



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA ELECTRÓNICO PARA UN BANCO HIDRÁULICO DE PRUEBAS DIDÁCTICO

T E S I S

**QUE PARA OBTENER EL TÍTULO
DE:**

**INGENIERO ELÉCTRICO
ELECTRÓNICO**

P R E S E N T A

**JOSÉ AGUSTÍN MORENO
LARIOS**

DIRECTOR DE TESIS: M.I. BENJAMÍN VALERA OROZCO



Deseo expresar mi agradecimiento a mi tutor, Benjamín, el cual me ha guiado durante éste trabajo. A su vez, a la gente de Ingeniería de Procesos del CCADET, que me ha acogido en su grupo de trabajo.

A mis amigos de la Facultad, con los cuales he compartido estos cinco años, trabajando, riendo y llorando.

Al proyecto PAPIME “Diseño e instrumentación de un banco hidráulico de pruebas para determinar caídas de presión en accesorios y tuberías” Clave PE103413 por el apoyo recibido

Finalmente, a mis padres y hermanas. Sin su apoyo, no sé donde estaría en éste momento.

¡Muchas gracias!

José Agustín Moreno.

Índice temático

Resumen de la tesis	1
Introducción	3
Definición del problema	3
Antecedentes	3
Descripción del problema a resolver	4
Relevancia y justificación	4
Alcance y limitaciones	4
Relación con otras áreas	4
Método	5
Inventario de materias	6
Resultados esperados	6
1. Capítulo 1: Conceptos básicos	7
1.1. Principios de mecánica de fluidos	7
1.1.1. Propiedades de fluidos	7
1.1.2. Presión	8
1.1.3. Hidrodinámica	9
1.1.4. Pérdidas en tuberías	12

1.2. Sistemas de adquisición y acondicionamiento de señales	13
1.2.1. Transductores	14
1.2.2. Acondicionamiento de señales	17
1.2.3. Adquisición de señales	19
1.3. Microcontrolador PIC18F4553	21
1.3.1. Especificaciones Generales	22
1.3.2. Interrupciones	25
1.3.3. Temporizadores	26
1.3.4. Módulo de Comparación, Captura y PWM (CCP)	29
1.3.5. Convertidor Analógico/Digital	30
1.3.6. Universal Serial Bus	31
1.4. Programación en la MFC de Visual C++	33
2. Capítulo 2: Descripción del proyecto	39
2.1. Esquema general	39
2.2. Sistema electrónico	40
2.2.1. Acondicionamiento de señales	41
2.2.2. Control de bomba hidráulica	45
2.2.3. Etapa de potencia para la electroválvula	46
2.2.4. Microcontrolador PIC18F4553	48
2.2.5. Firmware del Microcontrolador	49
2.2.6. Alimentación eléctrica	57
2.3. Software de operación	58
2.3.1. Descripción del software en el ámbito del operador	58
2.3.2. Descripción del software en el ámbito del programador	66

3. Capítulo 3: Resultados	77
3.1. Resultados	77
3.1.1. Circuito electrónico	77
3.1.2. Software	80
3.1.3. Calibración	82
3.1.4. Experimento	90
3.2. Discusión	92
3.3. Trabajo a futuro	93
Conclusiones	95
Anexo A	97
Anexo B	101
Anexo C	105
Bibliografía	111

Resumen de la tesis

El sistema electrónico desarrollado será usado en un banco de pruebas hidráulico didáctico, por ello, se revisaron conceptos básicos de hidráulica, electrónica y programación para el diseño y construcción de éste. El sistema consta de sensores de presión y flujo, una electroválvula y una bomba hidráulica. Estos componentes son vinculados a una computadora con ayuda de un microcontrolador y un programa especial, que enlaza el sistema mediante USB. Finalmente, se probó el sistema con diversas pruebas, que muestra el funcionamiento de los sensores en diferentes condiciones de operación.

En el capítulo 1 se tratarán los conceptos básicos para la realización del proyecto. Estos abarcan conceptos de mecánica de fluidos, de acondicionamiento y adquisición de datos, el manejo del microcontrolador PIC18F4553, y la creación de proyectos de Visual C++ en el entorno de Visual Studio 2012.

En el capítulo 2 se describirán los bloques básicos para el funcionamiento del proyecto: la electrónica para el control del sistema y acondicionamiento de señales, el firmware desarrollado para el PIC18F4553, y el software desarrollado para realizar una interfaz de control en una computadora.

En el capítulo 3 se mostrará el funcionamiento del sistema bajo ciertas pruebas, se discutirá acerca de los resultados obtenidos, y se establecerá el trabajo a futuro relacionado con el sistema desarrollado.

En conclusiones se expone el trabajo realizado en esta tesis.

En el anexo A se mostrará el diagrama esquemático del circuito desarrollado.

En el anexo B se mostrará el diseño para el circuito impreso utilizado por el sistema.

En el anexo C se lista el material electrónico usado para la construcción del circuito usado en el proyecto.

En la bibliografía se listan las referencias literarias usadas para la elaboración de este trabajo.

Introducción

Definición del problema

El desarrollo de los sistemas electrónicos para la adquisición de datos y las modernas computadoras permiten hoy en día la solución y simulación de muchos problemas en mecánica de fluidos. En realidad, una gran parte de los problemas sólo tienen solución experimental. Un procedimiento típico para evaluar un sistema hidráulico consiste de los siguientes elementos:

- Construir un prototipo en tamaño real o a escala.
- Definir las variables de interés.
- Analizar el desempeño de las variables para diferentes condiciones experimentales.

En éste sentido, las pérdidas de energía en conductos cerrados se analizan en los laboratorios de hidráulica. El enfoque ha demostrado gran utilidad en la enseñanza de la mecánica de fluidos a nivel licenciatura.

La utilización de un equipo instrumentado que permita identificar la principales variables involucradas en el diseño de redes hidráulicas de manera clara, mejorará la comprensión del tema y permitirá que los alumnos puedan aplicar estos conocimientos de manera más eficiente durante su desarrollo profesional, todo esto apoyado en la comparación con resultados y tablas disponibles en la literatura que reforzarán la confianza de los estudiantes y profesores en los métodos y técnicas utilizadas en el laboratorio.

Antecedentes

El principal antecedente para el desarrollo del presente proyecto es el equipo Armfield Fluid Friction Apparatus, instalado actualmente en el Laboratorio de Mecánica de Fluidos de la Facultad de Ingeniería UNAM. Este equipo permite el estudio de la fricción de fluidos incompresibles y pérdidas secundarias que ocurren a través de tuberías y accesorios. El equipo se ha utilizado como parte fundamental de las prácticas 7, 8 y 9 en la materia de Mecánica de Fluidos I. La experiencia adquirida en el manejo del equipo por parte de los académicos, nos conduce a

proponer mejoras en la infraestructura actual y nuevos experimentos para la enseñanza y aprendizaje de la materia.

Descripción del problema a resolver

La experiencia adquirida por parte de la planta académica de la materia de Mecánica de Fluidos I nos motiva a desarrollar un prototipo de banco hidráulico de pruebas que incorpore elementos de adquisición de datos con base en los recursos de cómputo e instrumentación actuales. En este sentido, se desea construir un aparato para el análisis de pérdidas en fluidos que registre las caídas de presión y controle el gasto del fluido desde una computadora personal con software de desarrollo propio específico para que el alumno desarrolle sus prácticas de manera amigable.

Relevancia y justificación

La relevancia del proyecto radica en dos aspectos que la planta académica en la materia ha identificado como elementos que enriquecen el desarrollo de los futuros profesionistas del área. En primer lugar motivar al alumno en el uso de recursos modernos de instrumentación y programación de computadoras aplicados en el área de mecánica de fluidos. En segundo lugar el sistema propuesto servirá como plataforma de desarrollo para futuras aplicaciones y para apoyar la creación de infraestructura de desarrollo propio que puede ser ofrecida como una transferencia tecnológica.

Alcance y limitaciones

El alcance del presente proyecto es la obtención del conocimiento elemental y el desarrollo de una plataforma hardware y software básica que pueda ser utilizada en la enseñanza de la hidrodinámica en canales confinados.

Las limitaciones del prototipo por desarrollar radican en lo elemental de su principio de operación. Sin embargo, constituye un elemento valioso en el momento de transmitir el conocimiento básico que puede ser utilizado en aplicaciones futuras a partir de una herramienta básica.

Relación con otras áreas

Existe una gran relación con las áreas de mecánica de fluidos, instrumentación virtual, programación de computadoras y medición e instrumentación.

Método

En este trabajo se propone diseñar y construir un sistema electrónico con base en un microcontrolador para un banco de pruebas hidráulico que permita registrar la caída de presión, en diversos accesorios, por medio de un sistema de adquisición de datos y control, que permita la variación del caudal del fluido que circula por los accesorios y que cuente con elementos que permitan la visualización del flujo. El esquema general se presenta en la Figura 1.

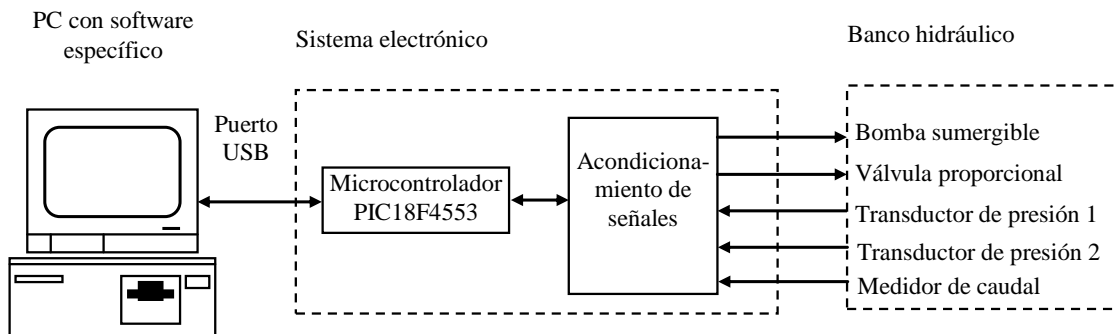


Figura 1: Método a emplear.

En el esquema de la figura 1 una PC con software de desarrollo específico para esta aplicación realiza las siguientes tareas.

- Registra las caídas de presión en los transductores 1 y 2 de acuerdo con requisitos previamente establecidos para el desarrollo de las prácticas.
- Controla el encendido y apagado de una bomba sumergible que impulsa la circulación del fluido por los diversos accesorios del banco hidráulico.
- Controla el caudal que circula por el banco hidráulico al estrangular el paso del fluido por medio de una válvula de control proporcional.
- Registra el caudal que circula en el circuito hidráulico por medio de un medidor de caudal.

El banco hidráulico en esencia es un circuito hidráulico de retorno de fluido en el cual se distribuyen diversos accesorios hidráulicos como codos, tes, yes, y válvulas que cuentan con conectores hidráulicos a los cuales se adaptan los transductores de presión. Conforme al desarrollo de las prácticas, los transductores de presión se conectan al accesorio en cuestión y se varía el caudal del fluido. De esta manera se grafican las caídas de presión en función del caudal.

Inventario de materias

- Análisis de circuitos eléctricos
- Máquinas eléctricas
- Diseño digital
- Electrónica digital
- Dinámica de sistemas físicos
- Fundamentos de control
- Sistemas embebidos
- Dispositivos electrónicos
- Sistemas electrónicos analógicos
- Circuitos integrados analógicos
- Análisis de circuitos eléctricos
- Amplificadores electrónicos
- Procesamiento digital de señales
- Electrónica de potencia
- Principios de termodinámica y electromagnetismo
- Medición e instrumentación
- Recursos y necesidades de México

Resultados esperados

Desarrollar y construir un sistema electrónico para un banco hidráulico de pruebas para determinar caídas de presión en accesorios con fines didácticos.

Capítulo 1

Conceptos básicos

Para la realización de este proyecto, se requiere una serie de conocimientos previos en los ámbitos de mecánica de fluidos, acondicionamiento y adquisición de señales, y de la programación en microcontroladores y computadoras. Éste capítulo trata de los conceptos básicos más importantes para el desarrollo del sistema.

1.1 Principios de mecánica de fluidos

La mecánica de fluidos es una rama de la física que se dedica a estudiar fluidos (líquidos, gases y plasmas), y para propósitos de este trabajo, se revisarán conceptos generales, tales como propiedades de fluidos, como Viscosidad, Densidad y Peso específico; de Presión, con Presiones Barométricas y Manométricas; de Hidrodinámica, con Flujo Laminar y Turbulento en tuberías y conceptos relacionados con las Pérdidas en Tuberías, como son la Ecuación de Darcy y el Número de Reynolds.

Estos conceptos involucran una descripción del fenómeno, unidades en el Sistema Internacional, y algunas ecuaciones básicas empleadas en este campo.

1.1.1 Propiedades de fluidos

Para la solución de cualquier problema de flujo de fluidos requiere conocimiento de las propiedades físicas del fluido a manejar. Se han determinado valores precisos para las propiedades que afectan el flujo de fluidos, principalmente, viscosidad y el peso específico, estos se encuentran en diversas tablas y gráficas.

Viscosidad

La viscosidad expresa la facilidad con la cual un fluido fluye cuando se ejerce una fuerza externa a este. El coeficiente de viscosidad absoluta (dinámica) es una medida de la resistencia a deformaciones internas o fuerzas de cizalla. Su símbolo es

μ y sus unidades son [Pa*s]. Por ejemplo, la miel es un fluido mucho más viscoso que el agua, y la viscosidad de los gases es muy pequeña comparada con ésta.

La viscosidad cinemática es la razón de la viscosidad absoluta a la densidad del fluido. Su símbolo es ν , y sus unidades son [m²/2], o comúnmente, Stoke (1 St = 10⁻⁴ [m²/s]). (Crane Engineering Department, 1969)

Densidad

La densidad de un fluido es la cantidad de materia en una unidad de volumen, comúnmente, se expresa en kilogramos por metro cúbico. A condiciones determinadas de temperatura y presión, un fluido posee una densidad definida. Aunque la densidad dependa de la temperatura y presión, la variación de la densidad al modificar estas variables puede ser grande o pequeña. Si la densidad varía poco para cambios moderados de presión y temperatura, se dice que el fluido es no compresible, si la densidad varía considerablemente con estos cambios, entonces se dice que el fluido es compresible.

Peso específico, volumen específico y gravedad específica

El peso específico de una sustancia es su peso por cada unidad de volumen. En el Sistema Internacional, la unidad manejada es Newton por metro cúbico. La expresión matemática es:

$$\gamma = \rho g \quad (3.1)$$

Dónde:

- γ es el peso específico, expresada típicamente en [N/m³]
- ρ es la densidad de la sustancia, expresada típicamente en [kg/m³]
- g es la aceleración debida a la gravedad, expresada típicamente en [m/s²]

El volumen específico es el recíproco del peso específico.

La gravedad específica es una medida relativa al peso específico. Debido a que la presión tiene un efecto insignificante en el peso específico de los líquidos, la temperatura es la única condición que debe ser considerada al designar la base para la gravedad específica. Se define como la razón de la densidad de un fluido a una temperatura dada, con respecto a la densidad del agua a esa misma temperatura.

1.1.2 Presión

La presión es la fuerza superficial que ejerce un fluido sobre las paredes del recipiente que lo contiene. En cualquier parte del interior de un fluido existe también una presión determinada. (McCabe, y Smith, 1972)

La Presión Barométrica es el nivel de la presión medido con respecto al vacío perfecto.

La Presión Manométrica es la presión medida con referencia a la presión atmosférica.

Las unidades empleadas en el Sistema Internacional son Pascales (Pa), se define como $[N/m^2]$

1.1.3 Hidrodinámica

La hidrodinámica es una rama dentro de la mecánica de fluidos, que se dedica al estudio de los líquidos en movimiento. En este apartado, se tratarán de los principios concernientes al flujo de fluidos en tuberías

Naturaleza del flujo en tuberías: Laminar y turbulento

Un experimento sencillo muestra de forma rápida la existencia de dos tipos de flujo completamente diferentes en una tubería. El experimento consiste en inyectar un pequeño chorro de un fluido coloreado en un líquido fluyendo en una tubería de vidrio y observar el comportamiento de estos chorros en diferentes secciones posteriores al punto donde se inyectaron.

Si la velocidad de descarga o velocidad promedio es pequeña, los hilos de fluido coloreado viajan en líneas rectas, Figura 1.1. Al aumentar la velocidad de flujo, estos hilos seguirán viajando en línea recta hasta alcanzar una velocidad cuando los hilos comenzarán a romperse en patrones difusos. La velocidad en la que ocurre esto es llamada “velocidad crítica”, Figura 1.2. A velocidades mayores a la “crítica”, los filamentos se dispersan de forma aleatoria a través del cuerpo principal del fluido, Figura 1.3. (Crane Engineering Department, 1969)



Figura 1.1: Flujo laminar.

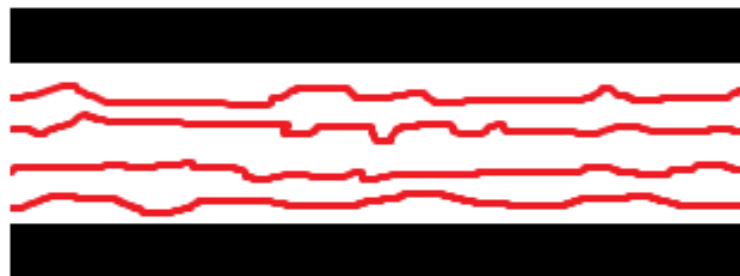


Figura 1.2: Flujo en Zona Crítica.



Figura 1.3: Flujo Turbulento.

El tipo de flujo que existe a velocidades menores a la “crítica” es conocido como flujo laminar. El flujo de esta naturaleza se caracteriza por capas deslizantes cilíndricas concéntricas, que pasan una sobre otra de forma ordenada. La velocidad del fluido es máxima en el eje de la tubería y decrece rápidamente a cero en la pared.

A velocidades mayores a la “crítica”, el flujo es turbulento. En flujo turbulento, existe un movimiento aleatorio irregular de partículas de fluido en direcciones transversales a la dirección del flujo principal. La distribución de velocidad en el flujo turbulento es más uniforme que en el flujo laminar. Aunque exista movimiento turbulento en la mayor parte de la tubería, siempre existe una capa delgada de fluido en la pared de la tubería, conocida como “capa límite” o “sub-capa laminar”, la cual tiene flujo laminar. (Crane Engineering Department, 1969)

Velocidad de flujo promedio

Se refiere a la velocidad de flujo promedio dada una sección transversal, como se muestra en la ecuación (1.2) para flujo en régimen permanente:

$$v = \frac{Q}{A} \quad (3.2)$$

Donde:

- v es la velocidad de flujo promedio de un fluido
- Q es el gasto volumétrico del fluido
- A es el área transversal de la tubería.

Número de Reynolds

El trabajo de Osborne Reynolds ha mostrado que la naturaleza del flujo en tuberías depende del diámetro de la tubería, la densidad y la viscosidad del fluido, así como la velocidad del flujo. El valor numérico de esta combinación adimensional de estas cuatro variables, conocida como el número de Reynolds, puede considerarse como la razón de las fuerzas dinámicas de flujo másico con respecto a las fuerzas cortantes debidas a la viscosidad. El número de Reynolds es:

$$R_e = \frac{Dv\rho}{\mu} \quad (3.3)$$

Donde:

- D es el diámetro de la tubería
- v es la velocidad de flujo del fluido
- ρ es la densidad del fluido
- μ es la viscosidad absoluta del fluido

Para ingeniería, se considera que el flujo de una tubería es laminar si el número de Reynolds es menor a 2000, y turbulento si es mayor a 4000. Entre estos dos valores está la “zona crítica” donde el flujo puede ser impredecible. (Crane Engineering Department, 1969)

Radio Hidráulico

Ocasionalmente se encuentra con un conducto cuya sección transversal es no circular. Para calcular el número de Reynolds en esta condición, el diámetro equivalente (cuatro veces el radio hidráulico) es sustituido por el diámetro circular. (Crane Engineering Department, 1969)

$$R_H = \frac{\text{Área seccional del conducto}}{\text{Perímetro Húmedo}} \quad (3.4)$$

Teorema de Bernoulli

El teorema de Bernoulli es una forma de expresar la aplicación de la ley de conservación de energía al flujo de fluidos en un conducto. La energía total en cualquier punto, arriba de algún nivel de referencia horizontal arbitrario, es igual a la suma de la cabeza de elevación, la cabeza de presión y la cabeza de velocidad, como sigue:

$$Z + \frac{P}{\rho g} + \frac{v^2}{2g} = H \quad (3.5)$$

Donde:

- Z es la cabeza potencial a partir del nivel de referencia
- P es la presión
- ρ es la densidad
- g es la aceleración debida a la gravedad
- v es la velocidad de flujo promedio
- H es la cabeza total de fluido

La cabeza hidráulica es la altura que tiene el fluido a partir del nivel de referencia.

Si se ignoran las pérdidas por fricción y no se agrega ni retira energía al sistema de tuberías (bombas o turbinas), la cabeza total, H , en la ecuación anterior, será una

constante en cualquier punto del fluido. Sin embargo, en la práctica, se encuentran pérdidas o incrementos o decrementos de energía, y estos deben ser incluidos en la ecuación de Bernoulli. (Crane Engineering Department, 1969)

1.1.4 Pérdidas en tuberías

El flujo en una tubería siempre es acompañado por fricción de partículas de fluido frotándose unas con otras, así como por el frotamiento de las partículas de fluido con las partículas de la tubería; en consecuencia, de pérdidas de energía.

En este apartado se tratará de la teoría que modela dichas pérdidas.

Fórmula de Darcy

Gracias a las fuerzas de fricción anteriormente descritas, debe de existir una caída de presión en la dirección de flujo. Si se conectan dos manómetros de Bourdon en una tubería, como se muestra en la Figura 1.4, P_1 indicaría una presión mayor que en P_2

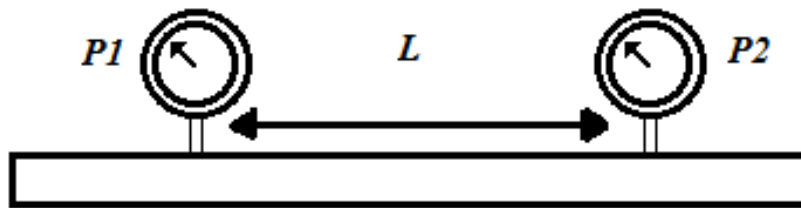


Figura 1.4: Tubería con manómetros de Bourdon.

La ecuación general para caída de presión, conocida como la Fórmula de Darcy, y expresada en metros de fluido es:

$$h_L = \frac{fLv^2}{2Dg} \quad (3.6)$$

Donde:

- h_L es la pérdida de cabeza debida al flujo de fluido.
- f es el factor de fricción de Darcy.
- v es la velocidad de flujo promedio.
- D es el diámetro de la tubería.
- g es la aceleración causada por la gravedad.
- L es la longitud de la tubería.

La ecuación de Darcy es válida para flujo laminar o turbulento de cualquier líquido en tuberías. Sin embargo, cuando existen velocidades extremas en la tubería que causan que la caída de presión alcance al nivel de presión de vapor del líquido,

se producen cavitaciones y los cálculos realizados serán imprecisos. (Crane Engineering Department, 1969)

Factor de fricción

La fórmula de Darcy puede derivarse racionalmente por análisis dimensional, con la excepción del factor de fricción f , el cual tiene que ser determinado experimentalmente. El factor de fricción para condiciones de flujo laminar ($R_e < 2000$) está en función exclusivamente del número de Reynolds; y para flujo turbulento ($R_e > 4000$), también es función de las características de la pared de la tubería.

Hay una región conocida como “zona crítica”, que existe en números de Reynolds entre 2000 y 4000, aproximadamente. En esta región, el flujo puede ser laminar o turbulento, dependiendo de varios factores; estos incluyen cambios en la sección o dirección del flujo y obstrucciones, tales como válvulas, en el circuito hidráulico. El factor de fricción de esta región es indeterminado y tiene límites inferiores basados en condiciones de flujo laminar y límites superiores basados en condiciones de flujo turbulento.

A números de Reynolds mayores de 4000, las condiciones de flujo se vuelven estables y se pueden establecer factores de fricción definidos. Esto es importante porque permite al ingeniero determinar las características de flujo de cualquier fluido en una tubería, si se conoce la viscosidad y peso específico para las condiciones de flujo. (Crane Engineering Department, 1969)

Efecto de la edad y uso de la tubería en la fricción

Las pérdidas por fricción en una tubería son sensibles al cambio en el diámetro y rugosidad de la tubería. Para un gasto y factor de fricción dados, la caída de presión por metro de una tubería varía inversamente con la quinta potencia del diámetro. Entonces, una reducción de 2% de diámetro causa un 10% de incremento en la caída de presión; una reducción de diámetro del 5% incrementa la caída de presión en un 23%. En varios servicios, las tuberías son incrustadas por sedimentos, tubérculos, u otros, por lo que, es prudente permitir cambios esperados en el diámetro de la tubería. (Crane Engineering Department, 1969)

1.2 Sistemas de adquisición y acondicionamiento de señales

Dentro de los sistemas de control electrónicos actuales, es necesario poder tomar “muestras” del mundo exterior. Para esto, se emplean transductores capaces de transformar algún fenómeno físico a una manifestación eléctrica, y posteriormente, se procesa para que ésta señal sea compatible con el sistema de conversión analógico/digital, con el fin de obtener una lectura lo más cercana a la realidad.

Para el sistema que se desarrolló, se emplearon tres tipos de transductores: Sensores de Presión basados en galgas extensiométricas, Sensores de flujo de paletas, y un detector de posición basado en un potenciómetro lineal con múltiples vueltas. En este trabajo, se tratará sobre su funcionamiento y procesamiento, además de unos conceptos básicos para la conversión analógico/digital realizada en el microcontrolador PIC18F4553.

1.2.1 Transductores

Un transductor es un dispositivo que actúa al recibir energía de un sistema, y suministra energía, usualmente en otra forma, a otro sistema. Se pueden clasificar en dos familias: transductores de entrada (sensores de presión, fuerza y temperatura) y transductores de salida (Bocinas y solenoides). Para fines de este trabajo, se revisarán conceptos de transductores de entrada, actuados por variables físicas, que entregan señales eléctricas a un sistema de control y medición.

Sensor de presión con base en galgas extensiométricas

La presión en líquidos es medida eléctricamente por una variedad de transductores de presión. Se usa una variedad de convertidores mecánicos, para medir presión mediante una longitud, distancia, desplazamiento y cambios en la presión por el movimiento producido. La salida de ésta interfaz mecánica se aplica a un convertidor electrónico.

Las galgas extensiométricas se pueden usar como transductores de presión. La galga extensiométrica es un elemento resistivo que cambia en longitud, lo que causa cambios en su resistividad, en función de la fuerza aplicada a la base en la cual esté montado. Al construirse como un puente de Wheatstone, la galga convierte una fuerza aplicada en un potencial de salida de puente, como se ve en la Figura 1.5.

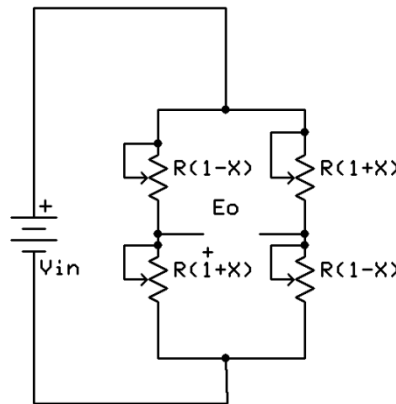


Figura 1.5: Las galgas extensiométricas se construyen como un puente de Wheatstone.

Las galgas son dispositivos de baja impedancia; requieren niveles significativos de energía de excitación para obtener un voltaje de salida a niveles razonables. Una celda de carga basada en galgas típica tendrá una impedancia de $350 \text{ } [\Omega]$ y su

sensibilidad se especifica en términos de milivolt por volt de excitación a escala completa. El máximo potencial de excitación y el potencial recomendado, son especificados por el fabricante. Para un dispositivo que trabaja a 10 [V], con una sensibilidad de 3 [mV]/[V], se tendrán 30 [mV] de señal de salida en escala completa. Se puede aumentar la amplitud de la señal de salida si se incrementa la alimentación al puente, pero los efectos de calentamiento se vuelven limitantes importantes: pueden causar lecturas erróneas o la destrucción del dispositivo si se prolonga el uso.

El bajo voltaje de salida de este tipo de transductores, en la mayoría de los casos, es debido a los pequeños cambios en la resistencia de las galgas. Sin embargo, mientras esto es una limitación importante para la ganancia, favorece a la linealidad. (Sheingold, 1980)

Sensores de flujo a paletas

El sensor a rueda a paletas a veces es descrito como un sensor de flujo a turbina. Ambos tienen un rotor, un rodamiento, y un eje, pero existe una diferencia importante: El eje de un sensor a turbina está en línea con la dirección de flujo, mientras que el eje del sensor a paletas se encuentra paralelo al flujo. Comparándolo con el eje de un sensor a turbina, el eje del sensor a paletas recibe una carga axial menor, y por lo tanto, menor fricción.

Un sensor a turbina tiene más álabes para recibir el momento del flujo, así que se necesita un impulso mayor para mover el rotor y superar la fricción del dispositivo. En lugar de hacer cada parte más resistente y grande, los desarrolladores de sensores a paletas toman ventaja del menor empuje para lograr una buena eficiencia y respuestas a bajos flujos.

La Figura 1.6 (Chen, sin fecha) muestra un diseño popular para el rotor y el eje. Se hace ver que el rodamiento está integrado al rotor, y que el eje se desliza a través del rotor como un eje central de giro. Una forma alternativa, el eje está conectado al rotor, y el rodamiento se construye en el cuerpo principal del sensor. En el ejemplo descrito en la figura, el rotor incorpora un imán que activa el transductor inherente al cuerpo principal.

La Figura 1.7 (Chen, sin fecha) ilustra un mecanismo de detección común. El transductor puede ser una bobina de inducción. El imán puede ser de alnico o cobalto samario. Si el rotor es un material ferromagnético, el sensor puede ser de tipo reluctante o una bobina de inducción. La frecuencia de salida de este dispositivo es proporcional a la velocidad del flujo que pasa a través de la tubería donde está instalado. (Chen, sin fecha)

Es posible conocer la posición de un motor con la ayuda de un potenciómetro fijado directamente en la flecha de éste, de forma que, al moverse, cambie un nivel de voltaje en proporción a dicho movimiento. Dicho concepto se muestra en la

Figura 1.8. Una limitación es que el motor no puede rotar libremente, pero es útil cuando este se va a mover en un rango limitado, como en una válvula.

Densidad de la rueda= (Peso imán + Peso del Aire + Peso del Plástico)/(Volumen del rotor) - Densidad del Agua

Brecha= (Diámetro de rodamiento - Diámetro del eje)/2

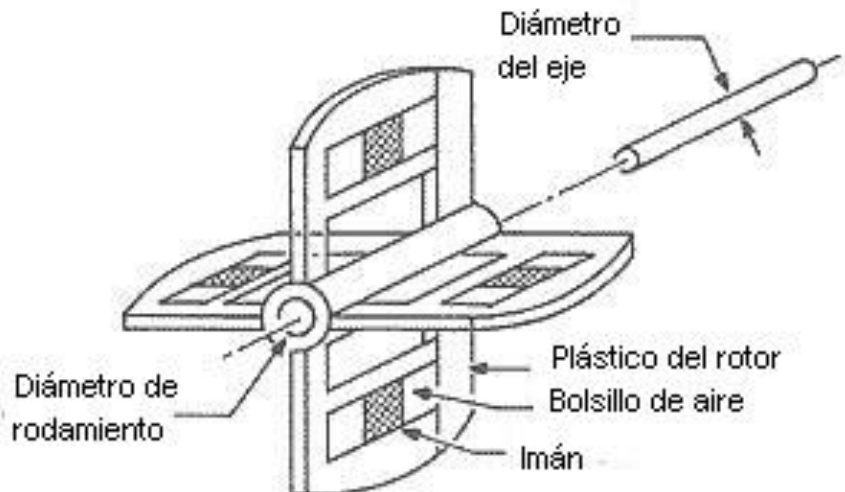


Figura 1.6: Diseño popular para sensor de flujo a paletas.

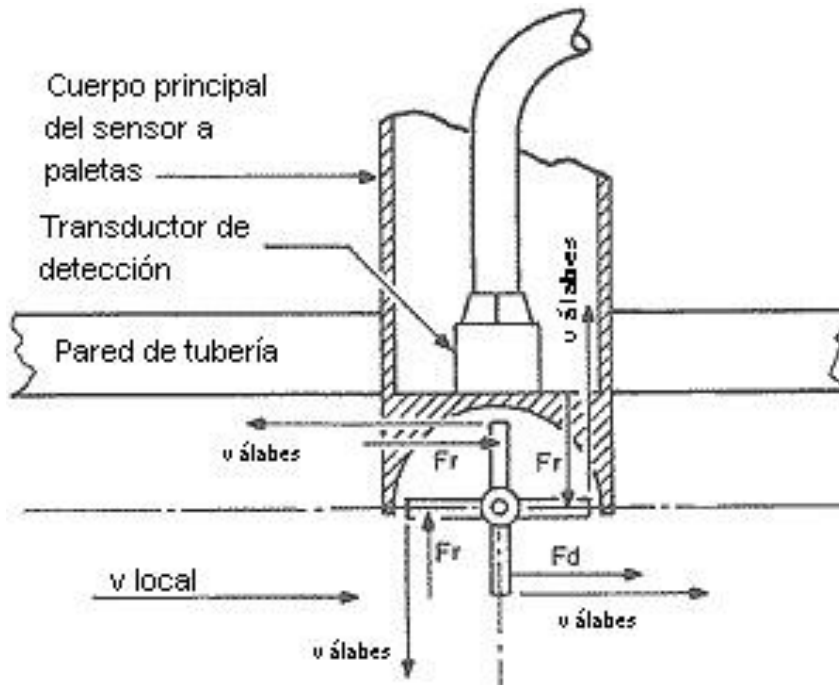


Figura 1.7: Mecanismo de detección de sensor de flujo.

Control de posición de motor a partir de un potenciómetro

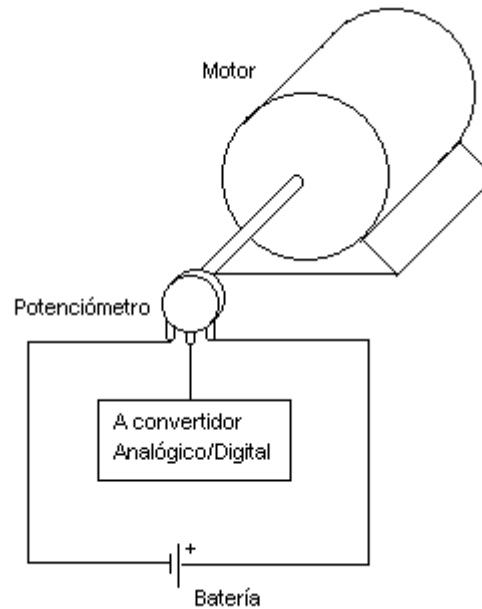


Figura 1.8: Control de posición empleando potenciómetro.

