significa que existe una chicharra conectada en el puerto E1 del PIC y se la ha llamado BIP, de modo que si el programa dice encender BIP la chicharra sonara hasta que el programa diga apagar BIP.

Como el par de ejemplos mencionados, todas y cada una de las partes del proyecto están declaradas dentro del programa, así como también variables auxiliares que si bien no están conectadas al PIC, nos sirven en el programa para realizar determinadas tareas como es almacenar un valor temporal o realizar un conteo.

Es sumamente importante tener clara la configuración del panel de botones, es decir, identificar perfectamente bien cada botón, cual es su función, su posición y a que pin del microcontrolador esta conectado, pues una vez que el robot esté listo, ensamblado y programado, el usuario tendrá acceso únicamente al panel de botones y si hubo una falla en el programa, al presionar un botón determinado, puede que realice una acción no requerida; para no tener complicaciones se debe revisar la Tabla. 3.2. antes y durante el desarrollo del programa, esta tabla contiene todo lo referente a los Botones.

4.2.1. Programa Principal

Una vez energizado el sistema, arranca el programa principal, lo primero que este hará es que el robot tome la posición inicial (0,0,0) mediante la llamada al subprograma ORIGEN y posterior a esto llamará al programa de menús para desplegar en pantalla (LCD) un primer menú de opciones, aquí, utilizando Flecha Arriba y Flecha Abajo, escogeremos: Control Manual, Control Computarizado, Imprimir Memoria o Figuras.

Cada uno de los menús mencionados hará la llamada a otros submenús que permitirán escoger las distintas opciones que brinda el robot.

Cabe mencionar que los botones del panel de control no trabajan igual dentro de todos los menús. Dependiendo del subprograma en el que se encuentre el robot, algunos botones pueden o no tener una acción asignada.

En la siguiente figura (Figura. 4.1.) se puede apreciar como está estructurado el programa principal del microcontrolador.

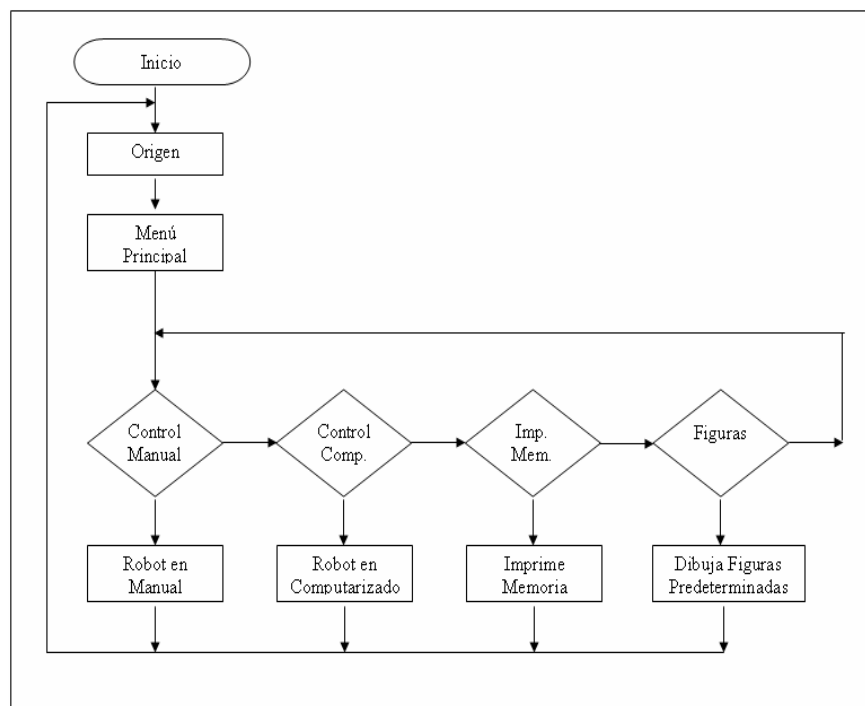


Figura. 4.1. Diagrama de flujo del Programa Principal

En el ANEXO B se presenta el código completo del programa, diseñado en PicBasic Pro, grabado en el microcontrolador.

4.2.2. Programa para el Control Manual

Si se opta por Control Manual en el menú principal, automáticamente se habilita un subprograma que permite que el robot se mueva libremente sobre el área de trabajo utilizando el panel de botones.

Cada vez que se oprime un botón, se está haciendo una llamada a otro subprograma, por ejemplo, si se oprime flecha abajo se está llamando al subprograma ABAJO, este al ser activado energiza las bobinas del motor Y en una secuencia programada que hace que de un giro en sentido positivo, esto quiere decir que la herramienta recorre hacia abajo en el plano de trabajo.

Si se da el caso en que se recorre la herramienta del robot hasta el tope del plano, allí se encontrará con un sensor fin de carrera que al ser activado, llama a otro subprograma que dependiendo de que sensor fue alcanzado, detiene el motor que esté llevando la herramienta en su dirección y activa la chicharra (Alarma auditiva). Con los mismos botones del panel se puede retroceder el movimiento.

La acción de elevar o descender la herramienta está a cargo del motor Z y para ello se ha designado un solo botón (Subir / Bajar), debido a que mediante el programa SUBEBAJA se ha conseguido que el motor Z se mueva en sentido positivo, o sea que baje la herramienta, solo si antes tuvo un movimiento negativo o viceversa, esto quiere decir que al pulsar el botón Subir / Bajar, el programa compara la posición de la herramienta y si esta se encuentra abajo la sube caso contrario, si se encuentra arriba la baja hacia el plano, de modo que no se necesita tener un botón para elevar la herramienta y otro para bajarla. Esto además, evita cometer el error de elevar la herramienta cuando ya está arriba o tratar de bajarla cuando se encuentra sobre el plano, lo que remordería los piñones del motor y reduciría la precisión y repetibilidad del robot.

La siguiente tabla (Tabla. 4.1.) presenta la acción que se ha designado, según el subprograma, a cada botón del panel, dentro del Control Manual.

Tabla. 4.1. Botones del Panel de Control dentro del menú Control Manual

Botón	Subprograma	Acción
Flecha arriba	ARRIBA	Mueve el motor Y en sentido (-)
Flecha abajo	ABAJO	Mueve el motor Y en sentido (+)
Flecha derecha	DER	Mueve el motor X en sentido (+)
Flecha izquierda	IZQ	Mueve el motor Y en sentido (-)
Diagonal arriba derecha	DIADARRI	Mueve el motor Y en sentido (-) y el X en sentido (+)
Diagonal arriba izquierda	DIAIARRI	Mueve el motor Y en sentido (-) y el X en sentido (-)
Diagonal abajo derecha	DIADABAJ	Mueve el motor Y en sentido (+) y el X en sentido (+)
Diagonal abajo izquierda	DIAIABAJ	Mueve el motor Y en sentido (+) y el X en sentido (-)
Subir / Bajar	SUBEBAJA	Mueve el motor Z en sentido (+) o (-) dependiendo el caso.
RUN	TECLAUNO	Activa el modo Movimiento Corrido
PAP	TECLADOS	Activa el modo Movimiento Paso a Paso
Home	ORIGEN	Eleva la herramienta y la lleva a la posición (0,0)
Stara	No asignado	No asignado
Stop	No asignado	No asignado
Enter	No asignado	No asignado
Menú	PROG2	Regresa al Menú Principal

Por otra parte, con respecto a las opciones de movimiento corrido y paso a paso, que brinda el robot, no es mas que utilizar o no un programa de antirrebote de teclas. Mas claro, cuando se oprime el botón PAP (Paso a Paso), se llama a un subprograma que está diseñado para que los motores avancen un paso por cada pulso, a esto se le llama antirrebote, de modo que, aunque se mantenga aplastando un botón, este reaccionará solo una vez, hasta que se lo suelte y se lo vuelva oprimir, lo contrario pasa si se activa el botón RUN (desplazamiento libre), este botón llama a otro subprograma que permite que los motores se muevan libremente mientras un botón de flecha esté oprimido.

4.2.3. Programa para el Control Computarizado

Si la opción escogida es la de Control Computarizado, el programa hará que los botones del panel se desactiven y se habilite el puerto de comunicación serial que posee el robot. A partir de ese momento, el robot se encuentra en la espera de los datos que a este se transmitan desde un computador que deberá tener cargado y ejecutado el programa específico del robot para uso desde un computador.

Cuando se finalice el Control Computarizado, se mandará un último dato al microcontrolador que reactivará su Panel de Control, deshabilitará el puerto de comunicación serial y nuevamente presentará el Menú Principal.

4.2.3.1. Programa diseñado para el uso del robot mediante computadora

La siguiente figura (Figura. 4.2.) muestra la pantalla a ser presentada en el computador, esto facilitará el entendimiento.

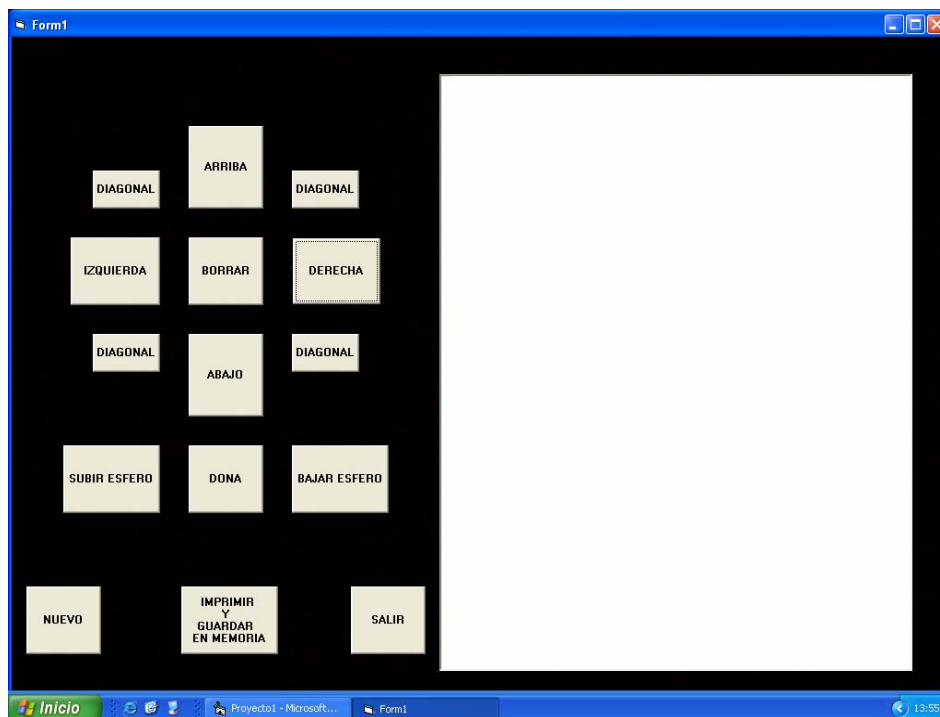


Figura. 4.2. Pantalla principal del programa para Control Computarizado

Como se habló desde el principio del proyecto, el robot debe cumplir el objetivo de poder ser manipulado manualmente o por medio de un computador. Para este efecto se ha desarrollado un programa en Visual Basic, el cual presenta en pantalla un panel de control, similar al que se tiene físicamente, y un plano en el que se va a ir presentando cada movimiento que se realiza, véase la Figura. 4.2.

En el ANEXO B se presenta el código del programa de computadora, diseñado en Visual Basic, para el control computarizado.

4.2.3.2. Funcionamiento del Programa para computador

Como se observa en la Figura 4.2., se tiene un panel de control de varios botones, similar al panel de botones físico que posee el robot. Estos se activan con un click del Mouse del computador.

Cada vez que se presiona una flecha, a más de realizarse el trazo correspondiente en pantalla, se manda a guardar ese movimiento como un dato en la memoria del microcontrolador.

El momento en que se presiona la tecla **IMPRIMIR Y GUARDAR EN MEMORIA**, se manda una señal al microcontrolador que hace que lea los datos guardados en su memoria. Tomando en cuenta que cada dato es un movimiento, al leer estos datos, el robot lo que hará es reproducir el gráfico que se haya diseñado en el computador.

El diseño permanecerá en la memoria del microcontrolador y podrá ser impreso las veces que así se requieran hasta cuando se lo remplace con otro diseño.

Se cuenta también con el botón **NUEVO** que es un reset dentro del programa, este hace que se borre el diseño presentado en pantalla y se aliste para realizar uno nuevo.

Por último, con el botón **SALIR** se manda un dato que el microcontrolador reconocerá y finalizará la opción Control Computarizado, reactivando el Panel de Control físico y deshabilitando la conexión al computador.

4.2.4. Programa para la opción IMPRIMIR MEMORIA

Al acceder a la opción Imprimir Memoria, el programa principal llama a un nuevo subprograma que tendrá como objeto leer los datos grabados en la memoria del microcontrolador, estos datos son los que previamente se cargaron en el modo Control Computarizado, es decir, el microcontrolador lee los datos y ejecuta las acciones correspondientes, dibujando el ultimo diseño que se haya realizado en el modo computarizado.

Se tiene aquí la ventaja de no necesitar el computador para imprimir un diseño previamente creado, las veces que así se requiera.

Una vez terminada la impresión, el subprograma que controla esta tarea lleva nuevamente al menú principal.

4.2.5. Programa para la opción FIGURAS

Esta opción hace el llamado a un subprograma similar al de Imprimir Memoria, con la diferencia de que en este caso presenta un nuevo submenú con figuras previamente creadas y grabadas en el microcontrolador. La diferencia está en que estas figuras no fueron diseñadas en el modo computarizado sino que han sido creadas en el programa cargado en el microcontrolador.

Para imprimir una figura predeterminada, no se necesita estar conectado a un computador ni antes ni después de la impresión, simplemente se debe ir hasta el submenú de figuras y oprimir la tecla ENTER posicionándose en la figura que se quiera ver impresa.

Una vez terminada la impresión, el subprograma envía al submenú de figuras y si se quiere salir completamente, se oprime la tecla MENU del panel de botones y este llevará, como en todos los casos, al menú principal.

Las figuras predeterminadas se las realiza en el programa MicroCode. La desventaja que presenta es que para realizar una nueva se debe volver a programar todo el PIC, lo que

implica desmontarlo y trasladarlo al grabador. Si bien, este es un proceso largo, se recompensa con la ventaja de tener una figura predeterminada permanente en la memoria del PIC.

CAPÍTULO 5

IMPLEMENTACIÓN DEL ROBOT

El presente capítulo detalla la construcción de cada pieza del robot y a su vez el ensamblaje entre ellas, de modo que se logra apreciar el conjunto completo, la parte mecánica y la electrónica enlazadas a su vez por el software diseñado, grabado en el microcontrolador. El resultado será entonces el robot cartesiano, objeto de este proyecto.

5.1. MESA DE TRABAJO

El primer paso en la construcción del robot es la elección del material de la mesa de trabajo. Por factor peso y sobre todo por la facilidad de cortar, clavar y atornillar, se elige un tablero de MDF con las medidas establecidas en el capítulo de diseño (80x70cm).

5.2. EJE Y

Este al igual que todos los ejes que se utilizan en este robot es de un material conocido como acero plata, ideal para el trabajo pues es muy liso, lo que permite que sobre el se desplacen otras piezas mediante bocines.

