



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

50
2ij

**DISEÑO DE UN ÓRGANO TERMINAL
DE MANIPULADOR DE ROBOT**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN

P R E S E N T A :

EDGAR GÓMEZ CRUZ



DIRECTOR DE TESIS: M. EN I. IGNACIO JUÁREZ CAMPOS

CIUDAD UNIVERSITARIA , D.F.

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatoria

A mis Padres, hermanos y familiares:

*Por el apoyo material y espiritual brindado,
por que este logro no es mío sino de ustedes,
por haberme dado vida de su vida
por su presencia, exigencia, deseo y atenciones,
porque siempre los he amado*

A mis amigos y amigas:

*por los momentos buenos y malos,
por habernos tropezado y levantado juntos,
a final de cuentas experiencias y recuerdos gratos*

*A la Facultad de Ingeniería y
sus profesores :*

*por lo conocimientos adquiridos
por la formación profesional brindada
por el orgullo de ser ingeniero*

A Dios:

*Porque siempre has estado
porque no nos has abandonado,
por las oportunidades y pruebas de todos los días
por la generosa oportunidad de la vida*

Gracias a todos quienes de una manera o de otra hemos tenido que ver en la elaboración de este trabajo, en especial al Ing. Ignacio Juárez Campos, Profesor de la Facultad de Ingeniería.

Objetivo

Desarrollar un órgano terminal servocontrolado tipo pinza que conste de sensores que indiquen las cualidades de contacto existentes.

Diseño de un órgano terminal de Manipulador de robot

Índice temático

	<u>Página</u>
Dedicatoria	ii
Objetivo	iii
Índice	iv
I -. Introducción	I-1
I.1 Revisión Bibliográfica	I-2
I.1.1 Definición de Robótica	I-2
I.1.2 Propósitos de la Robótica	I-3
I.1.3 Antecedentes históricos	I-4
I.1.4 Estado del Arte	I-5
I.2 Antecedentes	I-9
I.3.1 Sistema mecánico	I-9
I.3.2 Sistema de percepción	I-11
I.3.3 Sistema de decisión	I-14
I.3.4 Sistema de comunicación	I-14
II -. Análisis de necesidades	II-1
III -. Diseño	
III.1 Mecánico	III-1
III.1.1 Tipos de pinzas	III-2
III.1.2 Análisis mecánico	III-4
III.2 Electrónico	III-7
III.2.1 Controlador del motor	III-8
III.2.2 Sensor de Distancia	III-22

III.2.3 Sensores de Contacto	III-25
III.3 Software	III-29
III.3.1 Puertos de comunicación	III-30
III.3.2 Análisis del sistema	III-33
IV-. Implementación	
IV.1 Mecánica	IV-1
IV.2 Electrónica	IV-5
IV.2.1 Controlador del motor	IV-6
IV.2.2 Sensor de distancia	IV-7
IV.2.3 Sensores de Contacto	IV-8
IV.3 Software	IV-15
IV.4 Integración	IV-30
V -. Resultados y Conclusiones	V-1
VI-. Referencia bibliográfica	VI-1
<u>Apéndices</u>	
A Manual de usuario	VII-1
B Manual de mantenimiento	VII-7
C Piezas mecánicas	VII-10
D Hojas de Datos	VII-28

INTRODUCCIÓN

Introducción

I.1 Revisión Bibliográfica

I.1.1 Definición de Robótica

The Robot Institute of America define un robot como un "manipulador multifuncional reprogramable empleado para mover cargas, piezas, objetos o dispositivos especiales según variadas trayectorias establecidas para realizar diferentes tareas" (1979).

Para algunas personas ésta es una definición en rigor formal y poco sugestiva, puesto que la *robótica* también incluye aspectos tales como *fuerza a través de la inteligencia* o bien *donde la inteligencia artificial se aplica al mundo real*. Webster dice: *-Un dispositivo automático que desarrolla funciones normalmente atribuibles a los humanos o una máquina en la forma de un humano.*

El término de *Robótica*, se refiere al estudio y uso de los robots. Dicho término fue concebido y usado por primera vez por el científico y escritor norteamericano de origen ruso Isaac Asimov (1920-1992). Asimov escribió espléndidamente sobre una amplia variedad de temas, sin embargo, fue conocido principalmente por sus trabajos de ciencia ficción. Entre sus obras más famosas se incluye *Yo Robot* (1950).

Asimov también propuso sus *tres leyes de la robótica*, a las cuales agregaría más tarde la *ley cero*:

Ley Cero: Un robot no debe dañar a la humanidad o, por su inacción, dejar que la humanidad sufra daño.

Ley Uno: Un robot no debe dañar a un ser humano o, por su inacción, dejar que un ser humano sufra daño, excepto cuando esto violara la ley cero.

Ley Dos: Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto cuando estas órdenes están en oposición con una ley de orden mayor (ley cero y uno).

Ley Tres: Un robot debe proteger su propia existencia, hasta donde esta protección no esté en conflicto con las leyes anteriores.

Manual de Robótica

56ª Edición, año 2058

A pesar de tratarse de obras de ciencia ficción, no cabe duda que día tras día estas reglas van tomando mayor vigencia e importancia para generar una verdadera conciencia, sólo basta echar un vistazo a los actuales avances tecnológicos para darse cuenta que la realidad se acerca rápidamente a lo que antes era "ficción".

1.1.2 Propósitos de la Robótica

Los propósitos de la robótica, en general, son¹:

- Incrementar la velocidad de trabajo

¹ Apuntes tomados de la materia Robótica impartida por el Ing. Ignacio Juárez Campos (93-2), Prof. de la Facultad de Ingeniería

- Tener mayor flexibilidad (un mayor número de piezas diferentes producidas y posibilidad de efectuar grandes cambios)
- Homogeneización de la calidad del producto (el trabajo humano puede rendir mayor calidad pero no es constante)
- Mejorar el control, la organización y la programación de la producción
- Eliminación de tiempos muertos
- Liberar al obrero de condiciones de trabajo insalubres y de alto riesgo
- Eliminar la fatiga inherente al trabajo tedioso y repetitivo.

1.1.3 Antecedentes históricos

Los primeros robots industriales fueron desarrollados por George Devol y Joe Engleberger a finales de la década de los 50's y principios de los 60's. Las primeras patentes fueron creadas por Devol para transferencia de partes de máquinas. Engleberger fue el primero en promover el mercado de robots.

En la comunidad de la investigación el primer autómate fue probablemente la máquina de Grey Walter y la bestia de John Hopkins. Teleoperados ó dispositivos controlados remotamente habían sido construidos tan tempranamente como los primeros vehículos radiocontrolados hechos por Nikola Tesla en 1890. Tesla es mejor conocido como el inventor del motor de inducción, transmisión de potencia de corriente alterna, y otros numerosos dispositivos eléctricos. Tesla también había imaginado mecanismos inteligentes que fueran tan capaces como los humanos.

Existe un gran número de organizaciones y sociedades relacionadas con la robótica. Algunas se encuentran ligadas específicamente con la industria, otras

con la academia² y otro gran número son grupos de "hobbistas". Además, ciertos grupos, tales como ASME o IEEE, tienen a la robótica como una de sus varias subdisciplinas en sus respectivos campos. Varias de estas organizaciones también cuentan con publicaciones asociadas a la robótica.

1.1.4 Estado del Arte

Los brazos industriales modernos han incrementado su capacidad y rendimiento a través de controladores y lenguajes de desarrollo así como mejoras en sus mecanismos, sensores y sistemas de transmisión. En la primera mitad de los 80's el robot industrial creció rápidamente debido a las grandes inversiones hechas por la industria automotriz. El rápido salto hacia las fábricas del futuro se convirtió en un gran desastre cuando la integración y la viabilidad económica de estos esfuerzos entraron en crisis debido a los altos costos. En el mediano plazo hubo una enorme sacudida en la industria del robot. En los Estados Unidos por ejemplo, sólo una compañía (Adept), continúa con la producción de brazos industriales. La mayoría de las compañías quebraron o fueron vendidas a competidores europeos y japoneses.

A principios de los 90's los números y ventas emergieron para sobrepasar rápidamente los niveles alcanzados en los 80's. En la revista "Motion Control Magazine" de abril del 94 la Asociación de Industrias Robóticas publicó que sus ventas habían saltado rápidamente hasta un 40 % en Junio del 93. Se recibieron una nueva serie de pedidos por parte de compañías norteamericanas que basaban su producción en la robótica, totalizando 3,640 robots valuados en 306.2 millones de dólares.

² Comp. robotics, Frequently Asked Questions (FAQ), Carnegie Mellon University, Pittsburgh. Dirección Internet: nivek@cmu.edu

En el New York Times del miércoles 7 de septiembre del mismo año, se publicó que a finales de los 80's una crisis empujó fuera de los negocios de los robots a varias compañías norteamericanas. En la primera mitad de 1994, 4,335 robots con un valor total de 383.5 millones de dólares fueron vendidos. Fanuc es el líder con cerca de 360 millones en ventas este año. ABB es el segundo con ventas estimadas en 120 M. Después se encuentran varios japoneses: Motoman, Panasonic, Sony y Nachi. El productor mayor norteamericano que ha sobrevivido es Adept Technology con alrededor de 50 millones en ventas.

En este artículo se presenta la siguiente tabla de pedidos en los Estados Unidos:

Año	No. de robots	Ventas(\$US)
1984	5800	\$480 M
1985	6200	\$380 M
1986	5400	\$320 M
1987	3800	\$300 M
1988	4000	\$325 M
1989	4500	\$510 M
1990	5000	\$510 M
1991	4000	\$410 M
1992	5250	\$500 M
1993	6800	\$630 M
1994	4335 (6 meses)	\$383 M (6 meses)

Las compañías norteamericanas que basan su producción en robots están pasando por uno de sus mejores momentos. La asociación de industrias de

robótica (RIA), menciona que han surgido pedidos recientemente de compañías norteamericanas para alcanzar una de sus mejores cifras hasta el momento en nueve meses. En septiembre, los nuevos pedidos totalizaron 6,218 robots valuados en \$548 millones, un incremento en 12% de unidades y un 13% de incremento sobre el mismo periodo de 9 meses en el año anterior.

La creciente demanda, menciona el grupo, viene de los fabricantes de los Estados Unidos quienes están aprendiendo finalmente lo que los japoneses han sabido por años: Los robots pueden jugar un papel significativo en la mejora de la productividad, calidad, flexibilidad y negocios. Sin embargo, a pesar de que la demanda está en franco crecimiento y de que los Estados Unidos son el segundo universo mas grande de usuarios de robótica con alrededor de 53,000 sistemas, los japoneses los superan en un 700% de robots en uso.

Actualmente, se están haciendo grandes esfuerzos en lo que respecta a las áreas de la inteligencia artificial (redes neuronales), bases de datos y sistemas de procesamiento en paralelo. En el Instituto Federal de Tecnología Suizo se cuenta con la facilidad de un laboratorio de máquinas de percepción móvil, el cual está consagrado a la investigación básica e interdisciplinaria, transferencia de tecnología y educación autodidacta en inteligencia artificial a través de la robótica. Los esfuerzos de investigación y transferencia de tecnología están concentrados en la reducción del riesgo humano en situaciones peligrosas, transformación del medio y/o mejoramiento de la calidad de vida a través de nuevos procesos de manufactura.

Varios investigadores se encuentran con regularidad involucrados en competiciones de estudiantes en diseños aéreos de robots, en los cuales los conceptos de ingeniería concurrente están siendo aplicados para caracterizar los sistemas. Entre los tópicos actuales de mayor estudio se encuentran:

- Actuación en paralelo de manipuladores
- Arquitecturas de computadoras de procesamiento en paralelo para el sensado y control de robots
- Ultrasonido generado por láser para sensar estructuras de materiales
- Vehículos de posicionamiento autónomo
- Técnicas para evitar colisiones
- Control de brazos flexibles
- Movimientos coordinados de varios brazos
- Control avanzado de lazo cerrado
- Aprendizaje de control
- Control de fuerza
- Estructuras de control reactivo y planeación jerárquica, visión, retroalimentación
- Visión activa
- Multimedia
- Animación de un cuerpo humano
- Investigación basada en visión usando estimaciones.
- Máquinas interactivas inteligentes (Máquinas inteligentes que en cooperación con los humanos resuelven tareas complicadas)

¿Qué es multimedia en robótica³?

Multimedia en robótica es un área naciente de las ciencias computacionales concerniente a los nuevos mercados para tecnologías robóticas, enfatizando en la importancia de áreas tales como la realidad virtual, telepresencia, animación y entretenimiento.

³ Comp. robotics, Frequently Asked Questions (FAQ), Carnegie Mellon University, Pittsburgh. Dirección Internet: nivek@cmu.edu

1.2 Antecedentes

La robótica es por excelencia un área totalmente interdisciplinaria, los sistemas incluyen aspectos mecánicos, electrónicos, hardware, software y todos los tópicos relacionados a estas categorías. Como resultado, el diseño y construcción de tales sistemas requiere de una correspondiente variedad de componentes y partes.

Tomando en cuenta lo anterior, podemos mencionar que el robot industrial se encuentra constituido por cuatro subsistemas funcionales principales⁴:

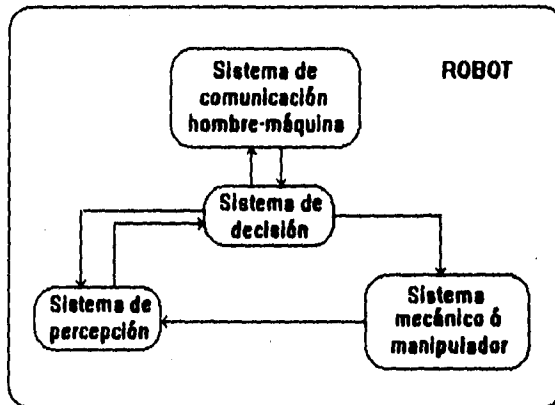


Fig. 1.1

1.2.1 Sistema mecánico

El sistema mecánico está constituido por la cadena cinemática asociada a la imagen del robot industrial, los actuadores que animan este mecanismo y todo

⁴Apuntes tomados de la materia Robótica impartida por el Ing. Ignacio Juárez Campos (93-2), Prof. de la Facultad de Ingeniería

el equipo complementario que permite la correcta realización de las tareas requeridas.

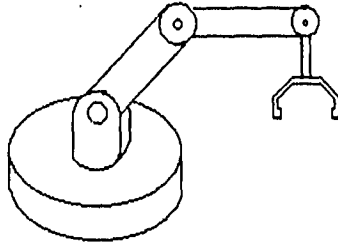


Fig. 1.2

Generalmente, el brazo del manipulador termina en una muñeca , a la que hay que acoplar un dispositivo que permita realizar la labor deseada. También cuenta con un brazo para posicionar la mano en el espacio. Si bien los movimientos de las articulaciones del manipulador colocarán al elemento terminal en cualquier posición y orientación, dentro del área accesible de trabajo, la pinza o herramienta final precisará de controles para su gobierno. La variedad de los elementos terminales confiere al robot una gran versatilidad así como la posibilidad de realizar labores muy diversas.

En particular, dos articulaciones o juntas o una articulación y el extremo del brazo (que es el órgano terminal), se encuentran unidas por eslabones. Dichas articulaciones de acuerdo a su movimiento pueden ser clasificadas como rotacionales ó prismáticas (traslacionales).

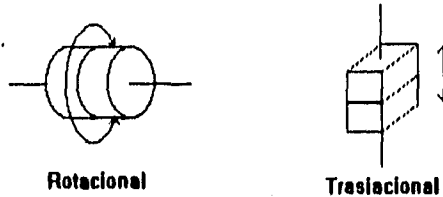


Fig. 1.3

Los grados de libertad son los parámetros que se precisan para determinar la posición y orientación del elemento terminal del manipulador. También se definen como los posibles movimientos básicos (giros y desplazamientos) independientes.

1.2.2 Sistema de percepción

El sistema de percepción está formado por los transductores y los circuitos asociados que permiten la generación de señales que informan al robot de su estado interno (posición y velocidad en cada articulación) ó bien que le proporciona la información sobre el medio ambiente que le rodea. Este sistema es un elemento que permite determinar la inteligencia de una máquina en la medida que pueda planear acciones a partir de la información obtenida de su capacidad de entender el medio ambiente.

Es posible hacer una clasificación de sensores de acuerdo a la información que aportan en propioceptivos y exteroceptivos. Los primeros son aquellos que dan información sobre el estado interno del robot y los segundos determinan las condiciones del medio ambiente en torno a la máquina. Estos elementos detectan

variables tales como contacto, posición, velocidad, esfuerzo y proximidad. Los datos pueden ser obtenidos por medio de microswitches, interruptores magnéticos, cámaras de televisión, elementos inductivo, resistivos, etc. A continuación se hace una breve descripción de algunos de los sensores disponibles para estos fines.

Existe un gran número de cámaras en el mercado que un consumidor normal puede adquirir, como algunas pequeñas camcorders que no son muy caras y pueden resultar adecuadas para una gran variedad de aplicaciones. Algo curioso que vale la pena mencionar, es que aunque estas cámaras son muy pequeñas, muchas de ellas se complementan con grandes cajas electrónicas que por medio de un cable generan potencia y transmiten el video.

En cuanto a la estimación de distancias, los sensores se basan en cuatro técnicas básicas para medirlas: tiempo de propagación de pulsos, comparación de fase, métodos Doppler e interferometría. Todas son usadas en la práctica dependiendo de la aplicación en particular.

Tiempo de propagación de pulsos, como su nombre lo indica, implica medir el ciclo de tiempo que tarda una señal en ser transmitida hacia una superficie reflejante y regresar. Este principio es usado en radares, altimetría de satélites, medición lunar con láser, dispositivos militares. También se le aplica para mandar señales desde pulsadores, cuyo tiempo es medido desde dos ó más radiotelescopios, para que la diferencia en tiempos de llegada sea convertida a distancias intercontinentales, con un margen de error de unos cuantos centímetros.

La diferencia de fase involucra el uso de una onda portadora la cual puede ser modulada en diferentes longitudes de onda. Midiendo la diferencia en fase

entre la señal transmitida y la señal recibida, después de que ha sido reflejada desde el otro extremo del objetivo, la distancia puede ser determinada como un número entero (desconocido), más una fracción de longitud de onda, la cual es conocida a partir de una comparación de fase. Usando un rango de modulación de frecuencias la ambigüedad puede ser resuelta.

Las técnicas Doppler fueron usadas en los sistemas de posicionamiento de los primeros satélites. La frecuencia recibida de una órbita de satélite es comparada con la señal actualmente transmitida como una función del tiempo. La velocidad de cambio de la frecuencia da un cierto rango de comportamiento entre el satélite y el observador, mientras que el instante cuando las dos frecuencias se igualan da el punto de aproximación más cercano. Conociendo los parámetros orbitales del satélite que transmite, la posición de los observadores puede ser determinada.

Los métodos interferométricos son usados por la metrología, alte precisión de medición de distancias sobre pequeñas longitudes (hasta 60 metros), y en la definición del metro.

La medición de distancias también se puede hacer por medio de sonidos. El tiempo que le toma a un pulso acústico propagarse a través del aire o del agua, reflejarse en un cierto medio y regresar a un detector, es proporcional a la distancia. Los dispositivos acústicos Polaroid son baratos y fácilmente integrables además de que se les ha encontrado un amplio campo de acción en la robótica: Algunas compañías han desarrollado otros dispositivos pensando en que Polaroid no será por mucho tiempo la única elección.

En cuanto a la medición de movimiento lineal existe una amplia gama de dispositivos que dan directamente la posición absoluta para movimientos lineales.

Frecuentemente los engranes son combinados con encoders rotatorios para obtener la posición absoluta..

El lector habrá observado que en estas descripciones se está haciendo énfasis en la medición de distancias. Esto es porque uno de los problemas a resolver, en el presente trabajo, es precisamente el determinar desplazamientos (distancias recorridas), para obtener los espesores de los objetos apresados por el órgano terminal propuesto.

1.2.3 Sistema de decisión

El sistema de decisión procesa la información sensorial que controla la estructura mecánica para que la tarea programada sea realizada. Además, permite el registro en memoria de dichas tareas. Obviamente, para ello se utilizan sistemas electrónicos que incluyen microcontroladores que gobiernan a toda la estructura. En varias ocasiones se llega a utilizar un microcontrolador por cada articulación para acelerar los procesos (ejecución en paralelo).

1.2.4 Sistema de comunicación

El sistema de comunicación permite al operador comunicarse con el robot, introducir las instrucciones que forman una tarea, modificarlas y, en general, activar los componentes del sistema robot. La comunicación hombre-máquina puede realizarse vía computadora u otros dispositivos especiales. Al establecerse una comunicación vía computadora, se necesita manejar un lenguaje de alto nivel orientado a la robótica.

A propósito de este tema, vale la pena hacer un comentario. Los lenguajes tradicionales (Basic, Pascal, Fortran, etc.) no son fácilmente aplicables en la robótica, debido a que no incluyen rutinas especiales que los manipuladores requieren para trabajar sus muy particulares interfaces y señales. Además, el robot requiere manejar de forma sencilla otros dispositivos periféricos, como cámaras de televisión, bandas u otros sensores exteroceptivos. Por ello se han desarrollado una serie de lenguajes orientados a simplificar y manejar de manera mas apropiada las señales y comandos utilizados por los manipuladores.

Apesar de ello, esta labor no ha sido sencilla, pues hasta ahora los lenguajes elaborados sólo cubren tareas bien definidas, como procesos de pintura, ensamble, soldadura; por lo que no son utilizables de manera general.

ANÁLISIS DE NECESIDADES

II Análisis de necesidades

En el Instituto de Ingeniería se han estado desarrollando algunos proyectos relacionados con la robótica. Uno de esos proyectos es un manipulador de "5" grados de libertad¹, el cual requiere de un órgano terminal que cumpla con ciertas características de funcionamiento.

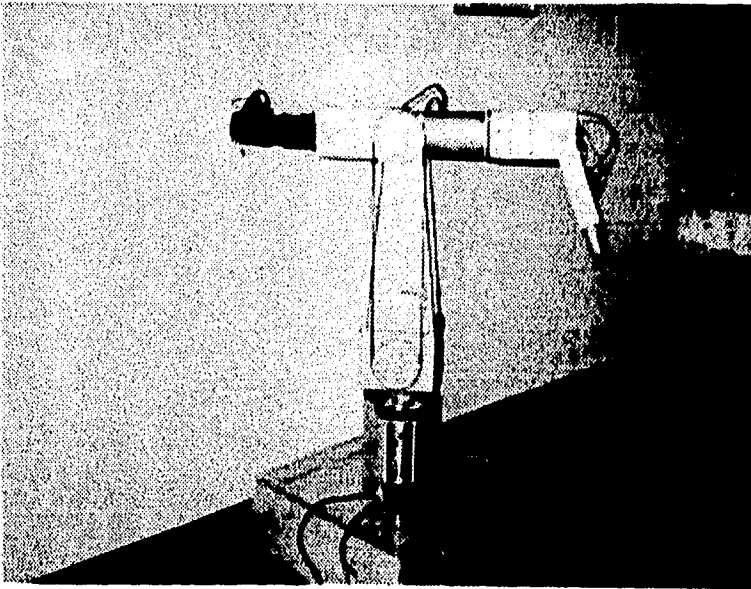


Fig. II.1

Como se mencionó en la Introducción, el órgano terminal de los manipuladores puede ser una herramienta muy específica para desarrollar una tarea, o bien simplemente una pinza para sujetar objetos. En este caso particular, se requiere de un órgano terminal que sea capaz de tomar objetos y además

¹ Proyecto desarrollado por el Ing. Ignacio Juárez Campos

caracterizar el tipo de contacto obtenido, esto con el fin de esbozar la forma del objeto apresado en la pinza.

La idea es la siguiente: El manipulador se posicionará en diferentes puntos del espacio de tres dimensiones. Gracias a las cualidades sensitivas al contacto de la pinza, la computadora podrá conocer las dos "caras" sujetas del objeto en estudio, además, determinará la distancia entre los dedos de la pinza para conocer el espesor del objeto apresado para esa posición tridimensional. El manipulador entonces podrá moverse a otro punto en el espacio para seguir caracterizando el objeto desde diferentes puntos.

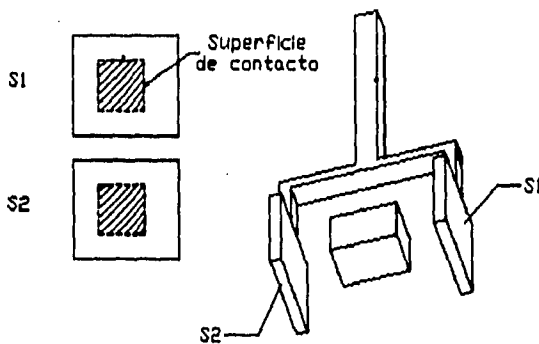


Fig. II.2

¿Cuáles son entonces las características deseadas para el órgano terminal? La pinza debe ser sensible al contacto, ya sea que haya tomado un objeto o bien que haya cerrado sin encontrar nada. En caso de haber apresado algún elemento, debe poseer una serie de sensores que le indiquen que parte de la superficie de la pinza está en contacto con el objeto, esto con el fin de poderlo caracterizar. También, para estar en posibilidad de crear un modelo mas completo, se hace necesario conocer la distancia que existe entre los dedos de la pinza.

Obviamente que al hablar de una pinza, de movimiento y de control se involucran elementos tales como un actuador (motor), circuitos electrónicos de potencia y lógicas digitales, sensores y software, es decir, el problema a atacar debe estructurarse en tres áreas bien definidas (mecánica, electrónica y computación), con las correspondientes subdivisiones que pudiera tener cada una de ellas.

