

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Sistema electromecánico para el control de la velocidad del viento en
el túnel de viento de la Universidad del Valle de Guatemala

Trabajo de graduación en modalidad de trabajo profesional presentado por
Josué Samuel Barrientos Castañeda
para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,
2013

**Sistema electromecánico para el control de la velocidad del viento en el
túnel de viento de la Universidad del Valle de Guatemala**


UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Sistema electromecánico para el control de la velocidad del
viento en el túnel de viento de la Universidad del Valle de
Guatemala


Trabajo de graduación en modalidad de trabajo profesional presentado por
Josué Samuel Barrientos Castañeda
para optar al grado académico de Licenciado en Ingeniería Mecatrónica


Guatemala,
2013

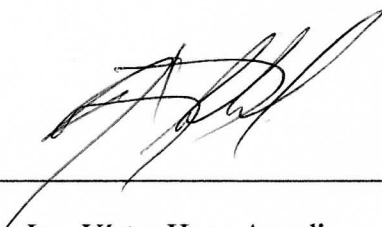
Vo. Bo. :

(f) 
MSc. Carlos Esquit

Tribunal Examinador:

(f) 
MSc. Carlos Esquit

(f) 
Ing. Roberto Delgado Rüegg

(f) 
Ing. Víctor Hugo Ayerdi

Fecha de aprobación: Guatemala 05 de diciembre de 2013

ÍNDICE

	Página
Lista de cuadros.....	ix
Lista de figuras.....	x
Resumen.....	xii
I. Introducción.....	1
II. Objetivos.....	2
A. Generales.....	2
B. Específicos.....	2
III. Justificación.....	3
IV. Marco teórico.....	4
A. Túneles de viento.....	4
B. Motores de corriente continua con escobillas.....	6
C. Drivers para motores DC con escobillas.....	11
D. Control de velocidad por modulación de ancho de pulsos.....	13
E. Tornillos de potencia.....	15
F. Acoples mecánicos.....	17
G. Sistemas de control automático.....	19
V. Alcances.....	26
VI. Diseño experimental.....	27
A. Descripción del túnel de viento.....	27

B.	Diseño general del sistema.....	29
C.	Selección de motor y caja reductora	30
D.	Driver y fuente de alimentación.....	33
E.	Mecanismo de acople	34
F.	Estructura de sujeción y protección	38
G.	Elementos finales del diseño físico	39
H.	Programación del microcontrolador.....	41
I.	Algoritmo de control automático.....	47
VII.	Resultados	55
A.	Construcción del sistema.....	55
B.	Eficiencia del sistema.....	58
C.	Listado de partes y costos.....	60
D.	Especificaciones	61
E.	Algoritmo de control automático	61
VIII.	Conclusiones	63
IX.	Recomendaciones.....	64
X.	Bibliografía.....	66
XI.	Anexos.....	68
A.	Explicación general del proyecto	69
B.	Cálculos para el motor seleccionado M5-RS540-12.....	72
C.	Planos de piezas fabricadas	74

D.	Código fuente del microcontrolador	81
E.	Diagrama del circuito	93
F.	Diagrama de PCB.....	95
G.	Planos de caja reductora.....	97
H.	Hoja de datos de fuente de voltaje	99

LISTA DE CUADROS

	Página
1. Efecto de incrementar los parámetros de un control PID en la respuesta a un escalón.....	22
2. Especificaciones del túnel de viento de la universidad.....	27
3. Características de los motores evaluados.....	31
4. Características de los motores a un torque de 5 lb in.....	31
5. Características del driver MD01B	33
6. Utilización de los módulos del microcontrolador.....	41
7. Listado de comandos que acepta el módulo	46
8. Ganancias de las acciones del controlador PD	53
9. Partes y costos del sistema diseñado.....	60
10. Especificaciones del sistema.....	61
11. Desempeño del control automático al cambiar la referencia de 5m/s a 14ms/.....	62

LISTA DE FIGURAS

	Página
1. Partes de un túnel de viento	5
2. Rotor de un motor dc con escobillas.....	7
3. Curva torque-velocidad típica para un motor DC.....	9
4. Curva voltaje-velocidad típica para un motor DC	10
5. Curva torque-corriente típica para un motor DC	10
6. Driver para motor unidireccional.....	12
7. Puente H.....	13
8. Modulación por ancho de pulsos	14
9. Tornillo de potencia.....	16
10. Esquema básico de un sistema de control.....	20
11. Túnel de viento modelo 303A de la Universidad del Valle de Guatemala.....	28
12. Detalle de la parte del túnel donde se encuentra el tornillo de potencia.....	28
13. Esquema general del sistema a diseñar.....	29
14. Motor M5-RS540-12	32
15. Caja reductora P60 26:1	32
16. Diagrama del driver MD01B	34
17. Acople Lovejoy L050	35
18. Espaciador fabricado en impresora de plástico.....	37
19. Pieza para centrar cojinete fabricada en impresora de plástico	37
20. Parte del túnel donde se encuentra el tornillo que debe girar el motor.....	38
21. Pieza fabricada para sujetar motor y caja	39
22. Circuito para activar contactor desde PIC por medio de relés.....	40

23. Encoder	40
24. Diagrama de flujo del programa	43
25. Filtro para la señal del microswitch	45
26. Relación entre velocidad de viento y distancia de apertura del túnel	47
27. Detalle de la conexión con el módulo de medición de velocidad	49
28. Diagrama de flujo del controlador PID	51
29. Prueba del sistema de control variando el setpoint	53
30. Prueba del sistema de control con varios setpoints utilizando dos rangos	54
31. Motor, caja, encoder y acople ensamblados	55
32. Acople, espaciador, pieza para centrar y rodamiento ensamblados en el tornillo	56
33. Sistema completamente ensamblado	56
34. Microswitches instalados	57
35. Relés para controlar bobina del contactor del ventilador	57
36. Circuito terminado en PCB	58
37. Prueba de desempeño del algoritmo de control automático	62
38. Esquema general del proyecto	70

RESUMEN

El presente trabajo de graduación forma parte de un proyecto dirigido por el departamento de Ingeniería Mecánica para reacondicionar el túnel de viento de la universidad. Dicho túnel carecía de toda la electrónica necesaria para ser funcional. El objetivo del proyecto era dotar al túnel con los sensores y actuadores para que pudiese ser usado para prácticas de laboratorio en cursos de la universidad y que todo esto fuese controlado y monitoreado desde una computadora. El proyecto fue dividido en varios trabajos de graduación realizados por distintos estudiantes de Ingeniería Mecatrónica.

Uno de los objetivos del proyecto era que el usuario del túnel pudiese desde la computadora ingresar la velocidad de viento deseada y que el sistema automáticamente la estableciera. Este fue el objetivo general de este trabajo de graduación. Con este fin se diseñó un sistema electromecánico que permite al usuario controlar la posición del ventilador del túnel y por consiguiente la velocidad del viento.

El sistema diseñado fue implementado en el túnel con resultados satisfactorios. El usuario puede controlar la posición del ventilador de manera manual y también entrar a un modo automático que implementa un controlador PID para fijar de manera automática la velocidad que el usuario ingrese.

I. INTRODUCCIÓN

El objetivo de este trabajo dentro del proyecto de rehabilitación del túnel de viento era que fuese posible controlar la velocidad del viento desde la computadora. El túnel ya contaba con una manera de hacer esto de forma manual que consistía en girar un timón que por medio de un tornillo de potencia separaba el ventilador de la sección de pruebas. Así se lograba disminuir la velocidad del viento. Al inicio del proyecto se exploraron varias alternativas, pero se decidió trabajar sobre el sistema existente sustituyendo el timón por un motor para que este fuese controlado desde la computadora.

El trabajo de graduación comprende todo el diseño del sistema para cumplir con los objetivos. Inicia con la selección del motor y parte de este para el diseño del resto del sistema. Se requirió de otros componentes mecánicos y electrónicos, algunos de los cuales fueron fabricados en la universidad, y del diseño de circuitos electrónicos y la programación de un microcontrolador.

Se creó un modo manual para controlar el motor e indirectamente la velocidad del viento. Se creó también un sistema de control del tipo PID que controla la posición del ventilador para mantener la velocidad del viento que el usuario define.

Dentro del proyecto existe un módulo maestro que se encargaba de la comunicación con la computadora y con todos los otros módulos del proyecto. La comunicación se hizo a través del protocolo I2C y se definió una interface (set de comandos) que enviaría el módulo maestro al sistema para funcionar.

El sistema fue completado e instalado en el túnel con resultados satisfactorios, cumpliendo con todos los objetivos planteados.

II. OBJETIVOS

A. Generales

Diseñar y construir un sistema electromecánico que permita establecer la velocidad del viento en la sección de pruebas del túnel de viento por el usuario desde una computadora.

B. Específicos

- Crear un mecanismo para que el tornillo que separa las dos partes del túnel sea accionado por un motor y no manualmente.
- Crear la estructura que permita sujetar el mecanismo al túnel y que proteja tanto al mecanismo como al usuario de las partes móviles.
- Crear un circuito eléctrico para el encendido y apagado del ventilador del túnel de viento desde la computadora.
- Crear la interface necesaria para la comunicación con la computadora a través del módulo central.
- Crear un modo manual en el que el usuario pueda controlar los movimientos que hace el motor desde la computadora.
- Crear el procedimiento o algoritmo para que el usuario seleccione una velocidad de viento desde la computadora y el sistema establezca automáticamente esa velocidad.
- Seleccionar una fuente de voltaje con la capacidad para suplir energía a todos los sistemas eléctricos que tendrá el túnel.
- Realizar el montaje de los sistemas diseñados en el túnel.
- Realizar pruebas para medir y evaluar el desempeño del sistema.

III. JUSTIFICACIÓN

Un túnel de viento en el cual se pueden realizar experimentos de laboratorio enriquece varios cursos que la universidad imparte, como el curso de Mecánica de Fluidos. Actualmente los temas sólo se enseñan por medio de clases magistrales y carecen de un laboratorio en el cuál los estudiantes puedan visualizar y reforzar los conceptos aprendidos. El túnel de viento será también una herramienta que podría ser utilizada en investigaciones como megaproyectos u otras.

El control de la velocidad del viento da flexibilidad a los experimentos al hacer posible realizar una mayor variedad de experimentos. Al poder modificar la velocidad del aire se puede examinar cómo esta variable afecta los sistemas bajo estudio. El cambio de un sistema manual a uno electrónico moderniza el equipo y al estar conectado todo a una computadora se puede registrar los datos y realizar las pruebas con mayor facilidad y rapidez.

IV. MARCO TEÓRICO

A. Túneles de viento

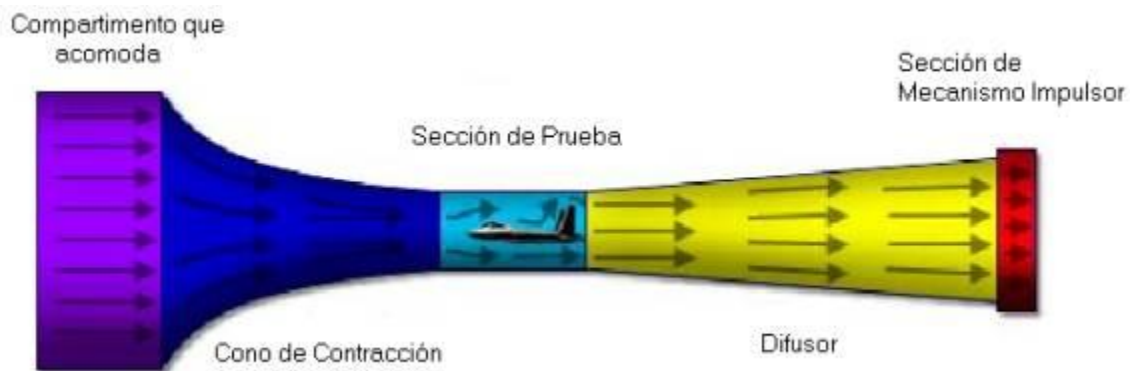
Un túnel de viento es una herramienta de investigación para el estudio de los efectos del viento sobre objetos. El objetivo de un túnel de viento es simular las condiciones que experimenta el objeto en una situación real. La premisa es que el resultado es el mismo ya sea que el objeto se mueva a través del aire o que el objeto permanezca estático mientras el aire pasa alrededor de él. De esta manera se pueden realizar pruebas para comprender el comportamiento del objeto y mejorar el diseño del mismo sin necesidad de probarlo en la situación real. Los túneles de viento son comúnmente utilizados para pruebas sobre aviones, automóviles, puentes y edificios (Benson, 1992).

Cuando se hicieron los primeros intentos de crear máquinas voladoras, se hizo evidente que era necesaria una herramienta que permitiera estudiar el comportamiento de estas máquinas sin necesidad de realmente volarlas. El primer túnel de viento fue creado por Francis Herbert Wenham en 1871. Los hermanos Wright construyeron un túnel que les facilitó el desarrollo del *Wright Flyer*. Hoy en día los túneles de viento son indispensables para el diseño de aeronaves (Benson, 1992).

Las pruebas que se realizan en un túnel de viento comúnmente sirven para determinar las fuerzas de arrastre y sustentación. También se puede utilizar para visualizar el flujo del aire. El método más común es utilizando humo. Otra prueba que se realiza en un túnel es determinar la velocidad y presión en distintos puntos del objeto (Benson, 2012).

1. Partes de un túnel de viento. En general los túneles de viento están compuestos por cinco partes básicas (véase Figura 1). La primera de estas es el ventilador que genera el flujo de aire en el túnel. Generalmente este es accionado por un motor eléctrico. El aire entra al túnel pasando primero acondicionadores de flujo cuyo objetivo es hacer que el flujo sea uniforme. Después atraviesa un cono de contracción que reduce el área transversal para aumentar la velocidad del viento. Después de este cono el aire entra en la sección de pruebas que es donde se coloca el objeto de estudio. Para túneles de velocidades bajas este es el punto con menor área transversal y por lo tanto mayor velocidad. Finalmente está el difusor donde el área se aumenta para bajar la velocidad del aire. El ventilador se coloca después del difusor. En los túneles de viento el aire no es empujado a través de la sección transversal sino succionado (Benson, 2012).

Figura 1 Partes de un túnel de viento¹



¹ Imagen por Víctor Martín tomada de:
[<http://fluidos.eia.edu.co/hidraulica/articulosos/flujodegases/tuneldeviento/tuneldeviento.html>]

2. Tipos de túneles de viento. Existe una gran variedad de tipos de túneles de viento. Se pueden clasificar de dos maneras. Según la velocidad de aire que manejen se clasifican como subsónicos, transónicos, supersónicos o hypersonicos. Este es un parámetro importante ya que la velocidad determina que tanto afecta la compresibilidad del aire. Si se trata de un túnel subsónico los efectos de la compresibilidad se pueden despreciar. La otra manera en que se pueden calificar los túneles de viento es según si son cerrados o abiertos. Los túneles cerrados recirculan el aire a través del túnel a diferencia de los abiertos que de un lado succionan aire y del otro lo expulsan (Benson, 2009).

B. Motores de corriente continua con escobillas

Existe una gran variedad de tipos de motores eléctricos. De forma general se clasifican como de corriente continua y de corriente alterna (de una o varias fases). Los motores de corriente continua con escobillas son utilizados en una gran variedad de aplicaciones debido a su bajo costo, facilidad de controlar electrónicamente, y la amplia disponibilidad comercial de distintos tamaños y formas.

Si bien hay distintos tipos de motores DC con escobillas todos se componen de cuatro partes básicas: el estator, el rotor, el conmutador y las escobillas. La diferencia radica en el estator. Como su nombre lo indica, este componente es estático y su función es crear un campo magnético estacionario que rodea al rotor. Este campo puede ser creado por medio de imanes permanentes o por embobinados (Mouton, 2008).

El rotor es la parte que gira del motor DC. Está compuesto por uno o más embobinados que al ser energizados generan un campo magnético (véase Figura 2). Los polos que se crean en el rotor son atraídos por los polos opuestos del estator. Así se crea el movimiento giratorio, pero para que pueda continuar los polos del rotor deben cambiar constantemente. Esta es la función del conmutador (Condit, 2004).

A diferencia de motores de otro tipo, como los DC sin escobillas, estos motores no requieren de un controlador electrónico para cambiar los polos en el rotor. En los motores DC con escobillas esta conmutación se realiza de manera mecánica por medio del conmutador y las escobillas. El conmutador está compuesto por un anillo montado sobre el eje del motor que posee una o varias láminas, generalmente de cobre, aisladas entre sí y conectadas a los embobinados del motor. Estas láminas son llamadas delgas. Sobre el conmutador se colocan las escobillas hacen contacto con las delgas, alimentando las bobinas del rotor. Puesto que las escobillas son fijas, al girar el motor entran en contacto con distintas delgas y así las bobinas son alimentadas con distinta polaridad. El roce de las escobillas con el conmutador las convierte en la parte que más desgaste sufre del motor y deben ser cambiadas con relativa frecuencia (Condit, 2004).

Figura 2 Rotor de un motor dc con escobillas²



² Imagen por Jjmontero9 [CC-BY-SA-3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>)], vía Wikipedia Commons

1. Motores de imán permanente. Los motores de imán permanente son los motores de corriente continua con escobillas más comunes. Estos utilizan imanes permanentes en el estator para crear el campo magnético fijo. Generalmente son utilizados en aplicaciones que requieren baja potencia porque son menos costos que los otros tipos. La desventaja es que con el tiempo los imanes se desmagnetizan. La relación entre voltaje y velocidad es altamente lineal en estos motores así como también la relación entre torque y corriente consumida. Estos motores responden rápidamente a cambios en el voltaje de alimentación. Estas características hace que sea posible controlar su velocidad con precisión y adecuados para un control por medio de PWM (Mouton, 2008).

2. Motores con embobinados. Estos motores utilizan embobinados en el estator en lugar de imanes permanentes. Generalmente son utilizados en aplicaciones que requieren de 5Hp en adelante. El embobinado del estator puede estar conectado en paralelo o en serie con el embobinado del rotor. Los motores con embobinados en paralelo poseen torque estable a velocidades bajas pero decrece rápidamente al aumentar la velocidad. Tienen también buenas características para el control de velocidad. Los motores con embobinados en serie poseen alto torque a velocidades bajas pero bajo torque a velocidades altas. La velocidad aumenta rápidamente al ser removida una carga. Estas características los hacen ideales para aplicaciones con cargas constantes, pero no para control de velocidad. También existen motores que combinan embobinados en serie y en paralelo. Estos motores combinan características de los anteriores. Son ideales para aplicaciones como generadores. También son los más costosos de todos los tipos de motores de corriente continua con escobillas (Mouton, 2008).

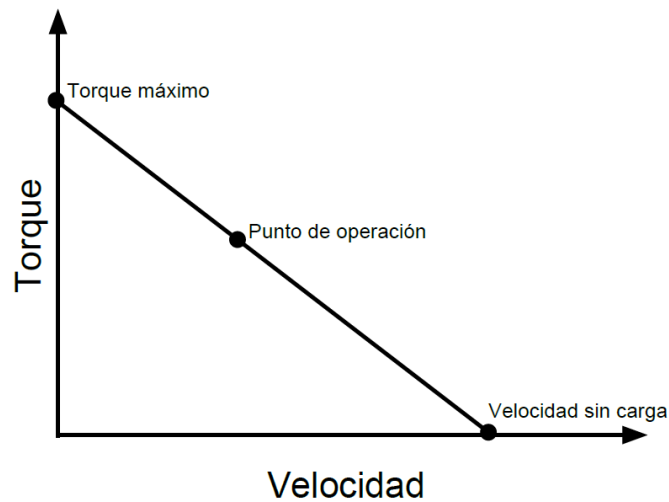
3. Curvas características de los motores de corriente continua. Las relaciones que existen entre torque-velocidad, voltaje-velocidad y torque-corriente caracterizan al motor de corriente continua. Normalmente estas relaciones se representan con curvas y es necesario considerarlas al seleccionar un motor.

La velocidad en un motor DC es inversamente proporcional al torque de salida. Al aumentar el torque requerido del motor la velocidad disminuye. La relación es lineal y la curva puede ser obtenida con dos puntos conocidos. Normalmente se utiliza la velocidad sin carga (torque cero) y el torque de parada (velocidad cero) para aproximar la curva. La relación se puede expresar con la Ecuación 1 (Page, 1999).

$$Torque = T_{m\acute{a}x} - \omega \frac{T_{m\acute{a}x}}{\omega_{m\acute{a}x}}$$

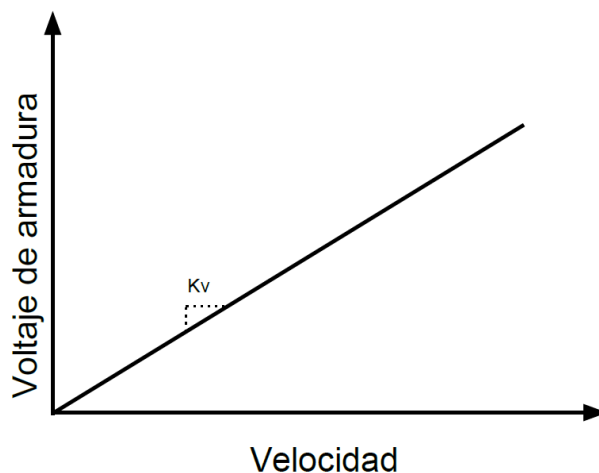
Ecuación 1

Figura 3 Curva torque-velocidad típica para un motor DC



La relación entre voltaje y velocidad es también lineal. Se aproxima tomando la velocidad a un voltaje y asumiendo velocidad cero a voltaje cero. La pendiente de esta curva es la constante K_v del motor. Hay que tener en cuenta que a voltajes bajos no se sigue la curva sino hasta que se vence la inercia del motor (Joliet Technologies, s.f.).