El eje Y está representado por dos ejes de 40cm de largo y 1cm de diámetro cada uno, se ubican paralelamente y de forma fija sobre el tablero, haciendo las veces de rieles sobre los cuales se transportará una plataforma que será el eje X.

Como no se puede hacer correr dicha plataforma directamente sobre los ejes, es necesario instalar en cada eje una pieza plástica a manera de soporte, estas piezas tienen un bocín que les permite sujetarse a los ejes. En la Figura. 5.1. se muestran dichos ejes con sus respectivos soportes y bocines.



Figura. 5.1. Ejes Y con respectivas piezas plásticas y sus bocines

Los dos ejes estarán interconectados por un tercer eje horizontal de 40cm de largo y 0.8cm de diámetro, el cual posee piñones en sus extremos (Véase la Figura. 5.2.).

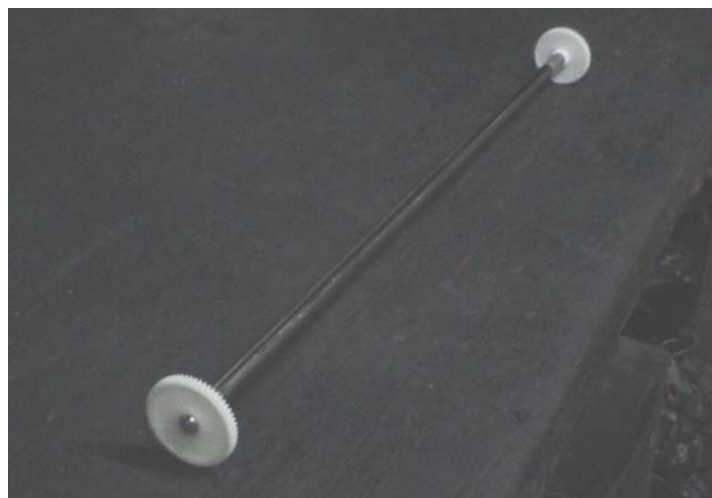


Figura. 5.2. Eje horizontal con piñones

Este eje horizontal no va a quedar fijo, estará girando pues en uno de sus piñones engranará un motor, de modo que se necesitará montarlo en una pieza que le permita dicho movimiento. La Figura. 5.3. muestra esta pieza y el eje desmontado.

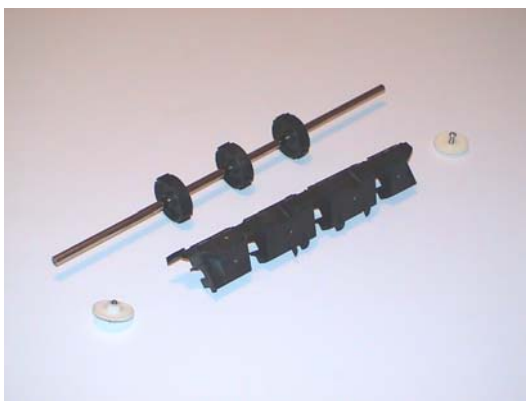


Figura. 5.3. Eje horizontal desmontado de la pieza base que lo permitirá girar

Se puede observar que en el eje se han colocado una especie de ruedas que le permitirán girar y a la vez lo mantendrán sujeto a la pieza.

En la siguiente figura (Figura. 5.4.) se puede apreciar el eje ya con sus piñones y ensamblado; a este conjunto se lo llamará en adelante Eje Auxiliar Horizontal. Cabe recalcar que la pieza plástica estará atornillada al tablero y es la primera parte del robot que se fija, es decir de ella dependerán la ubicación del resto de partes.



Figura. 5.4. Eje Auxiliar Horizontal

La Figura. 5.5. presenta los ejes Y y el eje Auxiliar Horizontal montados en el tablero. Observando este conjunto, se puede ir dando cuenta del área de trabajo que tendrá el robot, aproximadamente.

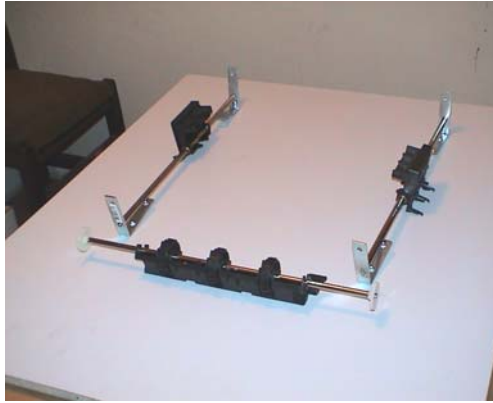


Figura. 5.5. Ejes Y paralelos y eje Auxiliar horizontal que gira

Una vez fijos los ejes paralelos y el eje horizontal, se coloca una banda en cada piñón y se la estira a lo largo de los ejes. Las plataformas de cada eje estarán sujetas a estas bandas, de modo que se debe calcular la distancia exacta para que cada banda quede perfectamente paralela a cada eje.

Una rodela móvil, montada en una pieza fija, permitirá templar las bandas, tal como se muestra en la Figura. 5.6.



Figura. 5.6. Bandas Transportadoras

El último paso en la construcción del eje Y es colocar el motor (Figura. 5.7.) que permitirá, por medio del eje horizontal, sus piñones y las bandas, mover hacia delante o hacia atrás los soportes que transportarán la plataforma que será el Eje X.

Cabe recalcar que el robot poseerá tres motores, uno para cada movimiento. Para facilitar el entendimiento, se los llamará Motor X, Motor Y y Motor Z.

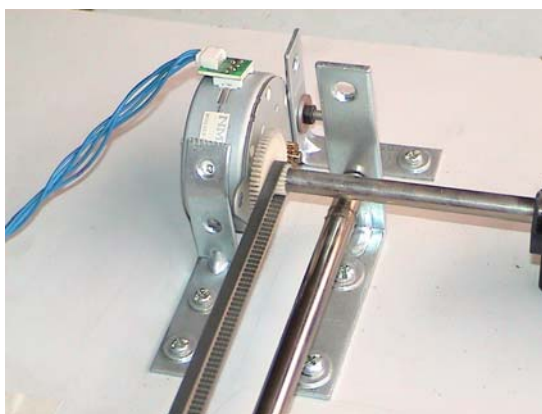


Figura. 5.7. Motor Y engranado al eje horizontal

El Motor Y es un motor paso a paso unipolar de cinco hilos, un común y los cuatro restantes correspondientes a las bobinas A, B, C y D; tiene un ángulo de giro de 7.5 grados, una resistencia en cada bobina de 30Ω y trabaja con una alimentación de 12V DC. Su forma física es cilíndrica como se aprecia en la Figura. 5.7., de 5.5cm de diámetro y 2cm de alto.

Observando la Figura. 5.8. y la Figura. 5.9., es posible entender como el robot describirá la trayectoria en Y. Simplemente el Motor Y hará girar el piñón del eje Auxiliar Horizontal, haciendo que la banda que tiene conectada se mueva y a su vez transporte el soporte que esta fijado en ella. Como el movimiento se transmite exactamente igual al piñón del otro extremo, la banda conectada a este también se moverá y transportará el otro soporte del otro eje.

Vale recalcar que para no tener problemas en el movimiento, los dos soportes deben estar perfectamente alineados, de esta forma se cumplirá el objetivo de que avancen los dos al mismo tiempo.

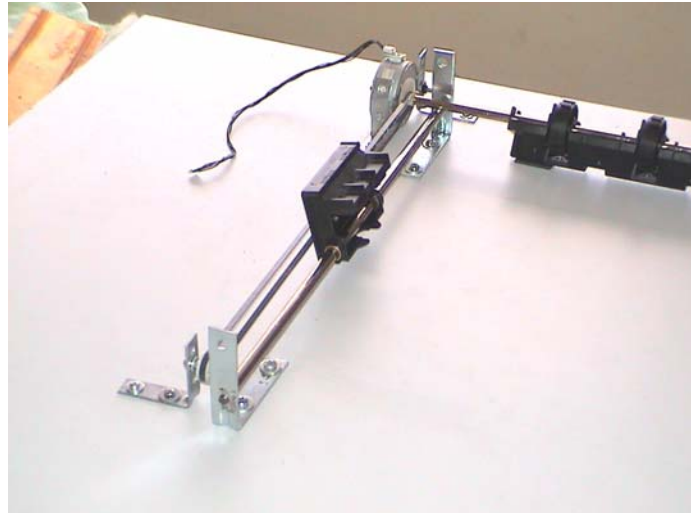


Figura. 5.8. Eje Y con motor

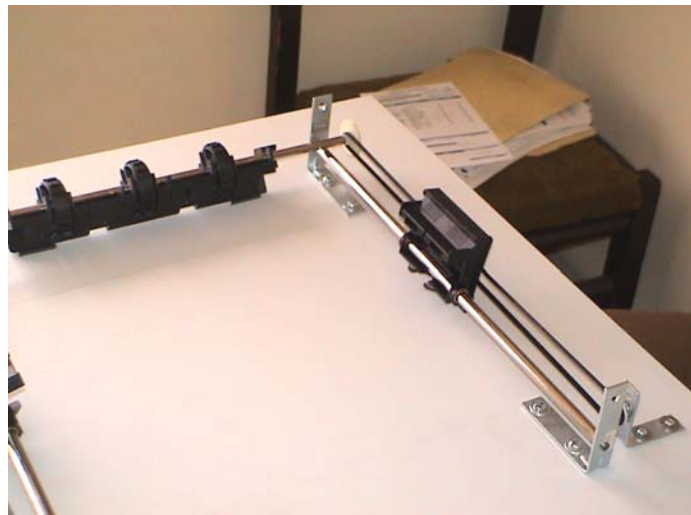


Figura. 5.9. Eje Y sin motor

A continuación se presentan tres figuras: Figura. 5.10., Figura. 5.11. y Figura. 5.12., en las que se puede apreciar los ejes Y completos visto desde distintos ángulos.

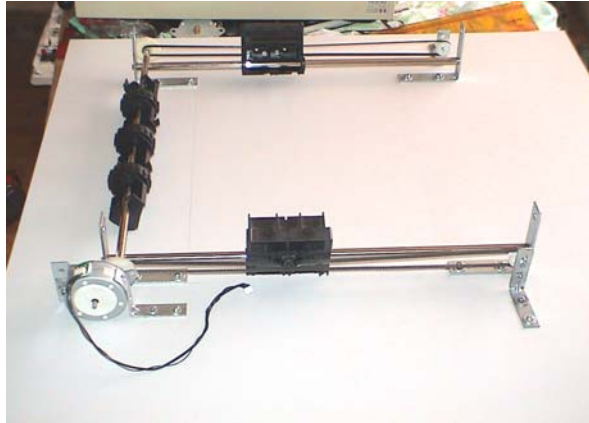


Figura. 5.10. Ejes Y (Vista Lateral)

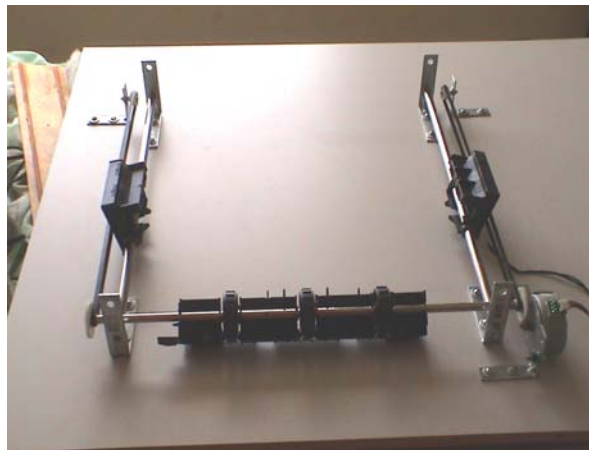


Figura. 5.11. Ejes Y (Vista Posterior)

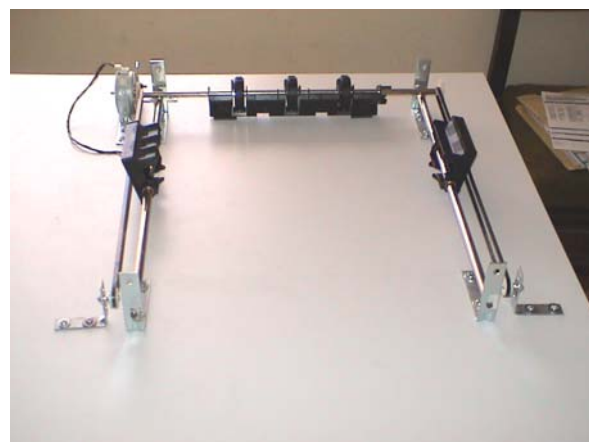


Figura. 5.12. Ejes Y (Vista Frontal)

5.3. EJE X

Como se mencionó antes, el eje X viene a ser una plataforma (Figura. 5.13.), la cual esta sujeta a los soportes de los ejes Y. Mide 44cm de largo y 12cm de ancho.

Dentro de la plataforma se instalan dos pequeños ejes fijos, cada uno con un piñón, el cual debe poder girar libremente. Como se ve en la Figura 5.13., estos se encuentran instalados uno a cada extremo de la plataforma.



Figura. 5.13. Eje X (plataforma)

Si se observa cuidadosamente en la figura anterior (Figura. 5.13.), se puede notar que el piñón de la izquierda tiene engranado a sus dientes un motor, el Motor X.

El Motor X es idéntico al Motor Y, con la diferencia que este tiene 6 hilos, de los cuales dos forman el punto común y los otros cuatro que son las bobinas A, B, C, y D.

El trabajo del Motor X es mover el piñón que está conectado a él, quien a su vez moverá una banda a la cual se va a sujetar el dispositivo que sostendrá la herramienta de trabajo del robot (Eje Z).

La siguiente figura (Figura. 5.14.), presenta la plataforma vista desde un ángulo que permite apreciar el Motor X.



Figura. 5.14. Vista frontal de la plataforma (Vista del Motor X)

La banda que transportará al Eje Z, conjunto con la herramienta de trabajo, sobre el Eje X, se encuentra templada entre los piñones antes mencionados, tal como se muestra en la Figura. 5.15.



Figura. 5.15. Banda templada en el Eje X

El Eje X (plataforma), se desplazará hacia adelante y hacia atrás por acción del Motor Y, el cual mueve los soportes donde estará montado todo este conjunto.

5.4. EJE Z

El Eje Z (Figura. 5.16.), es a su vez el dispositivo que sostiene la herramienta de trabajo, lo que lo convierte en la parte más importante del robot. Este se desplaza por toda el área de trabajo, de izquierda a derecha por medio de la banda del Eje X y de arriba hacia abajo, ya que todo el Eje X se mueve sobre el Eje Y, además permite ascender o descender la herramienta hacia el plano con la ayuda del Motor Z que está instalado en su interior. De esta forma se consiguen los tres desplazamientos: X, Y y Z.

