


# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

## INTERFAZ DE USUARIO PARA PROBADOR DE PRÓTESIS CARDIACAS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO - ELECTRÓNICO
P R E S E N T A

**JAIME MONROY ELIZONDO**

DIRECTOR DE TESIS: M. en I. MIGUEL ÁNGEL BAÑUELOS SAUCEDO

LABORATORIO DE ELECTRÓNICA
CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO




MÉXICO, D.F. 2006

*A Dios por darme la fuerza y salud necesarias para llegar a realizar una de mis tantas metas.*

*A mis padres, Don Aurelio Monroy Moreno y Doña Rosa Elizondo Silva, mis fuentes de inspiración y apoyo a largo de toda mi vida. Gracias por enseñarme a luchar y no dejarme vencer por los obstáculos que se presentan en la vida, pero sobretodo gracias por apoyarme en la realización de todos mis sueños.*

*A mis hermanos, Ismael y Angélica, gracias por todo su apoyo incondicional en las buenas y en las malas.*

*A David, Martín y Miguel, a quienes considero como mis hermanos, gracias por estar conmigo en mis momentos de alegría y tristeza.*

*A todos mis amigos, gracias por depositar su confianza en mí y seguir conmigo hasta el final de este trabajo.*

*A ti, que me has hecho sentir especial y me has alentado en esta última etapa de mi trabajo de tesis, gracias Mariana.*

*A mi director de tesis, el maestro Miguel Ángel Bañuelos, a los maestros José Castillo y Sergio Quintana, gracias por compartir su conocimiento y por haberme albergado en el Laboratorio de Electrónica por más de dos años.*

*A mis compañeros del Laboratorio de Electrónica, gracias por compartir momentos gratos y no tan gratos, pero sobre todo por ayudarme a la realización de este trabajo.*

*A mi querida Universidad Nacional Autónoma de México, gracias ser mi segunda casa durante 11 años.*

*A mi gran Facultad de Ingeniería y a todos sus profesores, porque sin ellos no habría podido lograr esto, gracias.*

*Al Instituto Nacional de Cardiología, gracias por permitirme ser partícipe de un gran proyecto.*

*A todos, gracias por creer en mi, este trabajo lleva un poco de cada uno de ustedes.*

# INDICE

1. INTRODUCCIÓN ..... 1

1.1 Antecedentes ..... 1
1.2 Definición del problema ..... 1
1.3 Alcances ..... 8

2. SENSORES ..... 9

2.1 Descripción ..... 9
2.2 Características ..... 11
2.3 Ventajas y desventajas ..... 11
2.4 Acondicionamiento de las señales de salida de los sensores ..... 12

3. INTERFAZ DE HARDWARE ..... 16

3.1 Microcontrolador ..... 16

3.1.1 Descripción ..... 16
3.1.2 Características ..... 18
3.1.3 Ventajas y desventajas ..... 21

3.2 Otros circuitos ..... 22

3.3 Desarrollo de la interfaz de hardware ..... 22

3.3.1 Implementación de la interfaz de hardware ..... 22
3.3.2 Uso de la comunicación serie para la interconexión de la interfaz de hardware con la de usuario ..... 27
3.3.3 Programación del microcontrolador para la recepción, procesamiento y transmisión de datos ..... 31

4. ETAPA DE POTENCIA ..... 34

4.1 Diseño y desarrollo del circuito de potencia ..... 34

5. INTERFAZ DE USUARIO ..... 36

5.1 ¿Qué es una Interfaz de Usuario? ..... 36

5.1.1 Conceptos de interfaz ..... 36
5.1.2 Clasificación ..... 37
5.1.3 Características humanas del diseño de la interfaz ..... 37

5.2 Visual Basic ..... 38

5.2.1 ¿Qué es Visual Basic? 38
5.2.2 Características 38
5.2.3 Ventajas y desventajas 40

5.3 Diseño de la interfaz 40

5.3.1 Análisis y recopilación de información del usuario final de la interfaz así como del uso final de la interfaz misma 41
5.3.2 Diseño de la interfaz de usuario 42

5.4 Implementación de la interfaz de usuario 43

5.4.1 Programación de la interfaz gráfica para el despliegue y control de la información, utilizando Visual Basic 6.0 43

6. PRUEBAS Y RESULTADOS 56

7. CONCLUSIONES Y COMENTARIOS 62

8. APÉNDICES 64

9. BIBLIOGRAFIA 75

# 1. INTRODUCCIÓN

## 1.1 Antecedentes

En el Instituto Nacional de Cardiología (INC) se desarrollan válvulas cardiacas para su implantación en pacientes que así lo requieren. Estas válvulas se elaboran a partir de válvulas porcinas o de tejido pericardio vacuno, a las cuales se les añade un refuerzo mecánico para ajustar su forma.

Actualmente se evalúa su desempeño mediante un instrumento para prueba de válvulas cardíacas que fue donado al Instituto. Sin embargo, no cuenta con las características adecuadas para hacer pruebas de fatiga acelerada, ni con la capacidad para actuar sobre múltiples válvulas simultáneamente.
Personal de dicho Instituto ha trabajado en el desarrollo de un nuevo probador, pero los resultados han sido parciales.

Para lograr las características antes mencionadas es que se desarrolla actualmente un nuevo probador de prótesis cardiacas, en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET).

## 1.2 Definición del problema

Para la definición de las características básicas del equipo a desarrollar se tomaron en cuenta las normas NOM-063-SSA1-1993, información técnica y comercial proporcionada por el INC, así como conversaciones con personal de dicho instituto.

A continuación se mencionan algunas características que la norma, que es de carácter obligatorio, establece como necesarias para el equipo de prueba:

- Debe producir las formas de las curvas de presión y flujo que se obtienen en el humano adulto sano.

- Debe tener la capacidad de variar ciclos por minuto de bradicardia severa (20 cpm) a taquicardia severa (200 cpm).

- La relación sístole - diástole debe ser variada en la bradicardia de 35 $\pm$ 5% del tiempo total del ciclo para la fase sistólica, de 45 $\pm$ 5% para la frecuencia normal y de 50% para la taquicardia.

- La simulación de la presión sistémica debe ser de 16 $\pm$ 1 kPa (120 $\pm$ 7.5 mm Hg) y la presión diastólica debe ser de 10 $\pm$ 0.5 kPa (80 $\pm$ 3.8 mm Hg).

- Debe proporcionar los medios para ver y fotografiar la válvula en cualquier momento del ciclo.

Respecto a las variables a medir, la misma norma establece que deben registrarse:

- La diferencia de presión media a través de la válvula.
- La frecuencia del ciclo.
- La presión arterial media durante todo el ciclo.

Respecto a la prueba de durabilidad, la norma menciona algunos puntos, pero no da parámetros para una prueba de fatiga acelerada. Especifica que las válvulas deberán ser probadas hasta que fallen o se completen 380 x $10^6$ ciclos, con exámenes cada 38 x $10^6$.

El probador de prótesis cardiacas a desarrollar constará de una cámara hecha de material transparente dentro de la cual se colocará la válvula a evaluar; por medio de un actuador se bombeará una solución salina a cierta presión y frecuencia, con esto se logrará que la válvula abra y cierre durante un periodo de tiempo establecido, simulando así ciertas condiciones a las que es sometida dentro del organismo humano.

Un actuador es un sistema encargado de convertir una variable eléctrica en una variable física.

En el probador a desarrollar, con ayuda de dos sensores de presión, colocados uno a la entrada y otro a la salida del flujo que atravesará la válvula, se podrá saber si las pruebas que se están realizando son las correctas, además de saber si el actuador está realizando las funciones encomendadas. Para lograr todo esto es que se desarrollará la interfaz, ya que con ésta se graficarán los datos provenientes de los sensores de presión y se enviarán las señales de control hacia el actuador. Esta manipulación se realizará desde una computadora.

En el desarrollo del probador se tienen contemplados dos diferentes actuadores: una electroválvula y un motor de pasos acoplado a un sistema biela-manivela.

En el presente trabajo se hablará del diseño de la interfaz tanto para el control de la electroválvula como para el del motor. Aunque ambas interfaces son muy similares, presentan algunas diferencias importantes.

A continuación se muestra un diagrama que en términos generales representa al probador.

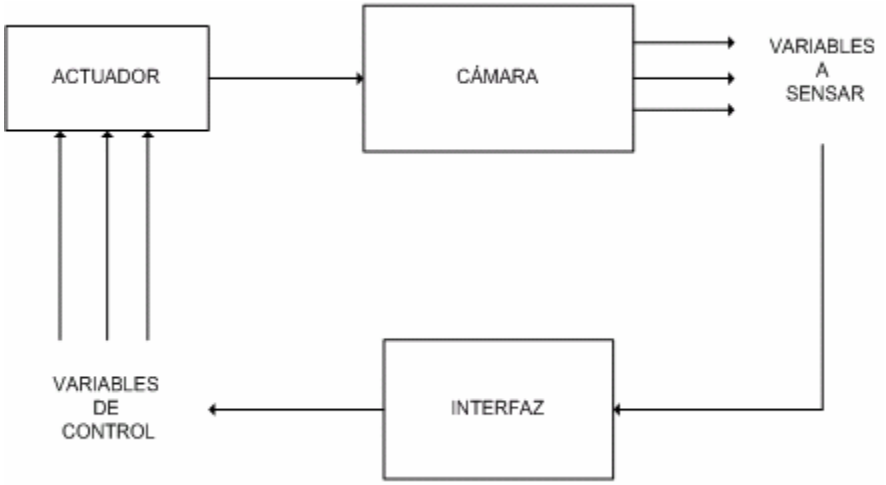



*Figura 1.1 Esquema general de un probador de prótesis cardiacas.*

Específicamente el desarrollo de la interfaz de usuario está a cargo del Laboratorio de Electrónica del CCADET y mi proyecto de tesis estuvo enfocado a la realización de la misma.

La necesidad de realizar la interfaz surge de poder monitorear y controlar varios procesos al mismo tiempo, tales como la recepción de datos provenientes de los sensores de presión colocados en la cámara y el envío de señales de control hacia el actuador, desde un solo equipo. Actualmente, en el INC esto se hace con varios osciloscopios y módulos de control, pero gracias a los avances tecnológicos, hoy en día es posible contar con circuitos electrónicos, computadoras y lenguajes de programación que en conjunto nos ayudan a crear software y hardware capaz de realizar estas tareas.

El objetivo de este trabajo fue desarrollar una interfaz de usuario amigable, tanto en software como en hardware, para realizar la comunicación vía el puerto serie, entre una computadora y los diferentes sensores y actuadores del probador, logrando así la medición y control de ciertas variables físicas durante la prueba de las prótesis cardiacas.

Además de la realización de la interfaz se trabajó con los sensores de presión que se utilizaron para monitorear la variable de interés, que en este caso fue la presión que experimenta el fluido antes y después de atravesar la válvula.

Así mismo, se habla de lo que se hizo con las señales de control a la salida del microcontrolador, ya que los actuadores necesitaron de una etapa de potencia previa.

Cabe señalar que los actuadores se realizaron en otro laboratorio, por lo que no se hablará mucho al respecto.

La principal diferencia entre el probador de prótesis donado al INC y el realizado en el CCADET es que el primero trabaja con varios módulos de adquisición y control, mientras que en el segundo estos módulos se agruparon en uno solo, por ende se redujo en costos de equipo, ya que sólo se necesitó la cámara con el actuador, los sensores, una interfaz de hardware y una computadora desde la cual se controla el instrumento. En ciertos módulos del diseño fue necesario el acondicionamiento de las señales para lograr la correcta interconexión de dichos módulos.




*Figura 1.2 Imágenes del probador de prótesis del INC y su sistema de control.*

El siguiente esquema muestra de manera general los módulos a desarrollar en el CCADET:

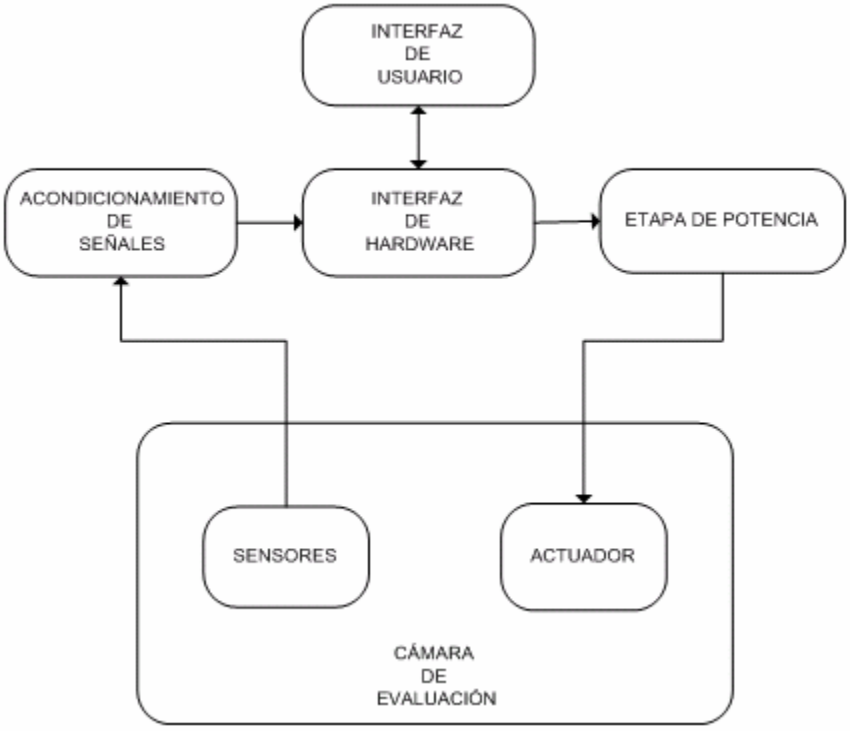



*Figura 1.3 Esquema del Probador de Prótesis Cardiacas a desarrollar.*

En el caso de la interfaz que se maneja actualmente en el INC, ésta cuenta con diversos módulos que sirven para el despliegue de los datos obtenidos de los sensores de presión. Muy aparte se tiene un sistema de control, el cual es el encargado de enviar los datos al sistema que simula las condiciones a las cuales se someten las válvulas cardiacas en el cuerpo humano.

Con el desarrollo de la nueva interfaz, el usuario tiene el control total del probador utilizando sólo un sistema que en este caso es una computadora, ya que las gráficas obtenidas a partir de los sensores son desplegadas en pantalla, y el envío de datos al actuador se hace desde el mismo programa.

En la figura 1.4 se muestran los módulos que se consideraron en el desarrollo de la interfaz:

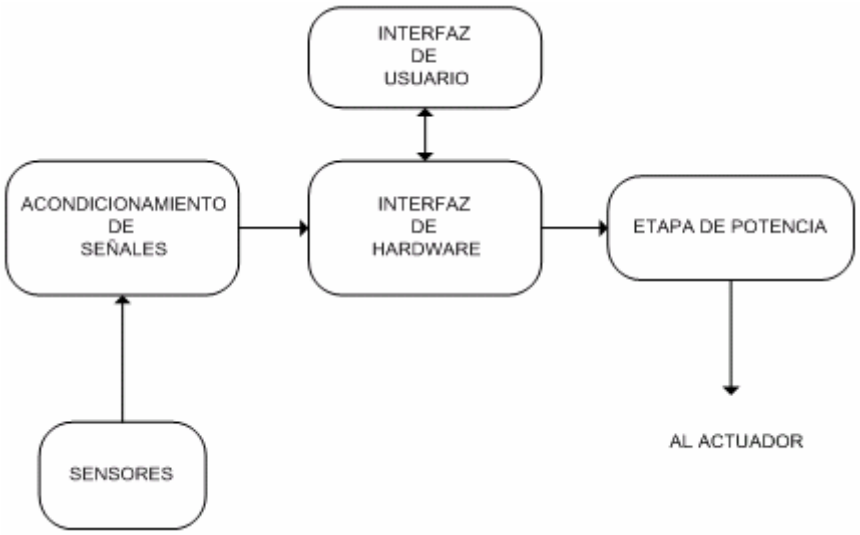



*Figura 1.4 Módulos del desarrollo del proyecto de tesis.*

De manera general lo que se realizó en cada bloque fue lo siguiente:

**Sensores**.-

La presión a medir por los sensores será de 120 mmHg para la presión sistólica y 80 mmHg para la presión diastólica, aproximadamente, pero para no encontrarnos restringidos se buscó un sensor que pudiera medir presiones mayores a las establecidas. Una vez que se eligió un sensor que cumpliera con las características deseables, se hicieron varias pruebas midiendo la variable de interés y con base en los resultados obtenidos se optó por elegir o no otro sensor. En este caso se utilizaron sensores de presión marca Honeywell, con un intervalo de medición de $\pm$ 5 psi, lo que equivale aproximadamente a $\pm$ 258 mmHg, con lo cual se cubrió el intervalo de presiones a medir en el probador.
Se utilizaron estos dispositivos ya que se pueden configurar fácilmente y son de fácil adquisición en el país.
Principalmente lo que se consideró del sensor fue que tuviera una respuesta rápida, buena resolución y un bajo costo.

**Acondicionamiento de señales.-**

Dado que los valores de voltaje de salida máximos de los sensores de presión se encontraron alrededor de los 115 mV, fue necesario amplificar dichas señales para lograr una mejor manipulación de las mismas. Además de esto, fue necesario el poder variar el nivel de referencia de las señales de salida, para lograr lo anterior

se utilizó un amplificador de instrumentación con ganancia y nivel de referencia variable. Gracias a esta configuración no fue necesario el uso de filtros para evitar los ruidos del ambiente y de la línea.

**Interfaz de hardware.-**

Debido a la necesidad de presentar en pantalla los datos provenientes de los sensores, fue necesario el utilizar una conversión analógica-digital para lograr dicho objetivo. Posteriormente esta información fue procesada de tal manera que pudiera ser enviada a una computadora a través del puerto serie de la misma.
Para lograr lo anterior se hizo uso de un microcontrolador PIC de Microchip de la serie 16F87X, el cual tiene incluido un convertidor analógico digital, además de contar con un módulo USART (*Addressable Universal Synchronous Asynchronous Receiver Transmitter*) gracias al cual se puede lograr una comunicación serial bajo la norma RS232.
Para este caso la programación se realizó en lenguaje C, ya que es mucho más rápida que en lenguaje Ensamblador, además de ser un poco más amigable.
La utilización de este dispositivo facilitó el control desde la computadora y la interacción de la interfaz de hardware con la de usuario, además de tener la ventaja de ser reprogramable.

**Interfaz de usuario.-**

Para la realización de la misma se hizo uso del lenguaje de programación Visual Basic 6.0, ya que con este lenguaje se pueden desarrollar interfaces de usuario amigables además de que para el programador el ambiente es totalmente amigable y muy parecido al lenguaje C en algunas funciones.
Este software incluye además un controlador para el puerto serie el cual fue de gran ayuda para lograr la interacción con la interfaz de hardware.
La interfaz realizada se encargó de desplegar en pantalla los datos provenientes de los sensores (además de otras funciones que se mencionan en los alcances) y al mismo tiempo enviar las señales de control hacia los actuadores.

**Etapa de potencia.-**

Esta etapa se realizó para obtener niveles de voltaje necesarios para el correcto funcionamiento de los actuadores, ya que al ser dispositivos de mayor consumo de corriente fue necesario darles un tratamiento especial, con esto se logró su correcto control.
Para el caso de la electroválvula la etapa constó del uso de un MOSFET de potencia y un diodo de protección. En el caso del control del motor de pasos, se diseñó un puente H, mismo que fue realizado por otra persona del Laboratorio de Electrónica.

## 1.3 Alcances

Con el desarrollo de la interfaz se tendrá un control total sobre el probador de prótesis cardiacas, tratando al mismo tiempo de brindar una mejor manipulación y una mayor facilidad de operación por parte del personal encargado.

La interfaz:

- Tendrá la opción de elegir el número de muestras a graficar y la frecuencia de muestreo.
- Recibirá y enviará datos en tiempo real.
- Desplegará en tiempo real la forma de onda de las señales provenientes de los sensores.
- Opcionalmente almacenará los datos recibidos, cada cierto tiempo, en un archivo de texto que contendrá además la fecha y hora de registro.
- Los datos de este archivo podrán ser exportados a una hoja de cálculo, esto para fines estadísticos.

Puesto que se trata de un proyecto que seguirá en desarrollo, el presente trabajo queda abierto a mejoras, por tal motivo el diseño presentado no será el definitivo, ya que atendiendo a las necesidades del personal del INC, la interfaz se verá modificada y mejorada.

# 2. SENSORES

## 2.1 Descripción

Los sensores de presión utilizados son de la marca Honeywell. Este tipo de dispositivos miniatura proveen de un sensado de presión confiable, siendo desarrollados en pequeños empaques. La siguiente figura muestra el sensor de presión utilizado.



*Figura 2.1 Sensor de presión modelo 24PCBFA6G.*

El sensor cuenta con 4 pines, dos para alimentación ($V_{CC}$ y GND) y los otros 2 para obtener el voltaje diferencial de salida ($V_O$).

La tecnología de estos dispositivos se basa en elementos de sensado que consisten de 4 elementos piezo-resistivos incrustados en la cara de un delgado diafragma de silicón, grabados químicamente. La característica principal de los elementos piezo-resistivos es la de modificar su resistencia cuando se les aplica una fuerza.  La figura 2.2 muestra cómo está constituido un sensor de este tipo.

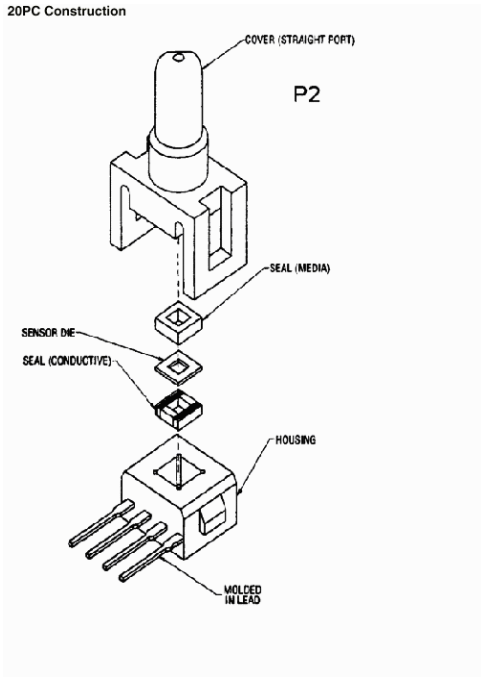



*Figura 2.2 Construcción del sensor de presión.*

Un cambio en la presión causa la flexión del diafragma induciendo una fuerza en el mismo y en las resistencias incrustadas. Los valores de las resistencias cambian en proporción a la fuerza aplicada y esto produce un cambio en la salida eléctrica. La configuración de estas resistencias es la de un circuito tipo puente.

En la figura 2.3 se puede observar el circuito básico de este tipo de puente, muchas veces llamado Puente de Wheatstone.

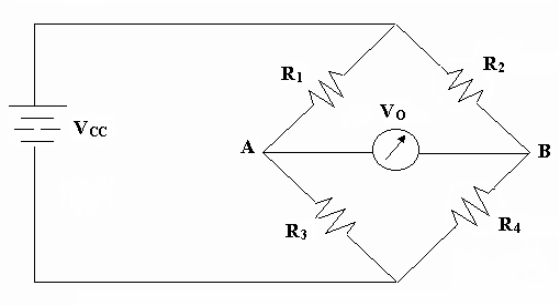



*Figura 2.3 Configuración básica de un puente de Wheatstone.*

Este tipo de circuito se usa principalmente para encontrar el valor desconocido de una de las resistencias del puente, conociendo previamente el valor de las otras tres. En este caso se usa como transductor en el que el cambio de valor en las resistencias tiene como resultado un cambio en el voltaje de salida.

En términos generales el sensor se somete a una presión lo que provoca la modificación de la resistencia de los elementos piezo-resistivos del puente obteniéndose un valor de voltaje diferencial de salida, proporcional a la presión ejercida sobre el sensor.

