

UNIVERSIDAD AUTONOMA DE GUADALAJARA

INCORPORADA A LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA MECANICA ELECTRICA

27² Gen.



TESIS CON
FALLA DE ORIGEN

"CONTROL POR COMPUTADORA DE UNA PLATAFORMA PARA LA
AYUDA EN LA REHABILITACION DE MINUSVALIDOS".

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A:

SR. ENRIQUE ZEPEDA SANCHEZ

GUADALAJARA, JALISCO. DICIEMBRE 1986



Universidad Nacional
Autónoma de México



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 está 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.

INDICE

I) INTRODUCCION -----	1
II) LA PLATAFORMA -----	5
III) INTERFASE COMPUTADORA-PLATAFORMA -----	10
a) Fase de control de velocidad -----	10
1) Oscilador -----	12
2) Comparador -----	13
3) Contador -----	15
b) Fase de control de giro -----	19
c) Fase de selección de motor -----	21
d) La interfase completa -----	22
IV) PROGRAMACION DE LA COMPUTADORA -----	28
a) Desplegados -----	33
b) Diagrama de flujo -----	40
c) Programa -----	43
V) EL SISTEMA COMPLETO -----	53
VI) CONCLUSIONES -----	59
BIBLICGRAFIA -----	61
Anexo 1 -----	63
Anexo 2 -----	66
Anexo 3 -----	67
Anexo 4 -----	68
Anexo 5 -----	70
Anexo 6 -----	71

C A P I T U L O I

I N T R O D U C C I O N .

INTRODUCCION

Desde que ha existido el ser humano, han vivido personas que por causas hereditarias, enfermedades, accidentes, etc., se han visto impedidas a continuar una vida de movimiento normal por causa de haber quedado imposibilitadas a mover sus extremidades o quizá todo su cuerpo.

Muchas cosas se han hecho para ayudarles a tener un proceso de rehabilitación a aquellas personas que todavía lo pueden tener, y a las que no, ayudarles a no tener defor~~maciones~~ maciones en sus miembros por causa de la inmovilidad.

Entre estas cosas se han ideado ejercicios que ayudan en este proceso. En la mayoría de los casos de personas minusválidas, una enfermera o un terapeuta debe de realizar junto con el paciente estos ejercicios, es decir, mover con sus propias manos las partes inmóviles del paciente, porque este último ha perdido la capacidad de manipularse a sí mismo. El terapeuta o la persona que ayuda al paciente a realizar este trabajo, tiene que realizarlo repetidas veces en un mismo día, con un gran número de personas y su consecuente fatiga. El terapeuta, al estar realizando los ejercicios con el paciente, no tiene la oportunidad de controlar, alcanzar, mover, quitar, sostener cosas o aparatos, por estar ocupado a toda su capacidad en la realización de los ejercicios que no pueda suspender bruscamente.

Conjuntamente con los ejercicios y con las ciencias médicas se han creado algunos aditamentos o aparatos que ayudan a realizar los ejercicios que requiere cada una de las personas imposibilitadas, en su proceso de rehabilitación.

En esta tesis yo propongo la automatización y control por medio de una microcomputadora de la plataforma, algo parecida a un brazo mecánico móvil, sobre ella se sujetaría al paciente con problemas de rehabilitación de sus extremidades inferiores, y esta plataforma, al tomar diferentes posiciones de sus partes móviles, haría que el cuerpo del paciente adoptara las posiciones que los ejercicios exigen de sus extremidades inferiores.

Así, con esta plataforma el terapeuta podría programar los movimientos que deberá realizar la plataforma y estos movimientos los ejecutaría posteriormente la plataforma ya con el paciente sujeto a ella. Esta programación se efectuaría con algunas teclas del teclado o por medio de una palanca de control conectada directamente a la computadora. Con esta última opción, el paciente que ya conoce sus ejercicios podría también programarlos.

-La microcomputadora tendría la opción de almacenar en memoria secundaria (disco flexible, cassette, disco duro) todos los datos de los movimientos diferentes de un gran número de personas, y así no tener la necesidad de repetir la programación día con día o paciente a paciente.

Este proyecto lo realizaré en cuatro etapas principales :

- A) Primero haré un esbozo general de la plataforma en sus dimensiones, grados de libertad, posición de motores, etc.
- B) Enseguida procederé al diseño de la interfase, que -- constará de elementos analógicos y digitales. Esta in-

terfase constará de tres elementos principales :

- a) Control de velocidad
 - b) Control de giro
 - c) Selección de motor
- C) Proseguiré a la programación del microcomputador codificada en BASIC, que dará los datos de control a la interfase, constará de algunas partes principales :
- a) Entrada de datos (teclas o palanca de control)
 - b) Grabación de datos en memoria secundaria
 - c) Desplegados en pantalla
 - d) Movimientos de prueba
- D) Por último haré la unión de las partes A,B,C, que formarán el sistema completo.

Mis expectativas para este proyecto son muchas y vislumbro ya algunas ventajas que aportará este sistema :

- i) El proceso de ejercicios pasivos estaría en gran parte completamente automatizado.
- ii) Tener una gran variedad de movimientos y posiciones, - como también la variación controlada de la velocidad - de cada una de las articulaciones y tener así una gran versatilidad en cuanto a ejercicios se refiere.
- iii) Poder tener un movimiento uniforme y constante para poder vigilar la consistencia de los ejercicios y que estos se mantengan dentro de los límites que no causen dolor.
- iv) Se mantendría además la longitud en reposo de los músculos (que es lo que se pretende con los ejercicios pasivos) y además prepararía para el ejercicio activo.

- v) El terapeuta puede estar vigilando todo el cuerpo del paciente y poder estar al tanto de que no existan contracturas (contracción involuntaria, duradera o permanente de uno o más músculos), en los músculos del paciente.
- vi) Para el paciente que pueda programar la plataforma, le ayudaría a desarrollar un sentido de independencia, - que creo que es muy importante desarrollarlo en estas personas.
-

C A P I T U L O II

LA PLATAFORMA.

LA PLATAFORMA

Esta plataforma tiene la función de ayudar al paciente minusválido a realizar sus ejercicios de rehabilitación.

Esta plataforma será como lo que hoy es una mesa balanceada (camilla rígida en la que se coloca al paciente para ayudarlo a comenzar a tomar una posición vertical), junto con un conjunto articulado, compuesto por 3 articulaciones para ayudar en los ejercicios de rehabilitación de las piernas.

Esta plataforma sería única y exclusivamente empleada en hospitales, con un terapeuta vigilando los ejercicios y las posiciones adoptadas por la plataforma.

El esbozo que se dará en este capítulo de la plataforma, no quieren llegar a ser normas de diseño, sino mostrar única y exclusivamente las características cualitativas que tendría la plataforma.

La plataforma tendrá partes móviles y partes fijas, a las primeras les llamaremos articulaciones y a las segundas partes estáticas.

La plataforma estará constituida casi en su totalidad de algún metal liviano como aluminio o algún tipo de acero o resina rígida. La Fig. 2.0 muestra en esbozo cómo estaría construida la plataforma.

En ella podemos distinguir 3 partes principales:

- a) BASE

- b) MESA BALANCIN
- c) CONJUNTO ARTICULADO

- LA BASE, servirá como apoyo mecánico a la plataforma. Es una parte fija.
- LA MESA BALANCIN, en ella se sujetará al paciente que realizará los ejercicios. El paciente estará acostado sobre ella y tendrá esta última correas de sujeción para asegurar al paciente.
- EL CONJUNTO ARTICULADO, constará de 3 articulaciones en las cuales se sujetará una de las piernas del paciente. La otra pierna quedará en el descanso de la mesa balancin.

Cada una de estas partes tendrá un determinado grado de libertad, esto es, cada una tendrá una libertad de movimiento diferente según se muestra en la tabla 2.1'.

Cada una de las articulaciones podrá ceder el número de grados allí especificados, siempre y cuando se haya movido con anterioridad en sentido opuesto, esa misma cantidad o más de grados.

Es decir, por ejemplo en el conjunto ARTICULADO, el movimiento máximo de la articulación 2 en el sentido opuesto a las manecillas del reloj, será cero, cuando llegue a estar en una posición colineal con la articulación 1, es decir, que estén formando una línea recta. Mientras no esté en esta posición, la articulación podrá moverse 20, 30, 50 grados en contra de las manecillas del reloj, previo movimiento de estos grados en contra de las manecillas del reloj.

Esto es por seguridad del paciente, dado que la rodilla no se puede doblar más allá de su posición normal en el sentido contrario a las manecillas del reloj, que cuando es

ta colineal el femur con la pantorrilla.

GRADOS DE LIBERTAD DE LA PLATAFORMA

PARTE DE LA PLATAFORMA	ARTICULACION	(0°) POSICION COLINEAL	
		SENTIDO HORARIO	SENTIDO ANTIHORARIO
B A S E	NINGUNA	-----	-----
M E S A BALANCIN	* UNICA	90°	90°
CONJUNTO ARTICULADO	1	90°	140°
	2	140°	0°
	** 3	90°	60°

Tabla 2.1

** Esta articulación, por ser la última y ser la que moverá los pies del paciente, no es colineal con las otras sino que es perpendicular a estas.

* Estos grados son tomados desde su posición horizontal.

Las articulaciones están controladas por motores Shunt de D.C., con reductor de velocidad integrado cada uno de ellos. El motor va dentro del cilindro que forma la visagra de las diferentes articulaciones. El reductor de velocidad, además de reducir la velocidad de giro, nos proporciona un cambio de velocidad por par.

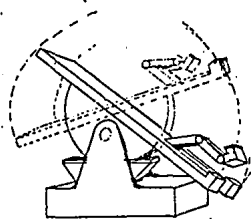
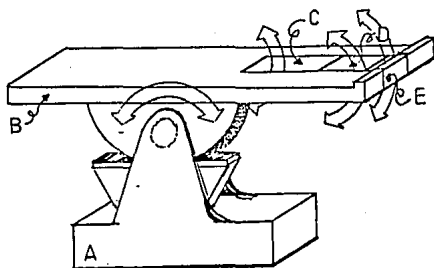
Se escogieron motores Shunt de D.C. por su facilidad de conmutación de sentido de giro!

Los tres motores del conjunto articulado serán de la misma potencia. El motor que moverá la mesa bajancín, por soportar más carga, tendrá que ser de mayor potencia.

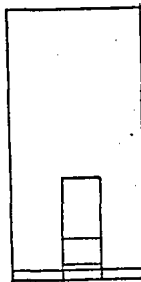
Se piensa puedan ser los primeros de 1H.P. y el segundo de 15H.P. a 115V, c-d.

Las visagras de las articulaciones serán recubiertas de plástico flexible para evitar raspaduras y mordeduras a los pacientes.

Se necesitará un rectificador de corriente alterna directa para alimentar a los motores, éste, no se incluye en este trabajo.



Vista
Superior



Vista
Frontal

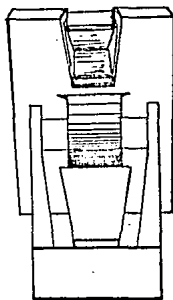


Fig. 2.0

C A P I T U L O I I I

INTERFASE COMPUTADORA-PLATAFORMA.

INTERFASE COMPUTADORA-PLATAFORMA

Esta interfase está diseñada para trasladar la información de control de la computadora a la plataforma. Esta interfase tiene tres diferentes secciones que interactúan entre sí :

- 1) Fase de control de velocidad
- 2) Fase de control de giro
- 3) Fase de selección de motor.

Como ya sabemos, para sus movimientos, la plataforma utiliza motores Shunt de d-c., por tanto, al controlar estos motores, controlaremos los movimientos de la plataforma y de esa manera, las tres fases anteriores se refieren a la velocidad, giro y selección de motores Shunt de d-c.

Fase de control de velocidad

Como ya sabemos, al variar el voltaje aplicado a las terminales de un motor Shunt de d-c., podemos variar su velocidad. Esto se comprueba por la siguiente fórmula:

$$S = \frac{V - I_a R_a}{K\Phi}$$

Donde: S = Velocidad en r.p.m.

V = Voltaje aplicado en terminales

I_a = Corriente de armadura

R_a = Resistencia de armadura

K = Constante (No. de bobinas, tipo de arrollamiento, etc.)

ϕ = Flujo por polo.

(Electrical Machines. Charles Siskind).

Esta variación de velocidad se basará en un tipo de modulación, la llamada (PWM, Pulse Width Modulation), modulada por ancho de pulso.

"La velocidad de un motor Shunt de d-c., puede ser controlada variando el voltaje aplicado a él. Desafortunadamente, el torque depende de la corriente y la corriente decrece cuando decrece el voltaje aplicado, hasta llegar a un punto cuando el torque, debido al voltaje aplicado, es menor que la fricción de los cojinetes y abrazaderas del motor, entonces el motor se para ..." (COMMODORE 64 INTERFACING BLUE BOOK).

"... Una solución a este problema es usar modulación de ancho de pulso (PWM). El voltaje total es aplicado por alguna cantidad de tiempo y después es quitado. Ajustando la razón de prendido (voltaje total aplicado), y apagado (quitado el voltaje total), la energía total va al motor (y su velocidad) es controlada ..." (COMMODORE 64 INTERFACING BLUE BOOK). La figura 3.2 muestra cuatro trenes de pulsos con frecuencia constante pero diferentes ciclos de trabajo. "Una señal con 50% de ciclo de trabajo, es una familiar señal cuadrada simétrica. El área bajo cada tren de pulsos es una medida de la potencia. La potencia se incrementa con el ciclo de trabajo." (MICROPROCESSORS FOR MEASUREMENT CONTROL).

"Usando (PWM), un motor puede ser controlado abajo de 1 r.p.m. o menos sin detenerse. El torque permanece constante mientras disminuye la velocidad, porque cuando está prendido el motor opera a voltaje (y corriente) plena". (COMMO-

DORE 64 INTERFACING BLUE BOOK).