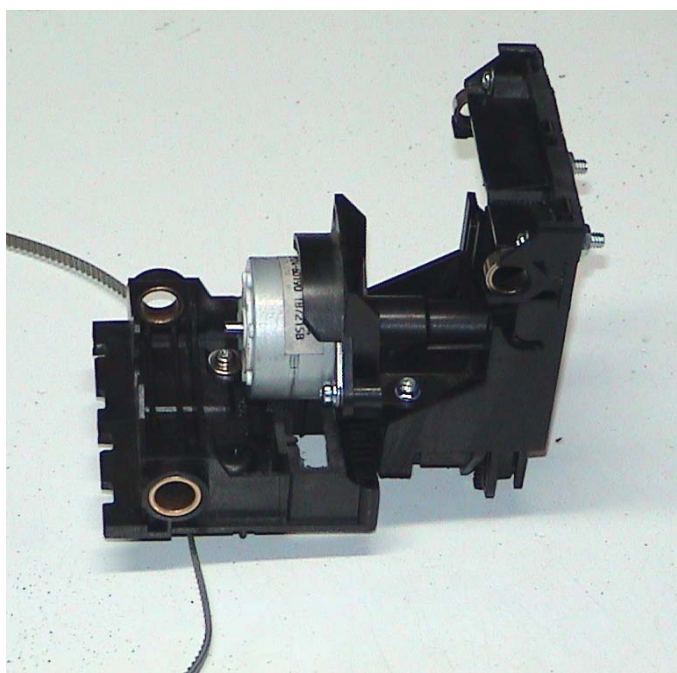


Figura. 5.16. Eje Z

El Motor Z es también es un motor de pasos unipolar de cinco hilos, de 7.5 grados de giro y alimentación de 12V. Es un poco más débil que los otros, pues tiene una resistencia en sus bobinas de 60Ω , pero es suficiente para realizar la tarea de bajar o elevar la herramienta de trabajo. También es cilíndrico, de 2cm de altura y 3.5cm de diámetro; tiene un piñón a manera de tornillo sin fin. Véase la Figura. 5.17.



Figura. 5.17. Motor Z

El tornillo sin fin antes mencionado, que puede ser apreciado en la figura anterior, gira y mueve un piñón que está engranado a él, quien a su vez desplaza una pieza plástica en sentido Z, donde irá sujeta una abrazadera metálica de 8x6cm que sostendrá el dispensador de tinta, tal como se muestra en la Figura. 5.18.

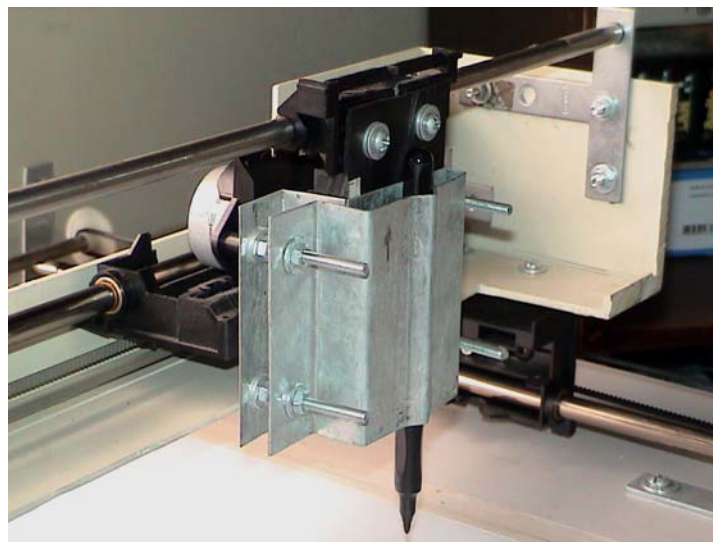


Figura. 5.18. Abrazadera metálica

El motor está configurado de tal manera que sube o baja la herramienta apenas 0,5cm, pues no se necesita más para lograr el objetivo de hacer o no contacto en el plano.

Tomando en cuenta que el buen desempeño del robot y la calidad de trabajo que brinde, dependerán de este último dispositivo, se necesita encontrar la mejor forma para que este se desplace sin errores, sin juego en el movimiento y sobre todo de manera segura, sin que se trabe. Esto fue posible instalando dos ejes a manera de riel elevado en la plataforma (Eje X), sobre los cuales se transporta la citada pieza.

Si se observa la Figura 5.16. que presenta todo el Eje Z, se pueden identificar cuatro bocines alineados de dos en dos. Por medio de ellos pasarán los ejes mencionados, tal como lo muestra la Figura 5.19 y la Figura. 5.20.

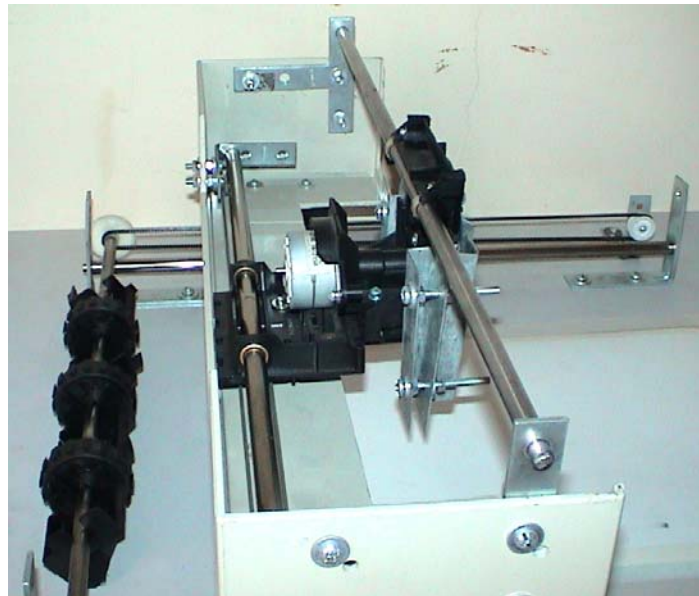


Figura. 5.19. Eje Z (Vista lateral)

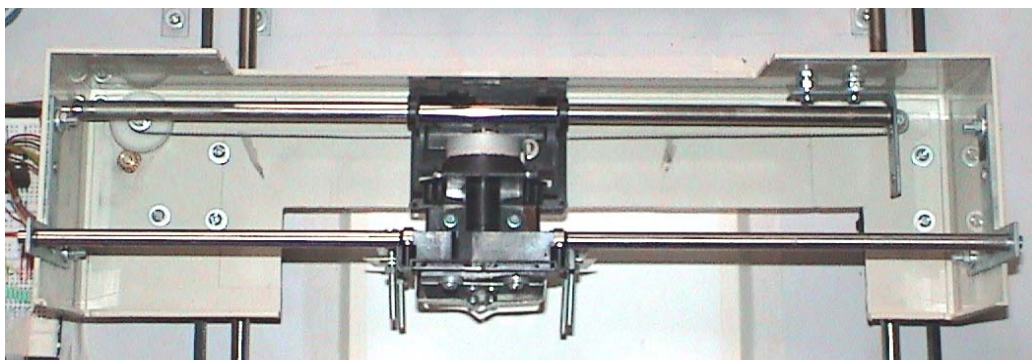


Figura. 5.20. Eje Z (Vista Superior)

El eje inferior de 40cm de largo y el superior de 44cm, están dispuestos de forma paralela, de modo que al desplazarse sobre ellos el Eje Z, lo hace de manera precisa y constante, logrando óptima calidad en las líneas de impresión.

5.5. SENSORES

Para evitar que el robot se salga de su área de trabajo y cause errores, se instalan sensores de fin de carrera al principio y al final de cada eje, de forma que si son alcanzados, generarán una señal que irá al microcontrolador y este parará los motores y encenderá una alarma auditiva.

El área definida de trabajo es 21cm en el sentido X y 30 en el sentido Y, es decir un formato A4.

La Figura. 5.21. presentada a continuación, muestra la conexión de los sensores fin de carrera.

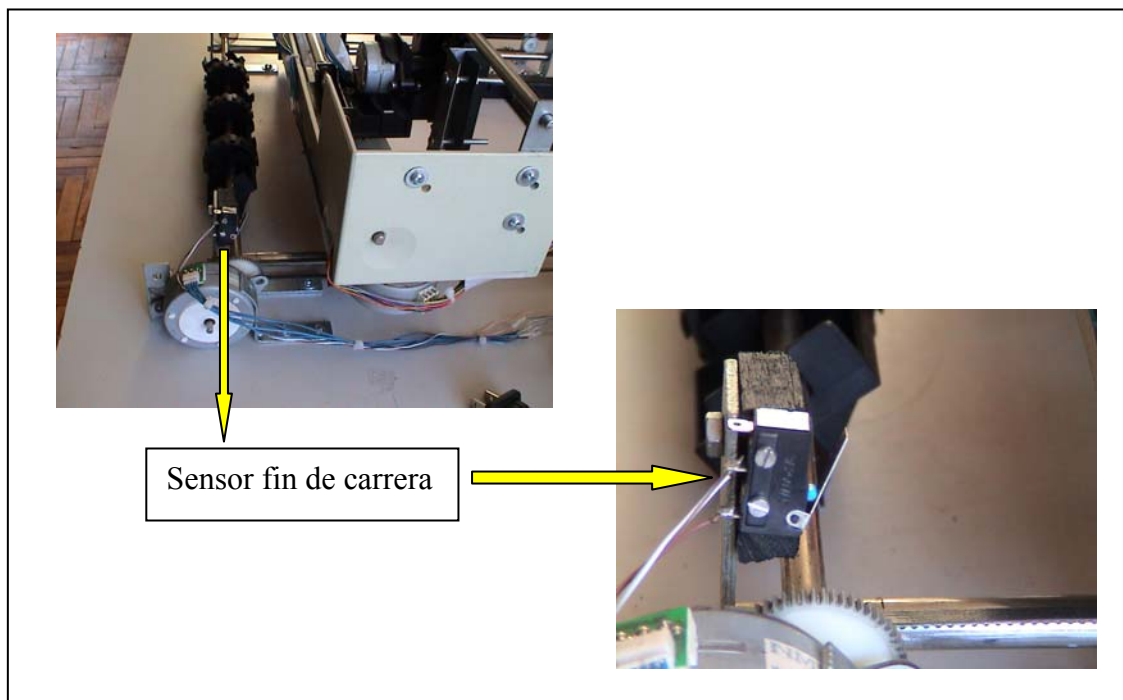


Figura. 5.21. Sensores fin de carrera

5.6. SUJECIÓN DE LA BAQUELITA

Para lograr que la baquelita no se mueva mientras se realiza la impresión sobre ella, se dispone de un sistema de sujeción a manera de marco. El lado superior está fijo sobre el tablero mientras que el inferior es móvil y ajustable por medio de pernos tipo mariposa instalados a sus extremos.

Como es fácil suponer lo que se hace es aprisionar la baquelita contra el lado fijo como muestra la Figura. 5.22.

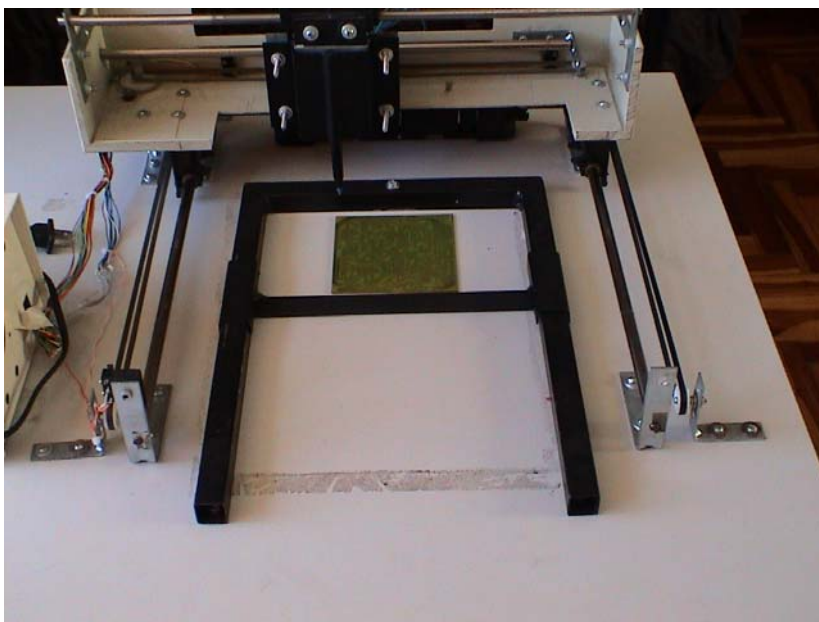


Figura. 5.22. Marco que sujeta la baquelita

5.7. CAJA PRINCIPAL

La Parte electrónica del robot, así como su fase de potencia y alimentación se encuentran dentro de una caja de latón que se la ha llamado Caja Principal. En su parte superior externa se encuentra instalado el LCD y el Panel de Botones y a un costado el puerto de comunicación serial y un swith de encendido tal como se muestra en la Figura. 5.23.



Figura. 5.23. Caja Principal

Cabe recalcar que ha sido fijada sobre la mesa de trabajo en un lugar estratégico a donde llegan fácilmente los cables de los elementos electrónicos externos como son los sensores y los motores.

5.8. PLACA ELECTRÓNICA DEL ROBOT

Esta placa, que contiene todo el hardware electrónico del robot, como ya se mencionó, se encuentra dentro de la Caja Principal conjunto con la fuente de alimentación, a ella van conectados todos los dispositivos electrónicos, pues aquí se encuentra el microcontrolador.

Se utiliza cable flexible para la conexión de los elementos que están fuera de la placa como es el LCD, el panel de botones, el puerto serial, el switch de encendido, los motores y los sensores fin de carrera, ya que al poner en marcha al robot, se estiran y recogen estos cables durante cada uno de sus movimientos y si se usa cable sólido, lo mas probable es que sufran fracciones internas.

La siguiente figura (Figura. 5.24.) muestra el diseño impreso de la placa, aquí se pueden apreciar las pistas del circuito.

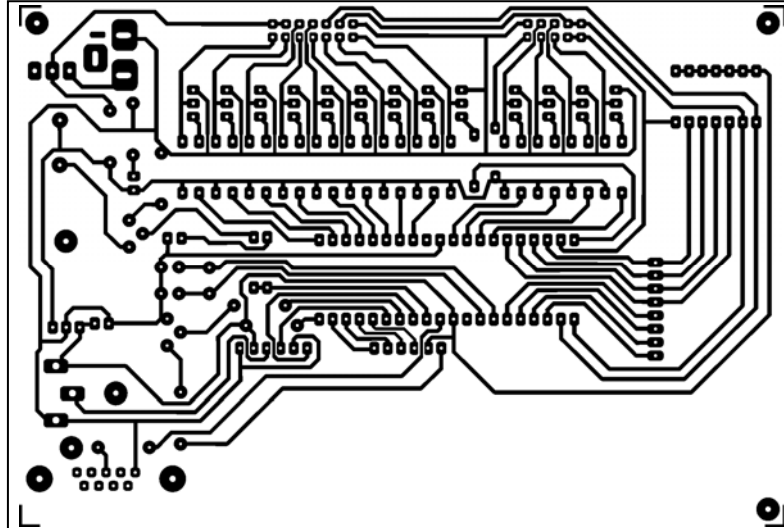


Figura. 5.24. Circuito Impreso

A continuación se presentan las figuras: Figura. 5.25., Figura. 5.26., y Figura. 5.27., en donde se puede observar la placa terminada y como se la instala en la Caja Principal.