1.2.2 Acondicionamiento de señales

Una vez que se ha seleccionado un tipo de transductor para un trabajo dado, se tiene que proporcionar una excitación adecuada y un tratamiento adecuado a la señal de salida. La naturaleza del condicionamiento depende de las características del transductor y el destino de la señal de salida. Procesos típicos incluyen aislamiento galvánico, transformación de impedancias, traducción de nivel, amplificación, linealización y cálculos diversos (analógicos o digitales). Estos procesos pueden ocurrir en las inmediaciones del transductor, en un sistema de adquisición de datos remoto, o paso a paso en diferentes lugares.

Sin importar la forma que el acondicionamiento de señales tenga, la circuitería y el desempeño serán gobernados por las características eléctricas del transductor y su salida. La caracterización precisa de los transductores en términos de parámetros apropiados para la aplicación, pueden establecer la diferencia entre una aplicación pobre o exitosa del dispositivo, especialmente en casos cuando se requiere una alta resolución y precisión, o mediciones de bajo nivel. (Sheingold, 1980)

Sensor de presión con base en galgas extensiométricas

Debido a la forma en que están configuradas las galgas (Puente de Wheatstone), la señal obtenida de estas está en el rango de los milivolt. Por eso, el amplificador de instrumentación, mostrado en la Figura 1.9, es una opción conveniente para su acondicionamiento. Este amplificador es un bloque de ganancia, generalmente caracterizado por una deriva baja, un alto rechazo al modo común, una impedancia de entrada alta, y tiene la capacidad de mantener su desempeño en un rango de

ganancias. La ganancia es función de la relación de dos resistores, los cuales no están conectados con las entradas; la ganancia puede ser ajustada al modificar la relación de resistores. (Sheingold, 1980)

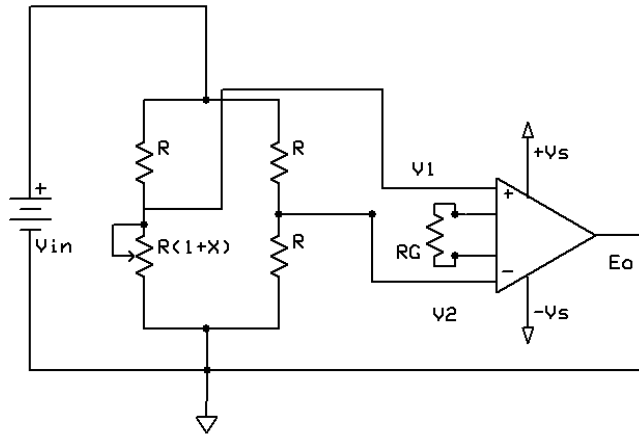


Figura 1.9: Amplificador de instrumentación para medición de puentes.

Donde:

$$E_o = G(V_1 - V_2) \quad (3.7)$$

Y G está en función de R_G .

Con referencia a la figura 1.9, cuando X es cero, la salida debe ser cero. Si el amplificador ha sido ajustado para cuando $V_{in}=0$, entonces cualquier error relacionado a V_{in} que aparezca en la salida es conocido como voltaje de error de modo común. La habilidad del circuito amplificador para minimizar este error es conocido como rechazo al modo común, una cantidad expresada logarítmicamente en decibeles.

El amplificador de instrumentación tiene una entrada diferencial balanceada. Esto quiere decir que el voltaje de salida es proporcional a la diferencia entre los voltajes de entrada; y que las terminales de entrada, las cuales representan una alta impedancia a la fuente de entrada, son eléctricamente similares. Un alto rechazo al modo común significa que el amplificador es sensible solamente a la diferencia entre los voltajes de entrada, incluso si están alternando en un rango amplio, y la diferencia es muy pequeña.

Debido a que el amplificador de instrumentación tiene entradas diferenciales, es útil para tomar lectura de una gran variedad de transductores, balanceados o desbalanceados. En aplicaciones de muy alta precisión, surgen problemas cuando los bajos niveles de deriva y error de modo común son excesivos. Además, los amplificadores de instrumentación pueden ser inadecuados en aplicaciones que requieran un alto rechazo al modo común con voltajes de falla u operación potencialmente destructivos. (Sheingold, 1980)

Sensor de flujo a paletas

Una ventaja importante de los sensores de flujo a paletas es que la señal se puede transmitir con una inmunidad al ruido considerable, a pesar del poco procesamiento (si hay) en el extremo final; además, la señal se puede aislar y eventualmente ser transformada en información analógica o digital. (Sheingold, 1980)

Para realizar la transformación de la señal generada por el sensor de flujo a una señal compatible con un sistema digital, se usa un comparador, como se muestra en la Figura 1.10.

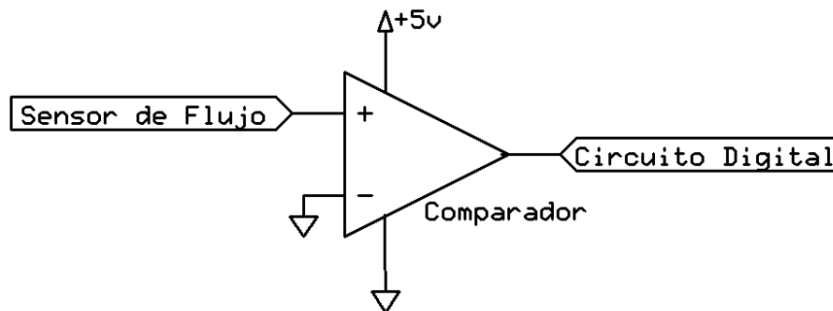


Figura 1.10: Conversión de una señal sinusoidal a cuadrada compatible con lógica TTL.

1.2.3 Adquisición de señales

El módulo de conversión analógico/digital del PIC18F4553 cuenta con 13 entradas para el PIC18F4553. Éste módulo permite la conversión de una señal de entrada analógica a un número digital de 12-bit. El microcontrolador tiene integrado un circuito de “sample and hold” (S/H), y utiliza un convertidor analógico/digital de aproximaciones sucesivas (Microchip, sin fecha). Ambos serán descritos a continuación.

Sample and Hold

El S/H ideal es simplemente un interruptor controlando un capacitor que sostiene un voltaje, seguido por un buffer de alta impedancia de entrada. La impedancia del buffer debe ser lo suficientemente alta para que el capacitor no se descargue más que un bit menos significativo durante el periodo de fijación. El S/H, como se ve en la Figura 1.11 (Kester, 2004) muestrea la señal en el modo de sample, y fija la señal a un nivel constante durante el periodo de hold. Se ajusta la sincronización del sistema para que el codificador realice la conversión durante el periodo de hold. Un convertidor analógico/digital de muestreo puede procesar señales rápidas – el límite superior de frecuencia es determinado por el propio S/H, no el codificador. (Kester, 2004)

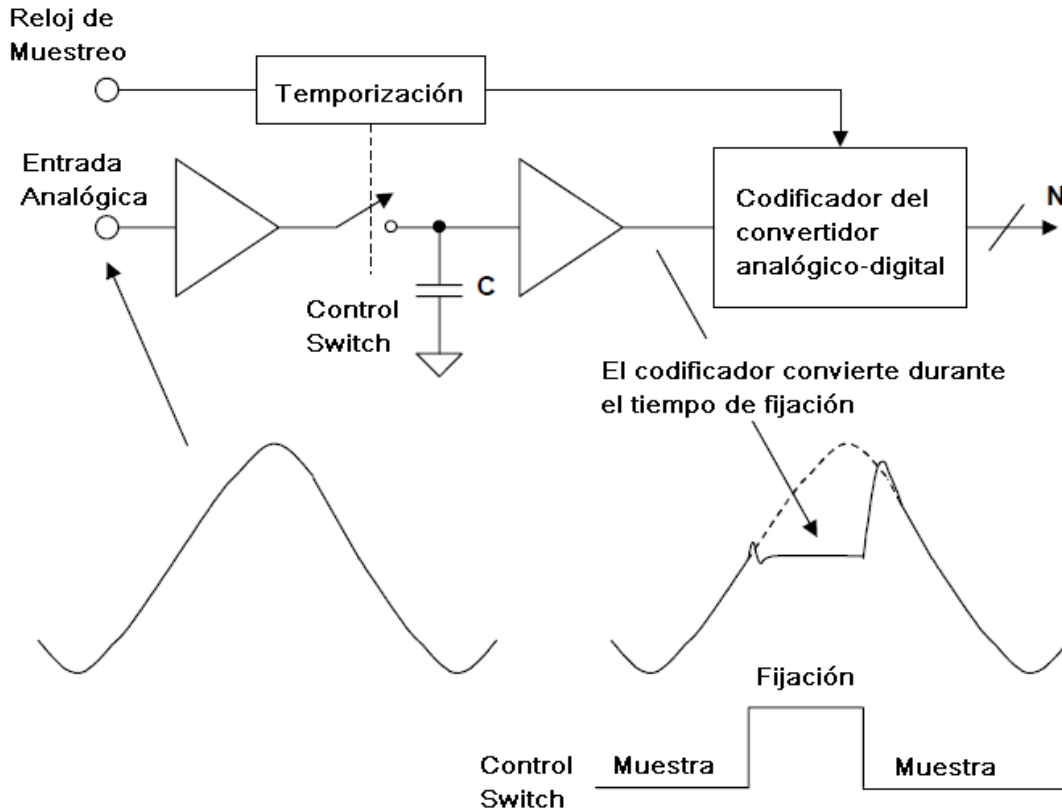


Figura 1.11: ADC con Sample-and-Hold.

Convertidor Analógico/Digital de Aproximaciones Sucesivas

Consiste en un registro de Aproximaciones Sucesivas, que realiza la cuenta al probar todos los valores de bits comenzando con el bit más significativo y termina en el bit menos significativo. Durante el proceso de conversión, el registro monitorea la salida del comparador para ver si la cuenta binaria es menor o mayor que la magnitud de la señal analógica de entrada, ajustando los valores en bits de forma correspondiente. La ventaja de éste método es que realiza la conversión analógico/digital mucho más rápido que el método de rampa (Kuphaldt, sin fecha). Un diagrama a bloques es presentado en la Figura 1.12. (Flye, 2006)

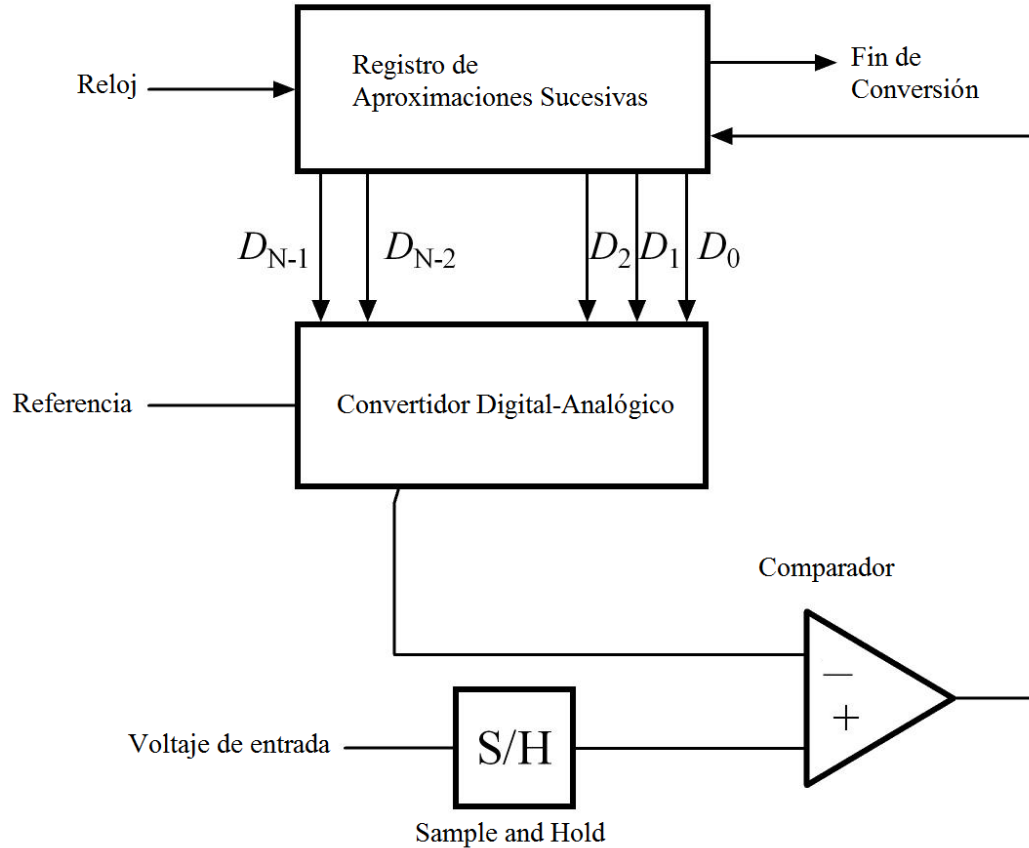


Figura 1.12: Diagrama simplificado de un convertidor analógico/digital de aproximaciones sucesivas.

La resolución del convertidor analógico/digital se determina por la siguiente ecuación:

$$1 \text{ LSB} = V_{\text{Referencia Negativa}} + \frac{V_{\text{Referencia positiva}} - V_{\text{Referencia Negativa}}}{2^{12}} \quad (3.8)$$

Donde LSB es un bit menos significativo.

Si el Voltaje de referencia negativa es igual a cero, y el Voltaje de referencia positiva es el de alimentación al microcontrolador (5 [V]), la ecuación se reduce a:

$$1 \text{ LSB} = \frac{5 \text{ [V]}}{2^{12} [\text{bit}]} = 1.22070312 \left[\frac{\text{mV}}{\text{bit}} \right] \quad (3.9)$$

Se hace notar que el módulo de conversión analógico/digital sólo admite voltajes positivos. Si se emplea un voltaje negativo en la entrada del convertidor analógico/digital, este puede sufrir daños severos.

1.3 Microcontrolador PIC18F4553

El microcontrolador PIC18F4553 fabricado por Microchip Technologies, es un componente fundamental para este sistema. El microcontrolador realiza las

funciones de convertidor analógico/digital, provee los medios de control para la bomba hidráulica y la electroválvula, además que establece la comunicación por USB con una computadora, la cual despliega la información adquirida, y permite el control del sistema electrónico. A continuación, se presentan las especificaciones generales del microcontrolador, así como los módulos usados dentro del proyecto.

La información con respecto a la programación en C del microcontrolador, fue obtenida de la ayuda incluida en el compilador C de CCS.

1.3.1 Especificaciones Generales

De acuerdo a la hoja de datos del PIC18F4553 de Microchip (Microchip, sin fecha).

Características de USB

- Sigue las características de USB V2.0
- Modo de Baja Velocidad (1.5 Mb/s) y Velocidad completa (12 Mb/s)
- Soporta transferencias de control, interrupción, isócronas y de volumen.
- Soporta hasta 32 “endpoints” (16 bidireccionales)
- 1-kbyte de memoria de acceso aleatorio de doble acceso para USB
- Tranceptor USB integrado con regulador de voltaje
- Interfaz para tranceptor USB fuera del chip
- Puerto paralelo para transmisión

Modos de administración de energía

- Run: CPU encendido, periféricos encendidos
- Idle: CPU apagado, periféricos encendidos
- Sleep: CPU apagado, periféricos apagados
- Corriente en el modo Idle: hasta 5.8 μA típicamente
- Corriente en el modo Sleep: hasta 0.1 μA típicamente
- Oscilador del TIMER1: 1.1 μA típicamente, 32 kHz, 2 V
- Temporizador Watchdog: 2.1 μA típicamente
- Inicio de Oscilador en dos pasos

Características especiales del microcontrolador:

- Arquitectura optimizada para compilador en C, con un set de instrucciones extendido opcional.
- Memoria de programación de 100000 ciclos de Borrado/Escritura
- EEPROM como 1000000 ciclos de Borrado/Escritura
- Retención de datos en EEPROM: Superior a 40 años
- Autoprogramable bajo control de Software

- Niveles de prioridad para interrupciones
- Multiplicador por Hardware de un solo ciclo 8x8
- Temporizador Watchdog Extendido:
 - Periodo programable desde 41 ms hasta 131 s
- Protección de Código programable
- Programación serie integrada, alimentada con una sola fuente de 5 V, por dos pines.
- Depuración integrada
- Opera en un rango amplio de voltajes (2.0 a 5.5 [V])

Estructura de oscilador flexible:

- Cuatro modos de cristal, incluyendo un PLL de alta precisión para USB
- Dos modos de Reloj Externos, hasta 48 [MHz]
- Bloque de oscilación interno
 - Ocho frecuencias seleccionables, desde 31 [kHz] hasta 8 [MHz]
 - El usuario puede compensar por deriva en frecuencia
- Oscilador secundario utilizando el TIMER1 a 32 [kHz]
- Las opciones para Oscilador Dual permiten al microcontrolador y al módulo USB operar a dos velocidades de reloj diferentes.
- Monitor de Reloj a Prueba de Fallas:
 - Permite un apagado seguro si algún reloj se detiene

Características resaltantes de periféricos:

- Alta corriente para Sink/Source: 25 [mA] para ambas
- Tres interrupciones externas
- Cuatro módulos de tiempo (TIMER0 a TIMER3)
- Hasta dos módulos de Captura/Compara/PWM (CCP):
 - La Captura es de 16-bit, con resolución máxima de 5.2 [ns]
 - La Comparación es de 16-bit, con resolución máxima de 83.3 [ns]
 - Salida PWM: La resolución del PWM es de 1 a 10-bits.
- Módulo mejorado de CCP:
 - Múltiples modos de salida
 - Polaridad seleccionable
 - Tiempo muerto programable
 - Auto apagado y auto reinicio
- Módulo USART Mejorado:
 - Soporte al bus LIN

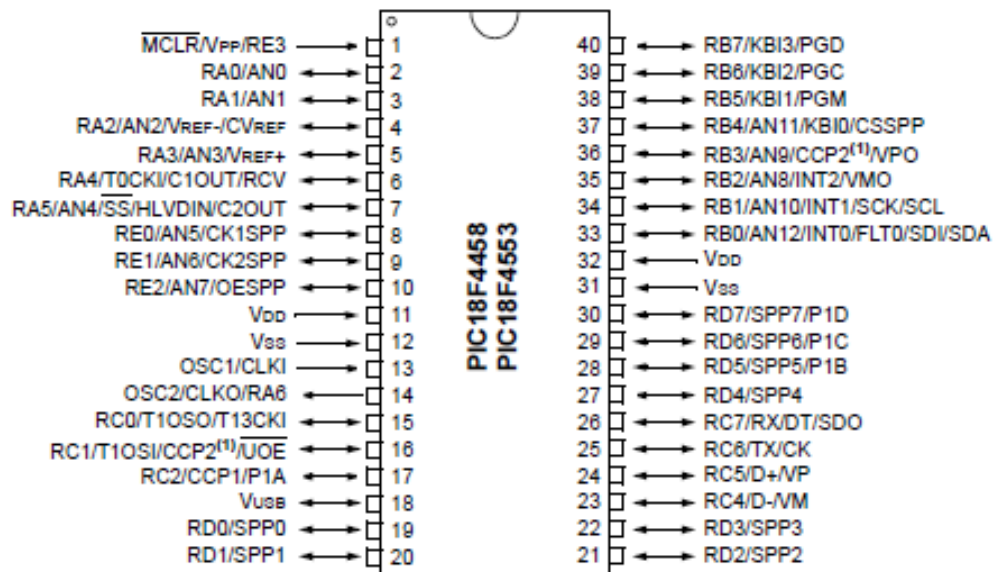
- Módulo de Puerto Serial Maestro Síncrono, soporta SPI de 3 alambres e I²C.
- Módulo de conversión Analógico/digital de 12-bits, hasta 13 canales con tiempo de adquisición programable
- Comparadores Duales Analógicos con conmutación de entradas.

Además, el dispositivo cuenta con 32 kilobytes de memoria Flash, una SRAM de 2048 bytes, una EEPROM de 256 bytes, 35 puertos de entrada/salida.

El desarrollo de software para este programa fue realizado en el ambiente de MPLAB versión 8.40, usando el compilador C de Custom Computer Services (CCS).

Se muestra un diagrama del PIC 18F4553, Figura 1.13 (Microchip, sin fecha), como referencia.

40-Pin PDIP



(1): RB3 es un pin alternativo para CCP2

Figura 1.13: Diagrama del PIC 18F4553.

Para establecer una base de tiempo adecuada para el uso de USB, se empleó un cristal de cuarzo cortado para resonancia en paralelo de 20 [MHz], y además, un Lazo de Seguimiento de Fase (Phase Locked Loop, PLL), para sintetizar una frecuencia de 48 [MHz].

1.3.2 Interrupciones

Como tal, una interrupción interrumpe la rutina ejecutada en el microcontrolador, estas pueden ser originadas de forma interna o externa. Cuando se produce una interrupción, el microcontrolador realizará un salto hacia la rutina donde se atiende dicha interrupción, la cual es definida por el programador. Una vez que se haya ejecutado esta rutina de interrupción, se regresa a la ejecución de la rutina principal, en la misma posición de memoria donde se produjo la interrupción.

Las interrupciones permiten elaborar programas que no requieran de un ciclo sólo para consultar algún suceso interno o externo (polling), las cuales provocan retardos en la ejecución del programa.

Para el PIC18F4553, existen 20 fuentes de interrupción: (Entradas digitales, temporizadores, CCP, Convertidores Analógico/digital).

Interrupciones en C

Las etiquetas para activar interrupciones en el CCS son:

- GLOBAL
- INT_RTCC – Desbordamiento del TIMER0
- INT_TIMER0 - Desbordamiento del TIMER0
- INT_TIMER1 - Desbordamiento del TIMER1
- INT_TIMER2 - Desbordamiento del TIMER2
- INT_TIMER3 - Desbordamiento del TIMER3
- INT_EXT – Interrupción externa en RB0
- INT_EXT1 – Interrupción externa 1
- INT_EXT2 – Interrupción externa 2
- INT_RB – Cambio de estado en los pines B4 a B7
- INT_AD – Conversión analógico/digital finalizada
- INT_RDA – Dato RS232 recibido
- INT_TBE – Buffer de transmisión RS232 vacío
- INT_SSP – Actividad en SPI o I²C
- INT_CCP1 – Unidad 1 de Captura, Comparación y PWM
- INT_CCP2 – Unidad 2 de Captura, Comparación y PWM
- INT_BUSCOL – Colisión de bus
- INT_LOWVOLT – Bajo voltaje detectado
- INT_COMP - Comparador
- INT_EEPROM – Escritura EEPROM finalizada
- INT_OSCF – Fallo en oscilador
- INT_SPP – Streaming Parallel Port

- INT_USB – Actividad en USB

El compilador de CCS permite emplear funciones especiales para el manejo de las directivas de interrupción. El comando **enable_interrupts(ID)**; configura los registros pertinentes para el uso de interrupciones. Las interrupciones (ID) que pueden ser activadas de esta forma son:

- GLOBAL
- INT_RTCC o INT_TIMER0
- INT_EXT
- INT_RB
- INT_AD
- INT_RDA
- INT_TBE
- INT_SSP
- INT_CCP1
- INT_TIMER2
- INT_TIMER1
- INT_EEPROM
- INT_BUSCOL
- INT_CCP2

La directiva GLOBAL controla la habilitación de interrupciones de forma global, además de controlar las interrupciones de periféricos, y esta debe activarse de forma independiente. Cada directiva activará los registros correspondientes.

Para desactivar las interrupciones, se emplea el comando **disable_interrupts(ID)**;

En el proyecto, se emplearon las interrupciones correspondientes al TIMER0, TIMER1 y al CCP. Se hablará sobre el uso de sus interrupciones en los apartados correspondientes.

1.3.3 Temporizadores

Los temporizadores, o “timers” son módulos que contabilizan pulsos de forma interna o externa. Cuando la cuenta es interna, se habla de temporización; si es externa, de contador.

TIMER0

El bloque TIMER0/WATCHDOG en el PIC18F4553 es un contador configurable por software de 8-bits o de 16-bits, que incrementa por hardware y es programable. La cuenta máxima es de 255 (0xFF) o 65535 (0xFFFF). La frecuencia de conteo es una cuarta parte de la frecuencia del reloj principal ($f_{osc}/4$). Si se requiere una

frecuencia menor, se puede emplear un divisor programable de frecuencia, o prescalador, en factores de 2, 4, 8, 16, 32, 64, 128 o 256.

Este bloque puede funcionar como WATCHDOG, lo que permite que durante el funcionamiento normal del microcontrolador, si hay un desbordamiento (timeout) del WATCHDOG, provoque un RESET al microcontrolador. Para prevenir esto, se tiene que restablecer la cuenta del WATCHDOG a cero con ayuda de CLRWDT. Si no se requiere la función de WATCHDOG, se configura el “fusible” (opciones para la operación del microcontrolador) NOWTD, de la siguiente forma: **#FUSE NOWTD**

El WATCHDOG TIMER está con base en un oscilador RC interno, independiente del oscilador principal del microcontrolador. Incluso el WATCHDOG TIMER detecta si el reloj conectado a OSC1/CLK1 o a OSC2/CLK0 está detenido.

Para determinar el tiempo de desbordamiento del TIMER0, se emplea la ecuación (1.10). (García, 2008)

$$T = \frac{4}{f_{osc}} * Prescalador * (65536 - Carga en TIMER0) \quad (3.10)$$

TIMER0 en C

Para configurar el TIMER0 en C, se emplea el siguiente comando:

setup_timer_0(opciones);

Donde las opciones son definidas en el archivo 18F4553.h del compilador C de CCS. Estas opciones son:

- RTCC_INTERNAL – Reloj Interno
- RTCC_EXT_L_TO_H – Cuenta cuando un pulso externo cambia de un nivel bajo a uno alto
- RTCC_EXT_H_TO_L - Cuenta cuando un pulso externo cambia de un nivel alto a uno bajo
- RTCC_DIV_1 – División de frecuencia entre 1
- RTCC_DIV_2 – División de frecuencia entre 2
- RTCC_DIV_4 – División de frecuencia entre 4
- RTCC_DIV_8 – División de frecuencia entre 8
- RTCC_DIV_16 – División de frecuencia entre 16
- RTCC_DIV_32 – División de frecuencia entre 32
- RTCC_DIV_64 – División de frecuencia entre 64
- RTCC_DIV_128 – División de frecuencia entre 128
- RTCC_DIV_256 – División de frecuencia entre 256

Para cargar un valor en el registro del contador, se usa:

set_timer0(valor);

Para obtener el valor actual del contador:

Valor = get_timer0();

TIMER1

Este módulo temporizador trabaja de forma independiente al TIMER0, cuenta con las siguientes características:

- Trabaja a 16-bit (está compuesto por dos registros de 8-bits cada uno).
- Permite lectura y escritura
- Desencadena una interrupción por desbordamiento de 0xFFFF a 0x0000
- Reinicio por disparo del módulo de Captura, Comparación y PWM.

El tiempo de desbordamiento se calcula de forma muy similar al TIMER0:

$$T = \frac{4}{f_{osc}} * Prescalador * (65536 - Carga en TIMER1) \quad (3.11)$$

TIMER1 en C

Para configurar este módulo, se emplea el comando:

setup_timer_1(opciones);

Donde estas opciones afectan la operación del TIMER1. Las opciones posibles son las siguientes:

- T1_DISABLED – Desactiva el contador
- T1_INTERNAL – Usa oscilador interno
- T1_EXTERNAL – Cuenta pulsos externos
- T1_EXTERNAL_SYNC – Sincroniza la entrada con reloj externo.
- T1_CLK_OUT – Habilita el oscilador del TIMER1
- T1_DIV_BY_1 – División en 1
- T1_DIV_BY_2 – División en 2
- T1_DIV_BY_4 – División en 4
- T1_DIV_BY_8 – División en 8

Para obtener y fijar valores en el contador, se sigue de una forma muy similar al TIMER0.

Para cargar un valor al contador:

set_timer1(Valor);

Para obtener un valor del contador:

Valor = get_timer0;

Donde “Valor” es un número entero de 16-bit

1.3.4 Módulo de Comparación, Captura y PWM (CCP)

Los módulos CCP permiten realizar tres funciones con base en el manejo de los temporizadores:

- **Modo de Comparación:** Realiza la comparación entre dos registros, uno de estos es el registro del TIMER1, y otro cualquiera. Esta función permite, además, disparar un evento externo cuando haya pasado una cantidad de tiempo predeterminada.
- **Modo de Captura:** Permite la medición de la duración de un evento. Esta función permite al usuario obtener información del estado actual de un registro que cambia constantemente. En este caso, es el registro correspondiente al TIMER1.
- **PWM (Pulse Width Modulation, Modulación de Ancho de Pulso):** Genera señales de frecuencia y ciclo de trabajo variables.

En este sistema, sólo se empleó el Modo de Captura, el que será descrito a continuación.

Modo de Captura

En este modo, el registro del módulo TIMER1 es copiado al registro del CCP1 en las siguientes situaciones:

- Cada flanco de bajada en el pin RC2/CCP1
- Cada flanco de subida en el pin RC2/CCP1
- Cada cuarto flanco de subida en el pin RC2/CCP1
- Cada decimosexto flanco de subida en el pin RC2/CCP1

Además, para que el módulo CCP1 trabaje de esta forma, se requiere que:

- El pin RC2/CCP1 esté configurado como entrada
- El módulo TIMER1 trabaje como temporizador o contador síncrono

Una vez que se registra una captura, se dispara una interrupción del módulo.

CCP en C

Para configurar el módulo CCP en el compilador C de CCS, se emplean los siguientes comandos:

setup_ccp1(modulo);

Donde **modulo** hace referencia a los registros de control del módulo. El listado se presenta en la Tabla 1.1.1

Modo	Función
CCP_OFF	Deshabilitación
CCP_CAPTURE_FE	Captura por flanco de bajada
CCP_CAPTURE_RE	Captura por flanco de subida
CCP_CAPTURE_DIV_4	Captura tras 4 pulsos
CCP_CAPTURE_DIV_16	Captura tras 16 pulsos
CCP_COMPARE_SET_ON_MATCH	Salida en alto en comparación
CCP_COMPARE_CLR_ON_MATCH	Salida en bajo en comparación
CCP_COMPARE_INT	Interrupción en comparación
CCP_COMPARE_RESET_TIMER	Reinicia el temporizador en comparación
CCP_PWM	Habilita PWM

Tabla 1.1: Modos para el módulo CCP.

1.3.5 Convertidor Analógico/Digital

El módulo de Convertidor Analógico/Digital del PIC18F4553 tiene 13 pines para los dispositivos de 40 pines. Éste módulo consiste en un Sample and Hold y un convertidor analógico/digital de aproximaciones sucesivas de 12-bits.

Convertidor Analógico/Digital en C

Para configurar el módulo de conversión analógica/digital en el compilador de C de CCS, se emplean los siguientes comandos:

setup_adc(modos);

Donde **modos** son los valores de configuración correspondientes al registro de control del módulo. Los modos son:

- ADC_OFF – Apaga el convertidor
- ADC_CLOCK_INTERNAL – Usa una base de tiempo interna
- ADC_CLOCK_DIV_2 – Divide la frecuencia del reloj entre 2
- ADC_CLOCK_DIV_8 – Divide la frecuencia del reloj entre 8
- ADC_CLOCK_DIV_32 – Divide la frecuencia del reloj entre 32
- ADC_CLOCK_DIV_64 – Divide la frecuencia del reloj entre 64

setup_adc_ports(Valor);

Hace que los pines del convertidor analógico/digital sean analógicos, digitales, o alguna combinación de estos, además de fijar la referencia a usar cuando se compute el valor de la conversión. Las combinaciones de pines analógicos permitidos dependen del chip. Para el PIC18F4553, éstas son:

Valor	Pines a activar como entradas analógicas
NO_ANALOGS	Ninguno
ALL_ANALOG	A0, A1, A2, A3, A5, E0, E1, E2, B2, B3, B1, B4 y B0
AN0_TO_AN11	A0, A1, A2, A3, A5, E0, E1, E2, B2, B3, B1 y B4
AN0_TO_AN10	A0, A1, A2, A3, A5, E0, E1, E2, B2, B3 y B1
AN0_TO_AN9	A0, A1, A2, A3, A5, E0, E1, E2, B2 y B3
AN0_TO_AN8	A0, A1, A2, A3, A5, E0, E1, E2 y B2
AN0_TO_AN7	A0, A1, A2, A3, A5, E0, E1 y E2
AN0_TO_AN6	A0, A1, A2, A3, A5, E0 y E1
AN0_TO_AN5	A0, A1, A2, A3, A5 y E0
AN0_TO_AN4	A0, A1, A2, A3 y A5
AN0_TO_AN3	A0, A1, A2 y A3
AN0_TO_AN2	A0, A1 y A2
AN0_TO_AN1	A0 y A1
AN0	A0

Tabla 1.2: Configuraciones permitidas para el ADC del PIC18F4553.

set_adc_channel(canal);

Donde **canal** establece que canal del convertidor analógico/digital se ha de usar para la próxima lectura del convertidor. Hay que mencionar que se tiene que dejar un lapso de tiempo después de hacer el cambio de canal para obtener lecturas válidas del convertidor analógico/digital. No es necesario cambiar el canal antes de cada lectura, si el canal a leer no cambia. Los canales están numerados de 0 a 11, y se denotan como AN0, AN1, ..., AN11.

valor = read_adc(modos);

El parámetro **modos** es de carácter opcional, los valores válidos son:

- ADC_START_AND_READ – Toma lectura de forma continua, es el modo predeterminado.
- ADC_START_ONLY – Comienza la conversión y regresa
- ADC_READ_ONLY – Lee el resultado de la última conversión

El rango de valores posibles en la conversión depende del número de bits en el convertidor analógico/digital y en la directiva #DEVICE ADC = (número de bits a usar)

1.3.6 Universal Serial Bus

El Bus Serie Universal (Universal Serial Bus), fue desarrollado en la década de los 90 con la idea de establecer el concepto *plug-and-play*, lo que permitiría conectar una gran gama de dispositivos a la computadora sin necesidad de reiniciar el equipo, y además, de que estos se configuran de forma automática. Además de poseer una

forma de establecer la comunicación entre un dispositivo y la computadora, el USB permite que el dispositivo se energice desde la computadora.

Éste bus tiene una estructura de árbol, con capacidad de hasta 127 dispositivos conectados a un mismo anfitrión (host), permitiendo transferencias síncronas y asíncronas.

El anfitrión es el equipo que inicia la comunicación hacia un dispositivo, o a un hub. El protocolo consiste en que el anfitrión envía un testigo al dispositivo, y éste le responde regresándole dicho testigo.

De forma física, la información del USB se transmite mediante un par trenzado (D+ y D-), además de que van acompañados por una línea de alimentación a 5 [V] y la referencia. (García, 2008)

USB en C

El compilador de C de CCS provee varios archivos de cabecera que permiten conectar el PIC con la computadora mediante USB.

Los archivos de cabecera son:

- `pic_18usb.h` – Capa de hardware para el driver USB de CCS
- `usb.h` – Maneja a los testigos USB y obtiene reportes de descripción
- `usb_desc_scope.h` – Configura al USB y los descriptores de ese dispositivo

Es importante que el *VENDOR ID* sea 0x04D8, que corresponde a Microchip, y que el *PRODUCT ID* concuerde con alguno listado en el archivo **mchpush.inf**, el cual se encuentra en la carpeta del driver de genérico de USB de Microchip, el cual puede obtenerse directamente de su sitio de internet. En caso contrario, el dispositivo no será reconocido por la computadora.