El problema es claro, tres bloques principales interrelacionados para conseguir los objetivos planteados. Hagamos las siguientes consideraciones:

1) Se trata de construir una pinza lo cual implica directamente una estructura mecánica.

2) La estructura mecánica va a tener un movimiento autónomo, es decir, habrá un actuador que la anime.

3) La pinza debe tener un comportamiento lógico de acuerdo a lo que el operador desee, por lo que se está hablando de programación vía computadora u otro dispositivo programable.

4) La computadora no puede interactuar directamente con una estructura mecánica que posea un actuador que disipe gran potencia (en comparación con la computadora), por lo que se necesita una interface electrónica entre el sistema digital y la pinza.

5) La pinza también debe contener sensores en sus "dedos" para indicar cualitativamente el contacto. La ubicación y el número de sensores debe ser apropiado para conseguirlo.

6) Se debe obtener el espesor de los objetos apresados. Para realizarlo se cuenta ya con un servomotor de CD, el cual genera una serie de pulsos acordes al giro del rotor lo cual puede relacionarse con el desplazamiento en los dedos de la estructura.

7) Toda esta información debe ser mandada y obtenida por la computadora, para desplegar datos, tomar decisiones y controlar nuevos movimientos. Por esta razón, sería muy conveniente que se aligerara en lo posible la carga de la misma, creando un sistema digital que le ahorre algunos procesos. Por ende, se involucra una conexión física con la circuitería creada, lo cual estará sujeto a las restricciones planteadas fundamentalmente en los puertos de comunicación del sistema computacional (líneas de entrada y salida).

8) Debe ser costeable.

Teniendo en cuenta las consideraciones anteriores, me permito plantear el siguiente esquema como solución:



Fig. II.3

El sistema de cómputo gobernará a la pinza vía una interface electrónica. El usuario indicará a la computadora iniciar un movimiento, ésta a su vez mandará una señal al sistema electrónico para que active la etapa de potencia correspondiente. La estructura mecánica iniciará su movimiento y se detendrá hasta que haya cerrado (ó abierto) totalmente ó bien que haya apresado un objeto. ¿Cómo detectar este momento? La pinza contará con una serie de sensores propioceptivos y exteroceptivos que avisarán al sistema electrónico que ha sucedido alguno de los eventos predeterminados para que desactive la etapa de potencia del motor.

En caso de haber tomado un objeto, los datos aportados por los sensores podrían obtenerse de dos maneras:

1) Estableciendo una conexión física desde el puerto de la computadora hasta los sensores para manejarlos directamente con la incomodidad que involucraría tener más de un cable saliendo de la computadora a otro dispositivo que está en constante movimiento, o bien:

2) Permitir que la interface electrónica reciba la información y posteriormente la mande al sistema de control. Esta opción resulta más apropiada. La misma situación se daría para la información mandada por el encoder.

Como se puede observar en este análisis, son varias las variables a trabajar en el sistema en general, entre ellas destacan el estado de los sensores de contacto de cada dedo (sensores exteroceptivos), el estado de los sensores internos de la pinza y la distancia recorrida.

A diferencia de la distancia recorrida, el estado de las variables de contacto externas e internas repercutirá en el sistema, ya que a partir de su valor se determinará la siguiente acción. De esta manera, es notorio que el problema involucra un sistema de control con realimentación (realmente se trata de un sistema de control secuencial por la naturaleza de los eventos involucrados). En la siguiente ilustración se presenta el diagrama a bloques del sistema a controlar.

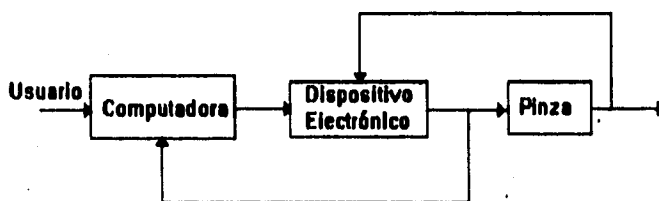


Fig. II.4

En el diagrama se pueden encontrar varios de los requerimientos mencionados. A la entrada el usuario presenta la salida deseada. A continuación, la computadora manda las señales pertinentes a la interfaz electrónica, la cual activa los movimientos en la pinza. Cabe destacar la presencia de dos lazos de realimentación a los controladores, uno de ellos va de la salida de la pinza hacia el controlador electrónico, el cual recibe específicamente el estado de los sensores internos del dispositivo. De esta manera, de ser necesario el controlador tomará la decisión de cambiar el estado de giro del rotor a un reposo total, no haciéndose necesaria la intervención de la computadora. En caso de que el controlador no pueda manejar una situación en especial, generará una señal que recibirá la computadora por medio del otro lazo de realimentación, entonces analizará la información y girará nuevas instrucciones al dispositivo electrónico.

El número de variables a tratar y las características de su comportamiento dependerán de la implementación de cada uno de los elementos que componen al sistema en total. Estos elementos se abordan individualmente en la sección correspondiente.

DISEÑO MECÁNICO

III.1 Diseño Mecánico

La mecánica es una parte de la física que se constituye por dos elementos fundamentales: estática (análisis de los cuerpos en reposo) y dinámica (se encarga del análisis de los cuerpos en movimiento).

Debido a la naturaleza de su funcionalidad, esta sección tratará sobre el diseño de un elemento eminentemente mecánico: la pinza ó "gripper".

III.1.1 Tipos de Pinzas

Para poder abordar este problema, es necesario reconocer las condiciones bajo las cuales se va a trabajar. Por ello, realicé una investigación previa sobre los tipos de modelos que ya existen en cuanto a "grippers", presentando aquí algunas posibles configuraciones.

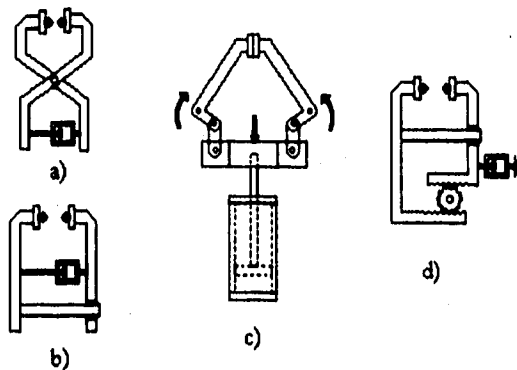


Fig. III.1.1

La figura III.1.1 a) muestra un sistema de pinza común actuado por un pistón neumático. Dicho sistema es de implementación sencilla pero presenta el

inconveniente de no tener un desplazamiento de puntas lineal (las superficies de los "dedos" no permanecen paralelas en todo momento, lo que complica el análisis de desplazamiento y superficie de contacto). El inciso b) ilustra un sistema de prensa con guía que también tiene como actuador un pistón neumático y que presenta la ventaja de tener un desplazamiento lineal en las puntas. Esta estructura también es relativamente sencilla de implementar.

El inciso c) muestra una pinza con un mecanismo de dos barras por dedo. Dicho mecanismo podría resultar mas complicado de elaborar por tener un número mayor de piezas, además de no tener un desplazamiento lineal. Finalmente, el inciso d) contiene un sistema de prensa con cremallera y pistón. La fabricación de este último esquema podría facilitarse mediante la utilización de piezas comerciales.

Estudiemos otros dos modelos que por su "sencillez" de estructura resultarían viables de construir¹:

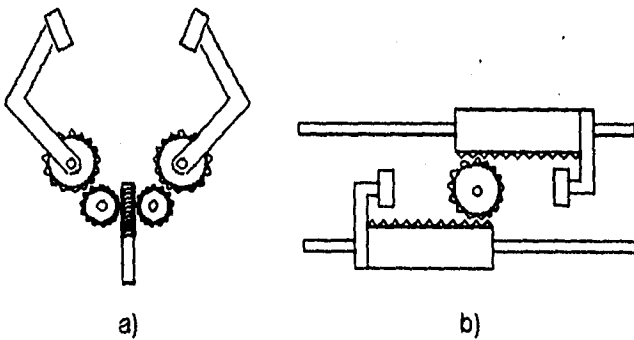


fig. III.1.2

¹ Considero oportuno dar las gracias por la orientación e información aportadas especialmente en esta materia al Ing. Ignacio Juárez Campos, profesor de la Facultad de Ingeniería

En la figura, el inciso a) muestra una pinza accionada por cuatro engranes y un tornillo sin fin. Al girar el tornillo acciona los engranes más pequeños los cuales debido a su posición giran en sentidos opuestos y a su vez hacen girar los engranes que soportan los dedos de la pinza. Los elementos importantes a considerar en este diseño son el tornillo sin fin y los engranes pues, deben tener las características de cuerda y número de dientes acordes para lograr un movimiento coordinado, además de la evidente no linealidad en el desplazamiento de las puntas.

En la fig. III.1.2 b), se puede apreciar un esquema similar a la figura III.1.1 d) pero sin el pistón (visto desde una perspectiva superior). En él se observan dos bloques dentados colocados sobre dos ejes que funcionan como guías. Al girar el engrane del centro hace que los bloques se desplacen en sentidos opuestos permitiendo la abertura o cierre de la pinza. La sencillez y funcionalidad de este modelo lo convierten, de entre todos los analizados, en el elegido para implementarse.

III.1.2 Análisis Mecánico

Antes de empezar a diseñar piezas, es necesario considerar las características de funcionamiento deseadas para la pinza, las cuales deben ser establecidas tomando en consideración los elementos con los que se cuenta previamente (un engrane de 2cm de diámetro y dos barras guías de 8.8 cm de longitud con 4.7625 mm de diámetro, una barra de 6 mm de diámetro y un motor de CD):

Abertura máxima: 8 cm

Superficie de contacto de los dedos: 25 cm² (5 cm x 5 cm)

Analicemos ahora el comportamiento del sistema elegido:

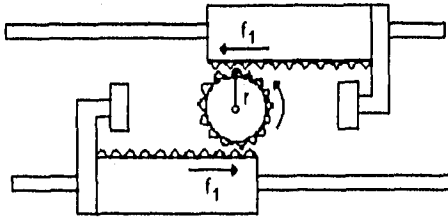


fig. III.1.3

La velocidad con que se moverán las placas esta en función del radio r del engranaje y la velocidad angular ω del eje del motor, es decir:

$$v = r\omega \Rightarrow v = (1 \text{ cm})(110 \times 2\pi) / 60 \text{ seg} = 11.5 \text{ cm/seg}$$

La fuerza aplicada por cada uno de los dedos está determinada por el radio r del engranaje y la fuerza de torque τ especificada para el motor.

$$\tau = fr \Rightarrow f = \tau / r \quad f = (60 \text{ Ncm}) / (1 \text{ cm}) = 60 \text{ N}$$

El número de revoluciones está determinado por:

$$\# \text{ rev} = \theta (1 \text{ rev} / 2\pi \text{ rad})$$

$$\text{donde: } \theta = \Delta x / r \Rightarrow \theta = 8 \text{ cm} / 1 \text{ cm} = 8 \text{ rad}$$

de aquí:

$$\# \text{ rev} = (8 \text{ rad}) (1 \text{ rev} / 2\pi \text{ rad}) = 1.27 \text{ rev}$$

En el apéndice C, se muestran las piezas que conforman a la pinza tanto individualmente como en forma de "explosión", además de una perspectiva del dispositivo final.

DISEÑO ELECTRÓNICO

III.2 Diseño Electrónico

Para iniciar el proceso de un diseño electrónico, se debe partir de la realidad que se quiere automatizar. Para este fin, se plantea una subdivisión en lo que respecta al diseño electrónico en tres partes principales:

- Diseño del controlador del motor
- Diseño del sensor de distancia
- Diseño de los sensores de contacto

III.2.1 Diseño del controlador del motor

En este caso, se cuenta ya previamente con un motor que puede tener la función de activar un dispositivo mecánico, específicamente, una pinza. Dicho motor funciona con alimentación de CD de 5 volts, el rotor del motor puede girar en dos sentidos (horario y antihorario), dependiendo de la diferencia de potencial aplicada a sus terminales. Cuando el rotor trabaja en sentido horario, la pinza cierra; cuando trabaja antihorario, la pinza abre.

De aquí, se infiere que el controlador del motor debe generar tres estados posibles para la pinza:

- Abrir
- Cerrar
- Inactivo

La siguiente tabla muestra la acción ejecutada por el motor dependiendo de la diferencia de potencial aplicada.

$V_{M0} - V_{M1}$ (V)	Acción
0	Inactivo
5	Abrir
-5	Cerrar

Tabla III.2.1

donde $V_{M0} - V_{M1}$ representa la diferencia del potencial aplicada al motor en volts.

Cabe mencionar que el control del motor y los sensores se hará por medio de computadora, por lo que debe incluirse una etapa de potencia que sirva como interface. A continuación, se presenta una descripción del diseño de dicha etapa de potencia.

Uno de los arreglos más comunes utilizados para incrementar corriente es el arreglo Darlington. Este arreglo se encuentra constituido básicamente por dos transistores conectados en serie como lo ilustra la siguiente figura.

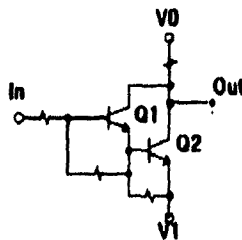


Fig. III.2.1

En el ejemplo, el transistor Q1 puede ser puesto en corte o saturación, dependiendo del valor de la entrada "In". Si Q1 es puesto en saturación, entonces drenará una cierta cantidad de corriente hacia V1, lo cual también le permitirá poner en saturación el transistor Q2 quien a su vez incrementará la capacidad de corriente a trabajar en "Out". Este tipo de arreglos se utilizan frecuentemente en los circuitos conocidos como drivers.

Teniendo en cuenta este arreglo, es posible hacer girar el motor en sentido horario conectando la referencia del motor (M0) a "Out" (Cuando los transistores se saturan lo conectarán a tierra), y su alimentación (M1) a 5V, de tal manera que se tendría lo siguiente:

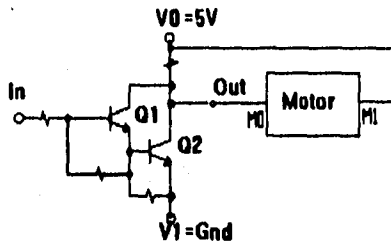


Fig. III.2.2

Sin embargo, por sencillez en la implementación del circuito, podría dejarse la referencia del motor conectada permanentemente a tierra y, la alimentación conectada al circuito que intercambiaría entre los tres valores posibles, es decir:



Fig. III.2.3

Esta última figura implica utilizar una fuente de alimentación simétrica para obtener los tres voltajes requeridos. Teniendo en cuenta estas consideraciones, la fig. III.2.2 actualizada quedaría de la siguiente manera:

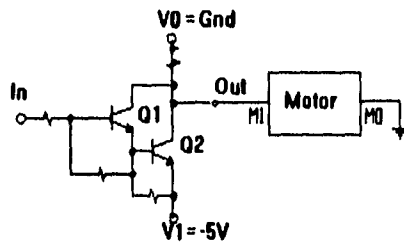


Fig. III.2.4

Funcionalmente este último diseño hace girar al rotor en sentido antihorario y no horario como en la fig. III.2.2. Para que se conserve dicha característica, es necesario que V_{M1} sea mayor que V_{M0} , es decir, $V_{M1}=5V$. Para que los transistores puedan ser puestos en operación, $V1$ debe ser menor que V_{in} y si se toma en cuenta que $V1=5V$ y que $5V$ es el máximo valor a manejar, no es posible conseguir el movimiento horario con este diseño. Para resolver este problema podrían utilizarse transistores PNP de tal forma que el diseño actualizado sería el siguiente.

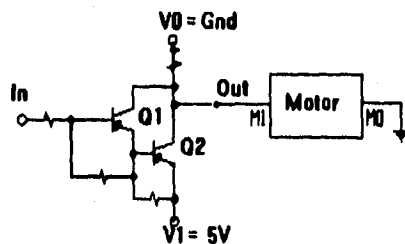


Fig. III.2.5

Si en la entrada "In" se tiene como valor GND, entonces los transistores se polarizarán y se irán a la saturación, por lo que M1 estará referenciado a 5V, permitiendo el movimiento horario.

Consideremos lo siguiente: los circuitos presentados van a ser controlados por una lógica digital. Los circuitos TTL manejan alimentaciones de 5V. En la fig. III.2.4 se puede ver que el estado de corte se consigue con un voltaje de -5V en "In" y la saturación llega con GND. Para la fig. III.2.5 el corte se consigue con 5V a la entrada y la saturación con GND. En la circuitería digital podría definirse como referencia GND y como Vcc (alimentación para los circuitos) 5V; empero, esto impediría poner en estado de corte al circuito de la fig. III.2.4.

Si se decidiera poner la referencia para la lógica digital en -5V y $V_{cc}=GND$, lo que sucedería ahora sería que no se podría poner el circuito de la fig. III.2.5 en corte, pues para ello se necesitaría un voltaje de 5V. La solución a este problema es establecer un circuito que convierta a un nivel de voltaje adecuado una señal "extra" cuando la lógica digital lo indique, ya que no puede generarla directamente.

Para diseñar este circuito, es posible utilizar el mismo esquema que se ha venido utilizando con algunas variantes, suponiendo que la referencia del circuito digital será establecida en -5V y que $V_{cc}=GND$, es decir, el problema a resolver sería para la fig. III.2.5.

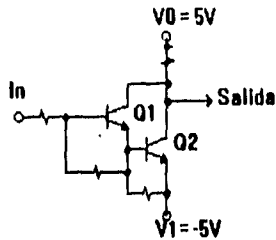


Fig. III.2.6

El funcionamiento del diagrama mostrado es el siguiente: para que la fig. III.2.5 pueda ser puesta en corte, se necesitan a su entrada 5V. En la fig. III.2.6, si $In = -5V$ provoca que la salida tenga un valor de 5V puesto que los transistores no están polarizados. En el caso contrario, cuando sea necesario que la fig. III.2.5 este en saturación, en su entrada deberá haber un voltaje menor a 5V. Este valor puede ser obtenido de la fig. III.2.6 si $In = GND$, es decir, su salida tendrá un valor de -5V, debido a que los transistores estarán polarizados.

Por cuestiones de mayor facilidad y debido a que cumple con los requerimientos máximos exigidos por el motor de CD (corriente máxima de 0.5 amper), se toma como modelo para establecer las características y valores de los elementos, la hoja de especificaciones de datos del circuito integrado XR-2002 así como el transistor BC547. Resumiendo, el diseño final, es el siguiente:

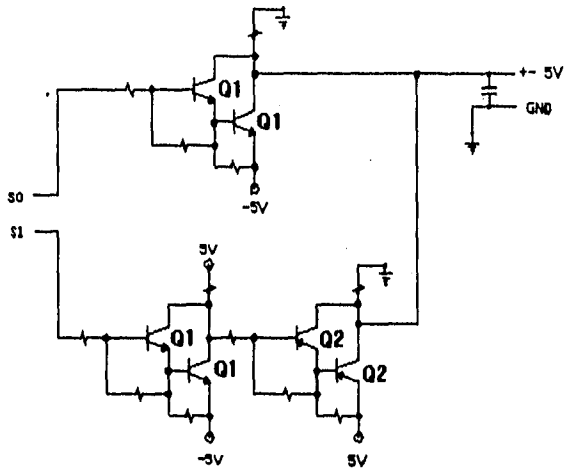


Fig. III.2.7

De esta manera, se encuentra ya solucionada la cuestión de la etapa de potencia que servirá como interface entre el sistema digital y el motor. Pasemos ahora con la fase del diseño digital propiamente dicho.

Tomando en cuenta la información mostrada por la tabla III.2.1, es posible generar el siguiente diagrama de flujo, el cual muestra la forma en que se sucederán los eventos:

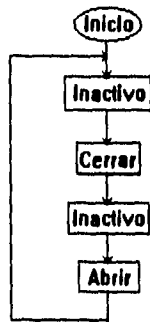


Fig. III.2.8

Como se puede observar, existe un ciclo. En este diagrama se hace la suposición de que inicialmente la pinza se encuentra abierta, e inactiva. La siguiente ilustración muestra la correspondiente carta ASM así como la tabla de estados:

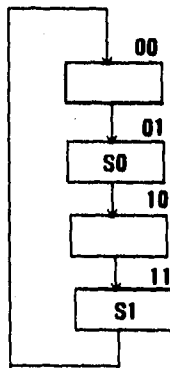


Fig. III.2.9

Nombre	Q(t)		Q(t+1)		Salidas
	B	A	B	A	
M	0	0	0	1	
N	0	1	1	0	S0
O	1	0	1	1	
P	1	1	0	0	S1

De este análisis se puede concluir que la sucesión de estados no va a variar bajo ninguna circunstancia, puesto que no depende de una variable de entrada en especial, es decir, se trata de una cuenta binaria ascendente de solo dos bits (B,A). Este hecho tan especial, hace posible el utilizar únicamente un contador binario para implementar el diseño. Obteniendo ahora las funciones de salida:

$$S0 = B'A$$

$$S1 = BA$$

Estas funciones podrían implementarse con compuertas AND y Negadores y para ello se necesitarían al menos dos circuitos integrados. Sin embargo, se puede reducir el número de circuitos utilizados si para la implementación se utiliza la técnica de multiplexores:

Mux S0(M)=0	Mux S1(M)=0
Mux S0(N)=1	Mux S1(N)=0
Mux S0(O)=0	Mux S1(O)=0
Mux S0(P)=0	Mux S1(P)=1

De tal manera que el circuito integrado a utilizar sería el 74LS153 que está compuesto por dos multiplexores de esta característica.

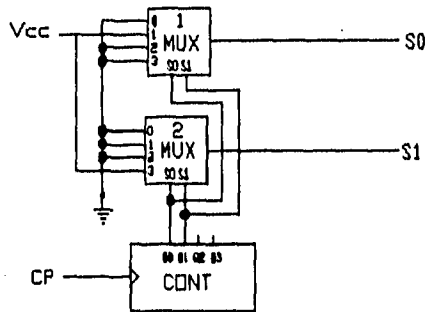


Fig. III.2.10

Es importante considerar que la señal de reloj (CP), puede ser mandada por la computadora o bien por alguno de los sensores internos de la pinza. Para este último caso, habrá sensores de tipo "push button" normalmente abiertos, que mandaran una señal permanente cuando la pinza haya llegado a un tope, ya sea abriendo o cerrando. Sin embargo, esta señal por ser permanente no genera el pulso que necesita el contador para pasar a un estado inactivo. El problema es entonces, a partir de una señal dada, generar un solo pulso de reloj que le permita al contador pasar a otro estado.

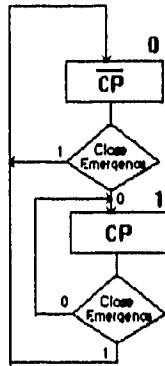


Fig. III.2.11

Q(t)	Entrada	Q(t+1)	Salida
A	Close Emergency	A	
0	1	0	CP'
0	0	1	CP'
1	0	1	CP
1	1	0	CP

Tabla de Estados

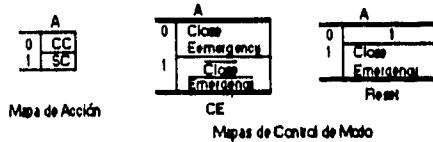
En el diagrama de estados, la entrada "Close Emergency" es el sensor interno de la pinza que determina en que momento el mecanismo debe parar su movimiento. Para este problema se ha elegido un diseño basado en contadores :

Reset	CE	Acción
*	0	Carga
0	1	Cuenta
1	1	Retener

Control de modo

Notación:

- CC Cuenta condicional
- CI Cuenta Incondicional
- SC Salto Condicional
- SI Salto Incondicional



$$\text{Reset} = A' + (\text{Close Emergency})'$$

$$\text{CE} = A'(\text{Close Emergency}) + A(\text{Close Emergency})'$$

Estas funciones pueden ya ser empleadas como lógicas de entrada a las señales de control Reset (Carga Paralela) y CE (Habilitador de cuenta) del contador. Sin embargo, tomando en cuenta las características del contador a utilizar, la lógica a alambrear puede aun simplificarse más con algunas pequeñas variaciones, de tal forma que si se omite la lógica asignada a CE por una condición OR entre el reloj del sistema y la salida TC (termino de cuenta) del contador 74LS191, se tendría lo siguiente:

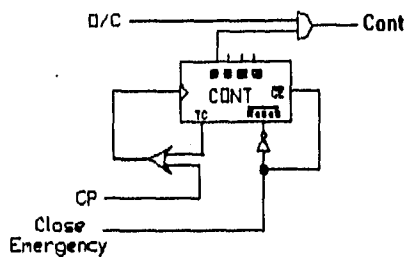


Fig. III.2.12

Es decir:

$$CE = (\text{Close Emergency})$$

$$\text{Reloj} = TC + CP$$

$$\text{Reset} = (\text{Close Emergency})'$$

De tal suerte el contador se deshabilita a si mismo al llegar a la cuenta máxima (1111) y ésta señal es dada por el mismo circuito, de ahí que se reduzca la lógica a implementar. La única restricción es que Reset debe cargar el número 1110.