Así se tiene que:

$$V_O = V_A - V_B$$

$$V_O = \frac{R_3}{R_1 + R_3} V_{CC} - \frac{R_4}{R_2 + R_4} V_{CC}$$

$$V_O = \frac{R_3(R_2 + R_4) - R_4(R_1 + R_3)}{(R_1 + R_3)(R_2 + R_4)} V_{CC}$$

Finalmente:

$$V_O = \frac{R_3 R_2 - R_4 R_1}{(R_1 + R_3)(R_2 + R_4)} V_{CC} = R V_{CC}$$

Donde R es una variable que depende de la presión ejercida.

Como se puede observar el voltaje de salida $V_O$ es proporcional sólo a los cambios en las resistencias ya que el voltaje de control $V_{CC}$ tiene un valor constante.

## 2.2 Características

Con base en el diseño las características más importantes son:

- El bajo consumo, la no amplificación y la no compensación del diseño del circuito puente de Wheatstone proveen salidas del orden de los mV intrínsecamente estables sobre un rango 0.5 a 250 psi.
- Puede ser usado para medir presiones tanto positivas como negativas.
- Este sensor posee un rango de medición de ± 5.0 psi (± 258 mmHg), lo que para nosotros es más que suficiente ya que la presión sanguínea en un ser humano sano oscila entre los 80 y los 120 mmHg, y en el caso de que se simulen presiones mucho más bajas o altas, estas últimas no sobrepasarán el valor máximo de medición del sensor.
- La sensibilidad típica del sensor oscila alrededor de los 23 mV/psi, y puesto que tenemos un valor máximo de presión de 5.0 psi obtendremos a la salida un voltaje máximo aproximado de 115 mV. Es por esto que tal vez sea necesario el contemplar una amplificación de la señal antes de procesarla.
- Tiene un tiempo de respuesta máximo de 1 ms. Considerando las frecuencias a manejar este es un buen tiempo de respuesta.
- Está fabricado en un empaque pequeño lo cual facilita la manipulación del mismo.

## 2.3 Ventajas y desventajas

Dentro de las ventajas más importantes se tienen:

- ✓ Uso en sistemas que utilizan líquidos (la presión que se medirá es en un líquido).
- ✓ No posee un sistema de amplificación incluido, lo cual nos ayuda ya que podemos adecuarlo a nuestras necesidades.
- ✓ Es compacto, lo que facilita su manipulación.
- ✓ Solo se requiere alimentarlo con corriente directa y no precisa de configuraciones adicionales salvo que sean necesarias.
- ✓ El fabricante provee los esquemas de los circuitos necesarios para su calibración y compensación.
- ✓ Fácil adquisición aquí en México.
- ✓ Soporta sobrepresiones de hasta 20 psi.

Aunque las desventajas no son muchas cabe mencionar algunas:

- × No está compensado con respecto a la temperatura.

- Su salida es diferencial por lo cual son necesarios circuitos adicionales para obtener una señal de más fácil manipulación.
- Al estar conformado por elementos muy sensibles, para lograr una correcta calibración es necesario utilizar elementos de precisión.

## 2.4 Acondicionamiento de las señales de salida de los sensores

Debido a los niveles obtenidos a la salida de los sensores fue necesario el llevar a cabo el acondicionamiento de las señales para lo cual se plantearon los siguientes módulos:

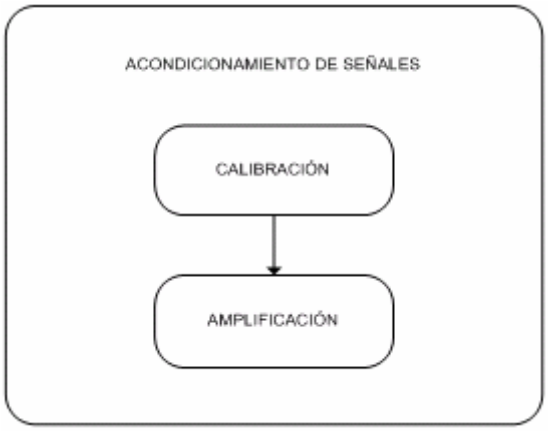



*Figura 2.4 Acondicionamiento de señales.*

CALIBRACIÓN

De acuerdo a especificaciones hechas por el fabricante se pueden hacer dos tipos de ajuste dependiendo del signo del offset para lograr un valor de voltaje igual a cero a la salida del sensor. Puesto que los sensores utilizados presentaron voltajes de offset positivos, se pudo haber utilizado la siguiente configuración para lograr un voltaje de salida igual a cero. Esto no se hizo ya que se utilizó otro tipo de configuración que se mostrará más adelante.
La figura 2.5 muestra la configuración que se utiliza para lograr un offset nulo a la salida del sensor.

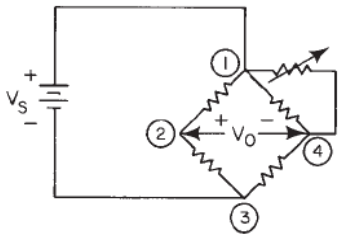



*Figura 2.5 Configuración para eliminar offset positivo.*

Como se puede observar en el circuito, sólo es necesario agregar una resistencia variable entre los pines 1 y 4 para lograr establecer un offset nulo, esto debido a que al tener dos resistencias en paralelo, el valor total se verá modificado al variar una de las resistencias, logrando así establecer un equilibrio del puente. Cabe señalar que la resistencia propuesta por el fabricante es de un valor alrededor de los 300k ohms. Lo antes mencionado sirve para calibrar el sensor dependiendo de las condiciones en las que tenga que ser operado.

AMPLIFICACIÓN

Una vez que se logra establecer el voltaje de salida a cero o a un cierto valor necesario, lo siguiente es conectar la salida del sensor a un circuito que amplifique la señal.

El uso de la configuración para la calibración se evitó ya que el mismo fabricante propone una configuración con cuatro amplificadores operacionales en la cual ya no es necesaria una etapa de calibración independiente. El circuito utilizado es el que se muestra en la siguiente figura:

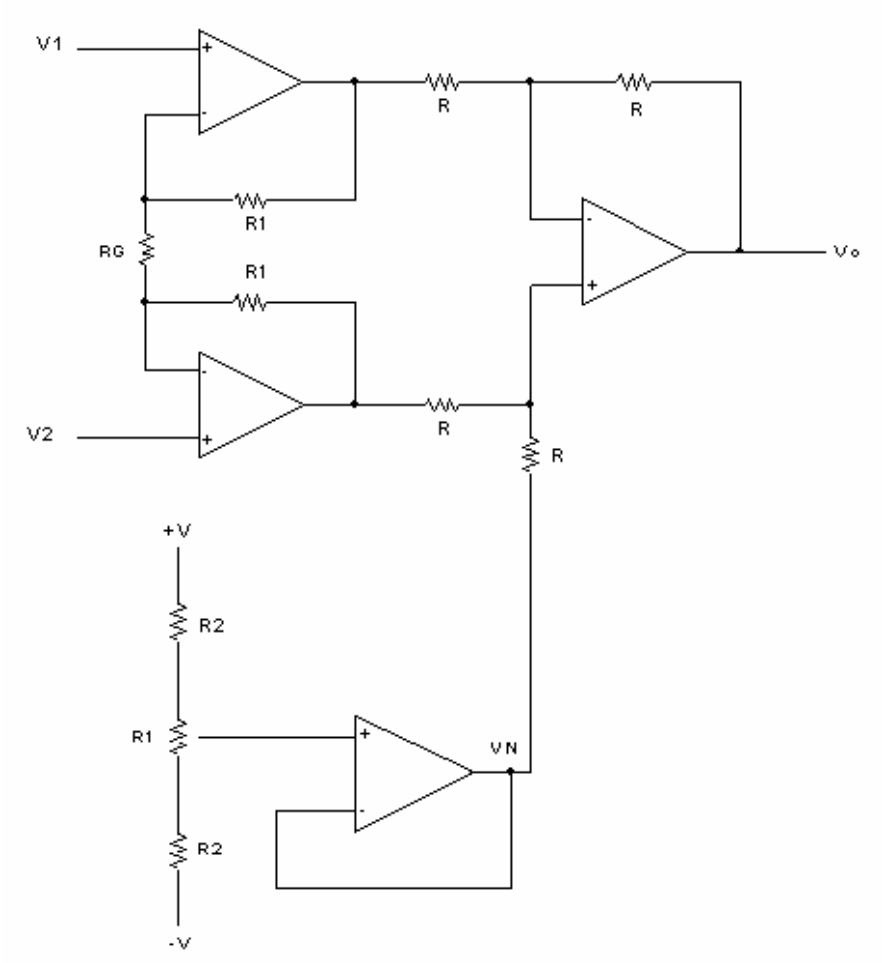



*Figura 2.6 Circuito para amplificación y calibración de la salida del sensor.*

La primera etapa, conformada por los primeros tres amplificadores, es conocida como amplificador de instrumentación. Esta configuración se caracteriza por tener una alta relación de rechazo al modo común (RRMC), lo que ayuda a eliminar gran parte del ruido presente en el medio ambiente. Esta configuración se utiliza mucho cuando es necesario amplificar señales pequeñas ya que se pueden ver más afectadas por las señales de ruido.
Como se puede ver en la figura 2.6 la referencia de este circuito no está conectada a tierra sino a un circuito amplificador seguidor que tiene la función de hacer variar el voltaje de referencia del amplificador de instrumentación, ya que a la señal de salida se le suma el voltaje obtenido del divisor de voltaje mostrado.
En términos generales toda la ganancia se coloca en los dos primeros amplificadores, con esto se logra hacer menos susceptible el circuito a errores en modo común. El tercer operacional se usa como un amplificador diferencial de ganancia unitaria y el cuarto como un seguidor cuya salida proporciona un nivel de referencia variable al tercer operacional.

En la primera etapa (amplificador de instrumentación) las señales provenientes de los sensores son sometidas a una diferencia entre ellas además de que la señal resultante es amplificada a un nivel deseado. Cabe señalar que las resistencias utilizadas para esta configuración fueron de precisión en cuanto a su valor ya que de ello depende el buen rechazo al modo común.
Para la segunda etapa (control del nivel de offset de salida) el valor de las resistencias del divisor de voltaje establece los niveles máximos y mínimos de voltaje que varían la referencia de la señal de salida del circuito. En este caso el divisor no se referenció a tierra sino a un voltaje negativo para obtener un mayor rango de variación.

Para la implementación del circuito mostrado anteriormente se utilizó el circuito integrado TL072 el cual es un amplificador operacional dual, es decir, cada circuito integrado contiene dos amplificadores operacionales internos. El diagrama de este circuito se muestra a continuación:

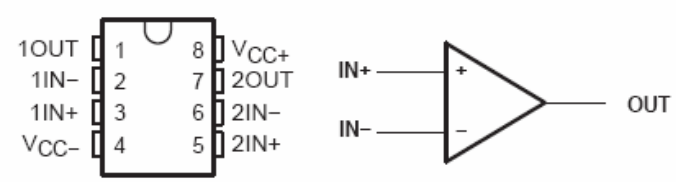



*Figura 2.7 Amplificador Operacional Dual TL072.*

Las principales características de este circuito son el bajo consumo de potencia, protección contra corto circuito a la salida, bajo ruido y una alta impedancia de entrada debido a que el amplificador está conformado por transistores de efecto de campo de juntura (JFET).

Tomando en consideración el circuito de la figura 2.6 la ecuación que lo rige es la siguiente:

$$V_O = (V_2 - V_1)\left(1 + \frac{2R_1}{R_G}\right) + V_N$$

donde:

$V_N$: voltaje de offset agregado a la señal de salida

Cabe mencionar que debido a que las frecuencias de las señales de entrada oscilan entre los 0.33333 y los 50 Hz, aproximadamente, no se tuvieron problemas con los tiempos de respuesta ni con los anchos de banda de los amplificadores, ya que por un lado los tiempos de respuesta son del orden de los microsegundos y el máximo valor del ancho de banda es del orden de los MHz.

La calibración y amplificación antes mencionadas, se hicieron por cada uno de los sensores, que en este caso fueron dos.

El circuito implementado quedó como se muestra a continuación:

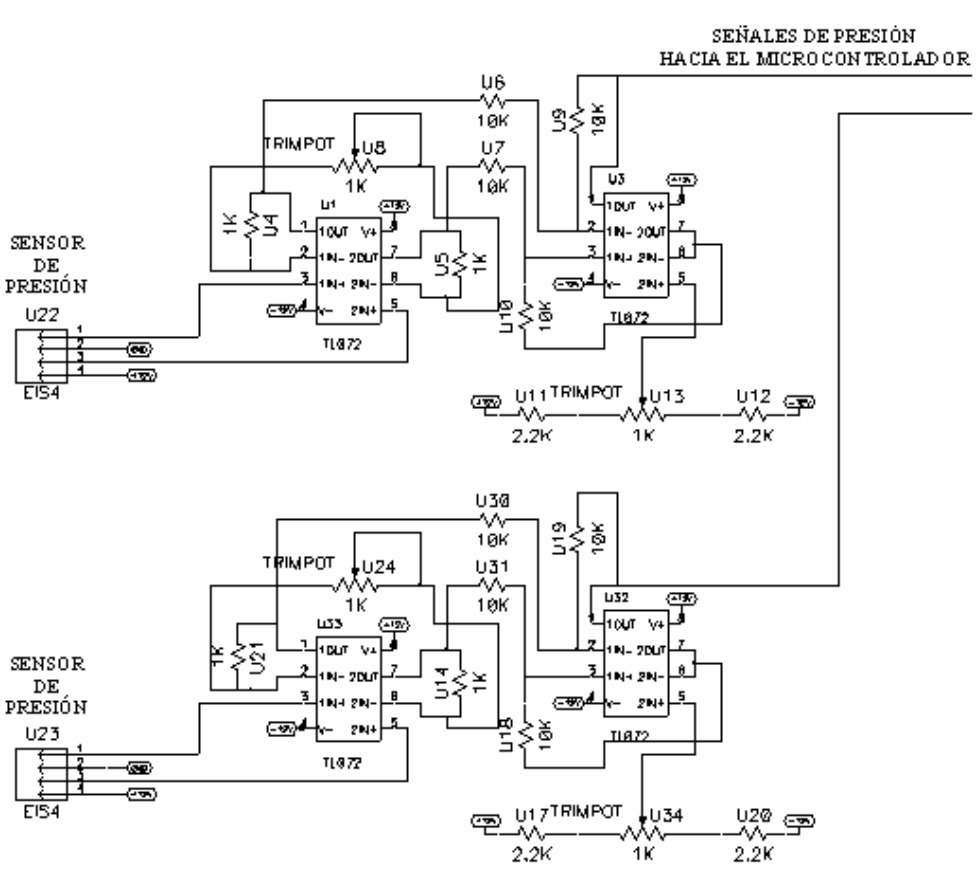



*Figura 2.8 Circuito utilizado para el acondicionamiento de señal.*

# 3. INTERFAZ DE HARDWARE

## 3.1 Microcontrolador

Un controlador es el dispositivo encargado de gobernar uno o varios procesos. Aunque el concepto permanece, su implementación física ha variado frecuentemente. En la actualidad, todos los elementos del controlador se incluyen en un chip el cual recibe el nombre de microcontrolador.
Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador consta de los siguientes componentes:

- Procesador o CPU (Unidad Central de Proceso)
- Memoria RAM para contener datos
- Memoria para el programa tipo ROM/PROM/EPROM/EEPROM
- Líneas de Entrada/Salida para comunicarse con el exterior
- Diversos módulos para el control de periféricos (temporizadores, puertos serie y paralelo, conversores analógico/digital (CAD), conversores digital/analógico (CDA), etc.
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema

### 3.1.1 Descripción

El microcontrolador utilizado es el PIC (Peripheral Interface Controller) de la serie 16F87X, el desarrollo de la interfaz se hizo con un 16F874A aunque se puede migrar fácilmente a otro de la misma serie, por ejemplo a un 16F877.

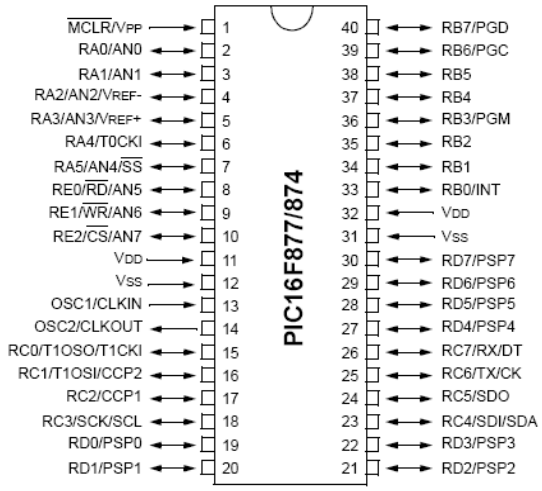



*Figura 3.1 Diagrama del PIC 16F874.*

Algunos aspectos importantes a considerar de estos microcontroladores son los siguientes:

*Arquitectura.-*

La arquitectura del procesador sigue el modelo Harvard. En esta arquitectura, la CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos.
La arquitectura Harvard permite a la CPU acceder simultáneamente a las dos memorias. Además, propicia numerosas ventajas al funcionamiento del sistema.

*Segmentación.-*

Se aplica la técnica de segmentación ("pipe-line") en la ejecución de las instrucciones.
La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).
Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

*Formato de las instrucciones.-*

El formato de todas las instrucciones es de la misma longitud.

Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

*Juego de instrucciones.-*

Procesador RISC (Computador de Juego de Instrucciones Reducido).

Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

*Instrucciones ortogonales.-*

Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

*Arquitectura basada en bancos de registros.-*

Esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

La arquitectura Harvard y la técnica de segmentación son los principales recursos en los que se apoya el elevado rendimiento que caracteriza estos dispositivos programables, mejorando dos características esenciales:

- Velocidad de ejecución
- Eficiencia en la compactación del código

### 3.1.2 Características

Para el caso de este proyecto se eligió el PIC 16F874A por tener convertidor analógico/digital, interrupciones, protocolo de transferencia de datos RS232 también llamado USART (Universal Synchronous Asynchronous Receiver Transmitter) que no es más que comunicación vía puerto serie; y varios puertos de entrada y/o salida.

En la siguiente tabla se muestran los elementos característicos de los PIC's que conforman la serie 16F87X.

| Key Features PICmicro™ Mid-Range Reference Manual (DS33023) | PIC16F873 | PIC16F874 | PIC16F876 | PIC16F877 |
|---|---|---|---|---|
| Operating Frequency | DC - 20 MHz | DC - 20 MHz | DC - 20 MHz | DC - 20 MHz |
| RESETS (and Delays) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) |
| FLASH Program Memory (14-bit words) | 4K | 4K | 8K | 8K |
| Data Memory (bytes) | 192 | 192 | 368 | 368 |
| EEPROM Data Memory | 128 | 128 | 256 | 256 |
| Interrupts | 13 | 14 | 13 | 14 |
| I/O Ports | Ports A,B,C | Ports A,B,C,D,E | Ports A,B,C | Ports A,B,C,D,E |
| Timers | 3 | 3 | 3 | 3 |
| Capture/Compare/PWM Modules | 2 | 2 | 2 | 2 |
| Serial Communications | MSSP, USART | MSSP, USART | MSSP, USART | MSSP, USART |
| Parallel Communications | — | PSP | — | PSP |
| 10-bit Analog-to-Digital Module | 5 input channels | 8 input channels | 5 input channels | 8 input channels |
| Instruction Set | 35 instructions | 35 instructions | 35 instructions | 35 instructions |

*Figura 3.2 Características clave de la familia de PIC's 16F87X.*

Tomando en consideración que el PIC 16F874A es el que se utilizó en el desarrollo del proyecto, a continuación se mencionan las características incluidas por las cuales se eligió:

## CONVERSIÓN ANALÓGICA/DIGITAL

Puesto que las señales obtenidas de los sensores de presión son del tipo análogo, es necesario el someterlas a un proceso de conversión analógico/digital. Con esto se establece una relación biunívoca entre el valor de la señal en su entrada y la palabra digital obtenida en su salida.

Módulo Conversor Analógico/Digital.- el PIC utilizado cuenta con 8 canales de conversión multiplexados hacia un solo convertidor. La siguiente figura muestra el diagrama a bloques del convertidor A/D del PIC.

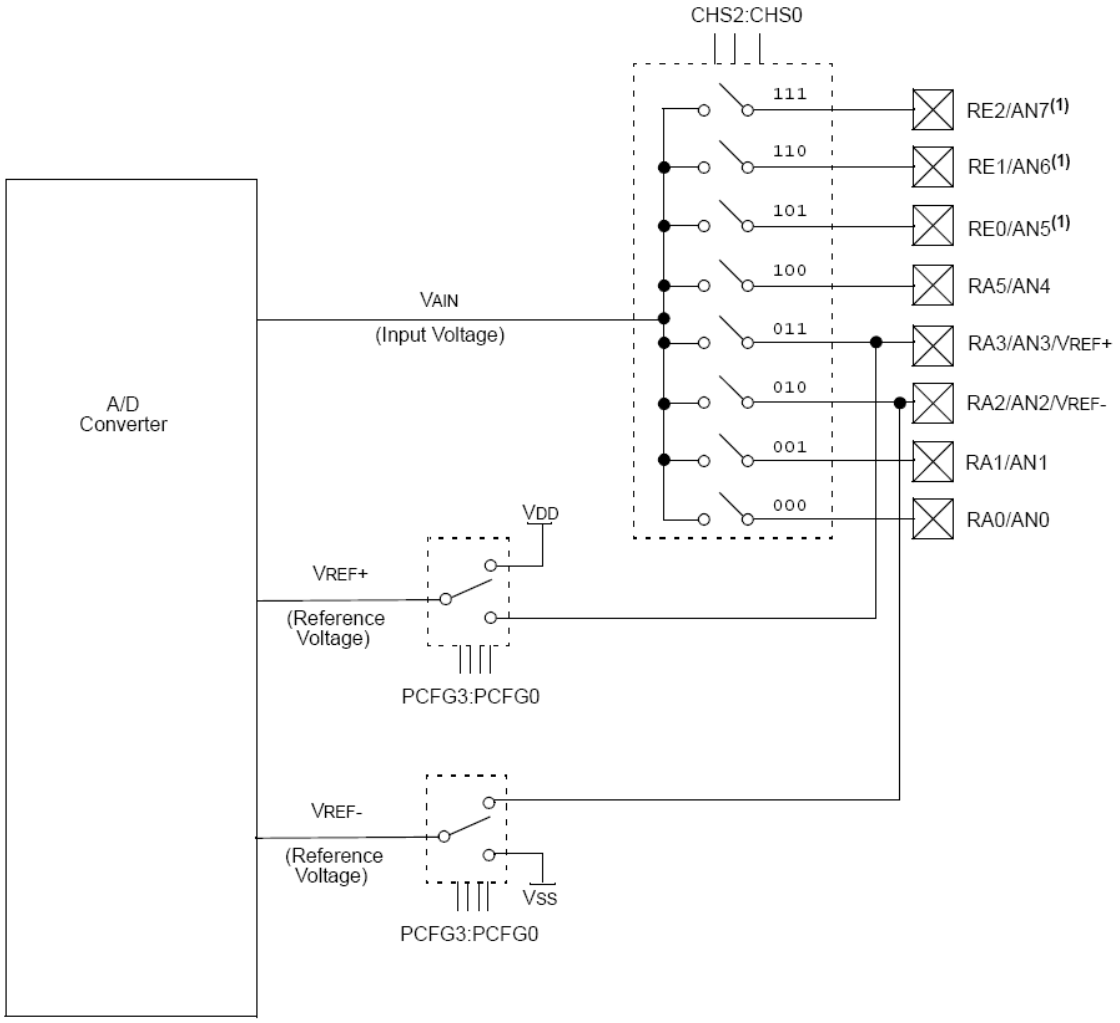



*Figura 3.3 Diagrama a bloques del Convertidor A/D.*

La entrada analógica carga un capacitor de muestreo y retención (sample and hold), la salida de este capacitor es la entrada hacia el convertidor. El convertidor genera entonces un resultado digital de este nivel analógico vía aproximación sucesiva.

La conversión analógico/digital de la señal analógica de entrada resulta en un número digital correspondiente de 10 bits.