El compilador de C de CCS provee varias funciones para trabajar con USB:

usb_init();

Inicializa el hardware de USB. Posteriormente esperará en un ciclo infinito a que el periférico USB sea conectado a un bus (lo cual no significa que tenga que ser numerado por la computadora). Activará y empleará la interrupción de USB.

usb_task();

Esta función prepara al periférico cuando el PIC sea conectado al bus. Si el PIC es desconectado del bus, reiniciará la pila USB y al periférico. Activará y empleará la interrupción de USB.

usb_enumerated();

Regresa TRUE si el dispositivo ha sido enumerado por la computadora. Si el dispositivo ha sido numerado, quiere decir que opera en modo normal y que uno puede enviar y recibir paquetes.

usb_put_packet(endpoint, datos, longitud, conmutación);

Coloca un paquete de datos en el buffer del endpoint especificado. Regresa TRUE si tiene éxito, FALSE si en buffer sigue lleno con el último paquete.

usb_get_packet(endpoint, apuntador, número de bytes máximo);

Lee hasta un número máximo de bytes del buffer de un endpoint especificado y los guarda donde indique el apuntador. Regresa el número de bytes guardados en el apuntador especificado.

1.4 Programación en la MFC de Visual C++

La librería de Microsoft Foundation Class (MFC) provee con un envoltorio orientado a objetos para casi todas las API (Interfaz de Programación de Aplicaciones) de Win32 y COM. Aunque pueda crear aplicaciones muy sencillas de escritorio, es más útil cuando se necesita desarrollar interfaces de usuario más complejas, con controles múltiples. Se puede utilizar al MFC para crear aplicaciones con interfaces de usuario semejantes a las vistas en Microsoft Office.

La referencia de MFC encontrada en línea, desde la página de Microsoft Developer Network, cubre las clases, funciones globales, variables globales y macros que componen a la librería MFC.

La librería de MFC provee mucho del código necesario para manejar ventanas, menús, y cajas de diálogo; realiza operaciones básicas de entrada y salida; almacena colecciones de objetos de datos; y más. Lo único que se necesita es agregar el código específico a cada aplicación en este Framework (marco de trabajo). Debido a la naturaleza propia de la programación de clases en C++, es fácil extender o sustituir la funcionalidad básica que la librería MFC provee.

El Framework de MFC es una forma poderosa que permite al usuario construir a partir del trabajo de programadores expertos para Windows. MFC acorta el tiempo de desarrollo; hace el código más portable; provee un gran apoyo sin reducir la libertad y flexibilidad de programación; y da un fácil acceso a elementos “difíciles de programar” en las interfaces de usuario, así como en tecnologías como Active, OLE y programación para internet. Además de que la librería MFC simplifica la programación de bases de datos a través de los Data Access Objects (DAO) y Open Database Connectivity (ODBC), y la programación de redes, a través de los Windows Sockets. MFC hace sencilla la programación de características como hojas de propiedades, vista previa de impresión, y barras de herramientas flotantes y personalizables. (Kruglinski, Wingo y Sheperd, 1998)

Para trabajar con esta librería, se usa el entorno de desarrollo de Visual Studio 2012. Para crear un programa bajo esta estructura, se tiene que elegir la opción “Nuevo Proyecto”, del menú principal (Figura 1.14), o en el menú superior (Archivo > Nuevo Proyecto). Una vez que se haya seleccionado esa opción, aparecerá un

diálogo como se ve en Figura 1.15, donde se muestran múltiples opciones para crear un programa nuevo. Al dirigirse hacia Visual C++ con la librería MFC, es posible crear una aplicación con base en esta librería, así como una Librería Vinculada Dinámicamente (DLL) o un control ActiveX, como se ve en la Figura 1.15.

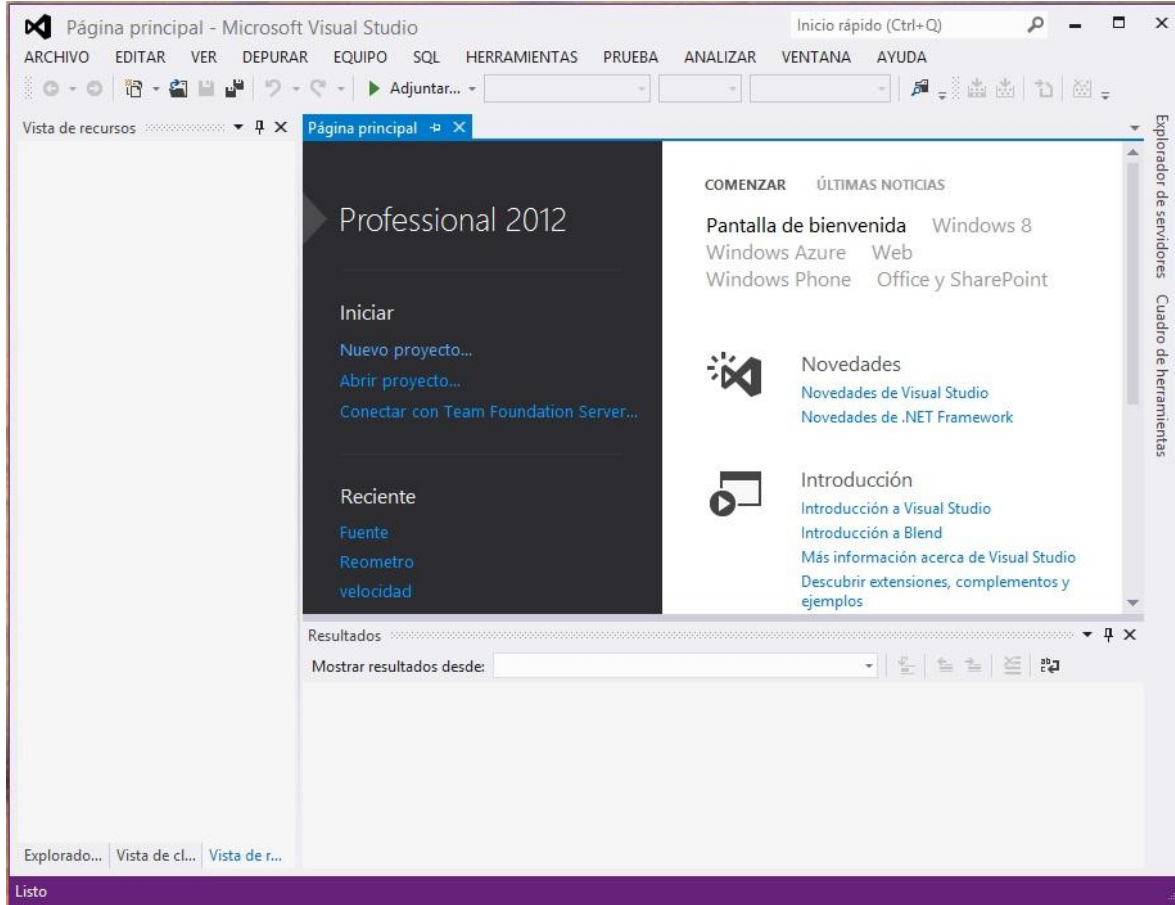


Figura 1.14: Ventana principal de Microsoft Visual Studio 2012.

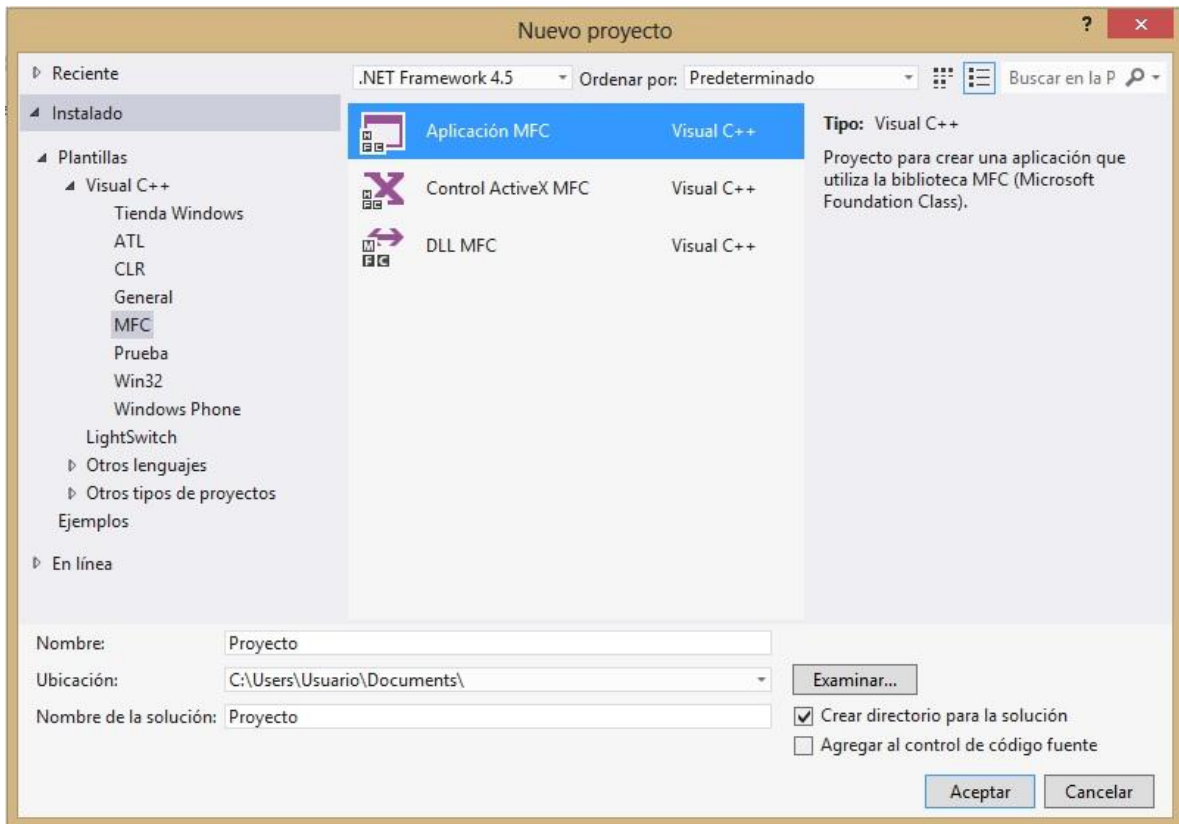


Figura 1.15: Menú para nuevo proyecto. Se visualizan las opciones basadas en la librería MFC.

Una vez que se ha creado un proyecto para una aplicación con base en MFC, se puede configurar el tipo de aplicación a desarrollar. Esta puede ser de Documento Único, Varios Documentos, o con base en Cuadros de Diálogo, además de poder elegir el lenguaje a emplear (Figura 1.16), como así como ciertas características de la interfaz de usuario (Cuadros de maximizar y minimizar, grueso del marco, menú de sistema), el nombre del diálogo, algunas opciones avanzadas (Controles ActiveX, Manifiesto de controles comunes), y las clases generadas (Figura 1.17).

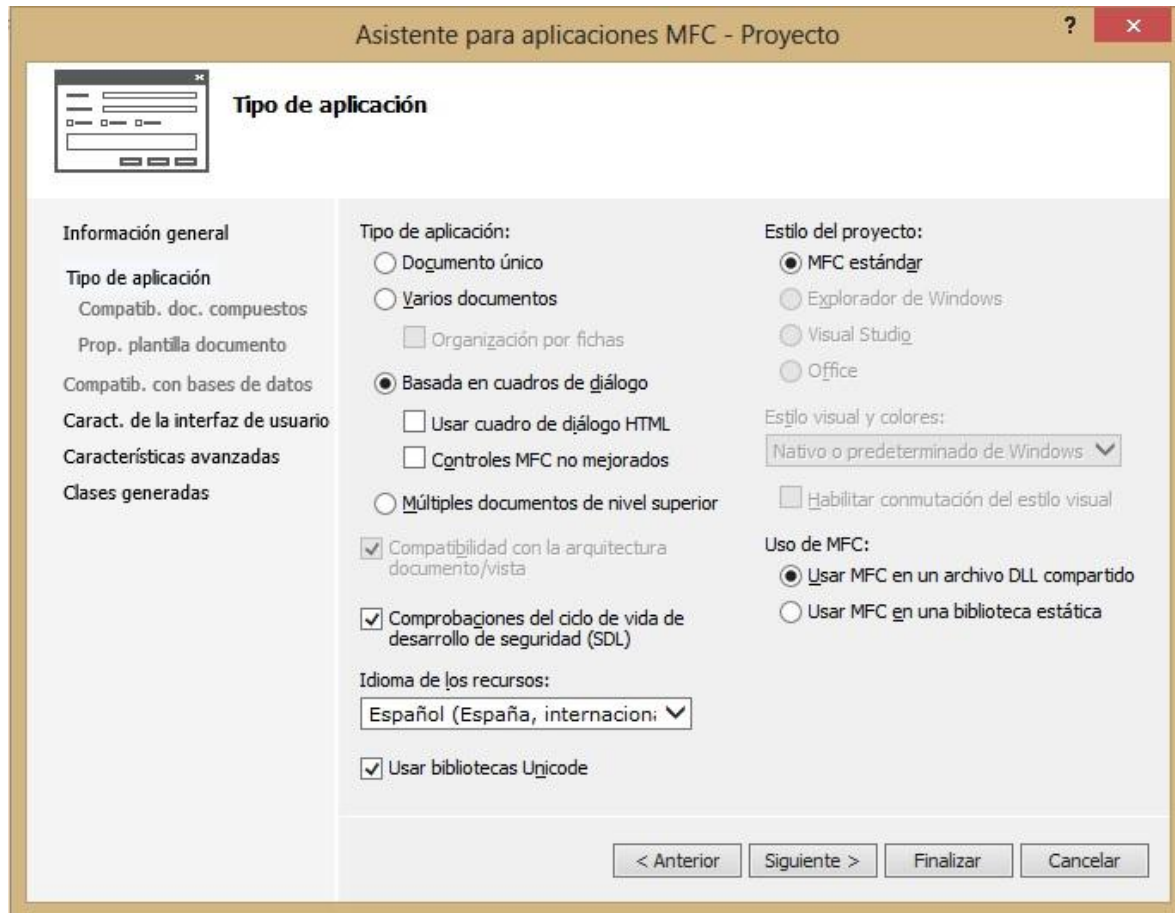


Figura 1.16: Configuración para una aplicación basada en diálogo.

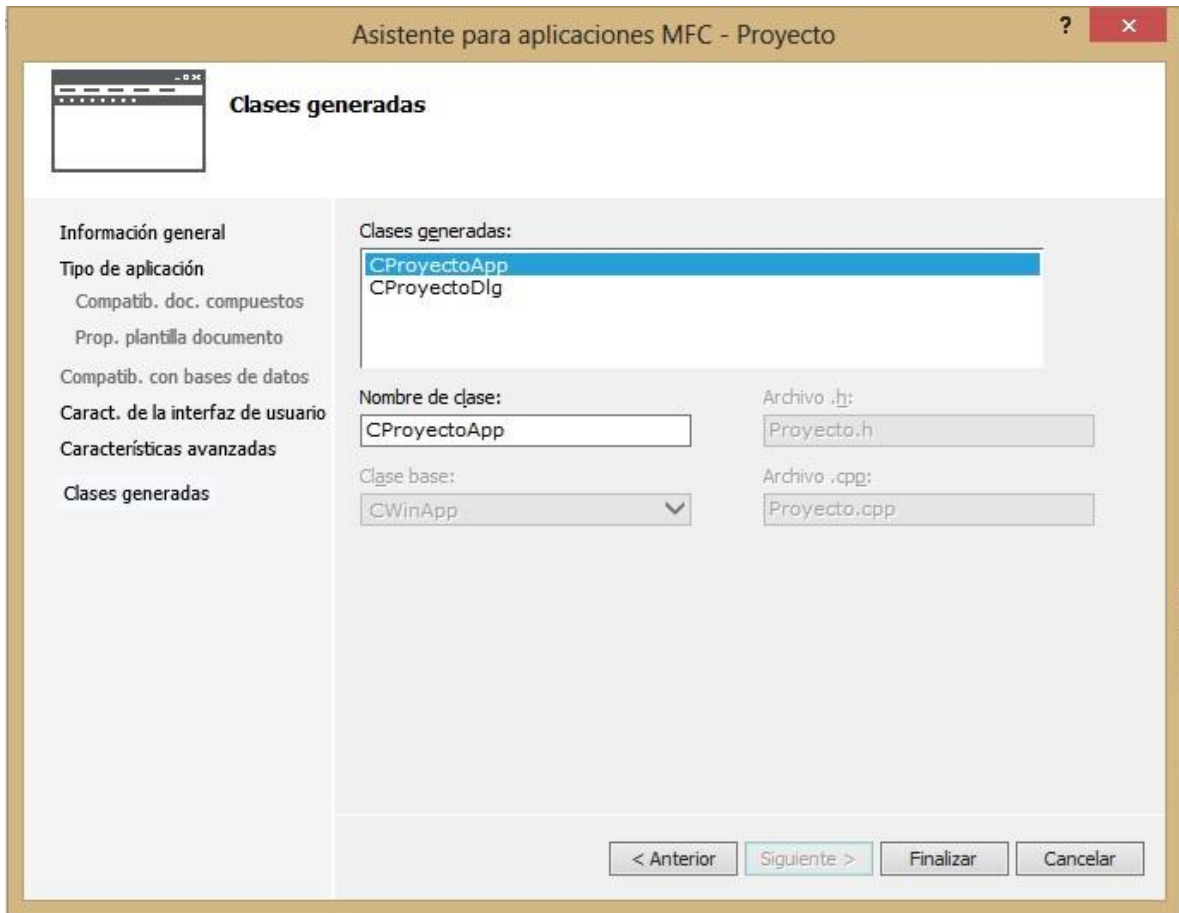


Figura 1.17: Clases generadas en el proyecto.

Una vez ocupado el asistente para la generación de un nuevo proyecto, Visual Studio muestra el esqueleto del cuadro de diálogo, en la cual se pueden agregar los controles que aparecen en la lista del lado izquierdo de la ventana principal. En el lado derecho, es posible explorar los archivos fuente y de cabecera que el asistente ha creado. Cada archivo con extensión .cpp es una clase dentro de Visual C++ 2012. Cabe mencionar que lo único que hace el asistente es crear un esqueleto del programa, para que el desarrollador comience a trabajar a partir de él.

Capítulo 2

Descripción del proyecto

Una vez sentadas las bases para el desarrollo del sistema, corresponde profundizar en su funcionamiento. A grandes rasgos, el proyecto consiste en un circuito electrónico que permite el acondicionamiento y adquisición de datos, como el control de una electroválvula y una bomba hidráulica; además de los algoritmos necesarios para el funcionamiento del sistema, como su visualización y control en una computadora usando el sistema operativo Microsoft Windows.

2.1 Esquema general

La Figura 2.2 muestra el sistema desarrollado. Se tiene un sistema electrónico que registra datos de los transductores de presión, flujo y posición; controla el funcionamiento de una bomba hidráulica y de una electroválvula. Además, se reciben y envían datos hacia una computadora, la cual cuenta con un programa elaborado específicamente para éste proyecto de tesis.

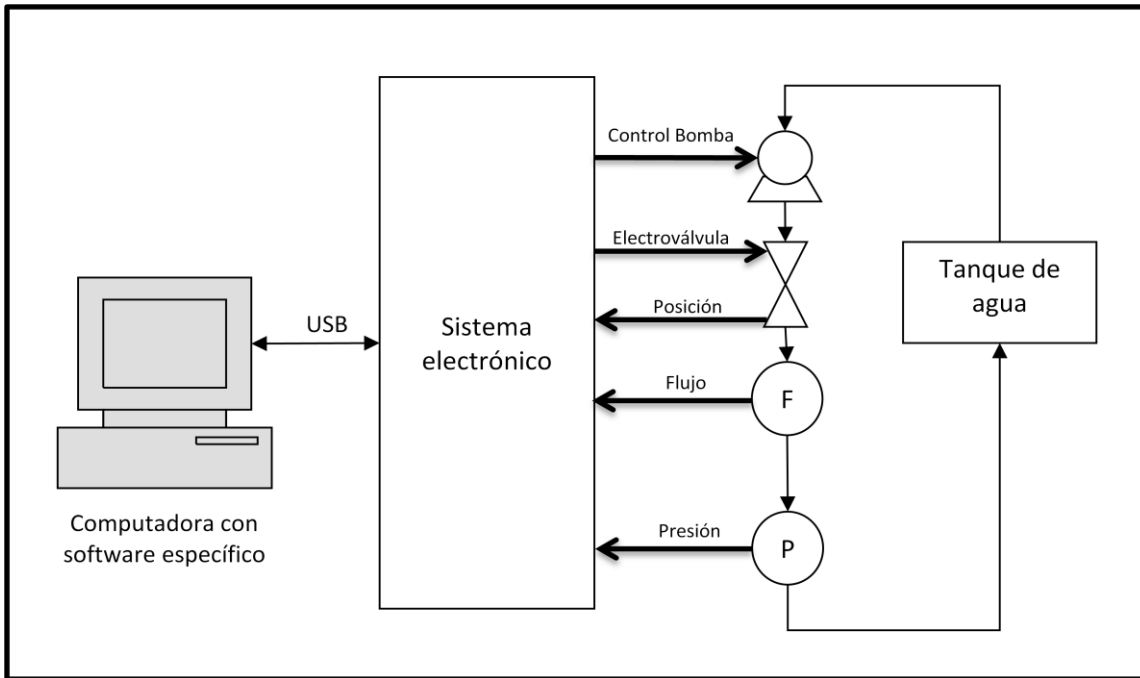


Figura 2.2: Esquema general del sistema propuesto.

2.2 Sistema electrónico

En esta sección se describen los detalles del circuito electrónico presentado en la Figura 2.2 con el fin de mostrar su funcionamiento a detalle. En la Figura 2.3 se muestra un diagrama a bloques general del sistema electrónico.

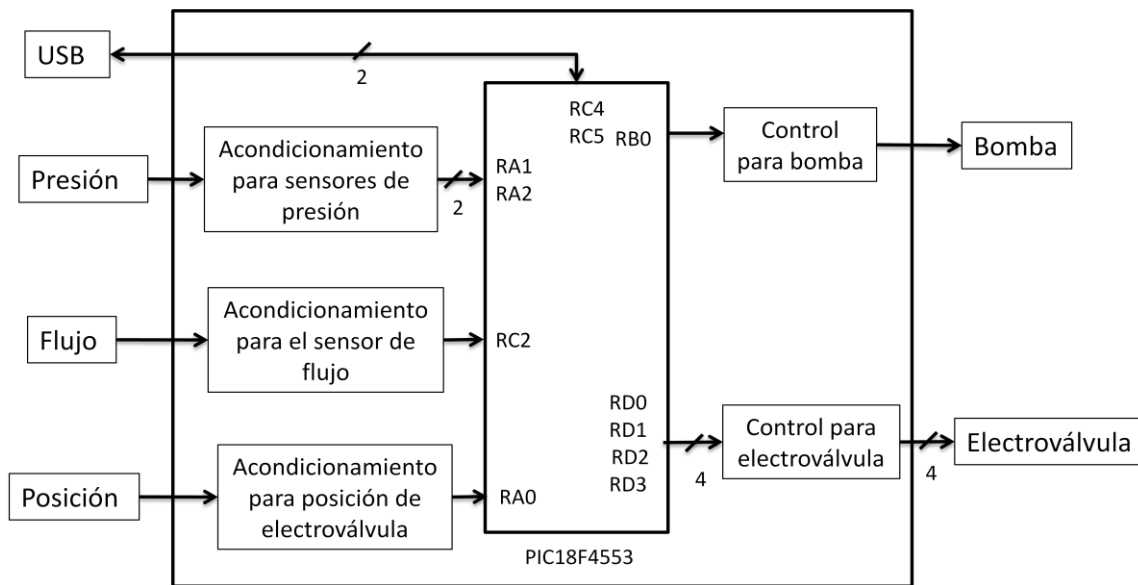


Figura 2.3: Esquema general del sistema electrónico.

Para consultar los diagramas esquemáticos y circuitos impresos de la implementación, refiérase a los anexos A y B. A continuación, una descripción funcional de los diversos subsistemas del circuito electrónico desarrollado.

2.2.1 Acondicionamiento de señales

Para procesar las señales provenientes de los sensores empleados es necesario acondicionar dichas señales para que satisfagan los requerimientos de la etapa de conversión analógico/digital. En los siguientes apartados se tratará sobre las etapas de acondicionamiento usadas en el sistema.

Acondicionamiento para los sensores de presión

Se emplearon transductores de presión modelo PX26-015DV (15 [psid]) y PX26-030DV (30 [psid]) fabricados por Omega. Dichos sensores son dispositivos activos con cuatro elementos piezorresistivos en un arreglo de puente de Wheatstone. De acuerdo al fabricante, estos sensores poseen una sensibilidad de 10 [mV/V]; una no linealidad de 1% en la escala completa y una desviación en la repetitividad del 0.2%. (Omega Engineering Inc., sin fecha)

El módulo de conversión analógico/digital del PIC18F4553 acepta señales entre 0 y 5 [V], por lo que el voltaje proporcional a la presión diferencial tiene que estar entre esos límites. La Figura 2.4 muestra el circuito básico para acondicionar dicha señal.

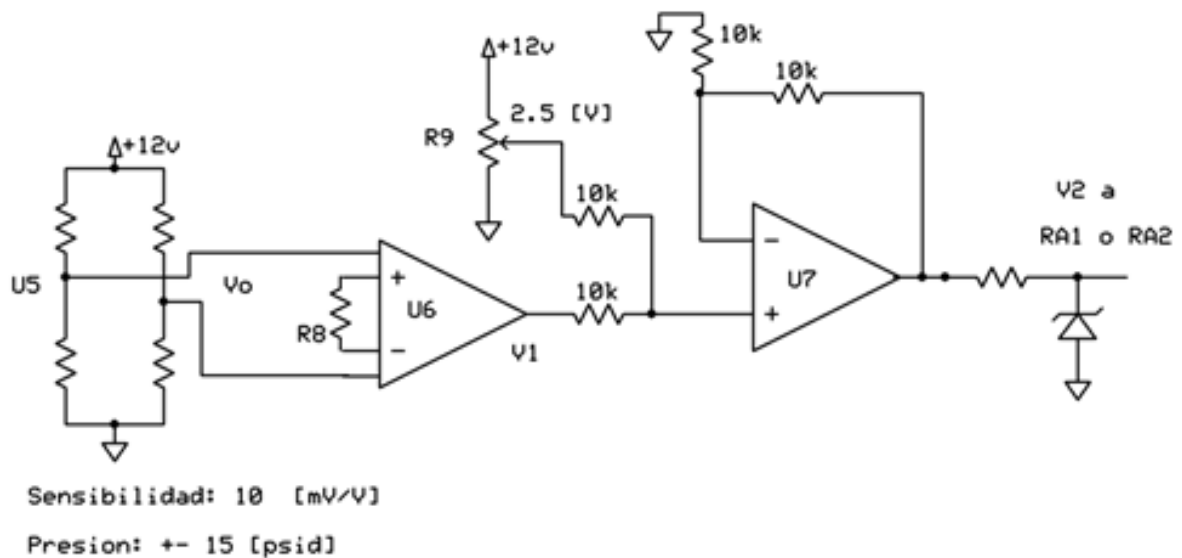


Figura 2.4: Acondicionamiento de sensor de presión.

El funcionamiento de ambos sensores es similar: la única diferencia es la presión diferencial a la que pueden ser sometidos, por lo que se usa el mismo circuito para cada uno de ellos.

De la Figura 2.4, U5 representa el sensor de presión PX26-015DV del cual se obtiene la señal diferencial V_0 .

El voltaje V_o máximo está dado por la ecuación (2.1)

$$V_{o_{\text{máximo}}} = \text{Voltaje de polarización} * \text{Sensibilidad} \quad (3.12)$$

Como se aprecia en la Figura 2.4, el voltaje de polarización del sensor es 12 [V] por lo que:

$$V_o = 12 [V] * 10 \left[\frac{mV}{V} \right] = 120 [mV] \quad (3.13)$$

Por lo tanto, $V_o = \pm 120 [mV]$ para ± 15 [psid]

Para realizar el acondicionamiento se amplifica a V_o con ayuda de U6, el cual es un amplificador de instrumentación y se agrega un offset de 2.5 [V] usando a U7 como amplificador sumador. Este proceso se visualiza en la Figura 2.5, donde se grafica el voltaje a la salida de U6 con respecto a la presión aplicada y en la Figura 2.6 donde se grafica el voltaje a la salida de U7 con respecto al voltaje a la salida de U6.

La ecuación (2.3) corresponde a la ganancia del amplificador de instrumentación AD620 (Analog Devices, sin fecha).

$$\frac{V_1}{V_o} = \frac{49.4 \text{ k}\Omega}{R_8} + 1 \quad (2.1)$$

Si $V_o = \pm 0.120$ [V] $V_1 = \pm 2.5$ [V]

$$R_8 = \frac{49.4 \text{ k}\Omega}{\frac{V_1}{V_o} - 1} = 2.49 [k\Omega] \quad (2.2)$$

Para obtener el comportamiento visto en la Figura 2.6, se emplea un amplificador sumador no inversor. Como los resistores usados son iguales, la expresión para la señal de salida se reduce a la suma de las señales de entrada, por lo que, el voltaje V_2 es:

$$V_2 = 2.5 + V_1 \quad (2.3)$$

Finalmente, se emplea un diodo Zener para garantizar que el nivel de voltaje proporcional a la presión diferencial ejercida en el sensor no exceda los 5 [V] del canal de entrada al microcontrolador.

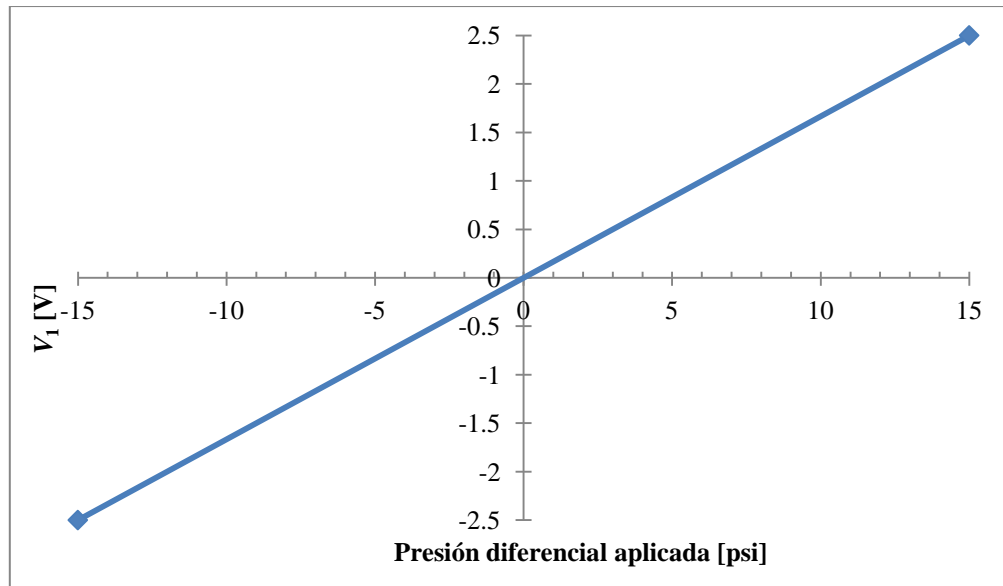


Figura 2.5: Señal amplificada con respecto a la presión aplicada.

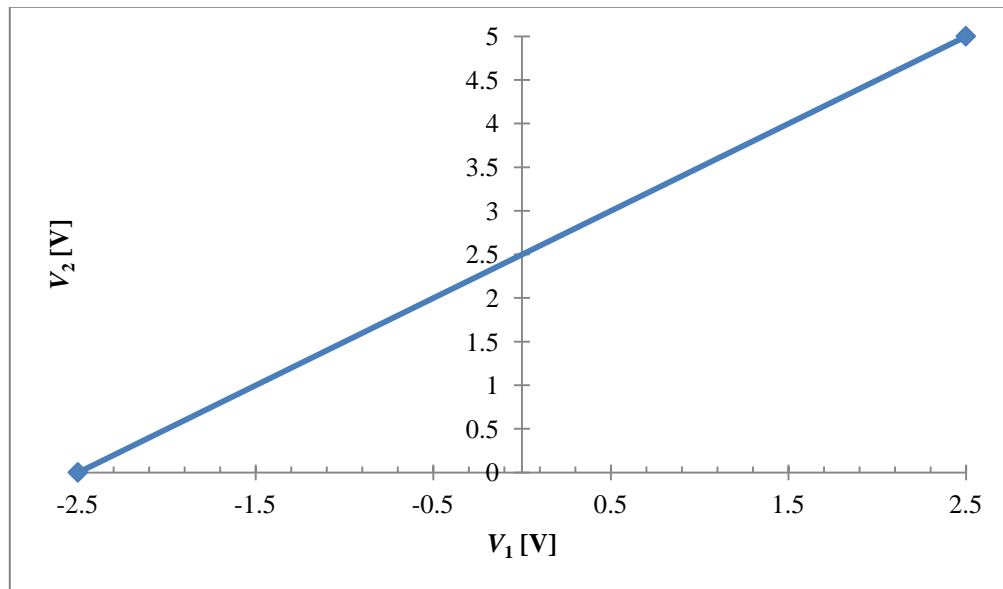


Figura 2.6: Salida del amplificador sumador.

Acondicionamiento para el sensor de Flujo

El sensor de flujo empleado fue el FP-5300 de Omega (Omega Engineering Inc., sin fecha). Éste es un sensor de flujo a paletas que puede registrar desde 0.3 hasta 6 [m/s]. Posee una no linealidad de $\pm 1\%$ en la escala completa, así como una desviación en la repetitividad de $\pm 0.5\%$ en la escala completa. La salida de este dispositivo tiene una frecuencia de 19.7 [Hz] por cada [m/s] y una amplitud de 3.3 [V_{pp}] por cada [m/s].

El parámetro de interés es la frecuencia de la señal generada por el sensor de flujo. Para poder contabilizar esta frecuencia en el módulo CCP del

microcontrolador, se requiere convertir la señal sinusoidal del sensor a una señal cuadrada entre 0 y 5 [V], tal como se muestra en la Figura 2.7.

