



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**Control de un motor de pasos  
utilizando un micro y un dsp**

**TESIS**

Que para obtener el título de  
**Ingeniero Eléctrico Electrónico**

**P R E S E N T A N**

Reyes Taboada Jiménez

Abraham Peña Del Arco

**DIRECTOR DE TESIS**

Ing. Joaquín Jorge Téllez Lendech



Ciudad Universitaria, Cd. Mx., 2016



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 esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

## AGRADECIMIENTOS

*A mis padres*

*Porque gracias a su cariño, apoyo, esfuerzos, sacrificios y confianza he llegado a realizar la culminación de mi carrera profesional y con ello, hacerlos sentirse orgullosos de esta persona que tanto los ama. Sabiendo que jamás encontraré la forma de agradecer su constante apoyo y confianza, sólo espero que comprendan que mis esfuerzos y logros han sido también suyos.*

*A mis hermanos*

*Porque aún sin saberlo me brindaron apoyo moral y fuerza para llevar a buen término este proyecto. Por su apoyo incondicional quiero compartir este momento con ustedes.*

*Ana Siri*

*Agradezco tu apoyo, comprensión y confianza esperando que comprendas que mis logros también son tuyos e inspirados por tí, considero este logro un triunfo y quiero compartirlo contigo.*

*A mis Profesores y Amigos*

*A todos aquellos con quien he compartido pequeños momentos de mi vida y que de alguna u otra manera fueron parte de mi formación profesional y personal, este es un pequeño reconocimiento con mucho cariño y respeto.*

**Reyes Taboada Jiménez**

*Gracias.*

**Abraham Peña Del Arco**

## Contenido

1. INTRODUCCIÓN .....	1
1.1 Tipos de sistemas de control.....	1
1.2 Controlador. ....	2
1.3 Justificación del uso del control de lazo abierto aplicado en este proyecto de tesis. ....	9
1.4 Clasificación de los motores eléctricos. ....	10
2. MOTORES A PASOS .....	20
2.1 Introducción. ....	20
2.2 Tipos básicos de motores de pasos de acuerdo a su principio de operación. ....	21
2.2.1 Reluctancia variable (VR).....	21
2.2.2 Imán Permanente (PM). ....	22
2.2.3 Híbrido.....	23
2.3 Tipos de motores paso a paso de imán permanente.....	24
2.3.1 Motor unipolar. ....	24
2.3.2 Motor bipolar. ....	25
2.4 Secuencias de manejo en motores a pasos. ....	26
2.4.1 Secuencias para manejar motores paso a paso unipolares. ....	26
2.4.2 Secuencias para manejar motores a paso bipolares.....	27
2.5 Parámetros de los motores a pasos. ....	27
2.6 Descripción del motor a pasos usado en este proyecto de tesis. ....	29
2.7 Ecuaciones electromagnéticas básicas para los motores a pasos genéricos. ....	30
3. CONTROL DE POSICIÓN Y VELOCIDAD EN UN MOTOR A PASOS .....	32
3.1 Ecuaciones mecánicas fundamentales de los motores a pasos.....	33
3.2 Diagramas de flujo para el control de posición.....	35
3.3 Diagramas de flujo para el control de velocidad.....	37

4. MICROCONTROLADOR ( $\mu$ C). TARJETA MINICON_11 (68HC11 DE MOTOROLA).....	39
4.1 Microcontrolador MC68HC11E1. ....	40
4.2 Registros empleados del MC68HC11. ....	46
4.3 La tarjeta MINICON_11B .....	51
5. PROCESADOR DIGITAL DE SEÑALES (DSP). TARJETA TMS320C50 DE TEXAS INSTRUMENTS.....	60
5.1 Características principales y ventajas del DSP TMS320C50. ....	60
5.2 Arquitectura del DSP TMS320C50.....	62
5.2.1 CALU. ....	63
5.2.2 Memoria. ....	64
5.2.3 Registros internos. ....	65
5.2.4 ARAU. ....	67
5.2.5 Registros de estado. ....	68
5.2.6.- Modos de direccionamiento .....	69
6. CONFORMACIÓN SISTEMA $\mu$ C – ETAPA DE POTENCIA – MOTOR DE PASOS.....	72
7. CONFORMACIÓN SISTEMA DSP – ETAPA DE POTENCIA – MOTOR DE PASOS.....	78
7.1 L298 Puente en H para un motor de dos fases. ....	78
7.2 Puertos paralelos de entrada y salida. ....	79
8. PRUEBAS DE POSICIÓN Y VELOCIDAD DE CADA UNO DE LOS SISTEMAS.....	83
8.1 Desarrollo de pruebas de posición y velocidad para el DSP C50 y el Microcontrolador MC68HC11. ....	83
8.1.1 Equipo requerido.....	83
8.1.2 Procedimiento. ....	83
8.2 Cálculos y mediciones para el DSP C50. ....	87
8.2.1 Tablas de resultados para el C50.....	88
8.3 Cálculos y mediciones para el HC11.....	90

8.3.1 Tablas de resultados para el HC11. ....	91
8.4 Código ASM para control de posición en el HC11. ....	92
8.5 Código ASM para el control de velocidad en el HC11. ....	94
8.6 Código ASM para el control de posición en el TMS320C50. ....	96
8.7 Código ASM para el control de velocidad en el TMS320C50. ....	98
9. CONCLUSIONES .....	102
BIBLIOGRAFÍA .....	103
MESOGRAFÍA .....	104
APÉNDICE A .....	106
APÉNDICE B .....	110

# 1. INTRODUCCIÓN

El objetivo de cualquier estrategia de control es mantener una variable llamada controlada, próxima a un valor deseado conocido como punto de ajuste “set-point”. El término regularización es usado para describir la acción de control de agentes de perturbación del estado de equilibrio de la variable controlada. Un sistema de control, solamente puede llegar a la regulación, aplicando en oposición a las fuerzas perturbadoras llamadas cargas, correcciones equivalentes en una o más variables denominadas manipuladas. La variable controlada permanecerá estable en el proceso mientras se encuentre en estado estacionario. Este equilibrio puede ser alcanzado usualmente por distintos sistemas de control. Hay varias clasificaciones dentro de los sistemas de controladores.

- Atendiendo a su naturaleza son analógicos, digitales o mixtos.
- Atendiendo a su estructura (número de entradas y salidas) puede ser control clásico o control moderno.
- Atendiendo a su diseño pueden ser por lógica difusa, redes neuronales...

## 1.1 Tipos de sistemas de control.

### 1.1.1 Sistemas de Control Clásico.

- **Sistemas de Control de Lazo Abierto:** se denominan sistemas de control de lazo abierto cuando la salida no tiene efecto sobre la acción de control, es decir no se compara la salida con la entrada de referencia. Por lo tanto, para cada entrada de referencia corresponde una condición de operación fija. Así, la precisión del sistema depende de la calibración del controlador.
- **Sistema de Control de Lazo Cerrado:** se denomina sistema de control de lazo cerrado cuando frente a presencia de perturbaciones, tiende a reducir la diferencia entre la salida del sistema y el valor deseado o “set point”; realizando el control de forma automática.

### **1.1.2 Sistemas de Control Moderno.**

- **Control Adaptativo:** es un método en el cual la respuesta de un controlador varía automáticamente basado en los cambios de las condiciones dentro del proceso y puede emplearse en diversas aplicaciones, como en el control del pH.
- **Control Difuso:** este control utiliza la lógica difusa a través de conceptos de inteligencia artificial capaz de convertir una muestra de la señal real a números difusos, para tratarlos según las reglas de inferencia y las bases de datos determinados en las unidades de decisión, logrando estabilizar el sistema sin la necesidad de fijar un punto de referencia.
- **Redes Neuronales Artificiales:** están diseñadas para actuar como lo hace el cerebro humano conectando la red entre los elementos de la forma más sencilla para poder ser entrenados y realizar funciones complejas en diversos campos de aplicación.

## **1.2 Controlador.**

El controlador es una componente del sistema de control que detecta los desvíos existentes entre el valor medido por un sensor y el valor deseado o “set point”, programado por un operador; emitiendo una señal de corrección hacia el actuador.

Un controlador es un bloque electrónico encargado de controlar uno o más procesos.

Al principio los controladores estaban formados exclusivamente por componentes discretos, conforme la tecnología fue desarrollándose se emplearon procesadores rodeados de memorias, circuitos de entrada y salida. Actualmente los controladores integran todos los dispositivos mencionados en circuitos integrados que conocemos con el nombre de microcontroladores.

Los controladores son los instrumentos diseñados para detectar y corregir los errores producidos al comparar y computar el valor de referencia o “set point”, con el valor medido del parámetro más importante a controlar en un proceso.

La actuación puede ser de forma clásica de acuerdo al tamaño y tiempo de duración del error, así como la razón de cambio existente entre ambos o aplicando sistemas expertos a través de la lógica difusa y redes neuronales.

El Control Realimentado es la propiedad de una sistema de lazo cerrado que permite que la salida (o cualquier otra variable controlada del sistema) sea comparada con la entrada al sistema (o con una entrada de cualquier componente interno del mismo o con un

subsistema) de manera tal que se pueda establecer una acción de control apropiada como función de la diferencia entre la entrada y la salida.

Más generalmente se dice que existe realimentación en un sistema cuando existe una secuencia cerrada de relaciones de causa y efecto entre las variables del sistema. Los controladores pueden ser del tipo: manual, eléctrico, electrónico, neumático ó digitales; así como las computadoras con tarjetas de adquisición de datos y los PLC (Controladores Lógicos Programables).

Actualmente en la industria se utiliza para controlar las variables de operación; sensores inteligentes, controladores lógicos programables (PLC), supervisando y adquiriendo los datos a través de las computadoras personales e integrándolas por una red y logrando un sistema de control distribuido (SOC).

### **1.2.1 Acciones de control de un regulador o controlador.**

El controlador o regulador constituye el elemento fundamental en un sistema de control, pues determina el comportamiento del bucle, ya que condiciona la acción del elemento actuador en función del error obtenido. La forma en que el regulador genera la señal de control se denomina acción de control. Algunas de estas acciones se conocen como acciones básicas de control, mientras que otras se pueden presentar como combinaciones de las acciones básicas.

#### **1. Acciones básicas:**

- Proporcional (P).
- Derivador (D).
- Integrador (I).

#### **2. Combinación de acciones básicas:**

- Proporcional - Integrador (PI).
- Proporcional - Derivador (PD).
- Proporcional – Integrador - Derivador (PID).

### 1.2.1.1 Controlador de acción Proporcional (P).

En este regulador la señal de accionamiento es proporcional a la señal de error del sistema. Si la señal de error es grande, el valor de la variable regulada es grande y si la señal de error del sistema es pequeña, el valor de la variable regulada es pequeño. Es el más simple de todos los tipos de control y consiste simplemente en amplificar la señal de error antes de aplicarla a la planta o proceso. La función de transferencia de este tipo de control se reduce a una variable real, denominada  $K_p$  que determinará el nivel de amplificación del elemento de control. (Ver figura 1.1).

Llamando  $y(t)$  a la señal de salida (salida del regulador) y  $e(t)$  a la señal de error (entrada al regulador), en un control proporcional tendremos que la función de transferencia del bloque controlador (no la total del sistema), es la mostrada en la siguiente ecuación.

$$Y(S) / E(S) = K_p \quad (1-1)$$

Donde:

- $Y(S)$  es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$  es la señal de error en el dominio de Laplace.
- $K_p$  es la ganancia del bloque de control.

Teóricamente, en este tipo de controlador, si la señal de error es cero, la salida del controlador también será cero. La respuesta, en teoría es instantánea, con lo cual el tiempo no interviene en el control. Sin embargo, en la práctica, esto no es así, de forma que, si la variación de la señal de entrada es muy rápida, el controlador no puede seguir dicha variación y seguirá una trayectoria exponencial hasta alcanzar la salida deseada.

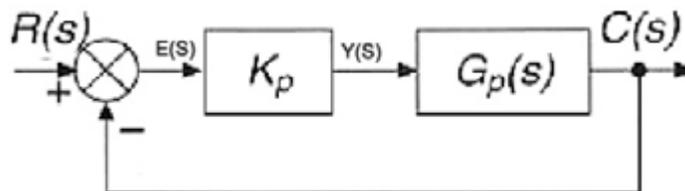


Figura 1.1 Controlador de acción Proporcional (P).

Una propiedad importante del regulador P es que como resultado de la rígida relación entre la señal de error del sistema y la variable regulada, siempre queda alguna señal de error del sistema. El controlador P no puede compensar esta señal de error remanente (permanente) del sistema (señal de OFFSET).

### 1.2.1.2 Controlador de acción Integral (I).

En un controlador integral, la señal de salida del mismo varía en función de la desviación y del tiempo en que se mantiene la misma, o dicho de otra manera, el valor de la acción de control es proporcional a la integral de la señal de error. Esto implica que mientras que en la señal proporcional no influía el tiempo, sino que la salida únicamente variaba en función de las modificaciones de la señal de error, en este tipo de control la acción varía según la desviación de la salida y el tiempo durante el que esta desviación se mantiene.

Llamando  $y(t)$  a la señal de salida (salida del regulador) y  $e(t)$  a la señal de error (entrada al regulador), en un control integral tendremos que la función de transferencia del bloque controlador es la mostrada en la siguiente ecuación.

$$Y(S) / E(S) = K_i / S \quad (1-2)$$

Donde:

- $Y(S)$  es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$  es la señal de error en el dominio de Laplace.
- $K_i$  es la pendiente de la rampa de acción integral, lo que implica que la velocidad de respuesta del sistema de control dependerá del valor de  $K_i$ .

El problema principal del controlador integral radica en que la respuesta inicial es muy lenta, y hasta pasado un tiempo, el controlador no empieza a ser efectivo. Sin embargo elimina el error remanente que tenía el controlador proporcional.

### 1.2.1.3 Controlador de acción proporcional e integral (PI).

En la práctica no existen controladores que tengan sólo acción integral sino que llevan combinada una acción proporcional. Estas dos acciones se complementan. La primera en actuar es la acción proporcional (instantáneamente) mientras que la integral actúa durante un intervalo de tiempo. Así y por medio de la acción integral se elimina la desviación remanente (proporcional). (Ver figura 1.2).

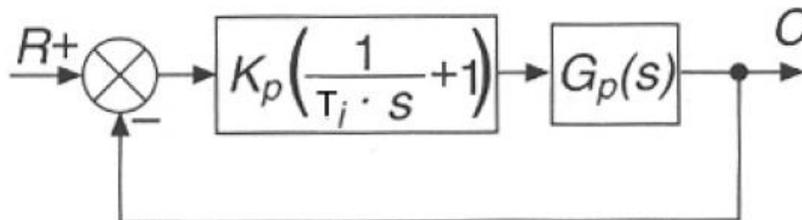


Figura 1.2 Controlador de acción Proporcional-Integral (PI).

La función de transferencia del bloque de control PI responde a la siguiente ecuación.

$$Y(S) / E(S) = K_p * (1 + 1 / T_i * S) \quad (1-3)$$

Donde:

- $Y(S)$  es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$  es la señal de error en el dominio de Laplace.
- $K_p$  es la ganancia del bloque de control.
- $T_i$  es el tiempo integral y controla la acción integral del sistema.

Si  $T_i$  es muy grande la pendiente de la rampa, correspondiente al efecto integral será pequeña y, por tanto, el efecto de esta acción suave, y viceversa. A  $T_i$  se le llama también tiempo de duplicación ya que es el tiempo que tarda la acción integral en igualar a la acción proporcional ante un error de tipo escalón.

La respuesta del controlador PI es la suma de las respuestas de un controlador proporcional y un controlador integral lo que proporciona una respuesta instantánea al producirse la correspondiente señal de error provocada por el control proporcional y un posterior control integral que se encargará de extinguir totalmente la señal de error.

#### 1.2.1.4 Controlador de acción proporcional y derivativa (PD).

Esta acción, al igual que la integral, no se emplea solo, sino que va unida a la acción proporcional (PD). (Ver figura 1.3).

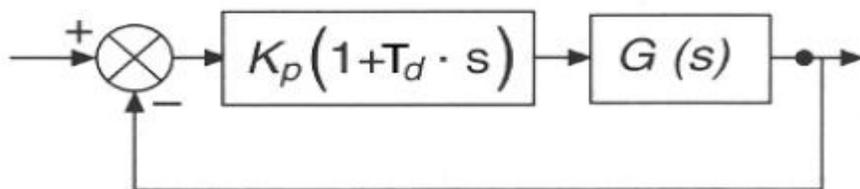


Figura 1.3 Controlador de acción Proporcional-Derivativo (PD).

En el control proporcional y derivativo PD, la función de transferencia del bloque de control responde a la siguiente siguiente ecuación:

$$Y(S) / E(S) = K_p * (1 + T_d * S) \quad (1-4)$$

Donde:

- $Y(S)$  es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$  es la señal de error en el dominio de Laplace.
- $K_p$  es la ganancia del bloque de control.
- $T_d$  es el tiempo derivativo o de adelanto, controla la acción derivativa del sistema.

A  $T_d$  se le llama también tiempo de duplicación ya que es el tiempo que tarda la acción proporcional en igualar el efecto de la acción derivativa ante una señal de error de tipo rampa (es una medida de la rapidez con que compensa un controlador PD un cambio en la variable regulada, comparado con un controlador P puro). En este tipo de controladores, debemos tener en cuenta que la derivada de una constante es cero y, por tanto, en estos casos, el control derivativo no ejerce ningún efecto, siendo únicamente útil en los casos en los que la señal de error varía en el tiempo de forma continua.

Este tipo de controlador se utiliza en sistemas que deben actuar muy rápidamente, puesto que la salida está en continuo cambio. La acción derivativa por sí sola no se utiliza, puesto que para señales lentas, el error producido en la salida en régimen permanente es muy grande y si la señal de mando deja de actuar durante un tiempo largo la salida tenderá hacia cero y no se realizará entonces ningún control. La utilidad de este tipo de controlador radica en aumentar la velocidad de respuesta de un sistema de control, ya que, como se comentó en los controladores proporcionales, aunque la velocidad de respuesta teórica de un controlador proporcional es instantánea, en la práctica no es así, pudiendo ser una rampa o una exponencial de una duración considerable.

Al incorporar a un controlador proporcional las características de un controlador derivativo, se mejora sustancialmente la velocidad de respuesta del sistema, a costa de una menor precisión en la salida (durante el intervalo de tiempo en que el control derivativo esté funcionando).

#### **1.2.1.5 Controlador de acción PID.**

Aprovecha las características de los tres reguladores anteriores, de forma, que si la señal de error varía lentamente en el tiempo, predomina la acción proporcional e integral y, si la señal de error varía rápidamente, predomina la acción derivativa. Tiene la ventaja de tener una respuesta más rápida y una inmediata compensación de la señal de error en el

caso de cambios o perturbaciones. Tiene como desventaja que el bucle de regulación es más propenso a oscilar y los ajustes son más difíciles de realizar. (Ver figura 1.4).

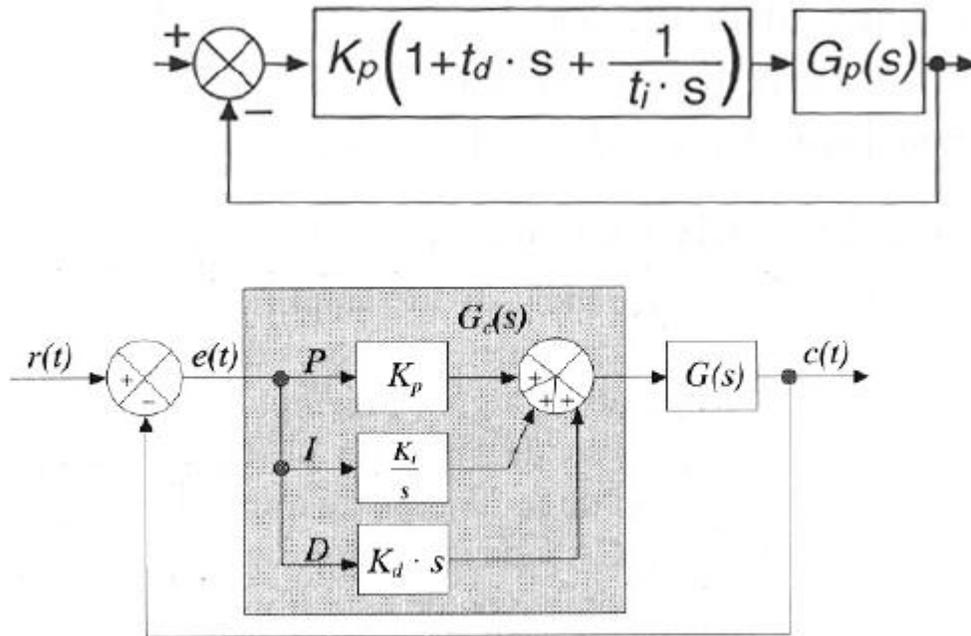


Figura 1.4 Controlador de acción Proporcional-Integral-Derivativo (PID).

La función de transferencia del regulador viene dada por la siguiente ecuación:

$$Y(S) / E(S) = K_p * (1 + T_d * S + 1 / T_i * S) \quad (1-5)$$

Donde:

- $Y(S)$  es la salida del regulador o controlador en el dominio de Laplace.
- $E(S)$  es la señal de error en el dominio de Laplace.
- $K_p$  es la ganancia del bloque de control.
- $T_d$  es el tiempo derivativo o de adelanto, controla la acción derivativa del sistema.
- $T_i$  es el tiempo integral y controla la acción integral del sistema.

Cuando se da una orden de cambio de magnitud, en una maniobra normal, la acción de control predominante del sistema es la proporcional, que aproximará la dirección al punto deseado de forma más o menos precisa. Una vez que la dirección esté cerca del punto deseado, comenzará la acción integral que eliminará el posible error producido por el control proporcional, hasta posicionar el volante en el punto preciso. Si la maniobra es lenta, la acción derivativa no tendrá apenas efecto. Si la maniobra requiere mayor velocidad de actuación, la acción de control derivativo adquirirá mayor importancia,

aumentando la velocidad de respuesta inicial del sistema y posteriormente actuará la acción proporcional y finalmente la integral. En el caso de una maniobra muy brusca, el control derivativo tomará máxima relevancia, quedando casi sin efecto la acción proporcional e integral, lo que provocará muy poca precisión en la maniobra.

### **1.3 Justificación del uso del control de lazo abierto aplicado en este proyecto de tesis.**

Dado que en los sistemas donde las entradas son conocidas previamente y en los que no hay perturbaciones se justifica el uso de lazo abierto, el desarrollo para conseguir la salida que en este caso es una velocidad o ángulo no requerirá de una realimentación.

## 1.4 Clasificación de los motores eléctricos.

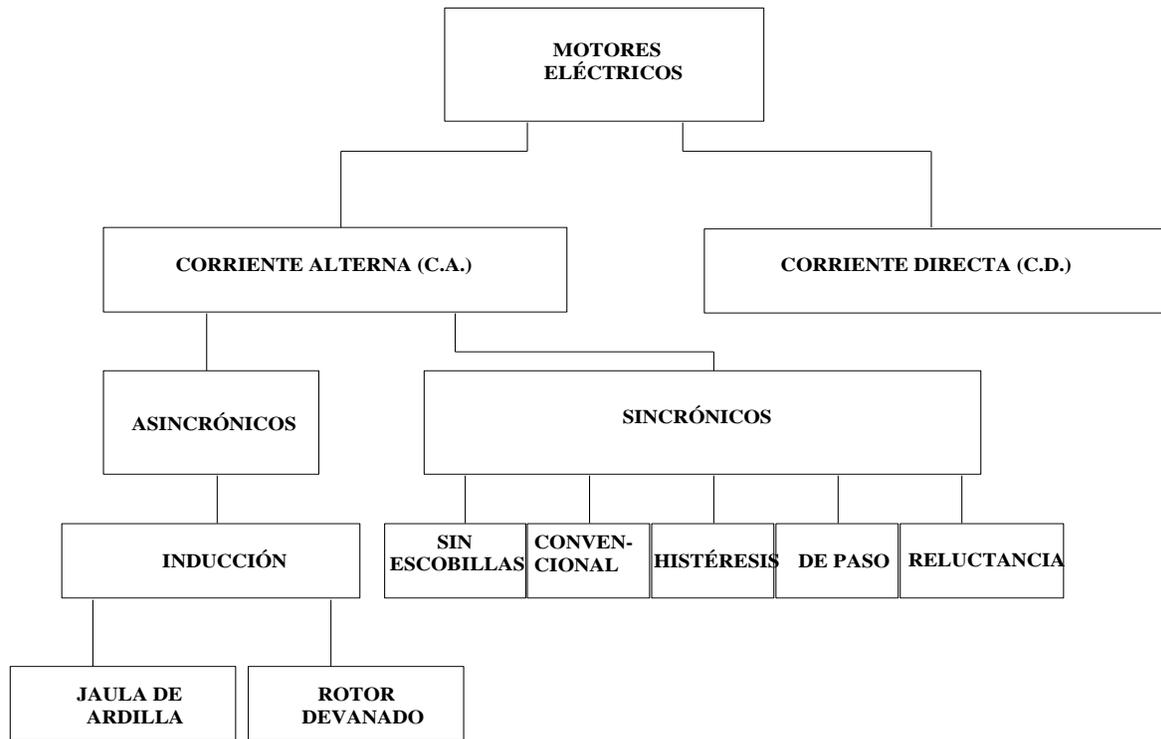


Figura 1.5 Clasificación de los motores eléctricos.

### 1.4.1 Motores.

Los dispositivos rotatorios de conversión de energía electromecánica son conocidos como máquinas rotatorias.

Están clasificadas como máquinas de corriente continua si sus salidas son en corriente directa o si la energía de entrada a las máquinas proviene de una fuente de corriente directa. Se llaman máquinas de corriente alterna si sus salidas son periódicas o si la energía primaria de entrada proviene de una fuente de corriente alterna.

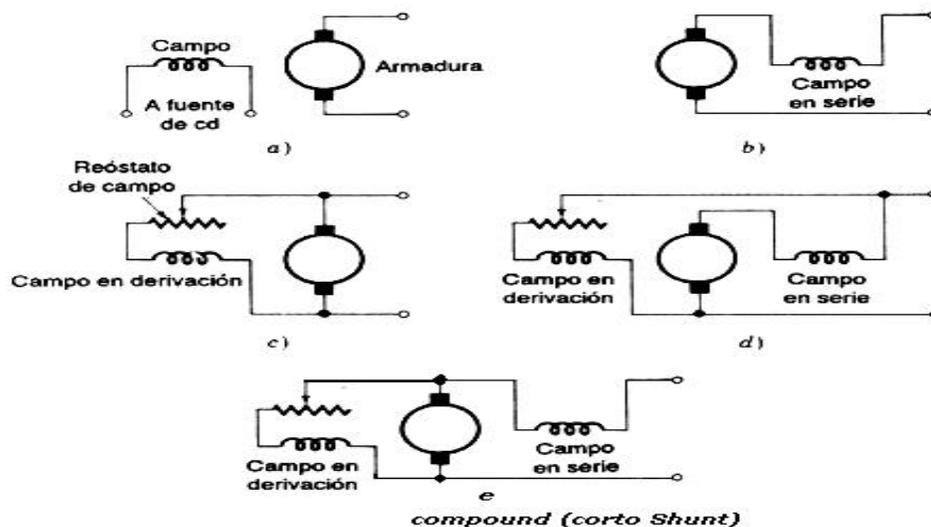
Una máquina rotatoria se llama motor si convierte energía eléctrica en mecánica y se llama generador si convierte energía mecánica en eléctrica.

Hay dos tipos principales de aplicaciones de las máquinas rotatorias. Como generadores, son usadas para proporcionar energía eléctrica industrial o domiciliaria y como motor para hacer girar dispositivos mecánicos tales como: ventiladores, bombas, etc. El otro tipo de aplicación se refiere como dispositivo de posición y/o para transportar información de una parte a otra.

Los métodos de análisis son determinados ampliamente por la aplicación de las máquinas y por las condiciones bajo las cuales operan. En algunos casos puede ser necesario determinar tanto el comportamiento transitorio como en estado estable de la máquina; en algunos otros podemos estar interesados solamente en una máquina aislada y en otros, la máquina bajo consideración, puede ser parte de un sistema complejo.

### 1.4.2 La Máquina de Corriente Continua.

Desde el punto de vista constructivo, esta máquina está constituida por dos núcleos de hierro, generalmente laminado, uno fijo (carcasa) y otro que gira (inducido o armadura). En ambos núcleos van ubicadas bobinas de diferentes configuraciones que crean campos electromagnéticos que interactúan, de modo que se pueda obtener o una tensión generada o bien un torque motriz en el eje, según la máquina trabaje como generador o motor. Las bobinas del rotor, que en este caso para la máquina de corriente continua (c.c.) se llama inducido o armadura, se definen como aquellas en las que se induce un voltaje, y las bobinas inductoras ubicadas en el estator, que en este caso para la máquina de c.c. se llama carcasa, se define como aquellas que producen el flujo magnético principal en la máquina.



Conexiones del circuito de campo de máquinas de cd: a) excitación separada, b) en serie, c) en derivación, d) compuesto.

Figura 1.6 Conexiones en máquinas de C.D.

En una máquina de c.c. normal, los embobinados del inducido se localizan en el rotor y los embobinados inductores se localizan en el estator (parte fija).

#### 1.4.2.1 Método de excitación de las máquinas de corriente continua.

Una de las ventajas de las máquinas de c.c. es su variedad de características de excitación, que está directamente ligado con el método que se utilice para excitar el campo.