INTERRUPCIONES

Debido a que es necesario enviar y recibir datos en tiempo real y al mismo tiempo, se decidió hacer uso de las interrupciones, una para la adquisición de datos a graficar, otra para generar la señal de control de los actuadores y finalmente una para detener el programa en cualquier momento de la prueba. El principio de funcionamiento de las interrupciones es el siguiente: cuando se da un evento en un determinado módulo, la bandera asociada se pone a 1 y, si las máscaras global (GIE) y particular (y en algunos casos la de los periféricos (PEIE)) están habilitadas, se acepta la interrupción. El programador debe asegurarse de que el código se encargue de identificar la fuente de la interrupción, guardar y recuperar el contexto existente antes de producirse la interrupción y poner a cero la bandera asociada a la misma para permitir posteriores identificaciones.

Interrupción del Módulo Timer0.- La interrupción del timer0 es generada cuando el registro TMR0 se desborda desde FFh a 00h.

Interrupción del Módulo Timer1.- Se genera la interrupción cuando el registro TMR1 se desborda de FFFFh a 0000h.

COMUNICACIÓN SERIE

Puesto que el control se haría desde una computadora se decidió utilizar la comunicación serial, en este caso los datos se envían bit a bit por una misma línea durante un tiempo fijo. La velocidad de transmisión se indica en baudios (número de bits enviados por segundo). La transferencia puede ser síncrona o asíncrona:

Síncrona: se envía la señal de reloj para sincronizar cada bit
Asíncrona: se necesitan “relojes” en el emisor y el receptor de la misma frecuencia y en fase.

La norma RS232 nace con la idea de comunicar una computadora o equipo terminal de datos (DTE – Data Terminal Equipment) y un módem o equipo de comunicación de datos (DCE – Data Communications Equipment). Su uso se extendió a otras aplicaciones y hoy se emplea para comunicar equipos que no respetan íntegramente la norma y que no necesitan la mayoría de las líneas de la norma original. Se emplea por ejemplo, para comunicar equipos que son ambos DTE, como dos PC’s o una PC y un microcontrolador.

Módulo Transmisor Receptor Síncrono Asíncrono Universal Direccionable (USART).- La USART puede ser configurada como un sistema asíncrono full duplex que se puede comunicar con dispositivos periféricos tales como una computadora personal. En este modo la USART usa el formato estándar NRZ

(non-return-to-zero), donde se tiene un bit de inicio, 8 o 9 bits de datos y un bit de parada.

PUERTOS DE ENTRADA/SALIDA

Tanto para el envío como para la recepción de datos y el control de los actuadores fue necesario establecer pines que cumplieran con las diferentes funciones. En este caso el PIC 16F874A cuenta con 5 puertos de entrada/salida configurables, algunos pines para estos puertos de entrada/salida son multiplexados con funciones alternas para las características periféricas del dispositivo. En general, cuando un periférico es habilitado ese pin podría no ser utilizado como un pin de entrada/salida de propósito general.
En el diagrama de la figura 3.1 se pueden observar las distintas funciones que pueden adquirir los pines del microcontrolador.

### 3.1.3 Ventajas y desventajas

Las ventajas que se mencionan a continuación son de las más importantes y gracias a las cuales los PIC's han tenido una gran demanda en los últimos años:

- ✓ Sencillez de manejo: tienen un juego de instrucciones reducido; 35 en la gama media. Además de que gracias a ciertos compiladores es posible programarlos en lenguaje C.
- ✓ Buena información, fácil de conseguir y económica.
- ✓ Precio: su coste es comparativamente inferior al de sus competidores.
- ✓ Poseen una elevada velocidad de funcionamiento. Buen promedio de parámetros: velocidad, consumo, tamaño, alimentación, código compacto, etc.
- ✓ Herramientas de desarrollo fáciles y baratas. Muchas herramientas software se pueden recoger libremente a través de internet desde Microchip.
- ✓ Existe una gran variedad de herramientas hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC's.
- ✓ Diseño rápido.
- ✓ La gran variedad de modelos de PIC's permite elegir el que mejor responde a los requerimientos de la aplicación.

Una de las razones del éxito de los PIC's se basa en su utilización. Cuando se aprende a manejar uno de ellos, conociendo su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo.

En relación con las desventajas no conozco hasta ahora ninguna, salvo que dependiendo del modelo de PIC puede tener algunas limitaciones pero por esta razón existen diferentes modelos de PIC dependiendo de la utilización que se les vaya a dar.

## 3.2 Otros circuitos

En la implementación del hardware y específicamente en la configuración de la comunicación serial fue necesario acoplar los niveles de voltaje del microcontrolador con los del puerto de la computadora (TTL con RS232). Existen varios circuitos integrados comerciales que realizan esta adaptación con muy pocos componentes, uno muy usado es el MAX232.

La figura 3.4 muestra el diagrama de este circuito integrado.

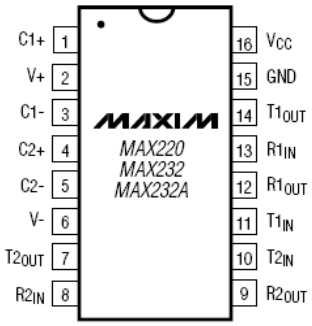



*Figura 3.4 Diagrama del circuito integrado MAX232.*

El circuito integrado MAX232 cambia los niveles TTL a los del estándar RS232 cuando se hace una transmisión, y cambia los niveles RS232 a TTL cuando se tiene una recepción. Esto se logra con ayuda de solo 4 capacitores electrolíticos de 22 μF.

Una de las ventajas de este circuito es que es de bajo consumo y se alimenta con 5 Volts (nivel TTL), lo que es de gran ayuda ya que el microcontrolador se alimenta de igual manera.

## 3.3 Desarrollo de la interfaz de hardware

A continuación se describen los circuitos electrónicos utilizados en el desarrollo de la interfaz de hardware, desde su implementación, la conexión con la interfaz de software a través de comunicación serial hasta la programación del microcontrolador.

### 3.3.1 Implementación de la interfaz de hardware

Una vez obtenidas las señales provenientes de los sensores, estas se conectaron a dos canales del convertidor analógico/digital del PIC. Aquí ya no fue necesario ningún tipo de circuito de acoplamiento, ya que los circuitos utilizados para el acondicionamiento de las señales de salida de los sensores proveen de una alta

impedancia de entrada antes de conectar los sensores los cuales presentan resistencias de salida de aproximadamente 5k ohms.
Básicamente en el microcontrolador se reciben los datos de los sensores, estos son procesados y finalmente son enviados a través del puerto serie hacia la computadora, aunque también cabe mencionar que desde la computadora se envían datos al PIC, los cuales son procesados y enviados hacia la etapa de potencia de los actuadores.

El siguiente diagrama muestra las etapas realizadas dentro del microcontrolador.

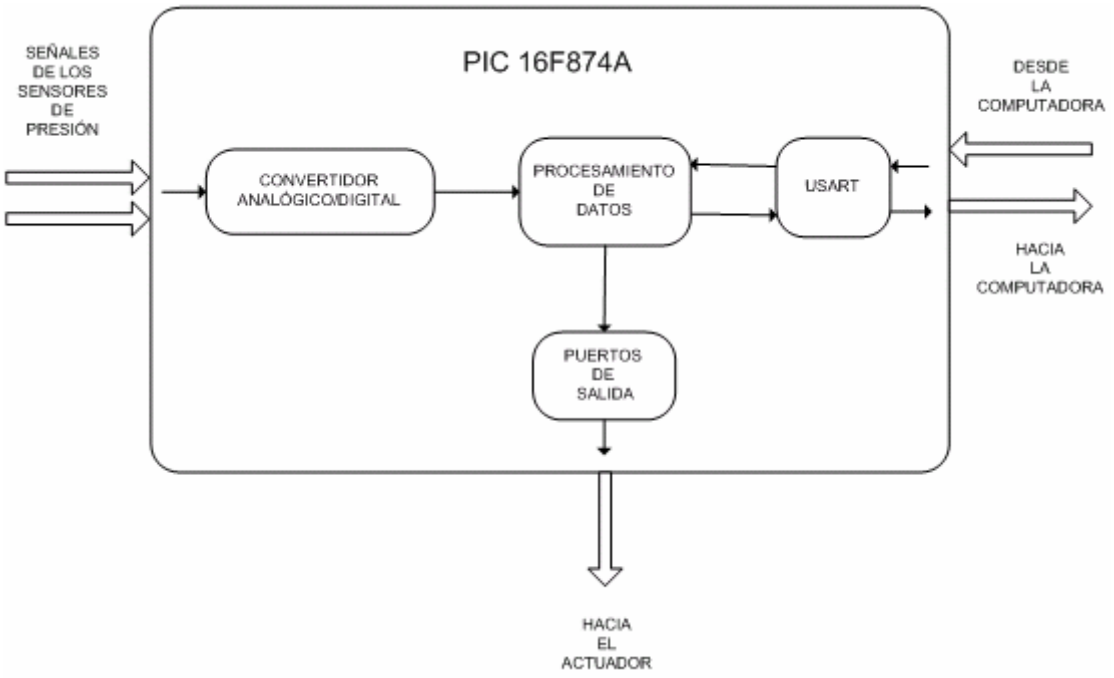



*Figura 3.5 Etapas a realizar en el microcontrolador.*

CONVERSIÓN ANALÓGICA/DIGITAL

Una vez que las señales provenientes de los sensores de presión entran por dos de los canales del PIC, estas se pasan por un circuito de muestreo y retención y posteriormente al convertidor analógico/digital donde por medio de aproximaciones sucesivas se digitaliza cada valor analógico obtenido de la señal.

Un circuito de muestreo y retención es un elemento de memoria analógica que almacena una tensión dada en un condensador. Este condensador está conectado a un interruptor electrónico de forma que cuando el interruptor se cierra el condensador se carga a la tensión de entrada; cuando el interruptor se abre el condensador retiene esta carga, congela la tensión durante un período de tiempo especificado. La figura 3.6 muestra el diagrama de un circuito de muestreo y retención básico.

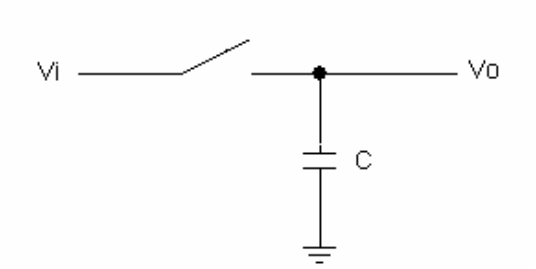



*Figura 3.6 Circuito básico de Muestreo y Retención.*

Un temporizador (dispositivo de control de muestreo) conectado al interruptor, hace que éste tome muestras de la amplitud de la señal analógica a intervalos de tiempo fijo. El número de muestras tomadas por segundo se conoce como Frecuencia de Muestreo.

Para el caso del muestreo de la señal se consideró el Teorema de Nyquist que dice que la frecuencia de muestreo debe ser mayor o igual a dos veces la frecuencia de la señal a muestrear. Para este caso práctico se consideró inicialmente una frecuencia de muestreo igual a 30 veces la frecuencia de la señal a muestrear aunque se observó que esta frecuencia puede ser menor.

Algo muy importante de considerar es el tiempo de adquisición y el tiempo de conversión. El tiempo de adquisición es el tiempo necesario para cargar el condensador de mantenimiento (el condensador del circuito de muestreo y retención). De la hoja de especificaciones del PIC se obtiene la ecuación para obtener el mínimo tiempo de adquisición que es:

$T_{ACQ}$ = Amplifier Settling Time +
Hold Capacitor Charging Time +
Temperature Coefficient

$T_{ACQ}$ = TAMP + TC + TCOFF

$T_{ACQ}$ = 2µs + TC + [(Temperature - 25°C)(0.05µs/°C)]

TC = CHOLD (RIC + RSS + RS) In(1/2047)

TC = - 120pF (1k + 7k + 10k) In(0.0004885)

TC = 16.47µs

$T_{ACQ}$ = 2µs + 16.47µs + [(50°C -25°C)(0.05µs/°C)

$T_{ACQ}$ = 19.72µs

Este último valor se obtiene considerando que la impedancia máxima de la fuente a conectar a la entrada del PIC debe ser de 10k ohms. Si ésta impedancia disminuye, el tiempo de adquisición también lo hace. Este valor es muy importante ya que de ello depende que se obtenga o no una buena conversión.
El tiempo de conversión se refiere al tiempo que el microcontrolador tarda en hacer la conversión del dato analógico a digital. Este tiempo se denomina $T_{AD}$ y para la conversión de un bit el tiempo es de un $T_{AD}$ que como mínimo debe ser de 1.6 microsegundos aunque este valor puede ser configurable.
El tiempo mínimo a considerar antes de comenzar otra adquisición es de $2T_{AD}$. Para el caso de 10 bits el tiempo debe ser de $12T_{AD}$. La siguiente figura muestra el tiempo de conversión dependiendo del número de bits.

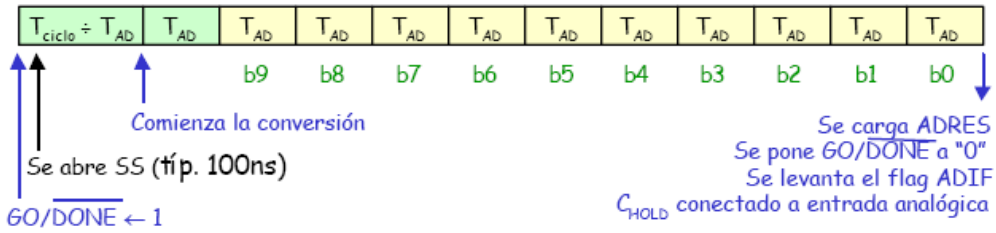



*Figura 3.7 Tiempos de conversión.*

El método de aproximaciones sucesivas es el implementado en los microcontroladores PIC. Consiste en ir poniendo a "1" cada bit comenzando por el más significativo. Por lo tanto, considerando que se trabaja con 4 bits de resolución, el primer paso es colocar a 1 el bit 3 (sería 1000=8 en decimal) y realizar la comparación.
Si por ejemplo, el comparador indica que la entrada analógica es menor que la salida del amplificador, significa que el valor es inferior a 8 con lo cual se sabe que se está en la mitad inferior del byte (0 a 7) y por lo tanto el bit 3 debe ser apagado (ponerse a 0), esto ya evita tener que comparar la mitad superior (8 a 16). El paso siguiente es poner a 1 el bit 2 y de esta forma se checa si el valor está entre 0 y 3 o entre 4 y 8. Dependiendo de la salida del comparador se deja en 1 o en 0 el bit 2. Y así sucesivamente se van verificando los restantes bits (0 a 1). El resultado es el valor digital correspondiente a la conversión.
Este método solo requiere de 4 comparaciones (trabajando con resoluciones de 4 bits) para detectar cualquier valor de entrada a diferencia del método secuencial que requeriría de 16 comparaciones en el caso extremo.
La siguiente figura muestra en términos generales como se realiza el proceso de aproximaciones sucesivas.

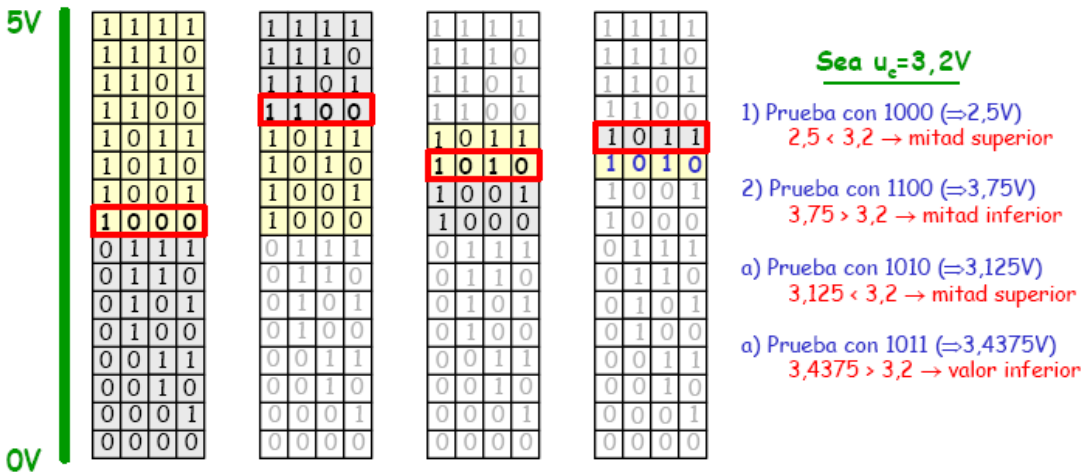



*Figura 3.8 Método de Aproximaciones Sucesivas.*

Dependiendo del número de bits de resolución es el número de comparaciones que realiza el microcontrolador en este proceso. En este caso la resolución del convertidor es de 10 bits por lo que el número de comparaciones es igual a 10.
Este método ofrece una gran exactitud combinada con una gran velocidad de conversión.

PROCESAMIENTO DE DATOS

Una vez que los datos analógicos son convertidos en datos digitales, estos son procesados de tal forma que puedan ser enviados hacia la computadora. Puesto que los datos tienen una resolución de 10 bits, es necesario “partir” ese dato ya que sólo pueden ser enviados datos de 8 bits a través del puerto serie.
En este caso gracias a funciones establecidas en el compilador del programador del PIC se pudo hacer esto de manera muy fácil.
Más adelante se mostrarán las funciones que se utilizaron para realizar este proceso.

Para el caso de los datos recibidos por el puerto serie desde la computadora, estos pasan por un proceso de “unión”, ya que llegan separados, por lo que al ser recibidos por el microcontrolador se les efectúan ciertas operaciones matemáticas o bien se les aplica una función, así por ejemplo si se envía desde la computadora un 1000 este dato tiene que ser “partido” en dos para poder ser enviado. Así entonces:

$$dato_alto = entero\left(\frac{1000}{256}\right) = 3$$

en este caso es igual a 3 ya que sólo se toma la parte entera del cociente de la división.

Para el caso del dato_bajo se aplica una función que sólo regrese el residuo de la división, por lo que se tiene:

$$dato_bajo = residuo\left(\frac{1000}{256}\right) = 232$$

Ya en el microcontrolador estos datos se tienen que unir nuevamente para poder ser interpretados correctamente, lo realizado fue lo siguiente:

$$dato = dato_alto * 256 + dato_bajo = 3 * 256 + 232 = 1000$$

con esto se obtuvo el dato original.

En general el proceso realizado a los datos en el microcontrolador se resume a operaciones en las que se tengan que “unir” o “partir” datos.

USART

En este caso gracias al módulo USART y con base en la norma RS232 fue posible enviar y recibir datos a través del puerto serie de la computadora. En el microcontrolador se tiene un pin de recepción y un pin de transmisión. Estos pines no son conectados directamente al puerto de la computadora sino que se conectan a un circuito acoplador de niveles de tensión, esto porque el PIC maneja niveles TTL y el puerto de la computadora maneja niveles establecidos por la misma norma RS232 y los cuales son positivos y negativos y de amplitud mayor a 5 volts.
En términos generales el módulo USART ayuda a enviar y recibir datos mediante la comunicación serial. Con esto se pueden controlar diversos dispositivos desde una computadora con ayuda del microcontrolador.

PUERTOS DE SALIDA

Aunque el microcontrolador tiene la ventaja de tener puertos de entrada/salida, estos sólo se utilizaron como salida para enviar señales que controlan ciertos dispositivos, en este fueron los actuadores antes mencionados.

### 3.3.2 Uso de la comunicación serie para la interconexión de la interfaz de hardware con la de usuario

La comunicación realizada fue del tipo serial asíncrona, ya que es una de las formas de conectar dos dispositivos, específicamente el microcontrolador con una computadora.
En este tipo de comunicación los bits de datos se transmiten en serie, es decir, uno detrás de otro y cada dispositivo tiene su propio reloj. Previamente se debe acordar que se transmitirán los datos a la misma velocidad.

Los datos son encapsulados en tramas como la que se muestra en la siguiente figura.




*Figura 3.9 Trama de un dato serie.*

Primero se envía un bit de inicio, a continuación los bits de datos (primero el bit más significativo) y finalmente el bit de parada.
El número de bits de datos y bits de parada es uno de los parámetros configurables, así como el criterio de paridad par o impar para la detección de errores. Normalmente las comunicaciones serie tienen los siguientes parámetros: 1 bit de inicio, 8 bits de datos, 1 bit de parada y sin paridad.
La siguiente figura muestra el ejemplo de la transmisión del dato binario 10011010.

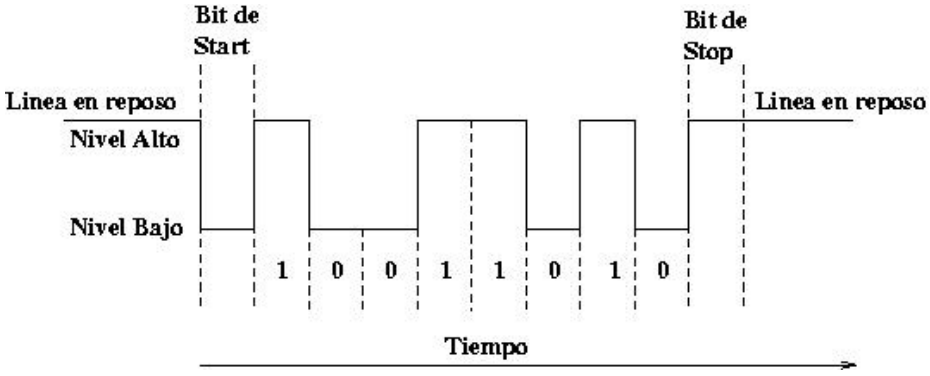



*Figura 3.10 Transmisión del dato binario 10011010.*

Para conectar la computadora a un microcontrolador a través del puerto serie, se utilizan las señales TX, RX y GND. La computadora utiliza la norma RS232 por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 volts. Los microcontroladores normalmente trabajan con niveles TTL (0-5V). Es por esto que es necesario el uso de un circuito acoplador de niveles de tensión como se muestra en la figura 3.11.

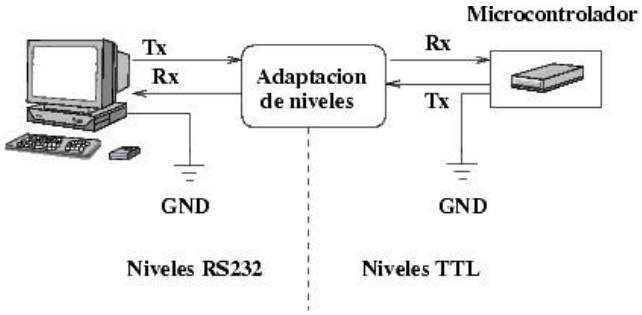



*Figura 3.11 Acoplamiento de niveles de tensión.*

Como ya se había mencionado anteriormente se hizo uso del circuito MAX232 el cual se configura como se muestra a continuación.

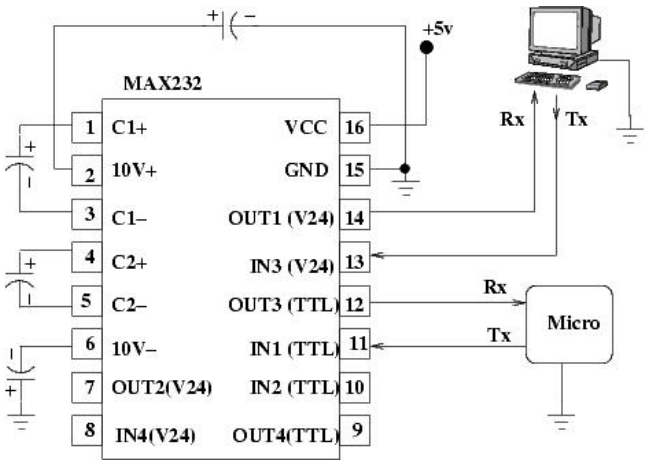



*Figura 3.12 Configuración del circuito integrado MAX232.*

Aunque existen otros procedimientos para realizar el acoplamiento de niveles de tensión el más fácil de todos es el uso de un circuito integrado ya que sólo se necesitan algunos componentes adicionales como capacitores lo cual facilita su configuración.