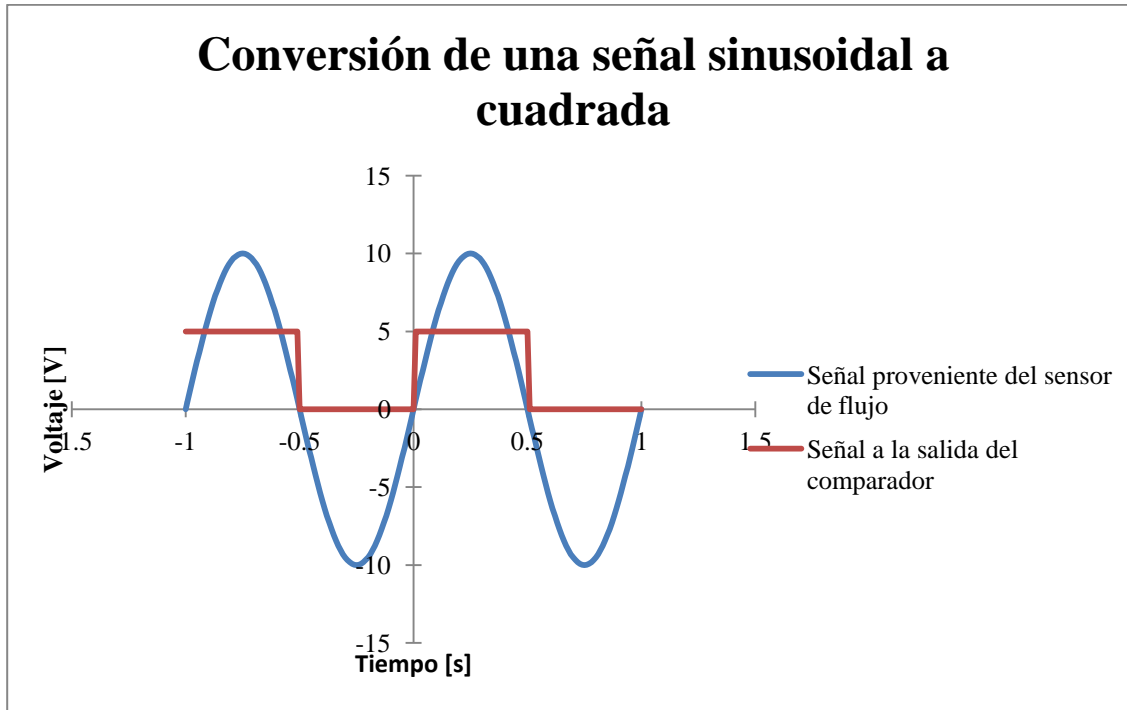


Figura 2.7: Conversión de una señal sinusoidal a una señal TTL.

Para obtener este comportamiento se empleó el circuito mostrado en la Figura 2.8

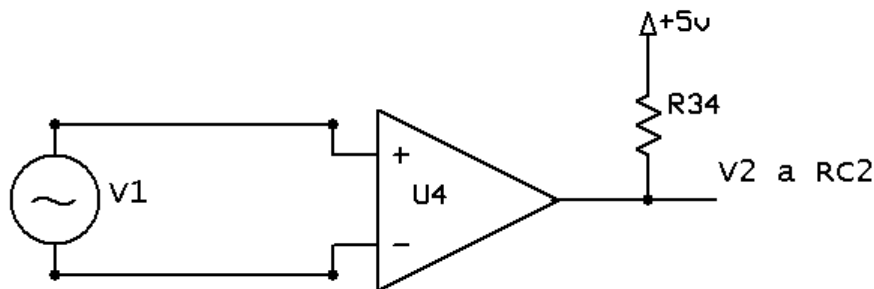


Figura 2.8: Acondicionamiento para el sensor de flujo.

El comparador empleado es un LM311. El resistor R34 es usado como pull-up, debido a que la salida de este comparador es de colector abierto.

Acondicionamiento de posición para Electroválvula

Para obtener la posición de apertura de la electroválvula se empleó un potenciómetro fijado a la flecha del motor de ésta. Este potenciómetro se muestra en

la Figura 2.9. La polarización de 0 a 5 [V] permite que el voltaje V_1 obtenido del cursor del potenciómetro esté en el rango del canal de conversión analógica a digital del microcontrolador, como se muestra en la Tabla 2.3.

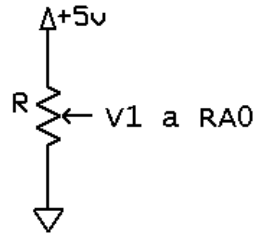


Figura 2.9: Circuito de acondicionamiento para posición de electroválvula.

Apertura de la electroválvula [%]	Resistencia [kΩ]	V_1 [V]	Valor registrado por el ADC
0	15.015	3.7537	0xC03
100	1.3712	1.3428	0x44C

Tabla 2.3: Datos del transductor de posición de la electroválvula.

2.2.2 Control de bomba hidráulica

Para el control de la bomba hidráulica es necesario emplear un relevador. Para esto, se empleó un optoacoplador con salida a transistor NPN, para aislar al relevador del microcontrolador y un relevador de tipo normalmente abierto. El relevador empleado fue un G5LE-1A4 (Omron, sin fecha).

En la Figura 2.10 se considera a V_1 en un estado bajo (L) o de alta impedancia (Z). Esto hace que el transistor de U8 y el transistor Q1 se encuentren en la región de corte, por lo que no habrá corriente en la bobina del relevador.

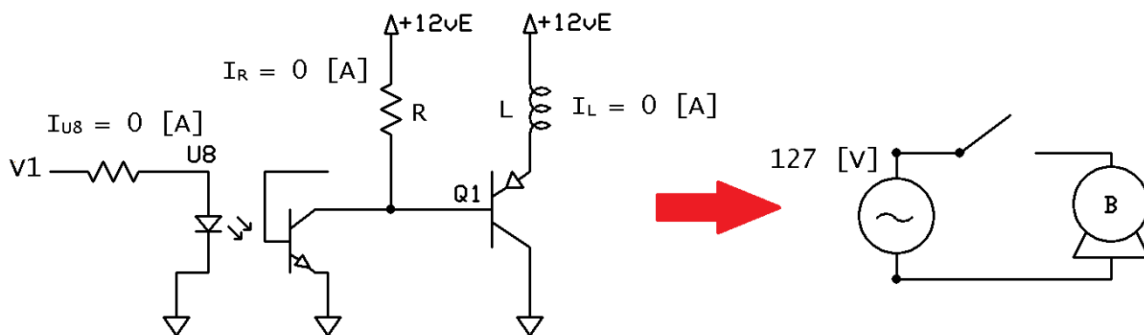


Figura 2.10: Control de Bomba, $V_1 = Z$ o L.

Cuando $V1$ se encuentra en un estado alto (H), ambos transistores operan en la región de saturación, cerrando el relevador, y en consecuencia, activando la bomba, como se ve en la Figura 2.11.

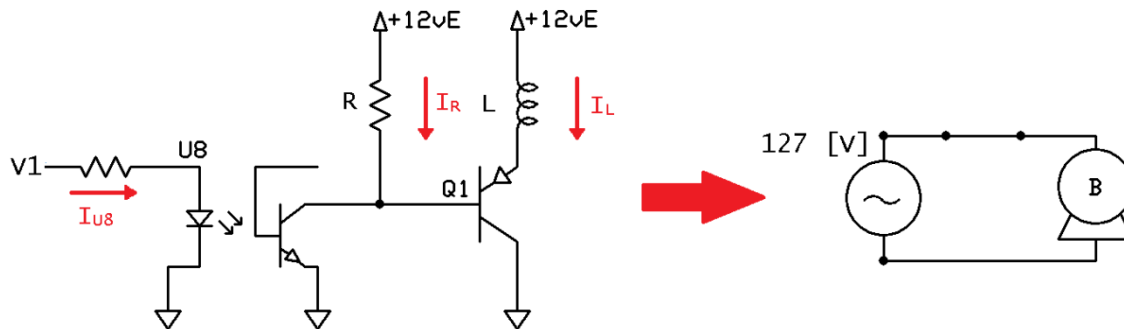


Figura 2.11: Control de Bomba, $V1 = H$.

2.2.3 Etapa de potencia para la electroválvula

La electroválvula utilizada es una PV34-B (Omega Engineering Inc., sin fecha). La tarjeta original resultó defectuosa, pero se observó que el motor de la electroválvula es un motor a pasos unipolar 4H8618S0676D de Telco, cuyas especificaciones son:

- 1.8 [°/paso]
- 12 [V] de polarización
- 0.67 [A]

Por lo que, para mover dicho motor, se emplea una etapa de potencia que consiste en optoacopladores y transistores de potencia NPN. El circuito empleado se muestra en la Figura 2.12 y en la Figura 2.13

Durante la operación de la electroválvula, por cada paso, las bobinas se cargan magnéticamente, como se ve en la Figura 2.12.

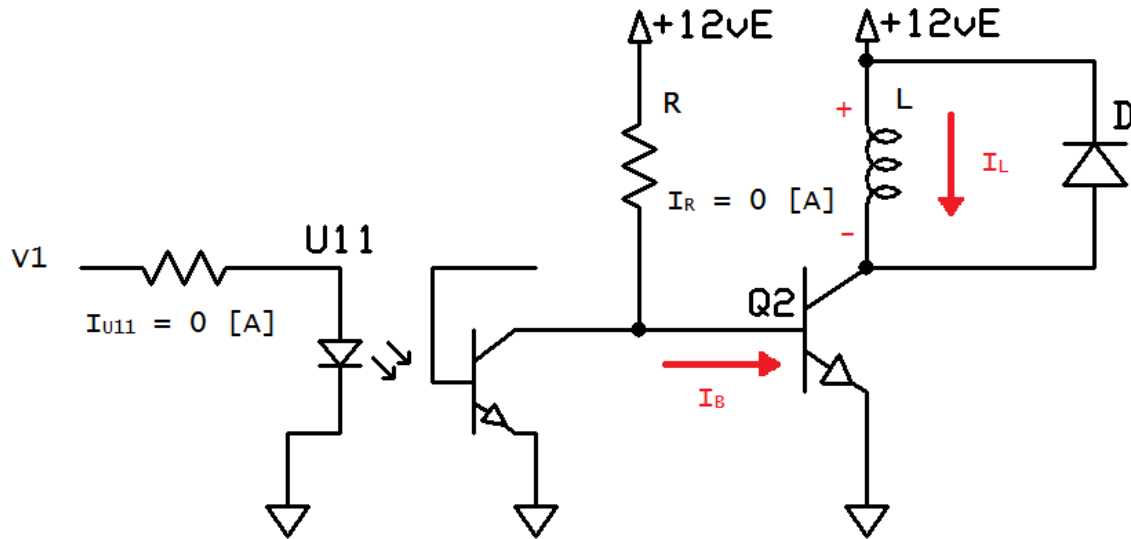


Figura 2.12: Etapa de potencia para electroválvula, $V_I = Z$ o L .

Al momento de hacer el cambio de paso, el inductor tratará de resistir la caída repentina de corriente (la caída de voltaje en la bobina es grande), usando la energía magnética que tenía almacenada. Esto hace que la polaridad de la bobina se invierta, sin embargo, el transistor podría no soportar este voltaje en reversa. Para contrarrestar esto, se coloca un diodo en el colector del transistor de forma que el diodo esté polarizado de forma inversa durante la carga de la bobina. Cuando deje de circular corriente en el transistor, el diodo formará un lazo con el inductor, como se aprecia en la Figura 2.13, de forma que éste se descargue sin dañar al transistor.

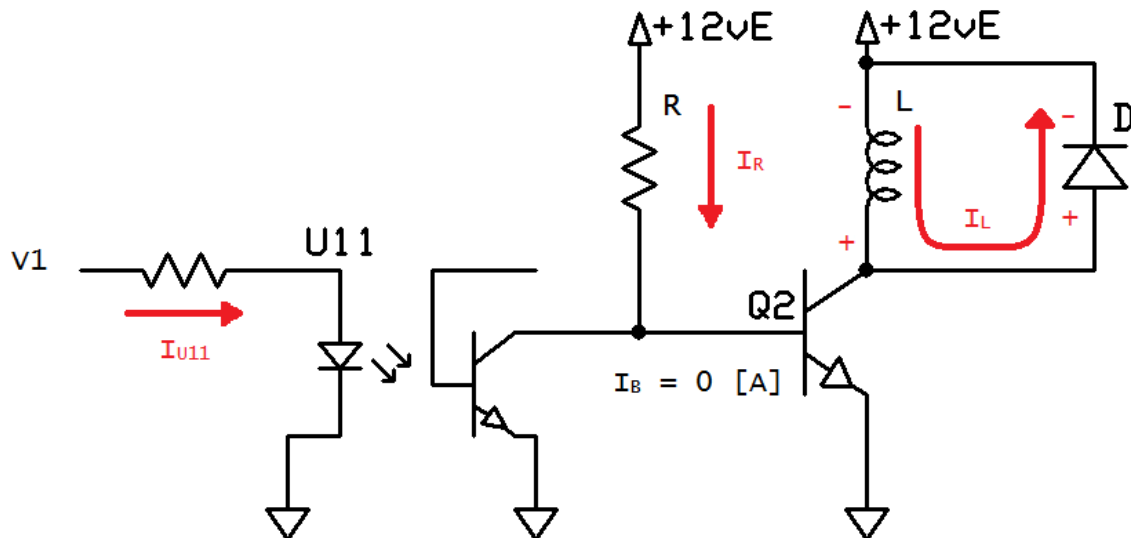


Figura 2.13: Etapa de potencia para electroválvula, $V_I = H$.

Como el motor a pasos de la electroválvula posee cuatro bobinas, se utilizan cuatro arreglos como el mostrado anteriormente.

2.2.4 Microcontrolador PIC18F4553

Para poder trabajar con el microcontrolador PIC18F4553 (Erdoğan, sin fecha), es necesario colocar unos cuantos componentes externos tal como se observa en la Figura 2.14.

Para usar la funcionalidad USB es necesario colocar un cristal de cuarzo cuya frecuencia de resonancia en paralelo es de 20 [MHz]. Un PPL interno (PLL5) se encarga de elevar esta frecuencia a 48[MHz], que es la frecuencia a la cual se realizan transferencias por USB.

Éste microcontrolador posee un regulador interno de 3.3 [V] para la comunicación USB, además de resistores de pull-up internos. Microchip recomienda un capacitor en el pin V_{USB} para hacer al regulador interno más estable.

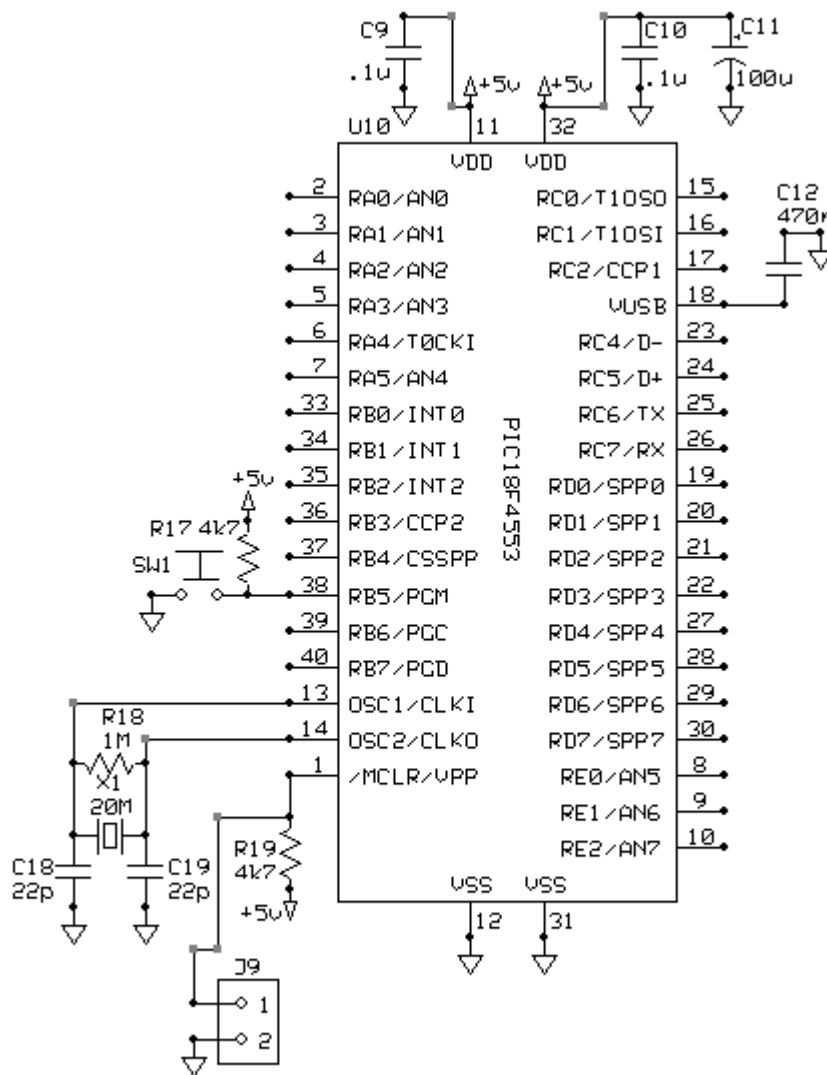


Figura 2.14: Componentes discretos básicos para el PIC18F4553.

Se coloca un conector Molex en J9 para instalar un botón de reset. Este botón provoca que el microcontrolador se reinicie y debe ser accesible para el usuario final.

Se incluye otro botón directamente en el circuito, que es útil si el microcontrolador tiene cargado un gestor de arranque (bootloader). El bootloader permite ejecutar el firmware actual o poder reprogramar el microcontrolador sin necesidad de removerlo del circuito. En este sistema el microcontrolador no tiene un bootloader.

Para energizar el microcontrolador se usa la propia entrada USB, la cual está compuesta por cuatro líneas: dos usadas para el intercambio de información, uno de 5 [V] y la tierra.

2.2.5 Firmware del Microcontrolador

El firmware del microcontrolador está compuesto por una rutina principal y tres rutinas de interrupción que serán explicadas a continuación.

No se incluye el código fuente, ya que éste es propiedad del Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la Universidad Nacional Autónoma de México.

Rutina principal

La Figura 2.15, la Figura 2.16 y la Figura 2.17 corresponden a la rutina principal del programa. Una vez que éste es iniciado, entra en un ciclo indefinido donde primero revisa si el microcontrolador es reconocido por la computadora (`usb_enumerated()`) y después revisa si existen datos en el buffer de entrada USB del microcontrolador. Si se cumplen ambas condiciones, se procede a ejecutar la rutina. Cabe mencionar que la frecuencia a la que opera el microcontrolador es 48 [MHz] y se configura al `TIMER0` para que desborde cada 1.3981 [s] y al `TIMER1` cada 5.46 [ms].

Esta rutina consiste en comparar el contenido de la primera localidad (llamada modo) de un arreglo de tres localidades, con respecto a una lista. En esta lista, se definen las acciones a realizar de acuerdo al modo recibido. Se envían datos promedio de las conversiones analógico/digitales realizadas (“valorCAD0”, “valorCAD1” y “valorCAD2”), se envía el periodo de la señal del sensor de flujo, se recibe la nueva posición para la electroválvula en “setpoint” y se enciende o apaga la bomba hidráulica.

Para obtener dichos valores se emplearon tres rutinas de interrupción, mostradas en la Figura 2.18, la Figura 2.19 y en la Figura 2.20.

Interrupción por CCP1

En la interrupción por captura en el CCP1 (Figura 2.18) se almacena el valor de la cuenta realizada por el `TIMER0` en la variable *periodo*. Una vez realizado esto, vuelve a iniciar la cuenta en cero. La variable *periodo* contiene la información

necesaria para que el programa en la computadora realice la conversión a una variable con unidades de flujo.

Interrupción por TIMER0

En la interrupción por el desbordamiento del TIMER0 (Figura 2.19), sólo cambia el estado del pin E0 (verificación de funcionamiento) y hace que *periodo* almacene un cero. Este cero será procesado en el programa desarrollado en Visual C++ para indicar que no hay flujo.

Interrupción por TIMER1

La rutina mostrada en la Figura 2.20 inicia cuando el TIMER1 del PIC18F4553 se desborda al cabo de 5.46 [ms] desde que inicia el programa. Esto genera una interrupción e inicia una rutina en donde realiza la conversión analógico/digital en los canales A0, A1 y A2, 45 veces, sumando estos datos en las variables “acumulador0”, “acumulador1” y “acumulador2”. Con estos datos se obtienen los valores promedio cada 0.25 [s] que se almacenan en las variables “valorCAD0”, “valorCAD1” y “valorCAD2”. La variable valorCAD0 contiene la información necesaria para que el programa en la computadora realice la conversión a una variable con unidades de porcentaje de apertura en la electroválvula. Las variables valorCAD1 y valorCAD2 contienen la información necesaria para que el programa en la computadora realice la conversión a variables con unidades de presión. Una vez realizado esto se reinicia el TIMER1.

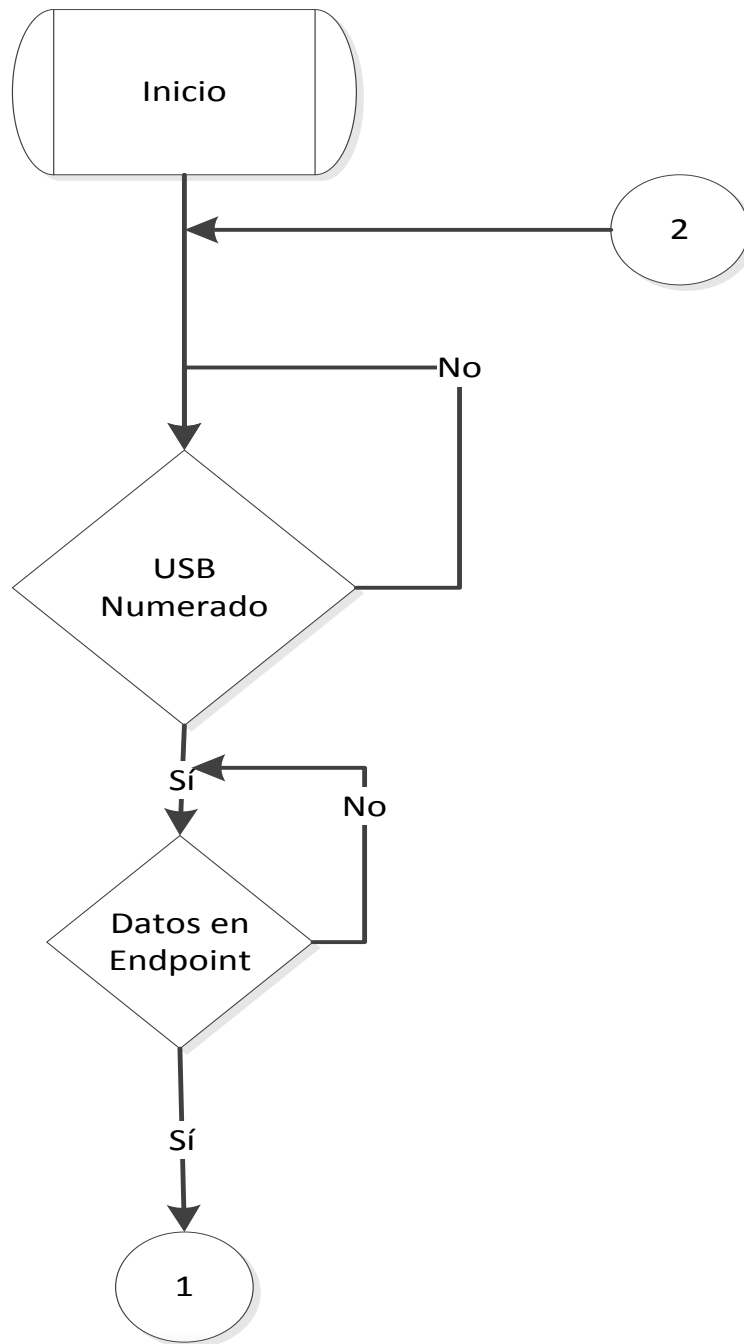


Figura 2.15: Primer parte del diagrama de flujo para la rutina principal.

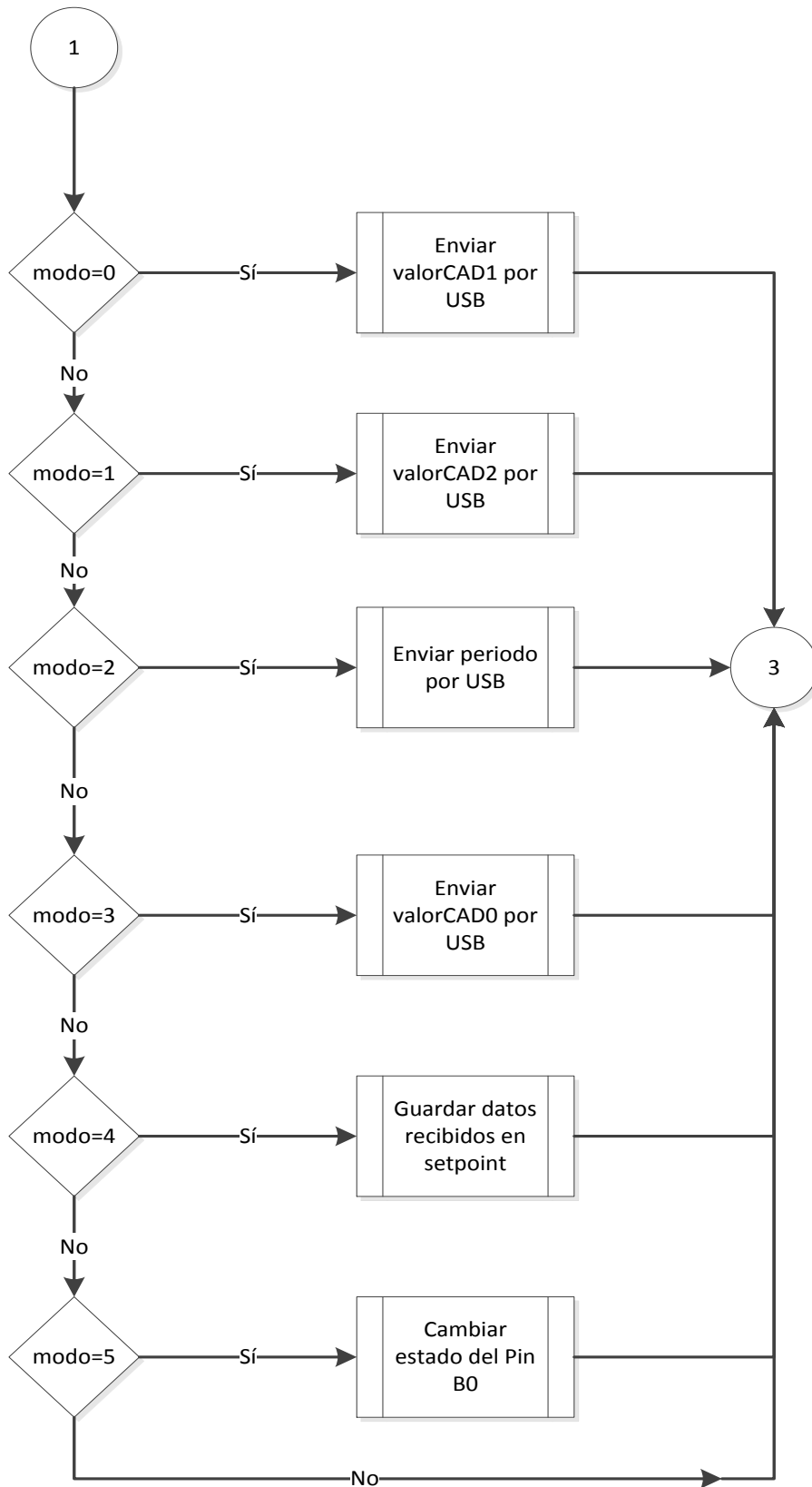


Figura 2.16: Segunda parte del diagrama de flujo para la rutina principal.

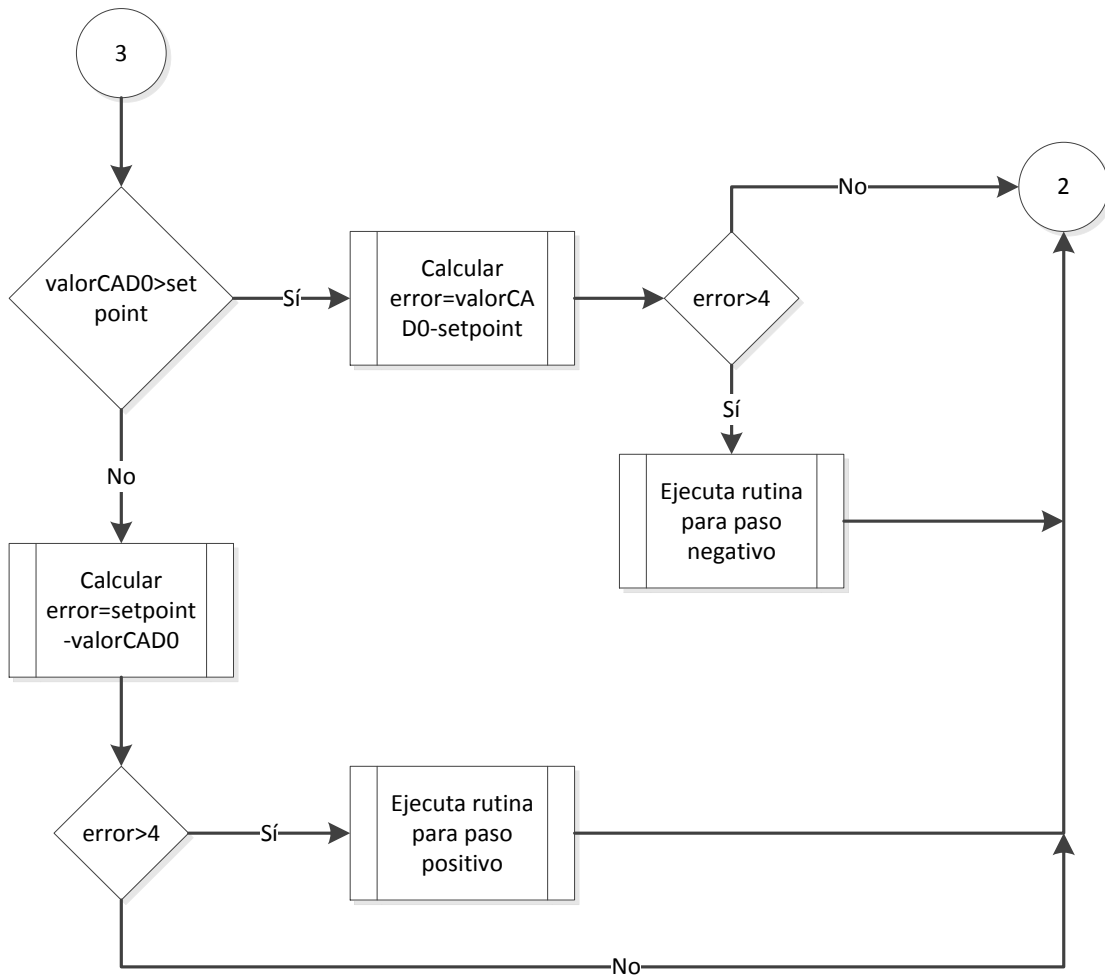


Figura 2.17: Tercera parte del diagrama de flujo para la rutina principal.

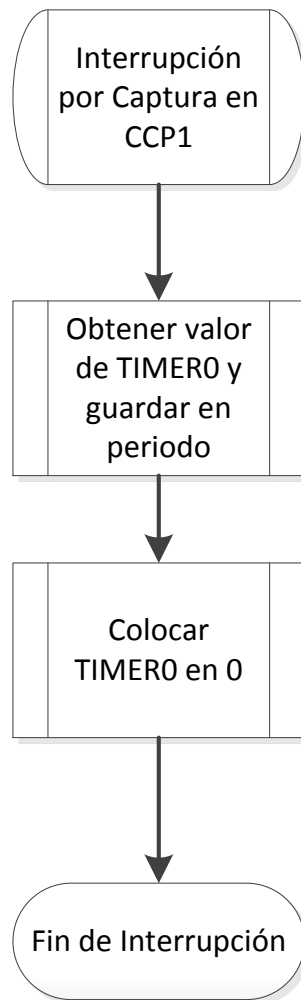


Figura 2.18: Rutina para manejo de interrupción de CCP1 por captura.

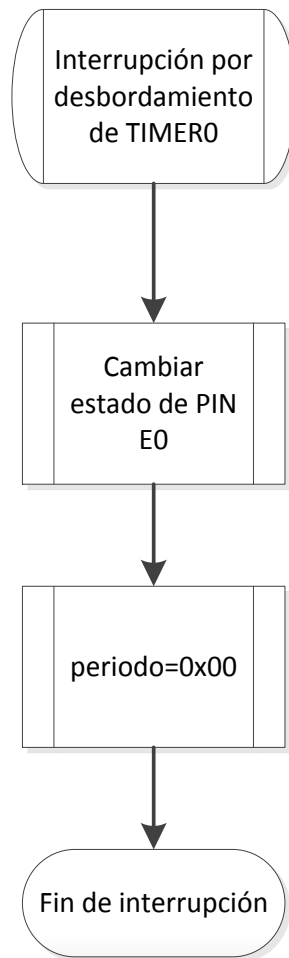


Figura 2.19: Rutina para manejo de interrupción por desbordamiento de TIMER0.

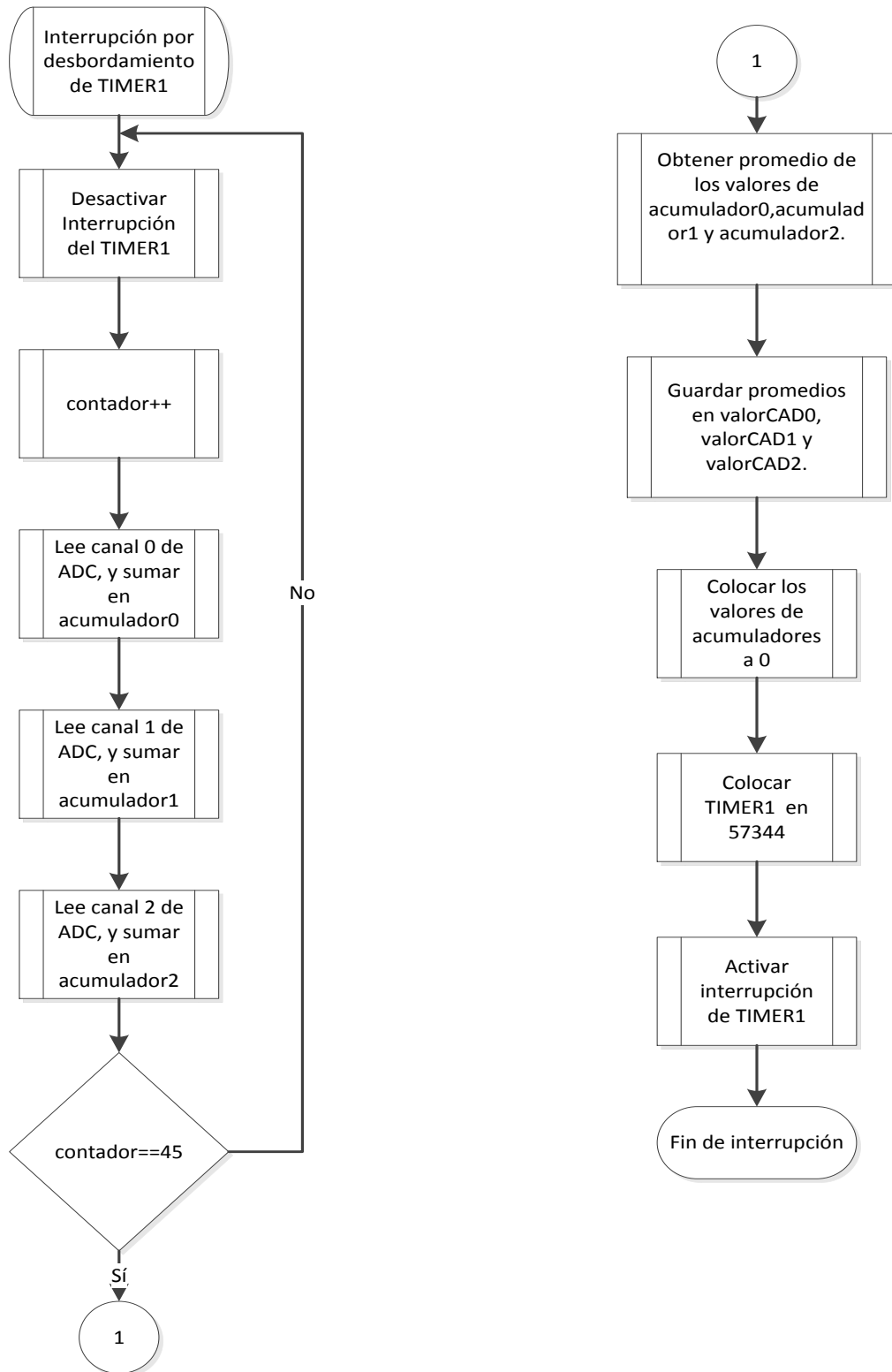


Figura 2.20: Rutina para manejo de interrupción por desbordamiento de TIMER1.