Según el tipo de excitación, se clasifican en:

- i) Excitación separada: el campo es excitado por una fuente externa de c.c.
- ii) Autoexcitadas: la propia máquina proporciona la excitación del campo.

Dentro de este último tipo existen a su vez los tipos serie, derivación y compound o compuesto; como se indica en la figura 1.6.

Las ecuaciones eléctricas de equilibrio son en términos generales:

$$v(t) = e_a + (R_a + L_a p) i_a \quad (1-6)$$

donde  $p = d/dt$

$$v(t) = e_a + (R_a + L_a p) i_a \quad (1-7)$$

En que se debe tener en cuenta:

- Conexión de campo y armadura.
- Resistencias:  $R_a$ , del devanado de armadura y  $R_f$  del enrollado del campo.
- $i_a$ : corriente de armadura.
- $e_a$  tensión de armadura.

y en régimen permanente :

$$V_T = E_a + I_a R_a \quad (1-8)$$

donde:

$V_T$  = voltaje aplicado a las terminales del motor

$I_a$  = corriente de armadura

$R_a$  = resistencia de armadura

### **1.4.3 Motores de corriente alterna.**

Existen dos tipos básicos de motores de corriente alterna polifásica: los motores síncronos y los motores de inducción.

#### **1.4.3.1 Motores de inducción.**

La máquina asíncrona o de inducción se caracteriza por que la corriente del devanado inducido (normalmente rotor) se debe a la fem inducida en un circuito cerrado.

La velocidad de la máquina asíncrona no está impuesta por la red, existe un pequeño deslizamiento.

Su uso habitual es como motor aunque es posible como generador (eólica).

##### **1.4.3.1.1 Descripción de una máquina de inducción trifásica.**

Estator.

Conjunto trifásico simétrico de bobinas. Estas bobinas son idénticas entre sí y se puede conectar en estrella o triángulo. El número de bobinas por fase con circuito magnético independiente establece el número total de polos de la máquina a través de las relaciones conocidas (ej. : máquina de 2 bobinas por fase es máquina de 4 polos). El estator es alimentado por un sistema equilibrado de tensiones sinusoidales.

Rotor.

Puede presentar cualquiera de las dos formas siguientes:

a) Rotor bobinado: compuesto de un enrollado similar al del estator con el mismo número de fases y de polos que él. Los terminales de estos enrollados se conectan a anillos aislados montados en el eje, de modo que, mediante escobillas de carbón, estas terminales se cortocircuitan entre sí a través de las resistencias externas.

b) Rotor tipo jaula de ardilla: está compuesto de barras conductoras alojadas en ranuras del fierro del rotor, cortocircuitadas en ambos extremos mediante anillos conductores. La disposición de las barras es simétrica en toda la periferia del rotor.

Velocidad síncrona.

La velocidad de rotación del campo magnético (C.M.) producido por el estator, es la llamada velocidad síncrona, es directamente proporcional a la frecuencia del voltaje de alimentación e inversamente proporcional al número de pares de polos.

$$n_s = \frac{f_s}{p/2} = \frac{2 \times f_s}{p} \quad \text{r/s} \quad (1-9)$$

$$n_s = \frac{120 \times f_s}{p} \quad \text{r/min} \quad (1-10)$$

donde:

$f_s$  = frecuencia de alimentación trifásica

$n_s$  = velocidad síncrona

P = número de polos en el estator

#### 1.4.3.1.2 Funcionamiento del motor de inducción.

La diferencia entre la velocidad del campo magnético rotatorio (C.M.R) y la velocidad del rotor es llamada velocidad de deslizamiento "s". Normalmente se da en porcentaje o en por unidad (p.u.). En otras palabras, el rotor atrasa al C.M.R. del estator en el valor del deslizamiento.

Se define la magnitud llamada deslizamiento "S", del motor, como:

$$n = n_s - n_r \quad (1-11)$$

$$s = \frac{n_s - n_r}{n_s} \quad (1-12)$$

donde:

$n$  = velocidad de deslizamiento (r/min)

$n_s$  = velocidad sincrona (r/min)

$n_r$  = velocidad del rotor (r/min)

$s$  = deslizamiento (PU)

La frecuencia del voltaje inducido en el rotor por el campo magnético rotatorio es:

$$f_r = \frac{P \times n}{120} \quad (1-13)$$

donde:

$f_r$  = frecuencia del rotor (Hz)

$P$  = número de polos

$n$  = velocidad de deslizamiento (r/min)

Sustituyendo la velocidad de deslizamiento

$$f_r = \frac{P(n_s - n_r)}{120} = \frac{s P n_s}{120} \quad \text{ya que } n_s - n_r = s n_s \quad (1-14)$$

Si el rotor está bloqueado  $s=1$ , y la ecuación anterior se convierte en:

$$f_{BR} = \frac{P n_s}{120} \quad (1-15)$$

donde  $f_{BR}$  = frecuencia del voltaje generado en el rotor bloqueado. Por lo anterior:

$$f_r = s f_{BR} \quad (1-16)$$

El voltaje generado en dos barras formando un circuito en el rotor es entonces:

$$E_r = 4.44 N s f_{BR} \Phi_{\max} \quad (1-17)$$

Con el rotor bloqueado,  $s=1$  por lo que:

$$E_{BR} = 4.44 N f_{BR} \Phi_{\max} \quad (1-18)$$

Y así se determina el valor del voltaje en el rotor a cualquier velocidad:

$$E_r = sE_{BR} \quad (1-19)$$

#### 1.4.3.1.3 Circuito equivalente simplificado y por fase del rotor del motor de inducción.

La reactancia de rotor expresada en términos de la frecuencia del rotor es:

$$X_R = 2\pi f_r L_r = 2\pi (s f_{BR}) L_r = s X_{BR} \quad (1-20)$$

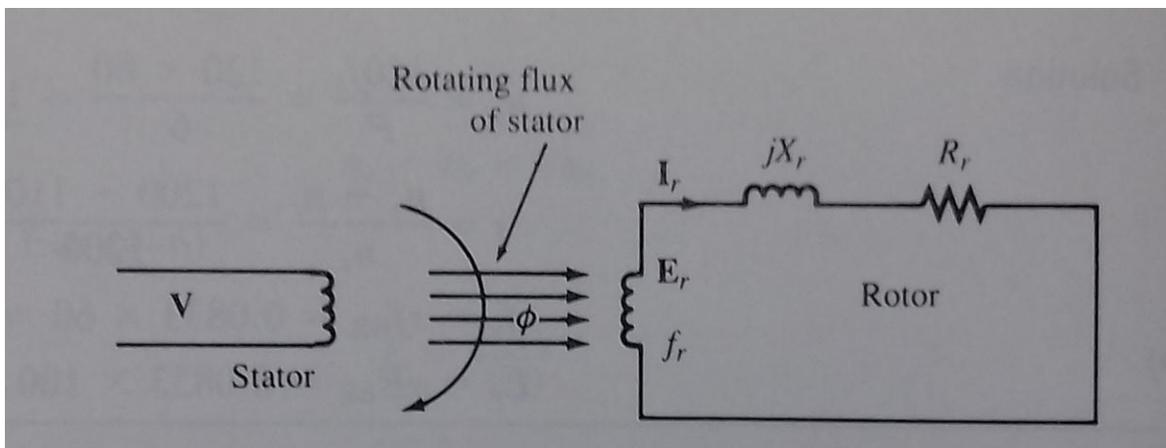


Figura 1.7. Circuito equivalente simplificado del rotor del motor de inducción

La impedancia del rotor es entonces:

$$Z_r = R_r + jsX_{BR} \quad (1-21)$$

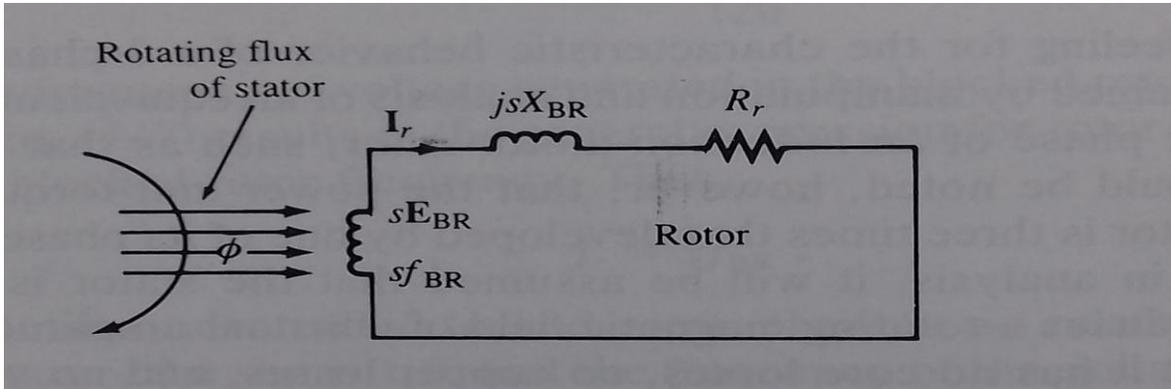


Fig. 1.8. Circuito equivalente simplificado del rotor del motor de inducción con valores equivalentes.

Aplicando la ley de Ohm, la corriente en el rotor es:

$$I_r = \frac{sE_{BR}}{Z_r} = \frac{sE_{BR}}{R_r + jsX_{BR}} \quad (1.22)$$

#### 1.4.3.2 Motores síncronos.

Los motores síncronos son un tipo de motor de corriente alterna en el que la rotación del eje está sincronizada con la frecuencia de la corriente de alimentación; el período de rotación es exactamente igual a un número entero de ciclos de C.A. Su velocidad de giro es constante y depende de la frecuencia de la tensión de la red eléctrica a la que esté conectado y por el número de pares de polos del motor, siendo conocida esa velocidad como "velocidad de sincronismo". Este tipo de motor contiene electro magnetos en el estator del motor que crean un campo magnético que rota en el tiempo a esta velocidad de sincronismo.

Durante el arranque el rotor es llevado a máxima velocidad como motor de inducción a través del uso de embobinados damper hasta alcanzar un deslizamiento muy pequeño, o si carece de estos embobinados damper, mediante un motor auxiliar o turbina, sin conectar carga alguna al motor síncrono. Posteriormente se aplica corriente directa a los embobinados magnéticos en el rotor formando polos norte y sur de manera consecutiva. En esos momentos los campos opuestos en el rotor y estator quedan alineados y el motor opera a velocidad síncrona.

$$n_s = \frac{120 \times f_s}{p} \quad \text{r/min} \quad (1-23)$$

donde:

f: frecuencia de la red a la que está conectada la máquina (Hz)

p: número de polos que tiene la máquina

$n_s$  : velocidad síncrona (r/min)

#### 1.4.3.2.1 Circuito equivalente para una fase del motor síncrono.

El circuito equivalente para una fase del motor síncrono de rotor cilíndrico es como se muestra en la figura 1.9. Todos los valores son por fase. Aplicando la ley de voltaje de Kirchhoff resulta,

$$V_T = I_a R_a + I_a jX_l + I_a jX_{ar} + E_f \quad (1-24)$$

sustituyendo reactancias,

$$X_s = X_l + X_{ar} \quad (1-25)$$

$$V_T = I_a (R_a + jX_s) + E_f \quad (1-26)$$

donde:

$R_a$  = resistencia de armadura  $\left(\frac{\Omega}{\text{fase}}\right)$

$X_l$  = reactancia de armadura ( $\Omega/\text{fase}$ )

$X_s$  = reactancia síncrona  $\left(\frac{\Omega}{\text{fase}}\right)$

$Z_s$  = impedancia síncrona  $\left(\frac{\Omega}{\text{fase}}\right)$

$V_T$  = Voltaje suministrado por fase (V)

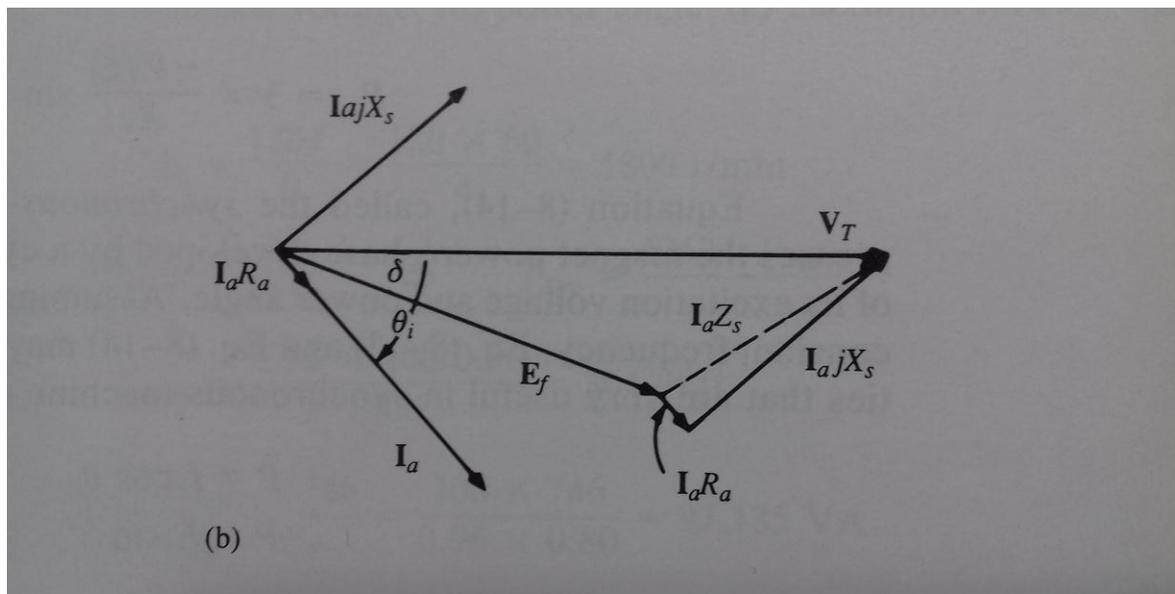
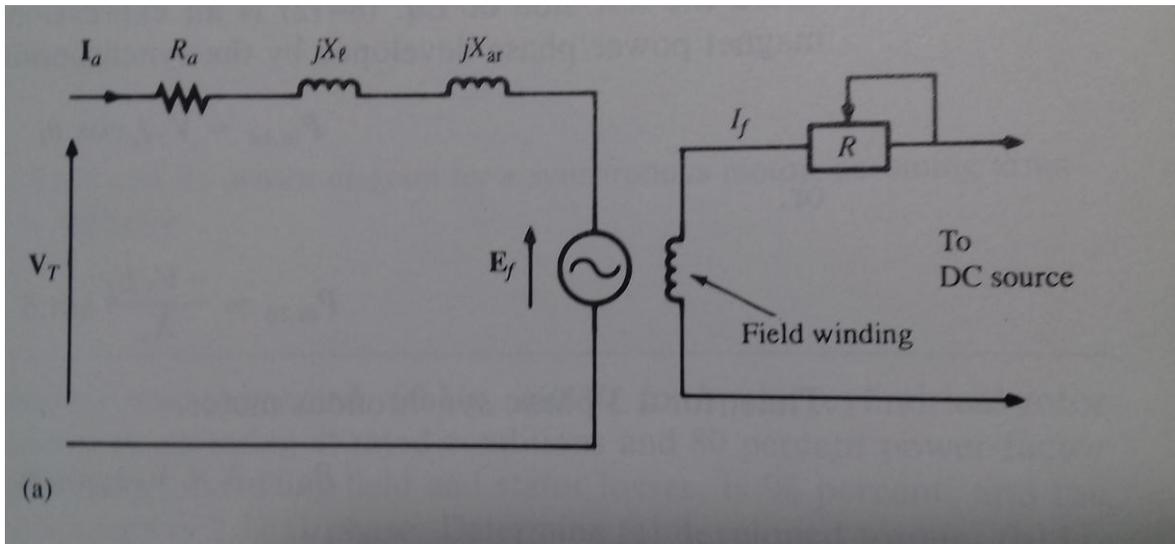


Figura 1.9 Circuito equivalente de una fase (a) y diagrama fasorial (b).

## 2. MOTORES A PASOS

### 2.1 Introducción.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos.

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$ , es decir, que se necesitarán 4 pasos en el primer caso ( $90^\circ$ ) y 200 para el segundo caso ( $1.8^\circ$ ), para completar un giro completo de  $360^\circ$ .

Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.

Los motores de pasos son motores eléctricos que se mueven a partir de pulsos. Los motores de pasos los encontramos en impresoras, manejadores de discos, alimentadores de papel, plotters, controles en aviones, brazos mecánicos, robots, etc.

Un motor de pasos es:

- Un actuador electromagnético rotatorio que convierte mecánicamente entradas de pulsos digitales a movimiento rotatorio de la flecha o eje.
- La rotación no solo tiene una relación directa al número de pulsos de entrada, sino que la velocidad está relacionada a la frecuencia de los pulsos .

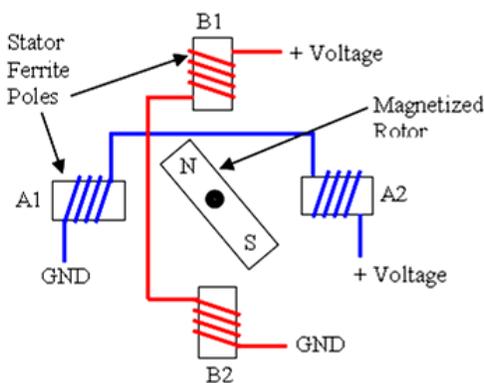


Figura 2.1 Motor a pasos.

## 2.2 Tipos básicos de motores de pasos de acuerdo a su principio de operación.

- Reluctancia Variable
- Imán Permanente
- Híbridos

### 2.2.1 Reluctancia variable (VR).

Estos motores se caracterizan por tener un rotor múltiple de hierro suave y un embobinado en el estator.

Operan generalmente con pasos de ángulo entre 5 y 15 grados a una velocidad de pasos relativamente alta, y no tienen par de retén (par de retén es el par de mantenimiento cuando no está fluyendo corriente en el motor).

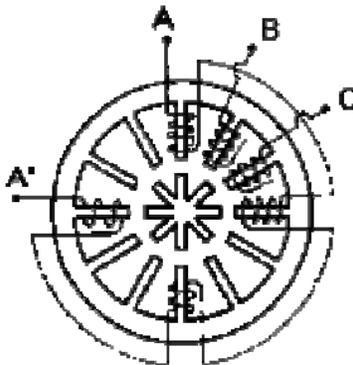


Figura 2.2 Motor de reluctancia variable.

Como se ven la figura 2.2, cuando la fase A es energizada, 4 dientes del rotor se alinean con los 4 dientes del estator de la fase A por atracción magnética.

El siguiente paso es adoptado cuando A es apagada y la fase B es energizada, rotando el rotor en sentido horario 15 grados. Continuando la secuencia, C es energizada en el siguiente y de nuevo la A.

Se consigue una rotación contra horario cuando el orden de las fases es invertido.

El motor de pasos de reluctancia variable rota cuando los dientes del rotor son atraídos a los dientes del estator electromagnéticamente energizados. La inercia del rotor de un motor de paso de reluctancia variable es pequeña y la respuesta es muy rápida, pero la inercia permitida de la carga es pequeña. Cuando los devanados no están energizados, el

par estático de este tipo de motor es cero. Generalmente, el paso angular de este motor de paso de reluctancia variable es de  $15^\circ$ .

### 2.2.2 Imán Permanente (PM).

- Estos motores difieren de los de VR en que tienen rotores de imán permanente sin dientes y son magnetizados perpendicularmente a los ejes.
- Energizando las cuatro fases en secuencia, el rotor gira conforme es atraído por los polos magnéticos.
- El motor mostrado en la figura 2.3 tomará pasos de 90 grados si la secuencia de alimentación de los embobinados es ABCD.

. Estos motores tienen pasos de ángulos entre 45 y 90 grados y la velocidad de pasos es relativamente baja, pero presentan un alto par y unas buenas características de amortiguamiento.

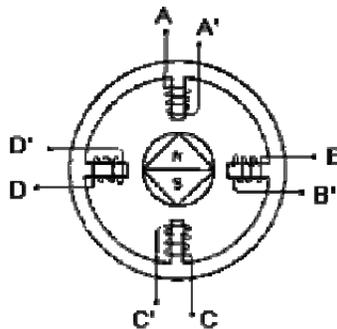


Figura 2.3 Motor de imán permanente.

El motor de pasos de rotor de imán permanente permite mantener un par diferente de cero cuando el motor no está energizado. Dependiendo de la construcción del motor, es típicamente posible obtener pasos angulares de 7.5, 11.25, 15, 18, 45 o  $90^\circ$ . El ángulo de rotación se determina por el número de polos en el estator.

### 2.2.3 Híbrido.

Combinan las cualidades de los VR y PM, los motores híbridos tiene algunas de los comportamientos deseables de cada uno de ellos.

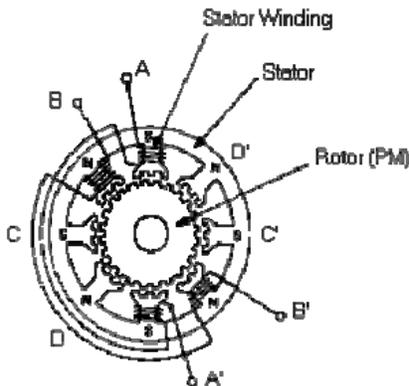


Figura 2.4 Motor Híbrido.

Tiene alto par de retén y un excelente par de sostenimiento dinámico, y pueden ser operados a una alta velocidad de pasos.

Normalmente exhiben ángulos de pasos de 0.9 a 5 grados

El motor de pasos híbrido se caracteriza por tener varios dientes en el estator y en el rotor como en la figura 2.4 con un imán concéntrico magnetizado axialmente alrededor de su eje. Se puede ver que esta configuración es una mezcla de los tipos de reluctancia variable e imán permanente. Este tipo de motor tiene una alta precisión y alto par, se puede configurar para suministrar un paso angular tan pequeño como 1.8°.

## 2.3 Tipos de motores paso a paso de imán permanente.

### 2.3.1 Motor unipolar.

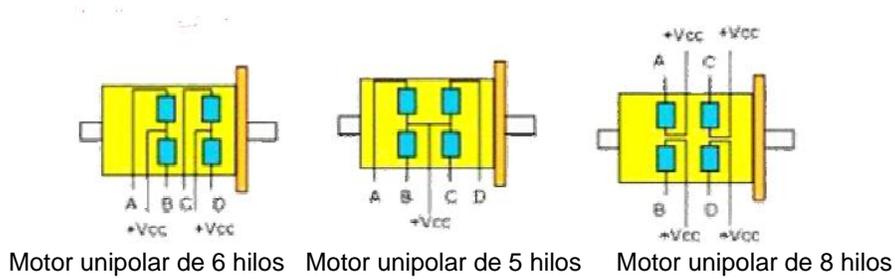


Figura 2.5 Tipos de motores unipolares.

- Estos motores suelen tener 6 ó 5 cables de salida, dependiendo de sus conexiones internas.
- Este tipo se caracteriza por ser más simple de controlar.

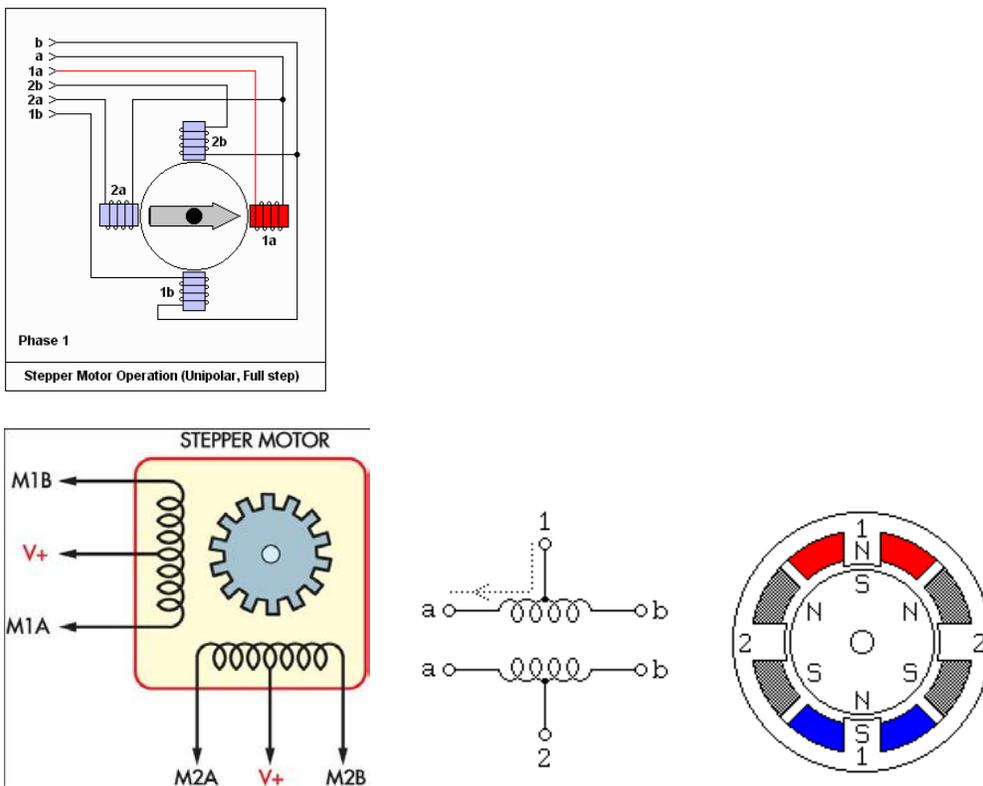


Figura 2.6 Esquema de un motor de pasos de 4 fases (bobinas) unipolar.

### 2.3.2 Motor bipolar.

- Estos tiene generalmente cuatro cables de salida.
- Se requiere de un cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

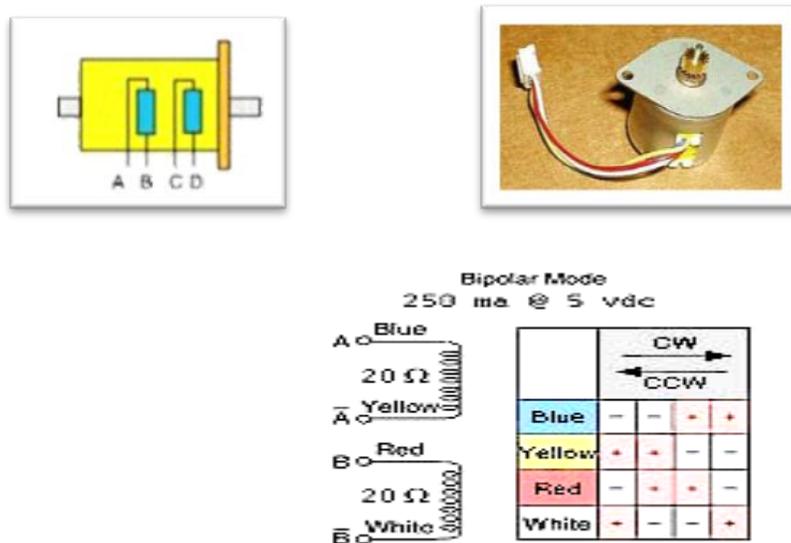


Figura 2.7 Motor bipolar y configuración de polarización de las bobinas para conducirlo en sentido horario o anti-horario.

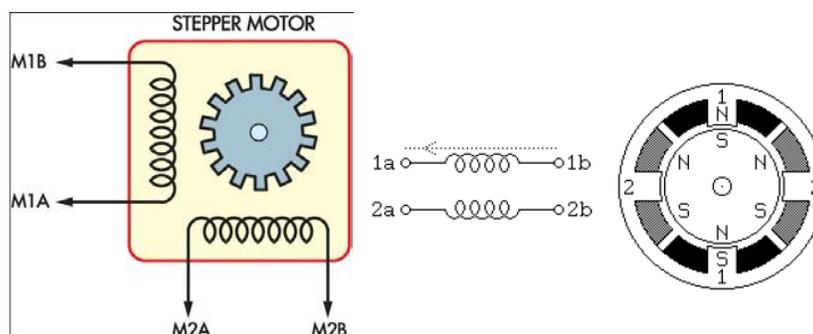


Figura 2.8 Esquema de un motor de pasos de 2 fases(bobinas) bipolar.

## 2.4 Secuencias de manejo en motores a pasos.

### 2.4.1 Secuencias para manejar motores paso a paso unipolares.

- Paso completo:
  - Secuencia del tipo fase simple (wave drive)
  - Secuencia normal (doble fase)
- Secuencia del tipo medio paso
- Micropaso

Secuencia del tipo fase simple (wave drive): en esta secuencia se activa sólo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. La desventaja es que al estar solo una bobina activada, el torque de paso y retención es menor.

N° de Pasos	A	B	C	D
Paso 1	1	0	0	0
Paso 2	0	1	0	0
Paso 3	0	0	1	0
Paso 4	0	0	0	1

Figura 2.9 Secuencia de fase simple.

Este modo solo será usado donde el desempeño de par y velocidad no sean importantes, por ejemplo, donde el motor sea operado a una velocidad fija y con condiciones de carga bien definidas.

Secuencia normal (doble fase): esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un alto torque de paso y de retención.

N° de Pasos	A	B	C	D
Paso 1	1	1	0	0
Paso 2	0	1	1	0
Paso 3	0	0	1	1
Paso 4	1	0	0	1

Figura 2.10 Secuencia normal.

Secuencia del tipo medio paso: en esta secuencia se activan las bobinas de tal forma que brindan un movimiento igual a la mitad del paso real. Para ello se activan primero 2 bobinas y luego sólo 1 y así sucesivamente.