Figura 4 Curva voltaje-velocidad típica para un motor DC

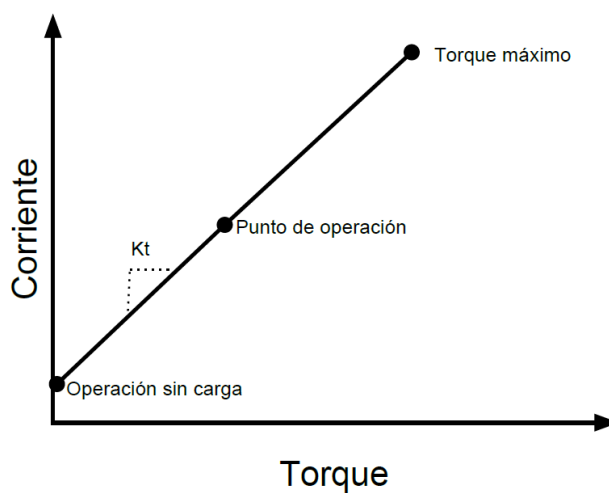


La corriente que consume el motor depende del torque, sin importar el voltaje de entrada. La relación, de nuevo, es lineal. La pendiente de la curva es la constante K_t . La curva se puede obtener con la corriente sin carga (torque cero) y algún otro punto como la corriente a torque máximo (Durfee, 2011). Esto resulta en la Ecuación 2.

$$\text{Corriente} = C_{T=0} + T/K_t$$

Ecuación 2

Figura 5 Curva torque-corriente típica para un motor DC



Otras curvas que normalmente se utilizan son la de potencia de salida y eficiencia del motor. La potencia de salida en cualquier sistema giratorio es el producto entre el torque y la velocidad. Si se combina esta relación con la Ecuación 1 se obtiene que:

$$Potencia\ de\ salida = T\omega = -\frac{\omega_{m\acute{a}x}}{T_{m\acute{a}x}}T^2 + \omega_{m\acute{a}x}T \quad \text{Ecuación 3}$$

La eficiencia se calcula dividiendo la potencia de salida entre la de entrada que es el producto entre el voltaje y la corriente.

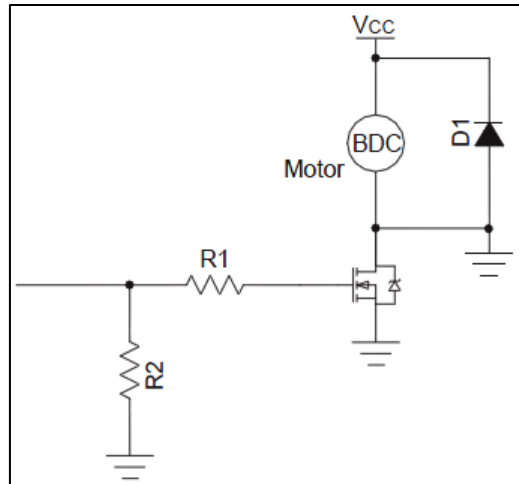
$$Eficiencia = \frac{P_{salida}}{P_{entrada}} = \frac{-\frac{\omega_{m\acute{a}x}}{T_{m\acute{a}x}}T^2 + \omega_{m\acute{a}x}T}{VI} \quad \text{Ecuación 4}$$

C. Drivers para motores DC con escobillas

Los drivers son circuitos que permiten el control de un motor. El control puede ser solamente de encendido y apagado o de velocidad y de una dirección o bidireccional. Estos circuitos proveen una manera de regular la corriente y voltaje en los devanados del motor.

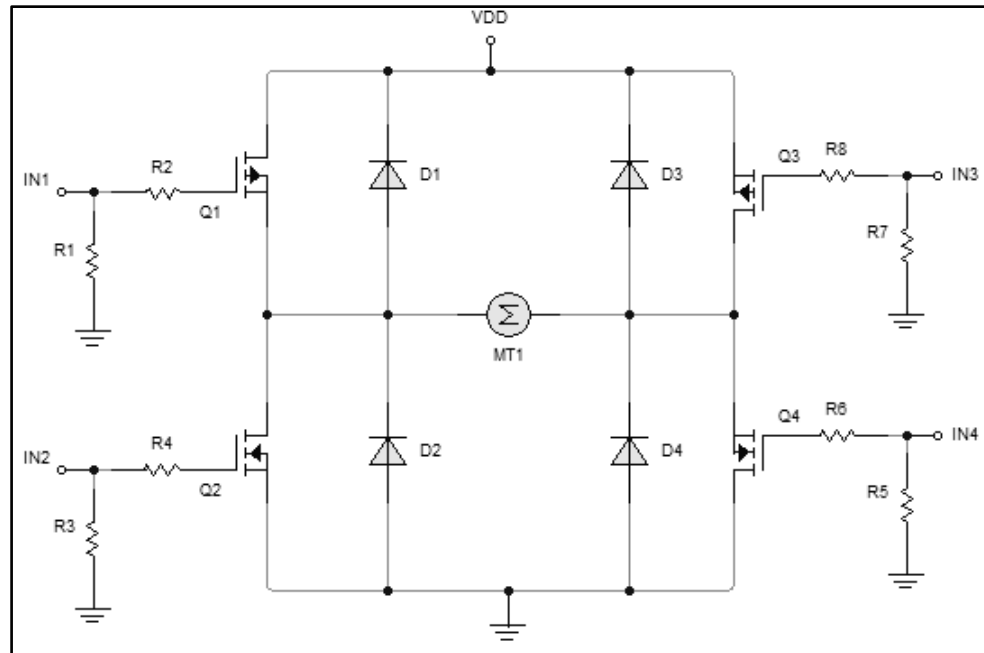
Para aplicaciones en las que se necesita que el motor gire en una sola dirección se utiliza un circuito como el mostrado en la Figura 6. El controlador aplica un voltaje en la compuerta del transistor habilitando el paso de corriente en el motor. En paralelo al motor se conecta un diodo que permite disipar la energía almacenada en la bobina del motor al deshabilitar el transistor. Sin este diodo la corriente dañaría el transistor. Las resistencias sirven para asegurar que la compuerta del transistor nunca esté flotando lo cual haría que su comportamiento sea impredecible (Condit, 2004).

Figura 6 Driver para motor unidireccional



Para aplicaciones en las que se requiere que el motor gire en ambas direcciones se utiliza un circuito denominado puente H (Figura 7). Este circuito permite que la corriente atraviese el motor en ambas direcciones. Al activar los transistores en esquinas opuestas mientras los otros permanecen desactivados la corriente fluye de la esquina superior a la inferior (Q1 y Q4). Al activar la pareja contraria la corriente en el motor cambia de sentido (Q3 y Q2). Si ningún transistor es activado el motor está libre y frena solo por la fricción. Si se activan parejas inferiores o superiores el motor está cortocircuitado y frena rápidamente. En este modo el motor actúa como generador. Al igual que para el driver de un solo sentido se deben colocar diodos y resistencias por los motivos expuestos anteriormente (Condit, 2004).

Figura 7 Puente H



D. Control de velocidad por modulación de ancho de pulsos

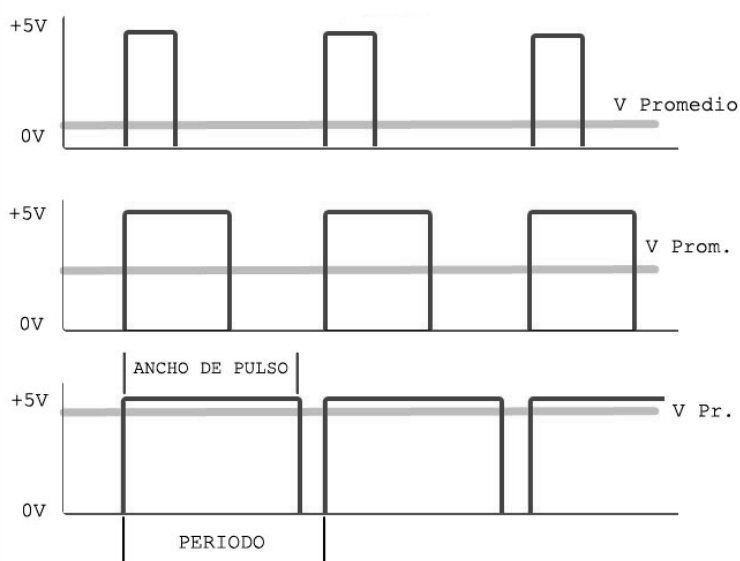
La velocidad de un motor dc puede ser variada al cambiar el voltaje en sus terminales o rápidamente conectando y desconectando la alimentación. Este último método es mucho más eficiente que el primero pues no requiere de una resistencia ineficiente que disipe la energía (Condit, 2004).

Para lograr la conexión y desconexión del motor se utiliza uno de los drivers mencionados en el apartado anterior y se les excita con una señal cuadrada. El ciclo de trabajo de esta señal se puede variar logrando así cambiar cuánto tiempo el motor pasa conectado lo cual determina su velocidad. Este método se denomina modulación por ancho de pulsos o PWM por sus siglas en inglés. Un microcontrolador puede generar esta señal fácilmente y es la razón por la que comúnmente se utiliza el método al usar estos dispositivos.

Al excitar a un motor con una señal cuadrada este actúa como un filtro pasa bajas. Si la señal tiene una frecuencia suficientemente alta la corriente en el motor tiende a ser continua. El valor de esta corriente y la velocidad del motor dependen del voltaje promedio de la señal de entrada. El voltaje promedio depende del ciclo de trabajo y del voltaje de alimentación según la Ecuación 5 donde D es el ciclo de trabajo (Condit, 2004). La Figura 8 muestra esto de manera gráfica.

$$V_{prom} = D * V_{cc}$$

Ecuación 5

Figura 8 Modulación por ancho de pulsos³

Como se mencionó anteriormente, la velocidad en un motor de corriente continua con escobillas es proporcional al voltaje aplicado en sus terminales. Entonces, la velocidad

³ Imagen por Pablo Murillo tomada de:
<http://www.arduteka.com/2011/11/tutorial-arduino-0003-entrada-analogica-y-salida-pwm/>

del motor es proporcional también al ciclo de trabajo pues este determina el voltaje promedio.

La frecuencia de la señal PWM es un parámetro importante. Esta debe ser suficientemente alta para lograr que el motor la filtre y se cumpla lo anterior. Tampoco puede ser demasiado alta pues disminuye la eficiencia del sistema al incrementar las pérdidas de conmutación de los transistores. Adicional a esto se debe considerar que al ser alimentado con una señal PWM el motor hará ruido que depende de la frecuencia de la señal. Normalmente se utilizan frecuencias entre 4kHz y 20kHz. Se recomienda que se experimente con distintas frecuencias para hallar la más adecuada (Condit, 2004).

E. Tornillos de potencia

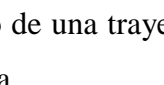
Los tornillos de potencia son elementos mecánicos utilizados para convertir movimiento rotatorio en movimiento lineal y ejercer la fuerza necesaria para mover una carga a lo largo de una trayectoria recta. El tornillo funciona en conjunto con una tuerca como muestra la . Si la tuerca se mantiene sin girar mientras el tornillo gira sin moverse axialmente, la tuerca avanza a lo largo del tornillo. Si la tuerca se fija y se gira el tornillo será este el que se mueve linealmente. Los tornillos de potencia pueden ser de rosca cuadrada, trapezoidal o Acme. Las primeras dos son más eficientes pero la Acme es más fácil de maquinar. La rosca trapezoidal se utiliza especialmente cuando sólo se transmite fuerza en una dirección (Mott, 2006).

Figura 9 Tornillo de potencia⁴

Los parámetros que definen a un tornillo de potencia son: roscas por pulgada (n), paso ($L = 1/n$), diámetro nominal (D), diámetro de paso (D_p) y ángulo de rosca (φ). Para calcular el torque necesario para mover una carga hacia arriba o de manera horizontal se utiliza la Ecuación 6 (Mott, 2006). En esta fórmula f es el coeficiente de fricción entre el tornillo y la tuerca.

$$T = \frac{FD_p}{2} \left(\frac{\cos \varphi \tan \lambda + f}{\cos \varphi + f \tan \lambda} \right) \quad \text{Ecuación 6}$$

$$\tan \lambda = \frac{L}{\pi D_p} \quad \text{Ecuación 7}$$

La eficiencia de un tornillo de potencia se puede calcular dividiendo el torque que sería necesario sin fricción dentro del real (Mott, 2006).

⁴ Imagen por Antonio Pérez González tomada de: http://www.mecapedia.uji.es/tornillo_de_potencia.htm

$$e = \frac{T_{f=0}}{T} = \frac{FL}{2\pi T}$$

Ecuación 8

Para los tornillos Acme de una sola entrada la eficiencia es baja. Normalmente está entre 20% y 30%. Aumentando el avance y el número de entradas se obtienen mejores eficiencias pero la ventaja mecánica disminuye. Otra opción es utilizar un tornillo de bolas que tiene una eficiencia de 90% comúnmente, pero tiene ciertas desventajas respecto a los tornillos de potencia (Mott, 2006).

F. Acoples mecánicos

Es muy común la necesidad de transmitir potencia de un eje a otro; por ejemplo, unir el eje de un motor al de una caja reductora. Esta unión se hace por un elemento mecánico denominado acoplamiento. Existen dos categorías generales de acoplamientos: rígidos y flexibles.

Los acoplamientos rígidos unen firmemente los dos ejes entre sí. No hay posibilidad de juego o movimiento relativo entre ellos. Los ejes quedan unidos como si fueran una sola pieza. Entre los acoplamientos rígidos se encuentran las uniones bridadas, las abrazaderas y los acoplamientos por sujeción cónica. Este tipo de acoplamiento es adecuado para aplicaciones en las que se necesita que no haya juego y en las que se puede asegurar un alineamiento altamente preciso al instalar y durante la operación. De lo contrario, cualquier desalineación podría generar esfuerzos en el acople que llevarían a una falla prematura (Mott, 2006).

En la mayoría de aplicaciones no se puede asegurar, ni es necesario, un alineamiento altamente preciso. Los acoples flexibles son los adecuados para estos casos pues toleran cierta desalineación. Estos acoples se componen de varias partes que pueden moverse al existir desalineación y así prevenir que se generen esfuerzos. Existe gran variedad de acoplamientos flexibles entre los cuales están: acoples tipo araña, acoples de engranajes,

acoples de cadena (Mott, 2006). La selección de uno u otro acople se realiza en base a los diámetros de los ejes a unir, el torque a transmitir y cualquier otro requerimiento especial como juego, características de falla y condiciones de operación.

Para fijar un acople a un eje se debe utilizar algún elemento de fijación. Comúnmente se utilizan cuñas, prisioneros, pasadores o conicidad en alguno de los elementos. En cualquier caso se debe diseñar las dimensiones del fijador para asegurar que el acople no fallará al accionarlo. Existen dos modos de falla posibles: por cortante y por compresión. El cortante se genera en la cuña, prisionero o pasador y la compresión se genera también en el eje y en el acople (Mott, 2006).

El esfuerzo cortante se calcula con la siguiente fórmula:

$$\tau = \frac{F}{A} = \frac{\frac{T}{D/2}}{(WL)} = \frac{2T}{DWL} \quad \text{Ecuación 9}$$

En esta fórmula D es el diámetro del eje, W es la longitud transversal al eje y L la longitud a largo del eje. El esfuerzo resultante debe ser menor al esfuerzo en cortante de diseño. Este generalmente se toma como:

$$\tau_d = \frac{0.5s_y}{N} \quad \text{Ecuación 10}$$

Para la anterior fórmula N es un factor de seguridad que queda a criterio del diseñador. Adicionalmente se debe asegurar que el acoplamiento no fallará por compresión. Esto se debe hacer considerando el material del ensamble con menor resistencia a la fluencia (Mott, 2006). Para una cuña cuadrada la fórmula se muestra en la Ecuación 11.

$$\sigma = \frac{F}{A} = \frac{\frac{T}{D/2}}{(LH/2)} = \frac{4T}{DLH}$$

Ecuación 11

en donde H es la altura de la cuña. De nuevo el resultado de este cálculo debe ser menor al esfuerzo de diseño por compresión (Mott, 2006). Este es:

$$\sigma = \frac{S_y}{N}$$

Ecuación 12

G. Sistemas de control automático

El objetivo de todo sistema de control es llevar un sistema, comúnmente llamado planta, a un estado deseado. El ser automático significa que no requiere de acción humana alguna.

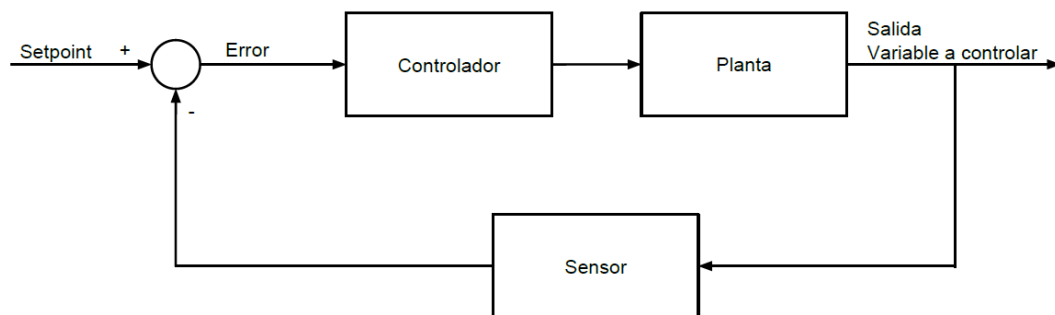
Los sistemas de control pueden ser clasificados como de lazo abierto o de lazo cerrado. La diferencia entre estos radica en que la acción del control de lazo cerrado depende de la salida, o variable a controlar, de la planta. En uno de lazo abierto la salida sólo depende de la entrada.

Los sistemas de lazo abierto son comúnmente utilizados porque son simples en su diseño y construcción, usualmente no tienen problemas de estabilidad y son convenientes cuando la salida es difícil de medir. La exactitud de estos sistemas depende de qué tan exacto es el modelo matemático que se tiene de la planta. La planta se debe conocer para poder predecir cómo se comportará ante una entrada para poder introducir la entrada requerida para llevar la salida al estado deseado. Si la planta cambia en algún parámetro, o se introduce una perturbación, el sistema tendrá un error que solamente se puede corregir modificando el sistema. Debido a esto requieren de calibraciones periódicas (Gosh, 2004).

Los sistemas de lazo cerrado miden la salida de la planta y producen la señal de control según el error que existe entre la medición y el valor deseado. Este enfoque permite gran exactitud pues se corrige el error. Puesto que se mide la señal de salida, el sistema es menos sensible a cambios en la planta o el sistema en sí. Las desventajas de estos sistemas son que son más complejos de diseñar e implementar y que pueden ser inestables (Gosh, 2004).

La Figura 10 muestra el esquema básico de un sistema de control de lazo cerrado. El controlador en un sistema de control de lazo cerrado es el elemento que procesa el error y produce la señal de control. La entrada al sistema es la referencia, o setpoint, que es el valor al cual se desea que la salida siga. El error se calcula restando el valor actual de la salida. La medición de la salida se hace a través de sensores y es importante notar que estos pueden no ser ideales, es decir, pueden introducir retrasos, ruido y desfases en la señal. El controlador es el elemento que procesa el error y produce la señal de control que excita la planta según sea necesario para llevar el error a cero. Existen distintos tipos de controladores pero el más común es el controlador PID debido a su simplicidad y buen desempeño en la mayoría de aplicaciones (Zhong, 2006).

Figura 10 Esquema básico de un sistema de control



1. Sistemas de control PID. El controlador PID consiste de tres parámetros: proporcional, integral y derivativo. La salida de este controlador depende del valor del

error, la suma del error pasado y la tasa de cambio del error. Cada uno de estos parámetros es escalado por una ganancia. La función de transferencia de un PID ideal es la siguiente (Zhong, 2006).

$$C(s) = K_p + \frac{K_I}{s} + K_d s$$

Ecuación 13

En la Ecuación 13, K_p , K_I , y K_d son las ganancias de las acciones proporcional, integral y derivativa, respectivamente. Al diseñar un sistema de control PID la principal tarea es determinar los valores de cada ganancia que producen el comportamiento deseado del sistema.

Los parámetros que comúnmente definen el comportamiento deseado de un sistema se establecen según la respuesta que tiene a un escalón en la entrada. Estos parámetros son:

- Tiempo de subida: el tiempo que le toma a la planta llegar al 90% del valor deseado.
- Sobreelevación: cuánto se pasa la salida de su valor final.
- Tiempo de asentamiento: el tiempo que le toma a la planta converger a su valor final.
- Error en estado estable: la diferencia entre el valor deseado de salida y el valor al cual converge la salida (Zhong, 2006).

La imagen muestra la curva típica de la salida de un sistema de control ante un escalón en la entrada y el valor de los parámetros.

Las ganancias del controlador PID afectan estos parámetros de distinta manera. El efecto de incrementar la ganancia de cada acción se resume en el Cuadro 1 (Zhong, 2006).

Cuadro 1 Efecto de incrementar los parámetros de un control PID en la respuesta a un escalón

<i>Parámetro</i>	<i>Tiempo de subida</i>	<i>Sobreelevación</i>	<i>Tiempo de asentamiento</i>	<i>Error en estado estable</i>
K_p	Decrece	Incrementa	-	Decrece
K_I	Decrece	Incrementa	Incrementa	Se elimina
K_d	-	Decrece	Decrece	-

Uno de los motivos de la popularidad del sistema PID es que es posible obtener buenos resultados simplemente modificando las ganancias según el comportamiento observado y la tabla antes mostrada. Es decir, es posible obtener buenos resultados sin necesidad de realizar cálculos matemáticos o de conocer a detalle la planta a controlar. El procedimiento generalmente es el siguiente (Zhong, 2006).

1. Incrementar K_p para reducir el tiempo de subida.
2. Incrementar K_d para reducir la sobreelevación y oscilaciones.
3. Incrementar K_I para eliminar el error en estado estable.

Si el PID es un controlador adecuado para la aplicación, este procedimiento resultará en valores adecuados para las ganancias. El problema es que puede llevar tiempo ajustar los valores. Existen métodos que permiten determinar valores iniciales que producen respuestas cercanas a la deseada. Después de aplicarlos se debe, de igual manera, ajustar manualmente.

2. Consideraciones adicionales para controladores PID. Si el controlador PID se implementa directamente como en 3, el sistema tendrá algunas características indeseables en la práctica que se deben a limitantes físicas.

La acción derivativa presenta dos problemáticas. La primera es que la acción derivativa amplifica cambios grandes del error en tiempos cortos. Es en esencia un filtro pasa altas. Esto provoca que el sistema sea sensible a ruido. Una solución es utilizar filtrar la señal de retroalimentación para eliminar el ruido. El filtro debe ser diseñado cuidando que no se filtren cambios reales en el sistema (Aström & Murray, 2008).

La segunda problemática con la acción derivativa sucede con cambios rápidos de la señal de referencia. Cuando esto sucede, como en un escalón, el error aumenta súbitamente. Este cambio es amplificado por la acción derivativa y se produce un pico en la salida del controlador. Este problema es conocido como *derivative kick*. Este pico puede dañar elementos del sistema especialmente si incluye elementos mecánicos como motores o engranajes. La manera en que se soluciona el problema es aplicando la acción derivativa sobre la salida del sistema y no sobre el error (Johnson, *et al.*, 2006). Se puede notar que:

$$E = R - S \quad \text{Ecuación 14}$$

$$\frac{dE}{dt} = \frac{dR}{dt} - \frac{dS}{dt} \quad \text{Ecuación 15}$$

Entonces, se puede concluir que si la derivada de la referencia es cero, la derivada del error es igual al negativo de la derivada de la salida. Por lo tanto al pasar la acción derivativa a la retroalimentación el comportamiento es el mismo excepto en momentos en los que cambia la referencia. Así si se pretende seguir solo escalones se evita que los cambios en el error por cambios en la referencia sean amplificados y se elimina el pico.

