

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

**SISTEMA PARA CONTROLAR LA VELOCIDAD DE UN
MOTOR DC UTILIZANDO MODULACIÓN DE ANCHO DE
PULSO**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**ALBA YADIRA CORREA ERAS
ESTHER PAOLA REMACHE ORTEGA**

DIRECTOR: ING. ALCÍVAR COSTALES

Quito, Julio del 2006

DECLARACIÓN

Nosotras, Alba Yadira Correa Eras y Esther Paola Remache Ortega declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado de calificación profesional; que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Alba Y. Correa E.

Esther P. Remache O.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Alba Yadira Correa Eras y Esther Paola Remache Ortega, bajo mi supervisión.

Ing. Alcívar Costales

DIRECTOR DE PROYECTO

Estudia y aprende como si
hubieras de vivir siempre. Vive
como si hubieras de morir
mañana.

J. Montalvo Tratado de
Geometría
Moral

AGRADECIMIENTO

Expresamos sinceros agradecimientos a los profesores de la Escuela de Formación Tecnológica y en especial a nuestro Tutor Ing. Alcívar Costales ya que sin su apoyo y guía no habría sido posible la culminación de este proyecto.

Alba y Paola

DEDICATORIA

Dedicado a mis padres y hermano
ya que con su amor, sus
acertados consejos y su apoyo
incondicional he conseguido la
culminación del presente trabajo

Paola

DEDICATORIA

A mi querida madre y a mis hermanos que con su comprensión y cariño me incentivaron a cada momento en el desarrollo del proyecto.

Alba

CAPITULO 1

1. Información general	1
1.1 Características del motor dc	1
1.1.1 Motor dc	1
1.1.2 Principio de funcionamiento del motor dc	1-3
1.1.3 Partes de un motor dc	4
1.1.3.1 Estator	4
1.1.3.1.1 Armazón	4
1.1.3.1.2 Imán permanente	4
1.1.3.1.3 Escobillas	5
1.1.3.2 Rotor	5
1.1.3.2.1 Eje	5
1.1.3.2.2 Núcleo	5-6
1.1.3.2.3 Devanado	6
1.1.3.2.4 Colector	6
1.1.4 Características nominales de un motor	6
1.1.4.1 Potencia nominal	6
1.1.4.2 Tensión nominal	6
1.1.4.3 Alimentación eléctrica	7
1.1.4.4 Sobre intensidad ocasional	7
1.1.4.5 Exceso momentáneo de par	8
1.1.5 Como se escoge un motor	8
1.1.6 Formas de control para motores d.c.	8
1.2 Generador	9
1.3 Modulación de ancho de pulso PWM [7]	10-11
1.4 Proporcional integral derivativa (PID)	11
1.4.1 Proporcional	11
1.4.2 Integral	11
1.4.3 Derivativa	11-12
1.5 Descripción general del PIC 16F877	12
1.5.1 La familia del PIC 16F877	12
1.5.1.1 Variantes principales	12-13

1.5.1.2	Empaquetados	13
1.5.1.3	Nomenclatura	13
1.5.1.4	Oscilador	13-14
1.5.1.4.1	Cristal externo:	15
1.5.1.4.2	Circuito RC externo	15-16
1.5.1.4.3	Rangos	16
1.5.1.4.4	Oscilador externo	16
1.5.1.4.5	Oscilador interno de 4Mhz.	16
1.5.1.4.6	Calibración del oscilador interno	17
1.5.2	Características generales del PIC 16F877	17
1.5.2.1	CPU	17
1.5.2.2	Memoria	17
1.5.2.3	Reset e interrupciones	17
1.5.2.4	Otros	18
1.5.2.5	Periféricos	18
1.5.2.6	Descripción de la CPU	19
1.5.2.6.1	Ciclo de instrucción	19
1.5.3	Recursos específicos	20
1.5.3.1	Temporizador o “timers”	20
1.5.3.2	Watchdog	20
1.5.3.3	Protección ante fallo de alimentación o “brownout”	21
1.5.3.4	Circuito de reloj	21
1.5.3.5	Reinicialización o reset	21
1.5.3.6	Estado de reposo o de bajo consumo	21-22
1.5.3.7	Convertor análogo/digital CAD	22
1.5.3.7.1	Registros de trabajo para la utilización del CAD	22
1.5.3.7.2	Registro ADCON0	23
1.5.3.7.3	Registro ADCON1	23-24
1.5.3.8	Modulador de ancho pulsos (PWM)	24
1.5.3.9	Comparador analógico	24
1.5.3.10	Puertas de entrada y salida digitales	24-25
1.5.3.11	Puertas de comunicación	25
1.5.4	Módulo de entrada de señales analógicas _[2]	25-27

1.5.5 Puertas y registros especiales	27
1.5.5.1 Registro de estado (STATUS)	27-28
1.5.5.2 Registro de control de interrupciones (INT CON)	28-29
1.5.5.3 Registro de opciones	29-30
1.5.5.4 Registro de permiso de interrupciones 1(PIE1)	30-31
1.5.5.5 Registro EECON1	31
1.5.6 Características generales de los timers	31
1.5.6.1 Timer 0 _[1]	31-32
1.5.6.2 Timer 1 _[2]	32-33
1.5.6.2.1 Registro T1CON	33
1.5.6.3 Timer 2 _[2] :	33
1.6 Control de sentido de giro para motores-cc _[13]	33-34

CAPITULO 2

2. Diseño y construcción del sistema del control	35
2.1 Descripción general del control de velocidad	35-36
2.1.1 Etapa de alimentación	37
2.1.1.1 Fuente para el circuito de control	37
2.1.1.2 Fuente para la parte de potencia	37-38
2.1.2 Etapa de ingreso de datos	38
2.1.3 Etapa de visualización de datos en LCD	38-39
2.1.4 Etapa de sensado	39
2.1.4.1 Amplificador operacional de precisión	39-40
2.1.5 Etapa de programación	40
2.1.6 Etapa de acoplamiento	40-42
2.2 Programación del PIC 16F877	43-44
2.2.1 Definición de constantes	45
2.2.2 Inclusión de ficheros	45
2.2.3 Configuración de bits	45
2.2.4 Definición de funciones	46
2.2.5 Definición de variables	46-50

CAPITULO 3

3. Ensamblaje y calibración del sistema del control	51
3.1 Diagramas del sistema de control	51-52
3.2 Dispositivos electrónicos utilizados	53
3.3 Pruebas del sistema de control	54
CONCLUSIONES	55
RECOMENDACIONES	56
BIBLIOGRAFÍA	57-58
ANEXOS	59-77

PRESENTACIÓN

El presente trabajo fue realizado con un objetivo definido: diseñar y construir un sistema de control de tres velocidades para un motor DC utilizando modulación de ancho de pulso .

En el capítulo 1 encontramos la información general de los diversos componentes que nos sirvió de punto de partida hacia la realización de nuestro proyecto. En este capítulo se describen: el principio de funcionamiento de los motores DC, control de velocidad con modulación de ancho de pulso, se realiza un enfoque superficial del PID y del Puente H, además, se realiza un estudio del microcontrolador PIC 16F877.

En el capítulo 2 se describe cada una de las etapas que componen el control de velocidad. En cada una de las etapas se realiza una breve explicación y se presenta su diagrama eléctrico.

En el capítulo 3 encontramos la lista de los componentes utilizados y las pruebas realizadas.

También encontraremos las conclusiones y recomendaciones obtenidas luego de la realización del proyecto.

En la parte final se presentan los anexos correspondientes y las referencias bibliográficas.

RESUMEN

Los requerimientos actuales de optimizar el uso de energía en motores DC que antes se gastaba en los bloques de control tradicionales, nos guiaron a la elaboración de un proyecto el cual nos permita conservar la energía controlando o modulando el ancho de pulso de energía según la demanda.

La información recopilada determino que para diseñar y construir un circuito de control que satisfaga los requerimientos antes mencionados y permita mantener constante la velocidad de un motor DC una alternativa es la modulación de ancho de pulso PWM. Para realizar la modulación y la conversión análogo digital utilizamos el PIC 16F877, que en conjunto con el sensor de velocidad que en nuestro caso es un taco generador podemos tener un control permanente de la velocidad a la que gira el motor.

La señal que obtenemos del taco generador es alterna y de amplitudes variables por lo que esta señal previamente debe ser acondicionada esto lo hacemos utilizando el circuito integrado LM 358, el mismo que toma la señal del taco generador y deja pasar señales positivas y de un valor comprendido entre 0v y 5v apropiadas para ser introducidas en el CAD del PIC para que este realice la modulación mediante el programa implementado. Cabe anotar que la etapa de acoplamiento entre el circuito de control y el circuito de potencia lo realizamos a través del puente h.

El programa grabado en el PIC es el que determina las características de funcionamiento. El programa fue desarrollado utilizando un lenguaje que brinda facilidad de comprensión al usuario como es el lenguaje C. Este programa fue desarrollado utilizando el paquete computacional HI- TECH PICC que es un compilador que no necesita ensamblador, y permite realización de operaciones con punto flotante.

El diseño se realizo de tal manera que el usuario pueda seleccionar por medio de pulsadores cualquiera de los tres valores de velocidad y visualizarlos en un LCD.

INTRODUCCIÓN

Nuestro objetivo es mantener constante la velocidad de un motor DC de acuerdo a una referencia elegida sin importar la variación de carga en el motor, para así solucionar los problemas de conservación de energía, así como también prevenir daños al motor limitando la corriente.

El reporte de investigación que a continuación se presenta, tiene como objetivo diseñar y construir un circuito de control de velocidad de un motor DC : ¿Como realizar el control y mantener constantes tres velocidades en un motor DC al detectar que se aumenta el valor de la carga? ¿Qué tipo de circuito y que componentes pueden implementarse para controlar y visualizar tres valores determinados de velocidad de un motor DC? ¿Cómo funciona este circuito de control? ¿Como detectar el cambio de velocidad (revoluciones) con que gira un motor DC? ¿Qué función realiza el PIC 16F877? ¿Qué ventajas presenta la modulación de ancho de pulso?.

Se determino la utilización del PIC 16F877, ya que tiene ventajas que son: modulador PWM y un conversor análogo digital lo que permite cumplir con el objetivo. Además todos los componentes de nuestro proyecto se encuentran en el mercado local a un precio relativamente bajo.

CAPITULO 1

1. INFORMACION GENERAL

1.1 CARACTERÍSTICAS DEL MOTOR DC

1.1.1 MOTOR DC

El motor es una maquina que convierte energía eléctrica en energía mecánica, accionar un motor DC es muy simple y solo es necesario aplicar la tensión de alimentación entre sus bornes. Los motores DC no pueden ser posicionados y/o enclavados en una posición específica, estos giran a la máxima velocidad y en el sentido que la alimentación aplicada se los permite.

El funcionamiento se basa en la interacción entre el campo magnético del imán permanente y el generado por las bobinas, ya sea una atracción o una repulsión, hacen que el eje del motor comience su movimiento.

1.1.2 PRINCIPIO DE FUNCIONAMIENTO DEL MOTOR DC

La base del funcionamiento de los motores es el principio de la inducción descubierto por el científico e inventor británico Michael Faraday en 1831.

La Ley de Faraday se refiere a la inducción de una fuerza electromotriz debido a la variación de flujo.

$$E = \frac{\Delta\phi}{\Delta t}$$

Si hay varias espiras será: $E = - N \frac{\Delta\phi}{\Delta t}$

Donde:

E = Fem inducida

N = numero de espiras de la bobina

$\frac{\Delta\phi}{\Delta t}$ = Variación de flujo con respecto al tiempo

El signo menos (-) se deriva de la ley de Lenz que nos dice: si una corriente inducida fluye, su dirección es siempre de tal forma que se opone al cambio de flujo que la produjo.

Por lo tanto se induce una corriente en la espira cuando hay variación de flujo en la espira, en este caso ello ocurre cuando hay movimiento relativo entre la espira y el imán.

Si un conductor se mueve a través de un campo magnético, o si está situado en las proximidades de otro conductor por el que circula una corriente de intensidad variable, se establece o se induce una corriente eléctrica en el primer conductor, consiste en una espira de alambre que gira entre los polos de un imán. Los extremos de la espira están conectados con delgas del colector que a su vez, hacen contacto con las escobillas. Las escobillas tienen conductores que van a una fuente de tensión de corriente continua.

Cuando la espira esta en la posición 1, la corriente que pasa por la espira hace que el lado superior de la espira se convierta en polo norte y la parte inferior en polo sur, según la regla de la mano izquierda. Los polos magnéticos de la espira serán atraídos por los polos opuestos correspondientes del campo, como resultado, la espira girara en el sentido de las agujas del reloj, colocando frente a frente a los polos contrarios, la espira ha girado 90 grados hasta la posición 2, se produce la conmutación y la corriente de la espira se invierte de sentido. A raíz de esto el campo magnético generado por la espira también se invierte. Ahora están frente a frente polos iguales, lo que significa que se rechazan, y la espira sigue girando para tratar de acercar los polos opuestos. Al girar 180 grados, mas allá de la posición 2, la espira se encuentra en posición 3. Entonces la situación

es la misma que cuando se hallaba en la posición 2. Se produce de nuevo la conmutación y la espira sigue girando.

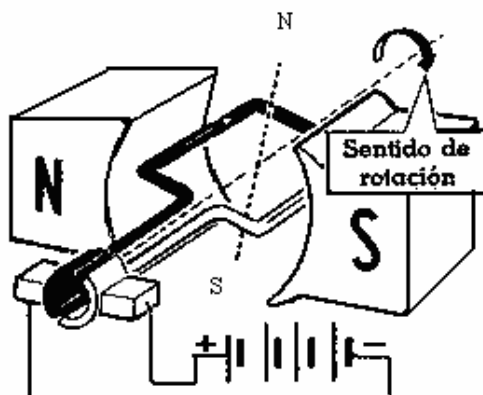


Figura 1.1 Posición 1

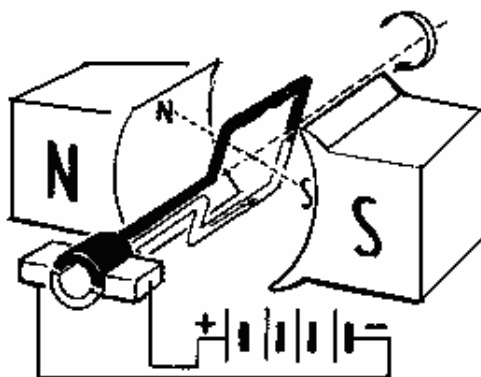


Figura 1.2 Posición 2

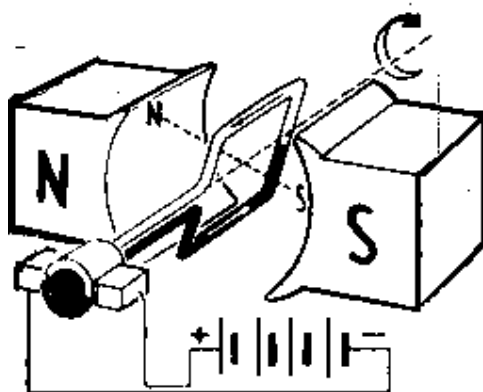


Figura 1.3 Posición 3

1.1.3 PARTES DE UN MOTOR DC

Un motor de corriente continua está compuesto de un estator y un rotor. (Figura 1.4).

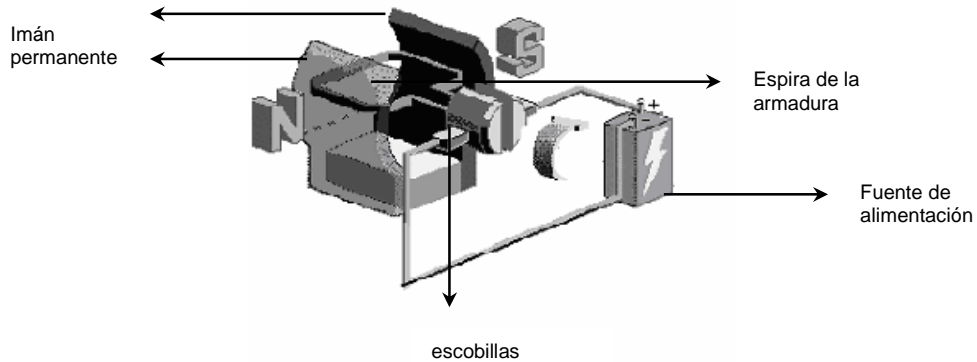


Figura 1.4 Partes del motor DC

1.1.3.1 Estator

Constituye la parte fija o estática de la máquina. Su función es suministrar el flujo magnético que será usado por el bobinado del rotor para realizar su movimiento giratorio. Está formado por: Armazón, imán permanente, escobillas.

1.1.3.1.1 Armazón

Denominado también yugo, tiene dos funciones primordiales: servir como soporte y proporcionar una trayectoria de retorno al flujo magnético del rotor y del imán permanente, para completar el circuito magnético.

1.1.3.1.2 Imán permanente

Compuesto de material ferromagnético altamente remanente, se encuentra fijado al armazón o carcasa del estator. Su función es proporcionar un campo magnético uniforme al devanado del rotor o armadura, de modo que interactúe con el campo formado por el bobinado, y se origine el movimiento del rotor como resultado de la interacción de estos campos.

1.1.3.1.3 Escobillas

Las escobillas están fabricadas de carbón, y poseen una dureza menor que la del colector, para evitar que éste se desgaste rápidamente. Se encuentran albergadas por los porta escobillas. Ambos, escobillas y porta escobillas, se encuentran en una de las tapas del estator.

La función de las escobillas es transmitir la tensión y corriente de la fuente de alimentación hacia el colector y, por consiguiente, al bobinado del rotor. La función del porta escobillas es mantener a las escobillas en su posición de contacto firme con los segmentos del colector. Esta función la realiza por medio de resortes, los cuales hacen una presión moderada sobre las escobillas contra el colector. Esta presión debe mantenerse en un nivel intermedio pues, de ser excesiva, la fricción desgastaría tanto a las escobillas como al colector; por otro lado, de ser mínima esta presión, se produciría lo que se denomina "chisporroteo", que es cuando aparecen chispas entre las superficies del colector y las escobillas, debido a que no existe un buen contacto.

1.1.3.2 Rotor

Constituye la parte móvil del motor es el dispositivo que gira en el centro del motor es el que proporciona el torque para mover la carga, está formado por: eje, núcleo, devanado, colector.