Para conectar la interfaz de hardware con la computadora se utilizó un cable cruzado en cuyos extremos se usaron conectores DB9. Aunque este tipo de conectores tiene nueve pines solo se utilizaron el de transmisión (Pin 3 - TX), el de recepción (Pin 2 - RX) y tierra (Pin 5 - GND). Las figuras siguientes muestran los conectores DB9 y un cable serie.

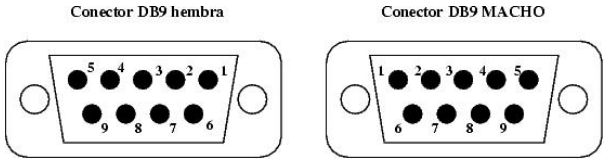



*Figura 3.13 Conectores DB9, hembra y macho.*



*Figura 3.14 Cable Serie.*

En esta conexión se pudo haber utilizado otro tipo de conector pero como se conectó a una computadora lo mejor fue usar el ya mencionado.

A continuación se muestra el circuito utilizado para la implementación de la interfaz de hardware:

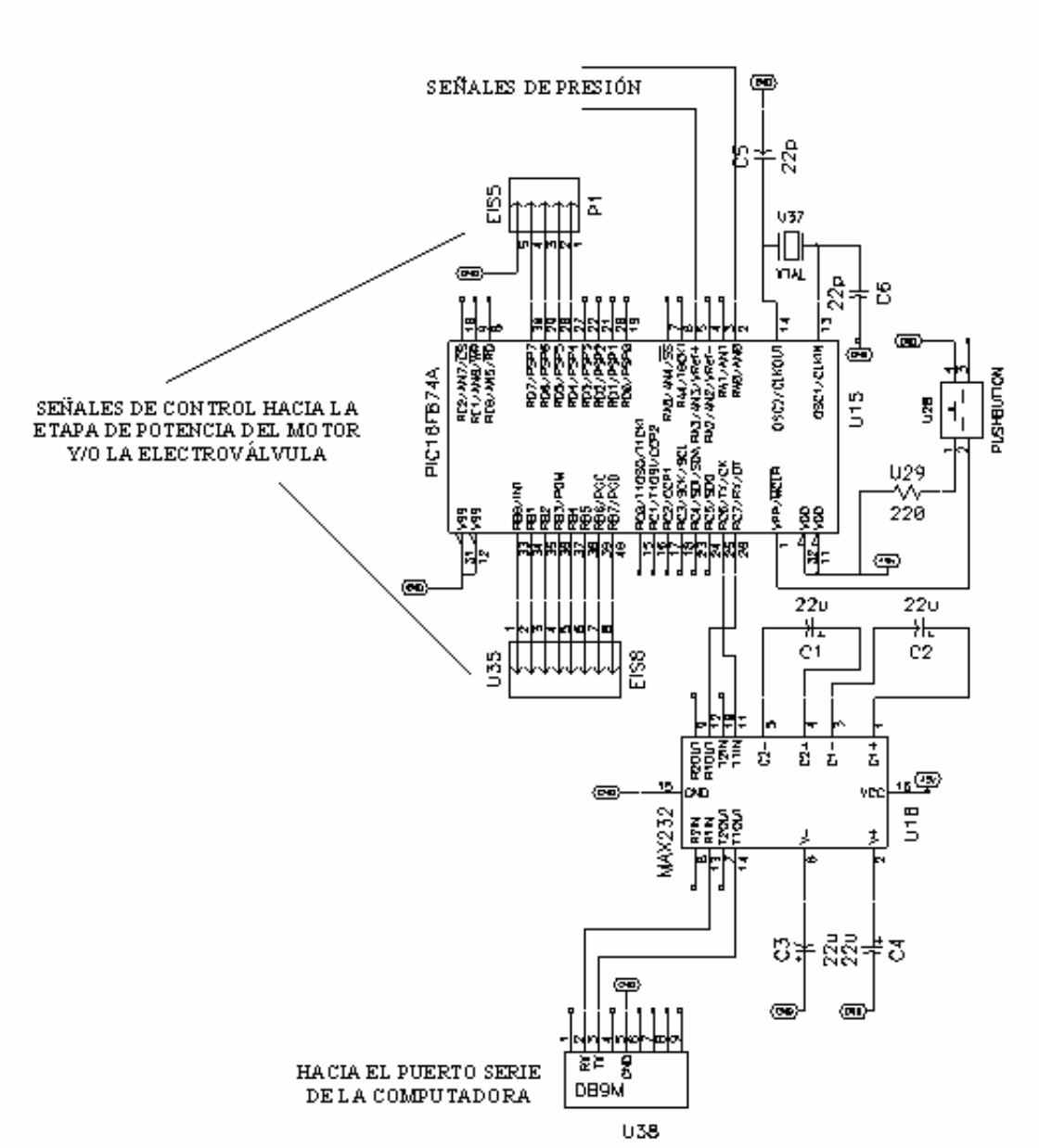



*Figura 3.15 Circuito para la interfaz de Hardware.*

### 3.3.3 Programación del microcontrolador para la recepción, procesamiento y transmisión de datos

La programación del microcontrolador se hizo en lenguaje C debido a que fue mucho más rápido hacerlo así que en ensamblador aunque se utilizó más memoria.

Las funciones principales del microcontrolador fueron:

- Convertir los datos provenientes de los sensores de presión en datos digitales para así poder procesarlos y enviarlos a la computadora.

- Recibir datos desde la interfaz para generar la señal de reloj que determina la frecuencia de operación del actuador. Parte de los datos recibidos son para enviar los bits de control al mismo actuador, y otros son para configurar la frecuencia a la cual son muestreadas las señales provenientes de los sensores.

El programa realizado se basó principalmente en el uso de interrupciones. Se utilizaron dos interrupciones por desbordamiento del timer interno del microcontrolador y la interrupción por puerto serie la cual se activa cuando se recibe un caracter en el microcontrolador.

En la función principal del programa se reciben los datos que sirven para establecer la frecuencia de muestreo de las señales, la frecuencia de la señal de reloj para el actuador, los bits de control del actuador; además de esto se configuran los timers utilizados. Puesto que muchos de los datos que recibe el microcontrolador son de más de 8 bits fue necesario el enviarlos separados desde la interfaz de usuario y ya en el microcontrolador se volvieron a unir. Esto se hizo debido a que el PIC no puede recibir datos de más de 8 bits.
Con ayuda de ciertas funciones del compilador fue posible lograr unir los datos, por ejemplo, si desde la interfaz de usuario se recibe un número mayor a 256 éste vendrá dividido en parte alta y parte baja. Entonces lo que se hace es tomar la parte alta del número y multiplicarla por 256 posteriormente se suma a la parte baja y finalmente se obtiene el número deseado.

```
while(!kbhit()); //Espera la recepción de un caracter
datoalto=getc();
while(!kbhit());
datobajo=getc();

freqmot=(datoalto*256)+datobajo;

o bien, se le puede aplicar la siguiente función:

freqmot=make16(datoalto,datobajo);
```

una vez recibidos todos los datos, algunos de ellos se utilizaron como bits de control, es decir, ciertos pines del microcontrolador se envían a niveles altos o bajos dependiendo de sus funciones. Para esto se utilizó la siguiente función del compilador:

```
output_bit(PIN_D7,1);
```

Con esta función se selecciona el pin a utilizar y se establece el nivel que tendrá de salida.

En el caso de la recepción de datos provenientes de los sensores se utiliza una de las interrupciones por desbordamiento. Aquí lo que se hace es establecer un tiempo de desbordamiento, con esto se fija una frecuencia de muestreo y se logra variar con ayuda de comparaciones. A continuación se muestra la función utilizada:

```
void adquisicion()
     {
          if(--conteo==0)
               {
                    set_adc_channel( 0 );
                    adquirir();
                    set_adc_channel( 3 );
                    adquirir();
                    conteo=periodo;
               }
          set_timer1(60536);
     }
```

En este caso el valor de la variable “conteo” es recibido desde la interfaz de usuario, este valor se decrementa y se compara con cero y si es igual se entra a la subrutina donde se establece el canal a muestrear y posteriormente se llama a la función “adquirir” que es la encargada de leer el convertidor, dividir el dato y enviarlo hacia la computadora. Aquí se divide el dato puesto que es de 10 bits.

```
void adquirir()
     {
          delay_us(30);
          dato = read_adc();
          datolow=dato;
          datohigh=make8(dato,1);
          putc(datohigh);
          putc(datolow);
     }
```

Si la variable “conteo” no es igual a cero, sólo se restablece el tiempo de desbordamiento del timer y se reinicia el conteo hasta que ocurra otro desbordamiento y se vuelva a llamar a la función de interrupción.

La otra de las interrupciones por desbordamiento se utilizó para generar la señal de reloj la cual se envía al actuador. Básicamente lo que se hace es cambiar el estado del pin cada que se desborda el timer. Puesto que la frecuencia de estos cambios tiene que ser variable se fija el tiempo de desbordamiento y con ayuda de comparaciones dentro de la función de interrupción se logra la variación. Así por ejemplo se tiene que:

```
void pulso()
{
     if(--retardo==0)
          {
               output_bit(PIN_D6,!bit_test(port_d,6));
               retardo=freqmot;
          }
     set_rtcc(62);
}
```

En este caso la función de interrupción se llama “pulso()”, dentro de esta se hace una pregunta de comparación, en la cual, si la variable decrementada “retardo” es igual a cero se cambia el estado del pin de un nivel alto a bajo o viceversa, según sea el caso.

A continuación se presenta la imagen de la interfaz de hardware.

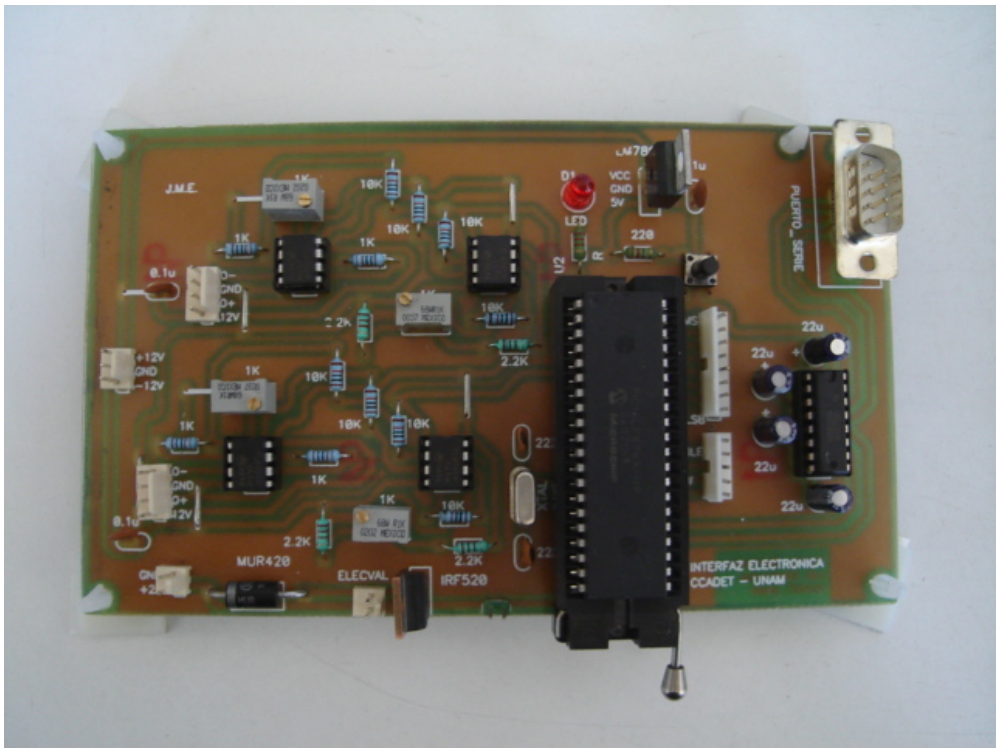



*Figura 3.16 Interfaz de hardware.*

# 4. ETAPA DE POTENCIA

En este capítulo se describirá el circuito encargado de interconectar la interfaz de hardware con el actuador, que en este caso es la electroválvula debido a que la etapa de potencia para el motor fue realizada por otra persona.

## 4.1 Diseño y desarrollo del circuito de potencia

El circuito de potencia tuvo su base en dos especificaciones importantes; el voltaje y la corriente que demandaría la electroválvula. Estos datos fueron establecidos por los compañeros encargados del diseño de la electroválvula y los valores especificados fueron: Voltaje = 25 V y Corriente = 0.4 A. Con estos datos se procedió al diseño del circuito.

Básicamente fueron dos los dispositivos empleados para la realización del circuito: un MOSFET de potencia (IRF520) y un diodo de potencia (MUR420). Las siguientes figuras muestran los símbolos de ambos dispositivos:

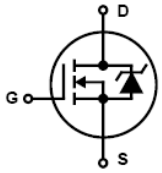



*Figura 4.1 Símbolo del IRF520.*

El IRF520 es un transistor de efecto de campo de poder, el cual es muy utilizado como switch en varias configuraciones. Las principales características por las cuales se eligió son:

- Manejo de corriente de 9.2 A y voltaje de 100 V
- Resistencia $r_{DS(ON)}$ = 0.270 ohms
- Velocidades de switcheo del orden de los nanosegundos
- Alta impedancia de entrada

Una de las principales ventajas que se tuvo con este transistor es que no fue necesario el uso de disipador ya que debido a las especificaciones de operación de la electroválvula, el valor aproximado de potencia disipada fue de 10 W aproximadamente, una sexta parte del valor máximo permitido por el dispositivo. Otra ventaja fue que el mismo dispositivo aísla el sistema de baja potencia del de alta potencia, debido a que se trata de un transistor de compuerta aislada. Con esto se evitó el uso de un optoacoplador.

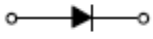


*Figura 4.2 Símbolo del MUR420.*

El MUR420 es un diodo rectificador de potencia diseñado especialmente para el uso en circuitos de switcheo, en este caso se usó como diodo de libre circulación (free wheeling diode) para proteger a la fuente que suministra los 25 V, ya que la carga es de tipo inductiva. Las características principales por las cuales se utilizó este tipo de diodo son:

- Tiempos de recuperación del orden de los nanosegundos
- Voltaje de conducción bajo
- Manejo de 50-600 V y 4 A

La ventaja de usar este tipo de diodo es la velocidad de conmutación entre el encendido y apagado y viceversa. Además de los valores de voltaje y corriente que maneja.

A continuación se muestra la configuración utilizada para accionar la electroválvula:

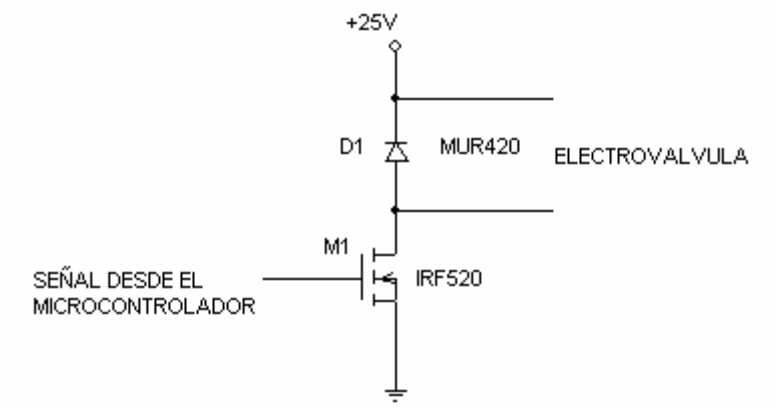



*Figura 4.3 Circuito de potencia utilizado.*

En la compuerta del MOSFET llega una señal de reloj proveniente del microcontrolador, la cual se encarga de encender el MOSFET y permitir la conducción de corriente a través de los extremos del diodo conectados a la electroválvula.

A continuación se muestra el circuito de potencia utilizado en la interfaz de hardware:

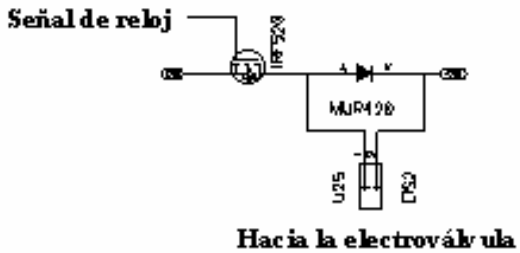



*Figura 4.4 Circuito de potencia.*

# 5. INTERFAZ DE USUARIO

Se utilizará el término interfaz de usuario para describir la interfaz de software ya que ésta será con quien esté en contacto directo el usuario y no así con la interfaz de hardware.

## 5.1 ¿Qué es una Interfaz de Usuario?

Hoy en día, gracias a los avances tecnológicos, se ha hecho necesaria la interacción hombre-máquina en muchos campos de la ciencia. Se puede ver por ejemplo que ahora no sólo los ingenieros de sistemas utilizan como herramienta la computadora, sino que también muchas otras personas lo hacen, como por ejemplo médicos y en general cualquier persona.
En la actualidad el uso de la computadora se ha convertido en algo indispensable para el ser humano.
Por este motivo ha sido necesario el crear un medio de interacción entre el hombre y la máquina llamada interfaz o mejor dicho interfaz de usuario.

### 5.1.1 Conceptos de interfaz

En términos generales las interfaces básicas se consideran como un conjunto de canales por medio de los cuales se permite la comunicación entre el hombre y la computadora, así por ejemplo se tienen menús, ventanas, teclado, ratón, algunos sonidos de la computadora, etc.

La idea principal en el concepto de interfaz es el de mediación, entre hombre y máquina. La interfaz es lo que “media”, lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, en este caso el ser humano y una máquina, como la computadora. Lo que implica además, una traducción, ya que los dos “hablan” lenguajes diferentes: verbo-icónico en el caso del hombre y binario en el caso de la computadora.

De una manera más técnica la interfaz de usuario se define como conjunto de componentes empleados por los usuarios para comunicarse con las computadoras. El usuario dirige el funcionamiento de la máquina mediante instrucciones, denominadas genéricamente entradas. Las entradas se introducen mediante diversos dispositivos, por ejemplo un teclado, y se convierten en señales eléctricas que pueden ser procesadas por la computadora. Estas señales se transmiten a través de circuitos, y son coordinadas y controladas por la unidad de proceso central y el sistema operativo. Una vez que la unidad de proceso central ha ejecutado las instrucciones indicadas por el usuario, puede comunicar los resultados mediante el mismo tipo de señales eléctricas, o salidas, hacia uno o más dispositivos de salida, por ejemplo una impresora o un monitor.

### 5.1.2 Clasificación

Dentro de las Interfaces de Usuario se distinguen básicamente dos tipos:

**Interfaz de hardware**, son todos aquellos dispositivos utilizados para ingresar, procesar y entregar los datos: teclado, ratón y pantalla, que es donde se visualizarán los resultados esperados, etc.
**Interfaz de software**, es aquella destinada a entregar información acerca de los procesos y herramientas de control, en otras palabras es lo que el usuario puede ver a través de la pantalla.

### 5.1.3 Características humanas del diseño de la interfaz

Cuando se diseña una interfaz de usuario se deben tomar en cuenta las habilidades cognitivas y de percepción de las personas, ya que el programa se debe adaptar a ellas. En muchos casos, algo muy importante es lograr la independencia de la persona de su propia memoria, por esto, la interfaz debe ser capaz de no forzar a la persona a recordar cosas innecesariamente (por ejemplo, información que apareció anteriormente) o a repetir operaciones ya realizadas (por ejemplo, introducir un mismo dato varias veces).

En términos generales se pueden enumerar los siguientes factores:

Velocidad de aprendizaje del usuario.- qué tan pronto el usuario aprende a manejar la interfaz.
Velocidad de respuesta de la interfaz.- qué tan rápido realiza las operaciones para las que fue diseñada.
Tasa de errores que comete el usuario.- porcentaje de errores que comete el usuario al manipular la interfaz.
Retención de información por parte del usuario.- cuánto recuerda el usuario sobre el uso de la interfaz en un periodo de tiempo dado.
Satisfacción del usuario.- qué tan a gusto está el usuario con la interfaz.

Otros a considerar con base en la adecuación son:

Características físicas de las personas.- puesto que no todas las personas son iguales, muchas optan por diferentes teclados, ratones, etc.
Ambiente.- se refiere al lugar donde va a ser utilizada la interfaz. En este caso la interfaz se debe adecuar al lugar.
Visibilidad.- también se refiere al lugar donde estará ubicada la interfaz, y puede ser por ejemplo que tanta luz se refleja en la pantalla.
Personalidad.- esto se refiere al nivel socio-económico, edad, etc., de la persona.
Cultura.- no todos los países manejan características iguales en sus equipos. Además se debe considerar si la interfaz será para uso internacional.

Y según la función se tienen:

Sistemas vitales.- en este caso la importancia de la medición es muy grande puesto que muchas personas dependen de ellos y pueden ser de vida o muerte.
Sistemas comerciales e industriales.- pueden ayudar a aumentar la productividad y así vender más.
Sistemas de oficina, hogar y juegos.- en este caso se considera el mercado a quien están dirigidos; deben ser amigables y satisfactorios.
Sistemas de investigación.- su función es realizar tareas específicas y tratan de imitar el medio en el que se desenvuelve el usuario.

## 5.2 Visual Basic

### 5.2.1 ¿Qué es Visual Basic?

En términos generales es un lenguaje de programación que se ha diseñado para facilitar el desarrollo de aplicaciones en un entorno gráfico, como Windows 98, Windows NT o superior.

Los programas de Visual Basic son creados en un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés). El IDE permite al programador crear, correr y depurar convenientemente los programas en Visual Basic. El IDE permite al programador crear programas funcionando, en una fracción del tiempo que esto podría tomar normalmente para programas de código sin usar IDE's.
El proceso de creación rápida de una aplicación es referido típicamente como Desarrollo Rápido de la Aplicación (RAD – *Rapid Application Development*). Visual Basic es el lenguaje RAD más ampliamente usado del mundo.
Visual Basic es un lenguaje distintivamente diferente, previendo características poderosas tales como interfaces de usuario gráficas, manejo de eventos, acceso a las API Win32, características orientadas a objetos, manejo de errores, programación estructurada y mucho más.

Microsoft Visual Basic es un sistema de desarrollo diseñado especialmente para crear aplicaciones con interfaz gráfica, de una forma rápida y sencilla. Para soportar este tipo de desarrollos, Visual Basic utiliza fundamentalmente dos herramientas, una que permite realizar los diseños gráficos y un lenguaje de alto nivel.

### 5.2.2 Características

Algunas de las características más importantes que incluye Visual Basic son las siguientes:

- Una biblioteca de clases que da soporte a los objetos Windows tales como ventanas, cajas de diálogo, controles (por ejemplo, etiquetas, cajas de texto, botones de pulsación, etc.).

- Un control que permite utilizar las cajas de diálogo más comúnmente utilizadas (abrir, guardar como, imprimir, color y fuentes).

- Un entorno de desarrollo integrado (editor de texto, intérprete, depurador, examinador de objetos, explorador de proyectos, compilador, etc.). Visual Basic fue diseñado para ser un intérprete, lo que favorece la creación y la depuración de una aplicación, y a partir de la versión 5 incluyó también un compilador que permite generar ficheros *.exe* favoreciendo así la ejecución. En adición, puede crear también ficheros *.ocx* y *.dll*, lo que permitirá manipular controles *ActiveX.*

- El editor de textos ayuda ahora a completar cada una de las sentencias visualizando la sintaxis correspondiente a las mismas.

- Asistentes para el desarrollo de aplicaciones: asistente para aplicaciones, asistente para barras de herramientas, asistente para formularios de datos, asistente para empaquetado y distribución, asistente para crear la interfaz pública de controles *ActiveX*, asistente para páginas de propiedades, asistente para objetos de datos, generador de clases, diseñador de complementos y asistente para migración de documentos *ActiveX*.

- Una interfaz para múltiples documentos (MDI – *Multiple Document Interface*) que permite crear una aplicación con una ventana principal y múltiples ventanas de documento. Un ejemplo de este tipo de aplicaciones es Microsoft Word.

- Editar y continuar. Durante una sesión de depuración, se pueden realizar modificaciones en el código de la aplicación sin tener que salir de dicha sesión.