A continuación se presenta el diseño final del sistema del controlador del motor:

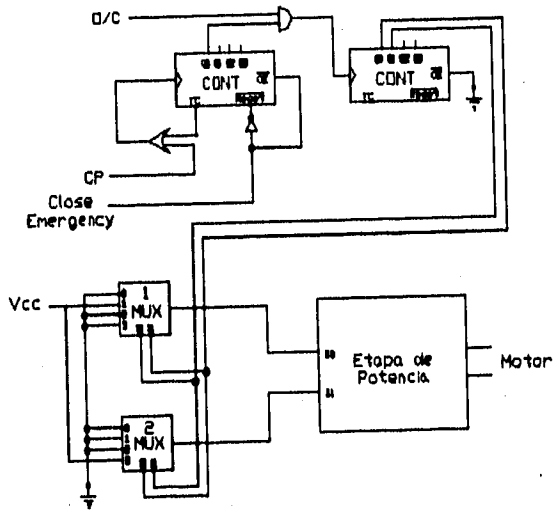


Fig. III.2.13

III.2.2 Diseño del Sensor de Distancia

El servomotor a utilizar cuenta con un codificador formado por un conjunto de ranuras que al giro del rotor pasan por un sensor óptico que genera una serie de pulsos que es la señal enviada por el codificador. Obviamente, el número de pulsos generado por el codificador tiene relación directa con el giro en grados del rotor y por lo mismo también tendrá relación con el desplazamiento sufrido por los dedos de la pinza.

De esta forma, es posible obtener la distancia recorrida por los dedos de la pinza al momento de detener su movimiento ¿cómo? Implementando un conjunto de contadores en cascada que tengan como señal de reloj los pulsos generados por el codificador.

El servoactuador es del tipo RH-5502-E036AO cuyas características son las siguientes:

RH-5502-E 036 A O

A B C D

A : Con codificador

B : Resolución del codificador 036=360P/rev

C : Voltaje del codificador A=5V

D : Tipo de salida del codificador O=colector abierto

La resolución al movimiento del rotor está dada por la resolución del codificador multiplicada por la reducción armónica del motor, es decir:

$$(360)(80)=28,800$$

en hexadecimal (7080)_{Hex}

A decir verdad, este número tan grande da una resolución bastante alta, lo cual en la práctica es imposible definir con tal exactitud. Considere lo siguiente: la pinza recorrerá en total 8 cm. (abertura máxima), para ello, el rotor habrá girado aproximadamente 360 grados (una vuelta completa). Entonces la resolución que se tendría sería $(360)/(28,800)=0.0125$, es decir, 125 diezmilésimas de grado lo cual equivaldría a 2.8 diezmilésimas de centímetro. Por lo anterior, considero oportuno incluir un divisor de frecuencia entre 2 a la salida de los pulsos del codificador.

Como el lector podrá observar en la parte correspondiente al diseño del software, el sistema estará limitado a utilizar cuatro bits de lectura y cuatro de escritura en su interface con la computadora, por lo que el número binario de 16 bits generado por los contadores, conviene que sea leído en forma serial, es decir, uno por uno. La computadora será quien se encargue de mandar la señal de reloj para indicar que sea leído el próximo bit.

Para la implementación del circuito, conviene conectar las salidas de los contadores a la carga paralela de 2 registros de corrimiento que manejen 8 bits. Obviamente, será necesaria una señal de control que active la carga paralela así como un reloj que sincronice a la computadora con la salida serial de los registros de corrimiento.

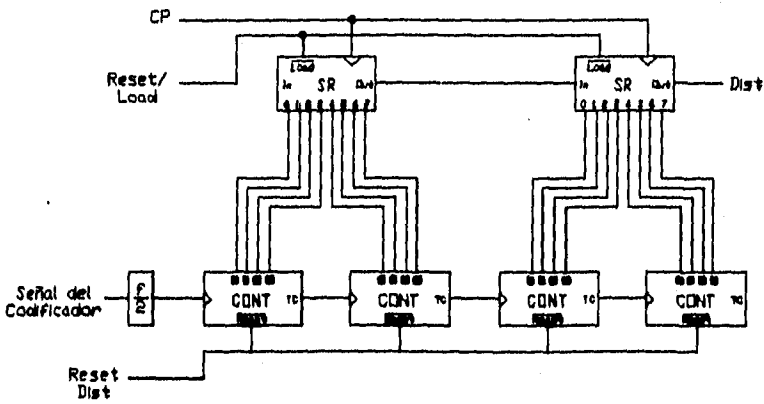


Fig. III.2.14

La figura III.2.14 muestra el diagrama del circuito a implementar, en el también se incluye una señal llamada "Reset Dist" que tiene como función inicializar los contadores con el número cero.

III.2.3 Diseño de los sensores de contacto

Una de las cuestiones a resolver es la de que el órgano terminal sea capaz de detectar el momento del contacto con un objeto así como caracterizar el tipo de contacto obtenido (superficie). Para ello, es necesario tener un arreglo de "células" que indiquen por sí solas cuáles de ellas están activadas y cuáles no, para de esta forma, determinar la superficie de contacto de acuerdo a la distribución de las señales de dichas células.

Si las células fueran totalmente independientes, entonces habría una señal por cada una de ellas que indicaría su estado. Piense, por restricciones de espacio, en un conjunto inicial de 64 células por cada dedo, serían necesarias 128 líneas ó cables para sensar el estado de estos elementos. Las líneas tendrían que estar conectadas a un número de circuitos suficiente para soportar todos los sensores. Podrían ser 16 multiplexores de 8 entradas cada uno. También haría falta una circuitería que habilitara y deshabilitara los multiplexores correspondientes. Pensando a futuro, algo semejante a lo propuesto sería terrible para implementar tanto en tiempo como en dinero.

Otra opción es la de realizar un "barrido" secuencial preguntando por el estado de cada una de las células. Propongamos un arreglo matricial de 8x8. Para la primera columna se preguntaría cuáles de sus ocho renglones están activados. Por supuesto, esto tendría que hacerse uno por uno empezando desde el renglón 1 hasta el 8 y para cada una de las columnas. El número de bits necesarios para direccionar las 64 células sería de 6 (3 para renglones y 3 para columnas).

Como de lo que se trata es de utilizar el menor número de líneas posible y en vista de que solo se elegirá un renglón de entre varios, podría utilizarse un

multiplexor de 8 a 1. Para trabajar una a una las columnas se "activaría" la columna correspondiente con una señal. En este caso se trataría de un switch que solo activaría una de ocho señales, como muestra la ilustración.

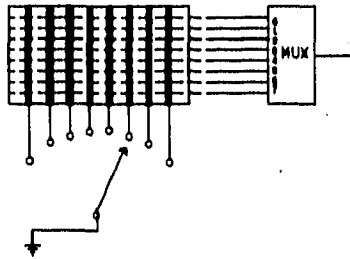


Fig. III.2.15

Para que un arreglo como el de la fig. III.2.15 pueda funcionar, es necesario que la línea 0 del MUX sea común a todas las columnas. Lo mismo sucede para las demás líneas del MUX. La teoría de funcionamiento es la siguiente: El switch conecta a tierra la columna cero, si alguna de las líneas o varias de ellas están en contacto con dicha columna, entonces a la entrada del MUX en esas líneas habrá como referencia una tierra; en las entradas restantes habrá Vcc (Valor de default para los circuitos TTL). De esta forma, es posible saber qué células de la columna cero están siendo oprimidas. El procedimiento se repite para las demás columnas.

Viendo ya hacia la implementación, los seis bits de direccionamiento para referenciar las células, podrían ser mandados directamente por la computadora, sin embargo, debido a restricciones del software (las cuales se abordan con amplitud en la sección de diseño de software), será mejor que la computadora solo maneje una señal de reloj que sincronice a dos contadores (uno para

columnas y otro para renglones), al tiempo que vaya leyendo los datos a la salida del MUX. Obviamente, también se necesitará una señal de inicialización para asegurar que los datos leídos sean de la célula correcta.

Hasta el momento y pensando nuevamente en la implementación, no hay mayor problema sobre como crear los elementos descritos, excepto por el "switch". Un decodificador tiene la función de habilitar solo una salida de entre varias. Pensemos en un decodificador de 8 líneas de salida. El elemento seleccionado será referenciado a tierra ó Vcc dependiendo de la lógica que maneje. Si consideramos el decodificador 74LS155 (el cual trabaja con lógica inversa), la salida seleccionada estará referenciada a tierra mientras que las demás serán referenciadas a Vcc.

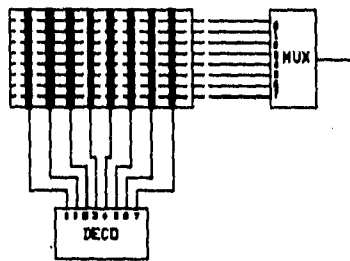


Fig. III.2.16

Considerando las características mencionadas para los sensores (un mismo renglón es común a todas las columnas), se corre el riesgo de provocar un corto, ya que un mismo renglón podría estar haciendo contacto en más de una columna, es decir, podría darse una unión directa entre Vcc y tierra.

Por ello, el diseño del "switch" debe prever que en las salidas no seleccionadas haya alta impedancia en vez de Vcc. Para lograrlo pueden utilizarse seguidores de tres estados, los cuales estarán referenciados a tierra y sus habilitadores se conectarán a las salidas del deco, como se muestra a continuación:

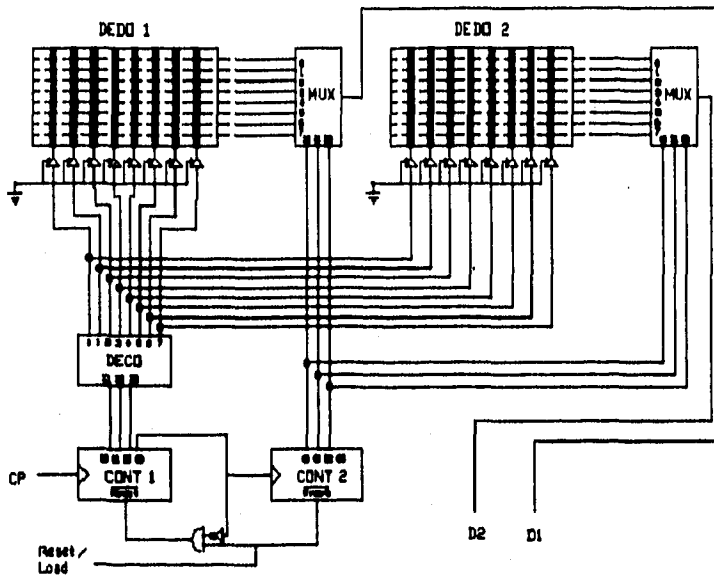


Fig. III.2.17

La fig. III.2.17 ilustra el diseño final para sensar el contacto en los dedos. Solo se incluye adicionalmente un negador y una compuerta AND que tienen la función de generar un pulso para el contador 2 cada vez que el contador 1 llega a un ciclo completo. Los detalles sobre la implementación de los sensores se tratan con amplitud en la sección correspondiente.

DISEÑO DE SOFTWARE

III.3 Diseño de Software

III.3.1 Puertos de comunicación

La función principal del software será controlar el funcionamiento y "percepciones" de la pinza mecánica, y para ello debe establecer un proceso de comunicación con los restantes sistemas. El realizar dicho proceso involucra la entrada y salida de datos entre la computadora y el sistema electrónico.

La entrada y salida de datos en la computadora se hace a través de sus puertos de comunicación (seriales y paralelos). Cuando la comunicación se establece vía puerto serie, los datos se transmiten secuencialmente, es decir, bit por bit. Es necesario establecer un puente de comunicación (cable), que contenga tres señales imprescindibles: transmisión (Tx), recepción (Rx) y una referencia (Gnd). El puerto serie es gobernado internamente por el controlador 8250. Dicho controlador debe ser configurado previamente en cuanto a las características de transmisión: velocidad, tamaño de la palabra, bit de paridad, bit de paro. El voltaje involucrado es de alrededor de 12 volts, por ello, el puerto serie resulta muy práctico para realizar comunicaciones entre dispositivos alejados.

Por otra parte, el puerto paralelo realiza una comunicación de datos síncrona, es decir, un conjunto de bits (palabra) pueden ser leídos o escritos simultáneamente. Este puerto es gobernado por el controlador 8255, el cual cuenta con las características de los circuitos TTL, por lo que maneja una diferencia de potencial de 5 volts. Internamente, el 8255 está compuesto a la salida por tres registros de 8 bits cada uno, además de un registro interno de 8 bits bidireccional¹ :

¹ Programmable Peripheral Interface, Intel.

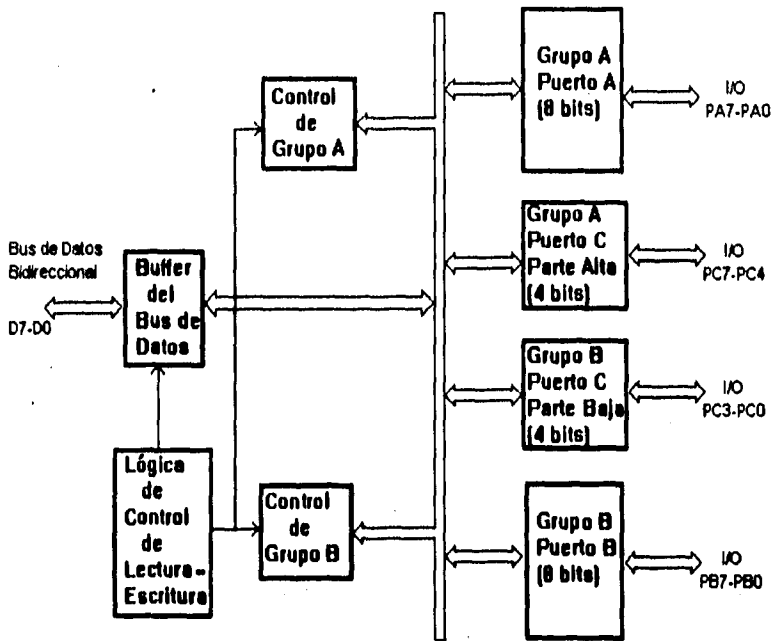


Fig. III.3.4

La fig. III.3.4 muestra la estructura interna de los registros. Los tres registros (A,B y C), son utilizables tanto para lectura como para escritura con la única diferencia de que el registro C puede programarse con funciones diferentes para cada una de sus partes. Esto se explica debido a la presencia de dos módulos controladores de cada grupo (A y B). El primer elemento controla el grupo A, el cual esta formado por el registro A y la parte alta del C. El segundo elemento controla el registro B y la parte baja del registro C. Como se puede observar en la figura, existe un bus de datos común a todos los registros y a un buffer de entrada/salida (para comunicación con el bus interno de la computadora). Cualquier orden de lectura, escritura ó configuración de los registros es especificada en la Lógica de Control de Lectura/escritura.

Cuando desde el sistema operativo se manda un trabajo a impresión, se activa la interrupción 17, la cual se encarga de escribir la orden en la Lógica de Control. Esta interrupción configura los registros internos del 8255 de manera que le permita mandar 8 bits de datos más otros tantos de control y comunicación con la impresora.

Cabe recordar que el problema a resolver en esta sección es establecer una comunicación entre la computadora y un sistema electrónico. Si se decidiera realizar la comunicación vía puerto serie, entonces resultaría ser muy laboriosa, pues habría que crear un reloj externo para sincronizar las señales además de crear una lógica digital más complicada para controlar estos procesos (o bien utilizar un microcontrolador con los consiguientes incrementos de costos).

Por esta razón, se ha elegido trabajar con el puerto paralelo y en especial con la instrucción `outportb` de lenguaje C, la cual permite escribir cuatro líneas de datos así como leer otras tantas. Pensé en utilizar esta instrucción para facilitar y hacer más rápida la implementación del software. De aquí que, el diseño del sistema electrónico se haya sujetado a las restricciones implicadas por esta instrucción.

¿Por qué lenguaje C? en la introducción se comentó algo referente a este punto. En efecto, existen varios lenguajes de desarrollo orientados a la robótica los cuales cubren necesidades muy específicas y con equipo especial. Las necesidades de este proyecto en particular, no van más allá del manejo de una rutina de comunicación con un dispositivo electrónico de creación propia, por lo que la utilización de lenguaje C resulta práctica y sencilla por contener los

elementos necesarios para la solución del problema, además de ser uno de los lenguajes de desarrollo de mayor auge en el mundo actual de la computación².

III.3.2 Análisis del sistema

El software entonces, está involucrado directamente en lo relativo al control. Obviamente, para realizar un diseño es necesario partir del modelo con el cual se va a interactuar. La siguiente figura ilustra el modelo a bloques del sistema a controlar.

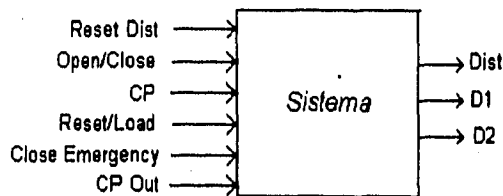


Fig. III.3.1

Este sistema aún puede desglosarse un poco más, esto es, traemos de diferenciar la interacción de los tres módulos principales:

² Probablemente por ser mi área de trabajo (ingeniería en computación), el diseño y la implementación electrónica se vieron supeditados a la investigación que previamente había realizado sobre la forma de comunicar la computadora con otros dispositivos externos, esto es, primero averigüé los detalles del área en la cual me desenvuelvo y, de acuerdo a las restricciones que implicaba, fue que realicé la parte electrónica. Bajo otras condiciones, si el proyecto se hubiera realizado por especialistas en cada materia, por ser la robótica un área multidisciplinaria, entonces el control (software) seguramente se habría supeditado a lo ya creado por los otros dos grupos (mecánica y electrónica).

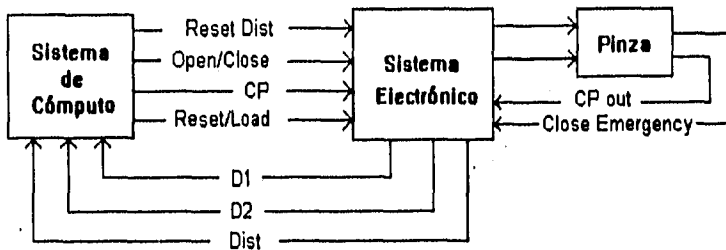


Fig. III.3.2

Como se puede observar en la fig. III.3.2, el sistema de cómputo se relaciona con el sistema electrónico por medio de 7 señales:

Reset Dist. Esta señal reinicializa a cero el conteo que se va haciendo de la distancia recorrida por la pinza (debe activarse antes de cada movimiento).

Open/Close. Es un pulso de reloj mandado por la computadora que da la señal para que la pinza inicie un nuevo movimiento.

CP. Es una señal de reloj que sincroniza la lectura del estado de los sensores de contacto. También se utiliza para la lectura de la distancia recorrida.

Reset/Load. Esta señal tiene un doble propósito: 1) inicializar los contadores de mapeo de los sensores y 2) cargar en los registros de corrimiento la distancia recorrida por la pinza.

D1. Contiene el estado del sensor (m,n) del dedo 1.

D2. Contiene el estado del sensor (m,n) del dedo 2.

Dist. Da la lectura en forma serial de la distancia recorrida.

Estas son las señales que van a permitir conseguir los objetivos establecidos: abrir y cerrar la pinza, obtener espesor del objeto apesado y caracterizar la superficie de contacto en los dedos de la pinza. De aquí, el algoritmo a seguir sería el siguiente:

- 1) Inicializar variables y sistemas mecánico y electrónico (pinza, contadores y registros).
- 2) Permanecer en estado de espera hasta que el usuario indique que desea que la pinza cambie de estado.
- 3) Inicializar contadores de mapeo de sensores y de distancia recorrida.
- 4) Iniciar movimiento en la pinza.
- 5) No parar el movimiento hasta que los sensores detecten un contacto o bien que se haya llegado a la abertura total de la pinza..
- 6) Desplegar el estado de los sensores.
- 7) Desplegar el espesor del objeto apresado.
- 8) Ir al paso 2.

El algoritmo anterior es ahora representado en el siguiente diagrama de flujo.

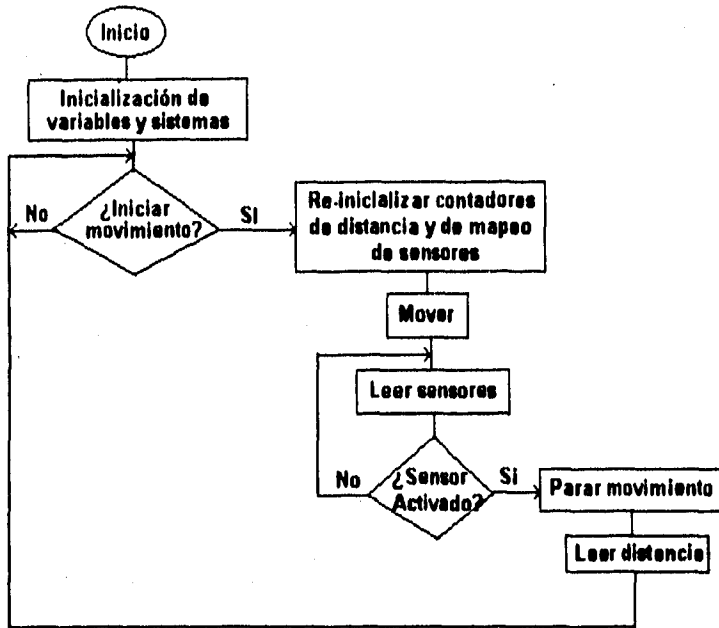


fig. III.3.3

Los detalles de la implementación se especifican en la sección IV.3.

IMPLEMENTACIÓN MECÁNICA

IV.1 Implementación Mecánica

En esta sección se abordan los detalles correspondientes a la implementación de las partes mecánicas, tratándose específicamente de la secuencia de armado.

El diagrama de la figura IV.1.1 muestra un esquema de explosión del dispositivo mecánico. De acuerdo a dicho diagrama, la implementación se dio en el siguiente orden:

Secuencia	Conjunto	Piezas
1	A	3-6-12
	B	4-7
	C	14-10
2	D	A-5-13
	E	C-11
3	F	D-8
	G	E-9
4	H	B-G
5	I	F-H
6	J	I-2-1

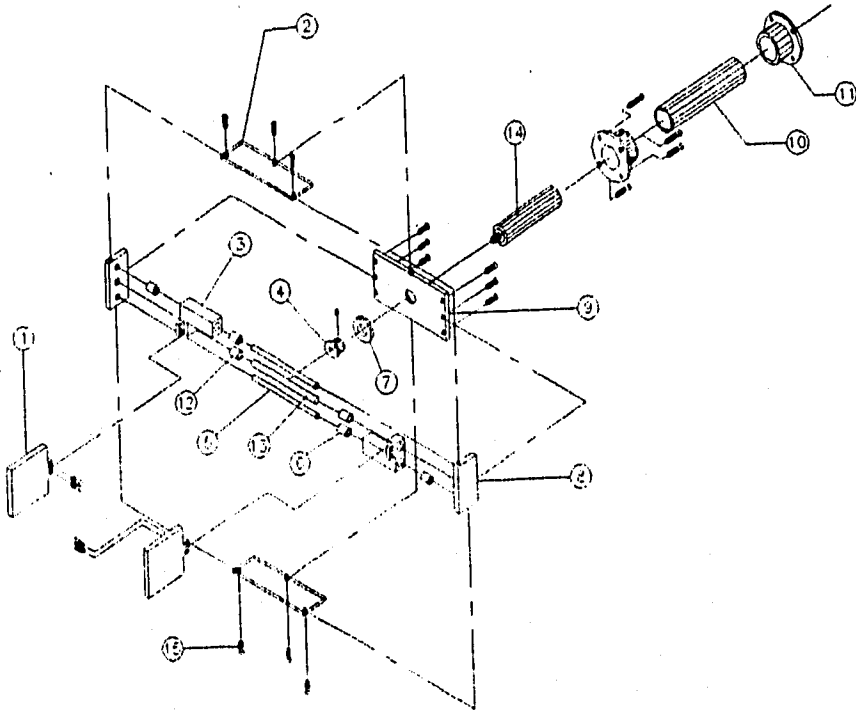
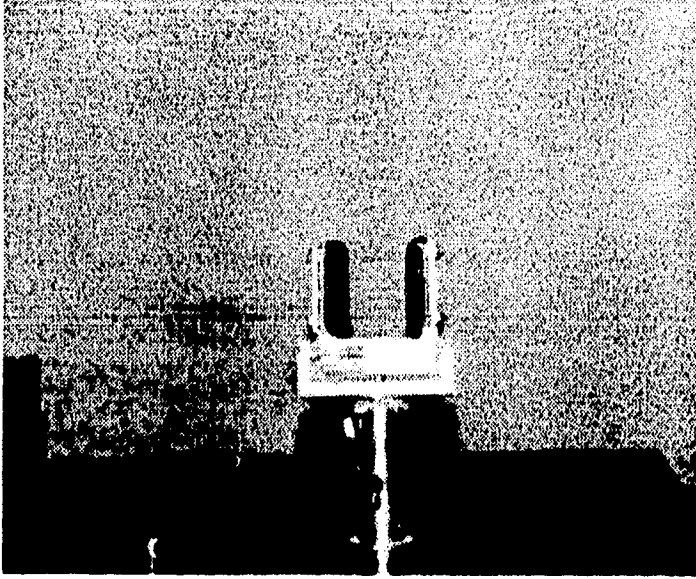


Fig. IV.1.1

Imagen del dispositivo final.



IMPLEMENTACIÓN ELECTRÓNICA

IV.2 Implementación Electrónica

En la sección correspondiente al diseño electrónico se abordaron tres aspectos principales: el controlador del motor, sensor de distancia y los sensores de contacto. La presente sección describe los detalles de tales implementaciones.