Nº de Pasos	A	B	C	D
Paso 1	1	0	0	0
Paso 2	1	1	0	0
Paso 3	0	1	0	0
Paso 4	0	1	1	0
Paso 5	0	0	1	0
Paso 6	0	0	1	1
Paso 7	0	0	0	1
Paso 8	1	0	0	1

Figura 2.11 Secuencia del tipo medio paso.

### 2.4.2 Secuencias para manejar motores a paso bipolares.

Para hacerlo se deben invertir las polaridades de los terminales de las bobinas 1 y 2 en una determinada secuencia para lograr un giro a derecha, y en secuencia opuesta para que gire a izquierda.

Nº de Pasos	1a	1b	2a	2b
Paso 1	+Vcc	GND	+Vcc	GND
Paso 2	+Vcc	GND	GND	+Vcc
Paso 3	GND	+Vcc	GND	+Vcc
Paso 4	GND	+Vcc	+Vcc	GND

Figura 2.12 Secuencia para motores a paso bipolar.

## 2.5 Parámetros de los motores a pasos.

Desde el punto de vista mecánico y eléctrico, es conveniente conocer el significado de algunas de las principales características y parámetros que se definen sobre un motor paso a paso:

- Par dinámico de trabajo (Working Torque): depende de sus características dinámicas y es el momento máximo que el motor es capaz de desarrollar sin perder paso, es decir, sin dejar de responder a algún impulso de excitación del estator y dependiendo, evidentemente, de la carga.

Generalmente se ofrecen, por parte del fabricante, curvas denominadas de arranque sin error (pull-in) y que relaciona el par en función el número de pasos.

Hay que tener en cuenta que, cuando la velocidad de giro del motor aumenta, se produce un aumento de la f.c.e.m. en él generada y, por tanto, una disminución de la corriente absorbida por los bobinados del estator, como consecuencia de todo ello, disminuye el par motor.

- Par de mantenimiento (Holding torque): es el par requerido para desviar, en régimen de excitación, un paso el rotor cuando la posición anterior es estable ; es mayor que el par dinámico y actúa como freno para mantener el rotor en una posición estable dada
- Par de detención (Detention torque): es una par de freno que siendo propio de los motores de imán permanente, es debida a la acción del rotor cuando los devanados del estator están desactivados.
- Angulo de paso (Step angle): se define como el avance angular que se produce en el motor por cada impulso de excitación. Se mide en grados, siendo los pasos estándar más importantes los siguientes:

Grados por impulso de excitación	Nº de pasos por vuelta
0,72°	500
1,8°	200
3,75°	96
7,5°	48
15°	24

Figura 2.13 Ángulo de paso (Step angle).

- Número de pasos por vuelta: es la cantidad de pasos que ha de efectuar el rotor para realizar una revolución completa.
- Frecuencia de paso máximo (Maximum pull-in/out): se define como el máximo número de pasos por segundo que puede recibir el motor funcionando adecuadamente.
- Momento de inercia del rotor: es su momento de inercia asociado que se expresa en gramos por centímetro cuadrado.
- Par de mantenimiento, de detención y dinámico: definidos anteriormente y expresados en miliNewton por metro.

## 2.6 Descripción del motor a pasos usado en este proyecto de tesis.

Para este trabajo usamos un motor de pasos de tecnología híbrida de la marca POLOLU, el modelo del motor es el SY35ST28-0504A; por su construcción, el motor da 200 pasos por revolución, consume 500mA por bobina y requiere un voltaje de polarización de 10V, la resistencia en sus devanados es de  $20\Omega$ , para más detalles referirse a la hoja de datos en el apéndice B. En la figura 2.14, se muestra una imagen del motor a pasos.

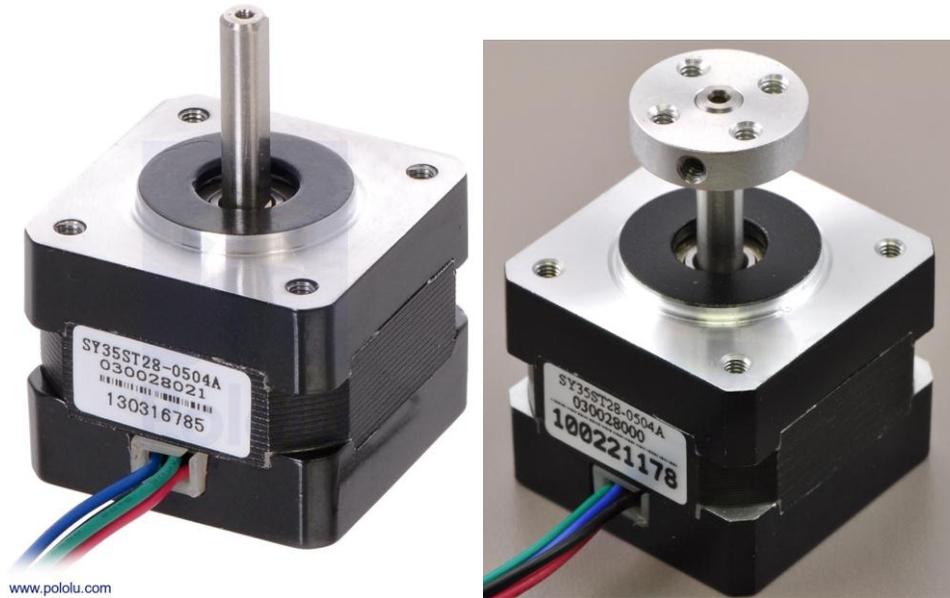


Figura 2.14 Motor a pasos Pololu modelo SY35ST280504A

La conexión interna de los devanados es como se indica en la figura 2.15.

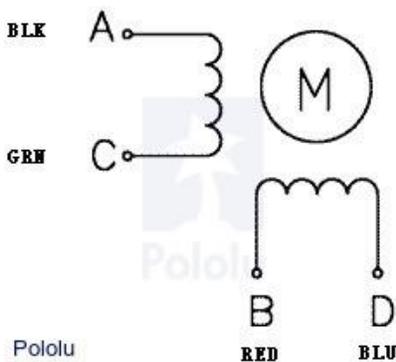
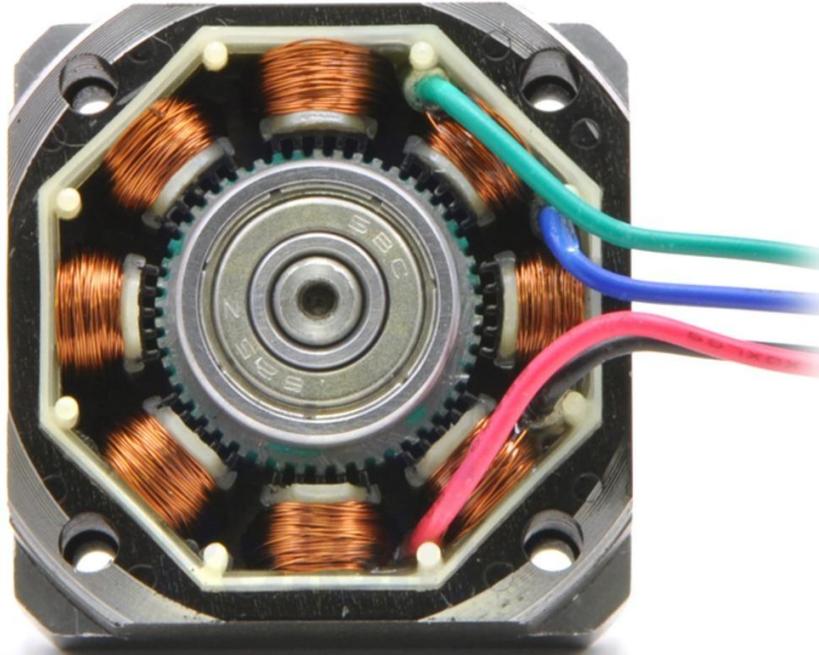


Figura 2. 15 Diagrama de cableado del motor a pasos.

En la figura 2.16 puede observarse el interior del motor a pasos empleado en este trabajo.



[www.pololu.com](http://www.pololu.com)

Figura 2. 16 Motor a pasos modelo SY35ST280504A (visto por dentro).

## 2.7 Ecuaciones electromagnéticas básicas para los motores a pasos genéricos.

La idea básica de funcionamiento de un motor paso-a-paso es similar a la de un motor síncrono, en donde existe un campo magnético rotatorio generado por las bobinas del estator, que se desplaza mediante conmutación de sus fases. A pesar de estas similitudes, el motor paso-a-paso tiene la ventaja de poder rotar al campo magnético a distintas

velocidades e incluso poder mantenerlo en alguna posición, según como sea la conmutación de sus fases.

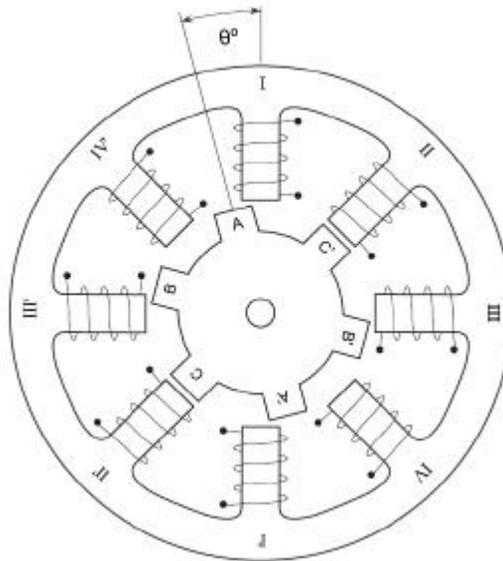


Figura 2.17 Motor paso a paso genérico (con P=3 y N=4).

Denominando P al número de pares de polos del rotor, y N al número de fases del estator. Designando por  $\Theta$  al ángulo de referencia tomado a partir de la alineación del polo de la fase I con el polo A del rotor, y con sentido creciente según la convención “regla de la mano derecha”.

Las ecuaciones eléctricas del sistema son las siguientes:

$$V_j(t) = i_j(t) * R_j + \frac{\partial \varphi_j}{\partial t}(i_1, \dots, i_N, \theta) \quad j=1, \dots, N \quad (2-1)$$

$$\frac{\partial \varphi_j}{\partial t} = \sum_{k=1}^N \frac{\partial \varphi_j}{\partial i_k} * \frac{\partial i_k}{\partial t} + \frac{\partial \varphi_j}{\partial \theta} * \omega \quad (2-2)$$

### 3. CONTROL DE POSICIÓN Y VELOCIDAD EN UN MOTOR A PASOS

Los motores de pasos pueden verse como actuadores electromecánicos que mueven su eje en función de los pulsos que reciben. Un pulso hace avanzar su eje unos cuantos grados, varios pulsos hará girar el eje un poco más de grados.

Por ejemplo, AIRPAX fabricante de motores de pasos, señala en los datos de placa de uno de sus motores: *STEP ANGLE: 7.5 Grados*.

Esto quiere decir que por cada pulso el motor avanza 7.5 grados. Claro que para girar una vuelta completa, es decir 360 grados se requieren 48 pulsos. Entonces, el avance del eje del motor está relacionada con el número de pulsos y la velocidad de rotación con su frecuencia. Entre cada pulso el motor mantiene su posición sin la ayuda de frenos o embragues.

El motor de pasos no responde en forma directa a los trenes de pulsos. Como el motor tiene varios devanados, estos deben ser energizados según una secuencia determinada, que al invertirse hace que el motor también invierta su sentido de rotación.

Los controles para los motores de pasos pueden ser unipolares o bipolares, según sí tienen la capacidad de invertir el voltaje de alimentación en los devanados del motor o no.

Los controles unipolares son los más sencillos y también los más económicos. Estos controles solo pueden usarse con motores de 5 ó 6 hilos. Cada devanado del motor cuenta con un tap central.

Los controles bipolares son un poco más sofisticados y pueden conectarse a motores de 4, 5 ó 6 hilos, debido a que tienen la capacidad de invertir el sentido de la corriente que pasa por cada devanado del motor de pasos. El control que se ocupa en el microcontrolador y el DSP es bipolar.

Los controles también pueden ser: Full step, half step o micro step. En los controles Full step un pulso hace girar el motor los grados nominales del motor.

En los controles Half step un pulso hace girar el motor la mitad de los grados nominales del motor. En nuestro ejemplo anterior como los grados por paso del motor AIRPAX son 7.5, en este modo de funcionamiento se requieren 96 pulsos para hacer girar el motor una vuelta completa.

En los controles micro step un pulso hace girar la décima parte de los grados nominales del motor.

Los controles para este tipo de motores se pueden implementar basándose en circuitos TTL, sin embargo los microcontroladores son particularmente útiles para elaborar desarrollos aún más complejos.

El control de un motor a pasos es relativamente sencillo al girar un paso a la vez o girar a una determinada tasa (resultado de la velocidad a la que los pasos son ejecutados), los motores a pasos pueden ser ligeramente complejos de alambrear al demandar más corriente con lo que una interfaz de potencia es necesaria para su operación .

Un motor bipolar consiste de un magneto anclado a la flecha del motor donde su posición se especifica por el par de embobinados localizados en el estator del motor. El rotor del motor se moverá debido a la serie de pasos discretos que le son suministrados a los embobinados en el estator.

A través del suministro de voltaje a los embobinados del motor por medio de la secuencia de estos pasos se puede controlar la dirección de rotación, el número de vueltas y la posición exacta de la flecha del motor. Controlando el tiempo entre los pasos o escalones, la velocidad y aceleración se regula.

Al suministrar un pulso a la vez el motor se moverá un paso y así sucesivamente por cada pulso. Esto nos permite controlar la posición del motor a pasos en lazo abierto. A diferencia de los motores de CA y DC, los motores a pasos no siempre necesitan sistemas de lazo cerrado para controlar la posición de la flecha.

Uno de los inconvenientes de los motores a pasos son las limitaciones de torque a altas velocidades, ya que el torque disminuye al aumentar la velocidad. El torque también disminuye en la velocidad resonante, en la que la flecha del motor vibrara mientras pierde el torque. La velocidad resonante depende del esquema Torque vs Velocidad del motor de pasos y de la carga.

### **3.1 Ecuaciones mecánicas fundamentales de los motores a pasos.**

Para iniciar el movimiento rotacional en el motor a pasos la corriente a través de los embobinados deberá cambiar de manera ordenada. Esto se consigue con un driver que da la secuencia de salida correcta mientras recibe un pulso y una señal de dirección.

Para girar el motor a velocidad constante, los pulsos deberán de generarse de manera controlada y constante.

Esto es:

$$\delta t = \frac{c}{f} [s] \quad (3-1)$$

donde  $f$  es la frecuencia de los pulsos,  $c$  será una constante y  $\delta t$  un instante de tiempo.

El ángulo del motor a pasos  $\alpha$ , la posición  $\theta$ , y la velocidad  $\omega$  están dados por:

$$\alpha = \frac{2\pi}{spr} [rad] \quad (3-2)$$

$$\theta = n\alpha [rad] \quad (3-3)$$

$$\omega = \frac{\alpha}{\delta t} \left[ \frac{rad}{s} \right] \quad (3-4)$$

Donde  $spr$  es el número de pasos por vuelta.  $n$  es el número de pasos.

$$1 \text{ rad/s} = 9.55 \text{ rpm} \quad (3-5)$$

3.2 Diagramas de flujo para el control de posición.

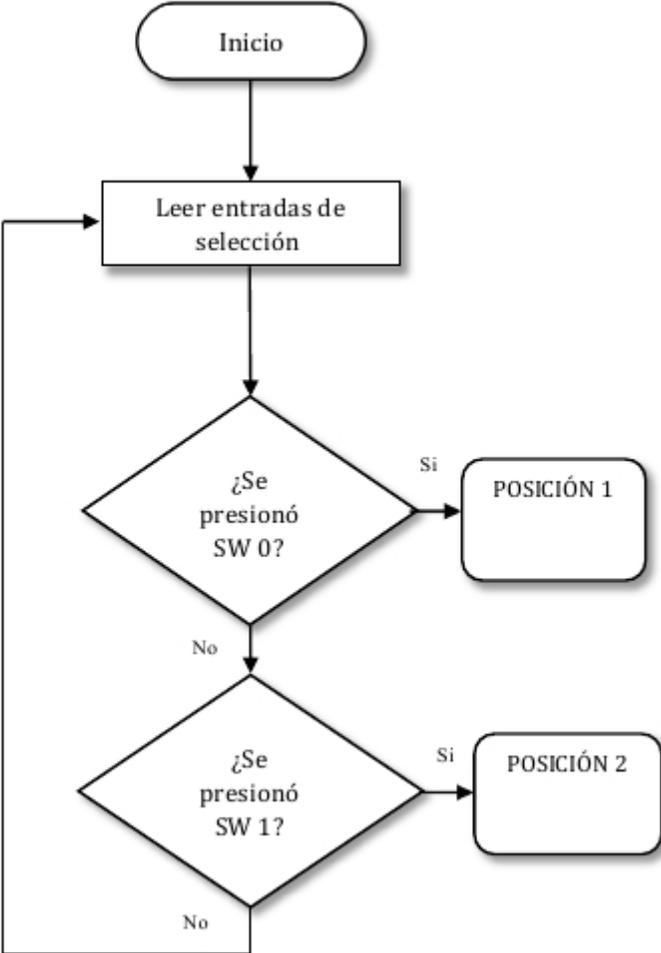


Figura 3.1 Inicio y ciclo de selección de posición.

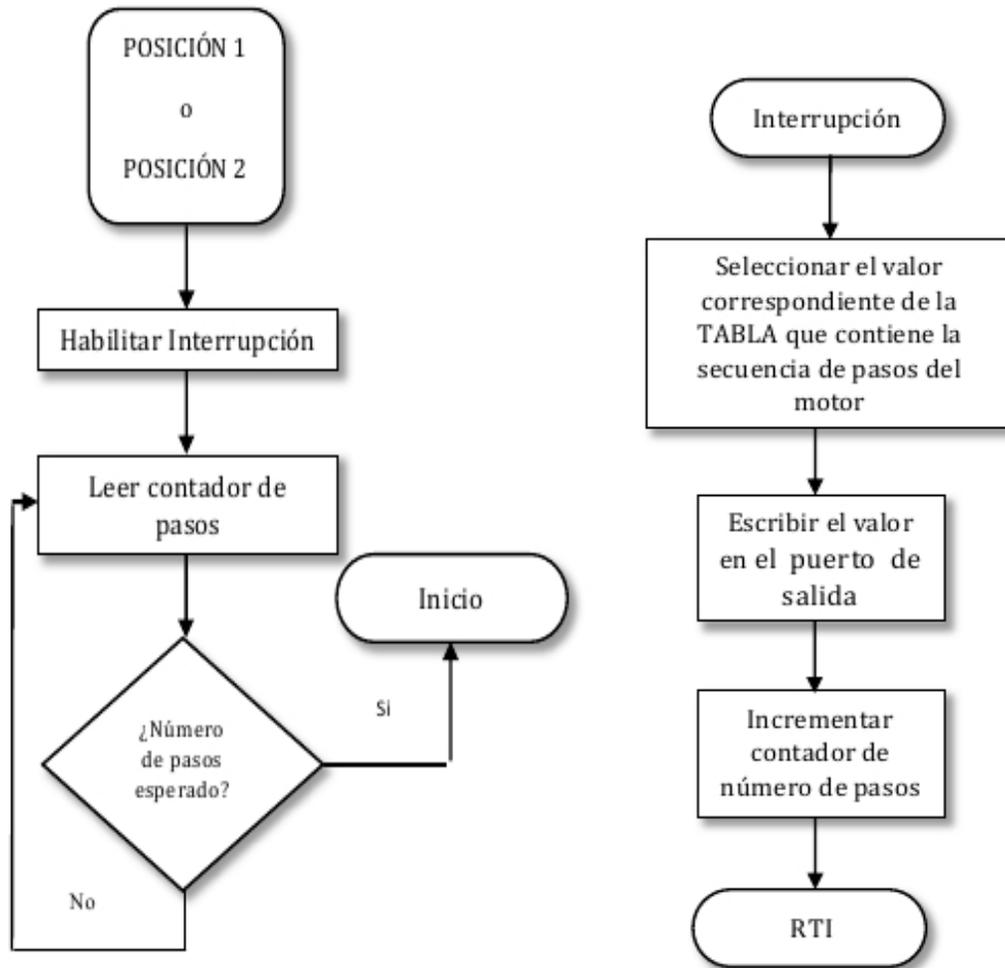


Figura 3.2 Ciclo de posición y ciclo de interrupción.

### 3.3 Diagramas de flujo para el control de velocidad.

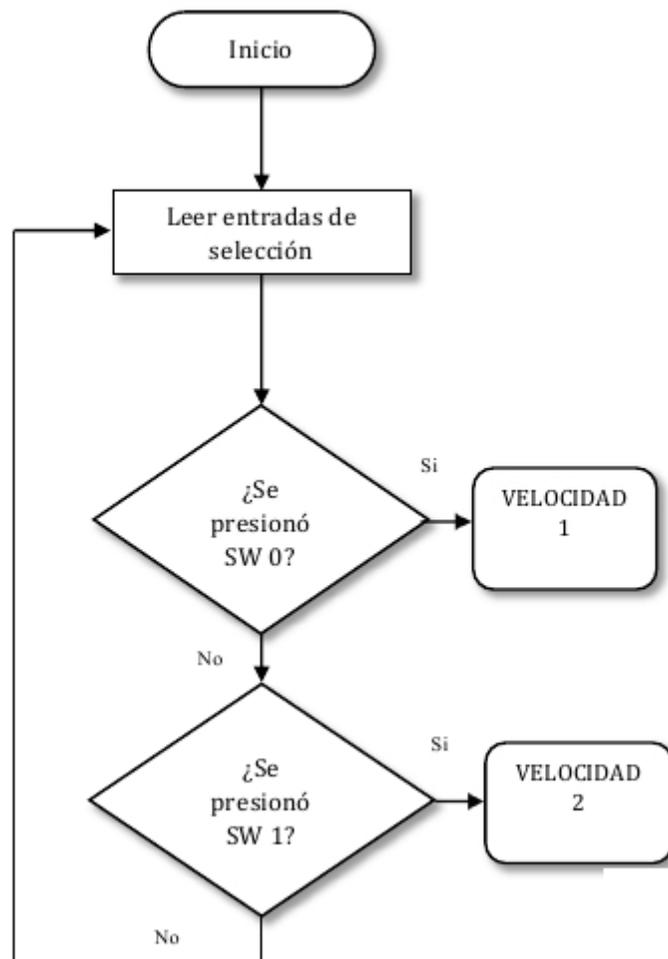


Figura 3.3 Inicio y ciclo de selección de velocidad.

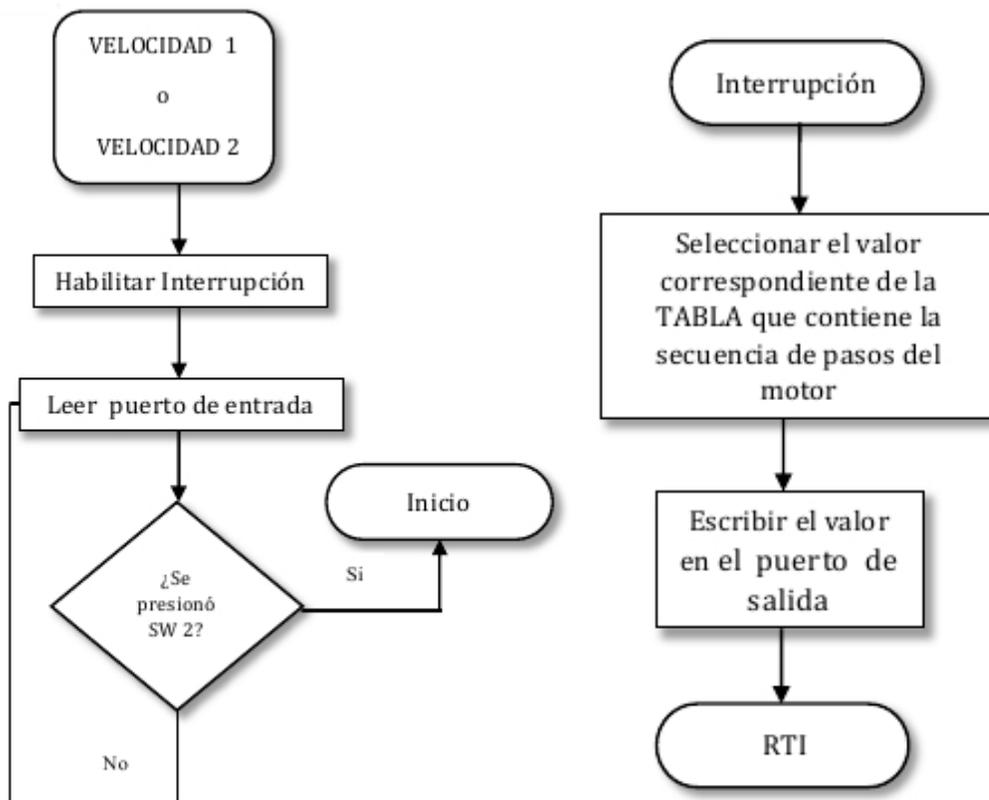


Figura 3.4 Ciclo de velocidad y ciclo de interrupción.

#### **4. MICROCONTROLADOR ( $\mu$ C). TARJETA MINICON\_11 (68HC11 DE MOTOROLA)**

Un microcontrolador es un circuito integrado que agrupa un microprocesador, memoria ROM, memoria RAM, puertos de entrada y puertos de salida para acceder a las memorias. El microprocesador a su vez se compone de un controlador, una unidad aritmético lógica (ALU) y un conjunto de registros; los registros están formados por flip-flops, en ellos podemos almacenar datos o instrucciones. El microprocesador tiene la capacidad de distinguir entre un dato, una instrucción o una dirección; de tal modo que, cuando recibe un dato lo envía a los registros o a la unidad aritmético lógica y cuando recibe una instrucción la envía al controlador. Un microprocesador es una máquina de estado, por lo que requiere de un pulso de reloj para funcionar, un pulso de reloj se utiliza para direccionar, buscar y decodificar y un segundo pulso para ejecutar; a esos cuatro pasos, que se realizan en dos ciclos de reloj, se les conoce como ciclo de máquina. Una instrucción de programa puede utilizar más de un ciclo de máquina, por lo que cada instrucción tiene un valor de ciclos de máquina que requiere para ejecutarse, al conjunto de ciclos de máquina que requiere una instrucción se le conoce como ciclo de instrucción. En la figura 4.1 se muestra la arquitectura del MC68HC11.

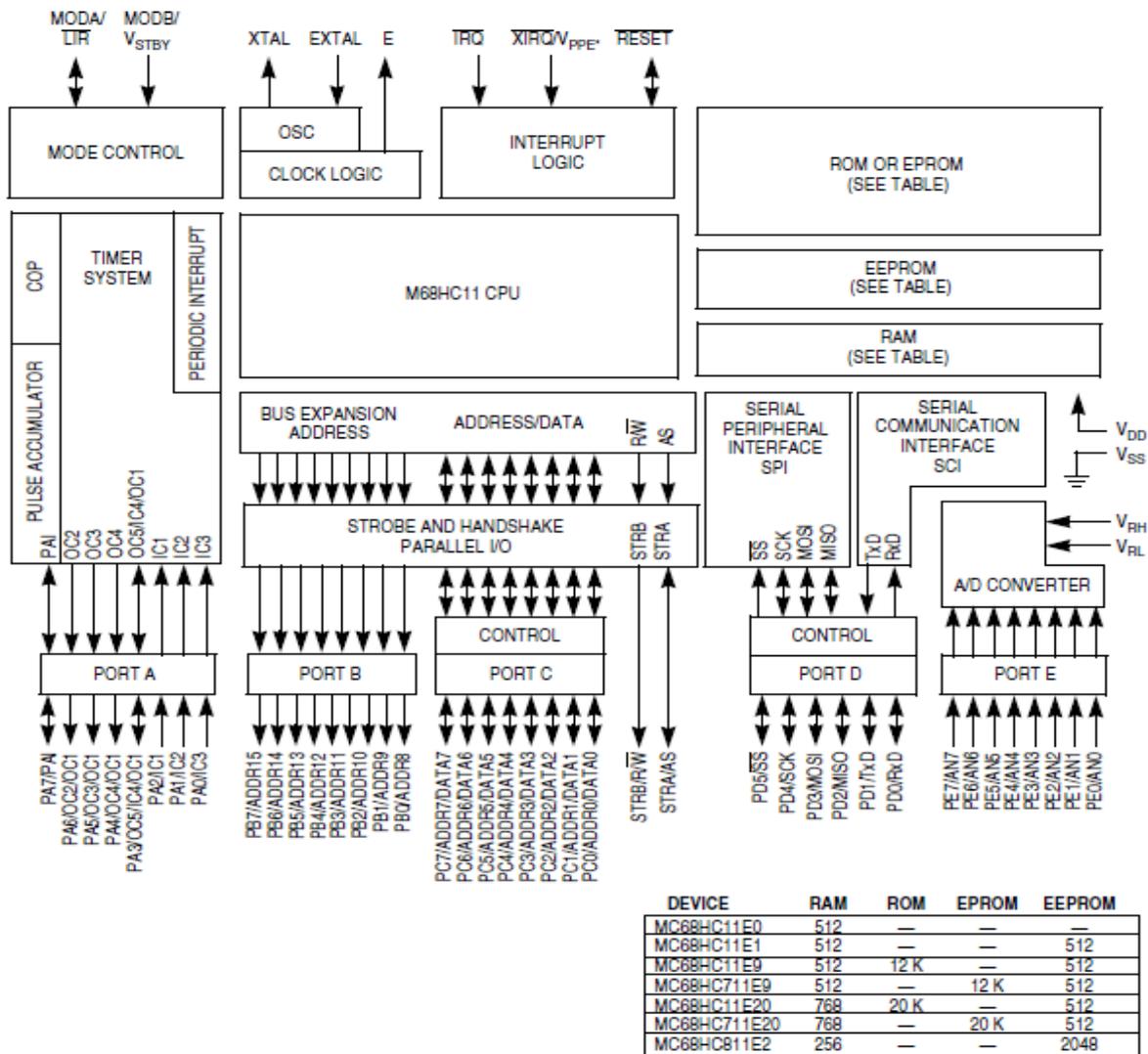


Figura 4. 1 Arquitectura de microprocesadores de la familia MC68HC11E.