1.1.3.2.1 Eje

Formado por una barra de acero fresada. Imparte la rotación al núcleo, devanado y al colector.

1.1.3.2.2 Núcleo

Se localiza sobre el eje. Fabricado con capas laminadas de acero, su función es proporcionar un trayecto magnético entre los polos para que el flujo magnético del devanado circule. Las laminaciones tienen por objeto reducir las corrientes parásitas en el núcleo. El acero del núcleo debe ser capaz de mantener bajas las

pérdidas por histéresis. Este núcleo laminado contiene ranuras a lo largo de su superficie para albergar al devanado de la armadura (bobinado).

1.1.3.2.3 Devanado

Consta de bobinas aisladas entre sí y entre el núcleo de la armadura. Estas bobinas están alojadas en las ranuras, y están conectadas eléctricamente con el colector, el cual debido a su movimiento rotatorio, proporciona un camino de conducción conmutado.

1.1.3.2.4 Colector

Denominado también conmutador, está constituido de láminas de material conductor (delgas), separadas entre sí y del centro del eje por un material aislante, para evitar cortocircuito con dichos elementos. El colector se encuentra sobre uno de los extremos del eje del rotor, de modo que gira con éste y está en contacto con las escobillas. La función del colector es recoger la tensión producida por el devanado inducido, transmitiéndola al circuito por medio de las escobillas.

1.1.4 CARACTERÍSTICAS NOMINALES DE UN MOTOR

1.1.4.1 Potencia nominal

La potencia nominal de un motor es la potencia entregada en el eje (en W) en condiciones nominales.

1.1.4.2 Tensión nominal

La tensión nominal es la tensión entre bornes de la máquina en condiciones de referencia definidas.

Para los motores en general con la tensión nominal el motor entrega la potencia nominal correspondiente a la velocidad denominada base.

1.1.4.3 Alimentación eléctrica

En los motores de corriente continua alimentados por medio de un convertidor estático de potencia, la ondulación de la tensión y de la corriente influyen sobre las características de funcionamiento de la máquina.

Al comparar un motor con este tipo de alimentación, con otro que se alimenta con una fuente de corriente continua pura, se observa que las pérdidas y el calentamiento se incrementan y la conmutación se dificulta para el primer caso.

Para reducir la ondulación en algunos casos se incluyen en el circuito de alimentación inductancias adicionales (que cumplen la función de filtros).

1.1.4.4 Sobre intensidad ocasional

Los motores de corriente continua deben poder soportar para la máxima velocidad, con la plena excitación y su correspondiente tensión de armadura, una corriente igual a 1.5 veces la corriente nominal durante un tiempo no menor de 1 minuto.

Para máquinas grandes se puede (previo acuerdo entre el constructor y el comprador), adoptar un tiempo menor, pero este no podrá ser inferior a 30 s.

La posibilidad de que se presente una sobre intensidad ocasional en una máquina rotante, se indica con el objeto de coordinar la máquina con los dispositivos de comando y protección, no estableciendo las normas ensayos para verificar esta condición.

Los efectos del calentamiento de los arrollamientos de la máquina varían aproximadamente como el producto del tiempo por el cuadrado de la corriente, en consecuencia una corriente superior a la nominal produce un incremento de la temperatura de la máquina.

Salvo que se especifique lo contrario, se supone que la máquina no será sometida a este tipo de sobrecargas, más que durante unos pocos cortos períodos durante toda su vida.

1.1.4.5 Exceso momentáneo de par

Para motores de corriente continua, el par se puede expresar en función de la sobre intensidad y el exceso momentáneo de par se debe acordar entre el fabricante y el comprador.

Algunas normas para aplicaciones particulares fijan valores de exceso momentáneo de potencia.

En muchos casos el usuario indica con grado importante de detalle como varían par y velocidad a lo largo del ciclo de utilización.

1.1.5 COMO SE ESCOGE UN MOTOR

Como hemos visto, no todos los motores pueden ser utilizados para toda clase de trabajo y cada actividad requiere un tipo de motor. Para elegir un motor hay que tener en cuenta:

- La carga de trabajo (Potencia)
- La clase de servicio.
- El ciclo de trabajo.
- Los procesos de arranque, frenado e inversión.
- La regulación de velocidad.
- Las condiciones de la red de alimentación
- La temperatura ambiente.

1.1.6 FORMAS DE CONTROL PARA MOTORES D.C.

- Control mediante Puente H con regulador de velocidad.
- Control mediante microrelés.
- Control de velocidad con modulación de ancho de pulso.
- Control de velocidad utilizando tiristores a base de la regulación del ángulo de disparo.
- Control con un servomotor.

1.2 GENERADOR [6]

Es una maquina rotativa que convierte energía mecánica en energía eléctrica.

El generador elemental consiste en una espira de alambre colocada de manera que pueda girar dentro de un campo magnético fijo produciendo una tensión inducida en la espira. Para conectar la espira con un circuito externo y aprovechar la FEM inducida se utilizan contactos deslizantes.

Las piezas polares son los polos norte y sur del imán que suministrará el campo magnético. La espira de alambre que gira a través del campo magnético se denomina armadura o inducido los extremos del inducido están conectados a unos anillos que se denominan anillos rozantes o de contacto para recoger la electricidad producida en la armadura y transportarla al circuito externo.(Figura 1.5)

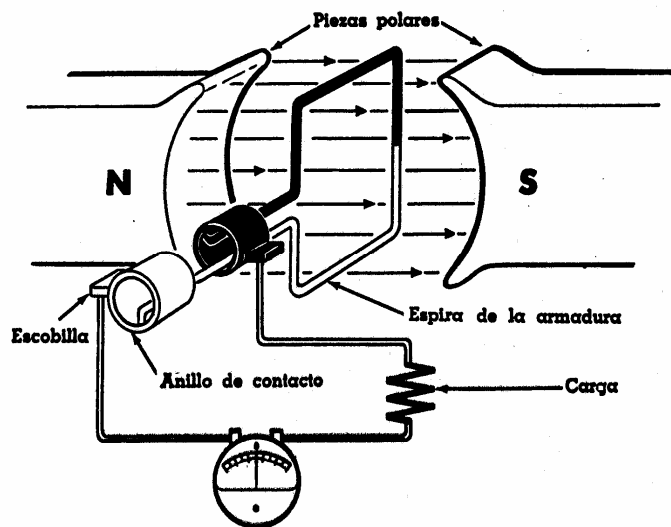


Figura 1.5 Partes del Generador

1.3 MODULACIÓN DE ANCHO DE PULSO PWM [7]

. La modulación de ancho de pulso es una técnica utilizada para controlar dispositivos, o para proveer un voltaje variable de corriente continua. Algunas aplicaciones en las que se utiliza PWM son controles de motores, de iluminación y de temperatura. La señal generada tendrá frecuencia fija y tiempos de encendido y apagado variables. En otras palabras, el período de la señal se mantendrá constante, pero la cantidad de tiempo que se mantiene en alto y bajo dentro de un período puede variar. (Figura 1.6).

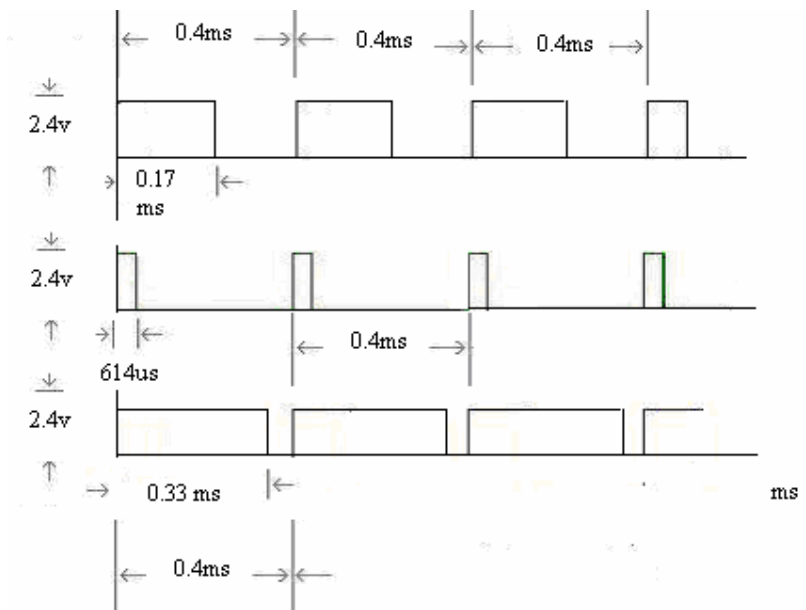


Figura 1.6 Modulación de ancho de pulso

En una onda cuadrada se varía el ancho de pulso positivo, manteniendo constante la frecuencia, ya que de esta manera el valor medio de la onda resultante es variable dependiendo de la duración del pulso positivo de la misma, es decir se varía el tiempo total (la frecuencia del ciclo). Con esta variación de frecuencia se varía el valor medio de la onda de salida, se intenta obtener un impulso cuyo nivel alto tenga un ancho de pulso variable dentro del intervalo del período de trabajo.

En PWM el ancho del pulso es proporcional al valor de la señal que se desea modular.

Cuando más ancho es el pulso, mayor es la velocidad del motor y viceversa, con la Modulación PWM la fuerza del motor se ve poco afectada incluso a velocidades mínimas.

1.4 PROPORCIONAL INTEGRAL DERIVATIVA (PID)

Para mantener una salida constante, el controlador mide la salida y controla la entrada (usualmente un sensor) y aplica modelos matemáticos a la diferencia de las dos (denominada error).

El error es luego tratado en tres diferentes maneras simultáneamente.

1.4.1 PROPORCIONAL

El error es multiplicado por una constante proporcional P, y enviada a la salida. P representa la banda donde la salida del controlador es proporcional al error del sistema.

1.4.2 INTEGRAL

El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un periodo de tiempo. Luego es multiplicado por una constante I y se agrega a la salida proporcional. I representa el error de estado estacionario del sistema.

1.4.3 DERIVATIVA

El error se deriva con respecto al tiempo y se multiplica por una constante D y luego se suma a las señales anteriores. La función de la derivativa es gobernar la respuesta de control a los cambios en el sistema ya que una mayor derivativa corresponde a un cambio más rápido y el controlador puede responder acordeamente.

La fórmula de cálculo de la proporcional integral derivativa es la siguiente:

$$\text{Out} = \kappa_p * (e + \kappa_i \Sigma[\text{errores anteriores}]) + \kappa_d * [e - e_{-1}]$$

κ_p = constante proporcional
 κ_d = constante derivativa
 e = error
 $e_{.1}$ = error anterior
 κ_i = constante integral
 e = set point – realimentación

Donde:

Set point: son las velocidades programadas en el PIC
realimentación: es la velocidad a la que gira el motor.

1.5 DESCRIPCIÓN GENERAL DEL PIC 16F877

1.5.1 LA FAMILIA DEL PIC 16F877

El microcontrolador PIC 16F877 de Microchip pertenece a una gran familia de microcontroladores de 8 bits (bus de datos) que tienen las siguientes características generales que los distinguen de otras familias:

- Arquitectura Harvard: dispone de dos memorias independientes una contiene solo las instrucciones y otra solo datos.
- Tecnología RISC: juego de instrucciones reducido
- Tecnología CMOS

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa y por lo tanto en la velocidad de ejecución.

1.5.1.1 Variantes principales

Los microcontroladores que produce Microchip cubren una amplio rango de dispositivos cuyas características pueden variar como sigue:

- Empaquetado (desde 8 terminales hasta 68 terminales)

- Tecnología de la memoria incluida (EPROM, ROM, Flash)
- Voltajes de operación (desde 2 v. h asta 5.5v.)
- Frecuencia de operación (Hasta 20 Mhz)

1.5.1.2 Empaquetados

Aunque cada empaquetado tiene variantes, especialmente en lo relativo a las dimensiones del espesor del paquete, en general se pueden encontrar paquetes tipo PDIP (Plastic Dual In Line Package), PLCC (Plastic Leaded Chip Carrier), QFP (Quad Flat Package) y SOIC (Small Outline I.C.).

1.5.1.3 Nomenclatura

En el nombre específico del microcontrolador pueden aparecer algunas siglas que dependen del rango de voltaje manejado y del tipo de memoria ROM incluida, como se muestra en la siguiente tabla 1.1:

Tipo de memoria	Rango de voltaje	
	Estándar (4.5 a 6 volts)	Extendido (2.5 a 6 volts)
EPROM, OTP	PIC16 C XXX	PIC16 LC XXX
ROM	PIC16 CR XXX	PIC16 LCR XXX
Flash	PIC16 F XXX	PIC16 LF XXX

Tabla 1.1 Rango de voltaje

1.5.1.4 Oscilador

Los PIC de rango medio permiten hasta 8 diferentes modos para el oscilador. El usuario puede seleccionar alguno de estos 8 modos, programando 2 bits de configuración del dispositivo denominados: FOSC1 y FOSC0, ubicados en un registro especial de configuración en la localidad 2007H de la memoria de programa (Tabla 1.2).

13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP1	CP0	DEBUG	-	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0

Tabla 1.2 Palabra de configuración (2007H):

En algunos de estos modos el usuario puede indicar que se genere o no una salida del oscilador (CLKOUT) a través de un terminal de Entrada / salida. Los modos de operación se muestran en la Tabla 1.3:

FOSC1	FOSC0	Modo de operación del oscilador	
0	0	LP	Baja frecuencia (y bajo consumo de potencia)
0	1	XT	Cristal / Resonador cerámico externos, (Media frecuencia)
1	0	HS	Alta velocidad (y alta potencia) Cristal/resonador
1	1	RC	Resistencia / capacitor externos

Tabla 1.3 Modo de operación del oscilador

Algunos PIC's poseen un modo de oscilación que les permite usar una resistencia y un capacitor interno calibrados para 4 Mhz. Los tres modos LP, XT y HS usan un cristal o resonador externo, la diferencia sin embargo es la ganancia de los drivers internos, lo cual se ve reflejado en el rango de frecuencia admitido y la potencia consumida. En la tabla 1.4 se muestran los rangos de frecuencia así como los capacitores recomendados para un oscilador en base a cristal.

Modo	Frecuencia típica	Capacitores recomendados	
		C1	C2
LP	32 khz	68 a 100 pf	68 a 100 pf
	200 khz	15 a 30 pf	15 a 30 pf
XT	100 khz	68 a 150 pf	150 a 200 pf
	2 Mhz	15 a 30 pf	15 a 30 pf
	4 Mhz	15 a 30 pf	15 a 30 pf
HS	8 Mhz	15 a 30 pf	15 a 30 pf
	10 Mhz	15 a 30 pf	15 a 30 pf
	20 Mhz	15 a 30 pf	15 a 30 pf

Tabla 1.4 Capacitores recomendados para un oscilador en base a cristal

1.5.1.4.1 Cristal externo:

En los tres modos mostrados en la tabla anterior se puede usar un cristal o resonador cerámico externo. En la figura 1.7 se muestra la conexión de un cristal a los terminales OSC1 y OSC2 del PIC.

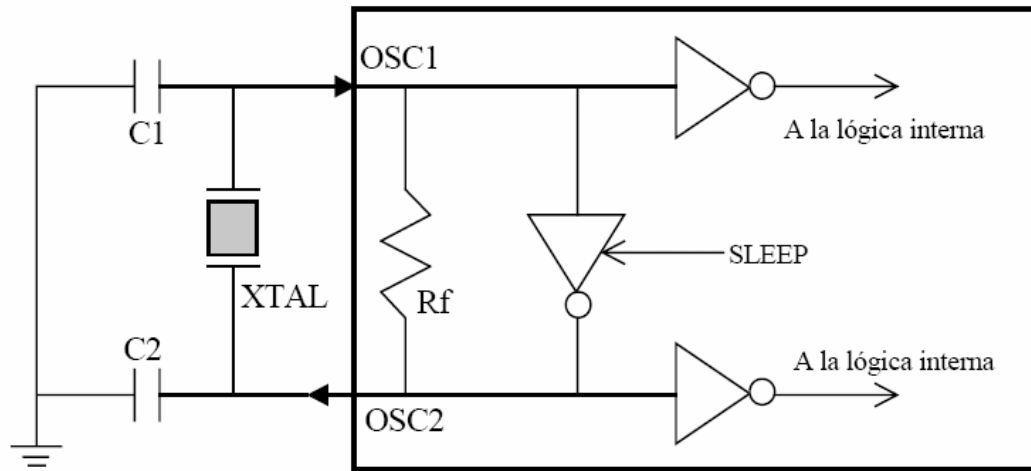


Figura 1.7 Conexión del cristal al PIC

1.5.1.4.2 Circuito RC externo

En los modos RC y EXTRC el PIC puede generar su señal oscilatoria basado en un arreglo RC externo conectado al terminal OSC1 como se muestra en la figura 1.8.

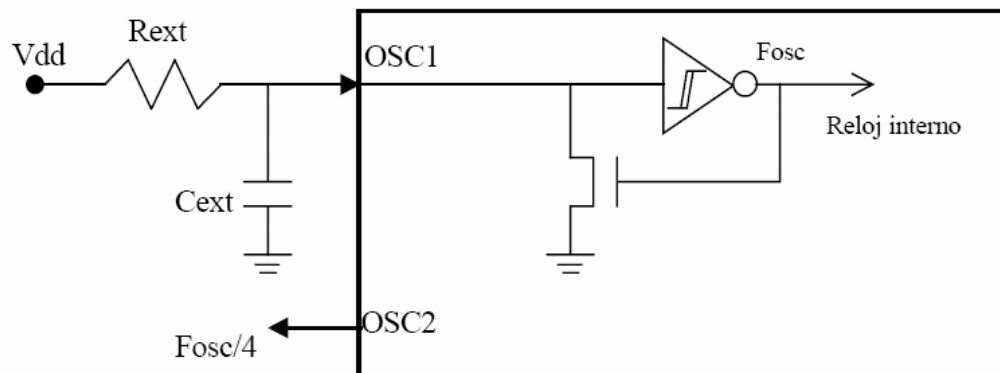


Figura 1.8 arreglo RC externo para OSC1

Este modo sólo se recomienda cuando la aplicación no requiera una gran precisión en la medición de tiempos.

