

**DISEÑO DE UN SISTEMA MODULAR DE PRESIÓN PARA EL  
LABORATORIO DE CONTROL**

**OSCAR FERNANDO MARTÍNEZ LAGUNA  
JONATHAN DAVID LEGUIZAMÓN TURCA**

**ESCUELA COLOMBIANA DE CARRERAS INDUSTRIALES  
FACULTAD DE INGENIERÍA  
ELECTRÓNICA INDUSTRIAL  
BOGOTÁ D.C.  
2014**

**DISEÑO DE UN SISTEMA MODULAR DE PRESIÓN PARA EL  
LABORATORIO DE CONTROL**

**OSCAR FERNANDO MARTÍNEZ LAGUNA  
JONATHAN DAVID LEGUIZAMÓN TURCA**

**TESIS PRESENTADA PARA OPTAR AL TÍTULO DE INGENIERO  
ELECTRÓNICO**

**DIRECTOR DE TESIS:  
Ing. Jonathan Avendaño Pérez**

**CO-ASESOR:  
Ing. Jaime Alberto Parra Plazas**

**ESCUELA COLOMBIANA DE CARRERAS INDUSTRIALES  
FACULTAD DE INGENIERÍA  
ELECTRÓNICA INDUSTRIAL  
BOGOTÁ D.C.  
2014**

**Nota de aceptación**

---

---

---

---

**Firma Presidente del Jurado**

---

**Firma Jurado**

---

**Firma Jurado**

Bogotá D.C. xx-de-xxxx-de 2014

## **DEDICATORIA**

A mi familia y amigos por el apoyo incondicional y su voz de aliento en este largo proceso. Gracias y una vez más ofrezco este logro a ustedes por siempre confiar en mí.

Jonathan David Leguizamón Turca

## **AGRADECIMIENTO**

A nuestro amigo Jonathan Avendaño por su apoyo y dedicación en cada paso de nuestro proyecto, gracias por tu ayuda. Adicional al profesor Andrés Torres por darnos una luz y una guía en lo que fue la culminación de nuestra carrera.

## RESUMEN

La ingeniería electrónica ha contribuido al desarrollo y aplicación de dispositivos para la adquisición de datos por sensores, que han tenido gran impacto en áreas como la Robótica, Mecatrónica, Electromedicina, Industrias energéticas, Industria Militar entre otras. Estos instrumentos se encargan de la adquisición de datos (DAQ), que es el proceso de monitorizar desde un PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido. Un dispositivo DAQ consiste de sensores, hardware de medidas y un PC con software programable. Entre los sensores comunes se encuentran los termopar, RTD, termistor para temperatura, foto sensor para luz, micrófono para sonido, transductor piezoeléctrico para fuerza y presión, acelerómetro para aceleración, entre otros.

En la industria la automatización de sistemas mediante controladores es algo tan necesario como habitual, lo vemos en algunos ejemplos como: Control de potencia en motores de inducción, control de nivel de agua, caudal y presión de procesos químicos entre otros, es muy amplio el uso de controladores PID y el uso de microcontroladores para el desarrollo de los mismos y de sus aplicaciones implementando lenguaje de programación libre que facilite la implementación y el bajo costo de adquisición de estos dispositivos.

El presente trabajo propone diseñar e implementar un prototipo de sistema de presión para la didáctica del laboratorio de sistemas de control, el proyecto se complementa con un sistema modular de hardware con microcontrolador con el algoritmo de un controlador PID que se adiciona a una interfaz gráfica hecha mediante Matlab® que permite reprogramar el controlador haciendo así un sistema de hardware para implementar con diferentes sistemas de Control siempre y cuando la base del control sea un PID. El sistema posee la parte de control y adicionalmente la transmisión de datos inalámbrica por puerto serial o inalámbricamente mediante la tecnología Zigbee. La culminación de este trabajo tendrá como mayor objetivo ser una herramienta práctica para los estudiantes del laboratorio de sistemas control.

# ÍNDICE DE CONTENIDOS

1. Introducción .....	14
1.1 Definición del problema .....	15
1.2 Preguntas de Investigación.....	15
1.3 Justificación .....	16
1.4    Objetivos .....	16
1.4.1 Objetivo General.....	16
1.4.2    Objetivos Específicos.....	16
1.5    Beneficios Esperados .....	17
1.6    Delimitación y limitaciones de la investigación .....	17
2. Fundamentación teórica .....	18
2.1    Controlador PID.....	18
2.1.1    Acción Proporcional .....	19
2.1.2    Acción Integral.....	19
2.1.3    Acción Derivativa .....	20
2.1.4 Estructura de un controlador PID Digital.....	20
2.1.5 Sintonización del Controlador PID .....	21
2.1.6    Método Ziegler-Nichols.....	21
2.2    PWM.....	23
2.2.1    Aplicaciones del PWM .....	24
2.3 Conversor Analógico-Digital y Digital- Analógico.....	25
2.4    Comunicación serial.....	26
2.4.1    Comunicación Asincrónica .....	27
2.4.2    Comunicación Sincrónica .....	27
2.5    Comunicación Zigbee .....	28
2.5.1    Tipos de Dispositivos.....	28
2.5.2    Zigbee VS Bluetooth.....	29
2.5.3    Ventajas y Desventajas .....	29
2.6    El Microcontrolador.....	31

2.6.1	Introducción al Microcontrolador .....	31
2.6.2	Microcontrolador Microchip.....	35
2.6.3	Microchip PIC18F452.....	35
2.6.4	Modulo USART:.....	39
2.6.5	Modulo MSSP.....	39
2.6.6	Comunicación Serial: .....	39
2.6.7	Modulación de ancho de pulsos (PWM).....	40
2.7	Presión.....	40
2.7.1	Definición Presión.....	40
2.7.2	Presión atmosférica.....	40
2.7.3	Presión Hidrostática .....	41
2.7.4	Presión Hidrodinámica.....	42
2.7.5	Presión Ejercida por Los Líquidos.....	42
2.7.6	Unidades de Presión y sus Factores de Conversión .....	42
2.7.7	El Efecto Venturi .....	44
2.8	Adquisición .....	44
2.8.1	Sistema de Adquisición de datos .....	44
2.8.2	Características del Acondicionamiento de señal.....	45
2.8.3	Conceptos necesarios para la Adquisición de señales.....	46
3.	Fundamentación Técnica .....	48
3.1	Proceso de selección del microcontrolador para la tarjeta programable.....	48
3.1.1	Encapsulado .....	50
3.1.2	Módulo de Conversión Analógico Digital .....	51
3.1.3	Módulo de Captura/Comparación/PWM (CCP) .....	52
3.1.4	Características del módulo en modo PWM .....	53
3.1.5	Módulo de transmisión Serial .....	54
3.1.6	Comunicación entre el PIC y el PC .....	55
3.2	¿Qué es un Sensor? .....	55
3.2.1	Especificaciones de un Sensor .....	55
3.3	Comparación de Sensores .....	56
3.4	Sensor de Presión diferencial MPX5100DP .....	57
3.4.1	Función de Transferencia.....	58
3.5	Consideraciones del Ruido.....	59

3.5.1 Efectos de ruido en el sistema del sensor.....	59
3.5.2 Técnicas de filtrado y consideraciones .....	59
3.5.3 Resolución, sensibilidad y número de bits .....	60
3.6 Cables USB y Norma RS-232.....	61
3.7 MAX232 .....	62
3.8 Módulo XBee Explorer Regulado.....	63
3.9 XBee Transceiver.....	63
4. Antecedentes .....	64
5. Metodología de Diseño y Construcción.....	66
5.1 Diseño Metodológico .....	66
5.2 Construcción del Modelo de Control .....	66
5.2.1 Especificaciones de los Elementos Principales .....	67
5.2.2 Procedimiento de Construcción del Modelo de Control .....	71
5.2.3 Montaje Final Del Prototipo .....	74
5.3 MATLAB.....	75
5.3.1 Identificación de sistemas .....	75
5.4 Adquisición de datos para Identificación del Sistema .....	77
5.4.1 Resultados de la adquisición de datos .....	81
5.4.2 Diseño del Controlador .....	83
5.4.3 Simulación en Simulink .....	90
5.5 Diseño de Interfaces Gráficas .....	93
5.5.1 Introducción a la GUIDE .....	93
5.5.2 Propiedades de los componentes.....	95
5.5.3 Funcionamiento de una aplicación GUI.....	96
5.6 Diseño electrónico del hardware.....	97
5.6.1 Especificaciones Generales.....	97
5.6.2 Diseño del Esquemático y el PCB mediante Kicad .....	98
.....	107
5.7 Configuración módulo Xbee.....	108
5.7.1 Configuración del Nodo de Recepción .....	111
5.7.2 Configuración del Nodo Sensor .....	113
6. Experimentación y Resultados.....	114
6.1 Acondicionamiento de Señales .....	114

6.1.1 Acondicionamiento de señales del sensor al convertor A/D .....	114
6.1.2 Filtración de señales del ruido generado por la bomba .....	115
6.2 Pruebas preliminares para la implementación del controlador .....	117
6.2 Medición de la presión con voltaje de entrada ascendente-descendente .....	119
6.3 Medición de voltaje del sensor MPX5100DP .....	121
6.4 Medición de valores con el set point.....	123
6.5 Resultados de la adquisición.....	124
6.6. Resultados del controlador.....	125
6.7 Resultado final del sistema de control .....	127
Conclusiones .....	128
Trabajos a Futuro .....	128
Referencias.....	129
ANEXO A.....	131
REGISTRO DE CONTROL (ADCON0) .....	131
REGISTRO DE CONTROL (ADCON1) .....	132
Calculo de tiempo de adquisición del convertor A/D.....	133
Anexo B .....	135
Circuitos Esquemáticos de Tarjeta Sensor y potencia .....	135
Diseño Esquemático de la Tarjeta de Control.....	136
Diseños de placas de circuito impreso .....	137
Anexo C .....	138
Controlador por software y modulo serial .....	138
Anexo D.....	139
Lista de componentes y relación de precios del proyecto.....	139

## ÍNDICE DE FIGURAS

Figura 1.1: Implementación de acción Integral como reset automático.....	21
Figura 1.1: oscilación sostenida con periodo $P_{cr}$ .....	23
Figura 1.3: Determinación de parámetros para el diseño del controlador.....	23
Figura 1.4: Modulación de la duración de impulsos.....	25
Figura 1.5: Conversión Analógica- Digital.....	26
Figura 1.6: Proceso de conversión Digital – Analógico.....	27
Figura 1.7: Comunicación Asincrónica.....	28
Figura 1.8: Comunicación Síncrona.....	29
Figura 1.9: Organización de un Microcontrolador.....	33
Figura 1.10: Elementos comunes de la CPU.....	34
Figura 1.11: Diagrama de Bloques PIC18F452.....	38
Figura 1.12: Pines del PIC 18F452.....	39
Figura 1.13: Tubo de Venturi.....	45
Figura 1.14: Diagrama de Bloques de un Sistema de adquisición de Datos.....	46
Figura 2.1: Comparación de microcontroladores en cuanto a la frecuencia de operación (MHz).....	50
Figura 2.2: Comparación de microcontroladores en cuanto a número de terminales de E/S.....	50
Figura 2.3: Comparación de microcontroladores en cuanto a memoria de programa (KB).....	50
Figura 2.4: PIC18F452 Encapsulado Tipo DIP.....	51
Figura 2.5: Sensor de presión diferencia MPX5100DP.....	58
Figura 2.6: Diagrama esquemático puerto diferencia.....	58
Figura 2.7: Curva característica salida vs presión diferencial.....	59
Figura 2.8: Respuesta del filtro RC pasa-bajos a una frecuencia de 650Hz.....	61
Figura 2.9: Cable Serial y Cable USB.....	63
Figura 2.10: Circuito Integrado MAX232.....	63
Figura 2.11: Modulo Xbee Explorer.....	64
Figura 2.12: XBee Transceiver.....	64
Figura 3.1: Equipo Patrón HEISE.....	68
Figura 3.2: Registro de bola de 3/8”.....	69
Figura 3.3: Válvula de aguja.....	69
Figura 3.4: tee unión de 1/4”.....	70
Figura 3.5: Unión de 1/4”.....	71
Figura 3.6: Tubería de 1/4”inoxidable.....	72
Figura 3.7: Flaches.....	72
Figura 3.8: Recipiente plastico.....	72
Figura 3.9: Foto de acoples y mangueras.....	73
Figura 3.10: Bomba de agua para parabrisas.....	73
Figura 3.11: Válvula de Aguja de 1/4”.....	74
Figura 3. 12: Conexiones modelo Venturi.....	74
Figura 3.13 Soporte y tubería prototipo.....	75
Figura 3.14: Montaje final del Modelo de control.....	75

Figura 3.15: Diagrama de bloques proceso de adquisición de datos.....	78
Figura 3.16: Diagrama de flujo del software implementado.....	79
Figura 3.17: Plano del circuito propuesto de adquisición de datos.....	80
Figura 3.18: Comparación señal de entrada vs salida.....	82
Figura 3.19: Ventana del Ident y funciona de transferencia.....	83
Figura 3.20: Repuesta de la señal en Lazo Abierto.....	84
Figura 3.21: Respuesta de la señal en Lazo Cerrado.....	85
Figura 3.22: Comparación Salida Lazo Cerrado vs Lazo Abierto.....	85
Figura 3.23: Polos del Sistema en Lazo cerrado.....	86
Figura 3.24: Polos del Sistema con compensador PI.....	87
Figura 3.25: Salida del Sistema Compensada mediante el Controlador Seleccionado.....	88
Figura 3.26: Frecuencia Natural del sistema expresada en radianes.....	89
Figura 3.27: Diagrama de simulación del sistema de control.....	91
Figura 3.28: Salida del Sistema Controlada en <i>Setpoint</i> de 0.7.....	92
Figura 3.29: Salida del Sistema Controlada en <i>Setpoint</i> de 0.5.....	92
Figura 3.30: Diagrama de Bloques en prueba de ruido aleatorio.....	93
Figura 3.31: Salida del sistema con ruido aleatorio en <i>setpoint de 0.7</i> .....	93
Figura 3.32: Ventana de Inicio de la GUI.....	95
Figura 3.33: Ventana de entorno y sus Herramientas.....	95
Figura 3.34: Opciones del componentes.....	96
Figura 3.35: Entorno <i>Property Inspector</i> permite ver y editar propiedades del objeto.....	96
Figura 3.36: Ventana de Inicio de Kicad.....	98
Figura 3.37: Ventana de Editor Esquemático KiCad.....	99
Figura 3.38: Selección de librerías de Componentes en Kicad.....	100
Figura 3.39: Selección de librerías de puertos de poder en Kicad.....	100
Figura 3.40: Ventana de Módulos de componentes.....	101
Figura 3.41: Visualización de dimensiones del Módulo.....	102
Figura 3.42: Visualización del Módulo en 3D.....	102
Figura 3.43: Entorno del editor de PCB de Kicad.....	103
Figura 3.44: Entorno Netlist para asignar componentes.....	104
Figura 3.45: Construcción de tarjeta PCB.....	105
Figura 3.46: Bloque PCB invertido.....	105
Figura 3.47: Módulo de Control.....	106
Figura 3.48: Módulo de acople de señal del sensor.....	106
Figura 3.49: Módulo de Potencia.....	107
Figura 3.50: Sistema modular conectado mediante módulo de alimentación.....	107
Figura 3.51: Entorno del programa X-CTU para configuración Zigbee.....	108
Figura 3.52: Ventana de configuración de puertos.....	109
Figura 3.53: Ventana de Configuración.....	110
Figura 3.54: Ventana de Testeo de puerto.....	111
Figura 3.55: Configuración por Terminal.....	112
Figura 3.56: Configuración a través de Modem Configuration.....	113
Figura 4.1: Suministro de alimentación recomendado con desacople y filtrado en la salida.....	115
Figura 4.2: Sensor de Presión con LPF-RC y buffer como entrada a la siguiente etapa.....	115
Figura 4.3: Señal de salida del sensor comparada con la salida del controlador.....	116
Figura 4.4: Perturbaciones de la señal por alta frecuencia.....	116
Figura 4.5: Cilindro de aluminio para filtración de ruido.....	117
Figura 4.6: Montaje de combinación de módulos.....	118

Figura 4.7: Prueba preliminar de funcionamiento de la planta.....	118
Figura 4.8: Patrón de presión HEISE en prueba de presión ascendente y descendente.....	119
Figura 4.9: Fuente de voltaje en prueba de presiones ascendentes y descendentes.....	121
Figura 4.10: Equipo patrón HEISE para medición de presión diferencial.....	121
Figura 4.11: Medición del voltaje del sensor y transmisión de datos por Zigbee.....	123
Figura 4.12: Comparación serial de entrada vs salida.....	124
Figura 4.13: Comparación señal de entrada vs salida Matlab.....	124
Figura 4.14: Salida del Sistema con restricción en la válvula en <i>Set point</i> de 0.8.....	125
Figura 4.15: Salida del Sistema con restricción en la válvula en <i>Set point</i> de 0.5.....	125
Figura 4.16: Salida del Sistema con restricción en la válvula en <i>Set point</i> de 0.7.....	126
Figura 4.17: Salida del Sistema Controlada en <i>Set point</i> de 0.7.....	126
Figura 4.18: Resultado final de planta de control.....	127

## ÍNDICE DE TABLAS

Tabla 1.1: Parámetros de sintonización en método de oscilación sostenida.....	22
Tabla 1.2: Parámetros de sintonización de método de curva de respuesta.....	24
Tabla 1.3: Características principales del PIC18F452.....	37
Tabla 1.4: unidades de presión y sus factores de conversión.....	44
Tabla 2.1: Tabla de comparación de Sensores en voltaje y escala de operación.....	57
Tabla 3.1: Características de la unión de 1/4".....	71
Tabla 3.2: Caja de Herramientas y Paquetes de Bloques de Matlab.....	76
Tabla 4.1: Tabla de presiones con voltaje de entrada ascendente.....	115
Tabla 4.2: Tabla de presiones con voltaje de entrada descendente.....	115
Tabla 4.3: Prueba de presión voltaje de entrada y salida del sensor de forma ascendente.....	118
Tabla 4.4: Prueba de presión voltaje de entrada y salida del sensor de forma descendente.....	118
Tabla 4.5: Medición de presión estableciendo set point.....	119

# CAPÍTULO 1

## 1. Introducción

Los controladores se usan comúnmente en la industria por su funcionalidad clara y fácil implementación, en el mundo actual el controlador más usado es el PID. El proceso para diseñar un controlador para la automatización industrial debe ser bastante elaborado y puede ser difícil en la práctica por múltiples objetivos que deben alcanzarse, esto puede ser incorporado en módulos de hardware, sin embargo desviaciones en los parámetros pueden generar conflictos en el rendimiento del sistema. En este trabajo se diseña e implementa un prototipo de sistema de presión acompañado de un sistema modular de hardware con controlador reprogramable mediante una interfaz gráfica, que adquiere la información del sistema de presión mediante un sensor de presión diferencial acoplado al módulo de conversión A/D del microcontrolador [1], se usó un sensor de presión diferencial por la importancia de dispositivos que actúan como elementos integradores y de la misma forma ofrecen un nivel de seguridad en sus datos y garantizan el desarrollo completo del proceso en ejecución [2].

El objeto de la demanda real en el seguimiento de la información ambiental ya sea un sistema de control de temperatura o humedad lleva a la segunda fase del desarrollo que se suma a la adquisición y el control del sistema. La transmisión de los datos adquiridos hacia la interfaz gráfica en el ordenador, se realizara de forma inalámbrica usando el módulo serial del microcontrolador pero adaptado con la tecnología Zigbee®, la cual permite lograr la creación de redes inalámbricas de sensores y agregación automática de los datos de seguimiento, se usó el algoritmo del PID para poder mejorar la exactitud de los datos de prueba y la regulación de la presión en el sistema. Los datos adquiridos se transmiten vía Zigbee al receptor conectado al ordenador y procesados por Matlab® mediante una interfaz gráfica que permite no solo cambiar las variables del controlador sino que permite adecuar el set point como el usuario lo desee. Este desarrollo proyecta un mejor uso en la transmisión de información por medio de la tecnología Zigbee® que está basada en el estándar IEEE 802.15.4 de redes inalámbricas que tiene como objetivo la comunicación más segura con baja tasa de envío de datos y maximización de la vida útil de la fuente de energía a usar.

## 1.1 Definición del problema

La enseñanza de la teoría de sistemas de control representa un reto en los planes de estudio de Ingeniería, una alternativa para atraer el interés de los estudiantes consiste en fomentar la construcción de sus propios sistemas de control, sin embargo esto no es del todo sencillo ya que los tiempos en un semestre ya están establecidos y no siempre se logra el perfecto desarrollo de un sistema de control aceptable [3]. En la actualidad es común ver instrumentos reprogramables en el caso de controladores, PLC's u otros dispositivos de automatización industrial, para el caso didáctico entrando en la formación de futuros ingenieros, se ve la necesidad de fomentar el aprendizaje práctico apoyado del teórico. En el caso del laboratorio de sistemas de control se ve la necesidad de diseñar un prototipo de sistema de presión para el laboratorio que viene implementado de un sistema modular de hardware y una interfaz gráfica que permite la reprogramación del controlador, es decir que da la posibilidad de cambiar las variables del controlador y graficar la salida del sistema de forma simultánea [19].

Esto será útil para estudiantes de laboratorio que podrán realizar la identificación del sistema, realizar modelamientos y podrán personalizar e implementar su propio controlador y modificando las constantes mediante la interfaz gráfica. La transmisión de información entre el sistema de hardware y el ordenador se realizara ya sea por puerto serial o vía inalámbrica por medio de tecnología ZigBee®, de esta forma se pueden suprimir cables de conexión a un computador y el mayor uso de vida útil de las baterías, ya que ZigBee® es una tecnología de bajo consumo de energía.

## 1.2 Preguntas de Investigación

Una solución a la necesidad del laboratorio de control es la posibilidad de elaborar un prototipo de sistema de control didáctico, una alternativa para que los estudiantes sientan la confianza hacia esta materia que de otra manera podría parecer árida y teórica. Diseñar un controlador a un sistema ya elaborado o implementar un sistema de hardware a un sistema de control diferente, brinda el inicio de una solución a los temas abordados en este trabajo y de igual forma plantea el interrogante ante la posibilidad del desarrollo de más elementos didácticos para los diferentes laboratorios de asignaturas de Ingeniería electrónica.

## 1.3 Justificación

La Escuela Colombiana De Carreras Industriales ECCI en auras del desarrollo de nuevas tecnologías al servicio del hombre, apoya la investigación y el desarrollo de un prototipo de sistema de control didáctico y así abarcar un mayor interés para el estudiante y solucionar el problema de la construcción de un sistema de control y beneficiar al estudiante en su formación práctica. Además de ello se pretende realizar un estudio de la factibilidad del proyecto y su aporte en la implementación de un sistema WINCS basado en la distribución de sistemas de control inalámbricos que integran la tecnología informática, de red y la de comunicación con los sistemas de control [20]. Este tipo de dispositivo tiene la ventaja de adquisición y transmisión de información inalámbrica y un controlador reprogramable mediante su interfaz en Matlab®, dando así el paso al desarrollo de futuros proyectos que se asemejen a famosos dispositivos como el RT121 y RT124 que son sistemas didácticos para control y control difuso pero más económicos y funcionales para la universidad.

## 1.4 Objetivos

### 1.4.1 Objetivo General

Diseñar un sistema prototipo de control de presión para el laboratorio de control.

### 1.4.2 Objetivos Específicos

1. Diseñar y Desarrollar una Planta para control de presión:
  - Identificar la Planta de control por medio de Ident de Matlab®
  - Hallar el modelo matemático basado en el Ident de Matlab®
  - Realizar Adquisición de datos y uso del Toolbox de Matlab®.
2. Diseñar e Implementar un controlador por Software:
  - Diseñar el controlador mediante parámetros de simulación obtenido en Simulink de Matlab®.
  - Implementar el controlador para el Sistema de Presión.
3. Implementar los protocolos de comunicación para una red ZigBee®
4. Desarrollar un sistema modular de Hardware.
5. Realizar una interfaz de visualización usando Matlab®.

## 1.5 Beneficios Esperados

- Un sistema modular de hardware para implementar el algoritmo de un control PID para un sistema de presión de agua.
- Una interfaz de usuario en Matlab® que permita la reprogramación de las variables del controlador PID para abarcar no solo un sistema sino varios sistemas a controlar.
- El desarrollo del proyecto y que los futuros trabajos se vean acogidos y replicados para el beneficio es la metodología de aprendizaje.

## 1.6 Delimitación y limitaciones de la investigación

El alcance del proyecto es diseñar e implementar un sistema de presión para el laboratorio de control, el cual sirva como herramienta didáctica para las prácticas de la materia y garantice un plus en la metodología del aprendizaje para el estudiante. Adicional aportar a la investigación de los sistemas WINCS que combinan los sistemas de control con la informática y redes de transmisión inalámbrica que en el caso de este proyecto se realiza mediante la tecnología Zigbee que se encarga de adaptar el hardware con su interfaz gráfica mediante Matlab®.

El proyecto se limita a la adecuación de un adaptador inalámbrico de mayor potencia al dispositivo reprogramable y que en si el proyecto se diseñó explícitamente para un controlador PID, ya en caso de desarrollar un Fuzzy u otro tipo de control el hardware permite la reprogramación de su código excepto la interfaz gráfica, que en el caso de este proyecto se diseñó para el sistema de presión de agua y la visualización de su señal de salida.

# CAPÍTULO 2

## 2. Fundamentación teórica

Es importante conocer aspectos relevantes en la construcción de este proyecto en donde a continuación se especificara detalladamente.

### 2.1 Controlador PID

La evolución de los proceso Industriales en los últimos años nos ha traído consigo la necesidad de controlar diferentes sistemas de una forma más precisa por medio de lo que se conoce como control automático ya que permite mantener la variable o un proceso en un punto deseado dentro del rango de una medición. Existen varios tipos de control en los cuales se encuentra el Proporcional-Integral-Derivativo, o comúnmente conocido PID basado en la electrónica analógica puede tener rangos variables en todos sus parámetros y así controlar diferentes procesos tanto simulados como reales, para poder estudiar, analizar y entender que tan importante son los sistemas de control.

Los controladores PID son suficientes para muchos problemas de control en particular cuando la dinámica del proceso es favorable y los requisitos de rendimiento son modestos y los PID se encuentran en gran número en toda nuestra industria. Más del 95% de los controladores son de tipo PID y a menudo se combinan con la lógica para maquinas secuenciales, sistemas de lectores y bloques de funciones simples para construir la automatización se sistemas utilizados en la producción de energía, transporte y manufacturación.

Los controladores comparan el valor real de la salida de un sistema con su entrada de referencia o valor deseado para así determinar el error o desviación y generar una señal de control que reducirá el error a cero o lo más aproximado al cero, están señal se denomina acción de control. El valor proporcional determina la reacción del error actual, el integral genera una corrección proporcional a la integral del error asegurando que al aplicar un control suficiente el valor del error se reduce a cero.

El proceso de selección de parámetros del controlador para que cumplan las especificaciones de desempeño se conoce como sintonización del controlador. Ziegler y Nichols sugirieron reglas de sintonización de PID lo cual implica establecer valores  $K_p$  (proporcional),  $K_i$  (Integral) y  $K_d$  (Derivativo).

### 2.1.1 Acción Proporcional

La razón por la cual el control on-off a menudo da lugar a oscilaciones es que el sistema reacciona de forma exagerada debido a que un pequeño cambio en el error hará que su sistema cambie, este efecto se evita con el control proporcional. La acción proporcional consiste en el producto entre señal de error y la ganancia proporcional, esto se expresa matemáticamente como:

$$e(t) = SP - PV \quad (\text{Ec.1})$$

$$P_{\text{out}} = K_p e(t) \quad (\text{Ec.2})$$

Donde  $P_{\text{out}}$  es la salida del controlador proporcional,  $K_p$  la Ganancia proporcional y  $e(t)$  el error de proceso instantáneo en tiempo real donde se representa de la siguiente forma:

$$e(t) = SP - PV \quad (\text{Ec.3})$$

Donde  $e(t)$  es el error de proceso en el tiempo,  $SP$  el punto establecido y  $PV$  el Proceso Variable.

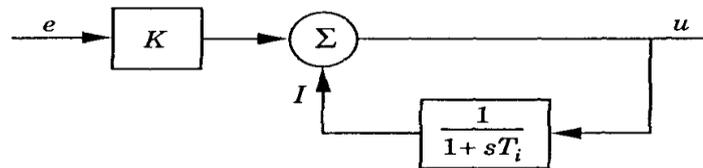
### 2.1.2 Acción Integral

La función principal de la acción integral es asegurar que la salida del sistema concuerde con su referencia en estado estacionario, al contrario del controlador proporcional que normalmente presenta error en estado estacionario con el integral un pequeño error positivo siempre producirá un incremento de la señal del sistema de control y un error negativo siempre dará una señal decreciente sin importar que tan pequeño sea el error. Eso indica que su propósito general es disminuir y eliminar el error en estado estacionario que es producido por el control proporcional, el error es integrado así que se suma al periodo y luego se multiplica por la constante integral. La respuesta posterior se suma al resultado proporcional para formar el control P+I para así obtener una respuesta estable del sistema sin error estacionario. La ganancia total del lazo debe ser menor a 1 y su fórmula está dada por los siguientes parámetros:

$$I_{\text{sal}} = K_i \int_0^t e(\tau) d\tau \quad (\text{Ec.4})$$

Donde  $I_{\text{sal}}$  (señal de salida),  $K_i$  (constante integral),  $e(t)$  (error de estado estacionario), un controlador con acción integral siempre dará error de estado estacionario cero. La acción integral también se puede visualizar como un dispositivo que se restablece automáticamente el término, esto se visualiza en la figura 1.0 en donde un controlador proporcional tiene un reinicio que se ajusta automáticamente mediante la alimentación de una nueva señal que es un valor filtrado, hasta el punto de que el controlador suma la señal, dado así el inicio del reajuste automático o actual control integral.

Figura 1.1: Implementación de acción Integral como reset automático.



Fuente: Libro Ingeniería de control Moderna- Ogata.

### 2.1.3 Acción Derivativa

El propósito de la acción derivativa es mejorar la estabilidad en lazo cerrado, la acción derivativa solo actúa cuando hay un cambio del valor absoluto de error, es decir que si el error es constante solo actuarán el control proporcional e integral. El error es la desviación entre el punto medido y el Set Point, así que la función derivativa mantendrá el error al mínimo corrigiéndolo proporcionalmente a la misma velocidad con la que se produce, de esta forma evita que el error se incremente. La acción derivativa se caracteriza por su tiempo de acción derivada en minutos de anticipo que es adecuada en casos de retraso entre el movimiento de la válvula de control y su repercusión a la variable controlada.

Cuanto más grande el tiempo, habrá inestabilidad en el proceso. Cuando el tiempo sea pequeño la variable oscilará en relación al punto de referencia, su tiempo óptimo es el que retorna la variable al punto de referencia con las mínimas oscilaciones. Su ecuación es dada de la siguiente forma:

$$D_{sal} = K_d \frac{de}{dt} \quad (\text{Ec.5})$$

Donde  $D_{sal}$  (señal derivativa de salida),  $K_d \cdot de/dt$  (constante derivativa con respecto al tiempo). Finalmente la acción derivativa ayuda a disminuir el desbordamiento de la variable en el inicio del proceso y puede emplearse en sistemas con tiempos de retardo largos.

### 2.1.4 Estructura de un controlador PID Digital

El controlador digital es básicamente un procesador que a cada segundo resuelve un algoritmo recursivo "Ecuación de diferencias", el diseño del controlador digital es realizado empleando técnicas propias de los sistemas muestreados que en un caso particular permiten obtener controles más versátiles que los que se pueden conseguir con compensadores analógicos. Sin embargo, existen casos en donde los compensadores analógicos demuestran ser más útiles y trabajan de forma satisfactoria, razón por la que muchas veces se prefiere diseñar un controlador digital basado en la aproximación de los controladores analógicos.

Los controladores digitales se basan en aproximaciones numéricas más comúnmente empleadas para discretizar la ecuación diferencial de un sistema analógico.

### 2.1.5 Sintonización del Controlador PID

La sintonización de los controladores PID consiste en la determinación del ajuste de sus parámetros ( $K_i$ ,  $K_d$ ,  $K_p$ ) y así lograr que el comportamiento del sistema de control sea aceptable y eficaz para el criterio preestablecido. Al poder obtener el modelo matemático de la planta es posible aplicar técnicas de diseño para poder determinar parámetros del controlador y así cumpla las especificaciones del estado transitorio y estado estacionario del sistema de lazo cerrado.

El primer paso de la sintonización consiste en identificar la estática y dinámica del proceso y a partir de esto determinar los parámetros del controlador usando el método deseado. Ziegler y Nichols sugirieron reglas para sintonizar controladores PID en base a su respuesta experimental al escalón o en base al valor de  $k_p$  que produce la estabilidad solo cuando se usa la acción proporcional.

### 2.1.6 Método Ziegler-Nichols

El primer procedimiento sistematizado en el cálculo de los parámetros de un controlador PID fue desarrollado por Ziegler y Nichols, sus ecuaciones fueron determinadas de forma empírica partir de pruebas realizadas en el laboratorio en diferentes procesos y se basan en el modelo de primer orden. Existen dos métodos denominados reglas de sintonización de Ziegler- Nichols y en ambos se puede obtener un 25% de sobrepaso en la respuesta escalón.

#### 2.1.6.1 Método en Lazo Cerrado o de Oscilación Sostenida

El método consiste en obtener la respuesta de la señal medida una perturbación como un pulso en el set point con controlador proporcional. Se observa la respuesta y si esta es amortiguada se incrementa su ganancia hasta lograr oscilaciones sostenidas. La ganancia  $K_p$  se denomina ganancia última y se nota  $K_{cu}$  y el periodo de oscilación es periodo último  $\tau_p$ . En la tabla 1.0 se visualizan los valores de sintonización:

**Tabla 1.1: Parámetros de sintonización en método de oscilación sostenida**

CONTROLADOR	$K_c$	$T_I$	$T_D$
P	$K_{cu}/2$	$\infty$	0
PI	$K_{cu}/2.2$	$\tau_u/1.2$	0
PID	$K_{cu}/1.7$	$\tau_u/2$	$\tau_u/8$

Fuente: Autores

Si la salida no presenta oscilaciones sostenidas para cualquier valor que pueda tomar  $K_p$ , no se aplica este método. Por lo tanto, la ganancia crítica  $K_{cr}$  y el periodo  $P_{cr}$  se determina experimentalmente como lo muestra la figura 1.2



**Tabla 1.2: Parámetros de sintonización de método de curva de respuesta**

CONTROLADOR	Kc	Ti	Td
P	$\frac{1}{K} \left(\frac{\tau}{L}\right)$	$\infty$	0
PI	$\frac{0.9}{K} \left(\frac{\tau}{L}\right)$	$\frac{L}{0.3}$	0
PID	$\frac{1.2}{K} \left(\frac{\tau}{L}\right)$	$\frac{L}{0.5}$	$\frac{L}{2}$

Fuente: Autores

### 2.1.6.3 Método de Asignación de Polos

Este método se basa en el conocimiento de la función de transferencia del sistema de control. Este método simplemente intenta encontrar el controlador que proporcione en un lazo cerrado los polos deseados para obtener determinadas especificaciones. En principio se obtiene el polinomio característico en lazo cerrado del controlador sea un (P, PI PD, PID), luego se calcula el segundo polinomio con los polos deseados en lazo cerrado para que el proceso cumpla las especificaciones. Luego de obtener ambos polinomios o sus ecuaciones se procede a igualar los coeficientes de las ecuaciones características para así determinar cada parámetro del controlador que se esté aplicando.

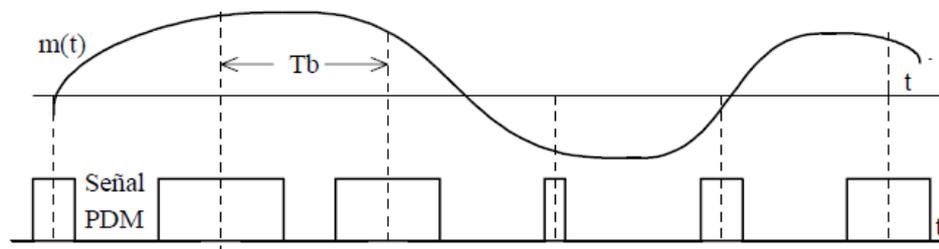
## 2.2 PWM

Un tren de pulsos se caracteriza por la amplitud, su periodo y duración de cada impulso, si de alguna forma se imprimiera sobre alguno de esos parámetros una variación entre señales y su operación fuera irreversible se tendrá lo que se conoce como Modulación Analógica de Impulsos. Un mensaje adecuadamente descrito en sus valores de muestra puede transmitirse mediante modulación analógica de un tren de impulsos, en la modulación analógica cada valor de muestra de la señal hace variar proporcionalmente cada parámetro de cada impulso modulado. Puede transmitirse y en su destino se puede extraer información del mensaje.

Hay varios tipos de modulación analógica de impulsos pero en este trabajo nos centraremos en la modulación de ancho de pulso PWM. La técnica de modulación PWM modifica el ciclo de trabajo de una señal sinusoidal o cuadrada ya sea para transmitir información a través de un canal de comunicación o para controlar la cantidad de energía que se envía a una carga.

En la modulación PWM la duración de cada impulso varía en función del valor de muestra de la señal del mensaje en donde el periodo y la amplitud de los impulsos no cambian. En la modulación PWM la duración de los impulsos varia de forma proporcional a los valores de muestra del mensaje  $m(t)$ , como se muestra en la figura 1.4 el valor más positivo de  $m(t)$  corresponde a un impulso más ancho, mientras el valor más negativo corresponde al impulso más angosto.

**Figura 1.4: Modulación de la duración de impulsos**



**Fuente: Transmisión de Datos, pagina 45, José E. Briceño Márquez**

### 2.2.1 Aplicaciones del PWM

Por lo general todo sistema que procese información binaria para controlar procesos analógicos requieren de una etapa de entrada analógica-digital y una de salida digital-análoga. Esto se puede lograr mediante un algoritmo de modulación por ancho de pulso PWM que permite asignar cierta duración de tiempo en alto o bajo. En la actualidad existen bastantes circuitos integrados que implementan esta modulación PWM para poder controlar fuentes conmutadas, control de motores, control de elementos termoeléctricos, sensores ambientales y otras aplicaciones

Una aplicación del PWM es enviar información de forma analógica y esto es muy útil para comunicarse con sistemas digitales, ya que en un sistema digital es relativamente fácil medir la señal de onda cuadrada, pero para esto se debe tener un conversor analógico digital para así obtener los valores del valor analógico de forma correcta.