#### 4.1 Microcontrolador MC68HC11E1.

Para poder programar el 68HC11 se requieren algunos conocimientos básicos como el modelo de programación, el mapa de memoria, los modos de direccionamiento y las instrucciones; de estas últimas mencionaremos sólo las que se emplearon en el desarrollo de la tesis, pues es muy extensa la lista de instrucciones.

Modelo de programación.- son los registros de que consta el microprocesador MC68HC11.

Registro D de 16 bits dividido en parte alta, acumulador A de 8 bits, y parte baja, acumulador B de 8 bits.

Dos registros de 16 bits cada uno para direccionamiento indexado llamados X e Y.

Puntero de Pila (SP) de 16 bits y el Contador de Programa (PC) también de 16 bits, lo que limita la longitud de un programa a 64Kb, espacio máximo direccionable por el PC del MC68HC11.

Mapa de memoria.- es una representación gráfica de todos los registros que puede acceder el procesador.

En nuestro caso particular el MC68HC11E1 tiene 512 bytes de memoria RAM y 512 bytes de memoria EEPROM. Según el mapa de memoria, la memoria RAM se encuentra en la página cero y su dirección 0 es la dirección 0000 del mapa de memoria. La memoria EEPROM se encuentra en la pagina B6 por lo que su dirección 0 será la dirección B600 del mapa de memoria. En la figura 4.2 se puede observar el mapa de memoria completo.

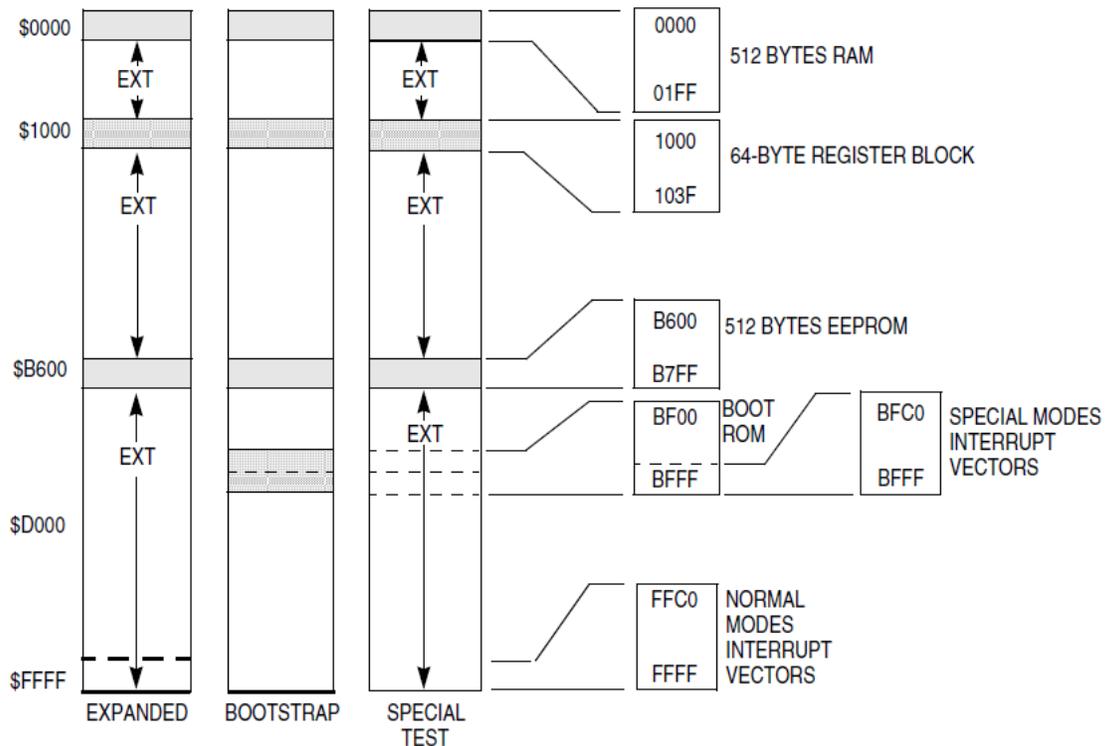


Figura 4. 2 Mapa de Memoria del MC68HC11E.

Modos de direccionamiento: son las maneras de acceder a los datos que están en memoria.

Inmediato: la instrucción contiene el dato al que hacemos referencia y no requerimos acceder a la memoria; el dato puede ser de 1 ó 2 bytes y debe ser precedido por el símbolo #; ej. LDAA #0A.

Extendido: el dato se encuentra en una dirección de memoria, por lo que la dirección se obtiene de la instrucción, la dirección de memoria ocupa 2 bytes y puede tomar valores entre 0000 y FFFF hexadecimal y debe ser precedido por el símbolo \$; ej. LDAA \$1F2A.

Directo: el dato se encuentra en una dirección de memoria comprendido entre 00 y FF hexadecimal, por lo que la dirección de memoria sólo ocupa 1 byte, también debe ser precedido por el símbolo \$; ej. LDAA \$0F.

Indexado: se utilizan los registros índice X o Y y se les suma un desplazamiento de 8 bits, este método se utiliza para acceder a tablas de memoria; ej. LDAB 5,X.

Relativo: sólo se utiliza para instrucciones de bifurcación, indica a la CPU que realice un salto hacia adelante o hacia atrás, tiene signo y utiliza un byte, por lo que los saltos sólo pueden realizarse hacia direcciones de memoria a menos de 128 bytes hacia abajo y a menos de 127 bytes hacia arriba. No hay que expresar o especificar la dirección del salto, de eso se encarga el ensamblador.

Inherente: los operandos se encuentran en un registro específico y no en memoria; el código de la instrucción le indica a la CPU de que registro se trata; ej. INCA incrementa el valor del acumulador A en una unidad, INCX incrementa el valor del índice X en una unidad.

Instrucciones: se expresan con mnemónicos para ayudarnos a comprender lo que estamos programando y hacer más simple reconocerlos y diferenciarlos entre ellos. Pues realmente lo que se carga en el microprocesador son secuencias de unos y ceros.

Para poder explicar algunas de ellas vamos a dividir las en grupos:

Manipular datos.

LDAA \$80 ; carga el dato de la dirección 80 en el acumulador A

LDAB #\$A0 ; carga el dato A0 en el acumulador B

LDX #\$1000 ; carga un dato de 16 bits en el índice X

LDY #\$1F00 ; carga un dato de 16 bits en el índice Y

STAA, STAB ; almacenan el dato de A, B en una dirección de memoria

STD, STX, STY ; almacenan datos de 16 bits en memoria.

DEC ; decrementa en uno el dato y lo deja en la misma localidad de memoria

DECA, DECB ; decrementan en uno lo que hay en A, B y lo dejan en A, B

DEX, DEY ; decrementan en uno lo que hay en X, Y y lo dejan en X, Y

INC ; incrementa en uno el dato y lo deja en la misma localidad de memoria

INCA, INCB, INX, INY; incrementan una localidad y la dejan en el mismo registro

CLR ; limpia una localidad de memoria, pone ceros

CLRA, CLRB ; coloca ceros en A, B

BCLR ; coloca ceros en uno o varios bits de una localidad de memoria

BSET ; coloca unos en uno o varios bits de una localidad de memoria

Aritméticas.

ADDA, ADDB ; suma al acumulador A, B el dato en una localidad y deja el resultado en A, B

ABA, ABX, ABY ; suma A+B, B+X o B+Y respectivamente dejando el resultado en A, X o Y respectivamente.

Prueba de datos.

CMPA, CMPB ; compara el acumulador A, B con un dato y activa las banderas Z o N del registro de estados CCR según: Z=0 (A o B)>Dato; Z=1 (A o B)=Dato; N=1 (A o B)<Dato

Bifurcaciones condicionadas.

BEQ ; A o B = M

BNE ; A o B ≠ M

BHI ; A o B > M

BLS ; A o B ≤ M

BCC ; A o B ≥ M

BCS ; A o B < M

Salto y bifurcaciones.

NOP ; se ejecuta no hace algo

JMP ; salta a una dirección de memoria

JSR ; salta a una subrutina

RTS ; retorna de una subrutina

BSR ; salta a una subrutina adelante

BRA ; salta siempre

BNR ; no salta

BRCLR ; salta si hay ceros

BRSET ; salta si hay unos

Instrucciones de manejo de registros de bandera

CLI ; borra el bit I, máscara de interrupción, del registro CCR

SEI ; pone un uno en el bit I, máscara de interrupción, del registro CCR

Directivas del ensamblador: también llamados pseudo-operandos, son instrucciones que indican al ensamblador realizar acciones específicas, las que utilizamos en el presente trabajo son:

EQU (EQUate symbol to a value); indica al ensamblador agregar una etiqueta en la tabla de símbolos e igualarla a un valor dado. Hay que tener cuidado de no usar etiquetas definidas o exclusivas del ensamblador; ej. PORTA EQU \$00 ; nos permite usar la etiqueta PORTA en el direccionamiento indexado en lugar de \$00, es útil cuando se refieren

diferentes registros dentro del programa y es más simple recordarlos por etiqueta que por dirección o posición en el mapa de memoria.

FCB (From Constant Byte); indica al ensamblador almacenar los operandos en bytes sucesivos, los operandos pueden ser de ocho bits o caracteres constantes simples, cada valor es separado por una coma. Se emplea para manejo de tablas; ej. TABLA FCB \$28, \$12, \$24, \$11.

ORG (set location counter ORiGin); permite al usuario establecer la posición del contador de programa en un valor específico. Se usa comúnmente para asegurar que el código es ensamblado en la posición correcta del mapa de memoria; ej. Inicio ORG \$0000; coloca el contador en la posición 0000 que es el inicio de la RAM.

RMB (Reserve Memory Byte); permite al usuario reservar direcciones de memoria y asociar una etiqueta a esas direcciones para variables, tablas, etc. ej. NUMPASO RMB 1.

## **4.2 Registros empleados del MC68HC11.**

El microprocesador MC68HC11 tiene registros específicos en donde se pueden configurar diferentes modos de operación del microcontrolador como son: los puertos de salida, entrada y convertidores analógico digital, el reloj interno, entre muchas otras características. Para comprender más a detalle el presente trabajo, vamos a mencionar sólo los que se usan en el desarrollo de esta tesis.

PORTA: el puerto A se encuentra en el registro con la dirección \$1000; es un puerto de entrada, tiene la particularidad de poder usar, si así se configuran, el bit 7 y el bit 3 como salidas.

PORTB: el puerto B se encuentra en la dirección \$1004 es un puerto de salida, y en el caso de la tarjeta Minicon11 tiene LEDs auxiliares para poder ver el estado de la salida en todo momento.

TCNT: el Timer Counter Register se encuentra en la dirección \$100E, es un registro de 16 bits, de sólo lectura que contiene un contador de 16 bits del temporizador pre escalado, según los valores mostrados en la figura 4.3. Este contador y el temporizador son independiente del reloj del sistema.

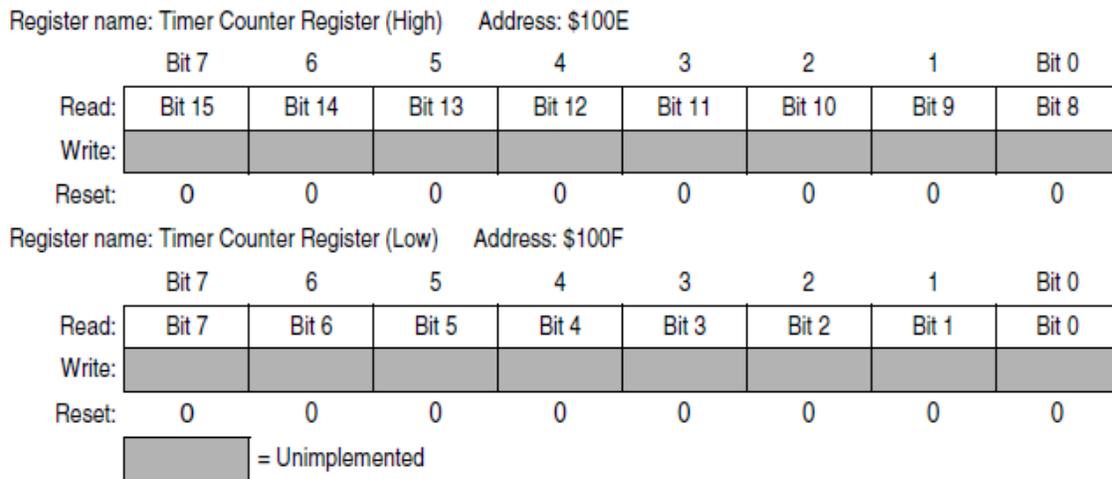


Figura 4. 3 Registro de contador de temporizador.

TOC2: el Timer Output Compare Register es un registro de escritura lectura de 16 bits y se encuentra en la dirección \$1018 la parte alta y \$1019 la parte baja; sirve para almacenar un valor de hasta 16 bits que será comparado con el temporizador.

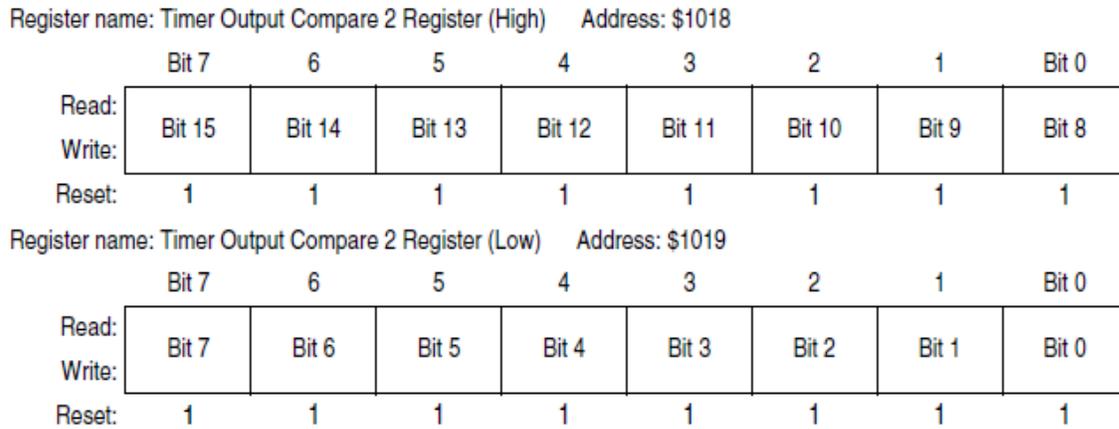


Figura 4. 4 Registro comparador de salida del temporizador.

TCTL1: en el Timer Control Register 1 los bits de este registro especifican las acciones a tomar como resultado de una comparación exitosa de los OC1 al OC4 y IC4/OC5; en nuestro caso particular, TOC2.

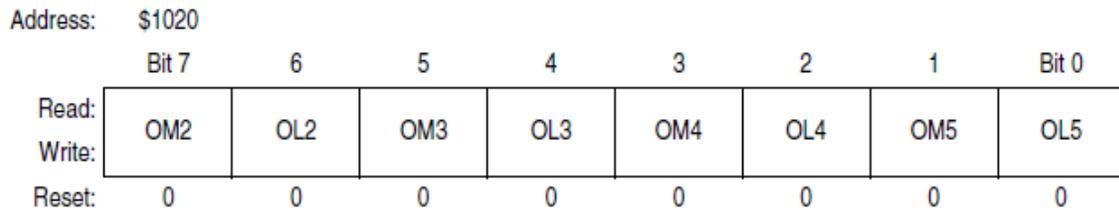


Figura 4. 5 Registro 1 de control de temporizador.

OMx	OLx	Action Taken on Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to 0
1	1	Set OCx output line to 1

Figura 4.6 Acciones a tomar para una comparación exitosa de TOCx.

TMSK1: el Timer Interrupt Mask 1 Register, es un registro de 8 bits que permite habilitar o inhibir las interrupciones de las comparaciones de salida OC y capturas de entrada IC del temporizador. Si un bit del registro es puesto a uno cuando el bit correspondiente del registro TFLG1 es uno, se solicita una secuencia de Interrupción de software SWI.

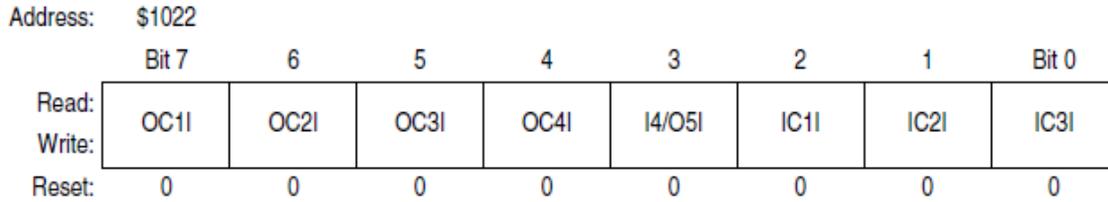


Figura 4. 7 Registro 1 máscara de interrupción del temporizador.

TMSK2: el Timer Interrupt Mask 2 Register, es un registro de 8 bits que sirve para habilitar o deshabilitar interrupciones de tiempo real o desbordamiento del temporizador; también incluye dos bits de control de pre escala del temporizador. Los Bits 1 y 0 del TMSK2, PR1 y PR0, respectivamente, controlan el pre escalado del temporizador según la figura 4.8

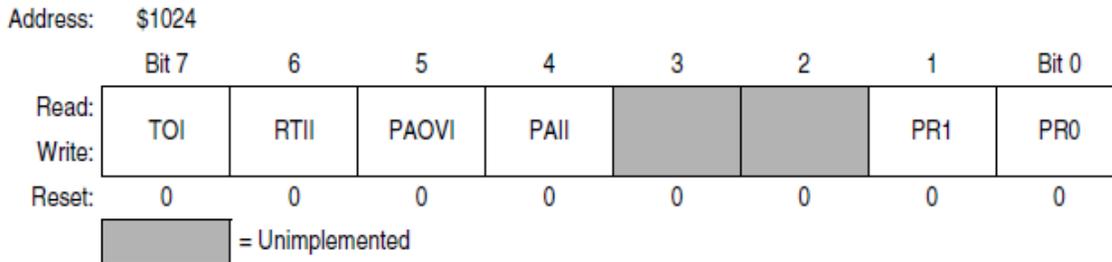


Figura 4. 8 Registro 2 máscara de interrupción del temporizador.

Control Bits PR1, PR0	XTAL Frequencies			
	4.0 MHz	8.0 MHz	12.0 MHz	Other Rates
	1.0 MHz	2.0 MHz	3.0 MHz	(E)
	1000 ns	500 ns	333 ns	(1/E)
	Main Timer Count Rates			
0 0 1 count — overflow —	1000 ns 65.536 ms	500 ns 32.768 ms	333 ns 21.845 ms	(E/1) (E/2 <sup>16</sup> )
0 1 1 count — overflow —	4.0 μs 262.14 ms	2.0 μs 131.07 ms	1.333 μs 87.381 ms	(E/4) (E/2 <sup>18</sup> )
1 0 1 count — overflow —	8.0 μs 524.29 ms	4.0 μs 262.14 ms	2.667 μs 174.76 ms	(E/8) (E/2 <sup>19</sup> )
1 1 1 count — overflow —	16.0 μs 1.049 s	8.0 μs 524.29 ms	5.333 μs 349.52 ms	(E/16) (E/2 <sup>20</sup> )

Figura 4.9 Parámetros de pre escalado del temporizador.

En nuestro caso particular con un periodo de 30h la interrupción se va a llevar cada  $48 \times 8 \mu s = 384 \mu s$ .

TFLG1: el Timer Interrupt Flag Register 1, es un registro de 8 bits que se encuentra en la localidad \$1023, los bits de este registro indican cuando ha ocurrido un evento del temporizador y acoplados con los bits del TMSK1 permiten al subsistema de temporizador operar ya sea monitoreando o interrumpiendo al sistema.

Address: \$1023

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
Write:								
Reset:	0	0	0	0	0	0	0	0

Figura 4. 10 Registro 1 bandera de interrupción de temporizador.

TFLG2: el Timer Interrupt Flag Register 2, es un registro de 8 bits, se encuentra en la localidad de memoria \$1025; los bits de este registro indican cuando se han producido ciertos eventos del temporizador. Los bits de este registro están acoplados a los bits del TMSK2.

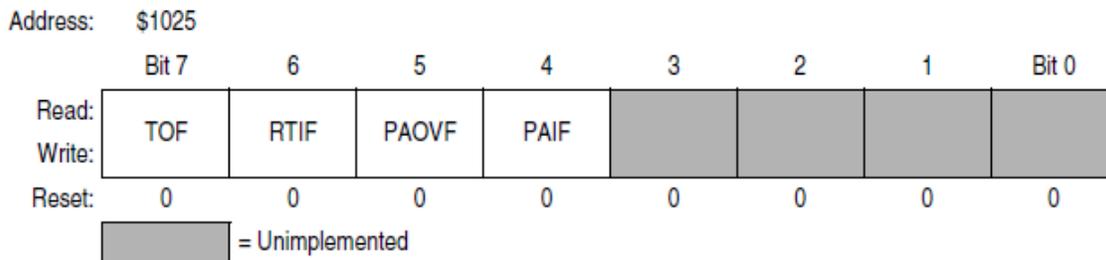


Figura 4. 11 Registro 2 bandera de interrupción de temporizador.

### 4.3 La tarjeta MINICON\_11B

El microcontrolador MC68HC11 de Motorola requiere de algunos elementos periféricos básicos para su funcionamiento, comunicación y programación. En el desarrollo de este trabajo se le utiliza con la tarjeta de desarrollo MINICON\_11 desarrollada por el Ing. Antonio Salva Calleja, profesor de esta facultad.

En este capítulo haremos una breve descripción de la tarjeta MINICON\_11 del microprocesador y el software de interfaz PUMA11, también desarrollado por el Ing. Salva.

La tarjeta MINICON\_11 emplea un microprocesador MC68HC11 de Motorola y algunos otros componentes periféricos para comunicarse con una computadora por medio de un puerto serie, para ser programado; cuenta con diferentes puertos de entrada y salida para recibir información, procesar y enviar información basada en el código de programa previamente cargado.

En la figura 4.12 se muestra la tarjeta MINICON11 y sus componentes.

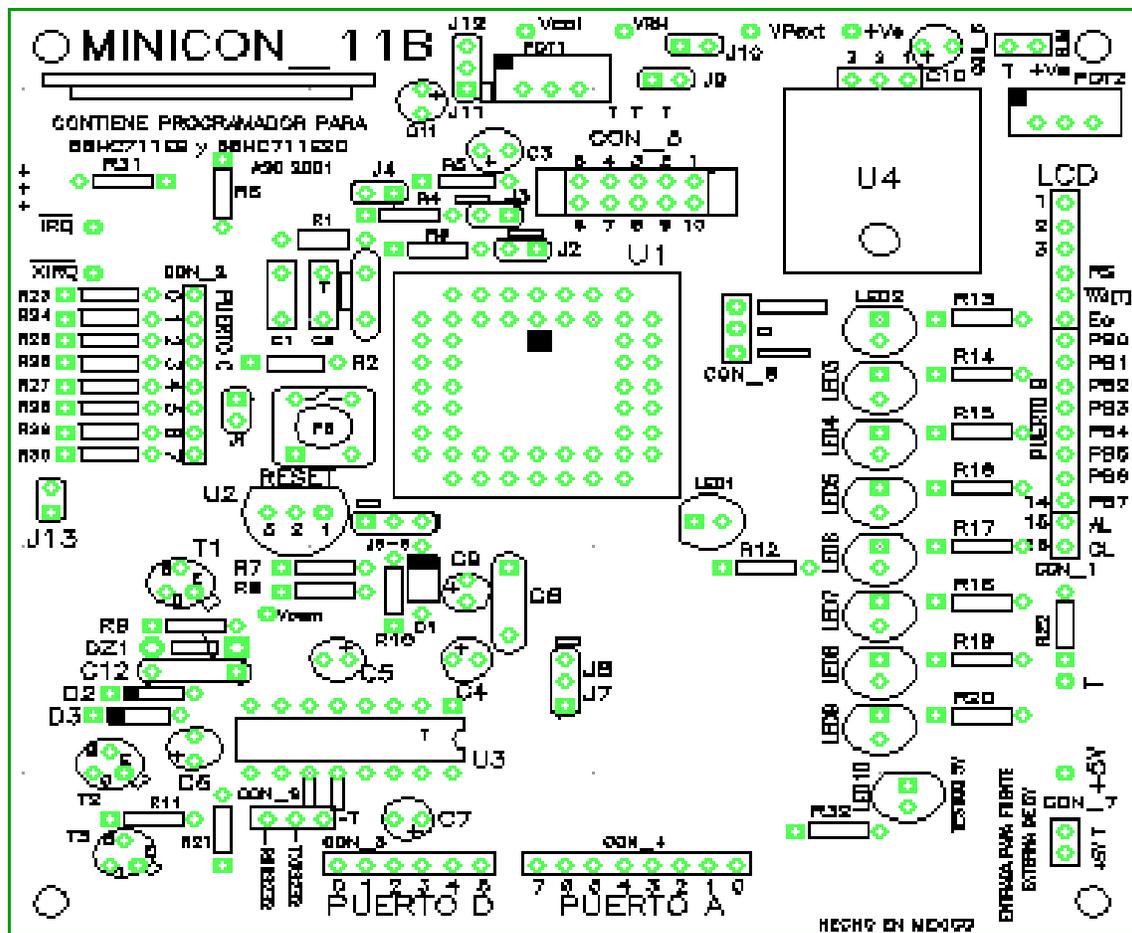


Figura 4. 12 Tarjeta MINICON\_11 y sus componentes.

En la figura 4.13 se muestra una foto real de la tarjeta MINICON\_11 con todos sus componentes y conectores.

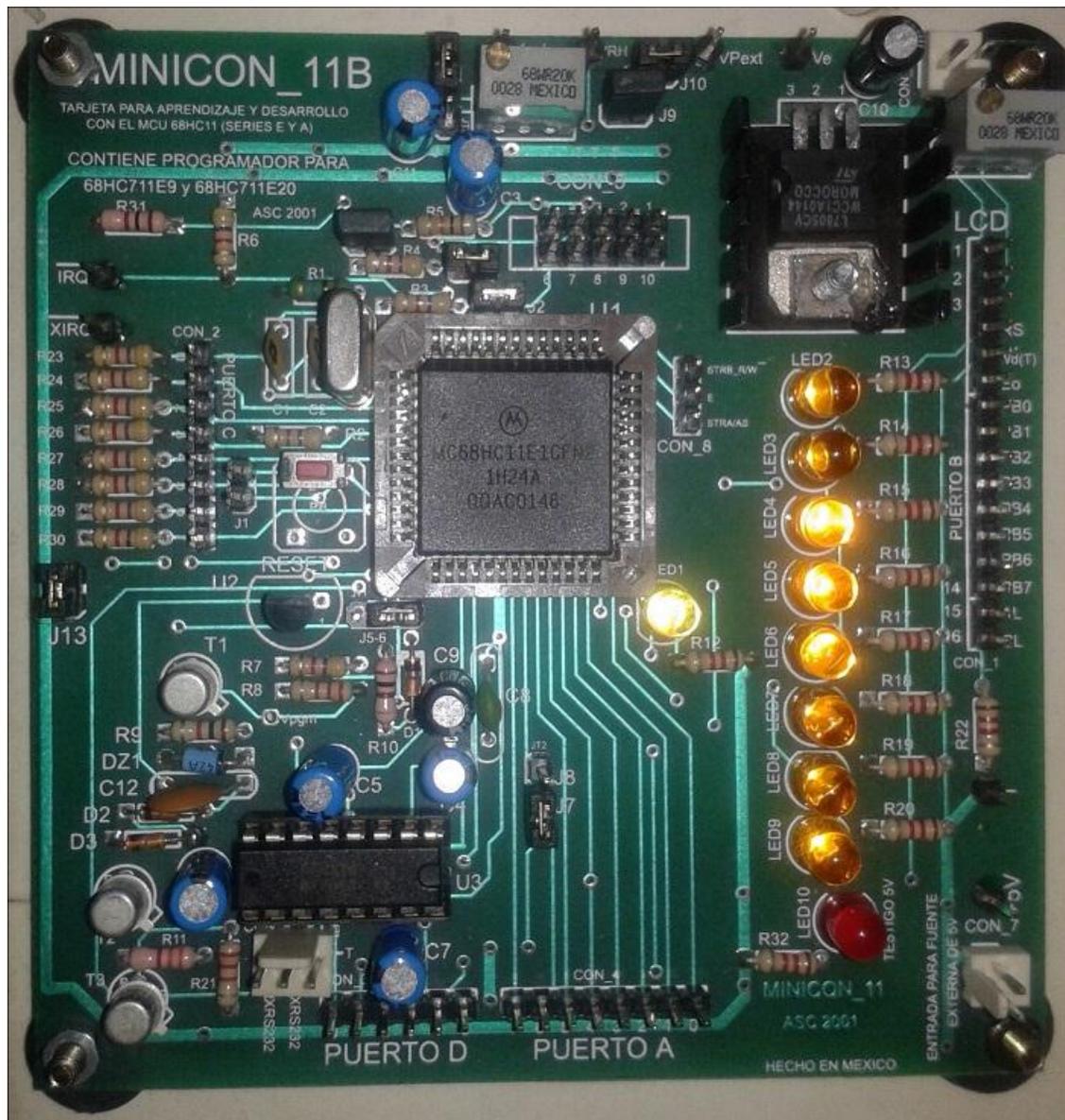


Figura 4. 13 Tarjeta MINICON\_11.