IV.2.1 Controlador del motor

La implementación del diseño final del controlador del motor (fig. III.2.7 y fig. III.2.13), es mostrada por el circuito de la fig. IV.2.1. Cabe recordar que la etapa de potencia esta constituida por tres arreglos tipo Darlington, dos arreglos con transistores NPN y el restante con transistores PNP. Gracias a las características del circuito integrado XR-2004, fue posible implementar los arreglos NPN de manera sencilla, sin embargo, el arreglo PNP tuvo que ser implementado con dos transistores BC547.

Para la lógica digital se utilizó un MUX 74LS153, dos contadores 74191 y los restantes circuitos corresponden a las compuertas AND, OR y NOT.

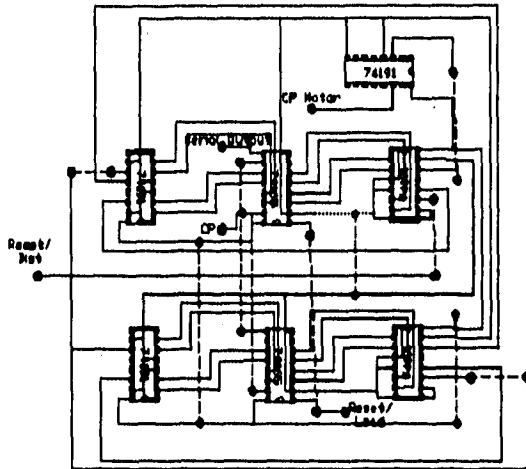


Fig. IV.2.2

IV.2.3 Sensores de Contacto

A decir verdad, ésta fue una de las etapas más críticas en cuanto a diseño e implementación. La implementación tuvo doble grado de dificultad, el primer problema fue diseñar un circuito impreso que tuviera el menor número de puentes posible con pistas lo suficientemente delgadas que permitieran a todos los circuitos comunicarse entre sí y que a su vez no invadieran el campo destinado a otras pistas. El segundo fue crear los sensores de contacto tan "idealmente" diseñados a un bajo costo.

Con respecto a esto último, fueron varios los intentos antes de llegar a la implementación final. La idea de funcionamiento ya estaba establecida: se trataba de un arreglo de ocho renglones y ocho columnas que serían sensados

secuencialmente por el circuito mostrado arriba. La cuestión era determinar de qué material serían hechos estos renglones y columnas.

El punto inicial de partida fue un circuito impreso que contenía ocho pistas como muestra la fig. IV.2.3.



Fig. IV.2.3

En la ilustración, las líneas anchas representan las pistas de cobre y, el resto de la superficie, es la bakelita. Después, a lo largo de cada una de las pistas se realizaron ocho perforaciones equidistantes pensando que en esas perforaciones podrían más adelante ubicarse una serie de resortes en forma de renglones, como muestra la ilustración:

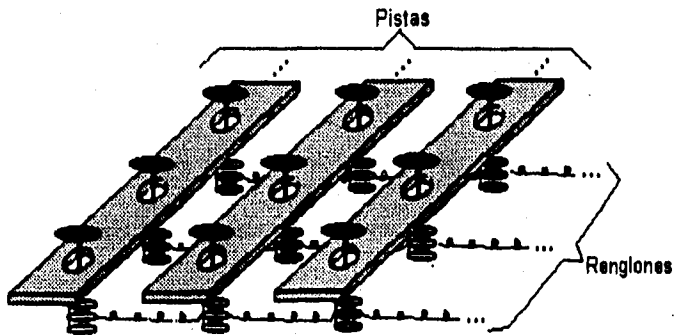


Fig. IV.2.4

El funcionamiento es el siguiente. En el tiempo t_1 , se hace pasar una señal por alguna de las pistas, si existe un objeto haciendo presión sobre un resorte, entonces la placa del extremo del resorte hará contacto con la pista, por lo que la señal escrita en la pista podrá ser leída en el cable correspondiente al renglón al que pertenece el resorte. Obviamente este proceso tendría que repetirse para cada uno de los renglones y para todas las pistas.

El funcionamiento entonces, de estos resortes, es similar al de un "push button". Por ello, la idea original fue implementar estos dispositivos con "push buttons", sin embargo, había varios inconvenientes. El arreglo sería de 8x8 en cada uno de los dedos, es decir, se necesitaban comprar 128 push buttons para implementar los sensores. El precio individual de cada botón era de N\$2.00 aproximadamente (antes de la devaluación), entonces el gasto implicado era de N\$256.00, lo cual representaba un gasto muy considerable.

Otro inconveniente era que se necesitaba que los sensores tuvieran la mayor sensibilidad posible para detectar el contacto y, los push buttons no garantizaban de ninguna manera esta característica amén de la problemática de implementación en cuanto a tamaño y soldadura, por lo que la idea fue desechada.

Entonces, surgió la posibilidad de implementar literalmente el diseño hecho con resortes. No tardó mucho en aparecer el primer inconveniente: ¿Cómo lograr una perfecta unión entre el resorte y la placa que se pensaba utilizar de manera que no se resquebrajara ante la presión? Los puntos de soldadura no eran lo suficientemente fuertes para soportar la presión además de que no podían ser muy grandes pues impedirían el libre movimiento a través del orificio hecho en la placa. Otra opción era la de deformar uno de los extremos del resorte para

utilizarlo como placa de contacto, sin embargo, la misma flexibilidad del resorte hacía que constantemente se deformara. La idea también fue desechada.

Tomando en cuenta la experiencia de los resortes, fue posible concluir que había dos requisitos muy claros e imprescindibles: 1) el material que daría forma a la placa que haría contacto con la pista debía tener cierta rigidez para que no se deformara con la presión y 2) el elemento que contrarrestara la presión en la placa de contacto debía ser flexible para detectar pequeños contactos además de que se pudiera unir fácilmente a la placa de contacto.

La característica 1 podía ser cubierta utilizando cable telefónico. Realicé pequeñas imitaciones de resortes con los cables usados en las tabletas experimentales, obteniendo un elemento más rígido que los resortes tradicionales. Obviamente no había nada de flexibilidad en ese elemento. Realmente no era necesaria. Para cumplir con la segunda característica, dos viles y simples globos fueron suficientes. ¿Cómo? agregando únicamente a la ilustración de la fig. IV.2.4 una capa extra en forma de "piso". Este "piso" era del material elástico obtenido de los globos, de tal forma que se obtuvo lo siguiente:

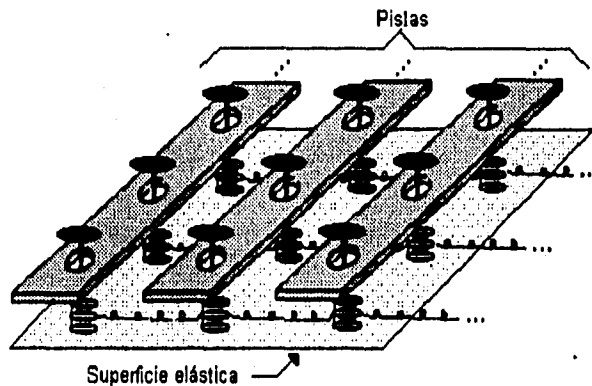


Fig. IV.2.5

En el esquema, la superficie elástica se encarga de ejercer la presión necesaria para que los nuevos "resortes" recuperen su posición inicial después de haber sido oprimidos. El inconveniente es que la presión ejercida por la superficie elástica no es uniforme en todas las regiones de la placa, por lo que la sensibilidad en los extremos de la placa resulta menor que en el centro, además de que los resortes suelen atorarse en los orificios. La idea también fue desechada

Finalmente llegué a la conclusión de que el hecho de hacer pasar un elemento a través de un orificio siempre iba a implicar cierta fricción por no ser el material muy apropiado para este tipo de movimientos. Como un nuevo intento, traté ahora con materiales radicalmente diferentes como papel aluminio y esponja. ¿Por qué estos materiales? porque el papel aluminio tiene la propiedad de ser un elemento conductor y además flexible y la esponja por ser muy suave al tacto.

Para la implementación, se cortó un cuadro de aproximadamente 5x5 cm en una esponja de .5 cm de ancho para que funcionara como "piel" para después unir a una de las superficies ocho tiras de papel aluminio en forma de renglones. La unión del aluminio con la esponja fue por medio de hilo de coser (sin ningún tipo de pegamento para no volver rígida a la esponja). Gracias a la presión del hilo de coser sobre los renglones de aluminio, el espesor en la superficie más interna de la esponja se redujo, por lo que en su periferia el ancho era mayor, permitiéndole al aluminio tener una cierta elevación con respecto a la superficie en la cual estaría apoyada la esponja.

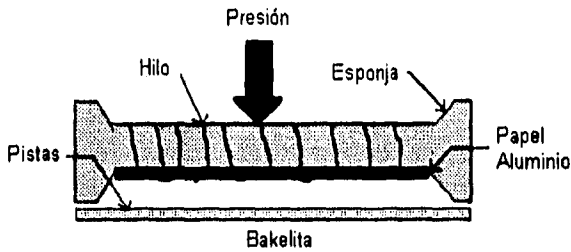


Fig. IV.2.6

Cuando se ejerce una fuerza sobre la superficie de la esponja, ésta se deforma haciendo que el papel aluminio entre en contacto con las pistas de la bakelita, por lo que se establece un circuito cerrado entre el aluminio y las pistas. Esta última idea dio paso a lo que sería la implementación final de los sensores con algunas pequeñas modificaciones:

Sobre cada una de las pistas de la superficie de bakelita soldé a lo largo un cable de cobre de un espesor aproximado de 1 mm para después ubicar entre pista y pista una pequeña tira de esponja, de tal forma que la esponja sobresalía de entre los cables de cobre. Esto permitió poder ubicar una serie de "renglones de aluminio", que tenían como soporte las tiras de esponja, en forma perpendicular a las pistas de la bakelita, de tal suerte que al haber una presión sobre cualquier tira de aluminio hacía que la esponja se comprimiera y entrara en contacto directo el aluminio con el cobre, como ilustra la fig. IV.2.7:

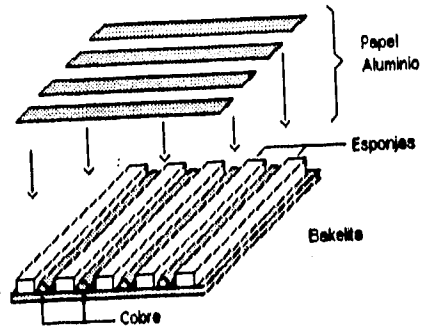


Fig IV.2.7

Esta fue la implementación final de los sensores de contacto por cumplir con las características inicialmente planteadas: bajo costo y sencillez de elaboración. A continuación se presenta el circuito electrónico asociado para el mapeo de los sensores cuyo diseño fue abordado en el capítulo III.

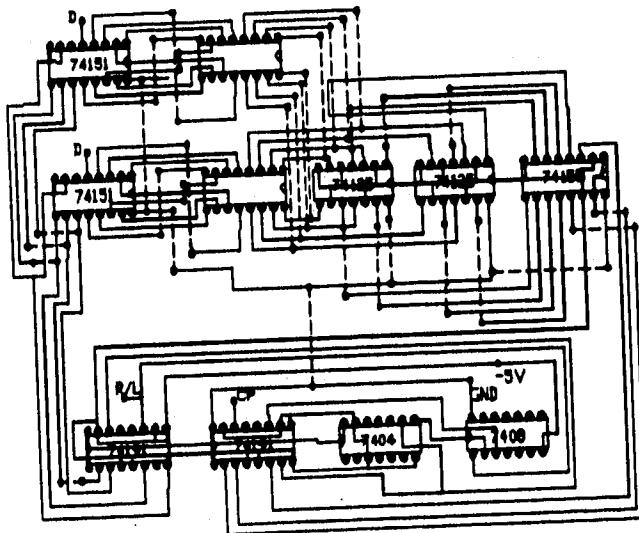


Fig. IV.2.8

IMPLEMENTACIÓN DE SOFTWARE

IV.3 Implementación de Software

En esta sección se presenta el código final del diseño abordado en el capítulo III correspondiente al software. La implementación final fue hecha en lenguaje de programación C. El proyecto está compuesto por tres archivos de encabezados para declaración de constantes y prototipos (headers) y cuatro archivos fuente.

```
.....
                                     Pinza.h
.....
                                     // Prototipos de funciones

// En la escritura al puerto, el bit más significativo
// corresponde a la señal CP. La distribución de bits de
// control es la siguiente:
//           Ms           CP
//           |           O/C
//           |           Reset/Load
//           Ls           ResetDist

// Para la lectura tenemos la siguiente correspondencia de
// bits:
//           Ms           Idle
//           |           D1
//           |           D2
//           Ls           Dist.

void Reoj(int Pulsos,int Periodo);
// Manda "n" pulsos de periodo "T" al
// puerto paralelo (CP)

void OpenClose();           // Escribe la señal de abrir o cerrar (O/C)

void ResetLoad();          // Resetea los contadores de la interface
// electrónica (Reset/Load)

void ResetDist();         // Resetea los contadores asociados al
// encoder del motor (ResetDist)

void ReadDist();          // Lee bit por bit en el puerto la Infor-
// mación mandada por los registros de
// corrimiento (Dist)

unsigned char ReadFinger(); // Lee la información de los sensores de
// los dedos

unsigned char Gripper();  // Prepara y pone en movimiento la mano
```



```
void Actualiza();           // Lee secuencialmente la información
                           // aportada por los sensores (D1 y D2)

void Prepara();           // Prepara pinza

unsigned char EdoSensor(); // Verifica estado del sensor

void home();              // Inicializa todos los sistemas

void SalidaError();      // Salida con mensaje de error
```

Raton.h

```
unsigned detectaEleccion(); // Detecta de acuerdo a la posición del
                           // ratón cual fue la opción elegida

void posicion(int *x, int *y); // Detecta la posición del cursor del ratón

int edoBoton();           // Verifica si se han oprimido los botones

void esperaPresion();     // Hace pausa hasta que se presione un
                           // botón

void esperaPresionIzq();  // Espera a que se presione el botón Izq
```

Mio.h

```
#define ipt1 0x378        // Direcciones de los puertos
#define ipt2 0x278

#define data 0           // Indices para los registros del puerto
#define status 1
#define control 2

#define esdx 15         // Posiciones para las ventanas
#define esdy 6
#define esd2x 50

                           // Rutina de creación de marcos
void CreateWindow(char PosEDx, char PosEDy, char Ancho, char Largo);

void DoScreen();        // Traza la ventana de trabajo y
                           // prepara el ratón

void beep();           // Tono agudo

void beep20();         // Tono grave
```

.....
Main.c
.....

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include "mio.h"
#include "raton.h"
#include "pinza.h"

extern int port; // Puerto a trabajar (Default Ipt1)
extern float distancia;
enum { NOBOTON, BOTONIZQ, BOTONDER, DOSBOTONES}; // Estados del ratón
union REGS r;
void Menu(); // Menú de trabajo
void Bye(); // Salida
void Guarda();

void main() // Bloque de control
{
    DoScreen();
    Prepara();
    Menu();
}

void Menu() // Menú de trabajo
int Opcion;

gotoxy(15,30);
while(1){
    while((edoBoton() != BOTONIZQ)&&!(kbhit()));
    if (edoBoton()==BOTONIZQ){
        while (edoBoton()==BOTONIZQ);
        Opcion=detectaEleccion();
    }else{
        Opcion=getch();
    }
    switch (Opcion){
        case '0' : Bye();
            break;
        case '1' : Gripper();
            break;
        case '2' : Actualiza();
            break;
        case '3' : Guarda();
            break;
        case '4' : ReadDist();
            OpenClose();
            while (kbhit()) Reloj(1,20);
    }
}
```

```

        break;
        default : beep2();
    }
}

// Rutina de salida

void Bye(){
char buffer[4096];
int i,x,y;
    gettext(1, 1, 80, 25, buffer); // Guarda pantalla de trabajo
    beep();
    CreateWindow(20,9,7,40);
    for (i=10;i<16;i++){
        gotoxy(21,i);cprintf("
    ");
    };
    textattr(0x74);
    gotoxy(28,11);cprintf(" REALMENTE DESEA SALIR ? ");
    gotoxy(33,13);cprintf(" [S]/[N]O : ");
    while((edoBoton() != BOTONIZQ)&&(!kbhit()));
    if (edoBoton()==BOTONIZQ){
        while (edoBoton()==BOTONIZQ);
        i=detectaEleccion();
    }else{
        i=getch();
    }
    if ((i=='s')||(i=='S')){
        clrscr();
        printf("
                Gracias por usar este programa.");
        r.x.ax = 0x02; // Oculta cursor del raton
        int86(0x33, &r, &r);
        exit(0);
    }
    puttext(1, 1, 80, 25, buffer); // Restablece pantalla
    gotoxy(45, 11);
}

// Almacena datos leidos en el archivo de
// salida especificado

void Guarda(){
char buffer[4096],d1,d2;
char nuevo[64];
static char archivo[50]="datos.dat";
unsigned char Lecture=0xFF;
static char entro=0;
int i,j;
FILE *f;
    if (entro){
        gettext(1, 1, 80, 25, buffer); // Guarda pantalla de trabajo
        beep();
        CreateWindow(20,9,7,40);
        for (i=10;i<16;i++){

```

```

        gotoxy(21,i);printf("                ");
    };
    textattr(0x74);
    gotoxy(28,11);cprintf("Archivo de almacenamiento ? ");
    gotoxy(33,12);cprintf("(%)s:",archivo);
    gotoxy(31,13);cprintf("Nuevo Archivo: ");
    if (strlen(gets(nuevo)))
        strcpy(archivo,nuevo);
    if ( (f=fopen(archivo,"w")) == NULL ){
        gotoxy(28,11);cprintf(" Error da escritura en el archivo");
        gotoxy(28,12);cprintf("                ");
        gotoxy(33,13);cprintf("                ");
        while (kbhit());
        exit(1);
    }else
        fclose(f);
    puttext(1, 1, 80, 25, buffer); // Restablece pantalla
    entro=1;
}

f=fopen(archivo,"a");
fprintf(f,"\n%f",distancia);
ResetLoad();

for (i=0;i<8;i++){
    for (j=0;j<8;j++){
        Lectura=inportb (port+status);
        *(nuevo+j+i*8)= (Lectura & 0x20)?' ':';
        Reloj(1,0);
    }
}

for (i=0;i<8;i++){
    putchar('\n',f);
    for (j=0;j<8;j++)
        putc("(nuevo+j*8+i),f);
}

putc('\n',f);
for (i=0;i<8;i++){
    for (j=0;j<8;j++){
        Lectura=inportb (port+status);
        *(nuevo+j+i*8)= (Lectura & 0x40)?' ':';
        Reloj(1,0);
    }
}

for (i=0;i<8;i++){
    putchar('\n',f);
    for (j=0;j<8;j++)
        putc("(nuevo+j*8+i),f);
}

putc('\n',f);
fclose(f);
}

```

Ratonc.c

```

.....
Ratonc.c
.....
#include <dos.h>
extern union REGS r;
extern enum { NOBOTON, BOTONIZQ, BOTONDER, DOSBOTONES};

void posicion(int *x, int *y){ // Obtiene posición del ratón
    r.x.ax = 0x03;
    int86(0x33, &r, &r);
    *x = r.x.cx;
    *y = r.x.dx;
}

void esperaPresion() {
    while(edoBoton() != NOBOTON) // Verifica que ningún botón esté
        // presionado
        while(edoBoton() == NOBOTON) // Espera hasta que se presione un botón
            ;
        while(edoBoton() != NOBOTON) // Espera hasta que se libere
            ;
}

int edoBoton(){ // Obtiene estado de los botones
    r.x.ax=0x03;
    int86(0x33,&r,&r);
    return(r.x.bx);
}

void esperaPresionIzq() {
    while(edoBoton() != BOTONIZQ) // Espera que se presione el botón izq.
        ;
    while(edoBoton() == BOTONIZQ) // Espera que se deje de presionar
        ;
}

unsigned detectaEleccion(){ // Determina opción elegida de
    // acuerdo a la posición del
    // cursor en la pantalla
int x,y;
    posicion(&x,&y);
    if (x>=259 && x<=276){
        if (y>=84 && y<=89)
            return '1';
        else if (y>=72 && y<=77)
            return '2';
        else if (y>=80 && y<=85)
            return '3';
        else if (y>=88 && y<=93)
            return '0';
        else
            return '4';
    }else if ((x>=275 && x<=292)&&(y>=96 && y<=102))

```

```

        return 'S';
    else
        return '4';
}

```

.....

Pinzac.c

.....

```

#include <dos.h>
#include <stdio.h>
#include <time.h>
#include <math.h>
#include "pinza.h"
#include "mlo.h"
#define CONTACTO 1
#define NO_CONTACTO 0
#define TIEMPO_TOLERA 1
#define TIEMPO_ABERTURA 4
#define ABERTURA_PINZA 4.6
extern enum { NOBOTON, BOTONIZQ, BOTONDER, DOSBOTONES};
char Comando=0xF;
int port=1pt1;
int factor;
float distancia;

void Prepara(){
char buffer[4096];
int i,x,y;
    gettext(1, 1, 80, 25, buffer); // Guarda pantalla de trabajo
    beep();
    CreateWindow(20,9,7,40);
    for (i=10;i<16;i++){
        gotoxy(21,i);cprintf("
    );
    textattr(0x74);
    gotoxy(28,11);cprintf(" PUERTO A TRABAJAR ? ");
    gotoxy(33,13);cprintf(" [2] / [1] : ");
    while((edoBoton() != BOTONIZQ)&&!(kbhit()));
    if (edoBoton()==BOTONIZQ){
        while (edoBoton()==BOTONIZQ);
        i=detectaEleccion();
    }else{
        i=getch();
    }
    if ((i=='2')||(i=='S')){
        port=1pt2;
        gotoxy(66,22);printf("LPT2");
    }
    textattr(0x70);
    gotoxy(28,11);cprintf("Inicializando Dispositivos.");
    gotoxy(33,13);cprintf("Espere por favor");
}

```

```

textatrr(0xF4);
gotoxy(50,13);cprintf("!!!");
home();
puttext(1, 1, 80, 25, buffer); // Restablece pantalla
if ((i=='2')||(i=='S')){
    gotoxy(66,22);printf("LPT2");
}
gotoxy(45, 11);
}

void ReleJ(int Pulsos,int Periodo ){
int i,j;
    for (j=0;j<Pulsos;j++){
        Comando=Comando&(0x7); // Apaga el bit Ms (3)
        outportb (port+control, Comando);
        delay(Periodo/2);
        Comando=Comando|(0x8); // Restablece el bit
        outportb (port+control, Comando);
        delay(Periodo/2);
    }
}

void OpenClose(){
int Periodo=400;
    Comando=Comando&(0xB); // Apaga el bit 2
    outportb (port+control, Comando);
    delay(Periodo/2);
    Comando=Comando|(0x4); // Restablece el bit
    outportb (port+control, Comando);
    delay(Periodo/2);
}

void ResetLoad(){
int Periodo=400;
    Comando=Comando&(0xD); // Apaga el bit 1
    outportb (port+control, Comando);
    delay(Periodo/2);
    Comando=Comando|(0x2); // Restablece el bit
    outportb (port+control, Comando);
    delay(Periodo/2);
}

void ResetDist(){
int Periodo=400;
    Comando=Comando&(0xE); // Apaga el bit 0 (Bit menos significativo)
    outportb (port+control, Comando);
    delay(Periodo/2);
    Comando=Comando|(0x1); // Restablece el bit
    outportb (port+control, Comando);
    delay(Periodo/2);
}

void ReadDist(){

```

```

int i,j;
double x=0;
int Distancia[16];
static double anterior;
unsigned char Lectura;
    ResetLoad();           // Carga el contenido de los contadores
                           // a los registros de corrimiento
    for (i=3;i<16;i+=4){
        for (j=0;j<4;j++){
            Lectura=inportb (port+status);
            if ((Lectura&(0x10))==0x10)
                Distancia[i-j]=1;
            else
                Distancia[i-j]=0;
            Reloj(1,0);
        }
    }
    for (i=0;i<16;i++)
        x=x+pow(((double)2,(double)i)*Distancia[i]);
    gotoxy(50,19);
    /*for (i=15;i>=0;i--)
        printf("%d",Distancia[i]);*/
    if (x<anterior)
        anterior= (pow(((double)2,(double)16)-anterior)*(-1));
    if (ABERTURA_PINZA>(x-anterior)/10500*ABERTURA_PINZA)
        anterior=ABERTURA_PINZA-(x-anterior)/10500*ABERTURA_PINZA;
    else
        anterior=0;
    printf("%lf",anterior);
    distancia=anterior;
    anterior=x;
}

```

```

unsigned char ReadFinger0(
unsigned char Lectura;
unsigned char i,j,k=0;

    textatrr(0x0B); //define color de los dedos (blanco con fondo negro)
    for (i=0;i<8;i++){
        for (j=0;j<8;j++){
            Lectura=inportb (port+status);
            gotoxy(esdx+i*2,esdy+j);
            if ((Lectura&(0x20))==0x20) // Lee Dedo 1
                cprintf("%c%c",176,176);
            else{
                cprintf(". ");
                k=CONTACTO;
            }
        }
        gotoxy(esd2x+i*2,esdy+j);
        if ((Lectura&(0x40))==0x40) // Lee Dedo 2
            cprintf("%c%c",176,176);
        else{
            cprintf(". ");
            k=CONTACTO;
        }
    }
}

```



```

    }
    Reloj(1,0);        // Pasa a otro dato
}
}
return k;
}

unsigned char EdoSensor()
unsigned char Lectura=0xFF;
int i;

for (i=0;i<=64 && (Lectura & 0x20) && (Lectura & 0x40);i++){
    Lectura=inportb (port+status);
    Reloj(1,0);
}
if (!( (Lectura & 0x20) && (Lectura & 0x40) ))
    return CONTACTO;

return NO_CONTACTO;
}

unsigned char Gripper()
char i=0,EdoAnt;
time_t t;
textatrr(0x8A);
//ResetDist(); // Inicializa contadores de distancia
ResetLoad(); // Prepara contadores de mapeo de sensores
gotoxy(65,2);cprintf("Trabajando ...");
if (EdoSensor()==CONTACTO){
    while (EdoSensor()==CONTACTO && i<3){
        t=time(NULL);
        OpenClose(); // Inicia movimiento
        if (EdoSensor()==CONTACTO) Reloj(200,10);
        gotoxy(1,1);printf("ya");
        i++;
    }
    if (EdoSensor()==CONTACTO){
        clrscr();
        SalidaError();
    }
    gotoxy(65,2);cprintf(" ");
    while(!ReadFinger() && !kbhit() && edoBoton()==NOBOTON) Reloj(10,10);
    if (ReadFinger()) { beep();OpenClose();}
}else{
    i=0;
    EdoAnt=NO_CONTACTO;
    while ((EdoAnt==NO_CONTACTO) && (i<3)){
        //ResetDist();
        ReadDist();
        t=time(NULL);
        OpenClose();
        while (((EdoAnt=EdoSensor())==NO_CONTACTO) && ((time(NULL)-
t)<TIEMPO_ABERTURA));
    }
}
}

```

```

        i++;
    }
    if (EdoAnt == NO_CONTACTO){
        clrscr();
        SalidaError();
    }
    beep();
    OpenClose();
    ReadDist(); // Registra la distancia recorrida
}
gotoxy(65,2);cprintf("      ");
Actualiza();
return i;
}

void Actualiza(){
    ResetLoad(); // Prepara contadores de mapeo de sensores
    ReadFinger(); // Lee sensores
}

void home(){
    unsigned char i=0,EdoAnt;
    time_t t;
    ResetDist(); // Inicializa contadores de distancia
    ResetLoad(); // Prepara contadores de mapeo de sensores
    do{
        if (EdoSensor()==CONTACTO){
            while (EdoSensor()==CONTACTO && i<4){
                t=time(NULL);
                OpenClose(); // Inicia movimiento
                if (EdoSensor()==CONTACTO) Reloj(200,10);
                i++;
            }
            if (EdoSensor()==CONTACTO){
                clrscr();
                SalidaError();
            }
            t=time(NULL);
            while(!EdoSensor() && ((time(NULL)-t)<(TIEMPO_ABERTURA*2))) Reloj(10,10);
            if (EdoSensor()) { beep();OpenClose();i=1;}
            else{
                gotoxy(27,13);cprintf("Oprima Enter para continuar");
                beep20;
            }
            //while(!kbhit()) Reloj(10,10);
        }else{
            i=0;
            EdoAnt=NO_CONTACTO;
            while ((EdoAnt==NO_CONTACTO) && (i<4)){
                //ResetDist();
                ReadDist();
                t=time(NULL);
                OpenClose();
            }
        }
    }while(1);
}

```

```

        while (((EdoAnt=EdoSensor())==NO_CONTACTO) && ((time(NULL)-
1)<TIEMPO_ABERTURA));
            i++;
        }
        if (EdoAnt == NO_CONTACTO){
            clrscr();
            SalidaError();
        }
        beep();
        OpenClose();
        ReadDist(); // Registra la distancia recorrida
    }while ( EdoSensor() == CONTACTO);
}

void SalidaError(){
    beep2();
    gotoxy(28,11);cprintf(" Error Critico      ");
    gotoxy(27,13);cprintf("Revisar Dispositivos físicos");
    while(!kbhit());
    clrscr();
    exit(1);
}

```

.....

Anexo.c

.....