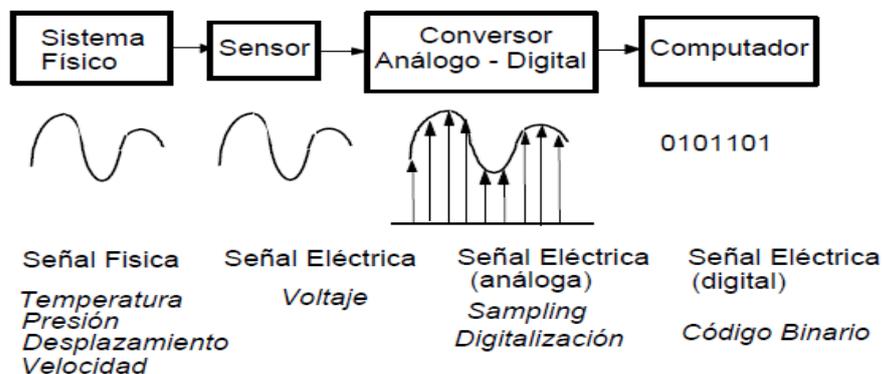
Con el modo de ancho de pulsos se pueden conseguir pulsos a nivel alto de ancho variable con módulos CCPx se puede obtener una resolución de 10 bits para la salida de PWM. Estas señales son muy aplicadas para el control de motores mediante un Triac o un driver para motores de corriente continua, en el caso de este trabajo se usó para controlar la bomba de agua mediante un TIP31C.

## 2.3 Conversor Analógico-Digital y Digital- Analógico

La tecnología ha seguido en avanzada a nivel mundial y operaciones realizadas por sistemas electrónicos analógicos se van haciendo parte de la historia. El desarrollo de microprocesadores y procesadores de seña digital (DSP) han reemplazado los sistemas analógicos, aun si como el mundo real es analógico una forma de enlazar los procesos analógicos con los sistemas digitales es el conversor analógico-Digital o ADC (Analogue to Digital Converter) y los conversores DAC (Digital to Analogue Converter).

Su objetivo principal es transformar la señal eléctrica analógica en un número digital equivalente, en donde la señal analógica que varía constantemente en el tiempo se conecta a la entrada de conversor y se somete a un muestreo a una velocidad fija, es decir, se toman valores discretos en instantes de tiempo de la señal analógica que recibe el nombre de *sampling*, esto matemáticamente es equivalente a multiplicar la señal analógica por una secuencia de impulsos de periodo constante obteniendo así una señal Digital de tren de impulsos con amplitudes limitadas por la envolvente de la señal analógica.. El diagrama de bloques de la figura 1.5 muestra la secuencia desde que la señal analógica entra al sistema hasta que es transformada a señal digital (código binario).

Figura 1.5: Conversión Analógica- Digital.



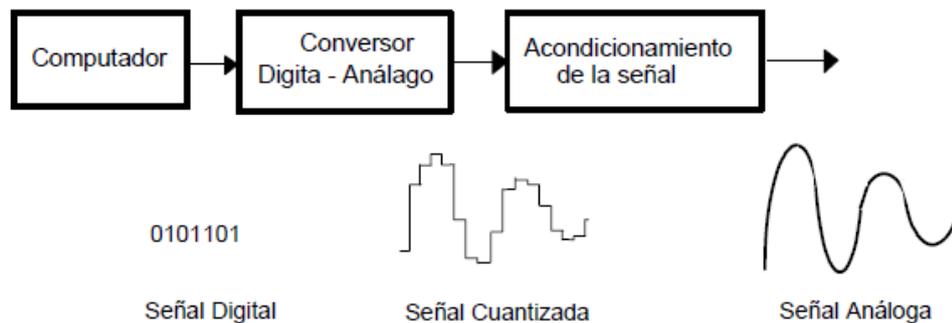
Fuente: Libro *Sistemas electrónicos de comunicaciones*- Roy Blake.

Para garantizar de forma correcta la toma de muestra y la conversión se debe considerar la velocidad del muestreo, para esto se implementa el teorema de Nyquist que establece la frecuencia de muestreo  $f_s$  como mínimo el doble del ancho de banda de la señal muestreada como se indica en la ecuación 6.

$$f_s > 2 \cdot f_m \quad (\text{Ec.6})$$

El proceso inverso (DAC) se muestra en la figura 1.6 En donde la señal digital es transformada en señal eléctrica, la señal digital se pasa por el convertidor DAC y es modulada y recuperada a través de un Filtro pasa bajos e interpolada así de esta forma se obtiene la señal analógica equivalente.

**Figura 1.6: Proceso de conversión Digital – Analógico.**



**Fuente: Libro Sistemas electrónicos de comunicaciones- Roy Blake.**

## 2.4 Comunicación serial

En la actualidad existen diferentes ejemplos de puertos que comunican información de manera serial (un bit a la vez), esto es conocido como puerto serial y ha sido gradualmente remplazado por el puerto USB que permite mayor versatilidad en la conexión de múltiples dispositivos. La mayoría de microcontroladores como el Arduino poseen puertos de comunicación serial para comunicarse con los ordenadores actuales que normalmente solo poseen puertos de USB

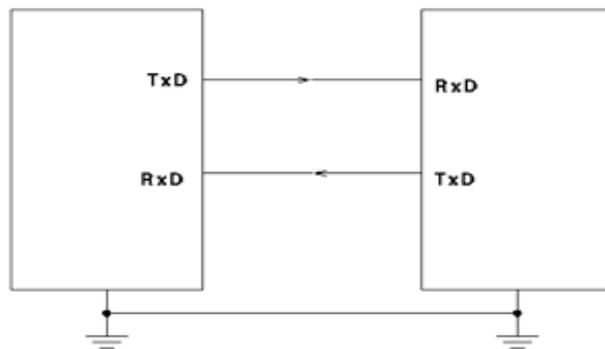
Una de las primeras comunicaciones serial fue el código morse y se llama serial porque los bits se reciben uno detrás del otro en serie. La comunicación RS-232 también es de tipo serial ya que los bits se transmiten uno detrás del otro, la comunicación serial en computadores ha seguido los estándares definidos en 1969 por el RS-232 que establece niveles de voltaje, velocidad de transmisión de datos etc. Este protocolo establece un nivel de -12V como un uno lógico y un nivel de voltaje de +12V como un cero lógico, al igual que los microcontroladores pero con 5V como un uno lógico y 0V como un cero lógico.

Normalmente la comunicación serial se utiliza para transmitir datos en formato ASCII para realizar la comunicación se necesitan 3 líneas de transmisión: 1 Tierra, 2 Transmisión, 3 Recepción. Debido a que la transmisión es asíncrona es posible enviar datos por una línea mientras se reciben datos por la otra. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada y la paridad, para que los puertos se puedan comunicar es necesario que las características sean iguales.

### 2.4.1 Comunicación Asincrónica

El puerto serie asincrónico envía en primer lugar una señal inicial anterior al primer bit de cada byte, carácter o palabra codificada, luego de enviar el código se envía inmediatamente una señal de stop después de cada palabra codificada. La señal de inicio sirve como mecanismo de recepción, llegada y registro de un símbolo, mientras que la señal de stop sirve para predisponer la recepción para que pueda preparar la recepción de un nuevo símbolo. El puerto serie RS-232 es de tipo asíncrono, utiliza cableado desde 3 hilos hasta 25 y conecta desde microcontroladores, computadoras y periféricos como impresoras o mouse. En la comunicación asincrónica, la velocidad de envío de los datos es acordada a priori entre el emisor y el receptor. En la figura 1.7 se puede ver el funcionamiento de la comunicación serial asincrónico.

**Figura 1.7: Comunicación Asincrónica.**

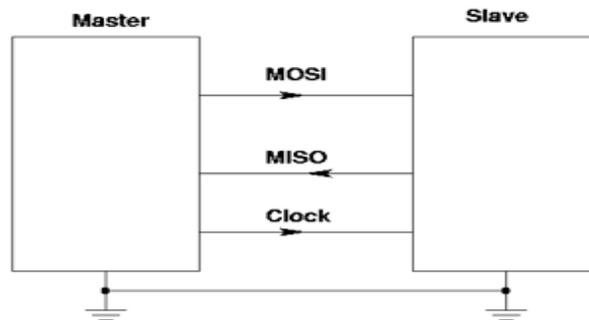


**Fuente: Autores**

### 2.4.2 Comunicación Sincrónica

Su creación data de 1998 aunque solo se empezó a conocer a principios del siglo XXI, se refiere al acceso inmediato en tiempo real de la información u de otros datos, por ejemplo la mensajería instantánea, las características de este tipo de comunicación suelen ser similares a el diálogo cara a cara, en donde una conversación evoluciona en tiempo real lo que desarrollo los sistemas de chat actuales. El envío de datos es sincronizado por el emisor a partir de un pulso constante de reloj (clock), con cada pulso envía un nuevo dato. En la figura 1.8 se visualiza la forma de comunicación Sincrónica.

**Figura 1.8: Comunicación Síncrona.**



**Fuente: Autores**

## 2.5 Comunicación Zigbee

Zigbee es un estándar de alto nivel de comunicación inalámbrica de bajo coste y consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (Wireless personal area network, WPAN), el Zigbee es un sistema ideal para redes domóticas diseñado específicamente para reemplazar la proliferación de sensores-Actuadores. El Zigbee es una arquitectura de red en malla es decir que cada nodo está conectado a todos los nodos, de esta manera es posible llevar la información de un nodo a otro por distintos caminos.

Su nivel de red soporta topologías distintas como estrella, árbol así como redes genéricas en malla. Zigbee utiliza la banda ISM ( Industrial, Scientific and Medical), que es reservada internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica, en concreto 868 MHz en Europa, 915 en estados unidos y 2.4Ghz en todo el mundo. Sin embargo las empresas al desarrollar sus dispositivos siempre optan por la banda de 2.4GHz, ya que esta es libre en todo el mundo.

### 2.5.1 Tipos de Dispositivos

Se definen tres tipos distintos de dispositivo ZigBee según su papel en la red:

- Coordinador ZigBee (ZigBee Coordinator, ZC): El tipo de dispositivo más completo. Debe existir uno por red. Sus funciones son las de encargarse de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos, requiere memoria y capacidad de computación.
- Router ZigBee (ZigBee Router, ZR): Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.

- Dispositivo final (ZigBee End Device, ZED): Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

### 2.5.2 Zigbee VS Bluetooth

El ZigBee es muy similar al Bluetooth pero con algunas diferencias y ventajas para domótica:

- Una red ZigBee puede constar de un máximo de 65535 nodos distribuidos en subredes de 255 nodos, frente a los ocho máximos de una subred (Piconet) Bluetooth.
- Menor consumo eléctrico que el de Bluetooth. En términos exactos, ZigBee tiene un consumo de 30 mA transmitiendo y de 3  $\mu$ A en reposo, frente a los 40 mA transmitiendo y 0,2 mA en reposo que tiene el Bluetooth. Este menor consumo se debe a que el sistema ZigBee se queda la mayor parte del tiempo dormido, mientras que en una comunicación Bluetooth esto no se puede dar, y siempre se está transmitiendo y/o recibiendo.
- Tiene una velocidad de hasta 250 kbit/s, mientras que en Bluetooth es de hasta 3000 kbps.

Debido a las velocidades de cada uno, uno es más apropiado que el otro para ciertas cosas. Por ejemplo, mientras que el Bluetooth se usa para aplicaciones como los teléfonos móviles y la informática casera, la velocidad del ZigBee se hace insuficiente para estas tareas, desviándolo a usos tales como la Domótica, los productos dependientes de la batería, los sensores médicos, y en artículos de juguetería, en los cuales la transferencia de datos es menor.

### 2.5.3 Ventajas y Desventajas

Dependiendo del trabajo a realizar hay que escoger el modo de transmisión que se adecue a la necesidad del proyecto, en este caso del Zigbee se muestra a continuación ventajas y desventajas del Zigbee.

## **Ventajas**

- Ideal para conexiones punto a punto y punto a multipunto
- Diseñado para el direccionamiento de información y el refrescamiento de la red.
- Opera en la banda libre de ISM 2.4 GHz para conexiones inalámbricas.
- Óptimo para redes de baja tasa de transferencia de datos.
- Alojamiento de 16 bits a 64 bits de dirección extendida.
- Reduce tiempos de espera en el envío y recepción de paquetes.
- Detección de Energía (ED).
- Baja ciclo de trabajo – Proporciona larga duración de la batería.
- Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla.
- Hasta 65.000 nodos en una red.
- 128-bit AES de cifrado – Provee conexiones seguras entre dispositivos.
- Son más baratos y de construcción más sencilla.

## **Desventajas**

- La tasa de transferencia es muy baja.
- Solo manipula textos pequeños comparados con otras tecnologías.
- Zigbee trabaja de manera que no puede ser compatible con Bluetooth en todos sus aspectos porque no llegan a tener las mismas tasas de transferencia, ni la misma capacidad de soporte para nodos.
- Tiene menor cobertura porque pertenece a redes inalámbricas de tipo WPAN.

## 2.6 El Microcontrolador

### 2.6.1 Introducción al Microcontrolador

Los microcontroladores forman una parte activa del mundo dado que se presentan en cualquier tipo de trabajo hasta en nuestro hogar es posible observarlos desde nuestro electrodomésticos hasta nuestros celulares o computadores en general como pueden estar controlando las funciones de un mouse del teclado y la pantalla de un Smartphone. Cabe destacar la diferencia entre un microprocesador y un microcontrolador está en que el microcontrolador posee memoria RAM como ROM y el microprocesador la tiene de manera externa direccionada mediante un circuito que multiplexa los canales.

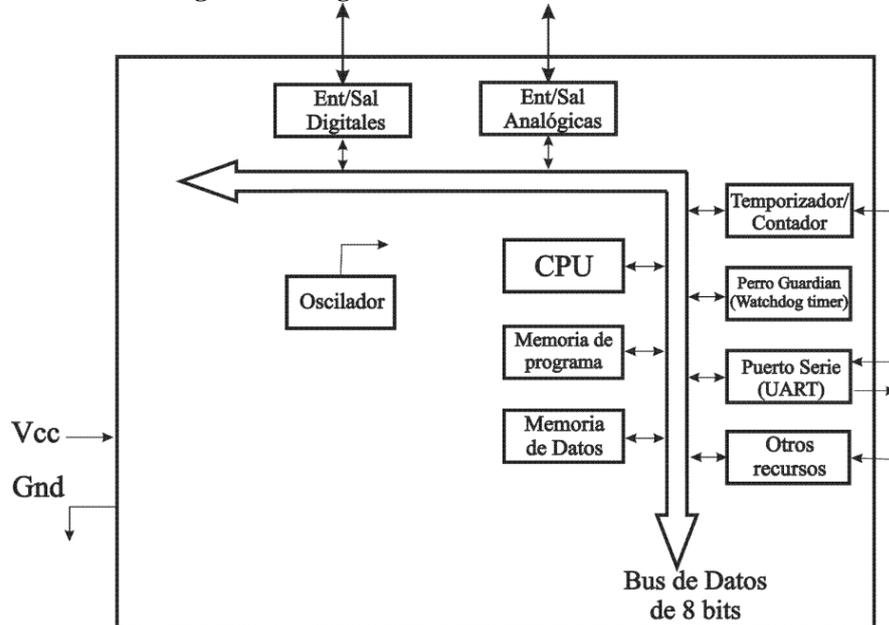
Si vamos más a fondo el microcontrolador es una computadora integrada en un chip por lo tanto consta de CPU, memoria y unidades de entrada y salida además de circuitos de apoyo como generador de reloj, módulos específicos, etc. Se puede escoger entre microcontroladores de 8, 16 y 32 bits (ancho de datos), simplemente dependiendo de la capacidad de proceso que se desea y su rapidez.

Dependiendo de su utilidad, rapidez, funcionalidad y complejidad de su cálculo en el entorno se dispone de varios fabricantes y varias familias de controladores como ATMEL, FREESCALE, MICROCHIP, ST, TEXAS. Una decisión fundamental es la correcta elección del microcontrolador que se necesita, esto dependerá de varios factores entre los más importantes las herramientas, periféricos internos como (temporizadores, CAD, PWM, cantidad de memoria, interrupciones, etc.), tamaño, documentación, precio, facilidad de programar, tipo de encapsulado y disponibilidad del fabricante.

#### 2.6.1.1 Organización del Microcontrolador

Existen diferentes fabricantes de microcontroladores y cada uno maneja diferentes familias con una variedad de modelos, sin embargo, hay bloques que son comunes a todos, en la Figura 1.9 se muestra la organización típica de un microcontrolador y en las siguientes secciones en donde se describen uno a uno los bloques que posee un microcontrolador, tales como la CPU, las entradas analógicas y digitales al igual que sus respectivas salidas, la memoria de programa, la memoria de datos, el temporizador, el módulo USART, el Oscilador etc.

Figura 1.9: Organización de un Microcontrolador



Fuente: Microcontroladores

### 2.6.1.2 Unidad Central de Procesamiento

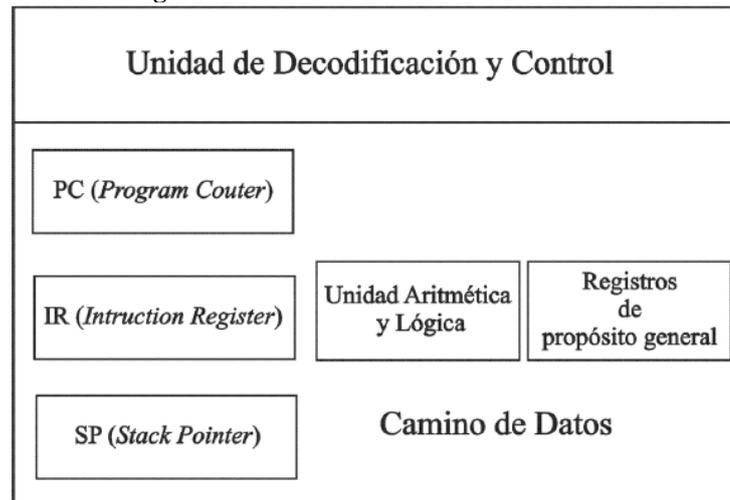
La CPU administra todas las actividades del sistema y ejecuta todas las operaciones sobre los datos, mediante las instrucciones ubicadas en la memoria de código, con las cuales se determina el comportamiento del sistema. El trabajo de la CPU puede resumirse en tres tareas fundamentales: captura de una instrucción, decodificación de la misma y ejecución. Las CPU pueden estar organizadas de diferentes maneras, de acuerdo con ello, pueden clasificarse como CISC o RISC. Con CISC se hace referencia a computadoras con un Repertorio de Instrucciones Complejo (CISC, *Complex Instruction Set Computer*) y RISC es para referir a computadoras con un Repertorio de Instrucciones Reducido (RISC, *Reduced Instruction Set Computers*).

En la Figura 1.10 se muestran los bloques típicos de una CPU, los cuales se pueden clasificar en dos grupos; por una parte el *Camino de Datos* y la *Unidad de Decodificación* que involucra los elementos en los que puede fluir la información cuando se ejecuta una instrucción y por la otra tenemos el *Control*, el cual determina qué elementos se activan dentro del camino de datos para la correcta ejecución de una instrucción. El Contador del Programa (PC, *Program Counter*), el Registro de Instrucción (IR, *Instruction Register*) y el Apuntador de Pila (SP, *Stack Pointer*), son registros con una función específica en una CPU.

El PC contiene la dirección de la instrucción que se va a ejecutar en un instante de tiempo determinado y mientras esa instrucción se ejecuta, el PC automáticamente actualiza su valor para apuntar a la siguiente instrucción a ejecutar. El registro IR contiene la cadena de bits que conforman la instrucción bajo ejecución, de esa cadena, la unidad de control considera el campo del opcode para determinar la activación de las

señales en los demás elementos de la CPU. El SP contiene la dirección del tope de la pila, que es un espacio de almacenamiento utilizado durante la invocación de las rutinas. La Unidad Aritmética y Lógica (ALU, *Arithmetic Logic Unit*) es el bloque que se encarga de realizar las operaciones aritméticas y lógicas con los datos, también opera sobre direcciones para calcular el destino de un salto o la ubicación de una localidad a la que se tendrá acceso para una transferencia de memoria a registro o viceversa.

**Figura 1.10: Elementos comunes de la CPU**



**Fuente: Microcontroladores**

### 2.6.1.3 Sistema de Memoria

Una computadora (y por lo tanto, también un microcontrolador) debe contar con espacios de memoria para almacenar los programas (código) y los datos. En relación a cómo se organizan estos espacios, se tienen dos modelos de computadoras, un modelo en donde el código y datos comparten el mismo espacio de memoria (arquitectura Von Neumann) y el otro en donde se tienen memorias separadas, una para código y otra para datos (Arquitectura Harvard).

### 2.6.1.4 Oscilador

La frecuencia a la que la CPU toma las instrucciones de la memoria del programa para su posterior ejecución, está determinada por el circuito de oscilación. El cual genera la frecuencia de trabajo a partir de un circuito RC, un resonador cerámico o un cristal de cuarzo externo.

### 2.6.1.5 Temporizador/Contador

Es un registro de n bits que se incrementa con una señal de reloj interna o con eventos externos. En el primer caso funciona como temporizador y en el segundo como contador de eventos. Como temporizador es utilizado para manejar intervalos de tiempo y como

contador es la base para programar alguna tarea cada que ocurra una cantidad predeterminada de eventos externos al microcontrolador.

#### 2.6.1.6 Perro Guardián

El Perro Guardián (WDT, Watch Dog Timer) es un temporizador que se compone de un registro de n-bits, cuando el WDT se desborda ocasiona un reinicio del sistema (reset). Su principal función es evitar que el microcontrolador se cicle en estados no contemplados.

#### 2.6.1.7 Puerto Serie

La mayoría de los microcontroladores cuentan con un receptor/transmisor universal asíncrono (UART, Universal Asynchronous Receiver Transceiver), para comunicación serial con dispositivos o sistemas externos, bajo protocolos y razones de transmisiones estándares. La comunicación serial puede ser de dos tipos:

□ Síncrona: Además de la línea de datos se utiliza una línea de reloj.

Asíncrona: Sólo hay líneas para los datos, el transmisor y el receptor se deben configurar con la misma velocidad de transferencia (bits/segundos, Baud Rate), además, se debe definir el mismo formato para cada trama.

#### 2.6.1.8 Entradas/Salidas Digitales

Los microcontroladores usan componentes de Entrada/Salida digitales para intercambiar datos con otros dispositivos. A diferencia de un puerto serie, en donde se transfiere un bit a la vez, en los puertos digitales es posible realizar un intercambio de bytes. Todos los microcontroladores cuentan con puertos digitales, aunque el número puede variar entre dispositivos.

#### 2.6.1.9 Entradas/Salidas Analógicas

Para entradas analógicas algunos microcontroladores incorporan Convertidores Analógico-Digital (ADC, Analogic-Digital Converter) o comparadores analógicos. Estos recursos permiten obtener información analógica del exterior, para monitorear parámetros como temperatura, velocidad, humedad, etc. Para salidas analógicas no es común que un microcontrolador incluya un Convertidor Digital-Analógico (DAC, Digital-Analogic Converter), para compensar esto, algunos microcontroladores incluyen salidas con Modulación por Ancho de Pulso (PWM, Pulse Width Modulation).

## 2.6.2 Microcontrolador Microchip

Los PIC pertenecen a una familia de microcontroladores de tipo **RISC** fabricados por la empresa **Microchip Technology Inc.** Y que fueron derivados de la familia PIC1650 desarrollado por **General Instrument**. Su nombre genérico es Peripheral Interface Controller (controlador de interfaz periférico). Los PICs con memoria PROM o EPROM se han renovado por chips con memoria Flash, en donde se pueden considerar tres altas gamas de Microcontroladores PICs en la actualidad como:

- Básicos (Linebase)
- Rango Medio (Mid Range)
- Alto Desempeño ( High Performance)

Los PIC 18 se consideran de alto desempeño y tienen entre sus miembros PICs que vienen con módulos de comunicación de protocolos avanzados (USB, Ethernet, Zigbee etc.) Además de estos microcontroladores existen otras marcas implementadas por el mismo o diferente fabricante, en este caso se escogen los PICs por su variedad de microcontroladores en el mercado por su rapidez en ejecución, eficiencia de código, juego de instrucciones reducido (RISC), compatibilidad con dispositivos de la misma familia y de otras distintas, gran soporte así como herramientas abundantes y de bajo costo y por ultimo protección de código frente a lecturas no deseadas.

## 2.6.3 Microchip PIC18F452

Para elegir este microcontrolador previamente se analizó su capacidad de procesamiento eso quiere decir la velocidad a la que puede procesar instrucciones ya que resulta importante dependiendo del proceso que se esté ejecutando, el PIC18F452 posee cinco puertos algunos de 8 bits y otros de 5 adicional pueden ser usados para señales de entrada y salida sean analógicas o digitales. El uso de la memoria lo realiza a través de instrucciones estáticas o dinámicas ya que en el caso de las dinámicas los ejecutara mediante la memoria RAM ya que se trata de un proceso más rápido y por otro lado es importante destacar que las instrucciones de la familia del microcontrolador son de tipo RISC en donde las instrucciones son de tamaño fijo y son presentadas en un reducido número de formatos además de que se crea una maquina multitarea y la programación estructurada logra que el código sea usado con cualquier otro fabricante de microcontroladores que usen ese tipo de programación.

### 2.6.3.1 Arquitectura del PIC18F452

El PIC18F452 posee 40 pines en su totalidad, 18 fuentes que podrían producir una interrupción, 5 puertos de comunicación, 4 Timers, 2 módulos generadores de señales PWM y un set de instrucciones de 75 palabras. Lo descrito anteriormente se visualiza en la tabla 1.4:

**Tabla 1.3: Características principales del PIC18F452**

<b>Frecuencia de operación</b>	<b>DC - 40MHz</b>
Memoria de programa	32K
Memoria de datos	1.5K
Memoria de dtos EEPROM	256 bytes
Fuentes de interrupción	18
Puertos de I/O	5
Timers	4
Módulos PWM	2
Comunicación serial	MSSP, USART
Comunicación en paralelo	1 PSP
Módulo ADC	8 canales de ingreso
Set de instrucciones	75 Instrucciones
Encapsulado	40 pines

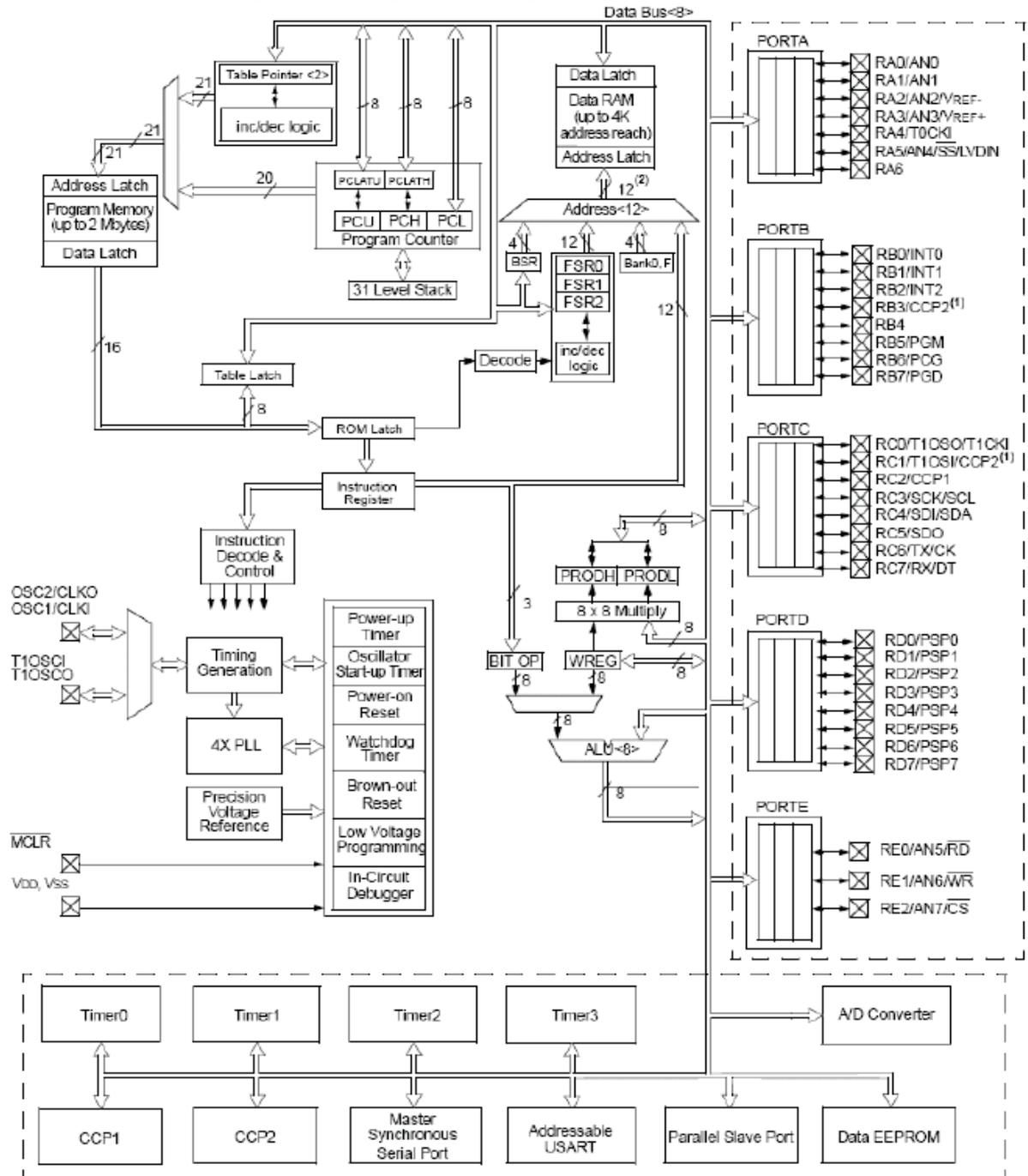
**Fuente: Manual PIC18F452**

El PIC18F452 está desarrollado con 5 puertos multiplexados que pueden ser configurados como entrada/salida o funciones adicionales específicas, los puertos A, B, C y D son de 8 bits mientras el puerto E es de solo 3 bits, estos puertos están internamente relacionados prácticamente con todos los módulos internos, a través de un bus de datos de 8 bits full dúplex. Los módulos TIMER0, TIMER1, TIMER2, TIMER3 acceden de forma directa a los puertos por medio del bus de datos, tiene dos módulos generadores de PWM, en los módulos CCP1 y CCP2 la memoria de datos EEPROM, adicional del módulo de conversión analógico/digital, unidad lógica y aritmética que posee el módulo de multiplicación de 8 bits que se usa frecuentemente con el registro acumulador (WREG).

Adicionalmente tiene registros de funciones especiales FSR de 12 bits, Registro BSR de 4 bits y el registro de los bancos seleccionados de 4 bits que pueden acceder a la memoria de los datos mediante direccionamiento, incluye un módulo contador PC que está directamente relacionado con la memoria de pila que posee una profundidad de 31 niveles, este módulo permite direccionar las instrucciones que se ejecutaran en el microcontrolador con un bus de salida de 20 bits para direccionar la memoria de programa, luego el módulo de eventos generales del micro son los encargados de generar las señales de reloj, Watchdog Timer, Power-on Reset, Power-up Timer En la

figura 1.11 se puede visualizar el diagrama de bloques del microcontrolador PIC18F452 en que se encuentran módulos anteriormente descritos y otros que son generalizados de la familia de microcontroladores 18F.

**Figura 1.11: Diagrama de Bloques PIC18F452**



Fuente: Manual PIC18F452

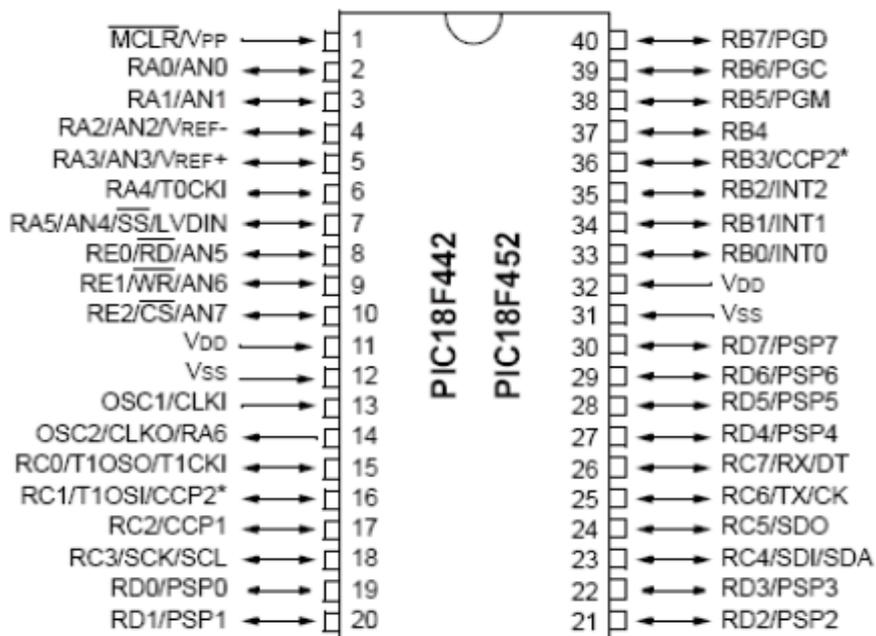
### 2.6.3.2 Descripción de pines

El PIC 18F452 tiene un diseño de 40 pines los cuales se detallan de forma gráfica en la figura 1.12, se visualiza el pin 1 MCLR/Vpp donde MCLR es el master clear que es activado al nivel lógico bajo y cuando recibe este nivel lo que hace es resetear el PIC, los pines 2, 3, 4, 5, 6 y 7 corresponden al puerto bidireccional A de 6 bits en donde están el conversor analógico a digital de 5 bits. Los pines 8, 9 y 10 corresponden al puerto bidireccional E de 3 bits que se encarga del control del puerto de comunicación paralelo encontrándose en los 3 pines la señal de lectura, escritura y habilitador.

Los pines 11 y 32 son alimentación positiva del integrado, mientras que los 12 y 31 son para la tierra. Los pines 13 y 14 corresponden a las señales de reloj externas (cristales) o circuitos RC. Los pines 15, 16, 17, 18, 23, 24, 25 y 26 pertenecen al puerto C en donde se alternan las funciones de un Timer, dos generadores de PWM, ingresos de captura y pines asociados a las comunicaciones serie del PIC siendo I2C, SPL y USART.

Los pines 19, 20, 21, 22, 27, 28, 29 y 30 pertenecen al puerto D con ancho de 8 bits, siendo de propósito general es decir que puede configurarse sus pines de entrada o salida para comunicación esclavo paralelo con todos sus 8 bits de ancho. Los pines 33, 34, 35, 36, 37, 38, 39 y 40 pertenecen al puerto B también de 8 bits también funcionando como propósito general tanto entrada como salida adicionalmente con funciones de interrupciones externas, un módulo generador de PWM y control de voltajes para las comunicaciones seriales.

Figura 1.12: Pines del PIC 18F452



Fuente: Manual PIC18F452

#### 2.6.4 Modulo USART:

Es la unidad síncrona y asíncrona de transmisión y recepción que se encarga de realizar comunicación serial en la familia de PICs 18F452, a la unidad síncrona se le conoce como SCI (interfaz de comunicación serial). Otros fabricantes de microcontroladores poseen unidad UART (Unidad asíncrona de recepción y transmisión) que recibe y transmite datos por comunicación serial tanto de manera síncrona como asíncrona, en el caso de la USART tiene tres configuraciones distintas:

- Transmisión Asíncrona: trabaja en modo full dúplex es decir puede recibir y transmitir al mismo tiempo.
- Transmisión Síncrona: Puede trabajar en dos modos como maestro o como esclavo siendo su forma de comunicación de un solo canal, donde recibe o transmite, pero no hace las dos cosas al mismo tiempo.

En caso del PIC posee dos pines de salida USART y entrada a esta, al igual que otro microcontroladores se debe hacer una configuración previa de los registros empleados por la USART.

#### 2.6.5 Modulo MSSP

El módulo MSSP (Master Synchronous Port) es un puerto de comunicación serial asíncrona half y full duplex y orientado para comunicaciones con componentes externos. Puede funcionar en dos modos de comunicación:

- SPI Serial Peripheral Interface.
- I2C Inter-Integrated Circuit.

#### 2.6.6 Comunicación Serial:

La comunicación de puerto serial es de las más usadas actualmente y posee ventajas de comunicación serial respecto a otras normas como la GPIB como son:

- Disponibilidad: la mayoría de computadores tiene puerto serial.
- Capacidad de controlar instrumentos a una mayor distancia.

El estándar serial más común es el RS-232C que requiere conexiones de transmisión, recepción y tierra. El estándar es capaz de transmitir a distancias mayores y mayor velocidad.

#### **RS232**

RS232 nace de la necesidad de interconectar equipos terminales de datos (STE) con equipos de comunicación de datos (DCE) empleando comunicación binaria, esto lleva a un conjunto de reglas (EIA-232) para poder administrar la manera de comunicarse entre dichos equipos. El estándar especifica como interconectar un DTE con un DCE como si fuera una llamada telefónica ya que la comunicación se establece punto a punto.

## 2.6.7 Modulación de ancho de pulsos (PWM)

Con el modo de modulación de ancho de pulsos se pueden conseguir pulsos a nivel alto de ancho variable, con los módulos CCPx se puede obtener una resolución de 10 bits para la salida PWM. Estas señales tienen muchas aplicaciones para el control de motores de corriente continua, en el proyecto se está usando una bomba de agua para. El CCP1 se multiplexa con el latch del PORTC, por lo que habrá que poner el bit TRISC<2> a cero para que trabaje como salida. La señal del PWM está a nivel alto y viene dada por el registro CCPR1H y el periodo de la señal por el registro PR2, así se borra el TMR2. La configuración anterior permite poder usar el módulo de PWM del Microcontrolador para la aplicación de este proyecto, claro está que esto solo es el principio y se explicará más detalladamente en el Capítulo 3 donde se explica la fundamentación técnica del microcontrolador.

## 2.7 Presión

### 2.7.1 Definición Presión

La presión ( $p$ ) es la magnitud física que mide la proyección de la fuerza en dirección perpendicular por unidad de superficie y sirve para caracterizar la aplicación de una fuerza determinada resultante sobre una línea. En el sistema internacional de unidades la presión se mide por el pascal (Pa) que es equivalente a la fuerza total de un newton que actúa uniformemente en un metro cuadrado. La presión es la magnitud escalar que relaciona la fuerza con la superficie sobre la cual actúa y equivale a la fuerza que actúa sobre esa superficie y se representa en la ecuación 7:

$$p = \frac{F}{A}$$

Ec 7:  $P$ = presión,  $A$ = área de superficie plana,  $F$ = fuerza aplicada de forma uniforme

### 2.7.2 Presión atmosférica

La Presión atmosférica es el peso que ejerce el aire de la atmósfera como consecuencia de la gravedad sobre la superficie terrestre o sobre una de sus capas de aire. La presión en un lugar determinado experimenta variaciones asociadas con los cambios meteorológicos por otra parte la presión atmosférica disminuye dependiendo de la altitud, eso se debe a que la presión decrece a razón de 1 mmHg o Torr (que es la unidad de presión) por cada 10 metros de elevación en niveles próximos al mar. Como se sabe, el planeta tierra está formada por una presión sólida (las tierras), una presión líquida (las aguas) y una gaseosa (la atmósfera).