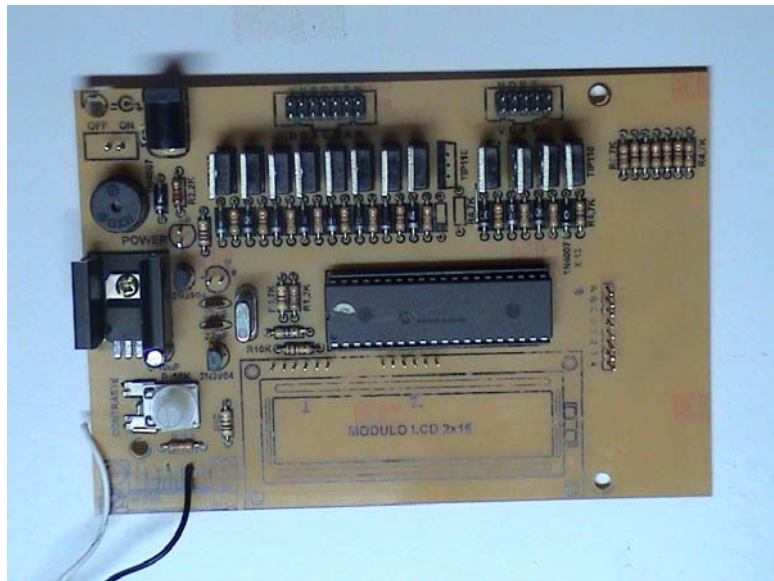


Figura. 5.25. Placa electrónica (Vista frontal)

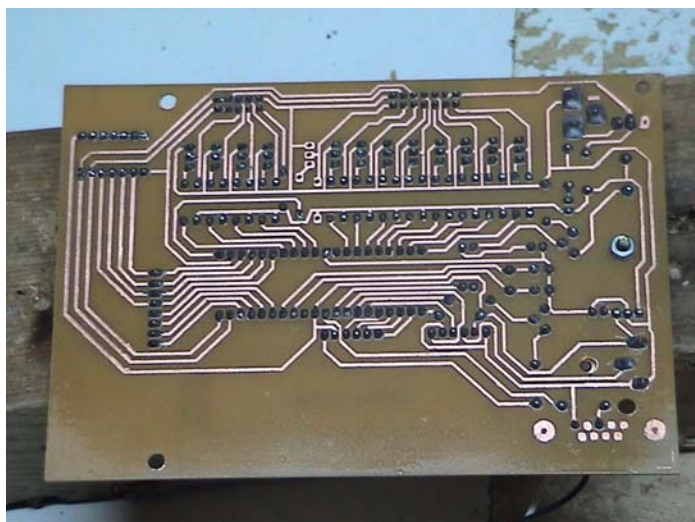


Figura. 5.26. Placa electrónica (Vista trasera)

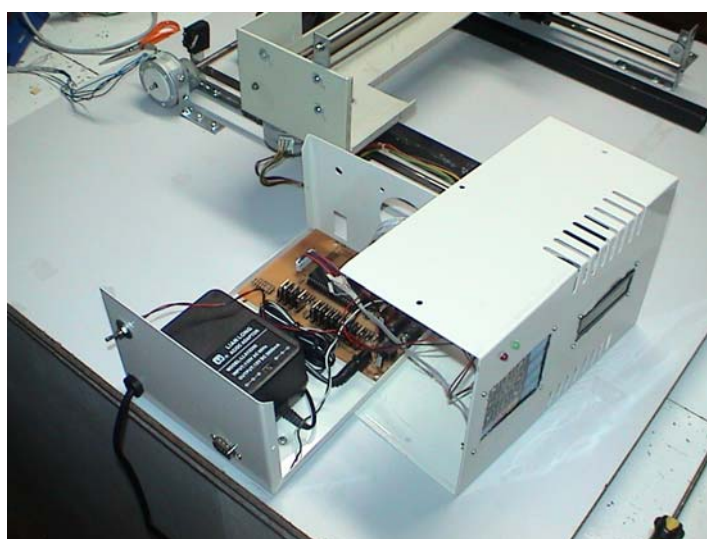


Figura. 5.27. Placa electrónica dentro de la Caja Principal

5.9. PANEL DE CONTROL Y PANTALLA LCD

Los botones que controlan el robot están fijos en un tablero acoplado a la Caja Principal al igual que la Pantalla LCD que despliega los mensajes del funcionamiento, tal como se muestra en la Figura. 5.23. presentada anteriormente.

A continuación se muestran dos figuras: la Figura 5.28. que presenta el Panel de Botones y el LCD instalados en la Caja Principal, visto desde adentro, y la Figura. 5.29. que permite ver un primer plano de los botones del Panel.

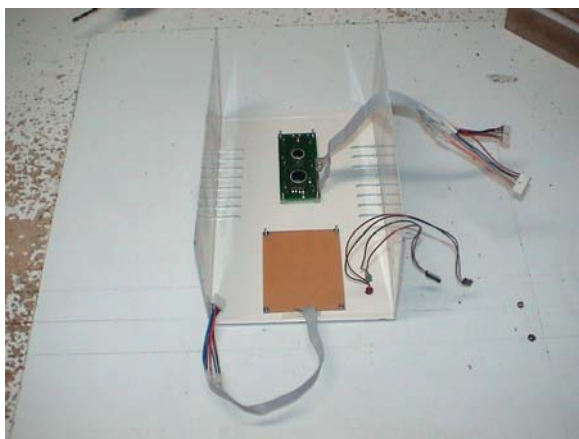


Figura. 5.28. LCD y Panel de Botones (Vista interna)



Figura. 5.29. Panel de Botones

5.10. PUERTO DE COMUNICACIÓN SERIAL

El puerto Db9 de comunicación serial que se utiliza para el control del robot por medio de computadora, se encuentra instalado a un costado de la Caja Principal, en el lado inferior derecho, como se presentó en la Figura. 5.23.

5.11. SWITCH DE ENCENDIDO (ON / OFF)

Como se habló en el capítulo de diseño, este switch está conectado a la salida de la fuente alimentadora de todo el sistema, de modo que dependerá de él que el robot se alimente de energía o no.

Se encuentra instalado a un costado de la Caja Principal como se mostró en la Figura. 5.23., en la parte superior izquierda.

El ANEXO A contiene un diagrama general del hardware del robot, allí se puede observar como están conectadas todas y cada una de las partes que lo componen.

CAPÍTULO 6

PRUEBAS Y RESULTADOS

El robot objeto de este proyecto trabaja en cuatro modos: Modo Manual, Modo Computarizado, Modo Imprimir Memoria y el Modo Figuras; en este capítulo se detallan las diversas pruebas a las que se ha sometido al robot para determinar si cumple o no con los objetivos planteados al inicio. A su vez, dichas pruebas han permitido validar su comportamiento y encontrar sus limitaciones.

6.1. PRUEBAS EN MODO MANUAL

6.1.1. Desplazamiento sobre el eje X

El desplazamiento de la herramienta de trabajo que realiza el robot en el sentido X, es decir de izquierda a derecha y viceversa sobre el plano de trabajo, no presenta inconveniente, ya sea que se esté usando el modo de desplazamiento corrido o paso por paso. Al ser una pieza sólida sostenida por dos ejes paralelos se mantiene firme durante sus trayectorias. Además, el motor utilizado es lo suficientemente fuerte para no trabarse durante el movimiento y el piñón que tiene instalado reduce su velocidad a una velocidad manejable, al mismo tiempo que aumenta su precisión.

Como fue planteado en el capítulo de diseño, el desplazamiento sobre el eje X es de 21cm, lo cual se cumple con la ayuda de los sensores fin de carrera instalados a los extremos.

6.1.2. Desplazamiento sobre el eje Y

El desplazamiento en el sentido Y, es decir de arriba hacia abajo y viceversa sobre el plano, presenta un único problema pero solucionable y es que deben posicionarse exactamente alineadas las bases que llevan la plataforma X sobre los ejes Y, caso contrario las líneas verticales toman un pequeño ángulo al final de la trayectoria sobre el plano, dicho de otra forma, los trazos en sentido Y no se presentan perfectamente rectos.

Debido a que el motor encargado del movimiento Y es igual al del movimiento X, inclusive en el piñón que tienen instalado, tanto la velocidad como el espacio recorrido por paso es idéntico en cualquier sentido, es decir, dando un paso a cada lado se obtiene un cuadrado perfecto.

El desplazamiento máximo que consigue el robot sobre este eje es de 30cm, lo cual, al igual que en el eje X, está controlado por los sensores fin de carrera instalados en sus extremos.

6.1.3. Desplazamiento sobre el eje Z

El movimiento en sentido Z (elevar o descender), es de apenas 3ml, suficientes para que la herramienta raye o no sobre el plano. Debe calcularse bien esta distancia de modo que la punta del dispensador de tinta haga contacto normal sobre la baquelita, lo necesario para que las líneas se impriman no muy gruesas ni muy delgadas.

Si se aprieta demasiado la punta contra el plano, al cambiar de sentido una línea no marca un ángulo recto como se espera y se corre el riesgo de forzar los motores, lo que haría trabar a las bandas, dando como resultado trazos no lineales.

El programar al robot para que utilice un mismo botón para elevar o descender la herramienta de trabajo ha marcado una gran ventaja, pues evita cometer el error de descender la herramienta cuando ya está sobre el plano o en el caso contrario, elevarla cuando ya se encuentra arriba, lo que descuadraría la herramienta de su eje a más de forzar el motor Z.

La siguiente figura (Figura. 6.1.), presenta una muestra de lo que se puede trabajar con el robot en el Modo Manual.

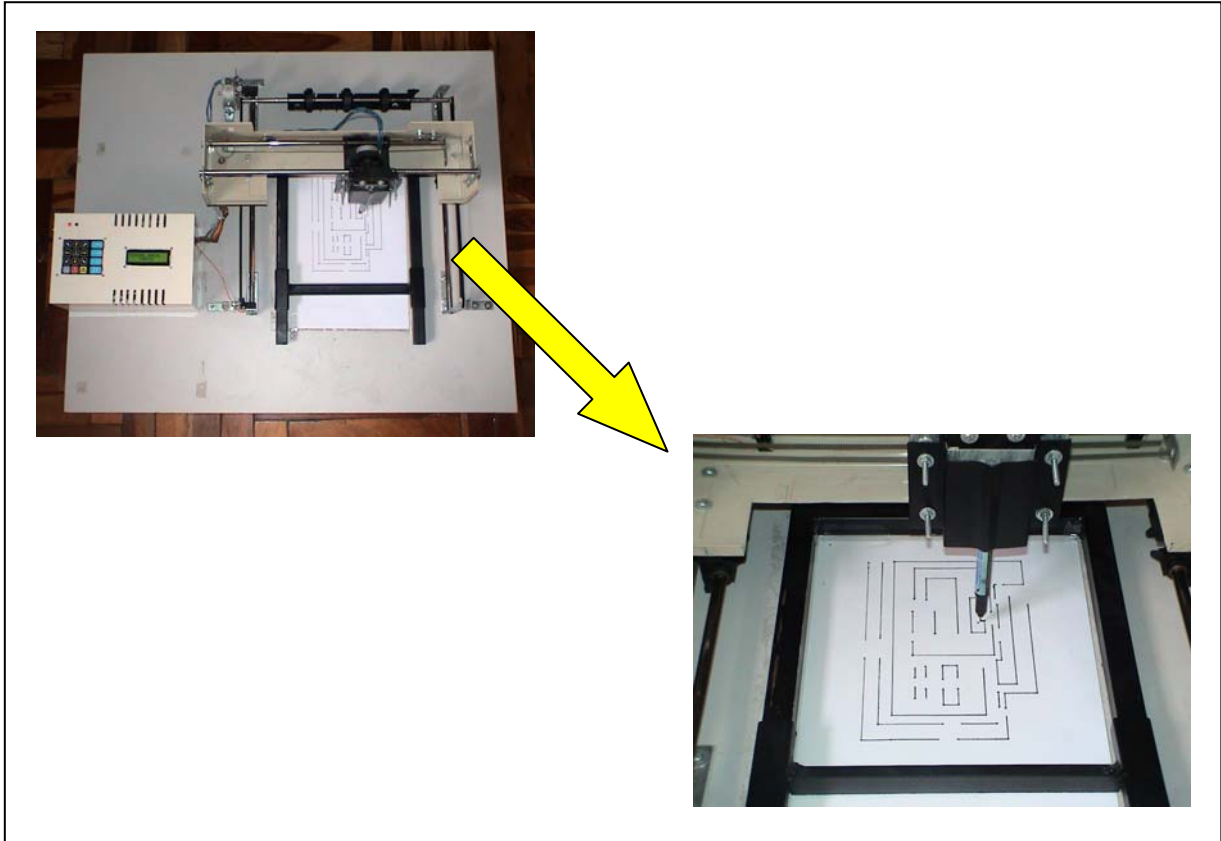


Figura. 6.1. Trabajo del Robot en Modo Manual

6.2. PRUEBAS EN MODO COMPUTARIZADO

Al trabajar el robot en modo computarizado se deshabilita por completo los botones del panel de control y se lo maniobra únicamente desde el computador al que esté conectado.

El programa para el manejo desde computador está diseñado para que el robot reproduzca los trazos que se presenten en pantalla. Este modo no tiene inconveniente, simplemente se trabaja un diseño en la computadora y se manda la señal al robot para que lo imprima.

Se debe revisar bien el cable serial que conecta el robot con el computador ya que si este llega a tener fisuras internas la señal emitida pasa de forma incorrecta o simplemente no se transmite, haciendo que el robot realice movimientos no requeridos.

Con el propósito de que el usuario se de cuenta que está transmitiendo correctamente los datos del computador hacia el robot, el programa cargado en el microcontrolador hará que suene un BIP y se encienda un LED cada vez que se recepte uno de ellos, es decir cada vez que se realice un trazo en pantalla, se generará un dato que se envía de inmediato a la memoria del microcontrolador, el cual encenderá un LED y hará sonar un BIP a manera de confirmación.

Las siguientes figuras (Figura. 6.2. y Figura. 6.3.), muestran el trabajo del robot en el modo computarizado, en ellas se puede apreciar como se ve el diseño en pantalla del computador e impreso en la máquina.