Si se pretende seguir cambios en la referencia esta topología no sería adecuada. (Johnson, *et al.*, 2006).

Otra problemática que se tiene al aplicar un sistema PID es la saturación de actuadores. Esto sucede cuando el controlador produce una señal que llevaría al actuador más allá de sus límites físicos. El problema que esto produce es en la acción integral. Mientras el actuador está en su límite la acción integral se mantiene incrementando. El resultado es que si el error cambia de signo y el actuador debe moverse alejándose del límite antes deberá reducirse el término integral. La manera para resolver este problema es no realizar la acción integral cuando el sistema está saturado (Johnson, *et al.*, 2006).

Al igual que la acción derivativa, la acción proporcional produce un pico en la salida del controlador cuando la referencia cambia súbitamente. Si bien esto produce tiempos de subida más rápidos, el pico puede dañar elementos del sistema. Se puede conseguir la misma rapidez sin el pico. La solución es la misma que resuelve el pico de la acción derivativa, actuar sobre la retroalimentación y no sobre el error. Para que sea posible es necesario que exista la acción integral (ganancia no igual a cero) ya que es esta acción la que llevará al sistema al valor deseado. La ganancia de la acción integral será mucho mayor que en la configuración original (Johnson, *et al.*, 2006).

3. Implementación del controlador PID en sistemas digitales. Es posible implementar un controlador PID en un sistema digital como una computadora o un microcontrolador. Para hacer esto se debe muestrear las señales de referencia y de retroalimentación, computar el error y la salida del controlador y excitar la planta con ella. El cálculo de la acción proporcional es directo: el error se multiplica por una ganancia K_p . Para las acciones derivativa e integral se debe realizar aproximaciones (Charais & Lourens, 2004). Para la acción derivativa la aproximación es la siguiente:

$$\frac{dE}{dt} \approx \frac{(E(n) - E(n - 1))}{T_s}$$

Ecuación 16

Para la acción integral se puede utilizar la siguiente aproximación:

$$\int_0^t E(t)dt = T_s \sum_0^N E(n)$$

Ecuación 17

La tasa de muestreo es un parámetro importante en un sistema de control digital. Si se elige una tasa de muestreo muy lenta se dificulta el control pues el tiempo de respuesta a cambios en el sistema será demasiado grande. Si se utiliza una tasa demasiado alta se sobrecarga al procesador y el sistema será más sensible a ruido en la señal de retroalimentación. La tasa de muestreo se debe elegir según las condiciones de cada aplicación tomando en cuenta estas implicaciones (Charais & Lourens, 2004).

V. ALCANCES

Si bien este trabajo de graduación tiene como objetivo crear un sistema que haga posible que un usuario defina la velocidad del viento en el túnel de viento desde una computadora, hay dos puntos necesarios para dicho fin que no son parte de este trabajo de graduación.

Este trabajo de graduación no incluye la medición de la velocidad del viento. Esto es parte del módulo realizado por Marco Guzmán y se tomará como válidos los datos de velocidad de viento que provea dicho módulo. Se asume que esta velocidad es un promedio representativo de lo que sucede dentro de la sección de pruebas.

De igual manera, no es parte de este trabajo de graduación el crear el programa de computadora que permita a un usuario interactuar con el sistema. Esto corresponde al módulo realizado por Roberto Moreno. Este trabajo de graduación se limitará a definir los comandos y crear los procedimientos internos del módulo para que el sistema pueda ser controlado por un usuario desde un programa de computadora si envía dichos comandos al módulo.

Por lo demás, este trabajo de graduación comprende el diseño de todo lo necesario para que el usuario pueda controlar satisfactoriamente la velocidad del viento en el túnel. Incluye, entonces, el diseño de todo mecanismo, estructura, circuito eléctrico o electrónico necesario y la programación de controladores requeridos.

VI. DISEÑO EXPERIMENTAL

A. Descripción del túnel de viento

El túnel de viento que posee la universidad es un túnel de fines didácticos. Las dimensiones de la sección de pruebas permiten solamente modelos pequeños. El Cuadro 2 resume las especificaciones del túnel. La Figura 11 es una foto del túnel. La parte de la derecha del túnel (azul metálico brillante) es donde se aloja el ventilador. Esta parte está sobre ruedas y se puede separar del difusor (azul opaco) para disminuir la velocidad del viento en la sección de pruebas (transparente). La Figura 12 muestra mayor detalle de la parte en la que se encuentra el tornillo de potencia (véase también Figura 20). La tuerca plástica que está en el tornillo está unida a la estructura donde está el ventilador por lo que al girar el tornillo el ventilador se mueve. La barra azul unía un timón al tornillo para que se pudiese girar manualmente.

Cuadro 2 Especificaciones del túnel de viento de la Universidad del Valle de Guatemala

Marca	Engineering Laboratory Design
Modelo	303A
Tipo	Subsónico abierto
Año de manufactura	1984
Dimensiones de la sección de pruebas	12'' x 12'' x 24''
Rango de velocidades	3m/s – 30 m/s
Motor	5Hp
Voltaje de alimentación	230VAC Trifásico

Figura 11 Túnel de viento modelo 303A de la Universidad del Valle de Guatemala



Figura 12 Detalle de la parte del túnel donde se encuentra el tornillo de potencia



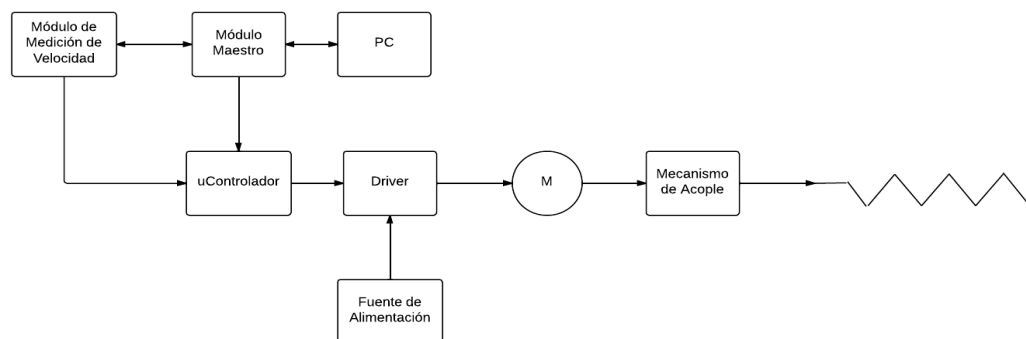
B. Diseño general del sistema

El diseño del sistema inició por definir un esquema general para luego diseñar cada subsistema por separado. La Figura 13 muestra el esquema. Se inició con la selección de un motor adecuado y de un driver y fuente de alimentación acordes a dicho motor.

A continuación se diseñó el mecanismo de acople que permite transmitir el movimiento del motor hacia el tornillo. Para controlar los movimientos del motor se decidió, debido a la experiencia que se tiene con estos elementos, utilizar un microcontrolador. Este microcontrolador se encarga de recibir comandos del módulo maestro (que se conecta a la computadora) y generar las salidas necesarias para mover el motor como sea necesario.

Teniendo completo el sistema físico se procedió a programar el microcontrolador para el modo manual. Una vez se tuvo funcionando este modo se inició el desarrollo del algoritmo para el control automático. Se finalizó con la comunicación al módulo maestro (véase anexo AXI.A para mayor información de los otros módulos del proyecto).

Figura 13 Esquema general del sistema a diseñar



C. Selección de motor y caja reductora

El motor fue elegido en base a la velocidad y torque requeridos por la aplicación. El torque de salida debe ser suficiente para poder girar el tornillo a una velocidad y un consumo de corriente razonables. Para conocer el torque necesario para girar el tornillo se realizó una medición simple. Se aplicó una fuerza en el timón, perpendicular al radio de este, utilizando un dinamómetro de resorte. La fuerza necesaria para mover el timón resultó ser de 2lb. La distancia a la que se aplicó la fuerza era de 2.5in. Por lo cual el torque resulta ser:

$$\tau = F \times r = (2 \text{ lb})(2.5 \text{ in}) \sin(90^\circ) = 5 \text{ lb} \cdot \text{in} \quad \text{Ecuación 18}$$

La velocidad de giro del motor determina la velocidad a la cual se separa el ventilador por consiguiente a la que será posible cambiar la velocidad del viento. Se decidió que el tiempo mínimo en el cual se debía separar o juntar completamente el ventilador a máxima velocidad fuera de 10s. La velocidad máxima del motor depende de este tiempo, la longitud máxima de apertura, y el número de vueltas por pulgada del tornillo:

$$\omega_{min} = \frac{14 \text{ in} * 6 \text{ hilos/in}}{10 \text{ s}} = 8.4 \text{ hz} = 504 \text{ rpm} \quad \text{Ecuación 19}$$

Esta es la velocidad a la que debe girar ya acoplado al tornillo, no libre.

Debido al bajo costo, facilidad de control y alta disponibilidad se decidió que el motor fuera del tipo DC con escobillas con voltaje nominal entre 12V y 24V. Este requisito eliminó la opción de acoplar al motor directamente al tornillo (sin pasar por una reducción) ya que los motores que cumplen con los requisitos de velocidad y torque son de tipo AC. Debido a esto se buscó una caja reductora que bajara la velocidad y aumentara el torque a los valores requeridos (504 rpm y 5 lb in).

Se analizaron 4 motores que se venden con cajas reductoras diseñadas especialmente: motor FR801-001 con caja “Toughbox Mini 9:1”, motor M5-RS540-12 con caja P60 26:1, motor M5-RS550-12 con caja P60 26:1 y el M5-RS775-12 con caja P60 16:1. Para elegir entre ellos se analizó la corriente, la velocidad y la eficiencia a la que operaría en el túnel. A continuación se muestran los datos que provee el fabricante de cada motor.

Cuadro 3 Características de los motores evaluados

<i>Motor</i>	<i>RPM sin carga</i>	<i>Corriente sin carga</i> (A)	<i>Torque Max</i> (mNm)	<i>Corriente Max</i> (A)	<i>Kt</i>	<i>Kv</i>
M5-RS775-12	7300	1.1	431.5	30	14.9	608.3
M5-RS550-12	19300	1.4	486.2	85	5.8	1608.3
M5-RS540-12	16800	1	278.8	42	6.8	1400.0
FR801-001	5310	2.7	2373	133	18.2	442.5

Con los datos anteriores y las ecuaciones Ecuación 1 a Ecuación 4 se calculó el torque que se demandaría de cada motor, la corriente, la velocidad de salida después de la caja reductora, el tiempo que le tomaría en abrir completamente el túnel y la eficiencia a la que trabajaría. La siguiente tabla resume estos datos. En el Anexo 0 se muestran los cálculos para el motor elegido.

Cuadro 4 Características de los motores a un torque de 5 lb in

<i>Motor</i>	<i>Reducción</i>	<i>Torque del motor</i> (mNm)	<i>Corriente</i> (A)	<i>RPM Salida</i>	<i>Eficiencia</i>	<i>Tiempo</i> (s)
FR801-001	9	62.77	6.15	574	46.1%	8.77
M5-RS550-12	26	21.73	5.14	709	68.1%	7.10
M5-RS540-12	26	21.73	4.20	596	70.0%	8.45
M5-RS775-12	16	21.73	3.47	419	59.6%	12.03

El motor FR801-001 se descartó por su alta corriente y baja eficiencia. De los tres restantes el M5-RS540-12 es el que provee la mejor combinación entre consumo de corriente, velocidad y eficiencia. Es el segundo con menor consumo de corriente detrás

del M5-RS775-12 pero provee mayor velocidad y eficiencia. Es también segundo en velocidad detrás del M5-RS550-12 pero con menor consumo de corriente. Por estas razones se seleccionó el motor M5-RS540-12 con una caja reductora P60 26:1 ambos de la marca Banebots. La caja reductora es de tipo planetaria con dos etapas y una relación total de 26:1. Los engranes son de acero. En el Anexo 0 se encuentran los planos de la caja reductora.

Figura 14 Motor M5-RS540-12



Figura 15 Caja reductora P60 26:1



D. Driver y fuente de alimentación

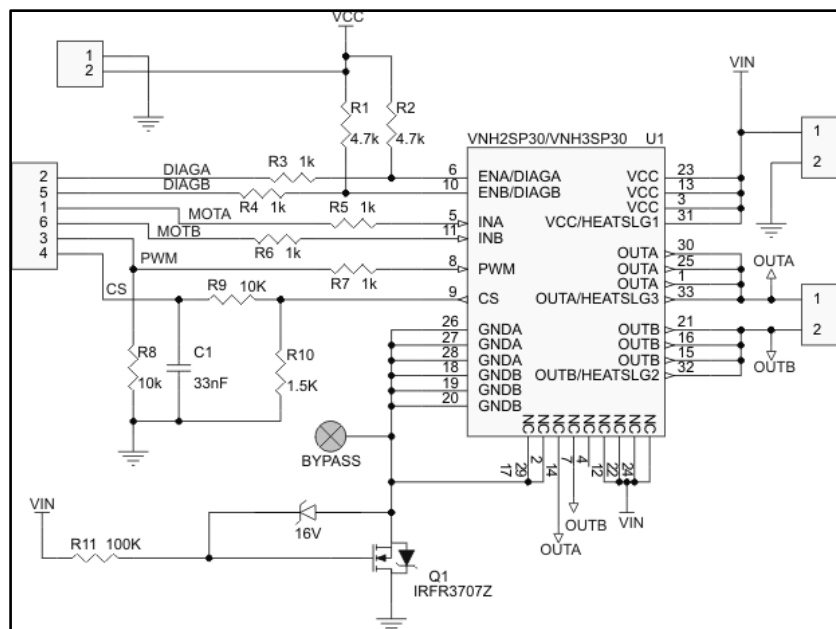
El microcontrolador no puede proveer la potencia necesaria para mover al motor y por esto se necesita un driver. Dado que el motor debe poder girar en ambas direcciones y variar su velocidad el circuito adecuado es el puente H. De los cálculos anteriores (Cuadro 4), el motor consume un aproximado de 4.2A a máxima velocidad, con el torque medido con el dinamómetro (5 lb in). El driver se seleccionó considerando un amperaje mayor, 10A, puesto que la demanda de corriente podría aumentar si por alguna razón aumenta la carga y para que en condiciones normales el driver no esté operando cerca de su límite.

El driver seleccionado fue la placa “MD01B”, del fabricante Pololu, que contiene el puente H “VNH2SP30” y todo el circuito externo que requiere este integrado. Este driver soporta hasta 30A continuos si se utiliza un disipador adecuado. Sin un disipador, como está en la placa, soporta hasta 14A continuos sin sobrecalentarse por lo que es adecuado para el motor elegido y no se necesitó instalar un disipador. A continuación se muestran las especificaciones del driver.

Cuadro 5 Características del driver MD01B

<i>Parámetro</i>	<i>Valor</i>
Voltaje de operación	5.5 – 16 V
Frecuencia de PWM máxima	20 kHz
Voltaje para medición de corriente	0.13 V/A
Voltaje máximo	16 V min. / 19 V typ.
Voltaje de uno lógico	3.25 V min.
Tiempo para sobrecalentamiento a 20 A	35 s
Tiempo para sobrecalentamiento a 15 A	150 s
Corriente máxima sin sobrecalentamiento	14 A

Figura 16 Diagrama del driver MD01B



Para alimentar el motor es necesaria una fuente de alimentación de potencia suficiente. Se calculó que el motor consumirá 4.2A a máxima velocidad. Puesto que al arrancar el motor consume más de esta corriente y la fuente también alimentaría otros circuitos se buscó una fuente que entregara al menos 15A. La fuente seleccionada es la TDK Lambda LS200-12 que provee 12V y hasta 16.7A .

E. Mecanismo de acople

Puesto que a la salida de la caja reductora la velocidad y torque ya son los requeridos, no se necesita de ninguna otra etapa reductora o multiplicadora. Entonces, solamente se necesitó de un elemento de acople para transmitir directamente el movimiento al tornillo. El tornillo posee una sección sin rosca de 3/8 in de diámetro en un extremo. Es en este extremo en el que se realizará el acople. La caja reductora, por su parte, tiene a su salida un eje de 1/2 in de diámetro con un cuñero para una cuña cuadrada de 1/8 in.

Los requisitos para la elección del acople únicamente eran que fuese capaz de transmitir al menos 5 lb in y que acepte ejes de los diámetros de la caja y el tornillo. Debido a que no se puede asegurar una alineación precisa se descartó el utilizar acoples rígidos y se decidió por un acople flexible tipo araña. Se verificó en el catálogo del fabricante Lovejoy y el acople más pequeño que admite el eje de $\frac{1}{2}$ in tiene un torque máximo de 26 lb in por lo que cubre sobradamente el requisito de la aplicación. Este es el acople L050.

Figura 17 Acople Lovejoy L050



Como se mencionó el eje de la caja reductora posee un cuñero por lo que se utilizó una cuña para la unión con el acople. En el tornillo se decidió realizar la unión solamente por medio de un prisionero que entraría en un agujero taladrado. Se calculó la profundidad mínima que debe tener este agujero. El cálculo se hizo evaluando las fuerzas de compresión en el tornillo pues es el elemento con menor resistencia a la fluencia. El prisionero tiene un diámetro de $\frac{1}{4}$ in por lo que según la Ecuación 11 y la Ecuación 12 la profundidad mínima es 0.04 in.

$$\sigma = \frac{F}{A} = \frac{\frac{T}{D/2}}{(D_p P)} = \frac{2T}{DD_p P}$$

Ecuación 20

$$P = \frac{2TN}{DD_p S_y} = \frac{(2)(5 \text{ lb in})(3)}{\left(\frac{3}{8} \text{ in}\right) \left(\frac{1}{4} \text{ in}\right) (8 \text{ ksi})} = 0.04 \text{ in}$$

Ecuación 21

Se necesitó fabricar dos piezas más e instalar un rodamiento para completar el mecanismo. La primera de estas piezas es un espaciador que se instala entre el perfil y el acople flexible. Este espaciador se fabricó con una impresora de plástico y su función es prevenir que, al girar, el tornillo se desenrosque alejándose del motor en lugar de que la tuerca avance. Solamente es necesario del lado del motor pues del otro lado existe un buje plástico que detiene al tornillo y sirve de tope ya que el tornillo cambia a un diámetro mayor. Se instaló un rodamiento axial para prevenir que el espaciador friccionara contra el perfil metálico. Debido a que en el medio solamente están disponibles rodamientos axiales métricos se decidió utilizar uno de 10mm de diámetro interno y se fabricó una pieza impresa en plástico que centra el rodamiento en el eje de 3/8in. El rodamiento es un 51100. Las figuras siguientes muestran las piezas fabricadas. Los planos de las piezas fabricadas se encuentran en el Anexo 0.

Figura 18 Espaciador fabricado en impresora de plástico



Figura 19 Pieza para centrar cojinete fabricada en impresora de plástico



F. Estructura de sujeción y protección

Por la ubicación del tronillo de potencia y las dimensiones de los componentes, surgió la necesidad de crear una estructura que sujete al motor y la caja reductora en su lugar. Esta estructura también cumple con el objetivo de proteger al usuario de las partes móviles del sistema dificultando el entrar en contacto con las mismas. La Figura 20 muestra la ubicación donde se decidió colocar el motor.

Figura 20 Parte del túnel donde se encuentra el tornillo que debe girar el motor



Después de evaluar varias ideas de diseño, se decidió por una estructura de tres partes que se atornilla a la caja reductora en los agujeros roscados que tiene a cada lado y que estuviese atornillada al perfil angular que posee la estructura del túnel (véase Figura 20). Se decidió, por facilidad de manufactura dentro de la universidad, que fuese de lámina negra de $\frac{1}{4}$ in de espesor y que las uniones fueran soldadas. Los agujeros para los tornillos se taladraron de 50 milésimas de pulgada de diámetro más para que fuese posible alinear los componentes y compensar errores de manufactura. A los costados se colocó acrílico de $\frac{1}{8}$ in de espesor que cumple con la función de proteger las partes móviles y permitir la visibilidad de los elementos para facilidad de inspección. Puesto

que esta pieza no está sometida a fuerzas significativas no se realizó un análisis de esfuerzos. La Figura 21 muestra la pieza terminada. En el Anexo 0 se encuentran los planos de la pieza.

Figura 21 Pieza fabricada para sujetar motor y caja

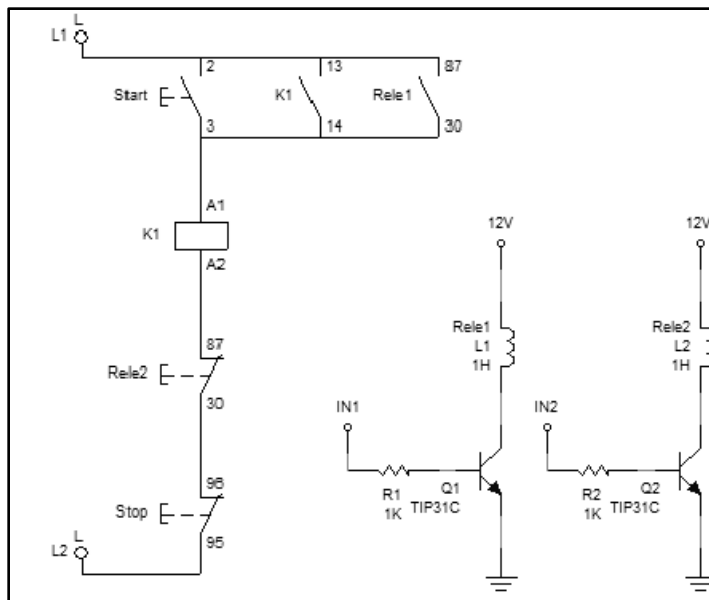


G. Elementos finales del diseño físico

Para completar el diseño físico se necesitó de algunos otros elementos. El primero son límites de carrera para indicar cuando el ventilador se encuentra en uno de los extremos y no se debe intentar mover más allá. De lo contrario el motor y las partes mecánicas se podrían dañar. Para este fin se instaló un microswitch en cada extremo.

Para poder encender desde la computadora el túnel de viento se diseñó un circuito eléctrico. El túnel ya contaba con un contactor que era accionado manualmente con pulsadores. Para que se pudiera encender y apagar desde la computadora se añadieron dos relés manteniendo funcionales los pulsadores. El circuito se muestra en. Los relés son de 12VDC y su bobina consume 0.14A. Para activarlos desde el microcontrolador se necesitó de un driver. El circuito se muestra en la **¡Error! No se encuentra el origen de a referencia..**

Figura 22 Circuito para activar contactor desde PIC por medio de relés



Para el modo manual se fabricó un encoder para que fuera posible mover el ventilador a pasos. Se imprimió en plástico una pieza que tiene cuatro secciones para utilizarla en conjunto con un fotointerruptor de ranura (Figura 23). Así se pueden contar los flancos de subida para que desde la computadora se pueda pedir que el tornillo gire una vuelta o un cuarto de vuelta.