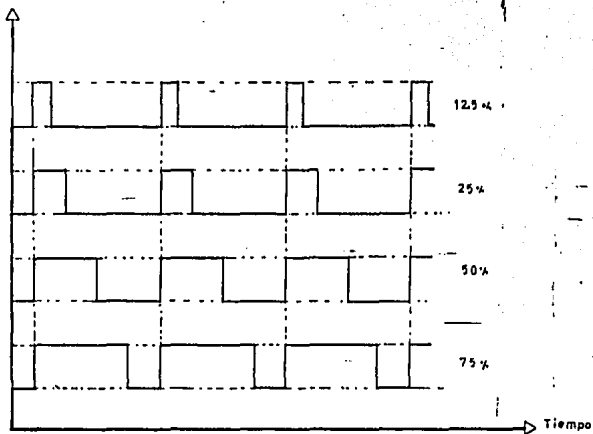
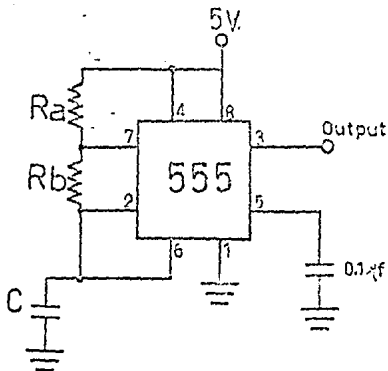


Fig:32

El circuito por mí empleado para realizar la PWM, es - el mostrado en la figura 3.3.

Consta de un oscilador, un contador, un comparador y - una entrada de control (puerto de salida de la computadora).

Oscilador : Está compuesto por un C.I. 555 en su configuración astable. Que es la siguiente :



Oscilador

R_a y R_b son resistencias calculadas según la fórmula siguiente, obtenida de Linear Data-Book, para una determinada frecuencia:

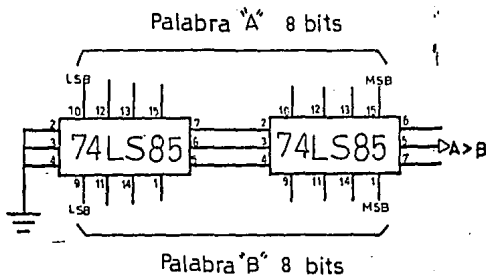
$$f = \frac{1.49}{R_a C + 2R_b C}$$

(Ver anexo 1 para especificaciones).

Comparador : Es un C.I. 74LS85, este es un comparador de 4 bits binarios. Puede conectarse con otros en cascada para comparar palabras de 8,12,16 bits. El control de velocidad maneja una palabra de 8 bits.

Sus interconexiones entre comparadores fueron hechas de la siguiente manera:

(Ver anexo 2 para especificaciones generales).



Comparador

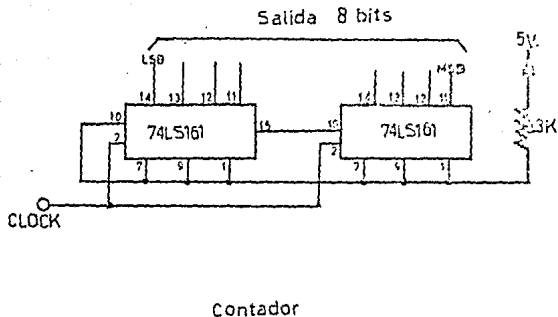
Al ser el valor de los primeros 4 bits de la palabra A mayores que sus correspondientes en la palabra B, se pondrá en estado lógico "1" la salida 5 del primer comparador. -- Llenando a la entrada 4 en cascada del segundo comparador.

Contador : Es un C.I. que nos da en su salida una secuencia de pulsos en un determinado orden, según sea el código empleado (BCD, binario natural, GRAY, etc.) dependiendo de una entrada generalmente llamada reloj (clock).

Se escogió el C.I. 74LS161, por ser un contador binario.

En este circuito se emplean dos contadores 74LS161 colocados en cascada para obtener un número a la salida de 8 bits.

Este 74LS161 es un contador binario, es decir, en su salida aparecerá un número binario sucesivo, cada vez que el clock da un pulso.



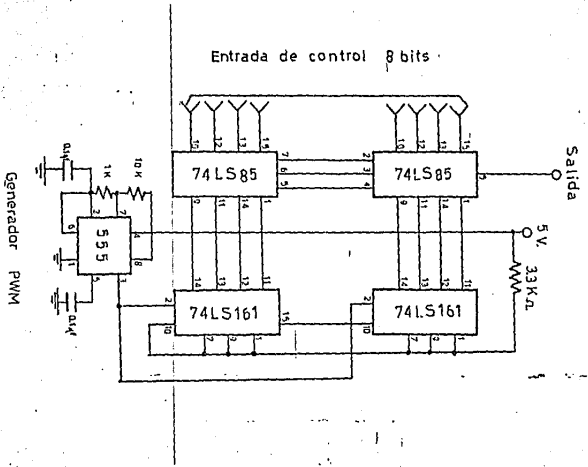
(Ver anexo 3 para especificaciones generales).

El circuito completo, es el mostrado en la figura 3.3 y funciona de la siguiente manera :

El oscilador controla directamente el contador, esto es, cada vez que el oscilador genera un pulso, el contador incrementa en 1 su conteo.

En el comparador se están comparando dos señales: La que proviene del contador, contra la que viene del puerto de salida de la computadora.

Mientras el número generado por el contador esté por debajo del de la entrada de control (puerto de salida), la salida del comparador será "1" lógico. Cuando el número generado por el contador es mayor que el de la entrada de control, la salida es un "0" lógico.

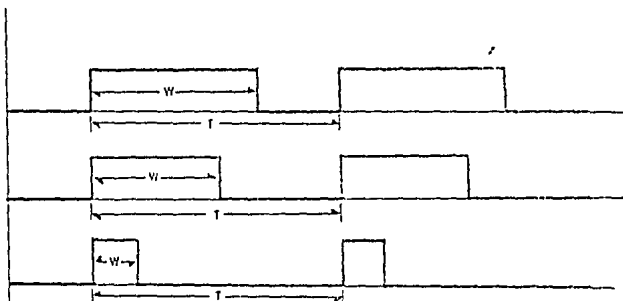


Generador PWM
Fig. 3.3

La salida del comparador cambia de estado lógico, es decir, de 0 a 1 o de 1 a 0, una vez cada ciclo completo del contador.

Como la frecuencia producida por el oscilador es constante, el ciclo completo del contador es también constante, por tanto el período de la onda de salida del comparador es constante (T). Cuando nosotros cambiamos la entrada de control al comparador, el ancho (W) varía dandonos una señal PWM, sin variar el período. Esto se ilustra en la figura 3.4.

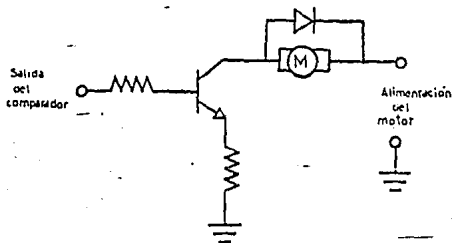
Señales obtenidas a la salida del comparador.



Señales PWM

Fig.34

Si esta señal controlara el disparo de un transistor de potencia, el cual controlara el flujo de corriente hacia un motor, como en el siguiente circuito:



Circuito que opera con señales PWM

Podríamos obtener un control sobre la continuidad o no continuidad del circuito, por tanto, sobre la alimentación del motor y por consecuencia de su velocidad.

NOTA : En realidad, no se opera directamente a la base del transistor de potencia con la señal de salida del comparador, sino en un arreglo Darlington, con otro transistor o transistores para obtener la suficiente corriente para el disparo de la base del transistor de potencia.

El diodo en paralelo con el motor es para disparar el pulso inductivo que ocurre cuando el transistor se pone en corte.

Fase de control de giro

En un motor d-c., el cambio en el giro del motor lo obtenemos simplemente, cambiando la polaridad de alimentación del motor, y esto puede ser hecho muy fácilmente con transistores de potencia, o con relevadores.

El circuito por mí diseñado es el mostrado en la figura 3.5 .

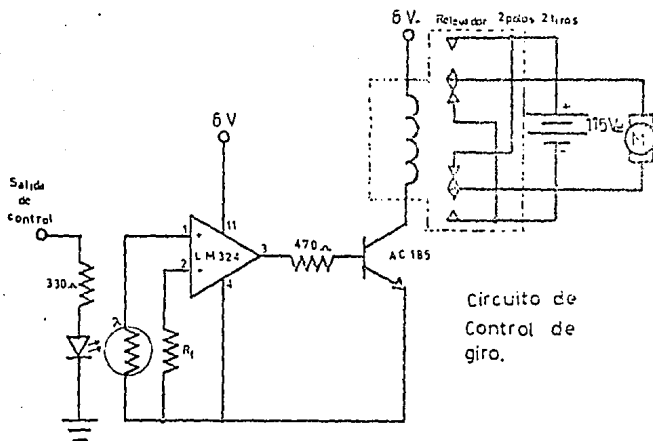


Fig. 3.5 .

El cual funciona de la siguiente manera :
 Se tiene una salida de control (puerto de la computadora), la cual está conectada a un diodo emisor de luz (LED), cuando la salida de control se pone en "1" lógico, es decir, 5V, el LED prende; lo cual ocasiona una variación en la fotore-

sistencia conectada al OPAMP, lo cual hará que exista un voltaje diferencial en las entradas al OPAMP, dado que la otra resistencia es fija, lo cual hará que exista un determinado voltaje a la salida del OPAMP, el cual hará que se dispare el transistor, cerrando el circuito que alimenta la bobina del relevador, haciendo esta bobina, cerrar los contactos normalmente abiertos y abrir los normalmente cerrados, con lo cual se ocasiona que el motor sea alimentado por la fuente de alimentación con una determinada polaridad y por tanto, con un determinado sentido de giro.

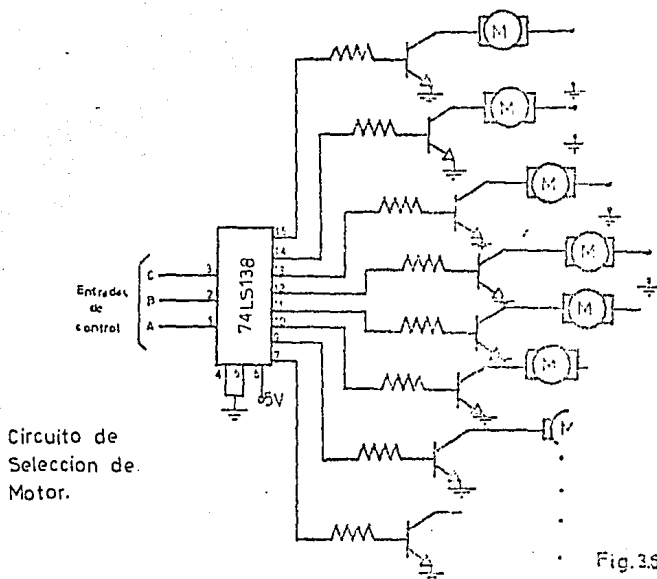
Cuando no existe señal en la salida de control, el LED está apagado, con lo cual la fotoresistencia no sentirá ningún cambio en su resistencia, por tanto no existirá ningún voltaje diferencial en el OPAMP, por tanto no existirá voltaje a la salida capaz de disparar el transistor, con lo cual no se cerrará el circuito que alimenta a la bobina y por tanto no cerrará los contactos normalmente abiertos, si no que quedarán cerrados los contactos normalmente cerrados que permiten que el motor sea alimentado por su fuente de alimentación pero con la polaridad opuesta a la del caso anterior, por tanto el motor girará en sentido opuesto.

NOTA : La fotoresistencia deberá de tener la misma resistencia que la resistencia de referencia R_f , para que cuando el LED esté apagado, no exista voltaje entre las dos terminales de entrada del operacional. Se está suponiendo que el OPAMP está completamente balanceado, es decir, cuando en las entradas existen cero volts, en la salida se obtiene cero volts. El relevador es un relevador 2 polos, 2 tiros.

Fase de selección de motor

Esta fase está diseñada con un decodificador de 3x8, - en el cual dada una señal de entrada de 3 bits, nos selecciona una de las 8 salidas, en nuestro caso estamos utilizando un decodificador TTL, 74LS138.

El circuito es el mostrado en la figura 3.6 .



Al poner un determinado código en las entradas A,B,C, - se selecciona una de las 8 salidas, la cual se pone en estado lógico "0" y las demás permanecen en estado lógico "1" ,

si se invierte esa señal y se alimenta un transistor que controle la alimentación o no-alimentación del motor, se podrá seleccionar así cada motor.

(Ver anexo 4 para especificaciones generales).

La interfase completa

Ahora que tenemos cada una de las fases de control, es decir, velocidad, giro y selección de motor, podremos ahora construir la interfase, que relacionará las tres fases unas con otras. Esta interrelación ha sido lograda con el circuito de la figura 3.7 .

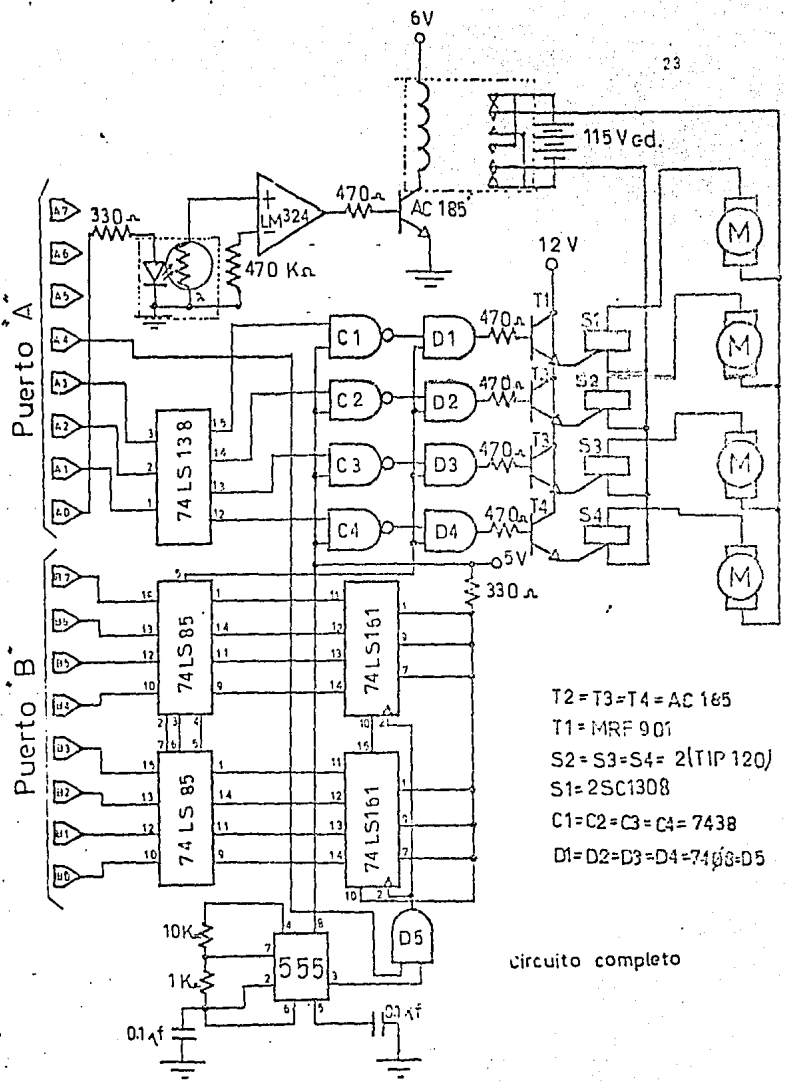
El circuito completo funciona de la siguiente manera :

La plataforma tiene 4 articulaciones, por tanto 4 motores a controlar.