La atmósfera es la capa gaseosa que envuelve todo el planeta y está formada por mezcla de gases que en conjuntos llamamos aire, como todos los cuerpos, tiene peso, el cual ejerce una fuerza sobre la superficie terrestre es lo que llamamos presión atmosférica. La presión atmosférica varía, no siempre es igual en los diferentes lugares de nuestro planeta y nuestro país, ni en la diferente época del año. Los expertos usan instrumentos llamados altímetros que son simples barómetros aneroides calibrados para alturas. La presión también varía dependiendo de la latitud en donde la menor presión atmosférica al nivel del mar alcanza latitudes ecuatoriales eso se debe al abombamiento ecuatorial de la tierra: la hidrósfera está aún más abultada por lo que las costas de la zona ecuatorial se encuentran varios km más alejadas del centro de la Tierra que en las zonas templadas y, especialmente, en las zonas polares. Y, debido a su menor densidad, la atmósfera está mucho más abultada en el ecuador terrestre que la hidrósfera, por lo que su espesor es mucho mayor que el que tiene en las zonas templadas y polares. Es por ello que la zona ecuatorial es el dominio permanente de bajas presiones atmosféricas por razones dinámicas derivadas de la rotación terrestre. Y es por ello que la temperatura atmosférica disminuye un grado por cada 154 m de altitud, mientras que en la zona intertropical esta cifra alcanza unos 180 m de altitud.

En resumen a mayor altura la presión disminuye y a menor altura aumenta, al ascender el aire se expande y ejerce menor presión. En términos de Temperatura el aire caliente pesa menos que el aire frío y tiende a elevarse, si se observa una olla de agua hirviendo se ve como el aire se va elevando y es porque el aire caliente, a altas temperaturas el aire se calienta y se hace liviano y origina baja presión.

Con bajas temperaturas el aire se enfría, se hace pesado descende y origina alta presión, en este caso se aplica la regla de a mayor altura menor temperatura y a menor altura mayor temperatura. En lugares donde hay mayor humedad hay menor presión y viceversa, si hay menor humedad, mayor presión, esta situación está relacionada con la altura.

### 2.7.3 Presión Hidrostática

Un fluido pesa y ejerce presión sobre las paredes del fondo del recipiente que lo contiene y sobre la superficie de cualquier objeto sumergido en él. Esta presión se conoce como presión hidrostática provoca en fluidos en reposo una fuerza perpendicular a las paredes del recipiente o a la superficie del objeto sumergido sin importar la orientación que adopten las caras. Si el líquido fluyera, las fuerzas resultantes de las presiones ya no serían necesariamente perpendiculares a las superficies, esta presión depende de la densidad del líquido en cuestión y de la altura del líquido por encima del punto en que se mida. Se calcula mediante la siguiente expresión:

$$P = \rho gh + P_0$$

Ec 8: P presión hidrostática en pascales,  $\rho$  densidad del líquido  $K/m^3$ , g aceleración de la gravedad  $m/s^2$ , h altura del fluido en metros,  $P_0$  presión atmosférica en Pa

#### 2.7.4 Presión Hidrodinámica

En un fluido en movimiento general, al medir la presión según diferentes direcciones alrededor de un punto, ésta no será constante, dependiendo la dirección donde la presión es máxima y mínima, y de la dirección y valor de la velocidad en ese punto. De hecho en un fluido newtoniano cuya ecuación constitutiva, que relaciona el tensor tensión con el tensor velocidad de deformación:

$$\sigma_{ij} = (-p + \lambda d_{kk})\delta_{ij} + 2\mu d_{ij} = \left(-p + \lambda \frac{\partial v_k}{\partial x_k}\right)\delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$$

Ec 9: Donde  $\sigma_{ij}$  son las componentes del tensor tensión.

$d_{ij}$  Son las componentes del tensor velocidad de deformación.

$v_i$  Son las componentes del vector velocidad del fluido.

$p$  Es la presión hidrodinámica.

$\lambda, \mu$  Son dos viscosidades que caracterizan el comportamiento del fluido.

#### 2.7.5 Presión Ejercida por Los Líquidos

La presión que se origina en la superficie libre de los líquidos contenidos en tubos capilares, o en gotas líquidas se denomina presión capilar. Se produce debido a la tensión superficial. En una gota es inversamente proporcional a su radio, llegando a alcanzar valores considerables. Por ejemplo, en una gota de mercurio de una diezmilésima de milímetro de diámetro hay una presión capilar de 100 atmósferas. La presión hidrostática corresponde al cociente entre la fuerza normal  $F$  que actúa, en el seno de un fluido, sobre una cara de un cuerpo y que es independiente de la orientación de ésta. Depende únicamente de la profundidad a la que se encuentra situado el elemento considerado. La de un vapor, que se encuentra en equilibrio dinámico con un sólido o líquido a una temperatura cualquiera y que depende únicamente de dicha temperatura y no del volumen, se designa con el nombre de presión de vapor o saturación

#### 2.7.6 Unidades de Presión y sus Factores de Conversión

La presión atmosférica media es de 101 325 pascales (101,3 kPa), a nivel del mar, donde 1 Atm = 1,01325 bar = 101325 Pa = 1,033kgf/cm<sup>2</sup> y 1 m.c.a = 9,81 kPa.

**Tabla 1.4: unidades de presión y sus factores de conversión.**

	Pascal	bar	N/mm <sup>2</sup>	kp/m <sup>2</sup>	kp/cm <sup>2</sup>	atm	Torr	PSI
<b>1 Pa(N/m<sup>2</sup>)=</b>	1	10 <sup>-5</sup>	10 <sup>-6</sup>	0,102	0,102×10 <sup>-4</sup>	0,987×10 <sup>-5</sup>	0,0075	0,00014503
<b>1 bar(10N/cm<sup>2</sup>) =</b>	10 <sup>5</sup>	1	0,1	10200	1,02	0,987	750	14.5036
<b>1 N/mm<sup>2</sup> =</b>	10 <sup>6</sup>	10	1	1,02×10 <sup>5</sup>	10,2	9,87	7500	145.0536
<b>1 kp/m<sup>2</sup> =</b>	9,81	9,81×10 <sup>-5</sup>	9,81×10 <sup>-6</sup>	1	10 <sup>-4</sup>	0,968×10 <sup>-4</sup>	0,0736	0.001422
<b>1 kp/cm<sup>2</sup> =</b>	9,81×10 <sup>4</sup>	0,981	0,0981	10000	1	0,968	736	14.22094
<b>1 atm (760 Torr) =</b>	101325	1,01325	0,1013	10330	1,033	1	760	14.69480
<b>1 Torr(mmHg) =</b>	133,32	0,0013332	1,3332×10 <sup>-4</sup>	13,6	1,36×10 <sup>-3</sup>	1,32×10 <sup>-3</sup>	1	0.019336
<b>1 PSI (libra / pulgada cuadrada) =</b>	6894.75729	0.68948	0.06894	703.188	0.0703188	0.68046	51.7149	1

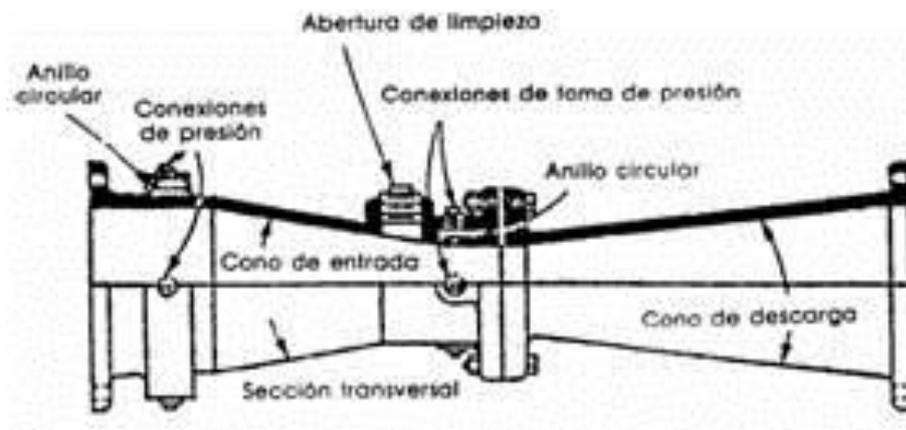
Fuente: Wikipedia

### 2.7.7 El Efecto Venturi

El efecto Venturi recibe su nombre del físico italiano Giovanni Battista Venturi (1746-1822). El Tubo Venturi (aplicación del efecto Venturi) es un dispositivo que origina una pérdida de presión al pasar por él un fluido. En esencia, consta de una tubería corta recta, o garganta, entre dos tramos cónicos. La presión varía en la proximidad de la sección estrecha; así, al colocar un manómetro o instrumento registrador en la garganta se mide la caída de presión y hace posible calcular el caudal instantáneo. El efecto Venturi se explica por el Principio de Bernoulli y el principio de continuidad de masa.

Si el caudal de un fluido es constante pero la sección disminuye, necesariamente la velocidad aumenta tras atravesar esta sección. Por el teorema de la conservación de la energía mecánica, si la energía cinética aumenta, la energía determinada por el valor de la presión disminuye forzosamente. Se utiliza frecuentemente para la fabricación de máquinas que proporcionan aditivos en una conducción hidráulica.

**Figura 1.13: Tubo de Venturi**



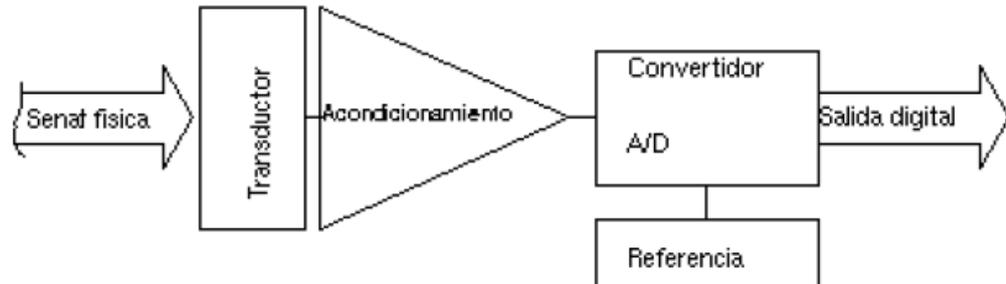
Fuente: Wikipedia

## 2.8 Adquisición

### 2.8.1 Sistema de Adquisición de datos

Un sistema de adquisición de datos es un dispositivo que permite tomar señales físicas de un entorno y convertirlas en datos que se procesaran y presentaran dependiendo de lo que se desee visualizar. En ocasiones un sistema de adquisición es parte de un sistema de control, por lo tanto la información recibida se procesa para obtener señales de control. En la figura 1.14 se ve el diagrama de bloques que componen un sistema de adquisición de datos.

**Figura 1.14: Diagrama de Bloques de un Sistema de adquisición de Datos**



**Fuente: Wikipedia**

En el diagrama de bloques se visualizan como principales bloques los siguientes:

- El transductor: Es el elemento que convierte la magnitud física que va a medirse en una señal de salida (sea tensión o corriente) para poder ser procesada con el sistema.
- El acondicionamiento de la señal: Es la etapa encargada de filtrar y adaptar la señal proveniente del transductor a la entrada del convertidor analógico-digital, adapta el rango de salida al del convertidor A/D acoplando sus respectivas impedancias de salida y entrada.
- El conversor analógico-digital: Es el sistema que presenta en su salida una señal digital a partir de una señal analógica de entrada y así realizar la respectiva cuantificación y codificación.

**Cuantificación:** Implica la división del rango continuo de entrada en una serie de pasos ya sea para los valores infinitos de la entrada o determinados valores de la salida. Lo anterior implica una pérdida de información que no debe olvidarse.

**Codificación:** Es cuando la señal digital se ofrece según un determinado código binario, de modo que las etapas posteriores al convertidor puedan leer los datos de forma correcta.

## 2.8.2 Características del Acondicionamiento de señal

En el sistema de acondicionamiento de señal se pueden encontrar algunas de las siguientes etapas aunque no todas están presentes en este sistema:

- **Amplificación:** Es la forma más común de acondicionamiento, para conseguir la mayor precisión posible la señal de entrada debe ser amplificada para que así su máximo nivel coincida con la máxima tensión que el conversor A/D pueda leer.
- **Aislamiento:** El aislamiento eléctrico entre un transductor y el ordenador, se realiza para proteger al mismo de transitorios de alta tensión que puedan

dañarlo, además de garantizar que las lecturas del conversor A/D no sean afectadas por diferencia de tensiones o en las tierras.

- Multiplexado: Es la conmutación de entradas del conversor, de modo que un solo con un conversor A/D puede medirse datos de diferentes canales de entrada, puesto que el conversor mide diferentes canales y la frecuencia máxima será la original dividida por el número de canales muestreados.
- Filtrado: Su fin es eliminar las señales no deseadas de la señal obtenida, señales alternas como la vibración necesitan de un filtro diferente, conocido como filtro antialiasing que en realidad es un filtro pasa bajo pero con un corte muy brusco que elimina todas las señales de mayor frecuencia al que se está midiendo.
- Excitación: La etapa de acondicionamiento puede generar en ocasiones excitación para algunos transductores, como por ejemplo galgas, termistores etc. que necesitan la misma señal por especificación interna del dispositivo o en su defecto por la configuración de la conexión.
- Linealización: Algunos transductores presentan respuestas no lineales antes cambios lineales en los parámetros que están siendo medidos. Aunque la Linealización se realice por métodos numéricos en un sistema de adquisición suele ser útil realizar esa corrección mediante circuitería externa.

### 2.8.3 Conceptos necesarios para la Adquisición de señales

Existen distintos factores a tener en cuenta cuando se quiere realizar adquisición de señales, a continuación se detallaran 3 que son esenciales en el proceso de adquisición.

#### 2.8.3.1 Estabilidad de la Tensión de Referencia

Los conversores usan distintos métodos para digitalizar la señal, pero siempre respecto a una tensión de referencia, en los casos que la señal de referencia sea externa se debe tener en cuenta lo siguiente:

- Usar elementos que den tensiones con poca deriva térmica.
- Adecuar la impedancia de salida y la impedancia de entrada del conversor.
- Filtrar adecuadamente la salida referente, así como la tensión de alimentación aplicada.

#### 2.8.3.2 Filtrado de las Líneas de Alimentación

Es imprescindible que las líneas de alimentación estén debidamente desacopladas con el uso de condensadores, además del condensador electrolítico que es adecuado para atenuar fluctuaciones de la alimentación debidas al izado de red, es imprescindible añadir condensadores cerámicos para evitar los transitorios de alta frecuencia, el rango del condensador podría estar bien en los 100nF y cercanos al conversor.

### 2.8.3.3 Trazado Adecuado y Separado de la Alimentación Analógica y Digital

En ocasiones este aspecto no se tiene en cuenta, pero la realidad es que es fundamental y puede llegar a causar muchos problemas y más tomando en cuenta cuando se miden tensiones de uno o dos mV. Esto se debe a que los conductores de alimentación tienen una resistencia no nula, así que si se dispone de un microcontrolador trabajando a 4MHz aparecerá en la alimentación picos de intensidad de la misma frecuencia.

Estos picos generan caídas de tensión cuando circulan por los caminos de una Board y esas tensiones harán que el nivel de Tierra (Masa) fluctúe. Se recomiendan los siguientes puntos de observación:

- Las pistas de Tierra deben ser anchas y ocupar la mayor extensión posible.
- Deben haber dos planos de Tierra separados, uno para circuitos digitales y otros para análogos.
- Los planos de la Tierra deben conectarse en un solo punto.
- Es recomendable usar reguladores separados, es decir un regulador para el bloque Digital y otro para el Analógico.
- Si se usan varios reguladores es importante dividir sus líneas de alimentación del mismo modo que la Tierra, es decir en conexión estrella.

# CAPÍTULO 3

## 3. Fundamentación Técnica

### 3.1 Proceso de selección del microcontrolador para la tarjeta programable.

La elección de un microcontrolador para una determinada aplicación se dificulta usualmente por la existencia de una enorme variedad de opciones y características de los chips y otros factores de consideración como son: existencia en el mercado, documentación y apoyo técnico, disponibilidad, costo de herramientas del desarrollo, conocimiento de otros productos del fabricante y seriedad del proveedor.

Los criterios considerados para determinar el microcontrolador son:

1. Frecuencia de operación: Da una idea de la velocidad de procesamiento del microcontrolador.
2. Número de canales analógicos/digitales de entrada y salida.
3. Tamaño de memoria de programa, capacidad de kilobytes para el programa de control.
4. Tipo de encapsulado, para identificar la dimensión y también la disposición de terminales para la realización del circuito impreso.
5. Interfaz de comunicación: Estándares que permitan la comunicación de la tarjeta programable con otros dispositivos como un Ordenador.
6. Lenguaje y/o Compilador: Lenguaje de Programación práctico y compilador fácil de trabajar.
7. Disponibilidad de Herramientas de programación y desarrollo: Facilidad de obtener la herramienta de desarrollo.
8. Costo: Costo del microcontrolador y de sus herramientas de Programación y Desarrollo.

En las Figuras 2.1, 2.2 y 2.3 se pueden visualizar con facilidad la comparación de diferentes microcontroladores de acuerdo a su frecuencia de operación, número de terminales de E/S y memoria del programa que posee cada uno. En las gráficas se marca con una línea para poder diferenciar cuales son los microcontroladores que tienen un valor adecuado para cada uno de los parámetros considerados de acuerdo con los criterios anteriormente descritos. En las Figuras se graficaron las características del más austero al más completo.

Fuente Figuras 2.1, 2.2, 2.3: Autores

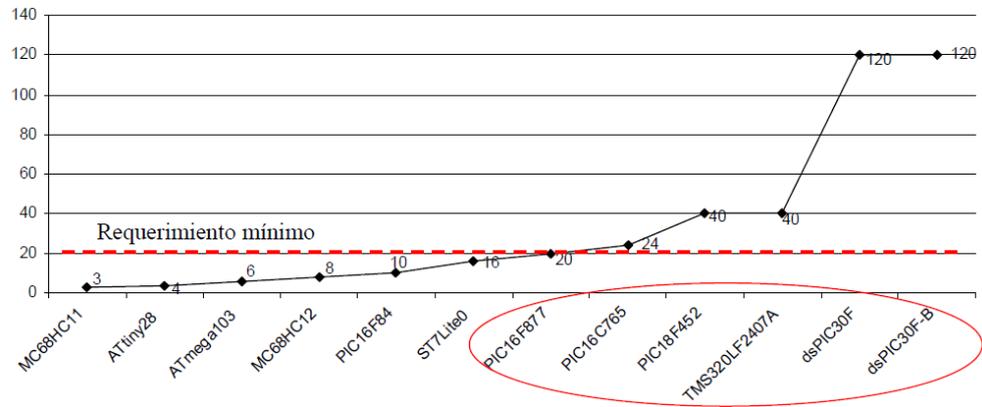


Figura 2.1: Comparación de los microcontroladores en cuanto a la frecuencia de operación (MHz)

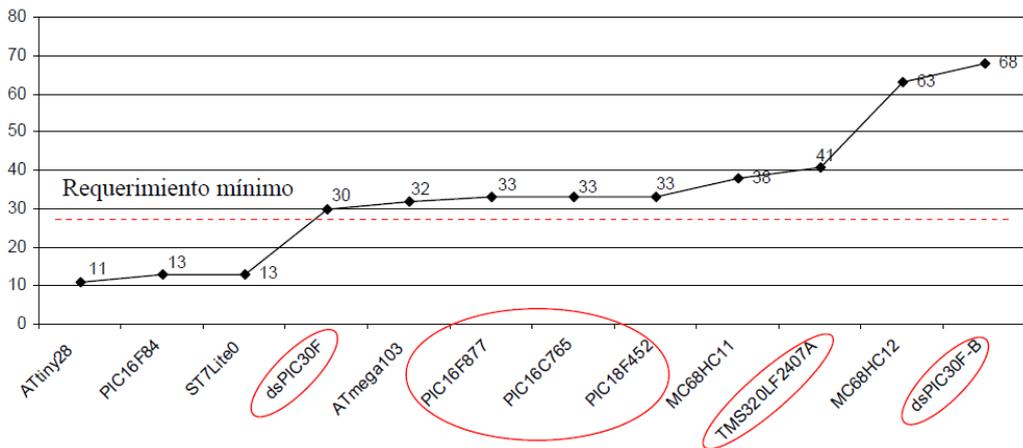


Figura 2.2: Comparación de microcontroladores en cuanto a número de terminales de E/S

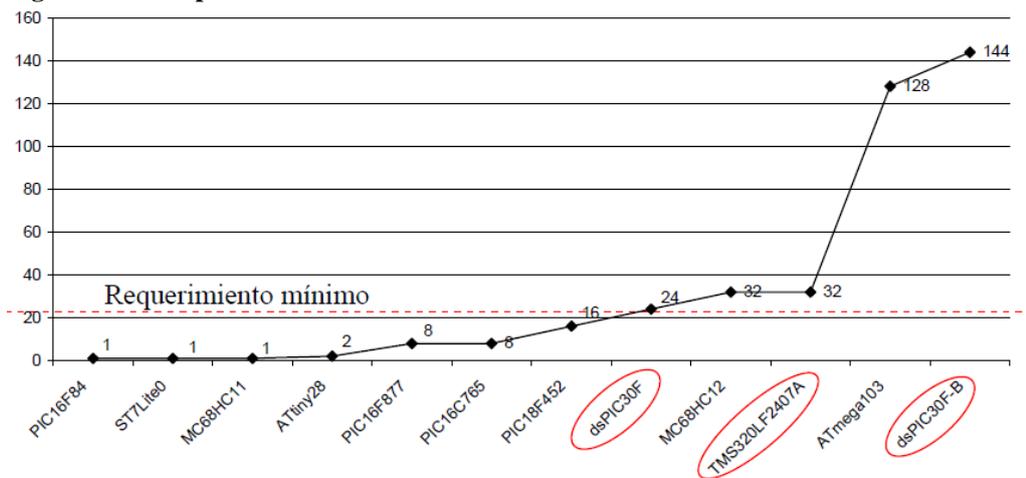


Figura 2.3: Comparación de microcontroladores en cuanto a memoria de programa (KB)

Los microcontroladores útiles para la mayoría de las plantas de control de procesos son los que se encuentran en la parte superior de la línea interrumpida de cada una de las gráficas anteriores; estos microcontroladores tienen costos ligeramente menores a otros citados en las gráficas que cuentan con mejores características, por lo que se optó por seleccionar un microcontrolador que ofreciera una mayor prestación a la tarjeta con disponibilidad de información, herramientas, apoyo técnico y que el costo del microcontrolador se mantenga en el presupuesto del proyecto. En base a los valores de los criterios anteriormente mencionados, se dio la tendencia a escoger el dsPIC30F para el desarrollo de este proyecto ya que proporciona las características idóneas para la tarjeta de adquisición y control. Por motivos de disponibilidad del microcontrolador, el programado de dsPICs ya que no se pueden programar con PICSTART, se escogió el PIC18F452 que tiene características similares cercanas al dsPIC30F como la velocidad de procesamiento 10 MIPS comparado con los 30 MIPS del dsPIC30F adicional que se consigue fácilmente y a un buen precio tanto el PIC como su información, características y soporte en internet sobre el PIC y sobre todo su herramienta de programación y compilación, en este caso se usó para programación en C en PIC C y para compilar y quemar el programa el MPLAB de Microchip, ambas herramientas totalmente gratuitas.

### 3.1.1 Encapsulado

El PIC18F452 está disponible en varios formatos se usó el encapsulado DIP. Este es el encapsulado más empleado en montajes por taladro pasante en placa. Este puede ser cerámico (marrón) o de plástico (negro). Un dato importante en todos los componentes es la distancia entre los pines que poseen, en los circuitos es de vital importancia este dato, así en este tipo el estándar se establece en 0.1 pulgadas (2.54mm).

**Figura 2.4: PIC18F452 Encapsulado Tipo DIP**



**Fuente:** [www.microchip.com](http://www.microchip.com)

### 3.1.2 Módulo de Conversión Analógico Digital

El PIC 18F452 tiene incorporado un módulo de conversión de señal analógica a señal digital, los módulos A/D que utiliza Microchip hacen un muestreo y retención (simple & hold) con un condensador y después utiliza el módulo de conversión. El módulo de conversión A/D tiene un conversor de 10 bits de resolución con 8 entradas analógicas y 13 canales multiplexados. Hay que recordar que la función de transferencia del conversor A/D es el resultado de que la primera transición ocurra cuando la tensión analógica de entrada es igual al voltaje de referencia dividido por 1024, la resolución vendrá dada por la siguiente ecuación:

Ec 10: Resolución de conversión

$$1LSB = V_{REF-} + \frac{(V_{REF+} - V_{REF-})}{1024}$$

Hay registros asociados a este periférico en donde se encuentran los siguientes 4 registros:

- A/D Registro de Alto resultado (ADRESH)
- A/D Registro de Bajo resultado (ADRESL)
- A/D Registro de Control 0 (ADCON0) (revisar Anexos)
- A/D Registro de Control 1 (ADCON1) (revisar Anexos)

El valor que se encuentra en los registros ADRESH/ADRESL no se modifican en caso de reinicio o encendido del PIC, en cambio al restablecer el encendido estos registros contendrán datos desconocidos. Luego de configurar el conversor A/D el canal que se seleccione debe ser adquirido antes de que inicie la conversión, es decir que se debe escoger el canal a usar antes de iniciar el proceso de conversión, los canales de entrada analógica deben corresponder con los bits seleccionados para entrada.

A continuación se enuncian los pasos requeridos para que el módulo de conversión A/D funcione correctamente.

1. Configurar el módulo A/D:
  - Configuración de pines analógicos, referencia de tensión y entradas y salidas digitales (ADCON1)
  - Seleccionar una canal de entrada A/D (ADCON0)
  - Seleccionar el reloj de conversión A/D (ADCON0)
  - Activar el Modulo A/D (ADCON0)

2. Configurar las interrupciones del A/D:
  - ADIF=0
  - ADIE=1
  - GIE=1
  - PEIE=1
3. Esperar el tiempo de adquisición
4. Iniciar Conversión:
  - Activar bit GO/DONE=1 (ADCON0)
5. Esperar a que la conversión A/D se complete, ya sea por:
  - Mediante lectura del bit GO/DONE hasta que sea 0
  - Esperar a que la conversión A/D sea interrumpida (interrupciones)
6. Leer el Registro de Conversión (ADRESH/ADRESL); borrar el bit ADIF si es necesario.
7. Para la siguiente conversión ir al paso 1 o paso 2 si es necesario. El tiempo de conversión A/D por bit es definido  $T_{AD}$ . Una espera mínima de dos  $T_{AD}$  es requerida antes de que se inicie la siguiente adquisición.

Existen dos tiempos básicos de trabajo: el tiempo de adquisición  $T_{ACQ}$  y el tiempo de conversión  $T_{AD}$ .

- **El tiempo de adquisición  $T_{ACQ}$ :** Es el tiempo necesario para que se cargue el condensador  $C_{HOLD}$  con la tensión de entrada. El tiempo de adquisición dentro de los márgenes típicos es de aproximadamente  $20\mu s$ , la adquisición no empieza hasta que no termine la conversión así que se debe esperar un tiempo  $T_{ACQ}$  para seleccionar un nuevo canal o encender el módulo A/D.
- **Tiempo de conversión  $T_{AD}$ :** Es el tiempo necesario para obtener el valor digital de la tensión analógica de entrada, este tiempo depende de la fuente de reloj que se seleccione para la conversión. Para que la conversión sea optima se debe asegurar un tiempo mínimo  $T_{AD}$  de  $1.6\mu s$

Para revisión de registros de control ADCON0, ADCON1 cálculo de tiempo de adquisición, selección del reloj de conversión consultar Anexo A.

### 3.1.3 Módulo de Captura/Comparación/PWM (CCP)

En el proyecto se usó un PWM para poder realizar un control de la bomba de Agua la cual hace parte del sistema de control, para este trabajo se hace uso del módulo CCP del PIC18F452. El módulo de captura, comparación y PWM es llamado CCP, este módulo consta de un registro de 16 bits CCPx que opera como un registro de captura de datos, comparación de datos o como registro para fijar el ciclo de un trabajo de la señal PWM. El PIC18F452 dispone de dos módulos denominados CCP1 y CCP2 asociados a los registros CCPR1 CCPR2 en otras palabras los pines RC2 (CCP1, pin 17) y RC1/RB3 (CCP2, pin 16). En el modo captura el módulo es capaz de almacenar el valor de TIMER1 o TIMER3 cuando se produce un evento (flanco de subida/bajada, cada 4/16 flancos de subida/bajada, etc.) en los pines CCP1 y CCP2, ya en el módulo de

comparación el valor del registro CCPRx está siendo constantemente comparado con el valor del TIMER1 o TIMER3, cuando los valores coinciden el pin CCPx se puede poner en nivel alto o bajo, es decir cambiar de estado o permanecer en su valor.

### 3.1.4 Características del módulo en modo PWM

El módulo en modo PWM utiliza el Timer2 para generar un tren de pulsos de amplitud y ciclo de trabajo variable (señal de PWM), para una frecuencia determinada **F<sub>pwm</sub>** que varía de 1 a 10Khz, se puede obtener un periodo **T<sub>pwm</sub>**. Fijando el ciclo de trabajo (duty\_cicle) podemos variar la potencia entregada en el pin de salida asociado RC<sup>2</sup>/RC1-RB3. En Modo PWM los pines CCPx pueden generar una señal de PWM de 10 bits de resolución, estos pines deben estar configurados como salidas para funcionar en modo PWM. El periodo de la señal PWM se obtiene de configurar TIMER 2 y el contenido del registro PR2 (dirección 0X092) ese registro de 8 bits tiene los más significativos de una cuenta de 10 bits. Para calcular el periodo de la señal PWM se utiliza la siguiente función:

$$PWMT = (PR2 + 1) * 4 * T_{osc} * (Valor\ Prescaler\ del\ TMR2) \text{ Ec.11}$$

Cuando el valor del TMR2 se iguala al valor de PR2, pueden ocurrir los siguientes eventos:

- TMR2 se borra
- El pin CCPx se pone a 1 (excepción: si el *Duty Cycle* es 0% el CCPx no se pone a 1).
- El valor de **CCPRxL** se carga en el **CCPRxH**, el cual es el que se compara con el TIMER2 para fijar el *duty cycle* (*ciclo útil*).

El ciclo útil (*duty cycle*) se define por el valor del registro **CCPRxL** y con los bits **CCP1CON<5:4>** antes de comenzar un nuevo periodo. El registro **CCPRxL** contiene los 8 bits de mayor peso y el **CCP1CON<5:4>** contiene los bits de menor peso. La siguiente ecuación permite calcular el valor del *Duty Cycle*:

$$PWM\ Duty\ Cycle = (CCPRxL:CCPxCON < 5: 4 >) * T_{osc} * TMR2\ prescaler \text{ Ec.12}$$

Si el ciclo de trabajo de la onda PWM es mayor que el periodo, la señal que sale por el pin CCPx está siempre a 1. La resolución máxima en bits viene dada por la expresión:

$$Resolución = \frac{\log\left(\frac{F_{osc}}{F_{pwm}}\right)}{\lg 2} \text{ bits}$$

Ec.13

Por ejemplo, para una frecuencia de PWM de 1.22kHz con un *prescaler* de TMR2 de 16 y una frecuencia de reloj de 20 MHz:

$$\frac{1}{1220} = (PR2 + 1) * 4 * \frac{1}{20 * 10^6} * 16$$

Ec.14

Se obtiene un valor de **PR2=255 (0xFF)**, y una resolución de:

$$Resolucion = \frac{\lg\left(\frac{20 * 10^6}{1200}\right)}{\lg 2} > 10 \text{ bits}$$

Ec.15

Para poder poner en marcha el modo PWM del PIC se deben dar los siguientes pasos:

1. Configurar el periodo PWM mediante escritura del registro PR2
2. Configurar el *Duty Cycle* escribiendo en el registro **CCPR1L** y los bits **CCP1CON<5:4>**
3. Configurar CCPx como salida o sea igual a 0 mediante el TRISC
4. Configurar el *prescaler* del TMR2 y habilitarlo mediante escritura en **T2CON**.
5. Configurar el módulo CCP para la operación en modo PWM.

### 3.1.5 Módulo de transmisión Serial

Para poder completar la parte de comunicación inalámbrica Zigbee, en este caso usando el PIC18F452, se ha hecho uso del módulo USART (Universal Synchronous Asynchronous Receiver Transmitter) donde la principal función es la de transmitir o recibir datos en serie, esta operación puede dividirse en dos categorías: comunicación serie del tipo síncrona o asíncrona, para la comunicación serial se hace uso de la norma RS-232 que normaliza la comunicación entre el PC y el modem, estableciendo el protocolo, señales de control, tensiones, conectores. El modo más habitual de transmisión usado por el USART es el modo asíncrono, puesto que permite comunicaciones en largas distancias (por tal motivo se usó este modo en el proyecto). Existen distintas normas de transmisión serie asíncrona, como la RS232, la RS485 etc. Los niveles de tensión empleados en estas normas son diferentes al empleado por el PIC, por ello suele ser necesaria la utilización de circuitos externos de adaptación.

El módulo USART puede ser configurado en los siguientes modos:

- Asíncrono (full-duplex)
- Síncrono Maestro (half-duplex)
- Síncrono Esclavo (half-duplex)

Los terminales utilizados en el módulo USART con el **RC6/TX/CK** y el **RC7/RX/DT**:

- En el modo síncrono maestro, el pin **RC6/TX/CK** se utiliza como señal de reloj (de salida) y el **RC7/RX/DT** como línea de datos a enviar o recibir.
- En el módulo síncrono esclavo, el pin **RC6/TX/CK** se utiliza como señal de reloj (de entrada) y el **RC7/RX/DT** como línea de datos a enviar o recibir.
- En el modo asíncrono, el pin **RC6/TX/CK** se utiliza como terminal de transmisión de datos y el **RC7/RX/DT** como terminal de recepción de datos.

En el caso de la programación en C como se realizará en este proyecto los parámetros de la USART son: velocidad de transmisión, pines utilizados etc. Se pueden modificar en cualquier programa pero siempre después de haber definido el #USE RS232 y el #USE DELAY, al usar el RS232 se definirá la velocidad en baudios, pin de transmisión, pin de recepción, número de bits y la paridad.

### 3.1.6 Comunicación entre el PIC y el PC

La comunicación entre el PIC y PC es una comunicación serie asíncrona full-duplex, donde principalmente usando el cable serial-USB tomamos 2 hilos para datos Tx y Rx adaptando los niveles de tensión mediante el circuito integrado Max232 estandarizando la velocidad en 9600 baudios (para más información revisar Datasheet del PIC 18F452 de la página 172 a la 180).

## 3.2 ¿Qué es un Sensor?

Se llama sensor al instrumento que produce una señal, normalmente eléctrica, que refleja el valor de una propiedad, mediante alguna correlación definida (ganancia). Un sensor es un instrumento que no altera la propiedad medida. Por ejemplo, un sensor de temperatura sería un instrumento tal que no agrega ni cede calor a la masa medida, es decir, en concreto, sería un instrumento de masa cero o que no contacta la masa a la que se debe medir la temperatura. Existe, además, el concepto de **transductor**: un instrumento que convierte una forma de energía en otra (o una propiedad en otra). Por ejemplo, un generador eléctrico en una caída de agua es un conocido transductor de energía cinética de un fluido en energía eléctrica; sobre esta base se podría pensar, por ejemplo, en un transductor de flujo a señal eléctrica consistente de un pequeño generador a paletas movilizadas por el caudal a medir.

### 3.2.1 Especificaciones de un Sensor

Todos los instrumentos deben ser especificados a un punto tal que aseguren la operación del proceso y que permita la estimación de sus costos. Estas especificaciones se pueden sistematizar, aplicándolas tanto a sistemas sensores como a sistemas actuadores, sin que todas y cada una de las definiciones que siguen a continuación sean aplicables a todo sensor o actuador:

- |                           |                     |
|---------------------------|---------------------|
| • Precisión (o exactitud) | • Reproducibilidad. |
| • Error                   | • Sensibilidad      |
| • Error de No-Linealidad  | • Resolución        |
| • Repetitividad           | • Rango             |

- Rango de Trabajo u Operación
- Banda Muerta
- Corrimiento del Cero
- Tiempo de Respuesta
- Histéresis
- Función de Transferencia

### 3.3 Comparación de Sensores

Al escoger el sensor con el cual se iba a trabajar en el proyecto se debía tener en cuenta primero que el sensor debía ser de presión diferencial (DP) y se tomaron en cuenta 13 sensores con características similares, en donde primaría el que se adecuara más al rango de operación y al voltaje de salida puesto que las características de la bomba son de 3 psi a 12 psi (20 kp a 83 kp) y para esto se necesita una señal analógica de 0 a 5 Vdc. En la tabla 2.1 se puede verificar la comparación de cada uno de los sensores:

**Tabla 2.1: Tabla de comparación de Sensores en voltaje y escala de operación.**

Referencia	Escala en Kp	Voltaje de Salida	Observación
MPX12DP	0-10	55mV full Scale	No sirve escala ni voltaje
MPX2010DP	0-10	25mV full Scale	No sirve escala ni voltaje
MPX2050DP	0-50	40mV full Scale	No sirve escala ni voltaje
MPX2053DP	0-50	40mV full Scale	No sirve escala ni voltaje
MPX2110DP	0-100	40mV full Scale	Sirve escala pero no voltaje
MPX2200DP	0-200	40mV full Scale	No sirve escala ni voltaje
MPX4250DP	0-250	0.2V – 4.9V	No sirve escala ni voltaje
MPX5010DP	0-10	0.2V – 4.9V	No sirve escala pero si el voltaje
MPX2053DP	0-50	40mV full Scale	No sirve escala ni voltaje
MPX2100DP	0-100	40mV full Scale	Sirve escala pero no su voltaje
MPX5050DP	0-50	0.2V – 4.7V	No sirve escala y si voltaje
MPX5100DP	0-100	0.2V – 4.7V	Este se utilizó por su escala y voltaje
MPX5700DP	0-700	0.2V – 4.7V	Escala muy alta, el voltaje es adecuado

**Fuente: Autores**

Se Escogió el sensor MPX5100DP por su rango de operación y el voltaje de salida según los datos de operación de la bomba que en escala de presión va de 20 Kp a 83Kp y señal analógica de voltaje de 0 a 5 Vdc.