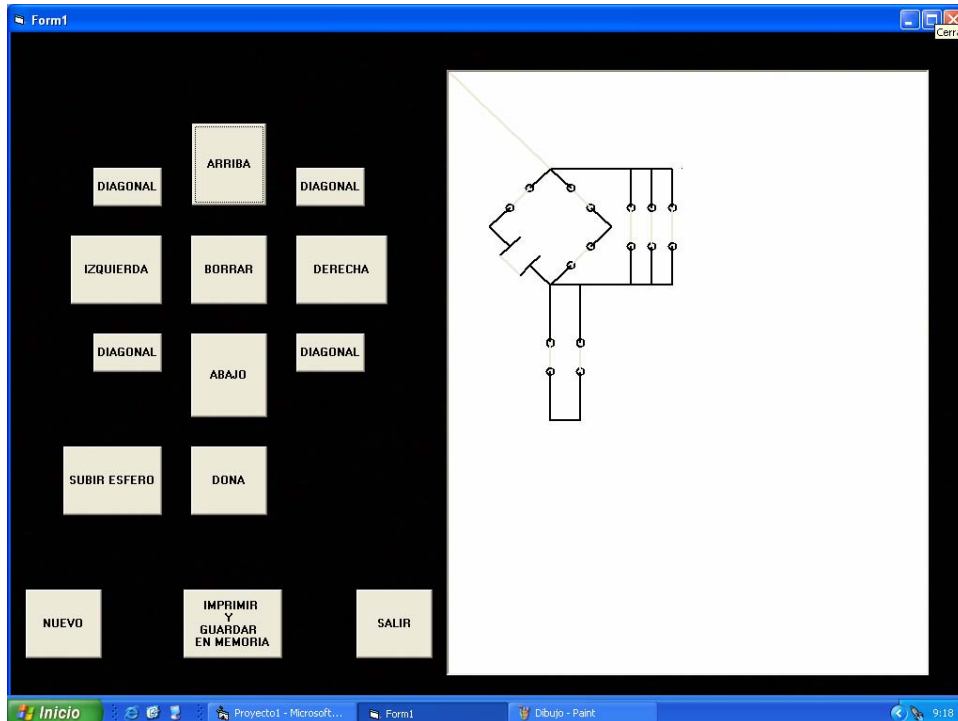


Figura. 6.2. Circuito Diseñado en el Computador

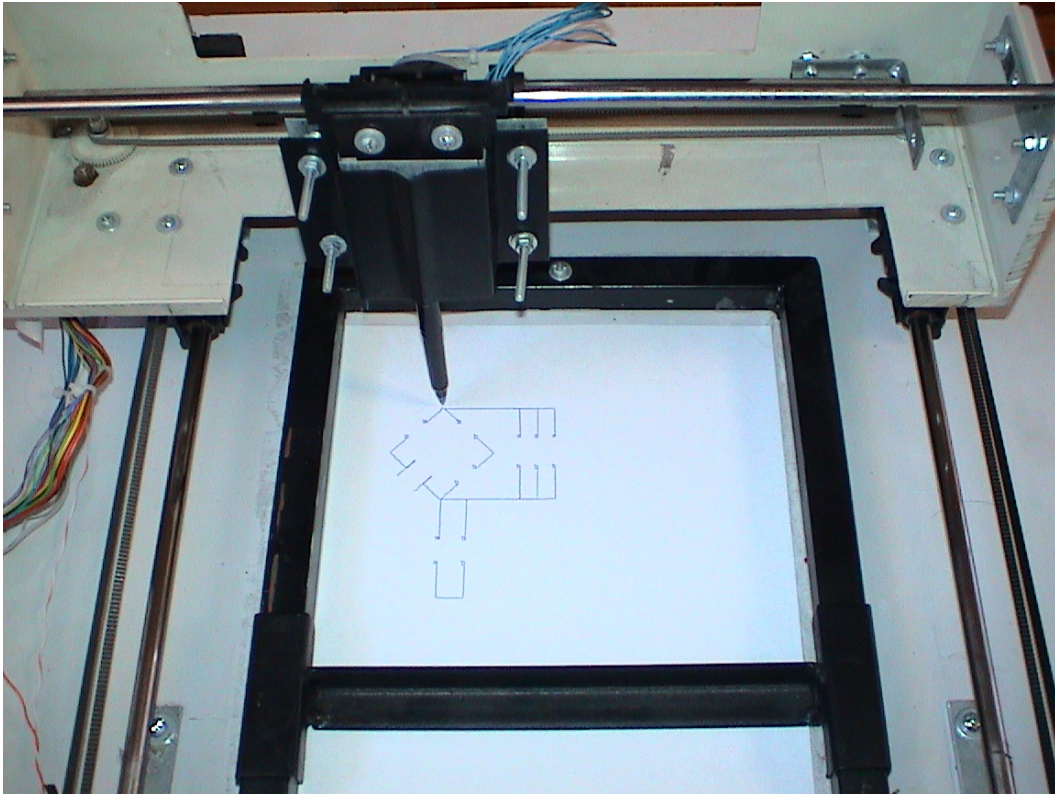


Figura. 6.3. Circuito Impreso desde el Computador

6.3. PRUEBAS EN MODO IMPRIMIR MEMORIA

El modo Imprimir Memoria realiza la acción de reproducir el último diseño grabado en la memoria del microcontrolador. Esto funciona sin ningún problema, incluso se ha diseñado el programa del microcontrolador para que al acceder a este modo, el robot pase primero por un reset que lo deja en la posición inicial (0,0,0) de tal forma que no se corre el riesgo de imprimir el diseño en un lugar del plano no deseado.

Lo único que se debe hacer es ingresar al Menú Imprimir Memoria y oprimir la tecla Enter del panel de control, sin olvidar en que punto se iniciará la impresión del diseño grabado con el objeto de colocar la baquelita en el lugar correcto.

6.4. PRUEBAS EN MODO FIGURAS

En esta opción, como se dijo en capítulos anteriores, se puede escoger entre varios diseños pregrabados en el microcontrolador para imprimirlos directamente. Lo único que se debe tomar en cuenta es que los diseños han sido creados para ser impresos a 3cm del margen superior y a 3cm del margen lateral izquierdo, esto con el objeto de colocar en el lugar correcto la baquelita en la que se quiera imprimir.

Al igual que en el modo Imprimir Memoria, el programa está diseñado para que lo primero que realice sea un reset o sea ir al punto origen (0,0,0), con lo que no hay que preocuparse de en que posición se encuentra la herramienta antes de iniciar la impresión.

En la siguiente figura (Figura. 6.4.), se puede observar, impreso por la máquina, uno de los diseños pregrabados y guardados en su memoria.

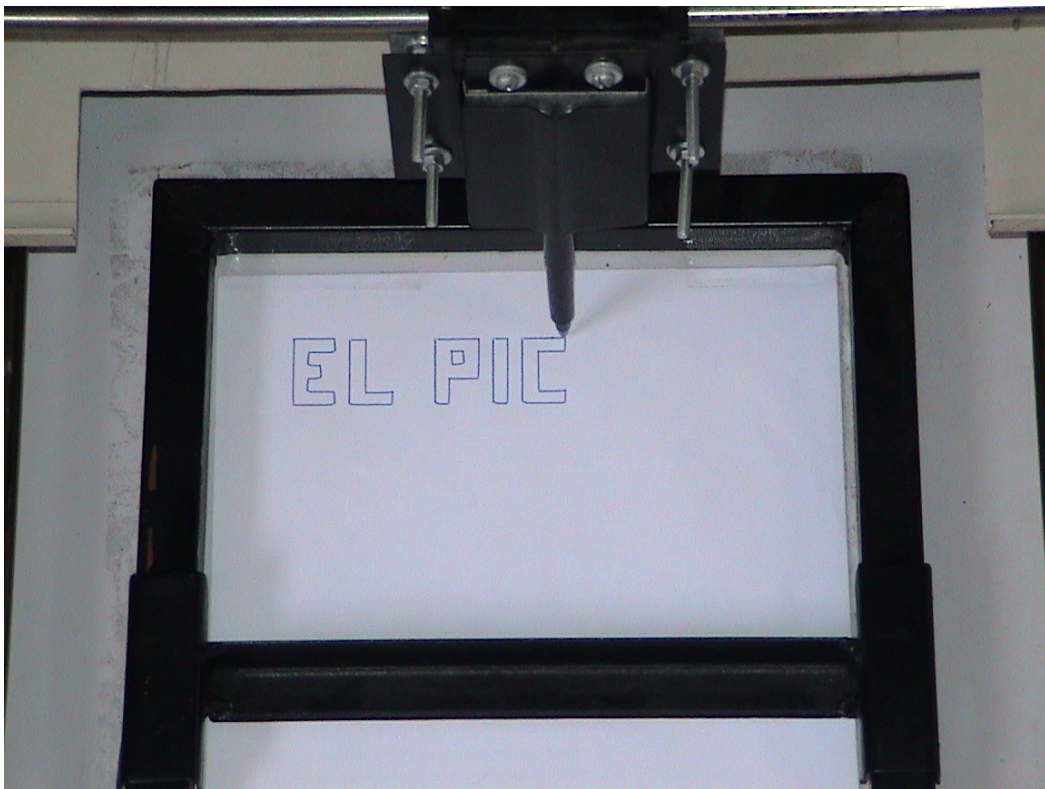


Figura. 6.4. Impresión en Modo Figuras

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

- Se logró a satisfacción el objetivo principal planteado en este proyecto que fue el de construir un robot electromecánico que imprima sobre baquelitas circuitos de electricidad y electrónica básica.
- El material utilizado para los ejes, conocido como acero plata, es adecuado para realizar el trabajo de transportar la herramienta ya que al ser suficientemente lisos permiten fácilmente desplazar piezas sobre ellos mediante bocines siendo mínima la resistencia por el rozamiento, más aún si son lubricados.
- La decisión de utilizar motores de pasos para la realización de este proyecto fue acertada ya que éstos presentan facilidades de programación y control, permitiendo así que los trazos que realiza el robot sean precisos y exactos.
- Se logró una estructura mecánica fuerte y suficientemente estable. El haber utilizado doble eje tanto en X como en Y, redujo por completo cualquier juego en el movimiento de la herramienta de trabajo.
- Se consiguió una repetibilidad aceptable combinando el buen manejo de los motores de pasos con la adecuada implementación de los ejes tanto X como Y.

- Utilizar bandas transportadoras dentadas evitó que los movimientos sean imprecisos. Una banda lisa tiende a resbalar y descalibraría el sincronismo de los motores.
- El modo computarizado brinda mejores resultados en cuanto al diseño de los circuitos ya que al graficar primero en pantalla se puede hacer correcciones antes de mandar a imprimir.
- El microcontrolador utilizado en el proyecto puede mantener en su memoria un diseño pregrabado aunque se desconecte la alimentación.
- Sin la ayuda de los sensores fin de carrera instalados al principio y fin de cada eje, no se hubiera logrado que la herramienta se mueva dentro de un plano especificado.
- Un robot industrial tiene la ventaja de poder ser reprogramado para realizar distintas tareas y el robot industrial cartesiano o XYZ elaborado en este proyecto no es la excepción. Su aplicación es la de imprimir circuitos electrónicos pero si se cambia la herramienta de trabajo por un taladro, una fresa, un pulidor, un esfero, una pinza, entre otros, podría ser programado para realizar una infinidad de tareas.

7.2. RECOMENDACIONES

- Los motores paso a paso utilizados en este proyecto tienen un ángulo de giro de 7.5 grados, esto quiere decir que necesitan dar 48 pasos para lograr una vuelta completa. Si se requiere mayor precisión se recomienda utilizar motores con menor ángulo de giro, por ejemplo: 3.75 grados, 1.8 grados, 0.72 grados.
- El robot puede guardar en su memoria interna un diseño de circuito de máximo 256 trazos. Si se requiere guardar un circuito mas extenso se recomienda utilizar una memoria externa.

- Si en vez de utilizar un dispensador de tinta como herramienta de trabajo se quiere usar un taladro, fresa u otra herramienta que tenga un peso considerable se recomienda utilizar motores de mayor torque.
- En la parte del diseño eléctrico y electrónico, se recomienda tomar muy en cuenta las protecciones para evitar cortocircuitos, esto se logra empleando diodos y resistencias.
- Para que el desplazamiento sobre los ejes no presente resistencia se recomienda mantenerlos lubricados.
- Se recomienda calibrar la distancia que existe desde el dispensador de tinta hacia el plano ya que si está muy apegado se puede generar resistencia al movimiento que forzará los motores o en caso contrario, si apenas logra contacto al plano es muy probable que al cambiar de sentido el movimiento pierda dicho contacto y no imprima, dejando espacios que serían un grave problema en un circuito cerrado.
- Para realizar diseños en modo manual, se recomienda utilizar la opción de desplazamiento paso a paso para trazos muy pequeños, ya que la opción de desplazamiento corrido no permite hacer paradas exactas.

REFERENCIAS BIBLIOGRÁFICAS

LIBROS

ANGULO, José, *Microcontroladores PIC. Diseño de aplicaciones*, segunda edición, McGraw Hill, Madrid – España 2000.

REYES, Carlos, *Microcontroladores PIC Programación en Basic*, segunda edición, Rispergraf, Quito – Ecuador 2006.

INTERNET

Robots Industriales, http://cfievalladolid2.net/tecno/cyr_01/robotica/aplicaciones.htm, consultado en Abril del 2006.

Tutorial sobre Motores Paso a Paso (Stepper motors), <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>, consultado en Abril del 2006.

Motores Paso a Paso,
http://autric.com/Microbotica%20y%20Mecatronica/motores_paso_a_paso.htm,
consultado en Abril del 2006.

Los LCD, <http://es.wikipedia.org/wiki/LCD>, consultado en Abril del 2006.

Hardware of the PIC16F877, http://www.interq.or.jp/japan/se-inoue/e_pic877.htm, consultado en Abril del 2006.

Procedimiento para generar placas de circuitos impresos (PCBs) de alta calidad y magnífica apariencia, http://www.geocities.com/oscar_andrade/papel.htm, consultado en Abril del 2006.

Introducción a los motores paso a paso, <http://www.redeya.com/>, consultado en Abril del 2006.