2.2.6 Alimentación eléctrica

Debido a que el puerto USB 2.0 de una computadora tiene un límite superior de 500 [mA] de corriente, para energizar los diversos componentes del sistema electrónico se requieren fuentes adicionales para impulsar la electroválvula así como energizar los amplificadores operacionales que utilizan tecnología bipolar.

Fuente para la electroválvula

Este circuito está diseñado para obtener los 12 [V] y 0.67 [A] que requiere la electroválvula para su operación. Esto se logró mediante un regulador lineal con un amplificador de corriente. El circuito se ilustra en la Figura 2.21.

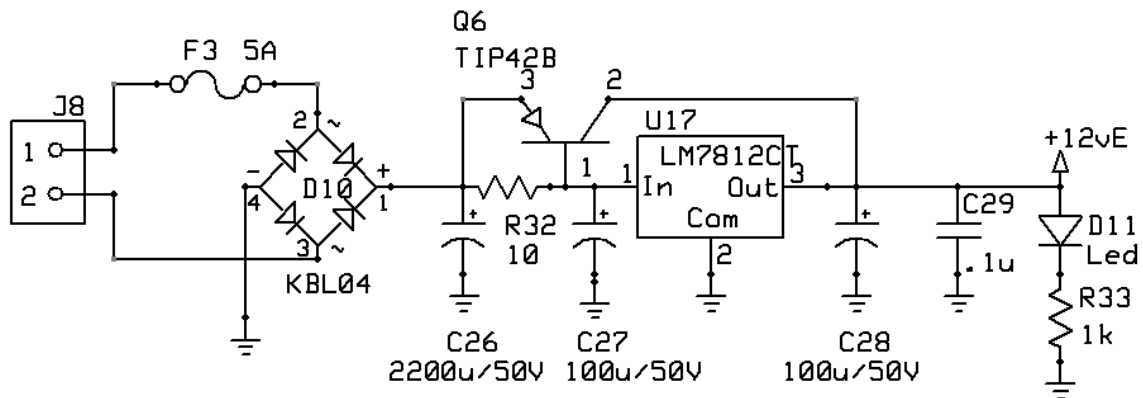


Figura 2.21: Fuente para la electroválvula.

Este circuito rectifica una señal de un transformador de 15 [V_{RMS}] a 6 [A_{RMS}] y después de que es filtrada, se emplea un regulador de voltaje LM7812 cuya salida es una tensión de 12 [V].

Una solución es usar un transistor PNP y un resistor tal como en la Figura 2.21. El voltaje en el resistor, así como el voltaje de la unión base-emisor del transistor, determinarán cuando el transistor conduzca. Debido al elevado nivel de corriente manejada en este circuito, se decidió por usarlo exclusivamente en la polarización y movimiento de la electroválvula, aislándolo del resto del sistema.

Fuente simétrica ±12 [V]

Para energizar a los amplificadores operacionales, como al relevador usado, se utiliza una fuente simétrica. Debido a que los requerimientos de corriente son por mucho menores a los usados en el caso de la electroválvula, no se emplea la configuración anterior. Como resultado, el circuito es mucho más sencillo.

Para este circuito, se empleó un transformador de 24 [V] a 0.5 [A] con derivación central, un regulador de voltaje positivo LM7812 y un regulador de voltaje negativo LM7912, como base del sistema. Los capacitores son utilizados para filtrar el voltaje de salida del rectificador a un nivel de DC casi constante y el

propio regulador remueve el rizo que existe. El LM7812 tiene una razón de rechazo al rizo de 55[dB] y el LM7912 de 60 [dB]. El circuito se muestra en la Figura 2.22.

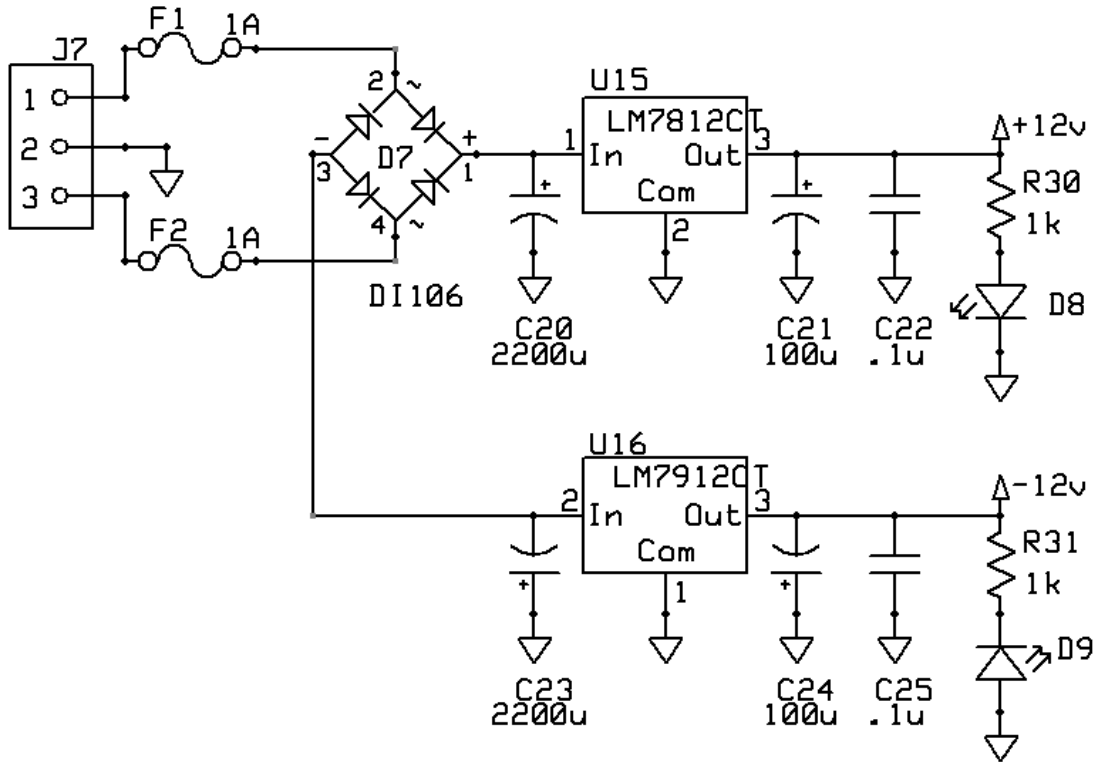


Figura 2.22: Fuente simétrica de ± 12 [V].

2.3 Software de operación

El programa desarrollado para este proyecto consiste de una interfaz de control entre la computadora y el sistema electrónico; la visualización de los datos adquiridos del microcontrolador en la pantalla, la posibilidad de realizar una gráfica de presión contra velocidad de flujo y la capacidad de guardar los datos en un archivo de texto.

2.3.1 Descripción del software en el ámbito del operador

La información adquirida por el microcontrolador y los comandos de control son transmitidos a través de una comunicación USB con una computadora. Se desarrolló un programa en Visual C++ utilizando las librerías MFC y la librería de enlace dinámico (DLL) mpusbapi.dll de Microchip para establecer la comunicación USB; interpretar los datos obtenidos de los sensores y mandar los comandos de control a la bomba y la electroválvula. En este apartado, se explicará la forma de instalar el programa en una computadora con Microsoft Windows, los requerimientos básicos del sistema, la operación del programa y por último las clases que conforman al programa y su funcionamiento.

Requerimientos mínimos de sistema

- Sistema operativo compatible
 - Windows 7 SP1 (x86 y x64)
 - Windows 8 (x86 y x64)
 - Windows Server 2008 R2 SP1 (x64)
 - Windows Server 2008 SP2 (x64)
 - Windows Vista SP2 (x86 y x64)
 - Windows Server 2012 (x64)
 - Windows XP
 - Windows 2003
- Requisitos de hardware:
 - Procesador a 1,6 GHz o más rápido
 - 1 GB de RAM (1,5 GB si se ejecuta en una máquina virtual)
 - 10 GB de espacio disponible en el disco duro
 - Unidad de disco duro de 5400 rpm
 - Tarjeta de vídeo compatible con DirectX 9 con una resolución de pantalla de 1024 x 768 o superior

Creación

El software fue desarrollado en Visual Studio 2012. El proyecto fue nombrado “Visual” y en la carpeta de éste se encuentran los archivos de código fuente y ejecutables. En la carpeta raíz del proyecto, existe una carpeta llamada “Release”, donde se colocaron los archivos necesarios para poder instalar el programa en otros equipos.

Estos archivos son propiedad del Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la Universidad Nacional Autónoma de México.

Instalación

1. Copie el contenido de la carpeta “Release”, encontrada dentro de la carpeta del proyecto “Visual” y dépositelo en un lugar conveniente.

Existen siete elementos en la carpeta “Release”, Figura 2.22:

- ControlBanco.exe: Es el programa de control y medición del sistema electrónico.
- mfc110.dll, mfc110u.dll, mfcm110.dll, mfcm110u.dll, msvcpl110.dll, msucr110.dll: Son las librerías correspondientes a la MFC requeridas por el programa.
- mpushapi.dll: Es la librería que permite la comunicación USB entre el microcontrolador y la computadora.
- La Carpeta “driver”: Esta carpeta contiene el driver necesario para la conexión USB con el microcontrolador PIC18F4553, es compatible con sistemas operativos Microsoft Windows de 32-bit y 64-bit.

- La Carpeta “NTGraph”: Contiene el control ActiveX usado para las funciones de graficación.

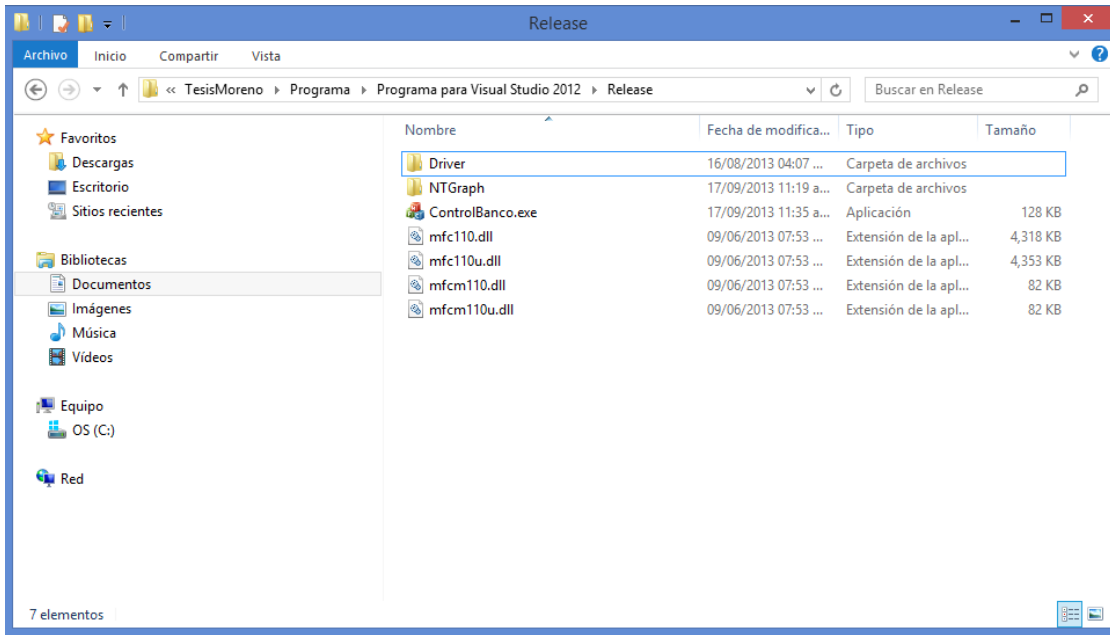


Figura 2.23: Contenido de Release.

2.- Las siguientes indicaciones son para instalar el control ActiveX NTGraph. Inicialice el Símbolo de Sistema con privilegios de administrador. Una vez realizado esto, se tiene que navegar a la carpeta que almacena a la herramienta “RegSvr32.exe” de 32-bit. En un sistema de 32-bit, se encuentra en “C:\windows\system32”, y en un sistema de 64-bit, en “C:\windows\syswow64”. (Microsoft Support, 2007)

Una vez localizado el programa, se introduce el siguiente comando para instalar el control ActiveX:

RegSvr32.exe C:\Ruta_del_ocx\NTgraph.ocx

La Figura 2.24 y la Figura 2.25 muestran un ejemplo de instalación

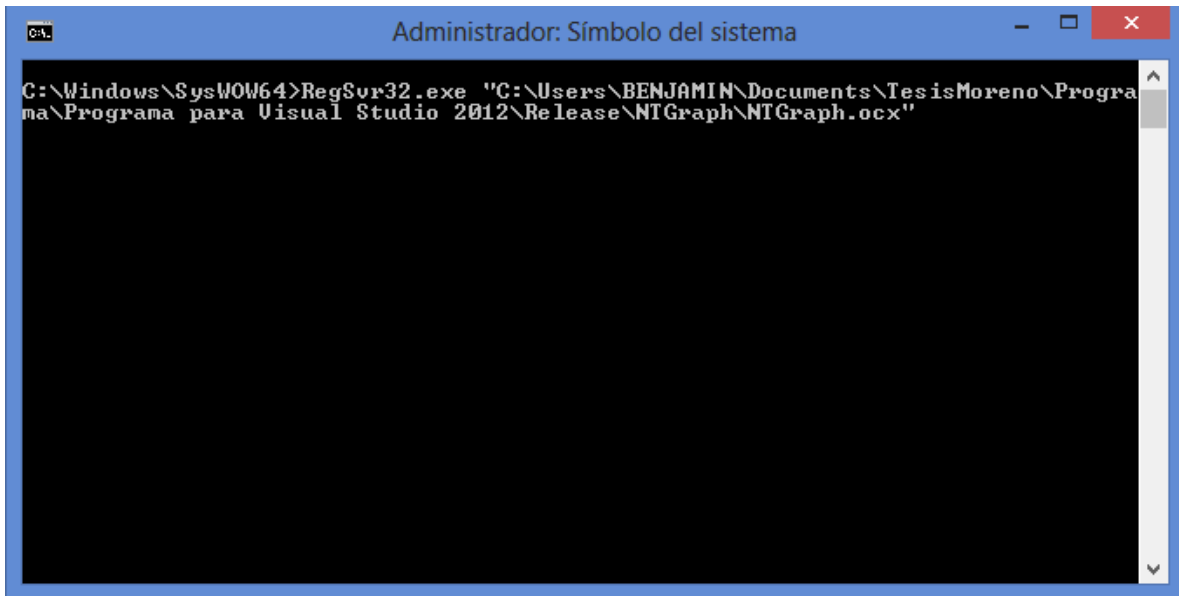


Figura 2.24: Instalación del control ActiveX NTGraph.

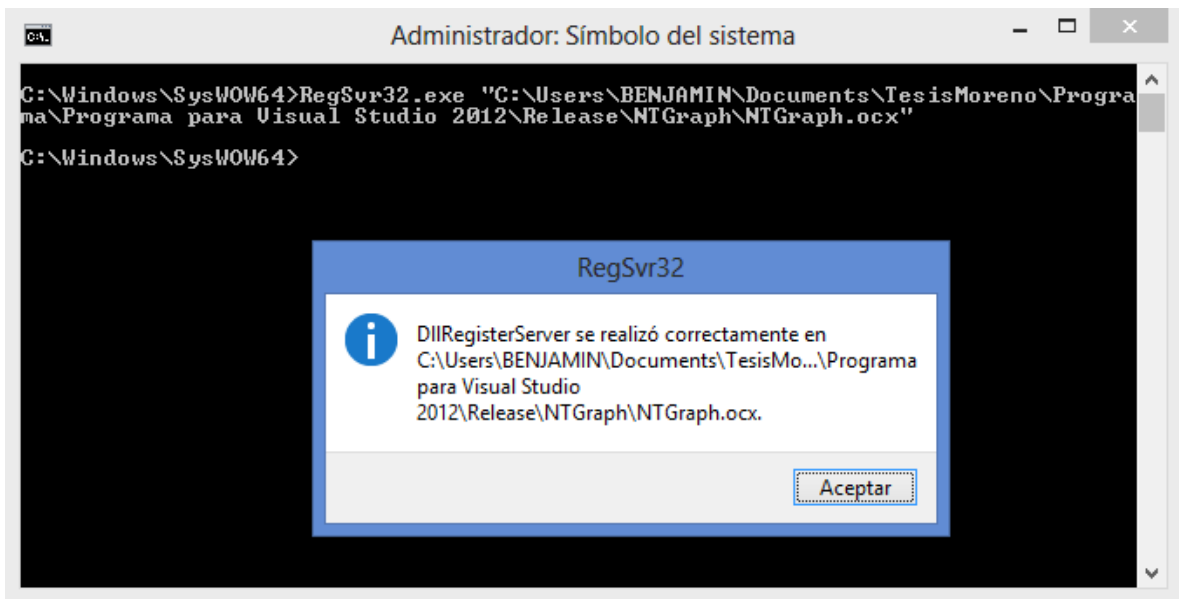


Figura 2.25: Confirmación de instalación del control ActiveX.

3.- Una vez instalado el control ActiveX, conecte el circuito electrónico mediante un cable USB a la computadora. Cuando el asistente de instalación de hardware lo solicite, instale el driver que se incluye en la carpeta “Release”, como se muestra en la siguiente serie de imágenes.

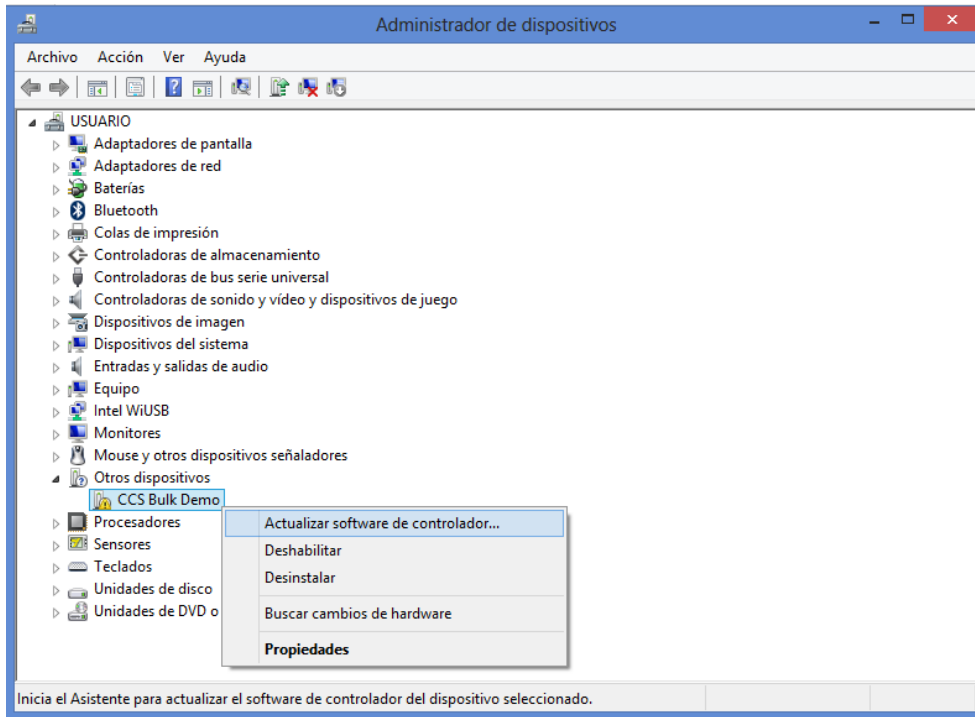


Figura 2.26: Pantalla del administrador de dispositivos.

En la Figura 2.26 se muestra al microcontrolador siendo reconocido por Windows 8, en este caso, el sistema operativo no cuenta con el driver necesario para operar, por lo que se utiliza el asistente de actualización de driver, como se ve en la figura anterior.

Cuando el asistente pregunte cómo se desea buscar el software del controlador, se elige la opción “Buscar software de controlador en el equipo” y en el siguiente diálogo se escoge la opción “Elegir en una lista de controladores de dispositivo en el equipo”. Una vez hecho esto, seleccionar la opción “Mostrar todos los dispositivos” y presionar “Siguiente”. En el dialogo de selección de controlador de dispositivo a usar, seleccionar el botón “Usar Disco”. Esto abrirá otro dialogo como en la Figura 2.27.

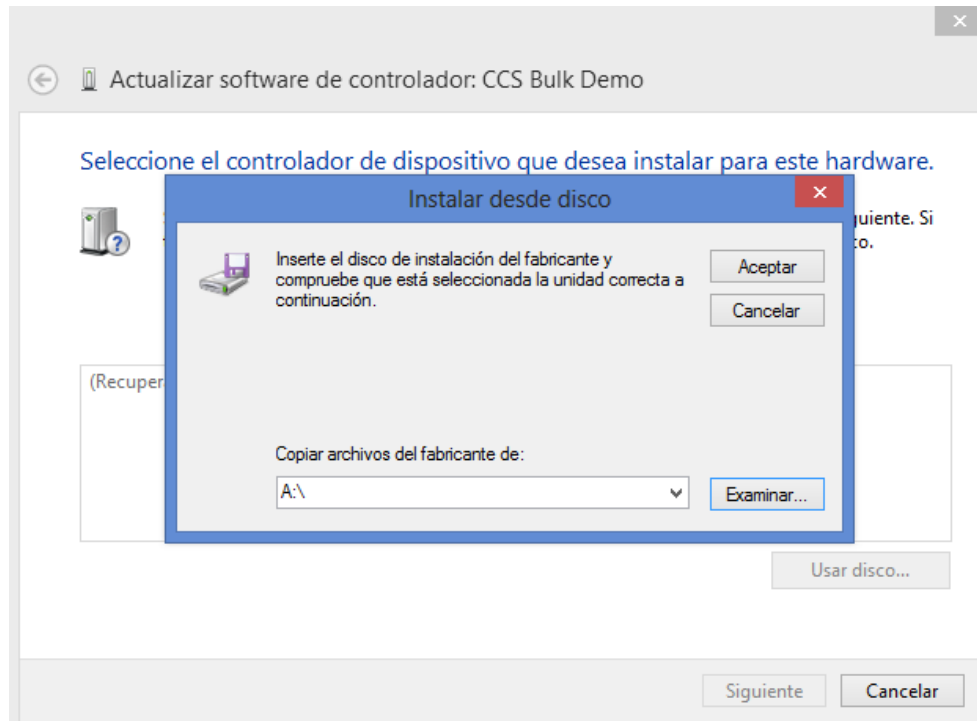


Figura 2.27: Instalación manual de controladores de dispositivo.

Una vez hecho esto, se selecciona “Examinar” y se dirige a la carpeta “driver”, para seleccionar al archivo “mchpub.ini”. La lista de hardware compatible debería verse como la Figura 2.28.

Una vez finalizado el proceso, en el Administrador de dispositivos se muestra una categoría llamada “USB Custom Devices”, en la cual está listado “Microchip Custom USB Device”. Si es así, se ha instalado con éxito el controlador del PIC18F4553.

Una vez realizados estos pasos, el sistema se encuentra correctamente instalado en el equipo. Para comenzar a trabajar hay que asegurarse que los transformadores del resto del sistema estén conectados, de otra forma, no se obtendrá información alguna ni se podrá controlar el sistema.

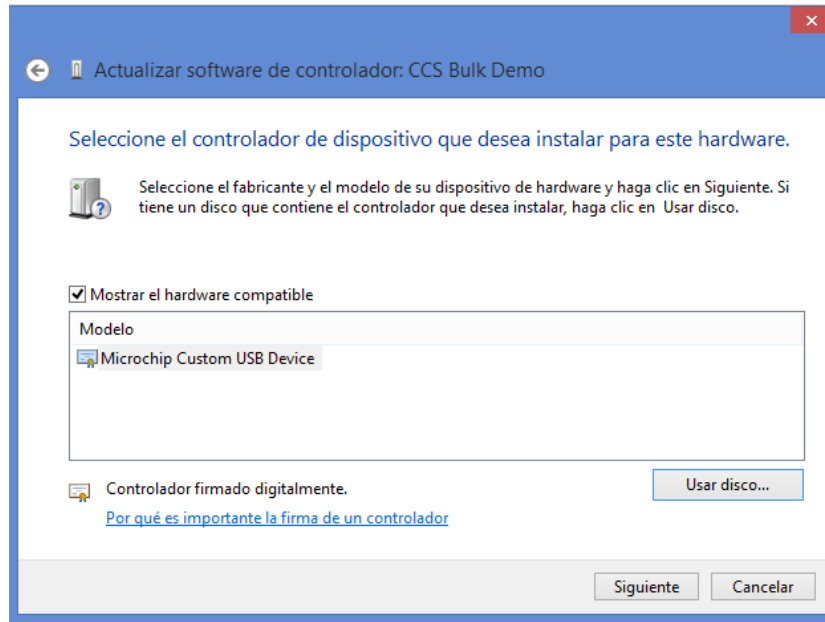


Figura 2.28: Listado de hardware compatible con el controlador instalado.

Uso del software

Una vez instalados los componentes mencionados anteriormente, se procede a ejecutar el archivo ControlBanco.exe, esto abrirá un dialogo como se ve en la Figura 2.29

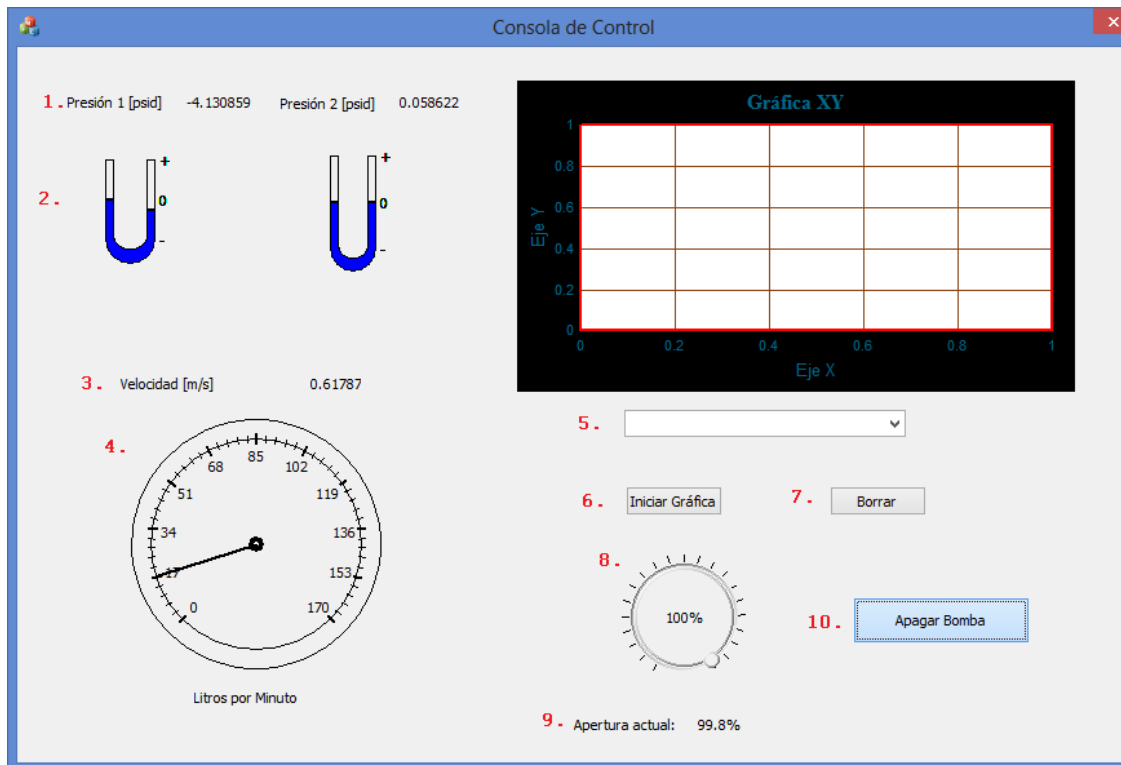


Figura 2.29: Ventana principal del programa en Funcionamiento.

Donde:

1. Son los indicadores en texto para los sensores de presión. Presión 1 es para el sensor de ± 15 [psid] y Presión 2 para el sensor de ± 30 [psid].
2. Son los indicadores visuales para los sensores de presión. El dibujo cambia en función de la presión medida por los sensores.
3. Es el indicador textual para el gasto volumétrico medido.
4. Es el indicador visual para el gasto volumétrico. Éste valor es obtenido a partir de la velocidad del fluido, su densidad y el diámetro de la tubería por donde pasa. En este caso, se toma el diámetro del aditamento con el cual se conecta al circuito (1”).
5. Es el selector de datos a graficar. Las opciones disponibles son Presión (± 15 [psid]) contra Velocidad de flujo [m/s] y Presión(± 30 [psid]) contra Velocidad de flujo [m/s].
6. Inicia la rutina para adquirir datos y posteriormente graficarlos. Una vez adquiridos, el programa solicita un lugar para guardar la información en un archivo con extensión txt, como se ve en la Figura 2.30

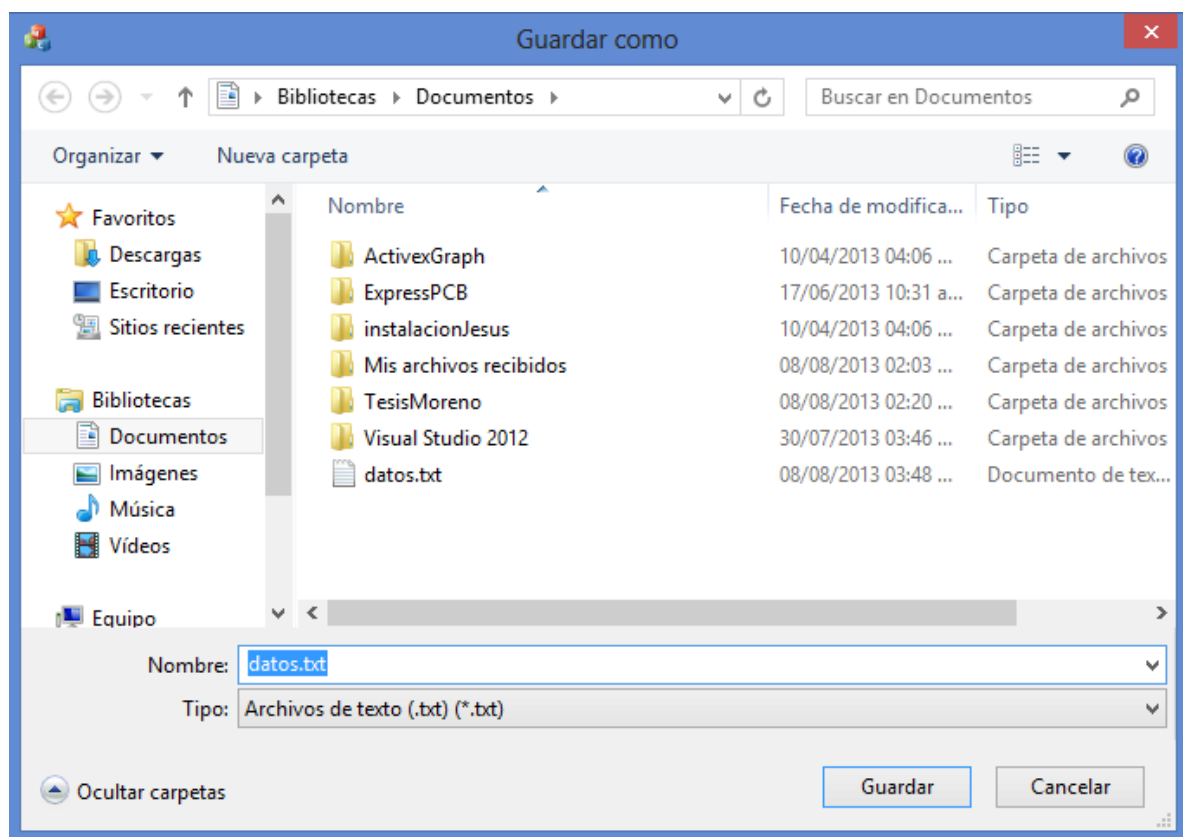


Figura 2.30: Diálogo para elegir lugar donde se guardarán los datos adquiridos.

7. Borra lo dibujado en la gráfica, como las variables donde se almacenaron los datos.
8. Es el control para la electroválvula. Al manipularse envía el porcentaje de apertura al microcontrolador para fijar la posición.

9. Es el indicador textual del porcentaje de apertura de la electroválvula.
10. Es el botón de Encendido/Apagado de la bomba hidráulica. El texto cambia en función de qué estado se encuentre la bomba. Como se puede ver en la Figura 2.29, la bomba está funcionando.

2.3.2 Descripción del software en el ámbito del programador

En esta sección se mostrará la estructura del programa desarrollado en Visual C++, describiendo las clases y los métodos que conforman el proyecto de programación. No se muestra el código fuente ya que es propiedad del Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM.

Diálogo de Control

La Figura 2.31 muestra como se aprecia el dialogo de control durante la etapa de desarrollo. En esta figura se aprecian los indicadores tipo barra (usados para la medición de presión y flujo), el control ActiveX para la gráfica, el menú usado para seleccionar los datos a graficar, una barra deslizable para el control de la electroválvula y diversos botones.