Figura 23 Encoder



El elemento final es una protección contra ruido eléctrico. Para prevenir que entre ruido al microcontrolador se incluyeron acopladores ópticos en las salidas del microcontrolador hacia los relés y hacia el motor. De esta manera se puede utilizar una batería para alimentar al microcontrolador y utilizar otra fuente para el motor y los relés. Así se garantiza que no entrará ruido al sistema.

H. Programación del microcontrolador

El microcontrolador que se utilizó fue el PIC18F45K22. Este microcontrolador posee todos los módulos necesarios para el control del motor y la comunicación. La tabla muestra los módulos que fueron utilizados y cuál es su función. Originalmente se inició utilizando el PIC16F887, pero luego se hizo el cambio debido a dos ventajas que posee el PIC18F45K22. La primera es que tiene un módulo USART más. Para realizar pruebas se estaba utilizando ya un módulo USART y se necesitaba uno más para la retroalimentación de la velocidad. La segunda ventaja es que este microcontrolador tiene más niveles en el stack de direcciones lo cual permite tener un programa con más funciones anidadas.

Cuadro 6 Utilización de los módulos del microcontrolador

<i>Módulo</i>	<i>Función</i>
I2C1	Comunicación con módulo maestro
CCP1	PWM
CCP2	Detección de flancos en el encoder
TIMER0	Utilizado para resolver rebote de microswitches
TIMER1	Utilizado en conjunto con CCP2
UART1	Realización de pruebas, mismas funciones que I2C
UART2	Retroalimentación de velocidad de viento
PORTB IOC	Utilizado para detectar que se presionó un microswitch

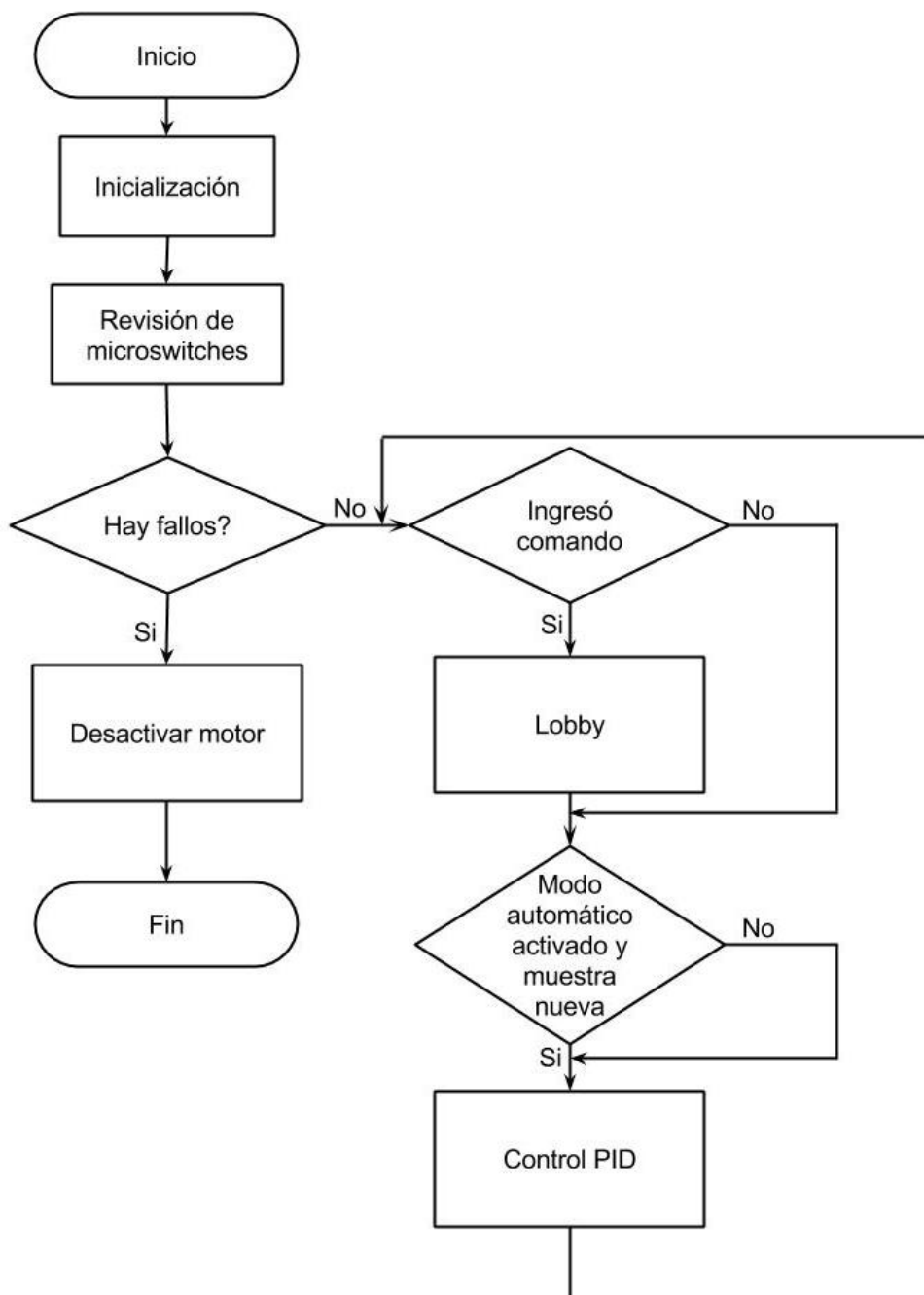
La Figura 24 muestra un diagrama de flujo simplificado del programa del microcontrolador. En el Anexo 0 se encuentra el código completo. La programación del microcontrolador se realizó en lenguaje C con el compilador mikroC. El bloque “Lobby” es la función que procesa los comandos que envía la computadora. Esta función realiza el procedimiento o llama a otras funciones para llevar a cabo la instrucción recibida. El bloque “Automático” contiene el algoritmo que establece la velocidad del viento de manera automática. Este algoritmo será explicado en otro apartado.

En la Figura 24 no se muestra el servicio de interrupciones pero este realiza varias funciones importantes. Se utilizan interrupciones para el conteo de las vueltas del motor (utilizado en modo manual), detección de cambios en los microswitches y recepción de datos vía I2C y USART.

1. Control del motor. El microcontrolador debía generar las señales para el control del motor. El driver seleccionado tiene tres entradas. INA e INB determinan la dirección de giro del motor. La tercera entrada es de PWM para regular la velocidad de giro.

El microcontrolador posee dos módulos de CCP (*Capture/Compare/PWM*). Ambos módulos fueron utilizados para el control del motor. El módulo CCP1 fue utilizado en modo de PWM. El módulo fue utilizado con una resolución del ciclo de trabajo de 8 bits. La frecuencia del PWM es un parámetro importante que está en función al motor y la aplicación. Se experimentó con varias frecuencias de PWM para obtener un comportamiento satisfactorio. Se seleccionó una frecuencia de 500hz porque esta produjo una relación lineal entre porcentaje de PWM y velocidad.

Figura 24 Diagrama de flujo del programa



Ya sea en modo manual o automático, se podría dar el caso en el que se desee cambiar la velocidad del motor súbitamente, por ejemplo, de un paro total a máxima velocidad. Esto sería perjudicial para los componentes mecánicos del sistema en especial la caja reductora. Por lo tanto se limitó el cambio máximo de PWM que se puede realizar de un momento a otro. Así para ir del paro a máxima velocidad se toma alrededor de 2 segundos. Con esto se logra una aceleración más suave.

El otro módulo CCP se utilizó en modo capture. El propósito es contar los flancos de subida que se producen cuando el encoder pasa por el fotointerruptor de ranura. El objetivo de esto es que en el modo manual se pueda hacer que el motor de un cuarto de vuelta o una vuelta completa. El módulo CCP2 se utiliza normalmente en conjunto con el TIMER2 para medir el tiempo entre flancos. Esta funcionalidad sólo se utilizó para realizar las pruebas de velocidad del motor. No es utilizada en el programa final.

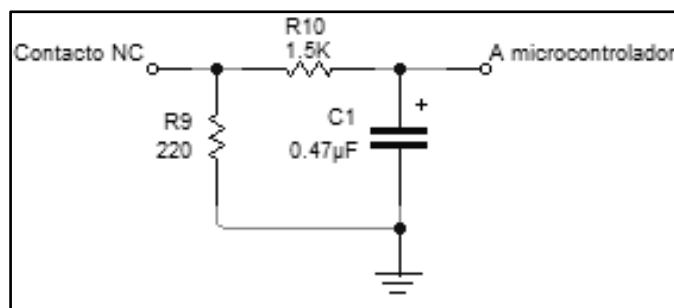
2. Microswitches. Los microswitches son una parte importante en el diseño porque son la única manera que tiene el microcontrolador de saber que se ha llegado a un límite y que no se debe mover el ventilador en cierta dirección. Al ser presionados el motor debe ser parado y mientras está presionado no se debe mover en la dirección que está el switch.

Cada microswitch tiene un contacto normalmente cerrado y un normalmente abierto. La detección de la activación del microswitch se realizó sobre el contacto normalmente cerrado para que en caso se desconecte el microswitch se tome como que está presionado y no se mueva el ventilador en ese sentido. También se utilizó el contacto normalmente abierto para que al iniciar el programa se pueda revisar si están conectados los microswitches. Si alguno no está conectado ambos pines del microcontrolador tendrán 0V lo cual sólo sucede en este caso.

La detección de la activación de los microswitches se decidió hacer detectando los flancos de bajada por interrupción para tener el menor retraso posible entre que se presiona el switch y se para el motor. Los contactos normalmente cerrados de los microswitches se conectaron a las entradas RB4 y RB5 que poseen *Interrupt On-Change (IOCB)*.

Al implementar esto se encontraron problemas con el rebote del switch. Esto es normal y lo que se hace comúnmente es que se desactiva la detección de cambios por cierto tiempo después de una primera detección. Se decidió realizar esto utilizando el TMR0 para que en los 10ms después del primer flanco no se detecte ningún otro. Esto resolvió el problema de detectar los cambios durante rebote pero en algunas ocasiones después del cambio el microcontrolador leía incorrectamente el estado de la entrada porque se producía un pico de muy corta duración y cuando el microcontrolador leía el estado el pico ya había pasado. La solución que se ideó fue filtrar la señal del microswitch para que este primer pico fuese eliminado. El diagrama del filtro se observa en la Figura 25. Este es un filtro pasivo de primer orden con frecuencia de corte de 226Hz y una constante de tiempo de 0.7ms.

Figura 25 Filtro para la señal del microswitch



3. Comunicación con módulo maestro. Como se mencionó en la sección anterior, el microcontrolador recibe comandos desde el módulo central. Se definió que la comunicación con el módulo maestro sería por I2C debido a que puede haber varios esclavos que serían los demás módulos del proyecto.

Se definió que los comandos fuesen de dos bytes. El primero indica la acción a realizar y el segundo es utilizado como parámetro para algunos comandos. El Cuadro 7 muestra los comandos que se utilizaron en la comunicación con el módulo maestro. Si se le envía un comando con un valor en el primer byte que no es parte de los comandos lo único que ocurrirá es que, de estar activado, se desactivará el modo automático. La dirección del módulo para el bus I2C es 0x50.

Cuadro 7 Listado de comandos que acepta el módulo

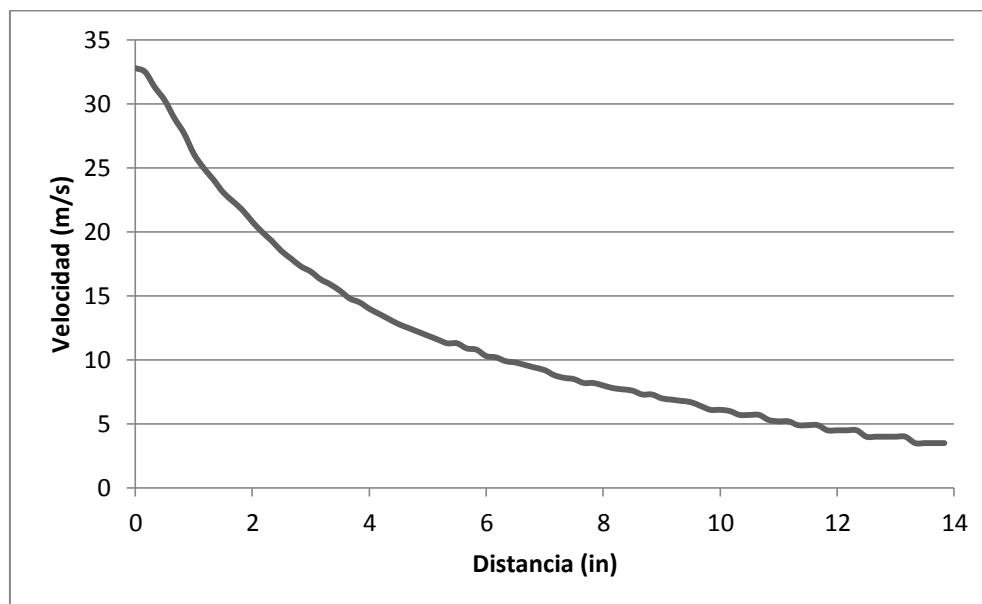
<i>Acción a realizar</i>	<i>Primer byte</i>	<i>Segundo byte</i>
Activar modo automático (PID)	‘p’	Velocidad de referencia (5-28)
Moverse de manera continua hacia adelante	‘a’	% de velocidad (0-100)
Moverse de manera continua hacia atrás	‘d’	% de velocidad (0-100)
Parar	‘f’	No importa
Paso corto hacia adelante (1/4 de vuelta)	‘-’	No importa
Paso corto hacia atrás (1/4 de vuelta)	‘+’	No importa
Paso largo hacia adelante (1/4 de vuelta)	‘n’	No importa
Paso largo hacia atrás (1/4 de vuelta)	‘m’	No importa
Encender ventilador	‘e’	No importa
Apagar ventilador	‘o’	No importa
Cambiar constante k de fórmula	‘k’	k x 100

I. Algoritmo de control automático

Uno de los objetivos de este trabajo de graduación era crear un algoritmo o procedimiento para que el usuario pudiese indicar una velocidad de viento deseada y que el sistema la estableciera.

Al inicio se consideró trabajar un sistema de lazo abierto. Obteniendo la relación entre la distancia de apertura del túnel y la velocidad del viento se podría calcular la distancia requerida para una velocidad dada y colocar el ventilador en esa posición. El microcontrolador llevaría un conteo del número de vueltas que ha realizado el tornillo y así se conocería la distancia. La Figura 26 muestra el resultado de las mediciones que se hicieron para obtener la ecuación entre velocidad y distancia.

Figura 26 Relación entre velocidad de viento y distancia de apertura del túnel



Este método fue descartado rápidamente porque cualquier cambio en el sistema haría inválida la ecuación y el algoritmo fallaría. Por esta razón se decidió por un sistema de lazo cerrado del tipo PID. Se tendría una retroalimentación de la velocidad del viento y

en base al error se genera una señal de PWM que movería el ventilador a la posición necesaria para llevar el error a cero.

1. Medición de la velocidad del viento. Para implementar un control de lazo cerrado es necesario medir la salida, en este caso la velocidad del viento. Antes de detallar la implementación del control PID se explicará cómo se obtiene la velocidad del viento.

Como se ha señalado la medición de la velocidad no es parte de este módulo (véase anexo A para mayor información de los otros módulos del proyecto). El sensor (DH008) que utiliza el módulo correspondiente se comunica vía Modbus RTU con un microcontrolador a través de un convertidor RS485 a RS232. Lo que se decidió hacer es conectarse a la línea RX del microcontrolador para recibir los datos que envía el sensor. La Figura 27 ilustra esta conexión. El sensor envía una secuencia fija de bytes. El primer byte es siempre 6 (dirección del sensor), el segundo es 3 (lectura), el tercero es siempre 2 (significa que la transmisión será de dos bytes) y en los siguientes dos bytes se encuentra el valor de la medición. Los datos se reciben por interrupción y se lleva control de los datos que se reciben para guardar los siguientes dos bytes después de que se haya recibido 6-3-2. De esta manera se asegura que se toman correctamente los datos. El tiempo entre muestras es de 203ms. Este tiempo es debido a cómo se programó el microcontrolador que hace la toma de datos. No tendría ventaja reducir este tiempo ya que el sensor tiene un tiempo de respuesta de 250ms.

El sensor realmente no mide velocidad de viento sino presión diferencial. El sensor mide la diferencia entre la presión de estancamiento y la presión estática. La relación entre esta diferencia de presiones y la velocidad está dada por la siguiente fórmula.

$$V = \sqrt{\frac{2\Delta P}{\rho}}$$

Ecuación 22

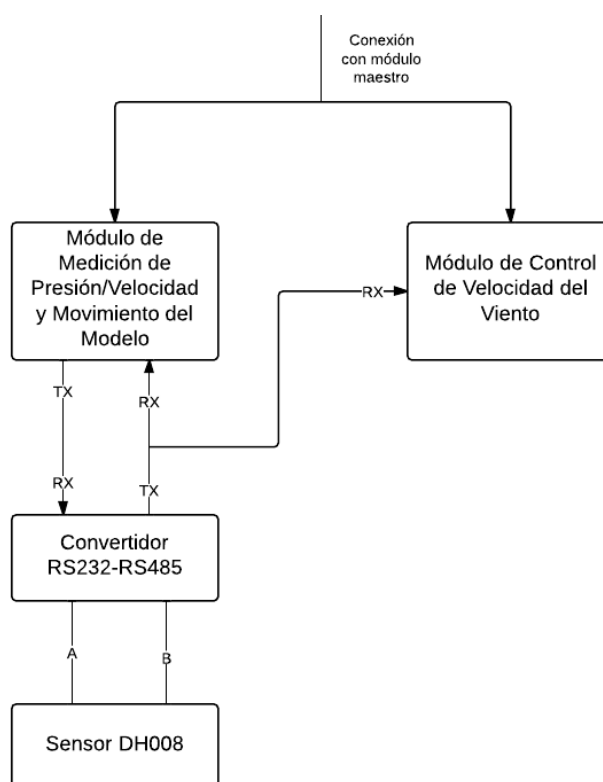
La fórmula que realmente se utiliza en el código tiene la siguiente forma:

$$V = 1.1743k\sqrt{\Delta P}$$

Ecuación 23

El valor 1.1743 incluye la conversión de psi (el sensor mide en psi) y la raíz de 2. El factor k es la raíz del inverso de la densidad. Este factor puede ser cambiado desde la computadora con un comando. El factor se guarda en la memoria EEPROM y se carga a una variable al iniciar el modo automático.

Figura 27 Detalle de la conexión con el módulo de medición de velocidad



2. Implementación del controlador PID. A continuación se explicará la implementación del control PID. La Figura 28 muestra de forma general la secuencia de pasos que sigue el algoritmo. El algoritmo se activa cuando se recibe el comando de

activar el modo automático y la velocidad deseada. También se revisa que el ventilador del túnel esté encendido.

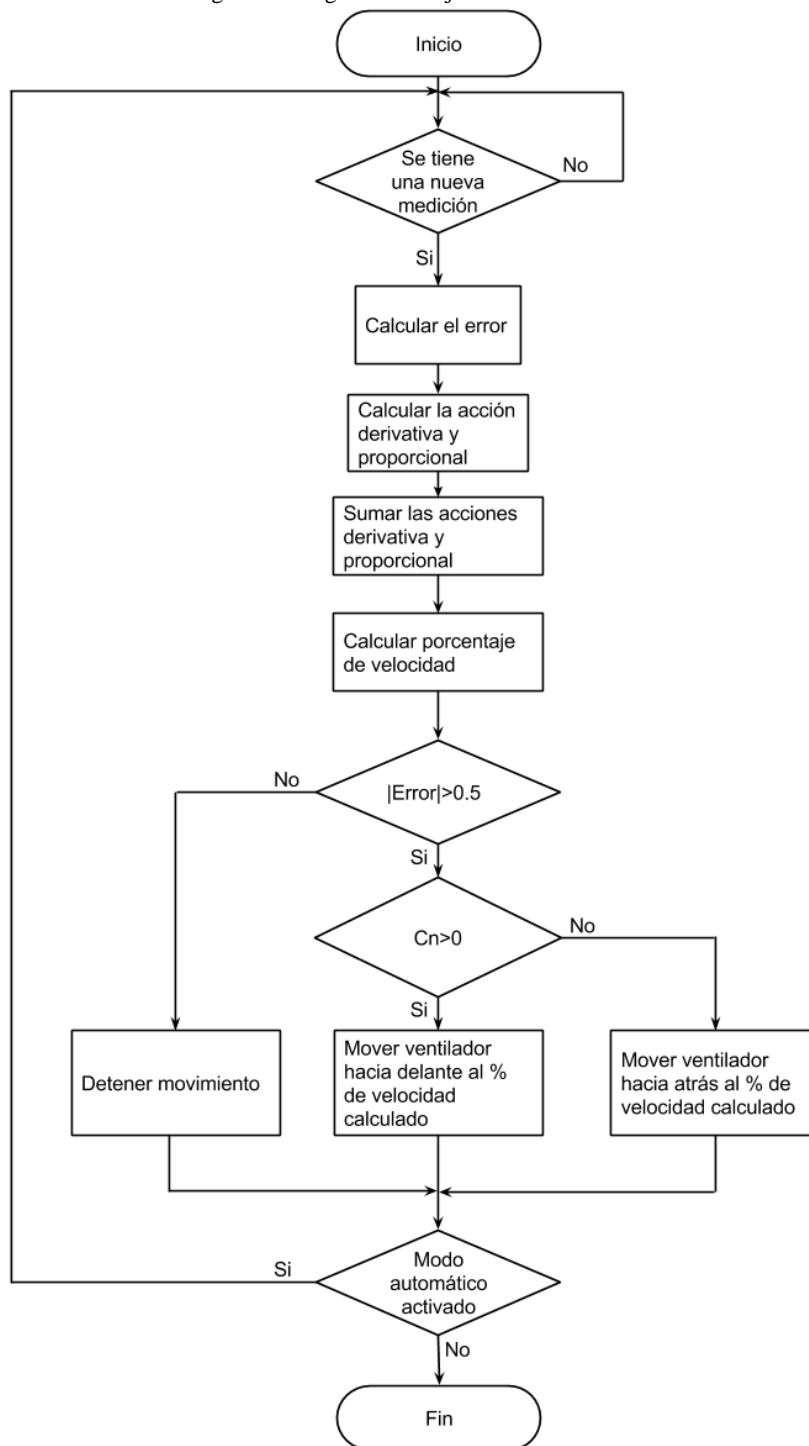
El algoritmo inicia cuando se obtiene una nueva lectura de la velocidad del viento. Esto ocurre por interrupción como se detalló en la sección anterior. Con este valor se calcula el error respecto del valor deseado. No es necesario muestrear una señal de referencia pues el valor deseado se envía desde la computadora solamente una vez cuando el usuario lo cambia.