En el decodificador utilizaremos únicamente 4 de sus salidas, las otras cuatro sirven para estados de no-operación, es decir, para no direccionar algún motor. Como la señal de salida del decodificador es "0" y todas las demás son "1", es necesario invertir esa salida y que únicamente tenga señal "1" a la salida direccionada, esto se logró por medio de compuertas lógicas NAND (7400), con una pata en común todas ellas hacia un "1" lógico.

Esto se ve más claro con la tabla de verdad de la compuerta NAND que a continuación muestro :

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



- T2 = T3 = T4 = AC 165
- T1 = MRF 901
- S2 = S3 = S4 = 2(TIP 120)
- S1 = 2SC1308
- C1 = C2 = C3 = C4 = 7438
- D1 = D2 = D3 = D4 = 7408 = D5

circuito completo

Fig. 3.7

Como las salidas no seleccionadas del decodificador son "1" y la otra entrada a la compuerta NAND es "1" lógico, entonces su salida será un "0" lógico como es de esperar. - En cambio, cuando la salida seleccionada del decodificador es "0" y la otra entrada a la compuerta NAND es "1" lógico, su salida será "1" lógico, nunca se dará el caso en que las dos entradas de la compuerta NAND sean ceros.

Una vez escogido uno solo de los motores, por la señal de entrada del decodificador, proporcionada por la computadora, uniremos ahora el circuito de control de velocidad, y esto lo haremos sencillamente con compuertas AND (7408).

Una de las entradas de cada una de las 4 compuertas AND van directamente a la salida del comparador que nos da la señal PWM, que es una señal con estados lógicos "1" y "0". La otra entrada de cada una de las compuertas AND va a la salida de las compuertas NAND de selección de motor.

Así como las compuertas AND tienen salida "1" lógico, cuando sus dos entradas son "1" lógicos, se tendrá a la salida de las compuertas AND el motor seleccionado y su señal PWM.

Esta última señal, hará que uno de los transistores se ponga en su estado de corte o saturación, dependiendo de la señal PWM, y el transistor activará el gatillo de los TR's para cerrar el circuito de alimentación de los motores, el TR estará en su estado de conducción hasta que se desactive su gatillo, el cual se desactivará cuando el transistor pase a su estado de corte, el cual lo estará dependiendo de la señal PWM, que llegue a su base.

El sentido de giro del motor lo dará la polarización que se le da al motor, la cuál está controlada por el abrir o cerrar de los contactos normalmente cerrados de los relevadores y abrir o cerrar los normalmente abiertos.

Esto se controla por la presencia o ausencia de la señal en A0, es decir, el prendido ó apagado del LED.

Como se ve, los motores tienen un sentido de giro ya asignado por los contactos normalmente cerrados y este se cambiará hasta que exista una señal de "1" lógico en A0.

Cuando no se haya escogido algún motor, con la señal de salida de control, que es la entrada al decodificador, - mandará la señal a una de las líneas de no-operación, lo que causará un "1" en las entradas de las compuertas NAND y esta ocasionará un "0" en su salida, lo cual hará tener en la salida de las compuertas AND un "0" lógico por lo cual, no habrá disparo para el transistor ni para el TR, por tanto el circuito de alimentación se encuentra abierto.

La tabla de verdad que controla el circuito está mostrada en la figura 3.8 .

La presencia o ausencia de señal en A4, hará que comience a contar el contador. Mientras esta señal sea "0" lógico, el contador no comenzará a contar.

Como 3 de los 4 motores son de las mismas características, los transistores, TR's y relevadores que los controlan serán iguales. El otro transistor, TR y relevador será por supuesto de mayor potencia.

El circuito TR como yo lo designé es el mostrado en la figura 3.9 .

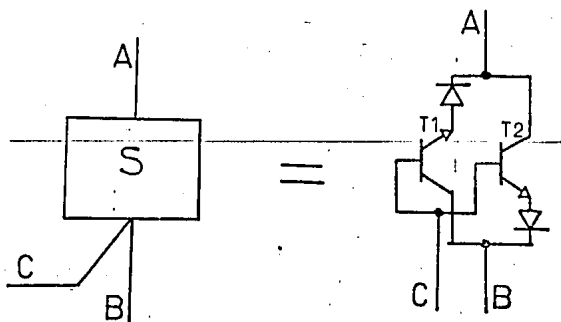
El TR trabaja como un triac, solamente con la diferencia que únicamente existe conducción entré los puntos A y B cuando existe corriente por las bases de los transistores . Cuando no exista, no habrá conducción.

La conducción se llevará a cabo solamente por uno de los transistores, dependiendo de la polaridad que exista entre los puntos A y B, es decir, cuando el punto A esté a un potencial mayor que B y exista corriente en su base conduci

rá T2; T1 estará polarizado inverso, o sea en su estado de corte, así que no existirá conducción por él, aunque exista corriente en su base. Cuando B esté a un potencial mayor que A y exista corriente en las bases de los transistores, conducirá T1; T2 estará en corte.

Esto se realiza así porque los puntos A y B cambiarán de polaridad, y de esta dependerá el sentido de giro del motor.

Los diodos puestos en los emisores de los transistores de potencia se utilizan para evitar una corriente inversa - al apagar el motor, por ser una carga inductiva.



Circuito TR.

Fig. 3.9

Tabla de verdad de la interfase









A0	A1	A2	A3	A4	MOTOR	GIRO
0	0	0	0	1	1	
0	0	0	1	1	2	
0	0	1	0	1	3	
0	1	1	0	1	4	
0	1	0	0	X	NO	OPERA
0	1	0	1	X	NO	OPERA
0	1	1	0	X	NO	OPERA
0	1	1	1	X	NO	OPERA
1	0	0	0	1	1	
1	0	0	1	1	2	
1	0	1	0	1	3	
1	0	1	0	1	4	
1	1	0	0	X	NO	OPERA
1	1	0	1	X	NO	OPERA
1	1	1	0	X	NO	OPERA
1	1	1	1	X	NO	OPERA

Fig. 3.8

C A P I T U L O I V

PROGRAMACION DE LA COMPUTADORA.

PROGRAMACION DE LA COMPUTADORA

Realizé el programa de control en BASIC 4.0, versión -
COMMODORE 64 por ser un lenguaje accesible a todas las per-
sonas, y a la mayoría de las microcomputadoras que tengan -
puertas de entrada/salida de 16 bits.

El programa nos presenta cinco opciones posibles (des-
plegado 4.1) (líneas 30-100).

- 1) Movimientos de control directo
- 2) Almacenamiento de información
- 3) Lectura y ejecución de información
- 4) Movimientos de prueba
- 5) Salida

- Los movimientos de control directo

Esto quiere decir que se podrá mover la plataforma in-
mediatamente, dependiendo de la opción escogida, del siguien-
te menú que nos aparecerá en pantalla (desplegado 4.2) (lí-
neas 170-230).

- 1) Movimientos controlados por teclas
- 2) Movimientos controlados por palanca
- 3) Menú principal
- 4) Salida

Si la opción escogida fué por teclas, nosotros podre-
mos controlar los movimientos de la plataforma por medio de
presionar algunas teclas que se irán indicando a través de

todo el programa.

Si la opción escogida fué por palanca de control, podremos controlar los movimientos de la plataforma por medio de mover la palanca que está conectada directamente al puerto 2 de la computadora (entrada para joystick) y todas las indicaciones irán apareciendo en pantalla para un control correcto.

Si la opción escogida fué menú principal, pasará de nuevo a las opciones generales, para poder escoger alguna de ellas.

Si la opción escogida fué salida, la máquina terminará el programa y pasará a estar en modo inmediato, teniendo aún así el programa en memoria.

En las opciones 1 y 2, se despliega una secuencia de instrucciones, para el manejo de estas. (desplegado 4.3).

- Almacenamiento de datos.

Con esta opción general, podremos almacenar una enorme cantidad de movimientos en diskette floppy 5"1/4, cada uno de los ejercicios que va a realizar el paciente está compuesto por varios movimientos, cada uno de estos movimientos quedarán almacenados en el diskette, con un determinado nombre para cada ejercicio completo.

El desplegado mostrado en la pantalla al escoger esta opción, será el mostrado en el desplegado 4.4.

Se están utilizando archivos secuenciales para almacenar los movimientos de cada ejercicio.

- Lectura y ejecución de datos almacenados.

Con esta opción general, leeremos los datos almacenados en el diskette y los ejecutará la máquina.

Aparecerá un desplegado en la pantalla como el siguiente : (desplegado 4.5).

Si se desea realizar algún ejercicio completo almacenado, se dará el nombre dado a ese ejercicio. El programa buscará ese nombre dentro de todos los nombres que tiene almacenados. Si no lo encuentra, despliega en pantalla : "ejercicio no grabado en el archivo", y pregunta otra clave. Si lo encontró, ejecuta cada uno de los movimientos en una secuencia y cuando lo termina todo el ejercicio, pregunta el programa si se desea repetir o buscar otro ejercicio, si se desea repetir, se repite la ejecución del ejercicio desde su primer movimiento; si no se desea repetirlo, nos manda al desplegado 4.5 .

Si no se recuerda la clave del ejercicio, el programa desplegará en pantalla todos los nombres de los ejercicios que se tienen grabados, después de esto, se podrá escoger uno de ellos o pasar al desplegado 4.5 .

- Movimientos de prueba.

Esta opción general, nos sirve para probar cualquier movimiento, ya sea de control directo (teclas o por palanca) o ya previamente almacenado en diskette.

Esta es otra manera de entrar a cada una de las dos opciones generales mencionadas.

- Salida.

Esta opción general termina el programa y deja a la computadora en su forma inmediata. Reteniendo en su memoria el programa.

El programa tiene por nombre REHABILITACION y el archivo de datos de los movimientos es un archivo secuencial llamado DATOS.

Se recomienda tener un respaldo (una copia) del programa y otra del archivo, por cuestiones de seguridad.

El programa no podrá listarse en pantalla por tener un seguro que deshabilita el comando LIST. Esto se logró por medio de la instrucción POKE 775,200 que deshabilita el comando. Esto se hizo por cuestiones de seguridad, para evitar cambios al programa.
(Ver la nota 1 anexa).

En el programa se utilizaron algunos caracteres imprimibles solamente en la computadora COMMODORE 64. Estos son los siguientes que pueden ser cambiados para ser aplicados en otras computadoras, aquí mencionamos los caracteres aplicados en computadora RADIO SHACK y APPLE IIe.

C A R A C T E R

<u>Comentario</u>	<u>COMMODORE 64</u>
Limpiar pantalla	PRINT " " "
Cambiar color pantalla	POKE 53281,1
Abrir un archivo secuencial	OPEN 3,8,3,"Ø:DATOS,S,W"
Escribir en archivo DATOS	PRINT#3
Leer del archivo DATOS	INPUT#3

<u>Comentario</u>	<u>RADIO SHACK</u>
Limpiar pantalla	CLS
Cambiar color pantalla	COLOR 1
Abrir un archivo secuencial	OPEN "DATOS",1
Escribir en archivo DATOS	PRINT#1
Leer del archivo DATOS	INPUT#1

<u>Comentario</u>	<u>APPLE IIe</u>
Limpiar pantalla	HOME
Cambiar color pantalla	COLOR=1
Abrir un archivo secuencial	OPEN "DATOS SECUENCIAL"

ComentarioAPPLE IIe

Escribir en archivo DATOS WRITE "DATOS"
 Leer del archivo DATOS READ "DATOS"

Se está utilizando para meter un valor a una localización de memoria la instrucción POKE teniendo el formato siguiente :

POKE Localidad de memoria, valor a poner en localidad

Para leer el valor puesto en una localidad específica se está utilizando la instrucción PEEK, teniendo el formato siguiente :

PEEK (Localidad de memoria)

Como se mencionó antes, el CIA (COMPLEX INTERFACE ADAPTER) tiene unas localidades de memoria para direccionar los puertos A y B como entrada o como salida, a estas localidades se les denomina DDR (Data Direction Register). El DDR de cada uno de los puertos es de 8 bits, es decir, podemos direccionar independientemente como entrada o como salida cada uno de esos 8 bits, para direccionarlos como salida, cualquiera de esos bits es necesario poner un "1" lógico en el bit requerido, y si se desea que sea entrada, se pondrá un "0" lógico en el bit requerido. Esto se puede resumir en la tabla siguiente :

LOCALIDAD DE MEMORIA	PUERTO HABILITADO	ESTADO LOGICO POR BIT	
		ENTRADA	SALIDA
5 9 5 7 8	A	0	1
5 6 5 7 9	B	0	1

M E N U

- 1) MOVIMIENTO DE CONTROL DIRECTO
 - 2) ALMACENAMIENTO DE INFORMACION
 - 3) LECTURA Y EJECUCION DE INFORMACION ALMECENADA
 - 4) MOVIMIENTOS DE PRUEBA
 - 5) SALIDA
- OPCION ?
-

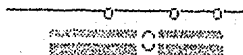
M E N U

- 1) MOVIMIENTOS CONTROLADOS POR TECLAS
- 2) MOVIMIENTOS CONTROLADOS POR PALANCA
- 3) MENU PRINCIPAL
- 4) SALIDA

OPCION 2

LA PLATAFORMA TIENE 4 ARTICULACIONES
LAS CUALES ESTAN ASIGNADAS (A,B,C,D)

B C D



A

MOVILES,
COMO SE MUESTRAN:

LA ASIGNACION DE CUAL ARTICULACION VA
REALIZAR POR MEDIO DEL TECLADO O POR
DE CONTROL; DEPENDIENDO DE CUAL OP -
SECUNDO MENU

A MOVER, LA PODRA
MEDIO DE LA PALANCA
CION ESCOGIO EN EL

PRESIONE LA BARRA ESPACIADORA PARA

CONTINUAR

UNA VEZ ESCOGIDA LA ARTICULACION A MO-
TENER UNO DE DOS MOVIMIENTOS: EN SEN-
LAS MANECILLAS

VER, ESTA PUEDE
TIDO DEL MOVIMIENTO DE

DEL RELOJ, O, EN CONTRA DEL MOVIMIENTO DE LAS MENCILLAS DEL
RELOJ, TAMBIEN ESTA ASIGNACION LA PODRA
REALIZAR DES- DE EL TECLADO O POR MEDIO DE LA
PALAN- CA DE CONTROL DEPENDIENDO
DE LA OPCION ESCOGIDA.