- Un programa para añadir ayuda en línea; esta herramienta permite la creación de ficheros de ayuda estilo Windows (*hcw.exe – Help Workshop*)

- Cuando se combinan estas características, algunas de ellas sólo disponibles en la versión profesional y empresarial, se dispone de un sistema de desarrollo que permite diseñar rápidamente aplicaciones sofisticadas.

Además de estas, Visual Basic en su versión 6 incorpora las siguientes características:

- Las funciones pueden devolver Arrays y los Arrays dinámicos se pueden asignar a funciones para el manejo de cadenas y arrays.
- Los tipos definidos por el usuario se pueden usar como parámetro y como valor devuelto por una función

### 5.2.3 Ventajas y desventajas

Ventajas:

- ✓ Facilidad en la programación ya que se realiza en un ambiente totalmente gráfico y no se trata de sólo código.
- ✓ Muy completo, flexible y fácil de comprender.
- ✓ Creación de programas desde los más sencillos a los más complejos.
- ✓ Rapidez en la creación de programas lo que diminuye el coste de la aplicación.
- ✓ La programación no es estructurada que empieza y acaba, sino que permite la interacción con el usuario final.
- ✓ El entorno de programación ayuda a encontrar fallos, simplifica la búsqueda de texto en el código, permite incluir comentarios, etc.
- ✓ Contiene herramientas de depuración que no existen en otros lenguajes.
- ✓ Comparado con otros lenguajes como Java, Visual Basic es sencillo pero a la vez relativamente potente ya que se pueden construir aplicaciones en muy poco tiempo.
- ✓ La ayuda de Visual Studio, tanto en disco como la que se encuentra en línea facilitan más la realización de los programas.
- ✓ Incorpora gran cantidad de componentes ya desarrollados lo que implica la no implementación por parte del programador.
- ✓ Una de sus limitaciones es la imposibilidad de realizar programas de alto y medio desarrollo, por ejemplo hablando en el tema de seguridad e internet.
- ✓ Es muy intuitivo y requiere poco aprendizaje.

Desventajas:

- × No es compatible con muchos lenguajes y programas que son creados por otras empresas (diferentes a Microsoft).
- × Sólo corre bajo Windows.
- × Al ejecutarse por eventos los errores en la aplicación no son vistos hasta llegar al punto de fallo.
- × Es un lenguaje orientado a eventos no a objetos.

## 5.3 Diseño de la interfaz

En el proceso de diseño de una interfaz de usuario se tienen cuatro pasos fundamentales:

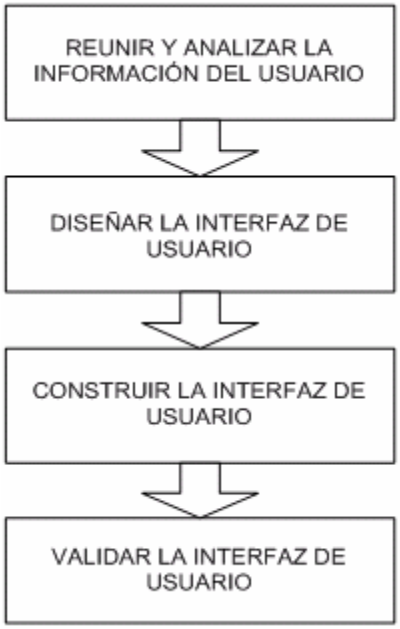



*Figura 5.1 Pasos en el proceso de diseño de la interfaz de usuario.*

*Reunir y analizar la información del usuario.-* en esta etapa se define qué tipo de usuario utilizará el programa, que tareas realizará y cómo lo hará. Qué exige el usuario del programa y en qué tipo de entorno se desenvuelve el usuario.

*Diseñar la interfaz de usuario.-* antes de entrar a la codificación es importante definir el uso final de la interfaz; conocer las tareas del usuario, los objetos y acciones de la interfaz, los íconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas.

*Construir la interfaz de usuario.-* aquí se sugiere realizar un prototipo de la versión para probarlo antes de realizar la codificación final.

*Validar la interfaz de usuario.-* en esta última etapa se deben realizar pruebas sobre el uso final de la interfaz, de ser posible con usuarios finales de la misma.

### 5.3.1 Análisis y recopilación de información del usuario final de la interfaz así como del uso final de la interfaz misma

El tipo de usuario que utilizará la interfaz es personal capacitado del Instituto Nacional de Cardiología. Por esta razón no habrá problema con respecto al diseño planteado. Las tareas a realizar por parte del usuario son el monitoreo de señales provenientes de los sensores de presión y el control del probador de prótesis cardiacas. Todas estas tareas se realizan desde una PC y para lo cual el usuario sólo tiene que introducir ciertos datos para poder comenzar con la prueba.
Puesto que en ciertas ocasiones el programa no será operado por ingenieros, el diseño comprendió todo tipo de usuarios con conocimientos previos de

computación. Por tal motivo se consideró el desarrollo de una interfaz capaz de ir guiando al usuario a través de todo el proceso de control y adquisición.

Como se había mencionado antes, el uso final que se le dará a la interfaz es el de monitorear y desplegar en pantalla las señales provenientes de sensores de presión colocados en la cámara de evaluación de las válvulas cardiacas, además de poder registrar los datos obtenidos, si el usuario así lo desea. También se controlará el actuador encargado de simular las condiciones a las que son sometidas las válvulas dentro del cuerpo humano.

### 5.3.2 Diseño de la interfaz de usuario

Algunos aspectos tomados en cuenta en el diseño y desarrollo de la interfaz de usuario fueron los siguientes:

*Presentación de información.-* en este caso no se deben colocar demasiados objetos en la pantalla y los existentes deben estar bien distribuidos. Cada elemento visual influye en el usuario, tanto por si mismo como por su combinación con el resto de los elementos presentes.
*Análisis de color.-* el color comunica información, por lo tanto deben utilizarse combinaciones adecuadas. El color debe atraer la atención del usuario, pero no debe cansarlo después de un rato de trabajo.
*Análisis de audio.-* en primer lugar es preciso saber cuándo es más apropiado que la información visual. En segundo lugar, se debe determinar el sonido adecuado y finalmente permitir la personalización del mismo.
*Análisis de animación.-* es el cambio en el tiempo de la apariencia de un elemento gráfico. Se aplica por ejemplo en: progreso de acciones, estado de procesos, acciones posibles.
*Diseño internacional.-* si el programa realizado será de uso internacional se deben tomar en cuenta las terminologías y hacer un uso adecuado de ellas. Así también con las diferencias culturales.
*Análisis y elección de controles.-* se debe saber qué tipo de controles se utilizarán ya que muchas veces un mismo control tiene varias características de presentación pero no todas gustan al usuario.
*Guías de expertos.-* actualmente existen guías de diseño realizadas por expertos y algunas de ellas son:

- Demasiada simetría puede hacer las pantallas difíciles de leer.
- Si se colocan objetos sin alinear lo mejor es hacerlo de forma drástica.
- Asimetría = activo, simetría = sereno.
- Cuando existen elementos de tamaño y color similares se pueden llegar a percibir como si fueran de un mismo grupo.
- Se debe diseñar para el usuario, no para demostrar los conocimientos tecnológicos ni propios.
- Gráficos espectaculares no salvarán a una mala interfaz.

La interfaz consta de sólo los objetos necesarios, es decir, no se utilizaron más botones de los necesarios, ni ventanas extras. Los menús son sólo los indispensables para llevar a cabo las tareas encomendadas.
En términos generales se planteó que la interfaz sólo tuviera un botón para el control general, es decir, que al presionarlo se inicie la prueba y al volverlo a oprimir se detenga todo el proceso. Pero antes de todo esto se deben de realizar ciertos pasos para configurar la adquisición de los datos y el control del actuador.

En el caso de la adquisición de datos es importante saber cuántas muestras serán graficadas y a qué velocidad serán muestreadas, aunque esto pudo haber sido incluido en la programación del microcontrolador se prefirió dejarlo a elección del usuario ya que las señales a muestrear no tendrán la misma frecuencia.
También se dio la opción de registrar los datos que son graficados, en este caso el usuario puede elegir el lugar donde se almacenarán los archivos de texto generados y cada cuánto tiempo se harán esos registros.
Otra opción que tiene el usuario es el exportar estos archivos a una hoja de cálculo, lo que ayuda bastante para poder manipular los datos con fines estadísticos.
Para el caso del control del actuador fue necesario enviar la frecuencia a la cual operará el mismo además de otros datos propios del control.

Ya después de haber ingresado todos estos datos lo único que debe hacer el usuario es presionar el botón general encargado de poner en marcha el sistema, cabe señalar que durante la prueba se pueden hacer cambios en algunos datos previamente ingresados, por lo que no es necesario detener la prueba para hacer cambios en los mismos.

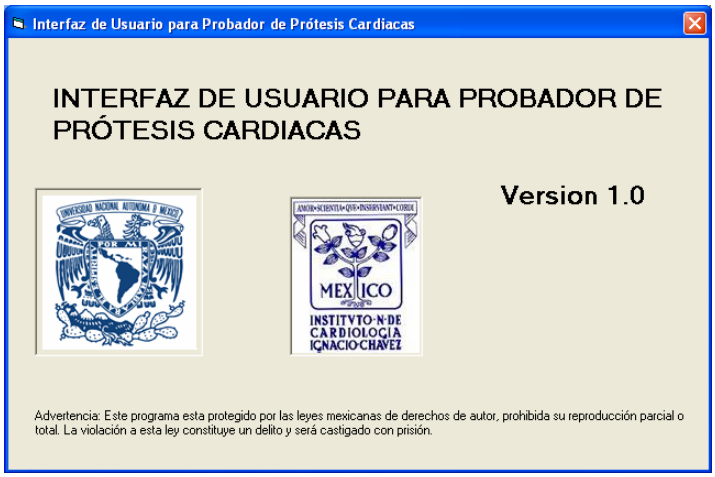
Para facilitar la interacción del usuario con la interfaz se envían ciertos mensajes en pantalla dependiendo si se han ingresado los datos o no y se ha hecho correctamente. Esto es la forma en la que se va guiando al usuario a través de todo el proceso.

## 5.4 Implementación de la interfaz de usuario

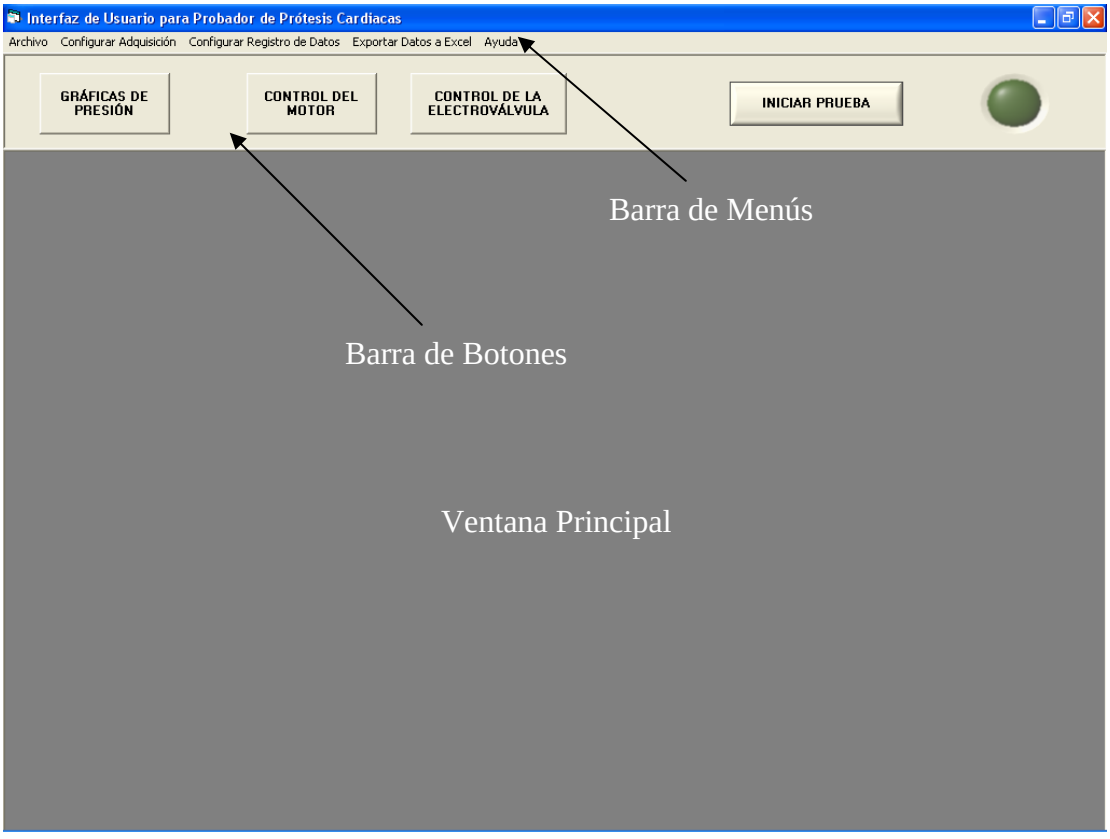
En el siguiente punto se describe el programa y lo que se hizo para lograr cada una de las funciones descritas anteriormente. Para ello se muestra la ventana y se describe cada componente así como la función principal de cada uno de los botones mostrados.

### 5.4.1 Programación de la interfaz gráfica para el despliegue y control de la información, utilizando Visual Basic 6.0

El programa inicia con una ventana de bienvenida donde se muestran el nombre del programa así como la versión, los nombres de los participantes y algunas advertencias sobre piratería.




Esta ventana será mostrada durante más o menos 3 segundos, posteriormente desaparece y da paso a la ventana principal del programa.




A continuación se describe la función de cada uno de los menús y los botones mostrados.

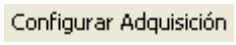
Menús.-



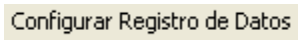

Al hacer clic en el menú Archivo se muestra cómo única opción la de salir.



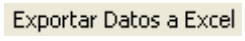

Esta opción permite cerrar el programa.




Este menú lleva a una ventana que ayuda a configurar el puerto serie para la adquisición de datos.




La ventana que se muestra después de hacer clic en este menú sirve para configurar opcionalmente el registro de datos.




Si se desea exportar los datos registrados, este menú es el encargado de presentar la ventana de configuración de la tarea.




Al hacer clic en este menú se muestra el submenú Acerca de…




Esta opción nos muestra la ventana con información acerca del programa.

Botones.-




Este botón permite cargar la ventana donde se mostrarán las gráficas de las señales provenientes de los sensores de presión.




Al presionar este botón se muestra en pantalla la ventana donde se introducen los datos para controlar el motor.




Al presionar este botón se muestra en pantalla la ventana donde se introducen los datos para controlar la electroválvula.



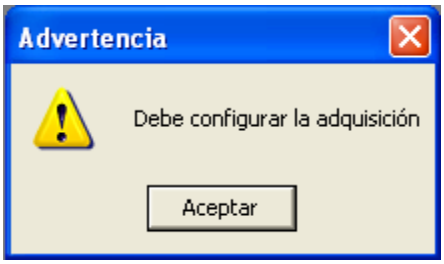

Este botón es el encargado de poner en funcionamiento el programa, así como de detenerlo en cualquier momento.



Este es sólo un indicador que nos muestra cuando el programa está en funcionamiento y cuando está inactivo. Cuando se encuentra en color oscuro la prueba no ha comenzado, pero cuando se torna color amarillo significa que la prueba se está realizando.

Antes de iniciar una prueba es necesario el configurar el puerto serie así como el control del motor o la electroválvula. El orden no importa pero si es importante hacerlo ya que de lo contrario se presentarán las siguientes ventanas:

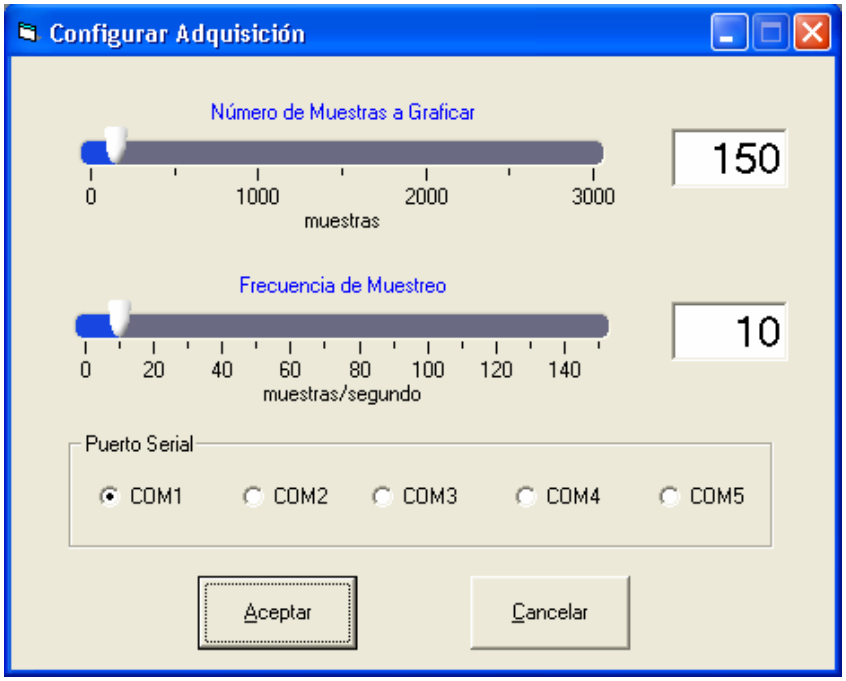
En caso de no configurar la adquisición:




En caso de no configurar el control del motor o la electroválvula:

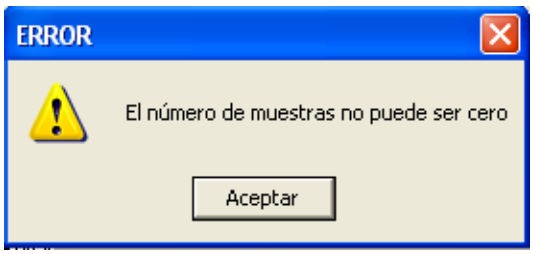


Cuando se oprime el botón Configurar adquisición la ventana que aparece en pantalla es la siguiente:




Como se puede observar se tienen datos previamente establecidos tanto para el número de muestras a graficar y la frecuencia de muestreo, como para la selección del puerto serie a utilizar. Todos estos pueden ser modificados por el usuario.

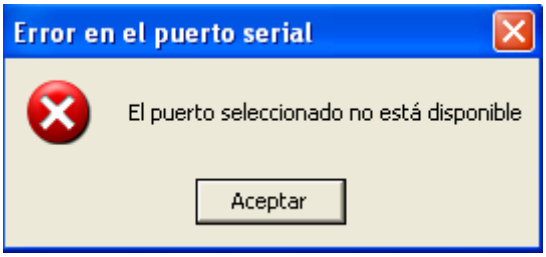
En dado caso que el número de muestras a graficar sea igual a cero, el programa manda el siguiente mensaje:




Para el caso de que la frecuencia sea igual a cero, el programa manda un mensaje similar.



Con el puerto serie sucede algo parecido ya que si se elige un puerto ocupado el programa manda el siguiente aviso:




En ocasiones, físicamente el puerto está ocupado pero el dispositivo que lo utiliza está inactivo lo que provoca que el puerto elegido sea aceptado como válido. Esto se verifica hasta que se inicia la prueba y en dado caso que haya un dispositivo ocupando el puerto se manda un mensaje de error parecido al anterior pero que dice: "No se ha seleccionado un puerto disponible".

En caso de oprimir el botón de Control del Motor, la ventana que aparece en pantalla es la siguiente:




Aquí se elige el sentido de giro, así como el número de pasos a oscilar y la frecuencia a la cual trabajará el motor. La parte de calibración sirve para generar un paso a la vez.

Si no se elige el giro del motor aparece la siguiente ventana:




En el caso de los pasos a oscilar y la frecuencia no hay problema puesto que ya se tienen cargados valores previamente.

En caso de oprimir el botón de Control de la Electroválvula, la ventana que aparece en pantalla es la siguiente:

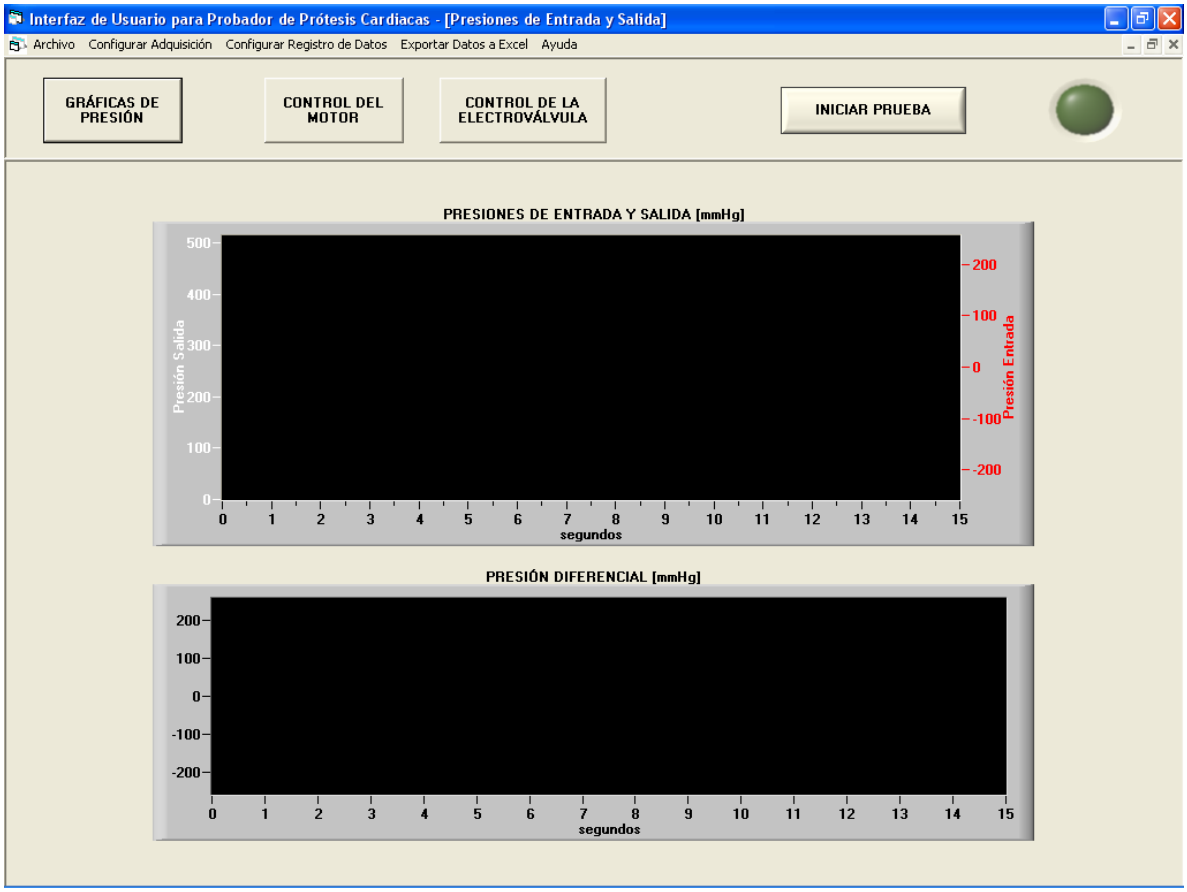



En caso de que sólo se oprima el botón Aceptar, el programa cargará el dato fijado en la perilla.

Para el caso tanto del control del motor como de la electroválvula, el botón de modificar que se encuentra inhabilitado se habilitará durante la prueba ya que con ello se podrán modificar los valores de frecuencia en tiempo real.

