



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO.**

FACULTAD DE INGENIERÍA.

**“VISCOSÍMETRO CON BASE EN UN
PIC18F4550”.**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO MECATRÓNICO**

PRESENTA:

MIGUEL ÁNGEL PEÑA MARTÍNEZ.

DIRECTOR DE TESIS:

M. I. BENJAMÍN VALERA OROZCO.

CODIRECTOR DE TESIS:

M. I. RIGOBERTO NAVA SANDOVAL.



MÉXICO, D.F.

2012

Agradecimientos

A mis padres y hermanas por su apoyo incondicional.

A mi familia y amigos.

A mi director de tesis, *M. I. Benjamín Valera Orozco*.

Al *M. I. Rigoberto Nava Sandoval*.

Al *Dr. Enrique Soto Castruita*.

Al *Centro de Ciencias Aplicadas y Desarrollo Tecnológico*, a la *Sección de Desarrollo de Prototipos* y al *Laboratorio de Metrología*.

A la *Facultad de Ingeniería*.

A la *Universidad Nacional Autónoma de México*.

Índice

Introducción

Relevancia y justificación.

Objetivo.

Alcances y limitaciones.

Organización de la tesis.

1.	Antecedentes	1
1.1	Medición de viscosidad	1
1.1.1	Definición de viscosidad.....	2
1.1.2	Métodos para la medición de viscosidad.....	6
1.1.3	Viscosímetros comerciales.....	12
1.2	Microcontrolador PIC18F4550	13
1.2.1	Especificaciones generales.....	14
1.2.2	Sistema de desarrollo.....	25
2.	Viscosímetro	28
2.1	Esquema general	28
2.2	Sistema electrónico.....	32
2.2.1	Subsistema de medición de desplazamiento angular	32
2.2.2	Subsistema de medición de temperatura	34
2.2.3	Subsistema impulsor de corriente directa	35
2.2.4	Algoritmo de medición y control	36
2.3	Software de operación	45
2.3.1	Descripción en el ámbito del usuario.....	46
2.3.2	Descripción en el ámbito del programador.....	49
2.4	Sistema mecánico	61

2.5	Procedimiento de calibración y medición	74
3.	Resultados	77
3.1	Resultados	77
3.2	Conclusiones.....	84
3.3	Trabajo a futuro	85
4.	Referencias	86
5.	Anexos.....	88

Introducción

Una de las propiedades físicas más importantes de los fluidos es la viscosidad, cuyo valor es constantemente utilizado como referencia en procesos experimentales e industriales como el mezclado, bombeo, extrusión, entre otros.

La viscosidad influye en la potencia necesaria para mover una superficie aerodinámica a través de la atmósfera. Responde a las pérdidas de energía asociadas con el transporte de fluidos en ductos, canales y tuberías. La viscosidad es una propiedad de fluido extremadamente importante en el estudio de flujos [8].

La viscosidad puede medirse de diferentes formas: mediante la ley de viscosidad de Newton [Ecuación.1], mediante la definición de viscosidad cinemática [Ecuación 4] o mediante métodos que requieren calibración con fluidos de viscosidad conocida. Es típico el empleo de equipo de laboratorio o de tipo industrial para medir las características de la mayoría de los fluidos, sin embargo, estos instrumentos pueden ser complicados y con un costo económico elevado para adquirirlos y posteriormente darles mantenimiento. Además de que puede ser necesario un operador con bastante experiencia.

Por ello, el viscosímetro propuesto en este trabajo tiene como características principales: basado en la definición de viscosidad dinámica, de tipo rotacional, método de medición simple, 16 bits que otorgan 65536 intervalos de medición de tiempo, comunicación USB que permita utilizarlo en prácticamente cualquier PC, código de control programado en PICC (código libre), desarrollo de interfaz gráfica (basada en Windows) práctica y sencilla de usar para el usuario.

El método propuesto en el presente proyecto de tesis consiste en integrar un sistema electromecánico, empleando técnicas de diseño mecánico y de sistemas digitales sobre la base de una PC y microcontrolador PIC18F4550 con interfase USB.

La figura 1 muestra un viscosímetro de cilindros concéntricos en donde un fluido depositado entre los cilindros fijo y rotatorio genera un par contrario al par de un motor de corriente directa.

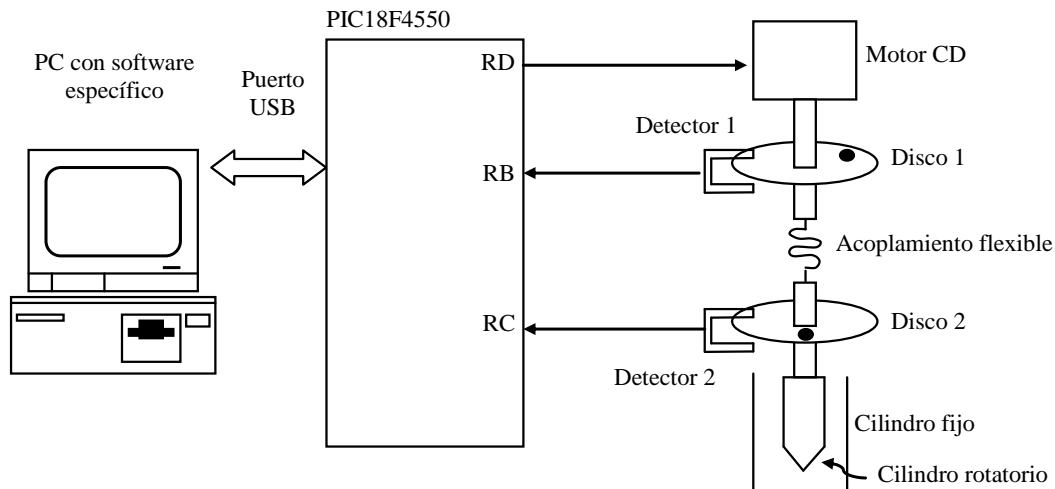


Figura 1. Esquema del prototipo electromecánico con interfase USB.

En la figura 1 el microcontrolador PIC18F4550 controla la velocidad del motor de corriente directa que impulsa al cilindro rotatorio mediante un acoplamiento flexible. El motor tiene soldado a su flecha una placa ranurada identificada como disco 1. El detector 1, fijo a la estructura, detecta el momento para el cual la ranura pasa sobre éste. De manera similar, el cilindro rotatorio tiene una placa ranurada identificada como disco 2 y su respectivo detector 2 registra el tiempo para el cual la ranura genera la señal sobre éste. En éste esquema, es posible determinar indirectamente el desplazamiento angular entre los discos 1 y 2 al registrar los tiempos en que se generan las señales de los detectores 1 y 2. La figura 2 muestra las señales que son enviadas al microcontrolador y el principio de medición del desplazamiento entre los discos.

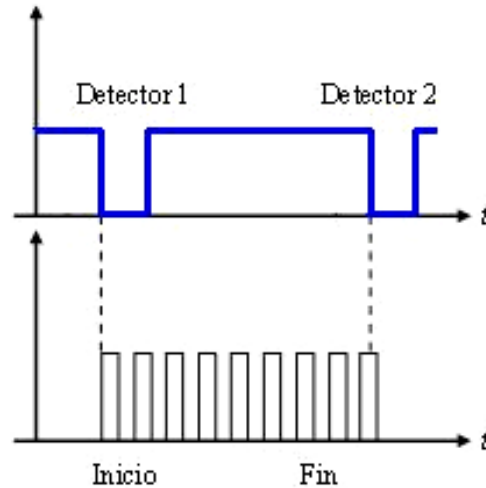


Figura 2. Diagrama de tiempos y principio de medición.

En la figura 2, el microcontrolador inicia un temporizador de 16 bits, en modo de contador, al registrar la interrupción que genera el detector 1. La señal del detector 2 genera una segunda interrupción que registra la cuenta alcanzada en el temporizador. La cuenta es proporcional al desplazamiento entre los discos 1 y 2 y a su vez es proporcional a la viscosidad. Con este esquema se piensa que un temporizador de 16 bits puede mejorar el desempeño en comparación con un convertidor analógico a digital de 10 bits.

Relevancia y justificación.

Este proyecto da continuidad a los desarrollos de tecnología original en el CCADET, con el objetivo de ofrecerse como una transferencia tecnológica realizando prototipos que cumplan con las necesidades específicas y que permitan ser utilizadas con una interfaz amigable.

Objetivo general.

Desarrollar un prototipo funcional para realizar una prueba de concepto del principio de medición de viscosidad propuesto.

Objetivo particular.

Desarrollar hardware y software para la implementación de un viscosímetro sobre la base de un sistema electrónico con microcontrolador PIC18F4550 y con interfase a una PC mediante el puerto USB.

Alcance y limitaciones.

El alcance del presente proyecto fue la construcción de un prototipo funcional cuyos resultados en cuanto a resolución y alcance de medición, pudieran ser mejores en comparación con el desarrollo previo reportado en [5] (revisar anexo 4).

Organización de la tesis.

El trabajo se compone de tres capítulos. En el primero se presentan fundamentos teóricos acerca de la viscosidad y los métodos comunes para su medida. Además se presenta una descripción breve del funcionamiento de los distintos tipos de viscosímetros. Finalmente se describen las características más importantes del microcontrolador PIC18F4550, así como una descripción general del sistema de control desarrollado para el proyecto. En el segundo capítulo se presenta el esquema general del equipo y cada uno de los subsistemas de medición y control. Después, se presentan las características y funcionalidades del software de operación desarrollado. También se mencionan aspectos del sistema mecánico y el procedimiento para realizar una medición. En el tercer capítulo se presentan los resultados de las mediciones realizadas con el prototipo para un par de fluidos y se realizan observaciones. Las últimas dos secciones son dedicadas a las referencias y anexos.

CAPÍTULO 1

1. Antecedentes

En esta sección se proporcionan las bases teóricas necesarias para abordar los capítulos subsecuentes, referentes al diseño y funcionamiento del viscosímetro propuesto en este trabajo. Primero se presentan definiciones de fluido, tipos de fluidos, viscosidad, técnicas de medición de viscosidad y el funcionamiento de algunos de los viscosímetros más comunes. También se abordan las características más importantes del microcontrolador PIC18F4550, como los registros de configuración y módulos de función especial. Finalmente se presentan generalidades del sistema de desarrollo electrónico.

1.1 Medición de viscosidad

La mecánica de fluidos es la ciencia que estudia el efecto de fuerzas aplicadas a los fluidos [7]. En contraste con un sólido, un fluido es una sustancia cuyas partículas pueden moverse fácilmente y cambiar su posición relativa [16]. Técnicamente, la diferencia entre ambos es que el sólido puede resistir un esfuerzo cortante con una deformación estática; mientras que el fluido no. Cualquier esfuerzo cortante aplicado a un fluido, no importa cuán pequeño sea, provoca el movimiento del fluido [2]. Un fluido es una sustancia que se deforma continuamente cuando se somete a un esfuerzo cortante, sin importar que tan pequeño sea ese esfuerzo cortante [15].

El grado de deformación del fluido ante un esfuerzo cortante está relacionado con la viscosidad, la cual es una propiedad del fluido [16].

Los procedimientos y equipos para medir viscosidad son numerosos. Algunos utilizan principios fundamentales de la mecánica de fluidos para expresar la viscosidad en sus unidades fundamentales. Otros, indican solo valores relativos de la viscosidad, usados para comparar fluidos diferentes. Los dispositivos para caracterizar el comportamiento del flujo de los líquidos se llaman *viscosímetros* o *reómetros* [1].

1.1.1 Definición de viscosidad

Viscosidad dinámica.

La viscosidad es aquella propiedad del fluido mediante la cual éste ofrece resistencia al esfuerzo cortante [15]. Es decir, determina la velocidad de deformación de un fluido cuando se le aplica un esfuerzo cortante. Conforme un fluido se mueve, dentro de él se desarrolla un esfuerzo cortante, cuya magnitud depende de la viscosidad del fluido [1]. Se define al *esfuerzo cortante*, denotado con la letra griega τ (tau), como la fuerza que se requiere para que una unidad de área de sustancia se deslice sobre otra. Entonces, τ es una fuerza dividida entre un área, y se mide en las unidades de N/m^2 (Pa) o lb/pie^2 .

La figura 1.1 ilustra el concepto de gradiente de velocidad en un fluido con el esquema de una capa delgada de fluido entre dos superficies, una de las cuales es estacionaria, en tanto que la otra está en movimiento. Una condición fundamental, cuando un fluido real está en contacto con una superficie de frontera, es que el fluido tenga la misma velocidad que ésta. Entonces, en la figura 1.1, la parte del fluido en contacto con la superficie inferior tiene una velocidad igual a cero, y aquélla en contacto con la superficie superior tiene velocidad v . Si la distancia entre las dos superficies es pequeña, entonces la tasa de cambio de la velocidad

con posición y es lineal. El *gradiente de velocidad* es una medida del cambio de velocidad, y se define como $\Delta v/\Delta y$. También se le denomina como *tasa cortante*.

El hecho de que el esfuerzo cortante en el fluido sea directamente proporcional al gradiente de velocidad se enuncia en forma matemática así:

$$\tau = \mu \left(\frac{\Delta v}{\Delta y} \right) \dots \dots \dots (1)$$

donde la constante de proporcionalidad μ (letra mu, en griego) se le denomina *viscosidad dinámica del fluido*, En ocasiones se emplea el término *viscosidad absoluta*.

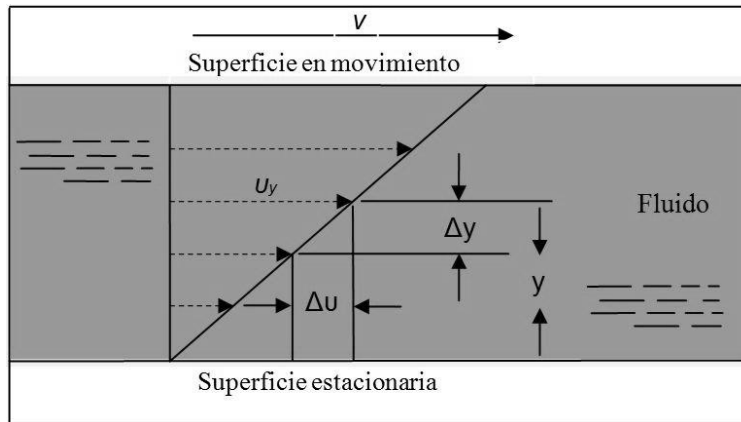


Figura 1.1. Concepto de gradiente de velocidad en un fluido.

La definición de viscosidad dinámica se obtiene al despejar a μ de la ecuación 1:

$$\mu = \frac{\tau}{\Delta v/\Delta y} = \tau \left(\frac{\Delta y}{\Delta v} \right) \dots \dots \dots (2)$$

Las unidades para μ se obtienen sustituyendo las del Sistema Internacional de Unidades (SI) en la ecuación 2:

$$\mu = \frac{N}{m^2} \times \frac{m}{m/s} = \frac{N \cdot s}{m^2} \dots \dots \dots (3)$$

Así, las unidades en el SI para μ pueden ser $N \cdot s/m^2$, $Pa \cdot s$ o $kg/m \cdot s$ (tabla 1.1).

Sistema de unidades	Unidades para la viscosidad dinámica
Sistema Internacional (SI)	$\frac{N \cdot s}{m^2}$, $Pa \cdot s$ o $\frac{kg}{(m \cdot s)}$
Sistema Tradicional de Estados Unidos	$\frac{lb \cdot s}{pie^2}$ o $\frac{slug}{pie \cdot s}$
Sistema cgs (obsoleto)	$poise = \frac{dina \cdot s}{cm^2} = \frac{g}{cm \cdot s} = 0.1 Pa \cdot s$ $centipoise = \frac{poise}{100} = 0.001 Pa \cdot s = 1.0 mPa \cdot s$

Tabla 1.1. Unidades para la viscosidad en distintos sistemas de unidades [1].

Viscosidad cinemática.

Muchos cálculos de la dinámica de fluidos involucran la razón de la viscosidad dinámica en la densidad del fluido. Por conveniencia, la viscosidad cinemática ν (letra nu en griego) se define como:

$$\nu = \frac{\mu}{\rho} \dots \dots \dots (4)$$

Debido a que μ y ρ son propiedades del fluido, ν también es una propiedad. Las unidades para la viscosidad cinemática en el SI (tabla 1.2) se obtienen con la sustitución de las unidades antes desarrolladas para μ y ρ :

$$\nu = \frac{m^2}{s} \dots \dots \dots (5)$$

Sistema de unidades	Unidades de la viscosidad cinemática
Sistema Internacional (SI)	$\frac{m^2}{s}$
Sistema Tradicional de Estados Unidos	$\frac{pie^2}{s}$
Sistema cgs (obsoleto)	$stoke = \frac{cm^2}{s} = 1 \times 10^{-4} \frac{m^2}{s}$ $centistokes = \frac{stoke}{100} = 1 \times 10^{-6} \frac{m^2}{s} = 1 \frac{mm^2}{s}$

Tabla 1.2. Unidades para la viscosidad cinemática en distintos sistemas de unidades [1].

Fluidos newtonianos y no newtonianos.

Cualquier fluido que se comporte de acuerdo con la ecuación 1 se le llama *fluido newtoniano*. La viscosidad μ sólo es función de la condición del fluido, en particular de su temperatura. La magnitud del gradiente de velocidad $\Delta v / \Delta y$ no tiene ningún efecto sobre la magnitud μ , como puede verse en la figura 1.2. A los fluidos más comunes como el agua, aceite, gasolina, alcohol, keroseno, benceno y glicerina, se les clasifica como newtonianos.

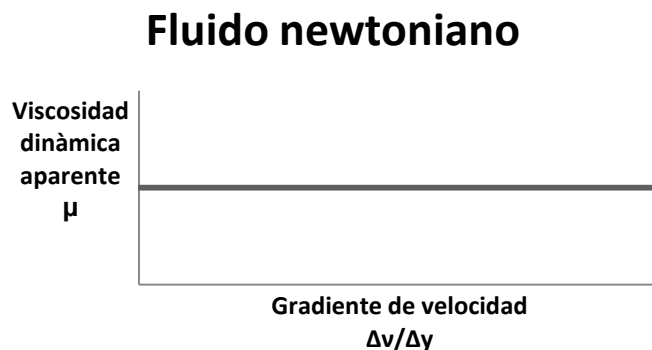


Figura 1.2. Viscosidad vs. Gradiente de velocidad.

En la figura 1.3, se muestra la relación esfuerzo cortante contra el gradiente de velocidad en un fluido newtoniano. En el caso de los fluidos newtonianos, la relación es lineal, entonces la pendiente es constante y por lo tanto la viscosidad es constante también. En el caso de fluidos no newtonianos, la pendiente es variable.

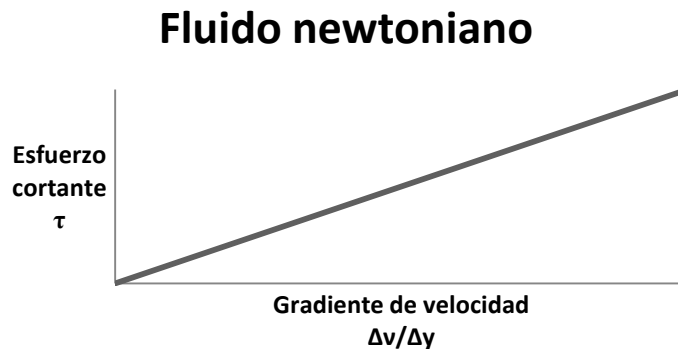


Figura 1.3. Esfuerzo cortante vs. Gradiente de velocidad.

A la inversa, a un fluido que no se comporte de acuerdo con la ecuación 1 se le denomina *fluido no newtoniano*. La viscosidad del fluido no newtoniano depende del gradiente de velocidad, además de la condición del fluido.

1.1.2 Métodos para la medición de viscosidad

La viscosidad es una propiedad que no se determina directamente sino que se calcula a partir de la medida de otras magnitudes y en base a las definiciones de viscosidad dinámica y cinemática. En la figura 1.4 se muestra el método general a seguir.

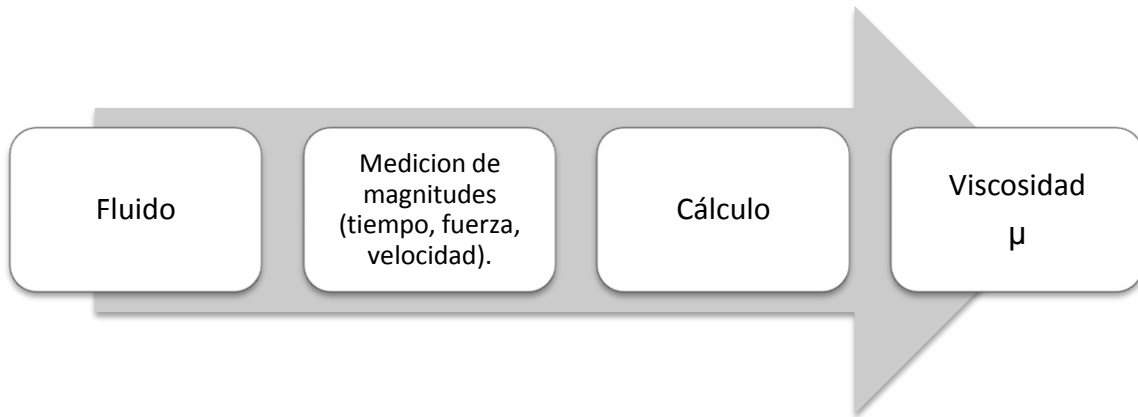


Figura 1.4. Metodología general para la medición de la viscosidad.

Los viscosímetros para fluidos newtonianos se pueden clasificar en tres grupos: Capilares, de Vibración y Rotacionales. A continuación se describen los más comunes.

Viscosímetro de tambor rotatorio.

El aparato que aparece en figura 1.5 mide la viscosidad por medio de la definición de viscosidad dinámica que se da en la ecuación 2, y que puede escribirse:

$$\mu = \frac{\tau}{\Delta v / \Delta y} = \tau \left(\frac{\Delta y}{\Delta v} \right) \dots\dots\dots (6)$$

El recipiente exterior se mantiene estático mientras que el motor acoplado al medidor hace girar el tambor rotatorio. El espacio Δy entre el tambor rotatorio y el recipiente es pequeño. La parte del fluido que está en contacto con éste es estacionaria, mientras que el fluido en contacto con la superficie del tambor interior se mueve a una velocidad similar a dicha superficie. Por tanto, en el fluido se establece un gradiente de velocidad conocido $\Delta v / \Delta y$. La viscosidad del fluido ocasiona en él un esfuerzo cortante τ que ejerce un torque de arrastre sobre el tambor rotatorio. El medidor detecta el arrastre e indica la viscosidad.

Los distintos modelos del aparato de prueba disponible en el mercado y los rotores diferentes para cada uno, permiten la medición de un rango amplio de viscosidades, desde 2.0 a 4.0×10^5 [mPa·s] ó hasta 400 [Pa·s]. Este probador se usa para fluidos muy variados: pintura, tintura, comida, derivados del petróleo, cosméticos y adhesivos.

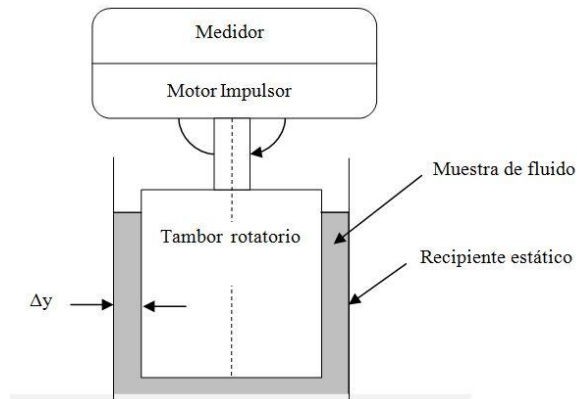


Figura 1.5. Esquema del viscosímetro de tambor rotatorio.

Viscosímetro de tubo capilar.

Conforme el fluido pasa por el tubo a velocidad constante, el sistema pierde alguna energía, lo que ocasiona una caída de presión que se mide por medio de manómetros (figura 1.6). La magnitud de la caída de presión se relaciona con la viscosidad del fluido.

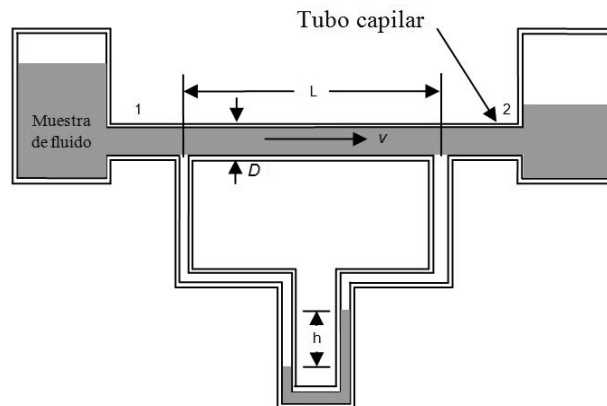


Figura 1.6. Esquema del viscosímetro de tubo capilar.