PRESIONE LA BARRA ESPACIADORA PARA CONTINUAR

CUANDO YA SE TIENE LA ARTICULACION
ESCOGIDA, ES NECESARIO ESCOGER LA VE-
ULTIMA NO SE ESPECIFICA, SE ASUME UNA

Y EL GIRO DE ESTA
LOCIDAD DE GIRO, SI ESTA
VELOCIDAD MEDIA

PARA ALMACENAR LA INFORMACION SE RE-
 COMO DE LA PALANCA DE CONTROL.
 EL PRIMER DATO A INTRODUCIR SERA,
 UNA LETRA Y UN NUMERO. EJEM: AS.
 QUE DESIGNARA EL INICIO DE UNA SERIE
 ENSEGUIDA SE DESIGNARA LA ARTICULA-
 ESTO SE HARA POR MEDIO DE ESAS TECLAS
 ENSEGUIDA SE ASIGNARA EL GIRO POR ME-
 (M)- GIRO EN SENTIDO DE LAS MANECILLAS
 (N)- GIRO EN CONTRA DE LAS MANECILLAS
 PARA CONTINUAR CUALQUIER TECLA

QUERIRA TANTO LAS TECLAS

ACION A MOVER (A,B,C,D)

DIO DE LAS TECLAS:
 DEL RELOJ
 DEL RELOJ

READY.

OTRO MOVIMIENTO (S/N)?

- A) MENU PRINCIPAL
 - B) SALIDA
-

OPCION?

Desplegado 4.5

Una vez que se tiene que tipo de direccionamiento es - en cada bit de cada puerto, es necesario poner datos si es salida, o leer datos si es entrada.

En este caso todos los bits de los dos puertos A y B - están direccionados como salidas, por tanto, para poner datos en cualquier puerto se requiere de otras localidades de memoria distintas del DDR de cada puerto para que contengan el dato que aparecerá en binario natural a la salida de los puertos, estas localidades de memoria son las siguientes :

LOCALIDAD DE MEMORIA	PUESTO
5 9 5 7 6	A
5 6 5 7 7	B

Por ejemplo : Si se quisiera tener el puerto B como salida y tener 5V ("1"-lógico) en los bits 2,4,6. Sería necesario dar las siguientes instrucciones:

POKE 56579,255 --- Habilita todos los pines del puerto B como salida
 POKE 56577,42 --- Pone en los pines 2,4 y 6 ; 5V

8	7	6	5	4	3	2	1
0	0	1	0	1	0	1	0

1 - 5V.

0 - 0V.

De esta manera, programamos los puertos de la computadora (ver nota 2).

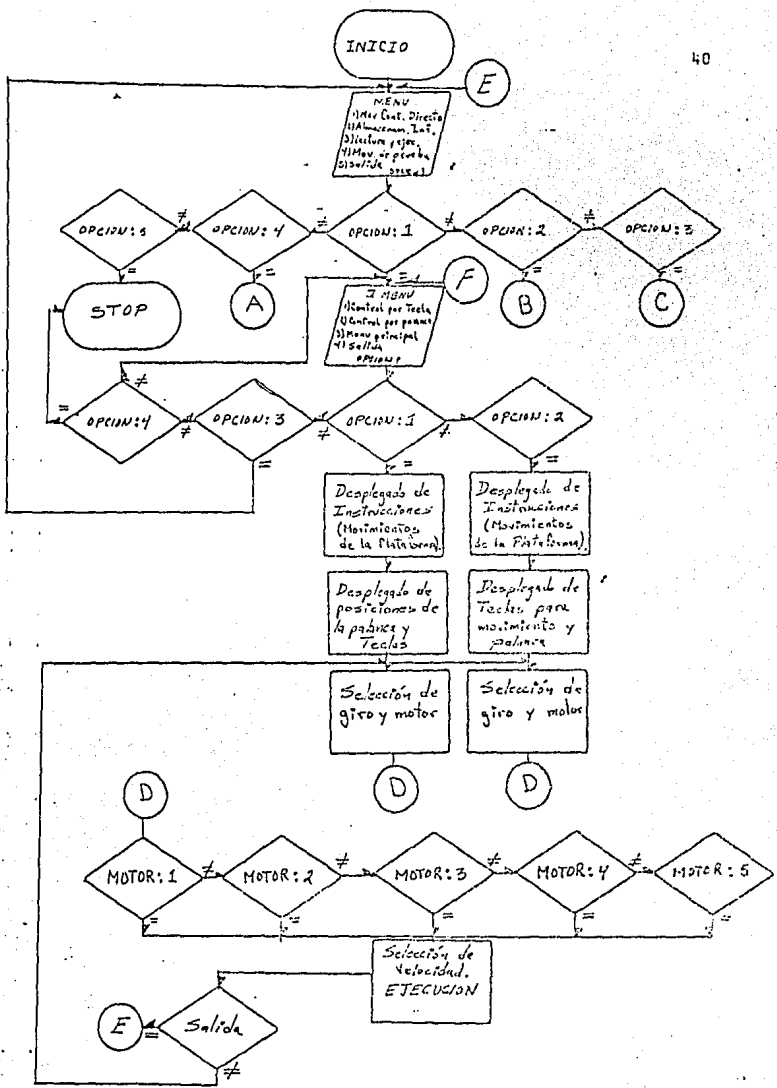
El desplegado 4.7 muestra el listado del programa completo. El desplegado 4.6 muestra el diagrama de flujo completo.

Como en el programa podemos cambiar la velocidad de giro de las articulaciones, si no se especifica la velocidad de giro, la máquina asume una velocidad media por $V = 128$.

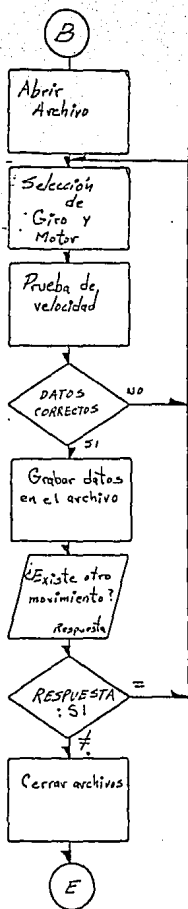
El tiempo de giro es también calculado por medio del programa con las constantes I y J y el tiempo total de cada uno de los movimientos se tiene almacenado en la variable numérica CO.

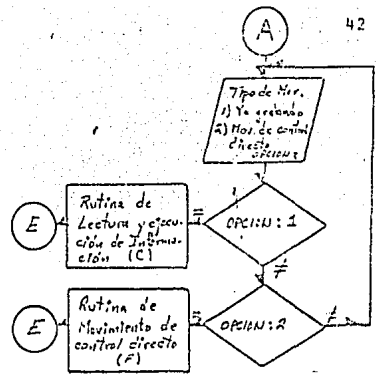
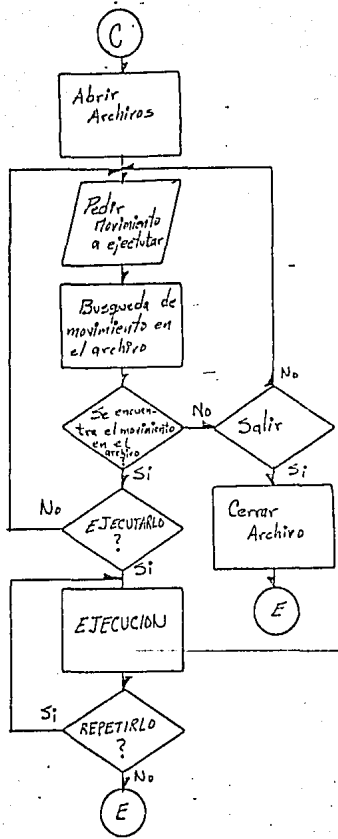
NOTA 1 : Para habilitar el comando LIST y poder hacer cambios o correcciones, teclear la siguiente instrucción : POKE 775,167 .

NOTA 2 : Las localidades de la memoria cambian para cada computadora, dependiendo del microprocesador empleado y de la fábrica que construye la computadora.



Desplegado 4.6a





```

3 POKE 775,200
5 DA=59578:DB=56579:PA=59576:PB=56577:V=128:I=0:J=0:CF="",
10 PRINT"D":REM LIMPIA PANTALLA
20 PRINT:PRINT:PRINT
30 PRINT "          M E N U"
40 PRINT:PRINT
50 PRINT " 1) MOVIMIENTO DE CONTROL DIRECTO":PRINT
60 PRINT " 2) ALMACENAMIENTO DE INFORMACION":PRINT
70 PRINT " 3) LECTURA Y EJECUCION DE INFORMACION... ALMECENADA":PRINT
80 PRINT " 4) MOVIMIENTOS DE RUEDA":PRINT
90 PRINT " 5) SALIDA ":PRINT:PRINT:PRINT
100 PRINT " OPCION ?"
110 GET A: IF A<=0 OR A>5 THEN 110
120 ON A GOTO 150,1000,3000,4000,410
150 REM OPCION DE MOVIMIENTOS DE CONTROL DIRECTO
160 PRINT"D":REM LIMPIA PANTALLA
170 PRINT:PRINT:PRINT "          M E N U"
180 PRINT:PRINT:PRINT
190 PRINT " 1) MOVIMIENTOS CONTROLADOS POR TECLAS ":PRINT
200 PRINT " 2) MOVIMIENTOS CONTROLADOS POR PALANCA" :PRINT
210 PRINT " 3) MENU PRINCIPAL" :PRINT
220 PRINT " 4) SALIDA"
230 PRINT:PRINT:PRINT:PRINT" OPCION ?"
240 GET B:IF B<=0 OR B>4 THEN 240
250 ON B GOTO 260,420,5,410
260 GOSUB 15000:REM RUTINA DE DESPLEGADO DE INSTRUCCIONES"
270 PRINT"D":PRINT:PRINT:PRINT
280 PRINT" ESCOJA ARTICULACION (A,B,C,D)          TECLANDO (1,2,3,4) RESPE"
290 PRINT "*****EFFECTIVAMENTE, SI TECLAS UN 5 NO SE MUEVE NINGUNA"
300 GET E:IF E<=0 OR E>5 THEN 300
310 POKE DA,255:POKE DB,255:REM DIRECCIONA LOS PUERTOS COMO SALIDA
320 GOSUB 5000:REM RUTINA DE SENTIDO DE GIRO(TECLAS)
330 IF E=1 THEN S=0+A0:GOTO390:REM *****
340 IF E=2 THEN S=8+A0:GOTO390:REM ***
350 IF E=3 THEN S=4+A0:GOTO390:REM ** ASIGNACION DE DATOS PUERTO A **
360 IF E=4 THEN S=12+A0:GOTO390:REM***
370 IF E=5 THEN S=2 :GOTO 390:REM *****
380 GOTO 300
390 GOSUB 6000:REM HACIA LA RUTINA DE CONTROL DE VELOCIDAD (TECLAS)
400 GOTO 5
410 END
420 GOSUB 15000
430 PRINT"D":PRINT:PRINT
440 PRINT " LA PALANCA DE CONTROL PUEDE ESTAR EN "
450 PRINT " PRIMERA: DESIGNACION DE ARTICULACION; SEGUNDA: GIRO Y VELOCIDAD"
460 PRINT " EN UN PRINCIPIO LA PALANCA ESTA EN LA PRIMERA POSICION"
470 PRINT " PARA PASAR A LA SEGUNDA OPCION PRECIO- NAR EL DOTON ROJO"
480 PRINT " EN LA PRIMERA OPCION SE ESCOGE LA AR- TICULACION."
490 PRINT " DE ACUERDO CON LA SIGUIENTE DESIG -- NACION:"