A continuación se calculan las acciones proporcional y derivativa. No fue necesario utilizar la acción integral pues, como se verá en los resultados, el sistema no tiene error en estado estacionario aún sin la acción integral. Puesto que en esta aplicación la referencia cambiará súbitamente y se mantendrá constante después de los cambios, se decidió realizar la acción derivativa sobre la velocidad y no sobre el error. Así se evita el *derivative kick* (véase sección IV.G.2). La acción proporcional si se calcula sobre el error.

Con la suma de las acciones (mostrado como C_n en el diagrama) se calcula el porcentaje de velocidad que debe aplicarse en el tornillo. Este simplemente es el valor redondeado del valor absoluto de la suma. Se decidió hacerlo de esta manera para que no hubiera otro parámetro que afecte el resultado además de la ganancia de cada acción.

Si el error es mayor a 0.5 m/s (o menor a -0.5 m/s) se revisa el signo de la suma de las acciones (C_n) para decidir en qué dirección se debe mover el tornillo. Con la dirección y el porcentaje de velocidad se cambian las señales de dirección que van al driver y el ciclo de trabajo del PWM. Si el error está entre -0.5 y 0.5 se para el movimiento. Se decidió hacer de esta manera para evitar que el sistema trate de corregir errores pequeños y que el motor se esté moviendo constantemente. Este error es aceptable en esta aplicación, especialmente porque la velocidad no será totalmente constante aún a un posición del ventilador fija.

Figura 28 Diagrama de flujo del controlador PID



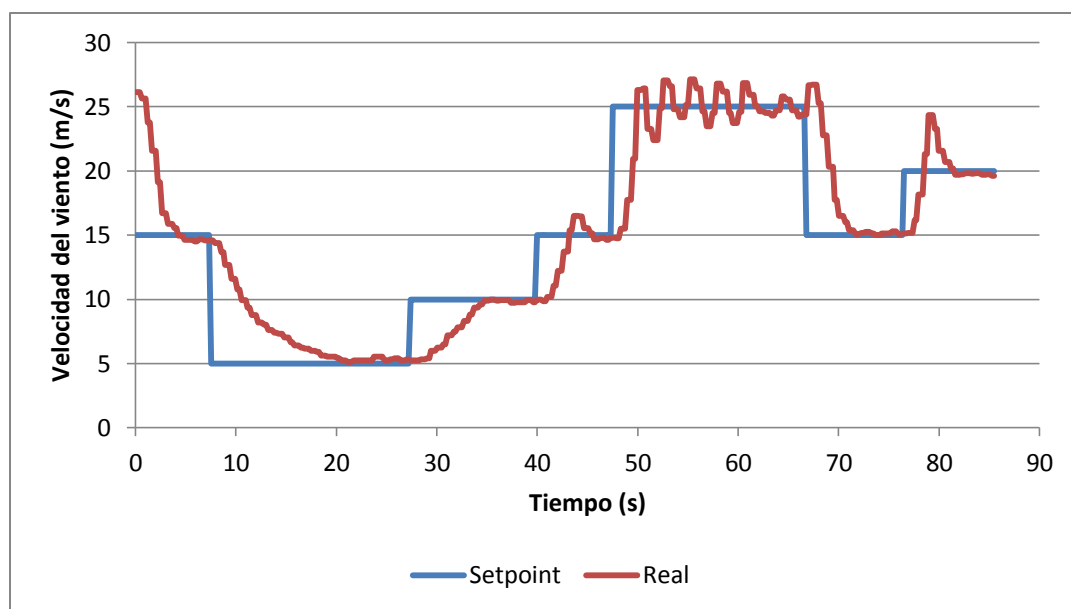
El procedimiento se repite cada vez que se obtiene una nueva lectura de la velocidad del viento. El modo automático se desactiva cuando se recibe un nuevo comando sin importar cuál sea.

3. Ajuste de las ganancias del controlador. La ganancia de cada acción del controlador PID afecta la respuesta que tendrá el sistema. El Cuadro 1 muestra el efecto que tiene el incrementar cada ganancia. Se debía encontrar una combinación de ganancias que resultara en un comportamiento adecuado del sistema. Los requisitos que se establecieron para el sistema son los siguientes:

- Error en estado estable de ± 0.5 m/s.
- Ninguna o muy poca oscilación.
- Tiempo de asentamiento razonable. No es crítico.

Puesto que no se posee un modelo matemático del sistema y obtenerlo tomaría demasiado tiempo las ganancias se encontraron a base de prueba y error. Al realizar esto se halló una complicación y es que como se puede observar en la Figura 26 el sistema no es lineal. Al mover el ventilador cierta distancia el cambio en la velocidad no es el mismo en todo el rango. Esto complica el ajuste del sistema de control ya que, como se observa en la Figura 29, el sistema se comporta de manera diferente según el setpoint. Para velocidades bajas el sistema es estable y tiene un tiempo de asentamiento bastante grande, pero para velocidades altas el sistema oscila.

Figura 29 Prueba del sistema de control variando el setpoint

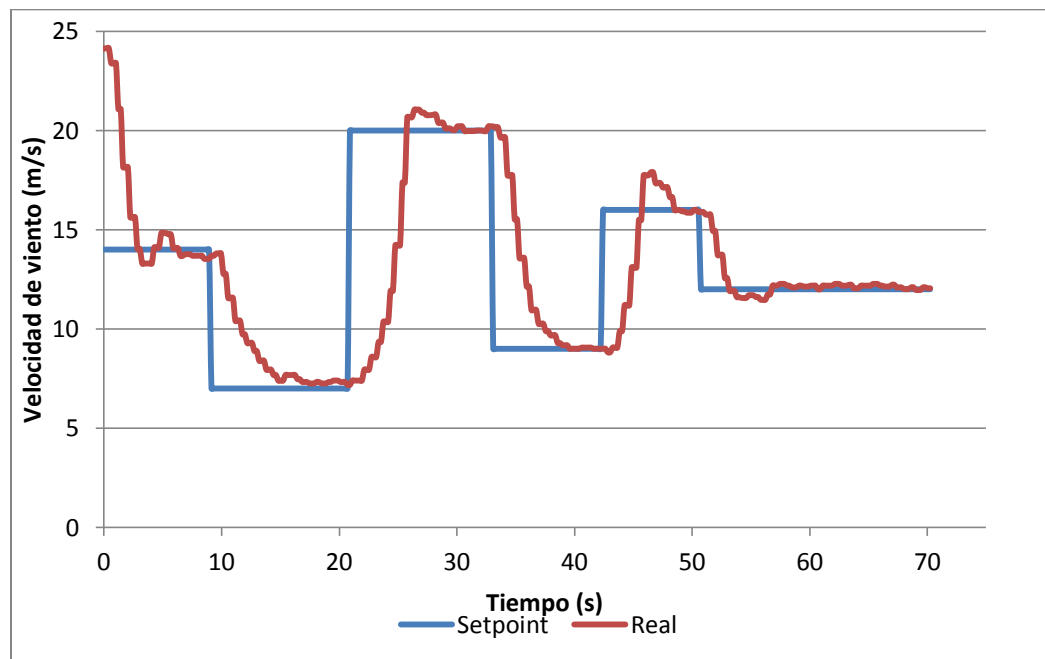


Por la aplicación era aceptable que el comportamiento no fuese igual en todo el rango de operación. Se evaluó la opción de ajustar el controlador para que el sistema fuese estable en todo el rango sin importar que el tiempo de asentamiento variara. Sin embargo, la solución que se eligió para resolver el problema fue dividir el rango en dos partes y tener un set de ganancias distinto para cada uno. Con este método se obtuvo buenos resultados. El Cuadro 8 muestra cómo se dividió el rango de operación y las ganancias que se fijaron para cada parte. La Figura 30 muestra el resultado. Como se observa el sistema es estable sin importar el setpoint, sólo hay una pequeña sobrelevación y el tiempo de asentamiento es aceptable.

Cuadro 8 Ganancias de las acciones del controlador PD

Condición	K_p	K_d
Setpoint y velocidad actual mayores a 15m/s	3	6
Setpoint o velocidad actual menores a 15m/s	12	3

Figura 30 Prueba del sistema de control con varios setpoints utilizando dos rangos



VII. RESULTADOS

A. Construcción del sistema

A continuación se muestran fotos del sistema construido. La Figura 31 muestra el motor con la caja reductora, el encoder y el acople. La Figura 32 muestra el lado del tornillo: el acople, espaciador y el rodamiento con el centrador. La Figura 33 muestra el sistema ensamblado. En el Anexo C se encuentran los planos de todo el sistema. La Figura 34 muestra los microswitches. En la Figura 35 se observan los relés que fueron utilizados para encender y apagar el túnel desde la computadora instalados en la caja del contactor.

Figura 31 Motor, caja, encoder y acople ensamblados

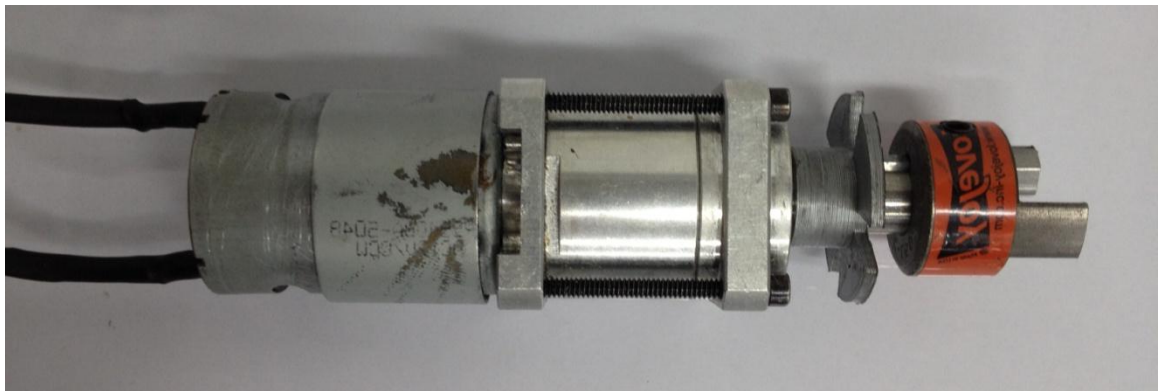


Figura 32 Acople, espaciador, pieza para centrar y rodamiento ensamblados en el tornillo



Figura 33 Sistema completamente ensamblado

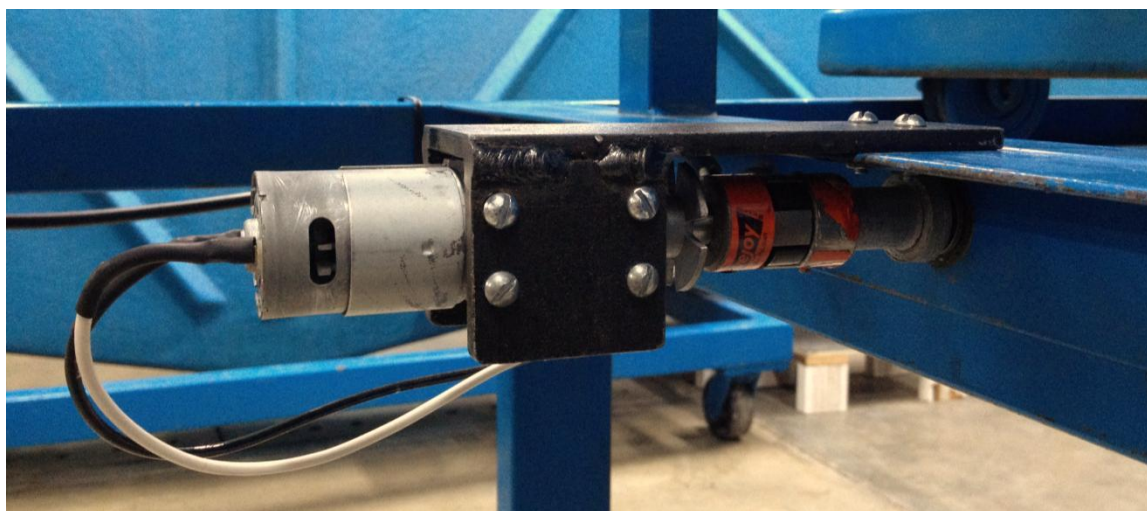


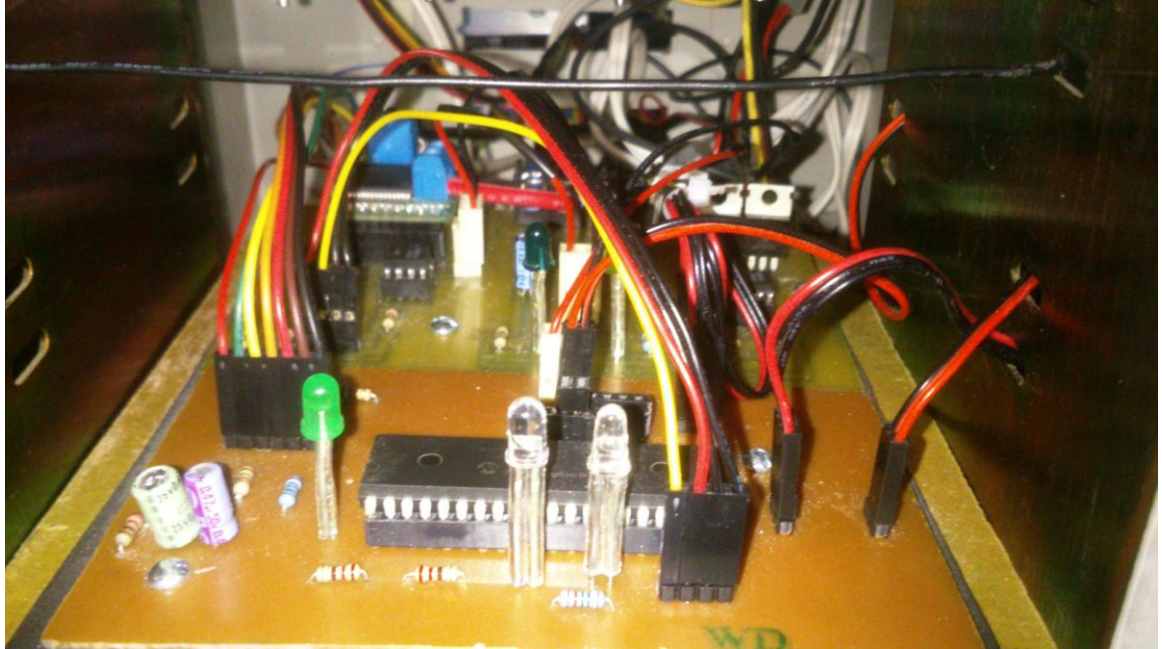
Figura 34 Microswitches instalados



Figura 35 Relés para controlar bobina del contactor del ventilador



Figura 36 Circuito terminado en PCB



B. Eficiencia del sistema

La eficiencia de los motores eléctricos varía de acuerdo a la velocidad y torque. Como punto de referencia se toma la eficiencia a máxima velocidad ya que este fue el punto que se utilizó para elegir el motor. Se midió la fuerza necesaria para mover el ventilador y resultó ser de 42 lb. El tiempo que le toma al sistema moverlo de un punto a otro es de 9s y el consumo de corriente se mantuvo alrededor de 4.5 A. La eficiencia se calcula de la siguiente manera.

$$P_{salida} = F * v = (42 \text{ lb} * 4.45 \text{ N/lb}) \left(\frac{14 \text{ in} * 0.0254 \text{ m/in}}{9 \text{ s}} \right) \quad \text{Ecuación 24}$$

$$= 7.39 \text{ W}$$

$$e_{total} = \frac{P_{salida}}{P_{entrada}} = \frac{7.39W}{(12V)(4.3A)} = 14.3\% \quad \text{Ecuación 25}$$

La eficiencia resulta ser realmente baja pero es de esperar puesto que la eficiencia de los tornillos de potencia es baja (véase sección IV.E). La eficiencia del tornillo se calcula según la Ecuación 8 .

$$e_{tornillo} = \frac{FL}{2\pi T} = \frac{(42 lb)(\frac{1}{6}in)}{2\pi(5 lb \cdot in)} = 22.28\% \quad \text{Ecuación 26}$$

La eficiencia total del sistema se puede calcular multiplicando la eficiencia de cada parte. Si sólo se considera el motor (véase Cuadro 4) y el tornillo la eficiencia resulta ser:

$$e_{motor-tornillo} = e_{motor} * e_{tornillo} = (70\%)(22.28\%)=15.60\% \quad \text{Ecuación 27}$$

Este valor es cercano a la eficiencia calculada en la Ecuación 25. La caja reductora, el acople y el rodamiento disminuyen la eficiencia del sistema. Dividiendo Ecuación 25 entre Ecuación 27 :

$$e_{otros} = \frac{14.3\%}{15.6\%} = 91.67\% \quad \text{Ecuación 28}$$

Esta es una eficiencia razonable para la combinación de caja reductora, acople y rodamiento. Es entonces entendible que la eficiencia total sea de sólo 14.3%. Mejorar la eficiencia sustancialmente implicaría cambiar completamente el sistema puesto que, debido al tornillo de potencia, aunque el motor y los demás componentes tuviesen una eficiencia del 100% el sistema no podría tener una eficiencia mayor a 22.28%.

C. Listado de partes y costos

La siguiente tabla detalla las partes del sistema y su costo. No se incluye el costo de envío de algunas partes que fueron compradas en el extranjero. Nótese que las piezas plásticas y la pieza de sujeción tienen costo cero pues tanto el material como la mano de obra fueron provistos por la universidad.

Cuadro 9 Partes y costos del sistema diseñado

<i>Descripción</i>	<i>Costo</i>
Motor Banebots RS-540	Q 50.00
Caja Reductora P60 26:1	Q 436.00
Driver Pololu MD01B	Q 279.60
Acople Lovejoy L050	Q 100.00
Rodamiento 51100	Q 50.00
Piezas plásticas impresas	Q 0.00
Pieza metálica de sujeción	Q 0.00
Fuente de Voltaje LS200	Q 466.64
Microswitches	Q 15.00
Microcontrolador PIC18F45K22	Q 50.00
Componentes electrónicos y eléctricos varios	Q 90.00
Tornillería	Q 20.00
Relés	Q 48.00
<i>Total</i>	Q 1605.24

D. Especificaciones

A continuación se muestran las especificaciones del sistema. El rango de velocidades se debe a las limitaciones físicas del túnel. Es decir, a la máxima distancia que puede estar el ventilador la velocidad está entre 1 m/s y 2 m/s y completamente cerrado está entre 27 m/s y 28 m/s. La resolución y el error máximo se definieron en base a las necesidades que tendría un usuario al usar el sistema.

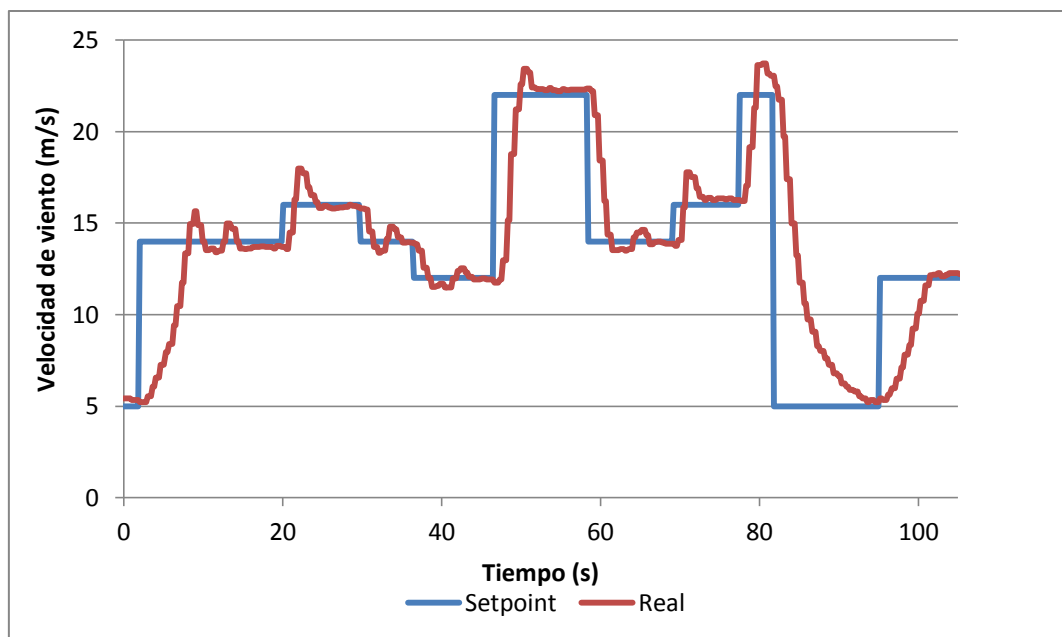
Cuadro 10 Especificaciones del sistema

<i>Parámetro</i>	<i>Valor</i>
Rango de velocidades de viento	2m/s - 27 m/s
Resolución	1m/s
Error	± 0.5 ms
Consumo máximo de corriente	8.5 A
Eficiencia a máxima velocidad	14.3%

E. Algoritmo de control automático

La Figura 37 muestra los resultados de una prueba que se realizó al controlador automático cambiando la referencia. Puesto que el sistema no es lineal (y el actuador limitado físicamente) los parámetros como tiempo de subida o tiempo de asentamiento no son constantes. El Cuadro 11 muestra los valores de estos parámetros para un caso, cambiando la referencia de 5 m/s a 14 m/s.

Figura 37 Prueba de desempeño del algoritmo de control automático



Cuadro 11 Desempeño del control automático al cambiar la referencia de 5m/s a 14ms/

<i>Parámetro</i>	<i>Valor</i>
Tiempo de subida	6s
Tiempo de asentamiento	8.7s
Sobreelevación	1.5 m/s
Error en estado estable	-0.34 m/s

VIII. CONCLUSIONES

- El mecanismo diseñado permite mover el ventilador por medio del motor y no manualmente y el microcontrolador permite que el movimiento sea controlado desde una computadora. El costo del sistema fue de Q1605.24.
- La eficiencia del sistema tiene un límite superior de 22.28% dado por el tornillo de potencia. Por esta razón para obtener una mejor eficiencia se tendría que cambiar completamente el sistema.
- El circuito diseñado con los relés y transistores permite que el ventilador sea encendido desde la computadora, a través del microcontrolador y también a través de los pulsadores.
- La interface creada (sección VI.H.3) permite que se envíen comandos al microcontrolador vía I2C o por el módulo USART. De esta manera una computadora conectada a un maestro puede enviar comandos o hacerlo directamente vía USART apegiándose a los comandos.
- El controlador PID implementado en el microcontrolador permite que se pueda fijar una velocidad de viento automáticamente desde la computadora sólo enviando al sistema la velocidad de viento deseada. El desempeño del controlador es adecuado a la aplicación. No se puede presentar los parámetros usuales para evaluar al controlador (tiempo de respuesta, tiempo de asentamiento) debido a que, por no ser lineal, el sistema se comporta de manera distinta dependiendo del setpoint y el valor inicial.
- Es posible controlar una planta no lineal si se puede tolerar que el desempeño no sea igual en todo el rango de operación. Se puede optimizar el desempeño si se divide el rango de operación en partes y se ajusta un controlador para cada parte.