Viscosímetro de vidrio capilar estándar calibrado.

El tubo del viscosímetro se carga con una cantidad específica de fluido de prueba. Después de estabilizar la temperatura de prueba, se aplica una succión para hacer pasar el fluido por el bulbo, ligeramente por arriba de la marca superior del tiempo (figura 1.7). Se suspende la succión y se permite que el fluido circule por gravedad. La sección de trabajo del tubo es la capilar por debajo de la marca inferior del tiempo. Se registra el tiempo requerido para que el borde superior del menisco pase de la marca superior del tiempo a la inferior. La viscosidad cinemática se calcula con la multiplicación del tiempo de flujo por la constante de calibración del viscosímetro, la cual suministra el fabricante de éste.

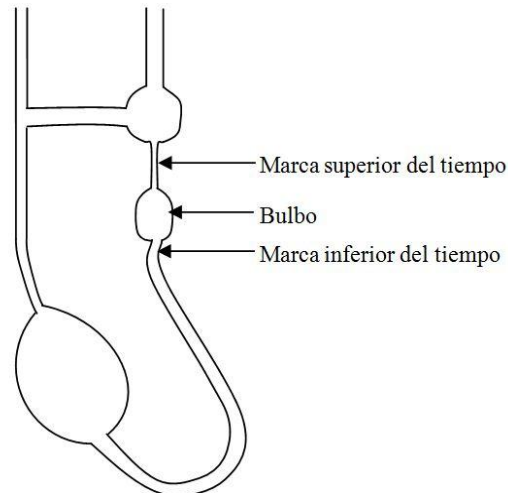


Figura 1.7. Esquema del viscosímetro de vidrio capilar.

Viscosímetro de bola que cae.

Conforme un cuerpo cae en un fluido solamente bajo la influencia de la gravedad (figura 1.8-a), acelerará hasta que la fuerza hacia abajo (su peso) quede equilibrada con la fuerza de flotación y la de arrastre viscoso que actúan hacia arriba. La velocidad que alcanza en ese tiempo se denomina *velocidad terminal*. En la figura 1.8-b se muestra un diagrama de cuerpo libre de la bola, donde w es el peso de la bola, F_b la fuerza de flotación y F_d la fuerza de arrastre viscoso sobre la bola. Cuando alcanza su velocidad terminal, la bola está en equilibrio.

Con estos datos es posible calcular la viscosidad dinámica haciendo uso de expresiones matemáticas. Es necesario que el fluido sea transparente, para que pueda observarse la caída de la bola. Sin embargo, algunos viscosímetros disponibles comercialmente disponen de un sensor que registra automáticamente la posición de la bola, de modo que es posible utilizar fluidos opacos.

El fabricante proporciona la calibración entre el tiempo que dura el recorrido y la viscosidad. Para que el viscosímetro se utilice con fluidos de viscosidades en un rango amplio, se dispone de varios tipos y tamaños de bola.

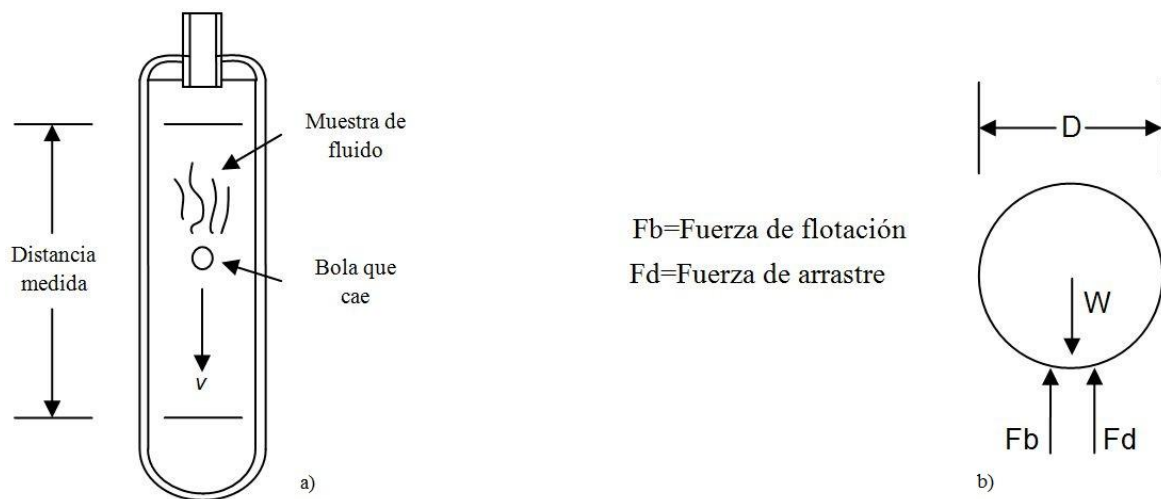


Figura 1.8. a) Esquema del viscosímetro de bola que cae. b) Diagrama de cuerpo libre de la bola que cae en un viscosímetro.

Viscosímetro de Saybolt universal.

La facilidad con que un fluido pasa por un orificio de diámetro pequeño es un indicador de su viscosidad. Éste es el principio en que se basa el viscosímetro de *Saybolt* [1]. Se utiliza un tubo capilar corto y se mide el tiempo que tardan 60cm^3 en fluir por el tubo bajo una cabeza decreciente. El tiempo en segundos es la lectura de *Saybolt*. Este aparato mide la viscosidad cinemática gracias a que la varía directamente con el tiempo [15].

La muestra de fluido se coloca en un aparato similar al que se ilustra en la figura 1.9. Después de que se establece el flujo se mide el tiempo que se requiere para reunir la cantidad en ml del fluido.

La relación aproximada entre la viscosidad y los segundos *Saybolt* se expresa mediante la ecuación 7.

$$\nu = 0.0022t - \frac{1.80}{t} \dots \dots \dots (7)$$

En la cual ν se encuentra en Stokes y t en segundos [15].

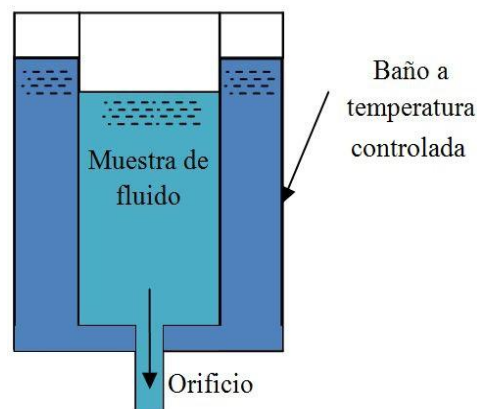


Figura 1.9. Viscosímetro de *Saybolt* universal.

A manera de resumen, a continuación se enumeran algunas ventajas y desventajas de los distintos viscosímetros:

- Viscosímetros de tambor rotatorio o cilindros coaxiales, cubren un amplio rango de medición y trabajan con fluidos newtonianos y no newtonianos, pero requieren de conjuntos mecánicos complejos
- Viscosímetros de tubo, pueden ser construidos a nivel planta piloto o en producción. Su limpieza resulta complicada [17].
- Viscosímetros de bola que cae, tienen un método de medición sencillo pero poca exactitud.

- Viscosímetros *Saybolt*, tienen una base teórica robusta. La medición depende de la precisión con que sea medido el tiempo

1.1.3 Viscosímetros comerciales

Los viscosímetros comerciales más comunes son del tipo capilares (tipo *Ubbelohde*, tipo *Cannon-Fenske*, Copas ISO, Copas *Ford*, copas *Zahn*, copas *Saybolt*), viscosímetros rotacionales y viscosímetros oscilatorios.

Algunas marcas comerciales de estos viscosímetros son: *Brookfield*, *Rheomat*, *Haake* y *Paar Physica*, entre otras. Los precios son variados, pero es común encontrarlos en el rango de los \$1,000 usd a \$6,000 usd.

Según la aplicación del viscosímetro, existen dos grupos: de laboratorio o de proceso. Los primeros son para ser utilizados dentro de condiciones de laboratorio; su exactitud de medición está entre $\pm 1.0\%$, con repetibilidad de $\pm 0.2\%$ [4], garantizan que la medición puede ser duplicada en cualquier lugar del mundo, siempre y cuando se use el mismo modelo y bajo condiciones controladas. En la figura 1.10 se muestra un par de viscosímetros de laboratorio comercializados por *Brookfield Engineering*.

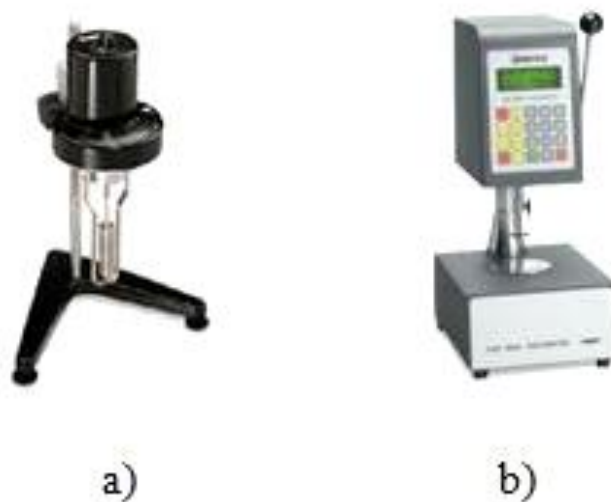


Figura 1.10. a) Modelo *Brookfield Dial Reading*. b) Modelo *CAP-2000+* (fotografías tomadas de [4]).

Los viscosímetros de proceso están destinados a ser parte de sistemas de control, en líneas de producción o bien procesos de los cuales no es posible extraer una muestra del fluido de trabajo. En la figura 1.11 se muestra un par de viscosímetros de proceso comercializados por *Brookfield Engineering*.

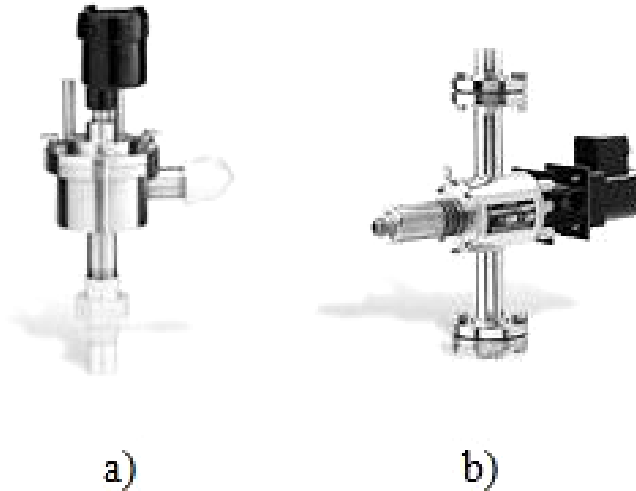


Figura 1.11. a) Modelo AST-100. b) Modelo TT-100 (fotografías tomadas de [4]).

En el sitio web de la empresa norteamericana *Brookfield Engineering Laboratories* [4] se puede acceder al catálogo completo de viscosímetros disponibles.

1.2 Microcontrolador PIC18F4550

Los microcontroladores son circuitos integrados que ofrecen un sistema completo de control configurable. La intención de usar un microcontrolador es la de ejecutar varias tareas al menor costo posible, ya que al ser un sistema embebido es capaz de resolver dichas tareas de forma autónoma. Por su versatilidad, los microcontroladores se encuentran actualmente en gran cantidad de dispositivos electrónicos de uso diario como monitores, sistemas de audio y electrodomésticos. El PIC18F4550 es un microcontrolador de propósito general, diseñado y producido por la empresa norteamericana *Microchip*. La razón principal por la que se eligió

este modelo en especial es que brinda soporte para el protocolo de comunicación USB (*Universal Serial Bus*) sin la necesidad de circuitos adicionales, es decir, directo del PIC a la PC.

1.2.1 Especificaciones generales

Las principales características del PIC18F4550 son las siguientes:

- Frecuencia máxima de operación de 48 MHz.
- Memoria Flash de 32 Kbytes.
- Memoria RAM de 2 Kbytes.
- Memoria EEPROM de 256 Bytes.
- 20 fuentes de interrupción.
- Encapsulado de 40 o 44 pines, de los cuales 35 son de entrada-salida divididos en 5 puertos (A, B, C, D, E).
- Un temporizador de 8 bits y tres de 16 bits.
- Módulo CCP de 16 bits para comparación, captura y generación de señal PWM de 10 bits.
- Módulo MSSP para comunicación I²C o SPI.
- Módulo USART.
- Módulo USB.
- 13 canales de conversión analógico-digital, con resolución de 10 bits.
- 2 comparadores.
- Voltaje de operación entre 2 y 5.5 volts.
- 75 instrucciones, 83 si se activan las instrucciones extendidas.
- Arquitectura de 8 bits.
- Oscilador interno configurable entre 32 kHz y 8 MHz.
- Prioridad en interrupciones.
- *Watchdog Timer*.
- Detección de voltaje Alto/Bajo.

Puertos del microcontrolador.

El microcontrolador PIC18F4550 tiene terminales de entrada/salida divididos en puertos, que se encuentran nombrados alfabéticamente A, B, C, D y E. Cada puerto puede tener hasta 8 terminales que, de forma básica, se comportan como una entrada/salida digital [9]. Según el tipo de encapsulado del PIC18F4550, cada puerto puede tener, además, asignado un bloque funcional: convertidor AD, USART, I2C, etcétera [10].

Considerando a los puertos como entradas y salidas digitales, los puertos se caracterizan por ser independientes, es decir, se puede programar cada terminal del puerto para que se comporte como una entrada o una salida digital [11].

En el caso del PIC18F4550 DIP40 (figura 1.12), éste tiene 5 grupos de puertos, cuya descripción general de sus pines y funciones es la siguiente:

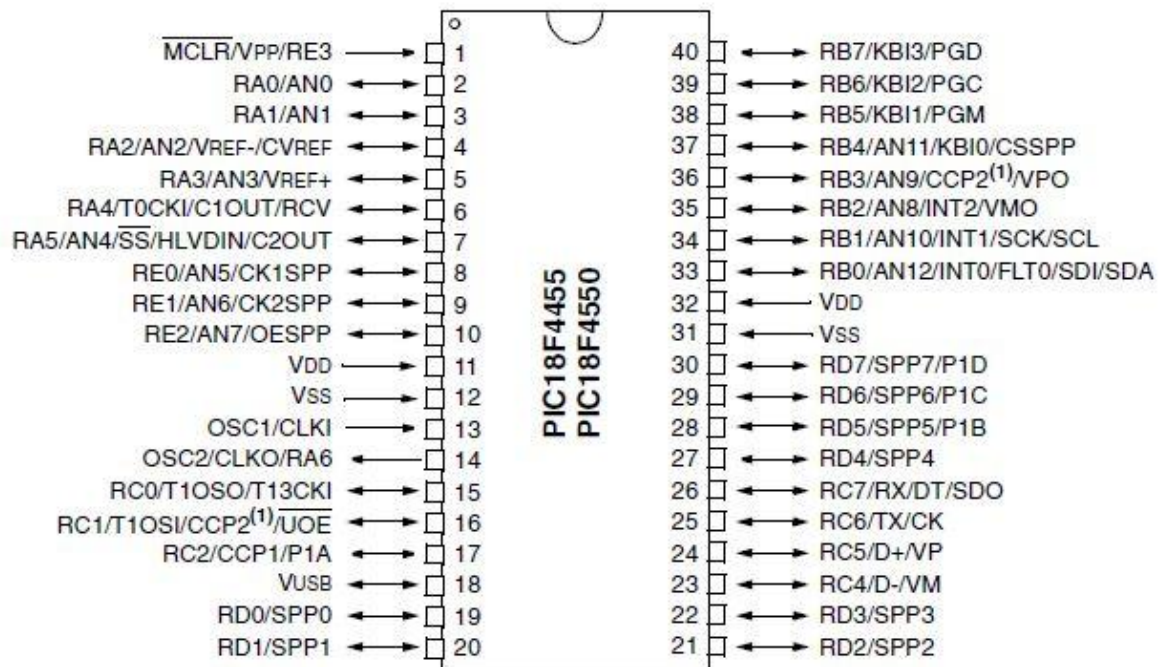


Figura 1.12. Diagrama de pines del PIC18F4550 (40 pines).

La información para la elaboración de las siguientes tablas fue consultada en la hoja de especificaciones [10].

Puerto A.

Es un puerto bidireccional de E/S. Sus funciones son como se muestran en la tabla 1.3.

Nombre del Pin	Descripción
RA0/AN0	E/S Digital/ Entrada analógica 0
RA1/AN1	E/S Digital/ Entrada analógica 1
RA2/AN2/V_{REF-}/CV_{REF}	E/S Digital/ Entrada analógica 2/ Entrada V _{REF-} A/D/ Salida V _{REF-} Comparador
RA3/AN3/V_{REF+}	E-S Digital/ Entrada analógica 3/ Entrada V _{REF+} A/D
RA4/T0CKI/C1OUT/RCV	E-S Digital/ Entrada de reloj Timer0/ Salida de Comparador 1/ Entrada RCV
RA5/AN4/SS/HLVDIN/C2OUT	E-S Digital/Entrada analógica 4/Entrada SPI/Entrada de detección voltaje alto/bajo/Salida de Comparador 2
RA6	Entrada del oscilador principal (OSC2)/salida de señal de reloj (CLKO)

Tabla 1.3. Descripción del puerto A.

Después del *reset*, todos los comparadores y convertidores Analógico/Digital configuran el respectivo pin en modo analógico. Para poder utilizar un pin en modo digital hay que desactivar la función analógica.

Puerto B.

Es un puerto bidireccional de E/S. Pueden activarse o desactivarse por software los *pull-ups* internos en todas las entradas. Si un pin se configura como salida, la resistencia *pull-up* correspondiente se desactiva automáticamente. Sus funciones son como se muestran en la tabla 1.4.

Nombre del Pin	Descripción
RB0/AN12/INT0/FLT0/SDI/SDA	E/S Digital / Entrada analógica 12 / Interrupción externa 0 / Entrada de fallo del ECCP / Entrada de datos del SPI / E/S de I ² C.
RB1/AN10/INT1/SCK/SCL	E/S Digital / Entrada analógica 10 / Interrupción externa 1 / Línea de reloj del SPI / Línea de reloj del I ² C
RB2/AN8/INT2/VMO	E/S Digital / Entrada analógica 8 / Interrupción externa 2 / Salida de datos del USB (VCMO)
RB3/AN9/CCP2/VPO	E/S Digital / Entrada analógica 9 / E/S de CCP2 / Salida de datos del USB (VPO)
RB4/AN11/KBI0/CSSPP	E/S Digital / Entrada analógica 11 / Interrupción por cambio en pin (KBI0) / Salida de CS del SSP (CSSP)
RB5/KBI1/PGM	E/S Digital / Interrupción por cambio en pin (KBI1) / Línea de programación (PGM)
RB6/KBI2/PGC	E/S Digital / Interrupción por cambio en pin (KBI2) / Línea de programación (PGC)
RB7/KBI3/PGD	E/S Digital / Interrupción por cambio en pin (KBI3) / Línea de programación (PGD)

Tabla 1.4. Descripción del puerto B.

Una característica importante de este puerto es el manejo de interrupciones externas. Las terminales RB0, RB1 y RB2 aceptan solicitudes de interrupción externa, por flancos de subida o bajada. Por su parte las terminales RB4 a RB7 pueden detectar un cambio en el nivel lógico (digital) de la señal de entrada, para generar una interrupción.

Puerto C.

Es un puerto bidireccional de E/S digital. Las terminales del puerto C son compartidas por dispositivos de entrada y salida (temporizadores, módulo CCP y módulo USB). Sus funciones son como se muestran en la tabla 1.5.

Nombre del Pin	Descripción
RC0/T1OSO/T13CKI	E/S Digital / Salida del oscilador del Temporizador 1 / Entrada del reloj externo Temp. 1 y Temp. 3
RC1/T1OSI/CCP2/ UOE	E/S Digital / Entrada del oscilador del Temporizador 1 / E/S del módulo CCP2 / Salida OE del transceptor del USB (UOE)
RC2/CCP1/P1A	E/S Digital / E/S del módulo CCP1 / Salida PWM del ECCP1 (P1A)
RC4/D-/VM	Entrada Digital / Línea menos del bus USB (D-) / Línea de entrada del USB (VM)
RC5/D+/VP	Entrada Digital / Línea más del bus USB (D+) / Línea de entrada del USB (VP)
RC6/TX/CK	E/S Digital / Salida de transmisión del EUSART (TX) / Línea de reloj del EUSART (CK)
RC7/RX/DT/SDO	E/S Digital / Entrada de recepción del EUSART (RX) / Línea de datos síncrona del EUSART (DT) / Salida de datos del SPI (SDO)

Tabla 1.5. Descripción del puerto C.

Después del *reset*, todos los pines quedan configurados como entradas.

Puerto D.

Puede funcionar como puerto bidireccional de E/S digital o como puerto SPP (*Streaming Parallel Port*). Esos pines tienen *buffers* de entrada cuando el módulo SPP está habilitado. Sus funciones son como se muestran en la tabla 1.6

Nombre del Pin	Descripción
RD0/SPP0	E/S Digital / Línea de datos del SPP (SPP0)
RD1/SPP1	E/S Digital / Línea de datos del SPP (SPP1)
RD2/SPP2	E/S Digital / Línea de datos del SPP (SPP2)
RD3/SPP3	E/S Digital / Línea de datos del SPP (SPP3)
RD4/SPP4	E/S Digital / Línea de datos del SPP (SPP4)
RD5/SPP5/P1B	E/S Digital / Línea de datos del SPP (SPP5) / Salida PWM del ECCP1 (P1B)
RD6/SPP6/P1C	E/S Digital / Línea de datos del SPP (SPP6) / Salida PWM del ECCP1 (P1C)
RD7/SPP7/P1D	E/S Digital / Línea de datos del SPP (SPP7) / Salida PWM del ECCP1 (P1D)

Tabla 1.6. Descripción del puerto D.

Pueden activarse o desactivarse por software los *pull-ups* internos en todas las entradas. Si un pin se configura como salida, la resistencia *pull-up* correspondiente se desactiva automáticamente.

Puerto E.

Es un puerto bidireccional de E/S digital. En el reset, RE2 a RE0 se configuran como entradas analógicas. Sus funciones son como se muestran en la tabla 1.7.

Nombre del Pin	Descripción
RE0/AN5/CK1SPP	E/S Digital / Entrada analógica 5 / Salida de reloj 1 del SPP
RE1/AN6/CK2SPP	E/S Digital / Entrada analógica 6 / Salida de reloj 2 del SPP
RE2/AN7/OESPP	E/S Digital / Entrada analógica 7 / Salida de habilitación del SPP
RE3	Línea de <i>reset</i> externo / Línea de programación (VPP)

Tabla 1.7. Descripción del puerto E.

Eventos y módulos principales del PIC18F4550.

Para este trabajo, los eventos y módulos más relevantes del PIC18F4550 (figura 1.13) son: interrupciones y temporizadores, convertidor AD, módulo CCP y módulo USB. A continuación se presenta un resumen de sus funciones y registros de configuración.



Figura 1.13. Imagen del PIC18F4550 (integrado de 40 pines).

Interrupciones.

Las interrupciones permiten a cualquier suceso interior o exterior interrumpir la ejecución del programa principal en cualquier momento. En el momento de producirse la interrupción, el PIC ejecuta un salto a la rutina de atención a la interrupción, previamente definida por el programador, donde se atenderá a la demanda de la interrupción. Cuando se termina de ejecutar dicha rutina, el PIC retorna a la ejecución del programa principal en la misma posición de la memoria del programa donde se produjo la interrupción.

El PIC18F4550 tiene múltiples fuentes de interrupción, así como niveles de prioridad que permiten que una interrupción con nivel de prioridad alto interrumpa a una con nivel de prioridad bajo, aun cuando ésta última se esté ejecutando.

Hay 10 registros de configuración que deberán ser empleados al programar interrupciones. Estos son:

- RCON.
- INTCON.
- INTCON2.
- INTCON3.
- PIR1, PIR2.
- PIE1, PIE2.
- IPR1, IPR2.

Temporizadores.

Los *TIMERS* o temporizadores son módulos integrados en el PIC que permiten realizar cuentas tanto de eventos internos como externos. Cuando la cuenta es interna se habla de temporización y cuando la cuenta es externa se habla de contador. Los *timers* están íntimamente ligados al uso de las interrupciones, pero no por ello se utilizan siempre de forma conjunta.