1.5.1.4.3 Rangos

La frecuencia de oscilación depende de los valores de R_{ext} , C_{ext} , y del voltaje de la fuente V_{dd} . Los rangos admisibles para resistencia y capacitor son:

R_{ext}: de 3 a 100 Kohms

C_{ext}: mayor de 20 pf

1.5.1.4.4 Oscilador externo

También es posible conectar una señal de reloj generada mediante un oscilador externo al terminal OSC1 del PIC. Para ello el PIC deberá estar en uno de los tres modos que admiten cristal (LP, XT o HS). La conexión se muestra en la figura 1.9.

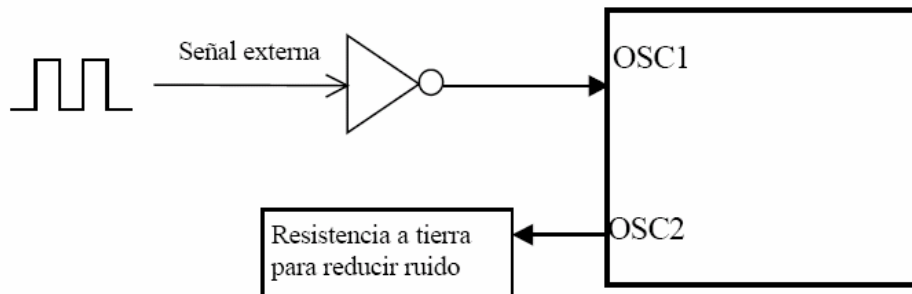


Figura 1.9 Oscilador externo conectado a OSC1

1.5.1.4.5 Oscilador interno de 4Mhz.

En los PIC's que poseen este modo de oscilación, (modo INTRC) el PIC usa un arreglo RC interno que genera una frecuencia de 4 Mhz con un rango de error calibrable de $\pm 1.5\%$.

1.5.1.4.6 Calibración del oscilador interno

El fabricante ha colocado un valor de calibración para estos bits en la última dirección de la memoria de programa. Este dato ha sido guardado en la forma de una instrucción RETLW XX. Si no se quiere perder este valor al borrar el PIC (en versiones EPROM con ventana) primero se deberá leer y copiar.

1.5.2 CARACTERÍSTICAS GENERALES DEL PIC 16F877

La siguiente es una lista que presenta las características del PIC 16F877

1.5.2.1 CPU

- Tecnología RISC
- Set de 35 instrucciones.
- Todas las instrucciones se ejecutan en un ciclo de reloj, excepto los saltos que requieren dos.
- Frecuencia de operación de 0 a 20 MHz (200 nseg de ciclo de instrucción).
- Opciones de selección del oscilador.

1.5.2.2 Memoria

- Hasta 8k x 14 bits de memoria Flash de programa
- Hasta 368 bytes de memoria de datos (RAM)
- Hasta 256 bytes de memoria de datos EEPROM
- Lectura / escritura de la CPU a la memoria flash de programa
- Protección programable de código
- Stack de hardware de 8 niveles

1.5.2.3 Reset e interrupciones

- Hasta 14 fuentes de interrupción
- Reset de encendido (POR)
- Timer de encendido (PWRT)
- Timer de arranque del oscilador (OST)
- Sistema de vigilancia Watchdog timer.

1.5.2.4 Otros

- Modo SLEEP de bajo consumo de energía.
- Programación y depuración serie "In-Circuit" (ICSP) a través de dos terminales.
- Rango de voltaje de operación de 2.0 a 5.5 voltios.
- Alta disipación de corriente de la fuente: 25mA.
- Rangos de temperatura: Comercial, Industrial y Extendido.
- Bajo consumo de potencia:
 - Menos de 0.6mA a 3V, 4 Mhz
 - 20 μ A a 3V, 32 Khz
 - Menos de 1 μ A corriente de standby (modo SLEEP).

1.5.2.5 Periféricos

- 3 a 5 puertos paralelos PORT A, B, C, D con líneas digitales programables individualmente.
- 3 Timers:
 - Timer 0 Contador/ temporizador de 8 bits con preescalador de 8 bits.
 - Timer 1 Contador/ temporizador de 16 bits con preescalador.
 - Timer 2 Contador/ temporizador de 8 bits con preescalador y post-escalador de 8 bits y registro de periodo.
- 2 Módulos CCP
 - Captura 16 bits, 1.5 η seg de resolución máxima.
 - Comparación 16 bits, 200 η seg de resolución máxima.
 - PWM 10 bits.
- Convertidor A/D de 10 bits, hasta 8 canales
- Puerto serie
- Puerto paralelo esclavo
- Temporizador

1.5.2.6 Descripción de la CPU

La CPU es la responsable de la interpretación y ejecución de la información (instrucciones) guardada en la memoria de programa. Muchas de estas instrucciones operan sobre la memoria de datos. Para operar sobre la memoria de datos además, si se van a realizar operaciones lógicas o aritméticas, requieren usar la Unidad de Lógica y Aritmética (ALU). La ALU controla los bits de estado (Registro STATUS), los bits de este registro se alteran dependiendo del resultado de algunas instrucciones.

1.5.2.6.1 Ciclo de instrucción

El registro Program Counter (PC) es gobernado por el ciclo de instrucción como se muestra en la figura 1.10. Cada ciclo de instrucción la CPU lee (ciclo Fetch) la instrucción guardada en la memoria de programa apuntada por PC y al mismo tiempo ejecuta la instrucción anterior, esto debido a una cola de instrucciones que le permite ejecutar una instrucción mientras lee la próxima:

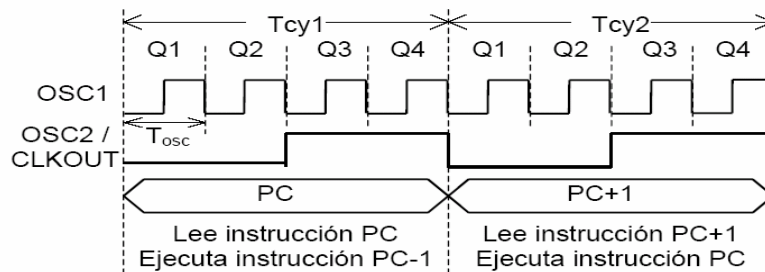


Figura 1.10 Ciclo de instrucción del registro PC

Como puede verse, cada ciclo de instrucción ($T_{cy} = 4T_{osc}$) se compone a su vez de cuatro ciclos del oscilador ($T_{osc} = 1/F_{osc}$). Cada ciclo Q provee la sincronización para los siguientes eventos:

Q1: Decodificación de la instrucción

Q2: Lectura del dato (si lo hay)

Q3: Procesa el dato

Q4: Escribe el dato

Debido a esto cada ciclo de instrucción consume 4 ciclos de reloj, de manera que si la frecuencia de oscilación es F_{osc} , T_{cy} será $4/F_{osc}$.

1.5.3 RECURSOS ESPECÍFICOS

Los principales recursos específicos que incorporan los microcontroladores son:

1.5.3.1 Temporizador o “timers”

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempo se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a cero, momento en el que se produce un aviso. Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en algunas de los terminales del microcontrolador, el mencionado registro se va incrementado o decrementado al ritmo de dichos impulsos.

1.5.3.2 Watchdog

Un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Watchdog, consiste en un temporizador que, cuando se desborda y pasa por cero, provoca un reset automáticamente o una reinicialización en el sistema cuando el programa queda bloqueado. Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Watchdog antes de que provoque el reset. Es un contador que funciona con los impulsos de su propio oscilador y que provoca un reset cuando se desborda en funcionamiento normal. Si el desbordamiento se produce cuando el microcontrolador se halla en estado de reposo, se despierta y sigue su comportamiento normal.

1.5.3.3 Protección ante fallo de alimentación o “brownout”

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD), es inferior a un voltaje mínimo (“brownout”). Mientras el voltaje de alimentación sea inferior al de “brownout” el dispositivo se mantendrá reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

1.5.3.4 Circuito de reloj

Encargado de generar los impulsos que sincronicen el funcionamiento de todos los sistemas.

1.5.3.5 Reinicialización o reset

Los bits TO# y PD# del Registro de Estado toman un valor determinado en cada tipo de Reset. La generación del Reset ocurre como consecuencia de la salida de un nivel lógico de la puerta OR1, la cual es controlada por todas las causas que reinician al PIC. El Vector de RESET ocupa la dirección 0000h y el vector de interrupción la 0004h, igual que en el PIC 16F84.

El temporizador OST produce un retardo de 1024 ciclos de la frecuencia aplicada por el terminal OSC, con el objetivo de asegurar que el cristal de cuarzo del oscilador principal esta estabilizado y en marcha.

1.5.3.6 Estado de reposo o de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Los microcontroladores disponen de una instrucción especial SLEEP en los PIC, que le pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho

estado se detiene el reloj principal y se “congelan” sus circuitos asociados como los temporizadores, el conversor A/D etc., quedando sumido en un profundo “sueño” el microcontrolador. Al activarse una interrupción ocasionado por el acontecimiento esperado el microcontrolador se despierta y reanuda su trabajo. Al entrar el microcontrolador en Modo de reposo o SLEEP, se detiene el oscilador interno y al no existir la señal Fosc/4 (ciclo de instrucción) deja de funcionar el TIMER2.

RB6: Recibe los impulsos de reloj.

RB7: Línea de datos con los bits en serie.

El sistema queda congelado y el consumo de energía se reduce al mínimo.

1.5.3.7 Conversor análogo/digital CAD

El microcontrolador PIC 16F877 posee un conversor A/D de 10 bits de resolución y 8 canales de entrada en los modelos con 40 terminales. La resolución que tiene cada bit procedente de la conversión tiene un valor que es función de la tensión de referencia Vref, de acuerdo con la fórmula:

Resolución: $(V_{ref+} - V_{ref-}) / 1024 = V_{ref} / 1024$

La tensión de referencia determina los límites máximo y mínimo de la tensión analógica que se puede convertir.

El Conversor A/D es el único dispositivo que puede funcionar en modo reposo (SLEEP), para ello el reloj del Conversor deberá conectarse al oscilador RC interno.

1.5.3.7.1 Registros de trabajo para la utilización del conversor análogo digital

- ADRESH: Parte alta del resultado de la conversión.
- ADRESL: Parte baja del resultado de la conversión.
- ADCON0: Registro del control 0.
- ADCON1: Registro de control 1.

En la pareja de registros ADRESH, ADRESL se deposita el resultado de la conversión, que al estar compuesto por 10 bits, solo son significativos 10 de los bits de dicha pareja.

1.5.3.7.2 Registro ADCON0

El registro ADCON0 (Tabla 1.5) controla la operación de C A/D, mientras que ADCON1 sirve para configurar los terminales de la puerta A como entradas analógicas o E/S digitales.

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE #	-	ADON
-------	-------	------	------	------	-----------	---	------

Tabla 1.5 Registro ADCON0

Estos valores determinan las siguientes características:

El bit GO/DONE# es el bit de estado de la conversión. Poniéndolo a 1 se inicia la conversión y mientras este a 1 está realizándose dicha operación. Cuando pasa a 0 confirma el final de la conversión y la puesta del resultado en la pareja de registros ADRESH:L

El bit ADON sirve para activar el C A/D poniéndole a 1 y para inhibir su funcionamiento poniéndolo a 0.

- La frecuencia del reloj que se emplea en la conversión $F_{osc} / 32$.
- El canal seleccionado para introducir la señal analógica a convertir es el canal 4 (RA5 / an4).
- GO/DONE # = 0 Confirma el final de la conversión y la puesta del resultado en la pareja de registros ADRESH - ADRESL
- ADON = 1 Con este valor se activa el C A/D

1.5.3.7.3 Registro ADCON1

ADFM = 1 Determina que el resultado esta justificado en el registro ADRESH que tiene sus 6 bits de más peso a cero.(Tabla 1.6).

ADFM				PCFG3	PCFG2	PCFG1	PCFG0
------	--	--	--	-------	-------	-------	-------

Tabla 1.6 Registro ADCON1

La combinación PCFG3 -0 = 0001 determina que el voltaje de referencia debe ser introducido por el canal 3 (AN3 / RA3).

Los microcontroladores que incorporan un conversor Analógico/Digital pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del C A/D diversas señales analógicas desde los terminales del circuito integrado. Se puede enviar y recibir señales analógicas.

1.5.3.8 Modulador de ancho pulsos (PWM)

Con este modo de trabajo se consiguen impulsos lógicos cuyo ancho del nivel alto es de duración variable, que son de enorme aplicación en el control de dispositivos como los motores y los TRIACS.

Son circuitos que proporcionan en su salida pulsos de ancho variable, que se ofrecen al exterior a través de los terminales del encapsulado, que básicamente consiste en enviar un tren de pulsos a los terminales de control, los cuales varían en mayor o menor medida el tiempo en que se mantiene el nivel alto de esos pulsos, manteniendo la frecuencia constante. Con esta forma de control, la cantidad de corriente sigue siendo la misma, la tensión no varía y en consecuencia el torque del motor se mantiene, que es justamente lo que estábamos buscando.

1.5.3.9 Comparador analógico

Algunos modelos de microcontroladores disponen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los terminales del microcontrolador. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra. Es decir para verificar el valor de una señal analógico.

1.5.3.10 Puertas de entrada y salida digitales

Todos los microcontroladores destinan algunas de sus terminales a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho

formando puertas. Las líneas digitales de las puertas pueden configurarse como entrada o como salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

La principal utilidad de las terminales que posee la cápsula que contiene un microcontrolador es soportar las líneas de entrada/salida que comunican al computador interno con los periféricos exteriores. Las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

1.5.3.11 Puertas de comunicación

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Las puertas son las líneas de entrada y salida que se agrupan en conjunto de 8 y manejan la información en paralelo.

La memoria de datos RAM contiene el banco de registros específicos y el banco de registros de propósito general.

1.5.4 MÓDULO DE ENTRADA DE SEÑALES ANALÓGICAS_[2]

Una de las características más importantes de los microcontroladores es la incorporación de conversores análogo digitales, simplificando de esta manera toda la circuitería externa que era necesario para manejar las señales análogas.(Tabla 1.7).

Pasos para realizar una conversión análogo/digital:

1) Configurar el módulo C A/D

- Configurar los terminales que actuarán como entradas analógicas, las que trabajan como entradas / salidas digitales y las usadas para la tensión de referencia (ADCON1).
- Seleccionar el reloj de la conversión (ADCON0)
- Seleccionar el canal de entrada A/D (ADCON0)
- Activar el módulo A/D (ADCON0)

- 2) Activar si se desea, la interrupción escribiendo sobre PIE1 y PIR1
 - Borrar la señalización ADIF
 - Poner a 1 el bit ADIE
 - Poner a 1 los bits habilitados GIE, PEIE
- 3) Tiempo de espera para que transcurra el tiempo de adquisición
- 4) Inicio de conversión

Poner en 1 el bit GO/DONE#/ ADCON0
- 5) Tiempo de espera para completar la conversión A/D que puede detectarse

Por la exploración del bit GO/DONE#, que al completarse la conversión pasa a valer cero.

Esperando a que se produzca la interrupción si se ha programado, al finalizar la conversión.

Aunque no se permita interrupción, el señalizador ADIF se pondrá a 1 al finalizar la conversión.
- 6) Leer el resultado en 10 bits válidos de ADRESH:L y borrar el flag ADIF
- 7) Para una nueva conversión se debe regresar al paso uno ó al paso dos. El tiempo de conversión por bit está definido por TAD. Se exige esperar un mínimo de 2 TAD para reiniciar una nueva conversión. (Tabla 1.7).

TAD	ADCS1:ADCS0	20 MHZ	5 MHZ	1.25 MHZ	333,33 KHZ
2 . Tosc	00	100 ns	400 ns	1,6 us	6 us
8 . Tosc	01	400 ns	1,6 ns	6,4 us	24 us
32 . Tosc	10	1,6 us	6,4 us	2,6 us	96 us
RC	11	2 – 6 us	2 - 6 us	2 - 6 us	2 – 6 us

Tabla1.7 frecuencia de CAD

Si se elige como reloj para la conversión al oscilador RC interno del C A/D, éste puede seguir funcionando cuando se introduce el microcontrolador al Modo de reposo o SLEEP.

El tiempo que determinamos para realizar una nueva conversión es de 1 segundo aproximadamente, esto se logra con lazos de repetición y utilizando el TMR0 temporizador.

1.5.5 PUERTAS Y REGISTROS ESPECIALES

A continuación se presenta la configuración de los registros que manejan el temporizador TMR0, configuración de los registros que controlan el conversor analógico / digital, y, de los registros que manejan interrupciones.

1.5.5.1 Registro de estado (STATUS)

IRP	RP1	RP0	TO#	PD#	Z	DC	C
-----	-----	-----	-----	-----	---	----	---

Tabla 1.8 Registro status

IRP: Selección del Banco RAM en direccionamiento indirecto

0	(Banco 0 y 1)	00h - FFh
1	(Banco 2 y 3)	100h- 1FFh

Z: Se pone a 1 cuando el resultado es cero.

DC: Acarreo / llevada en el cuarto bit. Funciona igual que el señalizador C, pero para el cuarto bit. Es muy útil en las operaciones con números expresados en BCD.

C: Acarreo / llevado del octavo bit: Se pone a 1 automáticamente cuando existe un acarreo en el bit de más peso en las instrucciones de suma. También actúa como señalizador de llevada en las instrucciones de resta

PD#: Se activa a 0 al ejecutarse la instrucción SLEEP (modo de reposo). Se pone a 1 automáticamente tras la conexión de la alimentación, o bien, al ejecutarse la instrucción clrwtd (refresco del perro guardián).

TO#: se activa a nivel bajo al desbordarse el perro guardián. Toma el valor de 1 tras la conexión de la alimentación o al ejecutarse las instrucciones clrwtd o SLEEP.

Los 3 bits destinados al direccionamiento de la RAM (IRP, RP1; RP0) toma el valor 0 después de la RESET, si se deseara acceder al Banco 3 de la RAM habría que emplear las siguientes instrucciones:

```
bsf STATUS, RP0
```

```
bsf STATUS, RP1
```

Son importantes en el proceso de RESET porque indican la causa que la ha originado y permiten actuar en consecuencia. No son escribibles.

Para seleccionar el banco al que se desea acceder en la RAM, se emplean los bits 6 y 5 del registro de estado, llamados RP1 y RP0 respectivamente, según el código de la tabla 1.9.