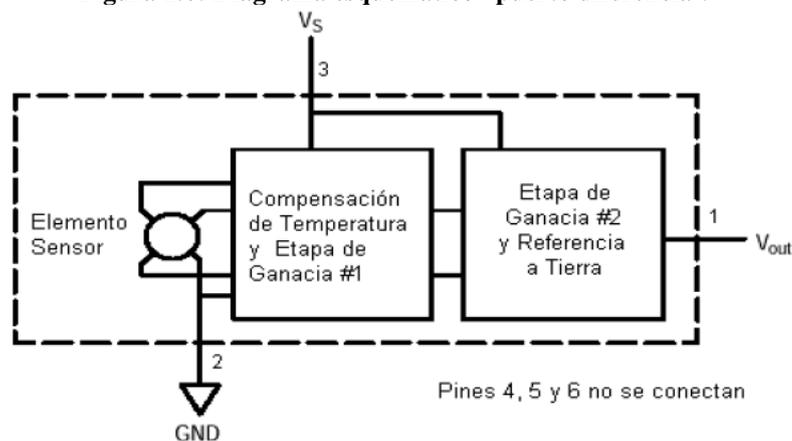
### 3.4 Sensor de Presión diferencial MPX5100DP

Es un transductor piezoresistivo de la serie MPX5100/MPXV5100, es un sensor de presión de silicio monolítico diseñado para una amplia gama de aplicaciones pero en particular las que emplean un microcontrolador o microprocesador con entradas A/D [4]. Este transductor combina avanzadas técnicas de micro mecanizado, metalización de película delgada y procesamiento bipolar para proporcionar una señal precisa de alto nivel de salida analógica que es proporcional a la presión aplicada. Cuenta con un error máximo de  $\pm 2.5\%$  sobre un rango de temperatura de 0 a 85° C, se alimenta con 5 Vdc para proporcionar 200 mV a 0 presión y 4.7 a 100 kPa (típico) En la Figura 2.5 se muestra el tipo de encapsulado del sensor y en la Figura 2.6 se muestra el diagrama esquemático a bloques del sensor completamente integrado. La integración de la compensación de la temperatura y la calibración en un solo chip han permitido una mejora significativa en la exactitud y la estabilidad térmica de la señal de salida de los sensores [5].

Figura 2.5: Sensor de presión diferencial MPX5100DP



Figura 2.6: Diagrama esquemático –puerto diferencia-.



Fuente: Datasheet MPX5100

### 3.4.1 Función de Transferencia

El valor nominal de transferencia para el MPX5100DP, está expresado por la ecuación 10:

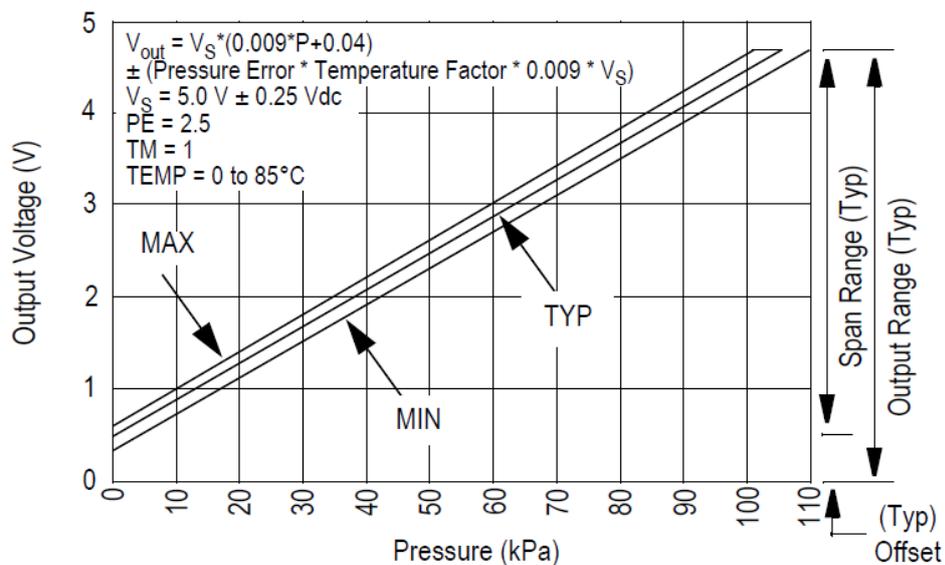
$$V_{out} = V_S (P \times 0.009 + 0.04) \pm (\text{Error en presión} \times \text{Mult.Temp.} \times 0.009 \times V_S)$$
$$V_S = 5.0 \text{ V} \pm 5\% \text{ P kPa}$$

Ec.16: Función nominal de transferencia para el MPX5100DP

Donde  $V_{out}$  es la señal de salida del sensor,  $V_S$  es el voltaje de alimentación, y  $P$  es la presión aplicada. El factor de error de temperatura (Mult. Temp.) Es una respuesta lineal desde  $0^\circ$  hasta  $-0^\circ$  C y desde  $85^\circ$  C a  $125^\circ$  C, mientras que es 1.0 en un rango de  $0^\circ$  a  $85^\circ$  C. El llamado error de presión está limitado por la banda de error, que para el MPX5100DP es de  $\pm 2.5$  kPa en el rango de 0 a 100.

La Figura 2.7 muestra la señal de salida del sensor con relación a la presión de entrada. Se midió con una manómetro de tubo abierto para comprobar la linealidad de señal de salida, tomando como referencia la presión atmosférica. Las curvas que se muestran son la de salida mínima, típica y máxima para una operación sobre una temperatura dentro del rango de  $0^\circ$  a  $85^\circ$  C. Cabe hacer notar que la señal de salida se saturará fuera del rango de presión especificado [6].

Figura 2.7: Curva característica salida vs presión diferencial.



Fuente: Datasheet MPX5100DP

### 3.5 Consideraciones del Ruido

Los convertidores A/D de 8 bits son los más comunes actualmente, convertidores A/D con una resolución alta han incrementado su disponibilidad. Sin embargo, para estos el ruido inherente a los puentes piezorresistivos se convierte en una consideración de diseño. Los dos tipos dominantes de ruido en un sensor de presión integrado piezorresistivo son el ruido blanco y 1/f (ruido flicker). El ruido blanco, es el resultado del flujo no uniforme de portadores a través de una unión y es independiente de la temperatura. El segundo, 1/f, resulta de los defectos del cristal y también es debido al procesamiento de las placas. Este ruido es proporcional al inverso de la frecuencia por lo que es más dominante a bajas frecuencias. El ruido también puede ser generado por circuitos externos. En un sistema sensor, la fuente de alimentación, las tierras y el layout en PCB son importantes y necesitan una consideración especial también [7].

#### 3.5.1 Efectos de ruido en el sistema del sensor

El puente en el transductor produce un voltaje diferencial muy pequeño, en el rango de los milivoltios. El amplificador diferencial del chip amplifica y traduce este voltaje a una terminal de salida de típicamente 0.2 voltios a 4.7 voltios. A pesar de que el transductor tiene una respuesta mecánica de aproximadamente 500 Hz, su ruido a la salida se extiende desde 500 Hz hasta 1 MHz. Este ruido es amplificado, el cual es suficiente como para afectar un conteo en un ADC de 8 bits, y cuatro o cinco conteos en un ADC de 10 bits. Es por lo tanto importante considerar el filtrado.

#### 3.5.2 Técnicas de filtrado y consideraciones

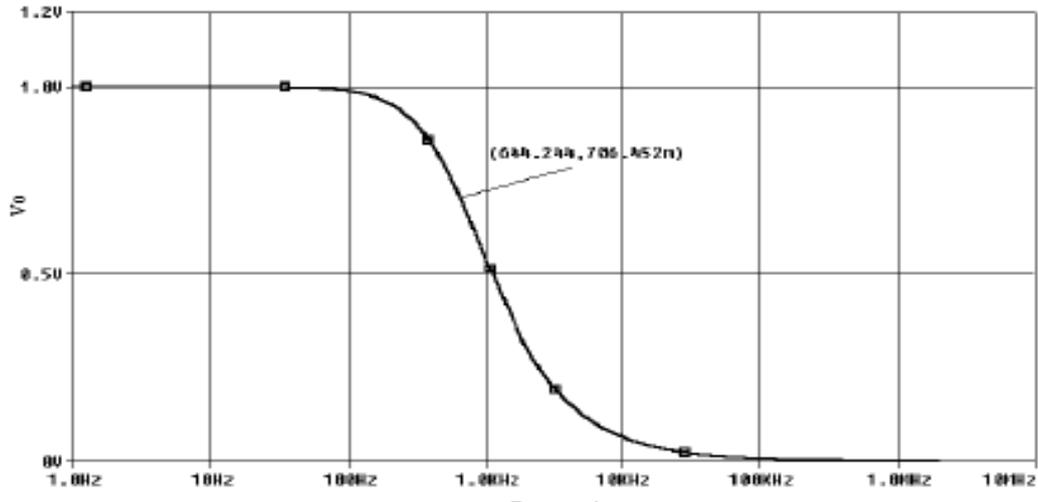
Para Disminuir los efectos del ruido, son efectivas dos propuestas generales, el filtro pasa-bajos mediante hardware o software, aquí solo se hablara del filtro implementado mediante hardware. Cuando se realiza el filtrado con hardware se recomienda un filtro RC pasa-bajos con una frecuencia de corte de 650Hz. Se puede hallar la función de transferencia del filtro de forma sencilla como se muestra a continuación:

Ec.17: Función de transferencia Filtro Pasa-bajos

$$H(j\omega) = \frac{1}{1 + j\omega RC}$$

Se puede seleccionar R con un valor de 750 ohmios y C con 0.33 micro faradios para poder dar un mejor resultado como se aprecia en la Figura 2.8. Algunos conversores analógicos a digital no trabajan bien teniendo como impedancia fuente un filtro RC de un polo, en aplicaciones en donde el conversor A/D es sensible a una impedancia de entrada alta es necesario utilizar un buffer.

Figura 2.8: Respuesta del filtro RC pasa-bajos a una frecuencia de 650Hz



Fuente: Libro Sistemas electrónicos de comunicaciones- Roy Blake.

Debido a que la salida del sensor guarda proporción con la fuente de voltaje, cualquier variación en el voltaje de alimentación se vera de forma proporcional en la salida. El sensor está diseñado, caracterizado y compensado para ser alimentado con una fuente de 5V y no se recomienda alimentar el sensor con un voltaje distinto porque el offset y la compensación del coeficiente de temperatura afectaran la exactitud del sensor.

### 3.5.3 Resolución, sensibilidad y número de bits

La resolución es el significado de del digita menos significativo (LSB - evisar Anexo A). La sensibilidad es el cambio incremental más pequeño que puede detectar un sensor, esto no significa que se deba mostrar el cambio mínimo detectado al usuario. A veces se expresa la sensibilidad de manera alternativa como la relación al cambio incremental en la entrada. Otra característica es el umbral que también está relacionado con la sensibilidad, este es el cambio mínimo que la entrada que provocara de inmediato un cambio en la salida [8]. Ahora bien, entrando en el tema del proyecto se sabe que el Sensor de presión a usar el MPX5100DP puede medir un mínimo de 0 kPa y un máximo de 100 kPa, además de forma ejemplar se podría asumir que la presión de 100 kPa resulta en + 4.5V, asumiendo que este voltaje es el máximo que se tendrá como alimentación, la sensibilidad que se obtendría seria la siguiente:

$$S = \frac{V}{P} = \frac{4.5V}{100kPa} = 45mV / kPa$$

Ec.18

Si la resolución de presión  $R_p$  que se necesita es de al menos 0.1Kpa, entonces

$$R_p = \frac{R_v}{S} \quad \text{Ec.19}$$

Donde  $R_v$  es la resolución de voltaje o LSB.

$$R_v = R_p \cdot S = 0.1 \frac{kPa}{bit} \cdot 45 \frac{mV}{kPa} = 4.5 \frac{mV}{bit} \quad \text{Ec.20}$$

Otra expresión para encontrar el LSB es

$$LSB = \frac{V_{FS}}{(2^n - 1)} \quad \text{Ec.21}$$

Donde  $V_{FS}$  es el voltaje de salida a escala de presión completa (Span +4.5V) y  $n$  el número de bits de salida del conversor A/D [12- 13]. Despejando para  $n$ , se tiene:

$$n = \frac{\log_{10} \left( \frac{V_{FS}}{LSB} + 1 \right)}{\log_{10} 2} = \frac{\log_{10} \left( \frac{4.5V}{4.5mV / bit} + 1 \right)}{\log_{10} 2} = 9.96 \quad \text{Ec.22}$$

Es decir, que por lo menos se requiere de un conversor A/D de 10 bits ( $2^{10} = 1024$  pasos). Para determinar el valor máximo de  $n$ , es necesario calcular la rapidez de respuesta (SR, en inglés) del sensor. Para ello se usa el modelo eléctrico equivalente del sensor de presión [9].

### 3.6 Cables USB y Norma RS-232

La norma RS232 es la más habitual en la comunicación serie, básicamente comunica un equipo terminal de datos (DTE o *Data terminal Equipment*) y el equipo de comunicación de datos (DCE o *Data Communications Equipment*). Las características eléctricas de la norma establecen que la longitud máxima entre el DTE y el DCE no debe ser mayor a 15 metros y su velocidad máxima de transmisión 20.000bps, los niveles de voltaje lógicos se sitúan como 0 lógico de -3V a -15V y 1 lógico entre +3V y -15V. En el caso del proyecto se usa un conector DB9.

Como los PIC usan señal TTL en el módulo USART se debe utilizar un conversor de nivel a RS232 como el MAX232. Para una comunicación Full-duplex del módulo USART del PIC se debe conectar el mismo número de señales TDX y RXD así como la

masa (GND). En la Figura 2.9 se puede visualizar los dos tipos de cables utilizados en la etapa de comunicación del proyecto y son dos cables porque el hardware de comunicación posee conexión por puerto serial y conexión vía Zigbee que en este caso debe usarse el cable USB.

**Figura 2.9: Cable Serial y Cable USB**



**Fuente: Autores**

El cable con conexión RS-232 de 9 pines se utiliza para la conexión de la tarjeta de comunicación al ordenador y el cable USB se usa para conectar y alimentar el módulo receptor Zigbee al puerto USB del ordenador. Finalmente la tarjeta de comunicación puede usar cualquiera de las dos comunicaciones dependiendo de la práctica que se desee realizar.

### 3.7 MAX232

Para la realización de la etapa de comunicación serial se utilizara el circuito integrado de referencia MAX232 que se encarga de cambiar los niveles TTL procedentes de cualquier integrado a los niveles estándar RS-232, cuando se hace una transmisión y cambian los niveles RS-232 a TTL cuanto se tiene una recepción. Finalmente el MAX232 sirve como interfaz de transmisión y recepción para las señales RX, TX, XTS y RTS. Este circuito integrado tiene salidas para manejar niveles de voltaje del RS-232 (aprox. -7.5V a +7.5V) que las produce a partir de una alimentación de +5V utilizando multiplicadores de voltaje que posee internamente y con la adición de condensadores externos y que es de mucha utilidad para dispositivos que tengan alimentación simple de +5 V.

En la figura 2.10 se puede ver el circuito integrado MAX232:

**Figura 2.10: Circuito Integrado MAX232**

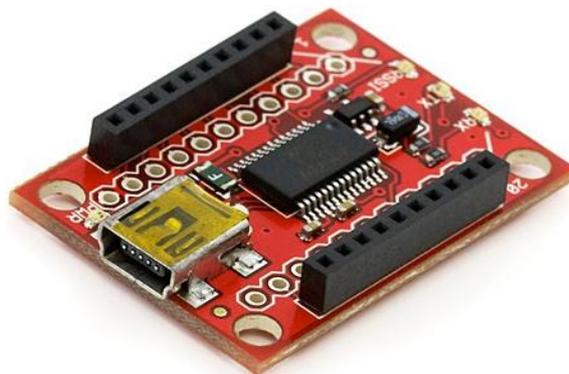


Fuente: Wikipedia

### 3.8 Módulo XBee Explorer Regulado

Este módulo XBee se encarga de la regulación a 3.3V, el acondicionamiento de la señal y los indicadores básicos de actividad (Led Power, Led RSSI y LED de actividad DIN/DOOUT). Este módulo convierte señales de 5V a 3.3V con lo que se puede conectar un sistema funcionando a 5V a cualquier XBee. Esta placa está diseñada para ser conectada directamente al ordenador.

Figura 2.11: Modulo Xbee Explorer



Fuente: [www.xbee.cl](http://www.xbee.cl)

### 3.9 XBee Transceiver

El módulo Xbee Transceiver es un Transceptor de RF XBee que utiliza 2.4 GHZ de frecuencia de funcionamiento y viene en configuraciones de antenas diferentes. Este tipo de dispositivo requiere un mínimo de energía, son fáciles de usar y de configurar adicional que proporciona una comunicación fiable entre dispositivos XBee.

Figura 2.12: XBee Transceiver



Fuente: [www.digi.com](http://www.digi.com)

# CAPÍTULO 4

## 4. Antecedentes

Este trabajo propone el diseño de un sistema de control implementando un hardware modular con fines educativos en donde se evidencia el análisis y la construcción de un proyecto real de sistema de control aplicado a un control de presión de agua o un sistema de abastecimiento de agua. En este trabajo se ha realizado una caracterización del sistema para poder diseñar el controlador PID necesario para la planta, como otros sistemas didácticos de alto coste como los GUNT con controladores PID o Fuzzy-PID, este proyecto aplica una interfaz gráfica diseñada para la reconfiguración de las variables del controlador PID usado, además de visualizar gráficamente la señal de salida del sistema, todo lo anterior fue desarrollado mediante Matlab y la comunicación entre hardware e interfaz por medio de la tecnología Zigbee.

En 2008 Rubaai, Castro-Sitiriche and Abdul R. Ofoli, [10] Desarrollaron un DSP basado en una implementación de laboratorio híbrida entre un controlador difuso y un PID para la optimización de un controlador de motores de alto rendimiento. Usaron un controlador PID y adicional un controlador PID Difuso (FPID) en una aplicación en tiempo real para motores industriales con el objetivo de utilizar los mejores atributos de los controladores PID y FPID para proporcionar un controlador con la mejor respuesta. El diseño, análisis e implementación de las etapas se llevaron por completo usando un dSPACE DS1104( Procesador de señal digital) Tarjeta de Hardware basada en tecnología Power PC para interfaces de E/S con controladores robóticos y aeroespaciales para la investigación creación y el diseño con la cual se realizó el sistema de control en tiempo real y adquisición de datos por Matlab/Simulink en donde los resultados demostraron que el híbrido FPID produce un mejor rendimiento en control que los controladores PID convencionales especialmente en el manejo de linealidades y perturbaciones externas.

En 2008 Urimov, Seong-Hyun and Jung Park [11] analizaron un sistema de control en tiempo real por red Zigbee, la limitaciones de las redes Zigbee en el tiempo de demora y beneficios de su difusión. Para poder realizar el control de un Motor se diseñó un controlador PID con sistema de predictor de Smith y un buffer para eliminación de varianza del tiempo, creando un sistema NCS que es una retroalimentación donde la información de sensores y controladores se envía a través de Zigbee.

En 2010 Zheg Yang, LiuWu and Xinfa Dong [12] después de analizar los inconvenientes en un sistema de abastecimiento de agua elaboraron una tesis con los requisitos necesarios para un sistema de control en donde combinaron un PLC y un convertidor de frecuencia y el control eléctrico donde el PLC ha incorporado funciones del PID, después de los parámetros tales como  $K_p$ ,  $K_i$ ,  $K_d$  rango de entrada, rango de salida, etc.

También en 2010 Zhang Chen- Shu, Ye Jian-Xiong [13] crearon un controlador híbrido combinando un controlador Fuzzy-PID con un predictor de Smith después de un análisis minucioso el controlador puede compensar el largo retardo de tiempo y cambiar un gran retardo a uno pequeño ajustando los parámetros del PID en línea con la lógica difusa en donde también se usa un PLC con algoritmo PID para una planta de suministro de agua en donde los resultados demostraron que la precisión y velocidad de convergencia es alta y la capacidad del modelo de hacer frente a la incertidumbre es fuerte.

En 2011 Mehta, Katrodiya, Mankad [14], en un proyecto de control de nivel de líquido en donde una bomba suministra el líquido dentro de un parámetro deseado, utilizan un algoritmo de PID en lenguaje C implementado en un microcontrolador PIC, El controlador genera la salida de acuerdo con la señal de error y el sistema deriva la señal alcanzando el cero. También implementando una interfaz con el PC a través implementando un MAX232 y el conector DB9 para la identificación del sistema y observación de la salida del sistema utilizando Matlab.

En 2011 Xinrong Zhang, Bo Chang [15] diseñaron un sistema de monitoreo por sensores de humedad y temperatura en el medio ambiente de la agricultura, basándose en un algoritmo de PID difuso con él con un Microprocesador CC2430 que integra el protocolo de comunicación Zigbee para él envió de la información de los sensores al microprocesador y de esta manera tomar perfecta recopilación de datos de temperatura y humedad con alta precisión por medio de una red inalámbrica de sensores y el algoritmo para mejorar la exactitud de los datos de prueba. Este sistema logro las ventajas de una red de nodos de bajo costo, bajo consumo de energía gracias al sistema Zigbee, no cables, interfaz amigable.

En 2011 ChenRujun, He Zhangxiang, QiuJieting, He Lanfang [16] se analiza el diseño y la implementación de una unidad de adquisición de datos distribuida (DDAU) que se basa en sincronización GPS y ZigBee. El DDAU se utiliza en la exploración electromagnética tridimensional para la detección del petróleo y gas (Hidrocarburo), se compone de adquisición de datos y un módulo de control incorporado DSP que posee cuatro canales que amplifican ,convierten y procesan señales débiles de campo eléctrico y sensores de campo magnético. Los DDAU`s se pueden sincronizar por GPS como una red inalámbrica con el apoyo de Zigbee.

En 2013 S.S.Sadistap, S.A.Akbar, B.A.Botre [17], -implementaron un sistema robótico móvil de monitoreo para el seguimiento de plantas en agricultura donde adquiriría datos de humedad y temperatura del suelo, en donde el robot por medio de su brazo basado en un servo recogía muestras de suelo. El algoritmo de lazo cerrado basado en el controlador PID digital hecho en lenguaje C ha sido desarrollado para la precisa posición y control de velocidad del robot, adicional se implementó el control inalámbrico y la adquisición de datos mediante el protocolo Zigbee en donde visualizaba lo datos mediante una interfaz en framework y el sistema de control diseñado por Matlab.

# CAPÍTULO 5

## 5. Metodología de Diseño y Construcción

En este capítulo se verán los esquemas necesarios para la construcción del proyecto, además una breve descripción del programa Matlab y algunas de sus herramientas, el programa de diseño electrónico con licencia libre KiCad que fue necesario para la creación de esquemáticos y el diseño de PCB's de los circuitos, además se verá una pequeña introducción en la configuración de los módulos ZigBee necesarios para la transmisión inalámbrica de los datos.

### 5.1 Diseño Metodológico

Por medio de la investigación de los elementos que se van a implementar para la construcción del Sistema de Presión de control, se procede a realizar el diseño de un prototipo de circuitos impresos (Sistema Modular), que se utilizara para la adquisición y el control del sistema de presión del agua. El Sistema lograra la adquisición de la señal de presión por medio del sensor de presión diferencial, la transmisión de la información será de forma inalámbrica por medio del protocolo ZigBee, la interfaz y visualización de los datos se trabajara con Matlab. Mediante la búsqueda de elementos que podrían ser útiles para la construcción de los circuitos y el sistema de control algunos otros elementos que deberán conseguirse para la integración de dos sistemas, adicionalmente el del hardware de acople al micro-controlador con la tecnología ZigBee. De aquí se procederá a realizar el diseño e implementación de elementos para la identificación del sistema y experimentar con la programación sobre el micro para obtener experiencia en su manejo. En el momento en que se dé por terminada la interfaz en Matlab, se procede a realizar las respectivas pruebas y emplear el sistema modular para verificar las señales recibidas, en cuanto se verifique el total funcionamiento de los módulos en conjunto, se empezarán las pruebas para optimizar el sistema de control. Durante el desarrollo del proyecto se irán tomando datos que se estarán acumulando para presentarlos dentro del libro correspondiente a esta tesis.

### 5.2 Construcción del Modelo de Control

Se diseñó un prototipo de una planta para presión de agua en cual se incluyeron todos los elementos posibles para poder realizar variaciones según las características de la bomba que se compró para el proceso.

Para esto se compraron los siguientes elementos:

- Tabla para base de 38 x 62cm aglomerada.
- Bomba de agua para parabrisas de 24 Vdc.
- Recipiente plástico de 3 litros
- Registro de bola inox. de 3/8.

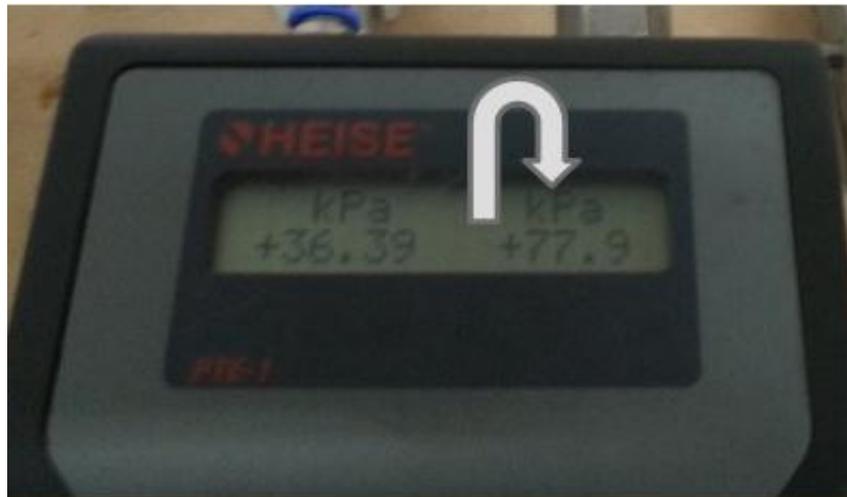
- Válvula de aguja en A. inox. de ¼
- tee inox .de ¼
- tee inox. de 1/8
- tubing inox. de 1/4 x 1 metro.
- unión Inox. de ¼
- Siete tuercas inox. con felulas y contrafelulas.
- Tres tuercas inox .con felulas y contrafelulas.
- Soporte de 1 vía aluminio
- Flanche en plástico de ½ de salida.
- Tres Racores en bronce de 3/8.
- Fungibles y consumibles (mangueras y abrazaderas plásticas).

### 5.2.1 Especificaciones de los Elementos Principales

#### 1. Bomba

La bomba se consiguió como para limpiaparabrisas a 24 voltios DC con un caudal de salida de 3.2 litros por minuto y según mediciones con un equipo patrón de presión (HEISE) Figura 3.1, maneja una presión de 20 KP a 80 KP.

**Figura 3.1: Equipo Patrón HEISE**



**Fuente: Autores**

## 2. Registro de Bola de 3/8

Figura 3.2: Registro de bola de 3/8"



Fuente: Autores

A continuación se muestran las características del registro de bola de 3/8".

- Cuerpo y tapa forjado: Latón EN 12165: 1998 CW617N
- Bola: Latón, macizo EN 12165: 1998 CW61N
- Vástago: Latón EN12164: 1998 CW614N
- Mango: Acero o Aluminio
- Tuerca (tornillo) para mango de acero (aluminio): Acero (Latón)
- Asiento: PTFE puro
- Anillo para Vástago: O-ring in VITON/NBR
- Sellador de rosca: Loctite 2701 (din 30661)
- N.B.: Todas las Válvulas tienen el vástago anti expulsión, montado interiormente.

## 3. Válvula de Aguja INOX. de 1/4".

Figura 3.3: Válvula de aguja



Fuente: Autores

- **Presión de trabajo:** Hasta 10 000 psig (689 bar)
- **Temperatura:** -65 A 1200 ° F (-53 a 648 ° C)
- **Material del cuerpo:** 316 de acero inoxidable, aleación 400, aleación 20, aleación 600, aleación C-276, de titanio
- **Material de Embalaje:** PTFE teflón asientos
- **Conexiones finales:**
  - **Tipo:** Racores Swagelok, NPT, para soldar el tubo, para soldar tubería, VCR macho ®, VCO accesorio macho ® ajuste
  - **Tamaño:** 1/8 a 1 pulgada; 6 a 12 mm
- **Características**
  - Aguja Unión Válvula Cutaway
  - Diseños madre ° no giratorio bola (NB)
  - Regulación (NR)
  - PCTFE ° regulación de asiento blando (NKR)
  - PTFE ° regulación de asiento blando (NTR)
  - Tamaños de orificio 0,156 a 0,437 pulgadas (4,0 a 11,1 mm)
  - Construcción de la Unión capó impide el desmontaje accidental de la válvula
  - Los pernos de embalaje diseño permite embalaje ajuste en la posición abierta
  - Seguridad de vuelta abrasador focas en la posición totalmente abierta
  - Recto y flujo ángulo patrones
  - Accionamiento manual y neumático

#### 4. TEE UNIÓN DE ¼” Y 1/8” INOX.

**Figura 3.4: Tee unión de ¼”**



**Fuente: Autores**

- EClass (4.1) 37020716
- EClass (6.0) 37-02-05-06
- Material del cuerpo Acero Inoxidable 316
- Proceso de limpieza y embalaje estándar (SC-10)
- Tamaño de conexión 1: 3/8 pulg.
- Tamaño de conexión 2: 1/4 pulg.
- Tipo de conexión 1 Racor Swagelok®
- Tipo de conexión 2 NPT macho
- UNSPSC (11.0501) : 40142605
- UNSPSC (4.03) :40141729
- UNSPSC (SWG01) :40141729

## 5. Unión de 1/4”

**Figura 3.5: Unión de 1/4”**



**Fuente: Autores**

**Tabla 3.1: Características de la unión de 1/4”**

Configuración	Unión
Material del cuerpo	Acero Inoxidable 316
Proceso de limpieza	Limpieza y embalaje estándar (SC-10)
Tamaño de conexión 1	1/4 pulg.
Tamaño de conexión 2	1/4 pulg.
Tipo de conexión 1	Accesorio Swagelok® de alta presión
Tipo de conexión 2	Accesorio Swagelok® de alta presión
UNSPSC (4.03)	40141700

**Fuente: Autores**

## 6. TUBO INOXIDABLE DE ¼ “

Las presiones de servicio admisibles están calculadas en base a las ecuaciones de ASME B31.3 y ASME B31.1, para tubo en ISO 1127, (Tolerancia D4, T4 para tubo de 3 a 12 mm; Tolerancia D4, T3 para tubo de 14 a 50 mm), usando un valor de esfuerzo de 137,8 MPa (20 000 psi) y una resistencia a la tracción de 516,4 MPa (74 900 psi), salvo excepciones.

**Figura 3.6: Tubería de ¼”inoxidable**



**Fuente: Autores**

### 5.2.2 Procedimiento de Construcción del Modelo de Control

1. Se instala flanches de salida de ½ pulgada al recipiente plástico con sus respectivos empaques como se muestra en las figuras 3.7 y 3.8:

**Figura 3.7 Flanches**



**Fuente: Autores**

**Figura 3.8: Recipiente plástico respectivamente.**



**Fuente: Autores**

2. Se instalan racores en bronce de 3/8" al registro y al flanche de salida del tanque para luego acoplar con la manguera plástica de 3/8" y sus respectivas abrazaderas como se visualiza en la figura 3.9.

**Figura 3.9: Foto de acoples y mangueras.**



**Fuente: Autores**

3. Se asegura la bomba de agua para parabrisas en la base y se conecta la manguera que viene del registro de 3/8" como se muestra en la figura 3.10.

**Figura 3.10: Bomba de agua para parabrisas**



**Fuente: Autores**

4. Se instala la válvula de aguja de  $\frac{1}{4}$ " con los racores de  $\frac{1}{4}$ " y la salida de la bomba con una manguera de  $\frac{1}{4}$ " plástica como se visualiza en la figura 3.11.

**Figura 3.11: Válvula de Aguja de  $\frac{1}{4}$ "**



**Fuente: Autores**

5. Se realiza el modelo de Venturi con un tramo de tubo de  $\frac{1}{4}$ " y de un  $\frac{1}{8}$ " de diámetro se introduce el tubo de  $\frac{1}{8}$ " al de  $\frac{1}{4}$ " y realiza soldadura en las dos puntas para generar el modelo de Venturi. Se instala la tee inox. De  $\frac{1}{8}$ " en el tramo de tubo para realizar la conexión del sensor y la otra parte se instala la tee inox. De  $\frac{1}{4}$ " para la otra señal, esto se visualiza en la figura 3.12.

**Figura 3.12: Conexiones modelo Venturi**



**Fuente: Autores**

6. Se instala en la punta la unión para conectar la salida con el tubo de  $\frac{1}{4}$ " hasta el tanque para terminar el proceso asegurándolo con un soporte de una vía en aluminio. Esto se muestra en figura 3.13.

**Figura 3.13: Soporte y tubería prototipo.**



**Fuente: Autores**

### 5.2.3 Montaje Final Del Prototipo

**Figura 3.14: Montaje final del Modelo de control**



**Fuente: Autores**

## 5.3 MATLAB

MATLAB es un lenguaje de programación de alto nivel, que tiene un enfoque directo hacia la computación científica, más concretamente es un programa de cálculo numérico con gran número de instrucciones dirigidas a la resolución de problemas científicos, adicional que dependiendo de su versión posee herramientas para procesamiento de imágenes, modelamiento de análisis financiero, estadística, desarrollo de aplicaciones etc. En sus diferentes versiones se van concatenando las herramientas anteriores además de la creación de interface (GUI), comunicación con programas en otros lenguajes y otros dispositivos de software. El nombre de MATLAB proviene de las palabras inglesas *Matrix Laboratory* (laboratorio de matrices), que evidentemente dan una idea de la utilidad primordial de dicho programa. Algunas de las Cajas de Herramientas del programa MATLAB se ven en la Tabla 3.1:

**Tabla 3.2: Caja de Herramientas y Paquetes de Bloques de Matlab**

<b>MATLAB (Cajas de herramientas)</b>	<b>Simulink</b>
Matemáticas y Optimización	Modelado de punto fijo
Estadística y Análisis de datos	Modelado basado en eventos
Diseño de sistemas de control y análisis	Modelado físico
Procesado de señal y comunicaciones	Gráficos de simulación
Procesado de imagen	Diseño de sistemas de control y análisis
Pruebas y medidas	Procesado de señal y comunicaciones
Biología computacional	Generación de código
Modelado y análisis financiero	Prototipos de control rápido y SW/HW HIL
Desarrollo de aplicaciones	Tarjetas integradas
Informes y conexión a bases de datos	Verificación, validación y comprobación

**Fuente: Autores**

### 5.3.1 Identificación de sistemas

Luego de haber realizado la construcción del modelo de control se debe proceder a la identificación del sistema, que en el caso de este proyecto se realizó mediante Ident. Luego de esto se debe proceder a hallar la función de transferencia del sistema y pasar la respectiva función de transferencia a la herramienta de Matlab llamada Sisotool y revisar el Overshoot y luego digitalizar el controlador mediante la transformada Z. La identificación de sistemas permite construir modelos matemáticos de sistemas dinámicos basándose en datos obtenidos de la experimentación, esto se logra ajustando parámetros de un modelo dado hasta que su salida coincida lo más cercano posible con la salida medida. A continuación se muestran los pasos del proceso de identificación:

1. Recolectar datos de entrada y salida del sistema de control
2. Examinar los datos, pulirlos para eliminar el ruido excesivo y seleccionar porciones útiles de los datos originales.
3. Hallar la Función de Transferencia mediante Ident de Matlab.
4. Pasar la función de transferencia a Sisotool para poder hallar el controlador del sistema y revisar el Overshoot.
5. Digitalizar el Controlador mediante transformada Z
6. Luego de tener el controlador en discreto realizar simulación mediante Simulink.

El System Identification Toolbox de MATLAB ofrece varias funciones para simplificar los pasos anteriores, adicional que permite crear una interfaz gráfica GUI para realizar la identificación de forma interactiva. Se usó la versión de Matlab 2013a en el diseño del sistema y su identificación.

De manera ejemplar se describirán a continuación los pasos para obtener el modelo matemático de una planta de control mediante adquisición de datos y la herramienta System Identification de Matlab.

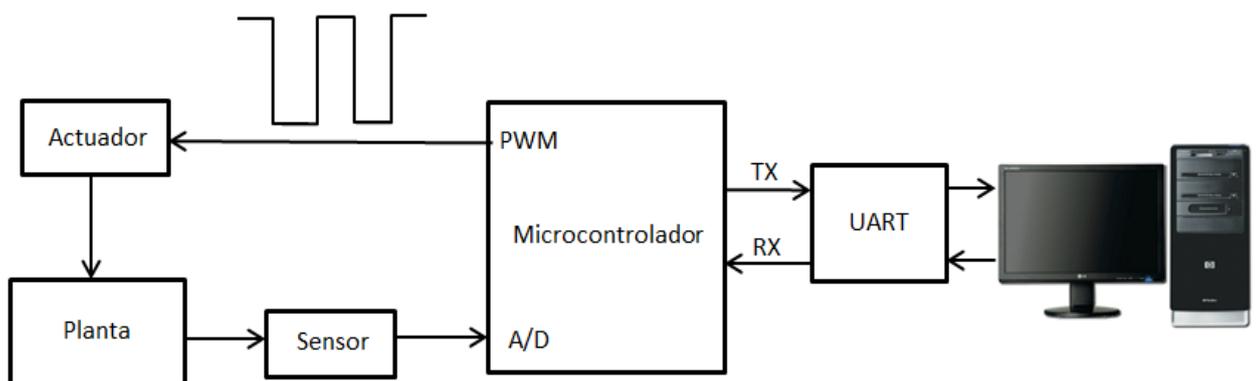
- 1. Recolectar datos de entrada y salida de la planta:** El primer paso para la identificación de la planta es la adquisición de los datos provenientes del experimento que consiste en generar una señal de estímulo al sistema y adquirir la respuesta del mismo ante dicho estímulo. La buena identificación de la planta depende de la correcta adquisición de datos que significa una buena representación del sistema.
- 2. Examinar los datos, pulirlos para eliminar el ruido excesivo y seleccionar porciones útiles de los datos originales:** Cuando los datos de entrada y salida estén libres de ruido se procede a seleccionar el modelo matemático que represente la dinámica de la planta. En este paso mediante los datos obtenidos se empieza a diseñar el modelo de control con Matlab para hallar su función de transferencia.
- 3. Hallar la Función de Transferencia mediante Ident de Matlab:** Como en el paso anterior se indica, para poder hallar la función de transferencia por medio de Matlab se usará la herramienta Ident y así poder calcular el mejor modelo en base a los datos obtenidos.
- 4. Pasar la función de transferencia a Sisotool para poder hallar el controlador del sistema y revisar el Overshoot:** Una vez obtenido el modelo matemático por medio de la herramienta Sisotool se procede a analizarlo y revisar el Overshoot del sistema.
- 5. Digitalizar el Controlador mediante transformada Z:** Luego del necesario análisis del Overshoot, se procede a digitalizar el controlador por medio de la transformada Z.
- 6. Luego de tener el controlador en discreto realizar simulación mediante Simulink:** Antes de pasar el controlador a Simulink, se debe revisar la frecuencia de muestreo para evitar un aliasing que como se indicó en el capítulo 2, para evitarlo se usa un tipo de filtro pasa bajo que elimina todas las señales de mayor frecuencia a la que se está midiendo. Luego de tener el controlador en discreto se pasará a Simulink para poder llevar a cabo la respectiva simulación del sistema.

## 5.4 Adquisición de datos para Identificación del Sistema

Para cumplir el primer pasó en la identificación de sistemas es necesaria la adquisición de datos ya que de otra forma no se podrían recolectar datos para luego identificarlos mediante Matlab. Como no se conoce la función de transferencia del sistema es necesario realizar un proceso de identificación con el cual llegue a la función de transferencia del sistema o a una muy buena aproximación de la misma.