El PIC18F4550 tiene 4 módulos de temporizadores: TIMER0, TIMER1, TIMER2 y TIMER3. Estas son sus características principales:

TIMER0

- Puede funcionar con 8 o 16 bits.
- Registros de lectura y escritura (TMR1H y TMR1L).
- Pre-escala programable de 8 bits.
- Fuente de reloj elegible (interna o externa).
- Genera una interrupción al desbordarse.

TIMER1

- Puede funcionar con 16 bits.
- Registros de lectura y escritura (TMR1H y TMR1L).
- Fuente de reloj elegible (interna o externa).
- Genera una interrupción al desbordarse.
- Se puede reiniciar vía el módulo CCP.

TIMER2

- Registros de 8 bits para temporizador y periodo.
- Registros de lectura y escritura (TMR2 y PR2).
- Pre-escalas programables (1:1, 1:4 y 1:16).
- Post-escalas programables (1:1 hasta 1:16).
- Interrupción cuando se iguala el valor de TMR2 a PR2.

TIMER3

- Registro de 16 bit para temporizador o contador.
- Registros de lectura y escritura (TMR3H y TMR3L).
- Fuente de reloj elegible (interna o externa).
- Genera una interrupción al desbordarse.
- Se puede reiniciar vía el módulo CCP.

Convertidor Analógico-Digital.

El microcontrolador incorpora un módulo de conversión de señal analógica a señal digital, con 13 canales disponibles. Permite la conversión de una señal de entrada analógica a un correspondiente número digital de 10 bits. El módulo tiene 5 registros:

- ADRESH (*A/D Result High Register*).
- ADRESL (*A/D Result Low Register*).
- ADCON0 (*A/D Control Register 0*).
- ADCON1 (*A/D Control Register 1*).
- ADCON2 (*A/D Control Register 2*).

El registro ADCON0 controla la operación del módulo A/D. El registro ADCON1 configura las funciones de los pines del puerto. El registro ADCON2 configura la fuente de reloj, el tiempo de adquisición de la señal y el formato del resultado.

Módulo CCP.

Los módulos CCP permiten realizar tres funciones básicas basadas en el manejo de los temporizadores (TIMERS):

- Comparador: compara el valor del temporizador con el valor de un registro y provoca una acción en el PIC.
- Captura: obtiene el valor del temporizador en un momento dado, fijado por la acción de una terminal del PIC.
- PWM: genera una señal modulada en amplitud de pulso.

El PIC18F4550 tiene 2 módulos CCP. Los módulos CCP se comportan prácticamente igual. Cada módulo CCP posee un registro de 16 bits que puede utilizarse de las tres siguientes formas:

- I. Registro de 16 bits para capturar el valor del temporizador al producirse un evento (CAPTURA).
- II. Registro de 16 bits para comparar su valor con el del temporizador, pudiendo provocar un evento cuando se alcanza el valor contenido en este registro (COMPARADOR).
- III. Registro de 10 bits para el ciclo de trabajo de una señal PWM (PWM).

Cada módulo CCP está asociado con un registro de control (llamado genéricamente CCPxCON) y un registro de datos (CCPRx). El registro de datos, está contenido en dos registros de 8 bits: CCPRxL (parte baja) y CCPRxH (parte alta)

El modulo CCP utiliza los temporizadores 1, 2 o 3 dependiendo del modo seleccionado. Los temporizadores 1 y 3 se usan en los modos Captura y Comparación, mientras el temporizador 2 se usa en el modo PWM.

Módulo USB.

El PIC 18F4550 contiene un motor de interfaz serial (*Serial Interface Engine*, SIE) USB de alta y baja velocidad, que permite comunicación rápida entre cualquier

anfitrión USB y el microcontrolador PIC. La idea principal del USB es mejorar las técnicas de *plug-and-play*, es decir, permitir a los dispositivos conectarse y desconectarse sin necesidad de reinicio, configurándose automáticamente al ser conectados; además puede suministrar energía eléctrica al dispositivo conectado.

La operación del módulo USB es configurada y controlada mediante registros de control, los más relevantes son:

- UCON (*USB Control register*).
- UCFG (*USB Configuration register*).
- USTAT (*USB Transfer Status register*).
- UADDR (*USB Device Address register*).

La configuración del módulo USB en lenguaje C, es posible gracias a las librerías incluidas en el compilador CCS. Las librerías son:

- “pic18_usb.h”.
- “PicUSB.h”.
- “usb.c”.

Dichas librerías incluyen definiciones y funciones que configuran fácilmente el módulo USB, ahorrando gran cantidad de tiempo y trabajo al programador.

1.2.2 Sistema de desarrollo

Al utilizar un microcontrolador como base del sistema de desarrollo, es necesario tener acceso a los puertos que serán utilizados y programarlo de manera sencilla. Por ello, se decidió integrar en una sola tarjeta de desarrollo el circuito de programación y el circuito de control del viscosímetro, teniendo como resultado el diseño electrónico de la figura 1.14.

Debido a la necesidad de reprogramar continuamente el microcontrolador y evitar el desgaste físico del chip, se optó por el uso de un *BootLoader*, que es un pequeño programa que se graba una sola vez en el PIC usando un programador específico (quemador). El *BootLoader* permite programar el PIC posteriormente, sin la necesidad de retirar el chip de la tarjeta y colocarlo en un quemador. Para usar el *BootLoader* se debe contar con la comunicación USB (tarjeta-PC), la aplicación PICDEM y los drivers suministrados por la empresa Microchip, los cuales deben ser instalados previamente en la PC. La aplicación y los drivers se encuentran en el sitio web de Microchip [12] para su descarga de manera gratuita.

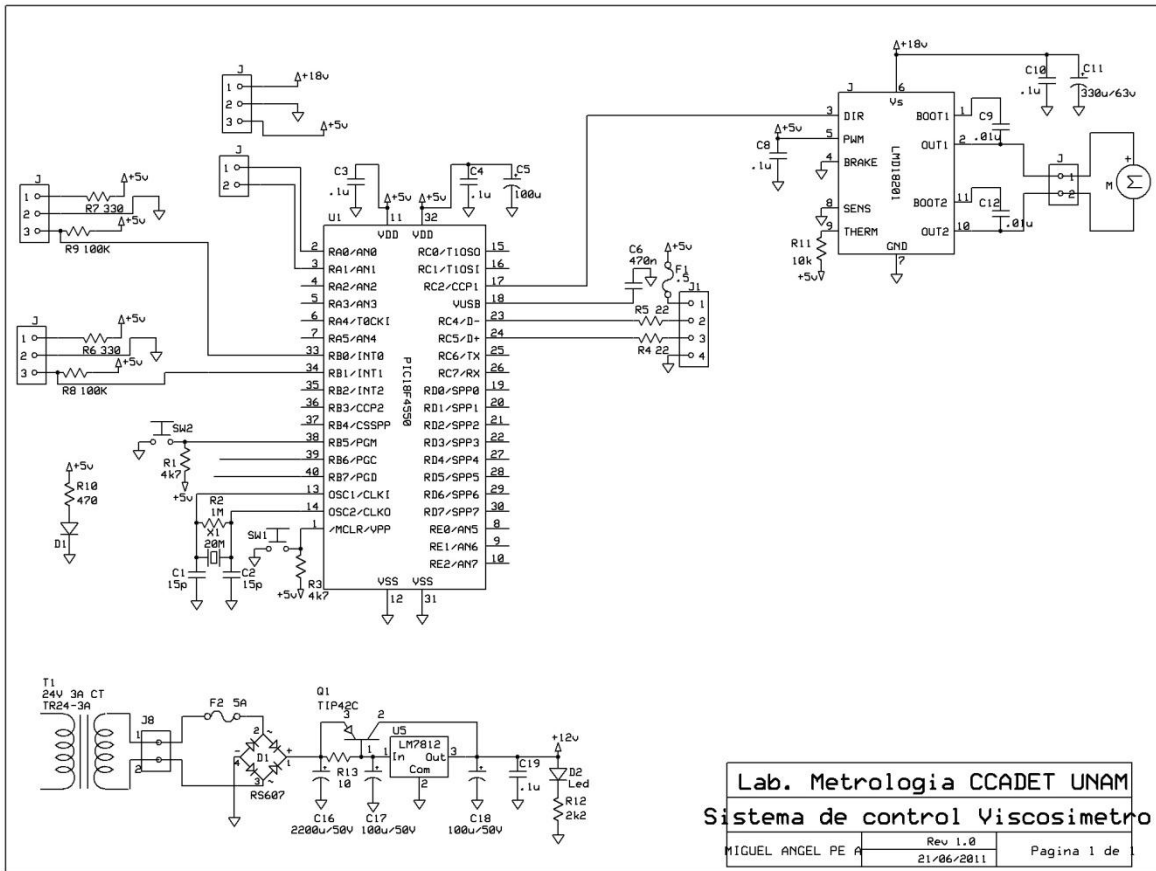


Figura 1.14. Diagrama esquemático del sistema de elaboración propia.

En cuanto al circuito de programación del microcontrolador, se incluyó conexión USB para comunicación con la PC y como medio de suministro de energía a la tarjeta (+5V), cristal de 20 MHz, botones de *reset* y de habilitación al modo *BootLoader*, *LED*'s indicadores y fusible de protección contra corriente excesiva.

Respecto al circuito de control del viscosímetro, se aborda con mayor detalle en el capítulo 2, aquí solo se menciona que se compone de los siguientes elementos principales: fuente regulada de voltaje (12V), controlador de motor de DC, 2 circuitos para opto-interruptor y entradas mediante borneras a los puertos A y B.

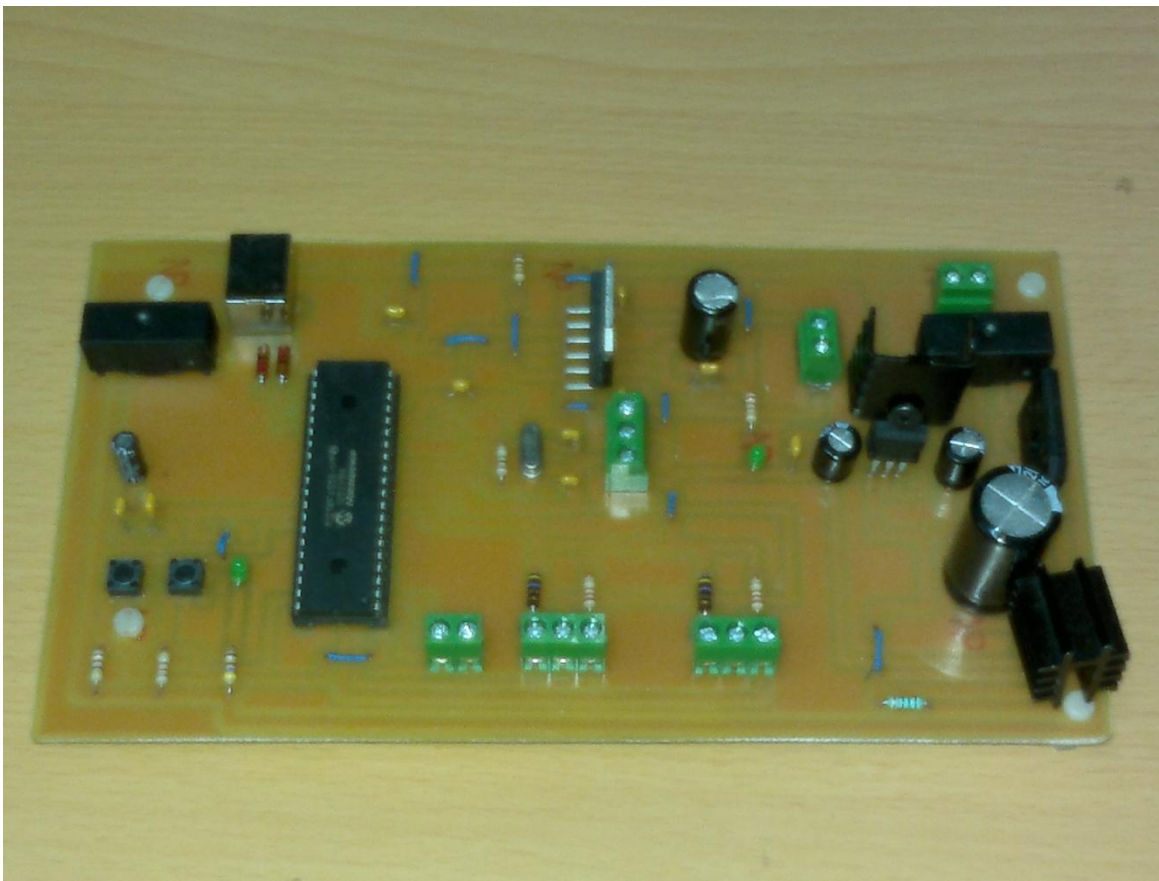


Figura 1.15. Imagen de la tarjeta de desarrollo de elaboración propia.

En la imagen (figura 1.15) se muestra el sistema de desarrollo usado para este proyecto; dicha tarjeta fue diseñada y fabricada en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) de la UNAM.

CAPÍTULO 2

2. Viscosímetro

En esta sección se abordan a profundidad los aspectos teóricos y de operación tomados en cuenta para el diseño y construcción del viscosímetro. Primero se presenta el método propuesto para la cuantificación de la viscosidad y los sistemas requeridos para llevarlo a cabo. Después, se presenta el sistema electrónico de medición y control, dividido en subsistemas, haciendo referencia a la función específica de cada uno de ellos. Además se incluye el algoritmo de medición y control. Se da la descripción del software de operación tanto en el ámbito del usuario como en el ámbito del programador. Para cerrar el capítulo, se presenta el diseño mecánico desde el concepto hasta el resultado final y se enumeran los pasos a seguir para realizar una medición.

2.1 Esquema general

El diseño del viscosímetro desarrollado toma como base teórica el principio de funcionamiento de los viscosímetros rotacionales (cilindros concéntricos), en los cuales se aplica directamente la definición de viscosidad dinámica vista en el capítulo anterior; por lo cual, el método cuantifica indirectamente la viscosidad del fluido, aprovechando la aparición de una fuerza de arrastre que ejerce el fluido sobre el cilindro de prueba (figura 2.1).

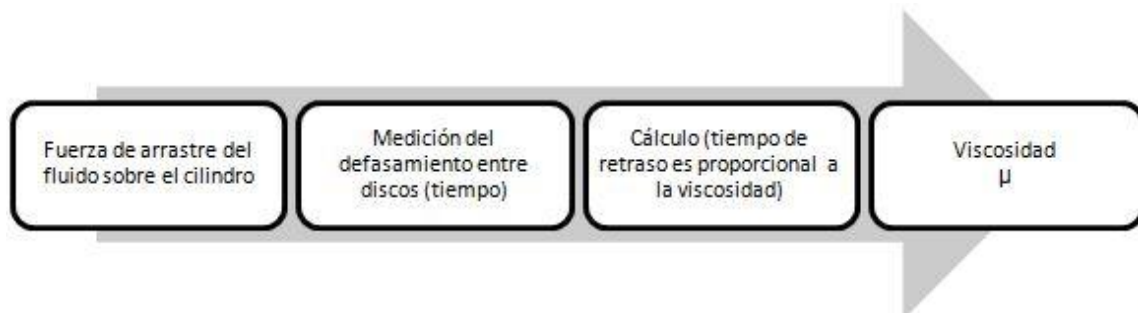


Figura 2.1. Método de medición propuesto.

La fuerza de arrastre provoca un desfase angular entre los discos que se ve reflejado en el tiempo de retraso de las señales generadas por los optointerruptores que registran el paso de las muescas en los discos. En la figura 2.2 se muestra un dibujo hecho en paquetería CAD, que ilustra los elementos mecánicos que conforman el sistema.

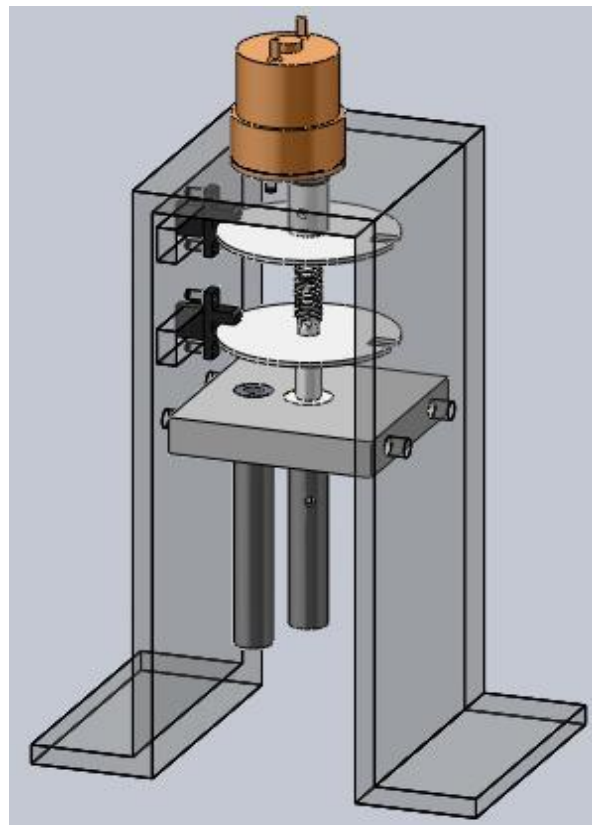


Figura 2.2. Vista isométrica del diseño mecánico del viscosímetro.

El viscosímetro se compone de un conjunto mecánico “eje-resorte-eje”. Dicho conjunto se encuentra acoplado a un motor de DC, que lo hace girar a velocidad constante. El cilindro de prueba es sumergido en la sustancia cuya viscosidad se requiere conocer. Al girar, el eje inferior es afectado por un esfuerzo cortante que hace que se “retrase” en el giro con respecto al eje superior. Dado que el resorte responde linealmente al esfuerzo; este retraso es proporcional a la resistencia que presenta el fluido a deformarse.

Una vez que los ejes superior e inferior giran a la misma velocidad pero con un desfase angular, se procede a cuantificar la viscosidad en función del tiempo de retraso. Para calcular el tiempo de retraso, se envían al PIC18F4550 un par de señales generadas por los opto-interruptores. Cuando el material obstruye al opto-interruptor, éste envía una señal de nivel lógico alto (5V), y en ausencia de material envía señal de nivel lógico bajo (0V). Por ello el diseño de cada eje con disco tiene una muesca rectangular como la que se indica en la figura 2.3.

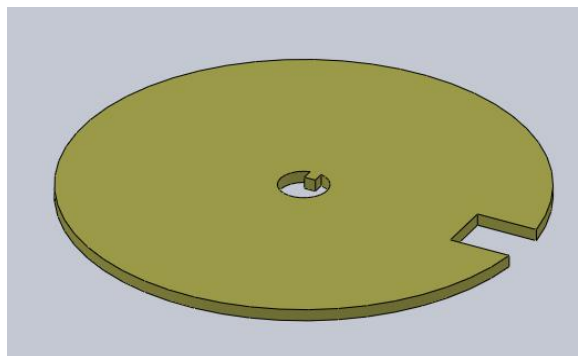


Figura 2.3. Disco con muesca.

Se utiliza un opto-interruptor para conocer cuando el primer disco “llega” y otro opto-interruptor para saber cuando lo hace el segundo disco. Esta diferencia de tiempos es cuantificada usando el modulo TIMER0 del PIC18F4550; el flanco de bajada de la señal del detector 1 dispara la cuenta del TIMER0, mientras que el flanco de bajada del detector 2 detiene la cuenta (figura 2.4); así es como se obtiene el tiempo requerido para el cálculo.

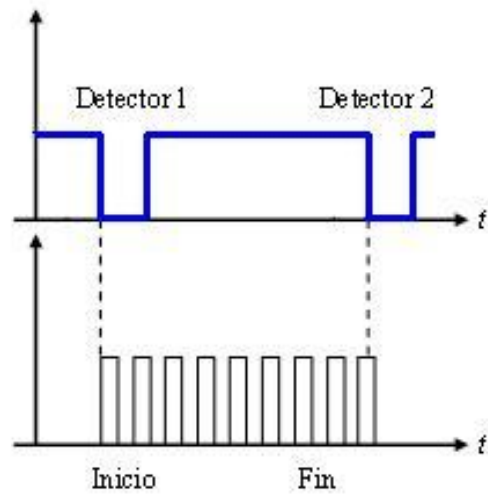


Figura 2.4. Diagrama de señales de control del TIMER0.

El prototipo (figura 2.5) resultante del diseño mecánico fue fabricado por personal del taller de la Sección de Desarrollo de Prototipos del CCADET.

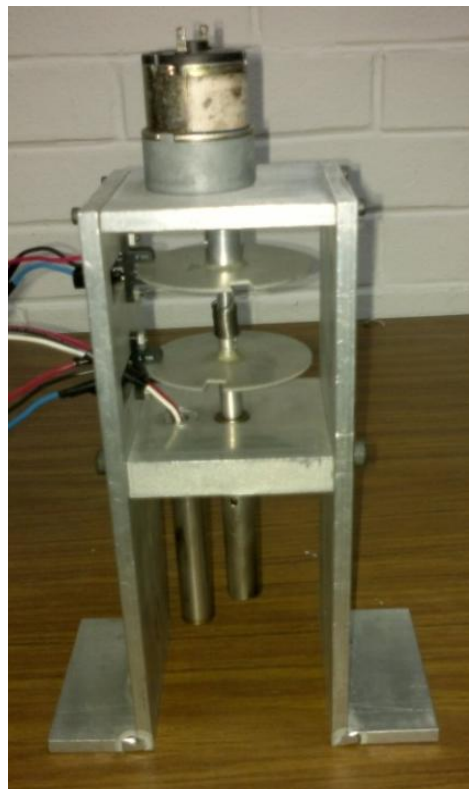


Figura 2.5. Imagen del prototipo fabricado.

2.2 Sistema electrónico

El sistema electrónico lleva a cabo las siguientes tareas: comunicación vía USB, control de velocidad del motor de CD, recepción de señales de los opto-interruptores, recepción de señal del sensor de temperatura y manejo de datos. El resumen de estas operaciones se aprecia en el diagrama de la figura 2.6.

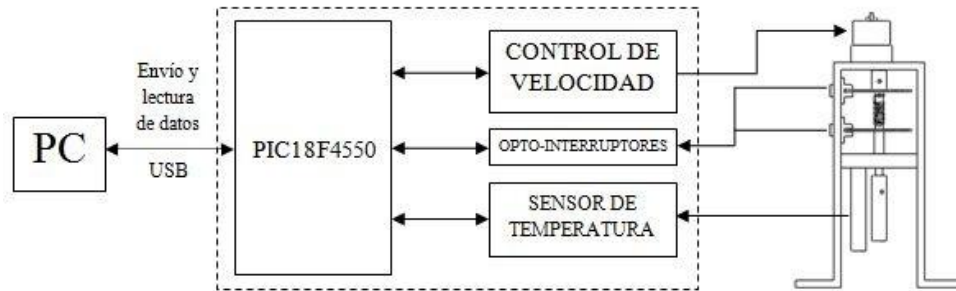


Figura 2.6. Diagrama de operación general.

El sistema está dividido en tres subsistemas:

1. Sistema de medición de desplazamiento angular (tiempo).
2. Sistema de medición de temperatura.
3. Sistema impulsor de corriente directa.

2.2.1 Subsistema de medición de desplazamiento angular

El sistema propuesto consiste en un conjunto mecánico que permite cuantificar la fuerza de oposición al movimiento que presenta el fluido de prueba. Al sumergirse el cilindro de prueba se presenta el desfase angular entre las muescas de los discos que giran a velocidad constante. El tiempo de desfase se cuantifica utilizando las señales generadas por los opto-interruptores.

El tipo de opto-interruptores utilizados es el *H21A1* [13], que consiste en un diodo infrarrojo emisor acoplado con un fototransistor dentro de un mismo empaque

plástico de color negro (figura 2.7). Este sistema de empaquetado está diseñado para optimizar la resolución mecánica, rechazo de luz ambiental, bajo costo y confiabilidad.

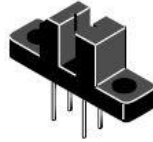


Figura 2.7. Fototransistor *H21A1*, imagen tomada de [13].

En la figura 2.8 se muestra el diagrama esquemático del componente. Cuando un material opaco se interpone en el espacio interrumpe la señal del emisor, cambiando el estado de la salida en el transistor.

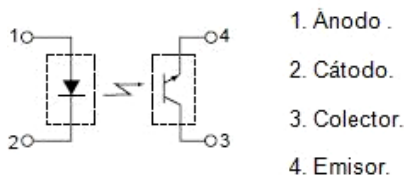


Figura 2.8. Diagrama esquemático del H21A1, imagen tomada de [13].