BANCO	RP1	RP0
0	0	0
1	0	1
2	1	0
3	1	1

Tabla 1.9 Estados de bits 6 y 7 para selección de banco

1.5.5.2 Registro de control de interrupciones (INT CON)

La mayor parte de los señalizadores y bits de permiso de las fuentes de interrupción en los PIC 16X8X están implementados sobre los bits del registro INTCON (Tabla 1.10), que ocupa la dirección 0Bh del Banco 0, hallándose duplicado en el Banco 1.

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
-----	------	------	------	------	------	------	------

Tabla 1.10 Registro INTCON

GIE: bit de permiso global de interrupciones

0 Prohibido (no acepta ninguna de las interrupciones)

1 Permitido (acepta interrupciones cuyo bit se lo consienta)

PEIE: Permiso de Interrupción por fin de la escritura en la EEPROM

- 0 Prohibido
- 1 Permitido

TOIE: Permiso de interrupción por sobrepasamiento del TMR0

- 0 Prohibido
- 1 Permitido

INTE: Permiso de Interrupción por activación de la terminal RB0/INT

- 0 Prohibido
- 1 Permitido

RBIE: Permiso de Interrupción por cambio de estado en RB7:RB4

- 0 prohibido
- 1 Permitido

TOIF: Señalizador de desbordamiento del TMR0

- 0 No se desbordado
- 1 Se ha desbordado

RBIF: Señalizador de cambio de estado en las terminales RB7:RB4

- 0 No ha cambiado el estado
- 1 Cambia de estado

INTF: Señalizador de activación de la terminal RB0/INT

- 0 No recibe un flanco activo
- 1 Recibe un flanco activo

1.5.5.3 Registro de opciones

El registro OPTION-REG (Tabla 1.11) toma el valor de 11111111 en cualquier tipo de reinicialización que se produzca es utilizado para la configuración del TMR0.

RBPU#	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
-------	--------	------	------	-----	-----	-----	-----

Tabla 1.11 Registro OPTION

Valor con el que actúa el divisor de frecuencia:

PS2	PS1	PS0	División del TMR0
1	1	1	1:256

Asignación del divisor de frecuencia

PSA = 0 El divisor de frecuencia se le asigna al TMR0

Tipo de flanco en tocki

TOSE = 0 Incremento del TMR0 con el flanco ascendente

Flanco activo de la interrupción externa

INTEDG = 0 Flanco descendente

Tipo de reloj para el timer0

TOCS = 0 Pulsos de reloj interno Fasc / 4(temporizador)

Resistencia de pull-up de la puerta B

RBPU# = 0 activadas

1.5.5.4 Registro de permiso de interrupciones 1(PIE1)

Contiene los bits que permiten (1) ó prohíben (0), las interrupciones provocadas por los periféricos internos del microcontrolador y que no estaban contemplados en INTCON. Para que cumplan su función es necesario que PEIE = 1.(Tabla 1.12).

PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
-------	------	------	------	-------	--------	--------	--------

Tabla 1.12 Registro PIE1

- ADIE permiso de interrupción para el conversor A/D al finalizar la conversión.
- PSPIE Permiso de interrupción para la puerta paralela esclava al realizar una operación de L/E.
- TMR1IE Permiso de interrupción para el TMR1 con su desbordamiento.
- CCP1IE Permiso de interrupción para el módulo CCP1 cuando se produce una captura o comparación.

Para contener la dirección y el dato de la memoria FLASH se necesitan 2 registros para cada valor:

EADRH(10Fh) y EEADR(10Dh) dirección de 13 bits (8K)

EEDATAH(10Eh) y EEDATA(10Ch) dirección de 14 bits (flash)

Para controlar la operación de L/E de la memoria EEPROM y FLASH hay 2 registros llamados: EECON1 y EECON2 .

1.5.5.5 Registro EECON1

EEPGD	-	-	-	WRERR	WREN	WR	RD
-------	---	---	---	-------	------	----	----

Tabla 1.13 Registro EECON1

EEPGD: Selecciona el acceso a la Flash(1) ó a la EEPROM(0)

WRERR: Señalizador de error en escritura

WREN: Permiso de escritura

WR: Hay que ponerlo a 1 para iniciar la escritura y pasa a 0 automáticamente cuando finaliza.

RD: Hay que ponerlo a 1 para iniciar la lectura.

1.5.6 CARACTERÍSTICAS GENERALES DE LOS TIMERS

1.5.6.1 Timer 0 [1]

El registro OPTION, encargado del control del TMR0. El TMR0 es un contador / temporizador de 8 bits.

- Leíble y escribible.
- Reloj interno o externo.
- Selección del flanco en el reloj externo.
- Predivisor de la frecuencia de reloj programable.
- Generación de interrupción opcional en el desbordamiento.

Una interrupción provoca que el procesador cancele temporalmente el trabajo en curso, también llamado programa principal, y pase a ejecutar otra tarea o programa llamado programa de tratamiento de la interrupción.

Cuando se produce una interrupción, el sistema guarda, de forma automática, la dirección actual del PC sobre un área de registros especiales llamados PILA.

Dicha PILA consta de 8 niveles o registros que se gestionan automáticamente y no son accesibles por los programas del usuario.

El TMR0 es uno de los periféricos que pueden provocar, si se desea interrupción cada vez que finalice la temporización. Para ello basta activar el bit GIE del registro INTCON, que otorga un permiso general de interrupciones, y el bit TOIE que habilita la interrupción del TMR0 en particular.

Cuando se provoca y acepta una interrupción el bit GIE se pone "0" automáticamente, volviendo a "1" cuando se ejecuta la instrucción RETFIE al final de tratamiento, de esta forma se garantiza que durante un tratamiento de interrupción no pueda provocarse ninguna otra.

1.5.6.2 Timer 1 [2]

Es un contador / temporizador de 16 bits, debe trabajar con entrada de reloj externo sincronizada, en Modo temporizador o Contador síncrono, nunca en modo asíncrono.

- Leíble y escribible.
- Selección del reloj interno o externo.
- Interrupción opcional por desbordamiento de FFFFh a 0000h.
- Posible reinicialización desde los módulos CCP

El Timer1 es el único Temporizador / contador ascendente con un tamaño de 16 bits, lo que requiere el uso de 2 registros concatenados de 8 bits: TMR1H, TMR1L, que son los encargados de guardar el valor del contaje en cada momento. Los impulsos de reloj que originan el contaje ascendente pueden provenir del exterior o de la frecuencia de funcionamiento del microcontrolador ($F_{osc}/4$).

Funciona de 3 formas:

- Como Temporizador
- Como Contador Síncronico
- Como contador Asíncronico

El funcionamiento del TIMER1 está gobernado por el valor con el que se programan los bits de registro T1CON, que ocupa la dirección 10h de la memoria RAM. El TMR1 puede generar una petición de interrupción cuando se produce el sobrepasamiento de contaje, es decir cuando se pasa desde FFFFh a 0000h. En esta situación, se pone automáticamente a 1 el flag TMR1F, que es el bit 0 del Registro Específico PIR1, que esta ubicado en la dirección 0Ch de la RAM. El permiso o prohibición de la Producción de Interrupción del TMR1, esta controlada por el bit TMR1IE , que ocupa la posición de menos peso del Registro Específico PIE1 (8Ch).

1.5.6.2.1 Registro T1CON

El predivisor de frecuencia (preescaler) es un simple divisor de la frecuencia de los impulsos que se aplican al TMR1 por 1,2,4 u 8.(Tabla 1.14).

		T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC#	TMR1CS	TMR1ON
--	--	---------	---------	---------	---------	--------	--------

Tabla 1.14 Registro T1CON

1.5.6.3 Timer 2 [2]:

Es un temporizador de 8 bits.

- Dispone de un registro de período de 8 bits (PR2).
- Leíble y escribible.
- Predivisor de frecuencia programable.
- Postdivisor de frecuencia programable.
- Interrupción opcional al coincidir TMR2 y PR2.
- Posibilidad de generar impulsos al módulo SSP.

1.6 CONTROL DE SENTIDO DE GIRO PARA MOTORES-DC_[13]

El uso de transistores conectados en modo corte y saturación actúan como interruptores, un análisis más completo de esta forma de conexión para transistores es la siguiente:

En este caso será necesario el uso de dos transistores complementarios, es decir uno PNP y otro NPN, de este modo sólo es necesario un terminal de control, el cual puede tomar valores lógicos "0" y "1", (Figura 1.11).

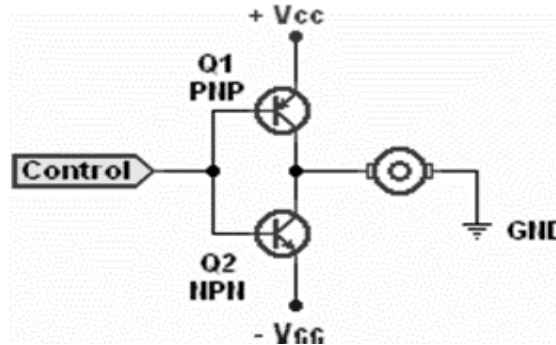


Figura 1.11 Configuración de transistores para control

Cuando utilizamos una fuente de alimentación simple, se implementa el circuito de la Figura 1.12.

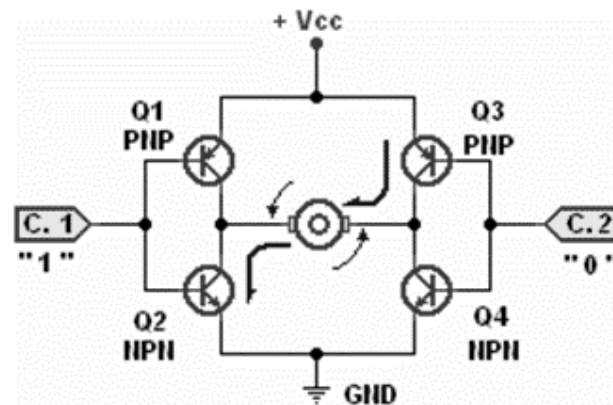
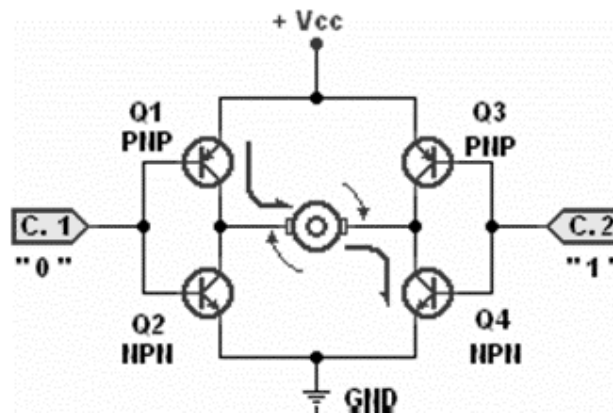


Figura 1.12 Puente H

Estos circuitos, son conocidos como puente en H o H-Bridge,

CAPITULO 2

2. DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DEL CONTROL

2.1 DESCRIPCIÓN GENERAL DEL CONTROL DE VELOCIDAD

El sistema de control de velocidad basa su funcionamiento en la utilización del microcontrolador PIC 16F877 el mismo que es el encargado de recibir las señales de entrada procesarlas y enviar señales de salida que son utilizadas para realizar el control del circuito de potencia. (Figura 2.1)

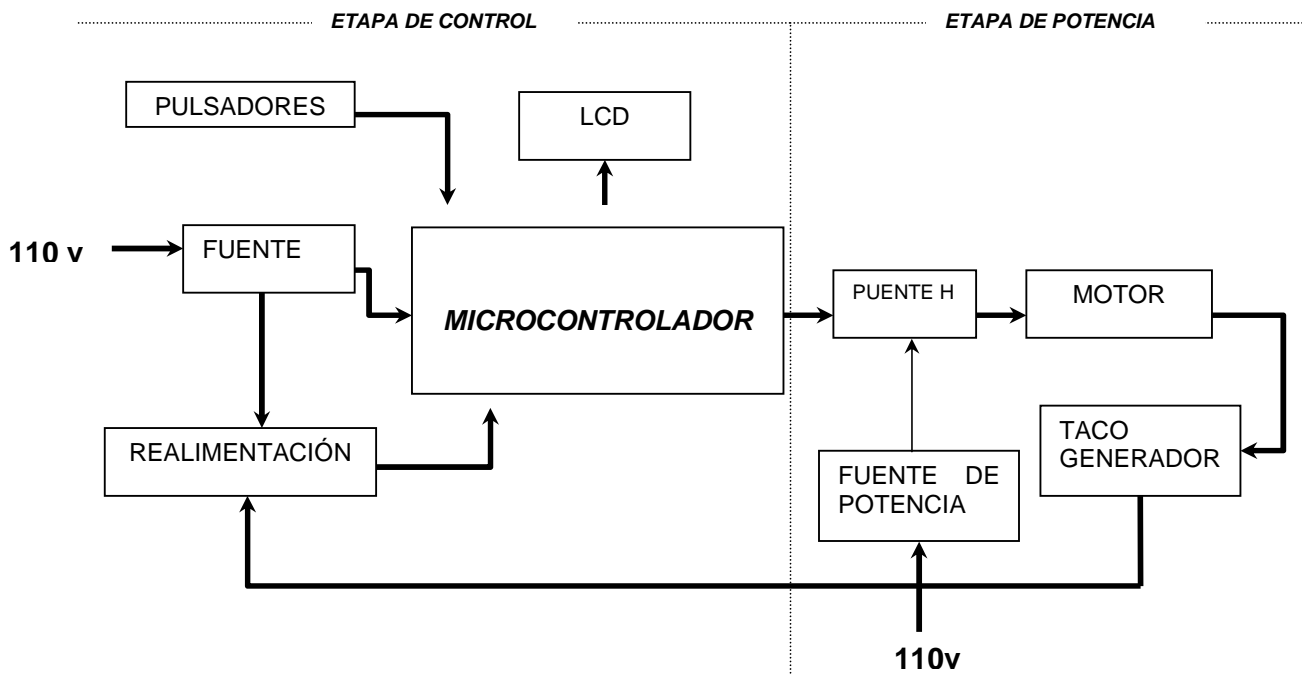


Figura 2.1 Descripción del sistema de control de velocidad

A través de pulsadores se envía al microprocesador el rango de velocidad al que debe mantenerse el motor y el sentido de giro, debemos destacar que una de las características del control de velocidad es mantener la velocidad estable independientemente de que la carga varíe o por influencia de factores aleatorios (variaciones de voltaje de la red pública, ruido, etc.) este valor de velocidad en revoluciones por minuto será visualizado en un LCD.

El conjunto pulsador LCD es manejado mediante un programa implementado en el microcontrolador.

Por otro lado la velocidad es sensada por un taco generador el cual genera un voltaje alterno proporcional a la velocidad con la que gira el motor esta señal no puede ingresar directamente al microcontrolador porque este acepta solamente voltajes positivos en sus entradas y de una amplitud determinada por lo que previamente acondicionamos la señal utilizando un circuito denominado amplificador operacional de precisión que entrega una señal positiva cuya amplitud varía entre cero y cinco voltios señal que puede ingresar al microcontrolador para su procesamiento.

Esta señal con las características antes señaladas ingresa al microcontrolador y en el módulo PWM es convertida en un tren de pulsos cuya duración de los estados altos y bajos nos permite enviar señales de salida que van a ser aplicados al circuito de potencia que basa su funcionamiento en conectar o desconectar el motor a la fuente de alimentación, a través de una banda se transfiere el movimiento al inducido del taco generador para que en este se genere un voltaje proporcional a la velocidad del motor y este realimente al PIC, para así continuar con un ciclo cerrado de este proceso.

Para la realización del diseño del circuito del proyecto lo hemos dividido en:

- Etapa de alimentación DC
- Etapa de ingreso de datos
- Etapa de visualización de datos en LCD
- Etapa de sensado de velocidad
- Etapa de programación
- Etapa de acoplamiento entre circuito de control y de potencia

2.1.1 ETAPA DE ALIMENTACIÓN

La etapa de alimentación esta dividida en dos partes:

- Fuente para el circuito de control
- Fuente para la parte de potencia

2.1.1.1 Fuente para el circuito de control

En este circuito el voltaje de la red publica es reducido en su amplitud por medio de un transformador 120v/12v con Tab central, a la salida del transformador tenemos un voltaje alterno el mismo es rectificado por un puente rectificador de onda completa implementado por dos diodos 1N4007 y filtrada por dos condensadores electrolíticos de 1000 μ f y 35v. Para obtener los valores deseados de voltaje utilizamos reguladores de voltaje AN7812, L7912CV, LM7805CV, que nos proporcionan 12v, -12v, 5v.