El siguiente diagrama de bloques (Figura 3.15) ilustra el proceso que se realizó para adquirir los datos que posteriormente fueron analizados con la herramienta Toolbox de MATLAB IDENT fue el siguiente:

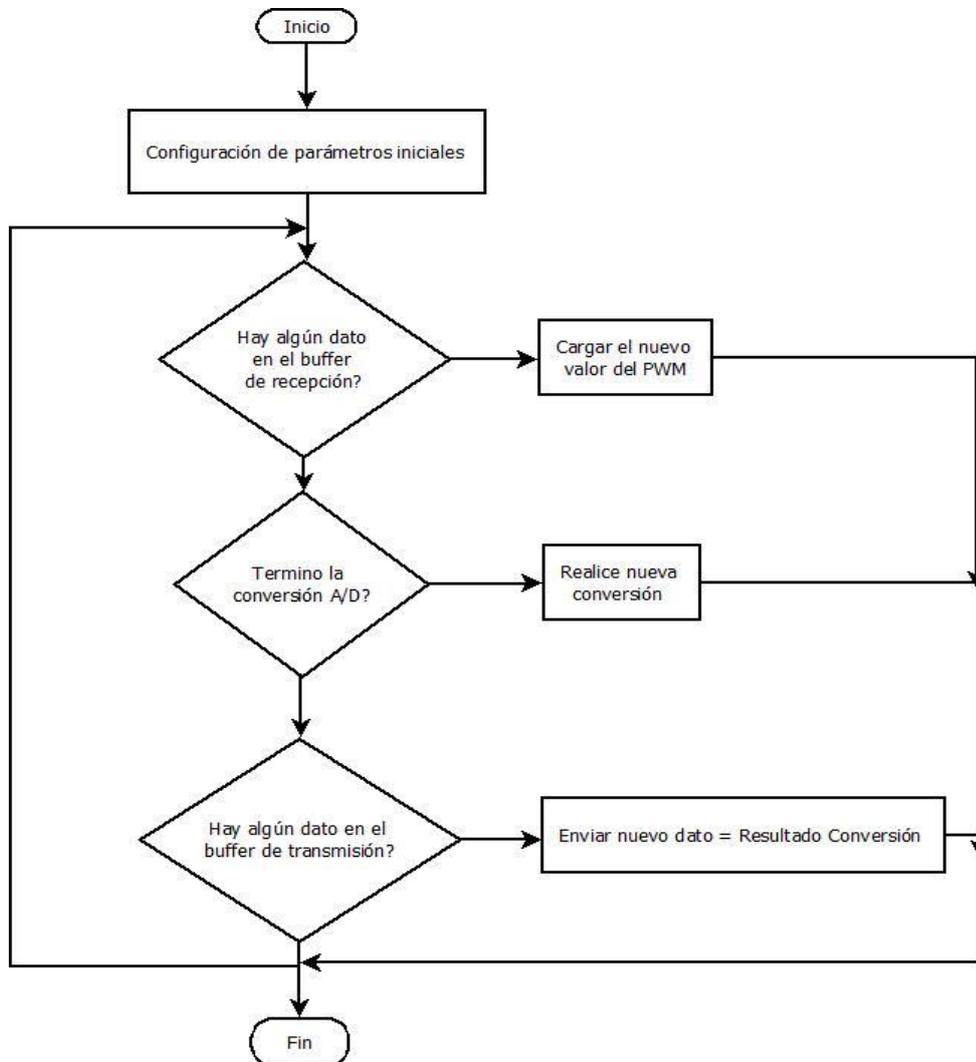
**Figura 3.15: Diagrama de bloques proceso de adquisición de datos**



**Fuente: Autores**

El microcontrolador del diagrama anterior debe realizar la conversión A/D para transmitir este dato al PC y recibir del PC el parámetro con el cual va a cargar el PWM, cabe resaltar que la transmisión y recepción es la interfaz entre el módulo USART del Microcontrolador y el módulo ZigBee. El siguiente diagrama de flujo (Figura 3.16) ilustra el proceso de la solución de software que implemento:

**Figura 3.16: Diagrama de flujo del software implementado**



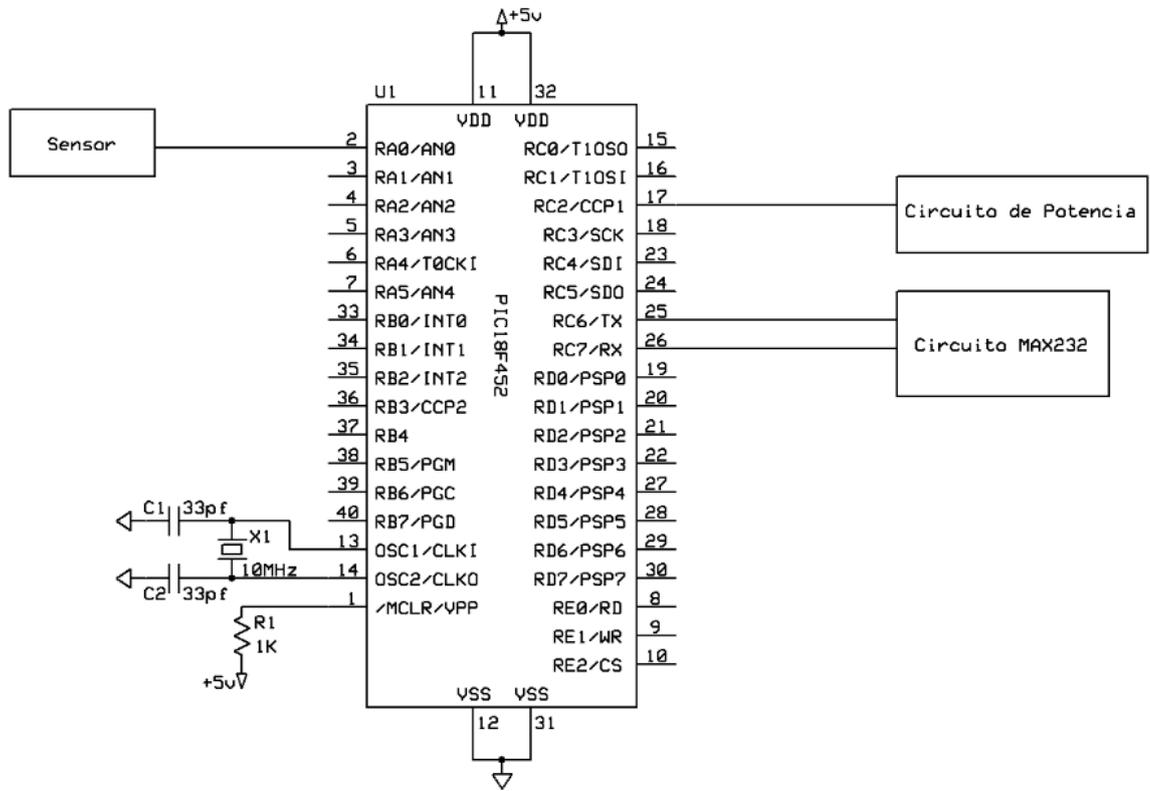
**Fuente: Autores**

La conversión A/D en el microcontrolador está configurada a 8 bits. La comunicación serial está configurada de la siguiente manera:

- Baudios = 115200
- Bits de datos = 8
- Paridad = Ninguna
- Bits de parada = 1
- Control de flujo = Ninguno

El siguiente es el plano del circuito propuesto para la adquisición de datos usando Matlab®:

Figura 3.17: Plano del circuito propuesto de adquisición de datos



Fuente: Autores

El script que se utiliza en Matlab®: para la adquisición de datos es el siguiente:

%% Configuración del puerto

```

clc, disp('Start');    %% Indica que se inició el script
SerPIC = serial('COMxx'); %% Reemplazar xx por el número de puerto Serial
% Parámetros de configuración serial
set(SerPIC, 'BaudRate', 115200);
set(SerPIC, 'DataBits', 8);
set(SerPIC, 'Parity', 'none');
set(SerPIC, 'StopBits', 1);
set(SerPIC, 'FlowControl', 'none');
SerPIC.terminator = { " " };
SerPIC.BytesAvailableFcnCount = 1;
SerPIC.BytesAvailableFcnMode = 'byte';
SerPIC.timeout = 2;
fopen(SerPIC);

```

%% Enviar Datos

```

y = [];
x = [];
% Limpiar el bufer
fclose(SerPIC);
fopen(SerPIC);
vout = 0;
for k=1:8
    pwm = 210;
    fwrite(SerPIC,pwm)
    for i=1:100
        tic
        x = [x;pwm];
        vout = (mean(fread(SerPIC,SerPIC.inputbuffersize))*(5/255));
        y = [y;vout];
        toc
    end
    pwm = 0;
    fwrite(SerPIC,pwm);
    for i=1:100
        tic
        x = [x;pwm];
        vout = (mean(fread(SerPIC,SerPIC.inputbuffersize))*(5/255));
        y = [y;vout];
        toc
    end
end

% Gráfica de la señal de entrada y salida
xt = x*(5/210);
plot(xt,'r','LineWidth',2)
hold on
plot(yt,'LineWidth',2)
grid on
axis([0 800 -1 7])

legend('Señal de entrada','Señal de salida')

%% CERRAR el puerto COM
fclose(SerPIC);
delete(SerPIC)
clear SerPIC
disp('Fin...')

```

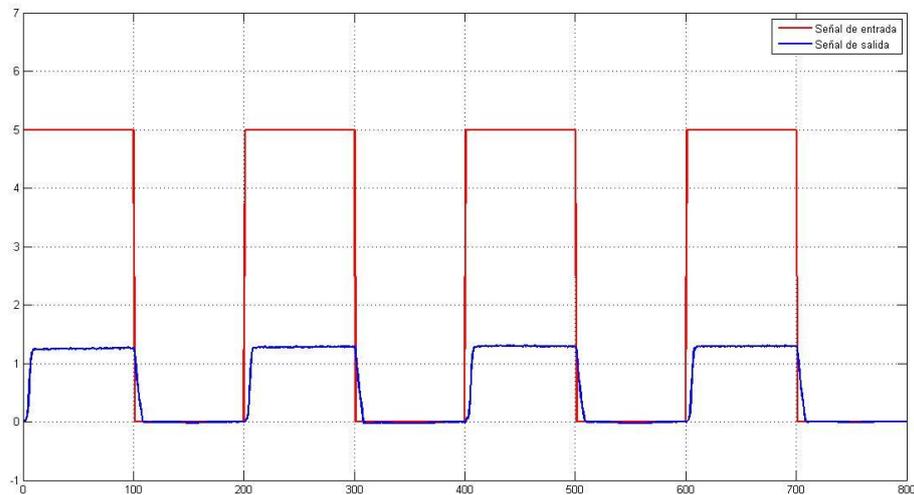
Los datos que son adquiridos pueden ser luego utilizados con las herramientas Toolbox del *Ident* para realizar la identificación del sistema. Después de ejecutar el script los datos a usar son:

- “xt” = Datos de entrada al planta.
- “y” = Datos de salida de la planta.

#### 5.4.1 Resultados de la adquisición de datos

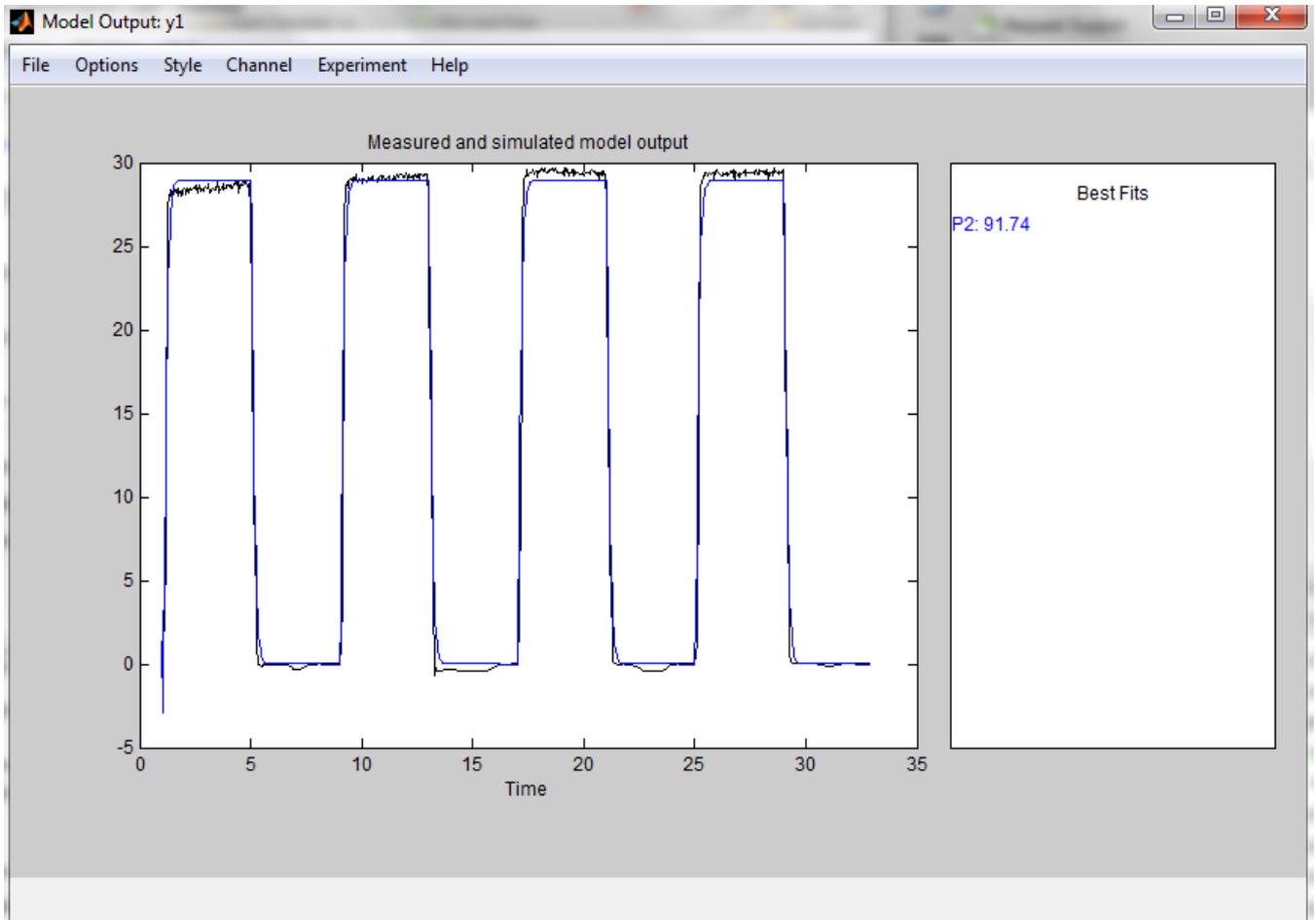
Como señal de entrada se implementó una señal pulsante que va de 0 a 5 voltios, los resultados de la adquisición son mostrados a continuación en la Figura 3.18:

**Figura 3.18: Comparación señal de entrada vs salida**



La línea roja indica la señal de entrada, mientras que la azul indica la salida del sistema. Con estas dos señales ya en el workspace se invocó la herramienta Toolbox del *Ident* de Matlab®. Después de varios ensayos, con diferentes funciones de transferencia de diferente orden se encontró la siguiente aproximación, la cual está un 91.74% correlacionada con la señal original.

**Figura 3.19: Ventana del Ident y función de transferencia**



**Fuente: Autores**

Dicha función de transferencia encontrada luego de adquirir los datos y utilizar la herramienta de Ident es:

$$G(s) = \frac{35.52}{s^2 + 23.63s + 139.5} \text{ Ec.23}$$

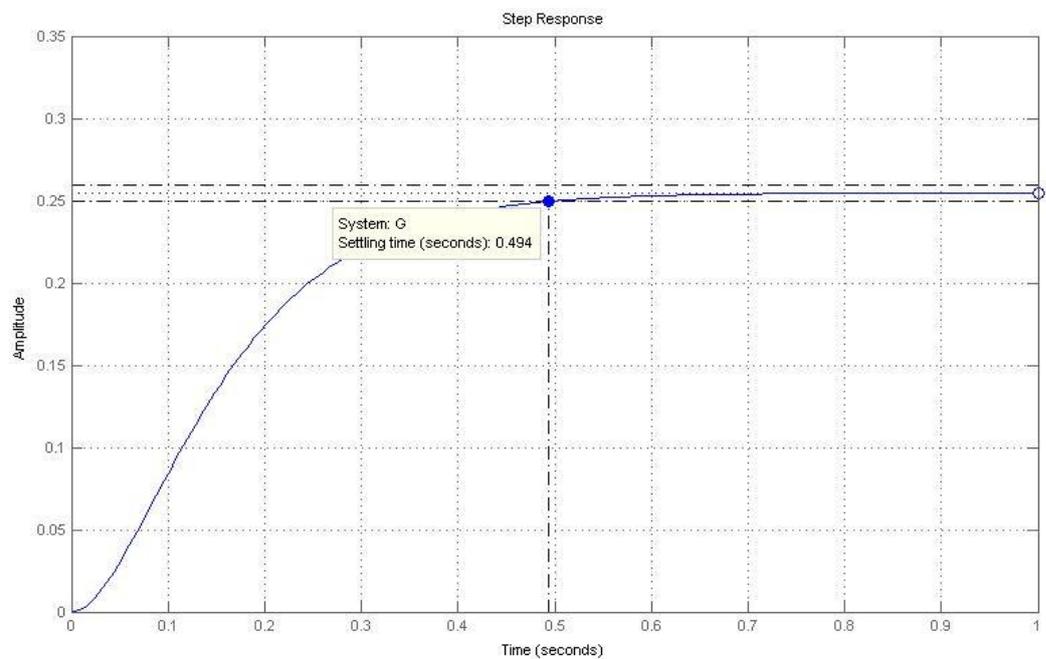
### 5.4.2 Diseño del Controlador

Como ya se tiene una función de transferencia, es posible diseñar un controlador usando el (LGR) lugar geométrico de las raíces. A continuación se enumeran los pasos desarrollados para llegar a la solución propuesta en este trabajo que es un controlador digital:

1. **Evaluar la respuesta del sistema en lazo abierto.** La función de transferencia al no tener polos en lado derecho del LGR indica que es un sistema estable; sin embargo, tiene un error de estado estacionario diferente de 0.

**Figura 3.20: Respuesta de la señal en Lazo Abierto**

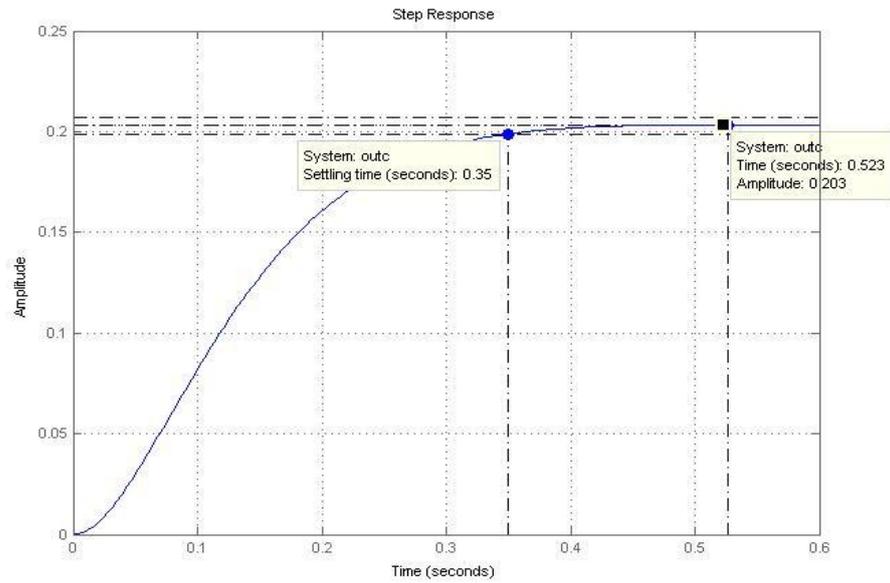
**Fuente: Autores**



Como se observa en la Figura 3.20 se observa un error estado estacionario de un 75% y un tiempo de respuesta de 0.494 segundos. Lo último indica que la planta es lenta en su respuesta.

2. **Evaluar la respuesta en lazo cerrado:** Se analiza ahora la respuesta del sistema ya en lazo cerrado.

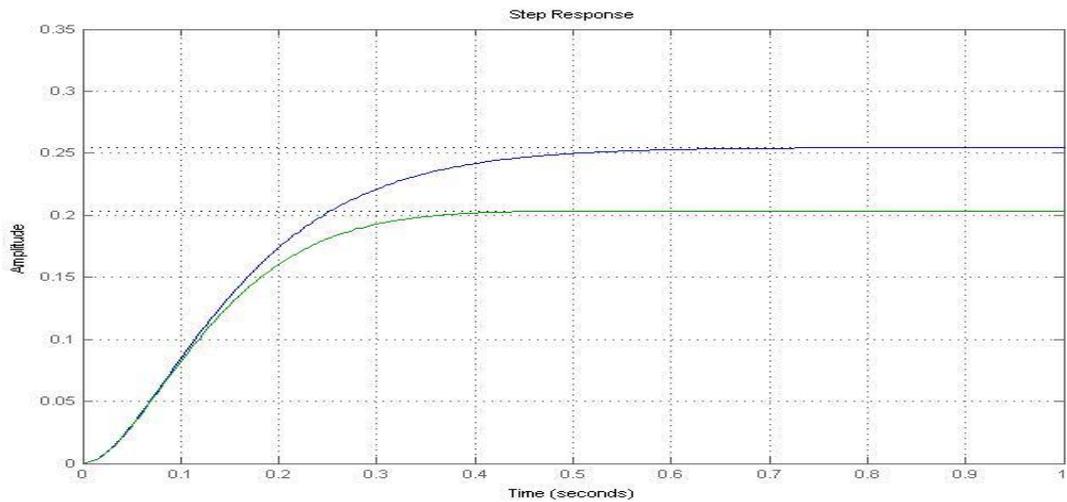
**Figura 3.21: Respuesta de la señal en Lazo Cerrado**



Fuente: Autores

Ahora se comparan las dos salidas lazo abierto y lazo cerrado.

**Figura 3.22: Comparación Salida Lazo Cerrado vs Lazo Abierto**

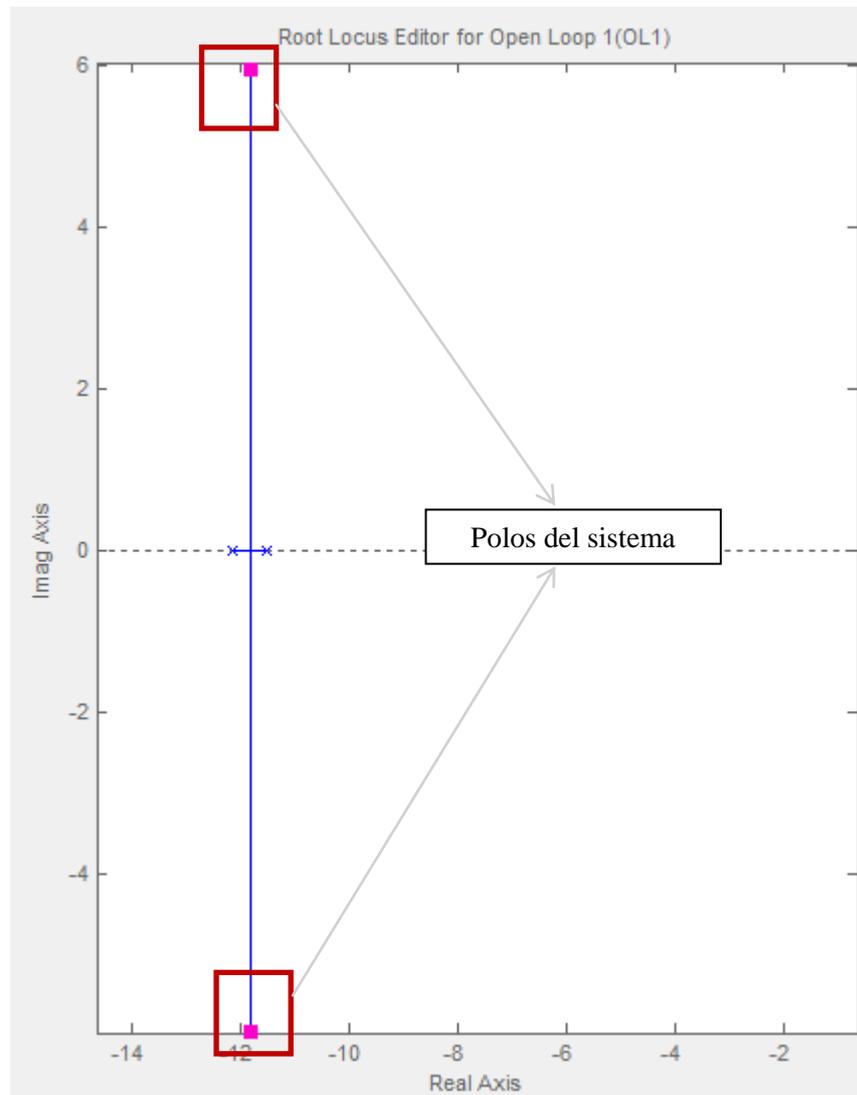


Fuente: Autores

### 3. SISOTOOL

Para elaborar el controlador se usó la Toolbox de MATLAB llamada SISOTOOL.

**Figura 3.23: Polos del sistema en lazo cerrado**



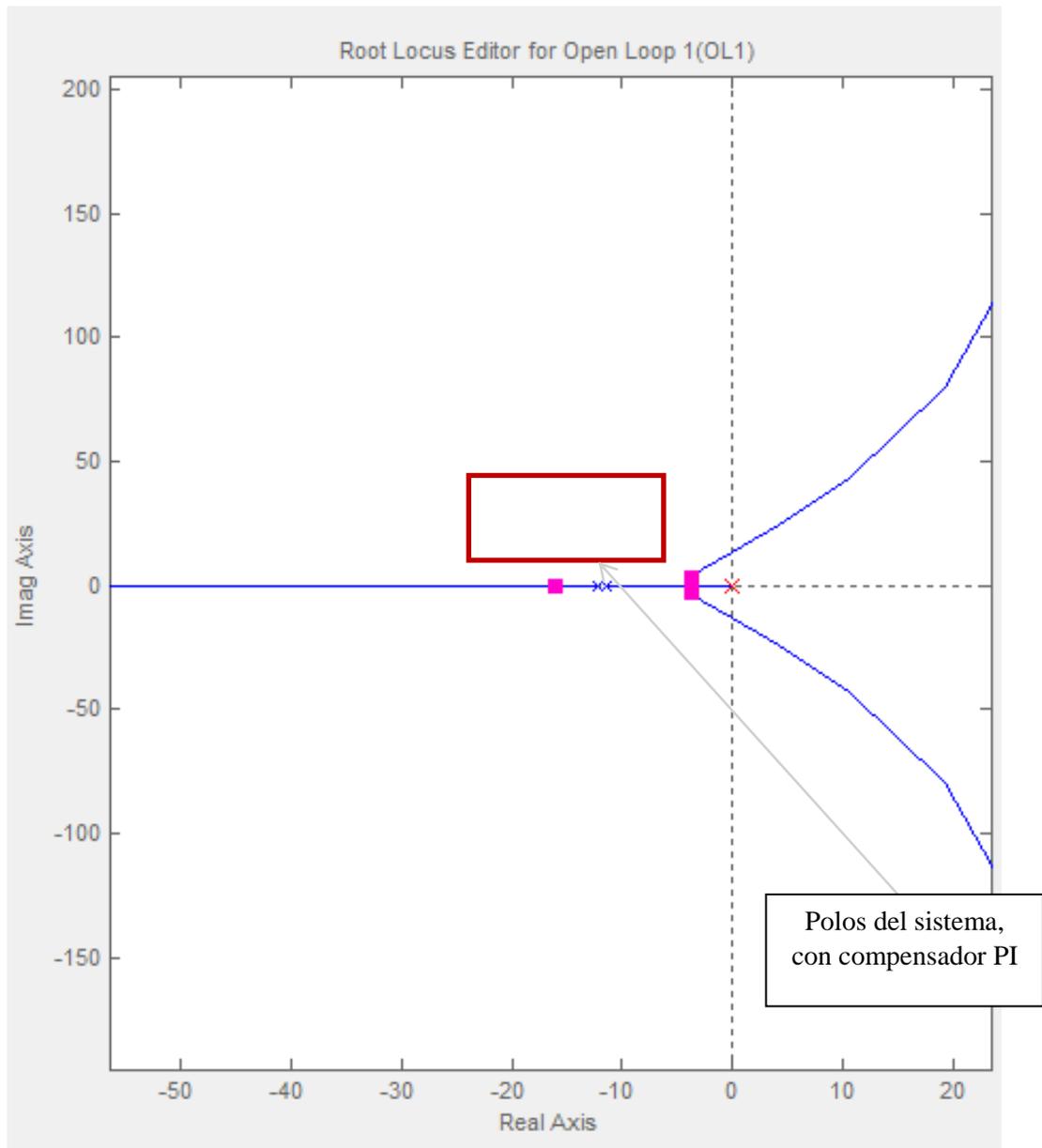
**Fuente: Autores**

En la Figura 3.23 se observa el lugar inicial (sin compensador) donde se encuentran los polos del sistema en lazo cerrado.

Luego de sintonizar el controlador con la finalidad de eliminar el error de estado estacionario, se llega a la siguiente solución:

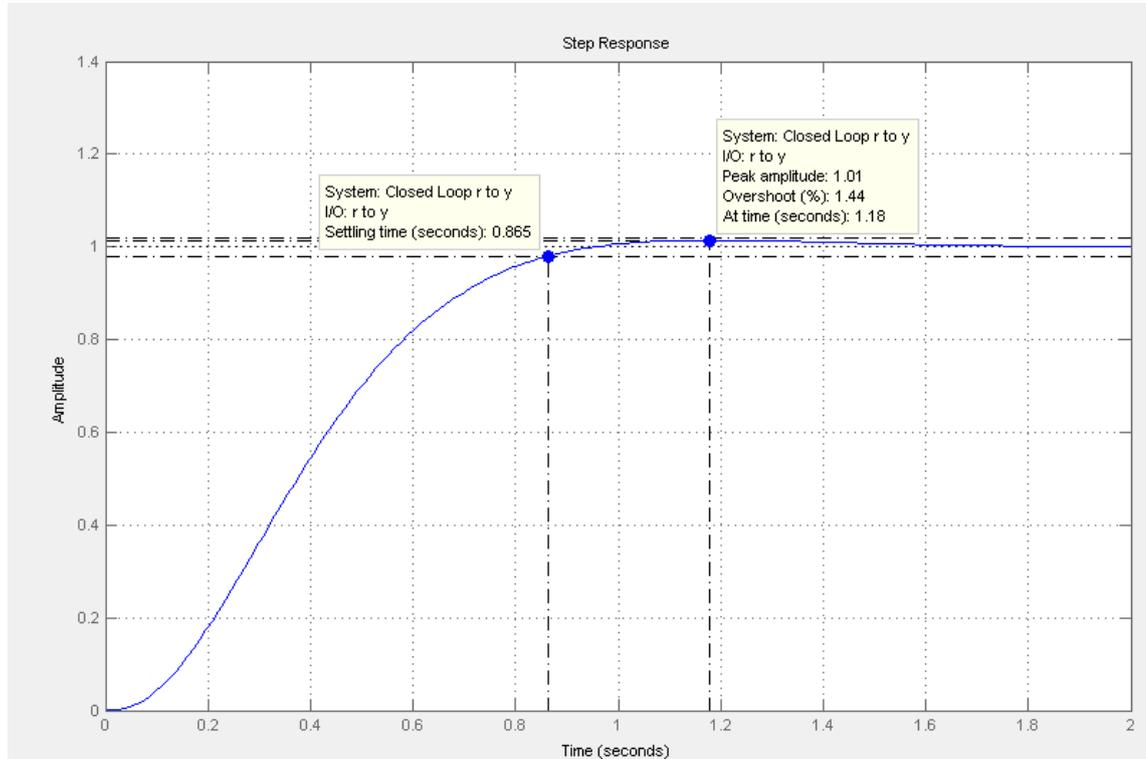
$$C(s) = \frac{0.1(s+100)}{s} \quad \text{Ec.24}$$

**Figura 3.24: Polos del Sistema con Compensador PI**



**Fuente: Autores**

**Figura 3.25: Salida del Sistema Compensada mediante el Controlador Seleccionado**

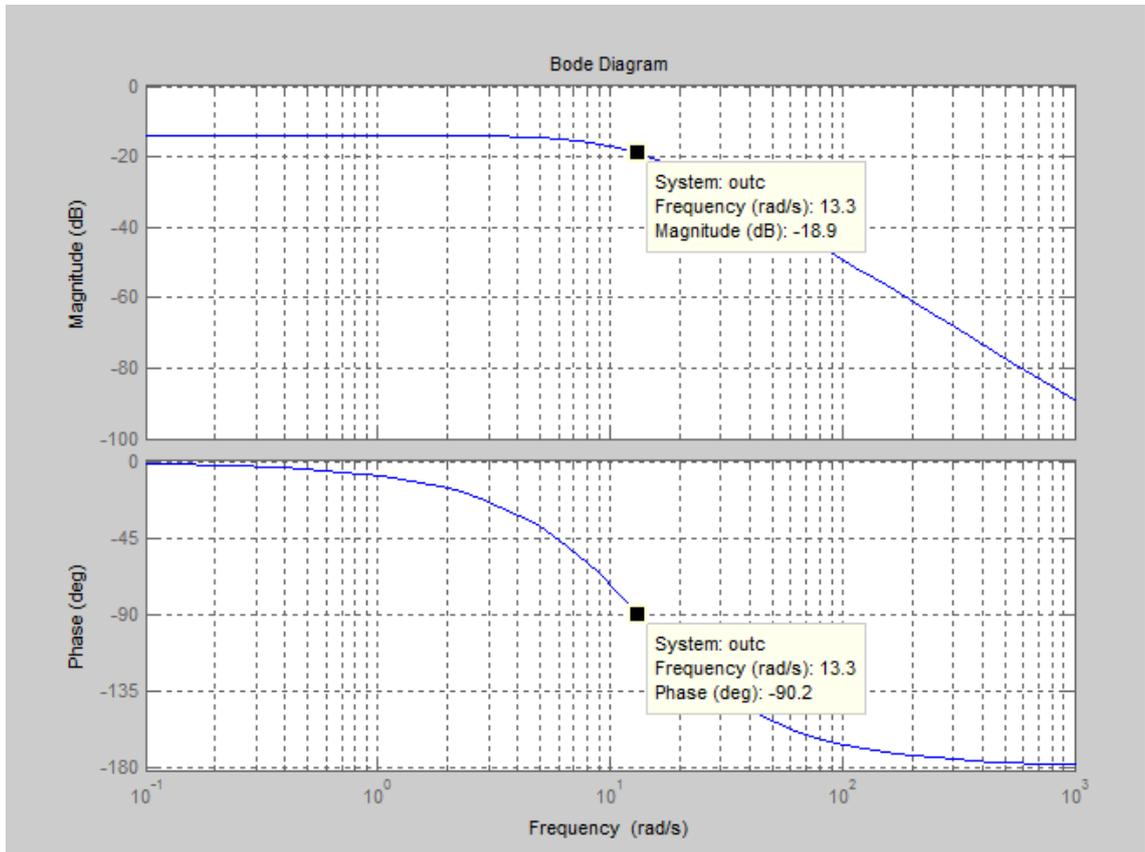


La Figura 3.25 ilustra la salida del sistema ya compensada con el controlador (C(s)) encontrado con la ayuda de la herramienta Toolbox de SISOTOOL. Se observa que efectivamente se eliminó el error de estado estacionario pero debido al efecto integrador el tiempo de respuesta aumenta a 0.865.

4. **Análisis de Bode:** Como lo que se busca es implementar un controlador digital y no uno analógico como el encontrado anteriormente es necesario realizar un análisis en frecuencia con la finalidad de encontrar la frecuencia de muestreo mínima del sistema y evitar el aliasing.

Se analiza entonces el bode del sistema sin compensador ( $k = 1$ ).

**Figura 3.26: Frecuencia Natural del sistema expresada en radianes**



**Fuente: Autores**

En la figura 3.26 se observa que la frecuencia natural del sistema expresada en radianes por segundos es 13.3 aproximadamente. Esta frecuencia sucede cuando la ganancia disminuye en 3dB y el ángulo de fase es  $90^\circ$ . Esta frecuencia esta expresada en radianes por segundo y se necesita pasar a Hertz por lo que es necesario dividirla por  $2\pi$ .

$$W_n \left( \frac{rad}{s} \right) = 13.3 \quad \text{Ec.25}$$

$$f(hz) = \frac{13.3}{2 * \pi} = 2.11Hz \quad \text{Ec.26}$$

La frecuencia de 2.1Hz es la frecuencia natural del sistema según el teorema de Nyquist la frecuencia de muestreo debe ser mayor o igual que dos veces la frecuencia natural del sistema [referencia]. En este caso se escogió 50 veces mayor lo que da una frecuencia de 105.83Hz que equivale a un periodo de muestreo de 9.4 ms.

Otra forma de verificar esta frecuencia es haciendo uso de los siguientes comandos.

```

[wn,z,p] = damp(outc);
Wmax = max(wn);
disp(Wmax)
w = (Wmax) / (2*pi);
wopt = w*50;
Topt = 1/wopt;
disp(Topt)

```

Donde outc es el sistema en lazo cerrado con ganancia de  $K = 1$ .

### 5. Hallar el controlador en discreto (Transformada Z)

Luego de encontrar el periodo de muestreo es posible hallar el controlador digital, por medio del comando c2d de MATLAB, que tiene como parámetros el controlador en continuo y el periodo de muestreo.

$$Cd = c2d(C, 0.001); \text{ Ec.27}$$

El anterior comando dio como resultado el siguiente controlador:

$$C(z) = \frac{0.1(z-0.5)}{z-1} \text{ Ec.28}$$

### 6. Ecuación de Diferencias

De la siguiente ecuación que muestra la forma de un control PID en discreto es posible encontrar la ecuación de diferencias con la cual se programe el microcontrolador.

$$C(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \text{ Ec.29}$$

Ecuación de diferencias:

$$u_k = u_{k-1} + q_0 e_k + q_1 e_{k-1} + q_2 e_{k-2} \text{ Ec.30}$$

Donde  $e_k$  es el error actual, y  $u_k$  es la salida al actuador.