```

#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include "mlo.h"
extern union REGS r;

void DoScreen(){
    int i;

    textbackground(1); // Definición de colores de trabajo
    textcolor(14);
    clrscr();
    CreateWindow(5,4,11,70);
    CreateWindow(47,18,2,21);
    gotoxy(48,17);printf("Abertura de la pinza:");
    gotoxy(47,22);printf("Puerto de trabajo: LPT1");
    gotoxy(5,17);printf("- Abrir/Cerrar permite cambiar el ");
    gotoxy(7,18);printf("estado de movimiento en la pinza");
    gotoxy(5,19);printf("- Actualizar hace un nuevo mapeo");
    gotoxy(7,20);printf("en el contenido de los sensores");
    gotoxy(5,21);printf("- Guardar almacena la información");
    gotoxy(7,22);printf("mostrada en un archivo");
    textattr(0x1A);
    gotoxy(5,17);cprintf("- Abrir/Cerrar");
    gotoxy(5,19);cprintf("- Actualizar");
    gotoxy(5,21);cprintf("- Guardar");
    textattr(0x30); //define color de los dedos
}

```

```

for (i=5;i<15;i++){
gotoxy(6,i);cprintf("                ");
}
CreateWindow(1,1,23,79);
textattr(0x1A);
gotoxy(22,2);cprintf(" <<< INGENIERÍA V1.0 >>> ");
textattr(0x1E);
gotoxy(22,3);cprintf(" Mando de un Organó terminal ");
textattr(0x1F);
gotoxy(28,2);cprintf("<<<");
gotoxy(50,2);cprintf(">>>");
textattr(0x0B); //define color de los dedos (blanco con fondo negro)
for (i=0;i<8;i++){
for (j=0;j<8;j++){
gotoxy(esdx+i*2,esdy+j);
cprintf("%c%c",176,176);
gotoxy(esd2x+i*2,esdy+j);
cprintf("%c%c",176,176);
}
textattr(0x30);
gotoxy(8,9);cprintf("Dedo 1");
gotoxy(8,9);cprintf("Dedo 2");
gotoxy(33,7);cprintf(" <- SENSORES -> ");
gotoxy(33,9);cprintf("[1] Abrir/Cerrar ");
gotoxy(33,10);cprintf("[2] Actualizar ");
gotoxy(33,11);cprintf("[3] Guardar ");
gotoxy(33,12);cprintf("[0] Salir ");
r.x.ax = 0x01; //dibuja cursor
int86(0x33, &r, &r);
beep20;
}

```

```

void CreateWindow(char PosEDx,char PosEDy,char Ancho,char Largo){
unsigned char i,j;
textattr(0x70); // Define color de la ventana
for (i=PosEDx;i<(PosEDx+Largo);i++){
gotoxy(i,PosEDy);
cprintf("%c",205);
gotoxy(i,PosEDy+Ancho);
cprintf("%c",205);
};
for (i=PosEDy;i<(PosEDy+Ancho);i++){
gotoxy(PosEDx,i);
cprintf("%c",186);
gotoxy(PosEDx+Largo,i);
cprintf("%c",186);
};
gotoxy(PosEDx,PosEDy);
cprintf("%c",201);
gotoxy(PosEDx+Largo,PosEDy);
cprintf("%c",187);
gotoxy(PosEDx,PosEDy+Ancho);
cprintf("%c",200);
gotoxy(PosEDx+Largo,PosEDy+Ancho);

```