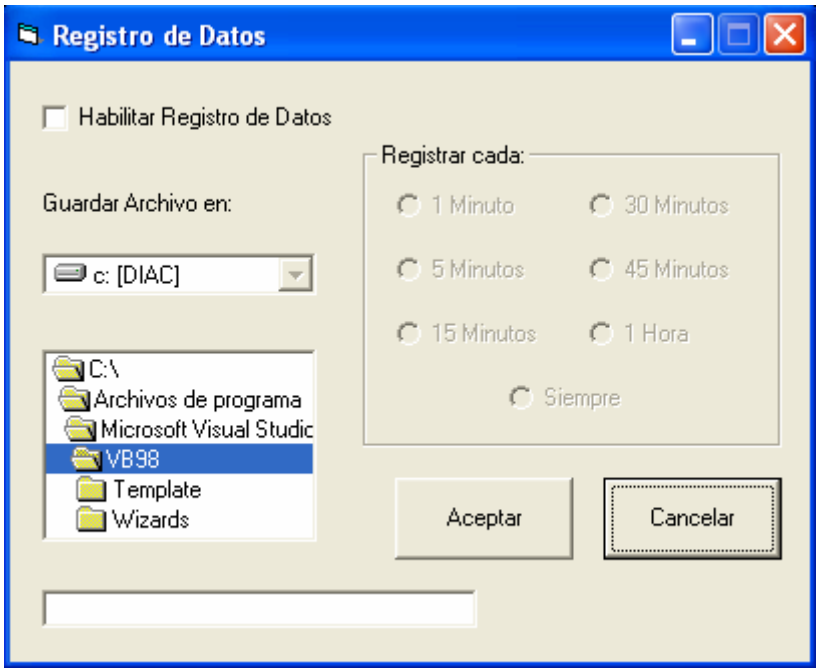
Como se observó, en las ventanas de configuración sólo se introducen datos, mismos que son necesarios para que el programa funcione correctamente. Los mensajes de error fueron pensados para de alguna forma ir guiando al usuario en el ingreso de los mismos.

Al oprimir el botón de Gráficas de Presión se muestra la ventana donde se presentarán las graficas de los sensores:



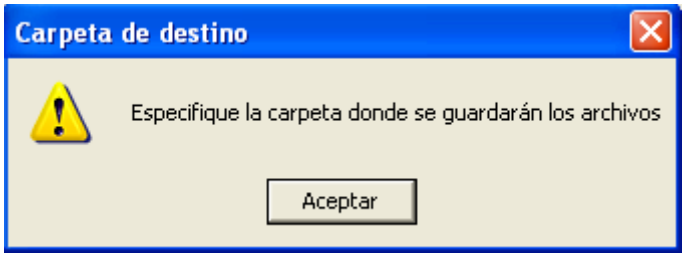

Aquí se mostrarán las gráficas de presión de entrada y salida a la válvula a evaluar, así como la presión diferencial generada a partir de las presiones anteriores.

Opcionalmente se puede configurar el registro de datos para que estos puedan ser almacenados en archivos durante la prueba, para esto se oprime el menú Configurar Registro de Datos y aparece la siguiente ventana:




Aquí se habilita o no el registro de los datos, posteriormente se debe elegir la carpeta donde serán almacenados los archivos de texto generados, así también se debe seleccionar cada cuánto tiempo se hará este registro. Si por alguna razón no se especifican los datos anteriores el programa mandará avisos diciendo que faltan datos para lograr el correcto registro. Cabe mencionar que la carpeta de destino puede ser seleccionada de una existente o bien, si se teclea directamente en el cuadro de texto en blanco el programa puede crearla o no a elección del usuario, en el lugar especificado.

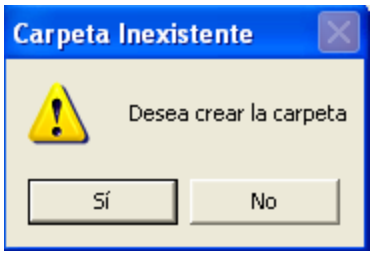
El mensaje de aviso para el caso de que no se especifique la ruta donde se guardará el archivo es:




En caso de que no se seleccione algún tiempo de registro el mensaje que aparece es el siguiente:

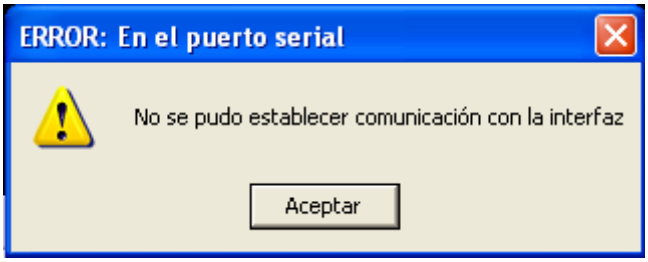



El mensaje que aparece cuando la carpeta especificada no existe es el siguiente:

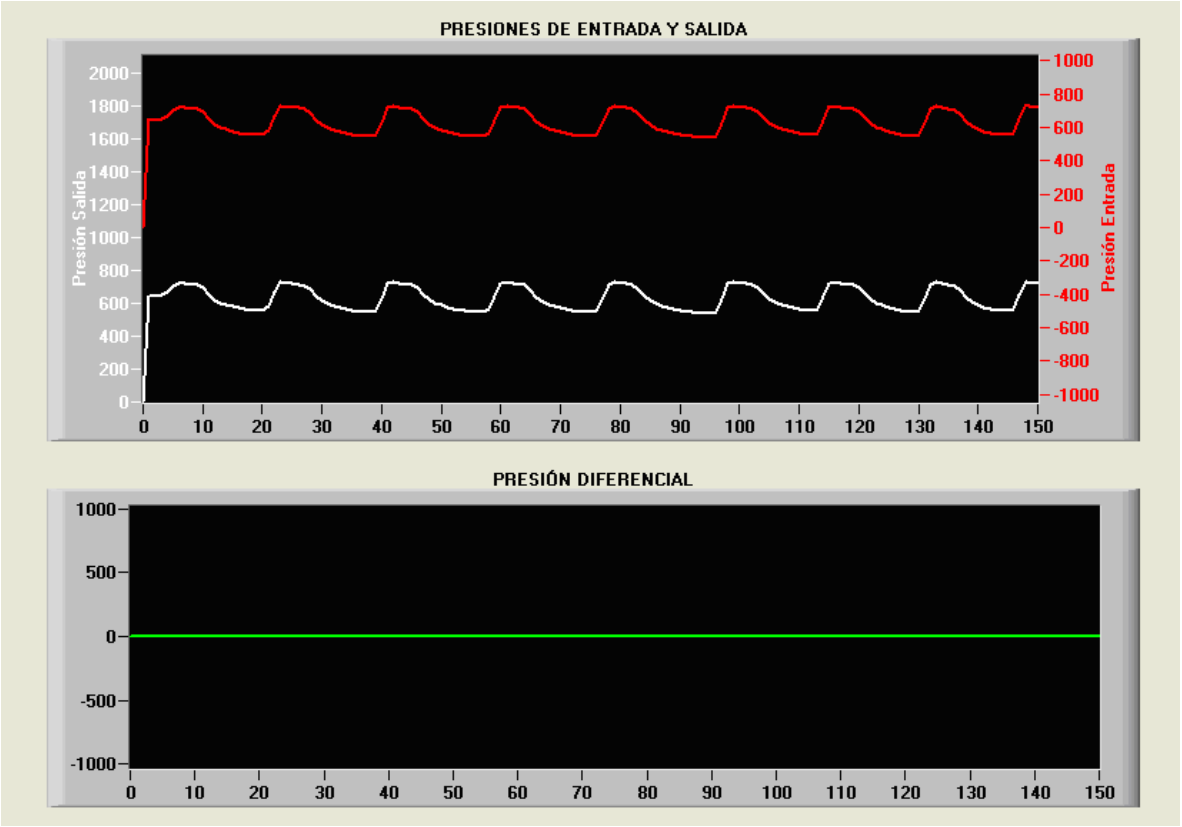



Una vez que se han realizado las configuraciones necesarias ya se puede iniciar la prueba para poder obtener los datos provenientes de los sensores así como el enviar las señales de control hacia el actuador.

Ahora ya sólo es necesario presionar el botón Iniciar Prueba; si por alguna razón no se puede establecer comunicación con la interfaz de hardware aparece el siguiente mensaje en pantalla:




Por el contrario, si todo es correcto, la interfaz comenzará a adquirir las señales provenientes de los sensores tal y como lo muestra la siguiente figura:




Para lograr la recepción y transmisión de datos desde la interfaz de usuario se hizo uso de un control ActiveX, encargado de controlar el puerto serie. En el caso de la transmisión de datos desde la computadora y como ya se había mencionado en la programación del PIC, fue necesario el dividir el dato para que pudiera ser recibido en el microcontrolador, para llevar a cabo esto se realizaron las siguientes operaciones:

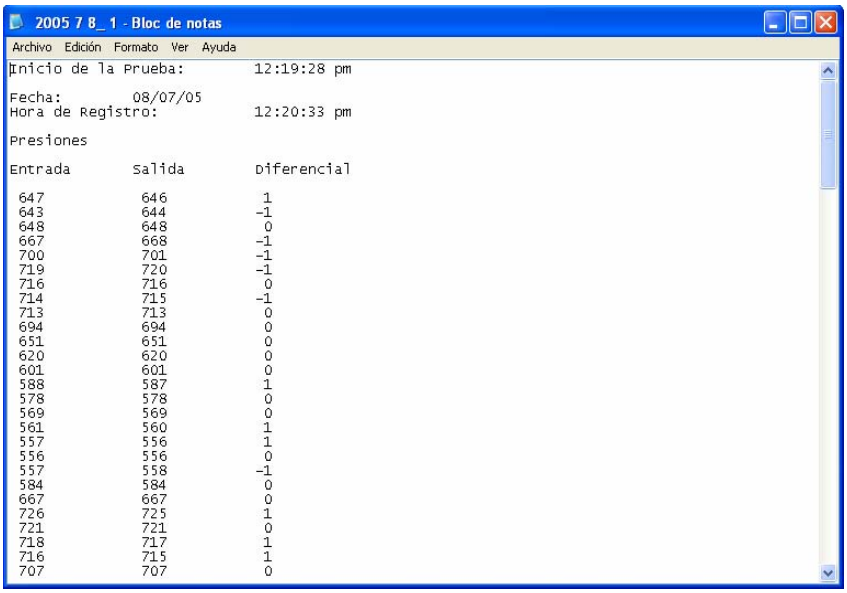
```
datoalto = Int(muestras / 256)
bytehigh = CByte(datoalto)
Serial.Output = Chr(bytehigh)
bytelow = muestras Mod 256
Serial.Output = Chr(bytelow)
```

A diferencia de la programación en C, la programación en Visual Basic brinda funciones que ayudan a simplificar el trabajo. Tal es el caso de la función Int, la cual nos regresa la parte entera del cociente de la división de dos números. Y para obtener la parte baja del dato se utilizó la función Mod, que al aplicarla da como resultado el residuo de la división de dos números.

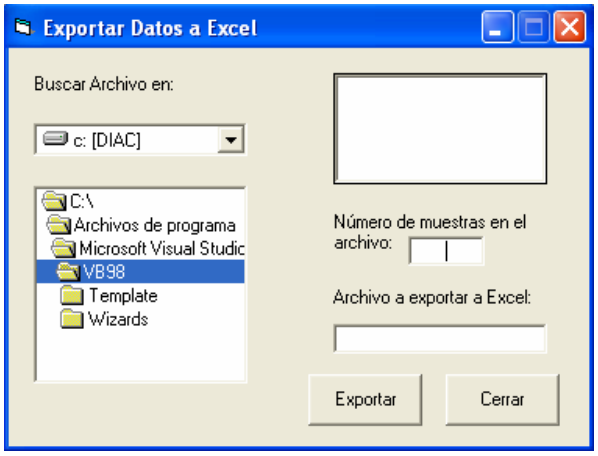
Para el caso de la recepción de datos en la computadora, el mismo control ActiveX tiene funciones que ayudan a tomar los datos que llegan al puerto. Una vez recibidos los datos, en Visual Basic se procede a transformarlos a valores enteros

decimales, puesto que los datos recibidos son caracteres ASCII. Posteriormente estos datos van siendo graficados uno a uno.

Si el registro de datos fue habilitado, cada cierto tiempo se crearán archivos de texto con los datos graficados. En este caso estos archivos pueden ser abiertos por cualquier procesador de texto o simplemente con el bloc de notas. La información que aparecerá en los archivos será similar la que se muestra a continuación:

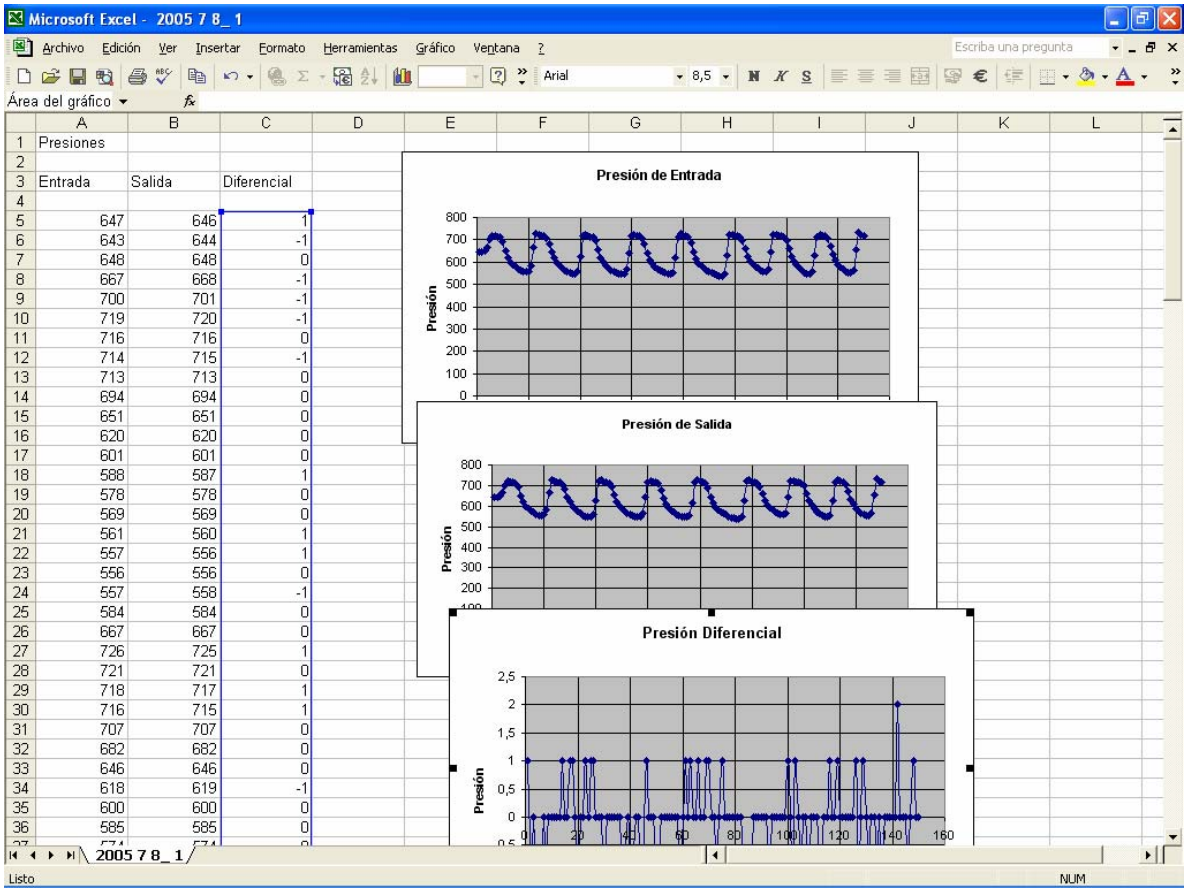



Si no se desea ver el archivo de texto, la interfaz cuenta con la función de exportar los datos contenidos en el archivo a una hoja de cálculo de Excel, para esto lo primero que se debe hacer es dar clic en el menú Exportar Datos a Excel con lo que aparece la siguiente ventana:




En esta ventana se especifica el archivo a exportar así como el número de muestras contenidas en él. Una vez hecho esto se da clic en exportar con lo que

finalmente se muestran en una hoja de cálculo de Excel tanto los datos como las gráficas de éstos.



# 6. PRUEBAS Y RESULTADOS

Desde el comienzo del proyecto se tuvieron que hacer pruebas continuamente; en un principio lo importante fue establecer la comunicación entre el software y el hardware y una vez logrado esto fue que se comenzaron a desarrollar en forma tanto la interfaz de hardware como la de usuario.

Las pruebas iniciales consistieron en poder enviar información desde la computadora hacia el microcontrolador y de éste hacia la computadora. Para realizar esto se hizo uso de la “hyperterminal” el cual es un programa de Windows con el que se puede establecer una comunicación bidireccional a través del puerto serie de la computadora.

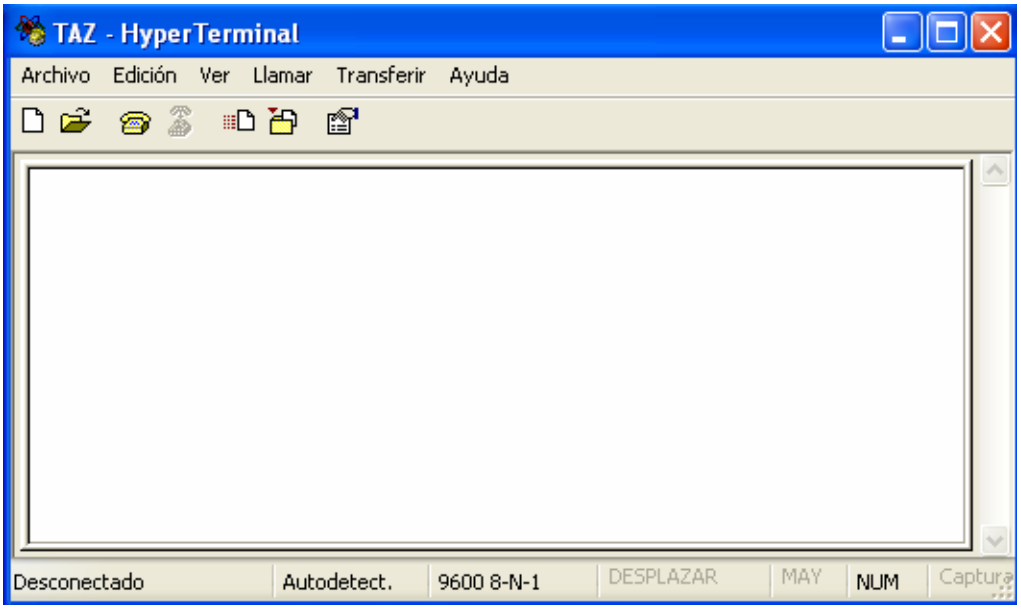



*Figura 6.1 Ventana principal de la hyperterminal.*

Una vez desarrolladas ambas interfaces se realizaron pruebas al conjunto de las mismas y a la interfaz de usuario de manera individual.

Como conjunto se verificó que las señales adquiridas por la interfaz de usuario fueran realmente las señales adquiridas por la interfaz de hardware, en este caso se utilizó un generador de señales para establecer una señal patrón. Al mismo tiempo se hizo variar en frecuencia la señal obtenida del generador para constatar que se estuviera adquiriendo en tiempo real.

Una vez comprobado lo anterior se realizaron pruebas pero ahora con los sensores de presión y con la señal generada por el mismo microcontrolador, ésta última encargada de controlar la frecuencia del actuador.

A continuación se muestran las gráficas obtenidas de un sensor y la generada por el microcontrolador.

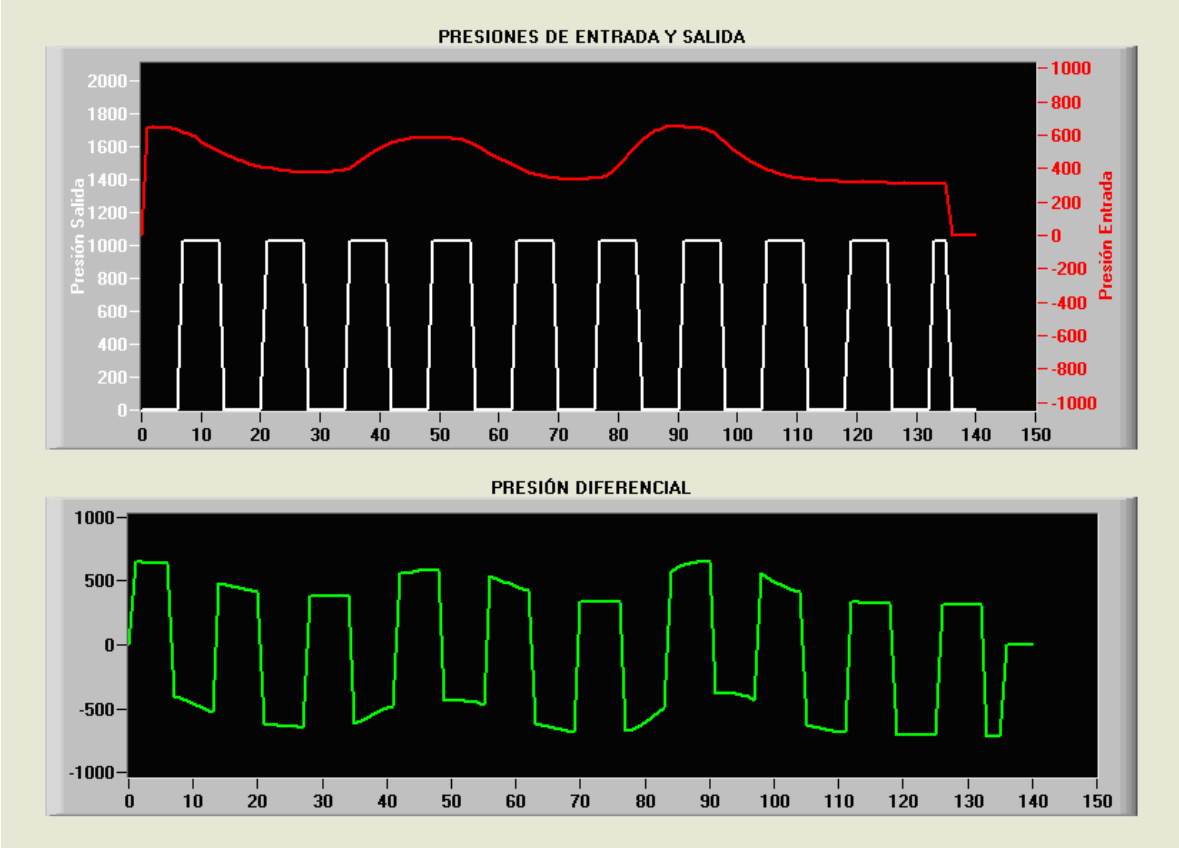



*Figura 6.2 Señal de presión, de control y la diferencial de éstas.*

Las pruebas realizadas a la interfaz de usuario consistieron en verificar el correcto funcionamiento de la misma en cuanto a la configuración del registro de datos y a la exportación de los mismos a una hoja de cálculo de Excel. Para el primer caso se observó que los tiempos de registro coincidieran con los establecidos en el programa además de que el número de datos registrados fuera el asignado previamente.
Una vez hecho el registro de datos se procedió a exportarlos a Excel donde sólo se mostraron las columnas con las presiones de entrada, salida y la diferencial, así como las gráficas de cada una de éstas últimas.
Después de realizadas las pruebas anteriores se procedió a hacer otras con la electroválvula las cuales resultaron satisfactorias, éstas consistieron en ponerla a funcionar y posteriormente modificar la frecuencia de operación de la misma en tiempo real. En este último caso también fueron conectados los sensores de presión.

Una de las pruebas más importantes que se le realizó a la interfaz de usuario fue el provocar errores en ella para posteriormente crear una función capaz de manipularlos, con esto se logró una especie de guía para el usuario que lo lleva en

el proceso de configuración de la interfaz para comenzar un proceso de adquisición.