IX. RECOMENDACIONES

El objetivo final de este trabajo de graduación era crear un sistema que hiciera posible establecer la velocidad del viento en la sección de pruebas del túnel. El sistema que se creó es relativamente complejo. Posee varios componentes mecánicos, circuitos electrónicos y requiere programación de un microcontrolador. Todo esto hace que sea difícil de reparar y modificar. Aún más al estar compuesto de muchas partes es más probable que falle que un sistema más simple. Una alternativa a este sistema que podría también cumplir con los objetivos sería instalar un variador de frecuencia que controle la velocidad del ventilador. Este sistema reduciría la cantidad de componentes del sistema, eliminando la necesidad de elementos mecánicos. Esto resultaría en un sistema menos propenso a fallar. Si se desea mejorar el sistema se recomienda que se explore esta posibilidad.

Si se desea mejorar el sistema existente y no cambiar por completo, existen varias mejoras que se podrían hacer. La primera es cambiar el tornillo, que es de aluminio, por uno de acero. El tornillo se tuvo que enderezar pues por alguna razón se había torcido. Si se cambia de material se reduciría la posibilidad de que esto pase de nuevo. De seguir esta recomendación se sugiere que se maquine un eje de 1/2in para que se pueda instalar un cuña que proveería una mejor interfaz eje-acople que el método actual.

Otra mejora sería colocar un cojinete axial igual al que se instaló entre el perfil cuadrado y el espaciador, pero en el lado opuesto del perfil cuadrado. La razón es que se observó que el tornillo fricciona en este punto al abrir el túnel. Esto hace que se requiera un torque distinto para girar el tornillo según la dirección. Instalar el cojinete reduciría esta diferencia y también incrementaría la eficiencia del sistema.

Si bien la estructura que se diseñó para sujetar la caja reductora es satisfactoria se podría mejorar. Una mejora sería diseñarla de manera que se pueda alinear el motor al tornillo de una manera más fácil y precisa.

El desempeño del modo automático es adecuado a la aplicación aunque se podría optimizar para obtener tiempos de subida y asentamiento menores. Por las características de la aplicación no es necesario buscar tiempos menores, pero si se deseara optimizar el sistema se recomienda dividir el sistema en más pedazos. En este diseño se dividió solamente en dos partes y el resultado fue satisfactorio pero se podría dividir en más partes para que cada parte sea relativamente lineal. Se tendría que ajustar las ganancias para cada rango.

X. BIBLIOGRAFÍA

- Aström, K. J. & Murray, R. M., 2008. PID Control. En: *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton : Princeton University Press, pp. 313-316.
- Benson, T., 1992. *NASA's Wind Tunnels*. [En línea]
Available at: <http://www.nasa.gov/centers/langley/news/factsheets/WindTunnel.html>
- Benson, T., 2009. *Types of Wind Tunnels*. [En línea]
Available at: <http://www.grc.nasa.gov/WWW/K-12/airplane/tuntype.html>
- Benson, T., 2012. *Wind Tunnel Parts*. [En línea]
Available at: <http://www.grc.nasa.gov/WWW/K-12/airplane/tunpart.html>
- Benson, T., 2012. *Wind Tunnel Testing*. [En línea]
Available at: <http://www.grc.nasa.gov/WWW/K-12/airplane/tuntest.html>
- Charais, J. & Lourens, R., 2004. *Software PID Control of an Inverted Pendulum Using the PIC16F684*. [En línea]
Available at: <http://ww1.microchip.com/downloads/en/AppNotes/00964A.pdf>
- Condit, R., 2004. *Brushed DC Motor Fundamentals*. [En línea]
Available at: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en012149
- Durfee, W., 2011. *Understanding DC Motors*. [En línea]
Available at: <http://www.me.umn.edu/courses/me2011/arduino/technotes/dcmotors/motor-tutorial/>
[Último acceso: 09 Septiembre 2013].
- Gosh, S., 2004. Open-Loop and Closed-Loop Systems. En: *Control Systems*. Singapur: Pearson Educación, pp. 3-7.

- Johnson, M. A., Crowe, J. & Moradi, M. H., 2006. PID Controller Implementation Issues. En: M. Moradi, ed. *PID Control: New Identification and Design Methods*. s.l.:Springer, pp. 17-28.
- Joliet Technologies, s.f. *DC Drive Fundamentals*. [En línea] Available at: http://www.joliettech.com/dc_drive_fundamentals.htm [Último acceso: 09 Septiembre 2013].
- Mott, R., 2006. Cuñas, Acoplamientos y Sellos. En: S. Helba, ed. *Diseño de Elementos de Máquinas*. Nauacalpan de Juárez: Prentice Hall, pp. 492-517.
- Mouton, J., 2008. *Brushed DC Motor Basics*. [En línea] Available at: http://www.microchip.com/stellent/groups/SiteComm_sg/documents/DeviceDoc/en543041.pdf
- Page, M., 1999. *Understanding DC Motor Characteristics*. [En línea] Available at: <http://lancet.mit.edu/motors/motors3.html> [Último acceso: 09 Septiembre 2013].
- Valenti, C., 2004. *Implementing a PID Controller Using a PIC18 MCU*. [En línea] Available at: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en020434
- Zhong, J., 2006. *PID Controller Tuning: A Short Tutorial*. [En línea] Available at: http://www.wdsa.uqac.ca/~rbeguena/Systemes_Asservis/PID.pdf [Último acceso: 23 Septiembre 2013].

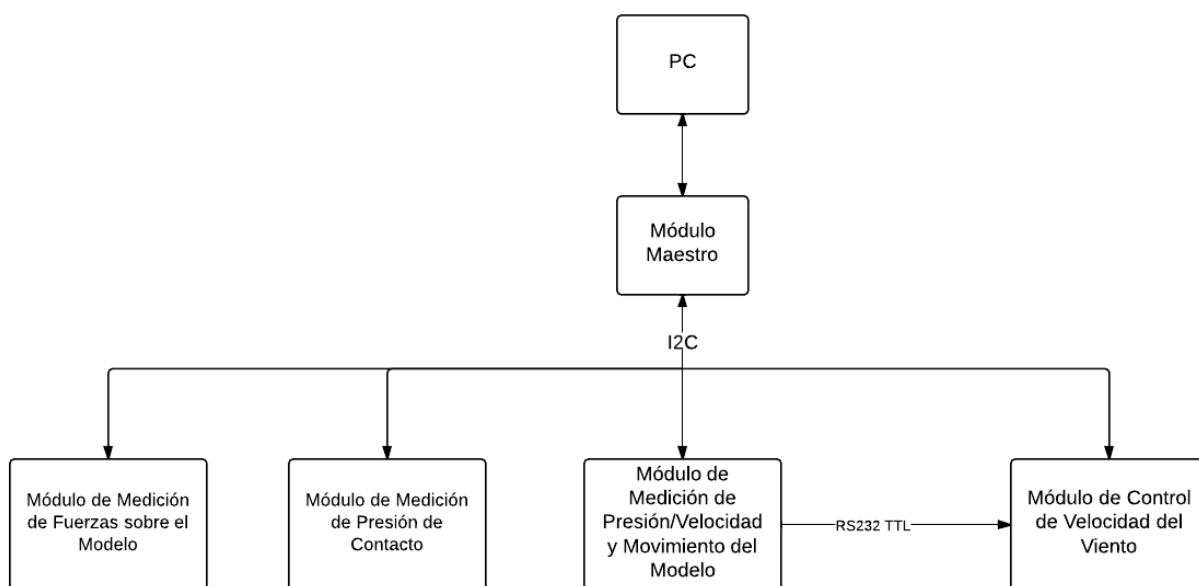
XI. ANEXOS

A. Explicación general del proyecto

Como se mencionó a lo largo del documento, este trabajo de graduación es parte de un proyecto mayor cuyo objetivo fue dotar al túnel de viento de los elementos necesarios para poder realizar prácticas de laboratorio con él. A continuación se dará una explicación de las otras partes de este proyecto y cómo este trabajo de graduación aporta al todo.

El proyecto fue dividido en cinco módulos, cada uno trabajado por un estudiante distinto. Al final se unieron estos módulos para crear un único sistema que cumple con el objetivo. La Figura 38 muestra estos módulos y cómo están conectados entre sí.

Figura 38 Esquema general del proyecto



El denominado módulo maestro incluye el programa en la PC y la comunicación entre este y los demás módulos. Periódicamente este módulo recopila los datos que generan los otros módulos y les envía comandos de ser necesario. Los otros módulos son: módulo de medición de fuerzas, módulo de medición de presión y movimiento del modelo, módulo de medición de presión de contacto y el módulo de control de la velocidad del viento, que fue el tema de este trabajo de graduación.

Dentro del sistema, el módulo de control de velocidad del viento es un actuador. Los otros módulos obtienen mediciones y las envían al módulo maestro para que la información sea entregada al usuario. Este módulo, en cambio, no hace ninguna medición, si no que cuando el usuario demanda un cambio de velocidad del viento, el módulo maestro la transmite y es este módulo en encargado de realizarla (la comunicación con el módulo maestro se detalla en la sección VI.H.3). Como es necesario conocer la velocidad del viento hay una conexión, como se observa en la Figura 38, por donde se transmite esta información. Esto se explicó con detalle en la sección VI.I.1.

Tal como es el caso con este trabajo de graduación, para cada módulo existe un trabajo de graduación asociado. Refiérase a estos para información específica de cada módulo.

1. “Implementación de la Interfaz de Usuario para el Túnel de Viento de la Universidad del Valle de Guatemala” por Roberto Moreno.
2. “Motricidad de la base para el modelo de experimentación e instrumentación electrónica para medición de presión del túnel de viento modelo 303A de la Universidad del Valle de Guatemala” por Marco Guzmán.
3. “Instrumentación Electrónica y Metodología de Obtención de Fuerzas para el Cálculo de los Coeficientes de Arrastre y Sustentación en Túnel de Viento” por Javier Castellanos
4. “Medición de presión superficial para un modelo a escala de un alerón dentro del túnel de viento académico de la Universidad del Valle de Guatemala” por Fernando Arroyo.

B. Cálculos para el motor seleccionado M5-RS540-12

Datos del fabricante:

Velocidad sin carga: 16800 rpm

Corriente sin carga: 1 A

Torque máximo: 278.8 mN · m

Corriente máxima: 42 A

$K_t = 6.8 \text{ mN} \cdot \text{m} / \text{A}$

Relación caja reductora: 26: 1

Torque requerido por la carga:

5 lb in \rightarrow 565 mN · m

Torque requerido del motor:

$$\frac{565 \text{ mN} \cdot \text{m}}{26} = 21.73 \text{ mN} \cdot \text{m}$$

Corriente:

$$\text{Corriente} = C_{T=0} + K_t T = 1 \text{ A} + \frac{(21.73 \text{ mN} \cdot \text{m})}{6.8 \text{ mN} \cdot \text{m} / \text{A}} = 4.2 \text{ A}$$

RPM del motor:

$$\begin{aligned} \text{Torque} &= T_{\text{máx}} - \omega \frac{T_{\text{máx}}}{\omega_{\text{máx}}} \\ 21.73 \text{ mN} \cdot \text{m} &= 278.8 \text{ mN} \cdot \text{m} - \omega \frac{278.8 \text{ mN} \cdot \text{m}}{16800 \text{ rpm}} \\ \omega_{\text{motor}} &= \frac{16800 \text{ rpm} (278.8 \text{ mN} \cdot \text{m} - 21.73 \text{ mN} \cdot \text{m})}{278.8 \text{ mN} \cdot \text{m}} = 15490 \text{ rpm} \end{aligned}$$

RPM en la salida de la caja:

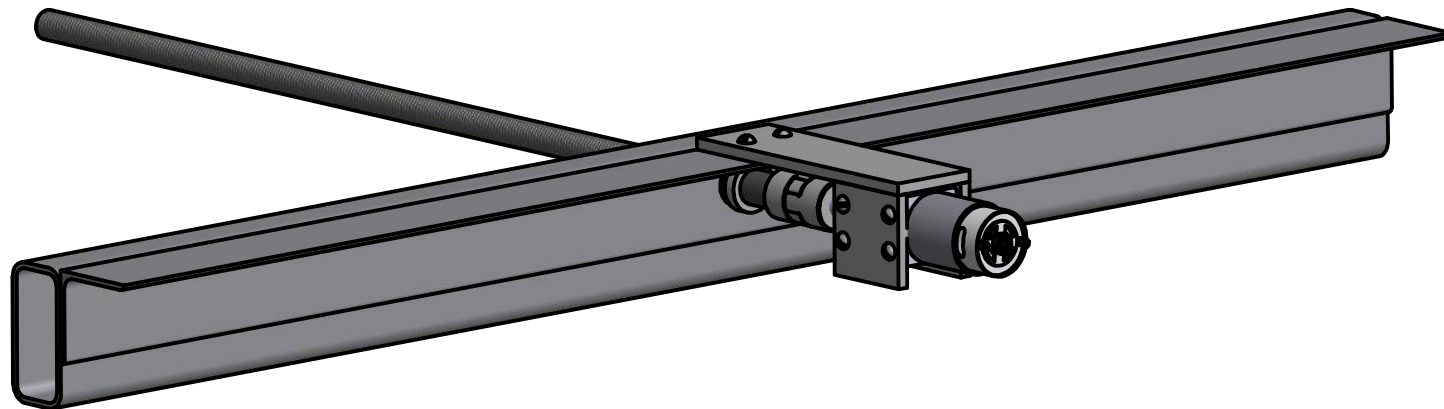
$$\omega_{\text{caja}} = \frac{15490 \text{ rpm}}{26} = 596 \text{ rpm}$$

Eficiencia:

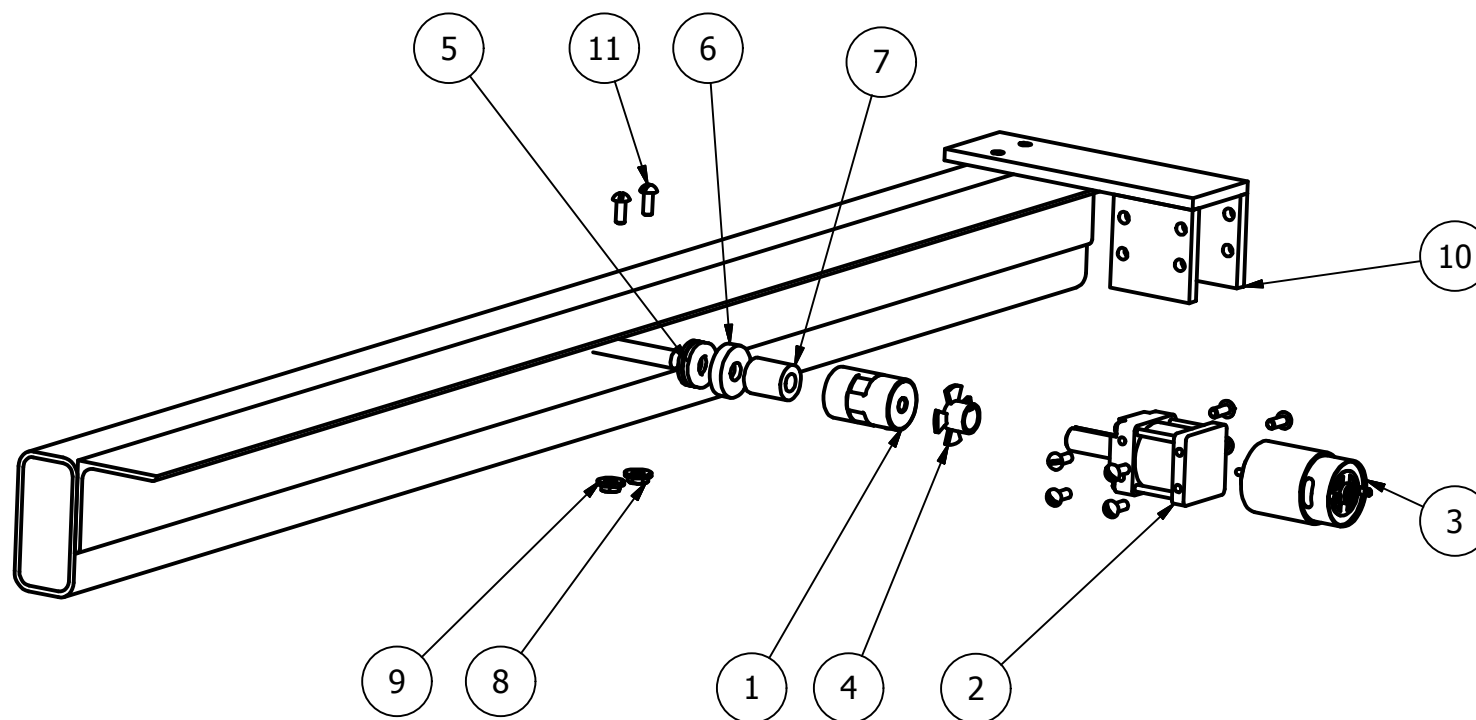
$$\eta = \frac{-\frac{1760 \text{ Hz}}{(0.2788 \text{ N} \cdot \text{m})} (0.02173 \text{ N} \cdot \text{m})^2 + 1760 \text{ Hz} (0.02173 \text{ N} \cdot \text{m})}{(12 \text{ V})(4.2 \text{ A})} = 0.700 \rightarrow 70\%$$

C. Planos de piezas fabricadas

Sistema electromecánico para el control de la velocidad del viento en el túnel de viento de la Universidad del Valle de Guatemala



Josué Samuel Barrientos Castañeda
Guatemala, Octubre 2013



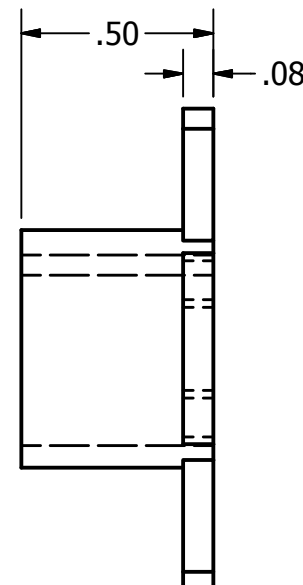
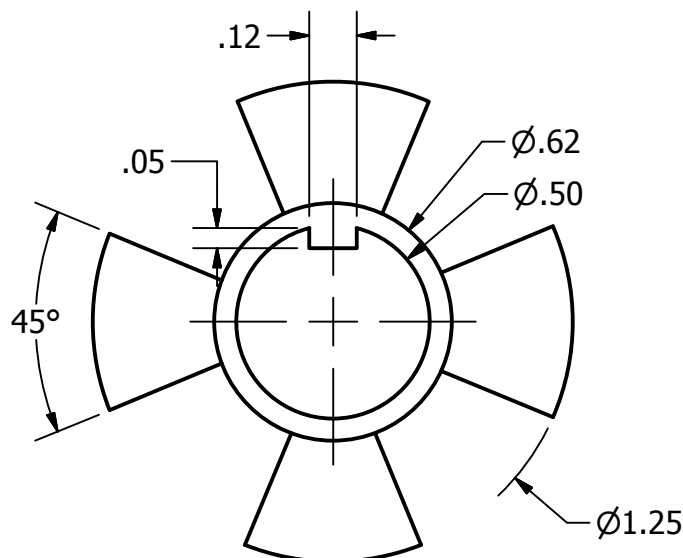
Listado de Partes

ITEM	CANTIDAD	PARTE
1	1	Acople L050
2	1	Caja Reductora
3	1	Motor
4	1	Encoder
5	1	Rodamiento 51100
6	1	Centrador
7	1	Espaciador
8	2	Tuerca
9	2	Roldana
10	1	Guarda
11	10	Tornillo 10-32

Título: Despiece		
Proyecto: Sistema electromecánico para el control de la velocidad del viento		
Autor: Josué Barrientos	Unidades mm	Escala 1/4
Fecha: 16.09.2013	No. de pieza 16	Página 2 /6

2

1



B

B

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

A

A

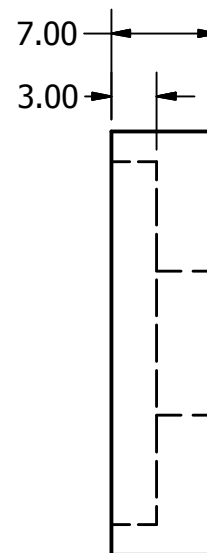
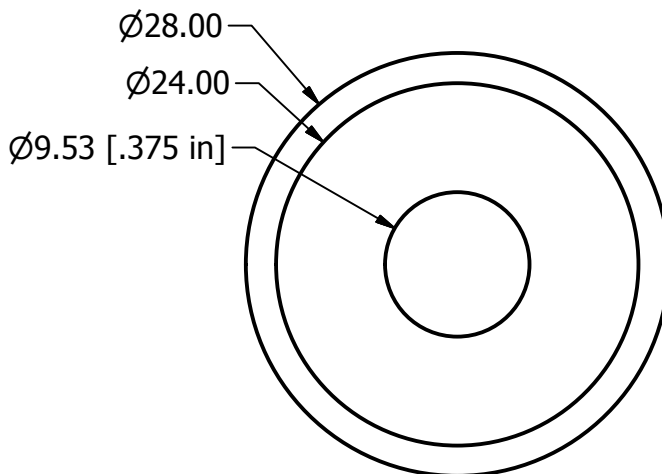
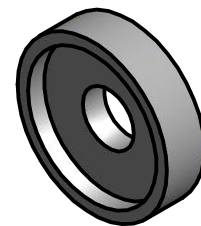
Título:		Encoder	
Proyecto: Sistema electromecánico para el control de la velocidad del viento			
Autor: Josué Barrientos	Unidades in	Escala 2:1	
Fecha: 16.09.2013	No. de pieza 4	Página 3 / 6	

2

1

2

1



B

B

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

A

A

Título: Centrador		
Proyecto: Sistema electromecánico para el control de la velocidad del viento		
Autor: Josué Barrientos	Unidades mm	Escala 2:1
Fecha: 16.09.2013	No. de pieza 6	Página 4 /6