```

READY.

```

495 PRINT
500 PRINT " MOVIMIENTO DE LA PALANCA ARTICULACION"
510 PRINT "     HACIA ARRIBA           A     "
520 PRINT "     HACIA ABAJO           B     "
530 PRINT "     HACIA IZQUIERDA        C     "
540 PRINT "     HACIA DERECHA          D     "
545 PRINT "     DIAGONAL              SALIDA "
550 PRINT "     PUNTO MUERTO          NINGUNA " :PRINT
555 FOR U= 1 TO 10000:NEXT U
560 PRINT " UNA VEZ ESCOGIDA LA ARTICULACION, RECESAR LA PALANCA EL "
565 PRINT "*****PUNTO MUERTO"
570 PRINT " ENSEGUIDA, PRECIONAR EL BOTON ROJO Y MANTENERLO OPRINIDO PARA
580 PRINT " PARA PASAR A LA SEGUNDA OPCION. "
590 PRINT " EN ESTA OPCION SE REALIZAN LAS SI- CUENTES OPERACIONES:" :PRINT
600 PRINT "MOVIMIENTO DE PALANCA ACCION REALIZADA" :PRINT
610 PRINT " HACIA ARRIBA -AUMENTA VELOCIDAD DE GIRO"
620 PRINT " HACIA ABAJO -DISMINUYE LA VELOCI- DAD DE "
625 PRINT " *****GIRO"
630 PRINT " HACIA DERECHA -GIRO EN SENTIDO DE LAS MANE
640 PRINT "*****BELLAS DEL RELOJ"
650 PRINT " HACIA IZQUIERDA -GIRO EN CONTRA DE LAS MANE
660 PRINT "*****BELLAS DEL RELOJ"
670 PRINT " PUNTO MUERTO -NINGUNA" :PRINT
675 FOR Q=1 TO 5000:NEXT Q
680 PRINT " CUANDO DESEE CONTINUAR PRESIONE CUAL-- QUIER TECLA"
690 GET F1:IF F1="" THEN 690
700 PRINT "Q" :PRINT:PRINT
710 PRINT " CUANDO QUIERA COMENZAR PRESIONE EL DO- TON ROJO"
720 A=PEEK(56320):REM LEE NO. DE MOVIMIENTO
730 IF A=111 THEN 720
740 COSUB 8000:REM RUTINA DE ESCOGENCIA DE ARTICULACION
750 POKE 56281,1:REM DESPLEGADO DE MOVIMIENTO ACELERADO
760 COSUB 8100:REM RUTINA DE GIRO Y VELOCIDAD
770 PRINT " OTRA ARTICULACION ?":INPUT E1
780 IF E1="SI" THEN I=0:J=0:GOTO 710
790 IF E1="NO" THEN GOTO 5
800 GOTO 770
999 REM RUTINA DE ALMACENAMIENTO
1000 INPUT " EXISTEN DATOS EN EL DISKETTE(S/N)":JMS:IF JMS="N" THEN 1010
1005 IF JMS="S" THEN OPEN 3,2,3,"0:DATOS.A":GOTO 1020
1006 GOTO 1000
1010 OPEN 3,2,3,"09:DATOS.S.M":REM ABRIR ARCHIVOS
1020 PRINT "Q" :PRINT:PRINT
1030 PRINT " PARA ALMACENAR LA INFORMACION SE RE- QUERIRA TANTO LAS TECLAS"

```

READY.

```

1030 PRINT "PARA ALMACENAR LA INFORMACION SE RE- QUERIRA TANTO LAS TECLAS"
1040 PRINT"COMO DE LA PALANCA DE CONTROL."
1050 PRINT"EL PRIMER DATO A INTRODUCIR SERA,"
1060 PRINT"UNA LETRA Y UN NUMERO. EJEM: AB."
1070 PRINT"QUE DESIGNARA EL INICIO DE UNA SERIE"
1081 PRINT"DE MOVIMIENTOS DE UN SOLO EJERCICIO."
1085 PRINT "EJERCICIO."
1090 PRINT " ENSEGUIDA SE DESIGNARA LA ARTICULA- CION A MOVER (A,B,C,D)"
1095 PRINT " ESTO SE HARA POR MEDIO DE ESAS TECLAS"
1100 PRINT " ENSEGUIDA SE ASIGNARA EL GIRO POR ME- DIO DE LAS TECLAS:"
1110 PRINT " (M)- GIRO EN SENTIDO DE LAS MANECILLAS DEL RELOJ"
1111 PRINT " (N)- GIRO EN CONTRA DE LAS MANECILLAS DEL RELOJ"
1111 PRINT"PARA CONTINUAR CUALQUIER TECLA"
1112 GET$=IF$="" THEN 1112
1120 PRINT " DESPUES SE ASIGNARA LA VELOCIDAD DE GIRO, SIGUIENDO LA SIG."
1125 PRINT "XXXXXXXXXXXXXXXXX TABLA : "
1130 PRINT " TECLA ACCION "
1140 PRINT " M VELOCIDAD NOMINAL "
1150 PRINT " S AUMENTO DE VELOCIDAD"
1160 PRINT " D DISMINUCION DE VELOCIDAD"
1170 PRINT " SI DESEA AUMENTAR LA VELOCIDAD DE GI- RO CON LA PALANCA DE "
1175 PRINT "XXXXXXXXXXXXXXXXXXXXX CONTROL"
1180 PRINT " EN MODO NORMAL (SIN PRECIONAR EL BO- TOM ROJO) MUEVALA HACIA "
1185 PRINT "XXXXXXXXXXXXXXXXXXXXX ARRIBA"
1190 PRINT " Y MANTENGA LA ALLI HASTA QUE SE ALCAN- CE LA VELOCIDAD DESERADA"
1200 PRINT " SI SE DESEA DISMINUIR LA VELOCIDAD MUE- VA LA PALANCA HACIA ABAJO"
1210 PRINT " CUANDO HAYA TERMINADO EL MOV. DE LA ARTICULACION PRECIONE EL"
1220 PRINT "XXXXXXXXXXXXXXXXXXXXX BOTON ROJO. ENSEGUIDA APARECERA UNA PREGUNTA "
1225 PRINT "XXXXXXXXX PRE- GUNTANDOLE"
1230 PRINT " SI EXISTE OTRO MOV. DEL MISMO EJERCI- CIO."
1240 PRINT " CUANDO ESTE LISTO PRECIONE LA BARRA ESPACIADORA"
1250 GET G$:IF G$="" THEN 1250
1260 IF G$=CHR$(32) THEN GOTO 1280
1270 GOTO 1250
1280 INPUT " LETRA QUE DESIGNARA AL EJERCICIO";H$
1290 INPUT " ARTICULACION A-MOVER";I$
1300 IF I$="A" THEN A=0:GOTO 1350
1310 IF I$="B" THEN A=8:GOTO 1350
1320 IF I$="C" THEN A=4:GOTO 1350
1330 IF I$="D" THEN A=12:GOTO 1350
1340 GOTO 1290
1350 PRINT " GIRO (M/N)":GET J$:IF J$="" THEN 1350
1360 IF J$="M" THEN A0=0:GOTO 1390
1370 IF J$="N" THEN A0=1:GOTO 1390
1380 GOTO 1350
1390 PRINT "J":PRINT:PRINT:PRINT

```

READY.

Desplegado.4.7c

```

1390 PRINT "D":PRINT:PRINT:PRINT
1400 PRINT " VELOCIDAD: N=NOMINAL; S=AUMENTO; D=DISMINUCION "
1410 GET L$:IF L$="" THEN 1410
1420 IF L$="N" THEN 1450
1430 IF L$="S" THEN 1570
1440 IF L$="D" THEN 1890
1450 GOTO 1410
1460 PRINT " LA ARTICULACION COMENZARA A MOVERSE, CUANDO LLEGUE A LA "
1470 PRINT " POSICION DESEADA PRECIONE EL BOTON ROJO"
1480 POKE PA,A+80:POKE PB,V
1490 K=PEEK(56320):IF K<111 THEN CO=CO+1:GOTO 1490
1500 POKE PA,Z:POKE PB,0
1510 PRINT " ES CORRECTO EL MOVIMIENTO (S/N)":GET RE$:IF RE$="" THEN 1510
1520 IF RE$="S" THEN GOTO 1550
1530 IF RE$="N" THEN CO=0:GOTO 1290
1540 GOTO 1510
1550 PRINT#3,H$:,";":I$;",";J$;",";V$;",";CO
1560 CO=0:GOTO 7000
1570 REM Rutina de aumento de velocidad
1580 K=PEEK(56320):IF K<126 AND K<111 THEN 1580
1585 IF K=111 THEN 1620
1590 AV=AV+1:FOR RE=1 TO 50:NEXT
1600 POKE PA,A+80:POKE PB,V+AV:GOTO 1580
1620 PRINT " REGRESAR LA ARTICULACION A SU POSICION INICIAL, DESPUES PRECIONE "
1630 PRINT " EL BOTON ROJO"
1640 K=PEEK(56320):IF K<125 THEN 1640
1650 IF K=111 THEN 1670
1660 POKE PA,A+(NOT A):POKE PB,V+AV:GOTO 1640
1670 PRINT " PRECIONE CUALQUIER TECLA PARA POSICIONAR LA ARTICULACION CON "
1685 PRINT " LA VELOCIDAD ESCOGIDA"
1690 GET L$:IF L$="" THEN 1690
1700 PRINT " NUEVA LA PALANCA HACIA ADELANTE HASTA LLEGAR A LA POSICION "
1710 PRINT " DESEADA"
1720 K=PEEK(56320):IF K<126 OR K<111 THEN 1700
1730 IF K=111 THEN DV=V+AV:GOTO 1730
1740 POKE PA,A+80:POKE PB,V+AV:CO=CO+1:GOTO 1700
1750 PRINT#3,H$:CO$;I$:CO$:J$:CO$:V$:CO$:CO
1760 CO=0:GOTO 7000
1800 REM Rutina de disminucion de velocidad
1810 K=PEEK(56320):IF K<125 OR K<111 THEN 1810
1820 IF K=111 THEN 1850
1830 AV=AV+1:FOR RE=1 TO 50:NEXT
1840 POKE PA,A+80:POKE PB,V-AV:GOTO 1810
1850 PRINT " REGRESAR LA ARTICULACION A SU POSICION ORIGINAL MOVIENDO LA "
1865 PRINT " PALANCA HACIA "
1880 PRINT " ATRAS,HASTA LLEGAR A LA POSICION INICIAL,DESPUES PRECIONE EL "
1895 PRINT " BOTON ROJO"
1900 K=PEEK(56320):IF K<126 OR K<111 THEN 1870
1910 IF K=111 THEN 1910
1920 POKE PA,A+(NOT A):POKE PB,V-AV:GOTO 1870
1930 PRINT " PRECIONE CUALQUIER TECLA PARA POSICIONAR LA ARTICULACION CON "
1945 PRINT " LA VELOCIDAD ESCOGIDA"
1950 GET L$:IF L$="" THEN 1950
1960 PRINT " NUEVA LA PALANCA HACIA ATRAS HASTA LLEGAR A LA POSICION "
1975 PRINT " DESEADA"

```

```
1935 PRINT "*****SE INICIA EL EJERCICIO DESEARDA"
1940 L=PEEK(56300):IF K=125 OR K=111 THEN 1940
1950 IF K=111 THEN DV=V-AV:GOTO 1970
1960 POKE PA:A+AV:POKE PB:V-AV:CO=CO+1:GOTO 1940
2000 REN LEER Y EJECUTAR DATOS ALMCEENADOS
2005 DIM C$(20),A$(20),G$(20),V$(20),CO(20)
2010 OPEN 4:R+:"00:DATOS.S:R"
2020 PRINT " TIENES LA CLAVE DEL MOVIMIENTO (S/N)"
2030 GET R$:IF R$="" THEN 2030
2040 IF R$="S" THEN 2070
2050 IF R$="N" THEN 2550
2060 GOTO 2030
2070 INPUT " TECLER LA CLAVE DEL MOVIMIENTO";MO#
2080 INPUT#4,C$(L),A$(L),G$(L),V$(L),CO(L):REN:LECTURA DE DATOS DE ARCHIVO
2090 IF ST=0 THEN CLOSE 4:PRINT " ESE MOVIMIENTO NO ESTA ALMACENADO":GOTO 3010
3100 IF C$=MO# THEN PRINT " CLAVE ENCONTRADA ":GOTO 3120
3110 GOTO 2030
3120 PRINT " EJECUTARLO(E); BUSCAR OTRO(O); SALIDA(S)"
3130 GET L$:IF L$="" THEN 3130
3140 IF L$="E" THEN 3120
3150 IF L$="O" THEN CLOSE 4:GOTO 3010
3160 IF L$="S" THEN CLOSE 4:GOTO 5
3170 GOTO 3130
3180 L=1
3190 C$(L)=C$:A$(L)=A$:G$(L)=G$:V$(L)=V$:CO(L)=CO
3200 L=L+1
3210 INPUT#4,C$(L),A$(L),G$(L),V$(L),CO(L)
3220 IF ST=0 THEN CLOSE 4:PRINT " EJERCICIO TERMINADO":GOTO 3010
3230 IF C$(L)=MO# THEN GOTO 3250
3240 GOTO 3200
3250 PRINT " CUANDO SE QUIERA EJECUTAR PRECIONE LA BARRA ESPACIADORA"
3260 PRINT " SI DURANTE SU EJECUCION SE REQUIERE PARAR DE INMEDIATO"
3270 PRINT " PRECIONE LA TECLA (S)"
3280 GET M$:IF M$="" THEN 3280
3290 IF M$=CHR$(169) THEN 3310
3300 GOTO 3280
3305 FOR N=1 TO L-1
3310 GET M$:IF M$="" THEN 3340
3320 IF M$=CHR$(13) THEN GOTO 3310
3330 POKE PA:Z:POKE PE:0:CLOSE 4:GOTO 5
3340 IF A$(N)="A" THEN A=0:GOTO 3390
3350 IF A$(N)="B" THEN A=8:GOTO 3390
3360 IF A$(N)="C" THEN A=4:GOTO 3390
3370 IF A$(N)="D" THEN A=12:GOTO 3390
3380 PRINT " ERROR EN LA ARTICULACION":CLOSE 4:GOTO 5
3390 IF G$(N)="M" THEN AO=0:GOTO 3420
3400 IF G$(N)="H" THEN AO=1:GOTO 3420
3410 PRINT " ERROR EN GIRO":CLOSE 4:GOTO 5
```

READY.

```

3410 PRINT " ERROR EN GIRO":CLOSE 4:GOTO5
3420 FOR N1=1 TO CO(ND)
3430 POKE PA:A+AO:POKE PB:V(ND)
3440 NEXT N1
3450 NEXT N
3460 PRINT " TAREA CONCLUIDA"
3470 PRINT " A) EJECUTARLO NUEVAMENTE"
3474 PRINT " B) EJECUTAR OTRO MOVIMIENTO"
3476 PRINT " C) SALIDA"
3480 GET N2$:IF N2$="" THEN 3480
3490 IF N2$="A" THEN GOTO 3300
3500 IF N2$="B" THEN GOTO 3010
3510 IF N2$="C" THEN GOTO 5
3520 GOTO 3480
3550 REM NO TIENE CLAVE
3560 PRINT " MOSTRARE LAS CLAVES DE MOVIMIENTO           ALMACENADAS"
3570 INPUT#4,C1,A1,C2,A2,C3,V1,C3
3580 IF ST(0) THEN CLOSE 4:GOTO 3600
3590 PRINT C1:GOTO 3570
3600 PRINT " 2) EJECUTAR ALGUNA DE LAS MOSTRADAS:  X) SALIDA"
3610 GET N3$:IF N3$="" THEN 3610
3620 IF N3$="1" THEN 3010
3630 IF N3$="X" THEN 5
3640 GOTO 3610
4000 REM MOVIMIENTOS DE PRUEBA
4010 PRINT "2":PRINT:PRINT:PRINT
4020 PRINT " MOVIMIENTOS YA ALMACENADOS(S=SI; N=NO)"
4030 GET MM$:IF MM$="" THEN 4030
4040 IF MM$="S" THEN 3000
4050 IF MM$="N" THEN 150
4060 GOTO 4030
4100 PRINT " PRECIONE LA BARRA ESPACIADORA CUANDO     DESEE 5ARRAS"
4110 GET RR$:IF RR$="" THEN POKE PA:AO:POKE PB:V:GOTO 4110
4120 IF RR$="CHR$(32) THEN POKE PA:Z:GOTO 5
4130 GOTO 4110
5000 REM RUTINA DE DESIGNACION DE GIRO(TECLAS)
5010 PRINT "2":PRINT:PRINT:PRINT
5020 PRINT "MOVIMIENTO:"
5025 PRINT:PRINT:PRINT " A) A FAVOR DE LAS MANECILLAS DEL RELOJ"
5030 PRINT " B) EN CONTRA DE LAS MANECILLAS DEL           RELOJ"
5040 PRINT " C) SALIDA A MENU PRINCIPAL":PRINT:PRINT
5050 PRINT " ESCOJA OPCION (A,B O C)"
5060 GET Y$:IF Y$="" THEN 5060
5070 IF Y$="A" THEN AO=0:GOTO5110
5080 IF Y$="B" THEN AO=1:GOTO5110
5090 IF Y$="C" THEN GOTO 10
5100 GOTO 5060
5110 RETURN

```

READY.