La configuración con la cual funcionan ambos H21A1 es la siguiente (figura 2.9):

- Señal Disco 1: Entrada por RB0.
- Señal Disco 2: Entrada por RB1.

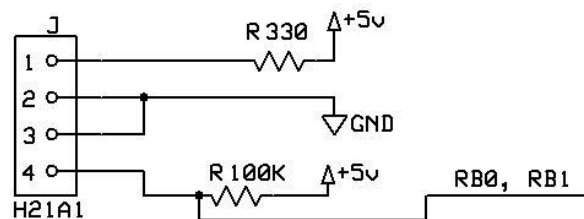


Figura 2.9. Diagrama de conexión de los H21A1.

En esta configuración, cuando se interpone un objeto opaco entre el emisor y receptor, a la salida el nivel lógico es “alto”, y “bajo” en caso de no interponerse algún objeto. Además, las señales de salida de los opto-interruptores son continuas. El flanco de bajada de cada señal genera una interrupción por puerto B. La primera interrupción inicializa el TIMER0, poniéndolo antes a cero. La segunda interrupción detiene el TIMER0 y guarda el valor del registro TMR0. El valor de ese registro representa el tiempo de desfase entre los discos, resultado de la fuerza de oposición del fluido y en consecuencia de la viscosidad del fluido.

2.2.2 Subsistema de medición de temperatura

La intención de incluir un sistema de medición de temperatura es únicamente la de contar con una medición de la temperatura del fluido de prueba al momento de realizar la caracterización del prototipo, ya que no se tiene forma de controlar esta variable. Sin embargo, el dato de temperatura es importante ya que al momento de consultar tablas en la literatura se observa que la magnitud de la viscosidad varía de manera importante con la temperatura.

Para este sistema se utiliza un sensor de temperatura LM35 [14], el cual está calibrado directamente en grados Celsius. El LM35 entrega una salida de voltaje proporcional a la temperatura en grados Celsius, con un factor de $10 \frac{mV}{^{\circ}C}$. Su bajo costo, rango de temperatura entre -55 y $+150$ $^{\circ}C$, rango de voltaje de alimentación entre $4V$ y $20V$ y dimensiones físicas del empaquetado tipo TO-92 (figura 2.10), permiten que sea compatible mecánica y electrónicamente con el prototipo.



Figura 2.10. Empaquetado del sensor de temperatura, imagen tomada de [14].

El circuito de aplicación del sensor se muestra en la figura 2.11.

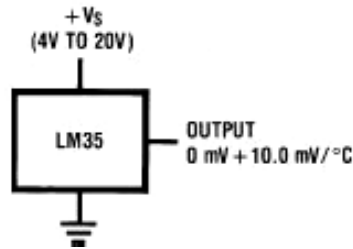


Figura 2.11. Diagrama de conexión del LM35, imagen tomada de [14].

La señal de salida del LM35 es recibida por el PIC18F4550 a través del canal analógico cero (figura 2.12).

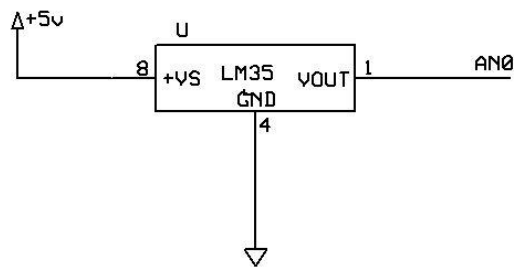


Figura 2.12. Diagrama de conexión del LM35.

En este caso el microcontrolador convierte el voltaje de entrada en el pin AN0 a un valor binario de 10 dígitos. Posteriormente envía ese valor por USB a la PC, donde el programa de Visual C++ ajusta ese número binario a un valor de temperatura en grados Celsius.

2.2.3 Subsistema impulsor de corriente directa

El sistema impulsor tiene como objetivo controlar la velocidad angular a la que gira el motor de corriente directa, que suministra el movimiento angular al eje para que pueda llevar a cabo la medición inmerso en el fluido.

En este caso, se utiliza el circuito de aplicación de la figura 2.13 para el control electrónico del motor de corriente directa, utilizando el driver LMD18201 [18]. Las principales características de este circuito integrado son la entrega de hasta 3 [A] de salida, voltaje de hasta 55 [V] y control de la dirección de giro por PWM.

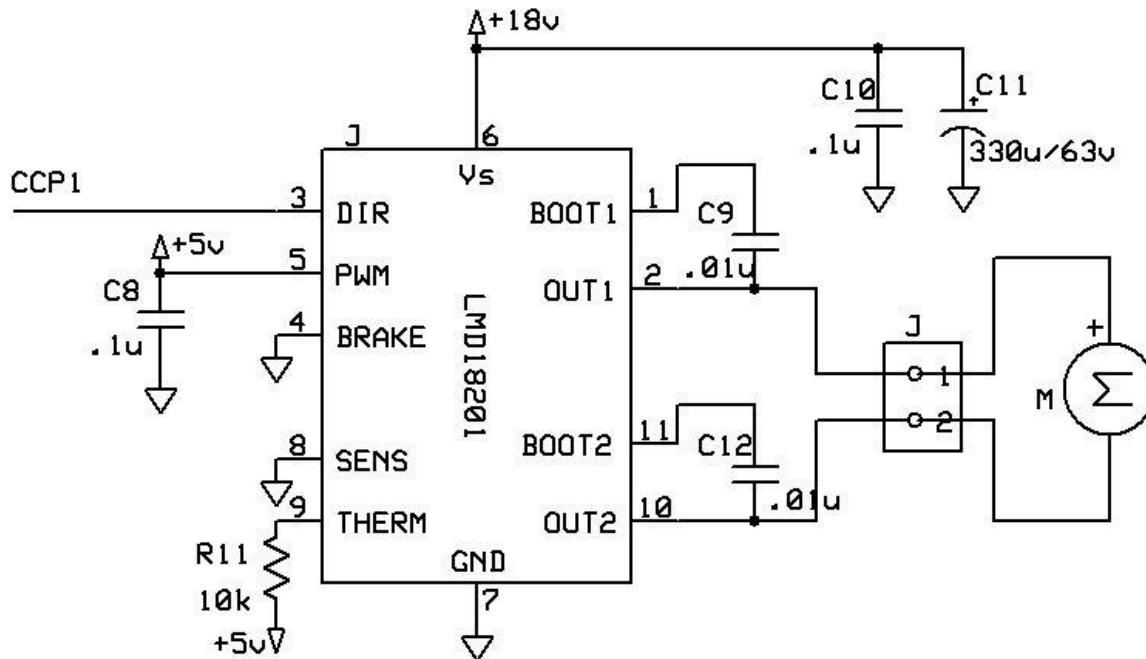


Figura 2.13. Circuito de control del motor de corriente directa.

La señal de control proviene del pin 17 (RC2/CCP1) del microcontrolador, el cual es la salida del modulo CCP1 que es generador de señal PWM.

2.2.4 Algoritmo de medición y control

El sistema se encarga de realizar la medición de tres valores (tiempo de retardo, revoluciones por minuto y temperatura) y controlar la velocidad del motor. Todas las operaciones son realizadas por el microcontrolador y a continuación se explica a detalle como éste realiza cada una de ellas.

Medición de tiempo.

La medición del tiempo de retardo, en realidad es el conteo de bits que realiza el Temporizador 0. Dicho conteo se realiza con base en el ciclo máquina (CM), es decir, el valor del temporizador aumenta un bit por cada CM (ecuación 8).

$$T_{CM} = \frac{4}{F_{OSC}} = \frac{4}{20 [MHz]} = 2 \times 10^{-7} [s] \dots \dots \dots (8)$$

Se debe establecer que cada lectura del temporizador debe hacerse antes de que se desborde, de lo contrario la cuenta se pierde. Por ello es importante conocer el tiempo de desbordamiento usando la ecuación 9.

$$T = T_{CM} \times Preescala \times (65536 - TMR0) \dots \dots \dots (9)$$

$$T = (2 \times 10^{-7} [s]) \times 256 \times (65536 - 0)$$

$$T = 3.355 [s]$$

Como el temporizador es de 16 bits, el rango de conteo va de 0 a 65536 bits. Finalmente, la configuración del temporizador 0 se muestra en la tabla 2.1.

Pre escala	256
# bits	16
Rango	0 – 65536
Tiempo de ciclo maquina (T_{CM})	$2 \times 10^{-7} [s]$
Tiempo de desbordamiento	3.355 [s]

Tabla 2.1. Configuración del temporizador 0.

Las mediciones del tiempo y de las RPM están muy relacionadas, ya que ambas dependen de la introducción del porcentaje de velocidad deseado y de la generación de las interrupciones. El tiempo de duración de la prueba es de 1 minuto.

La prueba comienza insertando el porcentaje de velocidad en el rango de 0 al 100% de ciclo de PWM por medio del programa de C++ (figura 2.15). Al oprimir el botón “Velocidad”, el dato es enviado al microcontrolador vía USB. El microcontrolador recibe los parámetros e inicia el movimiento del motor, y por consecuencia la generación de interrupciones.

La primer interrupción se encarga de inicializar el conteo del Temporizador 0, mientras la segunda interrupción detiene la cuenta y guarda el valor de los registros TMR0L y TMR0H usando la función *get_timer0()*. El código en lenguaje C de las interrupciones es el siguiente:

```
#INT_EXT//Atención a interrupción por cambio en RB0
ext_isr()//Función de interrupción
{
    set_timer0(0x00);
    bit_set(T0CON,7);//TIMER0 ON
    contador++;//indicador para el promedio
    pulso=0;//variable para la posición inicial
    eventos++;
}
#INT_EXT1 //Atención a interrupción por flanco de bajada en RB1
ext_isr1()
{
    bit_clear(T0CON,7);//TIMER0 DETENIDO
    t=get_timer0();
    acumulador=acumulador+t;//Se lleva la suma del tiempo 10 ciclos
    if(contador==10)//Al completarse 10 ciclos, se hace el promedio
    {
        promt=(int16)(acumulador/10);
        contador=0;//Se reinician variables
        acumulador=0;
    }
}
```

Las operaciones anteriores se repiten 10 veces para realizar un promedio de los valores del registro TMR0. En la figura 2.14 se muestra el proceso de medición de tiempo.



Figura 2.14. Diagrama de flujo del método de cuantificación del tiempo.

La lectura de tiempo que envía el PIC18F4550 a la PC es en realidad el promedio de 10 valores de tiempo acumulados. El objetivo de hacer un promedio es disminuir las variaciones entre lecturas, pues debido a que se tienen 16 bits de precisión el sistema tiene una sensibilidad alta. Para obtener el valor de tiempo desde la interfaz, se oprime el botón “Leer Timer”, con lo cual se desplegará el valor decimal de la cuenta en el “EditBox 2” (figura 2.15).

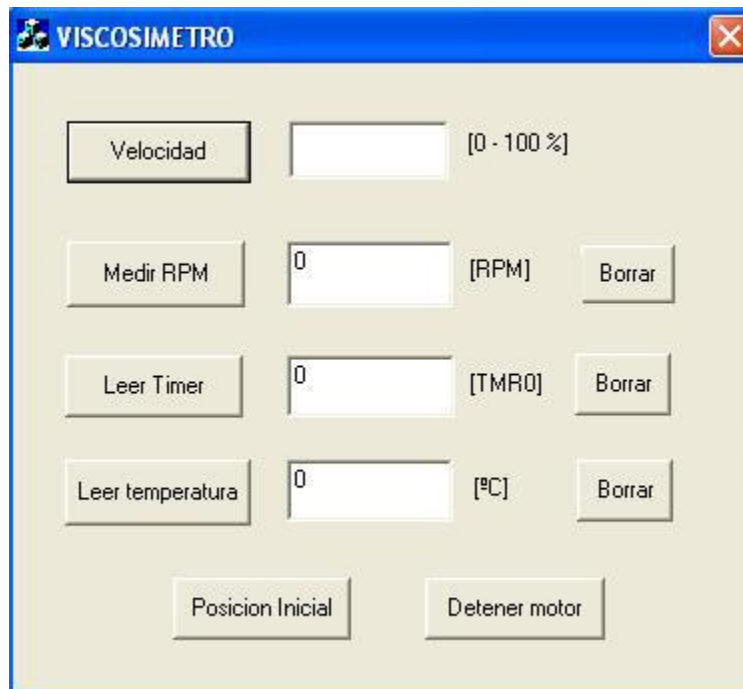


Figura 2.15. Programa de la PC.

Medición de RPM.

La medición de las RPM está muy relacionada con la medición del tiempo de retraso, ya que se lleva a cabo simultáneamente utilizando las interrupciones externas generadas. Simplemente se cuentan cuantas vueltas completa el disco 1 durante 1 minuto (revoluciones por minuto). Es decir, durante el minuto que dura la realización de la prueba, se realiza el conteo de pulsos generados por el disco 1; el valor se guarda en la variable “eventos”.

```
#INT_EXT//Atención a interrupción por cambio en RB0
ext_isr()
{//Función de interrupción
    set_timer0 (0x00);
    bit_set(T0CON,7);
    contador++;
    pulso=0;
    eventos++; //variable para el conteo de las RPM
}
```

Esta variable es enviada hacia la PC utilizando el siguiente código del PIC.


```

if (modo == 5)//Modo_medir RPM
{
    resultado=eventos;
    usb_put_packet(1, envia,1, USB_DTS_TOGGLE);//se envían 2 BYTES
}

```

Medición de temperatura.

Para la lectura del sensor de temperatura (LM35) se emplea el módulo ADC del PIC18F4550 configurado mediante el siguiente código.

```

#device adc=10
setup_adc(ADC_CLOCK_INTERNAL);//Fuente de reloj RC, se puede colocar el
voltaje de referencia
setup_adc_ports(AN0_ANALOG); //Canal 0 analógico

```

El rango de entrada del ADC se define de 0 a 5V pues el sensor entrega ese rango de salida. Con 10 bits la resolución del ADC se calcula con la ecuación 10.

$$V = \frac{V_{IN}}{2^N - 1} \dots \dots \dots (10)$$

$$\text{Voltaje de entrada: } V_{IN} = 5 \text{ [V]}$$

$$\text{Numero de bits: } N = 10 \text{ [bit]}$$

$$V = \frac{5 \text{ [V]}}{2^{10} - 1 \text{ [bit]}} = 4.8 \left[\frac{\text{mV}}{\text{bit}} \right]$$

El resultado anterior indica que por cada bit contabilizado por el microcontrolador, se tiene 4.8 [mV] provenientes del sensor. En la tabla 2.2 se muestra el resumen de la configuración del ADC.

Rango de entrada	0 – 5 [V]
Numero de bits	10
Resolución	4.8 [mV/bit]

Tabla 2.2. Características principales del ADC.

La lectura del ADC se realiza utilizando la función `read_adc()`, guardando el resultado en 2 variables de 8 bits cada una, `resultado` (parte baja) y `resultado2` (parte alta). Se utilizan dos variables por dos razones, primero porque el valor del ADC es de 10 bits, y segundo, porque la comunicación USB utilizada solo admite datos de 8 bits, tanto para envío como para recepción. El programa de C++ se encarga de juntar las partes alta y baja en la variable "`ResultadoPIC4()`"

Para obtener la lectura de temperatura del fluido se oprime el botón "Leer Temperatura" en la interfaz (figura 2.15). Al hacerlo el programa de Visual C++ enviará al microcontrolador la solicitud de lectura mediante el siguiente código:

```
BYTE SendData[1]; //Variable tipo Byte
SendData[0]=2;
myUSB.SendPacket(SendData,1); //Se envía el parámetro
m_Resultado3=(0.48)*(myUSB.ResultadoPIC4()); //Se recibe el resultado
UpdateData(FALSE);
UpdateData(TRUE);
```

El dato devuelto por el PIC18F4550 es un número binario de 10 dígitos equivalente a la temperatura, para hacer la conversión de bits a temperatura, se utiliza la ecuación 11. Debe tomarse en cuenta que la salida del sensor obedece a la relación $10 \left[\frac{mV}{^{\circ}C} \right]$ y que la resolución del ADC es de $4.8 \left[\frac{mV}{bit} \right]$.

$$T = \left(0.48 \left[\frac{^{\circ}C}{bit} \right] \right) * (ResultadoPIC4 [bit]) \dots \dots \dots (11)$$

A continuación se muestra un ejemplo numérico para comprender mejor este proceso. Si la salida de voltaje en el sensor es de 240 [mV] se tendría:

$$Lectura\ ADC = 50 [bit]$$

Para convertir este valor a voltaje.

$$V = 50[bit] \times 4.8 \left[\frac{mV}{bit} \right]$$

$$V = 240 [mV]$$

Y ajustando según la equivalencia del sensor, se obtiene la temperatura.

$$T = 240[mV] \times 0.1 \left[\frac{^{\circ}C}{mV} \right]$$

$$T = 24[^{\circ}C]$$

Finalmente, el valor de temperatura se muestra en el “EditBox4” convertido a grados Celsius.

Control de velocidad del motor DC.

Hay tres eventos principales relacionados con el control del motor DC que son activados desde la interfaz de C++. A nivel microcontrolador, se hace uso del modulo CCP1 en modo PWM (*pulse-width modulation*) que genera una señal modulada en amplitud de pulso basada en el temporizador 2 (TIMER2).

```
setup_timer_2(T2_DIV_BY_1,224,1); //PR2=224, T_pwm=225us
setup_ccp1(CCP_PWM); //CCP1 en modo PWM 10BITS
```

El primer evento es activado con el botón “Velocidad” y hace que el motor gire en sentido de las manecillas del reloj (*CW-clockwise*), regulando la velocidad en términos de porcentaje (0 al 100%), durante 1 minuto. Al completarse el minuto el motor se detiene automáticamente. A continuación se detalla el método.

Primero el usuario introduce el porcentaje de velocidad en la interfaz y oprime el botón “Velocidad” (figura 2.16). El microcontrolador recibe dos variables, “modo” y “param1”.

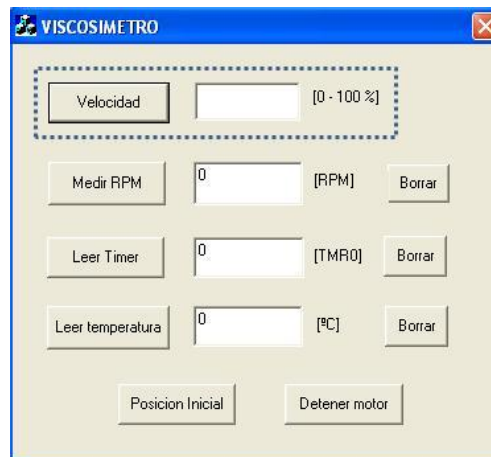


Figura 2.16. Sección de la interfaz para control de velocidad.

La variable “param1” asigna el valor del ciclo de trabajo. Como la resolución de salida de la señal PWM es de 10 bits, su rango de valores va desde 0 hasta 1023. Este valor se debe convertir de porcentaje de velocidad a un valor decimal entre 450 y 1023. Dicho rango de valores lleva al motor de estar detenido completamente (450) a girar a máxima velocidad (1023).

```
if (modo == 0)// MODO PWM CON VALOR DEFINIDO POR EL USUARIO
{
    i= param1;
    y=5.73*(i)+450;
    cont=0;
    eventos=0;
    bit_clear(T1CON,0);//TIMER1 DETENIDO
    set_timer1 (3036);//recarga del TMR1
    set_pwm1_duty(y);//a Duty de PWM
    bit_set(T1CON,0);//TIMER1 inicio
}
```

El segundo método se dispara al presionar el botón “Posición inicial”. Este establece una posición “cero” al sistema para garantizar que en todas las mediciones el motor inicie su movimiento desde la misma posición; mediante el modo 4 del programa del PIC.

```
if (modo == 4)// Modo_Posicion Inicial
{
    pulso=1;
    while(pulso==1)//pulso es igual a cero al presentarse INT_EXT
    {
```

```

        set_pwm1_duty(520); //a Duty de PWM
    }
    set_pwm1_duty(480); //a Duty de PWM
    ptimeL=0;
    ptimeH=0;
}

```

El código anterior también reinicia a cero las variables usadas en la medición del tiempo (*ptimeL*, *ptimeH*) para poder comenzar una nueva medición. Por último, con el botón “Detener motor” el PIC entra al modo 3 que solamente asigna el valor al ciclo de trabajo necesario para detener completamente el motor.

```

if (modo == 3) // Modo_detener motor
{
    set_pwm1_duty(450); //a Duty de PWM
    delay_ms(5);
}

```

2.3 Software de operación

El software de operación se divide en dos partes. La primera es una aplicación de Windows desarrollada con el compilador Visual 6.0 de Microsoft [19] en lenguaje de programación C++; el resultado es una aplicación del tipo “Cuadros de diálogo”, por medio de la cual el usuario tendrá interacción con el sistema. La segunda parte se trata del programa de control del PIC18F4550; éste consiste en código escrito en lenguaje C y compilado mediante la aplicación MPLAB IDE de la empresa Microchip [12], la cual a su vez utiliza librerías pertenecientes al compilador PIC C de la empresa CCS[20].

En cuanto las actividades específicas de cada programa, se puede señalar que el programa del microcontrolador está dedicado a llevar a cabo las operaciones de control de los sistemas de medición y de proveer los datos que le solicite la PC vía USB. Por su parte, el programa de Visual Studio lleva a cabo la recopilación de datos al momento que el usuario lo solicite mediante la interfaz desarrollada.

La relación entre ambos programas se limita a envío y recepción de datos. Para explicar más a detalle la interfaz con Visual Basic y el Microcontrolador es necesario entender que la comunicación USB cuenta con protocolos de comunicación definidos sobre el modo de operación en el que va a trabajar. Es decir, tanto la interfaz como la programación hecha en el microcontrolador realizan una comunicación *half-duplex* (en un solo sentido) una vez que se comunica con el programa de la PC, por ejemplo, este envía un carácter al Microcontrolador que más tarde enviará una respuesta de acuerdo a la acción que se tomó en este último.

2.3.1 Descripción en el ámbito del usuario

El programa consiste en una interfaz gráfica diseñada y programada en Visual 6.0. Al inicializar la aplicación *VISCOSIMETRO* se despliega el cuadro de diálogo que incluye 5 secciones:

1. Botón “Velocidad”, que envía el porcentaje de velocidad especificado por el usuario.
2. Botón “Medir RPM”, que solicita el valor de RPM de la última medición.
3. Botón “Leer Timer”, que inicia el proceso de lectura del temporizador.
4. Botón “Leer Temperatura”, que inician el proceso de lectura del sensor de temperatura.
5. Botones “Posición inicial” y “Detener motor”, que controlan el movimiento del motor de DC.

En la figura 2.17 se muestra completa la interfaz gráfica desarrollada.



Figura 2.17. Pantalla de la aplicación *VISCOSIMETRO*.

La forma de realizar la prueba es introducir el valor requerido de velocidad y oprimir el botón “Velocidad” para que el motor comience a girar (figura 2.18).

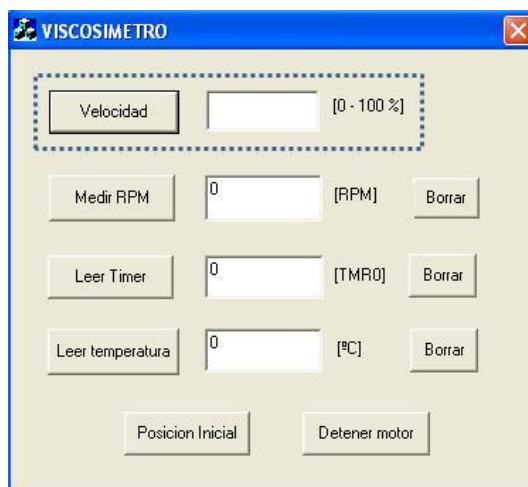


Figura 2.18. Sección de la aplicación destinada al movimiento del motor.

Posteriormente se debe esperar a que el sistema se detenga automáticamente, lo que ocurre 1 minuto después de haber presionado el botón “Velocidad”. Una vez transcurrido ese tiempo, se pide la lectura presionando el botón “Medir RPM” (figura 2.19).

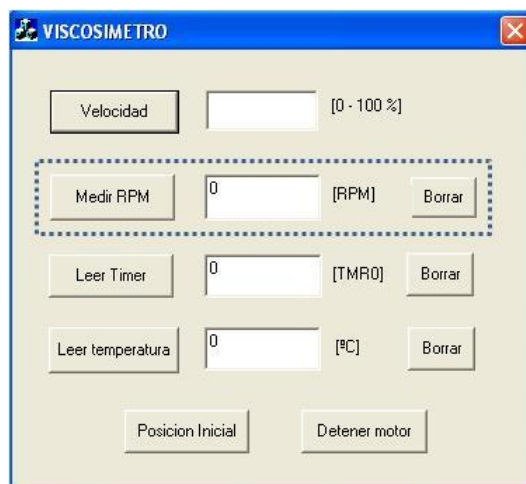


Figura 2.19. Sección de la aplicación destinada a la lectura de la velocidad.