Si se divide por  $z$  el controlador hallado se obtiene lo siguiente:

$$C(z) = \frac{0.1 - 0.05z^{-1}}{1 - z^{-1}} \quad \text{Ec.31}$$

Lo que indica que:

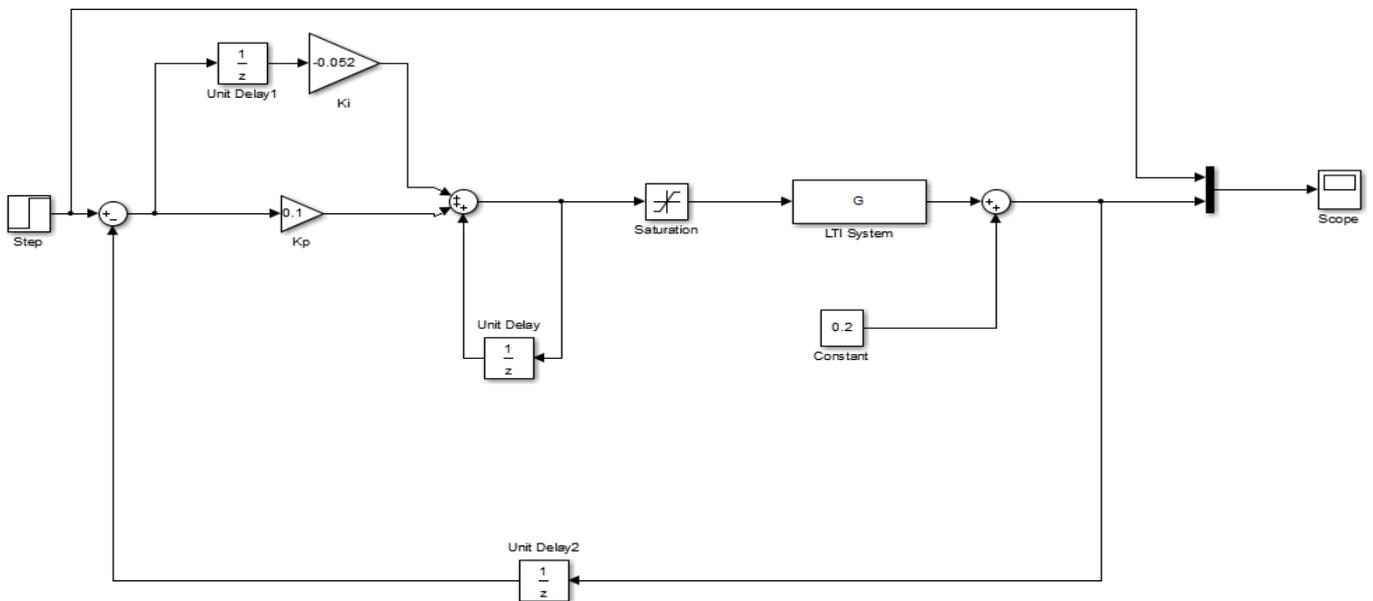
$$q_0 = 0.1 \text{ (Ganancia Proporcional)}$$

$$q_1 = -0.05 \text{ (Ganancia Integral)}$$

### 5.4.3 Simulación en Simulink

Finalmente después de hallar la función de transferencia del sistema y el controlador se procede a simular el sistema en SIMULINK.

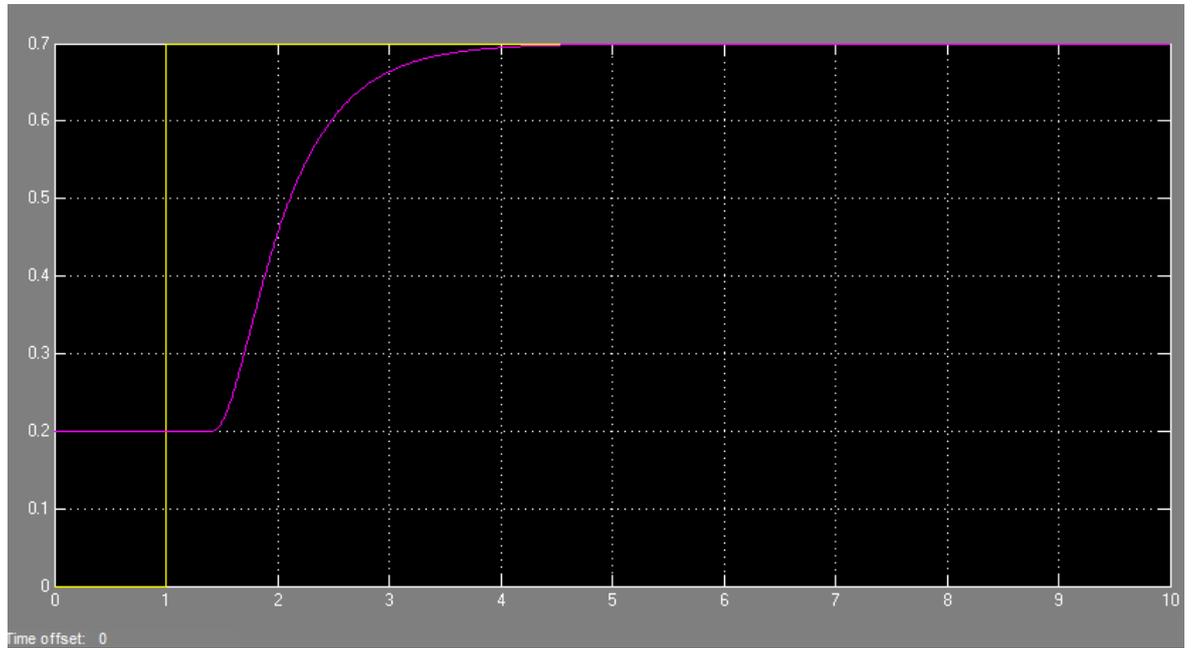
**Figura 3.27: Diagrama de simulación del sistema de control**



**Fuente: Autores**

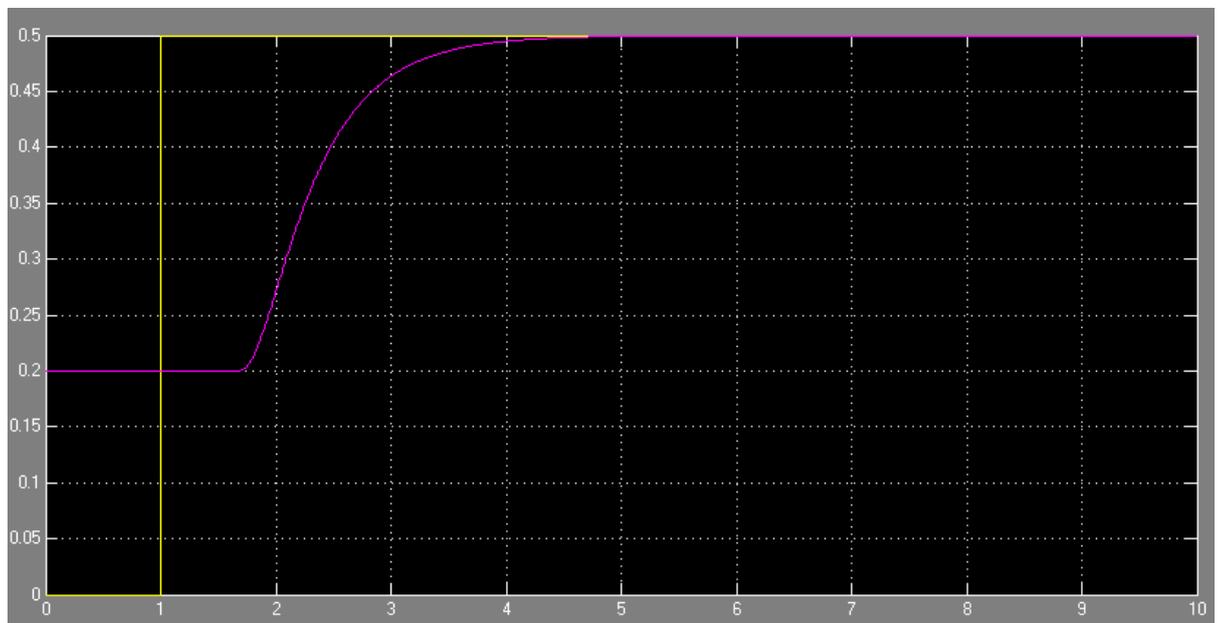
A continuación se pueden visualizar las figuras 3.28 y 3.29 que muestran las señales de salida del sistema de control ya simulado en la herramienta Simulink de Matlab®.

**Figura 3.28: Salida del Sistema Controlada en *Set point* de 0.7**



**Fuente: Autores**

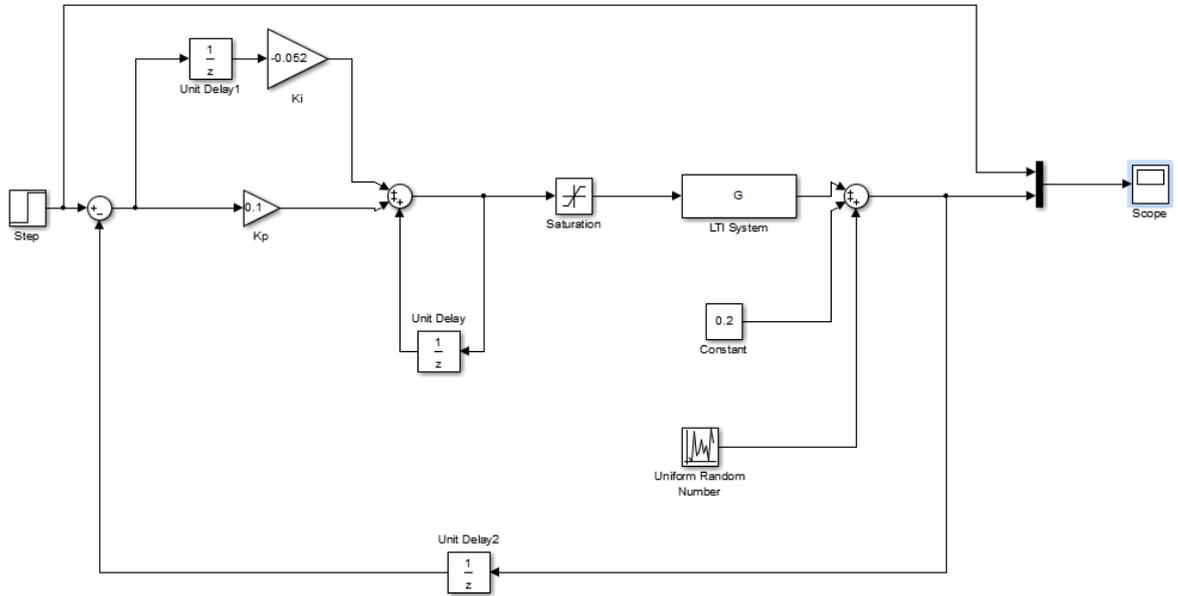
**Figura 3.29: Salida del sistema controlada a un *Setpoint* de 0.5**



**Fuente: Autores**

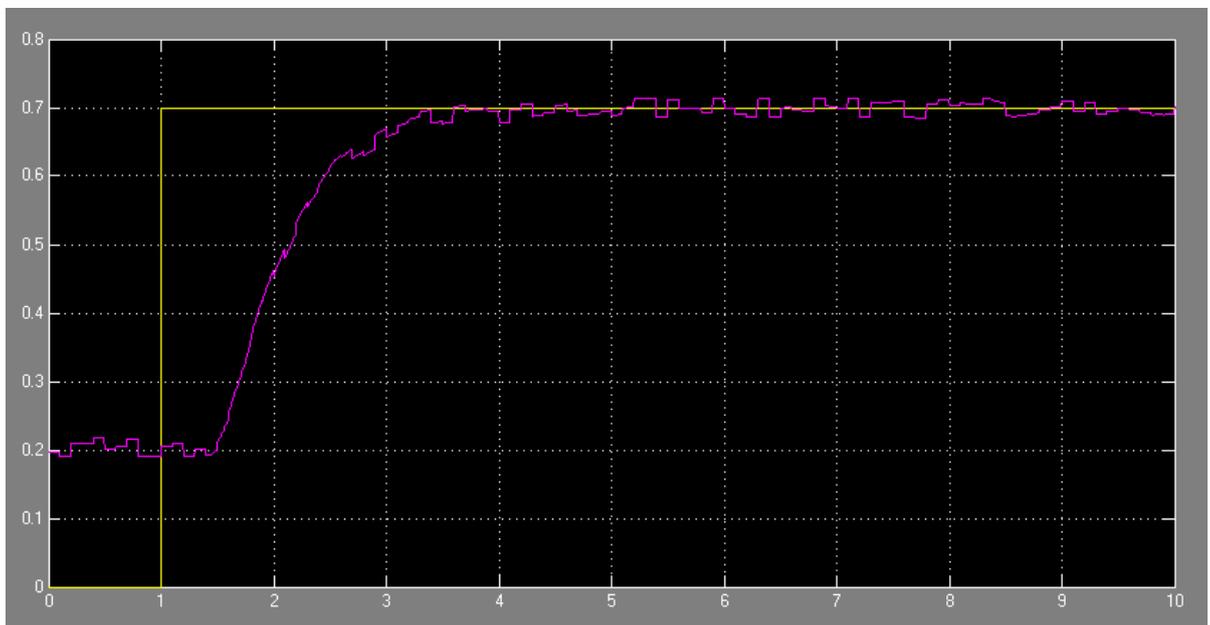
En la figuras 3.30 y 3.31 se visualiza el diagrama de bloques en Simulink y la gráfica de salida del sistema realizando una simulación agregando ruido aleatorio en un *setpoint* de 0.7-

Figura 3.30: Diagrama de Bloques en prueba de ruido aleatorio



Fuente: Autores

Figura 3.31: Salida del sistema con ruido aleatorio en *setpoint* de 0.7



Fuente: Autores

Para verificar el código del controlador digital consultar el Anexo C.

## 5.5 Diseño de Interfaces Gráficas

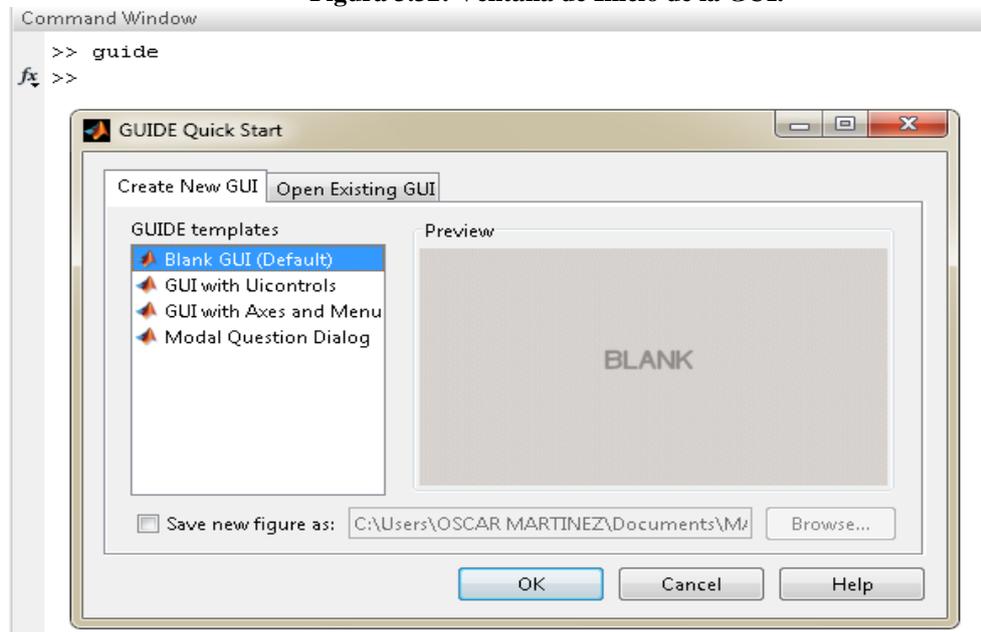
Una Interface gráfica de usuario (GUI) es una interface construida con objetos gráficos, los componentes del GUI dependen del diseño que se le quiera dar a la interface y puede tener botones, campos de texto, deslizadores y menús. El Matlab mantiene una caja de Herramientas para crear GUIs, estas herramientas simplifican el poner objetos y programar interfaces gráficas.

### 5.5.1 Introducción a la GUIDE

GUIDE es un entorno de programación visual disponible en MATLAB para realizar y ejecutar programas que necesiten ingreso continuo a datos. Tiene las características básicas de todos los programas de visualizaciones como visual Basic o Visual C++. Cuando se abre una GUI en GUIDE, se despliega en el Layout Editor que es el tablero de mando con todas la funciones y herramientas. Para Iniciar un proyecto GUI hay que escribir en la ventana de comandos `guide` al instante aparecerá el siguiente cuadro de dialogo que es la ventana de inicio de la GUI y se visualiza en la figura 3.32 que presenta las siguientes opciones:

- **Black GUIDE (default):** La opción de interfaz gráfica de usuario en blanco (viene predeterminada), nos presenta un formulario nuevo en el cual podemos diseñar nuestro programa.
- **GUI With Uicontrols:** Esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades.
- **GUI with Axes and Menu:** Esta opción es otro ejemplo del cual se obtiene el menu File con las opciones Open Print y Close. En el formulario tiene un *Popup menu, on push button* y un objeto *Axes*, se puede ejecutar el programa eligiendo alguna de las seis opciones que se encuentran en el menú desplegable y haciendo clic en el botón de comando.
- **Modal Question Dialog:** Esta opción muestra en la pantalla un cuadro de dialogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones *yes* y *No*, dependiendo del botón que se presione, el GUI retorna el texto seleccionado (la cadena de caracteres Yes o No).

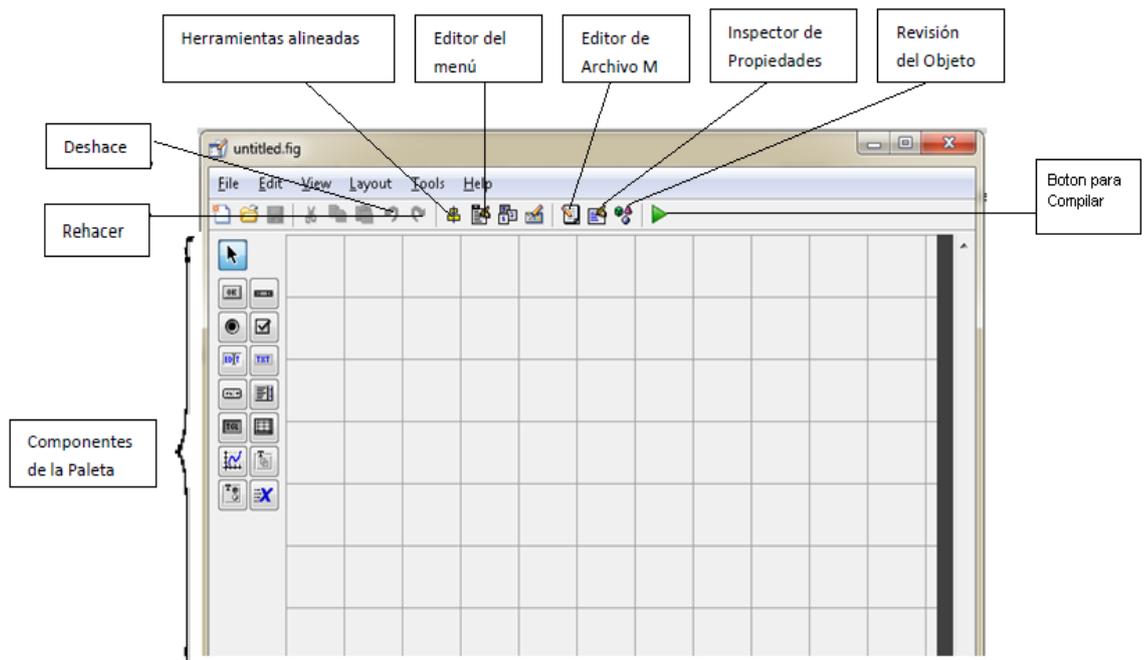
**Figura 3.32: Ventana de Inicio de la GUI.**



**Fuente: Autores**

Luego de escoger alguna de las opciones anteriormente descritas, nos aparecerá la ventana de entorno de la GUI, en la Figura 3.33 se visualiza la ventana con la descripción de sus Herramientas.

**Figura 3.33: Ventana de entorno y sus Herramientas.**

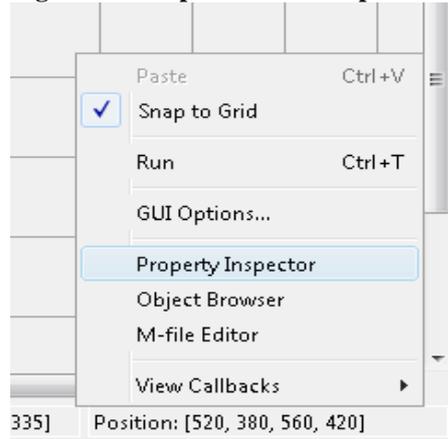


**Fuente: Autores**

## 5.5.2 Propiedades de los componentes

Cada uno de los elementos de GUI, tienen un conjunto de opciones que podemos acceder por medio del clic derecho, tal como lo muestra la figura 3.34.

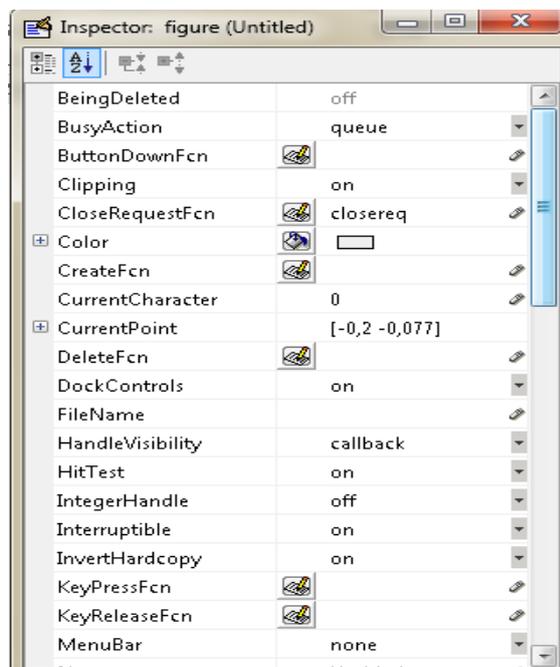
**Figura 3.34: Opciones del componentes**



**Fuente: Autores**

Luego de dar clic en *Property Inspector* abrirá una ventana en la cual podemos personalizar cada elemento, esto se visualiza en la figura 3.35.

**Figura 3.35: Entorno *Property Inspector* permite ver y editar propiedades de un objeto.**



**Fuente: Autores**

### 5.5.3 Funcionamiento de una aplicación GUI

Una aplicación GUIDE consta de dos archivos *.m* y *fig*. El archivo *.m* es el que contiene el código con las correspondencias de los botones de control de la interfaz y el archivo *fig* contiene los elementos gráficos. Cada vez que se adicione un nuevo elemento en la interfaz gráfica, se genera automáticamente código en el archivo *.m*. Para ejecutar una interfaz Gráfica, si la hemos etiquetado con el nombre *gui fig*, simplemente ejecutamos en la ventana de comandos **curso**. O haciendo clic derecho en el m-file y seleccionamos la opción *RUN*. El trabajo con GUIDE de Matlab, es una tarea similar a la labor que se realiza con cualquier lenguaje de programación gráfico como por ejemplo el Visual Basic. A continuación se muestra el resultado final de la creación de la GUI para la interfaz gráfica del proyecto. En la figura 3.35 podemos visualizar la parte de comunicación serial que al conectar el módulo XBEE o el cable serial será la primera parte de la GUI que aparezca en pantalla, en ella se deben configurar los siguientes datos:

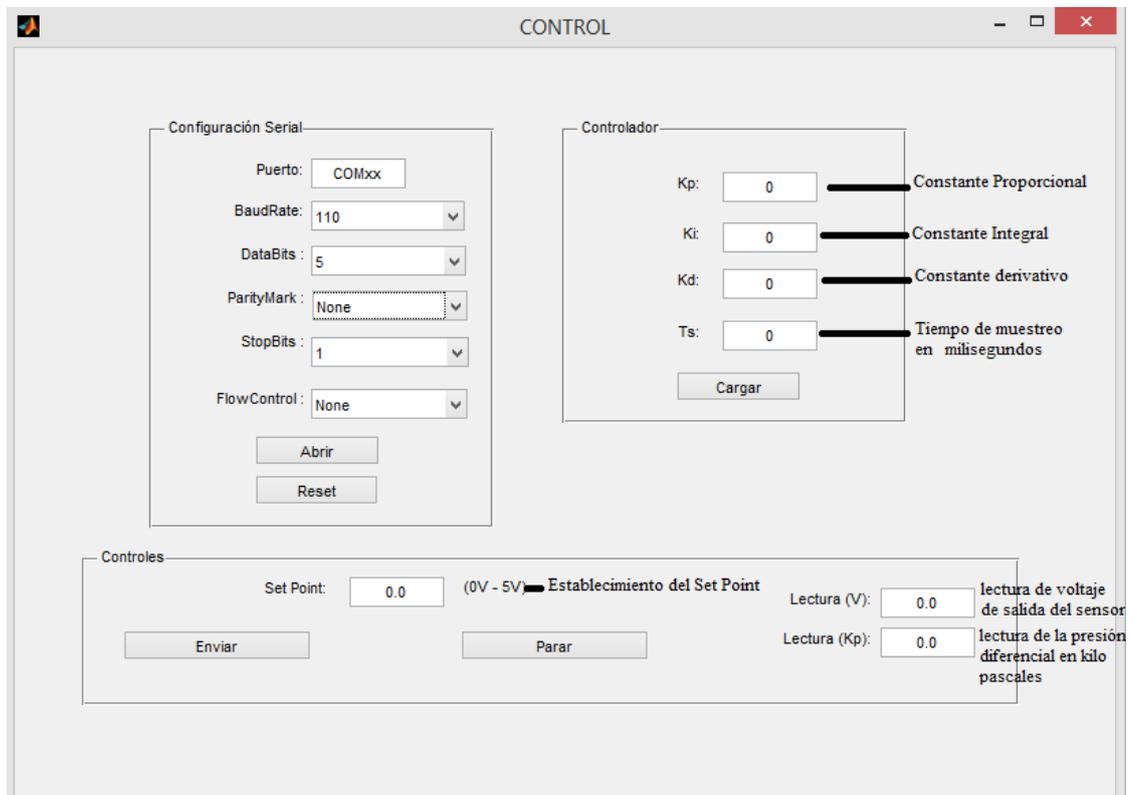
- Puerto: Configurar el puerto de entrada que fue reconocido por el PC.
- BaudRate: Velocidad de transmisión deseada, en el caso del proyecto es 9600.
- Data bits: El número de bits de transmisión que para el caso son 8.
- Party Mark: La paridad, que en este caso no se utiliza y se deja en None.
- Stop Bits: El bit de parada que en este caso es 1.
- Flow Control: El flujo de control que se dejara en None.

Luego de ingresar los anteriores datos se debe marcar el botón de abrir para que como siguiente paso muestre la parte del controlador en donde sencillamente se deben ingresar las constantes Proporcional, Integral y Derivativa del controlado PID que fue hallado y por último el tiempo de muestreo que se desea que en el caso de este proyecto es de 2500 micro segundos. La parte del controlador es la más importante del proyecto porque se aplica a un controlador reprogramable, lo cual indica que se puede cambia su configuración desentendiendo del sistema de control al que se aplique el sistema modular de hardware propuesto. Para el sistema de presión aplicado en este proyecto según el diseño del controlador propuesto por Matlab las contantes son las siguientes:

- KP: 1
- KI: -0.052
- KD: 0

Al ingresar la variables del controlador se pulsara el botón de cargar, entonces así se abrirá la parte de controles, que será donde se pueda establecer el Set point que se desee y al mismo tiempo se pueda parar el proceso mediante el botón parar. Así de esta forma se implementó esta GUI para la interfaz gráfica mediante Matlab, que utiliza el script del código hecho para la adquisición de datos, el código de control y la transmisión de información en un entorno sencillo y fácil de utilizar.

Figura 3.36: Interfaz gráfica para el sistema de control



Fuente: Autores

## 5.6 Diseño electrónico del hardware

El objetivo de llevar a cabo la identificación de la planta fue específicamente el diseñar un controlador para la misma, el controlador será un controlador PID por software el cual adoptara la mejor combinación que cumpla los requerimientos establecidos.

Para el diseño del circuito para la planta de control y su respectiva adquisición de datos, se debe separar inicialmente el trabajo que realiza el microcontrolador de la parte de transmisión serial y ZigBee hacia el ordenador, adicional a esto se suma la etapa de potencia y el circuito de trabajo del sensor.

### 5.6.1 Especificaciones Generales

A la hora de diseñar el circuito se ha tenido en cuenta ciertos parámetros para el desarrollo del hardware los cuales se nombran a continuación:

- Para controlar todo el sistema de control y el procesamiento de la información se usó el PIC18F452.
- Para la parte de comunicación serial se usó el MAX232.
- Para la comunicación Zigbee se usó el Módulo Xbee Explorer y para la transmisión una antena 100 mts Xbee Transceiver Series 2.
- Para la recepción de la señal Zigbee se utilizó otra antena Xbee Transceiver Series 2 en el mismo módulo Xbee Explorer pero conectado por cable USB al ordenador

- En la etapa de potencia se usó un TIP31C
- El Sensor MPX5100 se utilizó para el monitoreo de la presión mediante la conexión del tubo Venturi.

Después de identificar los componentes principales que llevara el hardware se dispuso a realizar el diseño del circuito para el sistema de control.

### 5.6.2 Diseño del Esquemático y el PCB mediante Kicad

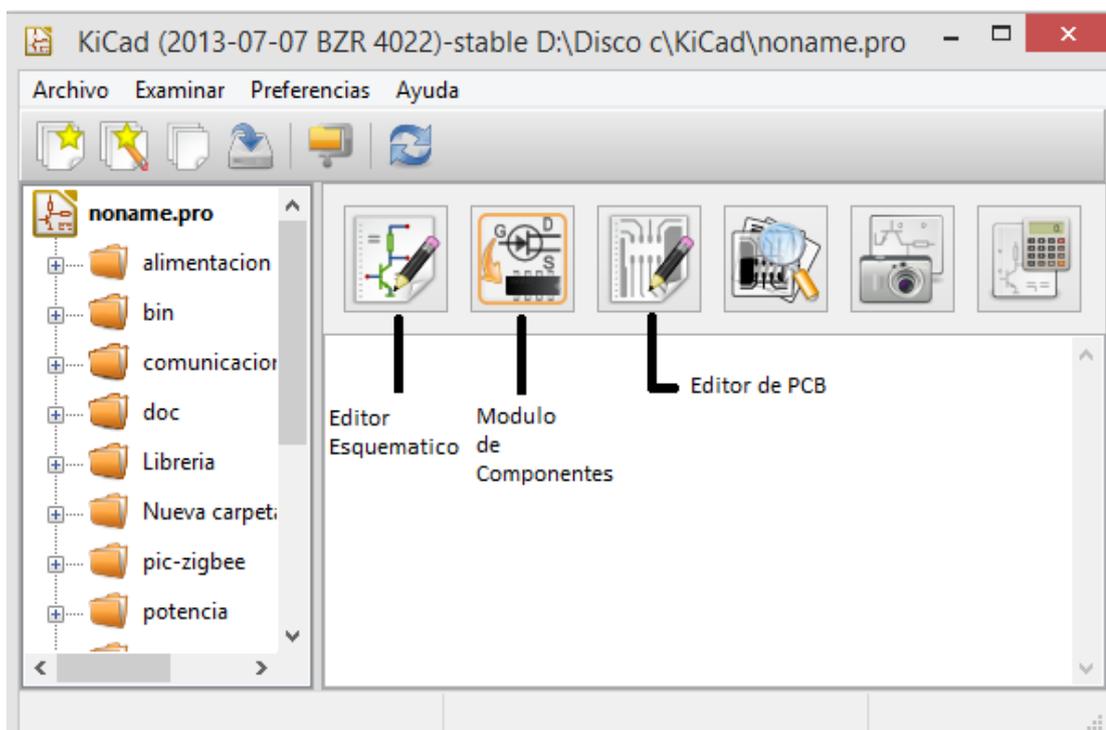
Antes de realizar el diseño del circuito se realizó la división del mismo por etapas para así crear el sistema modular para el control de la planta de presión. El primer módulo es el del controlador y su etapa de comunicación, luego siguen los módulos de la etapa de potencia, la del acople de la señal del sensor y por ultimo una de alimentación. Para el diseño de los esquemáticos y sus respectivos PCB's se usó el programa de licencia libre KiCad, primero porque es un software libre y segundo porque da la facilidad de crear librerías propias y así editar Las ya existentes.

El entorno de KiCad que se usó para el diseño se divide en 3 partes:

- Editor Esquemático
- Módulo de Componentes
- Editor PCB

Las opciones anteriormente enunciadas se pueden visualizar en la ventana de inicio de KiCad (Figura 3.37). Las demás opciones son para creación de mapa de bits y calculadora de PCB.

**Figura 3.37: Ventana de Inicio de Kicad**



**Fuente: Autores**

Para diseñar los circuitos esquemáticos se utilizó el editor de esquemático de Kicad, que es muy sencillo de usar y viene con las librerías de los elementos que fueron necesarios para crear los módulos de control, potencia, alimentación y acople de sensor, en el anexo B se puede consultar el esquemático del módulo de control donde los componentes que se encuentran son los siguientes:

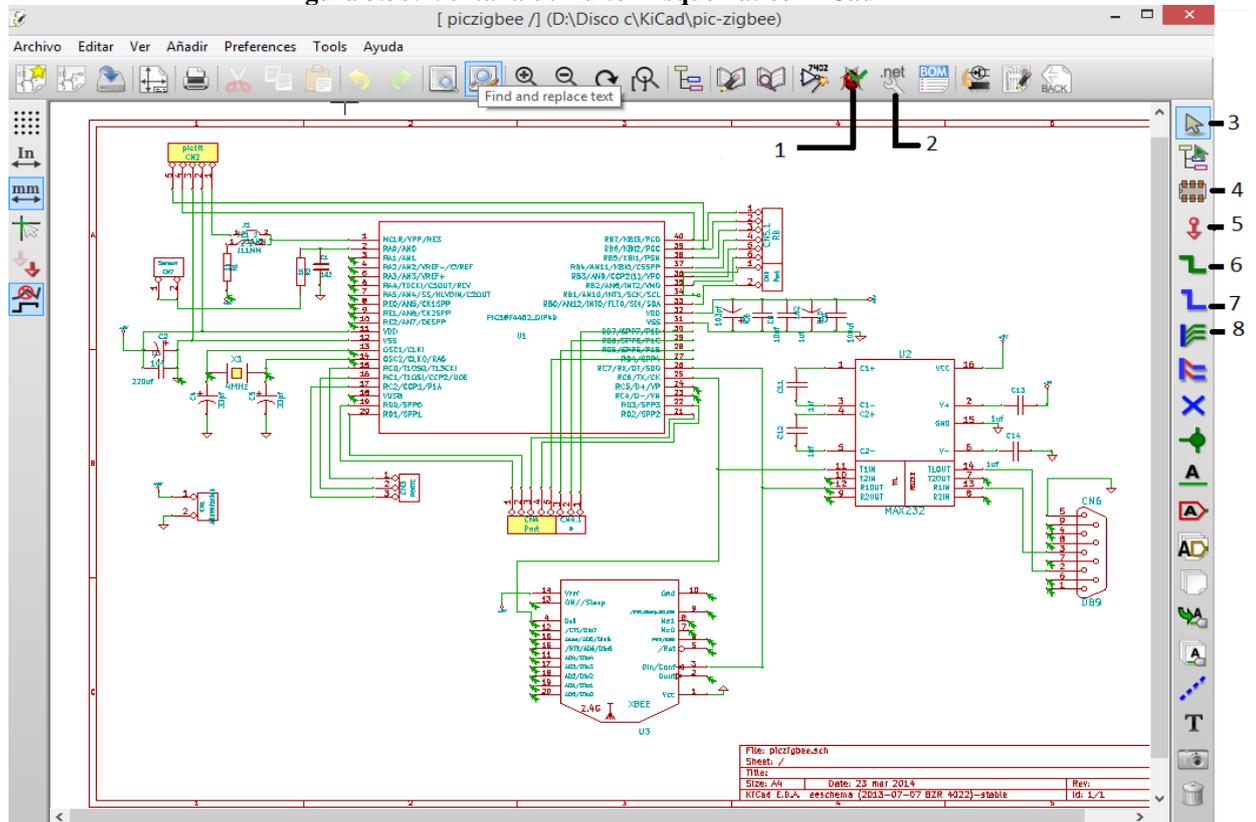
- Microcontrolador PIC18F452
- MAX232
- 6 Resistencias
- 14 condensadores
- 8 conectores
- 1 conector DB9 Hembra
- 1 Cristal de 10Mhz
- 2 Jumper
- 1 Modulo ZigBee
- 1 Amplificador Operacional

En anexo B también se pueden consultar los esquemáticos del módulo del sensor y el módulo de potencia, en donde se usaron los siguientes elementos:

- 6 conectores
- 3 condensadores
- 3 resistencias
- 1 TIP31C
- 1 Opto acoplador

La ventana de edición esquemática se aprecia en la Figura 3.38, en donde se visualizan el entorno de trabajo que KiCad ofrece al usuario y los botones de Herramientas que en este caso son enumerados para explicar su función.

**Figura 3.38: Ventana de Editor Esquemático KiCad**

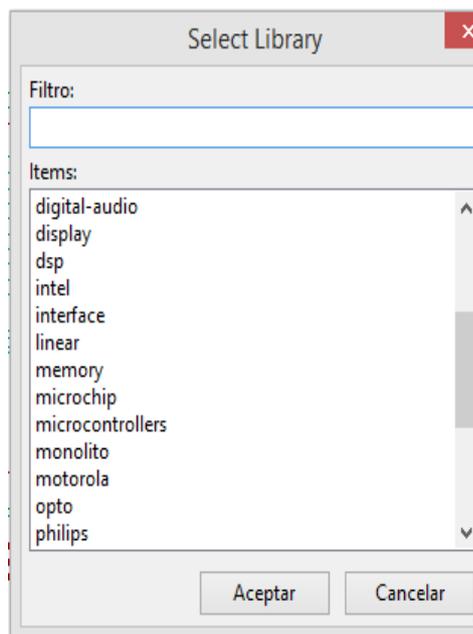


Fuente: Autores

A continuación se explica la función de los botones enumerados en la figura 3.37:

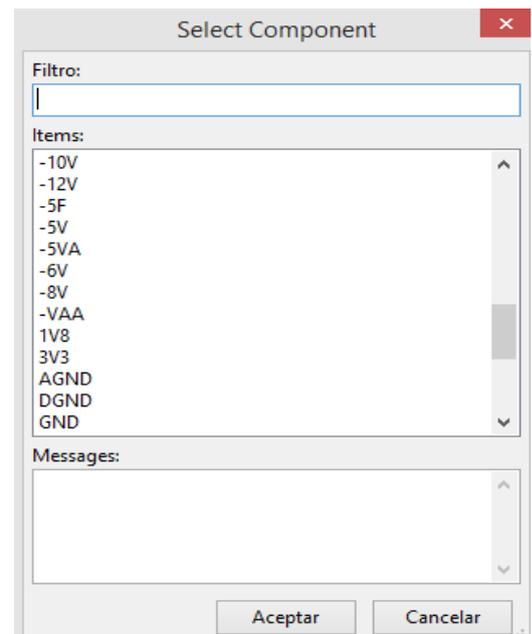
1. **Perform Electric Rules Check:** Su Función es checar las conexiones eléctricas luego de finalizar la creación del circuito esquemático y así verificar que no queden componentes sin conexión o componentes mal referenciados, es decir que no queden enumerados con letras o números.
2. **Generate netlist:** Su función es generar el netlist o lista de componentes usados en el esquemático. Al crear el archivo de extensión .net funciona para poder utilizar el Modulo de Componentes que más adelante será explicada su función. Cabe resaltar que para poder crear el netlist primero hay que checar las conexiones o sino no generara la lista.
3. **Herramienta de selección:** Su función es la de selección de componentes para poder moverlos a través de la hoja del esquemático o en su defecto editar funciones del componente.
4. **Colocar un componente:** Al utilizar esta función permitirá escoger de las librerías de Kicad los componentes que se desean utilizar para el esquemático, donde se dividirán por lista de componentes ya sean pasivos, TTL, operacionales, Micros Freescale,P PIC's etc. En la figura 3.39 de muestra las librerías de componentes.
5. **Poner un puerto de alimentación:** Funciona para seleccionar el tipo de puerto que se desea usar, al igual que la herramienta anterior esta también tiene lista de puerto de poder, que se dividen por voltajes positivos, negativos y puertos de tierra. En la Figura 3.40 se muestra parte del listado de esta herramienta.