El voltaje de 5v es utilizado para la polarización del PIC 16F877, el voltaje de \pm 12v es utilizado en la etapa de realimentación. (Figura 2.2)

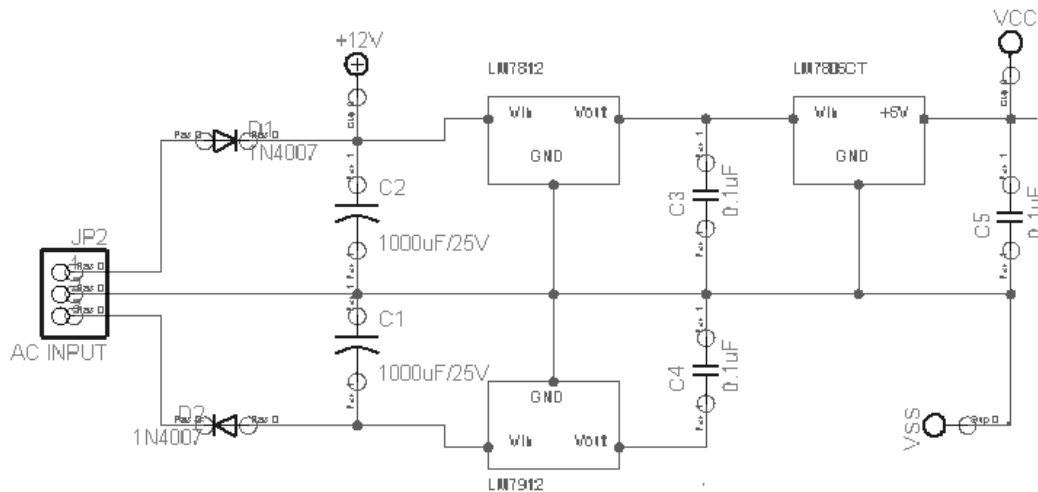


Figura 2.2 Diagrama de la fuente del circuito de control

2.1.1.2 Fuente para la parte de potencia

La fuente de potencia de igual manera el voltaje de la red publica es reducido en su amplitud por medio de un transformador 120v/12v con Tab central, a la salida

del transformador tenemos un voltaje alterno el mismo es rectificado por un puente rectificador de onda completa ECG 5330 filtrada por un condensador electrolítico 2200µf 50v. A la salida obtendremos 28v DC además incorpora un fusible de protección de 2A. (Figura 2.3)

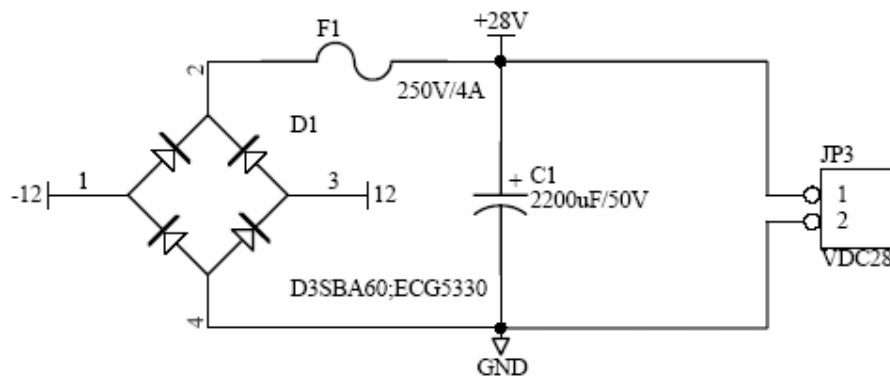


Figura 2.3 Diagrama de la fuente del circuito de Potencia

2.1.2 ETAPA DE INGRESO DE DATOS

Una de las ventajas que presenta el control de velocidad es que puede trabajar con tres diferentes valores de velocidad, además se puede invertir el sentido de giro, así como encender y apagar el circuito según sea la necesidad.

Estos tres valores de velocidad, el encendido / apagado y la inversión del sentido de giro son seleccionados por el usuario a través de cinco pulsadores que están conectados a las líneas del puerto B (RB0-RB4) del microcontrolador PIC 16F877 que es el receptor de la información, el mismo que la almacena y luego la procesa.

2.1.3 ETAPA DE VISUALIZACIÓN DE DATOS EN LCD

Una vez introducidos los datos a través de los pulsadores, utilizamos un módulo LCD para visualizar la velocidad requerida y también la velocidad a la que se encuentra girando el motor.

Este LCD tiene dos líneas de 16 caracteres cada una, es decir, en total 32 caracteres. El LCD solo puede visualizar 16 caracteres por línea pero puede almacenar un total de 40 caracteres.

2.1.4 ETAPA DE SENSADO

Para poder tener un control permanente de la velocidad a la que gira el motor necesitamos realimentar el programa con una señal proveniente de la velocidad real a la que gira el motor en ese instante; es decir sensor esa velocidad, esto lo logramos utilizando el efecto generador cuya fuerza mecánica necesaria para generar una tensión la obtenemos a través de la transmisión de movimiento a través de una banda del inducido del motor al inducido del generador aprovechando que el voltaje de salida de un generador depende de la velocidad con que se corten las líneas de fuerza del campo magnético del estator con los conductores del inducido.

2.1.4.1 Amplificador Operacional de precisión

La señal que obtenemos del taco generador es alterna y de amplitudes variables por lo que no es recomendable introducir esta señal directamente en la entrada analógica del conversor análogo digital del PIC que requiere determinada amplitud y una polaridad positiva, esta señal previamente debe ser acondicionada.

Esto lo hacemos utilizando el circuito de la (Figura 2.4) que es un amplificador operacional de precisión basado en la utilización del CI LM 358 el mismo que toma la señal del taco generador que previamente ha sido disminuida en su amplitud en un divisor de tensión implementado con el potenciómetro R0 y permitiendo que el ciclo positivo pase en forma original, mientras que el ciclo negativo se invierta.

Además le proporciona una ganancia determinada por las resistencias y capacitores que garanticen que a su salida obtengamos un valor de voltaje

positivo y de amplitud comprendida entre 0 y 5 v apropiada para ser introducida en el convertidor análogo digital del PIC. (Figura 2.4).

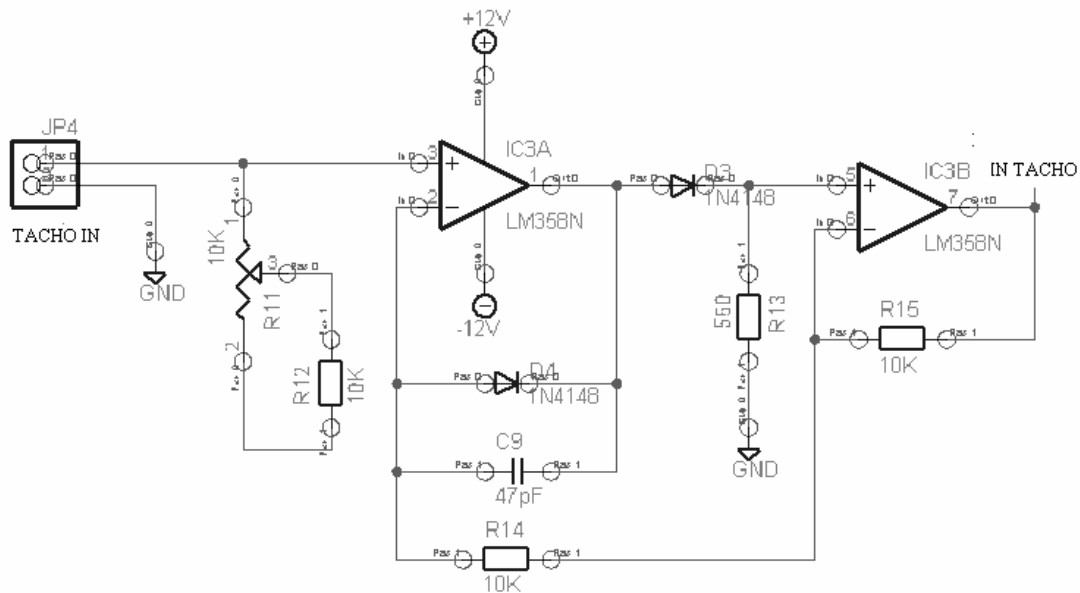


Figura 2.4 Diagrama del Amplificador Operacional de Precisión

2.1.5 ETAPA DE PROGRAMACIÓN

El funcionamiento óptimo del control de velocidad depende fundamentalmente del programa implementado en el PIC 16F877 ya que este programa es el eje de todas las etapas que conforman el control de velocidad. Esta etapa será descrita en la sección 2.2.

2.1.6 ETAPA DE ACOPLAMIENTO

La parte de potencia del control de velocidad que es la que va a manejar directamente la velocidad del motor y el sentido de giro, esta implementada utilizando un puente H, este puente H esta diseñado utilizando transistores NPN 3904, transistores de potencia TIP 127 PNP y TIP 110 NPN. Las señales de

control provienen de las salidas izquierda y derecha del PIC que determinan que en los puntos IZQ y DER tengamos 1L o 0L, si la señal de control determina que el giro debe ser a la izquierda tendremos 5 v en el punto IZQ y 0v en el punto DER, esto provoca que el transistor T1 entre en saturación y sature a su vez al transistor T6 permitiendo que el terminal positivo de la fuente de 28 voltios quede aplicado a una de las terminales del motor de igual forma el transistor T4 se satura y a la vez satura al transistor Q3 ocasionando que GND tierra se coloque en el otro terminal del motor haciendo que el motor gire en sentido antihorario, en estas condiciones, los transistores T2, T3, Q2, y T7 se encuentran en corte. Se debe señalar que en el punto IZQ los 5 voltios no llegan de manera constante sino que es un tren de pulsos provenientes del modulador de ancho de pulso cuya incidencia en el circuito de potencia es que al motor no lleguen los 28 v de manera permanente sino también un tren de pulsos de amplitud 28 cuya parte en alto depende de la velocidad escogida a través de los pulsadores.

Cuando escogemos que el motor gire en el sentido contrario el tren de pulsos se establece en el punto DER y el punto IZQ permanece en 0. En estas condiciones los transistores del puente H invierten sus condiciones: T1, T6, T4, y Q3 estén en corte mientras T2, T3, Q2, y T7 estén en saturación, provocando que el tren de pulsos de 28 voltios alimenten al motor en forma invertida con lo que el motor girará en el otro sentido. Es importante señalar que cuando el control proveniente del PIC esta a apagado los puntos IZQ Y DER se ponen en cero permanentemente lo que provoca que todos los transistores del puente H estén en corte por lo que el motor no recibe alimentación y no se mueve.

El programa implementado en el PIC genera estas señales garantizando también que no suceda el caso que en los puntos IZQ y DER sean alimentados con 5 voltios que pondrían a todos los transistores en saturación al mismo tiempo provocando cortocircuito en la fuente de alimentación. (Figura 2.5).

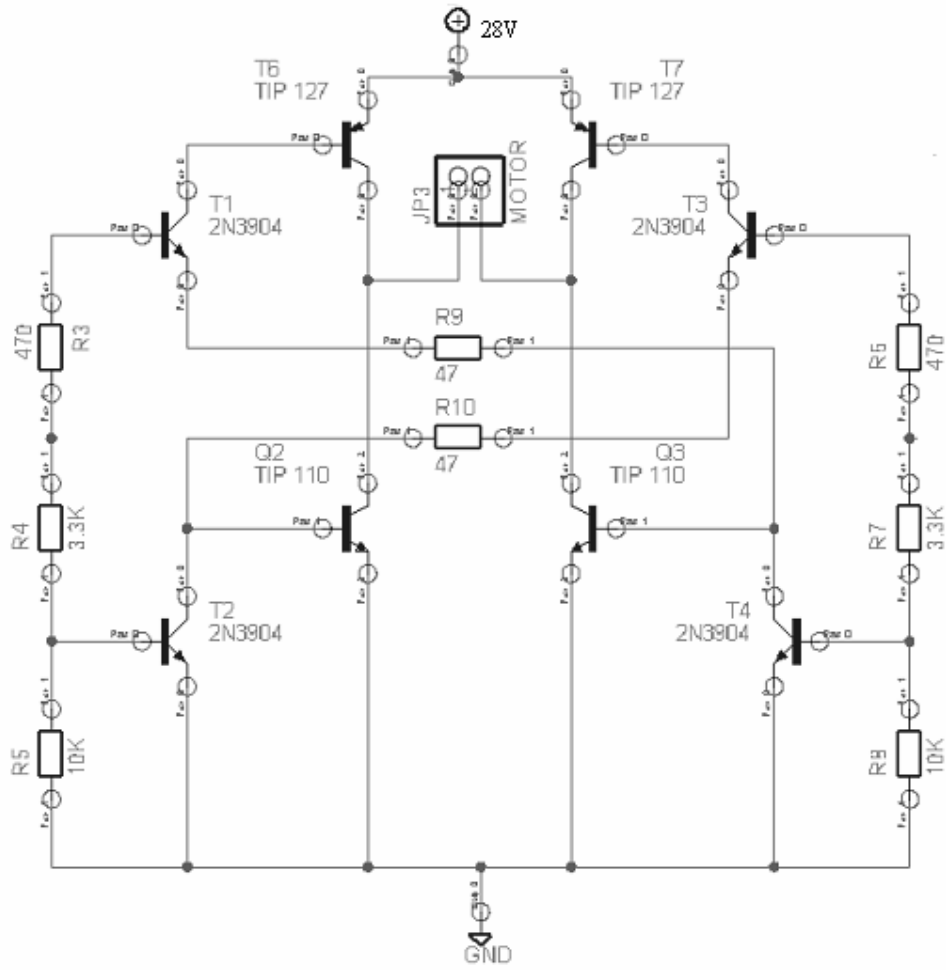


Figura 2.5 Diagrama del circuito de acoplamiento o puente H

2.2 PROGRAMACIÓN DEL PIC 16F877

El funcionamiento óptimo del control de velocidad depende fundamentalmente del programa implementado en el PIC 16F877 ya que este programa es el eje de todas las etapas que conforman el control de velocidad.

El programa implementado en el control de velocidad sigue la secuencia mostrada en el diagrama de flujo (Figura 2.6).

Se han determinado las siguientes fases que sigue el programa

- Definición de Constantes
- Definición de funciones
- Definición de variables
- Configuración de puertos y otros registros
- Conversión, comparación y generación de señales de control

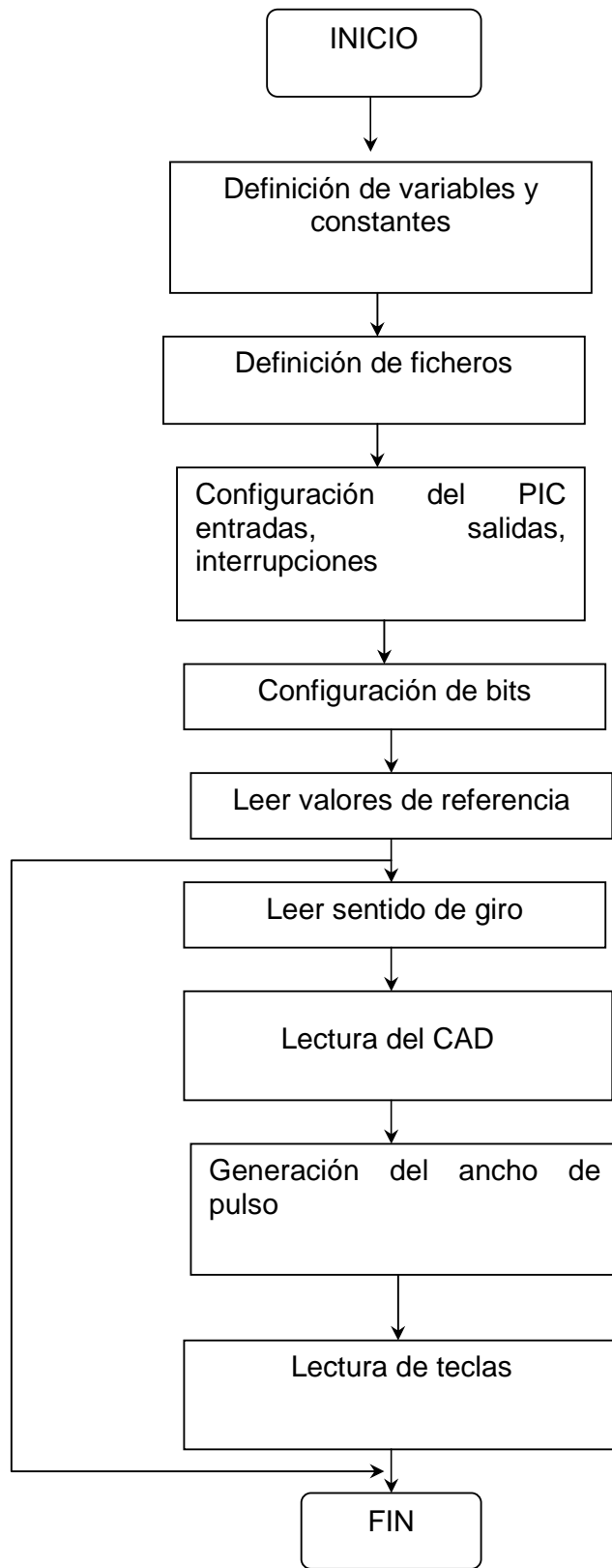


Figura 2.6 Diagrama de flujo del programa implementado

2.2.1 DEFINICIÓN DE CONSTANTES

Es la parte inicial del programa, con esta definición almacenamos datos en determinadas localizaciones de la memoria RAM bajo un nombre único. La característica principal de estas constantes es que el valor de estas no pueden variar durante la ejecución del programa.

Ejemplo: `#define SPEED1 100 // velocidad 1`

El `#define` obliga al compilador a hacer una sustitución en el programa cada vez que encuentre la cadena `SPEED1` por el valor de 100.

2.2.2 INCLUSIÓN DE FICHEROS

Para ello utilizamos la directiva `#include`, que nos permite utilizar funciones que no estén incluidas en el propio programa. De igual manera nos permite añadir librerías que se encuentren en otros ficheros a nuestro programa.

Ejemplo: `#include "lcd.h"`

Nos permite utilizar este fichero cuando estemos programando la parte relacionada al funcionamiento del LCD.

2.2.3 CONFIGURACIÓN DE BITS

En este paso asignamos un nombre específico a determinado bit de los puertos del PIC, el mismo que permanecerá constante en todo el programa.

Ejemplo: `static bit V1 @ BITNUM(PORT B, 0); //--velocidad 1`

Nos indica que el bit de menor valor del Puerto B tendrá como nombre específico V1.

2.2.4 DEFINICIÓN DE FUNCIONES

En esta etapa definimos las funciones principales que en conjunto forman el programa, están ordenadas y pueden tomarse como el reflejo del diagrama de flujo que va a seguir el programa específico del control de velocidad.

Ejemplo: `Void dutty _cal(int K6);`

Esta función contiene instrucciones que permiten calcular el pulso en alto del PWM.

2.2.5 DEFINICIÓN DE VARIABLES

En esta parte del programa identificamos a determinadas posiciones de la RAM del PIC con un nombre único, su función es la de almacenar datos en forma temporal ya que puede cambiar su valor mediante la ejecución del programa. Estas variables serán de distintas características de acuerdo a la necesidad del programa.

Ejemplo: `unsigned int x, setp, dutty;`

Con esto determinamos que en la memoria RAM vamos a tener localidades de memoria de nombres `x`, `setp`, `dutty` respectivamente que van almacenar datos enteros sin signo, es decir que su tamaño puede ser máximo de 2 bytes.