Para la lectura de los bits de retraso se presiona el botón Leer Timer (figura 2.20).

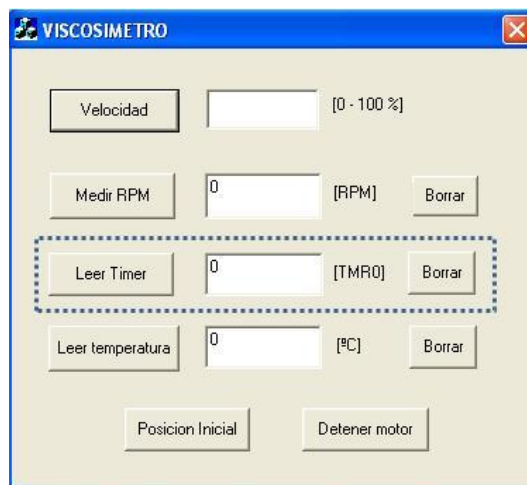


Figura 2.20. Sección de la aplicación destinada a la lectura de la velocidad real.

Adicionalmente, se incluye la medición de temperatura como un dato de referencia que será necesario para entrar a tablas de viscosidad (figura 2.21).

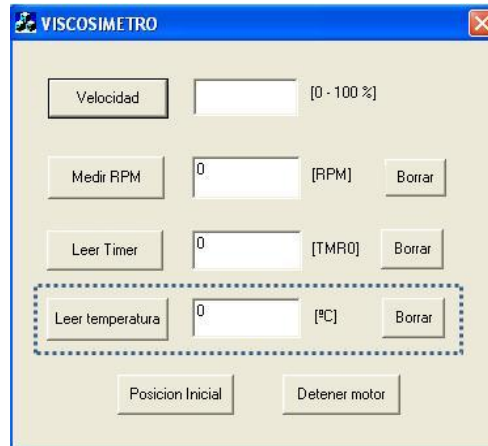


Figura 2.21. Sección de la aplicación destinada a la lectura de temperatura.

2.3.2 Descripción en el ámbito del programador

La aplicación fue desarrollada usando el compilador Visual C++ 6.0 de Microsoft. Es del tipo “cuadros de diálogo” y en base a MFC (*Microsoft Foundation Class*) [21], lo que le permite incluir Clases, librerías y archivos de cabecera. Para la aplicación *VISCOSIMETRO* se utilizaron archivos de cabecera y librerías que proporciona Microchip para facilitar la implementación de la comunicación USB en sus microcontroladores compatibles. En la figura 2.22 se muestran el mapa de clases del programa.



Figura 2.22. Clases del programa *VISCOSIMETRO*.

Clase CAboutDlg.

Es una clase que se genera automáticamente al momento de generar el proyecto. Se utiliza para mostrar la caja de diálogo “Acerca de”; sin embargo en este proyecto no es utilizada.

Clase CVISCOSIMETROApp.

Contiene las clases CVISCOSIMETROApp() y InitInstance(). La clase CVISCOSIMETROApp() no contiene código y la clase InitInstance() inicializa la apariencia de la caja de diálogo.

Clase CVISCOSIMETRODlg.

La clase CVISCOSIMETRODlg incluye vínculos con los archivos de cabecera *usb.h* y *usb2550.h*. Además contiene la definición de las variables utilizadas y los eventos presentes en el programa (figura 2.23).

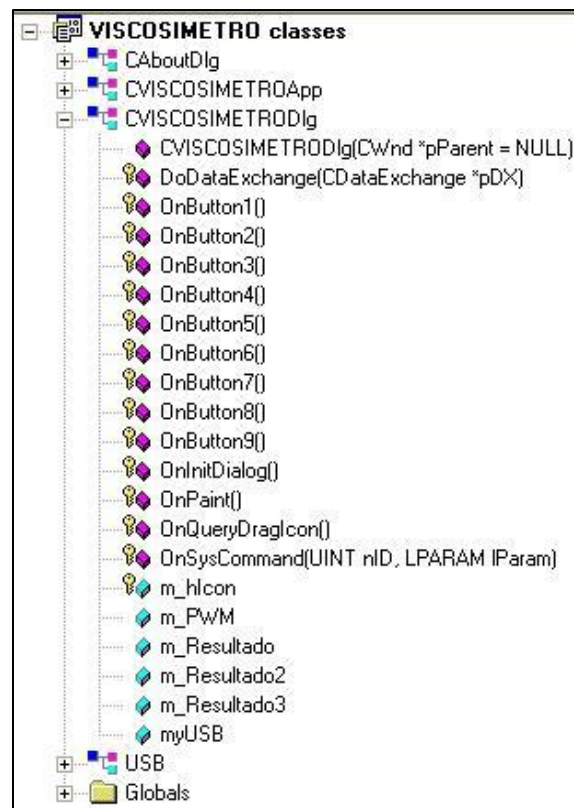


Figura 2.23. Elementos de la clase CVISCOSIMETRODlg.

DoDataExchange().

Intercambia y valida los datos del cuadro de diálogo.

OnButton1.

Es un método que se ejecutará cuando el usuario presione el botón “Velocidad”. Se encarga de enviar el valor de velocidad introducido en el campo correspondiente mediante el siguiente código.

```
void CVISCOSIMETRODlg::OnButton1()
{
    CString GuardaPWM;
    UINT resLB1, resHB1;
    int PWM=0;
    m_PWM.GetWindowText(GuardaPWM);
    PWM=atoi(GuardaPWM);
    resLB1=(PWM) & (0xFF);
    resHB1=PWM>>8;
    myUSB.ANDPIC(resLB1, resHB1);
    UpdateData(FALSE);
    UpdateData(TRUE);
}
```

OnButton2.

Es un método que se ejecutará cuando el usuario presione el botón “Leer Timer”. Se encarga de solicitar el valor del *TIMERO* mediante el siguiente código.

```
void CVISCOSIMETRODlg::OnButton2()
{
    BYTE SendData[1];
    SendData[0]=1;
    myUSB.SendPacket(SendData,1);
    m_Resultado2=myUSB.ResultadoPIC4();//Recogemos el resultado
    UpdateData(FALSE);
    UpdateData(TRUE);
}
```

OnButton3.

Es un método que se ejecutará cuando el usuario presione el botón “Leer Temperatura”. Se encarga de solicitar el valor de temperatura al microcontrolador mediante el siguiente código.

```
void CVISCOSIMETRODlg::OnButton3()
{
    BYTE SendData[1];
    SendData[0]=2;
    myUSB.SendPacket(SendData,1);
}
```

```

    m_Resultado3=(0.48)*(myUSB.ResultadoPIC4())
    UpdateData(FALSE);
    UpdateData(TRUE);
}

```

OnButton4.

Es un método que se ejecutará cuando el usuario presione el botón “Borrar”. Se encarga de borrar el valor de la caja de texto correspondiente a la parte de medición de revoluciones por minuto mediante el siguiente código.

```

void CVISCOSIMETRODlg::OnButton4()
{
    m_Resultado=0;
    UpdateData(FALSE);
    UpdateData(TRUE);
}

```

OnButton5.

Es un método que se ejecutará cuando el usuario presione el botón “Borrar”. Se encarga de borrar el valor de la caja de texto correspondiente a la parte de “Leer Timer” mediante el siguiente código.

```

void CVISCOSIMETRODlg::OnButton5()
{
    m_Resultado2=0;
    UpdateData(FALSE);
    UpdateData(TRUE);
}

```

OnButton6.

Es un método que se ejecutará cuando el usuario presione el botón “Borrar”. Se encarga de borrar el valor de la caja de texto correspondiente a la parte de “Leer Temperatura” mediante el siguiente código.

```

void CVISCOSIMETRODlg::OnButton6()
{
    m_Resultado3=0;
    UpdateData(FALSE);
    UpdateData(TRUE);
}

```

OnButton7.

Es un método que se ejecutará cuando el usuario presione el botón “Detener motor”. Se encarga de enviar el parámetro correspondiente al microcontrolador mediante el siguiente código.

```
void CVISCOSIMETRODlg::OnButton7()
{
    BYTE SendData[1];
    SendData[0]=3;
    myUSB.SendPacket(SendData,1);
    UpdateData(FALSE);
    UpdateData(TRUE);
}
```

OnButton8.

Es un método que se ejecutará cuando el usuario presione el botón “Posición inicial”. Se encarga de enviar el parámetro correspondiente al microcontrolador mediante el siguiente código.

```
void CVISCOSIMETRODlg::OnButton8()
{
    BYTE SendData[1]; //Se especifica que se enviará un byte al PIC.
    SendData[0]=4; //Se asigna el valor de 4 al byte
    myUSB.SendPacket(SendData,1); //se envía con el método SendPacket
    UpdateData(FALSE);
    UpdateData(TRUE); //Actualizamos la vista y datos de nuestro diálogo
}
```

OnButton9.

Es un método que se ejecutará cuando el usuario presione el botón “Medir RPM”. Se encarga de enviar el parámetro correspondiente al microcontrolador mediante el siguiente código.

```
void CVISCOSIMETRODlg::OnButton9()
{
    m_Resultado=myUSB.ResultadoPIC4();
    BYTE SendData[1];
    SendData[0]=5;
    myUSB.SendPacket(SendData,1);
    m_Resultado=myUSB.ResultadoPIC();
    UpdateData(FALSE);
}
```

```
        UpdateData (TRUE) ;  
    }
```

OnPaint.

Se encarga de dibujar la interfaz gráfica con los parámetros definidos, como dimensiones de ventana, posición de botones, etc.

OnQueryDragIcon.

El sistema llama a este método para obtener el cursor a mostrar en caso que el usuario arrastre la ventana minimizada.

OnSysCommand.

El marco de trabajo llama a esta función miembro cuando el usuario selecciona un comando del menú de control, o cuando el usuario selecciona el botón maximizar/minimizar.

Clase USB

La clase *USB* (figura 2.24) contiene los métodos que llevan a cabo la comunicación del PIC18F4550 con la PC. Esta clase se encontró disponible en el sitio web “muchotrasto” [22], el cual es un sitio dedicado a temas de robótica y tecnología, de libre acceso. La Clase USB fue desarrollada por terceros tomando como base la librería USB 2.0 Microchip, igualmente de libre uso.

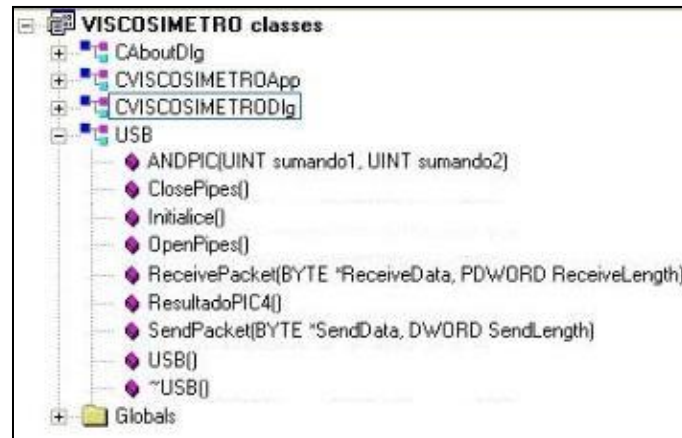


Figura 2.24. Métodos de la clase *USB*.

Los más importantes y su función se enlistan a continuación:

- *OpenPipes()*. Abre la comunicación USB-PC.
- *ClosePipes()*. Cierra la comunicación USB-PC.
- *SendPacket()*. Envía los datos al PIC.
- *ReceivePacket()*. Recibe los datos del PIC.
- *ResultadoPIC4()*. Recibe los datos del PIC utilizando el método *ReceivePacket()*, pero también integra los datos recibidos en un solo número de 16 bits. El siguiente código ilustra el método.

```
INT USB::ResultadoPIC4()//INT es de 16 bits en C++, en CCS es de 8 bits
{
    INT result = 0, result1=0, result2=0;
    byte* receive_buf = (BYTE*)malloc(2);
    DWORD RecvLength = 2;
    ReceivePacket(receive_buf, &RecvLength);
    result1 = receive_buf[0];
    result2 = receive_buf[1];
    result= result1 + result2*256;
    return result;
}
```

Programación en el microcontrolador PIC18F4550.

La herramienta que se utilizó para la programación fue el Compilador PCW [20]. El PCW emplea programación basada en lenguaje C aplicada a programar

microcontroladores PIC. Está constituido por librerías para manejar LCD, LCD gráfico, protocolos de comunicación, manejo de puertos, etc. En la actualidad el compilador PCW es una herramienta en desarrollo, al que se le han ido adaptando nuevas librerías, nuevos microcontroladores y ayudas.

El programa desarrollado para este trabajo es *viscosimetro.c*, y la descripción de sus segmentos de código más importantes se muestra a continuación.

Encabezados.

En esta sección se enlistan las librerías necesarias para efectuar el comienzo de la programación, como en todo código de programación se deben declarar las herramientas que van a estar disponibles; dentro de las escritas aquí están la librería de la familia del integrado, la resolución del ADC, las palabras de configuración cuya información es necesaria para algunos programadores de dispositivos y la frecuencia del procesador en ciclos por segundo (Hz).

```
#include<18F4550.h>
#deviceadc=10
#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN
#use delay(clock=20000000)
#usefast_io(B)
```

Las librerías necesarias para iniciar la configuración de las interrupciones para la comunicación USB en modo *Bulk* y el tamaño de paquetes de envío y recepción.

```
#defineUSB_HID_DEVICE FALSE
#defineUSB_EP1_TX_ENABLE USB_ENABLE_BULK
#defineUSB_EP1_RX_ENABLE USB_ENABLE_BULK
#defineUSB_EP1_TX_SIZE2//size to allocate for the tx endpoint 1 buffer
#defineUSB_EP1_RX_SIZE3//size to allocate for the rx endpoint 1 buffer
```

Las librerías necesarias para el funcionamiento de los controladores USB del microcontrolador.

```
#include<pic18_usb.h>
#include<PicUSB.h>
#include<usb.c>
```


Se crean identificadores para el correcto envío y recepción de datos.

```
#BYTE TRISD = 0xF95
#BYTE PORTD = 0xF83
#BYTE TRISB = 0xF93
#BYTE PORTB = 0xF81
#BYTE ADRESH = 0xFC4
#BYTE ADRESL = 0xFC3
#BYTETOCON = 0xFD5
#BYTEINTCON = 0xFF2
#BYTET1CON = 0xFCD

#define modo      recibe[0]
#define param1    recibe[1]
#define param2    recibe[2]

#define resultado      envia[0]
#define resultado2    envia[1]

#defineptimeLval[0]
#defineptimeHval[1]
```

Segmento Principal.

Esta sección de código tiene la función de iniciar las variables para el envío y recepción de datos en modo *half-dúplex* para la apertura y cierre de la línea de comunicación, las variables locales, la configuración del convertidor analógico-digital y la del timer0.

```
voidmain(void)
{
    int8 recibe[3]; //declaramos variables
    int8envia[2];
    int8 valor[2];
    int16i,q;
    contador=0;
    acumulador=0;
    t=0;
    promt=0;
    emprom=0;
    pulso=0;
    eventos=0;
    econt=0;
    y=0;
```

Inicialización de la comunicación USB.

```
usb_init();//inicializamos el USB
usb_task();//habilita periférico USB e interrupciones
```

```
usb_wait_for_enumeration();//esperamos hasta que el PicUSB sea
configurado por el host
```

Se establece la configuración de los puertos analógicos para la entrada de señal del sensor de temperatura.

```
setup_adc(ADC_CLOCK_INTERNAL);
setup_adc_ports(AN0_ANALOG); //Canal 0analógico
```

Se establece la configuración del Timer0.

```
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256); //Configuración timer0
bit_clear(T0CON,6);//TIMER0 de 16 bits
set_timer0 (0x00);
bit_clear(T0CON,7);//TIMER0 detenido
```

Se establece la configuración del Timer1.

```
setup_timer_1 (T1_INTERNAL | T1_DIV_BY_8);
set_timer1 (3036); //recarga del TRM1
enable_interrupts(INT_TIMER1);//habilitainterruptcion timer1
bit_clear(T1CON,0);//TIMER1 DETENIDO
```

Habilitación de las interrupciones.

```
disable_interrupts(INT_TIMER0);//DESHABILITA INT TMR0
enable_interrupts(int_ext);
enable_interrupts(int_ext1);//Habilita interrupción por puerto B
ext_int_edge(H_TO_L);//por flanco de bajada
enable_interrupts(GLOBAL);
```

Configuración de módulo PWM.

```
setup_timer_2(T2_DIV_BY_1,224,1); //PR2=224, Tpwm=225us
setup_ccp1(CCP_PWM); //CCP1 en modo PWM 10BITS
set_pwm1_duty(480); //a Duty de PWM
```

Una vez configurada las instrucciones necesarias para inicializar las acciones del PIC, este entra en un bucle infinito donde debe de esperar instrucciones provenientes de la computadora. Estos valores son recibidos y almacenados en el arreglo *recibe*.

```

while (TRUE)
{
    if(usb_enumerated())
    {
        if (usb_kbhit(1))
        {
            usb_get_packet(1, recibe, 3); //toma el paquete de
            tamaño 3bytes y almacenamos en recibe
        }
    }
}

```

El *carácter 0* recibido por parte del microcontrolador inicia la acción definida para la medición de viscosidad. Entra al *modo 0* que contiene el método de generación de la señal PWM.

```

if (modo == 0)// MODO PWM
{
    resultado = param1;
    resultado2 = param2;
    i=resultado+(resultado2*256);//El valor recibido se ajusta a 16bits
    y=4.59*(i)+ 450;
    cont=0;
    eventos=0;
    bit_clear(T1CON,0);//TIMER1 DETENIDO
    set_timer1 (3036);//recarga del TMR1
    set_pwm1_duty(i);
    delay_ms(5);
}

```

Después de inicializar el movimiento del motor se comenzarán a generar las interrupciones externas (*#INT_EXT*, *#INT_EXT1*) para efectuar el procedimiento de cálculo del promedio de tiempo de retardo. La interrupción *#INT_EXT* está dedicada a inicializar el timer0. La interrupción *#INT_EXT1* además de detener el temporizador 0, contiene el método para obtener el promedio de lecturas.

```

#INT_EXT//Atención a interrupción por cambio en RB0
ext_isr()
{//Función de interrupción
    set_timer0 (0x00); // Inicializa el valor del timer0
    bit_set(T0CON,7);//Arranca el timer0
    contador++;
    pulso=0;
    eventos++;
}
#INT_EXT1//Atención a interrupción por flanco de bajada en RB1
ext_isr1()

```

```

{
    bit_clear(TOCON,7);//TIMER0 DETENIDO
    t=get_timer0();
    acumulador=acumulador+t;//Se lleva la suma del tiempo 10
    ciclos
    if(contador==10)//Al completarse 10 ciclos, se hace el promedio
    {
        promt=(int16)(acumulador/10);
        contador=0; //Se reinician variables
        acumulador=0;
    }
}

```

A la llegada de un dato al PIC18F4550 corresponderá una de las acciones del “menú”. El carácter 1 indica que el usuario solicita la lectura del dato de tiempo, entonces el microcontrolador entra al modo correspondiente cuyo código es presentado a continuación.

```

if (modo == 1)//Modo_Promedio_de_tiempo
{
    ptimeL=promt&(0xFF);
    ptimeH=promt>>8;
    usb_put_packet(1, valor,2, USB_DTS_TOGGLE);
    delay_ms(5);
}

```

El carácter 2 recibido por parte del microcontrolador inicia la medición del sensor de temperatura registrado por el ADC.

```

if (modo == 2) // Modo_ADC CANAL 0
{
    set_adc_channel(0); //Habilitación canal 0
    delay_us(10);
    q=read_adc(); //Lectura canal0
    resultado = q&(0xFF);
    resultado2 = q>>8;
    usb_put_packet(1, envia,2, USB_DTS_TOGGLE);
}

```

El carácter 3 recibido por parte del microcontrolador inicia el paro del motor.

```

if (modo == 3)//Modo_detener motor
{
    set_pwm1_duty(450);//a Duty de PWM
}

```

```

    delay_ms(5);
}

```

El carácter 4 recibido por parte del microcontrolador lanza el modo “Posición inicial”.

```

if (modo == 4) // Modo_Posicion Inicial
{
    pulso=1;
    while(pulso==1)
    {
        set_pwm1_duty(520); //a Duty de PWM
    }
    set_pwm1_duty(480); //a Duty de PWM
    ptimeL=0;
    ptimeH=0;
}

```

El carácter 5 recibido por parte del microcontrolador inicia el modo “Medir RPM”.

```

if (modo == 5) // Modo_medir RPM
{
    resultado=eventos;
    usb_put_packet(1, envia,1, USB_DTS_TOGGLE); //se envían 2 BYTES
}

```

2.4 Sistema mecánico

Diseño conceptual.

Dado que el objetivo general del proyecto es el desarrollo de un prototipo para realizar una prueba de concepto y no el desarrollo de un producto para introducirse al mercado, los elementos mecánicos fueron elaborados con los materiales disponibles para este tipo de “proyectos semilla”, de bajo presupuesto, los cuales una vez probado el concepto podrían dar lugar al desarrollo de un producto terminado. Como base para diseñar el viscosímetro se tomaron ideas y bocetos relacionados con el desarrollado anteriormente [5] (revisar anexo 4). En la figura 2.25 se presenta la idea fundamental de un disco acoplado directamente a la flecha de un motor, que al girar transmitirá con un elemento elástico la fuerza necesaria para hacer girar a un segundo disco. El giro de cada disco será detectado por un opto-interruptor y contabilizado electrónicamente.

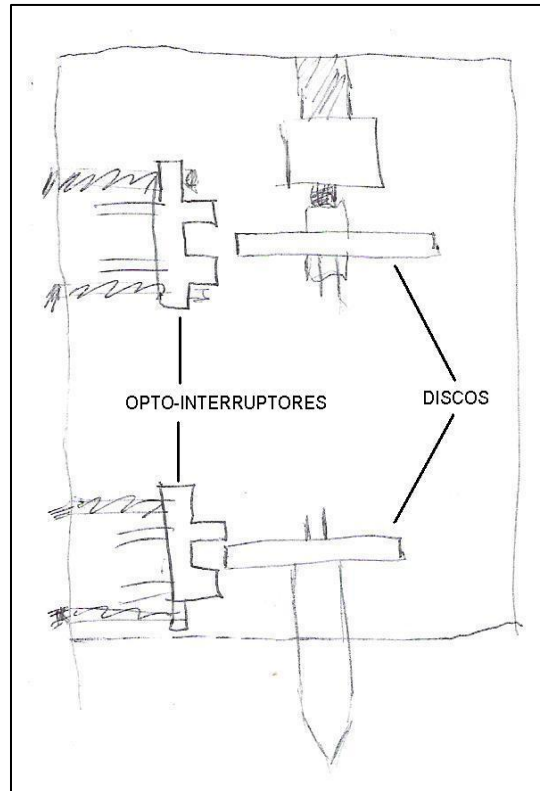


Figura 2.25. Idea básica de operación.

El eje inferior está acoplado al disco 2 y al elemento que será sumergido en el fluido. Para que pueda sostenerse y girar con la menor fricción posible se debe acoplar a un rodamiento adecuado (figura 2.26).

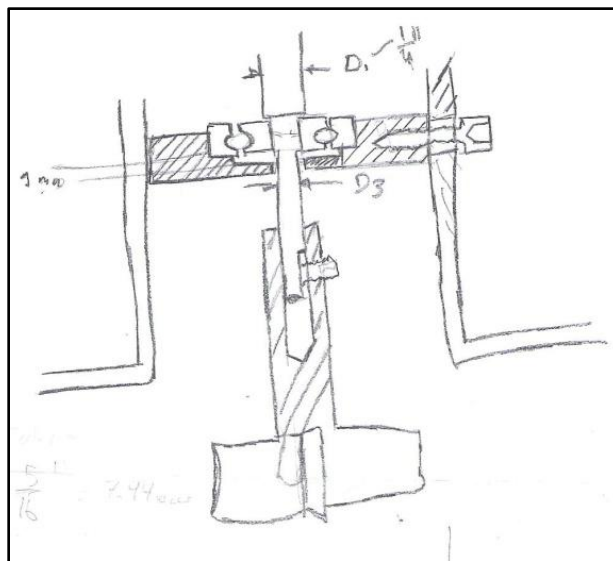


Figura 2.26. Boceto el eje inferior, base y rodamiento.