2

1

2

1

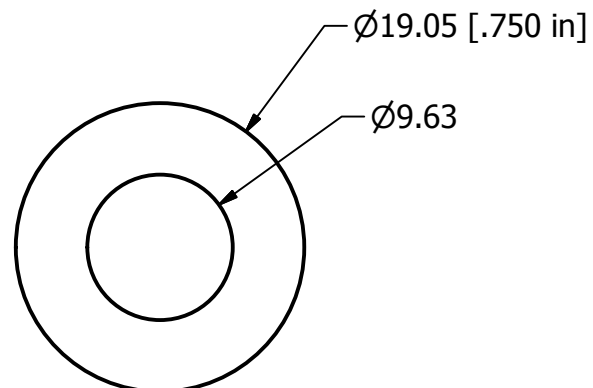
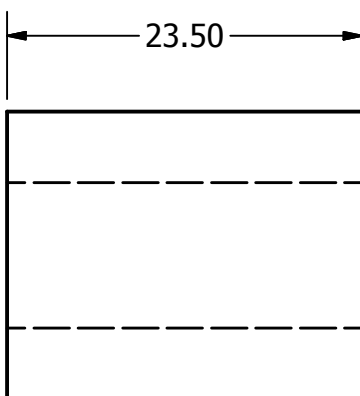


B

B

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT



A

A

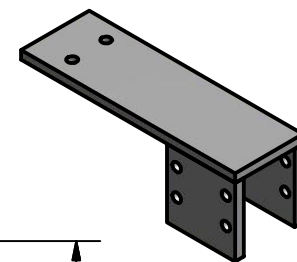
Título: Espaciador		
Proyecto: Sistema electromecánico para el control de la velocidad del viento		
Autor: Josué Barrientos	Unidades mm	Escala 2:1
Fecha: 16.09.2013	No. de pieza 7	Página 5 /6

2

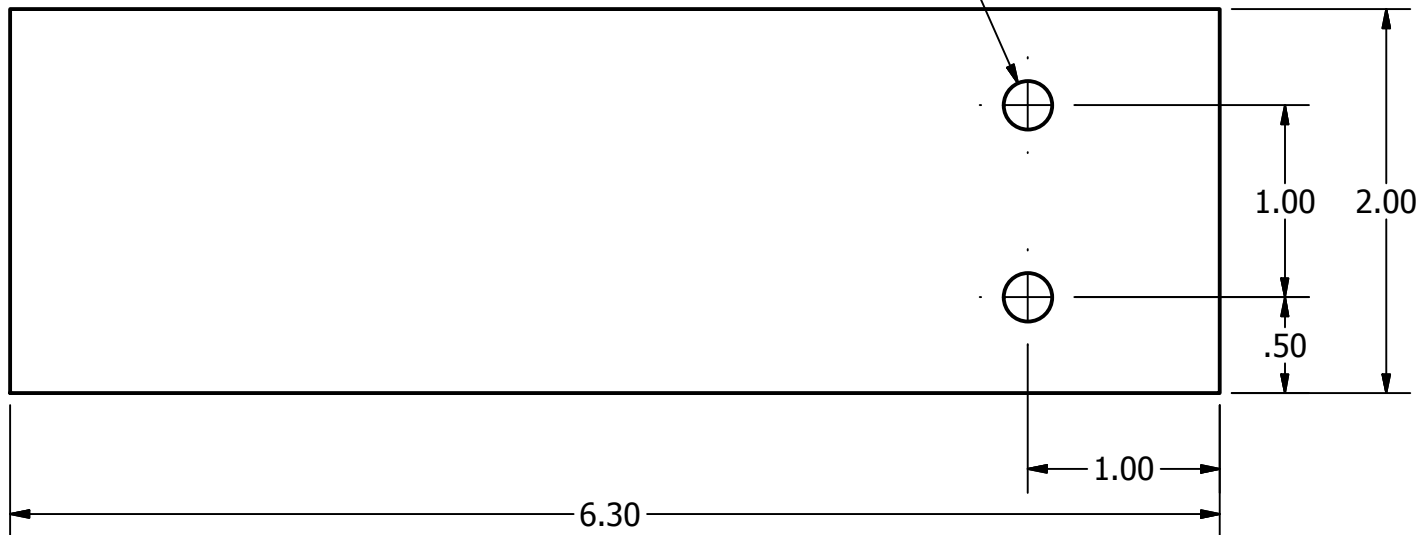
1

2

1



4X Ø.25 Pasado



B

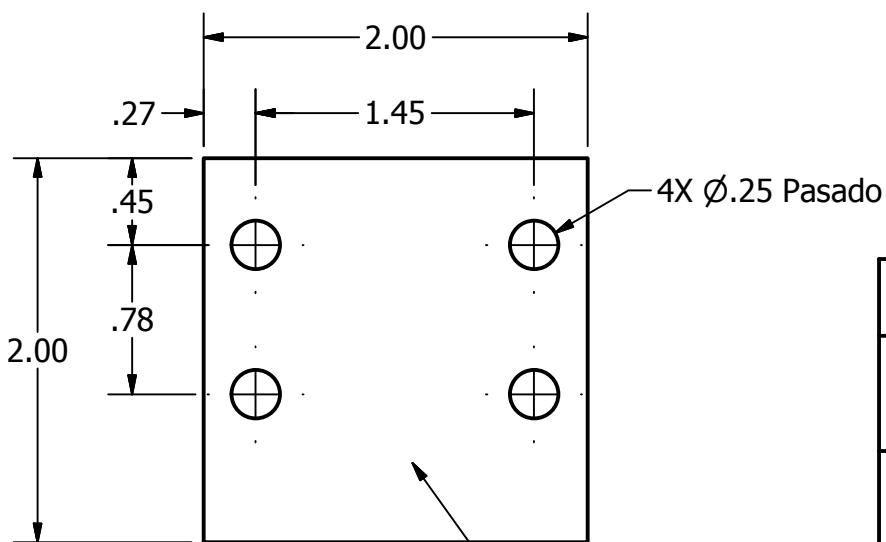
B

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

A

A



4X Ø.25 Pasado

2X

2

1

Título: Guarda		
Proyecto: Sistema electromecánico para el control de la velocidad del viento		
Autor: Josué Barrientos	Unidades in	Escala 1:1
Fecha: 16.09.2013	No. de pieza 10	Página 6 / 6

D. Código fuente del microcontrolador

```

1: /*****
2: Archivo: Control del Motor.c
3: Descripción: Programa que controla el motor que mueve el ventilador en el tunel
4: de viento de la UVG. Parte del trabajo de graduacion "Sistema electromecánico
5: para el control de la velocidad del viento en el túnel de viento de la
6: Universidad del Valle de Guatemala".
7: Autor: Josue Samuel Barrientos Castañeda
8: Compilador: Mikroc
9: Microcontrolador: PIC18F45K22
10: Fecha de modificacion: 14.10.2013
11: Diponible online en:
12: https://code.google.com/p/sistema-electromecanico-velocidad-viento/
13: *****/
14:
15: //*****Definición de Variables*****
16: const Dir=0x50, Mina=28, Mind=51;
17: unsigned short portbtemp, OK=1, cnti2c=0, cntuart=0, next, percentage;
18: unsigned short comando[2]={0,0}, P1, P2, uart20, uart21, uart22;
19: unsigned short tempshort, tempshort2, tempi2c, porcentaje_pasado=0;
20: char txtlong[15], txt[7];
21: float corriente, freq, k, Cn, error, vel_pasada, vel_actual, integral;
22: unsigned int cuartos, tempint, CCP, vel_deseada=14;
23: signed int presion_actual, derivativo;
24: signed short tempsshort;
25: bit automatico, procesar_comandos, sampled, encendido;
26:
27: //*****Definición de Macros*****
28: #define PARAR() PORTA.RA6=PORTA.RA7=0;
29: #define ADELANTE() PORTA.RA6=0; PORTA.RA7=1;
30: #define DETRAS() PORTA.RA6=1; PORTA.RA7=0;
31: #define PWM(valor) CCPRL=valor;
32: #define CONVER(min, per) (255-min)*per/100+min;
33: #define MICRO_ADELANTE PORTB.RB4!=0
34: #define MICRO_ATRAS PORTB.RB5!=0
35: #define ENCENDER_TUNEL() PORTE.RE0=1; delay_ms(1000); PORTE.RE0=0;
36: #define APAGAR_TUNEL() PORTE.RE1=1; delay_ms(1000); PORTE.RE1=0;
37:
38: //*****Inicializacion de Módulos y Puertos*****
39: void Inicializar(){ //Oscilador interno a 8Mhz
40: OSCCON.IRCF2=1;
41: OSCCON.IRCF1=1;
42: OSCCON.IRCF0=0;
43: ANSELA = 0B00000100; //Configuracion de pines analogicos
44: ANSELB = 0;
45: ANSELC = 0;
46: ANSELD = 0;
47: ANSELE = 0;
48: C1ON_bit = 0; //Comparadores deshabilitados
49: C2ON_bit = 0;
50:
51: PORTE.RE0=0; //Configuracion de puertos
52: Delay_ms(50);

```

```

53: PORTE.RE1=0;
54: PORTA = 0;
55: TRISA = 0b00000100;
56: TRISB = 0b11111111;
57: TRISC = 0b00011010;
58: TRISE = 0b00000100;
59:
60: UART1_Init(9600); //Inicializar Comunicacion serial a 9600
61: Delay_ms(100);
62:
63: UART2_Init(9600); //Inicializar Comunicacion serial a 9600
64: Delay_ms(100);
65:
66: PWM1_Init(500); //Inicializar PWM a 50Hz
67: Delay_ms(100);
68: PWM(0)
69: PWM1_Start();
70:
71: SSP1ADD = Dir; //Direccion de esclavo I2C
72: SSP1CON1 = 0x36; //Configuracion I2C esclavo 7bit adress
73: Delay_ms(100);
74:
75: ADC_INIT();
76: Delay_ms(100);
77:
78: CCP2CON=0b00000101; //Configuracion de modulo CCP2 en modo
79: //capture para cada flanco de subida
80: T1CON.TMR1CS1 = 0; //Configuracion del TMR1.
81: T1CON.TMR1CS0 = 0; //Prescaler= 8 | Precarga = 0
82: T1CON.T1CKPS1 = 1; //Frecuencia de interrupcion = 3.81 Hz
83: T1CON.T1CKPS0 = 1; //Periodo de interrupcion = 0.262144 s
84: T1CON.T1OSCEN = 0;
85: T1CON.T1SYNC = 0;
86: T1CON.T1RD16=0;
87: T1CON.TMR1ON = 1;
88:
89: TOCON.TMR0ON=1;
90: TOCON.T08BIT=0;
91: TOCON.T0CS = 0; //Configuracion del TMR1.
92: TOCON.T0SE = 0; //Usado para rebote de los microswitches
93: TOCON.PSA = 1; //Prescaler= 256 | Precarga = 49675
94: TOCON.T0PS2 = 0; //Frecuencia de interrupcion = 33.10 Hz
95: TOCON.T0PS1 = 0; //Periodo de interrupcion = 0.030208 s
96: TOCON.T0PS0 = 0;
97: TMR0H=0xC2;
98: TMR0L=0x0B;
99:
100: PIE3.RC2IE=1; //Habilitar interrupcion de recepcion de USART2
101: PIE1.RC1IE=1; //Habilitar interrupcion de recepcion de USART
102: IOCB.RB4=1; //Habilitar 'Interrupt on Change' en pin RB4
103: IOCB.RB5=1; //Habilitar 'Interrupt on Change' en pin RB5
104: PIE1.SSPIE = 1; //Habilitar interrupciones de I2C

```

```

105: PIE2.CCP2IE=1; //Habilitar interrupciones de CCP2
106: INTCON.RBIE=1; //Habilitar interrupciones del PORTB
107: INTCON.PEIE = 1; //Habilitar interrupciones de perifericos
108: INTCON.GIE = 1; //Habilitar interrupciones
109:
110: automatico=0;
111: procesar_comandos=0;
112: }
113:
114:
/*****
115: Función: Moverse_Continuo
116: Descripcion: Genera las salidas necesarias para que se mueva el motor en la
117: direccion y velocidad deseada. No acepta cambios bruscos de
118: velocidad. Solo hace cambios de 5%.
119: Parametros: Direccion de movimiento 'a'-adelante 'd'-atras
120: Porcentaje de velocidad 0-100
121: Resultado: El PWM se coloca al porcentaje necesario para dar el porcentaje de
122: velocidad y los pines de direccion se configuran.
123:
*****/
124: void Moverse_Continuo(short dir, unsigned short porcentaje){
125: if (porcentaje>100){porcentaje=100;}
126:
127: tempsshort=porcentaje-porcentaje_pasado;
128: if (tempsshort>5){porcentaje=porcentaje_pasado+5;}
129: else if (tempsshort<-5){porcentaje=porcentaje_pasado-5;}
130:
131: if(dir=='a' && MICRO_ADELANTE){
132: tempshort=CONVER(Mina, porcentaje)
133: if (PORTA.RA6 && PORTA.RA7==0 && CCP1L!=0){tempshort=0;porcentaje=1;};
134: ADELANTE()
135: PWM(tempshort)
136: }
137: else if (dir=='d' && MICRO_ATRAS){
138: tempshort=CONVER(Mind, porcentaje)
139: if (PORTA.RA7 && PORTA.RA6==0 && CCP1L!=0){tempshort=0; porcentaje=1;};
140: DETRAS()
141: PWM(tempshort)
142: }
143: UART2_WRITE_TEXT(",, ");
144: bytetostr(tempshort, txt);
145: UART2_WRITE_TEXT(txt);
146: porcentaje_pasado=porcentaje;
147: }
148:
149:
/*****
150: Función: Moverse_Pasos
151: Descripcion: Mueve el motor por pasos. Un paso es hasta que el encoder genera
152: un pulso.

```

```

153: Parametros: Parametros: Direccion de movimiento 'a'-adelante 'd'-atras
154: Porcentaje de velocidad 0-100
155: Numero de pasos
156: Resultado: La variable cuartos se hace cero. El ventilador se encuentra en la
157: posición cero.
158:
159: *****/
159: void Moverse_Pasos(char dir, short velocidad, short pasos ){
160: tempint=cuartos;
161:
162: if(dir=='a' && MICRO_ADELANTE){
163: tempint+=-pasos;
164: ADELANTE()
165: PWM(velocidad)
166: while(cuartos!=tempint && MICRO_ADELANTE){}
167: }
168: else if (dir=='d' && PORTB.RB5!=0){
169: tempint+=pasos;
170: DETRAS()
171: PWM(velocidad)
172: while(cuartos!=tempint && MICRO_ATRAS){}
173: }
174:
175: PARAR()
176: }
177:
178:
179:
180: /*****/
180: Función: Auto_Zero
181: Descripcion: Mueve el ventilador cerrandolo hasta que se activa el microswitch
182: del frente. Primero lo hace rápido, se separa de nuevo y lo hace de
183: nuevo lentamente.
184: Resultado: La variable cuartos se hace cero. El ventilador se encuentra en la
185: posición cero.
186:
187: *****/
187: void Auto_Zero(){
188:
189: Moverse_Continuo('a', 30);
190: while (MICRO_ADELANTE){}
191: PARAR()
192: delay_ms(250);
193:
194: Moverse_Continuo('d', 30);
195: delay_ms(500);
196: PARAR()
197:
198: Moverse_Continuo('a', 10);
199: while (MICRO_ADELANTE){}
200: PARAR()
201:

```

```

202: cuartos=0;
203: }
204:
205:
/*****
206: Función: Lobby
207: Descripción: Interpreta los datos recibidos por el puerto serial o por I2C.
208: Precondición: La variable comando ha sido cargada con el dato que llegó por el
209: puerto
210: Resultado: Se settean distintas banderas o se llama a distintas funciones
211: según el valor de la variable comando.
212:
*****/
213: void Lobby(){
214: automatico=0;
215: switch (comando[0]){
216: case 'w':
217: if (CCPR1L!=255){CCPR1L++;}
218: break;
219: case 's':
220: if (CCPR1L!=0){CCPR1L--;}
221: break;
222: case 'a':
223: do {
224: Moverse_Continuo('a', comando[1]);
225: delay_ms(100);
226: }while (comando[1]!=porcentaje_pasado);
227: break;
228: case 'd':
229: do {
230: Moverse_Continuo('d', comando[1]);
231: delay_ms(100);
232: }while (comando[1]!=porcentaje_pasado);
233: break;
234: case 'e':
235: ENCENDER_TUNEL();
236: encendido=1;
237: break;
238: case 'o':
239: APAGAR_TUNEL();
240: encendido=0;
241: break;
242: case 'f':
243: PARAR();
244: porcentaje_pasado=0;
245: break;
246: case '4':
247: PWM(30)
248: break;
249: case '5':
250: PWM(50)
251: break;

```

```
252: case '6':
253: PWM(75)
254: break;
255: case '7':
256: PWM(200)
257: break;
258: case '8':
259: PWM(250)
260: break;
261: case '0':
262: PWM(0)
263: porcentaje_pasado=0;
264: break;
265: case 'z':
266: Auto_Zero();
267: porcentaje_pasado=0;
268: break;
269: case '+':
270: Moverse_Pasos('d', 60, 1);
271: porcentaje_pasado=0;
272: break;
273: case '-':
274: Moverse_Pasos('a', 45, 1);
275: porcentaje_pasado=0;
276: break;
277: case 'm':
278: Moverse_Pasos('d', 55, 4);
279: porcentaje_pasado=0;
280: break;
281: case 'n':
282: Moverse_Pasos('a', 45, 4);
283: porcentaje_pasado=0;
284: break;
285: case 'p':
286: automatico=1;
287: PARAR();
288: porcentaje_pasado=0;
289: vel_deseada=comando[1];
290: break;
291: case 't':
292: vel_deseada+=1;
293: automatico=1;
294: porcentaje_pasado=0;
295: break;
296: case 'y':
297: vel_deseada-=1;
298: automatico=1;
299: porcentaje_pasado=0;
300: break;
301: case 'u':
302: vel_deseada=16;
303: automatico=1;
```

```

304: porcentaje_pasado=0;
305: break;
306: case 'i':
307: vel_deseada=12;
308: automatico=1;
309: porcentaje_pasado=0;
310: break;
311: case 'o':
312: vel_deseada=20;
313: automatico=1;
314: porcentaje_pasado=0;
315: break;
316: }
317: comando[0]=0;
318: comando[1]=0;
319: procesar_comandos=0;
320: }
321:
322:
/*****
323: Función: Revisar_Microswitches
324: Descripcion: Revisa que los pines a los que se conectan las terminales NC y la
325: NO de los microswitches tengan valores distintos. Si son iguales
326: indica que hay algun problema.
327: Resultado: Se modifica la variable OK lo que hara que el PIC salga del loop
328: principal.
329:
*****/
330: void Revisar_Microswitches(){
331: if (PORTB.RB4!=~PORTB.RB1){OK=10;}
332: if (PORTB.RB5!=~PORTB.RB2){OK=20;}
333: }
334:
335:
/*****
336: Función: Control_PID
337: Descripcion: Implementa un controlador PID para establecer la velocidad de
338: viento deseada.
339:
*****/
340: void Control_PID(){
341: vel_actual=1.026*sqrt(presion_actual);
342: error=vel_deseada-vel_actual;
343:
344: derivativo=vel_actual-vel_pasada+0.5;
345: // if (MICRO_ADELANTE && MICRO_ATRAS){integral=integral+error;}
346: // if (integral>3){integral=0;}
347: // else if (error<-3){integral=0;}
348:
349: if (vel_deseada>=15 && vel_actual>=15){Cn=error*3+derivativo*6;}//+integral;}
350: else {Cn=error*12+derivativo*3;}//+integral;}
351:

```

```

352: percentage=abs (Cn)+0.5;
353: // if (percentage>100){percentage=100;}
354:
355: if (error>0.5 || error<-0.5){
356: if (Cn>0){Moverse_continuo('a', percentage);}
357: else if(Cn<0){Moverse_continuo('d', percentage);}
358: }
359: else {PARAR()}
360:
361: UART2_WRITE_TEXT("\n\r");
362: floattostr(vel_deseada, txtlong);
363: UART2_WRITE_TEXT(txtlong);
364: UART2_WRITE_TEXT(",");
365: //Inttostr(integral, txtlong);
366: //UART2_WRITE_TEXT(txtlong);
367: //UART2_WRITE_TEXT(",");
368: Floattostr(vel_actual, txtlong);
369: UART2_WRITE_TEXT(txtlong);
370: UART2_WRITE_TEXT(",");
371: Bytetostr(percentage, txt);
372: UART2_WRITE_TEXT(txt);
373:
374: vel_pasada=vel_actual;
375: sampled=0;
376: }
377:
378:
379: Función: Reportar_Error
380: Descripcion: La funcion se llama si la variable 'OK' ha se modificó indicando
381: una falla. Segun el valor de la variable se envia por el modulo
382: UART la causa del fallo.
383: Resultado: Se modifica la variable OK lo que hara que el PIC salga del loop
384: principal.
385:
386: void Reportar_Error() {
387: if (OK==10){
388: UART1_WRITE_TEXT("\n\rREVISAR MICROSWITCH DE ADELANTE");
389: }
390: if (OK==20){
391: UART1_WRITE_TEXT("\n\rREVISAR MICROSWITCH DE ATRAS");
392: }
393: PORTE.RE0=1;
394: }
395:
396:
397: Función: main
398: Descripcion: La funcion se llama si la variable 'OK' ha se modificó indicando
399: una falla. Segun el valor de la variable se envia por el modulo
400: UART la causa del fallo.