A continuación se ejecuta la función principal del programa que contiene. Las funciones específicas del programa que van a ejecutarse apoyándose en las variables, constantes, ficheros, librerías y datos definidos inicialmente y otros obtenidos en el proceso. Comienza en la parte del programa que tiene el siguiente formato.

```
Void main (void)
```

```
{
```

Dentro de esta función principal se empieza determinando los parámetros iniciales de funcionamiento del PIC, del conversor analógico digital y del LCD, de igual manera se determina las instrucciones que nos permiten eliminar los rebotes producidos al utilizar un pulsador electromecánico para el ingreso de datos.

```
Initialise ADC (CHANNEL); // Inicializa el conversor y las entradas analógicas
lcd .init ( );           // Inicializa el LCD
DelayMs(50);           // Elimina los rebotes
```

A continuación Seteamos los parámetros de velocidad y sentido de giro que por defecto tendrá el control de velocidad al momento de encenderlo.

En el proyecto $V = 100$ y giro = - antihorario y damos valores a constantes como x , K_i , k_p , k_d que van a ser utilizadas en etapas posteriores del programa.

La siguiente parte del programa empieza en la instrucción While (1) terminando en el inicio de la interrupción static void interrupt. En esta etapa del programa utilizo las sentencias WHILE IF y IF anidado de un programa en C, con esto logramos leer los datos de entrada a través de los pulsadores que determinan la velocidad y el sentido de giro del motor, de igual manera leemos señales de encendido–apagado, reset para su procesamiento y generación de los respectivas señales de control que me permitirán actuar sobre el PWM y sobre el LCD para mostrar los resultados. En esta parte del programa actúa las instrucciones para la eliminación de los rebotes provocados por los pulsadores.

La instrucción lcd-puts (" "), muestra los caracteres en el LCD.

A continuación tenemos la función static void interrupt que son las instrucciones que deben ejecutarse debido al desbordamiento de los Timer 0 y Timer 1 que cumplen su propósito específico. El Timer 0, nos permite obtener una temporización de 33 ms, cada intervalo de tiempo de 33 ms provoca una interrupción que determina que se debe realizar una nueva lectura del conversor análogo/digital. La bandera que indica el desbordamiento del Timer 0 es TOIF y cumple la siguiente función:

TOIF = 0 contiene la temporización

TOIF = 1 indica que se debe leer el conversor

La bandera de desbordamiento del Timer1 es TMR1IF que cumple la siguiente función:

TMR1IF = 0 Contiene la temporización

TMR1IF = 1 Realiza la subrutina por desbordamiento del Timer1

La subrutina del desbordamiento del Timer1 nos permite obtener el periodo del pulso del PWM y el sentido de giro del motor en este intervalo de tiempo:

$$T_{pwm} = 4/f_{osc} * 1023 * preescaler = 4 \text{ ms}$$

El preescaler tiene un valor de 8. La resolución es de 1023 valores que se obtienen cargando con los valores adecuados al Timer1.

Por otro lado cuando se realiza el desbordamiento del Timer 1 se realiza a continuación la lectura del conversor y el cálculo del error existente entre la velocidad real y la determinada por el Setp, de igual manera cargamos los parámetros adecuados para el funcionamiento posterior del PID.

La función siguiente Void lcd_show (int y1 , char y2)

Esta función permite visualizar en el LCD los valores de la velocidad seteada y la velocidad real a la que gira el motor, de igual forma para poder indicar el sentido de giro. Se basa en el criterio de divisiones sucesivas para realizar el ordenamiento de los valores a presentar, determinando el valor y en que posición.

La función void resear _ pid (void), reinicia los parámetros a utilizar en el PID, se reinicia en cero.

La función `duty_cal` (int k6), permite calcular el pulso en alto del PWM tomando en cuenta lo realizado por el PID.

La función `void init_int` (char k7), se encarga de las habilitaciones o prohibiciones de las interrupciones ocasionadas por los Timer 0 y Timer1, de igual manera de los señalizadores del desbordamiento de los mismos.

TOIF y TMR1IF son señalizadores de desbordamiento de Timer 0 y Timer1 respectivamente.

TOIE y TMR1IE bit de habilitación / prohibición de interrupciones debido a los desbordamientos de Timer 0 y Timer1 respectivamente.

GIE: permiso global de interrupciones.

La función `Void init_uc` (void), configura los puertos como entradas o salidas y determina los parámetros de funcionamiento de los Timer 0 y Timer1.

OPTION: Controla el funcionamiento del Timer 0

T1CON controla el funcionamiento del Timer 1

Los registros TRISB, TRISC, TRISD gobiernan los puertos PA, PB, PC y PD respectivamente.

En la última parte del programa se tiene la función

`INT pid` (char k1, char k2, char k3, int e int e_1)

En esta parte del programa calculo el error total como suma de errores individuales de diferentes tipos, los mismos que se colocarán dividiendo en bloques.

Para esto definimos nuevamente variables del tipo long y del tipo short.

Long: Permite una longitud de hasta 4 bytes .

Short: Permite una longitud de hasta 2 bytes o números entre 0 – 65535 ó
–32767 - 32767.

Las variables long y short son muy necesarias pues en el calculo del PID se hacen operaciones que dan resultados de números muy largos que no permiten utilizar variables de otro tipo.

El resultado del PID se introduce luego en el calculo del ancho del pulso, que sucede cada 33 ms determinado por el Timer 0.

La operación principal que se realiza es la siguiente:

$$\text{Control_output_Temp.} = (\text{control_output_temp} * 1023) / 32767.$$

CAPITULO 3

3. ENSAMBLAJE Y CALIBRACIÓN DEL SISTEMA DEL CONTROL

3.1 DIAGRAMAS DEL SISTEMA DE CONTROL

En las figuras 3.1, 3.2, mostramos los circuitos impresos del circuito de control y de la fuente de potencia.

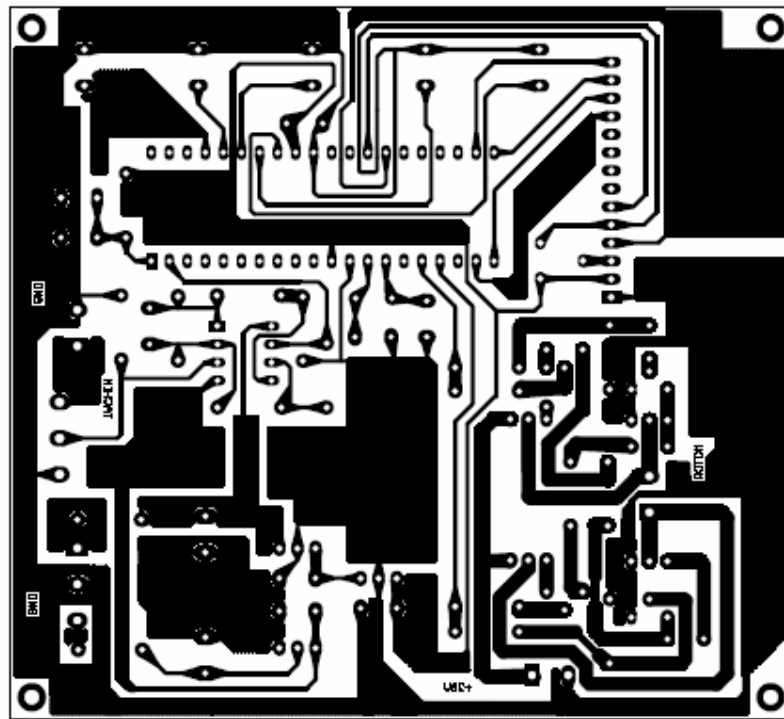


Figura 3.1 circuito impreso del circuito de control

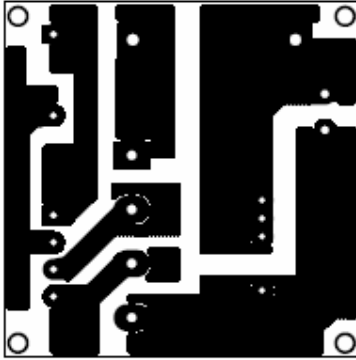


Figura 3.2 Circuito impreso de la fuente de potencia

3.2 DISPOSITIVOS ELECTRÓNICOS UTILIZADOS

DESCRIPCIÓN	CANTIDAD
transformador 110V/60Hz 12v/2A	1
transformador 110V/60Hz 12V/4A	1
pin de zócalo de 4	1
pin de zócalo de 6	1
condensadores cerámicos 0.1 uF	6
PIC 16F877	1
zócalo de 40 pines	1
4 diodos 1N4007	4
transistores 2N3904	4
TIP 110 NPN	2
TIP 127 PNP	2
zócalo de 8 pines	1
C.I HA17358 (LM 358)	1
crystal de 8 MHz	1
resistencias 10k, 1/4w	5
resistencia 47Ω, 1/4w	2
resistencias 470Ω, 1/4w	2
resistencias 3.3k ,1/4w	2
potenciómetro de 10K	2
condensadores electrolíticos de 1000uF, 35V	2
condensador de 2200uF 50V	1
punte de diodos KBJ(408)	1
Modulo LCD 2x16	1
Bus de datos (interface a 4 bits)	1
Fusible 2A – 250V	1

Tabla 3.1 Componentes electrónicos utilizados

3.3 PRUEBAS DEL SISTEMA DE CONTROL

En este sistema se realizaron pruebas y calibraciones continuas debido a la utilización de un tacogenerador como sensor, ya que la transmisión del movimiento entre el motor y el generador lo realizamos mediante una banda.

Para la programación del microcontrolador, el software utilizado nos facilito de gran manera la simulación de los programas, permitiendo observar el correcto funcionamiento de ciertas instrucciones.

En la etapa de acoplamiento correspondiente el Puente H, utilizamos los transistores en configuración Darlington, debido a que el circuito maneja corrientes altas.

Utilizamos el amplificador operacional de precisión para proporcionarle ganancia a la señal obtenida del sensor, así como voltajes positivos.

Se utiliza un potenciómetro de 10K, para mejorar el contraste de visualización en el LCD.

En el momento de encender el sistema de control, en el LCD se observan los valores seteados en el PIC los cuales son: -100 rpm en sentido horario.

Cuando no se encuentra conectado el sensor, el motor funciona a su máxima potencia, lo que repercute en un alto consumo de corriente.

CONCLUSIONES

- La utilización de microcontroladores y en particular del microcontrolador PIC 16F877 facilitan en gran medida la implementación de circuitos de control para diferentes procesos.
- La tecnología electrónica existente en nuestro país nos permitió que nuestro objetivo: realizar el sistema de control de velocidad con modulación de ancho de pulso se cumpla.
- Utilizamos un puente H con configuración Darlington lo que permite soportar corrientes altas en la etapa de acoplamiento; además esta etapa de acoplamiento garantiza una total independencia eléctrica entre el circuito de control y el de potencia.
- Con la modulación de ancho de pulso se obtiene una señal cuadrada cuyo valor medio es fácilmente variable, con esta señal se puede regular la velocidad de un motor c.c, cuanto mas ancho es el pulso mayor es la velocidad del motor y viceversa.
- El lazo de velocidad diseñado mantiene la velocidad del motor constante de acuerdo a una referencia elegida, sin importar la variación de la carga del motor
- Las librerías del programa que maneja el envío de datos y comandos a la pantalla LCD mediante rutinas nos permitieron el ahorro de líneas del programa principal y en proyectos posteriores facilitará la visualización de caracteres en la pantalla LCD.

RECOMENDACIONES

- Es importante no cambiar ningún elemento electrónico por otro, mientras no sea un elemento equivalente o de igual característica al implementado en este circuito de control.
- Al trabajar con el LCD, debemos tomar en cuenta la correcta polarización de lo contrario si trabajamos con tensiones no especificadas el LCD se puede quemar o deja de funcionar adecuadamente.
- Si el presente trabajo se va a tomar como referencia para la elaboración de un circuito de control de velocidad donde los rangos de voltaje y velocidad van a ser mas altos es necesario elegir un sensor de velocidad mas adecuado y realizar previamente el acondicionamiento de la señal para que sea compatible con los requerimientos del PIC.

BIBLIOGRAFÍA

- [1] ANASI T., César A. (2003) Prototipo automático programable del control de temperatura en una plantación de rosas de exportación. Tesis de Tecnología en Electrónica y Telecomunicaciones Escuela Politécnica Nacional

- [2] ANGULO U., José M. (2000) Microcontroladores PIC, Diseño práctico de aplicaciones Madrid: Mc Graw –Hill.-

- [3] AUCAPIÑA ESCOBAR, German. (2001) Sistemas de desarrollo para los microcontroladores PIC. Tesis de Ingeniería Eléctrica Escuela Politécnica Nacional.

- [4] CAMPAÑA, Jenny
TITO, Edgar (2003), Sistema De Alarma Con Microcontrolador PIC., Tesis de Tecnología en Electrónica y Telecomunicaciones Escuela Politécnica Nacional.

- [5] PHILIPS ECG Semiconductors, (1985) Guía de reemplazos. Canadá

- [6] VALKENBURGH, Van (1975) Electricidad Básica. Argentina,

REFERENCIAS

- [7] <http://www.todorobot.com.ar>
Modulación de ancho de pulso

- [8] <http://www.vestatech.com>
Proporcional Integral derivativa

- [9] <http://www.frino.com.ar>
Información elemental de LCD
- [10] <http://www.monografias.com>
Descripción del PIC 16F877
- [11] [http://www.rincón del vago.com](http://www.rincón-del-vago.com)
Motores DC
- [12] <http://www.microchip.com>
Especificaciones técnicas y gráficas del PIC 16F877
- [13] <http://www.automationdirect.com/static/manuals/d006usermsp/ch8.pdf>
Motores y Modulación de ancho de pulso

ANEXOS

LENGUAJE C

Su característica principal es ser portable, es decir, es posible adaptar los programas escritos para un tipo de computadora en otra.

Otra de sus características principales es el ser estructurado, es decir, el programa se divide en módulos (funciones) independientes entre sí.

El lenguaje C se conoce como un lenguaje compilado. Existen dos tipos de lenguaje: interpretados y compilados. Los interpretados son aquellos que necesitan del código fuente para funcionar (Por ejemplo: Basic). Los compilados convierten el código fuente en un fichero objeto y éste en un fichero ejecutable como en el caso del lenguaje C.

Podemos decir que el lenguaje C es un lenguaje de nivel medio, ya que combina elementos de lenguaje de alto nivel con la funcionalidad del lenguaje ensamblador. Es un lenguaje estructurado, ya que permite crear procedimientos en bloques dentro de otros procedimientos, es decir, el programa se divide en módulos (funciones) independientes entre sí.

A.1 COMPILADOR

La traducción de una serie de instrucciones en lenguaje ensamblador (el código fuente) a un código máquina (o código objeto) no es un proceso muy complicado y se realiza normalmente por un programa especial llamado compilador. La traducción de un código fuente de alto nivel a un código máquina también se realiza con un compilador.

A.2 TIPOS DE DATOS

En C' existen básicamente cuatro tipos de datos, podremos definir nuestros propios tipos de datos a partir de estos cuatro. A continuación se detalla su nombre, el tamaño que ocupa en memoria y el rango de sus posibles valores.

TIPO	Tamaño	Rango de valores
char	1 byte	-128 a 127
int	2 bytes	-32768 a 32767
float	4 bytes	3'4 E-38 a 3'4 E+38
double	8 bytes	1'7 E-308 a 1'7 E+308

A.3 VENTAJAS DEL LENGUAJE C

El lenguaje C aprovecha óptimamente las características del hardware de los microcomputadores, por este motivo los programas son mas compactos y se ejecutan con mayor rapidez.

A.4 DESCARGA DEL PROGRAMA VERSIÓN DEMO HT-PICC

Debido a que la licencia el programa HT-PICC es muy costosa, debemos descargar desde el internet la versión DEMO ya que se distribuye gratuitamente. Debemos ingresar a la página [www..microchip.com](http://www.microchip.com).

A.5 COMPILACION DEL ARCHIVO FUENTE

Una vez escrito el programa debe ser compilado, se lo realiza de la siguiente manera:

(Figura A.1)

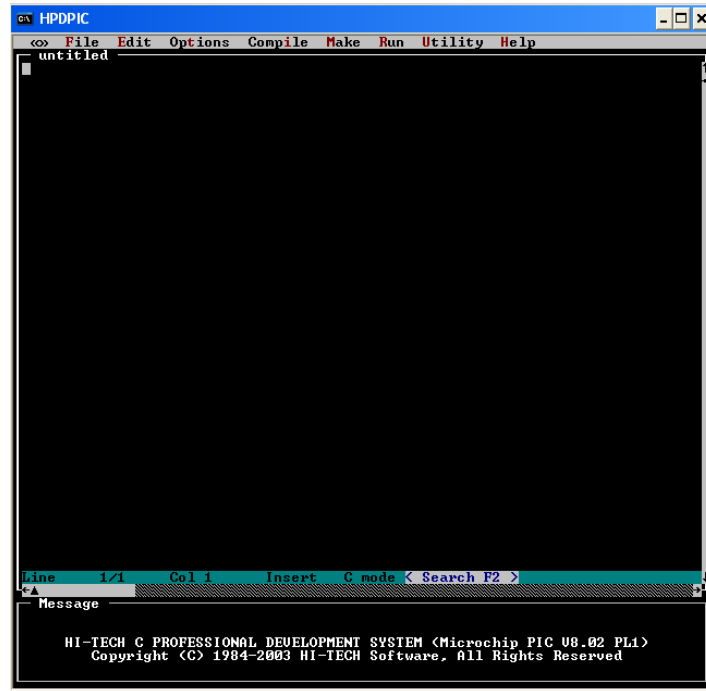


Figura A.1

Para empezar la simulación debemos elegir el modelo del PIC. (FiguraA.2).