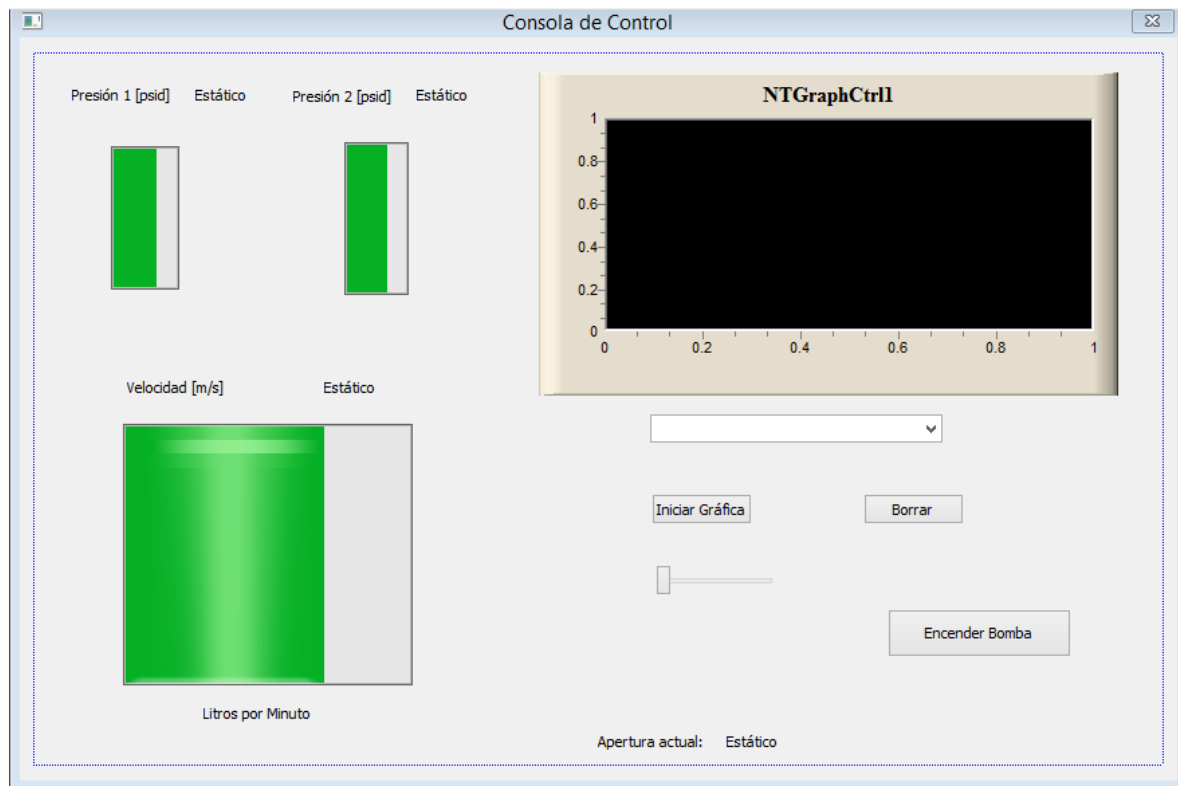


Figura 2.31: Consola de Control durante el desarrollo.

Clases empleadas

Las clases usadas en el proyecto de Visual C++ 2012 son las mostradas en la Figura 2.32.

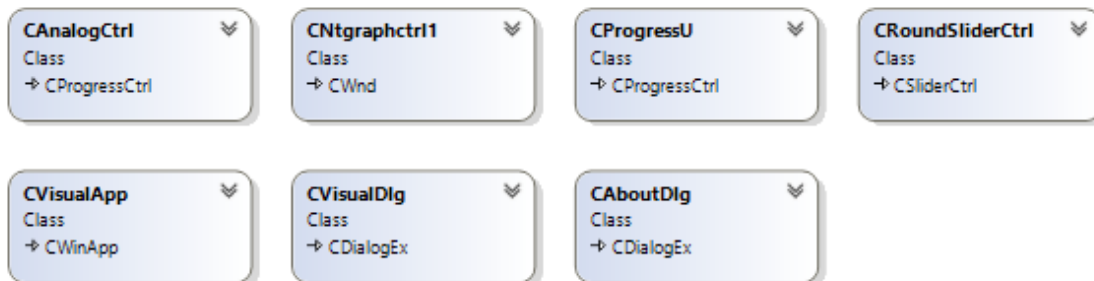


Figura 2.32: Clases usadas en el programa.

Donde:

- **CAalogCtrl**: Clase con base en CProgressCtrl. Modifica el dibujo original de un control “barra de progreso” para que se vea como un indicador analógico, como se muestra en el punto 4 de la Figura 2.29. El control conserva la funcionalidad original.
- **CNTgraphctrl1**: Es una clase generada por el control ActiveX NTGraphControl. Contiene los métodos necesarios para controlar la gráfica mostrada en la Figura 2.33.

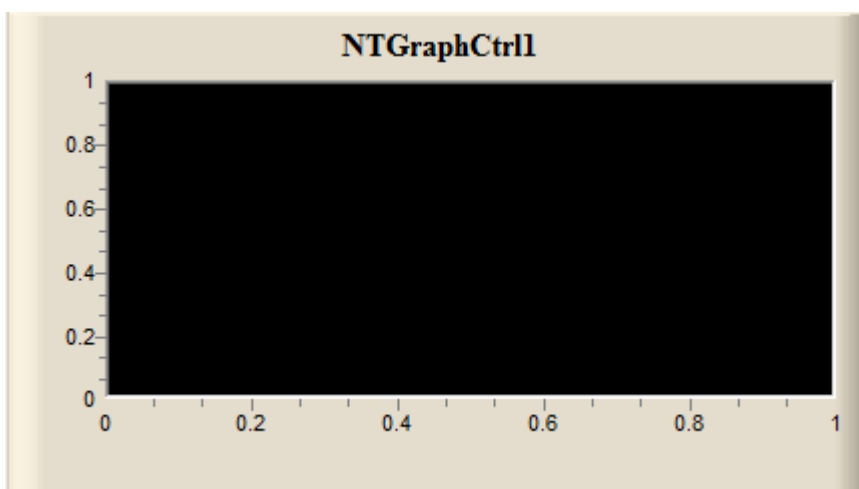


Figura 2.33: Gráfica generada por el Control ActiveX NTGraphControl.

- **CProgressU**: Clase con base en CProgressCtrl. Modifica el dibujo original de un control “barra de progreso”, para que se vea como un manómetro como se muestra en el punto 2 de la Figura 2.29. El control conserva la funcionalidad original.
- **CRoundSliderCtrl**: Clase con base en CSliderCtrl. Modifica el dibujo original de un control deslizable, para que se vea como una perilla de control como se muestra en el punto 8 de la Figura 2.29. El control conserva la funcionalidad original.
- **CVisualApp**: Esta clase, con base en CWinApp, es la que encapsula la inicialización, ejecución y terminación de un programa para el sistema

operativo Windows. Este objeto se construye antes de que se creen las ventanas.

- CVisualDlg: Clase con base en CDialogEx. Es la clase encargada de las acciones del diálogo. En esta se realizan los procesos de conversión de datos necesarios para la visualización de éstos en la ventana principal.
- CAboutDlg: Esta clase con base en CDialogEx se encarga de desplegar la ventana de “Acerca De”, cuando se solicita. No tiene impacto en la funcionalidad del programa.

Métodos empleados

Cada clase está compuesta de métodos, los cuales realizan diversas funciones dentro del programa. En esta sección se describirán los métodos que conforman a las clases anteriormente mencionadas.

CAnalogCtrl

Los métodos que conforman a CAnalogCtrl son mostrados en la Figura 2.34:

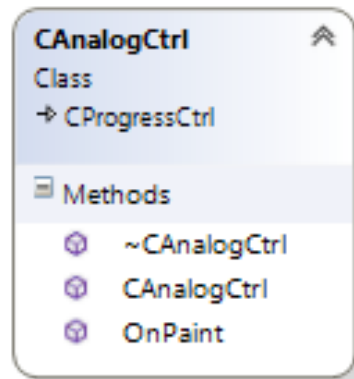


Figura 2.34: Métodos de la Clase CAnalogCtrl.

Donde:

- ~CAnalogCtrl: Método Vacío.
- CAnalogCtrl: Método Vacío.
- OnPaint: Responde a cada WM_PAINT que el sistema envíe. Este método se encarga de dibujar el medidor analógico y establecer la posición del indicador en función del flujo registrado.

CNTgraphctrl1

Los métodos fueron generados al importar el control ActiveX NTGraphControl de Nikolai Teofilov (2003) y no serán discutidos en este trabajo.

CProgressU

Los métodos que conforman a CProgressU son mostrados en la Figura 2.35:

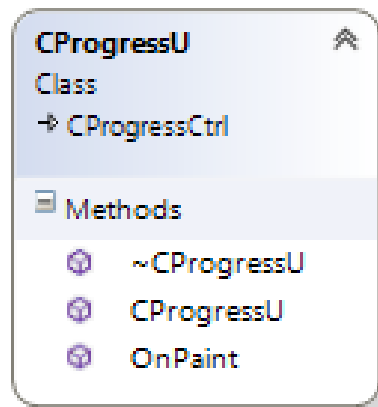


Figura 2.35: Métodos de la Clase CProgressU.

Donde:

- ~CProgressU: Método vacío
- CProgressU: Método vacío
- OnPaint: Responde a cada WM_PAINT que el sistema envíe. Este método se encarga de dibujar a un barómetro en lugar de la barra de progreso normal. A su vez, modifica el nivel actual del manómetro de acuerdo a los valores leídos.

CRoundSliderCtrl

Los métodos que conforman a CRoundSliderCtrl son mostrados en la Figura 2.36:



Figura 2.36: Métodos de la Clase `CRoundSliderCtrl`.

Donde:

- ~CRoundSliderCtrl: Método vacío.
- AssertValid: Verifica que este objeto sea válido. Por validez, se refiere a que revisa si el objeto tiene punteros hacia otros objetos y en caso de que así sea, que estos punteros no sean nulos.
- CRoundSliderCtrl: Contiene información para la inicialización del control.
- Dump: Permite extraer los datos de este objeto para ayudar con la depuración.
- GetInverted: Regresa el estado actual de inversión del control.
- GetKnobRadius: Obtiene el radio actual de la perilla.
- GetText: Obtiene la cadena almacenada en el control. Esta cadena es un porcentaje.
- GetZero: Obtiene el valor asignado a “cero”.
- Init: Hace al control cuadrado, obtiene sus estadísticas importantes (centro y radio), y asigna la región activa del control a sólo la sección circular del control de deslizamiento.
- OnEraseBkgnd: Regresa un valor distinto a cero si se ha borrado el fondo del control, cero si falla. Es llamado cuando el fondo necesita ser borrado (como cuando cambia de tamaño o se actualiza).
- OnFormatText: Agrega la cadena base definida en CRoundSliderCtrl y el valor actual del control.
- OnKeyDown: Define el movimiento del control en función de las teclas presionadas. Estas son Flecha Izquierda, Flecha Derecha, Flecha Arriba, Flecha Abajo, Retroceder Página y Avanzar Página.
- OnKeyUp: Tiene rutinas para cuando ciertas teclas dejen de ser presionadas. En este caso no se realiza ninguna acción.
- OnLButtonDown: Registra cuando se presiona el botón izquierdo del mouse. En esta rutina se permite al usuario arrastrar el control cuando mueva el mouse (esto es detectado por otro método), redibujándolo en el proceso.
- OnLButtonUp: Registra cuando se levanta el botón izquierdo del mouse. En esta rutina se desactiva el arrastre de la perilla.
- OnMouseMove: Detecta si el modo de arrastre está activado, en caso de que así sea, mueve la posición del control.
- OnPaint: Responde a cada WM_PAINT que el sistema envíe. Este método se encarga de dibujar el control tipo perilla en función del tamaño del control insertado así como la posición actual.
- OnSize: Ejecuta el método Init cuando se cambia el tamaño de la ventana.
- PostMessageToParent: Envía un mensaje a la ventana principal. Este mensaje puede ser el movimiento del control y la posición de éste.

- **PreCreateWindow:** Es un método que se ejecuta antes de la creación de la ventana que contiene al control, define aspectos como el rango y tamaño de línea de éste.
- **PreSubclassWindow:** Este método es ejecutado para permitir la creación de subclases antes de que esta ventana se vuelva una subclase. Ejecuta la misma rutina que **PreCreateWindow**.
- **SetInverted:** Permite al usuario invertir el control o no.
- **SetKnob:** Realiza los cálculos necesarios para la posición actual de la perilla.
- **SetKnobRadius:** Permite modificar el radio de la perilla.
- **SetText:** Permite modificar la cadena de texto en el control de la perilla.
- **SetZero:** Permite modificar el “cero” del control.

CVisualApp

Los métodos que conforman a **CVisualApp** son mostrados en la Figura 2.37:

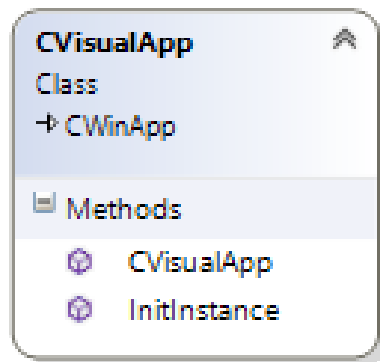


Figura 2.37: Métodos de la Clase CVisualApp.

Donde:

- **CVisualApp:** Es el constructor del objeto **CWinApp**. Además de que guarda el nombre de la aplicación.
- **InitInstance:** Contiene las rutinas de inicialización del diálogo principal. Permite la ejecución de varias instancias del programa al mismo tiempo.

CVisualDlg

Los métodos que conforman a **CVisualDlg** son mostrados en la Figura 2.38, donde:

- **CVisualDlg:** Es el constructor de la clase. Además, inicializa variables miembro que serán usadas a lo largo de esta clase en particular.
- **DoDataExchange:** Este método es usado para validar e intercambiar la información obtenida en un diálogo.

- OnBomba: Verifica el estado actual del botón “Encendido/Apagado” en el diálogo. Cambia la cadena de texto del botón, y envía los datos necesarios al microcontrolador por USB para encender o apagar la bomba.

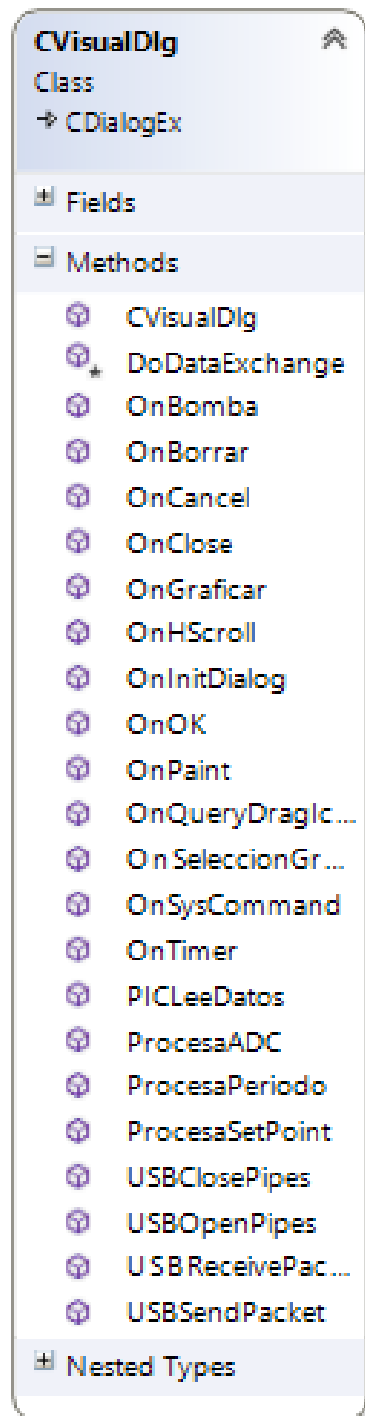


Figura 2.38: Métodos de la Clase CVisualDlg.

- OnCancel: Esta función está diseñada para que no haga nada. El comportamiento original es cerrar la ventana sin guardar datos al momento de presionar un botón de cancelación o presionando la tecla “Esc”.
- OnClose: Es ejecutada cuando se cierre el programa. Apaga la bomba hidráulica automáticamente.
- OnGraficar: Es ejecutado cuando el usuario activa el botón de “Graficar”. Desactiva este botón y el de “Borrar”, detecta el tipo de gráfica realizar y modifica los parámetros visualizados en el control ActiveX de la gráfica. Reserva espacio de memoria para cien muestras de los sensores de presión, flujo y velocidad de fluido. Activa la rutina para guardar los datos en el método OnTimer.
- OnHScroll: Es activado cuando se mueve el control deslizable. Transforma el valor porcentual de apertura en un número manejable para el microcontrolador. Se crea un arreglo para almacenar esta información y se envía por USB al microcontrolador.
- OnInitDialog: Agrega el elemento “Acerca de...” al menú de sistema, establece el ícono para el cuadro de diálogo, realiza la inicialización del control de apertura de la electroválvula, el indicador de flujo, un temporizador a 0.25 [s], y los parámetros iniciales para el control ActiveX de la gráfica. Enlaza la DLL de USB de Microchip con el programa y hace que la electroválvula se abra completamente.
- OnOk: Tiene la misma función que OnCancel. La diferencia es que esta función es activada al presionar la tecla “Enter”.
- OnPaint: Por defecto, dibuja el ícono de “Minimizar”, si éste es incluido.
- OnQueryDragIcon: Obtiene el cursor que se muestra mientras se arrastra la pantalla minimizada.
- OnSeleccionGrafica: Obtiene el tipo de gráfica a realizar cuando se manipula el control tipo lista.
- OnSysCommand: Despliega el diálogo “Acerca de...” cuando se selecciona del menú de sistema. Se puede acceder a éste si se hace click con el botón derecho del mouse en el título de la ventana.
- OnTimer: Cada 0.25 [s] solicita los datos de presión, periodo, y posición al microcontrolador, llama a los métodos que procesan los datos obtenidos en valores leíbles por el usuario y modifica los indicadores de texto para que muestren la información procesada. Si se ha activado el botón “Graficar”, almacena los datos que recibe en las localidades de memoria creadas al momento de pulsar ese botón. A su vez, grafica el conjunto de datos fijados por el menú de selección de gráfica. Una vez terminado el proceso, se procede a guardar los datos obtenidos en un archivo de texto. En esta sección del programa se realizan las conversiones de información transmitida por el sistema electrónico a variables con unidades de medida.

Los métodos empleados para la conversión son *ProcesaADC*, *ProcesaPeriodo* y *ProcesaSetpoint*.

- *PICLeeDatos*: Este método recibe el puntero donde se almacenará el dato solicitado así como el identificador para el PIC18F4553 y prepara el espacio de memoria para recibir la información solicitada. Envía el identificador por USB y a continuación recibe la respuesta del microcontrolador.
- *ProcesaADC*: Este método recibe la información de los transductores de presión y la procesa para que refleje el valor leído. Modifica la posición de los indicadores visuales de presión. Las ecuaciones empleadas para la conversión de los datos se muestran en (2.6) y (2.7).

$$Presion1 = (valorCAD1 - 2047) * \frac{15}{2048} \quad (2.4)$$

$$Presion2 = (valorCAD2 - 2047) * \frac{30}{2048} \quad (2.5)$$

Donde *valorCAD1* y *valorCAD2* son las variables que contienen los datos registrados de los sensores PX26-015DV y PX26-030DV, respectivamente.

- *ProcesaPeriodo*: A partir del tiempo medido entre flancos de subida de la señal generada por el sensor de flujo, calcula la velocidad de flujo y el gasto volumétrico a partir de la información del fabricante y el diámetro del aditamento instalado.

A partir de la ecuación (1.10), la velocidad se obtiene con la ecuación (2.8)

$$Velocidad = \frac{f_{osc}}{4 * Prescalador * periodo} * \left(\frac{1}{19.7} \right) \quad (2.6)$$

Donde f_{osc} es la frecuencia de oscilador establecida, *Prescalador* es el factor que divide a la frecuencia de oscilador para el *TIMER0*, y *periodo* es la variable que contiene el valor de la cuenta del *TIMER0* registrado en la interrupción por *CCP1* del microcontrolador. Este resultado está expresado en [m/s]

Para convertir esta información a gasto volumétrico [m³/s], se empleó la ecuación (2.9):

$$Q_{\frac{m^3}{s}} = \frac{Velocidad * \pi * D^2}{4 * 256 * periodo} \quad (2.7)$$

Finalmente, para convertir ésta información a [litros/min], se empleó la siguiente ecuación:

$$Q_{\frac{\text{Litros}}{\text{min}}} = Q_{\frac{\text{m}^3}{\text{s}}} * 1000 * 60 \quad (2.8)$$

Debido a que 1 [m³] son 1000 [litros].

- **ProcesaSetPoint:** Procesa el valor adquirido del ADC al cual está conectado el potenciómetro de la electroválvula para que sea un valor porcentual. La ecuación para convertir el valor adquirido por el ADC a un valor porcentual es la (2.11):

$$\text{Porcentaje de apertura} = \frac{\text{ValorCAD0} - 3075}{-19.75} \quad (2.9)$$

Donde el valor -19.75 es la pendiente de la recta formada a partir de los valores de la Tabla 2.3.

- **USBClosePipes:** Cierra la conexión de entrada y de salida hacia el microcontrolador.
- **USBOpenPipes:** Establece al dispositivo con el que existirá la comunicación USB, además de indicar a que Endpoint se enviará la información. Define a los canales de entrada y salida.
- **USBReceivePacket:** Recibe el paquete de datos enviados del microcontrolador. Se incluye el tamaño de arreglo esperado, la ubicación del arreglo donde se guardará la información y el tiempo de espera máximo para realizar la transferencia.
- **USBSendPacket:** Envía un paquete de datos hacia el microcontrolador. Tiene una estructura muy similar a USBReceivePacket.

CAboutDlg

Los métodos que conforman a CAboutDlg son mostrados en la Figura 2.39.

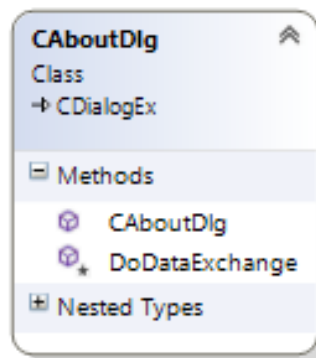


Figura 2.39: Métodos de la Clase CAboutDlg.

Donde:

- **CAboutDlg:** Es el constructor de la clase.
- **DoDataExchange:** Este método es usado para validar e intercambiar la información obtenida en un diálogo.

Capítulo 3

Resultados

En este capítulo se mostrarán los resultados de esta tesis: la electrónica y el software desarrollados. Además, se incluye información de unas pruebas realizadas con este sistema, para evaluar su desempeño. Se discutirá acerca del desempeño del sistema y el trabajo a futuro sobre este proyecto.

3.1 Resultados

Al final del desarrollo, se obtuvo un circuito electrónico capaz de interactuar con un programa desarrollado para visualizar información proveniente de diversos sensores, así como para controlar una bomba hidráulica y una electroválvula. Además, se realizó la comparación de la información de los sensores contra información obtenida por otros medios, y finalmente, se realizó un experimento común en las prácticas de Mecánica de Fluidos: observar la caída de presión y el flujo en una válvula de globo.

3.1.1 Circuito electrónico

El circuito final se presenta en la Figura 3.40:

Donde:

1. Es el circuito de control de la bomba hidráulica.
2. Es la etapa de potencia para el motor de la electroválvula.
3. Corresponde al potenciómetro que indica la posición actual de la electroválvula.
4. Es la etapa de acondicionamiento para los sensores de presión.
5. Es el microcontrolador PIC18F4553.
6. Es el circuito de acondicionamiento para el sensor de flujo.
7. Es la fuente de ± 12 [V] para los circuitos de acondicionamiento.

8. Es la fuente de 12 [V] dedicada a la electroválvula y a la bomba hidráulica.

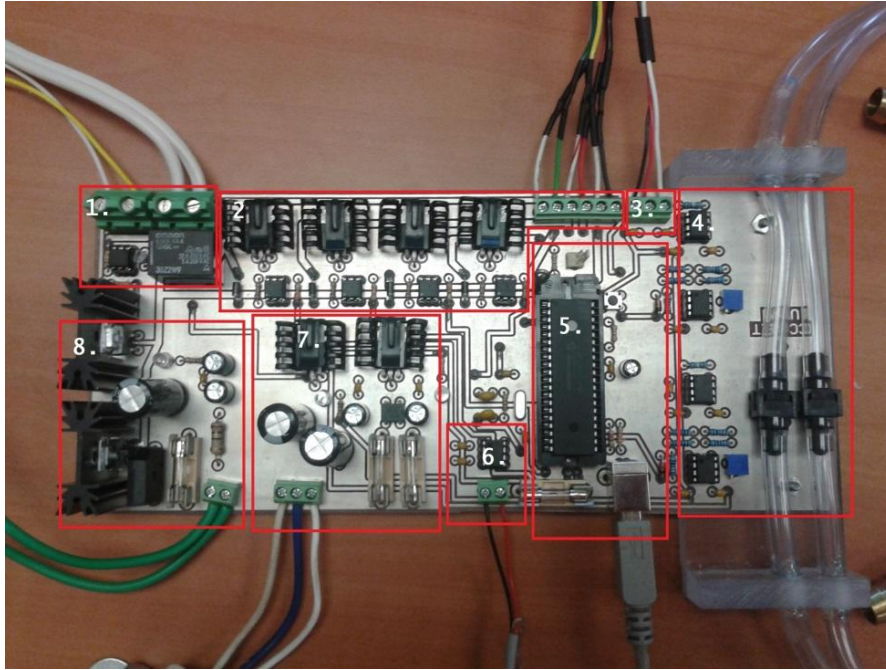


Figura 3.40: Circuito del sistema electrónico.

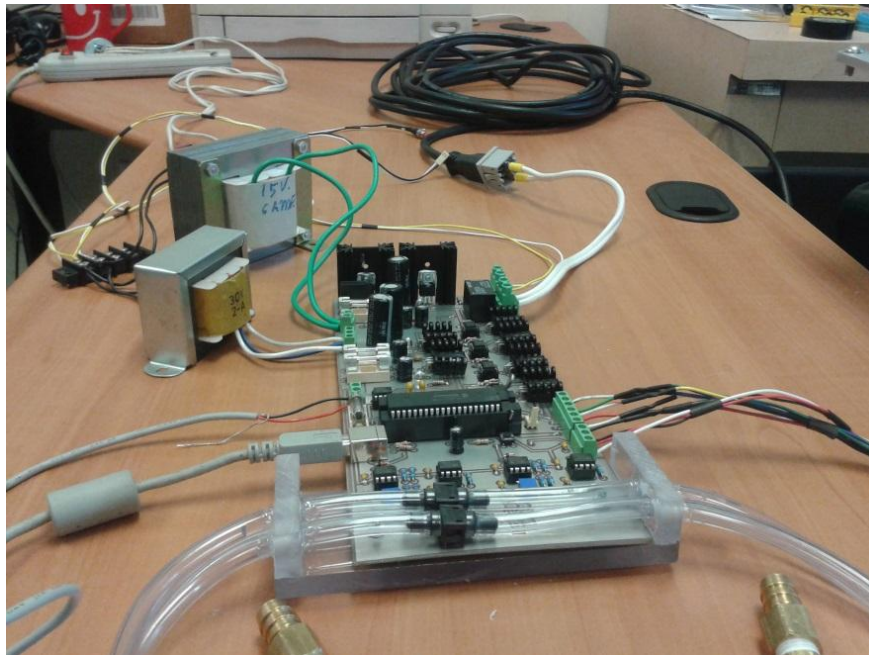


Figura 3.41: Circuito electrónico, con transformadores a la vista.

En la Figura 3.42, se aprecia el circuito energizado

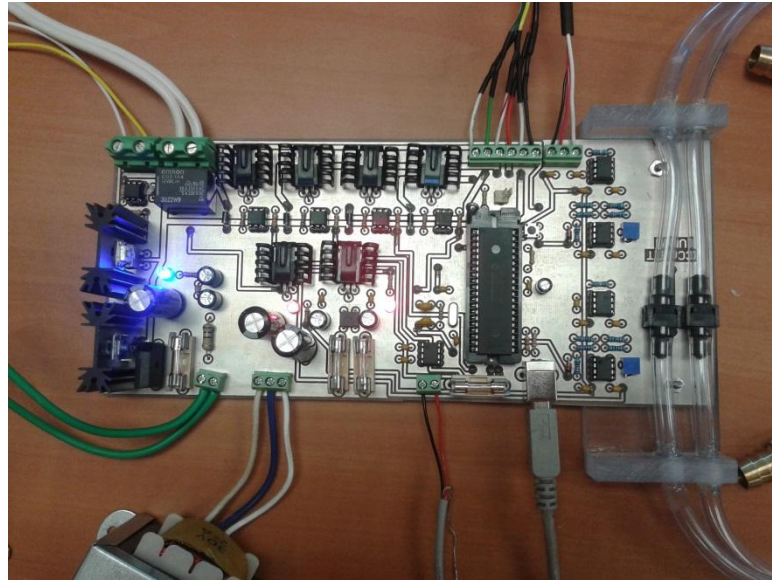


Figura 3.42: Circuito energizado.

La Electroválvula es mostrada en la Figura 3.43. En la parte superior se aprecia el potenciómetro usado para conocer la posición actual de la electroválvula, además de las conexiones del motor a pasos.



Figura 3.43: Electroválvula PV-34B con potenciómetro.

En la Figura 3.44 se pueden ver las especificaciones del motor a pasos. En la esquina superior izquierda, se aprecia la bomba hidráulica.

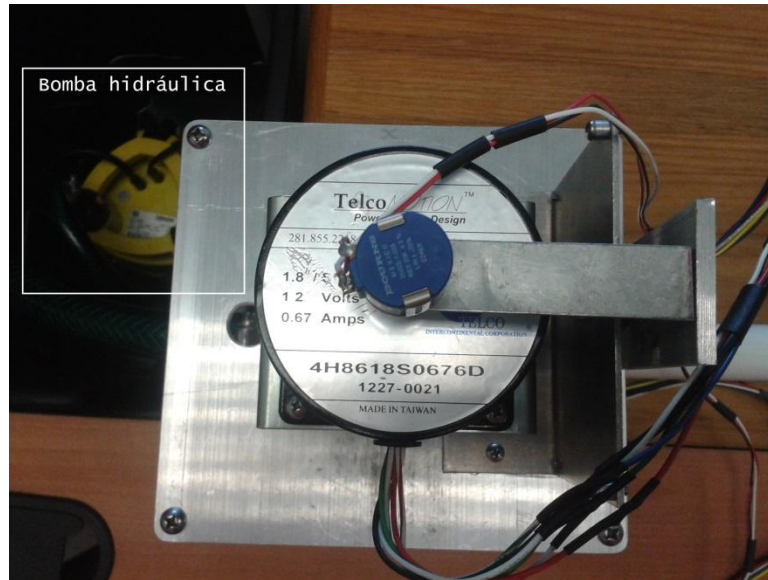


Figura 3.44: Vista superior de la electroválvula.

El sensor de flujo se muestra en la Figura 3.45.



Figura 3.45: Sensor de flujo instalado en circuito de prueba.

3.1.2 Software

Para recuperar los datos registrados por el microcontrolador PIC18F4553 y transformarlos en valores coherentes para el usuario, además de poder controlar el sistema, se desarrolló un programa en Visual C++. Las siguientes imágenes son capturas de pantallas del programa.

En la Figura 3.46, se ve la pantalla principal del programa en funcionamiento.

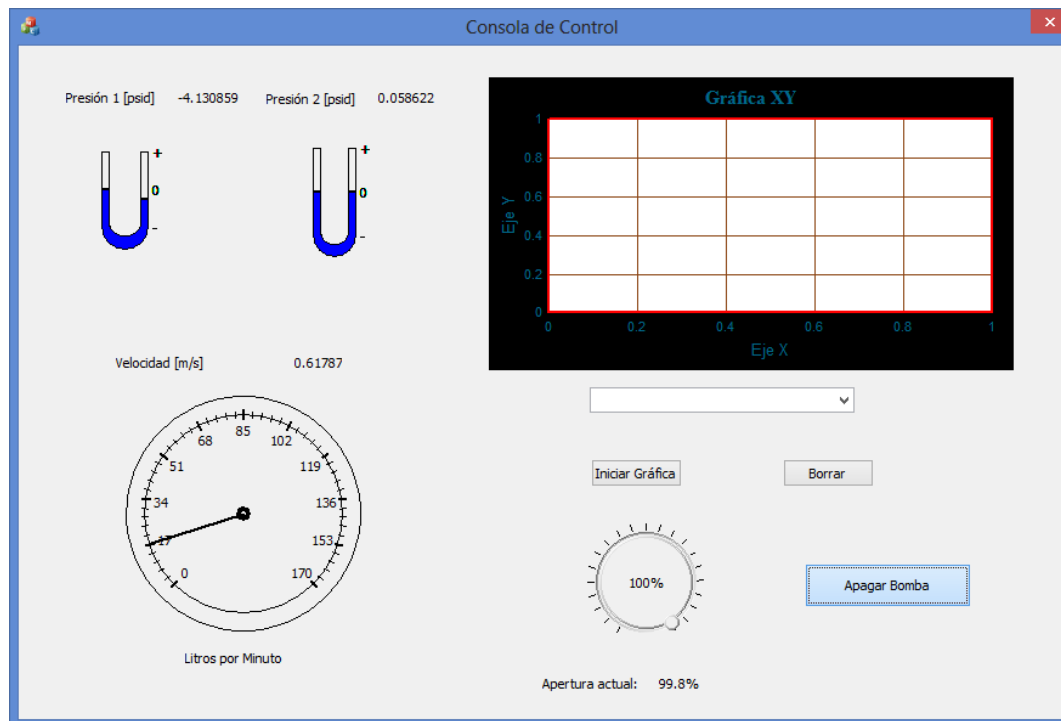


Figura 3.46: Programa de control en ejecución.

En la Figura 3.47, se muestran las opciones para graficar.

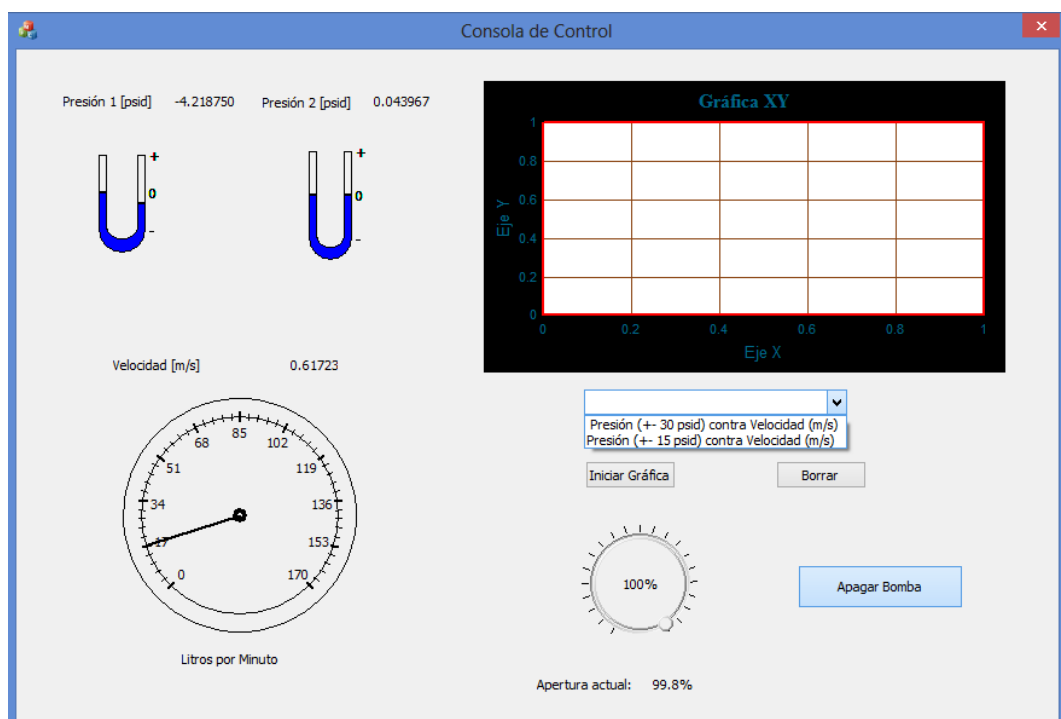


Figura 3.47: Opciones para graficar disponibles.

Y en la Figura 3.48, se muestra el programa una vez que ha almacenado los datos y graficado la información.