ANEXO A

DIAGRAMA GENERAL DEL HARDWARE DEL ROBOT

ANEXO B

CÓDIGOS DE PROGRAMAS

CÓDIGO DEL PROGRAMA CARGADO EN EL MICROCONTROLADOR

(Programador utilizado: PicBasic Pro – Microcode)

ADCON1=7

@ device BOD_OFF ;Apaga el Boden de consumo

Include "modedefs.bas" ;Incluye comunicación modo serial

Define LCD_DREG PORTA ; Define los pines del LCD

Define LCD_DBIT 0

Define LCD_RSREG PORTA

Define LCD_RSBIT 4

Define LCD_EREG PORTE

Define LCD_EBIT 0

BYA VAR PORTD.7 ;MOTOR Y

BYB VAR PORTD.6

BYC VAR PORTD.5

BYD VAR PORTD.4

BXA VAR PORTB.7 ;MOTOR X

BXB VAR PORTB.6

BXC VAR PORTB.5

BXD VAR PORTB.4

BZA VAR PORTB.3 ;MOTOR Z

BZB VAR PORTB.2

BZC VAR PORTB.1

BZD VAR PORTB.0

A VAR PORTC.0 ;Nombres para los pines de las filas

B VAR PORTC.1

C VAR PORTC.2

D VAR PORTC.3

UNO VAR PORTC.7 ;Nombres para los pines de las columnas
DOS VAR PORTC.6
TRES VAR PORTC.5
CUATRO VAR PORTC.4

SWY var PORTD.3 ;Nombres para los switch fin de carrera
SWX var PORTD.0
SWYF var PORTD.2
SWXF var PORTD.1

BIP VAR PORTE.1 ;Nombres para chicharra
LIGHT VAR PORTA.5 ;Nombres para led

X VAR BYTE ;Variables auxiliares
DATO VAR BYTE
NUMERO VAR BYTE
num var byte
num = 0
ESF VAR BYTE
ESF = 1

;***** AQUÍ EMPIEZA EL LCD *****

PROG:

HIGH BIP

PAUSE 500

LOW BIP

lcdout \$fe, 1, " INICIALIZANDO"

FOR X=1 TO 3

LOW LIGHT

GOSUB PTECLA

HIGH LIGHT

PAUSE 100

NEXT

GOSUB ORIGEN

lcdout \$fe, 1," LISTO"

PAUSE 1000

PROG2:

GOSUB ORIGEN

PROG2AA:

GOSUB PTECLA

GOSUB PTECLA

lcdout \$fe, 1," ESCOGER OPCION"

PAUSE 500

FOR X = 1 TO 2

LCDOUT \$fe,\$c3," "

pause 300

LCDOUT \$fe,\$c3,"USE FLECHAS"

pause 500

NEXT

PROG2A:

GOSUB BARRIDO :GOSUB PTECLA

IF NUMERO = 8 THEN PROG31

GOTO PROG2A

prog31:

lcdout \$fe,\$c0," Control Manual "

GOSUB BARRIDO :GOSUB PTECLA

IF NUMERO = 8 THEN PROG3

IF NUMERO = 15 THEN teclauno

GOTO PROG31

PROG3:

lcdout \$fe,\$c0," Control Compu. "

PROG3A:

GOSUB BARRIDO :GOSUB PTECLA

IF NUMERO = 2 THEN PROG31

```
IF NUMERO = 8 THEN PROG4
IF NUMERO = 15 THEN COMPU
GOTO PROG3A
```

PROG4:

```
lcdout $fe,$c0,"Imprimir Memoria"
```

PROG4A:

```
GOSUB BARRIDO :GOSUB PTECLA
IF NUMERO = 2 THEN PROG3
IF NUMERO = 8 THEN PROG5
IF NUMERO = 15 THEN PRINTM
GOTO PROG4A
```

PROG5:

```
lcdout $fe,$c0," Figuras  "
```

PROG5A:

```
GOSUB BARRIDO :GOSUB PTECLA
IF NUMERO = 2 THEN PROG4
IF NUMERO = 15 THEN FIGURAS
GOTO PROG5A
```

FIGURAS:

```
lcdout $fe, 1," ESCOJA FIGURA"
```

```
PAUSE 1000
```

```
LCDOUT $fe,$c0," USE FLECHAS  "
```

GOSUB ORIGEN

FIG:

```
GOSUB BARRIDO :GOSUB PTECLA
```

```
IF NUMERO = 8 THEN FIGE
```

```
IF NUMERO = 10 THEN PROG2
```

```
GOTO FIG
```

FIGE:

```
lcdout $fe, 1," LETRA E"
```

```
FIGE1:
```

```
GOSUB BARRIDO :GOSUB PTECLA
```

```
IF NUMERO = 8 THEN FIGPIC
```

```
IF NUMERO = 10 THEN PROG2
```

```
IF NUMERO = 15 THEN LETRAE
```

```
GOTO FIGE1
```

```
FIGPIC:
```

```
lcdout $fe,1," EL PIC"
```

```
FIGPIC1:
```

```
GOSUB BARRIDO :GOSUB PTECLA
```

```
IF NUMERO = 2 THEN FIGE
```

```
IF NUMERO = 15 THEN ELPIC
```

```
IF NUMERO = 10 THEN PROG2
```

```
GOTO FIGPIC1
```

```
COMPU:
```

```
lcdout $fe, 1," CONTROL"
```

```
LCDOUT $fe,$c0," COMPUTARIZADO"
```

```
GOSUB PTECLA
```

```
COMPUTER:
```

```
GOSUB ORIGEN
```

```
COMPU1:
```

```
SERIN PORTE.2,N9600,dato
```

```
IF dato="J" THEN Goto prog2
```

```
IF DATO = "E" THEN num=0 : GOSUB PTECLA : Goto compu1
```

```
IF dato="M" THEN GOSUB PTECLA : Goto compPRINTM
```

```
IF DATO = "N" THEN num = num-1 : gosub ptecla : goto compu1
```

```
num = num+1
```

```
Write num,dato
```

GOSUB PTECLA

IF dato = "P" THEN num=0 : goto print
goto compu1

print:

lcdout \$fe, 1," CONTROL"

LCDOUT \$fe,\$c0," COMPUTARIZADO"

num = num+1

Read num,dato

IF DATO = "A" THEN GOSUB compDIAIARRI

IF DATO = "B" THEN gosub compARRIBA

IF DATO = "C" THEN gosub compDIADARRI

IF DATO = "D" THEN gosub compIZQ

IF DATO = "F" THEN gosub compder

IF DATO = "G" THEN gosub compDIAIABAJ

IF DATO = "H" THEN gosub compABAJO

IF DATO = "I" THEN GOSUB compDIADABAJ

IF DATO = "K" THEN gosub SUBEBAJA

IF DATO = "L" THEN gosub dona

IF DATO = "P" THEN num = 0 : Goto computer

GOTO print

PRINTM:

lcdout \$fe, 1," IMPRIMIENDO"

LCDOUT \$fe,\$c4,"MEMORIA"

num = num+1

Read num,dato

IF DATO = "A" THEN GOSUB compDIAIARRI

IF DATO = "B" THEN gosub compARRIBA

IF DATO = "C" THEN gosub compDIADARRI

IF DATO = "D" THEN gosub compIZQ

IF DATO = "F" THEN gosub compder

IF DATO = "G" THEN gosub compDIAIABAJ

```
IF DATO = "H" THEN gOSUB compABAJO
IF DATO = "I" THEN GOSUB compDIADABAJ
IF DATO = "K" THEN gOSUB SUBEBAJA
IF DATO = "L" THEN gOSUB dona
IF DATO = "P" THEN num = 0 : GOTO PROG2
    goto PRINTM
```

compPRINTM:

```
num = num+1
```

```
Read num,dato
```

```
IF DATO = "A" THEN GOSUB compDIAIARRI
IF DATO = "B" THEN gOSUB compARRIBA
IF DATO = "C" THEN gOSUB compDIADARRI
IF DATO = "D" THEN gOSUB compIZQ
IF DATO = "F" THEN gOSUB compder
IF DATO = "G" THEN gOSUB compDIAIABAJ
IF DATO = "H" THEN gOSUB compABAJO
IF DATO = "I" THEN GOSUB compDIADABAJ
IF DATO = "K" THEN gOSUB SUBEBAJA
IF DATO = "L" THEN gOSUB dona
IF DATO = "P" THEN num = 0 : GOTO computer
goto compPRINTM
```

CAUTION:

```
LCDOUT $fe, 1," FUERA DEL AREA "
```

```
LCDOUT $fe,$c3,"DE TRABAJO"
```

```
GOSUB PTECLA
```

```
RETURN
```

TECLAUNO:

```
LCDOUT $fe, 1,"CONTROL MANUAL"
```

```
LCDOUT $fe,$c0," CORRIDO"
```

```

GOSUB BARRIDO           ;ir a barrido y retornar con un valor
IF NUMERO = 1 THEN GOSUB DIAIARRI
IF NUMERO = 2 THEN gosub ARRIBA
IF NUMERO = 3 THEN gosub DIADARRI
IF NUMERO = 4 THEN gosub IZQ
IF NUMERO = 5 THEN GOSUB origen
IF NUMERO = 6 THEN gosub der
IF NUMERO = 7 THEN gosub DIAIABAJ
IF NUMERO = 8 THEN gosub ABAJO
IF NUMERO = 9 THEN GOSUB DIADABAJ
IF NUMERO = 10 THEN prog2
IF NUMERO = 12 THEN GOTO TECLADOS
IF NUMERO = 13 THEN gosub SUBEBAJA
    GOTO TECLAUNO

```

BARRIDO:

```

LOW A : high B: HIGH C: HIGH D           ;sensar la fila A
IF UNO   = 0 THEN NUMERO =1 :RETURN   ;tecl. puls. retorne con var. cargada 1
IF DOS   = 0 THEN NUMERO =2 :RETURN   ;tecl. puls. retorne con var. cargada 2
IF TRES  = 0 THEN NUMERO =3 :RETURN   ;tecl. puls. retorne con var. cargada 3
IF CUATRO = 0 THEN NUMERO =10:RETURN   ;tecl. puls. retorne con var. cargada 10

```

HIGH A

```

LOW B           ;sensar la fila B
IF UNO   = 0 THEN NUMERO =4 :RETURN
IF DOS   = 0 THEN NUMERO =5 :RETURN
IF TRES  = 0 THEN NUMERO =6 :RETURN
IF CUATRO = 0 THEN NUMERO =11:RETURN

```

HIGH B

```

LOW C           ;sensar la fila C
IF UNO   = 0 THEN NUMERO =7 :RETURN
IF DOS   = 0 THEN NUMERO =8 :RETURN

```



```
IF TRES = 0 THEN NUMERO =9 :RETURN
IF CUATRO = 0 THEN NUMERO =12:RETURN
```

HIGH C

```
LOW D ;sensar la fila D
```

```
IF UNO = 0 THEN NUMERO =14:RETURN
```

```
IF DOS = 0 THEN NUMERO =0 :RETURN
```

```
IF TRES = 0 THEN NUMERO =15:RETURN
```

```
IF CUATRO = 0 THEN NUMERO =13:RETURN
```

HIGH D

```
pause 10
```

GOTO BARRIDO

```
; ***** Programa de antirrebote de teclas *****
```

PTECLA:

```
HIGH BIP ;genera sonido cada que se pulsa tecla
```

```
PAUSE 100 ;duración 100 milisegundos
```

```
LOW BIP ;apagar sonido y led
```

ESPACIO:

```
IF UNO = 0 THEN ESPACIO ;si la tecla sigue pulsada ir espacio
```

```
IF DOS = 0 THEN ESPACIO ;si la tecla sigue pulsada ir espacio
```

```
IF TRES = 0 THEN ESPACIO ;si la tecla sigue pulsada ir espacio
```

```
IF CUATRO = 0 THEN ESPACIO ;si la tecla sigue pulsada ir espacio
```

```
PAUSE 25
```

```
RETURN ;retorna si se suelta las teclas
```

TEMP:

```
PAUSE 5
```

```
RETURN
```

ESPERA:

```
PAUSE 200
```

```
RETURN
```

; ***** Movimiento de los Motores *****

IZQ: IF SWX = 0 THEN gOSUB CAUTION : RETURN

HIGH BXA

GOSUB TEMP

LOW BXA : HIGH BXB

GOSUB TEMP

LOW BXB : HIGH BXC

GOSUB TEMP

LOW BXC : HIGH BXD

GOSUB TEMP

LOW BXD

RETURN

compIZQ:

for x = 1 to 5

IF SWX = 0 THEN gOSUB CAUTION : RETURN

HIGH BXA

GOSUB TEMP

LOW BXA : HIGH BXB

GOSUB TEMP

LOW BXB : HIGH BXC

GOSUB TEMP

LOW BXC : HIGH BXD

GOSUB TEMP

LOW BXD

next

RETURN

DER:

IF SWXF = 0 THEN gOSUB CAUTION : RETURN

HIGH BXD

GOSUB TEMP

LOW BXD : HIGH BXC

```
GOSUB TEMP
LOW BXC : HIGH BXB
GOSUB TEMP
LOW BXB : HIGH BXA
GOSUB TEMP
LOW BXA
RETURN
```

```
compDER:for X = 1 to 5
  IF SWXF = 0 THEN gOSUB CAUTION : RETURN
  HIGH BXD
  GOSUB TEMP
  LOW BXD : HIGH BXC
  GOSUB TEMP
  LOW BXC : HIGH BXB
  GOSUB TEMP
  LOW BXB : HIGH BXA
  GOSUB TEMP
  LOW BXA
next
RETURN
```

```
abajo:
  IF SWYF = 0 THEN gOSUB CAUTION : RETURN
  HIGH BYA
  GOSUB TEMP
  LOW BYA : HIGH BYB
  GOSUB TEMP
  LOW BYB : HIGH BYC
  GOSUB TEMP
  LOW BYC : HIGH BYD
  GOSUB TEMP
  LOW BYD
RETURN
```

compabajo:

for x=1 to 5

IF SWYF = 0 THEN gOSUB CAUTION : RETURN

HIGH BYA

GOSUB TEMP

LOW BYA : HIGH BYB

GOSUB TEMP

LOW BYB : HIGH BYC

GOSUB TEMP

LOW BYC : HIGH BYD

GOSUB TEMP

LOW BYD

next

RETURN

arriba:

IF SWY = 0 THEN gOSUB CAUTION : RETURN

HIGH BYD

GOSUB TEMP

LOW BYD : HIGH BYC

GOSUB TEMP

LOW BYC : HIGH BYB

GOSUB TEMP

LOW BYB : HIGH BYA

GOSUB TEMP

LOW BYA

RETURN

comparrriba:

for x=1 to 5

IF SWY = 0 THEN gOSUB CAUTION : RETURN

HIGH BYD

GOSUB TEMP

LOW BYD : HIGH BYC

GOSUB TEMP

```
LOW BYC : HIGH BYB
GOSUB TEMP
LOW BYB : HIGH BYA
GOSUB TEMP
LOW BYA
next
RETURN
```

DIADABAJ:

```
IF SWYF = 0 THEN gOSUB CAUTION : RETURN
IF SWXF = 0 THEN gOSUB CAUTION : RETURN
HIGH BXD : HIGH BYA
GOSUB TEMP
LOW BXD : HIGH BXC :LOW BYA : HIGH BYB
GOSUB TEMP
LOW BXC : HIGH BXB :LOW BYB : HIGH BYC
GOSUB TEMP
LOW BXB : HIGH BXA :LOW BYC : HIGH BYD
GOSUB TEMP
LOW BXA : LOW BYD
RETURN
```

compDIADABAJ:

```
for x=1 to 5
IF SWYF = 0 THEN gOSUB CAUTION : RETURN
IF SWXF = 0 THEN gOSUB CAUTION : RETURN
HIGH BXD : HIGH BYA
GOSUB TEMP
LOW BXD : HIGH BXC :LOW BYA : HIGH BYB
GOSUB TEMP
LOW BXC : HIGH BXB :LOW BYB : HIGH BYC
GOSUB TEMP
LOW BXB : HIGH BXA :LOW BYC : HIGH BYD
GOSUB TEMP
```

LOW BXA : LOW BYD

next

RETURN

DIAIABAJ:

IF SWX = 0 THEN gOSUB CAUTION : RETURN

IF SWYF = 0 THEN gOSUB CAUTION :RETURN

HIGH BXA : HIGH BYA

GOSUB TEMP

LOW BXA : HIGH BXB:LOW BYA : HIGH BYB

GOSUB TEMP

LOW BXB : HIGH BXC :LOW BYB : HIGH BYC

GOSUB TEMP

LOW BXC : HIGH BXD :LOW BYC : HIGH BYD

GOSUB TEMP

LOW BXD : LOW BYD

RETURN

compDIAIABAJ:

for x=1 to 5

IF SWX = 0 THEN gOSUB CAUTION : RETURN

IF SWYF = 0 THEN gOSUB CAUTION :RETURN

HIGH BXA : HIGH BYA

GOSUB TEMP

LOW BXA : HIGH BXB:LOW BYA : HIGH BYB

GOSUB TEMP

LOW BXB : HIGH BXC :LOW BYB : HIGH BYC

GOSUB TEMP

LOW BXC : HIGH BXD :LOW BYC : HIGH BYD

GOSUB TEMP

LOW BXD : LOW BYD

next

RETURN

DIAIARRI:

```
IF SWX = 0 THEN gOSUB CAUTION : RETURN
IF SWY = 0 THEN gOSUB CAUTION : RETURN
HIGH BXA :HIGH BYD
GOSUB TEMP
LOW BXA : HIGH BXB :LOW BYD : HIGH BYC
GOSUB TEMP
LOW BXB : HIGH BXC :LOW BYC : HIGH BYB
GOSUB TEMP
LOW BXC : HIGH BXD : LOW BYB : HIGH BYA
GOSUB TEMP
LOW BXD : LOW BYA
RETURN
```

compDIAIARRI:

```
for x=1 to 5
IF SWX = 0 THEN gOSUB CAUTION : RETURN
IF SWY = 0 THEN gOSUB CAUTION : RETURN
HIGH BXA :HIGH BYD
GOSUB TEMP
LOW BXA : HIGH BXB :LOW BYD : HIGH BYC
GOSUB TEMP
LOW BXB : HIGH BXC :LOW BYC : HIGH BYB
GOSUB TEMP
LOW BXC : HIGH BXD : LOW BYB : HIGH BYA
GOSUB TEMP
LOW BXD : LOW BYA
next
RETURN
```

DIADARRI:

```
IF SWXF = 0 THEN gOSUB CAUTION :RETURN
IF SWY = 0 THEN gOSUB CAUTION : RETURN
HIGH BXD : HIGH BYD
```

```
GOSUB TEMP
LOW BXD : HIGH BXC :LOW BYD : HIGH BYC
GOSUB TEMP
LOW BXC : HIGH BXB :LOW BYC : HIGH BYB
GOSUB TEMP
LOW BXB : HIGH BXA :LOW BYB : HIGH BYA
GOSUB TEMP
LOW BXA :LOW BYA
RETURN
```

compDIADARRI:

```
for x=1 to 5
IF SWXF = 0 THEN gOSUB CAUTION :RETURN
IF SWY = 0 THEN gOSUB CAUTION : RETURN
HIGH BXD : HIGH BYD
GOSUB TEMP
LOW BXD : HIGH BXC :LOW BYD : HIGH BYC
GOSUB TEMP
LOW BXC : HIGH BXB :LOW BYC : HIGH BYB
GOSUB TEMP
LOW BXB : HIGH BXA :LOW BYB : HIGH BYA
GOSUB TEMP
LOW BXA :LOW BYA
next
RETURN
```

dona:

```
GOSUB ESPERA
gosub izq
gosub izq
GOSUB ESPERA
gosub arriba
gosub arriba
GOSUB ESPERA
```


gosub der

gosub der

GOSUB ESPERA

gosub abajo

gosub abajo

GOSUB ESPERA

GOSUB SUBEBAJA

GOSUB SUBEBAJA

return

SUBEBAJA:

IF ESF = 0 THEN GOSUB bajarESF :RETURN

if ESF = 1 then GOSUB subirESF :RETURN

RETURN

subirESF:

ESF = 0

for x = 1 to 75

HIGH BZD

pause 5

LOW BZD : HIGH BZC

pause 5

LOW BZC : HIGH BZB

pause 5

LOW BZB : HIGH BZA

pause 5

LOW BZA

NEXT

RETURN

bajarESF:

ESF = 1

for x = 1 to 75

```
HIGH BZA
pause 5
LOW BZA : HIGH BZB
pause 5
LOW BZB : HIGH BZC
pause 5
LOW BZC : HIGH BZD
pause 5
LOW BZD
NEXT
RETURN
```

```
; ***** Ir a origen (0,0,0) *****
```

```
ORIGEN:
```

```
IF ESF = 1 THEN GOSUB SUBIRESF
IF SWX=0 THEN ORIGEN0
GOSUB IZQ
GOTO ORIGEN
```

```
ORIGEN0:
```

```
IF SWY=0 THEN RETURN
GOSUB ARRIBA
GOTO ORIGEN0
```

```
; ***** Mover motores paso por paso *****
```

```
TECLADOS:
```

```
LCDOUT $fe, 1,"CONTROL MANUAL"
LCDOUT $fe,$c0," PASO A PASO"
GOSUB PTECLA
GOSUB BARRIDO ;ir a barrido y retornar con un valor
IF NUMERO = 1 THEN GOSUB DIAIARRI
IF NUMERO = 2 THEN gosub ARRIBA
IF NUMERO = 3 THEN gosub DIADARRI
IF NUMERO = 4 THEN gosub IZQ
```

```
IF NUMERO = 5 THEN GOSUB origen
IF NUMERO = 6 THEN gosub der
IF NUMERO = 7 THEN gosub DIAIABAJ
IF NUMERO = 8 THEN gosub ABAJO
IF NUMERO = 9 THEN GOSUB DIADABAJ
IF NUMERO = 10 THEN prog2
IF NUMERO = 11 THEN GOSUB PTECLA : GOTO TECLAUNO
IF NUMERO = 13 THEN gosub SUBEBAJA
    GOTO TECLADOS
```

```
; ***** Figura: LETRA E *****
```

```
LETRAE:
```

```
GOSUB PTECLA
```

```
FOR X = 1 TO 20
```

```
    GOSUB abajo
```

```
    NEXT
```

```
    GOSUB ESPERA
```

```
FOR X = 1 TO 20
```

```
    GOSUB DER
```

```
    NEXT
```

```
    GOSUB ESPERA
```

```
GOSUB bajarESF
```

```
FOR X = 1 TO 100
```

```
    GOSUB abajo
```

```
    NEXT
```

```
    GOSUB ESPERA
```

```
FOR X = 1 TO 60
```

```
    GOSUB DER
```

```
    NEXT
```

```
    GOSUB ESPERA
```

```
FOR X = 1 TO 20
```

```
    GOSUB ARRIBA
```

```
    NEXT
```

```
GOSUB ESPERA
FOR X = 1 TO 40
  GOSUB  IZQ
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  IZQ
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 40
  GOSUB  DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 60
```

```
GOSUB IZQ
NEXT
GOSUB      subirESF
```

```
GOSUB PTECLA
GOSUB PTECLA
```

```
GOTO FIGURAS
```

```
;***** Figura: EL PIC *****
```

```
ELPIC:
GOSUB PTECLA
FOR X = 1 TO 20
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 20
  GOSUB  DER
NEXT
GOSUB ESPERA
```

```
GOSUB      bajarESF
FOR X = 1 TO 18
  GOSUB  DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 12
```

```
GOSUB IZQ
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB IZQ
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 12
  GOSUB DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB IZQ
NEXT
```

```
GOSUB ESPERA
FOR X = 1 TO 30
  GOSUB  ARRIBA
  NEXT
GOSUB      subirESF
FOR X = 1 TO 24
  GOSUB  DER
  NEXT
GOSUB      bajarESF
FOR X = 1 TO 30
  GOSUB abajo
  NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB  DER
  NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB  ARRIBA
  NEXT
GOSUB ESPERA
FOR X = 1 TO 12
  GOSUB  IZQ
  NEXT
GOSUB ESPERA
FOR X = 1 TO 24
  GOSUB  ARRIBA
  NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB  IZQ
  NEXT
GOSUB      subirESF
FOR X = 1 TO 36
```

```
GOSUB    DER
NEXT
GOSUB      bajarESF
FOR X = 1 TO 30
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB    DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 12
  GOSUB    ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 12
  GOSUB    DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB    ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB    IZQ
NEXT
GOSUB      subirESF
FOR X = 1 TO 6
  GOSUB    DER
NEXT
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB      bajarESF
```



```
FOR X = 1 TO 6
  GOSUB  DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB  IZQ
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB  ARRIBA
NEXT
GOSUB      subirESF
FOR X = 1 TO 6
  GOSUB  ARRIBA
NEXT
FOR X = 1 TO 18
  GOSUB  DER
NEXT
GOSUB      bajarESF
FOR X = 1 TO 30
  GOSUB abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB  DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 30
  GOSUB  ARRIBA
NEXT
```

```
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB IZQ
NEXT
GOSUB      subirESF
FOR X = 1 TO 12
  GOSUB    DER
NEXT
GOSUB      bajarESF
FOR X = 1 TO 30
  GOSUB    abajo
NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB    DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
  GOSUB    ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 12
  GOSUB    IZQ
NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB    ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 12
  GOSUB    DER
NEXT
GOSUB ESPERA
FOR X = 1 TO 6
```

```
GOSUB  ARRIBA
NEXT
GOSUB ESPERA
FOR X = 1 TO 18
  GOSUB  IZQ
  NEXT
GOSUB      subirESF
GOSUB PTECLA
GOSUB PTECLA
GOTO FIGURAS

END
```

CÓDIGO DEL PROGRAMA DE COMPUTADORA
(Programador utilizado: Visual Basic – Microsoft)

```
Private Sub Command1_Click()
```

```
MSComm1.Output = "B"
```

```
Command63.Visible = False
```

```
Command64.Visible = False
```

```
Command65.Visible = False
```

```
Command66.Visible = False
```

```
Command67.Visible = False
```

```
Command58.Visible = False
```

```
Command59.Visible = False
```

```
Command61.Visible = False
```

```
Command62.Visible = True
```

```
Plano.Line Step(0, 0)-Step(0, -4)
```

```
End Sub
```

```
Private Sub Command10_Click()
```

```
MSComm1.Output = "K"
```

```
Command62.Visible = False
```

```
Command63.Visible = False
```

```
Command64.Visible = False
```

```
Command65.Visible = False
```

```
Command66.Visible = False
```

```
Command67.Visible = False
```

```
Command58.Visible = False
```

```
Command59.Visible = False
```

```
Command61.Visible = False
```

```
Command6.Visible = False
```

```
Command1.Visible = False
```

```
Command7.Visible = False
```

Command4.Visible = False
Command10.Visible = False
Command5.Visible = False
Command8.Visible = False
Command3.Visible = False
Command9.Visible = False
Command22.Visible = False

Command17.Visible = True
Command21.Visible = True
Command16.Visible = True
Command19.Visible = True
Command11.Visible = True
Command18.Visible = True
Command15.Visible = True
Command20.Visible = True
Command12.Visible = True

End Sub

Private Sub Command11_Click()
MSComm1.Output = "K"

Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False

Command6.Visible = True

```
Command1.Visible = True
Command7.Visible = True
Command4.Visible = True
Command10.Visible = True
Command5.Visible = True
Command8.Visible = True
Command3.Visible = True
Command9.Visible = True
Command22.Visible = True
```

```
Command17.Visible = False
Command21.Visible = False
Command16.Visible = False
Command19.Visible = False
Command11.Visible = False
Command18.Visible = False
Command15.Visible = False
Command20.Visible = False
Command12.Visible = False
End Sub
```

```
Private Sub Command12_Click()
MSComm1.Output = "I"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command65.Visible = True
Plano.Line Step(0, 0)-Step(4, 4), BackColor
End Sub
```

Private Sub Command13_Click()

MSComm1.Output = "P"

Command62.Visible = False

Command63.Visible = False

Command64.Visible = False

Command65.Visible = False

Command66.Visible = False

Command67.Visible = False

Command58.Visible = False

Command59.Visible = False

Command61.Visible = False

Command2.Visible = True

Command25.Visible = True

Command13.Visible = False

Command22.Visible = False

Command10.Visible = False

Command11.Visible = True

Command17.Visible = False

Command21.Visible = False

Command16.Visible = False

Command19.Visible = False

Command11.Visible = False

Command18.Visible = False

Command15.Visible = False

Command20.Visible = False

Command12.Visible = False

Command6.Visible = False

Command1.Visible = False

Command7.Visible = False

Command4.Visible = False

```
Command10.Visible = False
Command5.Visible = False
Command8.Visible = False
Command3.Visible = False
Command9.Visible = False
Command22.Visible = False
End Sub
```

```
Private Sub Command14_Click()
MSComm1.Output = "J"
```

```
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
```

```
Command6.Visible = False
Command1.Visible = False
Command7.Visible = False
Command4.Visible = False
Command10.Visible = False
Command5.Visible = False
Command8.Visible = False
Command3.Visible = False
Command9.Visible = False
Command22.Visible = False
```

```
Command14.Visible = False
Command25.Visible = False
```


Command17.Visible = False
Command21.Visible = False
Command16.Visible = False
Command19.Visible = False
Command11.Visible = False
Command18.Visible = False
Command15.Visible = False
Command20.Visible = False
Command12.Visible = False
Command11.Visible = False
Command13.Visible = False

Command17.Visible = False
Command21.Visible = False
Command16.Visible = False
Command19.Visible = False
Command11.Visible = False
Command18.Visible = False
Command15.Visible = False
Command20.Visible = False
Command12.Visible = False

Command23.Visible = False

Command2.Visible = False

Plano.Visible = False

End Sub

Private Sub Command15_Click()

MSComm1.Output = "G"

Command62.Visible = False

Command63.Visible = False

Command64.Visible = False

```
Command65.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command66.Visible = True
Plano.Line Step(0, 0)-Step(-4, 4), BackColor
```

```
End Sub
```

```
Private Sub Command16_Click()
MSComm1.Output = "C"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command67.Visible = True
Plano.Line Step(0, 0)-Step(4, -4), BackColor
```

```
End Sub
```

```
Private Sub Command17_Click()
MSComm1.Output = "A"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command59.Visible = False
```

```
Command61.Visible = False
Command58.Visible = True
Plano.Line Step(0, 0)-Step(-4, -4), BackColor
```

```
End Sub
```

```
Private Sub Command18_Click()
MSComm1.Output = "F"
Command62.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command63.Visible = True
Plano.Line Step(0, 0)-Step(4, 0), BackColor
```

```
End Sub
```

```
Private Sub Command19_Click()
MSComm1.Output = "D"
Command62.Visible = False
Command63.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command64.Visible = True
Plano.Line Step(0, 0)-Step(-4, 0), BackColor
End Sub
```

Private Sub Command2_Click()

MSComm1.Output = "E"

Plano.Cls

Command62.Visible = False

Command63.Visible = False

Command64.Visible = False

Command65.Visible = False

Command66.Visible = False

Command67.Visible = False

Command58.Visible = False

Command59.Visible = False

Command61.Visible = False

Command2.Visible = False

Command23.Visible = True

Command25.Visible = False

Command13.Visible = False

Command22.Visible = False

Command10.Visible = False

Command17.Visible = False

Command21.Visible = False

Command16.Visible = False

Command19.Visible = False

Command11.Visible = False

Command18.Visible = False

Command15.Visible = False

Command20.Visible = False

Command12.Visible = False

Command6.Visible = False

```
Command1.Visible = False
Command7.Visible = False
Command4.Visible = False
Command10.Visible = False
Command5.Visible = False
Command8.Visible = False
Command3.Visible = False
Command9.Visible = False
Command22.Visible = False
```

```
End Sub
```

```
Private Sub Command20_Click()
MSComm1.Output = "H"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = True
Plano.Line Step(0, 0)-Step(0, 4), BackColor
```

```
End Sub
```

```
Private Sub Command21_Click()
num = num + 1
MSComm1.Output = "B"
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
```

```
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command62.Visible = True
Plano.Line Step(0, 0)-Step(0, -4), BackColor
```

```
End Sub
```

```
Private Sub Command22_Click()
```

```
MSComm1.Output = "L"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command61.Visible = False
Command59.Visible = True
Plano.Circle Step(0, 0), 1.5
End Sub
```

```
Private Sub Command23_Click()
```

```
Command17.Visible = True
Command21.Visible = True
Command16.Visible = True
Command19.Visible = True
Command11.Visible = True
Command18.Visible = True
Command15.Visible = True
Command20.Visible = True
Command12.Visible = True
Command11.Visible = True
```