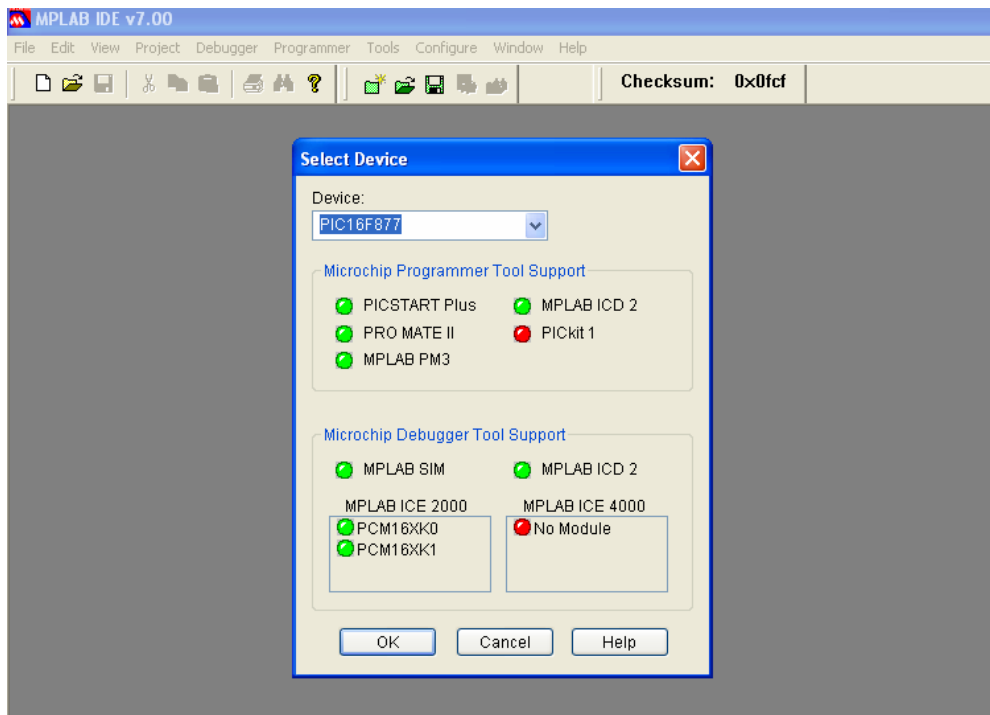


Figura A.2

Una vez que la compilación se ha completado satisfactoriamente para ponerlo a funcionar se debe seleccionar programa simulador. Una vez que se ingresa se puede observar las localidades de la memoria EEPROM . (Figura A.3).

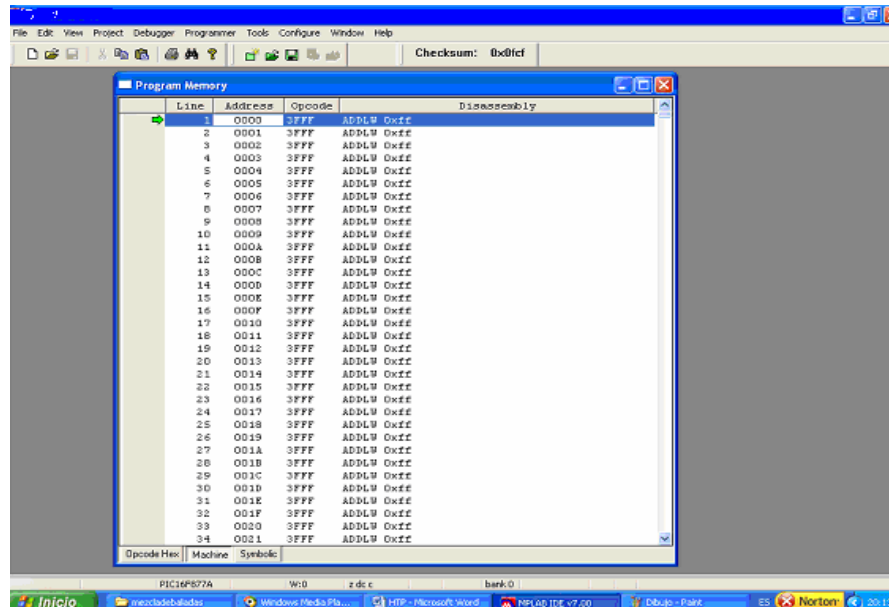


Figura A.3

También se puede seleccionar la frecuencia de trabajo del PIC. (Figura A.4).

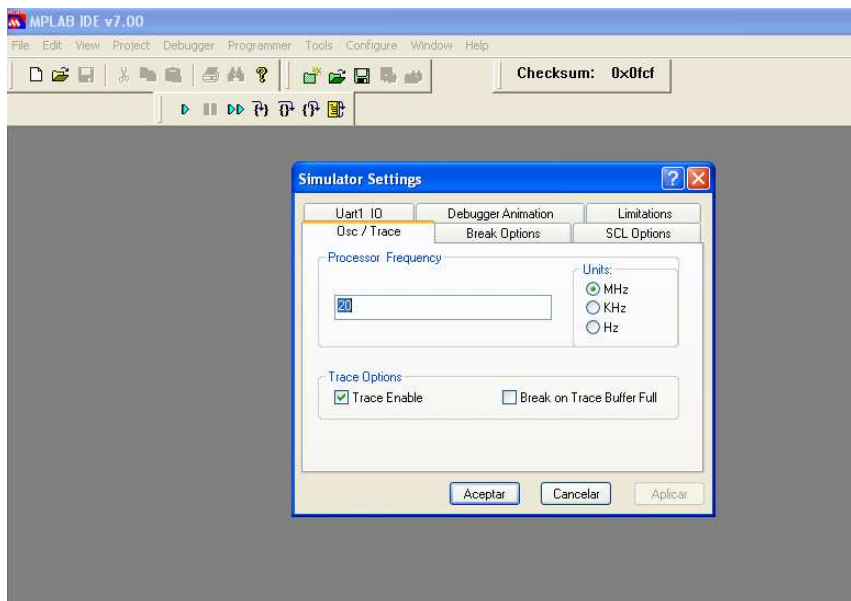


Figura A.4

Existen otras ventanas que permiten visualizar el programa y su ejecución, una de estas es la de registros de Función especial, la que permite observar el cambio en todos los registros especiales. (figura A.5,A.6)

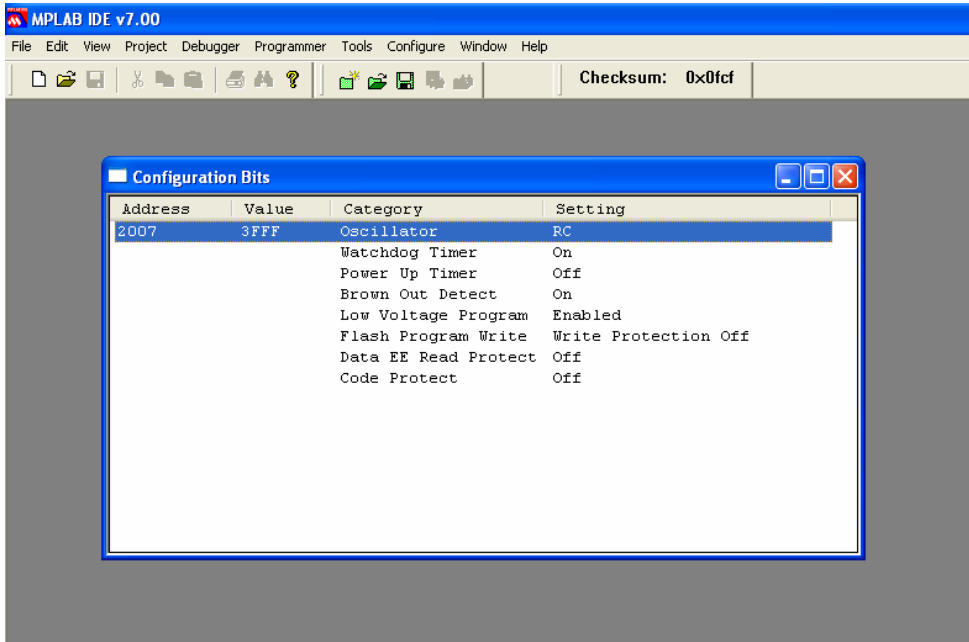


Figura A.5

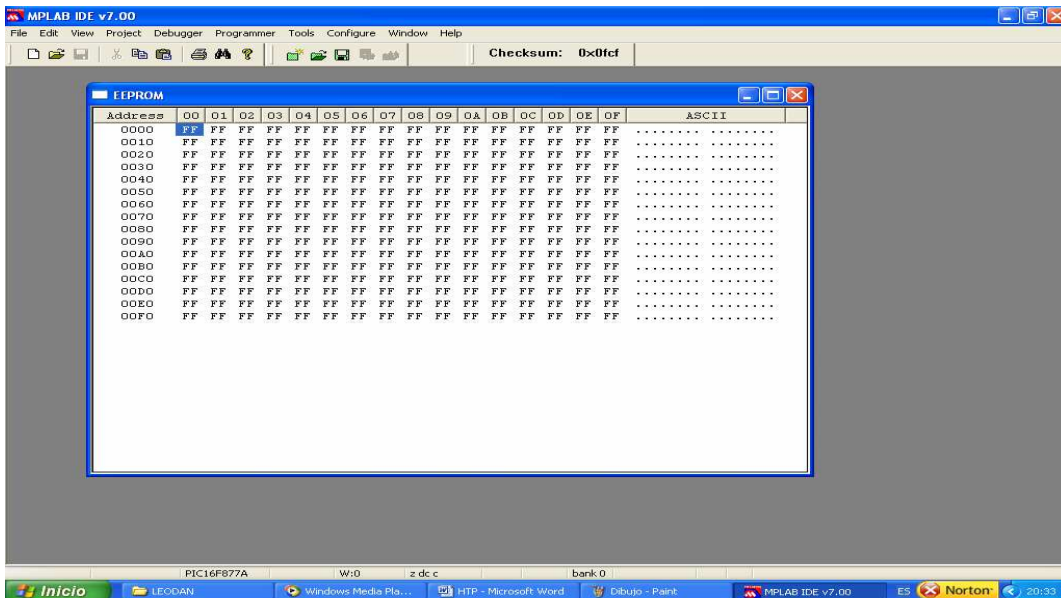


Figura A.6

El compilador HT-PICC ofrece muchas facilidades, de las cuales algunas no han sido utilizadas en el presente trabajo.


```

static bit    v2                @ BITNUM (PORTB, 1);/-- VELOCIDAD 2
static bit    v3                @ BITNUM (PORTB, 2);/-- VELOCIDAD 3
static bit    giro              @ BITNUM (PORTB, 3);/-- Inversión de giro
static bit    start_stop        @ BITNUM (PORTB, 4);/-- inicio
static bit    FW                @ BITNUM (PORTC, 1);/-- directo
static bit    RV                @ BITNUM (PORTB, 3);/-- reverse
#define       LCD_DATA          PORTD /-- LCD Data Port (Low 4 bits)
#define       T_LCD_DATA        TRISD          /-- LCD Data Port (Low 4 bits)
static bit    T_LCD_RS          @ BITNUM (TRISD, 4); /-- LCD RS Control
static bit    T_LCD_EN          @ BITNUM (TRISD, 5); /-- LCD E Control
static bit    LCD_RS            @ BITNUM (TRISD, 4); /-- LCD RS Control
static bit    LCD_EN            @ BITNUM (TRISD, 5); /-- LCD E Control
#include      "lcd.c"

```

```
// Definicion de funciones
```

```
void init_uc (void);
```

```
void init_int (char k7) ;
```

```
int pid (char k1, char k2, char k3, int e, int e_1);
```

```
void reset_pid (void);
```

```
void dutty_cal (int k6);
```

```
void lcd_show (int y1, char y2);
```

```
// Definicion de variables
```

```
int error;
```

```
int error_1;
```

```
bank1 char kp, ki, kd;
```

```
bank1 char d3, d2, d1, d0;
```

```
bank1 unsigned int count;
```

```
unsigned int out;
```

```
int PID_control_output;
```

```
int PID_accumulated_error;
```

```
short proportional_term;
```

```
short derivative_term;
```

```
short integral_term;
```

```

unsigned int x, setp, dutty;
unsigned char dutty_on_high, dutty_on_low, dutty_off_high, dutty_off_low;
bit f_dutty, f_giro, f_start_stop;
void main (void)
{
    DelayMs (50);                //temporizacion para eliminar rebotes
    Count=0;
    Init_uc ( );                //inicializa al microcontrolador
    f_dutty=1;
    InitialiseADC(CHANNEL);     // inicializa conversor adc
    lcd_init( );                // inicializa el LCD
    lcd_clear ( );
    lcd_cursor (OFF);
    lcd_puts (men_off_1);      // mensaje linea 1
    lcd_goto (0x40);           // linea 2
    lcd_puts (men_off_2);
    lcd_goto (0x46);           // linea 2
    lcd_puts (SPEED1);
    ki = 5;
    kp = 10;
    kd = 15;
    setp = SPEED1;            // velocidad por defecto
    x=0;
    f_giro=0;                  // giro rv
    f_start_stop=0;           // inicio off
    lcd_goto (0x0f);
    lcd_puts ("-");
    init_int(OFF);
    while (1)
    {
        if (f_start_stop= =1)
        {
            count++;

```



```

        if (count>=20000)
        {
            count=0;
            lcd_show (x,0x06);
        }
    }
if (v1==0)
    {
        setp=SPEED1;          // set1
        lcd_goto(0x46);
        lcd_puts (men_speed1);
        DelayMs (NOT_BOUN);
    }

if (v2 == 0)

    {
        setp=SPEED2;          // set2
        lcd_goto(0x46);
        lcd_puts (men_speed2);
        DelayMs (NOT_BOUN);
    }

if (v3 == 0)

    {
        setp=SPEED3;          // set3
        lcd_goto(0x46);
        lcd_puts (men_speed3);
        DelayMs (NOT_BOUN);          //elimina rebotes
    }

if (giro== 0)

    {

```

```

        f_giro^=1;
        lcd_goto(0x0f);
        if (f_giro== 1)
        {
            RV=0;
            lcd_puts("+");
        }
        else
        {
            FW=0;
            lcd_puts("-");
        }
        DelayMs(NOT_BOUN);
    }
if (start_stop== 0)
{
    f_start_stop^=1;
    lcd_goto(0x00);
    if (f_start_stop== 1)
    {
        lcd_puts(men_on_1);
        lcd_goto(0x40);
        lcd_puts(men_on_2);
        lcd_show(setp,0x46);
        reset_pid( );
        DelayMs(NOT_BOUN);
        Init_int(ON);
    }
    else
    {
        Init_int(OFF);
        PORTC=0;
        lcd_clear( );
        lcd_puts(men_off_1);
    }
}

```

```

        lcd_goto(0x40);
        lcd_puts(men_off_2);
        lcd_show(setp, 0x46);
        DelayMs(NOT_BOUN);
        lcd_goto(0x0f);
        if (f_giro==1)
            lcd_puts("+");
        else
            lcd_puts("-");
    }
}
}
(
static void interrupt //interrupciones debidas a los timers
isr(void)
{
    if (TMR1IF==1)
    {
        // interrupcion del timer
        TMR1IF = 0; // Limpia la bandera de la interrupcion
        if (f_dutty==1);
        {
            TMR1H = dutty_on_high; // RESOLUCION 1023 VALORES
            TMR1L = dutty_on_low;
            if (dutty==0)
            {
                TMR1IF=1;
            (
            else
            {
                if (f_giro==1)
                {
                    FW=1;
                    RV=0;

```

```

        (           else
        {
            FW=0;
            RV=1;
            (
            (
            else
            {
                if (duty= =1023)
                {
                    TMR1IF=1;
                    (
                    TMR1H = duty_off_high;           // RESOLUCION 1023 VALORES
                    TMR1L = duty_off_low;
                    FW=0;
                    RV=0;
                    (
                    f_duty ^=1;
                    goto finter;
                (
                if (TOIF= =1)
                {
                    TOIF=0;
                    x= ReadADC(CHANNEL);
                    error=setp-x;           // error
                    out=pid(kp,ki,kd,error,error_1);
                    error_1=error;
                    duty_cal(out);
                (
                finter: ;
                (void lcd_show (int y1, char y2)
                {

```

```

    int temp_val;
    temp_val = y1/1000;
    d3=temp_val;
    y1=y1%1000;
    temp_val= y1/100;
    d2=temp_val;
    y1=y1%100;
    temp_val= y1/10;
    d1=temp_val;
    y1=y1%10;
    d0=y1;
    lcd_goto(y2);
    putchar(d3+48);
    lcd_goto(y2+1);
    putchar(d2+48);
    lcd_goto(y2+2);
    putchar(d1+48);
    lcd_goto(y2+3);
    putchar(d0+48);
(
void reset_pid(void)
{
    out=0;
    error_1=0;
    error=0;
    PID_accumulated_error=0;
    out=0;
(
void dutty_cal (int k6)           //calculo del pulso en alto del PWM
{
    dutty = k6;
    dutty_on_high=(65535-k6)>>8;
    dutty_on_low=(65535-k6);
    dutty_off_high=(64512+k6)>>8;

```

```

        dutty_off_low=(64512+k6);
(
void init_int (char k7)
{
    TMR0=250;
    TMR1H=0xF0;
    TMR1L=0x00;
    TMR1IF=0;
    T0IF=0;
    TMR1IE=k7;
    T0IE=k7;
    PEIE=k7);
    TMR1ON=k7;
    GIE=k7;           // Permiso global para interrupciones
(
void init_uc (void)
{
    OPTION=0b00000110;           // pull ups enable prescaler timer 0 1:255
    T1CON=0b00000000;
    TRISB=255;
    TRISC=0;
    TRISD=0;
    PORTD=0;
    PORTC=0;
    PORTB=0;
(
int pid (char k1, char k2, char k3, int e, int e_1)
{
    short temp;
    long acc_error_temp, integral_term_temp, proportional_term_temp;
    long control_output_temp;
    if (e>=0)
        temp=e;
    else

```

```

temp= -e;
if (temp < =Deadband_Width)
{
e = 0; //Within deadband so zero error speed
(
proportional_term_temp = (long) e*(long)k1);
if (proportional_term_temp >32767)
proportional_term = 32767;
else
{
if proportional_term_temp < = -32768)
proportional_term = -32768;
else
proportional_term = (short) ( (long) proportional_term_temp);
(
derivative_term = (long) ( e - e_1)* k3);
acc_error_temp = (long) PID accumulated_error) + (long)e;
if (acc_error_temp > (long) 32767)
acc_error_temp = 32767; // limit to max + ve value
if (acc_error_temp < (long) -32768)
acc_error_temp= -32768; // limit to max - ve value

PID_accumulated_error = (short) acc_erroe_temp;
integral_term_temp = (long)PID_accumulated_error * (long)k2;
if (integral_term_temp >32767 )
integral_term = 32767;
else
{
if (integral_term_temp <= (long) ( (long) - 32768))
integral_term = -32768;
else
integral_term = (short)integral_term_temp;
(