Al centro de la tarjeta se observa el microcontrolador MC68HC11E1 de Motorola, cerca de la esquina superior izquierda del microcontrolador se observa el cristal de 8MHz, en la esquina superior derecha de la tarjeta esta un circuito L7805, regulador de voltaje de 5V y en el cuadrante inferior izquierdo se encuentra el circuito de comunicación serial MAX232 como interfaz con una computadora externa.

Además de los componentes antes mencionados, existen en la tarjeta los conectores CON\_1 al CON\_8 que permiten ingresar o tomar señales hacia o desde el microcontrolador para ser usadas con circuitos externos; los detalles de cada uno se explican a continuación.

En la figura 4.14 se muestra el detalle del conector CON\_1; el cual tiene las siguientes características: los pines 1 y 2 son tierra y 5V respectivamente, el pin 3 es para ajuste de contraste del LCD (externo), los pines 4, 5 y 6 son para RS, control lectura escritura y habilitación de LDC respectivamente, los pines 15 y 16 son para el ánodo y cátodo, respectivamente, del LED de retroiluminación del LCD. Los pines 7 al 13 corresponden a las salidas B0 al B7 del puerto B y permiten la salida de datos D0 al D7 de un display externo.

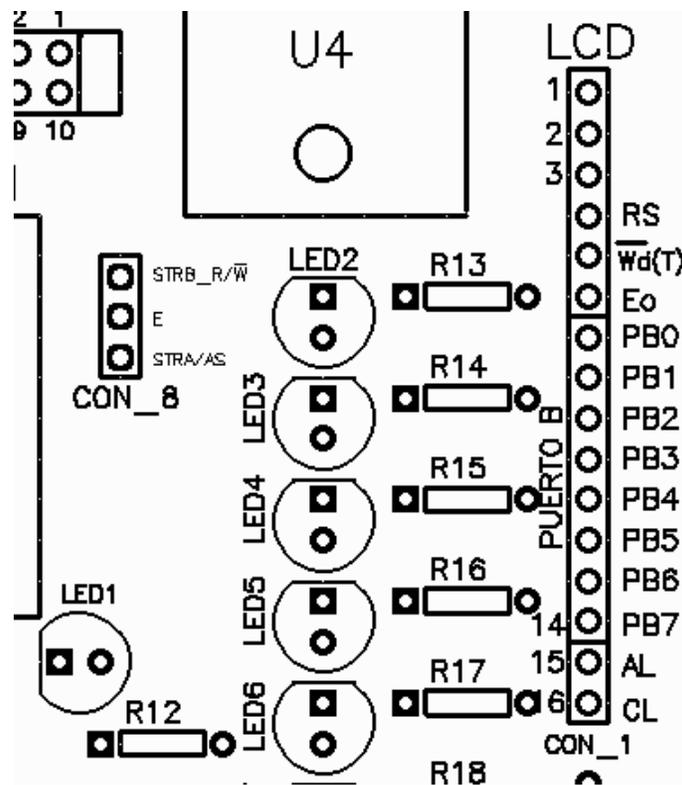


Figura 4. 14 Conector CON\_1 (Puerto B y control LCD).

En la figura 4.15 se muestra el detalle del conector CON\_2, este conector es en su totalidad el puerto C del C0 al C7 según se muestra en la imagen.

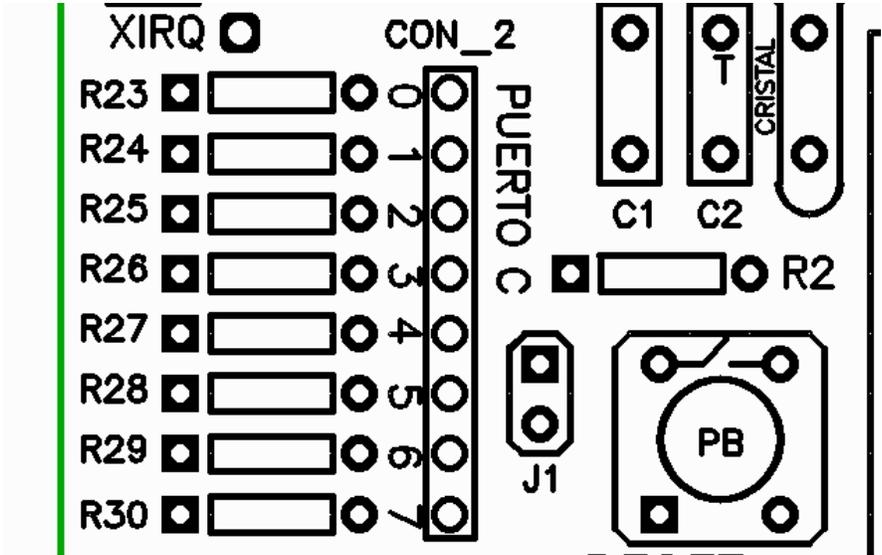


Figura 4. 15 CON\_2 (Puerto C).

El conector CON\_3 corresponde al puerto D, del D0 al D5 y se muestra el detalle en la figura 4.16.

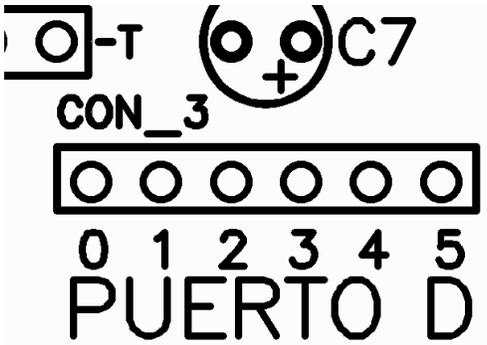


Figura 4. 16 CON\_3 (Puerto D).

El puerto A puede configurarse como entrada o salida tiene ocho pines del A0 al A7 y se encuentran en el conector CON\_4, según se muestra en la figura 4.17.

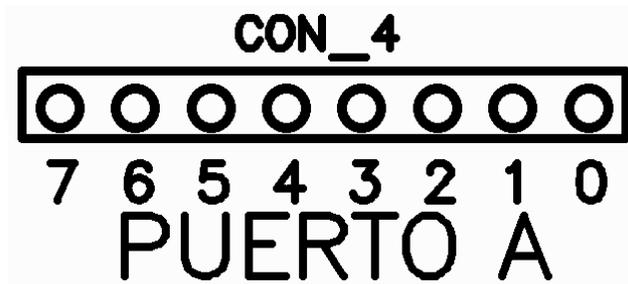


Figura 4. 17 (Puerto A).

El conector CON\_5 contiene los pines del puerto E con la siguiente relación: postes 1 al 4, E0 al E3, respectivamente; postes 7, 8, 9 y 10 corresponden con E7, E6, E5 y E4 en estricto orden; el poste 6 es tierra de señal y el poste 5 es VRH, voltaje de referencia, pues los pines de este puerto pueden usarse como entradas analógicas. El detalle se muestra en la figura 4.18.

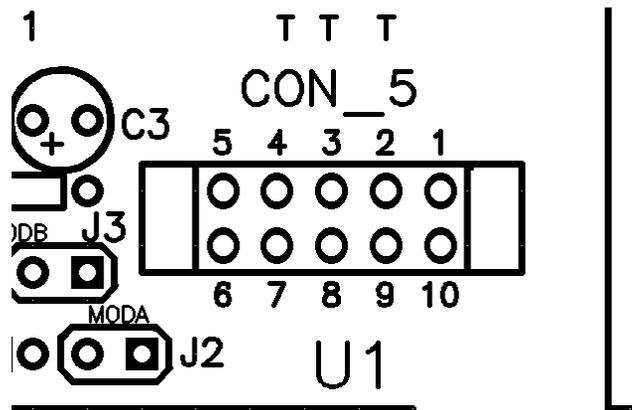


Figura 4. 18 (Puerto E).

La alimentación de la tarjeta puede ser por medio de un eliminador de baterías, con un rango de 7V mínimo a 19V máximo, conectado en el CON\_6, respetando la polaridad; T=0V y +Ve = Voltaje positivo del eliminador. El detalle del CON\_6 puede verse en la figura 4.19.

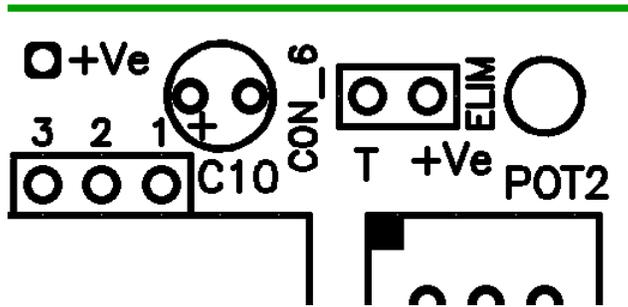


Figura 4. 19 CON\_6 Conexión de eliminador de baterías.

También es posible alimentar la tarjeta MINICON\_11, directamente, con una fuente regulada de 5V, conectada en el CON\_7 y respetando la polaridad indicada en el detalle de la figura 4.20.

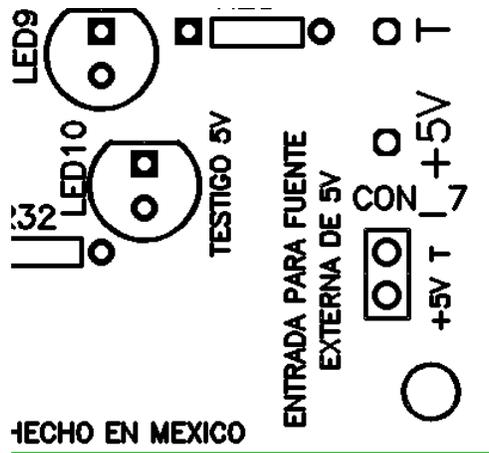


Figura 4. 20 CON\_7 Entrada para fuente externa 5V.

El detalle del conector CON\_8 se muestra en la figura 4.21.

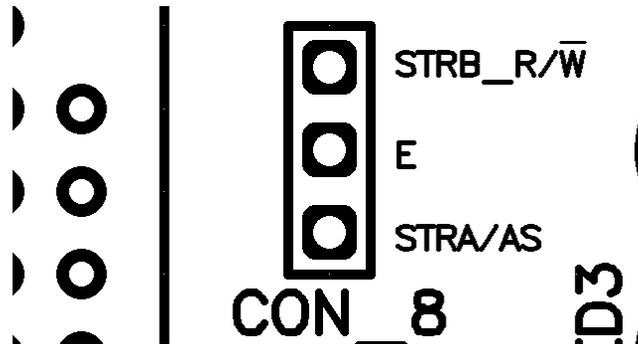


Figura 4.21 CON\_8 Lectura / Escritura.

El enlace de la tarjeta MINICON\_11 con una computadora externa se realiza por medio del MAX232 cuyas terminales TX y RX se encuentran en el CON\_9 según el detalle que se muestra en la figura 4.22.

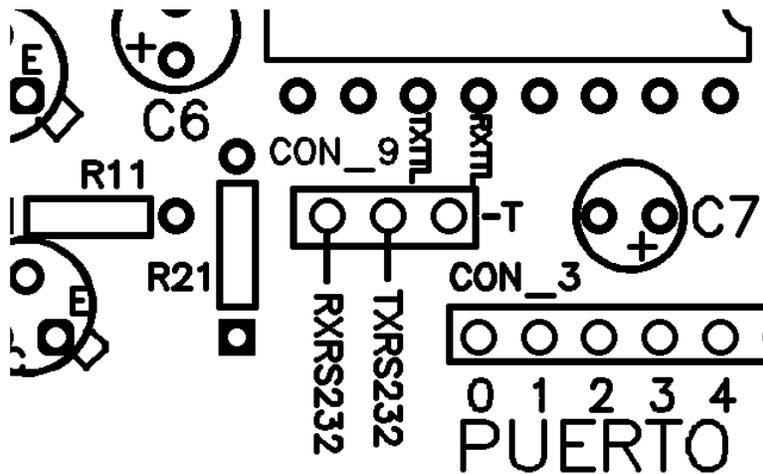


Figura 4. 22 CON\_9 Transmisión / recepción niveles RS232.

Para poder llevar a cabo el enlace de la tarjeta MINICON\_11 a la computadora externa se requiere además de un cable DB9 para ser conectada a un puerto serie RS232, el detalle

de la configuración del cable DB9 y el conector para la tarjeta MINICON\_11 se muestra en la figura 4.23.

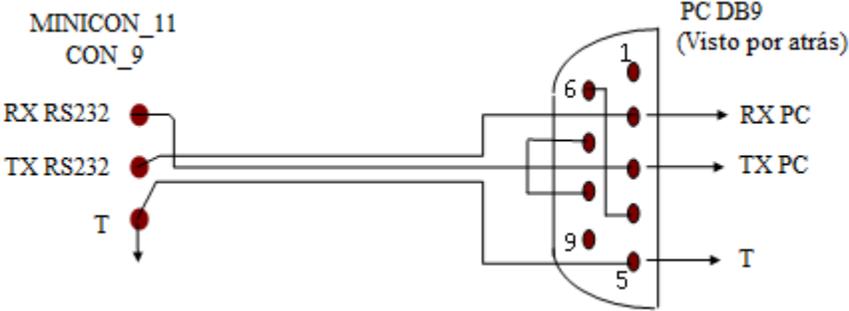


Figura 4. 23 Cable DB9 PC/MINICON\_11.

## **5. PROCESADOR DIGITAL DE SEÑALES (DSP). TARJETA TMS320C50 DE TEXAS INSTRUMENTS**

### **5.1 Características principales y ventajas del DSP TMS320C50.**

- El DSP está fabricado con tecnología CMOS estática integrada, su arquitectura esta basada en su antecesor el 'C25.
- Las bases de la flexibilidad de operación y velocidad de los DSP's son la combinación de una avanzada arquitectura Harvard (que tiene bus de memoria de programa y bus de memoria de datos separados), periféricos incluidos en el chip, más memoria integrada, y un conjunto de instrucciones altamente especializadas.
- Este dispositivo puede ejecutar hasta 28 millones de instrucciones por segundo.
- Un mejor diseño de arquitectura para un mejor desempeño.
- Diseño de arquitectura modular.
- Tecnología de procesamiento avanzada de circuito integrado.
- Compatibilidad de código fuente con otros DSP's de generaciones anteriores.
- Un amplio conjunto de instrucciones para la ejecución rápida de algoritmos y operación de lenguaje de alto nivel optimizado.
- Nueva técnica de diseño estático para minimización de consumo de potencia.
- Tiempo de ejecución de una instrucción de punto fijo que es de 10 us ciclo (35 a 50 ns).
- Operación de memoria basada en RAM.
- Operación de memoria basada en ROM.
- 9K x 16 bit de memoria RAM para programa y datos integrada.
- 2K x 16 bit de sección para ROM, 1056 x 16 bit de RAM para datos de acceso dual.

- 224K x 16 bit máximo de espacio de memoria externa direccionable, 64K de programa, 64K de datos, 64K de I/O (puertos paralelos de entrada y salida), 32K global).
- Unidad aritmética lógica (ALU) de 16 bit, acumulador (ACC) de 32 bit, y buffer acumulador (ACCB) de 32 bit.
- Unidad lógica paralela de 16 bit.
- Multiplicador paralelo de 16 x 16 bit con capacidad de 32 bit para el producto. Instrucciones de multiplicar/acumular en un solo ciclo.
- Ocho registros auxiliares con una unidad aritmética dedicada para direccionamiento indirecto.
- Once registros de contexto conmutado (registros fantasma) para almacenamiento estratégico controlados por el CPU durante una rutina de servicio de interrupción.
- Hardware de pila de 8 niveles. Corrimiento de datos de 0 a 16 bit a la izquierda y derecha, y un registro de corrimiento de datos de 64 bit que incrementa.
- Dos buffers circulares direccionados indirectamente para direccionamiento circular.
- Una sola instrucción para operaciones de repetir y repetir bloque para código de programa.
- Instrucciones para mover bloques de memoria para administración entre memoria de programa y datos.
- Puerto serial síncrono full-duplex para comunicación directa entre el 'C50 y otro dispositivo serial.
- Puerto serial con acceso múltiple por división en tiempo (TDM).
- Intervalo de timer con periodo, control y registro contador para inicio, alto y reset del software. 64K puertos paralelos de los cuales 16 están mapeados a memoria.
- Operación de espera extendida para concordar con DMA externo.
- Modo de direccionamiento indexado.

- Bit reservado en modo de direccionamiento indexado para radixF-F2T s.
- Opción de reloj dividido por uno.
- Generador de reloj integrado en el chip.
- Tecnología CMOS estática de 5V. Con dos modos de baja potencia.
- El propósito de los procesadores digitales de señales como el TMS320C50 es proporcionar un menor costo en la implementación de un sistema y la reducción del espacio requerido, al incorporar en un procesador la memoria y periféricos.
- Además de que proporciona un desempeño en tiempo real de aplicaciones como procesamiento de voz y filtrado de señales analógicas.
- Permiten una efectiva implementación en algoritmos de codificación y decodificación.
- Otra de las aplicaciones prácticas es el diseño de filtros adaptivos y la transformada de Fourier en las que se requiere que los programas realicen operaciones recursivas que el DSP ejecuta en tiempos reducidos gracias al bit reservado de modo de direccionamiento indexado.

## 5.2 Arquitectura del DSP TMS320C50.

Un DSP al igual que cualquier otro procesador contiene un registro donde se van a almacenar todos los resultados de las operaciones y que a la vez puede servir como fuente y como destino, es decir, se puede almacenar allí un dato que se va a utilizar en alguna operación y después de que se haya ejecutado el resultado se entrega en ese mismo registro. Dicho registro es el acumulador.

También existen otros 8 registros denominados registros auxiliares que se encargan de manipular las direcciones involucradas en las operaciones. Para entender su importante uso veamos el siguiente ejemplo: se desea sacar un valor de la memoria interna del DSP hacia alguno de los puertos se utiliza la instrucción OUT, digamos que el dato que queremos sacar se encuentra en la posición 2FFFh, no se puede poner directamente este valor en la instrucción sino hay que cargar un registro auxiliar con esta dirección y luego incluir este registro en la instrucción, solo así se reconocerá la dirección. La mayoría de las instrucciones funciona bajo este contexto por lo que es casi inherente el uso de los registros auxiliares.

Para ejecutar todas las instrucciones que implican una operación matemática como la suma, la resta o las operaciones lógicas como AND, OR, etc. en el caso de un procesador digital de señales, como el presente, estas operaciones se ejecutan en unidades separadas, así, las operaciones lógicas se realizan sobre la unidad lógica paralela (PLU) y las operaciones matemáticas se ejecutan en una unidad aritmética de propósito general (ALU) que también puede realizar operaciones booleanas cuando estas no se realizan sobre el acumulador.

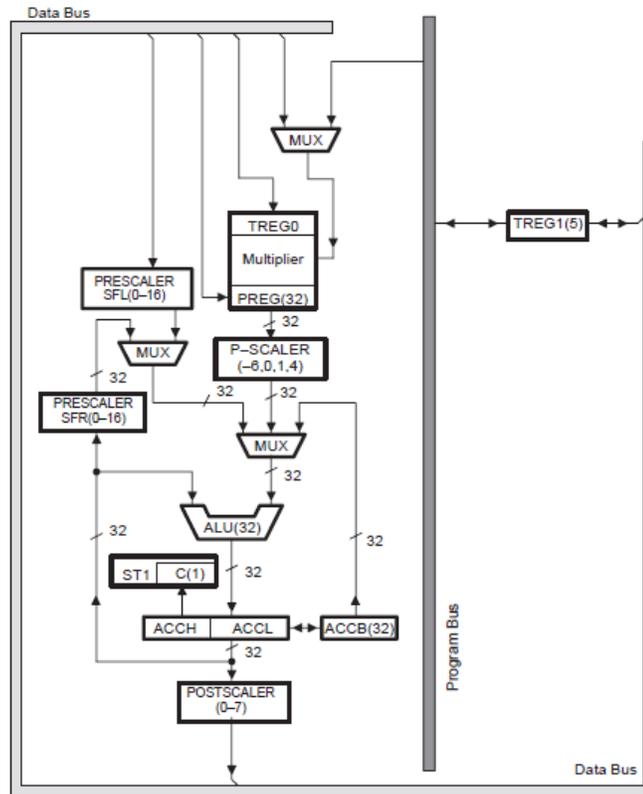
El DSP también cuenta con un multiplicador, que, como su nombre lo indica, realiza todas las multiplicaciones, esto es, porque muchos análisis de señales en frecuencia, como las transformadas de Fourier, hacen uso frecuente de esta operación y, cómo es sabido, las señales de voz es uno de los objetivos principales del DSP.

Esta división de las tareas del DSP lo convierten en un procesador muy eficiente porque se pueden estar ejecutando más de una instrucción al mismo tiempo, por ejemplo, podemos ordenarle al DSP, que haga una suma y si la siguiente instrucción se trata de alguna operación lógica, ésta se empieza a ejecutar en otra unidad antes de que terminen los ciclos de reloj que se invierten en una suma y, a todo esto, hay que añadir que su arquitectura es de tipo Harvard, es decir, tiene el bus de datos y el bus de direcciones físicamente separados por lo que, al mismo tiempo, se pueden estar manejando datos y direcciones. Esto facilita la ejecución paralela de dos instrucciones.

Otra característica exclusiva del 'C50 son los corrimientos. Cuando una instrucción carga un valor, éste se le puede recorrer hacia la izquierda determinado número de bits. En un programa que emplee filtros digitales esto puede ser de gran utilidad, además que el procesador no consume ciclos de reloj de más por hacer esta operación ya que cuenta con una unidad especializada en ello, el Scaling Shifter para ejecutar los corrimientos en paralelo a la ejecución de otras instrucciones.

### **5.2.1 CALU.**

El conjunto formado por el multiplicador, el acumulador la ALU y otros tres registros llamados registro de producto (PREG), el buffer del acumulador (ACCB) y el cambiador de escalamiento (Scaling shifer) forman la Unidad Central Aritmética Lógica (CALU) donde se procesan en general todas las instrucciones.



Notes: All registers and data lines are 16-bits wide unless otherwise specified.

Figura 5.1 CALU.

Scaling Shifter: es el encargado de hacer los corrimientos.

ACCB: registro para almacenar temporalmente el contenido del ACC.

PREG: registro donde se almacena temporalmente el producto de la multiplicación.

### 5.2.2 Memoria.

El DSP C50 puede direccionar hasta 0FFFFh localidades, todo este espacio lo constituye el espacio de memoria, de la localidad 0000h - 2BFFh se designa como la memoria interna del microprocesador y de 2C00h- FFFFh como la memoria externa; el total del espacio de

memoria se distribuye de dos formas de acuerdo al nivel de voltaje que haya en la terminal MP/MC.

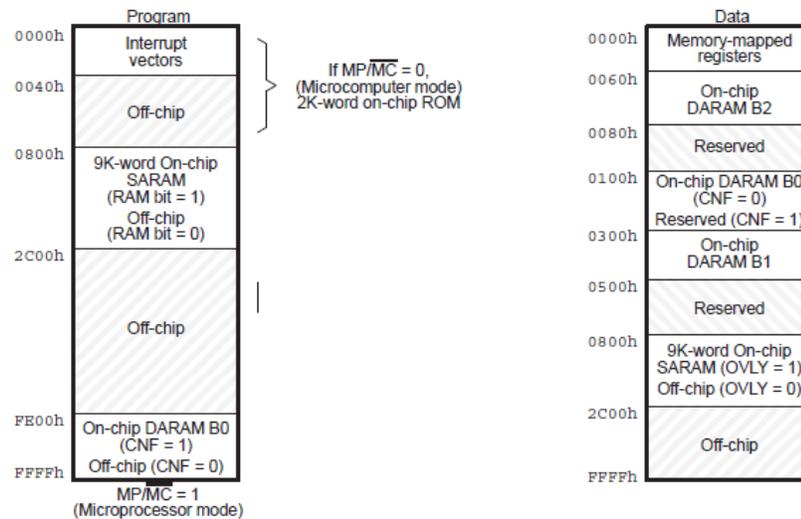


Figura 5.2 Mapa de memoria del C50.

Por default el procesador inicia con esta configuración puesto que no hay ningún nivel de voltaje en MP/" cuando se activa el RESET. Como se aprecia en la figura 5.1 hay que tener cuidado de direccionar todos las instrucciones y datos del programa hacia la SRAM para que no haya ningún conflicto de invadir algún sector reservado del DSP o de escribir en la memoria externa a menos que se tenga una Interface de acceso a la memoria externa. También se debe tener cuidado de poner un espacio suficiente entre las localidades del programa y la de los datos para que estos dos no se traslapen.

Cuando MP/"=1 la distribución de la memoria es igual sólo que las interrupciones se localizan en la memoria externa y el espacio que queda en la memoria interna se asigna a la localidad FE00h - FFFFh como una DRAM interna.

En el primer bloque (0h - 2Fh) se localizan todos los recursos del microprocesador, es decir todos aquellos registros que estén mapeados en memoria y las atenciones a las interrupciones. El tercer bloque (800h - 2BFFh) se destina a almacenar el código del programa y los datos del mismo.

### 5.2.3 Registros internos.

Algunos de estos registros estan mapeados en memoria y otros no. Sirven para almacenar o direccionar datos que auxilian en su ejecución al conjunto de instrucciones.

Estos son el tipo de registros que se utilizaron en el presente proyecto de tesis.

### **5.2.3.1 Registros ubicados en hardware.**

#### ACC (Acumulador)

Un registro de 32 bits que se divide en dos de 16 bits el ACCH y ACCL, sólo el ACCL y el ACC actúan como registros fuente y destino.

#### ARP (Apuntador de registro auxiliar)

Un registro de 3 bits. Es el que determina cuál de los ocho registros auxiliares se está utilizando, el número del registro en activo corresponde al equivalente en binario almacenado en el ARP. La instrucción para actualizar un nuevo registro:

MAR \*, AR[# de registro].

#### PC (Contador de programa)

El contador de programa de 16 bits almacena la dirección en memoria de la próxima instrucción y sirve para ejecutar secuencialmente un programa.

#### TC (Bandera de Test/Control)

Es un bit que reside en el registro de control ST1, su comportamiento difiere según la instrucción que se está usando, cuando se trata de la instrucción CMPR que compara el contenido de una localidad de memoria con un número este bit se activa si ambos números son iguales.

#### DP (Apuntador de página)

Es un registro de nueve bits que contiene la dirección de la página que se está utilizando actualmente. Cada página contiene 128 palabras, resultando todo el espacio de la memoria en 512 páginas direccionables.

#### STO, ST1, PMST (Registros de estado)

Son tres registros de estado que contienen el estado actual del microprocesador y los bits de control.

#### STACK (Pila)

Una pila de 8\*16 bits para almacenar el contenido del PC durante las interrupciones y las llamadas.

### **5.2.3.2 Registros mapeados en memoria.**

BRCR (Localidad 9)

Un registro de 16 bits mapeado en memoria en donde se almacena la cantidad de veces que se requiere repetir un bloque de instrucciones. Se usa junto con la instrucción RPTB.

ARO - AR7 (Localidades 10 h - 17h respectivamente)

Son los 8 registros auxiliares con los que las instrucciones accesan a la memoria en el direccionamiento indirecto (el más usado). También se le conoce como Archivo de registros auxiliares. En ocasiones se les utiliza para almacenar datos temporalmente.

DXR (Localidad 21 h)

Registro de transmisión de datos, cualquier valor que se coloque en este registro saldrá automáticamente por el puerto serial del DSP.

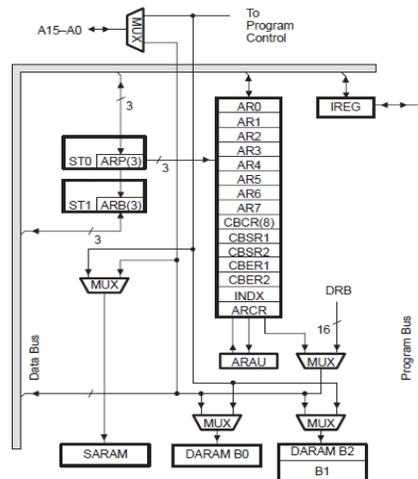
PA0 - PA15 (localidades 50h- 5F respectivamente)

Son los puertos de entrada salida. En realidad se puede usar cualquier localidad libre de todo el espacio de entrada salida para extraer o introducir datos pero estos ya están predeterminados.

### **5.2.4 ARAU.**

La ARAU es una unidad capaz de hacer operaciones sobre los registros auxiliares como sumar el contenido del registro INDX a cualquier registro auxiliar y hacer comparaciones sin la ayuda de la ALU lo que indica que puede ejecutar sus operaciones al mismo tiempo que la ALU ejecuta las suyas.

En figura 5.3 se observa como interactúan los registros auxiliares con la Unidad Aritmética de Registros Auxiliares (ARAU). Por lo general s610 se actúa sobre esta unidad y sobre el Apuntador de Registro Auxiliar, el resto entra en acción cuando se hacen uso de operaciones especiales como el direccionamiento bajo la columna de una matriz.