Después de considerar todas las ideas anteriores, se obtuvo el boceto final del concepto (figura 2.27), donde se identifican los siguientes elementos:

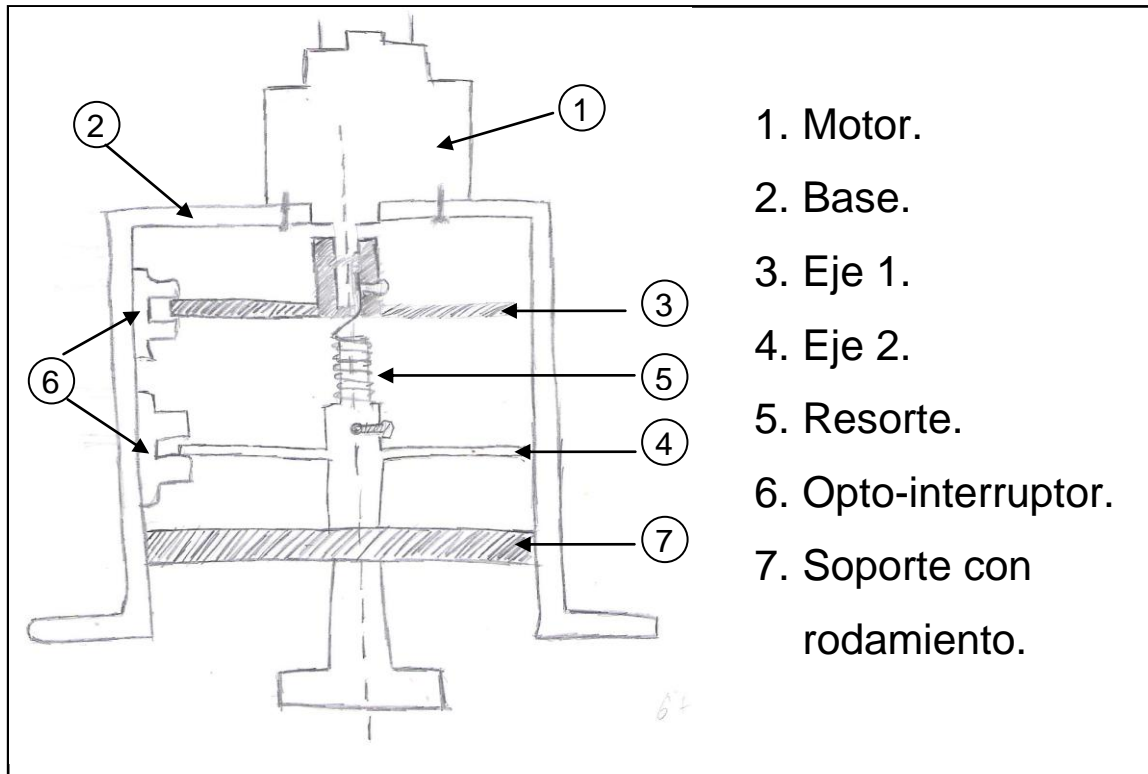


Figura 2.27. Idea completa del viscosímetro.

A partir de este diseño conceptual, se generó el diseño a detalle. El diseño del sistema mecánico tomó como restricciones de diseño las dimensiones físicas del motor, resorte y encapsulado de los opto-interruptores, debido a que se contaba con estos elementos previamente. A continuación se enlistan las características relevantes de cada uno de los componentes.

Motor.

Se trata de un motor KYSAN (1151243) con reductor (figura 2.28) que funciona a 12 [V], entregando 128 revoluciones por minuto en vacío y un consumo de corriente de 80 [mA]. La hoja de especificaciones del motor se encuentra en el anexo 3. En la tabla 2.3 se muestran las especificaciones más relevantes para este trabajo. El motor usado en el proyecto fue suministrado por la Sección de

Desarrollo de Prototipos (SDP) del CCADET, que cuenta con algunos motores de este tipo comprados para la realización de diversos mecanismos motorizados.



Figura 2.28. Motor de DC.

Voltaje nominal	12 [V]
Velocidad nominal a 12[V]	128 [rpm]
Consumo sin carga a 12[V]	80 [mA]
Consumo eje frenado a 12 [V]	1000 [mA]
Par máximo (detenido)	1.1 [kgf·cm]
Diámetro máximo	38 [mm]
Diámetro eje	6 [mm]

Tabla 2.3. Especificaciones del motor.

Elemento elástico.

Su función es la de permitir el desfase entre el giro del eje del motor y el eje de giro del cilindro de prueba y transmitir la fuerza rotativa del motor a dicho cilindro. Sus dimensiones se muestran en la figura 2.29. Este elemento fue seleccionado de una gama de resortes por el valor de k necesario para que exista el desplazamiento necesario entre el disco superior y el inferior. Para otros valores

de k el instrumento debe ser calibrado nuevamente. Para sustancias muy espesas k tiene que ser mayor y viceversa.

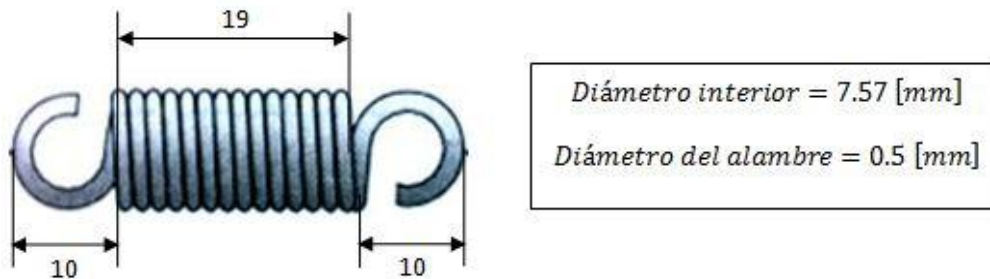


Figura 2.29. Dimensiones del resorte en mm.

Opto-interruptores.

Como en el caso del resorte y del motor, en la figura 2.30 se muestran las dimensiones del elemento. Este elemento se eligió porque ya se tiene la experiencia en el empleo del mismo, ampliamente usado en el desarrollo de otros prototipos.

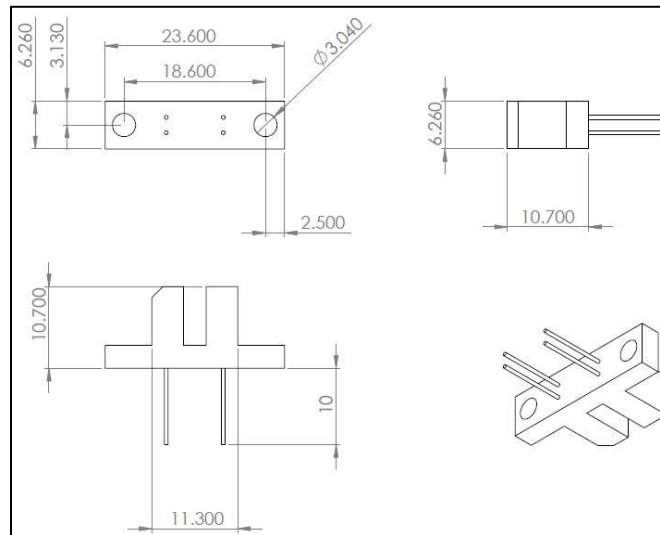


Figura 2.30. Planos del opto-interruptor (medidas en [mm]).

Diseño a detalle.

Para el diseño a detalle del prototipo se partió del concepto plasmado en la figura 2.27, pero incluyendo las dimensiones físicas del motor, opto-interruptores y

resorte como restricciones de diseño. Tomando como base esas medidas así como las características del material con que serían fabricadas las piezas y el proceso de maquinado necesario para cada una, se procedió al diseño de cada elemento. Todas las piezas fueron fabricadas de aluminio para evitar corrosión y no requerir pintado posterior. Las dimensiones de cada pieza y el conjunto final se muestran a continuación. Los planos a detalle no se incluyen en este trabajo, cualquier consulta o asunto acerca de los mismos favor de comunicarse con la SDP del CCADET.

Disco.

Se fabricaron 2 piezas. Cada una unida posteriormente a su respectivo eje. En el agujero se ha diseñado una pequeña lengüeta para transmisión del giro entre eje y disco, mientras que en el diámetro exterior se ha dejado una ranura, la cual permite el paso de luz en el opto-interruptor H21A1 (figura 2.7). El espesor de 1.6 [mm] es una medida estandarizada de lámina de 1/16" que deja suficiente holgura para el giro dentro del opto-interruptor H21A1.

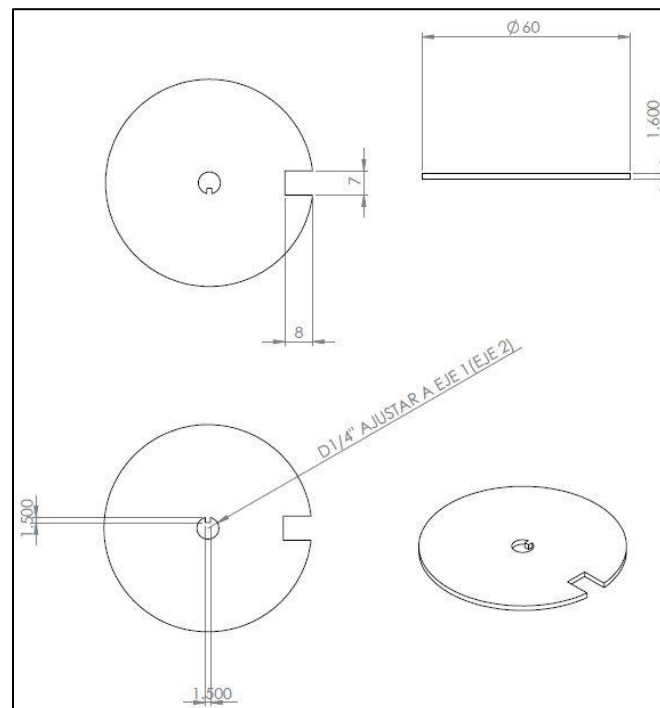


Figura 2.31. Planos de la pieza “Disco” (medidas en [mm]).

Eje superior.

Este elemento, en realidad es un acoplamiento que permite la unión de uno de los discos y el resorte con el eje del motor. Se diseñó con un agujero en la parte superior en el cual embona el eje del motor. Tiene un agujero roscado transversal para fijar su posición mediante un prisionero *Allen* al eje del motor. La parte inferior consta de un diámetro menor para producir un escalón en el cual descansa el disco. Se ideó otro agujero transversal pasado en el cual se inserta un extremo del resorte, que es sujetado por medio de un prisionero *Allen* que se introduce por otro agujero roscado transversal. En el diámetro menor lleva una ranura longitudinal en la que se acopla la lengüeta del disco.

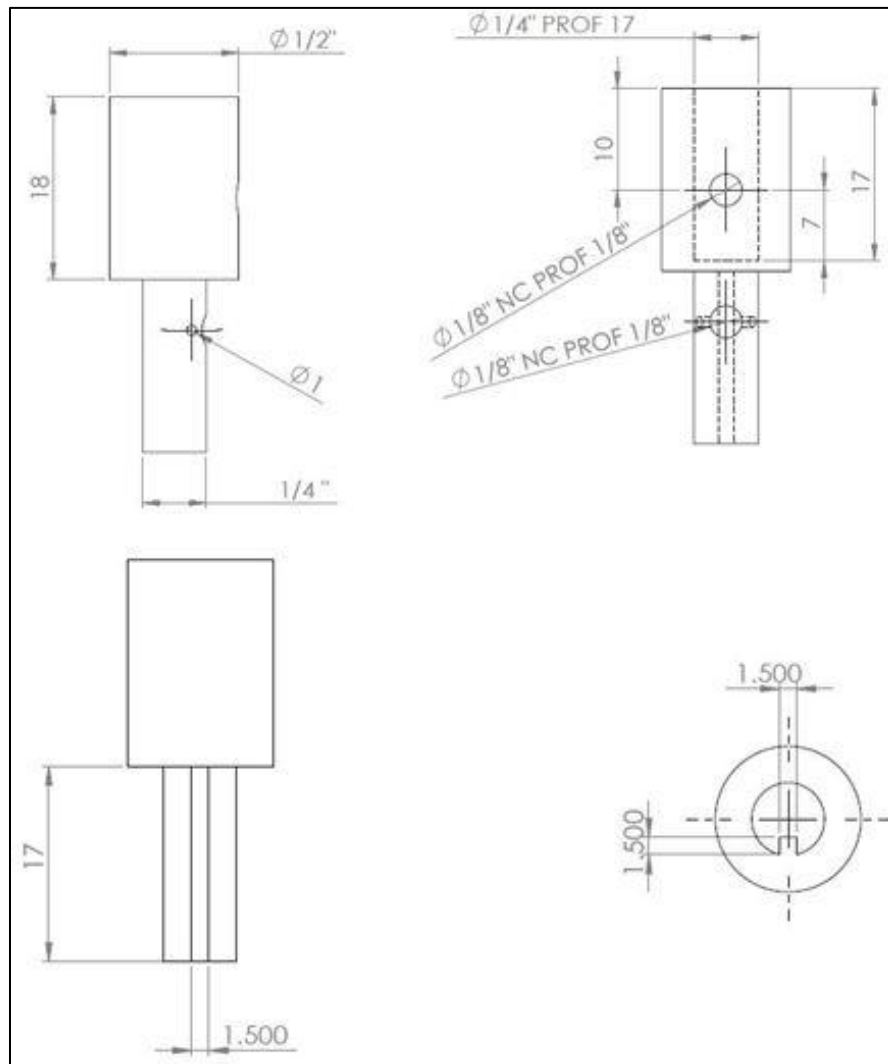


Figura 2.32. Planos de la pieza "Eje superior" (medidas en [mm]).

Eje inferior.

Este elemento es el que soporta al cilindro de prueba, además de estar acoplado al rodamiento para otorgar un movimiento de giro uniforme. Consta de cuatro secciones. En la parte superior (primer sección) consta de un agujero transversal pasado por donde se introduce el extremo inferior del resorte. Además, esta sección de la pieza tiene una ranura longitudinal donde se acopla la lengüeta del segundo disco. El diámetro mayor (segunda sección) funciona como un escalón donde descansa el disco. La tercera sección se ajusta al diámetro interior del rodamiento. La última sección tiene un plano en donde se acoplará la última pieza.

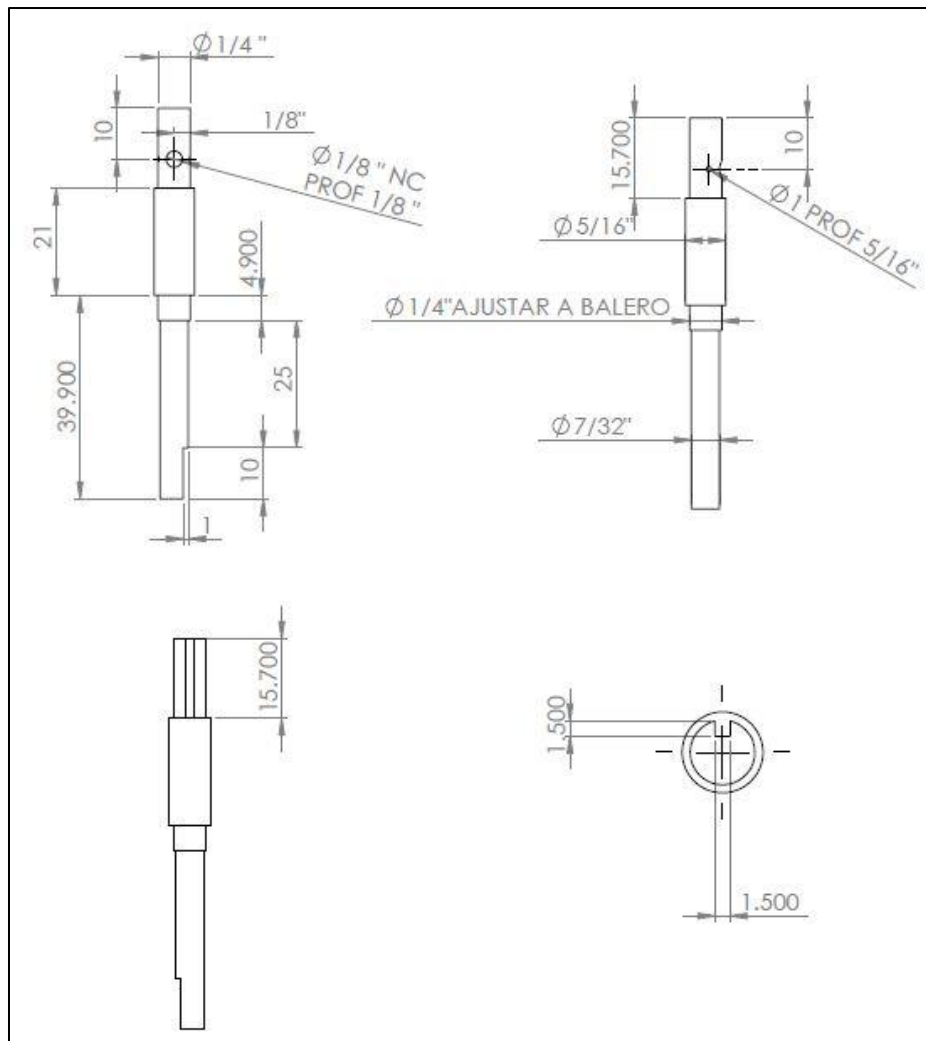


Figura 2.33. Planos de la pieza "Eje inferior" (medidas en [mm]).

Soporte.

Esta pieza se acopla directamente a la base. Su función principal es la de soportar al rodamiento y al cilindro termómetro. A la base se sujeta mediante cuatro tornillos *Allen*. En el centro tiene un agujero pasado y un escalón donde descansa el rodamiento. Además, se aprecia otro agujero pasado donde se hace el ajuste para sostener al cilindro termómetro.

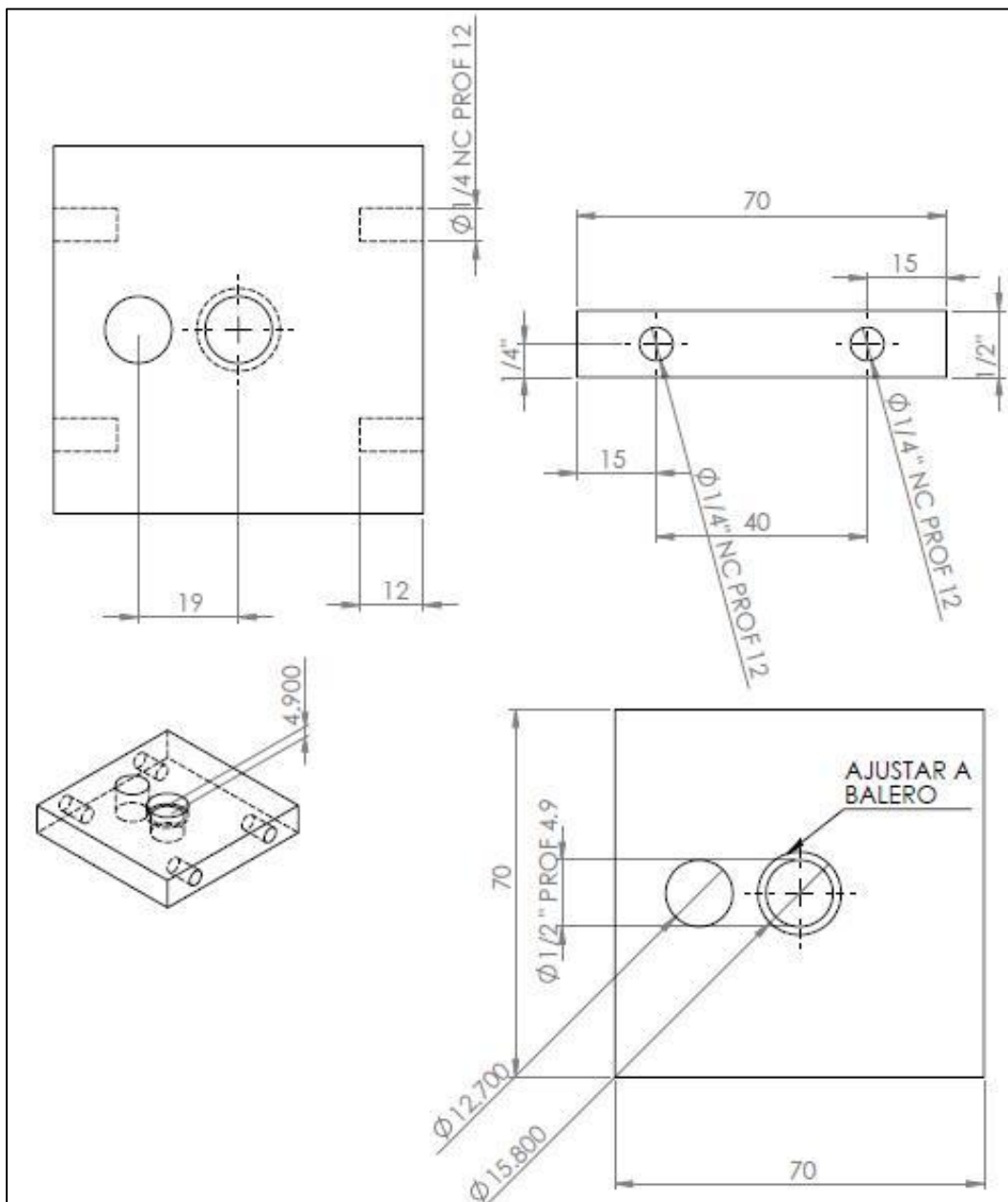


Figura 2.34. Planos de la pieza "Soporte" (medidas en [mm]).

Rodamiento.

Es un rodamiento de bolas, que fue seleccionado para que la medida de su diámetro interior ajustara con la tercera sección del eje inferior, y que la medida de su diámetro exterior no interfiriera con el agujero pasado del cilindro termómetro.

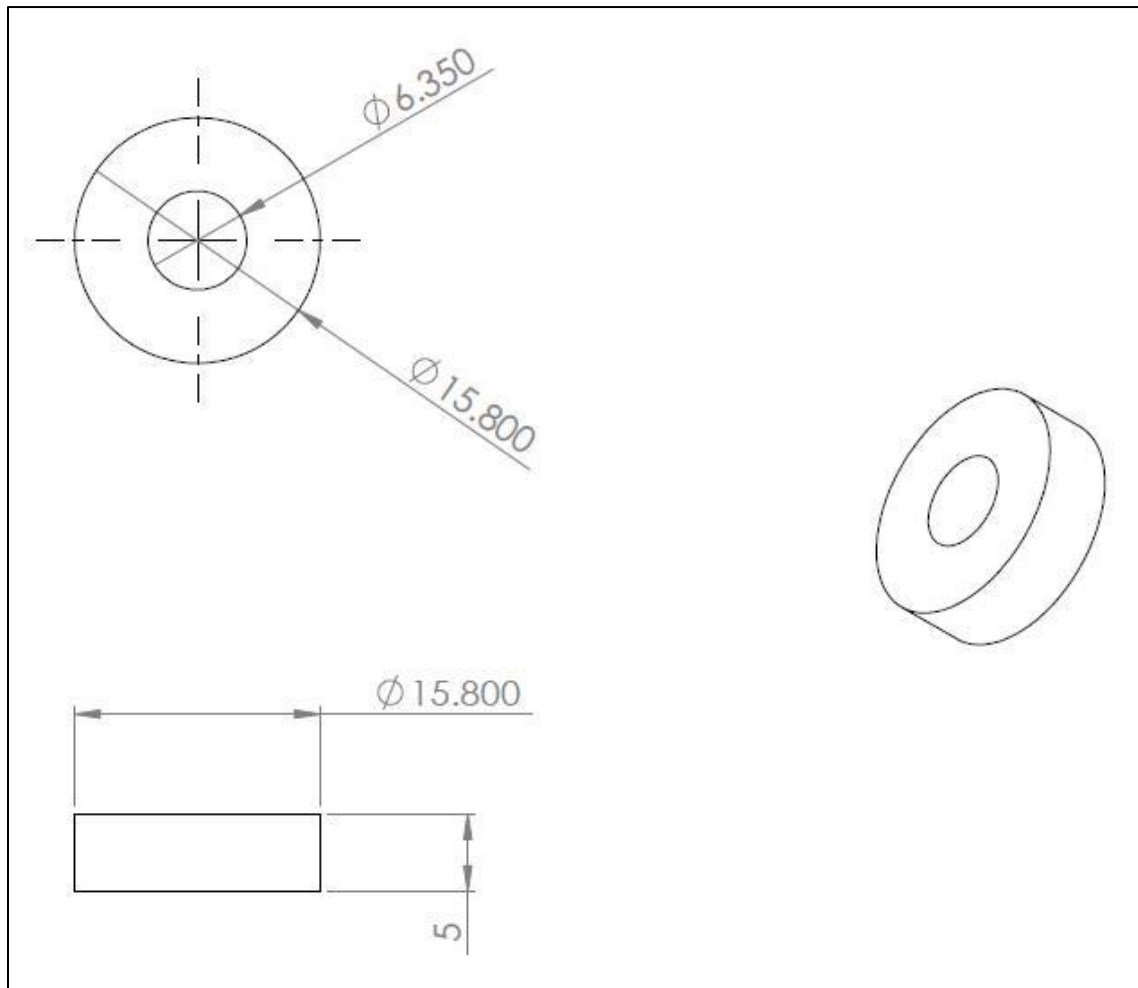


Figura 2.35. Planos de la pieza “Rodamiento” (medidas en [mm]).