Figura 3.39: Selección de librerías de Componentes en kicad



Fuente: autores

Figura 3.40: Selección de librerías de puertos de poder en Kicad

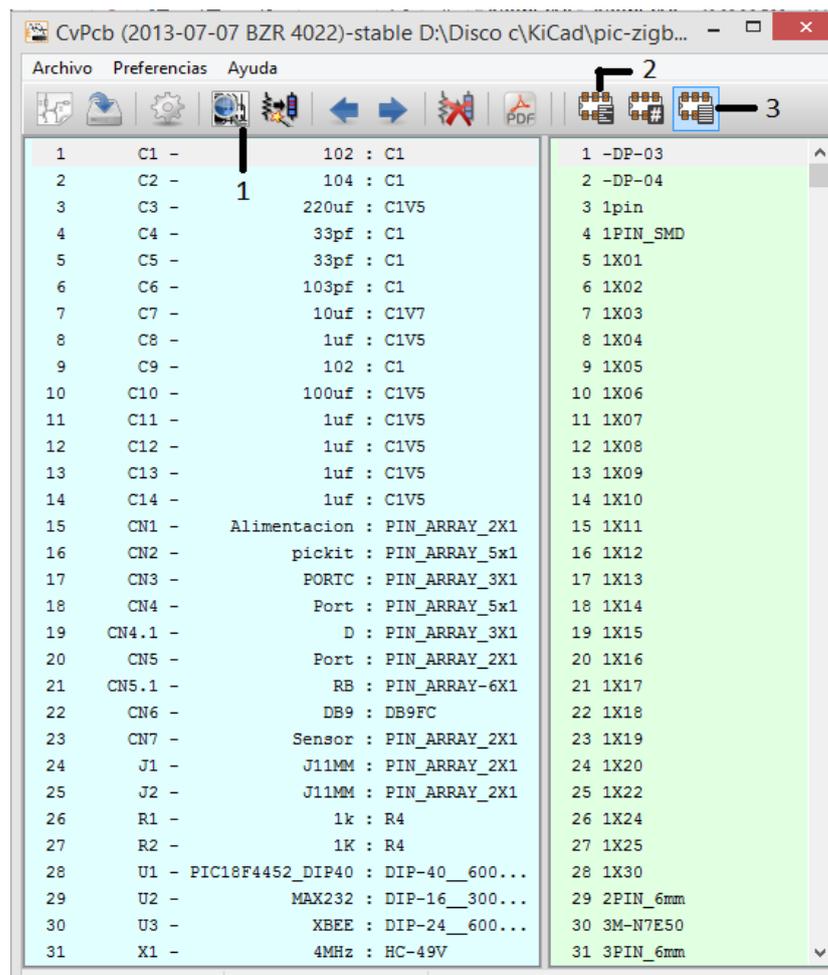


Fuente: Autores

6. **Poner una línea:** Con esta función podemos unir mediante caminos o alambres las entradas de componentes como el PIC a puertos de entrada/salida o componentes pasivos.
7. **Poner Bus:** Funciona para unir elementos mediante un bus, que se diferencia del camino de cobre por que al momento de la impresión del PCB aparecerán solo los orificios para incluir el bus, mas no un camino de cobre.
8. **Añadir Bus de datos:** Esta función como la anterior inserta un bus de datos que diferencia del anterior se puede escoger el número del buses que se necesiten para el circuito.

Luego de la creación del esquemático usamos la herramienta del módulo de componentes (Figura 3.41), se usa para implementar el grosor y tamaño del componente dependiendo de la lista creada para el mismo, es decir que podemos escoger el tamaño o grosor de las resistencias, condensadores o en su defecto escoger si se quiere usar un socket para el PIC o en su defecto un elemento superficial

**Figura 3.41: Ventana de Módulos de componentes**



**Fuente: Autores**

La herramienta de módulo de componentes en si es el seleccionador de huellas (footprint's) de los componentes usados en el esquemático, esencial para el trabajo en el editor de PCB,s de Kicad.

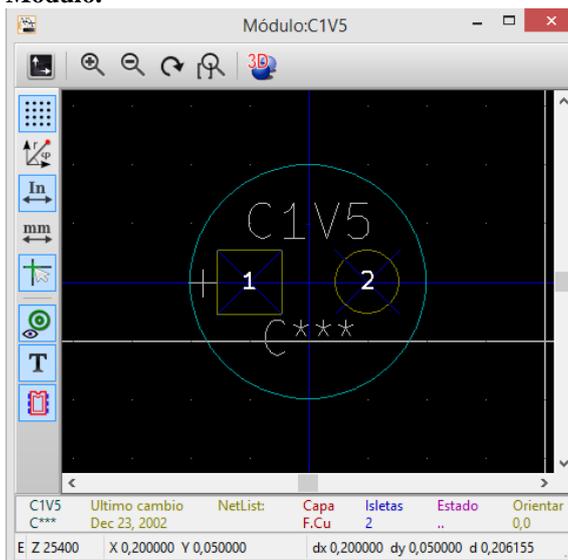
A continuación se explica la enumeración de los tres botones que se utilizaron en la asignación de módulos a los componentes de la netlist, que como se indicó anteriormente es necesaria para poder asignar las dimensiones que los componentes tendrán en la tarjeta.

1. **Muestra el módulo seleccionado:** Es la herramienta para poder verificar el tamaño del componente y la visualización en 3D del componente seleccionado.
2. **Muestra la lista de componentes actuales:** Visualiza la lista de módulos que ya han sido asignados a los componentes del esquemático.
3. **Muestra la lista completa de los módulos a asignar:** Visualiza la lista de todos los módulos que se pueden asignar a los componentes, cabe resaltar que está organizada alfabéticamente por referencia del módulo, es decir que los conectores condensadores resistencias, pines, y sockets van en orden alfabético.

A cada componente se le puede asignar el módulo deseado, claro está que se debe verificar el tamaño, tanto de ancho como de alto, ya que estos son los que se visualizaran en el diseño del PCB usando el editor de PCB's de Kicad.

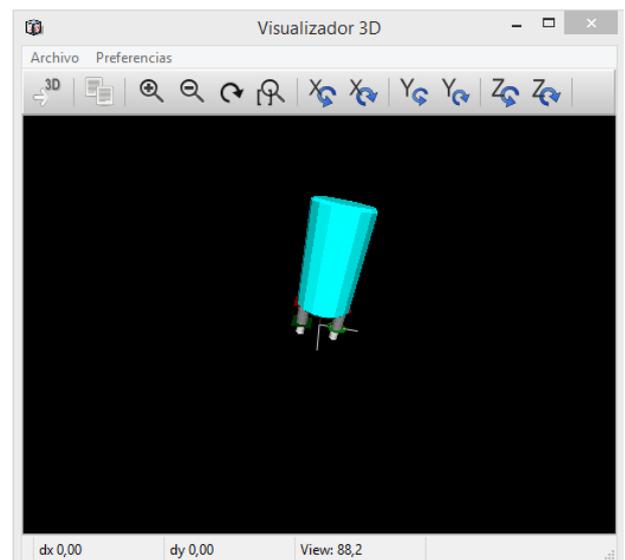
Las Figuras 3.42 y 3.43 hacen referencia del visor de módulos y su visualización en 3D

**Figura 3.42: Visualización de dimensiones del Módulo.**



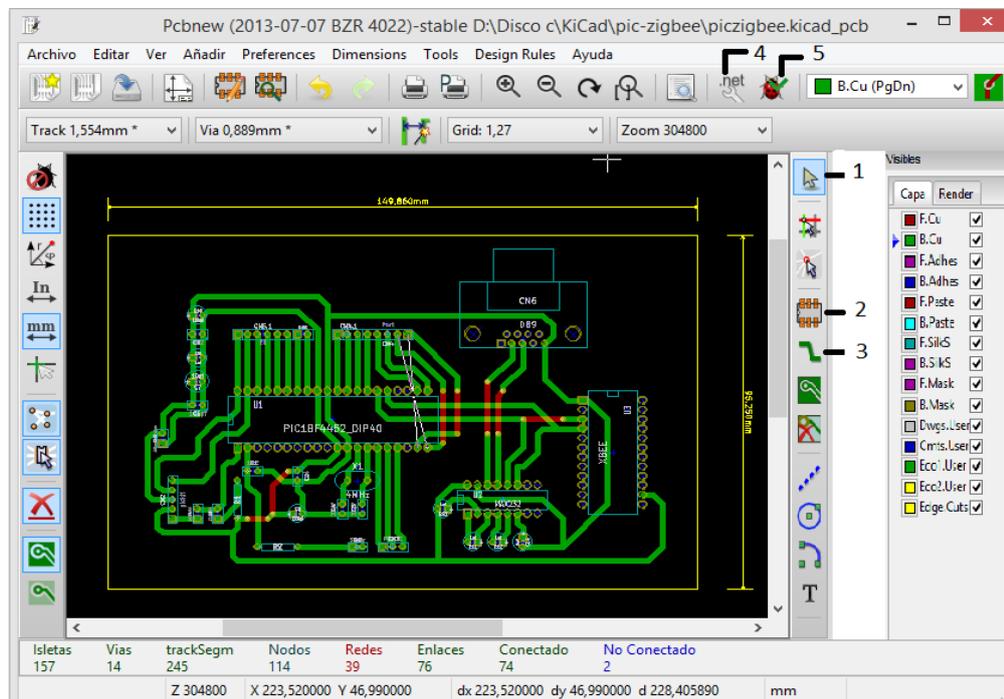
**Fuente: Autores**

**Figura 3.43: Visualización del módulo en 3D**



**Fuente: Autores**

Figura 3.44: Entorno del Editor de PCB de Kicad



Fuente:Autores

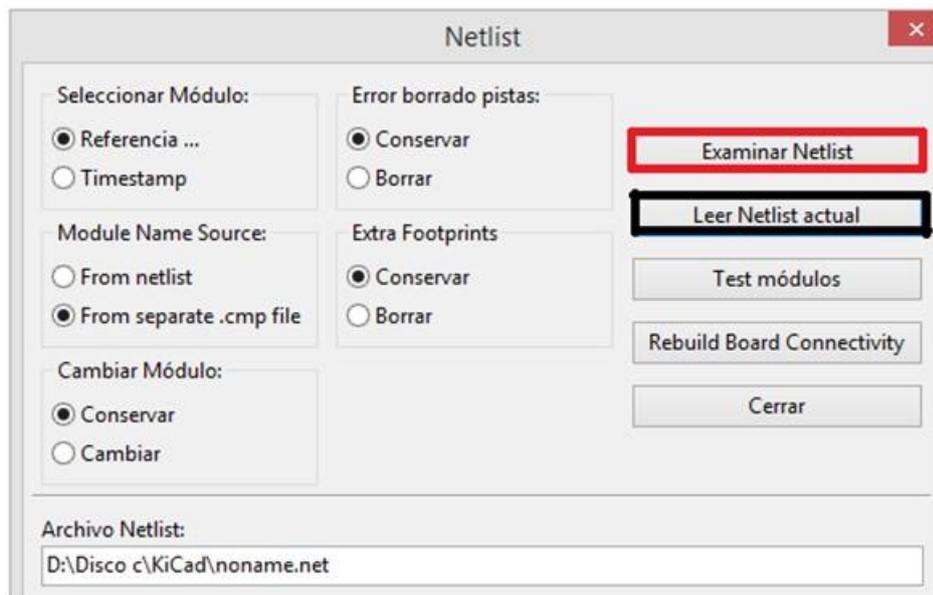
Luego de asignar los footprints a la lista de componentes que se implementaron en el esquemático, podemos empezar a crear nuestro PCB mediante el editor PCB de kicad, en la figura 3.44 muestra el entorno del editor de PCB que al igual que en los anteriores módulos se enumera las herramientas a utilizar y se explicaran a continuación:

1. **Herramienta de Selección:** Funciona para mover los elementos o seleccionar el área total de la tarjeta.
2. **Añadir Módulos:** En casos que se desee añadir un módulo adicional para correcciones o como mejora del circuito ya establecido esta herramienta permitirá añadir un módulo de un componente a un PCB ya creado.
3. **Añadir pista o vía:** Con esta herramienta se puede empezar a diseñar la forma de los caminos de cobre de la tarjeta, además de crear una pista de cobre también da la opción de crear una vía que es útil en caso de que por temas de obstrucción no se puedan conectar dos puntos, en ese caso se genera una vía que sería conectar con un alambre o cable que iría encima del trazado del PCB.
4. **Leer Netlist:** Esta herramienta es fundamental, ya que con ella se lee el archivo .net que se generó en el módulo de componentes y al momento de leer la lista genera los footprints para empezar a organizarlos en la hoja de edición.

5. **Perform Electric Rules Check:** Como en el editor esquemático su función es revisar que todos los camino de cobre y vías estén adecuado en base al esquemático, es decir que ningún camino puede quedar sin conectar, ya que al momento de empezar a dibujar las vías está siempre se conectara tal como se conectó en el esquemático, es decir que no se puede conectar una pista a un lugar que no sea el mismo que el del esquemático.

Luego de los pasos anteriores podemos empezar la creación de nuestro PCB, que según lo anterior debemos lo primero que debemos hacer al abrir el editor PDB de KiCad es abrir el netlist es decir compilar la lista que se creó en el módulo de componentes que básicamente era adjuntar los footprints a cada componente. Al leer la lista de netlist nos muestra la siguiente figura 3.45.

Figura 3.45: Entorno Netlist para asignar componentes

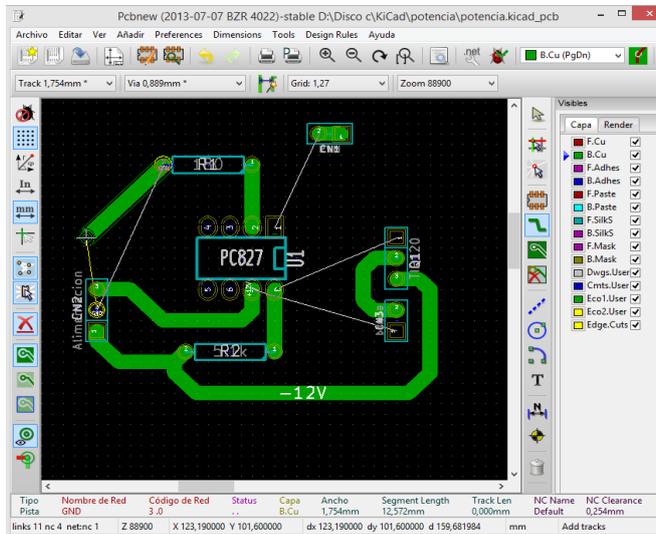


1. Examinar netlist, en ese momento se abriera la carpeta que contiene el archivo .net que es el que tiene la información de los footprints por componente, se selecciona el archivo y luego se continua con el paso 2.
2. Se pincha en el boton de Leer Netlist actual y de inmediato se colocaran los componentes con footprints asignados al lado de la hoja de edición para ser organizados

Fuente: Autores

Luego de leer la Netlist los componentes aparecen en la parte superior izquierda de la hoja de edición para poder ir acoplando los componentes en el orden que se requiera, luego de ya organizar los elementos como lo muestra la figura 3.46, el siguiente paso es conectar los componentes mediante la herramienta de añadir pistas.

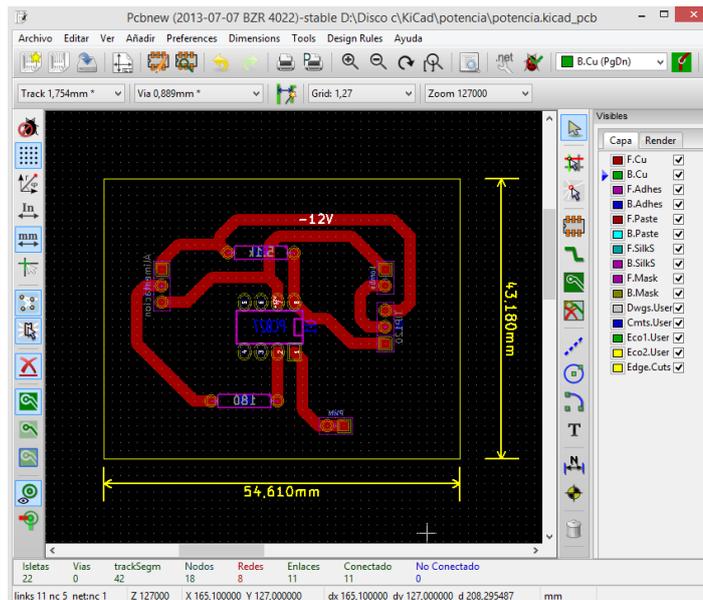
**Figura 3.46: Construcción de tarjeta PCB**



Fuente: Autores

Cabe resaltar que no hay forma de cometer errores de conexión ya que el programa no permite conexiones que no estén predeterminadas por el archivo esquemático. Mediante una línea amarilla el programa indica el destino de conexión. Luego de terminar la conexión se debe invertir la posición para que al momento de impresión se adecue la tarjeta para soldar los componentes, esto se hace seleccionando con el clic derecho el área donde está el diseño y seleccionar la opción Flip Block y de esta forma el PCB se invertirá y así estará listo para la impresión del PCB. En la Figura 3.47 se muestra la inversión del bloque PCB.

**Figura 3.47: Bloque PCB invertido**



Fuente: Autores

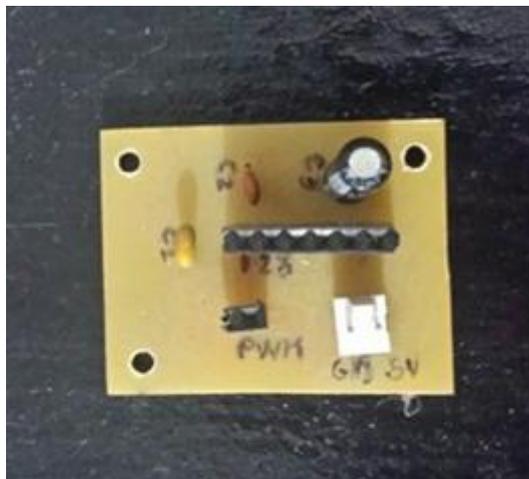
Para poder ver los esquemáticos y los PCB's de los tres módulos que se fabricaron para este proyecto consulte el Anexo C en donde se encuentran los esquemáticos, y PCB's de los módulos de control, sensor y potencia. El resultado final de las tarjetas se muestra a continuación en las figuras 3.48 a 3.50.

**Figura 3.48: Módulo de Control**



Fuente: Autores

**Figura 3.49: Módulo de acople de señal del sensor**



Fuente: Autores

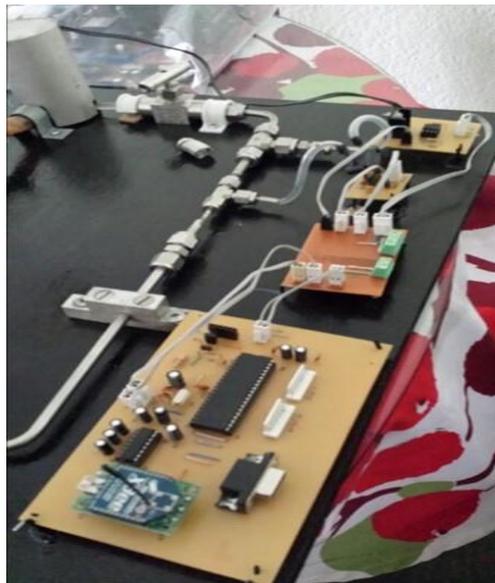
**Figura 3.50: Módulo de Potencia**



**Fuente: Autores**

Las anteriores figuras muestran los módulos que componen el sistema modular de hardware, hay que resaltar que el módulo de alimentación no se diseñó mediante Kicad y simplemente se usó una tarjeta universal para acoplar los puertos de alimentación mediante borneras e interconectar los tres módulos anteriores de una forma más ordenada. Como lo muestra la figura 3.51.

**Figura 3.51: Sistema modular conectado mediante módulo de alimentación**

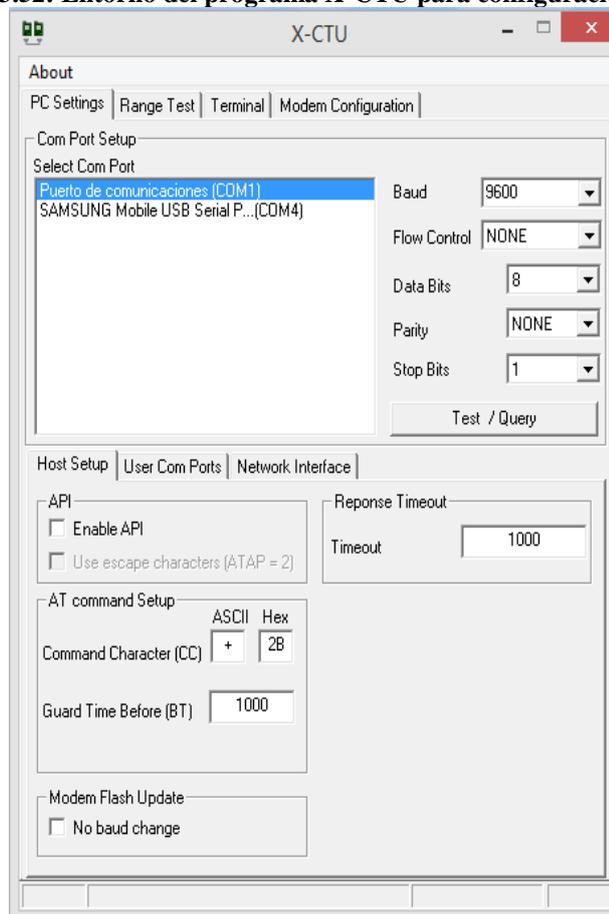


**Fuente: Autores**

## 5.7 Configuración módulo Xbee

Para poder usar los módulos XBee en el proceso de la transmisión por ZigBee es necesario instalar en programa X-CTU, aunque existen más programas en el mercado para la respectiva configuración del XBee, en este proyecto se ha escogido este porque es el más sencillo de usar. Este programa puede ser descargado de la página web de Digi International. Luego de instalado el programa de configuración XBee se visualizará una ventana como la mostrada en la figura 3.52.

**Figura 3.52: Entorno del programa X-CTU para configuración Zigbee**



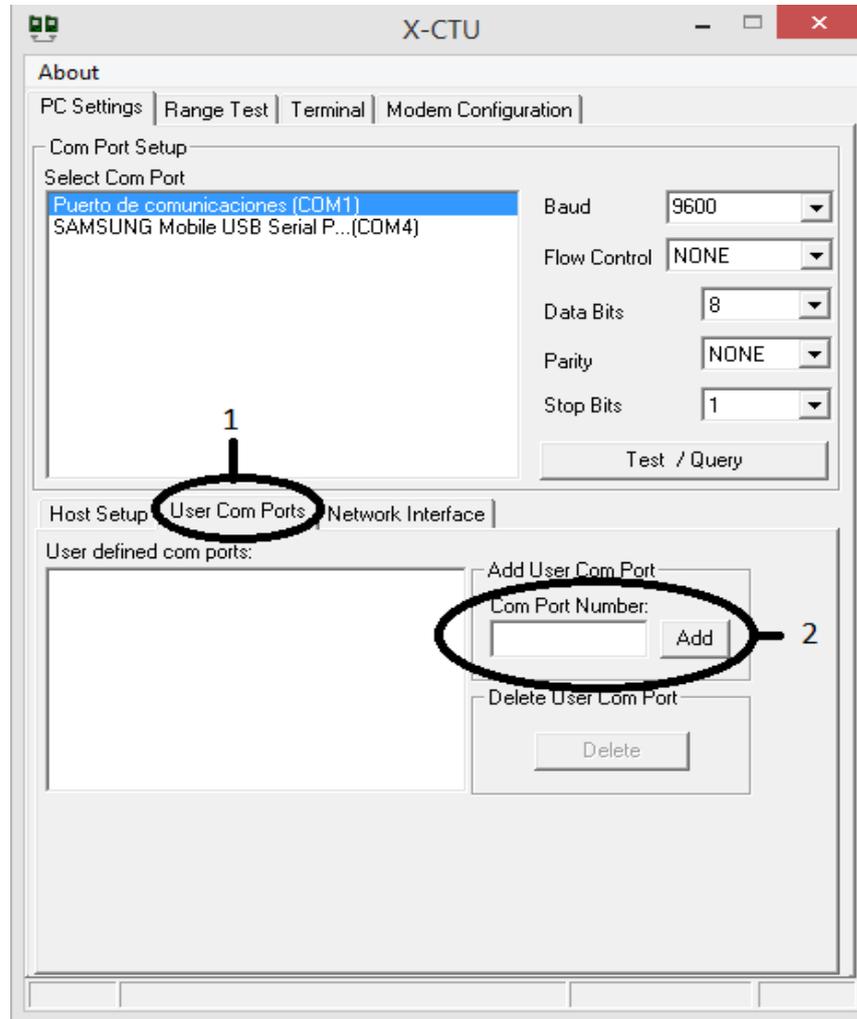
**Fuente: Autores**

Para poder configurar el módulo XBee se necesita de una interfaz USB para que pueda conectarse al ordenador y de esta manera llevar a cabo la configuración del Módulo, es por eso que se usó el módulo XBee Explorer que viene con el puerto ya adaptado.

Luego de tener conectado el módulo XBee Explorer, se procede a empezar la configuración del puerto en el programa X-CTU, al conectar módulo al puerto USB del ordenador este reconocerá el dispositivo indicando que ha detectado el puerto, como se muestra en la figura, primero debemos ir a la pestaña que dice “User Com Ports” y en la

parte “Com Port Number” ingresamos el nombre del puerto en donde hemos conectado el módulo sea Com1 o Com3 y pulsamos el botón “Add”.

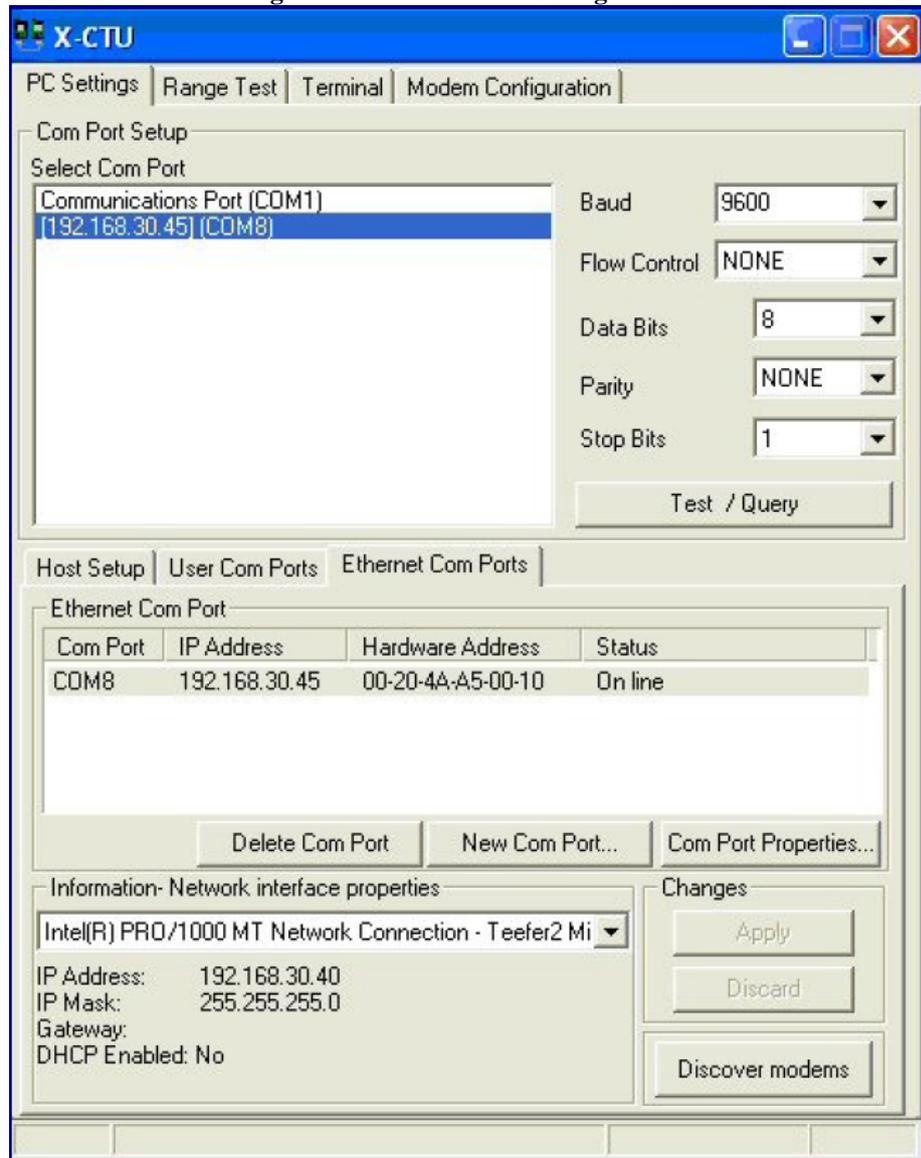
**Figura 3.53: Ventana de configuración de puertos**



**Fuente: Autores**

Después de añadir el nombre del puerto mediante el botón Add aparecerá la imagen de la figura 3.54, después de añadir el nuevo dispositivo se debe señalar la ventana “Select Com Port” dicho dispositivo y seleccionar el botón “Test/Query”. Luego de lo anterior aparecerá en pantalla la Figura 3.55 que es la ventana de Testeo de puerto.

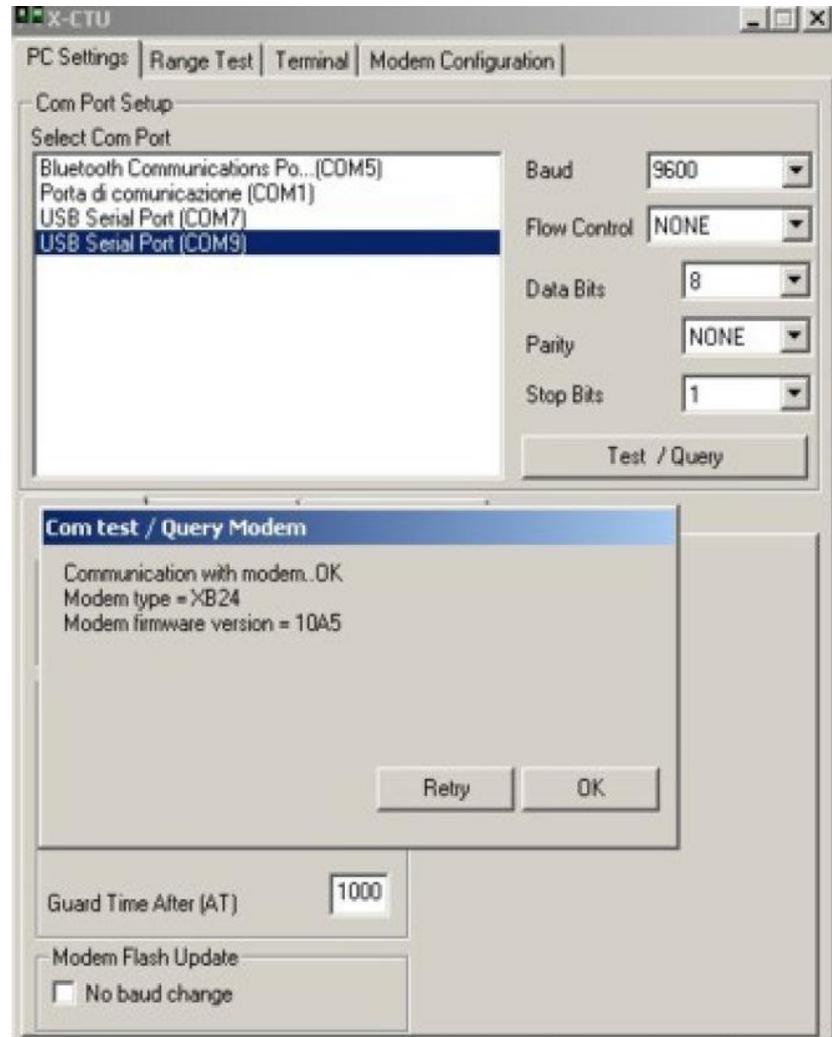
Figura 3.54: Ventana de Configuración



Fuente: Autores

Si nos aparece en la ventana de confirmación “Unable to open port” se le debe cambiar el valor a la casilla “Baud” aunque el valor configurado de fábrica es “9600” que es el mismo que se usa en este caso pero puede cambiarse dependiendo de lo que quiera el usuario. Después de efectuar los anteriores pasos ya se tiene añadido el usb para poder configurar el módulo Xbee.

Figura 3.55: Ventana de Testeo de puerto



Fuente: Autores

### 5.7.1 Configuración del Nodo de Recepción

A continuación se indicaran los pasos de cómo configurar el nodo encargado de la recepción de información del nodo sensor y enviar dicha información al servidor. Existen dos formas de realizar la configuración de nuestros nodos:

1. Terminal
2. Modem Configuration

Si utilizamos la configuración a través de “Terminal” debemos escribir +++ antes de escribir cualquier comando, tal y como se muestra en la Figura 3.56. Los comandos que se utilizaron para la configuración de nuestro nodo son:

**ATRE:** Restaura los valores predeterminados de fábrica antes de realizar cualquier modificación.

**ATAP\*:** Configuración de la API de XBee. Colocar el número que tiene la API de XBee, en nuestro caso 2, por lo que el comando sería ATAP2.

**ATCE1:** Configuración del módulo XBee en modo Coordinador.

**ATMY\*:** Dirección del módulo XBee en modo Coordinador. El valor de \* en nuestro caso será 1234. (ATMY1234)

**ATID\*:** ID de la conexión que vamos a crear entre nuestros módulos XBee. El valor de \* en nuestro caso es 1111. (ATID1111)

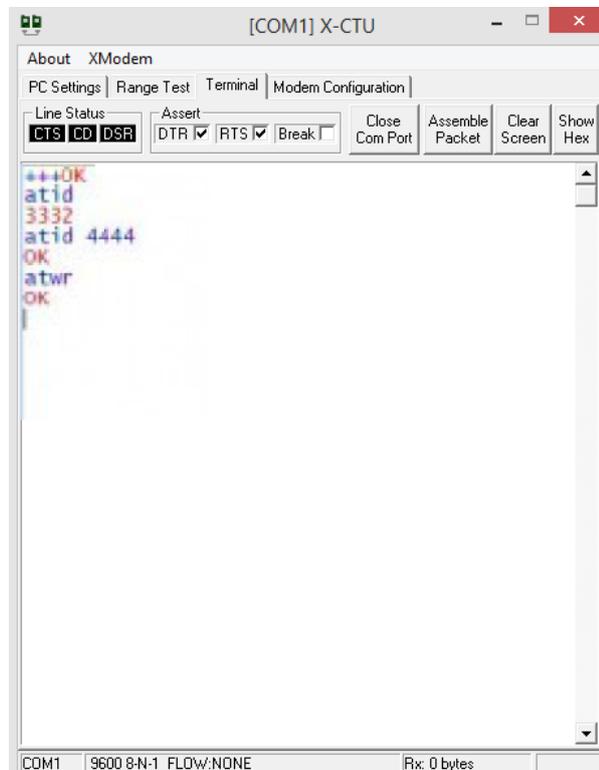
**ATCH\*:** Canal por el cual los módulos XBee se van a conectar. El valor de \* en nuestro caso será 0C. (ATCH0C)

**ATWR:** Escribe una nueva configuración en la memoria no volátil. Si no se escribiese este comando, las modificaciones realizadas solo durarían hasta que el módulo se quede sin batería.

**ATFR:** Reinicia el módulo XBee.

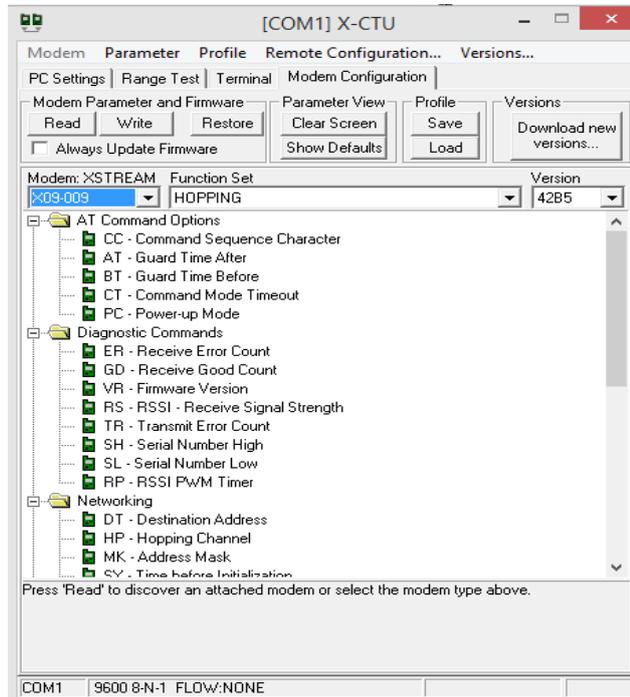
En caso de que se requiera realizar la configuración a través de “Modem Configuration” se deben buscar los nombres, por ejemplo si se quiere configurar el canal por el cual los módulos Xbee se van a conectar, tendríamos que buscar CH –Channel y poner el canal 1111 descrito anteriormente. Para escribir en la memoria no volátil se tendría que pulsar el botón “Write”. En la figura 3.57 se puede verificar la venta de “Modem Configuration”

**Figura 3.56: Configuración por terminal**



**Fuente: Autores**

**Figura 3.57: Configuración a través de Modem Configuration**



**Fuente: Autores**

### 5.7.2 Configuración del Nodo Sensor

Después de configurar el nodo pasarela, vamos a configurar los nodos sensores que son los encargados de enviar la información al nodo que hemos configurado anteriormente. Al igual que en el apartado anterior, si utilizamos la configuración a través de “Terminal” debemos escribir +++ antes de escribir cualquier comando. Los comandos que vamos a utilizar para la configuración de nuestros nodos son:

**ATRE:** Restaura los valores predeterminados de fábrica antes de realizar cualquier modificación.

**ATAP\*:** Configuración de la API de XBee. Colocar el número que tiene la API de XBee, en nuestro caso 2, por lo que el comando sería ATAP2.