```

```
control_output_temp = (long) integral_term;
control_output_temp += (long) derivative_term;
control_output_temp += (long) proportional_term;
control_output_temp += (long) PID_output_offset;
/*Limit Value of Control Term */
if (control_output_temp >32767)
{
    control_output_temp = 32767;
(
else
{
    if (control_output_temp < 0);
    {
        control_output_temp = 0;
(
(
/** update control variable***/
control_output_temp = (control_output_temp*1023)/32767;
PID_control_output = (int)control_output_temp;
return PID_control_output;
(
```


CONJUNTO DE INSTRUCCIONES DE RANGO MEDIO

A continuación (tabla A.1), se resumen las 35 instrucciones que reconoce la CPU de los PIC de medio rango, incluyendo su mnemónico, tiempo de ejecución, código de máquina y afectación de banderas:

Mnemónico	Descripción	Ciclos	Código de Máquina	Banderas afectadas
Operaciones con el archivo de registros orientadas a bytes				
ADDWF f,d	Suma f + W	1	00 0111 dfff ffff	C,DC,Z
ANDWF f,d	W AND f	1	00 0101 dfff ffff	Z
CLRF f	Limpia f	1	00 0001 1fff ffff	Z
CLRW	Limpia W	1	00 0001 0xxx xxxx	Z
COMF f,d	Complementa los bits de f	1	00 1001 dfff ffff	Z
DECf f,d	Decrementa f en 1	1	00 0011 dfff ffff	Z
DECFSZ f,d	Decrementa f, escapa si 0	1(2)	00 1011 dfff ffff	
INCF f,d	Incrementa f en 1	1	00 1010 dfff ffff	Z
INCFSZ f,d	Incrementa f, escapa si 0	1(2)	00 1111 dfff ffff	
IORWF f,d	W OR f	1	00 0100 dfff ffff	Z
MOVF f,d	Copia el contenido de f	1	00 1000 dfff ffff	Z
MOVWF f	Copia contenido de W en f	1	00 0000 1fff ffff	
NOP	No operación	1	00 0000 0xx0 0000	
RLF f,d	Rota f a la izquierda	1	00 1101 dfff ffff	C
RRF f,d	Rota f a la derecha	1	00 1100 dfff ffff	C
SUBWF f,d	Resta f – W	1	00 0010 dfff ffff	C,DC,Z
SWAPF f,d	Intercambia nibbles de f	1	00 1110 dfff ffff	
XORWF f,d	W EXOR f	1	00 0110 dfff ffff	Z
Operaciones con el archivo de registros orientadas a bits				
BCF f,b	Limpia bit b en f	1	01 00bb bfff ffff	
BSF f,b	Pone bit b en f	1	01 01bb bfff ffff	
BTFSC f,b	Prueba bit b en f, escapa si 0	1(2)	01 10bb bfff ffff	
BTFSS f,b	Prueba bit b en f, escapa si 1	1(2)	01 11bb bfff ffff	
Operaciones con literales y de control del programa				
ADDLW k	Suma literal k + W → W	1	11 111x kkkk kkkk	C,DC,Z
ANDLW k	k AND W → W	1	11 1001 kkkk kkkk	Z
CALL k	Llamado a subrutina	2	10 0kkk kkkk kkkk	
CLRWDT	Limpia timer del watchdog	1	00 0000 0110 0100	\overline{TO} , \overline{PD}
GOTO k	Salto a la dirección k	2	10 1kkk kkkk kkkk	
IORLW k	k OR W → W	1	11 0000 kkkk kkkk	Z
MOVLW k	Copia literal a W	1	11 00xx kkkk kkkk	
RETFIE	Retorna de interrupción	2	00 0000 0000 1001	
RETLW k	Retorna con literal k en W	2	11 01xx kkkk kkkk	
RETURN	Retorna de subrutina	2	00 0000 0000 1000	
SLEEP	Activa Modo standby	1	00 0000 0110 0011	\overline{TO} , \overline{PD}
SUBLW k	Resta k – W → W	1	11 110x kkkk kkkk	C,CD,Z
XORLW k	k EXOR W → W	1	11 1010 kkkk kkkk	Z
Notación: d= destino del resultado $\begin{cases} d = 0 & \text{destino W} \\ d = 1 & \text{destino registro} \end{cases}$ f =dirección del registro (memoria RAM), b= número de bit (0 a 7), k= dato de 8 bits				

Tabla A.1

DEFINICIONES

Señal de salida: es la variable que se desea controlar (posición, velocidad, presión, temperatura, etc.). También se denomina variable controlada.

Señal de referencia: es el valor que se desea que alcance la señal de salida.

Error: es la diferencia entre la señal de referencia y la señal de salida real.

Señal de control: es la señal que produce el controlador para modificar la variable controlada de tal forma que se disminuya, o elimine, el error.

Señal análoga: es una señal continua en el tiempo.

Señal digital: es una señal que solo toma valores de 1 y 0. El PC solo envía y/o recibe señales digitales.

Convertor análogo/digital: es un dispositivo que convierte una señal analógica en una señal digital (1 y 0).

Convertor digital/analógico: es un dispositivo que convierte una señal digital en una señal analógica (corriente o voltaje).

Sistema: consiste en un conjunto de elementos que actúan coordinadamente para realizar un objetivo determinado.

Sensor: es un dispositivo que convierte el valor de una magnitud física (presión, flujo, temperatura, etc.) en una señal eléctrica codificada ya sea en forma analógica o digital. También es llamado transductor. Los sensores, o transductores, analógicos envían, por lo regular, señales normalizadas de 0 a 5 voltios, 0 a 10 voltios o 4 a 20 mA.

Sistema de control en lazo cerrado: es aquel en el cual continuamente se está monitoreando la señal de salida para compararla con la señal de referencia y calcular la señal de error, la cual a su vez es aplicada al controlador para generar la señal de control y tratar de llevar la señal de salida al valor deseado. También es llamado control realimentado.

ESPECIFICACIONES DE ABREVIATURAS

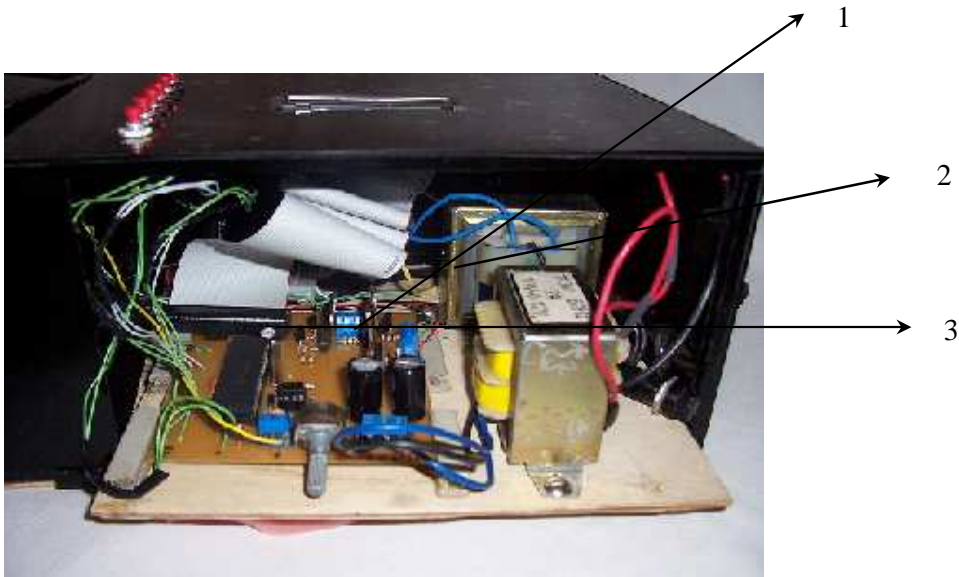
- E = Fem inducida
- FEM = fuerza electromotriz
- N = numero de espiras de la bobina
- PWM = modulación de ancho de pulso
- PID = proporcional integral derivativa
- k_p = constante proporcional
- k_d = constante derivativa
- k_i = constante integral
- e = error
- e_{-1} = error anterior
- e = set point – realimentación
- RISC = Reduced Instruction Set Computer (Conjunto de instrucciones reducidas de computación).
- CMOS = Complementary Metal Oxide Semiconductor (Semiconductor complementario de oxido metálico).
- EPROM = Erasable Programmable Read Only Memory (Memoria programable y borrrable de solo lectura).
- ROM = Read Only Memory (Memoria de solo lectura).
- PDIP = plastic dual in line package
- PLCC = plastic leaded chip carrier
- QFP = Quad Flat Package
- SOIC = Small Outline I. C.
- ALU = Unidad de Lógica y Aritmética
- LCD = Display de cristal liquido
- CAD = Conversor análogo digital

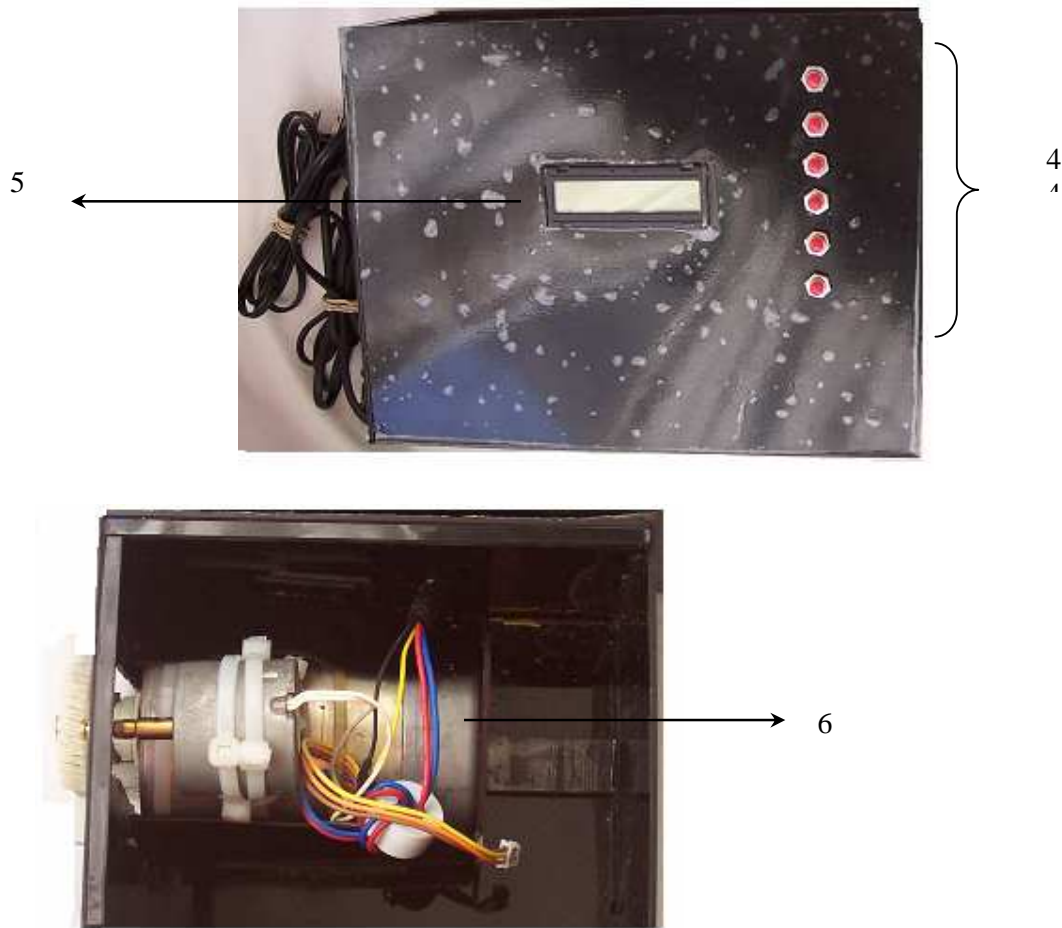
SISTEMA PARA CONTROLAR LA VELOCIDAD DE UN MOTOR DC UTILIZANDO MODULACIÓN DE ANCHO DE PULSO

INTRODUCCIÓN

Este manual contiene instrucciones sobre el uso y mantenimiento del Sistema de control de velocidad de un motor dc con modulación de ancho de pulso. Para el correcto funcionamiento de este equipo todos los usuarios deberán leer y seguir las instrucciones de este manual.

DESCRIPCIÓN DE LOS COMPONENTES





- 1) **ETAPA DE ACOPLAMIENTO:** conformada por el puente h que es el encargado de controlar el sentido de giro del motor
- 2) **FUENTE DEL CIRCUITO DE POTENCIA:** esta fuente proporciona la alimentación para el funcionamiento del motor.
- 3) **FUENTE DEL CIRCUITO DE CONTROL:** esta fuente proporciona la alimentación para el PIC y para el amplificador operacional de precisión.
- 4) **SELECTORES DE VELOCIDAD:** esta constituido por pulsadores que permiten seleccionar velocidades de 100, 300 y 500 RPM, además, permite seleccionar el sentido de giro, y también encender o apagar el circuito.

Reset

Es el valor programado en el PIC, al encender nos permite observar el valor seteado de 100 y el signo negativo (sentido horario).

Star/Stop

Nos permite encender / parada del motor.

Inversión de giro

Nos permite elegir el sentido de giro (horario / antihorario).

Velocidad 1

100 RPM: Nos permite seleccionar la velocidad programada de 100 RPM, además, podemos observar en el LCD, la velocidad real a la que gira el motor usando el tacogenerador como un sensor.

Velocidad 2

300 RPM: Nos permite seleccionar la velocidad programada de 300 RPM, además, podemos observar en el LCD, la velocidad real a la que gira el motor usando el tacogenerador como un sensor.

Velocidad 3

500 RPM: Nos permite seleccionar la velocidad programada de 500 RPM, además, podemos observar en el LCD, la velocidad real a la que gira el motor usando el tacogenerador como un sensor.

5) PANTALLA DE VISUALIZACIÓN: indicador de mensajes.

6) ETAPA DE SENSADO: esta conformada por un motor dc y generador (sensor) el cual dependiendo de la velocidad a la que gire el motor genera una tensión.

NOCIONES BÁSICAS:

Este circuito esta controlado por un **PIC** el cual va a recepatar y administrar valores de voltaje correspondientes a la salida del generador que esta siendo utilizado como sensor. Estos valores de voltaje serán comparados con valores fijos programados en el PIC 16F877 el cual ampliara o reducirá el ancho de

pulso de voltaje con el propósito de aumentar o disminuir la velocidad del motor y así mantenerla constante.

Para la utilización del sistema de control usted debe seguir las siguientes instrucciones:

- Conectar las fuentes del circuito de control y del circuito de potencia a la red de alimentación pública.
- En la pantalla de visualización se podrá observar.



- Mediante los pulsadores usted podrá seleccionar la velocidad de giro del motor y el sentido de giro. Oprima el pulsador con la opción que usted desea y podrá observar en la pantalla un mensaje que le indicara la velocidad y el sentido de giro.



- Para apagar el sistema usted solamente deberá oprimir el pulsador correspondiente al encendido / apagado.

SOLUCIÓN DE PROBLEMAS

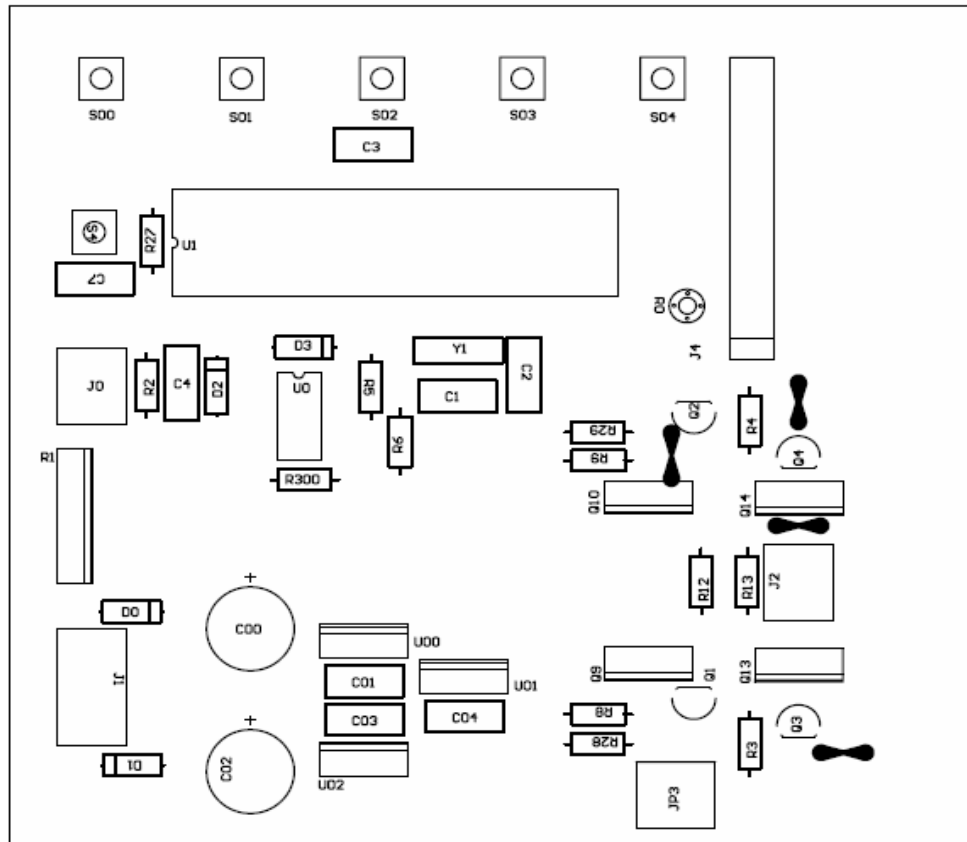
- **No se enciende la pantalla:** conecte el cable de la fuente de control con firmeza hacia la red de alimentación pública.
- **El contraste de visualización es borroso:** verifique que el voltaje que entrega la fuente es el correcto. Con la ayuda del potenciómetro usted puede ajustar el contraste de visualización, mueva el potenciómetro

delicadamente hasta que el contraste de la pantalla de visualización sea el apropiado.

- **El motor no gira:** asegúrese que la fuente del circuito de potencia este conectada a la red de suministro publico, verifique que el fusible este en buenas condiciones.
- **No se visualiza la velocidad de giro:** verifique las conexiones del circuito de control, verifique que exista transmisión de movimiento entre el motor y el generador.

UBICACIÓN DE LOS ELEMENTOS

CIRCUITO DE CONTROL



FUENTE DEL CIRCUITO DE CONTROL