A continuación se muestran las imágenes de la cámara de evaluación de las válvulas y la electroválvula controlada por la interfaz realizada:

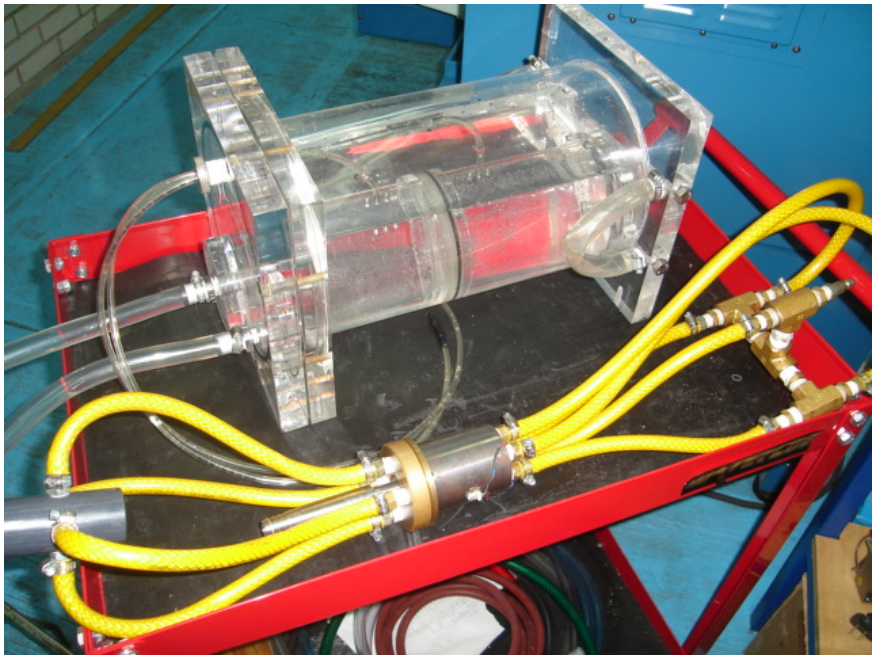


*Figura 6.3 Electroválvula y cámara de evaluación.*

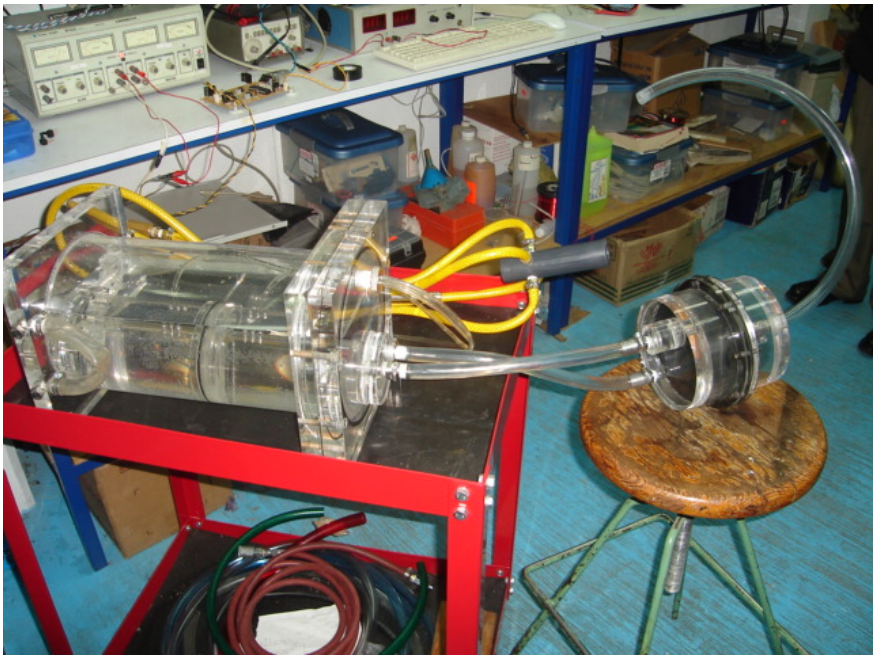


*Figura 6.4 Cámara de evaluación y sistema de bombeo.*

Las siguientes imágenes muestran el motor y el sistema biela-manivela controlado por la interfaz, así como la cámara de pruebas:

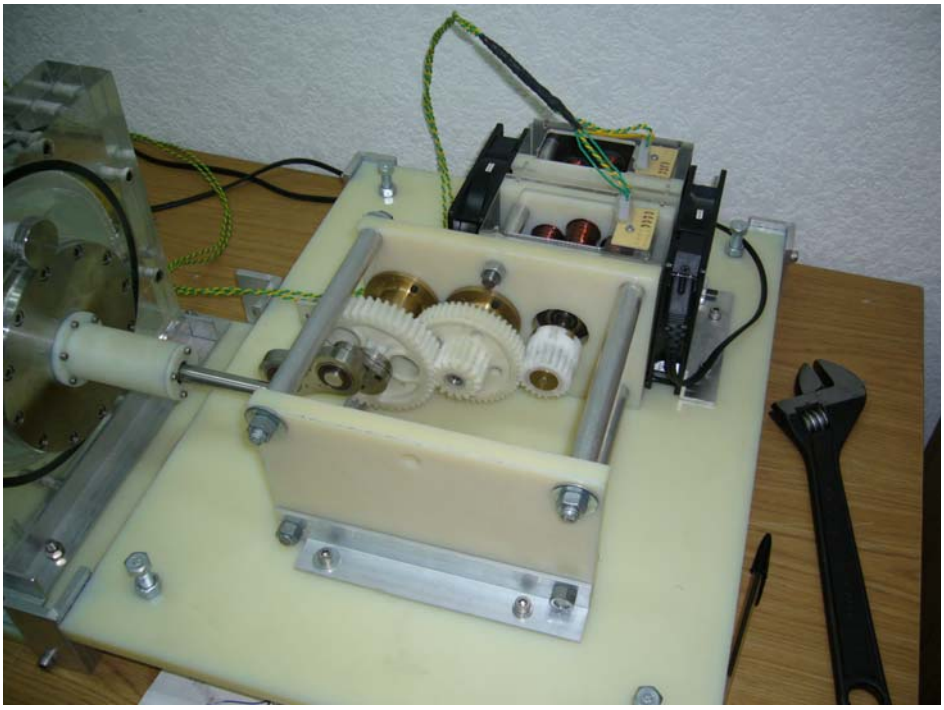


*Figura 6.5 Motor y sistema biela-manivela.*



*Figura 6.6 Cámara de pruebas.*

Las siguientes imágenes muestran las gráficas de presión en la interfaz de usuario realizada:

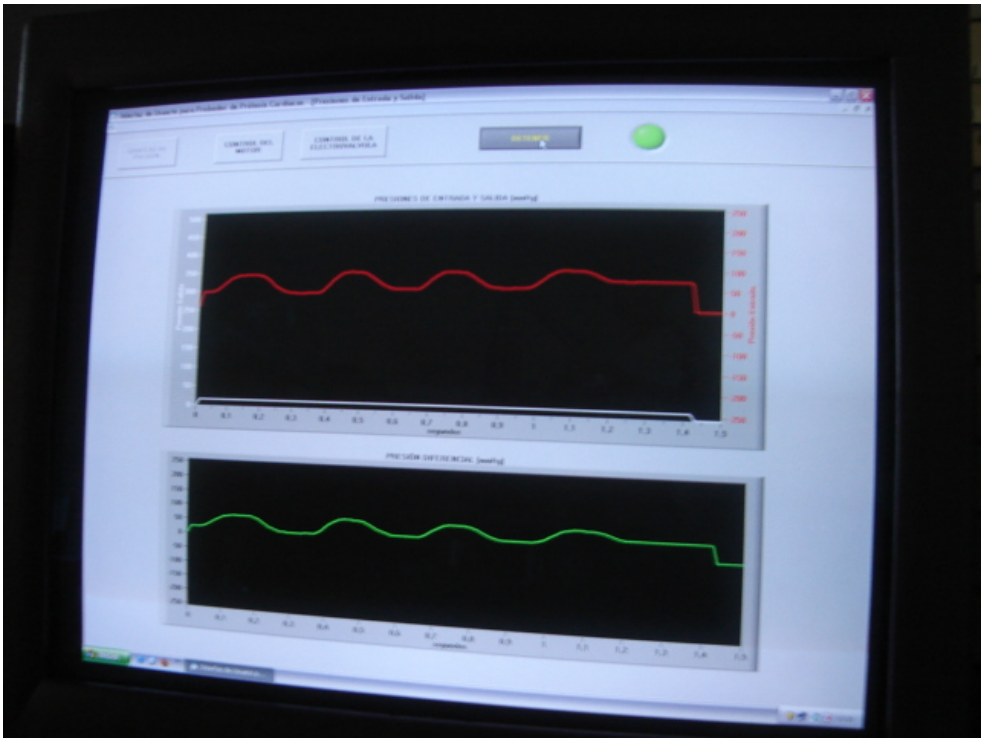


*Figura 6.7 Señales de presión.*




*Figura 6.8 Señales de presión.*

Finalmente se muestran imágenes de la etapa de potencia para el motor y la interfaz de hardware:



*Figura 6.9 Etapa de potencia para el motor.*



*Figura 6.10 Interfaz de hardware.*

# 7. CONCLUSIONES Y COMENTARIOS

De los resultados obtenidos y con base en los objetivos planteados y los alcances establecidos al inicio del presente trabajo, se puede decir que tanto los objetivos como los alcances se lograron satisfactoriamente.

Aunque el diseño actual no es el definitivo, la interfaz ya está en condiciones de ser sometida a varias pruebas más, con lo que ésta se irá mejorando hasta obtener un diseño final acorde con las necesidades del Instituto Nacional de Cardiología.

En cuanto a la parte electrónica no hubo demasiados problemas, salvo que debido a la presión a medir fue necesario el calibrar los sensores para obtener valores dentro del rango establecido por las características del microcontrolador. Cabe señalar que no fue necesario el utilizar algún tipo de filtro en el acondicionamiento de las señales de presión gracias a la configuración de amplificador de instrumentación y a que las señales a manejar no fueron demasiado pequeñas.

Las características de los sensores de presión utilizados fueron de gran ayuda ya que no fue necesario el realizar configuraciones adicionales para su correcto funcionamiento, además de que se encuentran disponibles en el país.

En general los elementos y dispositivos utilizados para la realización de hardware se conservaron desde el inicio del diseño y no hubo necesidad de cambiarlos salvo algunas resistencias que sólo fueron cambiadas en valor.

En cuanto a la interfaz de usuario, lo más problemático fue el provocar errores en la ejecución de la misma para así tratar de obtener el mejor método de solución de los mismos en tiempo de ejecución. Conforme se fue avanzando en el desarrollo de ésta se fueron encontrando detalles que se corrigieron al mismo tiempo.

En términos generales se logró desarrollar una interfaz que no tiene una electrónica muy compleja esto con el fin de que los usuarios sean capaces de detectar posibles fallas y solucionarlas de inmediato.

Respecto a la interfaz de usuario se trató de hacerla lo más amigable posible, el mandar avisos durante un proceso normal de configuración para puesta en marcha será de gran ayuda para el usuario final, ya que lo irá guiando paso a paso hasta obtener un resultado satisfactorio.

Como se mencionó anteriormente, este diseño no es el final y de acuerdo a ciertos requerimientos se irá modificando, preservando las características deseadas desde un principio.

Es importante mencionar que muchas de las características que rigen el funcionamiento de la interfaz de usuario se basan en la norma oficial mexicana

NOM-063-SSA1-1993, específicamente en el punto 8 el cual se refiere a las pruebas de los sustitutos de las válvulas cardiacas. Ya que es condición necesaria para realizar dispositivos encargados de probar las válvulas cardiacas.

En un futuro no muy lejano, este sistema sustituirá al que actualmente se encuentra en el INC con lo que se ahorrará en costos y personal para la operación del mismo.

# 8. APÉNDICES

## A.1 Análisis del Amplificador de Instrumentación

A continuación se muestra la configuración del amplificador de instrumentación:

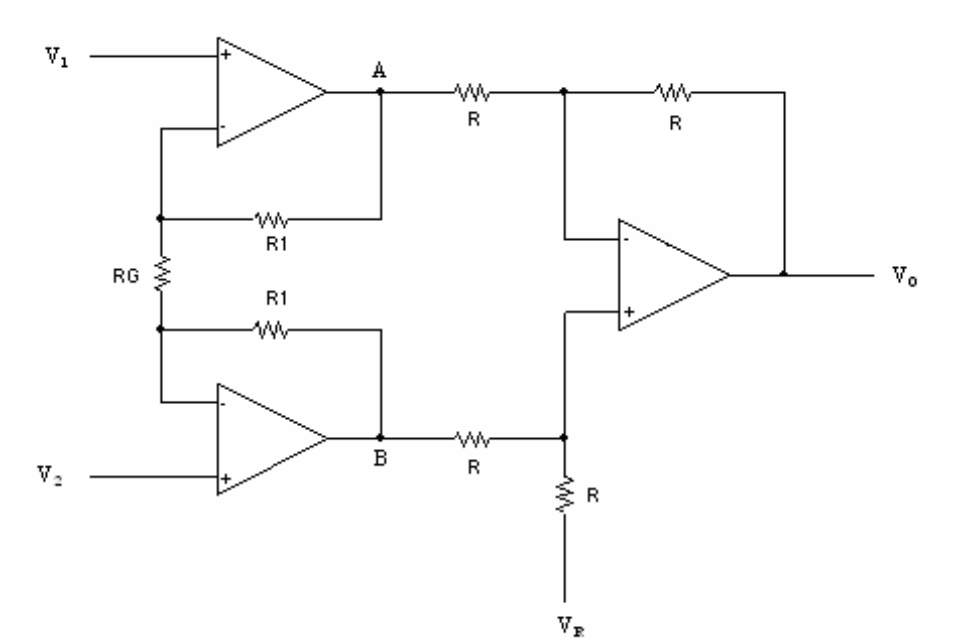



*Figura A.1.1 Amplificador de Instrumentación.*

Para encontrar el voltaje de salida $V_O$ en términos de $V_1$ y $V_2$ primeramente se calcula el voltaje en los puntos A y B.

Por superposición:

$V_1 \neq 0$ y $V_2 = 0$

en el punto A se tiene un amplificador no inversor, por lo tanto:

$$V_{A1} = \left(1 + \frac{R1}{RG}\right) V_1$$

mientras que en el punto B se tiene un amplificador inversor, así entonces se tiene que:

$$V_{B1} = -\frac{R1}{RG} V_1$$

Ahora considerando el caso inverso, es decir:

$V_1 = 0$ y $V_2 \neq 0$

para el punto A se tiene un amplificador inversor:

$$V_{A2} = -\frac{R1}{RG}V_2$$

y para el punto B se tiene un amplificador no inversor:

$$V_{B2} = \left(1 + \frac{R1}{RG}\right)V_2$$

De donde para obtener el voltaje total en A y B se tiene que:

$$V_A = V_{A1} + V_{A2} = \left(1 + \frac{R1}{RG}\right)V_1 - \frac{R1}{RG}V_2$$

$$V_B = V_{B1} + V_{B2} = -\frac{R1}{RG}V_1 + \left(1 + \frac{R1}{RG}\right)V_2$$

A partir de los puntos A y B se tiene un amplificador diferencial, por lo tanto:

$$V_O = V_B - V_A = -\frac{R1}{RG}V_1 + \left(1 + \frac{R1}{RG}\right)V_2 - \left(1 + \frac{R1}{RG}\right)V_1 + \frac{R1}{RG}V_2$$

$$V_O = \left(1 + \frac{2R1}{RG}\right)V_2 - \left(1 + \frac{2R1}{RG}\right)V_1$$

$$V_O = \left(1 + \frac{2R1}{RG}\right)(V_2 - V_1)$$

Para el caso de que la referencia del amplificador diferencial no sea cero, se tendrá lo siguiente:

$$V_O = \left(1 + \frac{2R1}{RG}\right)(V_2 - V_1) + V_R$$

Donde $V_R$ es un voltaje de offset.