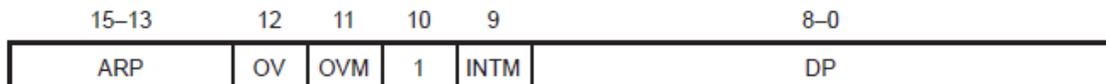
Notes: All registers and data lines are 16-bits wide unless otherwise specified.

Figura 5.3 ARAU.

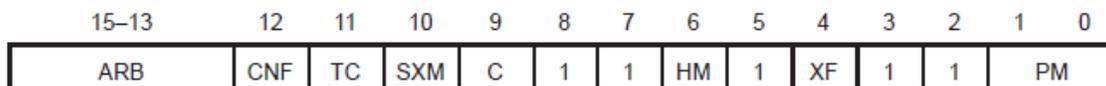
### 5.2.5 Registros de estado.

Estos son los cuatro registros de control y estado que contiene el DSP C50

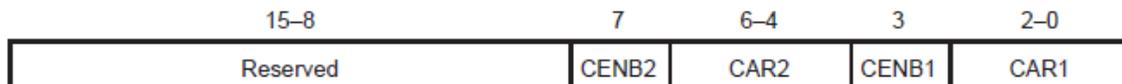
ST0



ST1



CBCR



PMST

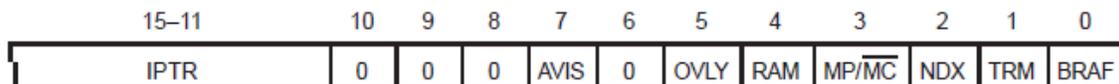


Figura 5.4 Registros de estado del DSP C50.

Los principales bits de control que se ocuparon en esta tesis se mencionan en seguida.

ARP - Apuntador de registro auxiliar.  
C - Bandera de acarreo.  
DP - Apuntador de página de la memoria de datos.  
OV - Bandera de sobreflujo.  
TC - Bandera de Test/Control.

## 5.2.6.- Modos de direccionamiento

Los modos de direccionamiento son las formas en que una instrucción puede acceder hacia una localidad de memoria o un registro. Dependiendo de la aplicación uno puede elegir el direccionamiento que más convenga para simplificar el código y hacerlo más eficiente. En nuestro caso siempre utilizamos el direccionamiento indirecto por que es al que se adaptan mejor el conjunto de instrucciones utilizadas, con lo cual reducimos el tamaño del código y aumenta la velocidad de nuestra aplicación. Dicho direccionamiento es muy sencillo, se trata de que todas las instrucciones que soliciten escribir o leer de la memoria lo hagan a través de los registros auxiliares, es decir, si quisiéramos cargar el acumulador con el contenido de la localidad de memoria 1000h tendríamos que cargar un registro auxiliar con el número 1000h y después actualizar al registro ARP para que la instrucción que ejecute la carga del acumulador lo tome en cuenta, esto se ve en el siguiente fragmento:

```
LAR ARO,#1000h ; se carga ARO con 1000h  
MAR *,ARO      ; se actualiza el registro ARO  
LACC *         ; se carga el acumulador con el contenido de la localidad que indica ARO
```

Pareciera que son muchas instrucciones para realizar una simple carga sobre el acumulador y por ende consumiría muchos ciclos de reloj, sin embargo, todas las operaciones que tienen que ver con registros auxiliares, ya sea, la carga de un número o su actualización por medio de la instrucción MAR, son realizadas por la unidad aritmética de registros auxiliares (ARAU) y no por la ALU. Entonces, en nuestro ejemplo anterior, las dos primeras instrucciones pueden empezar a ejecutarse aún cuando la instrucción anterior (supongamos que hay un LACC) no termine de realizarse, esto nos da un claro ejemplo de ejecución en paralelo, lógicamente este paralelismo no se podrá dar si ejecutamos solo instrucciones que sean procesadas por una sola unidad interna del DSP; en el anterior ejemplo vemos que entran en acción dos unidades la ALU y la ARAU. Se observa entonces que en realidad no se tiene que consumir los ciclos de reloj completos de las tres instrucciones en tiempo real.

La mayoría de las instrucciones permiten el direccionamiento indirecto.

Otro tipo de direccionamiento muy común es el direccionamiento directo que consiste en acceder a un dato que se le da explícitamente a la instrucción, por ejemplo, la carga de los registros auxiliares:

LAR AR3, #2000h ;se carga el registro auxiliar 3 con 2000h

Para acceder a la memoria con este direccionamiento se puede hacer mediante el apuntador de página (DP) o directamente.

La primera instrucción hace exactamente lo mismo que las últimas dos.

LACC 0406h ;carga lo que hay en la dirección 0406h  
LDP 8 ;carga el apuntador de página con 8 d  
LACC 6 ;carga lo que resulte de la concatenación  
;del apuntador de página con el número 6d  
;( 1030d o sea 406h)

Cada página se entiende como un bloque de 128 palabras (hay 512 paginas disponibles) entonces ocho páginas sumadas dan 1024d más las 6 localidades que están indicadas en la instrucción LACC dan 1030d o 406h.

Siempre que se desee cargar un dato inmediato a un registro mediante una instrucción se debe anteponer un símbolo # al dato en cuestión, si no se considera al dato como una localidad de memoria que contiene un dato.

LACC #0a00h ;carga el valor 0a00h  
LACC 0a00h ;carga el valor que hay en la localidad 0a00h

Cuando el dato en cuestión es menor que 0FFh se dice que el direccionamiento es corto inmediato, si sobrepasa este valor se dice que es largo inmediato.

La figura 5.5 detalla la estructura interna del DSP .



## **6. CONFORMACIÓN SISTEMA $\mu$ C – ETAPA DE POTENCIA – MOTOR DE PASOS**

Como ya se mencionó en el capítulo anterior, para este caso, un microcontrolador maneja únicamente señales discretas o analógicas con una tensión máxima de 5V y con una baja intensidad de corriente. Un motor de pasos es una máquina que convierte energía eléctrica en movimiento mecánico realizando un trabajo, dichas maquinas normalmente operan con tensiones e intensidades mayores a 5V, por lo que es necesaria una interfaz o etapa de potencia entre el microcontrolador y el motor. Dicha etapa intermedia tiene la particularidad de recibir señales compatibles, en tensión e intensidad, con el microcontrolador y traducirlas en señales que puedan realizar un trabajo o movimiento mecánico del motor, manteniendo en todo momento aislado un sistema del otro para evitar daños sobre todo del lado del microcontrolador.

Como el objetivo de esta tesis no es el desarrollo de la etapa de potencia, se emplea un circuito dedicado para realizar esta tarea.

El circuito empleado en el presente trabajo es el L298 de ST Microelectronics, del cual se hace una breve descripción, de sus principales características, a continuación:

- a.- Se trata de un manejador de puente completo doble; esto significa que puede controlar dos bobinados independientemente, es decir, puede controlar la polaridad de dos bobinas a la vez de manera independiente.
  
- b.- Tensión de alimentación de hasta 46V; podemos controlar cualquier motor a pasos que tenga una tensión de operación menor a 46V.
  
- c.- Corriente CD total hasta 4A; puede controlar cualquier motor que requiera hasta 2A o menos por bobina.

d.- Baja tensión de saturación, esto es importante pues la tensión de encendido y la intensidad de corriente determinan la potencia disipada por el circuito.

e.- Protección por sobre temperatura; nos permite apagar el circuito y/o desconectar el motor de manera automática cuando existe un calentamiento por falla o exceso de carga.

f.- Alta inmunidad al ruido; al tener un umbral de nivel bajo en 1.5V evita que se presenten disparos con señales indeseables que generalmente son de muy baja tensión, menores a 0.5V.

g.- Este circuito puede disipar por sí solo hasta 25W, y puede aumentarse la disipación usando un radiador de calor que permita aumentar el área en contacto con el aire.

El empaque empleado en el presente trabajo es del tipo Multiwatt15 y se muestra en la figura 6.1.

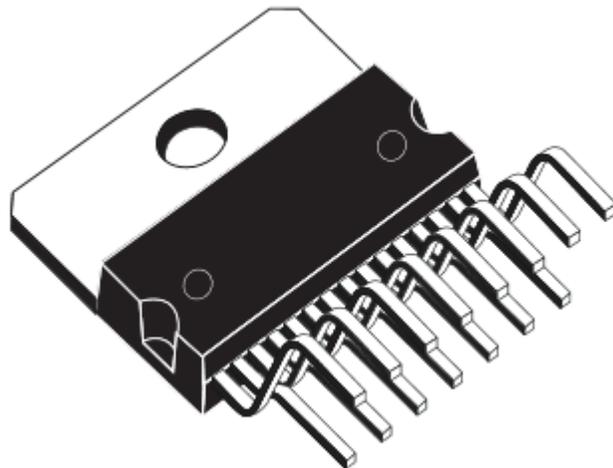


Figura 6. 1 Multiwatt 15.

La distribución de pines y sus funciones se encuentra en el apéndice A en la hoja de datos de dicho circuito.

El uso más común de este circuito es el que se muestra en el diagrama de la figura 6.2:

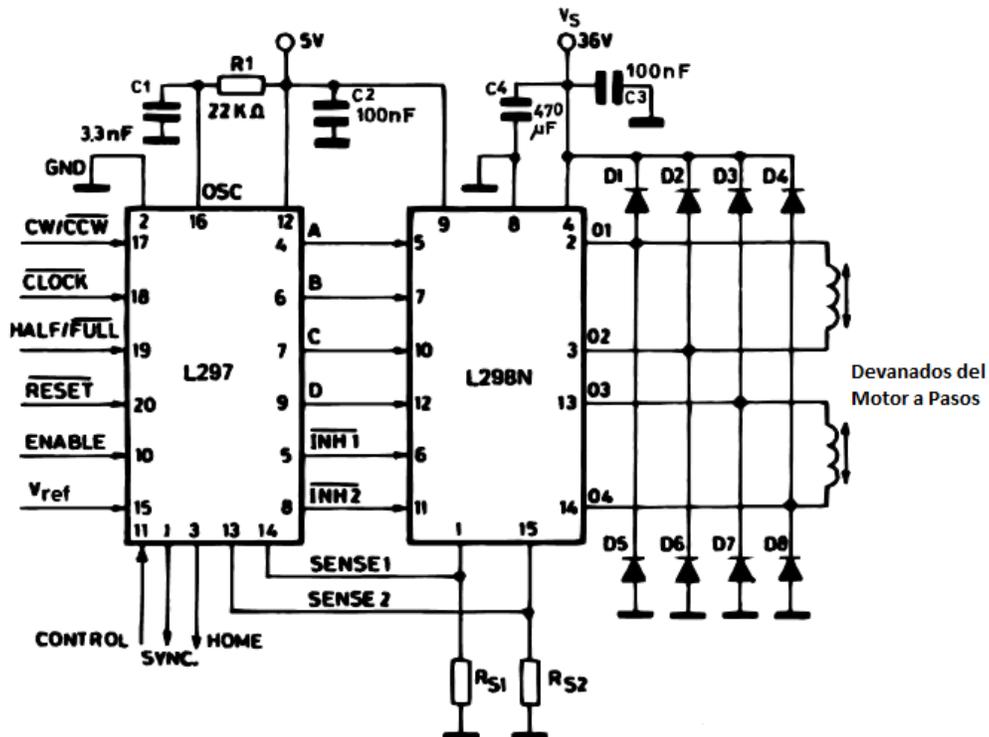


Figura 6. 2 Uso común del L298.

Para el caso de la figura 6.2 el circuito L297 se encarga de generar las secuencias de código para la polarización de los devanados del motor para cada una de las posibles posiciones; las instrucciones de los pines de control pueden ser de un micro o de interruptores manuales y se requiere además un pulso de reloj pues el circuito L297 se comporta como una máquina de estado.

Para lograr pasos más precisos usamos el método de control de medios pasos, para lo cual, según la tabla de estados del circuito L297, se emplea la tabla de la figura 6.3.

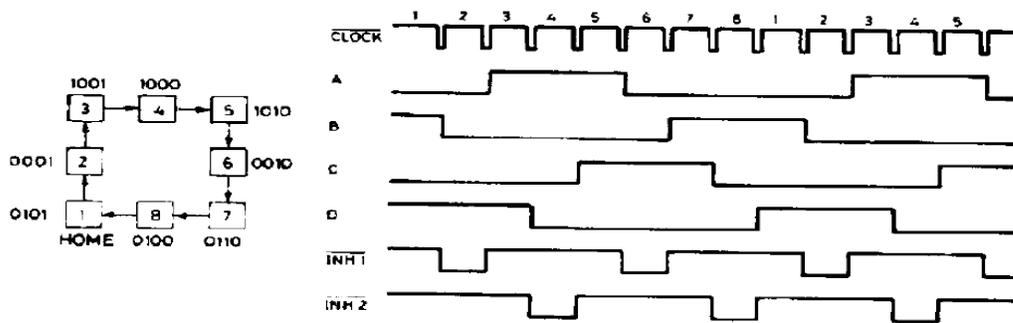


Figura 6. 3 Tabla de estados del circuito L297.

En la figura 6.3 pueden observarse los ocho pasos completos para lograr que el motor gire medios pasos y complete un ciclo.

Como, en nuestro caso, el objetivo es controlar el motor a pasos por medio de un microcontrolador, omitimos el uso del circuito L297, generando las señales de control A, B, C, D, INH1 e INH2 directamente de un puerto de salida, puerto B, del microcontrolador y recibiendo las opciones de movimiento en un puerto de entrada, puerto A. En este caso el reloj no es necesario pues el microcontrolador generará los cambios de estado en el puerto B configurado como puerto de salida y sólo es necesario cargar, dentro del código de programa, la tabla que se genera para cada uno de los pines del L298. La figura 6.4 muestra los valores de cada uno de los pines del L298 para cada uno de los 8 pasos.

PASO	INH2	INH1	D	C	B	A
1	1	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	1	0	0	1
4	0	1	0	0	0	1
5	1	1	0	1	0	1
6	1	0	0	1	0	0
7	1	1	0	1	1	0
8	0	1	0	0	1	0

Figura 6.4 Tabla de valores de los pines del L298 para cada uno de los 8 pasos.

El circuito completo del L298 que se conecta a los puertos de entrada y salida de la tarjeta MINICON11 es el que se muestra en la figura 6.5.

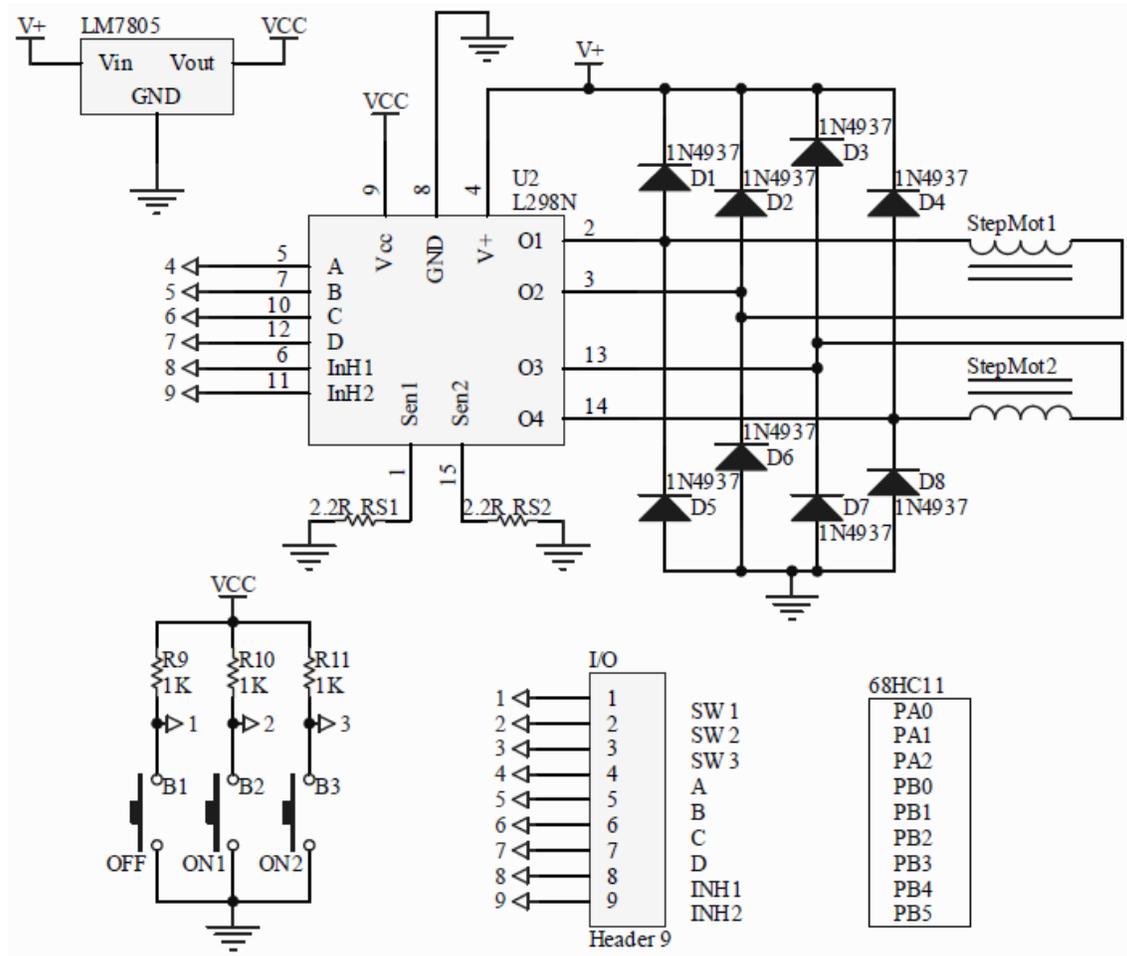


Figura 6. 5 Circuito del L298 y conexiones con la tarjeta MINICON11.

En este caso el microcontrolador MC68HC11 tiene puertos de entrada y salida independientes por lo que se usan los pines A0, A1 y A2 del puerto A para recibir las señales de control, que se obtienen presionando los botones B1, B2 o B3 del circuito. Una vez que el micro lee y procesa la información genera la secuencia de posiciones, para los pines INH 1, INH2, A, B, C y D, en los pines B5, B4, B3, B2, B1 y B0 respectivamente, del puerto de salida B.

Los botones B1, B2 o B3 mandan un pulso bajo al micro, al ser presionados, lo interpreta como un número binario de tres bits como un 7 si no se oprime ninguno, 6, 5 ó 3 si se presiona B1, B2 o B3 respectivamente.

Los diodos D1, D2, D3, D4, D5, D6, D7 y D8 sirven como diodos de rueda libre a los devanados el motor, tienen la función de proporcionar un camino a la corriente almacenada en los devanados del motor al momento de conmutar de encendido a apagado, es necesario colocarlos en puente pues la polaridad de los devanados cambia de acuerdo a las secuencias de polarización de los devanados.

Se usa también un regulador de voltaje de 5V para alimentar la parte analógica del circuito L298, y la parte de potencia se alimenta con un eliminador de 12V.

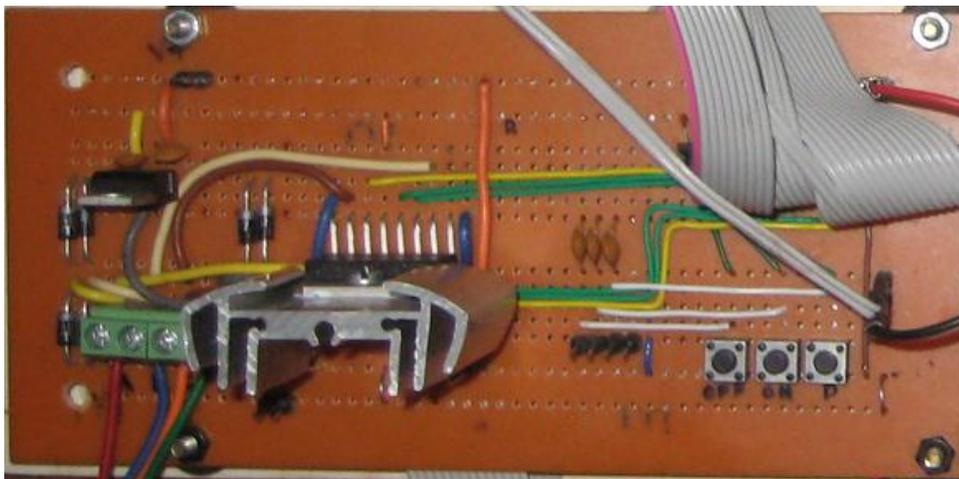


Figura 6.6 Interface de potencia.

## 7. CONFORMACIÓN SISTEMA DSP – ETAPA DE POTENCIA – MOTOR DE PASOS

### 7.1 L298 Puente en H para un motor de dos fases.

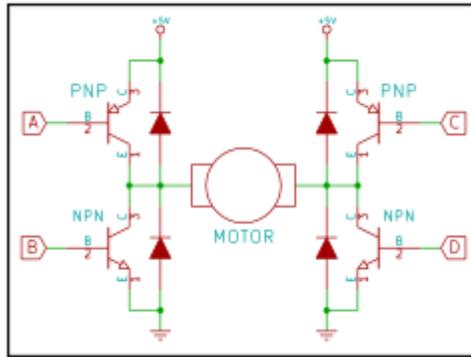


Figura 7.1. Circuito en puente H.

En los motores híbridos de dos fases, el driver bipolar da unos excelentes resultados por la siguiente razón, todos los bobinados se encuentran siempre excitados. En comparación con el driver unipolar, en éste la corriente fluye en una dirección del bobinado, pudiéndose obtener un incremento en el par de un 20 a un 35%.

Los motores deben de tener cuatro terminales para operar según el esquema de la figura 7.1. (dos terminales por fase o puente), los motores con un número de hilos diferentes a cuatro, se pueden utilizar si son de dos fases y admiten el control bipolar.

En el driver en puente se tiene que tener especial cuidado en la activación de los transistores de potencia, ya que éstos pueden resultar dañados; esto puede ocurrir si los dos transistores que se encuentran en cascada son activados al mismo tiempo. Cuando un transistor es desconectado, se tiene que dar un margen de tiempo antes de que el otro transistor se active, ya que ambos transistores podrían entrar en cortocircuito. Esto es debido a que normalmente el tiempo de desconexión es mayor que el tiempo de activación, por lo que, si uno se conecta y el otro se desconecta al mismo tiempo, se produce un cortocircuito.

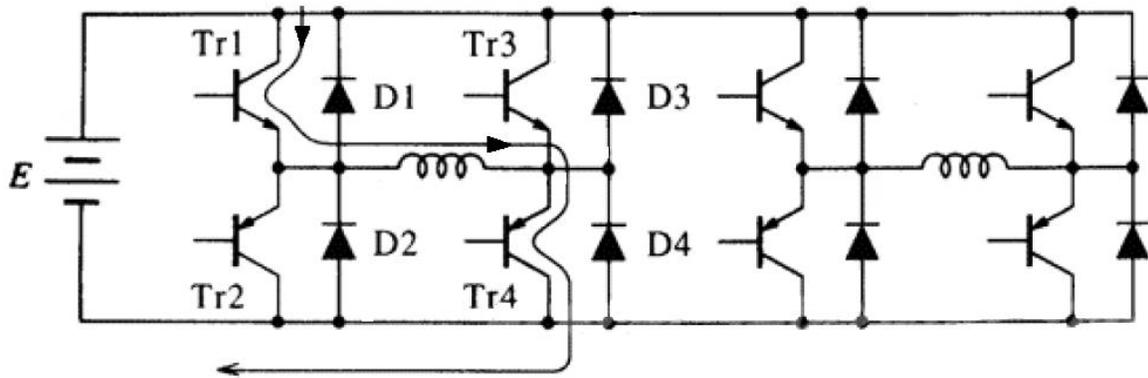


Figura 7.2. Puente completo para motor de dos fases.

Los diodos conectados en antiparalelo con los transistores Tr1 al Tr4, son para suprimir los picos de tensión generados por la energía almacenada en la inductancia de los devanados del motor, cuando los transistores pasan de saturación a corte de manera súbita. De acuerdo con la Figura 7.2 cuando están activos los transistores Tr1 y Tr4, la corriente sigue el circuito de la línea continua desde y hacia la fuente pasando por el devanado del motor; justo después que los transistores se ponen en corte, la energía almacenada sigue el circuito de la línea punteada a través del diodo D3 hacia la fuente y cierra el circuito con el diodo D2. En términos energéticos, la energía magnética presente en el bobinado es realimentada a la fuente de alimentación.

## 7.2 Puertos paralelos de entrada y salida.

La etapa de potencia para el DSP es la misma que se emplea con el microprocesador MC68HC11 y que se explicó en el capítulo anterior; sin embargo, el para el caso particular del TMS320C50 se usan los puertos de entrada y salida paralelos, específicamente el PA0, y el PA1, que son dos de los 16 puertos localizados en la memoria datos (Page 0, del mapa de memoria dato). Para lo anterior se requiere de un circuito interfaz de puerto, desarrollado con base en el manual del TMS320C5x que sugiere la figura 7.3.

El TMS320C50 cuenta con instrucciones de control para poder seleccionar la función del puerto como entrada o salida.

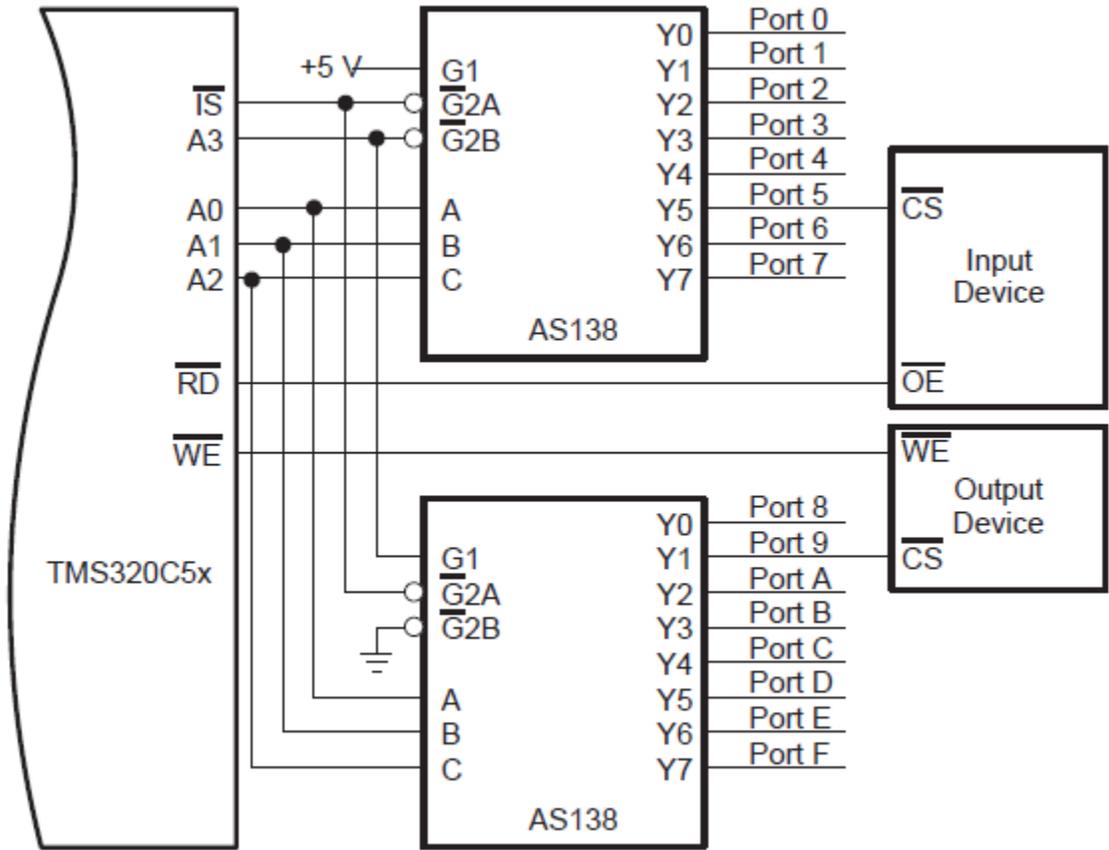


Figura 7.3 Circuito interfaz de puertos.

Como la señal de control en los pines IS, A3, A2, A1 y A0 activan por un instante el puerto como entrada o como salida, esto a través del demultiplexor AS138; y como sólo en ese instante es posible leer los datos en el puerto, es necesario usar otros circuitos periféricos que permitan mantener la señal hasta el siguiente ciclo de lectura o escritura y ésta pueda usarse. Por este motivo se usan los circuitos SN74LS373N conocidos como LATCH, los cuales tienen la función de mantener en su salida, una señal de entrada mientras no exista una nueva señal de cambio.

El circuito completo para el control de puertos de entrada y salida del TMS320C50 se muestra en la figura 7.4.