Cilindro de prueba.

Es el elemento que se sumerge en el fluido de prueba. Consta de un agujero en la parte superior que permite el acoplamiento con el eje inferior. Tiene un agujero roscado transversal para fijar su posición mediante un prisionero *Allen*.

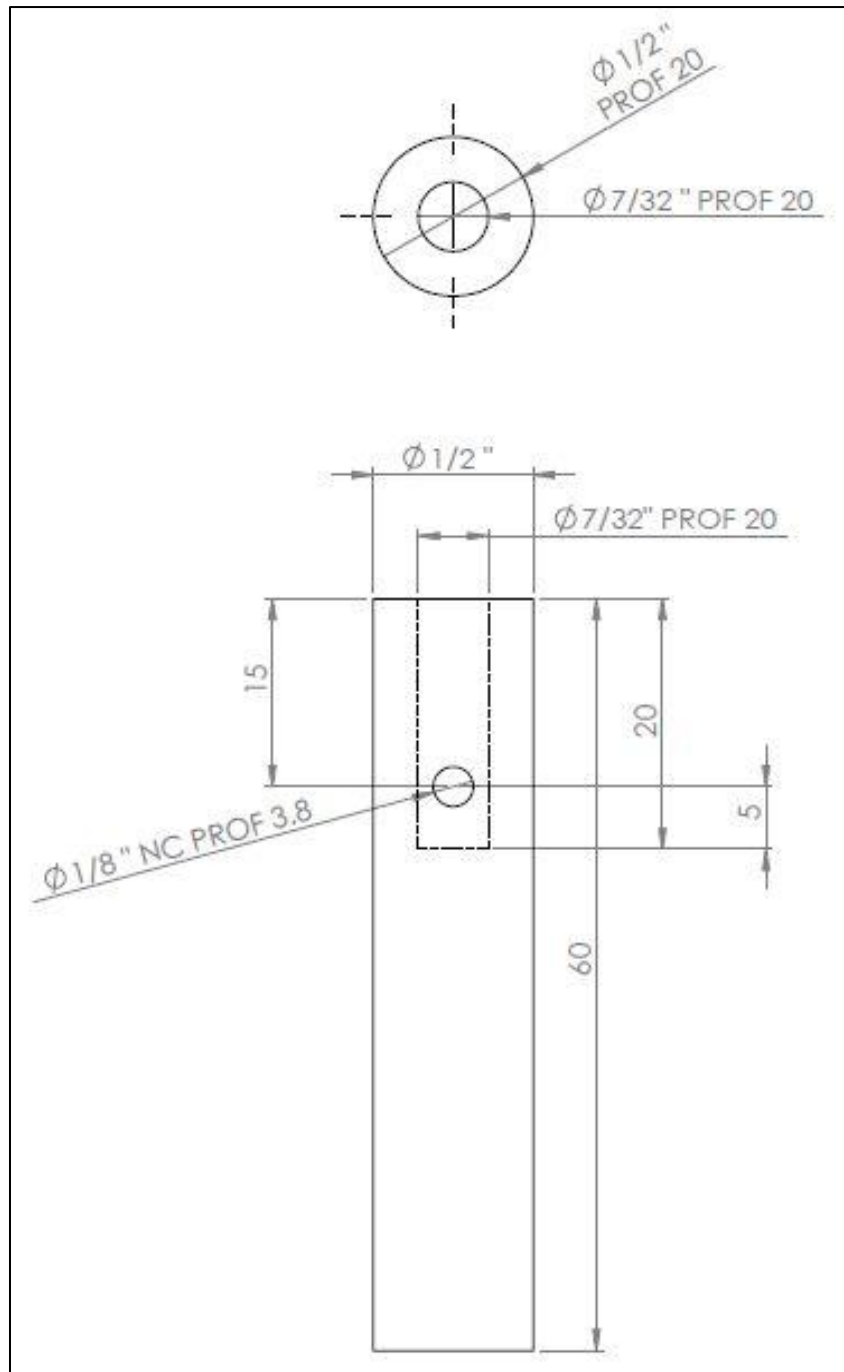


Figura 2.36. Planos de la pieza "Cilindro de prueba" (medidas en [mm]).

Cilindro termómetro.

Se ajusta al agujero pasado correspondiente en la pieza “soporte”. Se diseñó con un agujero en la parte superior en donde se introducirá el sensor de temperatura LM35, en empaquetado TO-92. Su función es proteger el circuito integrado de los fluidos de prueba.

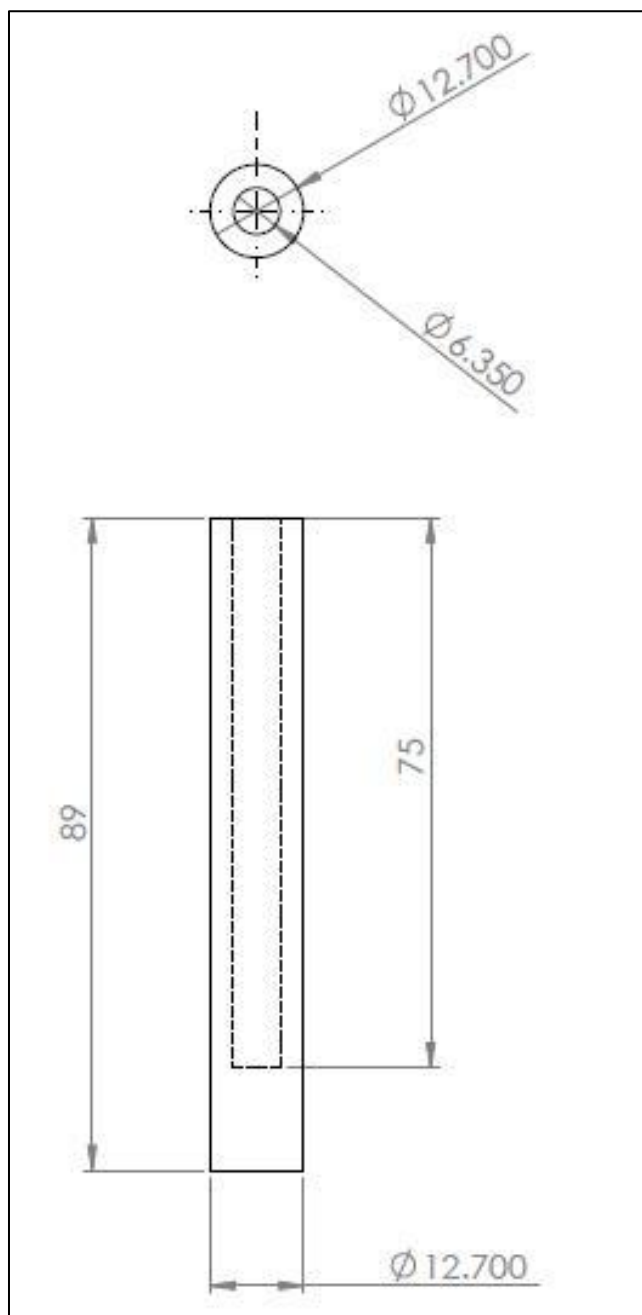


Figura 2.37. Planos de la pieza “Cilindro termómetro” (medidas en [mm]).

Base.

Esta pieza permite el posicionamiento de todos los elementos para realizar su función adecuadamente. Se diseñó pensando en que la configuración fuera funcional y didáctica, para mostrar el principio de operación.

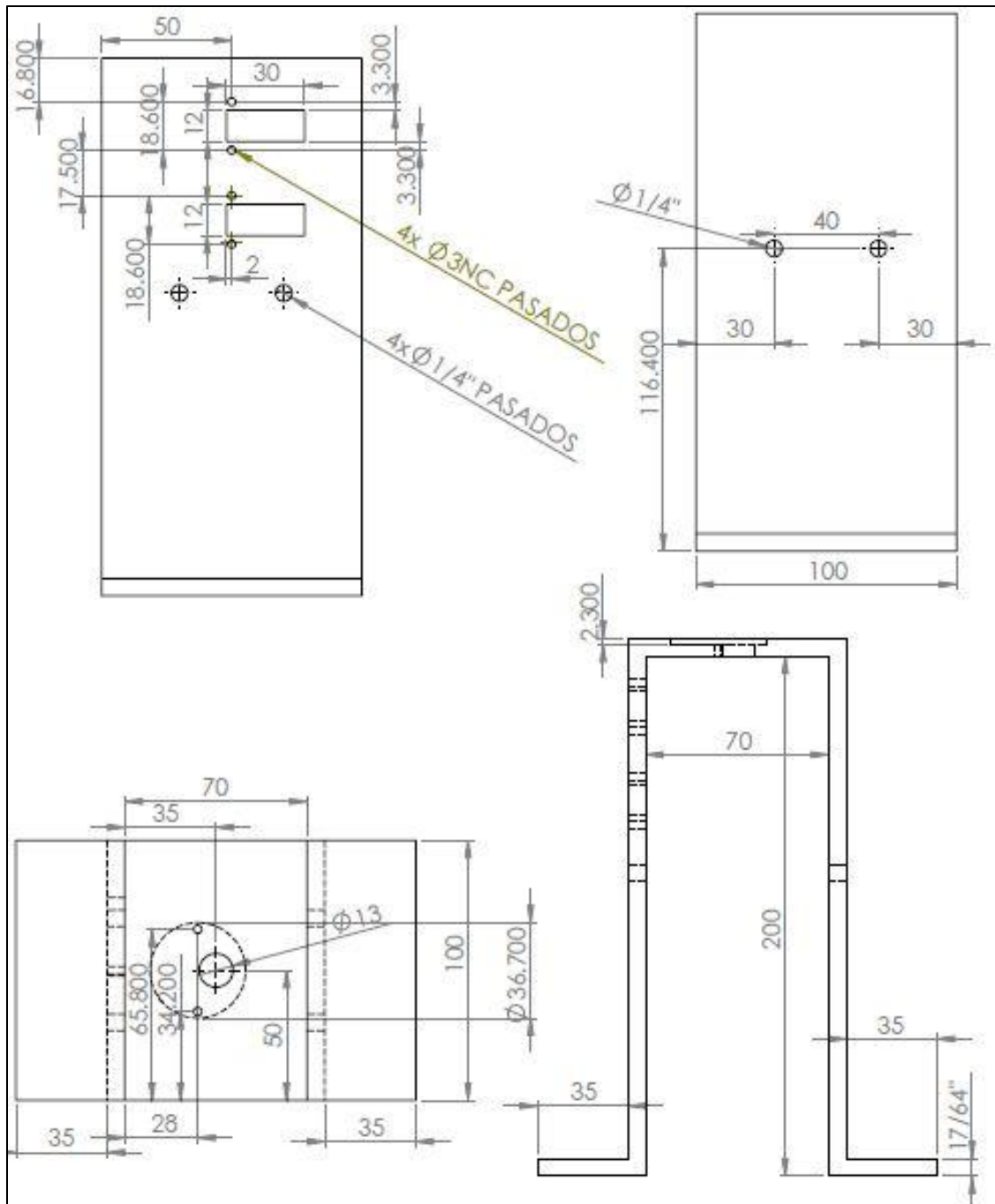


Figura 2.38. Planos de la pieza "Base" (medidas en [mm]).

En la figura 2.39 se muestra el ensamblaje y la enumeración final del prototipo.

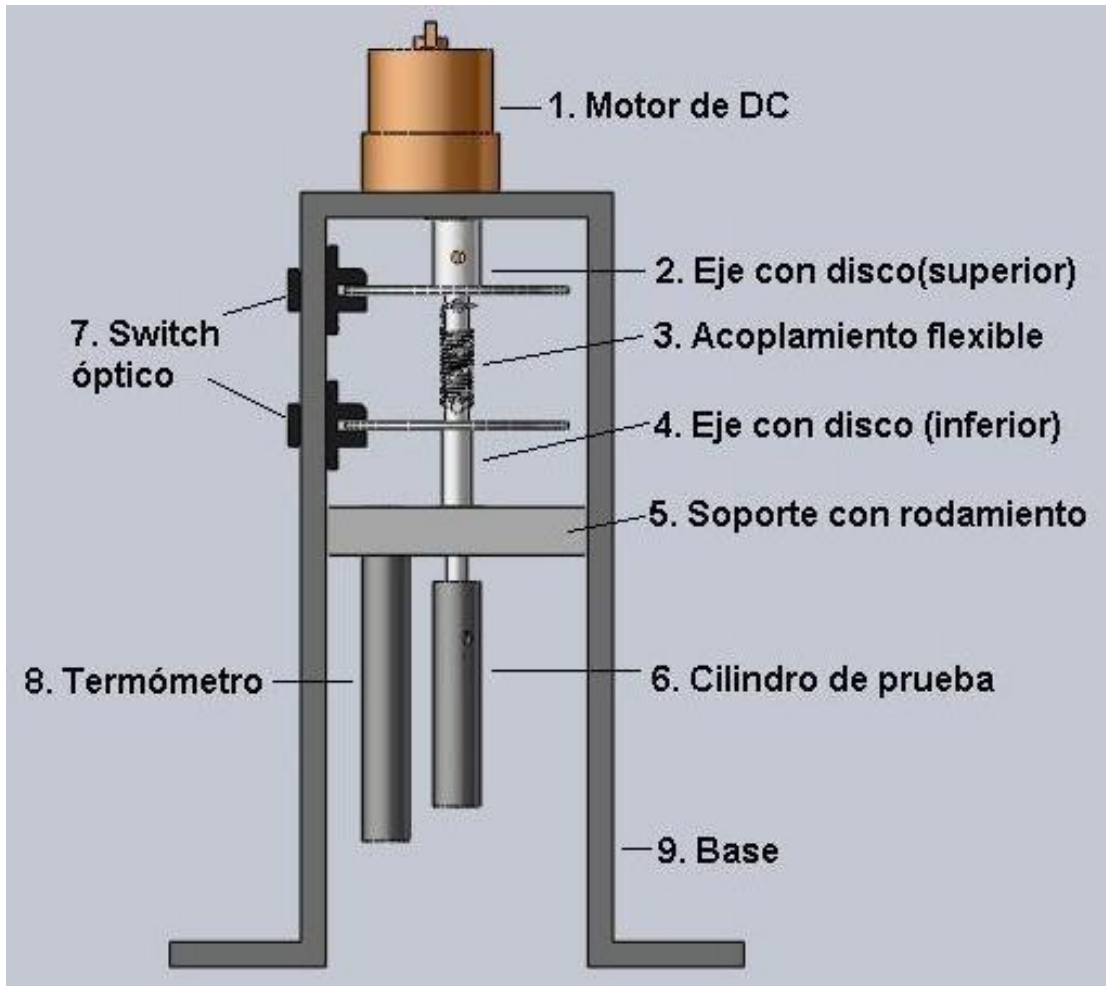


Figura 2.39. Enumeración de componentes del prototipo.

2.5 Procedimiento de calibración y medición

A continuación se muestran los pasos a seguir para realizar la una prueba para obtener el factor de proporcionalidad (F.P.) que relaciona los valores de tiempo de retraso con la viscosidad. Para la medición de viscosidad es necesario conocer tiempo de retardo entre los discos, resultado del esfuerzo cortante que se opone al giro del cilindro de prueba. Para este trabajo, el objetivo es realizar pruebas utilizando fluidos newtonianos (para los cuales es válida la ecuación 8 y así

obtener valores de tiempo de retraso, revoluciones por minuto y temperatura del fluido. La temperatura del fluido sirve como valor de entrada a tablas de viscosidad.

$$\mu [Pa \cdot s] = TMR0 [bits] \times F.P. \left[\frac{Pa \cdot s}{bits} \right] \dots \dots \dots (8)$$

donde: μ = Viscosidad

TMR0= tiempo de retraso

F.P.= factor de proporcionalidad

Despejando de la ecuación 8, se tiene la ecuación 9.

$$F.P. \left[\frac{Pa \cdot s}{bits} \right] = \frac{\mu [Pa \cdot s]}{TMR0 [bits]} \dots \dots \dots (9)$$

Sustituyendo en la ecuación 9 los valores de viscosidad (de tablas) y tiempo de retraso (del prototipo) se puede conocer el F.P. Conocer el F.P. permite que posteriormente, ante un fluido de características desconocidas podamos calcular su viscosidad con la ecuación 8.

Material y equipo necesario para realizar la medición.

- 100 ml del fluido de prueba.
- Computadora personal.
- Tarjeta electrónica.

Pasos a seguir para realizar la prueba.

1. Preparar la muestra de fluido.
2. Abrir la aplicación “Viscosimetro.exe”.

3. Conectar la tarjeta electrónica a la computadora mediante el cable USB.
4. Conectar la tarjeta electrónica a la corriente alterna.
5. En la ventana del programa, oprimir el botón "Posición inicial" y esperar a que se posicione el sistema.
6. Asignar un valor de 0 a 100 a la casilla correspondiente a la velocidad. Oprimir el botón "Velocidad". Se sugiere iniciar desde 100 e ir reduciendo en las mediciones siguientes de 10 en 10.
7. Esperar durante 1 minuto a que el dispositivo se detenga.
8. Oprimir los botones "Medir RPM", "Leer Timer" y "Leer temperatura".
9. Recopilar los resultados en una tabla e iniciar una nueva prueba de ser el caso.

CAPÍTULO 3

3. Resultados

En este capítulo se presentan los resultados de las pruebas realizadas y se hace el análisis necesario para obtener la medición de viscosidad de un par de fluidos de prueba. Con base en estos resultados se determina si el método de medición propuesto es viable y si el hardware desarrollado cumple los objetivos propuestos. Se propone también el trabajo a futuro en base a los resultados entregados por este prototipo.

3.1 Resultados

El prototipo desarrollado (figura 3.1) consta de las tres componentes mecatrónicas siguientes:

Componente mecánica

- Prototipo mecánico.

Componente electrónica

- Circuito electrónico de medición y control.

Componente software

- Interfaz de usuario para PC.

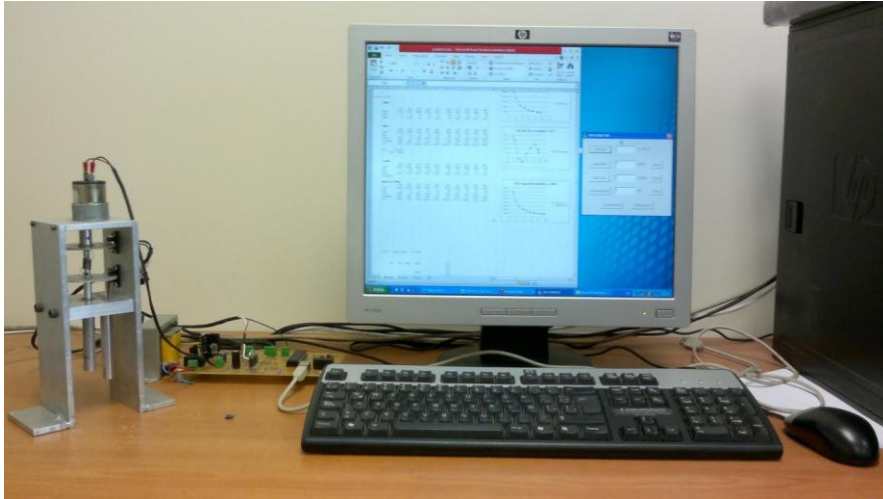


Figura 3.1 Viscosímetro conectado a la PC.

El software desarrollado es capaz de recibir el parámetro de velocidad por parte del usuario y enviarlo al PIC18F4550, vía USB. Posteriormente recibe del microcontrolador los valores resultantes de la prueba para el tiempo de retardo, temperatura y número de revoluciones por minuto. Dicha interfaz fue programada en Visual Studio 6.0 usando lenguaje C++. Para el programa del microcontrolador se utilizó el compilador CCS, y está escrito con base en lenguaje C.

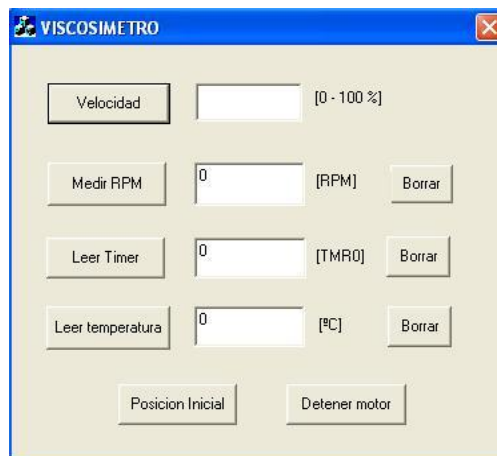


Figura 3.2. Aplicación *Viscosímetro*.

El usuario debe coleccionar los resultados de cada prueba en una tabla de datos. Primero se realiza la prueba en vacío (aire), ya que estos valores de tiempo representan el “cero” del sistema. Además, es importante que los fluidos de prueba se encuentren a la misma temperatura promedio (ambiente) que el cero del sistema. Los resultados se muestran en la tabla 3.1 y figura 3.3. En esta parte se consideran las contribuciones debidas a los rodamientos y a la inercia de la geometría de medición.

Aire a 24[°C] (cero del sistema)

Porcentaje de velocidad	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Timer [bits]	29291	22259	13966	9995	7826	6415	5431	4710	4203	3870
Velocidad angular [rpm]	11	16	26	36	46	56	66	76	86	96

Tabla 3.1. Resultados para aire a 24°C.

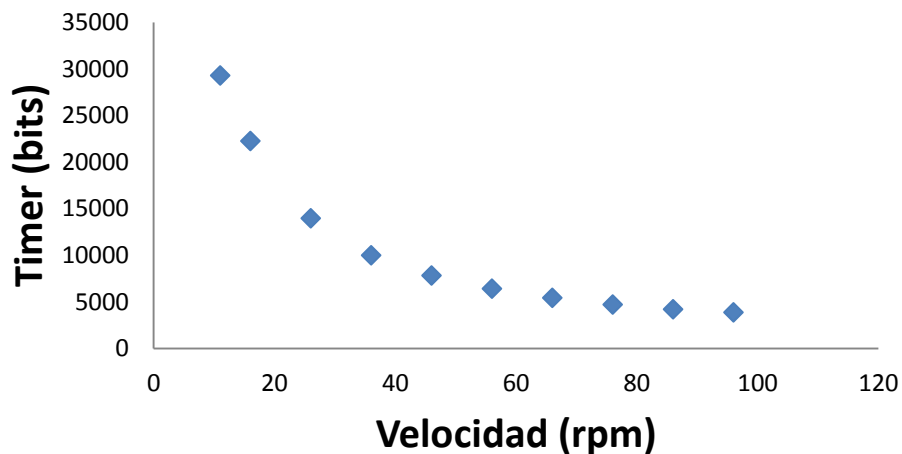


Figura 3.3. Gráfica para aire a 24°C.

Los fluidos de prueba fueron dos: miel y mezcla de agua con glucosa al 90%. La caracterización de los fluidos se realizó en el reómetro *Physica MCR 101* (figura 3.4), disponible en el *Laboratorio de Metrología* del CCADET.



Figura 3.4 Reómetro *Physica MCR 101*.

Se obtuvieron las siguientes viscosidades promedio a 24 [°C] de cada fluido de prueba:

$$\mu_{\text{Miel}} = 29.9 \text{ [Pa}\cdot\text{s]}$$

$$\mu_{\text{Glucosa al 90\%}} = 17.5 \text{ [Pa}\cdot\text{s]}$$

Un aspecto importante, es que en el reómetro MCR-101 se tiene control de la temperatura, por lo que previamente fueron realizadas las mediciones en el prototipo, para poder obtener una temperatura promedio, en este caso 24[°C]; y que al momento de hacer la prueba en el MCR-101 se pudiera especificar que la temperatura fuera igual a 24[°C]. Era importante manejar en ambos casos la misma temperatura, ya que la viscosidad varía principalmente debido a la misma.

Los datos recopilados durante los ensayos realizados en el prototipo son los que se muestran en las tablas 3.2 y 3.3, de las cuales se explica a continuación cada una de sus renglones.