Desplegado 4.7f

```

5110 RETURN
6000 REM RUTINA DE CONTROL DE VELOCIDAD(TECLAS)
6010 PRINT"O":PRINT:PRINT:PRINT
6020 PRINT " LA PLATAFORMA TIENE UNA VELOCIDAD NOMINAL, PERO TAMBIEN SE PUEDE"
6030 PRINT "REGULAR ESTA VELOCIDAD SEGUN SE ESCOJA EL SIGUIENTE"
6040 PRINT "ENTE MENU:"
6050 PRINT:PRINT:PRINT TAB(15);"M E N U"
6060 PRINT:PRINT " A) VELOCIDAD NOMINAL"
6070 PRINT:PRINT " B) ELEVAR VELOCIDAD"
6080 PRINT:PRINT " C) DISMINUIR VELOCIDAD"
6090 PRINT:PRINT " D) MENU PRINCIPAL"
6100 PRINT:PRINT " E) SALIDA "
6110 PRINT:PRINT:PRINT " OPCION ?"
6120 GET V$:IF V$="" THEN GOTO 6110
6130 IF V$="A" THEN V=128:GOSUB 5000:GOTO 4100
6140 IF V$="B" THEN GOTO 6190
6150 IF V$="C" THEN GOTO 6620
6160 IF V$="D" THEN GOTO 5
6170 IF V$="E" THEN END
6180 GOTO 6110
6190 REM ELEVAR VELOCIDAD
6200 L1=1
6210 PRINT"O":PRINT:PRINT:PRINT
6220 PRINT " CUANDO SE DESEE ELEVAR LA VELOCIDAD, PRESIONE LA TECLA"
6230 PRINT "MARCADA CON UNA (E); LA VELOCIDAD DE LA"
6240 PRINT "ARTICULACION CORMAN AUMENTARA."
6250 PRINT " CUANDO LA ARTICULACION OBTenga LA VELOCIDAD DESEADA,"
6260 PRINT " PRESIONE LA TECLA MARCADA CON UNA (S);"
6270 PRINT " CONENZARA A MOVERSE LA ARTICULACION Y PODRA REGULAR SU VELOCIDAD"
6280 PRINT "DESPUES, PODRA MOVER LA ARTICULACION A ESA"
6290 PRINT "VELOCIDAD."
6300 PRINT " CUANDO ESTE LISTO PRESIONE LA BARRA ESPACIADORA"
6310 GET V$: IF V$="" THEN GOTO 6270
6320 IF V$(0)="$E" THEN GOTO 6270
6330 GET O$: IF O$="" THEN GOTO 6290
6340 IF O$="E" THEN GOTO 6290
6350 GET O$: IF O$="" THEN GOTO 6340
6360 IF O$="S" THEN GOTO 6360
6370 GOTO 6310
6380 IF V/354 THEN PRINT "VELOCIDAD MAXIMA TOPE":GOTO 6360
6390 I=I+1 :REM CONTADOR DE AUMENTO DE VELOCIDAD
6400 FOR R=1 TO 500:NEXT R:REM CICLO DE RETARDO
6410 V=V+1
6420 POKE PA,S:POKE PB,V:GOTO 6310:REM PONER DATOS EN PUERTO
6430 PRINT " TAREA CONCLUIDA"

```

READY.


```

6390 PRINT " TAREA CONCLUIDA"
6390 PRINT " *1) SEGUIR HOVIENDO ESTA ARTICULACION"
6400 PRINT "     CON LA VELOCIDAD ESCOGIDA"
6410 PRINT " * 2) REGULAR DE NUEVO SU VELOCIDAD"
6420 PRINT " * 3) MENU PRINCIPAL"
6430 PRINT " * 4) SALIDA "
6440 PRINT:PRINT:PRINT
6450 PRINT " OPCION ?"
6460 GET M$: IF M$="" THEN 6460
6470 IF M$="1" THEN GOTO 6510
6480 IF M$="2" THEN V=128:I=0:ON L1 GOTO 6190,6620
6490 IF M$="3" THEN GOTO 5
6495 IF M$="4" THEN END
6500 GOTO 6450
6510 PRINT "C":PRINT:PRINT
6520 PRINT " PARA CAMBIAR DE GIRO PRECIONE LA TECLA (C)"
6525 PRINT " PARA IR AL MENU PRINCIPAL PRECIONE LA TECLA (M)"
6530 PRINT " PARA PARAR PRECIONE LA TECLA (P)"
6540 PRINT " ESPERO OPCION "
6550 GET L$: IF L$="" THEN POKE PA,S:POKE PB,V:GOTO 6550
6560 IF L$="C" THEN 6600
6570 IF L$="M" THEN 5
6580 IF L$="P" THEN 6300
6590 GOTO 6550
6600 IF A0=0 THEN POKE PA,S+1:POKE PB,V:GOTO 6550
6610 POKE PA,S-1:POKE PB,V:GOTO 6550
6620 PRINT "I":PRINT:PRINT
6625 L1=2
6630 PRINT " CUANDO DESEES DISMINUIR LA VELOCIDAD, PRECIONE LA TECLA "
6640 PRINT " *****"
6650 PRINT " LA VELOCIDAD DE LA ARTICULACION CO- MENCERA A DISMINUIR"
6660 PRINT " CUANDO LLEGUE A LA VELOCIDAD DESEADA PRECIONE (S)"
6670 PRINT " CUANDO QUIERA COMENEAR PRECIONE LA PARA ESPACIADORA"
6680 GET R$: IF R$="" THEN 6680
6690 IF R$<"C" THEN 6690
6700 GET S$: IF S$="" THEN 6700
6710 IF S$<"D" THEN 6700
6720 J=J+1
6725 FOR I=1 TO 500: NEXT
6730 IF V<2 THEN PRINT "VELOCIDAD TOPE MINIMA":FORM#1701000:NEXT:GOTO 5
6735 V=V-J
6740 GET P$: IF P$="" THEN POKE PA,S:POKE PB,V:GOTO 6720
6750 IF P$<"S" THEN 6720
6755 GOTO 6300
7000 PRINT " EXISTEN MAS MOVIMIENTOS (S/N)"
7010 GET J$: IF J$="" THEN 7010
7020 IF J$="S" THEN 7050
7030 IF J$="N" THEN CLOSE 3:GOTO 5
7040 GOTO 7010
7050 PRINT " SON DE OTRO EJERCICIO (O):O SON DEL MISMO EJERCICIO (M)"
7060 GET J$: IF J$="" THEN 7060
7070 IF J$="O" THEN GOTO 1280
7080 IF J$="M" THEN GOTO 1290
7090 GOTO 7060
7999 REM Rutina de ESCOGENCIA DE MOTOR

```

```

7999  REM  RUTINA DE ESCOGENCIA DE MOTOR
8000  A=PEEK(50320):REM LEE DATOS DE PALANCA
8010  IF A=125 THEN POKE PA,8:GOTO 8000
8020  IF A=125 THEN POKE PA,8:GOTO 8000
8030  IF A=123 THEN POKE PA,4:GOTO 8000
8040  IF A=119 THEN POKE PA,12:GOTO 8000
8050  IF A=117 THEN POKE PA,2:RETURN:REM NO OPERA
8060  GOTO 8000

```

51

```

8100  A=PEEK(50320):REM LEE DATOS DE PALANCA
8110  IF A=107 THEN POKE PA,(PEEK(PA))+1:GOTO 8100
8120  IF A=103 THEN POKE PA,(PEEK(PA))+0:GOTO 8100
8130  IF A=110 THEN J=J+1:POKE PB,V+J:GOTO 8100
8140  IF A=109 THEN I=I+1:POKE PB,V-1:GOTO 8100
8150  IF A=101 THEN POKE PA,2: RETURN
8160  GOTO 8100

```

```

15000  REM  DESPLEGADO DE INSTRUCCIONES
15010  PRINT "D"

```

```

15020  PRINT:PRINT:PRINT

```

```

15030  PRINT " LA PLATAFORMA TIENE 4 ARTICULACIONES
15040  PRINT " LAS CUALES ESTAN ASIGNADAS (A,B,C,D)

```

```

MOVILES, "
COMO SE MUESTRA: "

```

```

15050  PRINT " B C D "

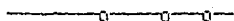
```

```

15060  PRINT

```

```

15070  PRINT "  "

```

```

15080  PRINT "  "

```

```

15090  PRINT " "

```

```

15095  PRINT " A "

```

```

15099  PRINT " "

```

```

15095  FOR H=1 TO 5000:NEXT H

```

```

15100  PRINT:PRINT:PRINT

```

```

15110  PRINT " LA ASIGNACION DE CUAL ARTICULACION VA
15120  PRINT " REALIZAR POR MEDIO DEL TECLADO O POR

```

```

A MOVER, LA PODRA "
MEDIO DE LA PALANCA "
CION ESCOGIO EN EL "

```

```

15120  PRINT " DE CONTROL; DEPENDIENDO DE CUAL OP
15140  PRINT " ***** SEGUIR MENU "

```

```

15150  PRINT:PRINT

```

```

15160  PRINT " PRECIONE LA BARRA ESPACIADORA PARA

```

```

CONTINUAR "

```

```

15170  GET Z$:IF Z$="" THEN 15170

```

```

15180  IF Z$="C" THEN PRINT " ESA NO ES LA BARRA ESPACIADORA " GOTO 15160

```

```

15190  PRINT:PRINT:PRINT

```

```

15200  PRINT " UNA VEZ ESCOGIDA LA ARTICULACION A MO-
15210  PRINT " TENER UNO DE DOS MOVIMIENTOS: EN SEN-

```

```

VER, ESTA PUEDE "
TIDO DEL MOVIMIENTO: DE "

```

```

15215  PRINT " ***** LAS MANECILLAS "

```

```

15220  PRINT " DEL RELOJ, O, EN CONTRA DEL MOVIMIENTO

```

```

DE LAS MANECILLAS DEL "
ESTA ASIGNACION LA PODRA "

```

```

15230  PRINT " ***** RELOJ, TAMBIEN
15235  PRINT " ***** REALIZAR DES-

```

```

DE EL TECLADO O POR MEDIO DE LA

```

```

15240  PRINT " ***** PALAN-

```

```

CA DE CONTROL DEPENDIENDO "

```

```

15250  PRINT " DE LA OPCION ESCOGIDA. "

```

```

15260  PRINT:PRINT:PRINT " PRECIONE LA BARRA ESPACIADORA PARA

```

```

CONTINUAR "

```

```

READY.

```

Desplegado 4.7i

```

15260 PRINT:PRINT:PRINT" PRECIONE LA BARRA ESPACIADORA PARA CONTINUAR"
15270 GET W$:IF W$="" THEN 15270
15280 IF W$=CHR$(32) THEN PRINT "ESA NO ES LA BARRA ESPACIADORA" GOTO 15260
15290 PRINT:PRINT:PRINT
15300 PRINT " CUANDO YA SE TIENE LA ARTICULACION Y EN GIRO DE ESTA "
15310 PRINT " ESCOGIDA, ES NECESARIO ESCOGER LA VE- LOCIDAD DE GIRO, ESTA "
15320 PRINT " ULTIMA NO SE ESPECIFICA, SE ASUME UNA VELOCIDAD MEDIA "
15330 PRINT " LA VELOCIDAD TAMBIEN ES ESCOGIDA POR MEDIO DEL TECLADO O POR "
15340 PRINT " LA PALANCA DE CONTROL DEPENDIENDO DE LA OPCION"
15350 PRINT " SELECCIONANDO LA OPCION ESCOGIDA"
15360 PRINT:PRINT:PRINT
15370 PRINT " DESEA VOLVER A VER LAS INSTRUCCIONES (S=SÍ; N=NO)"
15380 GET W$:IF W$="" THEN 15380
15390 IF W$="S" AND W$="N" THEN 15370
15400 IF W$="S" THEN 15010
15410 RETURN :REM REGRESO DE LA SUBROUTINA

```

READY.

C A P I T U L O V

EL SISTEMA COMPLETO .

EL SISTEMA COMPLETO

Se conectará la interfase a la computadora por medio de un conector a los puertos de salida de esta, y la interfase se conectará a los TR's que estarán montados en disparadores por medio de cable plañar. De los TR's se conectarán a los motores por medio de cable elicoidal, para darle flexibilidad. (Los TR's y sus disparadores estarán en la base de la plataforma).