```
    cprintf("%c",188);  
}  
  
void beep(){  
    sound(1100);  
    delay(100);  
    nosound();  
    delay(40);  
    sound(1100);  
    delay(100);  
    nosound();  
}  
void beep2(){  
    sound(150);  
    delay(100);  
    nosound();  
}  
}
```

INTEGRACIÓN

IV.4 INTEGRACIÓN

Desde el momento en que se planteó el problema tratado en este trabajo, se vislumbró la obvia necesidad de integrar los tres sistemas que conforman al proyecto: mecánica, electrónica y software. A decir verdad, la integración final, una vez construidos los módulos correspondientes independientemente, no representó mayor problema, pues todos los aspectos relativos a la integración se previeron en el diseño de cada uno de dichos módulos.

De esta manera, cabe recordar la relación de señales que había entre todos los módulos, ilustrado por la siguiente figura.

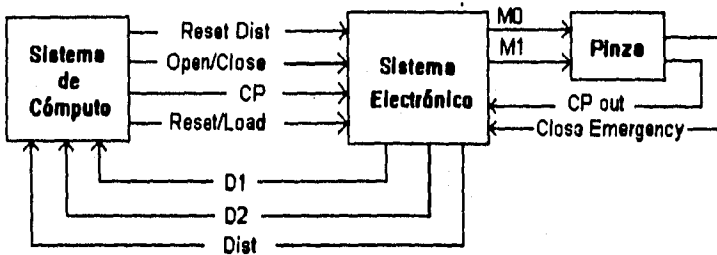


Fig. IV.4.1

Como se puede observar, el sistema electrónico es la unión entre la pinza y el sistema de cómputo, por lo que se implementó un circuito extra donde se unen todos los elementos:

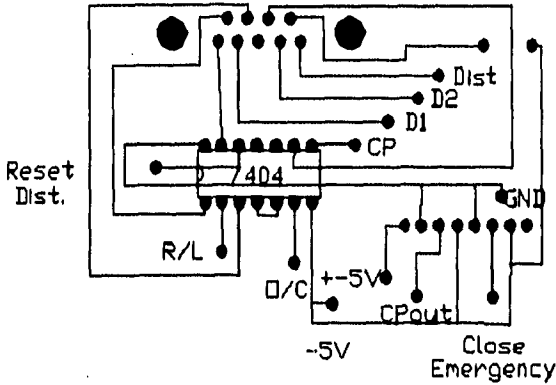
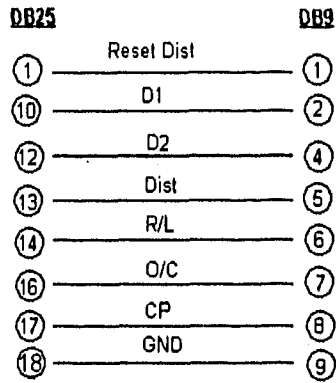


Fig. IV.4.2

En la ilustración se muestran las señales de datos D1, D2 y Dist que serán leídas por la computadora por medio de un conector DB9. La computadora también escribirá por medio del mismo conector las señales de control Reset Dist, R/L, O/C y CP las cuales irán directamente a un negador que funcionará como driver antes de enviar las señales al resto del circuito.

La comunicación física entre el puerto paralelo de la computadora y el conector DB9 del sistema digital es establecida por medio de un cable que tiene la siguiente correspondencia de señales:



En lo que respecta a la parte mecánica, la comunicación se hace por medio de un conector de ocho pines el cual manda las señales de alimentación al motor así como recibe los pulsos del encoder y las señales de los sensores propioceptivos.

RESULTADOS
Y
CONCLUSIONES

V. RESULTADOS Y CONCLUSIONES

En el capítulo III se abordó el diseño del proyecto en general especificando una serie de condiciones iniciales de trabajo. En los capítulos IV y V, se comentaron los detalles de las implementaciones. En la presente sección se analizan los resultados obtenidos del sistema funcionando.

En efecto, el proyecto consistió en construir una pinza sensorizada que funcionara como órgano terminal de un manipulador de 5 grados de libertad. La pinza ha sido construida y hé aquí algunas observaciones.

- Se ejecutó el software de control, el cual se encarga de mandar y leer señales para verificar que el sistema responda adecuadamente. El sistema respondió con un mensaje de error y finalizó inmediatamente el programa en los siguientes casos:

- 1) No había ninguna conexión física
- 2) Las conexiones físicas estaban establecidas pero el controlador electrónico no estaba energizado
- 3) Los buses de los sensores no estaban conectados
- 4) Existía algún problema mecánico (tornillo del engrane del eje del motor sin ajustar)

Todos estos casos fueron preparados previamente y el diagnóstico fue correcto, se detectó el error. Cuando todas las conexiones fueron establecidas correctamente, el tornillo del engrane del motor se ajustó y se energizó el

controlador electrónico, el software de control pasó a la siguiente etapa (permitir al usuario maniobrar con la pinza).

- Una vez pasada la etapa de inicialización, se probó el comportamiento de la pinza: abriendo, cerrando y sujetando objetos.

1) La pinza abrió y cerró en todos los casos de acuerdo a lo previsto, presentándose como único detalle la generación ocasional de un pulso extra por parte de los sensores propioceptivos (provocado por el mismo movimiento), lo cual llevaba a un estado de inestabilidad momentáneo en el movimiento de la pinza, del cual el sistema siempre se recuperó.

2) Al sujetar diversos objetos, los sensores de contacto trabajaron mostrando la información obtenida en la pantalla de la computadora, en la mayoría de los casos la superficies desplegadas fueron figuras muy cercanas a las "caras" del objeto que estaban en contacto con la pinza.

3) Se tomaron varias lecturas de un objeto de espesor conocido (1.5 cm). Los valores registrados fueron los siguientes:

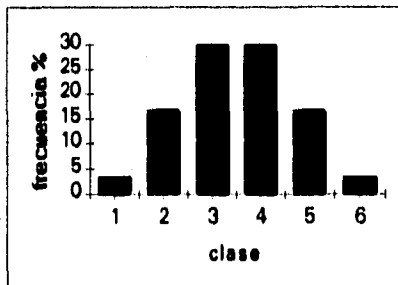
1.4	1.48	1.54
1.41	1.48	1.54
1.41	1.48	1.54
1.44	1.5	1.55
1.44	1.5	1.56
1.45	1.51	1.58
1.46	1.51	1.59
1.46	1.51	1.6
1.46	1.52	1.6
1.47	1.52	1.63

Del análisis estadístico de los datos mostrados, se obtuvieron los siguientes parámetros:

$\bar{X} = 1.504667$ (Media aritmética)

$\nu_x = 1.495977$ (Media de la distribución de medias)

$\sigma_x = 0.013726$ (Desviación estándar de la distribución de medias)



Gráfica de distribución muestral

donde:

Clase	Rango
1	1.36 - 1.40
2	1.41 - 1.45
3	1.46 - 1.50
4	1.51 - 1.55
5	1.56 - 1.60
6	1.61 - 1.65

La media de la población en un cierto intervalo de confianza se define por:

$$\bar{X} - Z_c \sigma_x < \nu < \bar{X} + Z_c \sigma_x$$

Para un intervalo de confianza del 95 % tenemos un error probable de:

$$Z_c \sigma_x = (2.042) (0.013726) = 0.028$$

4) Finalmente, la máxima abertura total de la pinza fue de 5 cm.

CONCLUSIONES

-Las lecturas de los espesores de los objetos apesados fueron un tanto variadas, sin embargo, mantenían un comportamiento constante, es decir, la pinza mostró una ligera tendencia a descalibrarse pero mantuvo el mismo nivel de precisión sobre dichos desajustes. Esto fue debido principalmente por una pequeña anomalía que se presentó en el acoplamiento del engrane con el eje del motor provocado por el mismo movimiento (el tornillo de ajuste va sufriendo pequeñas deformaciones), por lo que en determinado momento, fue necesario utilizar la opción de ajustar en el software de control. Para mejoras posteriores en la pinza se recomienda utilizar un tipo de material menos maleable.

- La abertura total de la pinza no fue la esperada inicialmente, debido principalmente a la presencia del sensor propioceptivo que en conjunto con el espesor de los dedos (sensores exteroceptivos), redujeron en aproximadamente 3 cm la abertura total.

- Durante cierto momento se presentó una inestabilidad en el movimiento de la pinza debido a la fricción que se daba entre los bloques en desplazamiento y el sensor propioceptivo. Aunque no tuvo mayor relevancia en el funcionamiento final de la pinza, podría darse una mejora en la implementación de dicho sensor utilizando probablemente un dispositivo óptico en vez de uno de contacto.

- Finalmente, puede decirse que se cumplieron en forma general los objetivos planteados inicialmente: se tiene un órgano terminal capaz de tomar y caracterizar objetos por medio de dedos sensorizados y un servomotor.

**REFERENCIA
BIBLIOGRÁFICA**

VI. REFERENCIA BIBLIOGRÁFICA

- Comp Robotics, Frequently asked questions (FAQ)
Kevin Dawling
Carnegie-Mellon University, Pittsburgh
<http://ri.cmu.edu>
email: nivek@cmu.edu

- Figueredo J.
Elastic-link manipulators: Modelling, simulation and experiments
Vol 11, pag. 13-21
International Journal of Robotics and Automation, 1995

- Honda Satoshi
Development of an intelligent vibration gripper
Vol 2595, pag. 170-177
Science Univ. of Tokyo, Jpn, 1995

- Bright G.
Generic Gripper - grabs your components
Vol. 15
Assembly Automation, 1995

- Billatos Samir
Novel approach to flexible robotic assembly systems
Vol 13
Robotics, 1995

- Boylestad Robert, Nashelsky Louis
Electrónica, teoría de circuitos
4a. edición
Prentice Hall

- Ogata Katsuhiko
Ingeniería de control moderna
1a. edición
Prentice Hall

- Clark Christopher
Designing Logic Systems using state machine
McGraw Hill, 1970

- Fletcher William
An engineering approach to digital design
McGraw Hill

- Mata Gloria
Prácticas de laboratorio de medición e instrumentación
Departamento de Ingeniería de Control,
Fac. de Ingeniería, U.N.A.M.

- Canavos George
Probabilidad y estadística, Aplicaciones y métodos
McGraw Hill, 1987

- Beer Ferdinand, Johnston Russell

Mecánica Vectorial para Ingenieros (Dinámica)

4a. edición

McGraw Hill

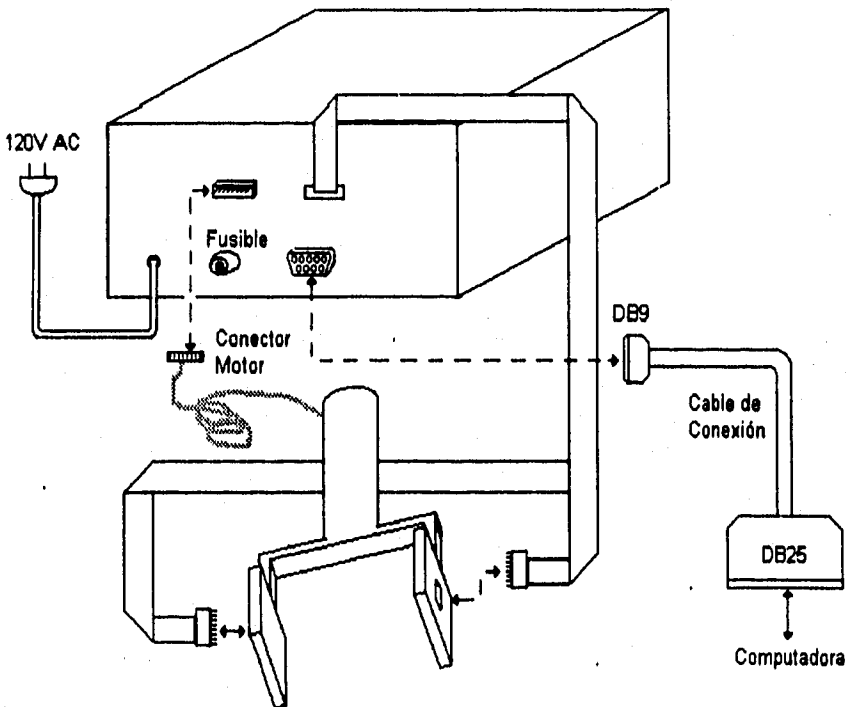
A) MANUAL DE USUARIO

Manual de usuario

Preparación del equipo.

Para poner en funcionamiento el equipo por primera vez se recomienda seguir las siguientes indicaciones:

- 1-. Verificar que el equipo de cómputo y el sistema digital se encuentren apagados.
- 2-. Conectar a la parte posterior del chasis del sistema digital el conector de ocho entradas que proviene de la pinza como lo ilustra la fig.1.
- 3-. Conectar a las placas de los dedos de la pinza los cables planos que provienen del chasis del sistema digital (ver fig.1)



4-. Conectar el cable de comunicación al puerto paralelo de la computadora con el conector DB9 del sistema digital.

5-. Finalmente, conectar a la alimentación de 120 V el cable proveniente de la fuente.

Puesta en operación.

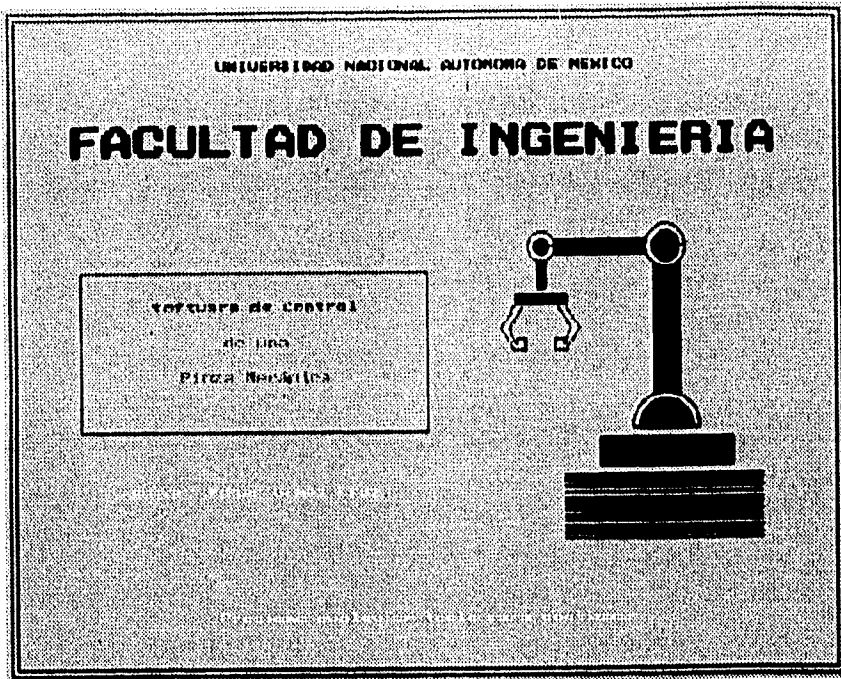
Una vez cumplida la fase anterior podrá continuar con los siguientes pasos:

1-. Oprimir el botón de encendido del sistema digital. Si las señales de AC y DC se encienden entonces el sistema se encontrará energizado (en caso contrario consultar la sección correspondiente a solución de problemas en el apéndice B).

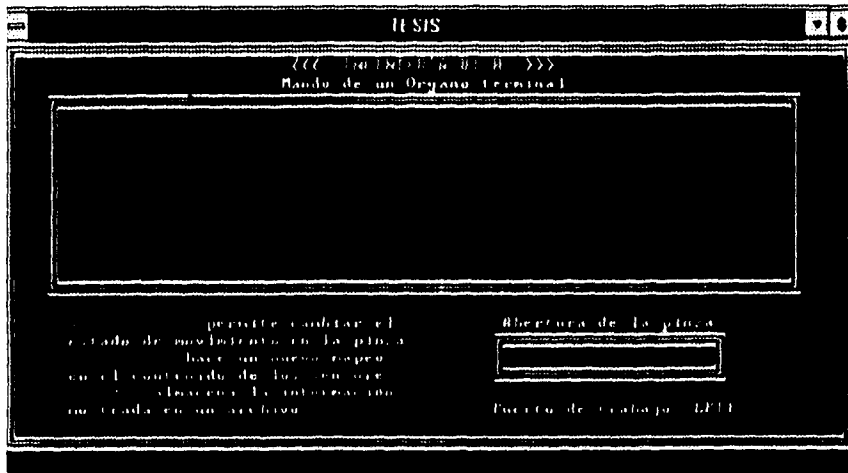
2-. Para poder interactuar con la pinza mecánica, es necesario ejecutar desde el sistema operativo el software de control llamado INGENIERÍA v1.0 de la siguiente manera:

`c:\> tesis`

3-. Al ejecutarse INGENIERÍA v1.0 aparecerá el siguiente mensaje de bienvenida:



4-. Para abandonar esta pantalla, sólo deberá oprimir cualquier tecla. Posteriormente se efectuarán una serie de rutinas previas de verificación e inicialización del sistema. En caso de no haberse encontrado ningún error, aparecerá una pantalla de trabajo, de lo contrario, un mensaje le alertará sobre anomalías en el sistema. Para mayor información refiérase al apéndice B (manual de mantenimiento).



Esta pantalla presenta cuatro opciones principales:

- 1) Abrir/Cerrar
- 2) Actualizar
- 3) Guardar
- 4) Salir

Abrir/Cerrar permite cambiar el estado de la pinza. Si la pinza se encuentra actualmente abierta entonces al seleccionar esta opción iniciará su movimiento hasta que haya cerrado totalmente ó bien haya apresado algún objeto. De lo contrario, si se encuentra cerrada, entonces la pinza abrirá.

Actualizar hace que se lea nuevamente la información captada por los sensores y actualiza la pantalla.

Guardar permite almacenar los datos mostrados en pantalla en un archivo. La primera vez que seleccione esta opción, **Guardar** le pedirá que introduzca el nombre del archivo en que se almacenarán los datos. En posteriores selecciones, no se le solicitará el nombre del dispositivo de salida, los datos se agregarán automáticamente al final del archivo que especificó inicialmente.

Salir permite finalizar la ejecución de INGENIERÍA v1.0.

Para seleccionar cualquiera de las opciones basta con oprimir la letra encerrada entre corchetes ó bien hacer 'click' en la zona correspondiente con el mouse

Existen tres secciones en la pantalla que muestran la información obtenida de la pinza: dedo1, dedo2 y abertura de la pinza. En dedo1 y dedo2 se muestra el estado de los sensores de contacto y en abertura de la pinza, la distancia que existe entre los dedos hasta el momento de detenerse.

La combinación de la información aportada por este programa desde diferentes puntos del espacio tridimensional puede servir para esbozar la forma del objeto apresado.

**B) MANUAL DE
MANTENIMIENTO**

Manual de Mantenimiento

Antes y después de la operación de todo el sistema deberán seguirse una serie de pasos para prevenir daños (los cuales se describen en el apéndice A). Sin embargo, existe la posibilidad de falla debido al uso constante del equipo. A continuación se plantean algunas fallas probables así como su correspondiente solución. Cabe mencionar que no está de más hacer una revisión periódica de los elementos tratados como posibles fallas aun y cuando no se hayan presentado inconvenientes.

I-. Revisar constantemente que los sensores trabajen correctamente. Para ello, será necesario seleccionar la opción de Actualizar en el software de control cuando la pinza se encuentre abierta y presionar la superficie de los dedos para corroborar que la presión es detectada en la región adecuada., en caso contrario, verificar los siguientes elementos:

1) Apagar y encender el sistema electrónico y observar nuevamente el trabajo de los sensores (la sincronización en la lectura de los sensores pudo haberse perdido si hubo algún incidente en la alimentación de la energía eléctrica o bien se provocó algún pequeño "corto" en el sistema).

2) Debe existir continuidad en los conectores de los extremos del cable plano de 16 hilos que une los sensores de la pinza con el sistema electrónico. Revise que se encuentren bien conectados. Si la continuidad falla reemplace el cable (el movimiento constante podría afectar en un cierto momento la calidad de contacto de los conectores).

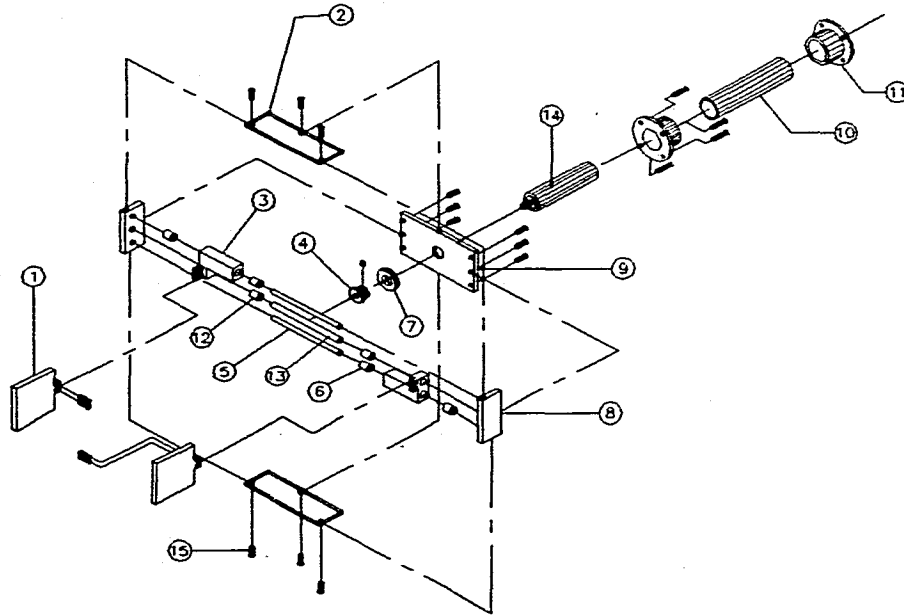
3) Si los anteriores elementos fueron verificados y la falla persiste, reemplace la capa de sensores por una nueva.

II-. Si después de preparar todo el equipo para trabajar, el sistema mecánico no respondiera a las ordenes mandadas por la computadora, entonces habrán de revisarse los siguientes elementos:

- 1) Que estén encendidas las señales de AC y DC. Si no están encendidas ninguna de las dos verifique internamente el estado de la fuente de poder. Si la señal de AC funciona y la de DC no, entonces cambie el fusible actual por uno nuevo de 1Amp.
- 2) Si el punto anterior no fue el problema, revise que el conector de alimentación del motor se encuentre conectado correctamente.
- 3) Revisar el cable de comunicación DB9 a DB25 (en los detalles de la implementación se muestra la relación que debe haber entre ellos).
- 4) Finalmente, revise el estado interno de los puertos de su computadora.

C) PIEZAS MECÁNICAS

11-III

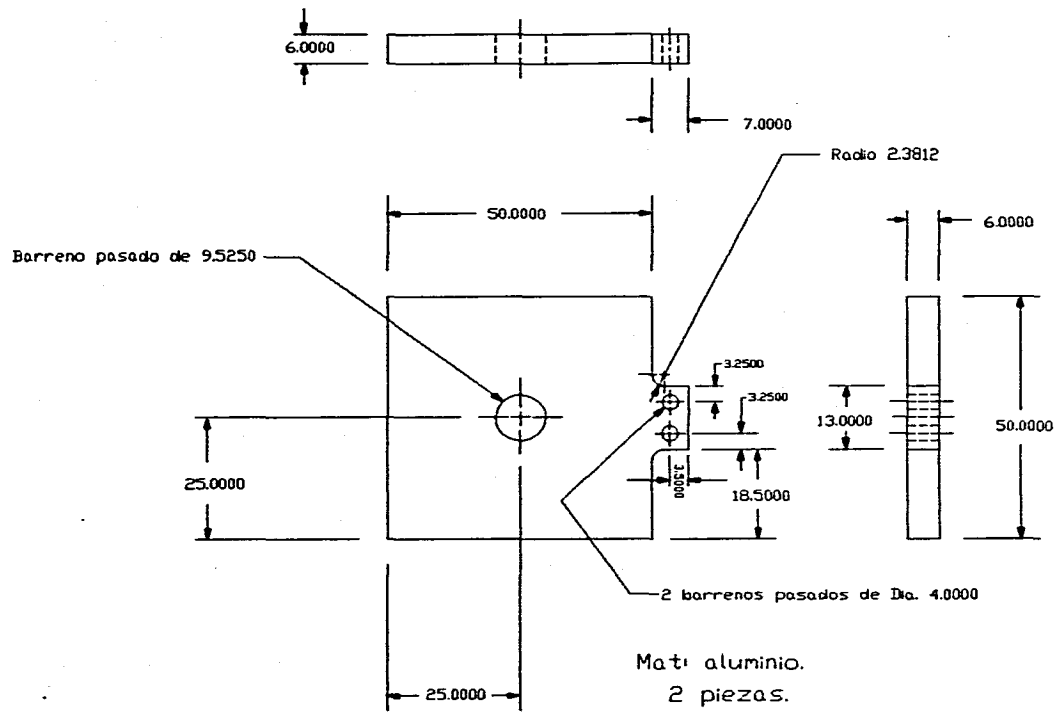


Dibujo de explosión		
FACULTAD DE INGENIERIA U.N.A.M.		Isometrico
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

Lista de partes

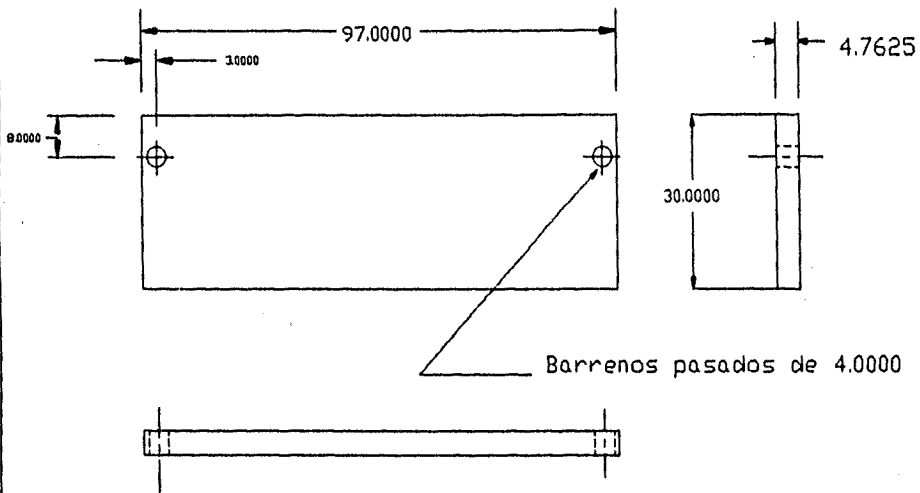
- 1) Placa dedo
- 2) Placa lateral
- 3) Bloque de desplazamiento
- 4) Acoplamiento
- 5) Guía lateral
- 6) Buje lateral
- 7) Engrane
- 8) Placa de soporte
- 9) Placa base
- 10) Cubierta de motor
- 11) Cilindro base
- 12) Buje central
- 13) Guía central
- 14) Motor RH-5502-E036AO

VII-13



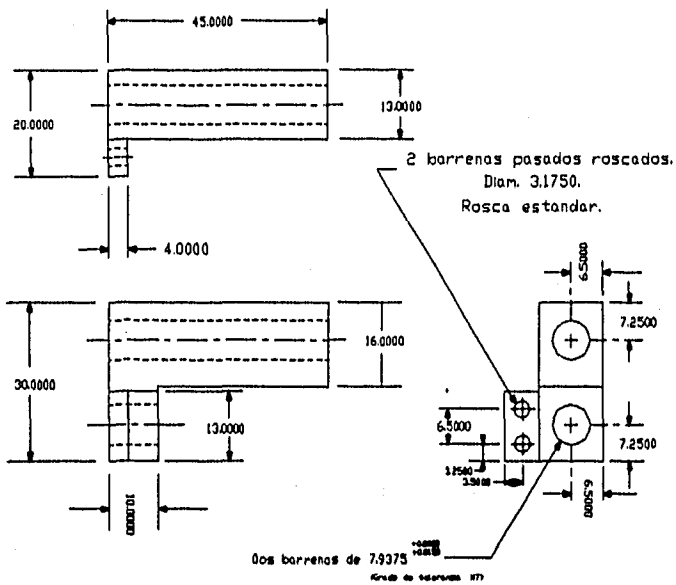
Mat: aluminio.
2 piezas.

Pieza 1: Placa Dedo		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proj. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1



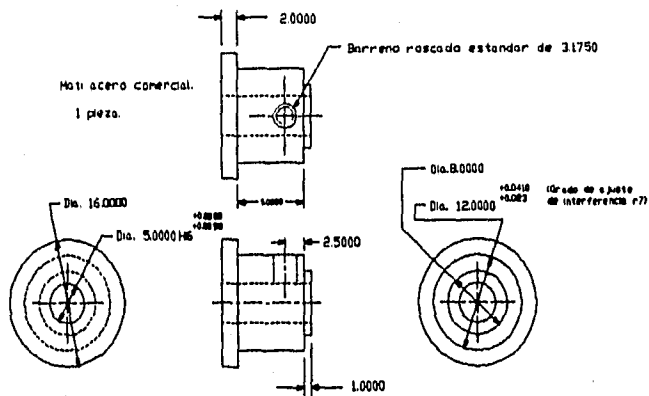
Materia: acrílico
 2 piezas

Pieza 2 : Placa lateral		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

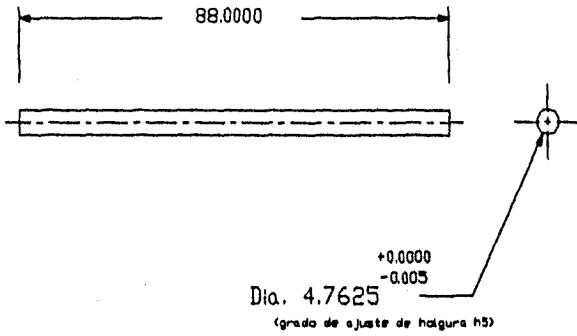


2 piezas de aluminio.

Pieza 3 : Bloque de desplazamiento		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

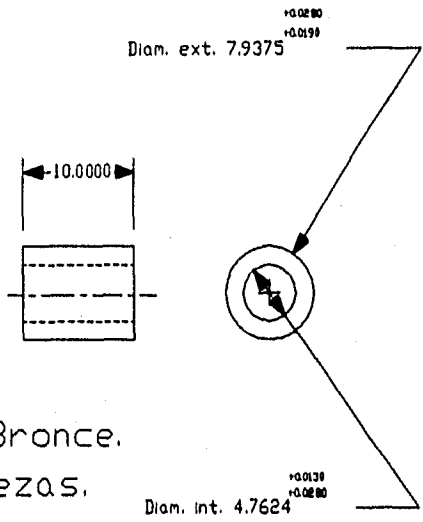


Pieza 4 : Acoplamiento		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1,5 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1



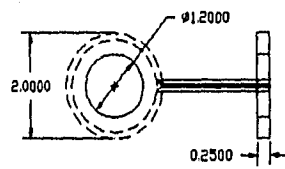
Barra redonda de acero inoxidable.
 2 piezas.

Pieza 5 : Guía lateral		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proj. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1



Mat: Bronce.
4 piezas.

Pieza 6 - Buje lateral		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 2.25 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1



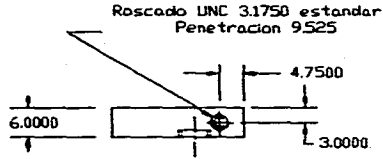
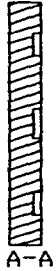
Engrane de 32 dientes

Mat: Acero

1 Pieza

Pieza 7 : Engrane		
FACULTAD DE INGENIERIA UNAM.	Escala 1 : 1	Unidades cm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

VII-20

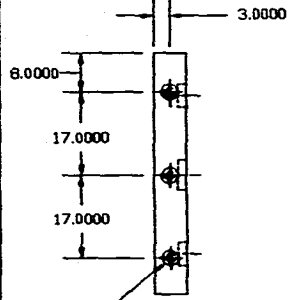


NOTA a.

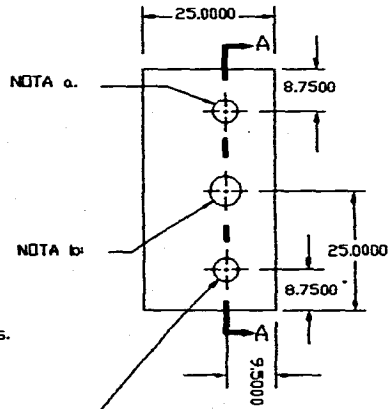
Dia. 4.7625. Penetracion 1.5

NOTA b.

Dia. 6.0000. Penetracion 1.5

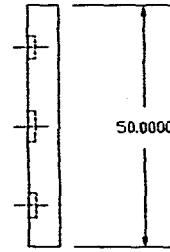


3 roscados iguales.
NOTA d.



NOTA a.

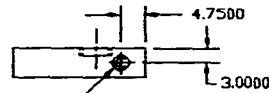
NOTA b.



NOTA d.

Roscado UNC 3.1750 estandar
Penetracion 9.525

NOTA a.

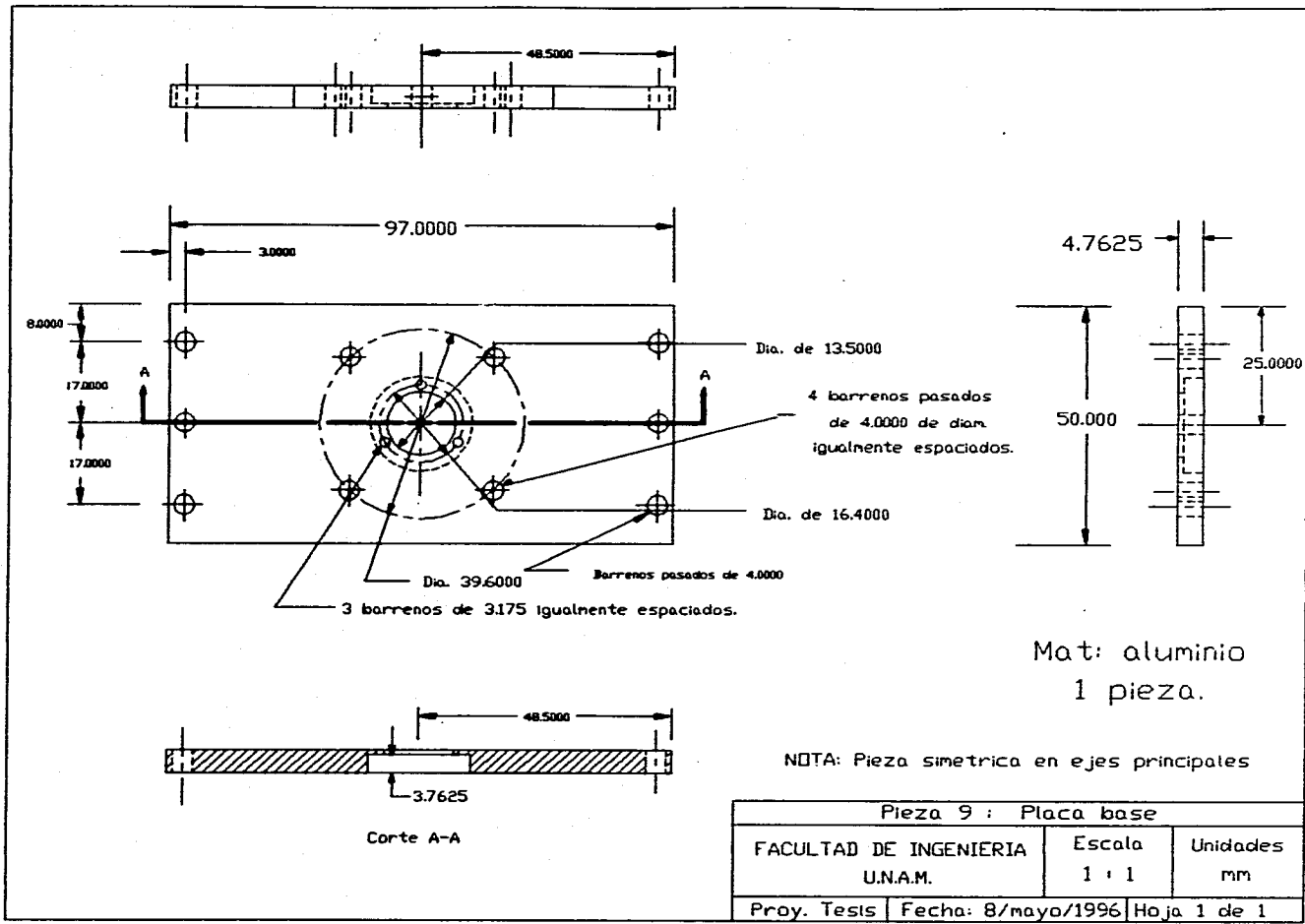


Roscado UNC 3.1750 estandar
Penetracion 9.525

Material: aluminio.
1 pieza.

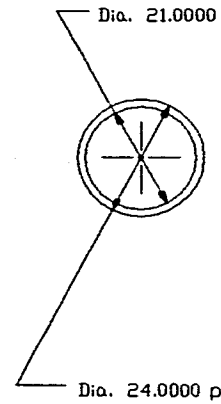
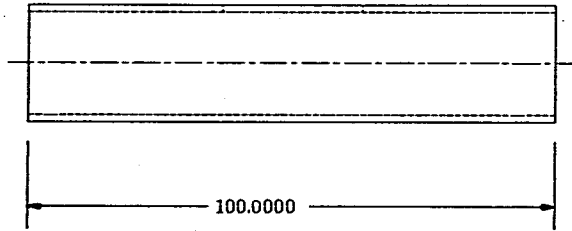
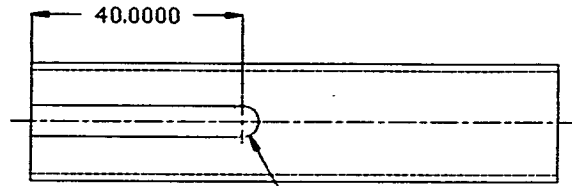
Pieza 8 : Placa de soporte		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

VII-21



Pieza 9 : Placa base		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

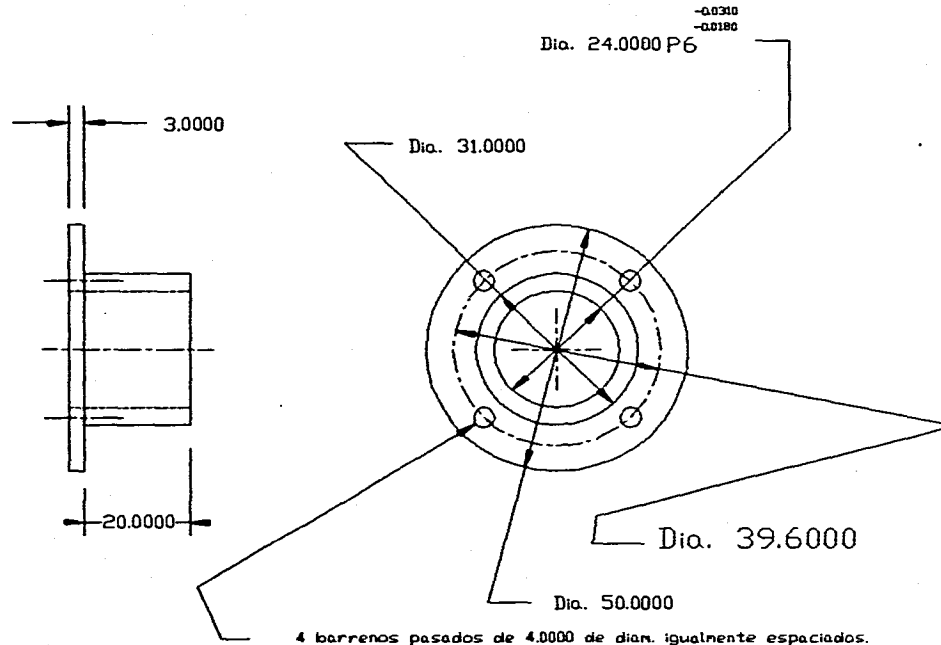
VII-22



Mat: aluminio.
1 pieza.

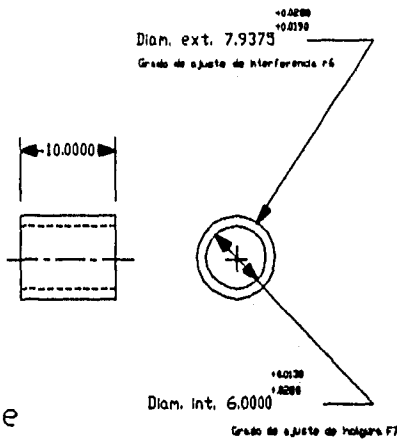
Pieza 10 : Cubierta de motor		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

VII-23



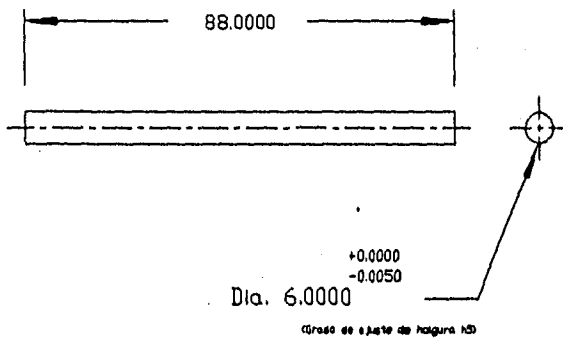
Mat: aluminio.
1 piezas

Pieza 11 : Cilindro base		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tests	Fecha: 8/mayo/1996	Hoja 1 de 1



Mat: Bronce
 2 piezas.

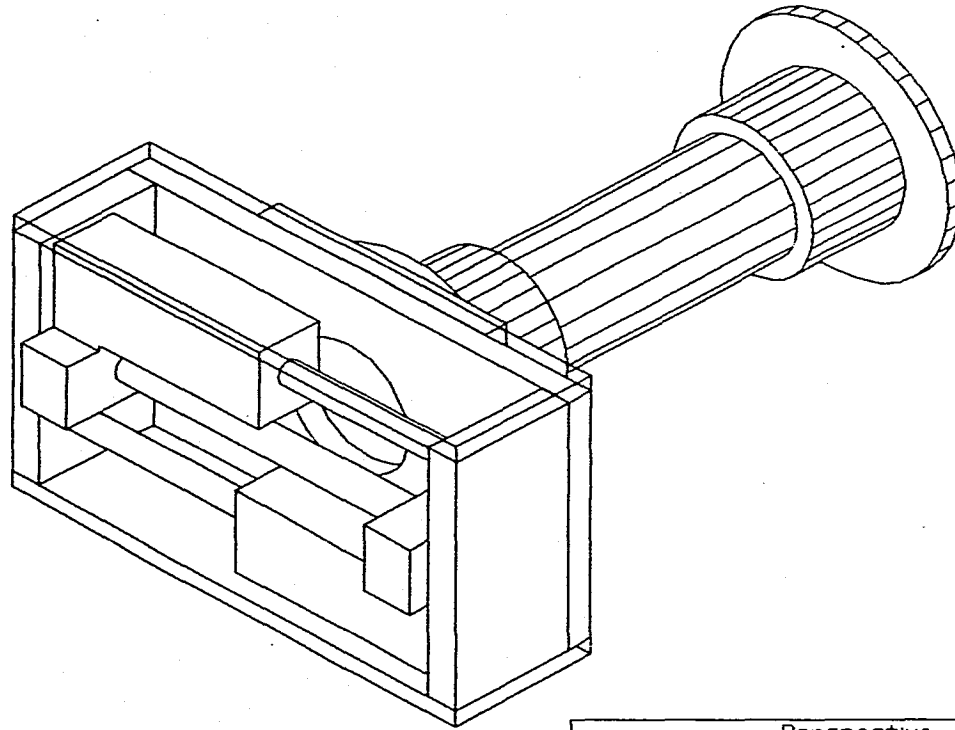