Command23.Visible = False

Command13.Visible = True

Command2.Visible = True

Plano.Visible = True

End Sub

Private Sub Command24_Click()

Command23.Visible = True

Plano.Visible = True

Command24.Visible = False

Command14.Visible = True

End Sub

Private Sub Command25_Click()

MSComm1.Output = "M"

End Sub

Private Sub Command3_Click()

MSComm1.Output = "H"

Command62.Visible = False

Command63.Visible = False

Command64.Visible = False

Command65.Visible = False

Command66.Visible = False

Command67.Visible = False

Command58.Visible = False

Command59.Visible = False

Command61.Visible = True

```
Plano.Line Step(0, 0)-Step(0, 4)
End Sub
```

```
Private Sub Command4_Click()
MSComm1.Output = "D"
Command62.Visible = False
Command63.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command64.Visible = True
Plano.Line Step(0, 0)-Step(-4, 0)
End Sub
```

```
Private Sub Command5_Click()
MSComm1.Output = "F"
Command62.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command63.Visible = True
Plano.Line Step(0, 0)-Step(4, 0)
End Sub
```

```
Private Sub Command58_Click()
MSComm1.Output = "N"
Plano.Line Step(0, 0)-Step(4, 4), vbWhite
```



```
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
```

End Sub

```
Private Sub Command59_Click()
MSComm1.Output = "N"
Plano.Circle Step(0, 0), 2, vbWhite
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
```

End Sub

```
Private Sub Command6_Click()
MSComm1.Output = "A"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
```

```
Command67.Visible = False
Command59.Visible = False
Command61.Visible = False
Command58.Visible = True
Plano.Line Step(0, 0)-Step(-4, -4)
```

End Sub

```
Private Sub Command61_Click()
MSComm1.Output = "N"
Plano.Line Step(0, 0)-Step(0, -4), vbWhite
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
```

End Sub

```
Private Sub Command62_Click()
MSComm1.Output = "N"
Plano.Line Step(0, 0)-Step(0, 4), vbWhite
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
```

```
Command61.Visible = False
```

```
End Sub
```

```
Private Sub Command63_Click()
```

```
MSComm1.Output = "N"
```

```
Plano.Line Step(0, 0)-Step(-4, 0), vbWhite
```

```
Command62.Visible = False
```

```
Command63.Visible = False
```

```
Command64.Visible = False
```

```
Command65.Visible = False
```

```
Command66.Visible = False
```

```
Command67.Visible = False
```

```
Command58.Visible = False
```

```
Command59.Visible = False
```

```
Command61.Visible = False
```

```
End Sub
```

```
Private Sub Command64_Click()
```

```
MSComm1.Output = "N"
```

```
Plano.Line Step(0, 0)-Step(4, 0), vbWhite
```

```
Command62.Visible = False
```

```
Command63.Visible = False
```

```
Command64.Visible = False
```

```
Command65.Visible = False
```

```
Command66.Visible = False
```

```
Command67.Visible = False
```

```
Command58.Visible = False
```

```
Command59.Visible = False
```

```
Command61.Visible = False
```

```
End Sub
```

```
Private Sub Command65_Click()
```

```
MSComm1.Output = "N"  
Plano.Line Step(0, 0)-Step(-4, -4), vbWhite  
Command62.Visible = False  
Command63.Visible = False  
Command64.Visible = False  
Command65.Visible = False  
Command66.Visible = False  
Command67.Visible = False  
Command58.Visible = False  
Command59.Visible = False  
Command61.Visible = False
```

```
End Sub
```

```
Private Sub Command66_Click()  
MSComm1.Output = "N"  
Plano.Line Step(0, 0)-Step(4, -4), vbWhite  
Command62.Visible = False  
Command63.Visible = False  
Command64.Visible = False  
Command65.Visible = False  
Command66.Visible = False  
Command67.Visible = False  
Command58.Visible = False  
Command59.Visible = False  
Command61.Visible = False
```

```
End Sub
```

```
Private Sub Command67_Click()  
MSComm1.Output = "N"  
Plano.Line Step(0, 0)-Step(-4, 4), vbWhite  
Command62.Visible = False  
Command63.Visible = False
```

```
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
```

```
End Sub
```

```
Private Sub Command7_Click()
MSComm1.Output = "C"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command66.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
Command67.Visible = True
Plano.Line Step(0, 0)-Step(4, -4)
End Sub
```

```
Private Sub Command8_Click()
MSComm1.Output = "G"
Command62.Visible = False
Command63.Visible = False
Command64.Visible = False
Command65.Visible = False
Command67.Visible = False
Command58.Visible = False
Command59.Visible = False
Command61.Visible = False
```

```
Command66.Visible = True  
Plano.Line Step(0, 0)-Step(-4, 4)
```

```
End Sub
```

```
Private Sub Command9_Click()  
MSComm1.Output = "I"  
Command62.Visible = False  
Command63.Visible = False  
Command64.Visible = False  
Command66.Visible = False  
Command67.Visible = False  
Command58.Visible = False  
Command59.Visible = False  
Command61.Visible = False  
Command65.Visible = True  
Plano.Line Step(0, 0)-Step(4, 4)  
End Sub
```

```
Private Sub Form_Load()  
MSComm1.PortOpen = True  
End Sub
```

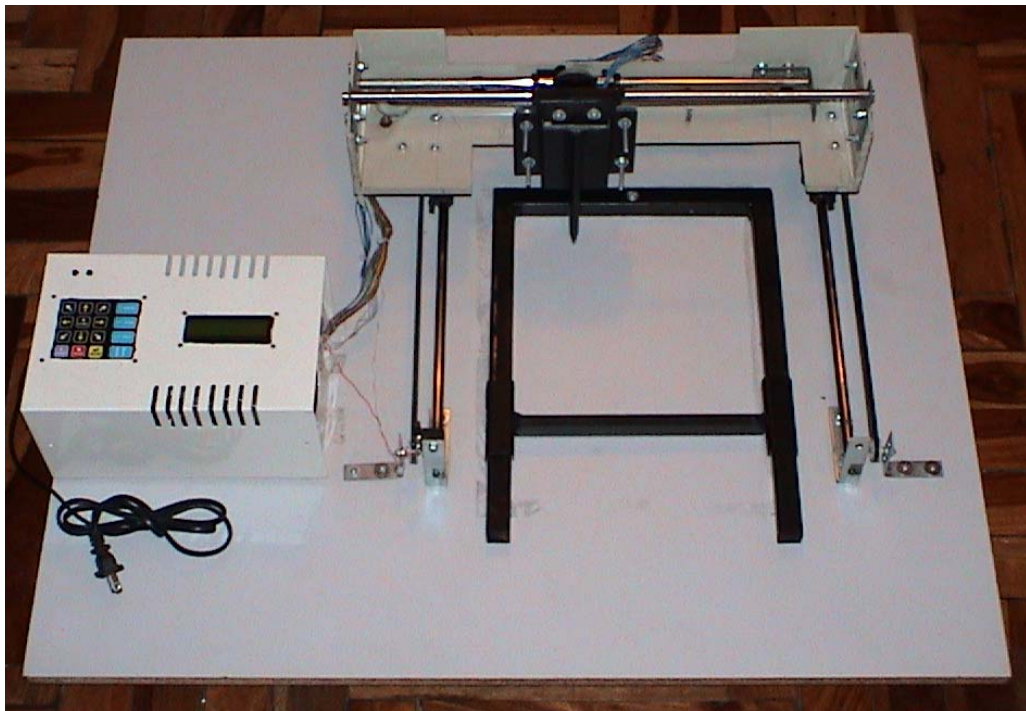
```
Private Sub MSComm2_OnComm()  
End Sub
```

ANEXO C

MANUAL DEL USUARIO

ROBOT CARTESIANO

CAROLO XYZ



MANUAL DEL USUARIO

INTRODUCCIÓN

CAROLO XYZ es un Robot Industrial Cartesiano. Ha sido especialmente implementado y programado para el diseño e impresión de circuitos de electricidad y electrónica básica sobre baquelita.

CAROLO XYZ puede ser manipulado manualmente por medio de los botones que brinda su panel de control, o desde un computador, el cual debe tener instalado el programa para control computarizado, que es suministrado con el robot.

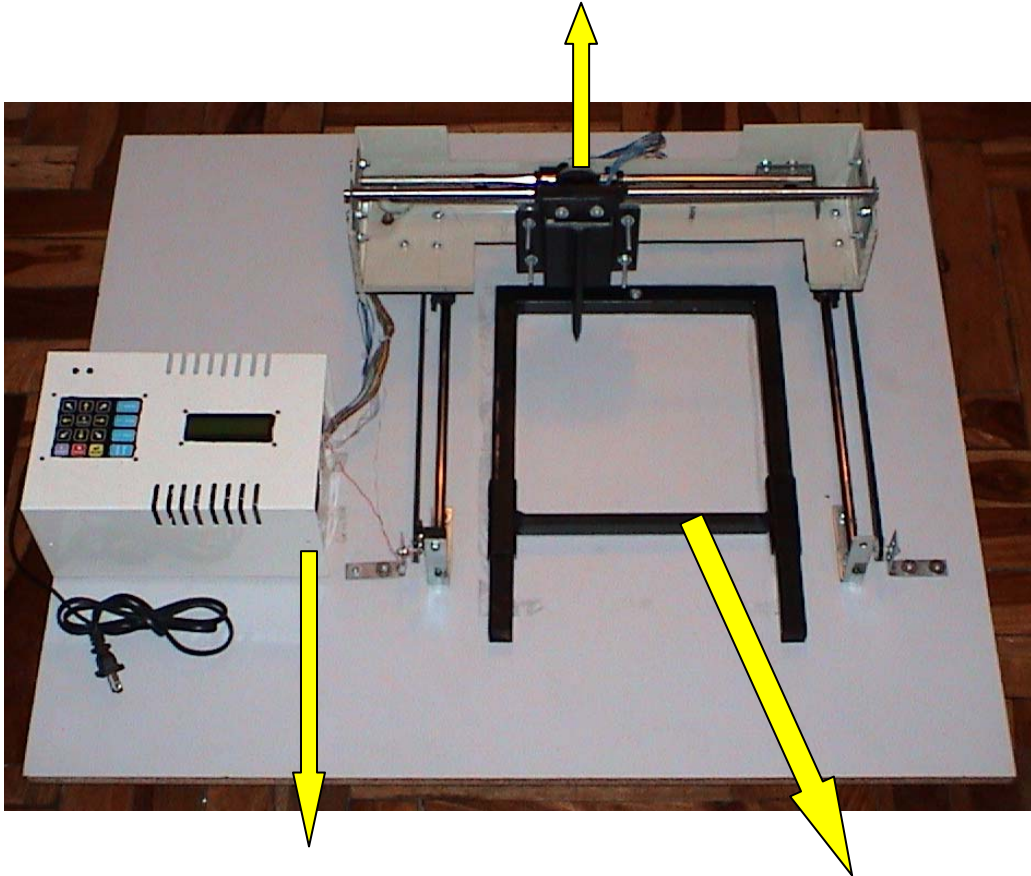
Para poder ser utilizadas las placas impresas, deben ser sometidas a un proceso químico conocido como “atacado con ácido”. Este proceso elimina el cobre de la placa que no esté protegido con tinta, es decir, deja únicamente sobre su superficie las pistas impresas de los circuitos diseñados.

El cerebro del robot es un microcontrolador fabricado por la casa Microchip, específicamente el PIC 16F877A, el mismo que permite manejar los motores, sensores y demás elementos electrónicos que posee.

Este manual lo instruirá para poder utilizar de forma correcta el robot. Lo guiará gráficamente para que se familiarice con cada una de sus partes y sobre todo, para que entienda como funciona cada una de ellas. Por favor, léalo atentamente antes de operar el robot. Guárdelo para posible consulta.

PARTES DEL ROBOT:

Abrazadera + Herramienta (Dispensador de tinta)



Caja Principal (LCD y Panel de Botones)

Marco Sujetador de Baquelita

ACCESORIOS SUMINISTRADOS:



Cable de Comunicación Serial



CD con programa para Modo Computarizado

FUNCIONAMIENTO

El primer paso es posicionar la baquelita:

1. Colocar la placa dentro del Marco Sujetador.
2. Ajustar la placa.

El segundo paso es colocar la Herramienta de Trabajo:

1. Afloje los pernos de la abrazadera.
2. Coloque el dispensador de tinta.
3. Calcule la distancia correcta entre la punta y la superficie de trabajo.
4. Apriete los pernos de la abrazadera.

El tercer paso es energizar el robot:

1. Conectar el cable de poder de la Caja Principal a la toma de energía (110V AC).
2. Encender el Robot (Poner el Switch de la Caja Principal en la posición ON).

Como cuarto paso se tiene escoger en que modo se va a trabajar:

1. Usando las teclas arriba y abajo del Panel de Botones, posicione en uno de los cuatro modos de operación que se presentan en el LCD: Modo Manual, Modo Computarizado, Modo Imprimir Memoria, Modo Figuras.
2. Oprima Enter.

Si escoge el Modo Manual:

1. Posicione la herramienta sobre el lugar de la baquelita donde va a trabajar.
2. Utilice las flechas del Panel de Botones para trabajar su diseño.
3. Para salir de este modo oprima la tecla Menú.

Si escoge el Modo Computarizado:

1. Cargar y correr el programa suministrado para control computarizado en la computadora con la que vaya a trabajar.
2. Conectar el robot con el computador mediante el cable serial.
3. Realizar el diseño del circuito requerido en la pantalla del computador utilizando las opciones que brinda el programa. Se trabaja con el Mouse.
4. Enviar a imprimir, que al mismo tiempo guarda en memoria el diseño creado.
5. Dar click sobre la tecla SALIR para salir de este modo y habilitar nuevamente las teclas del Panel de Botones. El LCD presentará el menú principal.

Si escoge el MODO Imprimir Memoria:

1. Presione Enter y automáticamente el robot imprimirá el último diseño grabado en su memoria. Al terminar volverá al menú principal.

Si escoge el Modo Figuras:

1. Utilizando las flechas arriba y abajo del panel de control, escoger la figura que se quiera imprimir. Los nombres de las figuras serán presentados en el LCD.
2. Oprimir la tecla Enter, esto dará inicio a la impresión de la figura escogida. Al terminar se regresará al menú de figuras.
3. Para salir de este modo se debe oprimir la tecla Menú, lo cual llevará nuevamente al menú principal.

Fecha de entrega:

Roberto Harnisth Pinos

AUTOR

Ing. Víctor Proaño

**COORDINADOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**