Se realizará un ejemplo de como funcionará el sistema completo.

Supongamos que la plataforma está en su posición de descanso (horizontal tanto la mesa balancin como el conjunto articulado), y se requiere un ejercicio con los siguientes pasos :

- a) Posicionar la mesa balancin a 90° de su posición horizontal en el sentido de giro de las manecillas del reloj.
- b) Posicionar la articulación 1, 45° de su posición en ese momento (horizontal) en el sentido de giro contrario al de las manecillas del reloj.
- c) Posicionar la articulación 2, 30° de su posición en el sentido de giro de las manecillas del reloj.

Y se requiere almacenar estos movimientos en diskette.

NOTA : Proponemos este ejercicio de tres fases para ilustrar todo el proceso que sigue la máquina para realizarlo.

Lo que aparesca entre paréntesis son los números de línea del programa REHABILITACION que se están ejecutando en ese momento.

PASOS A SEGUIR

- 1) Encender la computadora COMMODORE 64, en switch de ensendido, también la unidad lectora-grabadora de diskettes y la fuente de alimentación de la plataforma (rectificador)
- 2) Meter en la unidad de disco el diskette que contiene el programa REHABILITACION y cargarlo en la memoria de la computadora con el comando :

LOAD"REHABILITACION",8

Una vez cargado, sacar el diskette de la unidad de disco y meter el diskette de almacenamiento de movimientos en la unidad de disco.

- 3) Correr el programa con el comando RUN.
 - 4) Enseguida aparecerá en la pantalla el desplegado 4.1, del cual se escogerá la opción No. 2) Almacenamiento de información, (30-90) enseguida se nos mostrará el desplegado 4.4 (1000-), de este desplegado se contestarán las preguntas pertinentes por medio del teclado.
- Al haber realizado el paso 4 se inicializan todas las variables del programa a sus valores nominales (5), se limpian los puertos de salida y se direccionan como tal -- (310). También se crea el archivo si es que no existe en el diskette de almacenamiento; si el archivo ya existe solamente se abre para introducir más datos (1000-1010).

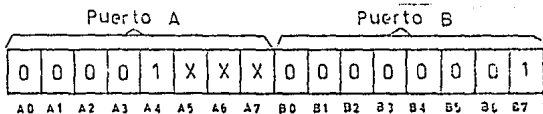
El archivo llamado DATOS es un archivo de forma secuencial.

- 5) Segudamente aparecerá en pantalla el desplegado 4.4 que nos indica las secuencias de pasos a seguir (1733-1743).

Ya sea presionando teclas del teclado o moviendo la palanca de control a una posición indicada. Por tanto, se escoge la articulación A, presionando la tecla con la letra "A" marcada.

Se designará el giro en sentido de las manecillas del reloj presionando la tecla "M", después se le asigna una velocidad nominal presionando la tecla "W".

Al haber realizado los pasos anteriores en los puertos de salida A y B se obtiene el siguiente estado:



Los "1" son 5V y los "0" son 0V, las "X" son estados de no importancia.

En combinación con el puerto B nos controla la velocidad del motor.

El estado de A3 nos controla el giro del motor.

El estado de A1, A2, y A3, nos determinan que el motor se escogerá.

El estado de A4, nos determina el momento en que comienza a contar el contador.

Por tanto en este caso A5 contiene un 0 y el giro será en sentido de las manecillas del reloj, A1 contiene un 0; A2 contiene un 0; A3 contiene un 0, por tanto el

motor seleccionado es 1; A4 contiene un 1; por tanto se inicia el conteo del contador.

El puerto B da el dato que se presentará en uno de los lados del comparador, mientras que en el otro se encontrará el número producido por el contador, el cuál proporcionará una salida alta, hasta que el número del contador sea mayor que el proporcionado por el puerto B de la computadora.

Lo que hará que una vez pasado el "1" lógico a través de las compuertas lógicas, puesto que en el puerto A existe la combinación 0001 el decodificador pondrá un cero en la línea de motor escogida, el cuál pasará por la compuerta C1, lo que nos proporcionará un "1" en su salida que estará a la entrada de la compuerta NAND con su otra entrada, conectada a la salida, del comparador, dando en la salida de la compuerta un "1" lógico que disparará el transistor y este a su vez el gatillo del TR, lo que cerrará el circuito de alimentación del motor No. 1, el cual hará que se mueva la articulación. Esta seguirá moviéndose hasta que la palanca de control se posicione en su punto muerto, lo que ocasionará que en el puerto A se ponga el valor numérico de 01000000, el cual será en el decodificador señal de no operación, con lo cual no se disparará ningún transistor y por tanto ningún TR.

Mientras la palanca de control este en su posición indicada, o la tecla "S" no se haya pulsado, existirá un contador que contará el tiempo en que el motor está funcionando (1490); este contador almacenará ese tiempo, para después reproducirlo cuando se lea ese archivo.

Cuando se haya terminado ese movimiento es necesario presionar el botón rojo, para indicarle a la máquina que continúa otro movimiento, esto ocasionará que este movi-

miento sea almacenado en el disco (1550), los datos guardados en disco son los siguientes : Nombre del ejercicio, articulación, giro, velocidad, tiempo. Al efectuar lo anterior, la mesa balancin se encuentra a 90° de su posición horizontal, en el sentido de giro de las manecillas del reloj.

- 6) Por tanto se prosigue con el siguiente movimiento, este movimiento es el movimiento de la articulación B.

Para decir que se va a mover la articulación B se presiona la tecla "B", en seguida es necesario dar el sentido de giro de la articulación, esto se logra presionando la tecla "N", al tener lo anterior, se les da valor a algunas variables como A1, A, A4, para que cuando se escoja la velocidad, sean puestos estos cinco datos en los puertos A y B.

Una vez seleccionada la velocidad, se ponen en el puerto A los datos de articulación, giro y comienzo de conteo y en el puerto B los datos de velocidad.

Los datos del puerto B estarán en una de las entradas del comparador, mientras que los datos del puerto A, seleccionan por el decodificador el motor.

El dato A0 selecciona el giro y el dato A4 da comienzo al conteo del contador. Sucederá lo anteriormente explicado pero ahora con el motor que controla la articulación B, hasta que llegue a su posición requerida entonces se graba en el diskette con los mismos datos anteriormente citados.

- 7) Exactamente el mismo procedimiento sucederá con la articulación C, hasta que llegue a su posición requerida.
- 8) Cuando se hayan terminado todos los movimientos que requiere el ejercicio, presionar el botón rojo, indicando así que se ha terminado el ejercicio, esto cerrará el ar

chivo DATOS (730), y se desplegará enseguida el menú de opciones generales, para así escoger cualquiera de sus opciones.

- 9) Si se requiere terminar, es necesario escoger del menú principal la opción : Salida, lo que terminará el programa y la computadora pasará a estar en modo inmediato.

Para volver a reproducir este ejercicio, se tendría que escoger la opción general de lectura y ejecución de datos que leería el archivo DATOS, sacando los datos del ejercicio y ejecutandolos en su misma secuencia.

Si en algún momento el paciente tiene alguna contracción muscular que le cause dolor, la persona puede inmediatamente suspender el ejercicio presionando la tecla "S" -- (3270-3330) y en ese momento, se da instrucción al decodificador que pase a un estado de no operación (3330), lo que hará que se detenga por completo la plataforma, para poder atender al paciente. Si se requiere comenzar de nuevo el ejercicio, la computadora después de detenerse nos mostrará el menú principal con el cual se podrá escoger una nueva opción.

C A P I T U L O VI

CONCLUSIONES.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA 59

CONCLUSIONES

En mi primera idea acerca de la plataforma y sus problemas de manipulación, tenía un concepto muy vago acerca de lo que se podría lograr, pero en el transcurso de este proyecto he podido constatar como su aplicación, sus ventajas, sus articulaciones, su control es realmente flexible, para añadir y mejorar lo que hasta ahora logré.

Algo que se le podría añadir al proyecto podría ser: - añadir un conjunto articulado más, para poder realizar con las dos piernas los ejercicios, sin necesidad de mover al paciente; o también, poder mover más de una articulación al mismo tiempo, dado que en este trabajo solamente puede moverse una articulación por vez, etc.

En este proyecto, el programa de control, es la parte modular del diseño, porque en él se coordinan y dirigen todas las demás acciones, se puede notar lo largo del programa, esto es debido al gran número de variantes que se pueden presentar. Creo que una de las tendencias modernas de la Electrónica de Control Programable, es tratar de realizar la mayoría sino es que todas las funciones de control por medio de programación (software) y dejar a la circuitería externa (hardware) únicamente la función de recibir o dar los datos de la computadora al sistema a controlar.

Como se puede notar en este proyecto, todos los tiempos,

secuencias, interrelaciones se llevan a cabo en el programa y única y exclusivamente la interfase da a la plataforma - los datos que recibe de la computadora. Esto se realizó de esta manera por la confiabilidad, la rapidez y el fácil acceso a los datos, que se puede tener por medio de la programación.

Creo además que este tipo de equipos en la ayuda de la rehabilitación, son muy necesarios para ayudar a que los pacientes que necesitan realizar este tipo de ejercicios para no llegar al punto de tener que sustituir parte de su anatomía por partes mecánicas (brazos, piernas de plástico, etc) por falta de una buena rehabilitación o por falta de tiempo. No estoy en contra de esta sustitución, pero si se puede evitar caminar con piernas de plástico ajenas al paciente, para caminar con sus propias piernas aunque esto lleve gran tiempo de ejercicios de rehabilitación y algo de dolor, creo que estos equipos serán de gran beneficio para estas personas.

BIBLIOGRAFIA

Carl T. Helmers. Robotics Age, In the Beginning. Hasbrouck Heights, New Jersey. EUA. Hayden Book, Company, Inc. 1981.

David M Auslander and Paul Sagues. Microprocessors for Measurement Control. Berkley, California. E U A. Osborne-Mc.Graw Hill. 1981.

Edward L. Safford, Jr. Handbook of advanced Robotics. Blue Ridge Summit, P.A. E U A. Tab Books Inc. 1982.

Howard A. Rusk. Medicina de Rehabilitacion. Centro Médico de la Universidad de New York. E U A. Interamericana. 1966.

Nacional Semiconductor Corporation., Logic Databook. Santa Clara, California. E U A. Volume II. 1984.

Nacional Semiconductor Corporation. Linear Databook. Santa Clara, California. E U A. 1983.

Nacional Semiconductor Corporation. Transistor Databook.
Santa Clara, California. E U A. 1983.

V.J. Geogiou. Commodore 54 Interfacing Blue Book. Goleta,
California. E U A. Microsignal Press. 1984.

A N E X O S


**National
Semiconductor**
Industrial Blocks

LM555/LM555C Timer

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free-running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and its output circuit can source or sink up to 200 mA or drive TTL circuits.

- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output

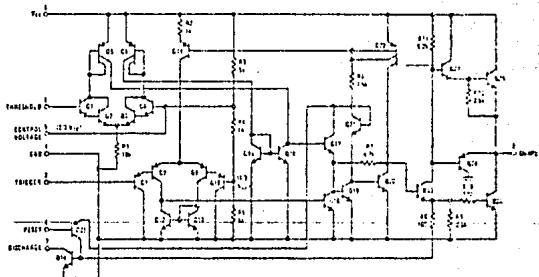
Applications

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

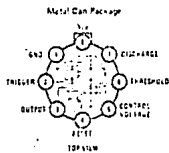
Features

- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes

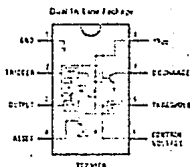
Schematic Diagram



Connection Diagrams



Order Number LM555H, LM555C1
See NS Package MODC



Order Number LM555N
See NS Package MODC
Order Number LM555C and LM555C1
See NS Package MODC

Applications Information

MONOSTABLE OPERATION

In this mode of operation, the timer functions as a one-shot (Figure 1). The external capacitor is initially held discharged by a transistor inside the timer. Upon application of a negative trigger pulse of less than $1/3 V_{CC}$ to pin 2, the flip-flop is set which prevents the short circuit across the capacitor and drives the output high.

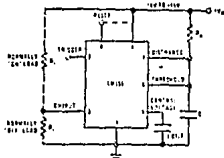


FIGURE 1. Monostable

The voltage across the capacitor then increases exponentially for a period of $t = 1.1 R_A C$, at the end of which time the voltage equals $2/3 V_{CC}$. The comparator then resets the flip-flop which in turn discharges the capacitor and drives the output to its low state. Figure 2 shows the waveforms generated in this mode of operation. Since the charge and the threshold level of the comparator are both directly proportional to supply voltage, the timing interval is independent of supply.

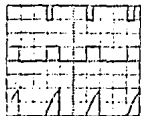


FIGURE 2. Monostable Waveforms

During the timing cycle when the output is high, the further application of a trigger pulse will not effect the circuit. However the circuit can be reset during this time by the application of a negative pulse to the reset terminal (pin 4). The output will then remain in the low state until a trigger pulse is again applied.

When the reset function is not in use, it is recommended that it be connected to V_{CC} to avoid any possibility of false triggering.

Figure 3 is a nomograph for easy determination of R, C values for various time delays.

(NOTE: In monostable operation, the trigger should be driven high before the end of timing cycle)

ASTABLE OPERATION

If the circuit is connected as shown in Figure 4 (pins 2 and 6 connected) it will trigger itself and free run as a

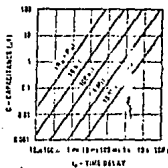


FIGURE 3. Time Delay

multivibrator. The external capacitor charges through $R_A + R_B$ and discharges through R_B . Thus the duty cycle may be precisely set by the ratio of these two resistors.

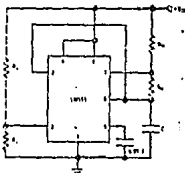


FIGURE 4. Astable

In this mode of operation, the capacitor charges and discharges between $1/3 V_{CC}$ and $2/3 V_{CC}$. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

Figure 5 shows the waveforms generated in this mode of operation.