Pieza 12 : Buje central		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 2 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1



Barra redonda de acero inoxidable.
1 pieza.

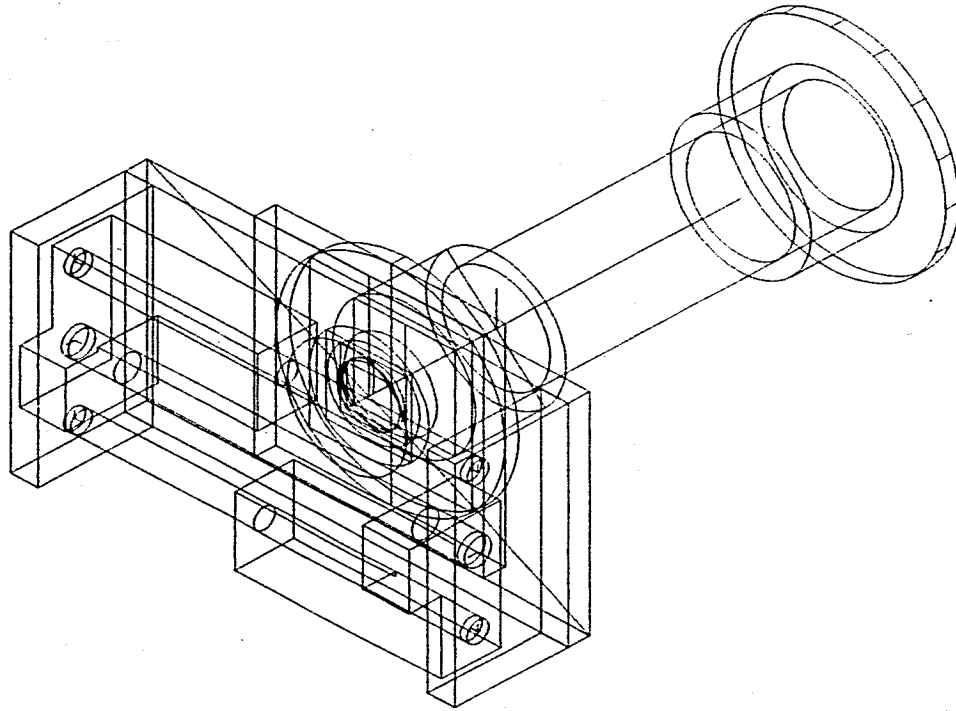
Pieza 13 : Guía central		
FACULTAD DE INGENIERIA U.N.A.M.	Escala 1 : 1	Unidades mm
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

VII-26



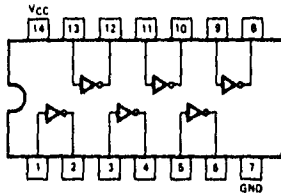
Perspectiva		
FACULTAD DE INGENIERIA U.N.A.M.		Isonétrico
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

VII-27



Dibujo de alambre		
FACULTAD DE INGENIERIA UNAM.		Isométrico
Proy. Tesis	Fecha: 8/mayo/1996	Hoja 1 de 1

D) HOJAS DE DATOS



J Suffix — Case 632-08 (Ceramic)
N Suffix — Case 646-08 (Plastic)

SN54/74LS04

HEX INVERTER
LOW POWER SCHOTTKY

GUARANTEED OPERATING RANGES

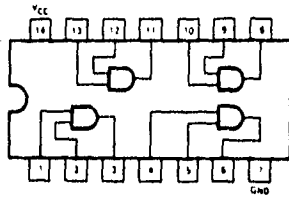
SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T _A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IK} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IH} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.6	V	
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	V _{CC} = V _{CC} MIN, V _{IH} = V _{IH} or V _{IH} per Truth Table
		74	0.35	0.5	V	
I _{IH}	Input HIGH Current			20	µA	V _{CC} = MAX, V _{IH} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IH} = 7.0 V
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IH} = 0.4 V
I _{OS}	Short Circuit Current	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total, Output HIGH Total, Output LOW			2.4	mA	V _{CC} = MAX
				8.8		

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t _{PLH}	Turn Off Delay, Input to Output		9.0	15	ns	V _{CC} = 5.0 V
t _{PHL}	Turn On Delay, Input to Output		10	15	ns	C _L = 15 pF



J Suffix — Case 632-08 (Ceramic)
N Suffix — Case 646-08 (Plastic)

SN54/74LS08

QUAD 2-INPUT AND GATE
LOW POWER SCHOTTKY

GUARANTEED OPERATING RANGES

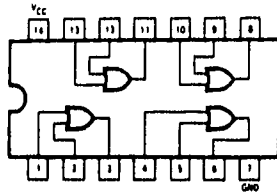
SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T _A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.85	-1.8	V	V _{CC} = MIN, I _{IH} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.5	V	
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	I _{OL} = 4.0 mA V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table
		74	0.35	0.5	V	
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total Output HIGH Total Output LOW			4.8	mA	V _{CC} = MAX
				8.8		

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t _{PLH}	Turn Off Delay, Input to Output		8.0	15	ns	V _{CC} = 5.0 V
t _{PHL}	Turn On Delay, Input to Output		1.0	2.0	ns	C _L = 15 pF



J Suffix -- Case 632-08 (Ceramic)
N Suffix -- Case 646-06 (Plastic)

SN54/74LS32

QUAD 2-INPUT OR GATE
LOW POWER SCHOTTKY

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current -- High	54, 74			-0.4	mA
I _{OL}	Output Current -- Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IK} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.8		
V _{OL}	Output LOW Voltage	54, 74	0.28	0.4	V	I _{OL} = 4.0 mA, V _{CC} = V _{CC} MIN, V _{IN} = V _{IH} or V _{IH} per Truth Table
		74	0.35	0.5		
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
I _{IL}	Input LOW Current			0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{OS}	Short Circuit Current	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total Output HIGH Total Output LOW			6.2	mA	V _{CC} = MAX
				9.8		

AC CHARACTERISTICS: T_A = 25 °C

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t _{PLH}	Turn Off Delay, Input to Output		14	22	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Turn On Delay, Input to Output		14	22		



SN54/74LS125A
SN54/74LS126A

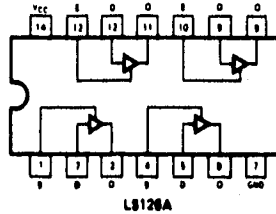
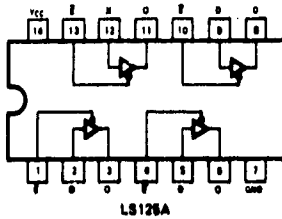
TRUTH TABLES

LS125A			LS126A		
INPUTS		OUTPUT	INPUTS		OUTPUT
E	D		E	D	
L	L	L	H	L	L
L	H	H	H	H	H
H	X	(Z)	L	X	(Z)

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
(Z) = High Impedance (off)

J Suffix — Case 632-08 (Ceramic)
N Suffix — Case 646-06 (Plastic)

QUAD 3-STATE BUFFERS
LOW POWER SCHOTTKY



DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	0.4		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IH} = -18 mA
V _{OH}	Output HIGH Voltage	0.4	2.4		V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.4		V	
V _{OL}	Output LOW Voltage	0.4, 74	0.25	0.4	V	I _{OL} = 12 mA I _{OL} = 24 mA V _{CC} = V _{CC} MIN, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	0.35	0.5	V	
I _{OZH}	Output OH Current HIGH			20	μA	V _{CC} = MAX, V _{OUT} = 2.4 V
I _{OZL}	Output OH Current LOW			-20	μA	V _{CC} = MAX, V _{OUT} = 0.4 V
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
I _{IL}	Input LOW Current			0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current			-0.6	mA	V _{CC} = MAX, V _{IN} = 0.6 V
I _{OS}	Short Circuit Current	-40		-225	mA	V _{CC} = MAX
I _{CC}	Power Supply Current	LS125A		20	mA	V _{CC} = MAX V _{IN} = 0 V, V _E = 4.5 V
		LS126A		22	mA	

5



SN54/74LS151

DESCRIPTION — The TTL/MSI SN54LS/74LS151 is a high speed 8-Input Digital Multiplexer. It provides, in one package, the ability to select one bit of data from up to eight sources. The LS151 can be used as a universal function generator to generate any logic function of four variables. Both assertion and negation outputs are provided.

8-INPUT MULTIPLEXER

LOW POWER SCHOTTKY

- SCHOTTKY PROCESS FOR HIGH SPEED
- MULTIFUNCTION CAPABILITY
- ON-CHIP SELECT LOGIC DECODING
- FULLY BUFFERED COMPLEMENTARY OUTPUTS
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

PIN NAMES

$S_0 - S_2$	Select Inputs
\bar{E}	Enable (Active LOW) Input
$I_0 - I_7$	Multiplexer Inputs
Z	Multiplexer Output (Note b)
\bar{Z}	Complementary Multiplexer Output (Note b)

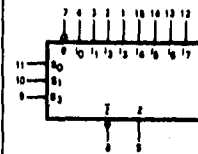
LOADING (Note a)

	HIGH	LOW
$S_0 - S_2$	0.5 U.L.	0.25 U.L.
\bar{E}	0.5 U.L.	0.25 U.L.
$I_0 - I_7$	0.5 U.L.	0.25 U.L.
Z	10 U.L.	5 (2.5) U.L.
\bar{Z}	10 U.L.	5 (2.5) U.L.

NOTES:

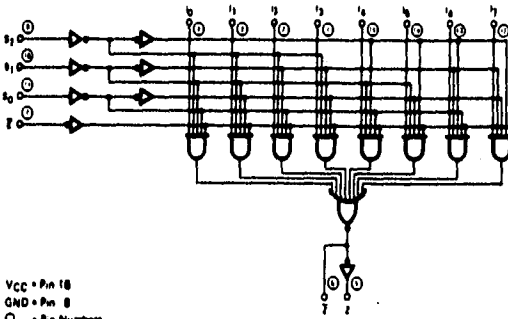
- a. 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.
- b. The Output LOW drive factor = 2.5 U.L. for Military (154) and 5 U.L. for Commercial (174) Temperature Ranges.

LOGIC SYMBOL



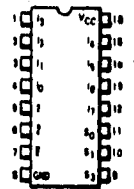
VCC = Pin 18
GND = Pin 8

LOGIC DIAGRAM



VCC = Pin 18
GND = Pin 8
○ = Pin Numbers

**CONNECTION DIAGRAM
DIP (TOP VIEW)**



J Suffix — Case 820-08
Ceramic
N Suffix — Case 840-08
(Plastic)

5

SN5474LS151

FUNCTIONAL DESCRIPTION - The LS151 is a logical implementation of a single pole, 8-position switch with the switch position controlled by the state of three Select inputs, S_0, S_1, S_2 . Both assertion and negation outputs are provided. The Enable input (E) is active LOW. When it is not activated, the negation output is HIGH and the assertion output is LOW regardless of all other inputs. The logic function provided at the output is:

$$Z = \bar{E} \cdot (I_0 \cdot \bar{S}_0 \cdot \bar{S}_1 \cdot \bar{S}_2 + I_1 \cdot \bar{S}_0 \cdot \bar{S}_1 \cdot S_2 + I_2 \cdot \bar{S}_0 \cdot S_1 \cdot \bar{S}_2 + I_3 \cdot \bar{S}_0 \cdot S_1 \cdot S_2 + I_4 \cdot S_0 \cdot \bar{S}_1 \cdot \bar{S}_2 + I_5 \cdot S_0 \cdot \bar{S}_1 \cdot S_2 + I_6 \cdot S_0 \cdot S_1 \cdot \bar{S}_2 + I_7 \cdot S_0 \cdot S_1 \cdot S_2)$$

The LS151 provides the ability, in one package, to select from eight sources of data or control information. By proper manipulation of the inputs, the LS151 can provide any logic function of four variables and its negation.

TRUTH TABLE

E	S ₂	S ₁	S ₀	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	Z	\bar{Z}
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	H	X	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	H	L
L	H	H	H	X	X	X	X	X	X	H	X	L	H

- H = HIGH Voltage Level
- L = LOW Voltage Level
- X = Don't Care



GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T _A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current - High	54, 74			-0.4	mA
I _{OL}	Output Current - Low	54			4.0	mA
		74			8.0	



SN54/74LS153

DUAL 4-INPUT MULTIPLEXER

LOW POWER SCHOTTKY

DESCRIPTION — The LSTTL/MSI SN54LS/74LS153 is a very high speed Dual 4-input Multiplexer with common select inputs and individual enable inputs for each section. It can select two bits of data from four sources. The two buffered outputs present data in the true (non-inverted) form. In addition to multiplexer operation, the LS153 can generate any two functions of three variables. The LS153 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- MULTIFUNCTION CAPABILITY
- NON-INVERTING OUTPUTS
- SEPARATE ENABLE FOR EACH MULTIPLEXER
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

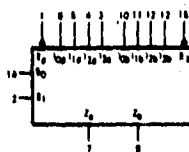
PIN NAMES

S ₀	Common Select Input
E	Enable (Active LOW) Input
I ₀ , I ₁	Multiplexer Inputs
Z	Multiplexer Output (Note b)

LOADING (Note a)	
HIGH	LOW
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	8 (2.8) U.L.

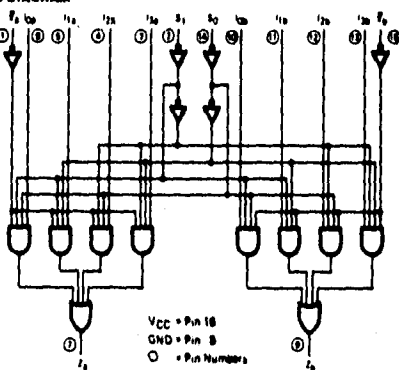
NOTES:
 a. 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.
 b. The Output LOW drive factor is 2.8 U.L. for Military (54) and 8 U.L. for Commercial (74) Temperature Range.

LOGIC SYMBOL

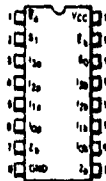


VCC = Pin 16
 GND = Pin 8

LOGIC DIAGRAM



CONNECTION DIAGRAM - DIP (TOP VIEW)



J Suffix — Case 620-08 (Ceramic)
 N Suffix — Case 646-08 (Plastic)

NOTE
 The Tripple version has the same pinout as the standard version.



SN5474LS153

FUNCTIONAL DESCRIPTION — The LS153 is a Dual 4-Input Multiplexer fabricated with Low Power, Schottky barrier diode process for high speed. It can select two bits of data from up to four sources under the control of the common Select Inputs (S_0, S_1). The two 4-input multiplexer circuits have individual active LOW Enables (\bar{E}_a, \bar{E}_b) which can be used to strobe the outputs independently. When the Enables (\bar{E}_a, \bar{E}_b) are HIGH, the corresponding outputs (Z_a, Z_b) are forced LOW.

The LS153 is the logic implementation of a 2-pole, 4-position switch, where the position of the switch is determined by the logic levels supplied to the two Select inputs. The logic equations for the outputs are shown below.

$$Z_a = \bar{E}_a \cdot (I_{0a} \cdot \bar{S}_1 \cdot \bar{S}_0 + I_{1a} \cdot \bar{S}_1 \cdot S_0 + I_{2a} \cdot S_1 \cdot \bar{S}_0 + I_{3a} \cdot S_1 \cdot S_0)$$

$$Z_b = \bar{E}_b \cdot (I_{0b} \cdot \bar{S}_1 \cdot \bar{S}_0 + I_{1b} \cdot \bar{S}_1 \cdot S_0 + I_{2b} \cdot S_1 \cdot \bar{S}_0 + I_{3b} \cdot S_1 \cdot S_0)$$

The LS153 can be used to move data from a group of registers to a common output bus. The particular register from which the data came would be determined by the state of the Select inputs. A less obvious application is a function generator. The LS153 can generate two functions of three variables. This is useful for implementing highly irregular random logic.

TRUTH TABLE

SELECT INPUTS		\bar{E}	INPUTS (a or b)				OUTPUT Z
S_0	S_1		I_0	I_1	I_2	I_3	
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

H = HIGH Voltage Level
 L = LOW Voltage Level
 X = Don't Care

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
VCC	Supply Voltage	54	4.5	5.0	V
		74	4.75	5.0	
TA	Operating Ambient Temperature Range	54	-55	25	°C
		74	0	25	
I _{OH}	Output Current — High	54, 74		-0.6	mA
I _{OL}	Output Current — Low	54		4.0	mA
		74		5.0	



**SN54/74LS155
SN54/74LS156**

**DUAL 1-OF-4 DECODER/
DEMULTIPLEXER
LS156-OPEN-COLLECTOR
LOW POWER SCHOTTKY**

DESCRIPTION — The SN54LS/74LS155 and SN54LS/74LS156 are high speed Dual 1-of-4 Decoder/Demultiplexers. These devices have two decoders with common 2-bit Address inputs and separate gated Enable inputs. Decoder "a" has an Enable gate with one active HIGH and one active LOW input. Decoder "b" has two active LOW Enable inputs. If the Enable functions are satisfied, one output of each decoder will be LOW as selected by the address inputs. The LS156 has open collector outputs for wired-OR (DOT-AND) decoding and function generator applications.

The LS155 and LS156 are fabricated with the Schottky barrier diode process for high speed and are completely compatible with all Motorola TTL families.

- SCHOTTKY PROCESS FOR HIGH SPEED
- MULTIFUNCTION CAPABILITY
- COMMON ADDRESS INPUTS
- TRUE OR COMPLEMENT DATA DEMULTIPLEXING
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

PIN NAMES

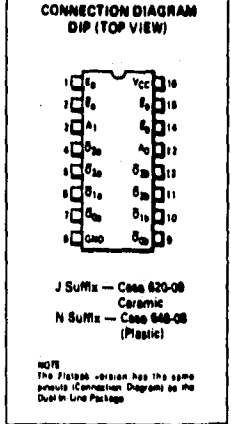
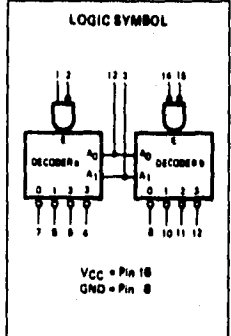
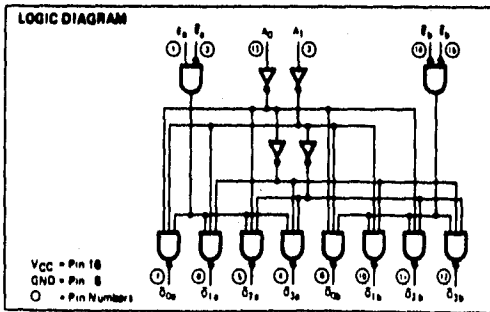
A_0, A_1	Address Inputs
E_a, E_b	Enable (Active LOW) Inputs
E_s	Enable (Active HIGH) Input
$O_0 - O_3$	Active LOW Outputs (Note b)

LOADING (Note a)

	HIGH	LOW
0.5 U.L.	0.25 U.L.	
0.8 U.L.	0.25 U.L.	
0.8 U.L.	0.25 U.L.	
10 U.L.	5 (2.8) U.L.	

- NOTES:**
- a. 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.8 mA LOW.
 - b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges. The HIGH level drive for the LS156 must be established by an external resistor.

5



SN54/74LS155 • SN54/74LS156

FUNCTIONAL DESCRIPTION — The LS155 and LS156 are Dual 1-of-4 Decoder/Demultiplexers with common Address inputs and separate gated Enable inputs. When enabled, each decoder section accepts the binary weighted Address inputs (A_0, A_1) and provides four mutually exclusive active LOW outputs (\bar{O}_0 – \bar{O}_3). If the Enable requirements of each decoder are not met, all outputs of that decoder are HIGH.

Each decoder section has a 2-input enable gate. The enable gate for Decoder "a" requires one active HIGH input and one active LOW input (E_a, \bar{E}_a). In demultiplexing applications, Decoder "a" can accept either true or complemented data by using the \bar{E}_a or E_a inputs respectively. The enable gate for Decoder "b" requires two active LOW inputs (\bar{E}_b, \bar{E}_b). The LS155 or LS156 can be used as a 1-of-8 Decoder/Demultiplexer by tying E_a to \bar{E}_b and relabeling the common connection as (A_2). The other \bar{E}_b and E_a are connected together to form the common enable.

The LS155 and LS156 can be used to generate all four minterms of two variables. These four minterms are useful in some applications replacing multiple gate functions as shown in Fig. 6. The LS156 has the further advantage of being able to AND the minterm functions by tying outputs together. Any number of terms can be wired-AND as shown below.

$$f = (E + A_0 + A_1) \cdot (E + \bar{A}_0 + A_1) \cdot (E + A_0 + \bar{A}_1) \cdot (E + \bar{A}_0 + \bar{A}_1)$$

where $E = E_a + \bar{E}_a$; $\bar{E} = E_b + \bar{E}_b$

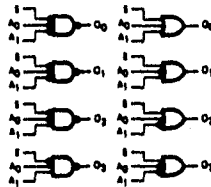


Fig. 6



TRUTH TABLE

ADDRESS		ENABLE "a"		OUTPUT "a"				ENABLE "b"		OUTPUT "b"			
A_0	A_1	E_a	\bar{E}_a	\bar{O}_0	\bar{O}_1	\bar{O}_2	\bar{O}_3	\bar{E}_b	\bar{E}_b	\bar{O}_0	\bar{O}_1	\bar{O}_2	\bar{O}_3
X	X	L	X	H	H	H	H	H	X	H	H	H	H
X	X	X	H	H	H	H	H	X	H	H	H	H	H
L	L	H	L	L	H	H	H	L	L	L	H	H	H
H	L	H	L	H	L	H	H	L	L	H	L	H	H
L	H	H	L	H	H	L	H	L	L	H	L	L	H
H	H	H	L	H	H	H	L	L	L	H	H	H	L

H • HIGH Voltage Level
 L • LOW Voltage Level
 X • Don't Care



SN54/74LS165

DESCRIPTION — The SN54LS/74LS165 is an 8-bit parallel load or serial-in register with complementary outputs available from the least stage. Parallel inputting occurs asynchronously when the Parallel Load (PL) input is LOW. With PL HIGH, serial shifting occurs on the rising edge of the clock; new data enters via the Serial Data (DS) input. The 2-input OR clock can be used to combine two independent clock sources, or one input can act as an active LOW clock enable.

8-BIT PARALLEL-TO-SERIAL CONVERTER LOW POWER SCHOTTKY

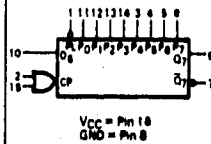
PIN NAMES

CP ₁ , CP ₂	Clock (LOW-to-HIGH Going Edge) Inputs
DS	Serial Data Input
PL	Asynchronous Parallel Load (Active LOW) Input
P ₀ -P ₇	Parallel Data Inputs
Q ₀	Serial Output from Last Stage (Note b)
\bar{Q}_0	Complementary Output (Note b)

LOADING (Note a)

	HIGH	LOW
CP ₁ , CP ₂	0.5 U.L.	0.25 U.L.
DS	0.5 U.L.	0.25 U.L.
PL	1.5 U.L.	0.75 U.L.
P ₀ -P ₇	0.5 U.L.	0.25 U.L.
Q ₀	10 U.L.	5 (2.5) U.L.
\bar{Q}_0	10 U.L.	5 (2.5) U.L.

LOGIC SYMBOL



NOTES

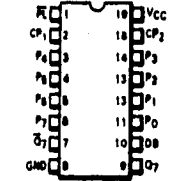
- a. 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW
- b. The Output LOW drive factor is 3.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Range.

TRUTH TABLE

PL	CP		CONTENTS								RESPONSE
	1	2	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	
L	X	X	P ₀	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	Parallel Entry
H	L	X	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	Right Shift
H	H	L	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	No Change
H	H	H	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	Right Shift
H	X	X	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	No Change

H = HIGH Voltage Level
L = LOW Voltage Level
X = Indifferent

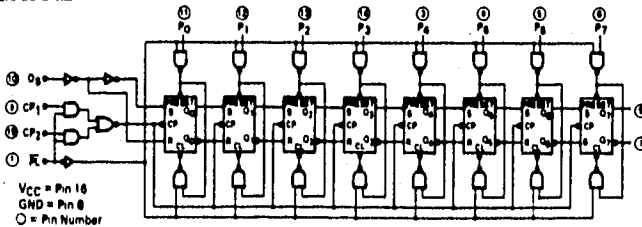
CONNECTION DIAGRAM (TOP VIEW)



J SuMx — Case 820-08 (Ceramic)
N SuMx — Case 848-08 (Plastic)

NOTE: The Plastic version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

LOGIC DIAGRAM





**SN54/74LS190
SN54/74LS191**

DESCRIPTION — The SN54LS/74LS190 is a synchronous UP/DOWN BCD Decade (8421) Counter and the SN54LS/74LS191 is a synchronous UP/DOWN Modulo-16 Binary Counter. State changes of the counters are synchronous with the LOW-to-HIGH transition of the Clock Pulse input.