El *porcentaje de velocidad* se varía para comprobar que la ecuación 1 (vista en el capítulo 1) es válida y se puede obtener viscosidad dinámica de estos fluidos de prueba (newtonianos).

La fila *Timer* contiene las mediciones directas de tiempo de retraso, en función de bits.

La *velocidad angular* es la variable contra la que se graficaran los valores de bits obtenidos.

El renglón *Timer corregido* es resultado de restar a cada elemento del renglón *Timer*, el valor correspondiente de la tabla *cero del sistema*. Se debe restar el cero ya que dichos valores representan un error de medición, producto de la propia inercia y fricción del sistema mecánico.

La fila *Viscosidad* contiene los valores resultado de aplicar el F.P. (Factor de Proporcionalidad) a cada elemento de la fila *Timer corregido*. Para ejemplificar lo anterior, en el caso de la miel, se tiene como valores de referencia la viscosidad y el tiempo promedio (TMR0).

$$\mu_{\text{Miel}} = 29.9 \text{ [Pa}\cdot\text{s]} \text{ a } 24[^\circ\text{C}]$$

$$\text{TMR0} = 7495.88 \text{ [bits]}$$

Como la prueba en el prototipo fue realizada a una temperatura promedio de $24[^\circ\text{C}]$, se puede establecer que el promedio de lecturas de la fila *Timer corregido* es proporcional al valor de viscosidad. Primero se obtiene el F.P. usando la ecuación 9 (vista en el capítulo 2):

$$F.P. \left[\frac{\text{Pa}\cdot\text{s}}{\text{bits}} \right] = \frac{\mu \text{ [Pa}\cdot\text{s}]}{\text{TMR0}[\text{bits}]} \dots\dots\dots(9)$$

$$F.P. = \frac{29.9 [Pa \cdot s]}{7495.88 [bits]}$$

$$F.P. = 0.0039 \frac{[Pa \cdot s]}{[bits]}$$

Luego, multiplicando el F.P. por cada uno elementos de la fila *Timer corregido*, se obtiene el valor de viscosidad correspondiente en unidades del SI (Sistema Internacional) Pascal-segundo.

Miel a 24°C

La tabla 3.2 y la figura 3.5 muestran las mediciones obtenidas con el viscosímetro desarrollado en este proyecto de tesis para el caso de miel a 24°C.

Porcentaje de velocidad	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Timer [bits]	38585	31758	21593	17660	15281	13815	12753	12113	11903	11516
Velocidad angular [rpm]	11	16	26	36	46	56	66	76	86	94
Timer corregido [bits]	9294	9499	7376	7665	7455	7400	7322	7403	7700	7646
Viscosidad [Pa-s]	37.0725	37.8902	29.4218	30.5746	29.737	29.5175	29.2064	29.5295	30.7142	30.4988
% Error relativo	23.98	26.72	1.59	2.25	0.54	1.27	2.31	1.23	2.72	2

Tabla 3.2. Resultados de la prueba de miel a 24[°C].

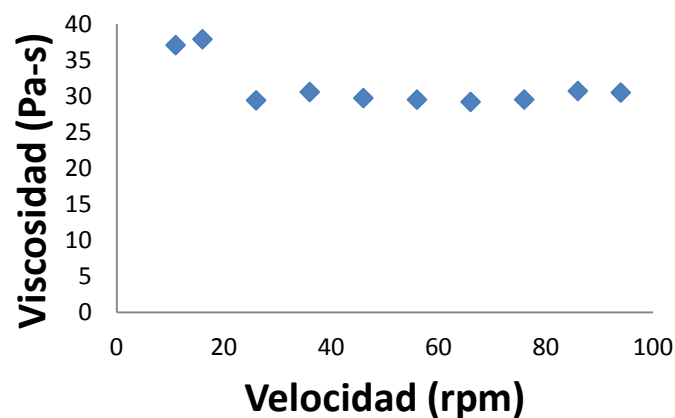


Figura 3.5. Gráfica para la miel a 24°C.

El proceso anterior se aplica igualmente para el caso de la Glucosa a 90% obteniéndose un F.P. muy semejante al de la miel y la tabla 3.3 y su gráfica de la figura 3.6.

$$M_{\text{Glucosa}} = 17.5 \text{ [Pa}\cdot\text{s]} \text{ a } 24[^\circ\text{C}]$$

$$\text{TMR0} = 4589.86 \text{ [bits]}$$

$$F.P. = \frac{17.5 \text{ [Pa}\cdot\text{s]}}{4589.86 \text{ [bits]}}$$

$$F.P. = 0.0038 \frac{\text{[Pa}\cdot\text{s]}}{\text{[bits]}}$$

Glucosa al 90%.

Porcentaje de velocidad	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Timer [bits]	39280	29113	19080	14664	12477	11089	10085	9217	8710	8337
Velocidad angular [rpm]	11	16	26	36	46	56	66	76	87	94
Timer corregido [bits]	9989	6854	5114	4669	4651	4674	4654	4507	4507	4467
Viscosidad [Pa·s]	38.2079	26.2166	19.5611	17.8589	17.7901	17.878	17.8016	17.2393	17.2393	17.0863
% Error relativo	118.33	49.8	11.77	2.05	1.65	2.16	1.72	1.48	1.48	2.36

Tabla 3.3. Resultados para la glucosa al 90%.

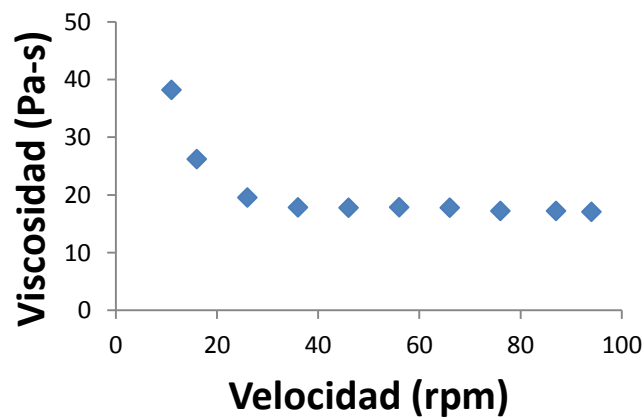


Figura 3.6. Gráfica para la glucosa al 90%.

Cabe mencionar que los valores correspondientes a 15% y 20% en el caso de la miel, y del 15%, 20% y 30% en el caso de la glucosa, no fueron incluidos en los cálculos para obtener el F.P., ya que al ser tan dispersos deben interpretarse como fuera del rango de medición del prototipo. Se incluyen en las gráficas para hacer evidente su desviación.

Finalmente las características del instrumento de medición son las siguientes:

División mínima: 1 [bit]

Alcance de medición: 17 a 31 [Pa•s]

3.2 Conclusiones

Se diseñó y construyó un prototipo funcional del viscosímetro basado en el microcontrolador PIC18F4550, que mejora en alcance de medición respecto al reportado en [5]. El prototipo permitió probar el concepto de medición de viscosidad propuesto. Los valores de viscosidad obtenidos se consideran aceptables, tomando en cuenta que en la mayoría de ellos los porcentajes de error no superan el 3%, respecto al valor de referencia. Se debe observar que en porcentajes de velocidad del motor bajos (de 15 a 30%), el error se incrementa considerablemente; esto se debe a que en esas etapas el motor no provee del par suficiente para vencer las inercias de los elementos mecánicos y por lo tanto el movimiento de giro es irregular

También, se debe resaltar que los resultados obtenidos son válidos únicamente para fluidos newtonianos, como se estableció en la base teórica del capítulo 1 de este trabajo.

Respecto al software, la aplicación *VISCOSIMETRO* resulta simple y fácil de operar y el programa del microcontrolador realiza correctamente la recopilación de datos. La comunicación USB resultó adecuada para este proyecto, debido a los tipos de datos utilizados y estabilidad.

En cuanto al hardware, se puede decir que tanto el motor como el resto del sistema mecánico cumplieron con su objetivo adecuadamente. Por la parte electrónica, la tarjeta de desarrollo fabricada cumplió ampliamente su objetivo de control y recopilación de datos. La ventaja de ser un sistema con base en el microcontrolador PIC, es que permitió que durante la etapa de pruebas, el código del programa se fuese modificando, adaptándose a los cambios en el algoritmo de medición y control surgidos en el proceso. Es por eso que dicha tarjeta electrónica puede ser utilizada en un futuro, ya sea para controlar una versión mejorada del prototipo u otro sistema que requiera control de motor de corriente directa.

3.3 Trabajo a futuro

Con base en los resultados obtenidos, se pueden sugerir las siguientes mejoras que permitirían tener mayor control sobre el sistema y perfeccionar las condiciones para un nuevo prototipo:

- Diseño de una base y vasos de prueba fabricados de material adecuado que facilite la limpieza.
- Un mejor acabado en los discos y ejes, ya que presentan un “cabeceo” considerable.
- Diseño y selección de un resorte o elemento elástico, del cual se conozcan a profundidad sus características, para conocer teóricamente el intervalo de deformación que soportaría. Además, el resorte debe ser *resorte a torsión*.
- Mejorar la aplicación de Visual Studio, para que automáticamente entregue las tablas de datos y genere las graficas necesarias.

REFERENCIAS

4. Referencias

1. Mott, Robert L., Mecánica de fluidos. Sexta edición. Pearson Educación, México 2006. Pp 26-43.
2. White, Frank M. Mecánica de fluidos. Quinta edición. McGraw-Hill, España 2003. Pp-4-5.
3. <http://www.cenam.mx/materiales/mr-viscosidad.aspx>, noviembre de 2011.
4. <http://www.brookfieldengineering.com/products/viscometers/index.asp>, mayo de 2012.
5. Nava S. R., Valera O. B., Pérez L. J. S. y Vázquez P. L. A., 2008, Viscosímetro portátil USB, XXIII Congreso de Instrumentación, SOMI, Jalapa Veracruz, 1 al 3 de octubre.
6. Parra R. G. A., “Instrumentación de un viscosímetro de cilindros concéntricos”, 2010.
7. Clayton T. Crowem Elger. Mecánica de fluidos. Primera edición. Compañía Editorial Continental., México 2002. Pp 2-3.
8. Merle C. Potter, David C. Wiggert. Mecánica de fluidos. Tercera edición. Ed. Thomson, México 2002. Pp 14-17.
9. Valdés, Pallás. Microcontroladores: fundamentos y aplicaciones con PIC. Alfaomega Grupo Editor, México 2007. Pp 14-17, 29-34.
10. PIC18F2455/2550/4455/4550 Data Sheet, Microchip, U.S.A. 2006.
11. Eduardo García Breijo., Compilador C CCS y simulador PROTEUS para Microcontroladores PIC., Alfaomega Grupo Editor, México 2008. Pp 23-143, 251-254.

12. <http://www.microchip.com>. Microchip, enero 2012.
13. "H21A1" Data Sheet, Fairchild Semiconductor Co.; USA, 2001.
14. "LM35" Data Sheet, National Semiconductor Co.; USA, 2000.
15. Victor L Streeter, E. Benjamin Wylie, Keith W. Bedford. Mecánica de fluidos. Novena edición. Mc Graw Hill, Colombia, 2000. Pp 3-5,8-11, 493-497.
16. Crowe, C. T. Engineering fluid mechanics. Seventh Edition. John Wiley & Sons, Inc. USAM 2001. Pp 2-3.
17. Castro González, Rodolfo. Diseño conceptual de un viscosímetro de cilindros concéntricos. Universidad Nacional Autónoma de México. Facultad de Ingeniería. 2007. Pp 21.
18. "LMD18201" Data Sheet. National Semiconductor Co. USA, 2011.
19. <http://www.microsoft.com>. Microsoft Corporation.
20. <http://www.ccsinfo.com/>. CCS Inc.
21. Pappas, Murray. Microsoft Visual C++ 6.0 Manual de referencia. Mc Graw Hill/Interamericana de España, 1999.
22. <http://www.muchotrasto.com/>. Muchotrasto, junio 2011.
23. <http://www.kysanelectronics.com>. Kysan electronics, mayo 2012.

ANEXOS

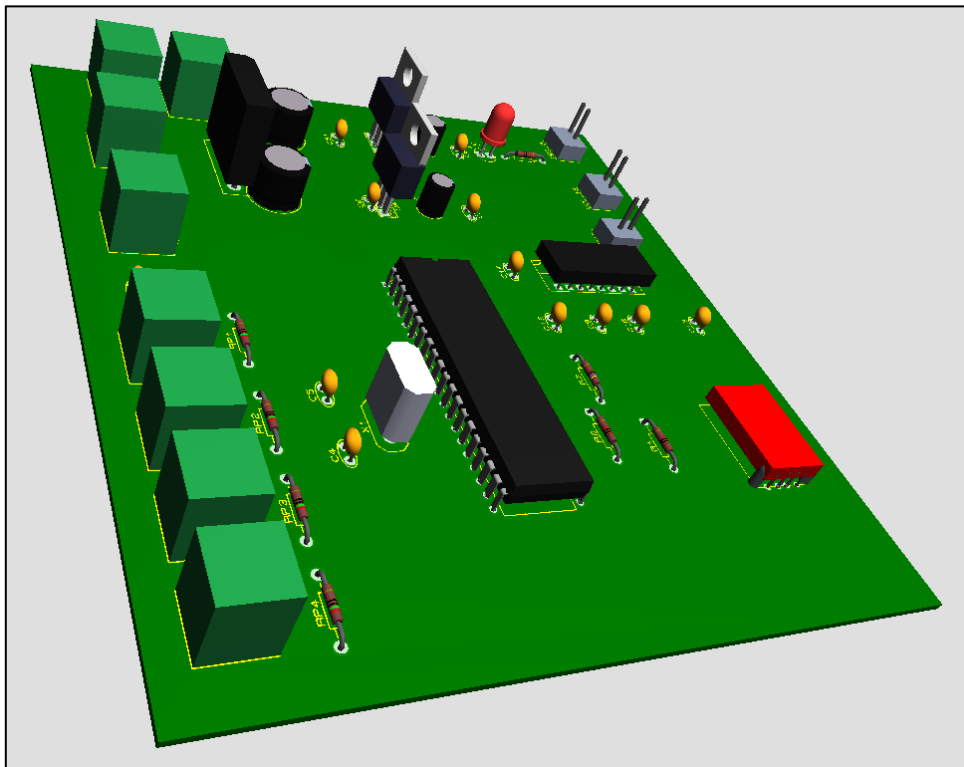
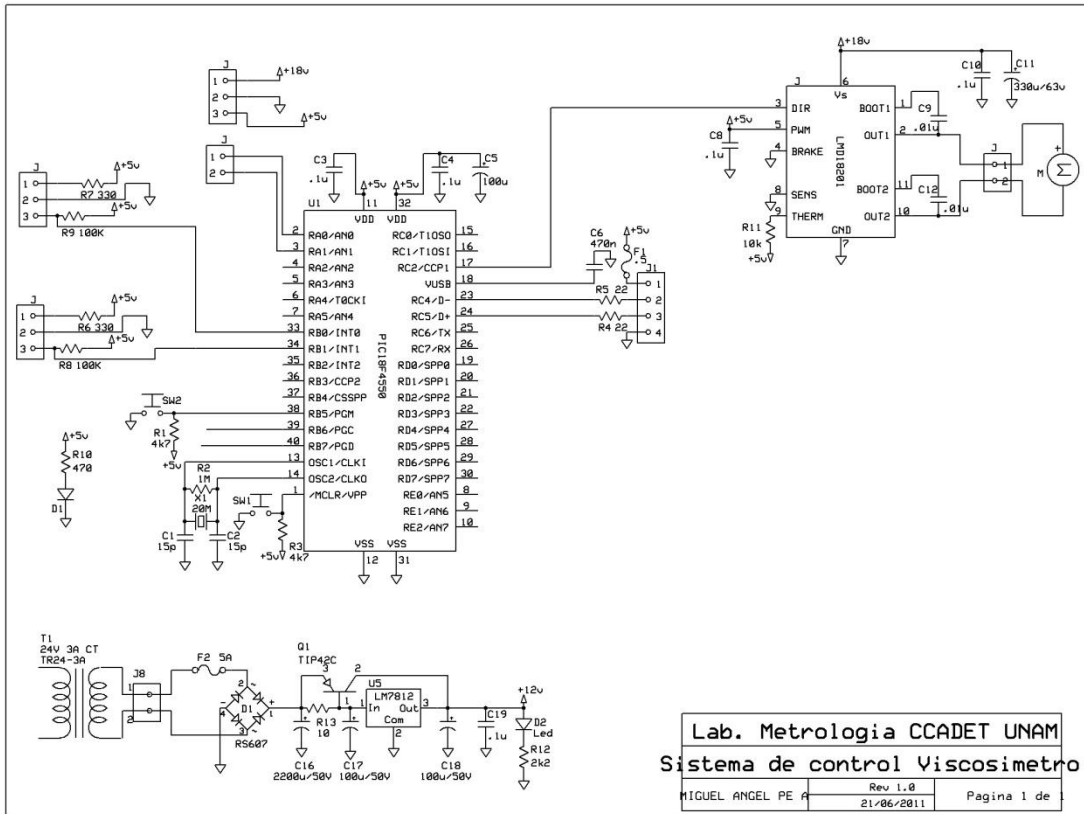
5. Anexos

1. Lista de material utilizado en la tarjeta de desarrollo.

Componente	Número de piezas
Microcontrolador PIC18F4550	1
Dip socket 40 pines	1
Conector hembra USB tipo B	1
LMD18201	1
Cristal 20 MHz	1
Fusible 0.5A	1
Sujetador de fusible	2
Capacitor electrolítico 100uF, 10V	1
Capacitor electrolítico 2200uF, 50V	1
Capacitor electrolítico 100uF, 50V	2
Capacitor electrolítico 330uF, 63V	1
Capacitor cerámico 0.47uF, 50V	1
Capacitor cerámico 15pF	2
Capacitor cerámico 100nF	5
Capacitor cerámico 0.01uF	2
Led 3mm	2
Switch push	2
Resistencia 1MΩ ½ W	1
Resistencia 4.7KΩ ½ W	1

Resistencia 470 Ω ½ W	1
Resistencia 10K Ω ½ W	1
Resistencia 22 Ω ½ W	2
Resistencia 4K Ω ½ W	2
Resistencia 2KM Ω ½ W	1
Resistencia 10 Ω ½ W	1
Transformador 24V 3A	1
H21A1	2
Terminal para PCB 2 vías	3
Terminal para PCB 3 vías	3
Fusible 5A	1
TIP42C	2
RS607	1
LM7812	1
LM7805	1

2. Diagrama esquemático de la tarjeta de desarrollo.



3. Hoja de especificaciones del motor [23].

21/05/2012

Kysan Electronics Data Book



Spur Geared Motors



- Data Group: 88
- SKU: 1151245
- Weight (Kg): 0.5000
- Description: HN-GH12-1640Y
- HS Code: 8483.40.9000
- Applications: Toy, Medical equipment

KYSAN SKU	MFG Model	Rated Voltage (V)	No Load		Gear Ratio	At Maximum Efficiency			Motor Size (mm)	Gear Case Size (mm)	Shaft Size (mm)
			Current (A)	Speed (RPM)		Speed (RPM)	Current (A)	Torque (Kgi-cm)			
1151201	38-001	12	<0.12	720	10:01	600	<0.4	0.35	Φ33x25.9	Φ38x18.5	Φ6x12
1151202	38-002	12	<0.12	240	30:1	200	<0.37	1	Φ33x25.9	Φ38x18.5	Φ6x12
1151203	38-003	12	<0.12	216	33:1	180	<0.37	1.1	Φ33x25.9	Φ38x18.5	Φ6x12
1151204	38-004	12	<0.12	144	50:1	120	<0.35	1.5	Φ33x25.9	Φ38x18.5	Φ6x12
1151205	38-005	12	<0.12	130	56:1	110	<0.35	1.6	Φ33x25.9	Φ38x18.5	Φ6x12
1151206	38-006	12	<0.12	80	90:1	66	<0.35	2.9	Φ33x25.9	Φ38x18.5	Φ6x12
1151207	38-007	12	<0.12	72	100:1	60	<0.35	3	Φ33x25.9	Φ38x18.5	Φ6x12
1151208	38-008	12	<0.12	48	150:1	40	<0.35	4.5	Φ33x25.9	Φ38x25.5	Φ6x12
1151209	38-009	12	<0.12	43	167:1	36	<0.29	4.7	Φ33x25.9	Φ38x25.5	Φ6x12
1151210	38-010	12	<0.12	29	250:1	24	<0.29	5.8	Φ33x25.9	Φ38x25.5	Φ6x12
1151228	38-028	12	0.08	8	900:1	6.5	0.1	24.5	Φ32x26	Φ38x25.5	Φ6x17
1151229	38-029	12	0.08	14	500:1	12	0.1	13.6	Φ32x26	Φ38x25.5	Φ6x17
1151243	38-043	12	0.08	128	56:1	107	0.1	1.1	Φ32x26	Φ38x18.5	Φ6x17
1151237	38-037	12	0.08	48	150:1	40	0.1	7.5	Φ32x26	Φ38x25.5	Φ6x17
1151215	38-015	24	0.06	24	300:1	20	0.085	11.6	Φ32x26	Φ38x25.5	Φ6x17
1151231	38-031	24	0.06	8	900:1	6.5	0.085	32.2	Φ32x26	Φ38x25.5	Φ6x17
1151232	38-032	24	0.06	14	500:1	12	0.085	19.4	Φ32x26	Φ38x25.5	Φ6x17
1151233	38-033	24	0.06	29	250:1	24	0.085	10.6	Φ32x26	Φ38x25.5	Φ6x17
1151234	38-034	24	0.06	43	167:1	36	0.085	7	Φ32x26	Φ38x25.5	Φ6x17
1151235	38-035	24	0.06	48	150:1	40	0.085	6.3	Φ32x26	Φ38x25.5	Φ6x17
1151236	38-036	24	0.06	80	90:1	66	0.085	3.8	Φ32x26	Φ38x18.5	Φ6x17
1151238	38-038	24	0.06	128	56:1	107	0.085	2.3	Φ32x26	Φ38x18.5	Φ6x17
1151239	38-039	24	0.06	218	33:1	181	0.085	1.5	Φ32x26	Φ38x18.5	Φ6x17
1151240	38-040	24	0.06	240	30:1	200	0.085	1.3	Φ32x26	Φ38x18.5	Φ6x17
1151241	38-041	24	0.06	720	10:01	600	0.085	0.5	Φ32x26	Φ38x18.5	Φ6x17

38 series is a small-sized DC geared motor. It has low noise, long service life, large output torque, wide speed reducing rang1, and excellent servo performance, it widely apply to occasions such as instrument and meter, automatic control, bank currency count machine, stage lighting, advertising exhibition instrument, monitor, valves of various kinds, medical instrument, scientific research facility, sports instrument, window automatic control, soft handkerchief machine, automobile television, automatic voltage regulator, building water and electricity control system, etc.

21/05/2012

Kysan Electronics Data Book

Shaft Detail

DATE DRAWING PREPARED		DESIGN	DATE	BY	IN KIT B38 SERIES	REVISED
CHECKED AND IN ALLIGATION					Unit: mm	
DRAWN BY					Kysan Electronics	
CHECKED BY					B38 Series Geared Motor	
DATE					M	
BY					PAGE 11	
REVISION					SHEET 11	

i	30	33	50	65	90	100
L	18.5					
i	100	166	270	300		
L	450	500	810	900		
L	25.5					

i: Ratio L: Gearbox length

KYSAN ELECTRONICS
Unit: mm

©2003 KYSAN ELECTRONICS All Rights Reserved | www.kysanelectronics.com

4. Características del viscosímetro desarrollado en [5].

En el artículo “VISCOSÍMETRO PORTÁTIL USB” [5] se presenta el diseño mecánico, electrónico y de programación en el desarrollo de un viscosímetro portátil con interfase USB. El sistema está desarrollado con base al principio del viscosímetro de cilindro, en donde un rotor, impulsado por un motor CD, gira con relación a un estator y entre ambos es contenido el fluido por analizar. En ese esquema, el fluido opone resistencia al giro del rotor y demanda mayor consumo de corriente en el motor. Se demuestra que la corriente y viscosidad conservan una relación lineal dentro de un intervalo de 0.003 [Pa•s] a 0.27 [Pa•s]. Las variaciones de corriente son detectadas por un sistema electrónico y transformadas a unidades de viscosidad. Las mediciones del sistema electrónico son transportadas a una computadora huésped por medio de la interfase USB. Se presentan resultados que demuestran el desempeño del viscosímetro desarrollado.

Las características del equipo de medición son las siguientes:

Alcance: 0.002 – 0.4 [Pa•s]

División mínima: 0.0001 [Pa•s]

El instrumento ha demostrado su utilidad en aplicaciones que no requieren un alto alcance de operación.