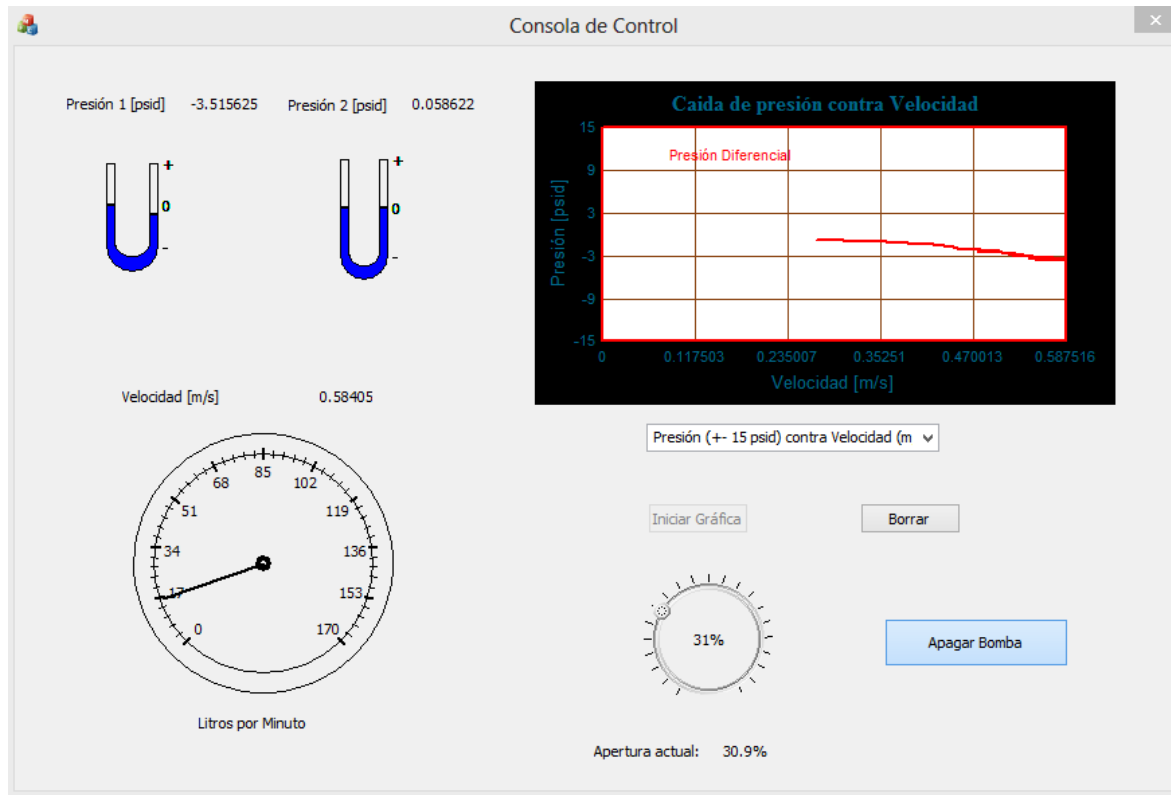


Figura 3.48: Programa después de graficar datos.

3.1.3 Calibración

Se realizaron dos procesos de calibración para evaluar el desempeño de los dispositivos de medición utilizados en el sistema electrónico desarrollado (sensores de presión y flujo).

A continuación se presentan las pruebas realizadas.

Presión

Para la calibración del sensor de presión, se empleó un compresor de aire de 2.5 [hp] de Adir Herramientas y el manómetro incluido en éste como instrumento de referencia. Como se puede ver en la figura 3.10, la división mínima del manómetro de referencia es de 5 [psi] y su alcance de medición es de 200 [psi]. Comparando estas características con los alcances de medición en el sistema electrónico a calibrar (± 15 psid y ± 30 psid) observamos que se trabajará en el extremo inferior del manómetro de referencia, lo cual ocasionará cierta incertidumbre, ya que representa una zona difícil de observar a simple vista.

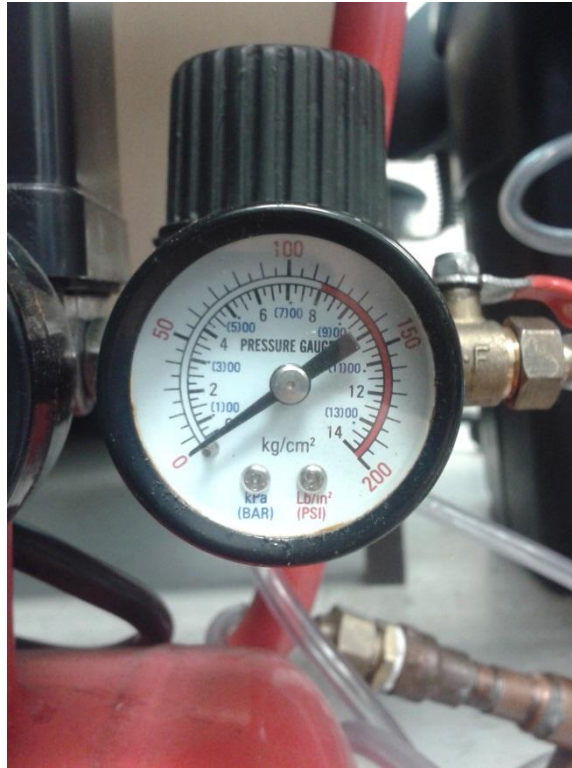


Figura 3.49: Manómetro de referencia usado para calibración.

La prueba consiste en colocar el compresor a una presión de referencia, la cual se registra con el manómetro de referencia en uno de los extremos del sensor piezorresistivo del sistema electrónico y el otro extremo del sensor dejarlo abierto a la presión atmosférica. La presión de referencia se regula manualmente mediante la válvula de estrangulamiento del compresor. En la Tabla 3.4 y la Tabla 3.5 se muestran las mediciones realizadas, donde:

- P_M es la lectura tomada del manómetro mecánico, que se usa como referencia. Es expresada en [psi].
- P_S es el promedio de las lecturas realizadas con el sistema electrónico. Es expresada en [psi].
- *Error* es la diferencia entre P_M y P_S .
- σ es la desviación estándar de cada familia de lecturas.

P_M	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	P_S	Error	σ
-15	-14.99	-15	-15	-14.99	-14.99	-14.994	-0.006	0.253
-14	-14.5	-14.99	-14.99	-14.53	-14.39	-14.68	0.68	0.163
-12	-13.11	-13.15	-13.01	-12.78	-13.16	-13.042	1.042	0.377
-10	-10.72	-11.06	-11.09	-10.56	-10.56	-10.798	0.798	0.179
-8	-8.8	-8.48	-8.64	-8.76	-8.85	-8.706	0.706	0.473
-6	-7.06	-7.05	-6.66	-6.59	-6.34	-6.74	0.74	0.181
-4	-4.62	-5.02	-4.79	-4.63	-4.58	-4.728	0.728	0.022
-2	-2.18	-2.9	-2.38	-2.12	-2.73	-2.462	0.462	0.339
0	0.02	0.01	-0.03	0.007	-0.04	-0.0066	0.0066	0.456
2	2.44	2.52	2.59	2.23	2.45	2.446	-0.446	0.202
4	5.03	4.68	4.62	4.53	4.62	4.696	-0.696	0.205
6	6.58	6.28	6.34	6.79	6.72	6.542	-0.542	0.184
8	8.28	8.38	8.13	8.53	8.54	8.372	-0.372	0.124
10	9.8	10.51	10.18	10.01	10.25	10.15	-0.15	0.253
12	11.25	11.34	11.33	11.45	11.34	11.342	0.658	0.163
14	11.95	12.2	11.96	12.12	12.2	12.086	1.914	0.377
15	12.38	12.62	12.43	12.6	12.7	12.546	2.454	0.179

Tabla 3.4: Prueba con PX26-015DV (± 15 [psid]).

P_M	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	P_S	Error	σ
-30	-29.5	-29.5	-29.97	-30	-29.9	-29.774	-0.226	0.005
-25	-26.42	-26.7	-26.39	-26.3	-26.6	-26.482	1.482	0.288
-20	-20.22	-21.07	-20.56	-20.12	-20.35	-20.464	0.464	0.158
-15	-15.72	-15.67	-15.65	-15.79	-16.09	-15.784	0.784	0.261
-10	-10.43	-10.33	-9.37	-9.55	-9.73	-9.882	-0.118	0.148
-5	-5.01	-4.77	-4.61	-4.9	-5.05	-4.868	-0.132	0.311
0	-0.01	-0.02	0.01	-0.05	-0.01	-0.016	0.016	0.182
5	4.7	4.95	5.46	5.23	5.49	5.166	-0.166	0.342
10	9.68	9.75	10.62	10.59	9.95	10.118	-0.118	0.027
15	15.09	14.78	14.99	14.96	14.58	14.88	0.12	0.135
20	18.52	18.5	18.23	18.51	18.07	18.366	1.634	0.194
25	21.26	20.94	20.94	20.78	20.85	20.954	4.046	0.226
30	22.76	22.65	22.58	22.48	22.46	22.586	7.414	0.173

Tabla 3.5: Prueba con PX26-030DV (± 30 [psid]).

La Figura 3.50 y la Figura 3.51 muestran las gráficas de presión de referencia contra el error calculado.

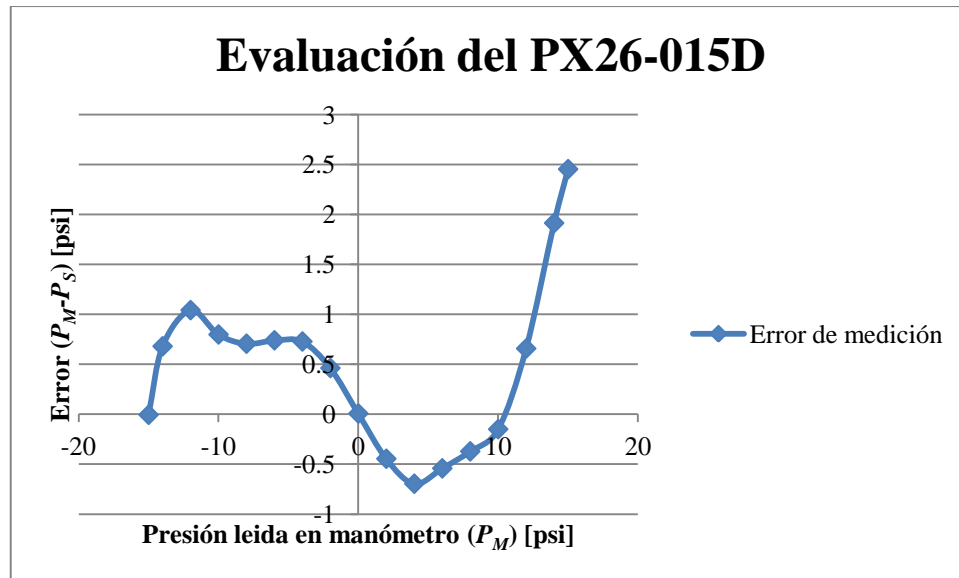


Figura 3.50: Evaluación del PX26-015D contra Manómetro mecánico.

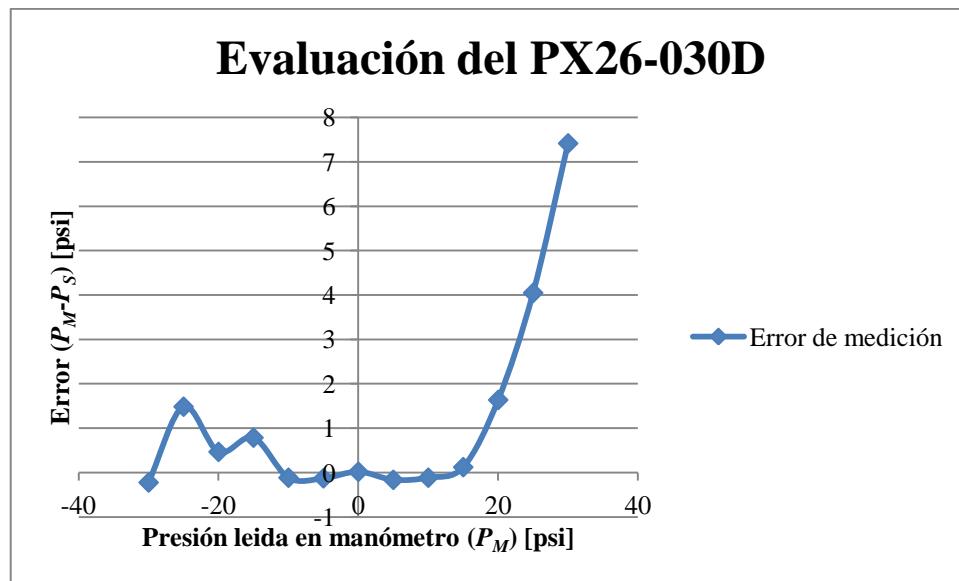


Figura 3.51: Evaluación del PX26-030D contra Manómetro mecánico.

En ambas gráficas, se identificaron dos regiones: una donde el comportamiento varía menos al 10% y otra región, menos confiable, donde se excede ese 10%.

Se aprecia una desviación estándar máxima de 0.456 [psi] en los datos medidos del PX26-015D, y de 0.342 [psi] para los datos del PX26-030D.

Para analizar este comportamiento anómalo, se realizó un experimento que tiene por objetivo observar detalladamente la etapa de acondicionamiento de señal para el sensor de presión. El experimento consiste en medir los voltajes a la salida del amplificador sumador del circuito de acondicionamiento para el PX26-030D (V_X) y en la entrada del canal de conversión del microcontrolador, es decir, en el cátodo del

diodo Zener (V_2), como se muestra en la Figura 3.52. Debido a que el PX26-015D y el PX26-030D poseen el mismo circuito, sólo se hizo la prueba con un solo sensor.

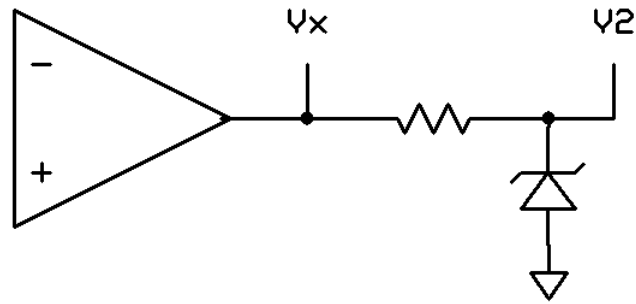


Figura 3.52: Comparación de V_x y V_2 .

En la Tabla 3.6, se muestran los valores registrados para el experimento anteriormente descrito.

P_M [psi]	V_x [V]	V_2 [V]	Variación de voltajes $ V_x - V_2 $
-30	-0.45	-0.36	0.09
-25	0.173	0.171	0.002
-20	0.56	0.57	0.01
-15	1.36	1.36	0
-10	1.45	1.54	0.09
-5	2.068	2.02	0.048
0	2.497	2.493	0.004
5	3.059	3.058	0.001
10	3.39	3.37	0.02
15	3.96	3.81	0.15
20	4.395	4.045	0.35
25	4.94	4.27	0.67
30	5.27	4.36	0.91

Tabla 3.6: Datos recolectados de la medición de V_x y V_2 para una presión específica.

De la Tabla 3.6 se aprecian variaciones muy marcadas en 15, 20, 25 y 30 [psi], lo que indica que el diodo Zener está conduciendo, distorsionando en este rango el voltaje que registra el microcontrolador.

Flujo

Para ésta prueba se construyó un circuito hidráulico para probar la funcionalidad del sistema. Se presenta en la Figura 3.53.



Figura 3.53: Circuito hidráulico de prueba.

El circuito consiste de una bomba hidráulica Karcher SCP 12000 (0.7 [bar], 200 [litros/s]), una electroválvula, un sensor de flujo, dos codos rectos, tres conectores rápidos, una válvula de globo de 1'', una válvula de bola de 1/2'', y dos adaptadores de tubo de 1'' a tubo de 1/2''. En la Figura 3.54 se muestran las mangueras del sensor de presión conectadas en ambos extremos de una válvula de globo.

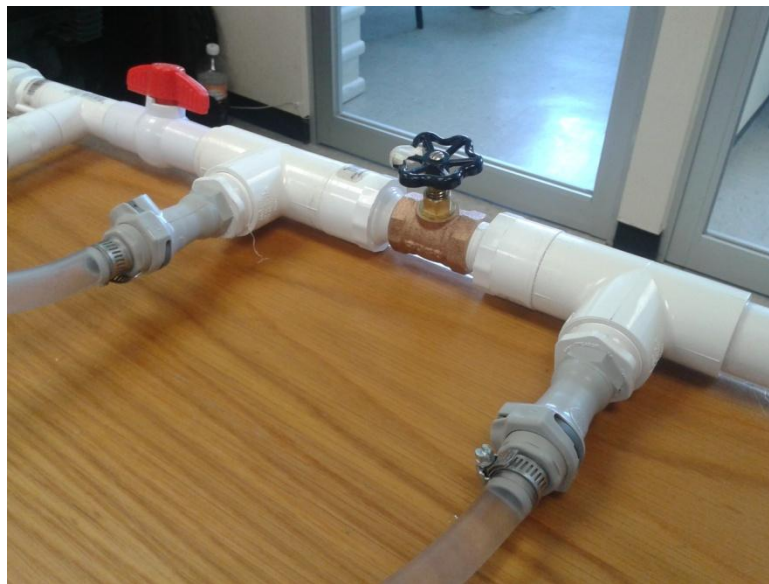


Figura 3.54: Válvula utilizada para pruebas..

Para apreciar el desempeño del sensor de flujo, se realizó un experimento que consiste en recolectar el agua que atraviesa el circuito hidráulico durante cierto tiempo, y estimar el gasto másico de referencia (\dot{m}_{ref}) al pesar la cantidad de agua recolectada y compararlo contra el gasto másico determinado por el sistema electrónico (\dot{m}_{med}).

El gasto másico de referencia se determina por la ecuación (3.1).

$$\dot{m}_{\text{ref}} = \frac{P}{T} \quad (3.14)$$

Donde:

- P es el peso medido en kilogramos.
- T es el tiempo en segundos transcurrido durante el llenado del recipiente.

La relación del gasto másico con la velocidad de flujo determinada por el sistema electrónico se da por la siguiente ecuación

$$\dot{m}_{\text{med}} = \rho * v * A \quad (3.15)$$

Donde:

- \dot{m}_{med} es el gasto másico experimental.
- ρ es la densidad del fluido.
- v es la velocidad del fluido determinada por el sistema electrónico.
- A es el área de la tubería donde se realiza la medición.

Se hizo esto en dos condiciones de variación de flujo: colocando la válvula de globo (Figura 3.54) abierta al 60% y abierta al 100%. El resto de las válvulas están completamente abiertas. Se realizaron veinte mediciones en este experimento.

Se toma a ρ como $1000 \text{ [kg/m}^3\text{]}$, y el área de la tubería A es $0.000464068 \text{ [m}^2\text{]}$

Los resultados se muestran en las Tablas 3.4 y 3.5 donde el *Error* está definido por la siguiente ecuación:

$$\text{Error} = \frac{|\dot{m}_{\text{med}} - \dot{m}_{\text{ref}}|}{|\dot{m}_{\text{med}}|} * 100 \text{ [%]} \quad (3.1)$$

Número de Prueba	P [kg]	T [s]	\dot{m}_{med} [kg/s]	v [m/s]	\dot{m}_{ref} [kg/s]	Error [%]
1	8.16	35	0.23314286	0.472	0.21903988	6.04907196
2	8.02	35.2	0.22784091	0.461	0.21393514	6.10328238
3	8.49	35.9	0.23649025	0.47	0.21811174	7.77135961
4	8.185	36.2	0.22610497	0.461	0.21393514	5.38238377
5	8.125	36.7	0.22138965	0.456	0.2116148	4.41522385
6	8.12	36.5	0.22246575	0.44	0.20418972	8.21521354
7	8.25	37.5	0.22	0.446	0.20697412	5.92085366
8	8.21	36.8	0.22309783	0.45	0.20883039	6.39514702
9	8.29	38.5	0.21532468	0.43	0.19954904	7.32644039
10	8.345	38.4	0.21731771	0.44	0.20418972	6.04092124
11	8.395	35.6	0.23581461	0.48	0.22275242	5.53917701

Tabla 3.7: Prueba de sensor de flujo, con válvula de globo al 60% de apertura.

Número de Prueba	P [kg]	T [s]	\dot{m}_{med} [kg/s]	v [m/s]	\dot{m}_{ref} [kg/s]	Error [%]
12	8.425	35.9	0.23467967	0.467	0.21671954	7.65303863
13	8.275	36.5	0.22671233	0.46	0.21347107	5.84055633
14	8.33	36.8	0.2263587	0.458	0.21254293	6.10348244
15	8.225	36.4	0.22596154	0.466	0.21625547	4.29545037
16	8.225	35.8	0.2297486	0.48	0.22275242	3.04514804
17	8.39	36.5	0.22986301	0.47	0.21811174	5.11229303
18	8.45	37.4	0.22593583	0.46	0.21347107	5.51694764
19	8.635	38.7	0.22312661	0.45	0.20883039	6.40722438
20	8.155	38.1	0.21404199	0.44	0.20418972	4.60296496

Tabla 3.4 (Continuación).

De la ¡Error! No se encuentra el origen de la referencia. se aprecia que el error promedio es de 5.88%, el error máximo registrado es 8.21%, y el mínimo de 3.04%. Los datos obtenidos del sensor poseen una desviación estándar de 0.01386884 [m/s].

Número de Prueba	P [kg]	T [s]	\dot{m}_{med} [kg/s]	v [m/s]	\dot{m}_{ref} [kg/s]	Error [%]
1	8.57	25.4	0.33740157	0.71	0.32948795	2.34546114
2	8.3	24.4	0.34016393	0.72	0.33412863	1.77423484
3	8.67	25.7	0.33735409	0.72	0.33412863	0.95610469
4	8.62	25.3	0.34071146	0.72	0.33412863	1.932085
5	8.63	25.5	0.33843137	0.72	0.33412863	1.27137893
6	8.755	25.7	0.34066148	0.7	0.32484728	4.64220435
7	8.4	24.8	0.33870968	0.71	0.32948795	2.72260465
8	8.28	24.4	0.33934426	0.713	0.33088015	2.49425396
9	8.47	25.2	0.33611111	0.73	0.3387693	0.79086694
10	8.33	24.5	0.34	0.71	0.32948795	3.09177882
11	8.54	25.3	0.33754941	0.72	0.33412863	1.01341601
12	8.54	25.3	0.33754941	0.72	0.33412863	1.01341601
13	8.51	25.1	0.33904382	0.71	0.32948795	2.81847714
14	8.21	25.4	0.32322835	0.7	0.32484728	0.50086268
15	8.42	24.7	0.34089069	0.71	0.32948795	3.3449832
16	8.65	25.5	0.33921569	0.72	0.33412863	1.4996532
17	8.32	24.6	0.33821138	0.71	0.32948795	2.57928342
18	8.34	24.4	0.34180328	0.71	0.32948795	3.60304522
19	8.34	24.9	0.33493976	0.71	0.32948795	1.62769779
20	8.6	25.5	0.3372549	0.71	0.32948795	2.30299098

Tabla 3.8: Prueba de sensor de flujo, con válvula de globo al 100% de apertura.

De la Tabla 3.8, se aprecia que el error promedio es 2.11%, el error máximo es de 4.64% y el mínimo es de 0.5%. Los datos obtenidos del sensor poseen una desviación estándar de 0.0074075 [m/s].

Comparando ambas tablas, se aprecia un porcentaje de error menor a mayor porcentaje de apertura en la válvula de globo, es decir, a mayor velocidad de flujo. Esto es apreciable en la Figura 3.55.

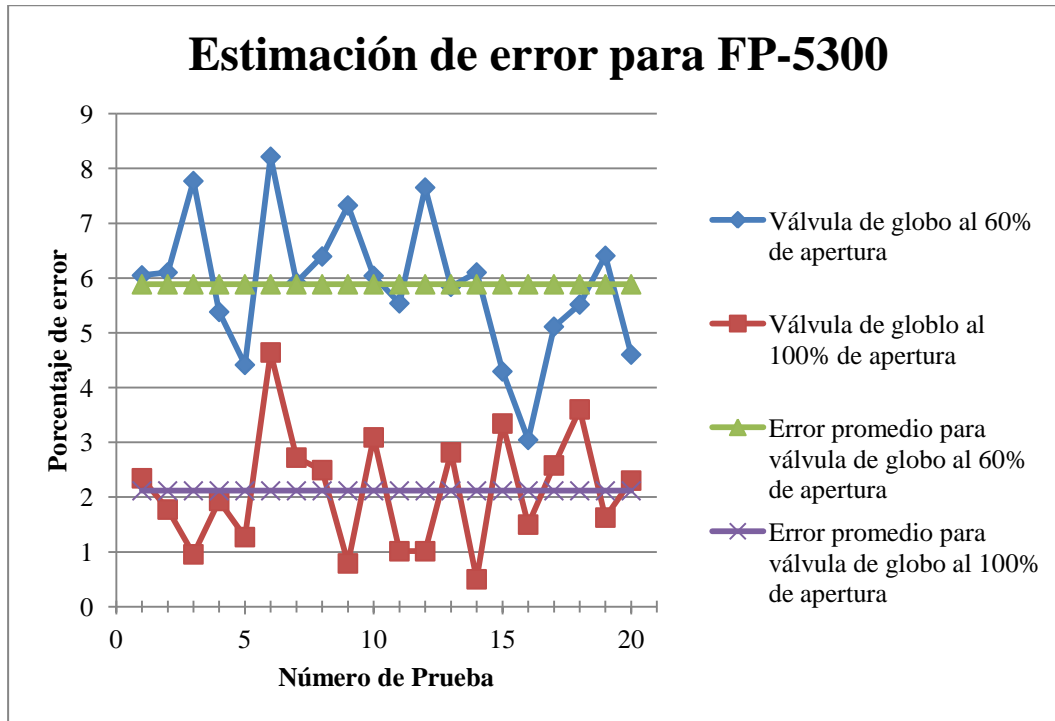


Figura 3.55: Estimación de error para el sensor de flujo FP-5300.

3.1.4 Experimento

Este experimento tiene como base la Práctica #8 de la materia “Mecánica de Fluidos I” de la carrera de Ingeniería Mecánica.

El primer experimento consistió en medir la diferencia de presión en la válvula de globo de la figura 3.15, contra el porcentaje de apertura de la electroválvula, para diferentes porcentajes de apertura de la válvula de globo. La válvula de bola se coloca en posición abierta. La gráfica obtenida se observa en la Figura 3.56. El segundo experimento consistió en medir la velocidad de flujo en el circuito hidráulico de la figura 3.14, contra el porcentaje de apertura de la electroválvula, para diferentes porcentajes de apertura de la válvula de globo. La válvula de bola se coloca en posición abierta. La gráfica obtenida se observa en la Figura 3.56.

De la Figura 3.56, se observa que a partir del 30% de apertura de la electroválvula, la válvula de globo domina el flujo de agua. A partir de que la válvula de globo está abierta en un 40%, el periodo de la señal proveniente del sensor de flujo excede a 1.3981 [s] (0.7152 [Hz]) y no es leída por el

microcontrolador, además, el fabricante indica que el rango de lectura del es de 0.3 [m/s] a 6 [m/s] con una frecuencia nominal de 19.7 [Hz] por cada [m/s]. Esto quiere decir que la menor frecuencia generada por el dispositivo es aproximada a 5.9 [Hz]. Cualquier valor menor a esto, está fuera del rango especificado por Omega.

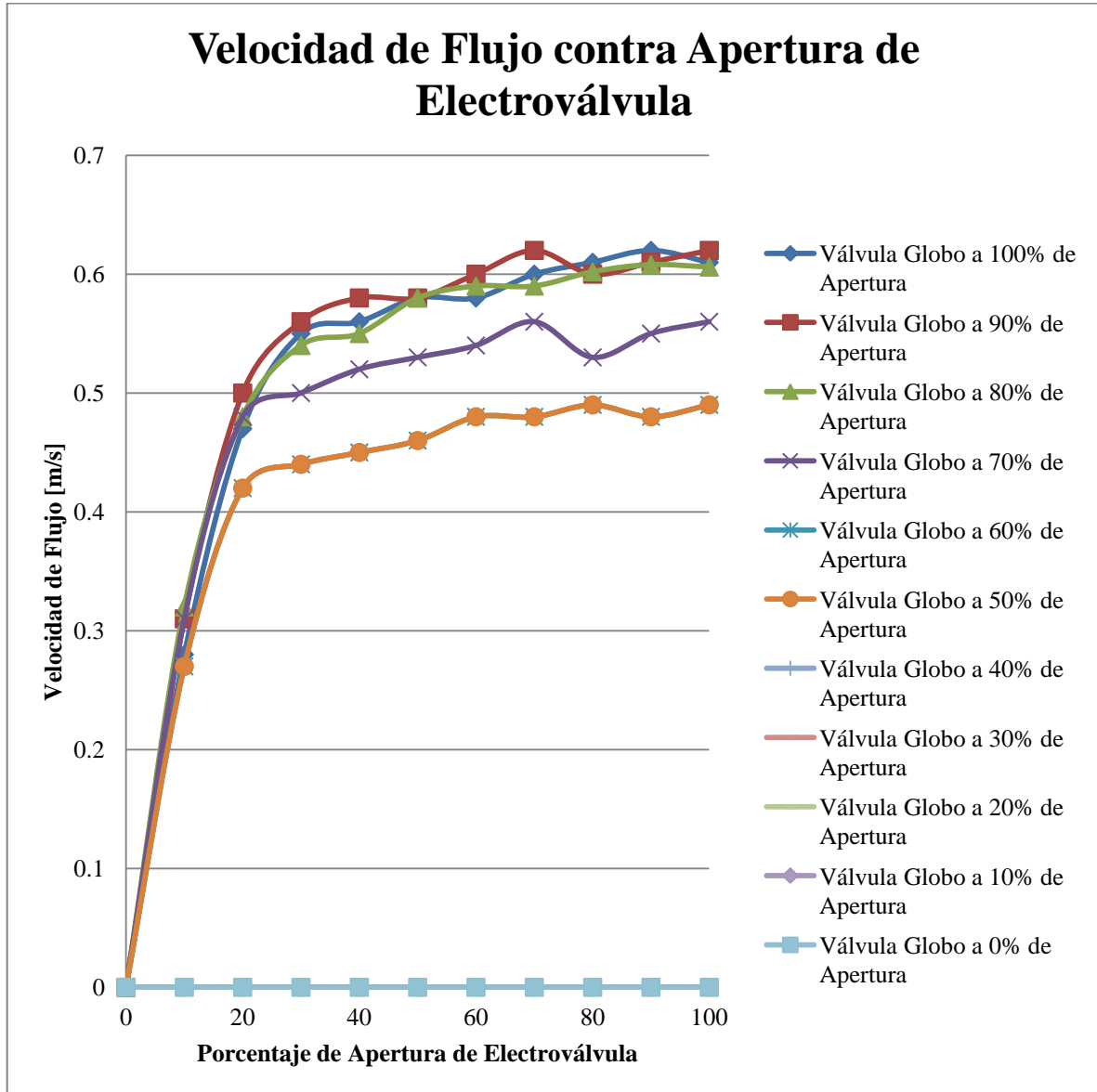


Figura 3.56: Velocidad de Flujo contra Apertura de Electroválvula.

De la Figura 3.57, se aprecia un incremento en la presión diferencial en la válvula de globo en función de la apertura de la electroválvula, hasta alrededor del 30%, donde la presión diferencial se queda casi constante debido al fenómeno explicado anteriormente. A partir de 40%, la presión se mantiene en un nivel constante, gracias a que la válvula de globo domina el flujo de agua en el circuito.

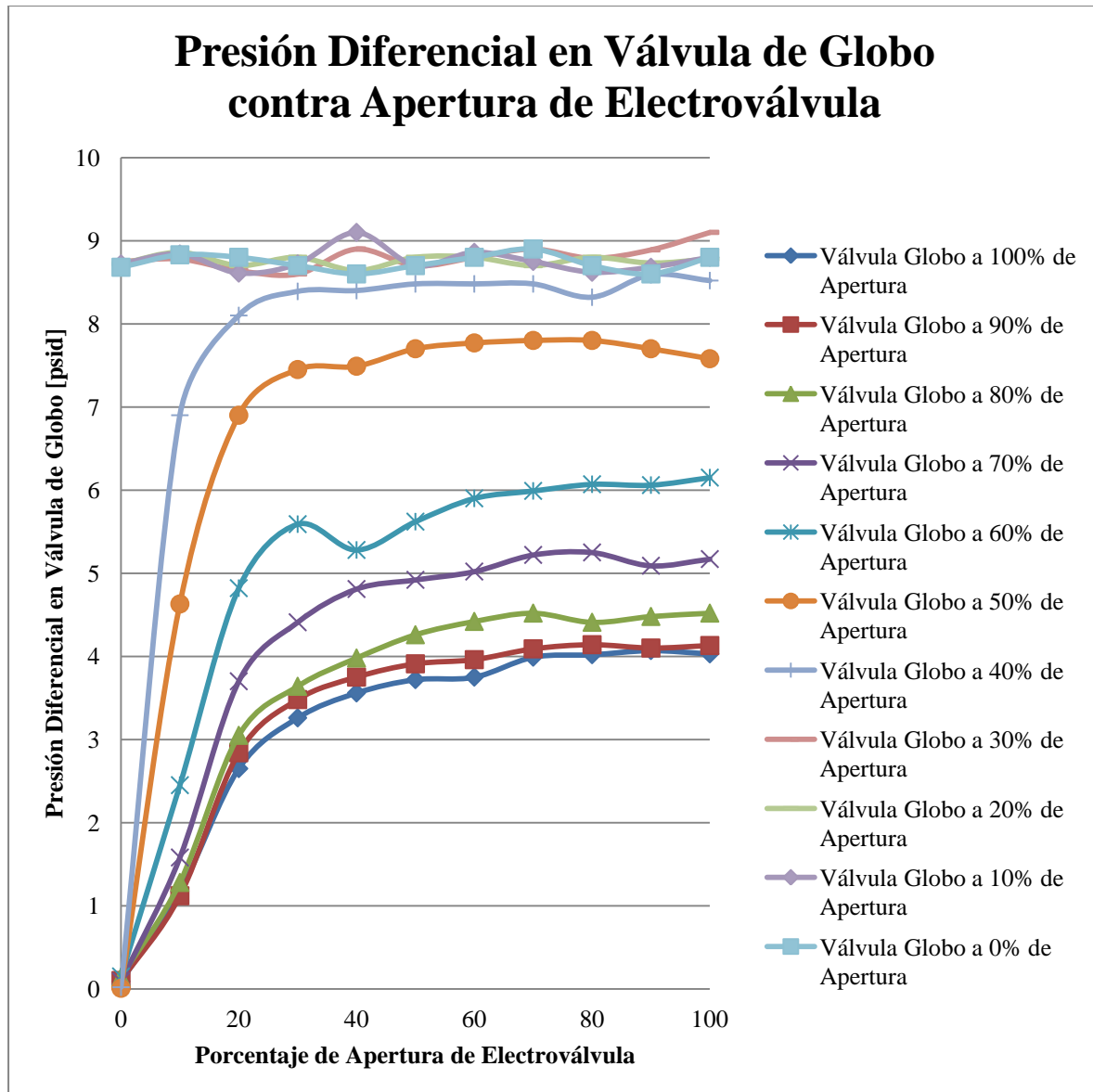


Figura 3.57: Presión Diferencial en Válvula globo contra Apertura de Electroválvula.