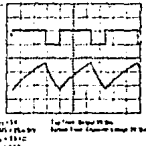


FIGURE 5. Astable Waveforms

The charge time (output high) is given by:

$$t_1 = 0.693 (R_A + R_B) C$$

And the discharge time (output low) by:

$$t_2 = 0.693 (R_B) C$$

Thus the total period is:

$$T = t_1 + t_2 = 0.693 (R_A + 2R_B) C$$

Applications Information (Continued)

The frequency of oscillation is:

$$f = \frac{1}{T} = \frac{1}{(R_A + 2R_B)C}$$

Figure 6 may be used for quick determination of these RC values.

The duty cycle is:

$$D = \frac{R_B}{R_A + 2R_B}$$

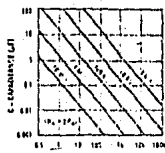


FIGURE 6. Free Running Frequency

FREQUENCY DIVIDER

The monostable circuit of Figure 1 can be used as a frequency divider by adjusting the length of the timing cycle. Figure 7 shows the waveforms generated in a divide by three circuit.

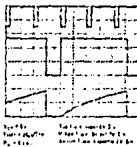


FIGURE 7. Frequency Divider

PULSE WIDTH MODULATOR

When the timer is connected in the monostable mode and triggered with a continuous pulse train, the output pulse width can be modulated by a signal applied to pin 5. Figure 8 shows the circuit, and in Figure 9 are some waveform examples.

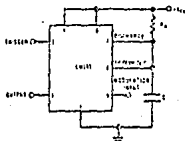


FIGURE 8. Pulse Width Modulator

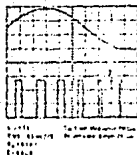


FIGURE 9. Pulse Width Modulator

PULSE POSITION MODULATOR

This application uses the timer connected for astable operation, as in Figure 10, with a modulating signal again applied to the control voltage terminal. The pulse position varies with the modulating signal, since the threshold voltage and hence the time delay is varied. Figure 11 shows the waveform generated for a triangle wave modulation signal.

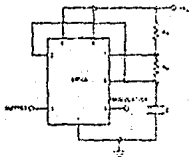


FIGURE 10. Pulse Position Modulator

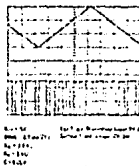


FIGURE 11. Pulse Position Modulator

LINEAR RAMP

When the pullup resistor, R_B , in the monostable circuit is replaced by a constant current source, a linear ramp is



DM54LS85/DM74LS85 4-Bit Magnitude Comparators

General Description

These four-bit magnitude comparators perform comparison of straight binary or BCD codes. They fully decoded outputs about two, 4-bit words (A, B) are made and are externally available at three outputs. These devices are fully expandable to any number of bits without external gates. Words of greater length may be compared by connecting comparators in cascade. The A>B, A<B, and A=B outputs of a stage handling less-significant bits are connected to the corresponding inputs of the next stage handling more-significant bits. The stage handling the most-significant bits must have a high-level voltage applied to the A=B input. The cascading path is implemented with only a two-gate level delay to reduce overall comparison times for long words.

Features

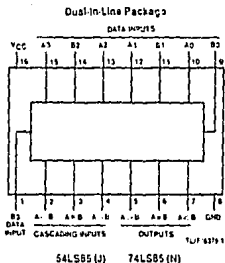
- Typical power dissipation 52 mW
- Typical delay (4-bit words) 24 ns

Absolute Maximum Ratings (Note 1)

Supply Voltage	TV
Input Voltage	TV
Storage Temperature Range	-65°C to 125°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the reliability of the device cannot be guaranteed. The device should not be operated at these limits; the characteristic values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table defines the conditions for actual device operation.

Connection Diagram



Function Table

Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	A>B	A<B	A=B	A>B	A<B	A=B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	H	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 < B3	A2 < B2	X	X	X	X	X	L	H	L
A3 > B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 < B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 < B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 > B3	A2 < B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 < B3	A2 < B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 > B3	A2 = B2	A1 < B1	A0 = B0	L	L	H	L	L	H
A3 < B3	A2 = B2	A1 < B1	A0 = B0	X	X	H	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 > B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

H = High Level, L = Low Level, X = Don't Care



DM54LS160A/DM74LS160A, DM54LS161A/DM74LS161A, DM54LS162A/DM74LS162A, DM54LS163A/DM74LS163A Synchronous 4-Bit Counters

General Description

These synchronous, presetable counters feature an internal carry look-ahead for applications in high speed counting designs. The LS160A and LS162A are decade counters, and the LS161A and LS163A are 4-bit binary counters. The carry output is created by means of a NOR gate, thus preventing spikes during the normal counting mode of operation. Synchronous operation is provided by having all flip-flops clocked via a common clock. This mode of operation eliminates the output counting spikes which are normally associated with asynchronous (ripple clock) counters. A buffered clock input bypasses the four flip-flops on the rising (positive-going) edge of the clock input waveform.

These counters are fully programmable, that is, the outputs may be preset to either level. As presetting is synchronous, setting up a few extra bit load inputs defines the counter and causes the outputs to agree with the set-up data after the next clock pulse, regardless of the levels of the enable inputs. Low to high transitions at the load input of the LS160A through LS163A are perfectly acceptable, regardless of the bit levels on the clock or enable inputs. The clear function for the LS160A and LS161A is asynchronous, and a low level at the clear input sets all four of the flip-flop outputs low, regardless of the levels of clock, load, or enable inputs. The clear function for the LS162A and LS163A is synchronous, and a low level at the clear input sets all four of the flip-flop outputs low after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily, as does the maximum count displayed, can be accomplished with one external NAND gate. The gate output is connected to the clear input to synchronously clear the counter to all low outputs.

The carry look-ahead circuitry provides for cascading counters for non-bit synchronous applications without adding extra timing instrumental in accomplishing this function are two complementary inputs and a ripple carry output.

Both count enable inputs (P and T) and the ripple carry output are input tri-stated for low impedance. The ripple carry output is a high impedance output with a propagation delay that is greater than the high level portion of the Q_n output. This ripple carry output is a carry pulse suitable for use in synchronous cascaded stages. High to low level transitions at the enable Pin T inputs of the LS160A through LS163A may occur at any point of the edge level of the clock.

LS161A through LS163A feature a fully independent clock output. Changes made to control inputs (enable Pin T or LS4) that will modify the operating modes have no effect on the clock output. The function of the enable inputs, when enabled, disabled, holding, or counting will be dictated solely by the conditions meeting the enable setup and hold times.

Features

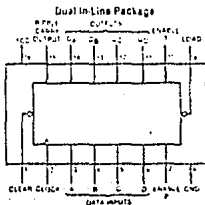
- Synchronously programmable
- Internal look-ahead for fast counting
- Carry output for bit cascading
- Synchronous counting
- Low cost design
- Data-stamped inputs
- Typical propagation time, max. clock to Q output 14 ns
- Typical clock frequency 20 MHz
- Typical power dissipation 100 mW

Absolute Maximum Ratings (Note 1)

Supply Voltage	V _{CC}
Input Voltage	5.5V
Storage Temperature Range	-55°C to 125°C

Note 1: The Absolute Maximum Ratings are those values beyond which the reliability of the device cannot be guaranteed. The device should not be operated at these limits. The manufacturer assumes no responsibility for any consequences arising from the use of the device in excess of the recommended operating conditions. See the end of the data sheet for actual device ordering.

Connection Diagram



54LS160A (2)	74LS160A (16)
54LS161A (2)	74LS161A (16)
54LS162A (2)	74LS162A (16)
54LS163A (2)	74LS163A (16)



National
Semiconductor

DM54LS138/DM74LS138, DM54LS139/DM74LS139 Decoders/Demultiplexers

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory decoding and data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress ringing and simplify system design.

Features

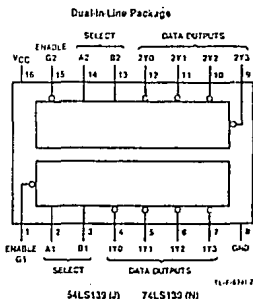
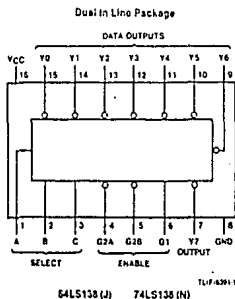
- Designed specifically for high speed:
 - Memory decoders
 - Data transmission systems
- LS138 3-to-8-line decoder incorporates 3 enable inputs to simplify cascading and/or data reception
- LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
 - LS138 21 ns
 - LS139 21 ns
- Typical power dissipation
 - LS138 33 mW
 - LS139 34 mW

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	7V
Storage Temperature Range	-65°C to 150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. These values should not be exceeded at these limits. The maximum values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Connection Diagrams



LS139 Switching Characteristics at $V_{CC} = 5V$ and $T_A = 25^\circ C$
(See Section 1 for Test Waveforms and Output Load)

Parameter	From (Input) To (Output)	$R_L = 210$						Units
		$C_L = 15\text{ pF}$			$C_L = 45\text{ pF}$			
		Min	Typ	Max	Min	Typ	Max	
t_{PLH} Propagation Delay Time Low to High Level Output	Select to Output		13	18		16	27	ns
t_{PHL} Propagation Delay Time High to Low Level Output	Select to Output		17	27		23	40	ns
t_{PLH} Propagation Delay Time Low to High Level Output	Enable to Output		13	18		16	27	ns
t_{PHL} Propagation Delay Time High to Low Level Output	Enable to Output		18	24		22	40	ns

Function Tables

LS138

Inputs				Outputs								
Enable	Select											
G1	G2	C	D	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	H	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	L	H	H	H	H
H	L	H	L	H	H	H	H	L	H	H	H	H
H	L	H	H	L	H	H	H	H	L	H	H	H
H	L	H	H	H	L	H	H	H	H	L	H	H
H	L	H	H	H	H	L	H	H	H	H	L	H
H	L	H	H	H	H	H	L	H	H	H	H	L
H	L	H	H	H	H	H	H	L	H	H	H	H

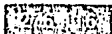
LS139

Inputs				Outputs			
Enable	Select						
G	B	A		Y0	Y1	Y2	Y3
H	X	X	X	H	H	H	H
L	L	L	L	L	H	H	H
L	L	L	H	L	H	H	H
L	L	H	L	L	H	H	H
L	L	H	H	L	L	H	H
L	H	L	L	L	L	H	H
L	H	L	H	L	L	L	H
L	H	H	L	L	L	L	H
L	H	H	H	L	L	L	L

H = HIGH level, L = LOW level, X = don't care

*G2 = G2A + G2B

H = HIGH level, L = LOW level, X = don't care



CADMIUM SULPHIDE PHOTOCELL

GENERAL DESCRIPTION

A cadmium sulphide photo cell is a light variable resistor which is most sensitive in the green to yellow portion of the light spectrum. With it you can use light to control many electronic devices. Max. resistance .5 meg., min. resistance 100 ohms, max. voltage 170 V, max. wattage .2 watts, rugged epoxy case.

APPLICATIONS

- Night light
- Light control
- Burglar alarm
- Relay

SPECIFICATIONS

- Shape..... Round
- Sensitive Area..... .07 sq. in.
- Weight..... 1.59 gms.
- Resistance at 1 Fc (28°C/K)..... 1.7k Ohms . 49%
- Typical Resistance 100 Fc (28°C/K)..... 100 Ohms
- Resistance Dark Minimum (1 Minute)..... 0.5 Megohms

ABSOLUTE MAXIMUM RATINGS

- Max. Applied Voltage (ac or dc)..... 170 V peak
- Max. Power Dissipation at 25°C..... 2 watts
- Power Derating..... Linearly to 0 @ 75°C
- Operating Temp. Range..... -40 to +75°C

CONNECTIONS

BOTTOM VIEW





QUAD OP AMP

GENERAL DESCRIPTION

The 324 series consists of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc-a.c. blocks and all the conventional op-amp circuits which now can be more easily implemented in single power supply systems. For example, the 324 series can be directly operated off of the standard $\pm 5 V_{DC}$ power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional $\pm 15 V_{DC}$ power supplies.

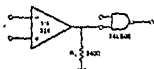
FEATURES

- Internally frequency compensated for unity gain
- Large dc voltage gain 102 dB
- Wide bandwidth (unity gain) 1 MHz (temperature compensated)
- Wide power supply range:
 - Single supply: $3 V_{DC}$ to $30 V_{DC}$
 - or dual supplies: $\pm 1.5 V_{DC}$ to $\pm 15 V_{DC}$
- Very low supply current drain (800 μA)—essentially independent of supply voltage (1 mW/op amp at $\pm 5 V_{DC}$)
- Low input biasing current 45 nA_{DC} (temperature compensated)
- Low input offset voltage 2 mV_{DC} and offset current 5 nA_{DC}
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing 0 V_{DC} to $V^+ - 1.5 V_{DC}$

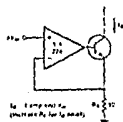
ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V^+	$32 V_{DC}$ or $\pm 16 V_{DC}$
Differential Input Voltage	$32 V_{DC}$
Input Voltage	$-0.3 V_{DC}$ to $+32 V_{DC}$
Power Dissipation	
Molded DIP	570 mW
Cavity DIP	900 mW
Output Short-Circuit to GND (One Amplifier)	Continuous
$V^+ < 15 V_{DC}$ and $T_A = 25^\circ C$	
Input Current ($V_{IN} < -0.3 V_{DC}$)	53 mA
Operating Temperature Range	0 to $+70^\circ C$
Storage Temperature Range	-65 to $+150^\circ C$
Lead Temperature (Soldering, 10 seconds)	$200^\circ C$

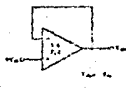
TYPICAL APPLICATIONS



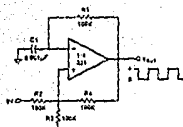
Driving TTL



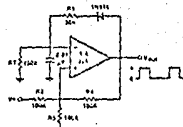
High Compliance Current Sink



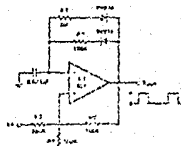
Voltage Follower



Squarewave Oscillator

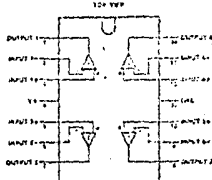


Pulse Generator



Pulse Generator

PIN CONNECTION



INTERNAL CIRCUIT (Each Amplifier)