```

```

401:
*****/
402: void main() {
403: Inicializar();
404: OK=1;
405: Revisar_Microswitches();
406:
407: while (OK==1){
408:
409: if (procesar_comandos){Lobby();}
410: if (automatico && sampled && encendido){Control_PID();}
411:
412: }
413:
414: PARAR()
415: PWM(0)
416: Reportar_Error(); //Se envia un reporte del fallo
417:
418: }
419:
420: //*****Rutina de Servicio de
Interrupción*****
421: void Interrupt() {
422:
423: if (PIR1.SSPIF == 1){
424:
425: PIR1.SSPIF = 0;
426: if (SSPSTAT.D_A==1 && procesar_comandos==0){
427: if (cnti2c==0){
428: comando[0] = SSPBUF;
429: //UART1_Write(comando[0]);
430: cnti2c=1;
431: }
432: else {
433: comando[1] = SSPBUF;
434: //UART1_Write(comando[1]);
435: procesar_comandos=1;
436: cnti2c=0;
437: }
438: }
439: tempi2c = SSPBUF;
440: }
441:
442: if (PIR1.RC1IF) {
443: if (cntuart==0){
444: comando[0]=UART1_READ(); //Leer dato recibido
445: //UART1_Write(comando[0]);
446: cntuart=1;
447: }
448: else{
449: comando[1]=UART1_READ();
450: //UART1_Write(comando[1]);

```

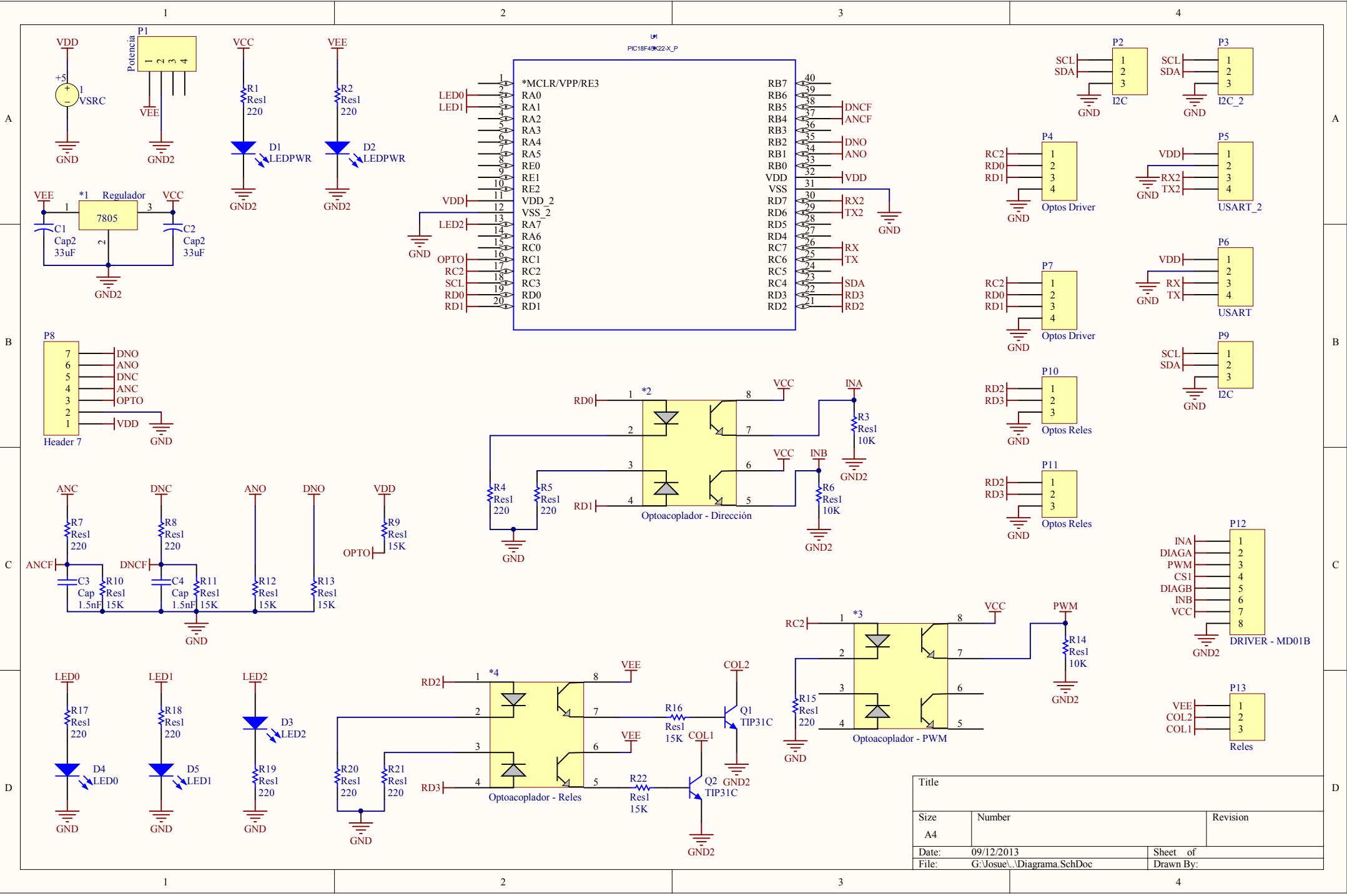
```

451: procesar_comandos=1;
452: PORTD.RD2=~PORTD.RD2;
453: cntuart=0;
454: }
455: }
456:
457: if (PIR3.RC2IF) {
458: uart20=UART2_READ();
459: if (next==2){
460: P2=uart20;
461: presion_actual=P1*256+P2;
462: next=0;
463: sampled=1;
464: PORTA.RA1=~PORTA.RA1;
465: }
466: else if (next==1){
467: P1=uart20;
468: next=2;
469: }
470: else if (uart22==6 && uart21==3 && uart20==2){
471: next=1;
472: }
473: else {
474: uart22=uart21;
475: uart21=uart20;
476: }
477: }
478:
479: if ((INTCON.RBIF==1) && (INTCON.RBIE==1)){
480: portbtemp=PORTB;
481: if ((portbtemp.b4==0) || (portbtemp.b5==0)){
482: //Si el cambio fue de bajada en RB1 o RB5
483: PARAR() //Se para el motor
484: PWM(0); //Ciclo de trabajo del PWM se hace cero.
485: CCP1CON.DC1B1=0;
486: CCP1CON.DC1B0=0;
487: }
488: //uart1_write('!');
489:
490: if (portbtemp.b4==0){cuartos=0;}
491:
492: TMR0H=0xC2; //Se inicia el TMR0 para que no afecte
el rebote
493: TMR0L=0x0B;
494: INTCON.TMR0IF=0; //Limpiar bandera de interrupcion del TMR0
495: INTCON.TMR0IE=1; //Habilitar interrupción del TMR0
496: INTCON.RBIF=0; //Limpiar bandera del puerto B
497: INTCON.RBIE=0; //Habilitar interrupciones dle puerto B.
498: }
499:
500: if (PIR2.CCP2IF) {
501: TMR1H = 0; //Se reinicia el TMR1 a 0

```

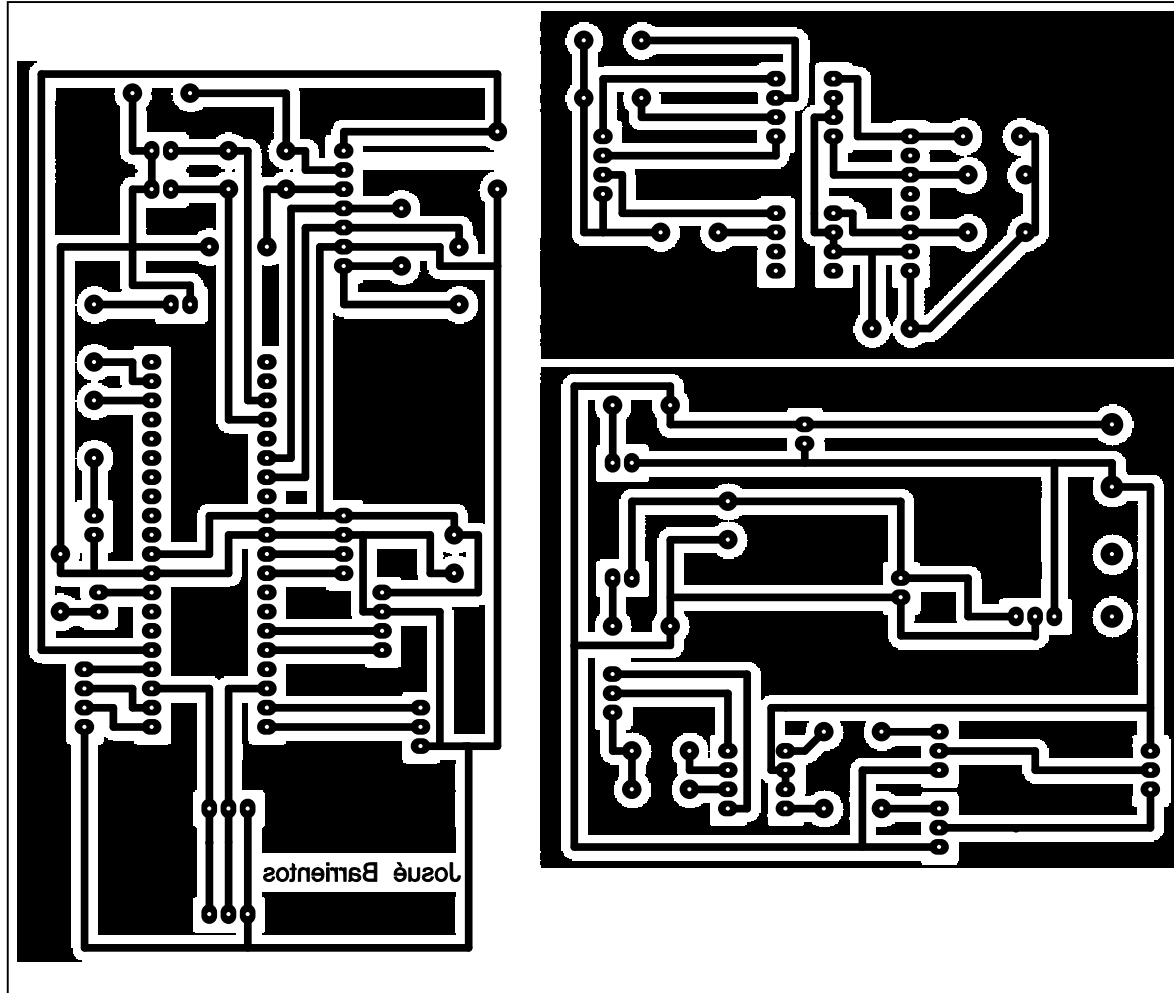
```
502: TMR1L = 0;
503: PIR2.CCP2IF=0; //Limpar bandera de interrupcion
504: PORTA.RA0=~PORTA.RA0; //Cambiar estado del LED
505: tempshort2=1;
506: if(PORTA.RA7){cuartos--;}
507: if(PORTA.RA6){cuartos++;}
508: }
509:
510: if ((INTCON.TMR0IF==1) && (INTCON.TMR0IE==1)){
511: INTCON.TMR0IF = 0; //Limpiar bandera de interrupcion
512: INTCON.TMR0IE = 0; //Deshabilitar la interrupcion
513: INTCON.RBIF=0; //Limpiar bandera de interrupcion
514: INTCON.RBIE=1; //Habilitar interrupcion del puerto B
515: }
516: }
```

E. Diagrama del circuito



Title		
Size	Number	Revision
A4		
Date:	09/12/2013	Sheet of
File:	G:\Josue\Diagrama.SchDoc	Drawn By:

F. Diagrama de PCB



G. Planos de caja reductora

BANEBOTS P60 GEARBOX PRELIMINARY

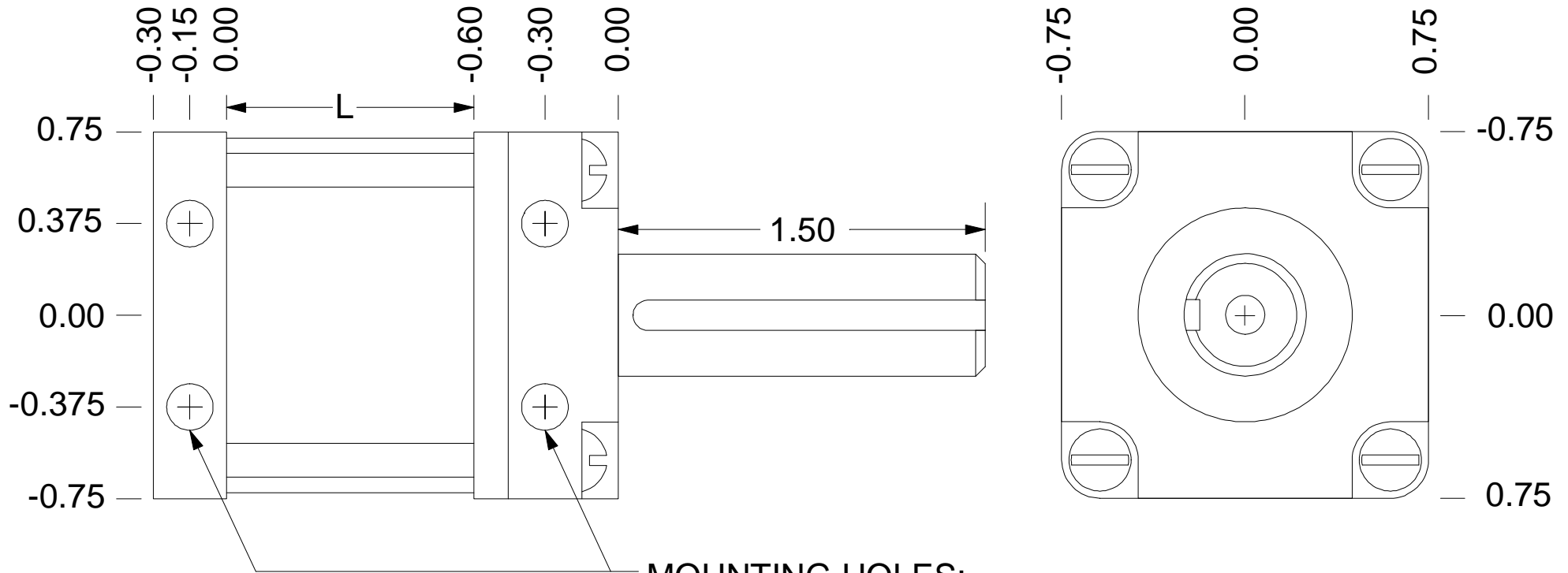
RINGGEAR LENGTH (L)

- 1 STAGE: 0.60
- 2 STAGE: 1.00
- 3 STAGE: 1.40
- 4 STAGE: 1.80

(TOTAL LENGTH = RINGGEAR LENGTH + 0.90)

STANDARD SHAFT:

4140 HARDENED STEEL
0.500 DIAMETER
0.125 KEYWAY TO SHAFT END
#10-32 END TAP, 0.300 MIN DEPTH



MOUNTING HOLES:

#10-32
8 TOTAL
4 TOP, 4 BOTTOM
0.30 MIN DEPTH

ALL DIMENSIONS INCHES
NO HIDDEN LINES SHOWN

PRELIMINARY
11 JAN 09

H. Hoja de datos de fuente de voltaje

Single Output 150 - 200W General Purpose Power Supplies

Features

- ◆ Very low cost
- ◆ Small Size
- ◆ Wide Range AC Input
- ◆ Convection or Fan Cooled
- ◆ Three year warranty
- ◆ 1.6" high (For 1U racking)



Key Market Segments & Applications



Specifications			
Model		LS200 (Enclosed style with internal fan)	LS200/L (U channel style - no internal fan)
AC Input Voltage (300VAC for 5s)	VAC	85 - 264VAC	85 - 264VAC (3)
Input Frequency	Hz	47 - 63Hz	
DC Input Voltage	VDC	120 - 373VDC	
Inrush Current (230VAC, cold start)	A	60	
Power Factor	-	Meets EN61000-3-2, -3 (Typical PF 0.98/0.95)(1)	
Input Current (115/230VAC)	A	3.5 / 1.7 (typical)	
Temperature Coefficient	-	<0.02%/°C (0 - 50°C)	
Overcurrent Protection	-	>105% of nominal or peak. Constant current style	
Overvoltage Protection (2)	V	3.3V: 3.8 - 4.45V, 5V: 5.75 - 6.75V, 7.5V: 8.6 - 10.1V, 12V: 15.1 - 17.75V 15V: 17.25 - 20.25V, 24V: 30.25 - 35.5V, 36V: 41.4 - 48.6V, 48V: 60 - 69.6V	
Overtemperature Protection (2)	-	Yes	
Hold Up Time (115/230V input)	ms	20ms	
Leakage Current (230VAC 60Hz)	mA	<1mA	
Remote Sense	-	Yes	
Remote On/Off	-	On: 0 - 0.8V; Off: 3 - 12V	
LED Indicator	-	Green LED = On	
Operating Temperature	°C	Fan or forced air rating: -25 to +70°C. Derate linearly to 60% load from +50 to +70°C	
Storage Temperature	°C	-40 to +85°C	
Operating Humidity	-	20 - 90% RH (non condensing)	
Storage Humidity	-	10 - 95% RH (non condensing)	
Cooling	-	Internal Fan (air exhausts from fan end)	Convection or customer supplied airflow
Withstand Voltage	-	Input to Ground 1.5kVAC, Input to Output 3kVAC, Output to Ground 500VAC for 1 min.	
Isolation Resistance	-	>100M at 25°C & 70%RH, Output to Ground 500VDC	
Vibration (non operating)	-	10 - 55Hz: 19.6m/s ² constant sweep 1 min X, Y, Z for 1 hour	
Shock	-	< 196.1 m/s ² (20G)	
Immunity	-	IEC61000-4-2, -3, -4, -5, -6, -8, -11	
Safety Agency Approvals	-	UL60950-1, CSA60950-1 (cUL), IEC60950-1 2nd Edition, CE Mark	
Conducted & Radiated EMI	-	EN55011/EN55022-B, FCC--B	
Weight (Typ)	g	700	600
Size (LxWxH)	in	7.8 x 3.9 x 1.61"	
Warranty	yrs	Three Years	

Notes:

- (1) 115 / 230VAC input
- (2) Recycle AC to reset
- (3) (Derate linearly to 65% load from 115VAC to 85VAC)

Output Ratings

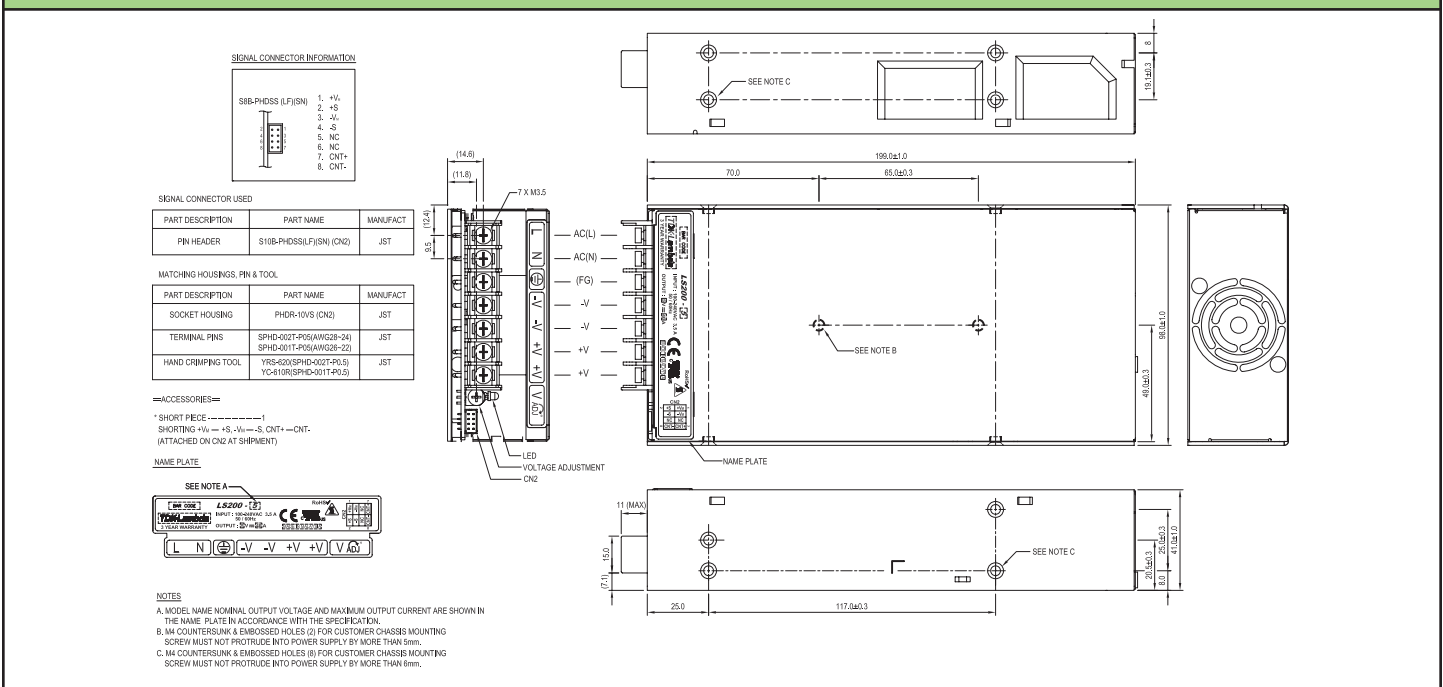
Model	Adjust Range Voltage (V)	Max Current Forced Air(A) ⁽⁴⁾	Max Current Convect.(A) ⁽⁵⁾	Peak Current (A)	Load Reg (mV)	Line Reg (mV)	Ripple Noise (mV)	Efficiency (typ) % ⁽¹⁾
LS200-3.3	3.3V	3.0 - 3.6	40.0	N/A	-	40	16	80 / 68
LS200-5	5V	4.75 - 5.5	40.0	N/A	-	40	20	80 / 75
LS200-7.5	7.5V	6.8 - 8.2	26.7	N/A	-	40	20	80 / 77
LS200-12	12V	10.8 - 14.4	16.7	N/A	-	96	48	76 / 79
LS200-15	15V	13.5 - 16.5	13.4	N/A	-	120	60	80 / 83
LS200-24	24V	22 - 28.8	8.4	N/A	10.4	192	96	82 / 84
LS200-36	36V	32 - 40	5.6	N/A	6.9	288	144	82 / 85
LS200-48	48V	42 - 57.6	4.2	N/A	-	384	192	82 / 85
LS200-3.3/L	3.3V	3.0 - 3.6	40.0	26	-	40	16	67 / 68
LS200-5/L	5V	4.75 - 5.5	40.0	26	-	40	20	72 / 75
LS200-7.5/L	7.5V	6.8 - 8.2	26.7	17.3	-	40	20	74 / 77
LS200-12/L	12V	10.8 - 14.4	16.7	11.6	-	96	48	76 / 79
LS200-15/L	15V	13.5 - 16.5	13.4	9.3	-	120	60	80 / 83
LS200-24/L	24V	22 - 28.8	8.4	5.8	10.4	192	96	82 / 84
LS200-36/L	36V	32 - 40	5.6	3.9	6.9	288	144	82 / 85
LS200-48/L	48V	42 - 57.6	4.2	2.9	-	384	192	82 / 85

Notes

(4) With internal fan version LS200-xx or LS200-xx/L version with external airflow.

(5) See Installation manual for derating curve.

Outline Drawing



Other Related Products

LS25 - 150	25W to 150W low cost
HWS15 - 1800	15W to 1800W limited lifetime warranty
SWS300 - 1000	300 to 1000W single output

Options

Suffix	Description
/L	No cover or fan (U channel)
Blank	Cover and fan

For Additional Information, please visit
us.tdk-lambda.com/lp/products/lis-series.htm