## A.2 Diagramas de flujo del programa desarrollado para el microcontrolador

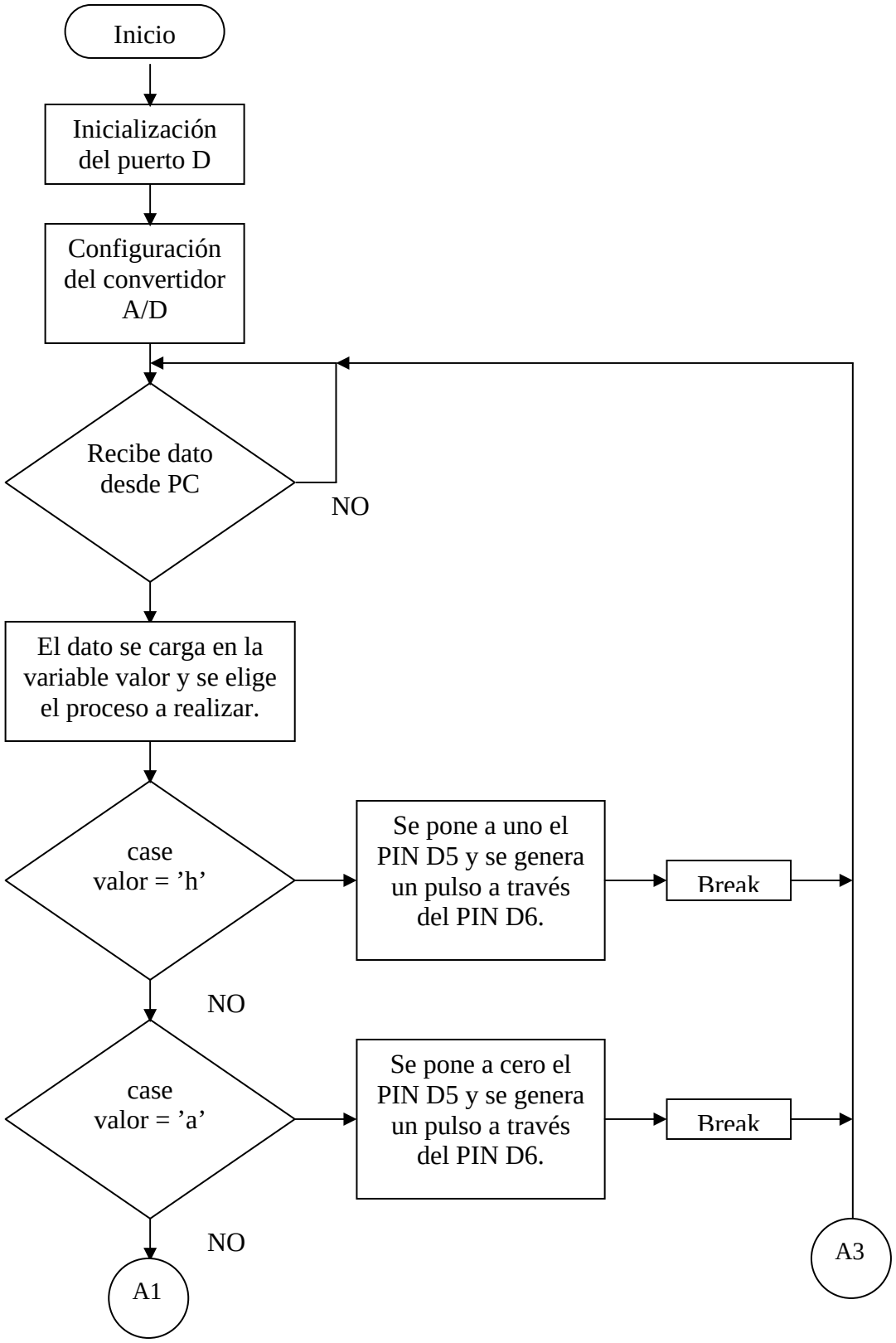



*Figura A.2.1 Diagrama de flujo del programa principal.*

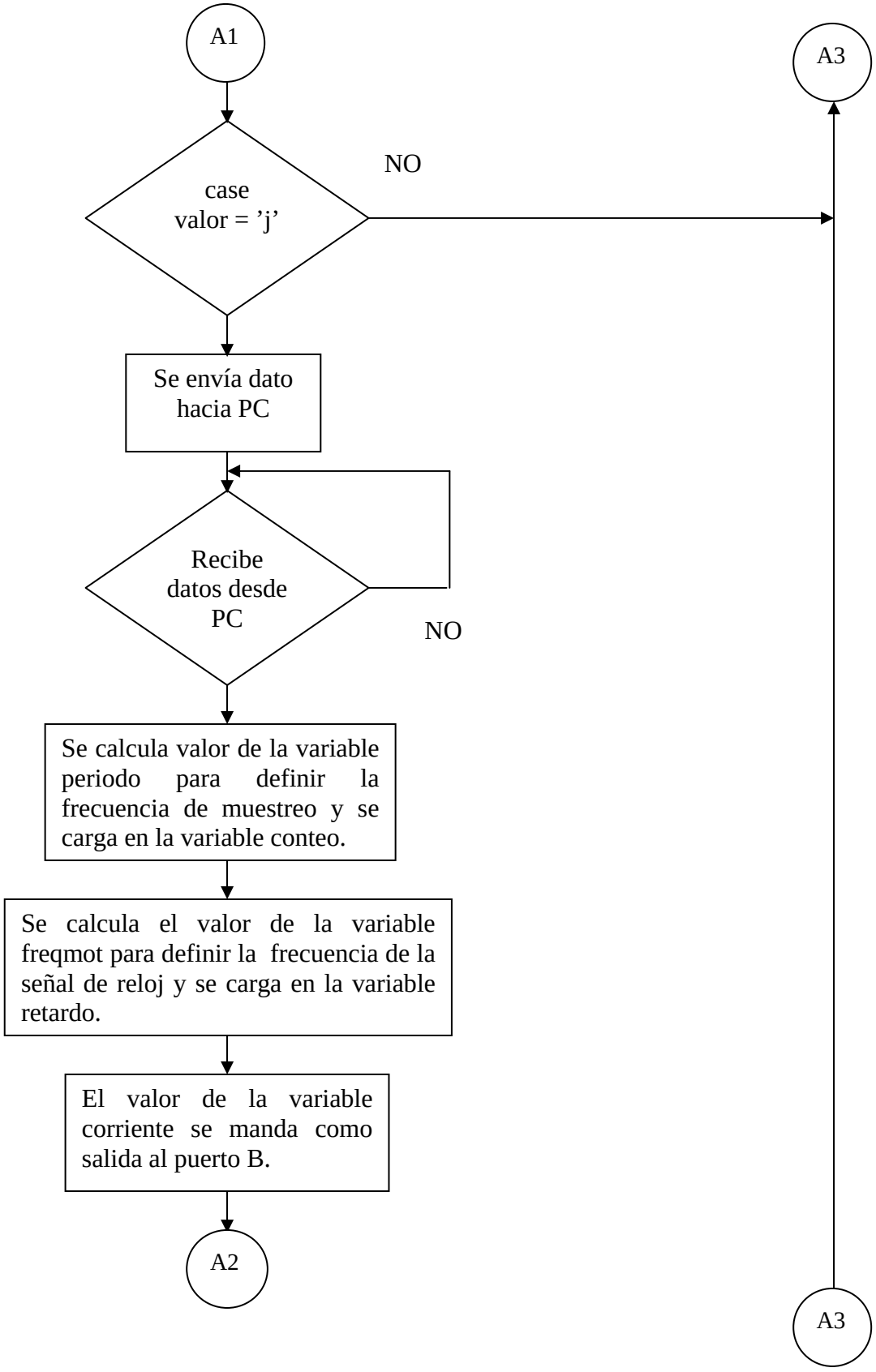



*Figura A.2.2 Continuación del diagrama de flujo del programa principal.*

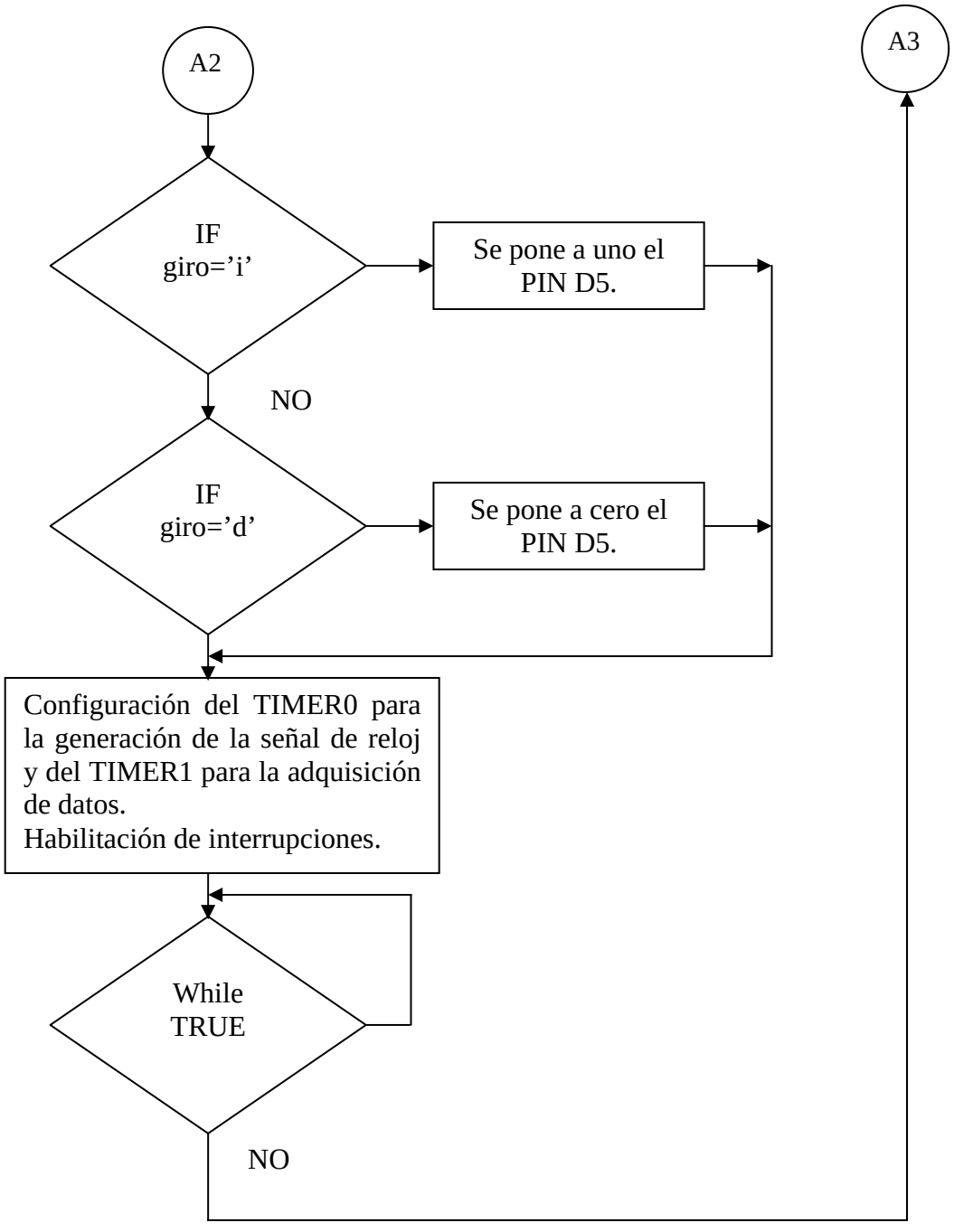



*Figura A.2.3 Continuación del diagrama de flujo del programa principal.*

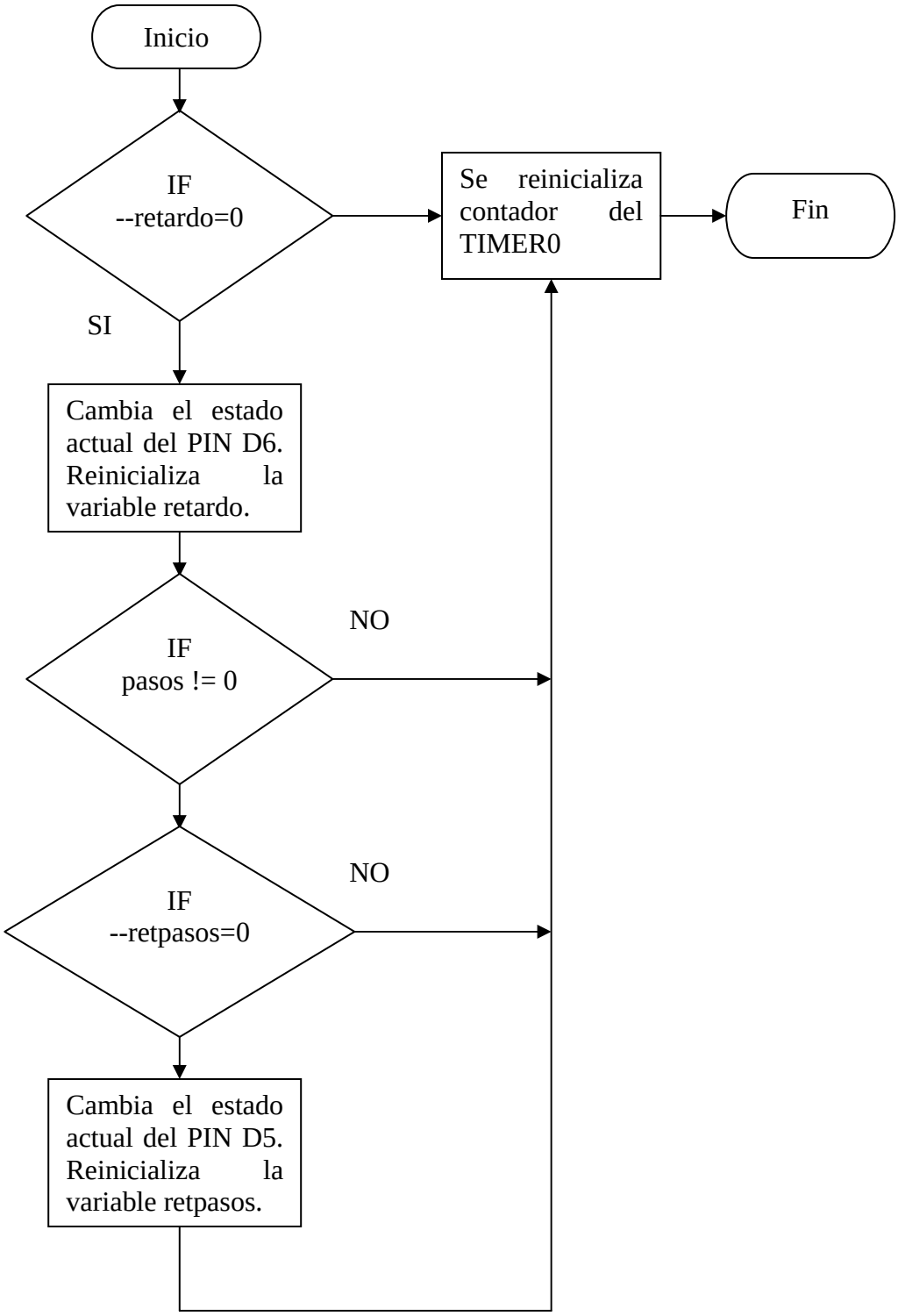



*Figura A.2.4 Diagrama de flujo de la interrupción del TIMER0.*

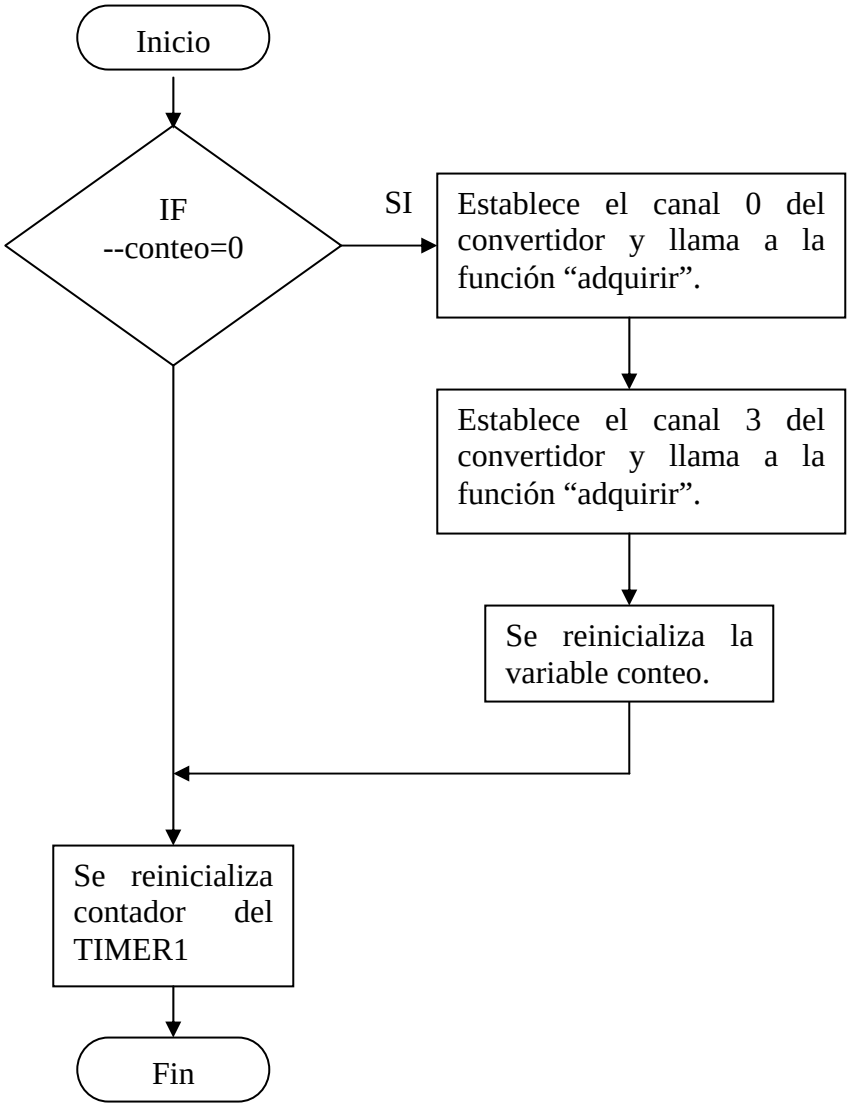



*Figura A.2.5 Diagrama de flujo de la interrupción del TIMER1.*

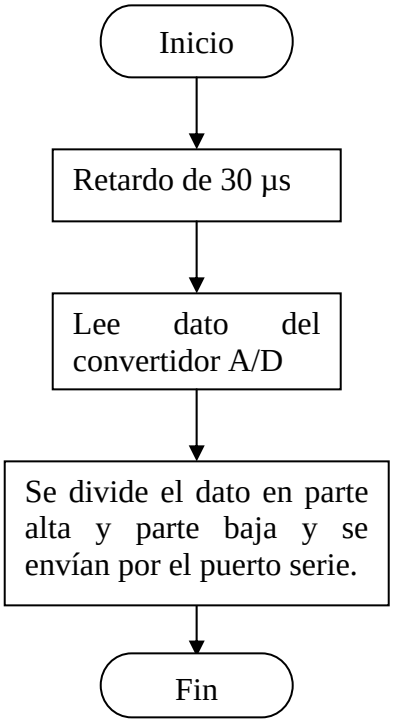



*Figura A.2.6 Diagrama de flujo de la función "adquirir".*

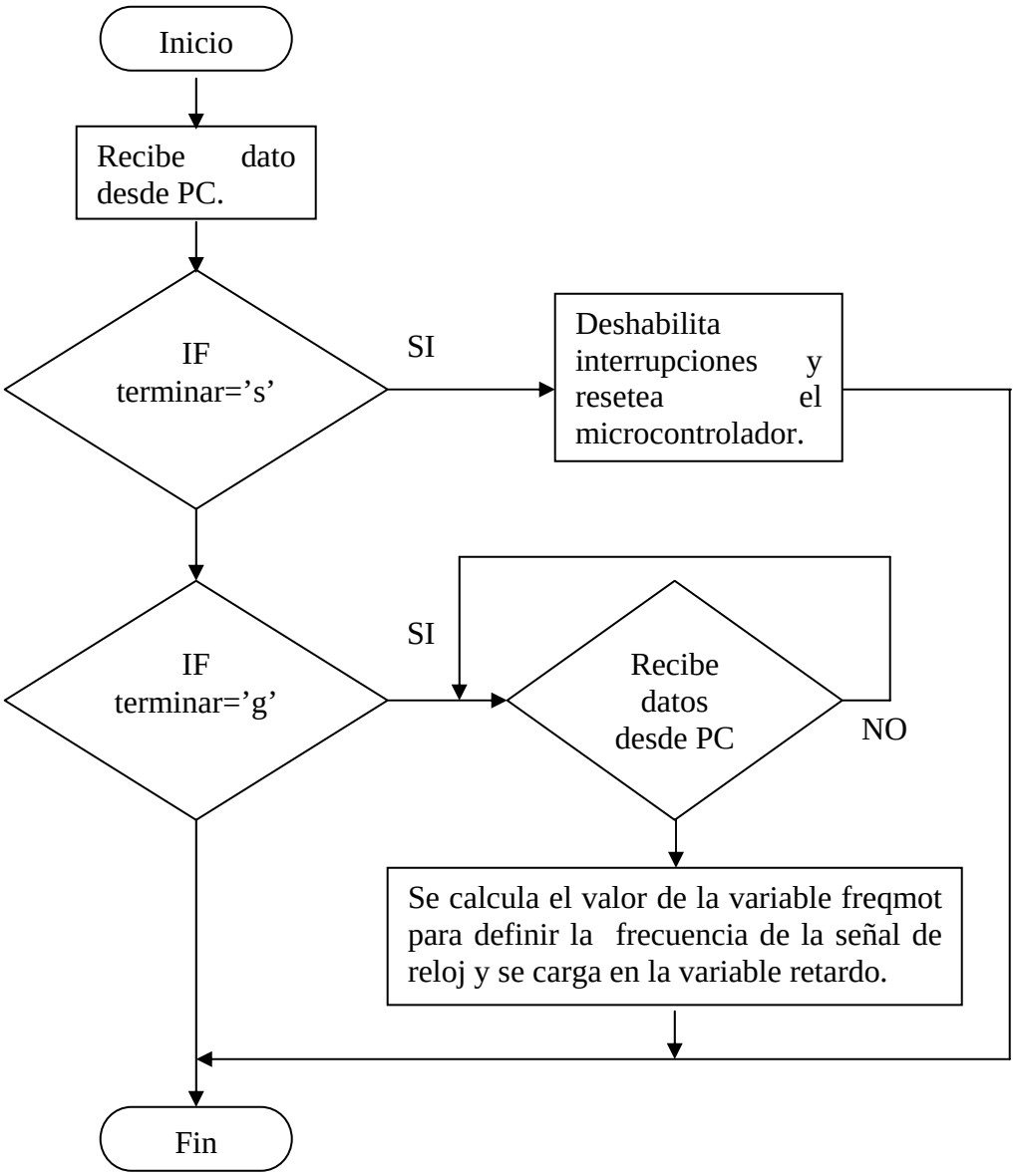



*Figura A.2.7 Diagrama de flujo de la interrupción RDA.*

## A.3 Fuente de alimentación para la interfaz de hardware

A continuación se muestran los diagramas de las fuentes utilizadas para la alimentación de la interfaz de hardware y la electroválvula. Cabe señalar que el diseño fue tomado de uno existente y no se realizaron modificaciones al original. Para el caso de la interfaz se realizó una fuente simétrica de ± 12 V y para la electroválvula se realizó una fuente variable considerando una alimentación de 25 V.

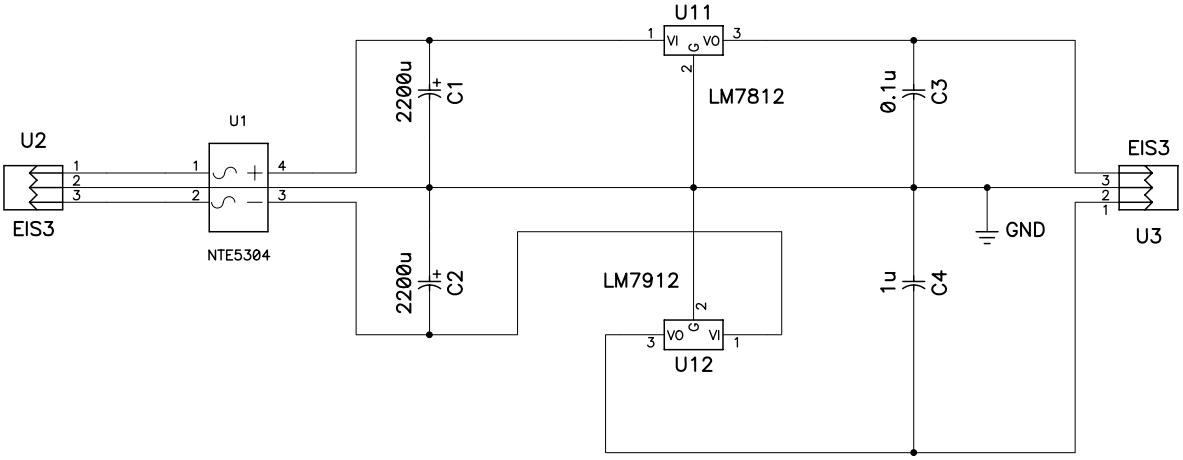



*Figura A.3.1 Fuente de alimentación simétrica ± 12 V.*

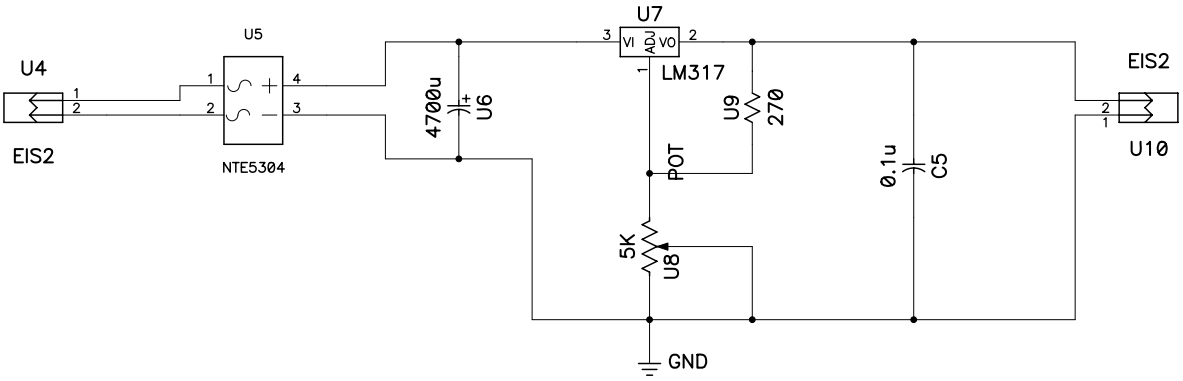



*Figura A.3.2 Fuente de alimentación de 25 V.*

En los conectores U2 y U4 se conectaron los transformadores correspondientes. Mientras que en los conectores U3 y U10 se obtuvieron los voltajes de salida deseados.

## A.4 Diagrama esquemático de la interfaz de hardware

El siguiente diagrama muestra el conjunto de módulos anteriormente mencionados y que constituyen la interfaz de hardware.

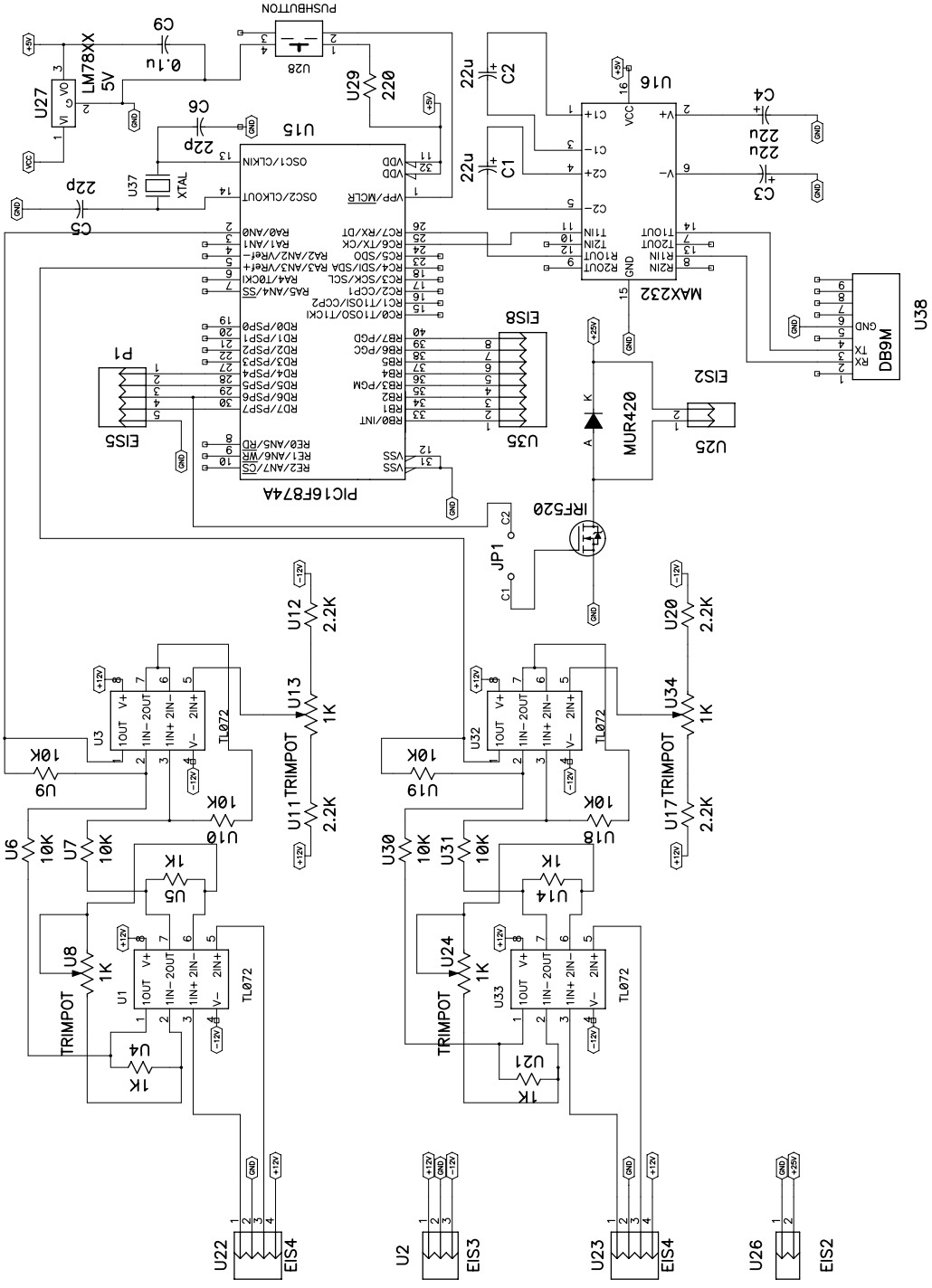



*Figura A.4.1 Interfaz de hardware.*

# 9. BIBLIOGRAFIA

Sergio Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*, México, Ed. McGraw-Hill, 1988.

Thomas Floyd, *Dispositivos Electrónicos*, México, Ed. Limusa, 2000.

Honeywell, *24PCBFA6G*, Data Sheet, 2004.

Honeywell, *Bridge Amplifiers for 20PC,170PC and FS Series*, Data Sheet

Texas Instruments, *TL072*, Data Sheet, 2005.

Intersil, *IRF520*, Data Sheet, 1999.

ON Semiconductor, *MUR420*, Data Sheet, 2005.

Microchip, *PIC16F87X*, Data Sheet, 1999.

Maxim, *MAX232*, Data Sheet, 2004.

Deitel, H. M. Deitel P. J. Nieto T. R. ,*Visual Basic 6: How to program*, Prentice Hall, New Jersey, 1999

Francisco Javier Ceballos Sierra, *Enciclopedia de Microsoft Visual Basic 6*, Alfaomega, México, 2000

**Sitios de Internet:**

Juan González Gómez, Cuaderno Técnico I: Comunicaciones Serie, http://www.iearobotics.com/proyectos/cuadernos/ct1/ct1.html

Manual del PIC, Simulador SIM 2000 Aprende a simular: http://www.simupic.com/

Carlos Aimacaña Toledo, Interfaz de usuario - Monografías.com, http://www.monografias.com/trabajos6/inus/inus.shtml