3.2 Discusión

El sistema electrónico desarrollado está pensado para un banco de pruebas hidráulico similar al encontrado en el Laboratorio de Termofluidos de la Facultad de Ingeniería. En los experimentos diseñados como parte de este trabajo, se emplea una versión simplificada de circuito hidráulico que no corresponde al equipo con el que cuenta el Laboratorio de Termofluidos. Sin embargo, las mediciones realizadas de presión y flujo con el sistema electrónico desarrollado, son confiables ya que de los resultados de calibración se obtuvo una desviación estándar máxima de 0.456 [psi]

para el PX26-015DV, de 0.342 [psi] para el PX26-030DV, y de 0.01386884 [m/s] para el FP-5300, lo que asegura la reproducibilidad de las mediciones. Adicionalmente, los resultados obtenidos comprueban los parámetros especificados por los fabricantes de los sensores de presión y flujo. En nuestro concepto, el comportamiento del sistema electrónico debe ser igualmente confiable en el momento que se desarrolle un circuito hidráulico con mayor número de componentes o ramas, semejante al encontrado en el Laboratorio de Termofluidos.

No se agregaron rutinas de automatización para la realización de experimentos del Laboratorio, ya que la intención del sistema es acercar a los estudiantes a la tecnología actual de instrumentación, no exentarlos de su trabajo en el desarrollo de las prácticas.

En caso de que los niveles de flujo sean consistentemente menores a lo que el sensor de flujo puede registrar, se recomienda emplear una bomba hidráulica de mayor presión que la actualmente instalada.

Para compensar el error causado por los diodos Zener en los circuitos de acondicionamiento de los sensores de presión, es posible emplear técnicas de linealización mediante software, sustituir el diodo Zener por uno cuyo voltaje Zener sea mayor, o investigar otras técnicas de limitación de voltaje para este circuito.

Los datos obtenidos en las mediciones de flujo y de presión demuestran la reproducibilidad de las mediciones. Las pruebas realizadas muestran el comportamiento del sistema con respecto a los métodos e instrumentos disponibles en el laboratorio.

La versatilidad del sistema electrónico desarrollado abre la puerta a que éste pueda ser usado como base para otros sistemas de medición y control hidráulicos. El diseño y construcción de un circuito para tales condiciones proveen de la experiencia necesaria para trabajar en sistemas más complejos. Además, con el software desarrollado, es posible extenderlo a aplicaciones más complejas, como puede ser un sistema de control realimentado.

3.3 Trabajo a futuro

El trabajo a futuro para este sistema se presenta en los siguientes puntos:

Elaborar un gabinete para el circuito realizado

Este gabinete debe permitir que las conexiones eléctricas e hidráulicas sean rápidas y seguras. Además, éste no debe ser abierto por el usuario del equipo, sólo por el personal capacitado para darle mantenimiento.

Construir el banco de pruebas hidráulico

Se requiere construir el banco de pruebas hidráulico para realizar las prácticas 7, 8 y 9 del laboratorio de la materia Mecánica de Fluidos I, el cual contiene diversos tramos de tubería, válvulas y accesorios comunes en los circuitos hidráulicos.

Instalación del prototipo funcional

Una vez construido el banco de pruebas, es necesario instalar el sistema hidráulico y electrónico en el Laboratorio de Termofluidos. Además, es necesario instalar el programa del sistema en una computadora, siguiendo los pasos descritos en el apartado 2.3.1 de esta tesis.

Como el programa utiliza pocos recursos computacionales para la operación, se puede utilizar una computadora de bajo costo para el sistema. Se recomienda que ésta tenga dos puertos USB: uno para conectar el circuito electrónico y otro para conectar una memoria extraíble, en caso de que el estudiante desee extraer los datos recolectados en el sistema.

Capacitar a los responsables de la operación del equipo

Es necesario entrenar al personal responsable para el buen funcionamiento del equipo. Además de instruirlos en el uso del sistema, hay que informarlos acerca de las precauciones que hay que tomar para evitar problemas.

Estas precauciones son:

- Asegurarse de que las mangueras conectadas al sistema electrónico no tengan fugas.
- Asegurarse de que no existan fugas en los ramales del banco hidráulico.
- Verificar que las conexiones eléctricas no tengan daños.

Se puede energizar al sistema electrónico antes de conectarlo a la computadora, ya que el sistema está diseñado para que no realice acción alguna hasta que sea reconocido por el sistema operativo.

Conclusiones

Se diseñó y construyó un sistema electrónico para un banco hidráulico de pruebas para determinar caídas de presión en accesorios con fines didácticos. El sistema desarrollado es capaz de obtener información necesaria para el desarrollo de las prácticas de la materia Mecánica de Fluidos I, además que le permite a los estudiantes conocer la instrumentación electrónica usada hoy en día, ya que en la industria, éstos dispositivos no se encuentran a simple vista. Además, la electrónica y el software desarrollados dan pie al desarrollo de aplicaciones propias mucho más complejas para uso de las instituciones educativas.

Al no agregarse rutinas de automatización para la realización de los experimentos del Laboratorio de Termofluidos, se logra acercar al estudiante a la tecnología actual de instrumentación, sin exentarlos de su trabajo en el desarrollo de las prácticas.

Los experimentos realizados comprueban su buen desempeño y viabilidad para incorporar al sistema electrónico como parte de circuitos hidráulicos más complejos.

Del análisis de la información recolectada en los experimentos realizados, se identificaron oportunidades de mejora para trabajos futuros.

Anexo A

Diagramas esquemáticos

En este anexo se presentan los diagramas esquemáticos elaborados para la construcción del sistema electrónico.

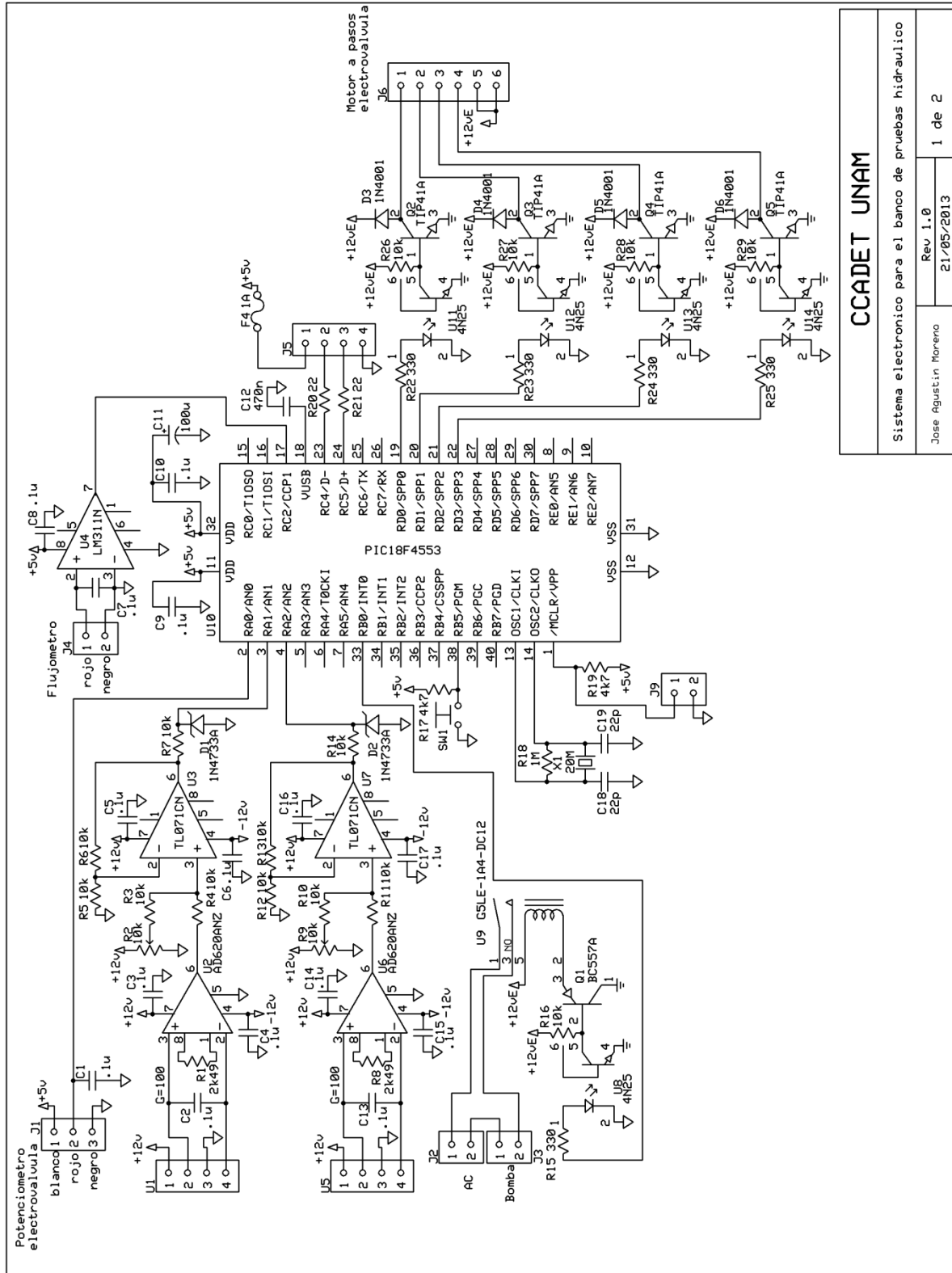
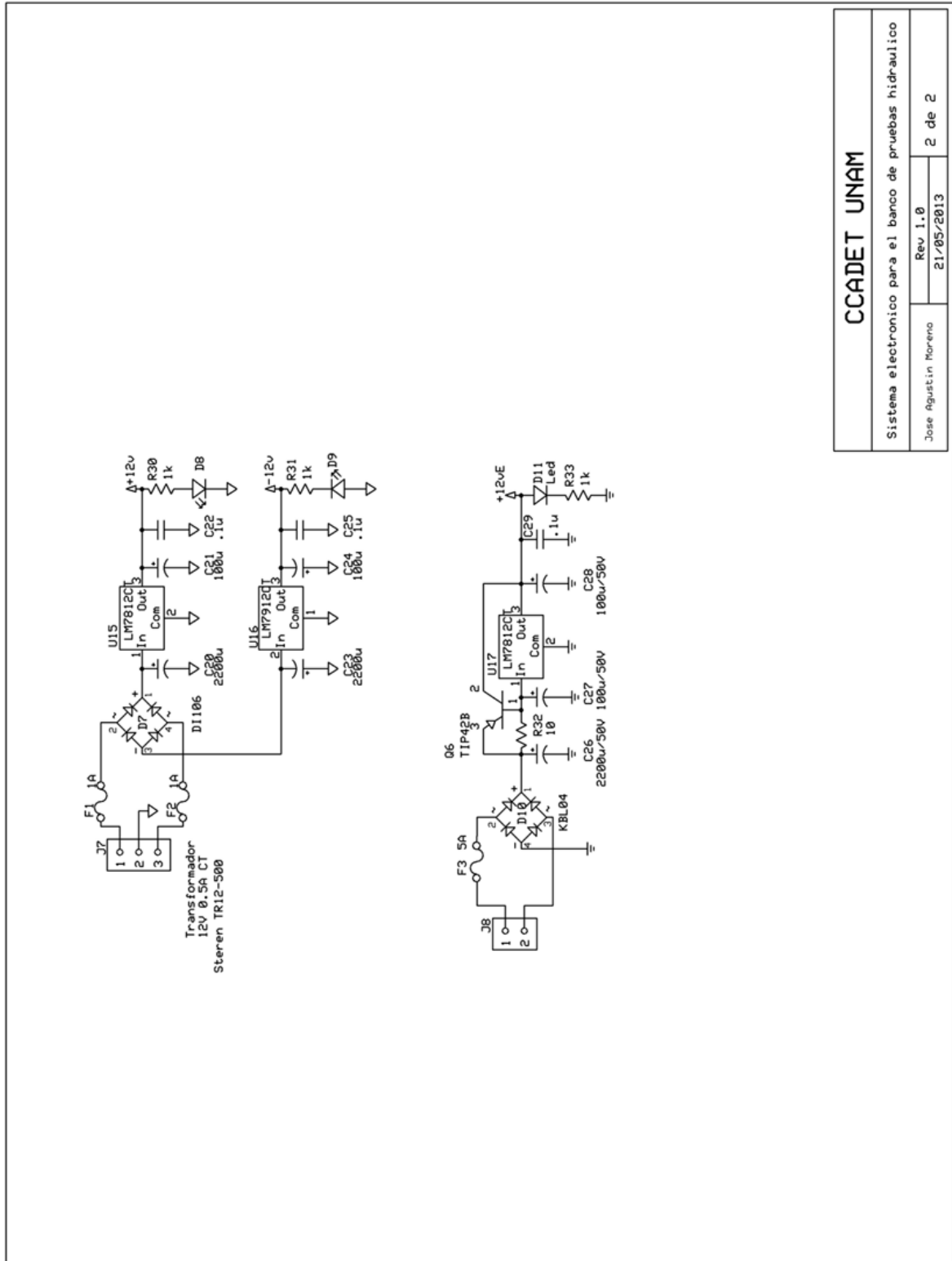


Figura A.1: Diagramas esquemáticos de las fuentes de alimentación.



CCADET UNAM

Sistema electrónico para el banco de pruebas hidráulico

Jose Agustín Moreno

Rev 1.0

21/05/2013

2 de 2

Figura A.2: Diagrama del circuito principal.

Anexo B

Circuito Impreso

En este anexo se presentan las caras del circuito impreso fabricado.

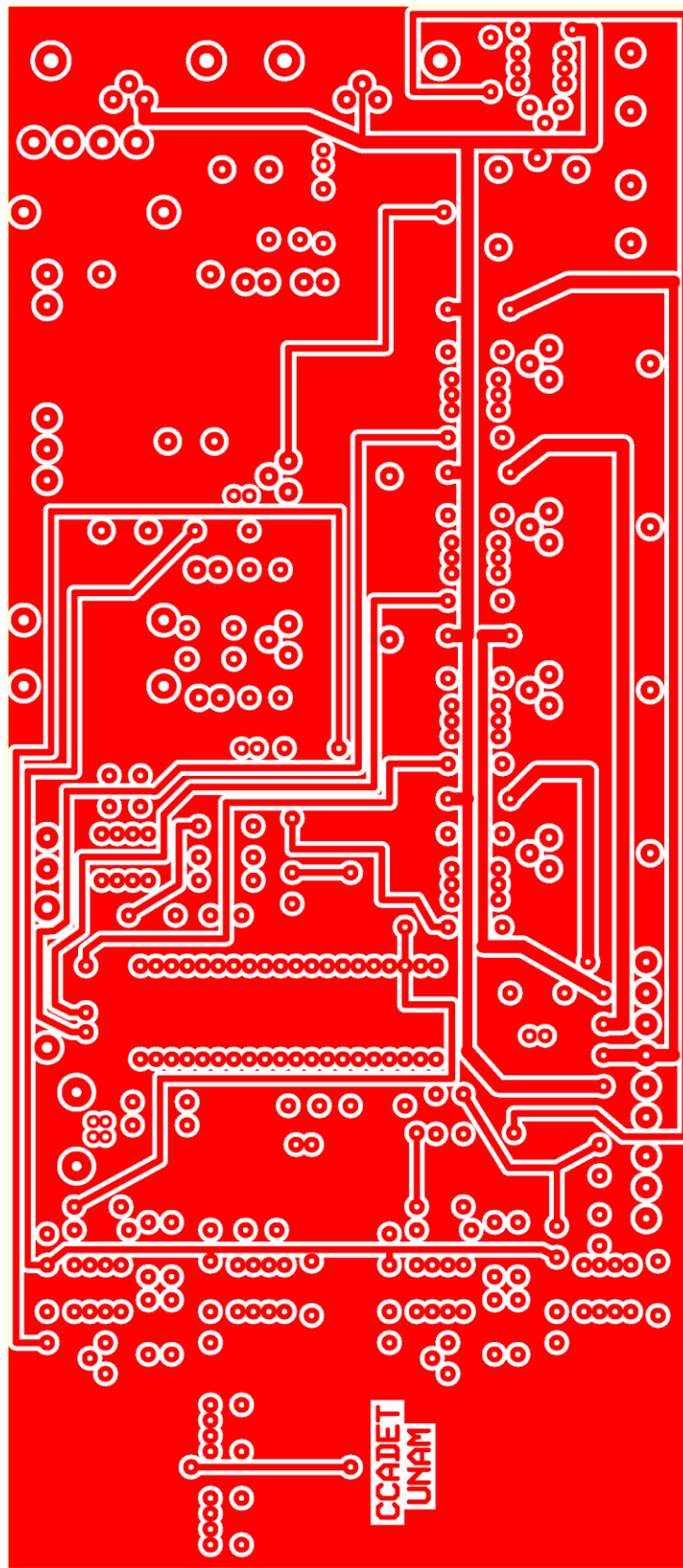


Figura B.1: Capa superior del circuito impreso.

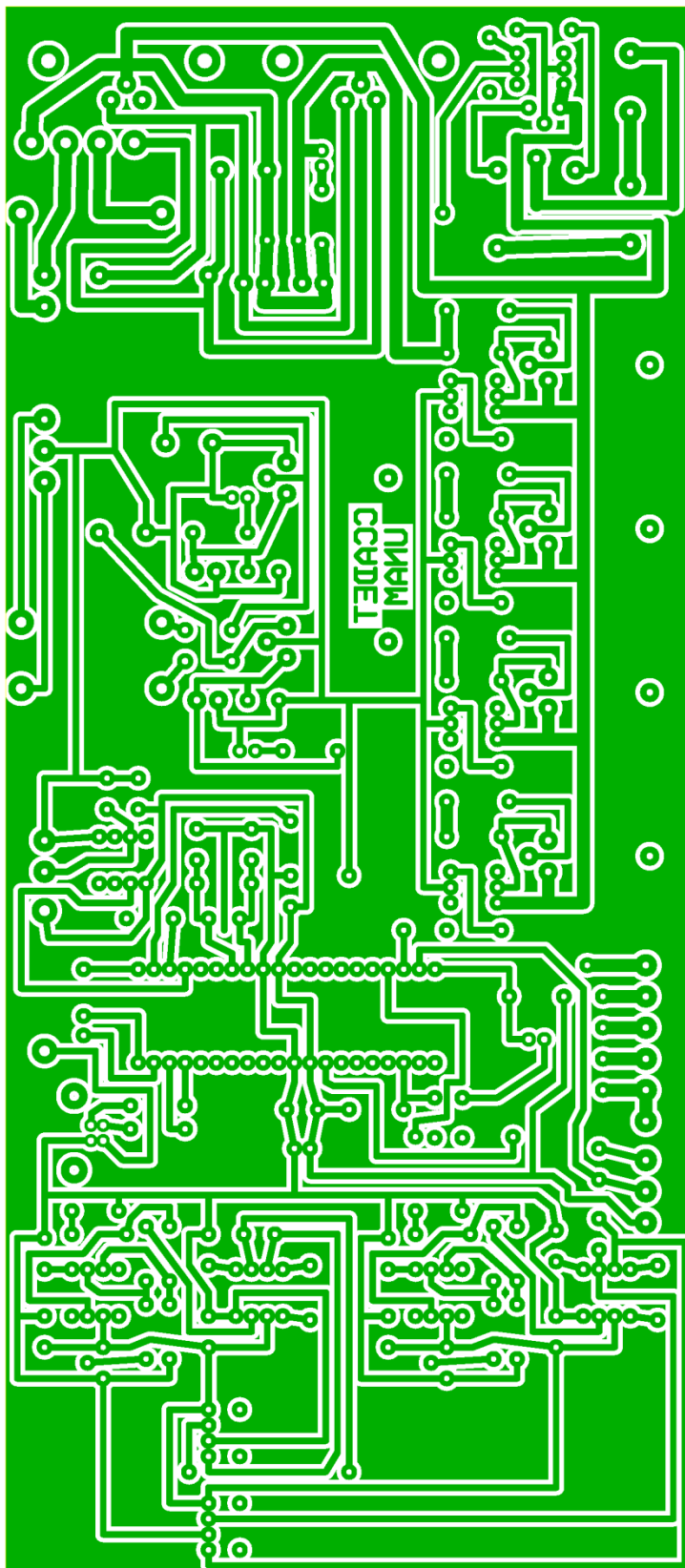


Figura B.2: Capa inferior del circuito impreso.

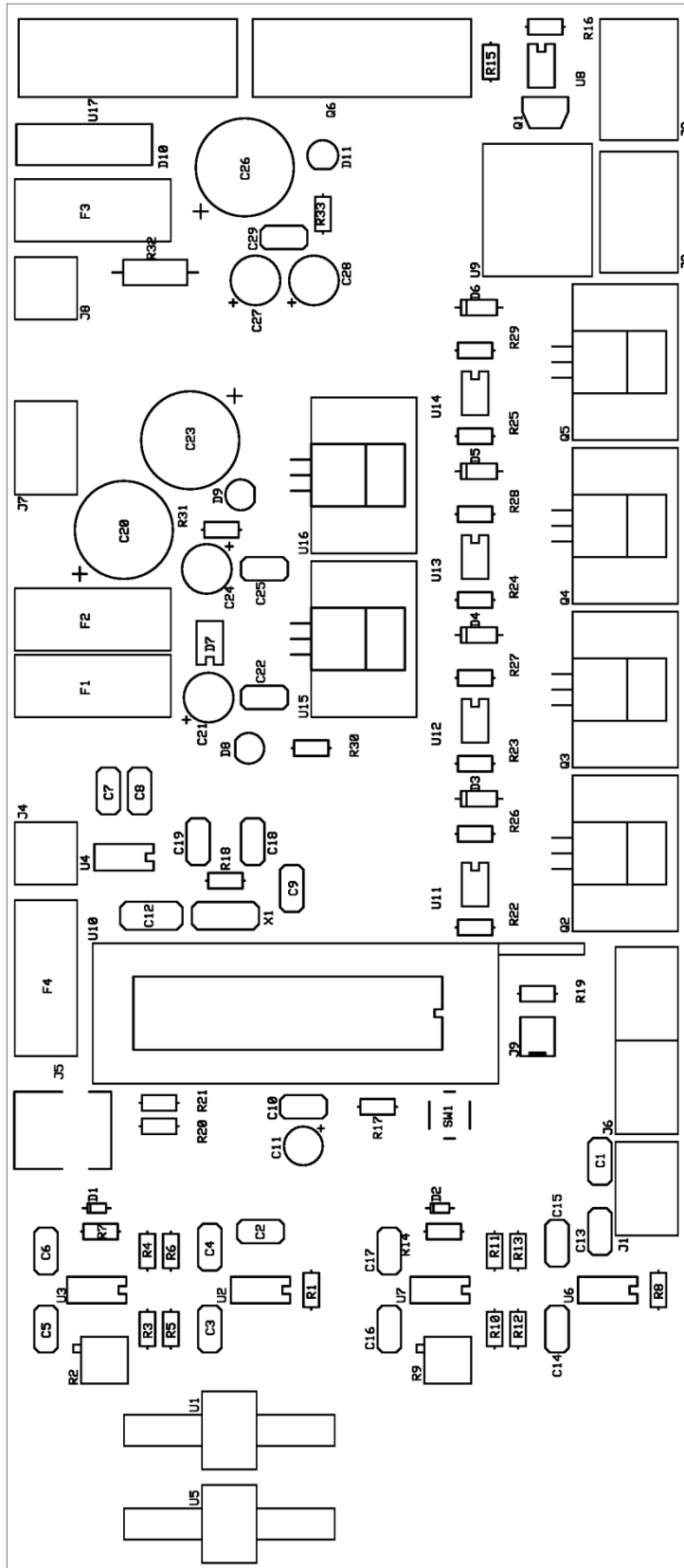


Figura B.3: Delineado de componentes electrónicos en el circuito impreso.

Anexo C

Lista de material

En este anexo, se presenta la lista de material electrónico empleado para la construcción de este circuito.

Identificador en el diagrama esquemático	Componente	Proveedor	Clave de producto
C1	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C2	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C3	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C4	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C5	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C6	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C7	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C8	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C9	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C10	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C11	Capacitor electrolítico de 100 [μ F]	AG Electrónica	CE-100/16V
C12	Capacitor cerámico de 0.47 [μ F]	Newark	46P6326
C13	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C14	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C15	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C16	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C17	Capacitor cerámico de 0.1 [μ F]	Newark	95C0843
C18	Capacitor cerámico de 22 [pF]	Newark	92K9515
C19	Capacitor cerámico de 22 [pF]	Newark	92K9515

Identificador en el diagrama esquemático	Componente	Proveedor	Clave de producto
C20	Capacitor electrolítico de 2200 [μF]	AG Electrónica	CE-2200/50V
C21	Capacitor electrolítico de 100 [μF]	AG Electrónica	CE-100/50V
C22	Capacitor cerámico de 0.1 [μF]	Newark	95C0843
C23	Capacitor electrolítico de 2200 [μF]	AG Electrónica	CE-2200/50V
C24	Capacitor electrolítico de 100 [μF]	AG Electrónica	CE-100/50V
C25	Capacitor cerámico de 0.1 [μF]	Newark	95C0843
C26	Capacitor electrolítico de 2200 [μF]	AG Electrónica	CE-2200/50V
C27	Capacitor electrolítico de 100 [μF]	AG Electrónica	CE-100/50V
C28	Capacitor electrolítico de 100 [μF]	AG Electronica	CE-100/50V
C29	Capacitor cerámico de 0.1 [μF]	Newark	95C0843
D1	Diodo Zener 1N4733A	AG Electrónica	1N4733A
D2	Diodo Zener 1N4733A	AG Electrónica	1N4733A
D3	Diodo de propósito general 1N4001	Newark	10M8464
D4	Diodo de propósito general 1N4001	Newark	10M8464
D5	Diodo de propósito general 1N4001	Newark	10M8464
D6	Diodo de propósito general 1N4001	Newark	10M8464
D7	Puente rectificador DI106	Newark	96K8607
D8	LED	Steren	5/ROJO ULTRA
D9	LED	Steren	5/AZUL ULTRA
D10	Puente rectificador KBL04	Newark	76K6344
D11	LED	Steren	5/AZUL ULTRA
F1	Fusible 1 [A]	Último Recurso	-
F2	Fusible 1 [A]	Último Recurso	-
F3	Fusible 5 [A]	Último Recurso	-
F4	Fusible 1 [A]	Último Recurso	-
J1	Terminal con 3 tornillos	Steren	TRT-03
J2	Conector con 2 tornillos	Newark	71C4093

Identificador en el diagrama esquemático	Componente	Proveedor	Clave de producto
J3	Conector con 2 tornillos	Newark	71C4093
J4	Terminal con 2 tornillos	Steren	TRT-02
J5	Conector USB Tipo B	Newark	10R6833
J6	Dos terminales con 3 tornillos	Steren	TRT-03
J7	Terminal con 3 tornillos	Steren	TRT-03
J8	Terminal con 2 tornillos	Steren	TRT-02
J9	Molex 2 pines horizontal	Último Recurso	-
Q1	Transistor PNO BC557A	AG Electrónica	BC557A
Q2	Transistor NPN TIP41A	Newark	01H1005
Q3	Transistor NPN TIP41A	Newark	01H1005
Q4	Transistor NPN TIP41A	Newark	01H1005
Q5	Transistor NPN TIP41A	Newark	01H1005
Q6	Transistor PNP TIP42B	AG Electrónica	TIP42B
R1	Resistor de 2.49 [kΩ]	AG Electrónica	RP-2K49
R2	Potenciómetro de 10 [kΩ]	Newark	62J1261
R3	Resistor de 10 [kΩ]	Newark	58K3797
R4	Resistor de 10 [kΩ]	Newark	58K3797
R5	Resistor de 10 [kΩ]	Newark	58K3797
R6	Resistor de 10 [kΩ]	Newark	58K3797
R7	Resistor de 10 [kΩ]	Newark	58K3797
R8	Resistor de 2.49 [kΩ]	AG Electrónica	RP-2K49
R9	Potenciómetro de 10 [kΩ]	Newark	62J1261
R10	Resistor de 10 [kΩ]	Newark	58K3797
R11	Resistor de 10 [kΩ]	Newark	58K3797
R12	Resistor de 10 [kΩ]	Newark	58K3797
R13	Resistor de 10 [kΩ]	Newark	58K3797
R14	Resistor de 10 [kΩ]	Newark	58K3797
R15	Resistor de 330 [Ω]	Steren	R330 1/4
R16	Resistor de 10 [kΩ]	Steren	R10K 1/4
R17	Resistor de 4.7 [kΩ]	Steren	R4K7 1/4
R18	Resistor de 1 [MΩ]	Steren	R1M 1/4
R19	Resistor de 4.7 [kΩ]	Steren	R4K7 1/4
R20	Resistor de 22 [Ω]	Steren	R22 1/4
R21	Resistor de 22 [Ω]	Steren	R22 1/4
R22	Resistor de 330 [Ω]	Steren	R330 1/4
R23	Resistor de 330 [Ω]	Steren	R330 1/4
R24	Resistor de 330 [Ω]	Steren	R330 1/4

Identificador en el diagrama esquemático	Componente	Proveedor	Clave de producto
R25	Resistor de 330 [Ω]	Steren	R330 1/4
R26	Resistor de 10 [$k\Omega$]	Steren	R10K 1/4
R27	Resistor de 10 [$k\Omega$]	Steren	R10K 1/4
R28	Resistor de 10 [$k\Omega$]	Steren	R10K 1/4
R29	Resistor de 10 [$k\Omega$]	Steren	R10K 1/4
R30	Resistor de 1 [$k\Omega$]	Steren	R1k 1/4
R31	Resistor de 1 [$k\Omega$]	Steren	R1k 1/4
R32	Resistor de 10 [Ω]	Steren	R10 1
R33	Resistor de 1 [$k\Omega$]	Steren	R1k 1/4
SW1	Push button	Steren	AU-101
U1	Sensor diferencial de presión ± 15 [psid]	Omega	PX26-015DV
U2	Amplificador de instrumentación AD620ANZ	Newark	59K4387
U3	Amplificador operacional TL071CN	Newark	89K1100
U4	Comparador de voltaje LM311N	Newark	89K0675
U5	Sensor diferencial de presión ± 30 [psid]	Omega	PX26-030DV
U6	Amplificador de instrumentación AD620ANZ	Newark	59K4387
U7	Amplificador operacional TL071CN	Newark	89K1100
U8	Optoacoplador con salida a transistor 4N25	Newark	88K1384
U9	Relevador G5LE-1A4-DC12	Newark	36K7449
U10	Microcontrolador PIC18F4553	Newark	54M5025
U11	Optoacoplador con salida a transistor 4N25	Newark	88K1384
U12	Optoacoplador con salida a transistor 4N25	Newark	88K1384
U13	Optoacoplador con salida a transistor 4N25	Newark	88K1384
U14	Optoacoplador con salida a transistor 4N25	Newark	88K1384
U15	Regulador de voltaje LM7812CT	Newark	86K1600
U16	Regulador de voltaje LM7912CT	Newark	41K6357
U17	Regulador de voltaje LM7812CT	Newark	86K1600

Identificador en el diagrama esquemático	Componente	Proveedor	Clave de producto
X1	Cristal de 20 [MHz]	Newark	96F2832
-	Potenciómetro de 20 [kΩ]	AG Electrónica	3540S-1-203
-	Sensor de flujo	Omega	FP-5300
-	Electroválvula	Omega	PV34-B
-	Transformador 12 [V], 0.5 [A] con tap central	Steren	TR12-500
-	Transformador 15[V], 6 [A]	Transformadores Ávila	-

Bibliografía

- [1]. Crane Engineering Department (1969), *Flow of Fluids Through Valves, Fittings and Pipe*, Nueva York, Estados Unidos.
- [2]. McCabe, W. y Smith, J. (1972), *Operaciones Básicas de Ingeniería Química*, Barcelona, España.
- [3]. Sheingold, D (ed.) (1980), *Transducer Interfacing Handbook: A Guide to Analog Signal Conditioning*, Massachusetts, Estados Unidos.
- [4]. Chen, J (sin fecha), *Paddlewheel Flow Sensors: The Overlooked Choice*, [En línea], obtenida de:
http://archives.sensormag.com/articles/1299/63_1299/index.htm [Accedida el 28/08/13].
- [5]. Microchip (sin fecha), *PIC18F2458/2553/4458/4553 Data Sheet*, [En línea], obtenida de:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39887c.pdf>. [Accedida el 03/09/2013].
- [6]. Kester, W. (ed.) (2004), *Analog-Digital Conversion*, Massachusetts, Estados Unidos.
- [7]. Kuphaldt, T. (sin fecha), *Successive Approximation ADC*, [En línea], obtenida de http://www.allaboutcircuits.com/vol_4/chpt_13/6.html. [Accedida el 03/09/13].
- [8]. Flye, W. (2006), *Successive Approximation ADC Block Diagram*, [En línea], obtenida de
http://en.wikipedia.org/wiki/File:SA_ADC_block_diagram.png. [Accedida el 03/09/13].
- [9]. García Breijo, E. (2008), *Compilador C CCS y Simulador Proteus para Microcontroladores PIC*, Barcelona, España.
- [10]. Kruglinski, D., Wingo, S. y Sheperd, G. (1998), *Programming Microsoft Visual C++*, Washington, Estados Unidos.
- [11]. Omega Engineering Inc. (sin fecha), *PX26 Series Pressure Transducers*, [En línea], obtenida de <http://www.omega.com/manuals/manualpdf/M1608.pdf>. [Accedida el 17/09/13].

- [12]. Analog Devices (sin fecha), *AD620 (Rev. H)*, [En línea], obtenida de http://www.analog.com/static/imported-files/data_sheets/AD620.pdf. [Accedida el 17/09/13].
- [13]. Omega Engineering Inc. (sin fecha), *FP-5300, FP-5100, FP8500 and FP-319X Flow Sensors*, [En línea], obtenida de <http://www.omega.com/manuals/manualpdf/M1931.pdf>. [Accedida el 17/09/13].
- [14]. Omron (sin fecha), *PCB Relay G5LE*, [En línea], obtenida de <http://www.farnell.com/datasheets/1509313.pdf>. [Accedida el 17/09/13].
- [15]. Omega Engineering Inc. (sin fecha), *PV SERIES Electronic Proportional Control Valves*, [En línea], obtenida de <http://www.omega.com/manuals/manualpdf/M1655.pdf>. [Accedida el 18/09/13].
- [16]. Erdoğan, B. (sin fecha), *Implementation of a USB based PIC-to-PC Communication*, [En línea], obtenida de http://www.eee.metu.edu.tr/~design/lib/exe/fetch.php?media=lecture_notes:tutorial_1_-_implementation_of_a_usb_based_pic-to-pc_communication.pdf. [Accedida el 28/08/13].
- [17]. Microsoft Support (2007), *Error Message When You Run Regsvr32.exe on 64-bit Windows*, [En línea], obtenida de <http://support.microsoft.com/kb/282747/en-us>. [Accedida el 17/09/13].
- [18]. Teofilov, N. (2003), *2D Graph ActiveX Control*, [En línea], obtenida de <http://www.codeproject.com/Articles/3214/2D-Graph-ActiveX-Control>. [Accedida el 28/08/13].