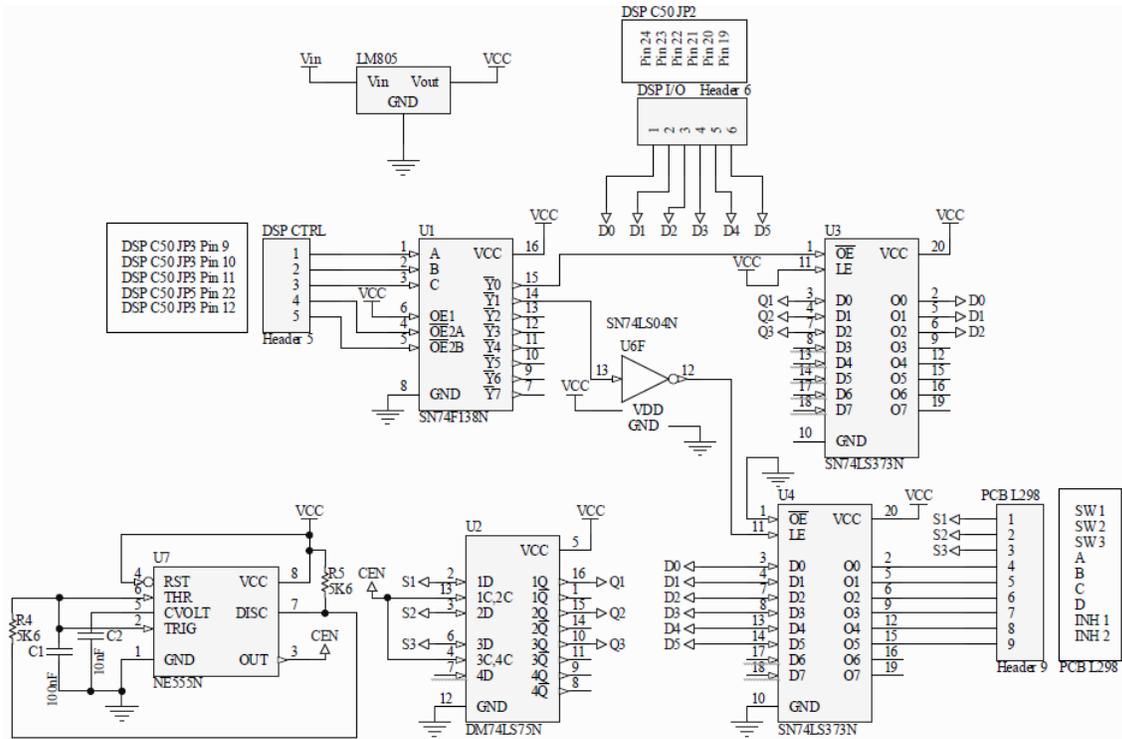


Figura 7. 4 Circuito de control de puertos de entrada y salida del TMS320C50.

Y en la figura 7.5. se muestra el prototipo del esquemático de la fig.7.4.

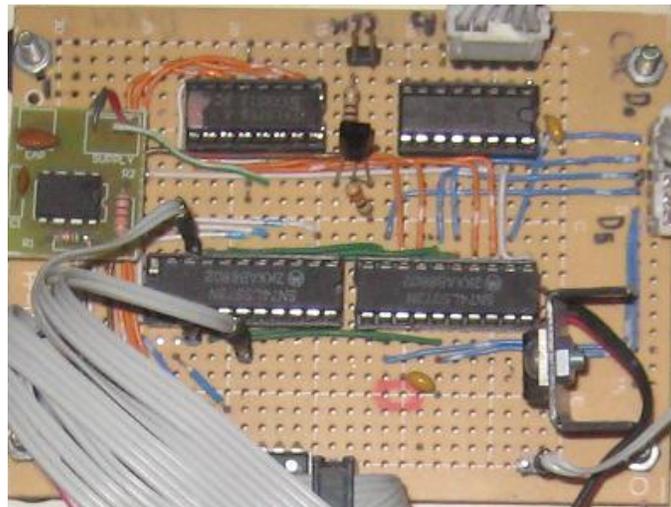


Figura 7.5 Prototipo del circuito de puertos del TMS320C50.

En el caso de las señales de entrada hacia el DSP para seleccionar una opción de movimiento del motor, es necesario mantener “el pulso” del botón, ya que en el ciclo de

decisión el DSP llevará a cabo la orden de leer el puerto de entrada; por esta razón es necesario colocar un circuito que mantenga los estados de la botonera durante un lapso de tiempo para que el DSP lleve a cabo la instrucción indicada para el puerto de salida o de entrada. Para lograr esta condición empleamos el circuito flip-flop, DM74LS75N y un oscilador NE555N como reloj para actualizar los cambios que corresponderán a las entradas.

En el diagrama de la figura 7.4 se ve claramente como los pines D0, D1, D2, D3, D4 y D5 del conector JP2 de la tarjeta de desarrollo del TMS320C50 se comportan como puerto de entrada y salida quedando el D0, D1 y D2 como entrada de la botonera.

El prototipo de la etapa de potencia para el TMS320C50 se muestra en la figura 7.6. Usando los pines de control 9, 10, 11 y 12 del conector JP3 y el pin 22 del conector JP5 y los pines 24, 23, 22, 21, 20 y 19 del conector JP2 como puerto de entrada/salida del lado del DSP y el conector I/O Header 9 de la tarjeta del circuito L298 o puente H, (la cual se explica en la figura 6.5).

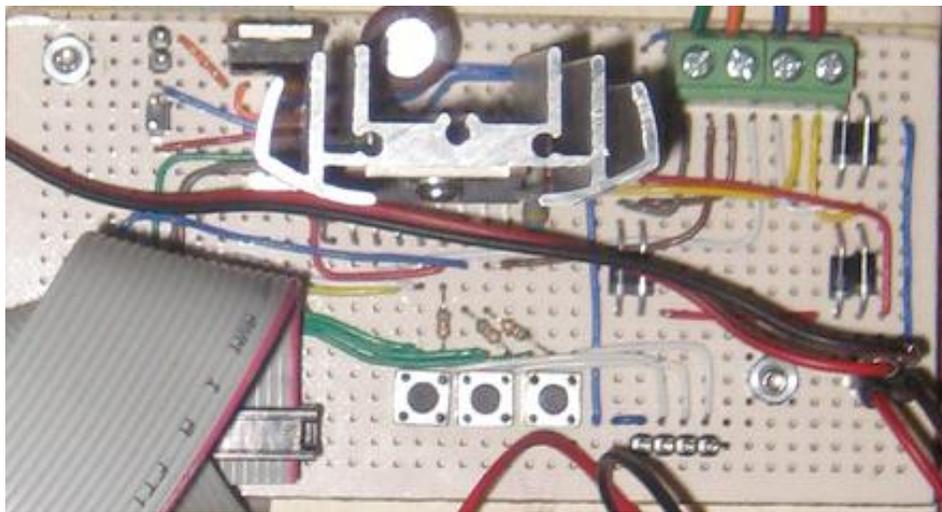


Figura 7.6 Prototipo de la etapa de potencia para el TMS320C50.

## **8. PRUEBAS DE POSICIÓN Y VELOCIDAD DE CADA UNO DE LOS SISTEMAS**

Una vez explicadas las etapas de potencia de cada uno de los sistemas, Microcontrolador MC68HC11 de Motorola y DSP TMS320C50 de Texas Instruments, procederemos a desarrollar los códigos de programa que se cargarán en cada uno de los procesadores para lograr el control de posición y velocidad de un motor a pasos, pero antes veamos los resultados en las pruebas de velocidad y posición para cada sistema.

### **8.1 Desarrollo de pruebas de posición y velocidad para el DSP C50 y el Microcontrolador MC68HC11.**

#### **8.1.1 Equipo requerido.**

- Computadora.
- TMS320C5x DSK.
- Minicon HC11.
- Cable USB a DB9 Macho.
- Fuente de alimentación de 24 Vca.
- Fuente de voltaje de 12 Vcd.
- Interfaces digitales (para el DSP TMS320C50).
- Etapas de potencia.
- Motor a pasos.
- Plataforma Virtual Box.
- Tacómetro.

#### **8.1.2 Procedimiento.**

a) Realizar las conexiones del cable USB serie en la computadora del lado USB como se muestra en la figura 8.1 (de lo contrario no funcionará la máquina virtual) y de lado serie a la tarjeta de desarrollo TMS320C50 o Minicon HC11 según sea el caso, adicionalmente conectar la tarjeta de desarrollo a la interface y etapa de potencia

(estas conexiones pueden estar ya hechas). Finalmente, energizar la TMS320C50 o Minicon HC11 así como las etapas que lo requieran.

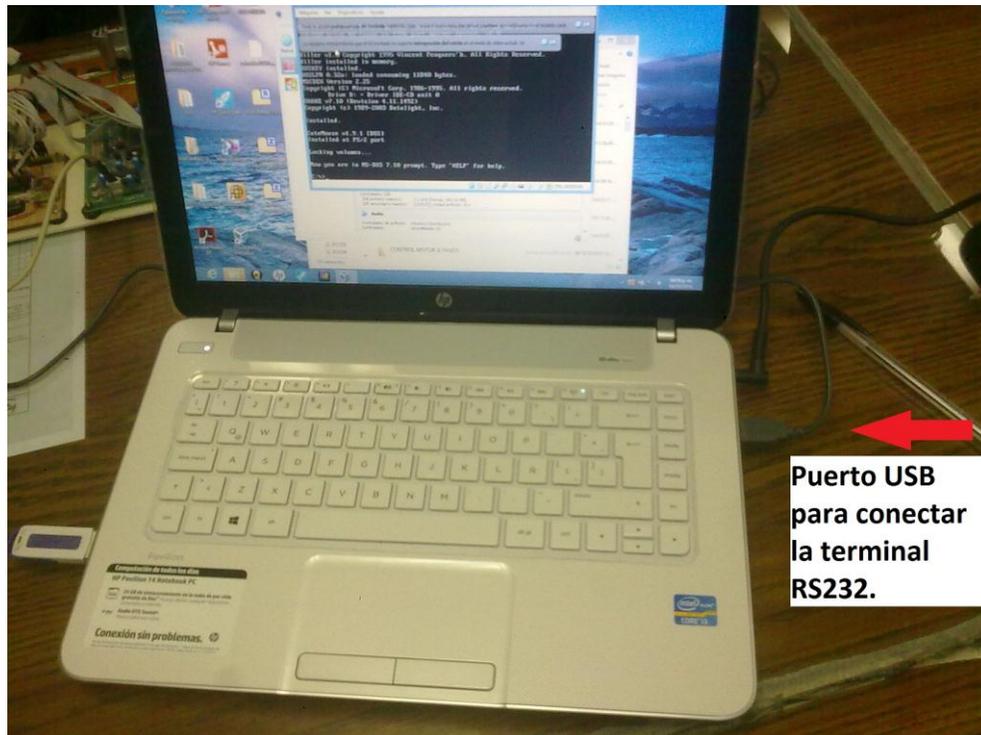


Figura 8.1 Conexiones a la computadora.

b) Ejecutar la Virtual box dando doble clic en el icono:



Figura 8.2 Icono de Virtual Box.

- c) Seleccionar MS-DOS para el DSP C50 o XP para el HC11 y dar clic en Start.
- d) Estando en el ambiente DOS, seleccionar el directorio DSK para el C50, o en Windows XP ir a Menú Inicio / Programas y seleccionar PUMMA\_11 con doble clic para Minicon HC11 mostrada en la figura 8.3.
- e) Teclear entonces DSK5D b38400 (para el DSP C50), se ejecutará la ventana del depurador del DSP mostrada en la figura 8.4.

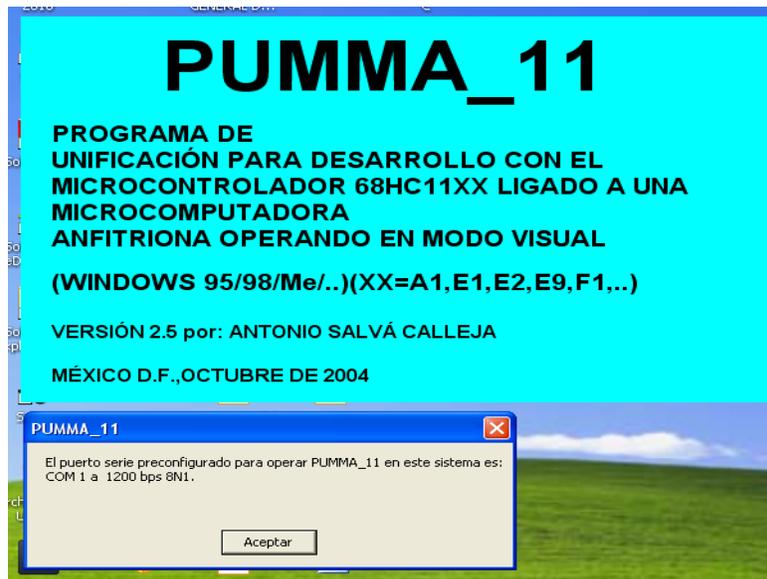


Figura 8.3 PUMMA\_11.

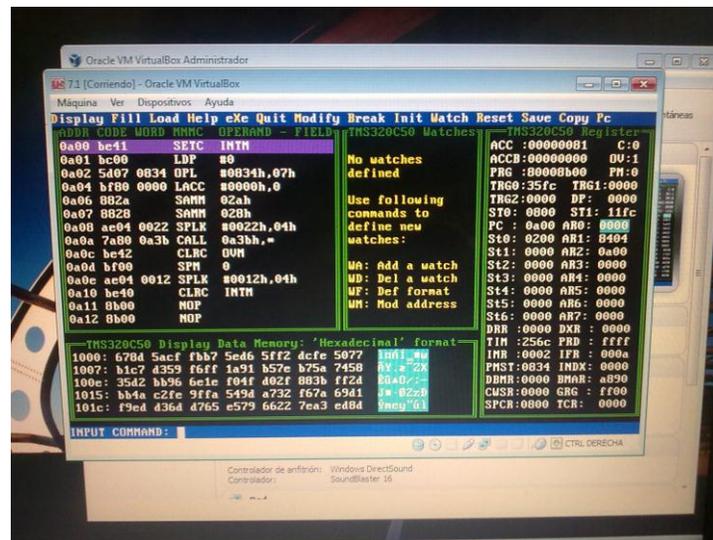


Figura 8.4 Depurador del DSP.

En caso contrario modificar las opciones de acuerdo a las sugerencias en pantalla.

- f) Para el C50 una vez en el ambiente del depurador en la línea de comando teclear L (en referencia a load) luego teclear D (en referencia a un archivo previamente ensamblado, que tienen extensión DSK).

g) Para el HC11 cargar los archivos con extensión “.s19” a la memoria del Microprocesador como muestra la figura 8.5.

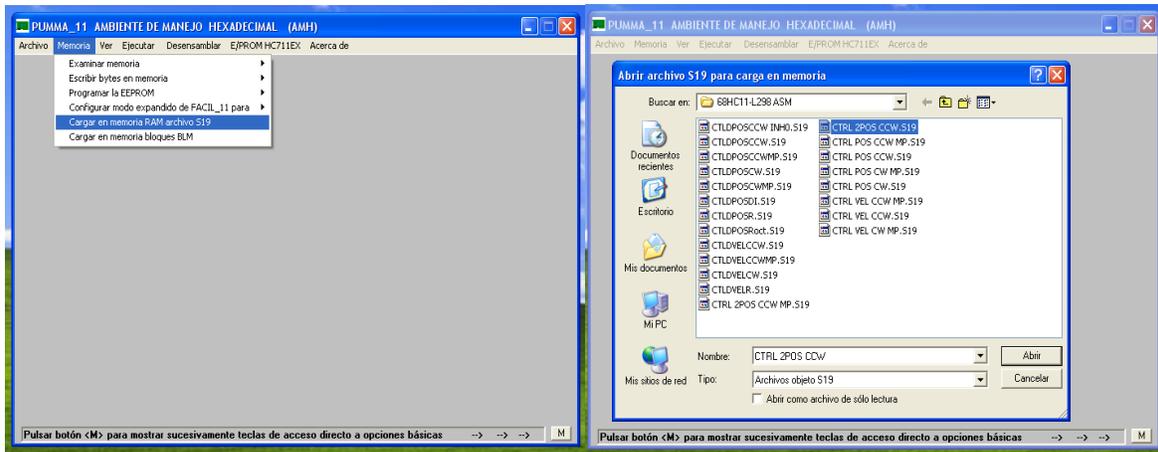


Figura 8.5 Archivos con extensión “.S19”.

h) Escribir o seleccionar la ruta completa correspondiente al archivo que se desea utilizar, esto es para control de velocidad o para control de posición de acuerdo a la figura 8.6.

SISTEMA	VELOCIDAD	POSICIÓN
C50	C:\DSK\V12.DSK	C:\DSK\P12R.DSK
HC11	C:\HC11\CTRL VEL CCW MP.S19	C:\HC11\CTRL 2POS CCW MP.S19

Figura 8.6 Rutas a utilizar.

## 8.2 Cálculos y mediciones para el DSP C50.

$$f_{TIM} = \frac{20 \text{ MHz}}{(PRD+1)(TDDR+1)} \quad (8-1)$$

donde:

$f_{TIM}$  = frecuencia de interrupción del timer y para la generación de cada paso

PRD = registro mapeado y preescalador del timer

TDDR = preescalador del timer, localizado en los bits(0 al 3)del TCR (Registro de control del timer)

Recordando la ecuación 3-2:

$$\alpha = \frac{2\pi}{spr} \quad (3-2)$$

donde:

$\alpha$  = ángulo del motor a paso (rad)

$spr$  = número de pasos por revolución

$$\omega = \frac{\alpha}{\delta t} = \alpha f_{TIM} \quad (8-2)$$

donde:

$\omega$  = velocidad del motor  $\left(\frac{\text{rad}}{\text{s}}\right)$

$\alpha$  = ángulo del motor a paso (rad)

$f_{TIM}$  = frecuencia de interrupción del timer y para la generación de cada paso

Recordando la ecuación 3-5:

$$1 \frac{\text{rad}}{\text{s}} = 9.55 \text{ rpm} \quad (3-5)$$

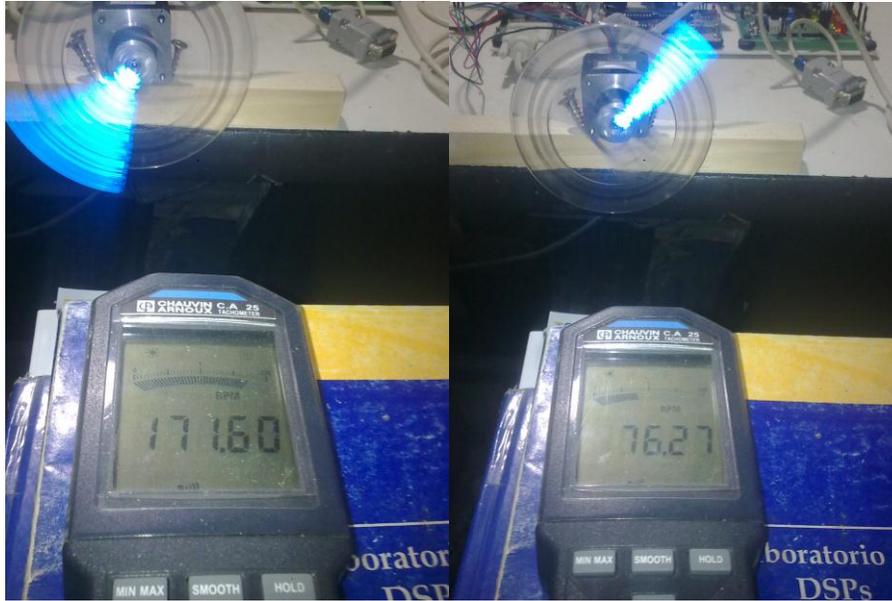


Figura 8.7 Medición de las velocidades configuradas en el código del C50.

### 8.2.1 Tablas de resultados para el C50.

DATOS DEFINIDOS EN EL CÓDIGO	PRD h	PRD d	TDDR h	TDDR d	$f_{T/M}$ (Hz)	$\omega$ ( $\frac{rad}{s}$ )	$\omega$ (rpm)	$\alpha/2$ (rad)	$\alpha/2$ (grados)	$\omega$ (rpm)
										tacómetro
	FFFF	65535	D	13	21,80	0,34	3,27	0,02	0,9	Fuera de rango
	0FBA	4026	4	4	993,30	15,60	149,01	0,02	0,9	149,1
Velocidad 1	1111	4369	3	3	1144,16	17,97	171,64	0,02	0,9	171,7
	5555	21845	0	0	915,50	14,38	137,34	0,02	0,9	137,3
	9999	39321	3	3	127,16	2,00	19,07	0,02	0,9	Fuera de rango
	ba98	47768	0	0	418,68	6,58	62,81	0,02	0,9	Fuera de rango
	aaaa	43690	0	0	457,76	7,19	68,67	0,02	0,9	68,5
Velocidad 2	9999	39321	0	0	508,62	7,99	76,30	0,02	0,9	76,26
	8888	34952	0	0	572,20	8,99	85,84	0,02	0,9	85,8
	6666	26214	0	0	762,92	11,98	114,45	0,02	0,9	114,4
	1234	4660	8	8	476,77	7,49	71,52	0,02	0,9	71,5

Figura 8.8 Tabla de velocidad programada contra velocidad obtenida con un tacómetro de laboratorio y el C50.

DATOS						(# de pasos)				Posición	
DEFINIDOS	PRD h	PRD d	TDDR h	TDDR d	$f_{TIM}$ (Hz)	Programado	(# de pasos)	$\alpha/2$ (rad)	$\alpha/2$ (grados)	programada	Posición
EN EL						en	Programado			(grados)	medida
CÓDIGO						hexadecimal	en decimal			(grados)	
Posición 1	FFFF	65535	4	4	61,04	190	400	0,02	0,9	360	360
	0FBA	4026	4	4	993,30	C8	200	0,02	0,9	180	180
Posición 2	ffff	65535	4	4	61,04	32	50	0,02	0,9	45	45

Figura 8.9 Tabla de posición programada contra posición obtenida al medir el ángulo total en un cambio de posición en el C50.

### 8.3 Cálculos y mediciones para el HC11.

$$f_{\text{TIMER}} = \frac{2 \text{ MHz}}{(\text{PR1})(\text{PR0})} \quad (8-3)$$

donde:

$f_{\text{TIM}}$  = frecuencia de interrupción del timer y para la generación de cada paso

$\text{PR1}$  = BIT 1 del registro TMSK2

$\text{PR0}$  = BIT 0 del registro TMSK2

Para nuestro desarrollo :

$$f_{\text{TIMER}} = 125 \text{ KHz} \quad (8-4)$$

$$\omega = \frac{\alpha}{\delta t} = \alpha f_{\text{PASO}} \quad (8-5)$$

donde:

$\omega$  = velocidad del motor  $\left(\frac{\text{rad}}{\text{s}}\right)$

$\alpha$  = ángulo del motor a paso (rad)

$f_{\text{PASO}}$  = frecuencia para la generación de cada paso

Periodo = valor numérico usado para variar la duración de cada paso

$$f_{\text{PASO}} = \frac{f_{\text{TIMER}}}{\text{PERIODO}} \quad (8-6)$$



Figura 8.10 Medición de las velocidades configuradas en el código del HC11.

### 8.3.1 Tablas de resultados para el HC11.

Ya que  $E=2$  MHz y que se utiliza un divisor de frecuencia de valor 16, se tiene que el temporizador tiene una frecuencia = 125 KHz.

DATOS DEFINIDOS EN EL CÓDIGO	PERIODO h	PERIODO d	FRECUENCIA DEL TIMER (Hz)	tiempo (s)	$f_{Paso}$ (Hz)	$\omega$ ( $\frac{rad}{s}$ )	$\omega$ (rpm)	$\alpha/2$ (rad)	$\alpha/2$ (grados)	$\omega$ (rpm) tacómetro
	58	88	125000	0,000008	1420,45	22,31	213,08	0,02	0,9	213
	5A	90	125000	0,000008	1388,89	21,82	208,35	0,02	0,9	208,3
	BB	187	125000	0,000008	668,45	10,50	100,27	0,02	0,9	100,24
	EA	234	125000	0,000008	534,19	8,39	80,13	0,02	0,9	80,12
	100	256	125000	0,000008	488,28	7,67	73,25	0,02	0,9	73,25
	110	272	125000	0,000008	459,56	7,22	68,94	0,02	0,9	68,71
	125	293	125000	0,000008	426,62	6,70	64,00	0,02	0,9	63
Velocidad 1	135	309	125000	0,000008	404,53	6,35	60,68	0,02	0,9	60,68
Velocidad 2	70	112	125000	0,000008	1116,07	17,53	167,42	0,02	0,9	167,4
	65	101	125000	0,000008	1237,62	19,44	185,66	0,02	0,9	185,6
	60	96	125000	0,000008	1302,08	20,45	195,33	0,02	0,9	195,34

Figura 8.11 Tabla de velocidad programada contra velocidad obtenida con un tacómetro de laboratorio y el HC11.

DATOS DEFINIDOS EN EL CÓDIGO	PERIODO h	PERIODO d	FRECUENCIA DEL TIMER (Hz)	tiempo (s)	$f_{TIM}$	n (# de pasos) Programado en hexadecimal	n (# de pasos) Programado en decimal	$\alpha/2$ (rad)	$\alpha/2$ (grados)	Posición programada (grados)	Posición medida (grados)
Posición 1	30	48	125000	0,000008	2604,17	32	50	0,02	0,9	45	45
	30	48	125000	0,000008	2604,17	64	100	0,02	0,9	90	90
Posición 2	30	48	125000	0,000008	2604,17	c8	200	0,02	0,9	180	180

Figura 8.12 Tabla de posición programada contra posición obtenida al medir el ángulo total en un cambio de posición en el HC11.

## 8.4 Código ASM para control de posición en el HC11.

Del archivo: CTRL 2POS CCW MP.asm

```

PORTA    EQU    $00    ;DIRECCIÓN DE PUERTO A
PORTB    EQU    $04    ;DIRECCIÓN DE PUERTO B
TMSK2    EQU    $24    ;DIRECCIÓN DEL REGISTRO DE MASCARA 2 DEL TIMER
TFLG2    EQU    $25    ;DIRECCIÓN DEL REGISTRO DE BANDERAS 2 DEL TIMER
TCNT     EQU    $0E    ;DIRECCIÓN DEL TIMER
TCTL1    EQU    $20    ;DIRECCIÓN DE CONTROL DE LOS COMPARADORES
TFLG1    EQU    $23    ;DIRECCIÓN DEL REGISTRO DE BANDERAS 1 DEL TIMER
TMSK1    EQU    $22    ;DIRECCIÓN DEL REGISTRO DE MASCARA 1 DEL TIMER
TOC2     EQU    $18    ;DIRECCIÓN DEL REGISTRO DEL COMPARADOR 2
CONT     RMB    1      ; RESERVAR BYTE DE MEMORIA PARA CONT
CERO     EQU    $00    ; VALOR INICIAL PARA CERO
SECUEN   RMB    1      ; RESERVAR BYTE DE MEMORIA PARA SECUEN
NUMPASO  RMB    1      ; NUMERO DE PASOS A MOVERSE
PERIODO  EQU    $0030  ; VALOR INICIAL PARA PERIODO

        ORG    $0000
        CLR   SECUEN    ;DEJAR EN CERO SECUEN
        LDX   #$1000
        BSET  TMSK2,X,$03 ;CONFIGURAR REGISTO 2 DEL TIMER
        BCLR  TCTL1,X,$C0 ;TOC2 TIMER DISCONEC FROM OUTPUT
        LDD  TCNT,X      ;SE IGUALAN TOC2 Y TEMPORIZADOR
        STD  TOC2,X      ;GUARDAR REGISTRO D

        ;CICLO PARA DECIDIR ENTRE DOS POSICIONES

INIC     CLR   PORTB,X    ;DEJAR EN CERO EL PUERTO B
        LDAA #$40

```

```

STAA  TFLG1,X
STAA  TMSK1,X      ;HABILITO INTERRUPCION DE COMPARADOR2
LDAB  PORTA,X      ;SE oprimio bit A2 para inicio?
CMPB  #$06         ;CICLO PARA DECIDIR ENTRE DOS POSICIONES
BEQ   POS1
CMPB  #$05
BEQ   POS2
BNE   INIC         ;FIN DE CICLO

```

*;CICLO DE POSICIÓN 1*

```

POS1  CLR  NUMPASO  ;OPCIÓN 1
      CLI
PAS1  LDAA NUMPASO
      CMPA #$32     ;ESTE ES EL VALOR MENOR DE PASOS A MOVERSE
      BNE  PAS1
      SEI          ;DESHABILITA INTERRUPT ya no borra puerto B
      BRA  INIC

```

*;CICLO DE POSICIÓN 2*

```

POS2  CLR  NUMPASO  ;OPCIÓN 2
      CLI
PAS2  LDAA NUMPASO
      CMPA #$C8     ;ESTE ES EL VALOR MAYOR DE PASOS A MOVERSE
      BNE  PAS2
      SEI          ;DESHABILITA INTERRUPT ya no borra puerto B
      BRA  INIC

```

*;RUTINA DE INTERRUPCIÓN*

```

OVER  ORG  $0062    ;RUTINA DE INTERRUPCIÓN PARA EJECUCIÓN DE
      BSET TFLG1,X,$4 ;UN AVANCE
      LDD  PERIODO  ;CARGA EL PERIODO DE TIEMPO DE CADA PASO
      ADDD TOC2,X
      STD  TOC2,X
      LDY  #TABLA
      LDAB SECUEN   ;APUNTADOR A LOS ELEMENTOS DE LA TABLA
      ABY
      LDAA #$00
      ORAA CERO,Y   ;SELECCIÓN DE UN VALOR EN LA TABLA
      STAA PORTB,X  ;ESCRIBIR EN LA SALIDA EL VALOR DEL PASO
      INC  SECUEN
      LDAB SECUEN
      CMPB #$08
      BNE  SALT
      CLR  SECUEN

```

```
SALT      INC      NUMPASO      ; CONTADOR DE PASO DADOS
OUT       RTI
```

### *;TABLA*

```
TABLA     ORG      $0100      ;son ocho pasos EL ORDEN ES IN2 IN1 D C B A
step2     FCB      $39        ;11 1001
step3     FCB      $3A        ;11 1001
step4     FCB      $12        ;01 0010
step5     FCB      $36        ;11 0110
step6     FCB      $24        ;10 0100
step7     FCB      $35        ;11 0101
step8     FCB      $11        ;01 0001

          ORG      $00DC      ;DIRECCION DE RTI
          JMP      OVER
```

Es posible disminuir el tamaño del paso o de los grados con el método de medios pasos. En el código escrito para este proyecto de tesis se usaron medios pasos.