An asynchronous Parallel Load (\overline{PL}) input overrides counting and loads the data present on the P_n inputs into the flip-flops, which makes it possible to use the circuits as programmable counters. A Count Enable (\overline{CE}) input serves as the carry/borrow input in multi-stage counters. An Up/Down Count Control ($\overline{U/D}$) input determines whether a circuit counts up or down. A Terminal Count (TC) output and a Ripple Clock (RC) output provide overflow/underflow indication and make possible a variety of methods for generating carry/borrow signals in multi-stage counter applications.

**PRESETTABLE BCD/DECADE
UP/DOWN COUNTERS
PRESETTABLE 4-BIT BINARY
UP/DOWN COUNTERS
LOW POWER SCHOTTKY**

- LOW POWER . . . 80 mW TYPICAL DISSIPATION
- HIGH SPEED . . . 28 MHz TYPICAL COUNT FREQUENCY
- SYNCHRONOUS COUNTING
- ASYNCHRONOUS PARALLEL LOAD
- INDIVIDUAL PRESET INPUTS
- COUNT ENABLE AND UP/DOWN CONTROL INPUTS
- CASCADABLE
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

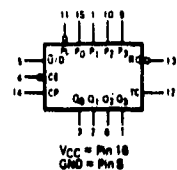
PIN NAMES

CE	Count Enable (Active LOW) Input		
CP	Clock Pulse (Active HIGH going edge) Input		
$\overline{U/D}$	Up/Down Count Control Input		
\overline{PL}	Parallel Load Control (Active LOW) Input		
P_n	Parallel Data Inputs		
Q_n	Flip-Flop Outputs (Note b)		
RC	Ripple Clock Output (Note b)	TC	Terminal Count Output (Note b)
TC	Terminal Count Output (Note b)		

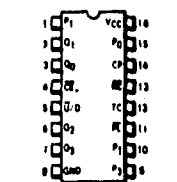
LOADING (Note a)	
HIGH	LOW
1.5 U.L.	0.7 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	5 (2.5) U.L.
10 U.L.	5 (2.5) U.L.
10 U.L.	5 (2.5) U.L.

NOTES:
a. 1 TTL Unit Load (U.L.) = 40 μ A HIGH/16 mA LOW.
b. The Output LOW drive factor is 2.5 U.L. for Military (84) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC SYMBOL



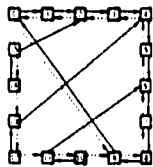
**CONNECTION DIAGRAM
DIP (TOP VIEW)**



J Suffix — Case 620-08 (Ceramic)
N Suffix — Case 648-08 (Plastic)

NOTE:
The Flipgate version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

STATE DIAGRAMS

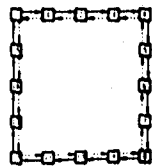


190

LS190
UP: $TC = Q_0 \cdot Q_3 \cdot (\overline{U/D})$
DOWN: $TC = \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{Q_3} \cdot (\overline{U/D})$

LS191
UP: $TC = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot (\overline{U/D})$
DOWN: $TC = \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{Q_3} \cdot (\overline{U/D})$

Count Up ———
Count Down ·····



191

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

6-Pin DIP Optoisolators
Transistor Output

These devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Convenient Plastic Dual-in-Line Package
- Most Economical Optoisolator
- High Input-Output Isolation Guaranteed — 7500 Volts Peak
- Meets or Exceeds All JEDEC Registered Specifications
- UL Recognized. File Number E54915
- VDE approved per standard 0883/8.80 (Certificate number 41953), with additional approval to DIN IEC380/VDE0808, IEC438/VDE0805, IEC85/VDE0860, VDE110b, covering all other standards with equal or less stringent requirements, including IEC204/VDE0113, VDE0180, VDE0832, VDE0833, etc.
- Special lead form available (add suffix "T" to part number) which satisfies VDE0883/8.80 requirement for 8 mm minimum creepage distance between input and output solder pads.
- Various lead form options available. Consult "Optoisolator Lead Form Options" data sheet for details.

4N25
4N25A
4N26
4N27
4N28

6-PIN DIP
OPTOISOLATORS
TRANSISTOR OUTPUT



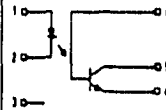
CASE 720A-08
PLASTIC

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Rating	Symbol	Value	Units
INPUT LED			
Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	80	mA
LED Power Dissipation — $T_A = 25^\circ\text{C}$ with Negligible Power in Output Detector Derate above 25°C	P_D	120	mW
		1.61	mW/°C
OUTPUT TRANSISTOR			
Collector-Emitter Voltage	V_{CE0}	30	Volts
Emitter-Collector Voltage	V_{EC0}	7	Volts
Collector-Base Voltage	V_{CB0}	70	Volts
Collector Current — Continuous	I_C	160	mA
Detector Power Dissipation in $T_A = 25^\circ\text{C}$ with Negligible Power in Input LED Derate above 25°C	P_D	160	mW
		1.70	mW/°C
TOTAL DEVICE			
Isolation Surge Voltage (1) (Peak at Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac
Total Device Power Dissipation in $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	260	mW
		2.84	mW/°C
Ambient Operating Temperature Range	T_A	-55 to +100	°C
Storage Temperature Range	T_{STG}	-55 to +150	°C
Soldering Temperature (10 sec, 1/16" from case)	T_{SOL}	260	°C

(1) Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

SCHEMATIC



- 1 LED ANODE
- 2 LED CATHODE
- 3 N.C.
- 4 EMITTER
- 5 COLLECTOR
- 6 BASE

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristics	Symbol	Min	Typ	Max	Unit
INPUT LED					
Forward Voltage (I _f = 10 mA)	T _A = 25°C T _A = -55°C T _A = 100°C	V _f	— 1.15 —	1.5 1.3 —	Volts
Reverse Leakage Current (V _R = 3 V)		I _R	—	100	μA
Capacitance (V = 0 V, f = 1 MHz)		C _J	—	18	pF
OUTPUT TRANSISTOR					
Collector-Emitter Dark Current (V _{CE} = 10 V, T _A = 25°C)	4N25,25A,25,27 4N28	I _{CEO}	—	1 100	nA
(V _{CE} = 10 V, T _A = 100°C)	All Devices	I _{CEO}	—	1	μA
Collector-Base Dark Current (V _{CB} = 10 V)		I _{CBO}	—	0.2	nA
Collector-Emitter Breakdown Voltage (I _C = 1 mA)		V _{(BR)CEO}	30	45	Volts
Collector-Base Breakdown Voltage (I _C = 100 μA)		V _{(BR)CBO}	70	100	Volts
Emitter-Collector Breakdown Voltage (I _E = 100 μA)		V _{(BR)ECO}	7	7.0	Volts
DC Current Gain (I _C = 2 mA, V _{CE} = 5 V)		h _{FE}	—	500	—
Collector-Emitter Capacitance (f = 1 MHz, V _{CE} = 0)		C _{CE}	—	7	pF
Collector-Base Capacitance (f = 1 MHz, V _{CB} = 0)		C _{CB}	—	18	pF
Emitter-Base Capacitance (f = 1 MHz, V _{EB} = 0)		C _{EB}	—	9	pF
COUPLED					
Output Collector Current (I _f = 10 mA, V _{CE} = 10 V)	4N25,25A,25,27 4N27,28	I _C	2 1	7 —	mA
Collector-Emitter Saturation Voltage (I _C = 2 mA, I _f = 50 mA)		V _{CE(sat)}	—	0.15 0.5	Volts
Turn-On Time (I _f = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)		t _{on}	—	2.8	μs
Turn-Off Time (I _f = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)		t _{off}	—	4.5	μs
Rise Time (I _f = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)		t _r	—	1.2	μs
Fall Time (I _f = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)		t _f	—	1.3	μs
Isolation Voltage (f = 60 Hz, t = 1 sec)		V _{ISO}	7500	—	Vac(p-p)
Isolation Resistance (V = 500 V)		R _{ISO}	10 ¹¹	—	Ω
Isolation Capacitance (V = 0 V, f = 1 MHz)		C _{ISO}	—	0.2	pF

6

TYPICAL CHARACTERISTICS

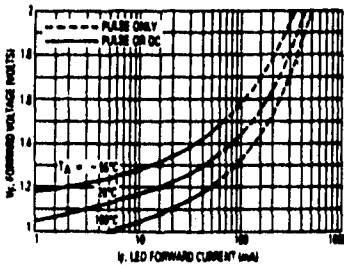


Figure 1. LED Forward Voltage versus Forward Current

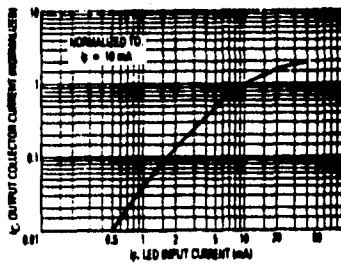


Figure 2. Output Current versus Input Current

EXAR

XR-2001/2/3/4

High-Voltage, High-Current Darlington Transistor Arrays

GENERAL DESCRIPTION

The XR-2001/2002/2003/2004 are high-voltage, high-current Darlington transistor arrays consisting of seven silicon NPN Darlington pairs on a common monolithic substrate. All units feature open collector outputs and integral protection diodes for driving inductive loads. Peak inrush currents of up to 600 mA are allowed, which makes the arrays ideal for driving tungsten filament lamps. The outputs may be paralleled to achieve high load current capability although each driver has a maximum continuous collector current rating of 500 mA. The arrays are directly price competitive with discrete transistor alternatives.

FEATURES

- Peak Inrush Current Capability of 600 mA
- Internal Protection Diodes for Driving Inductive Loads
- Excellent Noise Immunity
- Direct Compatibility with Most Logic Families
- Opposing Pin Configuration Eases Circuit Board Layout

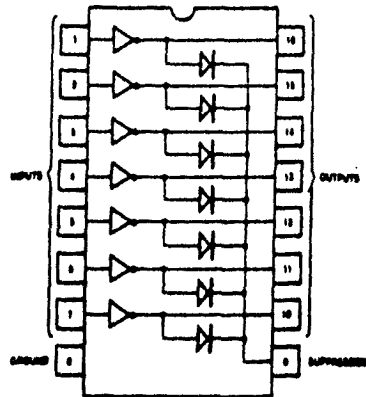
APPLICATIONS

- Relay Drivers
- High Current Logic Drivers
- Solenoid Driver

ABSOLUTE MAXIMUM RATINGS $T_A = 25^\circ\text{C}$

Output Voltage, V_{CE}	50V
Input Voltage, V_{IH}	30V
Continuous Collector Current, I_C (Each Driver)	500mA
Continuous Base Current, I_B (Each Driver)	25mA
Power Dissipation, P_D (Each Driver)	1.0W
(Total Package)	See graph
Derate Above 25°C	18 mW/ $^\circ\text{C}$
Storage Temperature Range	-55 $^\circ\text{C}$ to +150 $^\circ\text{C}$

FUNCTIONAL BLOCK DIAGRAM



ORDERING INFORMATION

Part Number	Package	Operating Temperature
XR-2001CN	Ceramic	0 $^\circ\text{C}$ to +70 $^\circ\text{C}$
XR-2002CN	Ceramic	0 $^\circ\text{C}$ to +70 $^\circ\text{C}$
XR-2003CN	Ceramic	0 $^\circ\text{C}$ to +73 $^\circ\text{C}$
XR-2004CN	Ceramic	0 $^\circ\text{C}$ to +70 $^\circ\text{C}$

SYSTEM DESCRIPTION

The XR-2001 interfaces with bipolar digital logic (with external current limiting), or with CMOS or PMOS directly.

The XR-2002 was specifically designed to interface with 14V to 25V PMOS devices.

The XR-2003 permits operation directly with CMOS or TTL operating at a supply voltage of 5 volts. Interface requirements beyond the scope of standard logic buffers are easily handled by the XR-2003.

The XR-2004 requires less input current than the XR-2003 and the input voltage is less than that required by the XR-2002. The XR-2004 operates directly from PMOS or CMOS outputs using supply voltages of 6 to 15 volts.

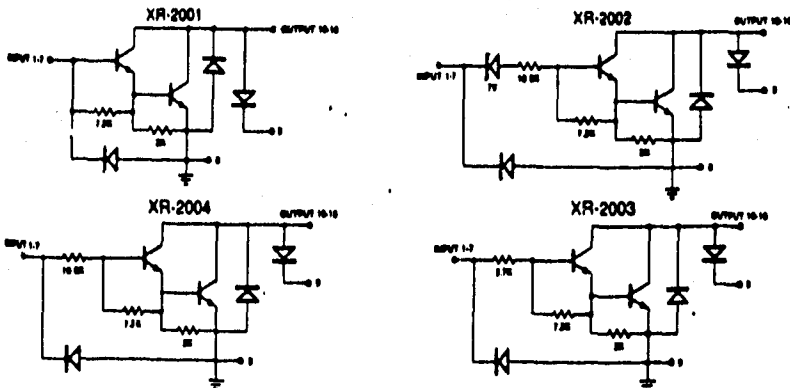
XR-2001 XR-2002 XR-2003 XR-2004

ELECTRICAL CHARACTERISTICS

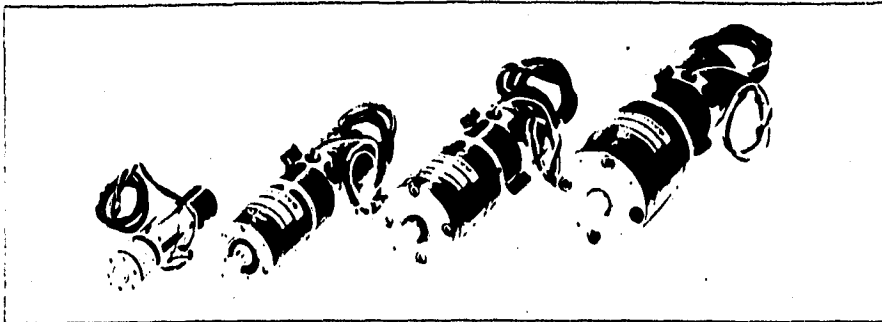
Test Conditions: T_A 25°C, unless otherwise specified

SYMBOL	PARAMETERS	LIMITS			UNITS	CONDITIONS
		MIN	TYP	MAX		
I_{CEX}	Output Leakage Current XR-2002 XR-2004			100	μA	$V_{CE} = 50 V, T_A = 70^\circ C$ $V_{CE} = 50 V, T_A = 70^\circ C, V_{IN} = 6V$ $V_{CE} = 50 V, T_A = 70^\circ C, V_{IN} = 1V$
				500	μA	
				500	μA	
V_{CE}	Collector-Emitter Saturation Voltage		1.25	1.6	V	$I_C = 350mA, I_B = 500\mu A$ $I_C = 200mA, I_B = 350\mu A$ $I_C = 100mA, I_B = 250\mu A$
			1.1	1.3	V	
			0.9	1.1	V	
I_{IN}	Input Current (on) XR-2002 XR-2003 XR-2004		0.85	1.25	mA	$V_{IN} = 17V$ $V_{IN} = 3.85V$ $V_{IN} = 5V$ $V_{IN} = 12V$
			0.93	1.35	mA	
			0.35	0.5	mA	
			1.0	1.45	mA	
I_{IN}	Input Current (off)	50	65		μA	$I_C = 500\mu A, T_A = 70^\circ C$
V_{IN}	Input Voltage XR-2002 XR-2003 XR-2004			13	V	$V_{CE} = 2 V, I_C = 300mA$ $V_{CE} = 2 V, I_C = 200mA$ $V_{CE} = 2 V, I_C = 250mA$ $V_{CE} = 2 V, I_C = 300mA$ $V_{CE} = 2 V, I_C = 125mA$ $V_{CE} = 2 V, I_C = 200mA$ $V_{CE} = 2 V, I_C = 275mA$ $V_{CE} = 2 V, I_C = 350mA$
				2.4	V	
				2.7	V	
				3.0	V	
				5.0	V	
				6.0	V	
				7.0	V	
				8.0	V	
h_{FE}	D-C Forward Current Transfer Ratio XR-2001	1000				$V_{CE} = 2 V, I_C = 350mA$
C_{IN}	Input Capacitance		15	30	pF	
I_R	Clamp Diode Leakage Current			50	μA	$V_R = 50V$
V_F	Clamp Diode Forward Voltage		1.7	2.0	V	$I_F = 350mA$
t_{PLH}	Turn-On Delay		0.25	1.0	μS	$0.5 E_{IN}$ to $0.5 E_{OUT}$
t_{PHL}	Turn-Off Delay		0.25	1.0	μS	$0.5 E_{IN}$ to $0.5 E_{OUT}$

SCHEMATIC DIAGRAMS (One of 7 Identical Drivers is shown for each device)



RH MINI SERIES



The RH mini series actuators, combining precision harmonic drive gearing with DC servo motors utilizing rare earth magnets, offer unique features unsurpassed by conventionally geared drives. Used in highly demanding industrial servo systems they can provide precision motion control and high torque capacity in very compact packages. Encoders and tachometers are built integral with the motor to make the axial length minimum.

Harmonic drive gearing is lubricated with a grease specially conceived and made for it for low maintenance and prolonged service life. Motor brushes are O-ring sealed to contain dust.

APPLICATIONS

Industrial Robots Positioning and Indexing Tables Welding and Plasma Equipments
 Medical Equipments Valve Drives Testing, Analyzing, Measuring Instruments

TYPE AND DESIGNATION

ACTUATOR

RH-8-30 06
A B C D

WITH TACHOMETER

RH-8-3006-T
E

(optional)

WITH ENCODER

RH-8-3006-E 050 A
F G H

(optional)

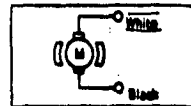
A : RH Series
 B : Unit Size
 C : Rated Speed, rpm
 D : Rated Output
 E : With Tachometer. (Not available for RH5)
 F : With Encoder
 G : Encoder Resolution, 050=500P/rev.
 H : Encoder Voltage, A=5V

SPECIFICATION

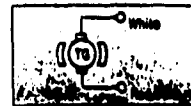
RATING : Continuous
 EXCITING METHOD : Permanent Magnet
 MOTOR INSULATION : Class B
 DIELECTRIC STRENGTH : AC500V 1min.
 INSULATION RESISTANCE : 100M Ω (500VDC Megger)
 ROTATION : Bidirectional
 CONSTRUCTION : Totally Closed
 LUBRICATION : Grease (Harmonic Grease SK-2)
 COLOR : Black
 AMBIENT TEMPERATURE : 0~40°C

WIRING

Rotation CW facing mounting end, with a positive current applied to white leads of motor with respect to black leads.



Tacho polarity positive white leads and negative red leads, when rotation CW facing mounting end.



RH MINI SERIES

SERVO ACTUATOR MODEL	RH-6				RH-8		RH-11			RH-14			
	5502	3002	1502	6906	3006	6001	3001	4002	3402	3002	2702	1901	
ACTUATOR													
Rated Output Power ¹⁾	W	1.7	2.5	2.5	8.6	6.2	13.6	12.3	20.3	18.5	18.5	18	12.7
Rated Output Torque ¹⁾	Ncm lb-in	30 2.6	80 7.1	160 14	140 12	200 17	220 19	380 34	320 28	520 46	590 52	640 56	640 56
Rated Voltage ¹⁾	V	24				24			24				
Rated Current ¹⁾	A	0.5	0.4	1.0	0.8	1.3			1.8				
Rated Output Speed ¹⁾	rpm	55	30	15	60	30	60	30	60	34	30	27	19
Max. Continuous Stall Trq ¹⁾	Ncm lb-in	40 3.5	150 13	230 20	150 13	230 20	250 22	440 39	540 48	780 69	780 69	780 69	780 69
Max. Output Torque ¹⁾	Ncm lb-in	60 5.2	260 23	350 31	260 23	350 31	490 43	780 69	1,370 122	1,960 174	1,960 174	1,960 174	1,960 174
Max. Current ¹⁾	A	0.78	1.0	0.7	1.6	1.1	2.4	2.1	5.4	4.6	4.1	3.9	3.0
Max. Output Speed ¹⁾	rpm	110	60	25	100	50	100	50	100	57	50	45	34
Torque Constant	Ncm/A lb-in/A	111 10	333 30	664 59	210 19	420 37	248 22	491 43	292 26	508 45	976 86	634 56	763 68
Voltage Constant (B.E.M.F.)	V/rpm	0.12	0.35	0.70	0.22	0.44	0.26	0.50	0.30	0.53	0.60	0.66	0.80
Moment of Inertia ¹⁾	kg-cm sec ² lb-in sec ²	0.0159 0.0138	0.03 0.026	0.12 0.104	0.038 0.033	0.151 0.131	0.110 0.095	0.440 0.382	0.208 0.181	0.644 0.559	0.832 0.722	1.010 0.877	0.644 0.559
Starting Current	A	0.13	0.07	0.07	0.11	0.14	0.17	0.12	0.18	0.17	0.16	0.15	0.12
No-load Running Current ¹⁾	A	0.24	0.19	0.21	0.38	0.38	0.61	0.55	0.89	0.90	0.91	0.70	0.55
Mechanical Time Constant	m-sec	13.3	9.5		8.5		8.5			7.0			
Rated Power Rate ¹⁾	W/sec	0.055	0.21	0.51	0.26	0.43	0.36	0.51	0.43	0.42	0.41	0.84	
Thermal Time Constant ¹⁾	min	5.2	6		9		10			11			
Thermal Resistance ¹⁾	°C/W	11.4	7.2		4.2		3.3			2.8		4.1	
Reduction Ratio	1:R	1:80	1:50	1:100	1:50	1:100	1:50	1:100	1:50	1:80	1:100	1:110	1:80
Max. Radial Load	kg(lb)	6.0(13)	20(44)				25(55)			40(88)			
Max. Thrust Load	kg(lb)	3.0(7)	10(22)				20(44)			40(88)			
Weight: Actuator Only	kg(lb)	0.070(15)	0.22(0.5)	0.27(0.6)		0.47(1.0)			0.74(1.6)				
: With Tach	N.A.	0.30(0.7)	0.36(0.8)		0.55(1.2)			0.82(1.8)					
: With Encoder	kg(lb)	0.08(0.2)	0.26(0.6)	0.31(0.7)		0.51(1.1)			0.78(1.7)				
MOTOR													
Rated Power ¹⁾	W	2.6	3.4	10		20			30			17	
Rated Speed ¹⁾	rpm	4,500	1,500	3,000		3,000			3,000			1,650	
Armature Resistance	Ω	8.6	36	10		4.7			2.7			6.5	
Armature Inductance	mH	2.7	6.9	2.2		1.6			1.1			2.4	
Electrical Time Constant	m-sec	0.31	0.19	0.22		0.34			0.41			0.37	

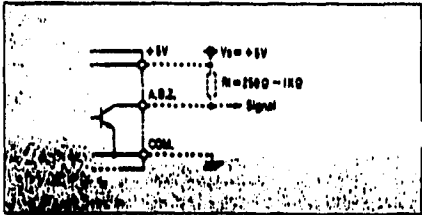
RH6, RH8, RH11, RH14 Performance Data Notes

- All specifications are applicable for actuators mounted on aluminum heat sink of the following sizes:
 RH-6 100x100x3(mm)
 RH-8 150x150x6(mm)
 RH-11 150x150x6(mm)
 RH-14 150x150x6(mm)
- Actuator specifications include the efficiency of harmonic drive gearing.
- Values for saturated actuator temperature. Other values for actuator temperature of 20°C.
- Maximum allowable values. Under no circumstances the limits may be exceeded.
- Moment of inertia is the sum of motor and harmonic drive gearing converted to the output side of actuator.
- Values at rated speed.

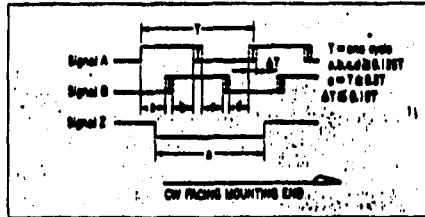
ENCODER			
SERVO ACTUATOR MODEL		RH-5	RH-8, RH-11, RH-14
Output Circuit			Open Collector
Resolution ⁷⁾	P/rev.	100, 200 300, 360	200, 360, 600, 1,000
Output Signal			3-Channel (A, B, Z)
Power Supply	V, mA	+5VDC, ±5% 60mA Max.	+5VDC, ±5%, 80mA Max.
Output Voltage	V		V _{OL} = 0.5V Max.
Max. Permissible Voltage	V DC	+38	+30
Max. Permissible Current	mA		20
Max. Response Frequency	kHz	80	88
Moment of Inertia ⁸⁾	kg ² -cm sec ² lb-in sec ²	1 × 10 ⁻⁴ 8.8 × 10 ⁻⁵	5 × 10 ⁻⁴ 4.3 × 10 ⁻⁴
Vibration	G		2.5 Max.
Shock	G		30 Max.
Lead Wire	mm	φ3 × 300l φ0.08/17ph	φ4 × 300l, φ0.12/7ph
TACHOMETER GENERATOR			
Output Voltage ⁹⁾	V/T, 000rpm	N.A.	3 ± 10%
Linearity ⁹⁾	% Max.	N.A.	1/200rpm Min. 1
Ripple ⁹⁾	% Max.	N.A.	1RMS/3(r-p) (200rpm Min. 1)
Armature Resistance	Ω	N.A.	65
Armature Inductance	mH	N.A.	7
Moment of Inertia ¹⁰⁾	kg ² -cm sec ² lb-in sec ²	N.A.	1.2 × 10 ⁻³ 1.0 × 10 ⁻³

- 7) Resolution of encoder only. Resolution at the output of actuator is equal to the encoder resolution multiplied by R (reduction ratio).
- 8) Value at motor shaft. To convert it to the value at the output of actuator, multiply by R² (reduction ratio squared).
- 9) Values for tach only.
- 10) To convert it to the value at the output of actuator multiply by R² (reduction ratio squared).

ENCODER OUTPUT CHARACTERISTICS
OUTPUT CIRCUIT



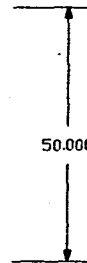
OUTPUT WAVEFORMS



LEAD WIRES

BROWN	A SIGNAL
RED	B SIGNAL
YELLOW	Z SIGNAL
WHITE	+5VDC
BLACK	GROUND (COMMON)
SHIELD	FLOATING

6.0000



50.0000

0

a Dedo

Escala	Unidades
1 : 1	mm

yo/1996 Hoja 1 de 1