## **8.5 Código ASM para el control de velocidad en el HC11.**

Del archivo: CTRL VEL CCW MP.asm

```
PortB     EQU      $04        ;DIRECCIÓN DE PUERTO B
PORTA     EQU      $00        ;DIRECCIÓN DE PUERTO A
TCNT      EQU      $0E        ;DIRECCIÓN DEL TIMER
TCTL1     EQU      $20        ;DIRECCIÓN DEL REGISTRO DE BANDERAS 1 DEL TIMER
TMSK2     EQU      $24        ;DIRECCIÓN DEL REGISTRO DE MASCARA 2 DEL TIMER
TFLG1     EQU      $23        ;DIRECCIÓN DEL REGISTRO DE BANDERAS 1 DEL TIMER
TMSK1     EQU      $22        ;DIRECCIÓN DEL REGISTRO DE MASCARA 1 DEL TIMER
TOC2      EQU      $18        ;DIRECCIÓN DEL REGISTRO DEL COMPARADOR 2
CERO      EQU      $00        ; VALOR INICIAL PARA CERO
SECUEN    RMB      1          ; RESERVAR BYTE DE MEMORIA PARA SECUEN
NUMPASO   RMB      1          ; NUMERO DE PASOS A MOVERSE
PERIODO   FCB      $FF,$01    ; VALOR INICIAL PARA PERIODO

          ORG      $0000
          CLR      SECUEN
          LDX      #$1000
          BSET    TMSK2,X,$03
          BCLR   TCTL1,X,$C0    ;oc2 TIMER DISCONEC FROM OUTPUT
          LDD     TCNT,X        ;SE IGUALAN TOC2 Y TEMPORIZADOR
          STD     TOC2,X
          BSET    TFLG1,X,$40
```

CLR NUMPASO

*;CICLO PARA DECIDIR ENTRE DOS VELOCIDADES*

INIC        BSET    TMSK1,X,\$40        ;HABILITO INTERRUPTON DE COMPARADOR2  
CON        LDAA    PORTA,X            ;Lee el Puerto A  
           CMPA    #\$06                ;Pregunta cual boton fue presionado  
           BEQ    VEL1  
           CMPA    #\$05            ;SELECCION DE VELOCIDAD1 ó VELOCIDAD2  
           BEQ    VEL2  
           BRA    CON

*;CICLO DE VELOCIDAD 1*

VEL1        LDD    #\$0135            ;NUMERO DE VELOCIDAD,ESTE VALOR ES EL ALTO  
           STD    PERIODO  
           JMP    ENAB

*;CICLO DE VELOCIDAD 2*

VEL2        LDD    #\$0070            ;NUMERO DE VELOCIDAD,ESTE VALOR ES EL BAJO  
           STD    PERIODO  
ENAB        CLI  
INF        LDAA    PORTA,X  
           CMPA    #\$03            ;SALIR?  
           BNE    INF  
           CLR    PORTB,X  
           SEI                    ;DESHABILITA INTERRUPTON  
           BRA    INIC

*;RUTINA DE INTERRUPTON*

OVER        ORG    \$0060            ;RUTINA DE INTERRUPTON PARA EJECUCION DE  
           BSET    TFLG1,X,\$40        ;UN AVANCE  
           LDD    PERIODO        ;CARGA EL PERIODO DE TIEMPO DE CADA PASO  
           ADDD    TOC2,X  
           STD    TOC2,X  
           LDY    #TABLA  
           LDAB    SECUEN            ;APUNTADOR A LOS ELEMENTOS DE LA TABLA  
           ABY  
           LDAA    #\$00  
           ORAA    CERO,Y  
           STAA    PORTB,X        ;ESCRIBIR EN LA SALIDA EL VALOR DEL PASO  
           INC    SECUEN  
           LDAB    SECUEN  
           CMPB    #\$08  
           BNE    SALT

```

SALT   CLR   SECUEN
OUT    INC   NUMPASO
      RTI

```

*;TABLA*

```

TABLA  ORG   $0100   ;son cuatro pasos el orden es INH2, INH1, D, C, B, A
step2  FCB   $39     ;11 1001
step3  FCB   $28     ;10 1000
step4  FCB   $3A     ;11 1001
step5  FCB   $12     ;01 0010
step6  FCB   $36     ;11 0110
step7  FCB   $24     ;10 0100
step8  FCB   $35     ;11 0101
      FCB   $11     ;01 0001

      ORG   $00DC   ;DIRECCION DE RTI
      JMP   OVER

```

## 8.6 Código ASM para el control de posición en el TMS320C50.

En el caso del DSP, se utiliza el siguiente código (listado abajo), para generar secuencias de movimiento una con un ángulo de giro de 360° en caso de seleccionar posición1 y de 45° en caso de seleccionar posición 2.

Del archivo: P12R.asm

```

      .mmregs           ;Registros mapeados a memoria
      .ds               1B00h ;dirección para segmento de datos
SAL1  .word             5
SAL2  .word             5
KONS  .word             0
dato  .word             0
dato2 .word             0
z     .word             0
TABLA .word             11h,35h,24h,36h,12h,3ah,28h,39h

      .ps 00808h      ;
      B TINT          ; Set transmit interrupt vector
      .PS 00A00h      ;Segmento de programa
      .TEXT           ;inicio de Código de programa

```

\* ASIGNACION DE AR7 COMO BUFER CIRCULAR

```

LACC      #TABLA      ;
SMM      CBSR2
SMM      AR7          ;apuntador a la tabla de datos
ADD      #07h
SMM      CBER2
LACL     #0f0h
SMM      CBCR

```

\*fin de asignacion de ar7

\*configuracion de timer para utilizarse en interrupcion

```

SPLK     #ffff, PRD
OPL      #8,IMR      ; mascara de interrupcion
SPLK     #424h,TCR
SPLK     #1000b,IFR

```

\*FIN DE CONFIGURACION DE TIMER

*;CICLO PARA DECIDIR ENTRE DOS POSICIONES*

```

SEL:     nop                ;ciclo de selección entre posiciones
        lar                ar0, #dato
        MAR                *, AR0
        Lacl               #07h,0
        IN                 *, PA0      ;leer entradas
        and                *
        SACL               *,0,AR0

        CPL                #03h,*
        bcnd               POS1,tc    ;posición uno
        CPL                #05h,*
        BCND               POS2,TC    ;posición dos

        LACL               #0h,0
        MAR                *,AR5
        SACL               *,0,AR5
        MAR                *, AR5
        OUT                *, PA1
        LAR                AR2,#KONS
        LACL               #0h,0
        MAR                *,AR2
        SACL               *,0,AR2

        B                  SEL        ; fin de selección

```

*;CICLO DE POSICIÓN 1*

```

POS1:      CLRC      INTM      ;posición 1

REP1:
MAR        *,AR2
CPL        #0190h,*      ; cc vuelta completa 360º, valor a modificar
BCND       SEL,TC
B          REP1

                ;CICLO DE POSICIÓN 2

POS2:      CLRC      INTM      ;posición 2

REP2:
MAR        *,AR2
CPL        #032h,*      ; valor a modificar para variar el angulo recorrido
BCND       time,TC      ; 32h equivale a 45grados a medios pasos
B          REP2

time:
splk       #0ffffh,BRCR      ;retardo para permitir avance en
rptb       RETARDO          ;posiciones pequeñas
rpt        #0ffffh
lacc       #0,0
rpt        #0ffffh
lacc       #0,0
rpt        #0ffffh
RETARDO:   lacc       #0,0
B          SEL

                ;RUTINA DE INTERRUPCIÓN

TINT:      LACL     *,AR2      ;interrupción para dar salida a un paso al motor
ADD        #1h
SACL      *,0,AR7
OUT       *+,PA1
RETE

.END

```

## 8.7 Código ASM para el control de velocidad en el TMS320C50.

Del archivo: V12.asm

\*\* SE;AL DE MOTOR A PASOS A DSP TMS320C50\*

```
.mmregs          ;Registros mapeados a memoria
.ds 1B00h        ;dirección para segmento de datos
SAL1             .word 5
SAL2             .word 5
KONS             .word 0
dato             .word 0
dato2            .word 0
z                .word 0
TABLA            .word 11h,35h,24h,36h,12h,3ah,28h,39h
```

```
.ps 00808h      ;
B TINT          ; Set transmit interrupt vector
.PS 00A00h      ;Segmento de programa
.TEXT           ;inicio de Código de programa
```

\* ASIGNACION DE AR7 COMO BUFER CIRCULAR

```
LACC #TABLA
SMM CBR2
SMM AR7         ;apuntador a la tabla de datos
ADD #07h
SMM CBER2
LACL #0f0h
SMM CBCR
```

\*fin de asignacion de ar7

\*configuracion de timer para utilizarse en interrupción

```
SPLK #ffffh, PRD
OPL #8,IMR ; mascara de interrupcion
SPLK #424h,TCR
SPLK #1000b,IFR
```

\*FIN DE CONFIGURACION DE TIMER

*;CICLO PARA DECIDIR ENTRE DOS VELOCIDADES*

```
SEL:          nop          ;ciclo de selección entre velocidades
              lar ar0, #dato
              MAR *, AR0
              lacl #07h,0
              IN *, PA0
              and *
              SACL *,0,AR0
```

```

CPL #03h,*
BCND VEL1,TC ; velocidad 1
CPL #05h,*
bcnd VEL2,TC ; velocidad 2

```

```

LACL #0h,0
MAR *,AR5
SACL *,0,AR5
MAR *,AR5
OUT *,PA1 ;borrar salida
LAR AR2,#KONS
LACL #0h,0
MAR *,AR2
SACL *,0,AR2

```

```
B SEL
```

*;CICLO DE VELOCIDAD 1*

```

VEL1: CLRC INTM ;ciclo de velocidad uno
SPLK #1111h,PRD ;valor a modificar para variar la velocidad 1
SPLK #423h,TCR ;valor a modificar para variar la velocidad 1
V1: lar ar1,#dato2
MAR *,AR1
lacl #07h,0
IN *,PA0 ;leer entrada
and *
SACL *,0,AR1

CPL #06h,* ;SALIR? ;detener movimiento?
BCND SEL,TC
B V1

```

*;CICLO DE VELOCIDAD 2*

```

VEL2: CLRC INTM ;ciclo de velocidad dos
SPLK #9999h,PRD ;valor a modificar para variar la velocidad 2
SPLK #420h,TCR ;valor a modificar para variar la velocidad 2
V2: lar ar1,#dato2
MAR *,AR1
lacl #07h,0
IN *,PA0 ;leer entrada
and *
SACL *,0,AR1
CPL #06h,* ;SALIR? ;detener movimiento?
BCND SEL,TC
B V2

```

*;RUTINA DE INTERRUPCIÓN*

```
TINT:    LACL  *,AR2      ;interrupción para dar salida a un paso al motor
         ADD   #1h
         SACL  *,0,AR7
         OUT  *,PA1
         RETE
         .END
```

## 9. CONCLUSIONES

El diseño del código para el control del motor se desarrolla con marcadas diferencias, el HC11 utiliza comparadores y el timer para obtener las variantes de velocidad y posición, el DSP ocupa las interrupciones del timer y un registro circular que resuelve que dato de la tabla de valores debe colocarse a la salida.

En el caso del microprocesador MC68HC11E1, el contar con puertos paralelos de entrada y salida, facilita la conexión con la etapa de potencia, el motor a pasos y los interruptores de control; es decir, a cada pin del micro le corresponde una señal, ya sea de entrada o salida. Esto simplifica bastante la lógica de control y las conexiones.

En cuanto al código en el HC11 puede ser muy flexible para lograr el control de un motor a pasos empleando diferentes técnicas para variar los periodos de tiempo entre pasos, en este trabajo se emplean interrupciones derivadas por el acumulador por brindar la posibilidad de calcular de manera más directa los periodos y que sean independientes de los ciclos de instrucción.

Podemos decir que la comparación que resulta al poder usar ambos sistemas, (el HC11 y el C50) y el interactuar con la misma etapa de potencia es muy equilibrada.

Ambos sistemas cumplen bien con las condiciones de posición y velocidad, en el caso del HC11 es más práctico al contar con puertos, en el DSP se tienen que integrar al hardware, pero es más completo en cierta forma al desarrollarlo. Como punto en contra podemos mencionar que es de 8 bits, a diferencia del C50 que es de 16 bits.

Para el C50 las instrucciones, registros auxiliares y un diseño enfocado al procesamiento de algoritmos matemáticos le ayudan cuando se trata de hacer lo que un microcontrolador comúnmente hace.

## **BIBLIOGRAFÍA**

T.I. TMS320C5x User's Guide. Literature Number: SPRU056D. June 1998.

T.I. TMS320C5x DSP Starter Kit User's Guide. Literature Number: SPRU101A. Reprinted November 1997.

ESCOBAR S.L. Laboratorio de DSPs, familias TMS320C5X y TMS320C54X. Facultad de Ingeniería, UNAM, 2002, México.

SPASOV P. Microcontroller Technology: The 68HC11, Prentice Hall, 4th Edition, 2002, U.S.A.

HC11. Reference Manual M68HC11, Motorola, 1991, U.S.A.

HUBERT Ch. Electric Machines, Theory, Operation, Applications, Adjustment, and Control. Macmillan Publishing Company, 1991, U.S.A.

OGATA, K. Ingeniería de Control Moderna. Prentice Hall, 1980, México.

## MESOGRAFÍA

MC68HC11 Assembler THRAss11, 2002, HC11 Demon, The Netherlands Harry Broeders.

<http://www.hc11.demon.nl/thrsim11/infoass.htm>. Consulta: Marzo 2016.

Microcontrolador 68HC11: Fundamentos, recursos y programación.

<http://www.iearobotics.com/proyectos/libro6811/libro6811.html> . Consulta: Junio 2015.

Motores a paso. <http://www.todorobot.com.ar/tutorial-sobre-motores-paso-a-paso-stepper-motors/> Consulta: Marzo 2015.

Motores a paso híbridos. [http://www.alciro.org/alciro/Plotter-Router-Fresadora-CNC\\_1/Motores-hibridos\\_158.htm](http://www.alciro.org/alciro/Plotter-Router-Fresadora-CNC_1/Motores-hibridos_158.htm). Consulta: Febrero 2016.

Virtual Box. <https://www.virtualbox.org/> . Consulta Marzo 2015.

TMS320C5x DSP Starter Kit User's Guide <http://www.ti.com/lit/ug/spru101a/spru101a.pdf>.  
Consulta: Febrero 2015.



## APÉNDICE A



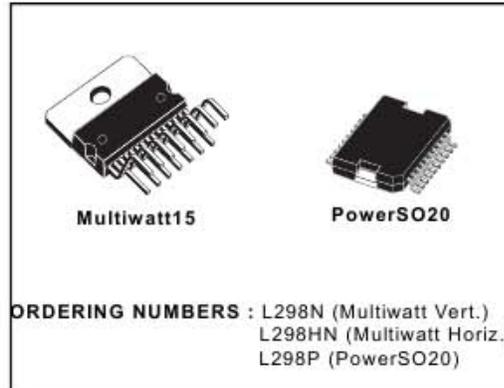
**L298**

### DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

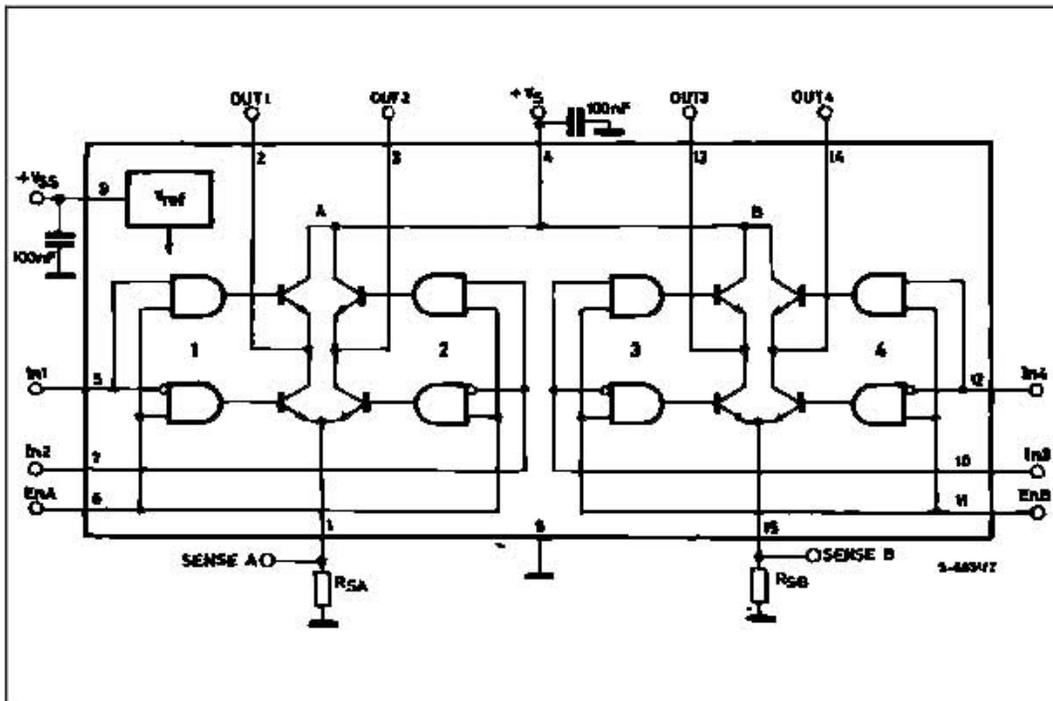
#### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-load Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

#### BLOCK DIAGRAM

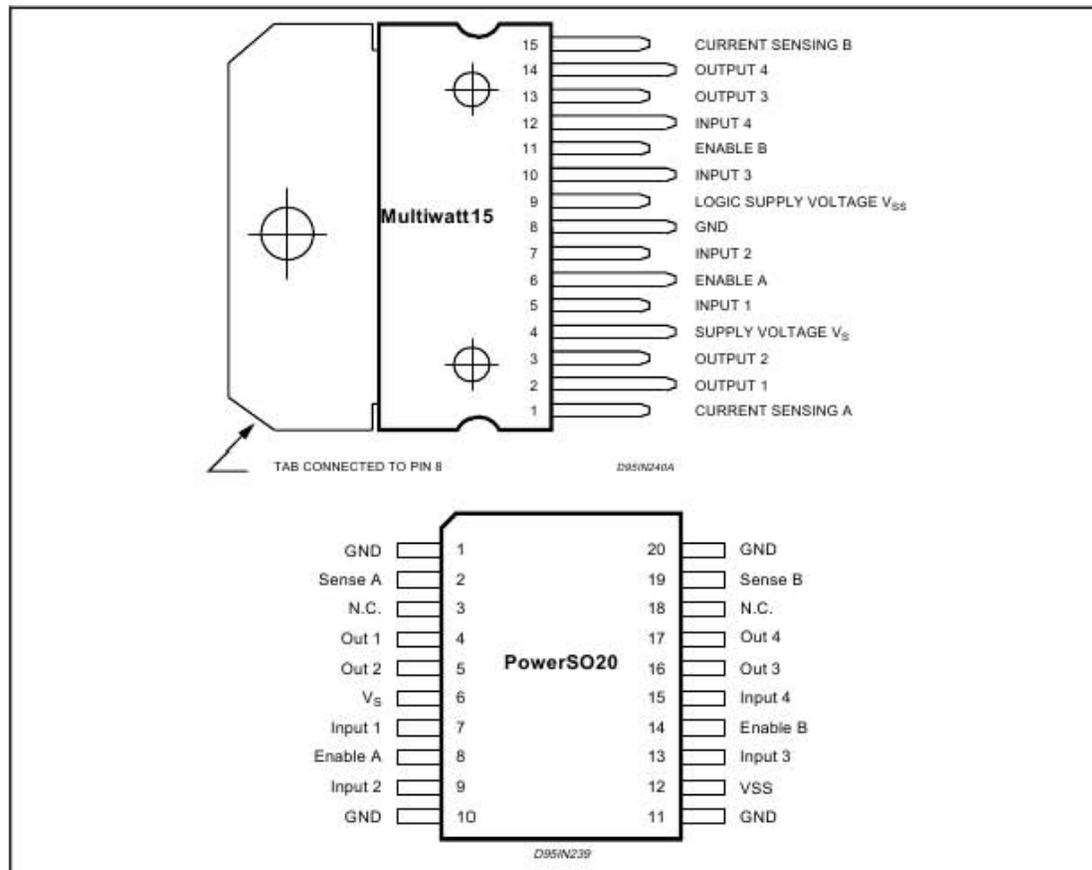


## L298

### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_i, V_{en}$	Input and Enable Voltage	-0.3 to 7	V
$I_O$	Peak Output Current (each Channel)		
	– Non Repetitive ( $t = 100\mu s$ )	3	A
	– Repetitive (80% on –20% off; $t_{on} = 10ms$ )	2.5	A
	– DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^{\circ}C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^{\circ}C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^{\circ}C$

### PIN CONNECTIONS (top view)



### THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	3	$^{\circ}C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	35	$^{\circ}C/W$

(\*) Mounted on aluminum substrate

## PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>S</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V<sub>S</sub> = 42V; V<sub>SS</sub> = 5V, T<sub>J</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>IH</sub> +2.5		46	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>en</sub> = H; I <sub>L</sub> = 0 V <sub>i</sub> = L V <sub>i</sub> = H		13 50	22 70	mA mA
		V <sub>en</sub> = L V <sub>i</sub> = X			4	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>en</sub> = H; I <sub>L</sub> = 0 V <sub>i</sub> = L V <sub>i</sub> = H		24 7	36 12	mA mA
		V <sub>en</sub> = L V <sub>i</sub> = X			6	mA
V <sub>IL</sub>	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V <sub>IH</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.3		V <sub>SS</sub>	V
I <sub>L</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = L			-10	αA
I <sub>IH</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = H * V <sub>SS</sub> -0.6V		30	100	αA
V <sub>en</sub> = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V <sub>en</sub> = H	Enable High Voltage (pins 6, 11)		2.3		V <sub>SS</sub>	V
I <sub>en</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = L			-10	αA
I <sub>en</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = H * V <sub>SS</sub> -0.6V		30	100	αA
V <sub>CEsat(H)</sub>	Source Saturation Voltage	I <sub>L</sub> = 1A I <sub>L</sub> = 2A	0.95	1.35 2	1.7 2.7	V V
V <sub>CEsat(L)</sub>	Sink Saturation Voltage	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V <sub>CEsat</sub>	Total Drop	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	1.80		3.2 4.9	V V
V <sub>sens</sub>	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

# L298

## ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T <sub>1</sub> (V <sub>i</sub> )	Source Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
T <sub>2</sub> (V <sub>i</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
T <sub>3</sub> (V <sub>i</sub> )	Source Current Turn-on Delay	0.5 V <sub>i</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
T <sub>4</sub> (V <sub>i</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
T <sub>5</sub> (V <sub>i</sub> )	Sink Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
T <sub>6</sub> (V <sub>i</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>7</sub> (V <sub>i</sub> )	Sink Current Turn-on Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		1.6		μs
T <sub>8</sub> (V <sub>i</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
f <sub>c</sub> (V <sub>i</sub> )	Commutation Frequency	I <sub>L</sub> = 2A		25	40	KHz
T <sub>1</sub> (V <sub>en</sub> )	Source Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (2); (4)		3		μs
T <sub>2</sub> (V <sub>en</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
T <sub>3</sub> (V <sub>en</sub> )	Source Current Turn-on Delay	0.5 V <sub>en</sub> to 0.1 I <sub>L</sub> (2); (4)		0.3		μs
T <sub>4</sub> (V <sub>en</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
T <sub>5</sub> (V <sub>en</sub> )	Sink Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
T <sub>6</sub> (V <sub>en</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.35		μs
T <sub>7</sub> (V <sub>en</sub> )	Sink Current Turn-on Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>8</sub> (V <sub>en</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

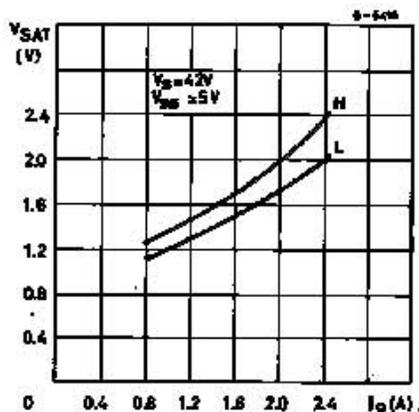
1) Sensing voltage can be -1 V for t > 50 μsec; in steady state V<sub>en</sub> min ≥ -0.5 V.

2) See fig. 2.

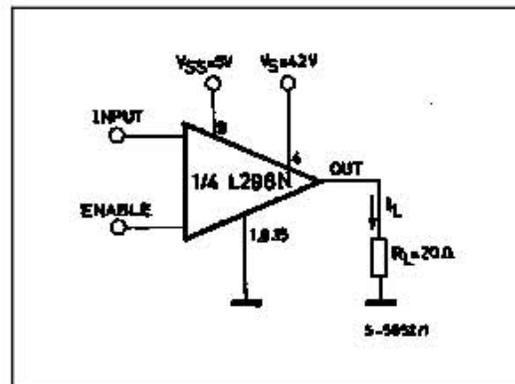
3) See fig. 4.

4) The load must be a pure resistor.

**Figure 1 :** Typical Saturation Voltage vs. Output Current.



**Figure 2 :** Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = H

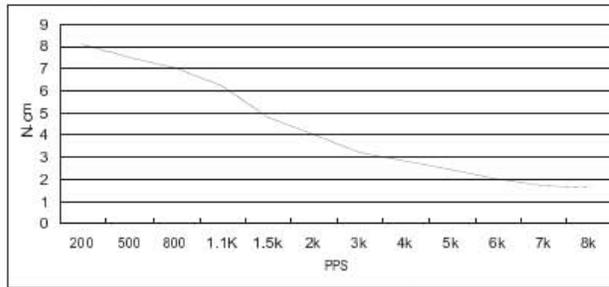
# APÉNDICE B

## HIGH TORQUE HYBRID STEPPING MOTOR SPECIFICATIONS

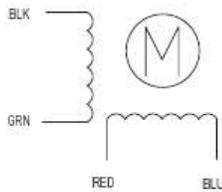
General specifications		Electrical specifications	
Step Angle (°)	1.8	Rated Voltage (V)	10
Temperature Rise (°C)	80 Max (rated current, 2 phase on)	Rated Current (A)	0.5
Ambient temperature (°C)	-20 ~ +50	Resistance Per Phase ( $\pm 10\% \Omega$ )	20 (25°C)
Number of Phase	2	Inductance Per Phase ( $\pm 20\% \text{mH}$ )	13.5
Insulation Resistance	1.00M $\Omega$ , Min (500VDC)	Holding Torque (Kg.cm)	1
Insulation Class	Class B	Detent Torque (g.cm)	80
Max.radial force (N)	28 (20mm from the flange)	Rotor Inertia (g.cm <sup>2</sup> )	10
Max.axial force (N)	10	Weight (Kg)	0.125

● Pull out torque curve:

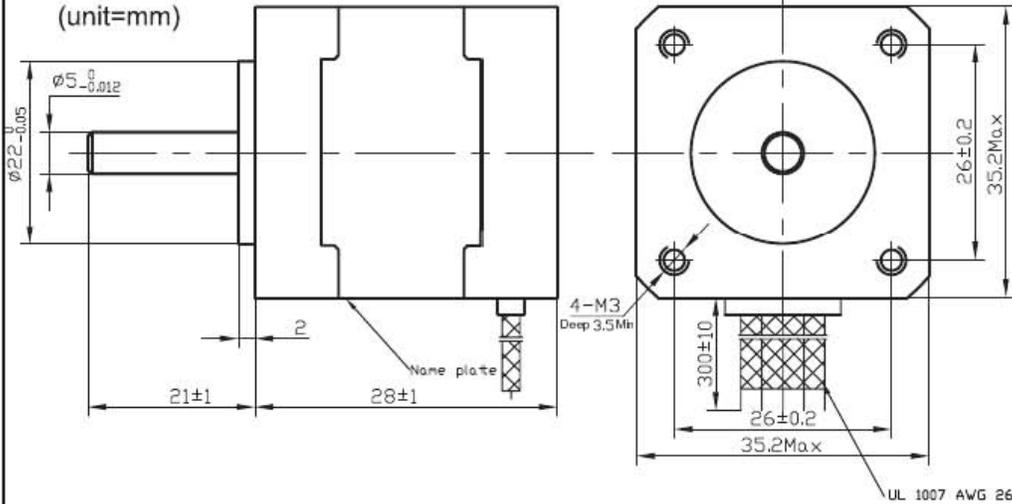
VOLTAGE: 24VDC, CONSTANT CURRENT: 0.5A, HALF STEP



● Wiring Diagram:



● Dimensions:  
(unit=mm)



△					SY35ST28-0504A	TECHNICAL CONDITIONS
△						
REV	REVISIONS	DESCRIPTION	BY	DATE	CHANGZHOU SONGYANG MACHINERY & ELECTRONICS NEW TECHNIC INSTITUTE	030028000
DRAW						
CHECK						
APPROVE						