**ATCE0:** Configuración del módulo XBee en modo Nodo Transmisión.

**ATMY\*:** Dirección del módulo XBee en modo Nodo Transmisión. El valor de \* en nuestro caso será 5678. (ATMY5678)

**ATID\*:** ID de la conexión que vamos a crear entre nuestros módulos XBee. El valor de \* en nuestro caso es 1111. (ATID1111)

**ATCH\*:** Canal por el cual los módulos XBee se van a conectar. El valor de \* en nuestro caso será 0C. (ATCH0C)

**ATWR:** Escribe una nueva configuración en la memoria no volátil. Si no se escribiese este comando, las modificaciones realizadas solo durarían hasta que el módulo se quede sin batería.

**ATFR:** Reinicia el módulo XBee.

# CAPÍTULO 6

## 6. Experimentación y Resultados

### 6.1 Acondicionamiento de Señales

Una vez que los datos han sido recolectados del proceso físico, es probable que los mismos no estén en condiciones de ser usados por algoritmos de identificación. Esto se debe a que hay varias deficiencias posibles en los datos que deberían ser tenidas en cuenta, tales como:

1. Perturbaciones de alta frecuencia en la adquisición de datos, por encima de las frecuencias de interés del sistema dinámico.
2. Bursts y/o outliers ocasionales
3. Drift, offset y perturbaciones de baja frecuencia.

Aclaraciones:

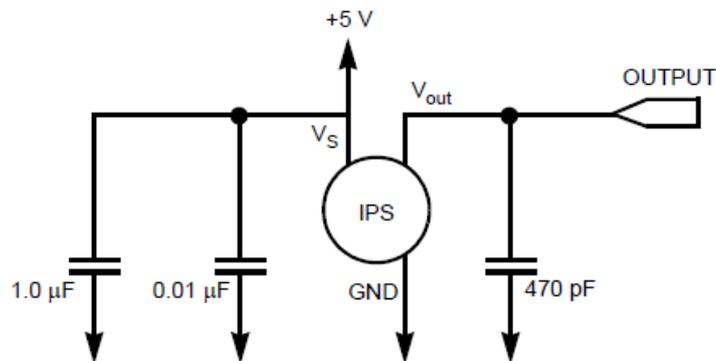
- Bursts: Es una secuencia o ráfaga de datos de valor unitario
- Outliers: Son observaciones infrecuentes o puntos que no parecen seguir la distribución característica del resto de los datos
- Drift: Variación lenta de una característica del sistema como la amplitud, frecuencia o potencia de salida.

#### 6.1.1 Acondicionamiento de señales del sensor al conversor A/D

El sensor de presión MPX5100DP envía una señal analógica que requiere ser convertida a Digital para que pueda ser procesada por el PIC18F452. Debido a que la salida del sensor guarda una porción de voltaje de la fuente de alimentación la señal de salida se verá comprometida.

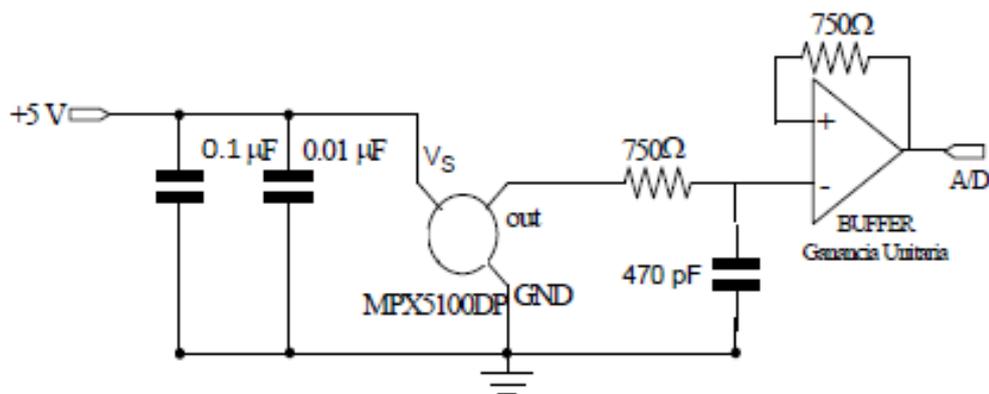
En principio la señal de salida del sensor de presión era levemente afectada por una distorsión de voltaje y la exactitud de los datos a medir. Así que en el circuito principal de filtrado de la señal de salida del sensor (Figura 4.1) que es recomendado en el Datasheet como circuito de filtrado de la señal funciona de forma correcta aunque con una leve distorsión en la señal de salida, para solucionar esto, se implementó un buffer de ganancia unitaria a la salida de la señal del sensor, puesto que la resolución del conversor A/D es también proporcional a su voltaje de referencia así gracias a esto las variaciones en el voltaje de alimentación serán canceladas por este sistema (Figura 4.2).

Figura 4.1: Suministro de alimentación recomendado con desacople y filtrado en la salida



Fuente: Datasheet MPX5100DP

Figura 4.2: Sensor de Presión con LPF-RC y buffer como entrada a la siguiente etapa

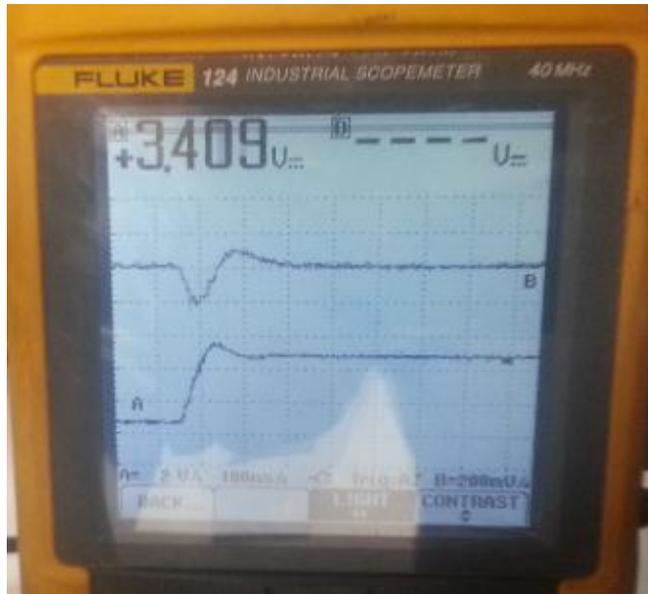


Fuente: Autores

### 6.1.2 Filtración de señales del ruido generado por la bomba

El anterior montaje va relacionado directamente con el sensor de presión diferencial y la entrada del convertor A/D del PIC, adicional a esto se debe sumar el ruido generado por la bomba de agua. Debido al excesivo ruido en la señal de salida no se podía tomar una muestra de calidad de la señal para poder realizar la parametrización de los datos, tal como lo muestra la Figura 4.3. Cabe resaltar que es muy importante asegurar las conexiones a tierra lo más cerca del cristal del PIC, como la bomba de agua es controlada por frecuencia la parte de alimentación de la tarjeta del controlador es filtrada mediante una serie de condensadores en paralelo entre los +5V y la tierra.

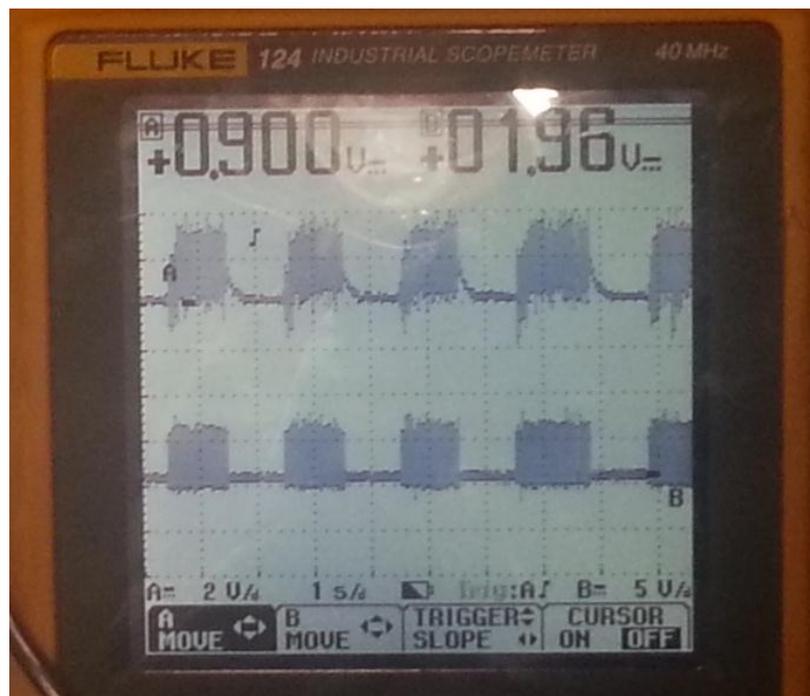
**Figura 4.3: Señal de salida del sensor comparada con la salida del controlador**



**Fuente: Autores**

Las perturbaciones por alta frecuencia afectan las señales de salida del sensor (Figura 4.4), por lo tanto adicional al filtrado en la fuente de alimentación, también se realizó un aislamiento de la bomba de agua mediante un cilindro de aluminio (Figura 4.5) aterrizado que mejoró considerablemente el ruido en las señales tomadas por el osciloscopio.

**Figura 4.4: Perturbaciones de la señal por alta frecuencia**



Fuente: Autores

Figura 4.5: Cilindro de aluminio para filtración de ruido



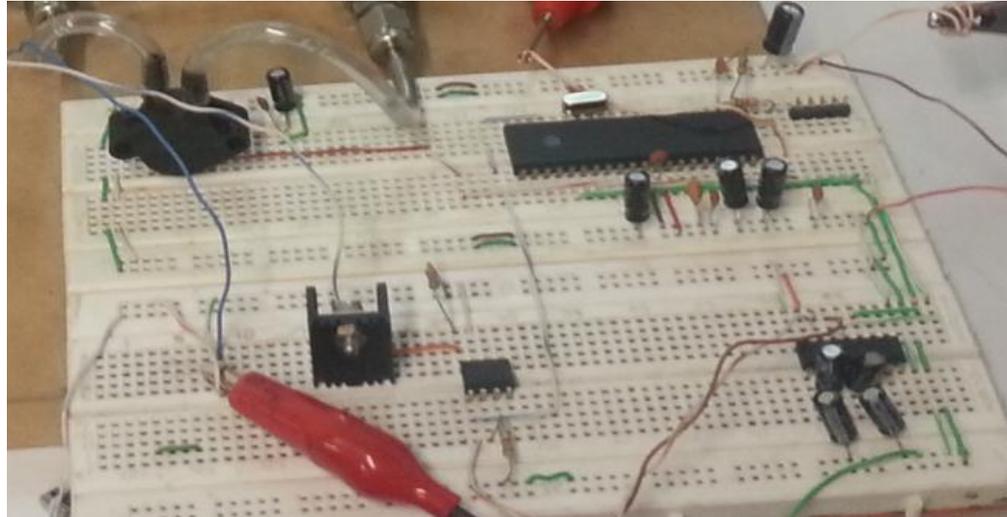
Fuente: Autores

## 6.2 Pruebas preliminares para la implementación del controlador

La construcción del diseño circuital del módulo del controlador se inició realizando el montaje del PIC18F452 programado con el algoritmo del controlador, el sensor con su respectiva configuración y la parte de potencia para la bomba de agua en un protoboard para poder comprobar el funcionamiento del controlador digital diseñado en el capítulo anterior antes de realizar su respectivo circuito impreso.

El proceso del montaje fue el siguiente: se montó el PIC con su determinada configuración y un cristal de 10MHz, la entrada del conversor A/D (pin 2) se conectó mediante un cable a la salida del sensor de presión diferencial que también se montó guiándose en la configuración citada en el capítulo 3, se anexo el circuito MAX 232 para la transmisión por puerto serial conectado al módulo USART del PIC (pin 25 y 26) y la salida del PWM (pin 17) del PIC se conectó a un optoacoplador que sirvió para separar la etapa de 5 voltios de la etapa de potencia que se generó mediante un transistor de potencia TIP 41C con disipador, finalmente fue la combinación de los módulos de control y comunicación, sensor y potencia tal como lo muestra la Figura 4.6.

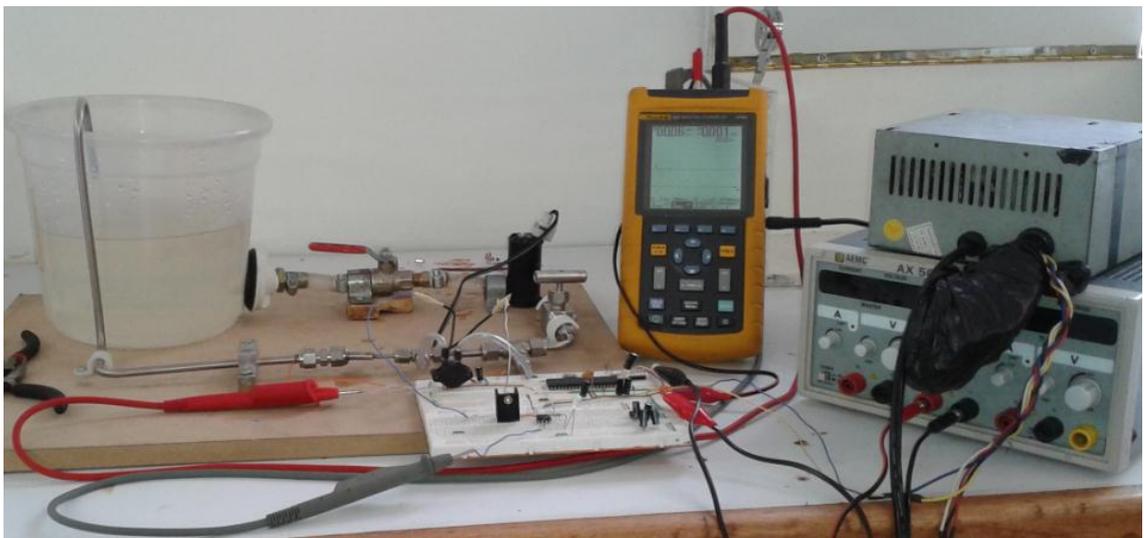
**Figura 4.6: Montaje de combinación de módulos**



**Fuente: Autores**

Se seleccionaron fuentes de +12 y -12 para la parte de potencia y para comunicación, sensor y control una fuente de +5V y se conectaron las sondas del osciloscopio de forma alterna entre la salida del sensor y el PWM. El procedimiento para poder verificar si el controlador digital funcionaba fue revisando la señal de salida del sensor de presión que mediante Matlab se cambiaba el set point y se empezaba a realizar perturbaciones al sistema mediante la válvula de aguja o simplemente tapando la salida de la tubería al balde, como se muestra en la Figura 4.7.

**Figura 4.7: Prueba preliminar de funcionamiento de la planta**



**Fuente: Autores**

## 6.2 Medición de la presión con voltaje de entrada ascendente-descendente

Para identificar el rango de presiones de la planta se toma los valores según el voltaje de entrada, esto con el fin de identificar entre que rangos debemos buscar el sensor de presión diferencial. Al utilizar el voltaje entre 5 y 22 Vdc que es lo máximo de la fuente se halla el máximo dato de presión que es 72 kilo pascales, el cual se concluye que debemos buscar un sensor con un rango de 0-100 Kilo pascales. En la figura 4.8 y tablas 4.1 y 4.2 se documentan los valores y resultados de las pruebas.



**Figura 4.8: Patrón de presión HEISE en prueba de presión ascendente y descendente**

**Fuente: Autores**

**Tabla 4.1: Tabla de presiones con voltaje de entrada ascendente**

<b>Vin (V)</b>	<b>P1 (kp)</b>	<b>P2 (kp)</b>
4.48	1.6	0.93
5.02	2.1	1.19
6	3.9	2.06
6.91	5.6	3.38
7.98	8.1	5.01
8.91	10.4	6.37
10.05	13.3	7.94
11.10	16.4	9.72
12.07	20.1	11.85
13.08	26.1	15.16
14.06	30.4	17.48
15.03	32	18.46
16.01	36.6	20.93
17.06	42.2	23.96
18.09	48.5	27.27
19.07	55.2	31.08
19.89	61.8	34.40
20.7	68	37.80
21.2	72.1	40

Fuente: autores

**Tabla 4.2: Tabla de presiones con voltaje de entrada descendente**

<b>Vin (V)</b>	<b>P1 (kp)</b>	<b>P2 (kp)</b>
20.6	69.2	38.62
19.7	64	35.56
18.7	58.3	32.59
17.8	53.2	29.92
16.7	47.8	26.93
16	43	24.28
15	38	21.60
14	32.8	18.86
13	28.3	16.38
12	24.1	14.04
11	19.7	11.60
10	16	9.55
9	12.5	7.59
8	9.3	5.70
7	6.3	3.98
6	4.1	2.39
5	2.3	1.45
4	1	1.88

Fuente: Autores

### 6.3 Medición de voltaje del sensor MPX5100DP

Se realizó mediciones de presión y salida de voltaje del sensor para saber hasta que rango máximo de voltaje nos variaba el sensor de presión diferencial MPX5100DP, a continuación se documentan en las figuras 4.9, 4.10 y los resultados en las tablas 4.3 y 4.4.

Figura 4.9: Fuente de voltaje en prueba de presiones ascendentes y descendentes

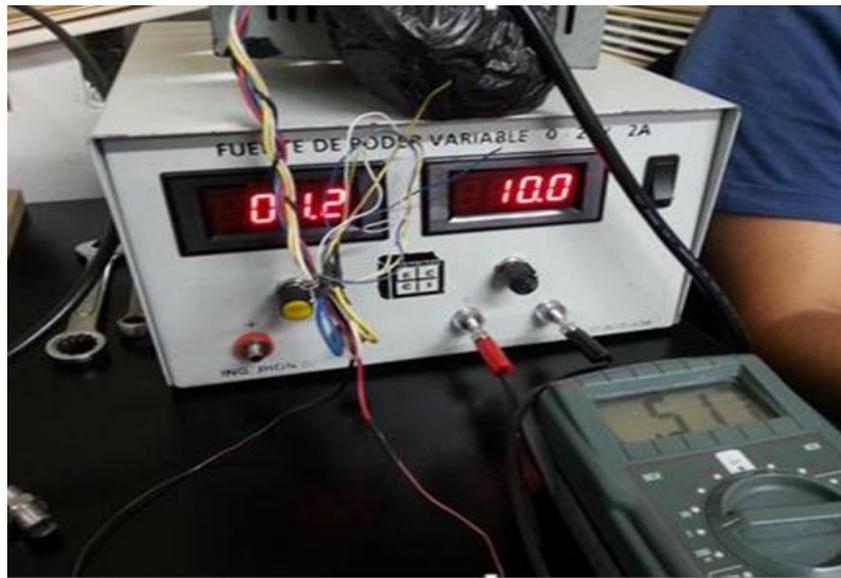


Figura: Autores

Figura 4.10: Equipo patrón HEISE para medición de presión diferencial



Fuente: Autores

**Tabla 4.3: Prueba de presión voltaje de entrada y salida del sensor de forma ascendente**

<b>V in (V)</b>	<b>P1 (kp)</b>	<b>P2 (kp)</b>	<b>Vout sensor (V)</b>
4.5	2.1	1.38	0.200
5	6	1.7	0.300
6	8.5	2.63	0.300
7	12.8	4.13	0.370
8	20.1	5.65	0.420
9	23.2	7.40	0.475
10	26.4	8.64	0.520
11	31.5	11	0.603
12	33	12.4	0.650
13	34	14.79	0.745
14	41.8	17.20	0.830
15	44	16.4	0.885
16	49.8	19.98	0.940
17	52.2	23.90	1.055
18	56.8	26.8	1.20
19	66.1	30.20	1.370
20	75.8	34.73	1.517
21	81.1	38.26	1.642
22	87	41.28	1.78

**Fuente: Autores**

**Tabla 4.4: Prueba de presión voltaje de entrada y salida del sensor de forma descendente**

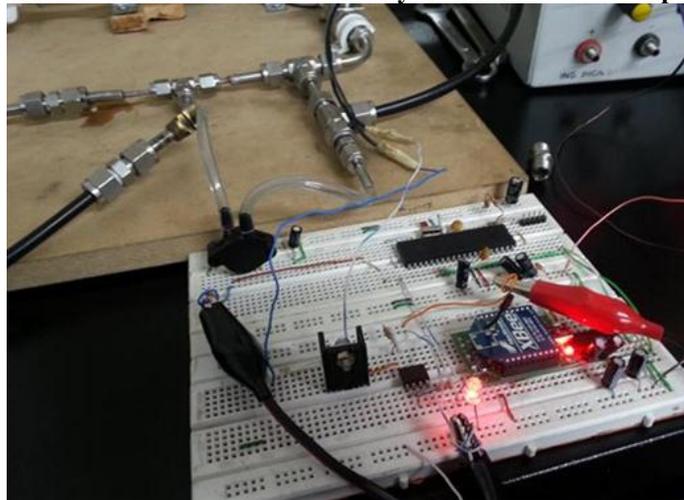
<b>V in (V)</b>	<b>P1 (kp)</b>	<b>P2 (kp)</b>	<b>Vout sensor (V)</b>
21	80.3	39.24	1.690
20	78.1	36.57	1.580
19	72.3	33.70	1.476
18	64.2	30.71	1.350
17	59.1	27.54	1.232
16	53	24.71	1.125
15	47	21.82	1.015
14	41.5	19.22	0.907
13	35.7	16.60	0.813
12	30	13.91	0.712
11	25.1	11.72	0.633
10	20.3	9.48	0.553
9	18.1	7.36	0.482
8	15.5	5.69	0.420
7	12.8	4.05	0.365
6	10.8	2.44	0.333
5	9.9	1.52	0.292
4	7.5	0.92	0.256

**Fuente: autores**

## 6.4 Medición de valores con el set point

Los datos obtenidos en la siguiente prueba son la comparación de presiones cuando establecemos el set point en Matlab para dar arranque a la planta por medio del cable serial o el puerto inalámbrico Xbee. A continuación en las figuras 4.11 y la tabla 4.5 se documentan valores y resultados de las pruebas realizadas.

**Figura 4.11: Medición del Vout del sensor y transmisión de datos por Zigbee**



Fuente: Autores

**Tabla 4.5: Medición de presión estableciendo set point**

<u>V<sub>o</sub></u>	D <sub>p</sub> (V)	P1 (kP)	P2 (kp)
0.1	0.256	-	-
0.2	0.256	-	-
0.3	0.256	-	-
0.4	0.256	11.2	5.86
0.5	0.537	17.1	9.01
0.6	0.641	20.5	12.97
0.7	0.741	25.3	14.85
0.8	0.842	30.3	17.61
0.9	0.946	34.6	20.45
1.0	0.947	34.4	20.42
1.1	0.948	34	20.45
1.2	1.24	47.7	28
1.3	1.350	53.5	30.75
1.4	1.45	57.5	33.29
1.5	1.55	61.9	35.90
1.6	1.65	66.4	38.45
1.7	1.65	66.4	38.46

Fuente: Autores

## 6.5 Resultados de la adquisición

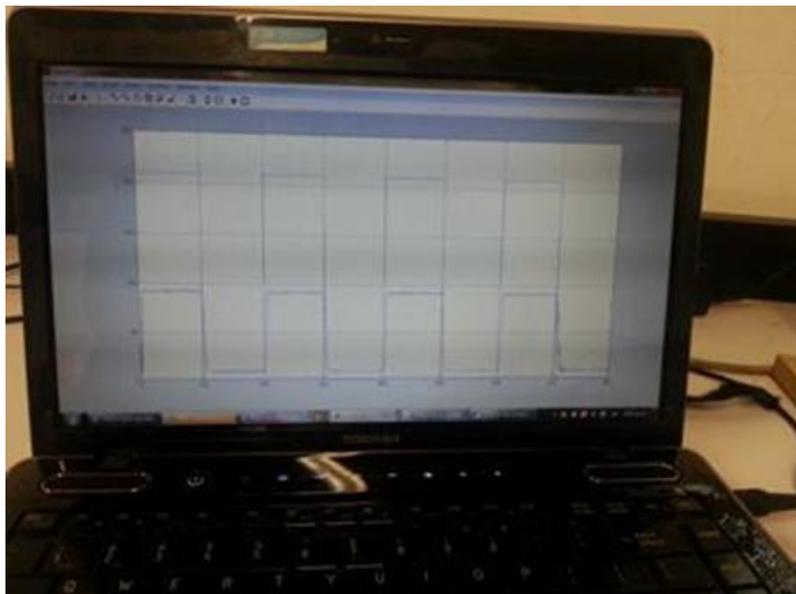
Luego de realizar las pruebas de adquisición y recolectar datos para hallar la función de transferencia del sistema mediante Matlab, se realizó la prueba mediante el osciloscopio de medir las señales de entrada y salida de la planta de presión, representadas en las figuras 4.12 y 4.13.

**Figura 4.12: Comparación señal de entrada vs salida**



Fuente: Autores

**Figura 4.13: Comparación señal de entrada vs salida Matlab**

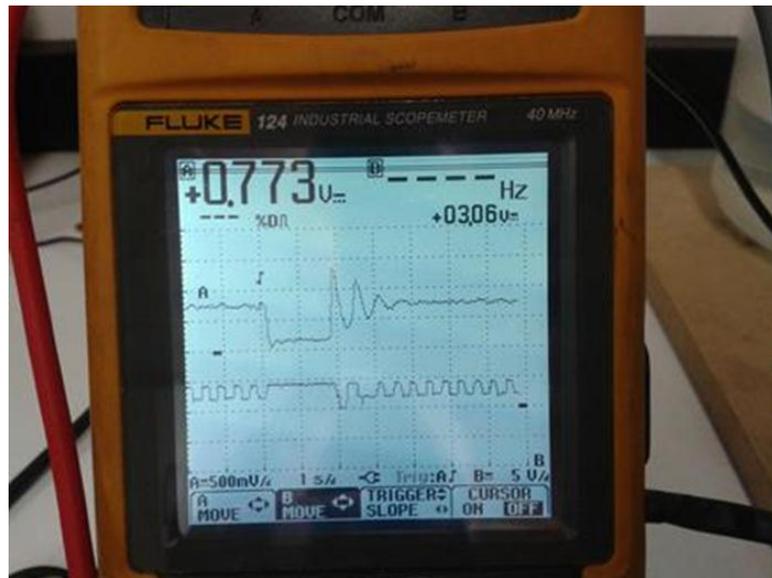


Fuente: Autores

## 6.6. Resultados del controlador

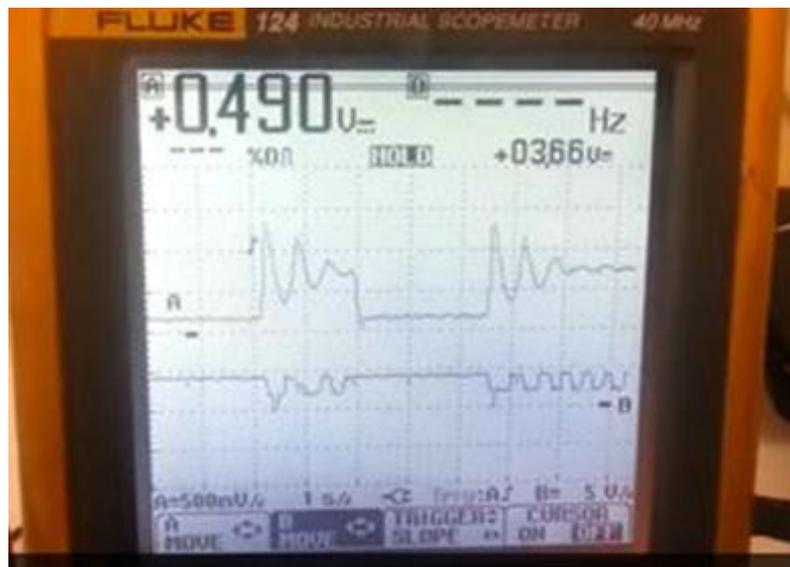
Al realizar las pruebas del controlador y tener varios puntos de set point se generan las gráficas 4.14 al 4.17. Que se describen a continuación donde se distinguen las variables cuando se generó obturación al proceso de control y sistema modificaba el control.

**Figura 4.14:** Salida del Sistema con restricción en la válvula en *Set point* de 0.8



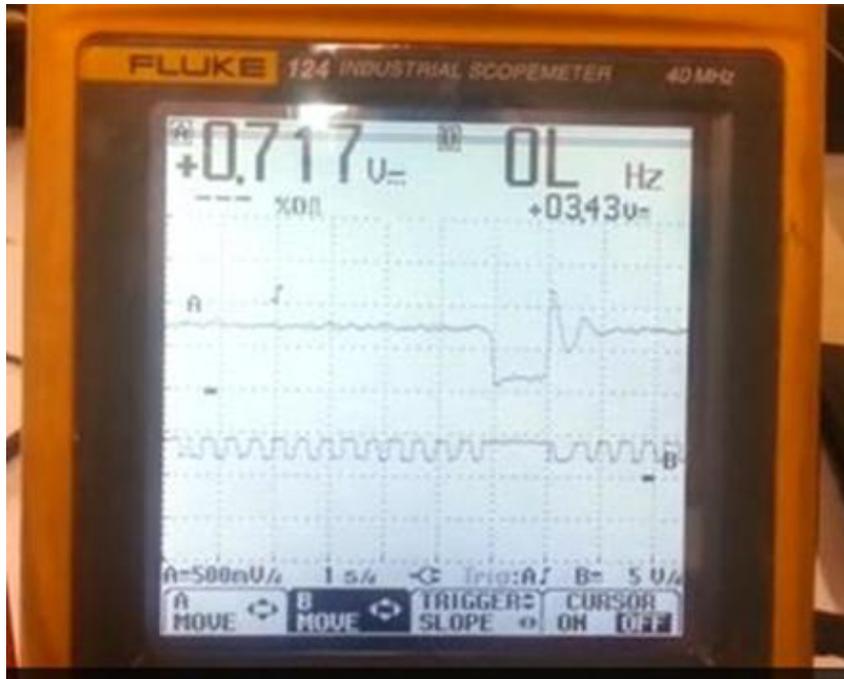
Fuente: Autores

**Figura 4.15:** Salida del Sistema con restricción en la válvula en *Set point* de 0.5



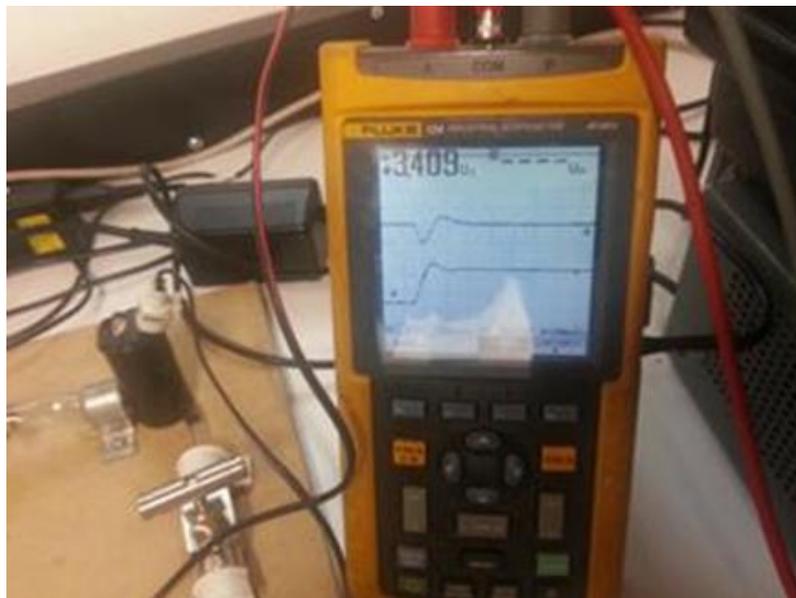
Fuente: Autores

Figura 4.16: Salida del Sistema con restricción en la válvula en *Set point* de 0.7



Fuente: Autores

Figura 4.17: Salida del Sistema Controlada en *Set point* de 0.7



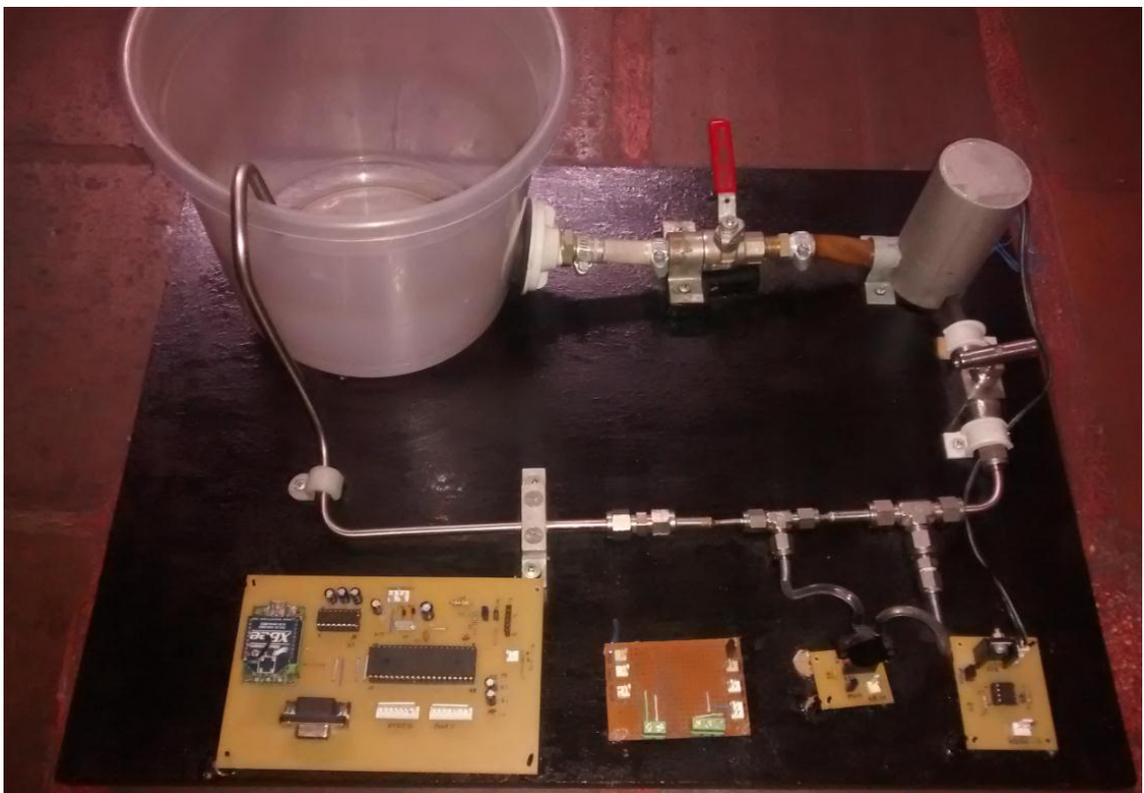
Fuente: Autores

## 6.7 Resultado final del sistema de control

Luego de las pruebas documentadas anteriormente se procedió al diseño de las tarjetas del sistema modular que en el capítulo 5 se documenta como se realizó el diseño de las tarjetas PCB mediante el programa KiCad. Se realizaron 3 diseños de tarjeta, primero la tarjeta de control que también tiene parte de la comunicación serial y ZigBee, segundo la tarjeta del sensor que viene con la configuración detalla en el Datasheet del sensor y por último la tarjeta de potencia que va relacionada directamente a la bomba de presión de agua.

A continuación se puede ver el resultado final de la construcción de las tarjetas en las figura 4.18-

**Figura 4.18: Resultado final de planta de control**



Fuente: Autores

## Conclusiones

- Se diseñó e implementó un prototipo de control de presión en función de la bomba y sus características, acoplado un sensor de presión diferencial.
- Usando la herramienta Ident de Matlab, se identificó la planta de presión y se obtuvo la función de transferencia.
- Por medio de la simulación en discreto del modelo matemático generado en Matlab, se diseñó e implementó un algoritmo de control para el sistema de presión.
- Se implementó una red Zigbee coordinador-router para la transmisión Full dúplex entre hardware y la interfaz gráfica.
- Se diseñó mediante la herramienta KiCad un sistema modular de hardware, para la implementación del módulo de control de presión, sensor de presión diferencial y acople para potencia de la bomba.
- Mediante la herramienta GUI de Matlab se diseñó una interfaz de usuario que permite reprogramar las variables del controlador el tiempo de muestreo y punto de trabajo. Además de visualizar la salida del sistema.

## Trabajos a Futuro

- Aumentar la frecuencia del microcontrolador para poder mejorar el tiempo de muestreo.
- Mejorar la comunicación Zigbee a un punto de acceso wifi para mejorar el alcance de la señal entre hardware y ordenador
- Mejorar la interfaz de usuario usando Labview o Visual Basic para mejorar la implementación y comunicación entre interfaz

## Referencias

- [1] A. J. Bouwens, *Digital Instrumentation*, (Ed, McGraw-Hill, USA), cap. 3 y 7 (1984).
- [2] E. Aldrete-Vidrio, F. Sandoval-Ibarra, y Juan Santana, Modelo *Eléctrico Equivalente para Sensores de Presión de Efecto Piezorresistivo*, Memorias CONCI'2000, ITP Puebla, pp: 15-19, mayo 2000.
- [3] M. Bañuelos S., J. Castillo H., G. Rayo L. S. Quintana T., R. Damián Z., J. Pérez S. Controlador PID de temperatura de tipo didáctico” Congreso Nacional de Instrumentación, UNAM México, D.F (2010)
- [4] MOTOROLA, Inc., Semiconductor Technical Data, *Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated*, Rev 6 (1998).
- [5] C. Winkler & J. Baum, MOTOROLA, Inc., Semiconductor Application Note, *Barometric Pressure Measurement Using Semiconductor Pressure Sensors*, Rev 1, (1997).
- [6] MOTOROLA, Inc., Semiconductor Technical Data, *Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated*, Rev 6 (1998).
- [7] A. Reodique and Warren Schultz, MOTOROLA, Inc., *Semiconductor Application Note, “Noise Considerations for Integrated Pressure Sensors*, Rev 1, (1998).
- [8] W. Stanlay & R. F. M. Smith, Guía para Mediciones Electrónicas y Practicas de Laboratorio, (Ed. Prentice Hall Hispanoamericana, México), (1992)
- [9] <http://www.mot-sps.com/models/bin/sensor2.html>
- [10] A. Rubaai, M. Castro-Sitiriche y A. Ofoli, «DSP-Based Laboratory Implementation of Hybrid Fuzzy-PID Controller Using Genetic Optimization for High-Performance Motor Drives,» *Industry Applications, IEEE Transactions on*, vol. 44, nº 6, pp. 1977-1986, 2008.
- [11] U. Umirov, S.-H. Jeong y J.-I. Park, «Applicability of ZigBee for real-time networked motor control systems,» de *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, 2008.
- [12] Z. Yang, L. Wu y X. Dong, «Control System Design for Contant-pressure Water Supply,» de *Multimedia and Information Technology (MMIT), 2010 Second International Conference on*, 2010.
- [13] C.-s. Zhang y J.-x. Ye, «Controller Design for Constant Water Pressure,» de *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, 2010.
- [14] S. Mehta, J. Katrodiya y B. Mankad, «Simulation, design and practical implementation of IMC tuned digital PID controller for liquid level control system,» de *Engineering (NUICONE), 2011 Nirma University International Conference on*, 2011.

- [15] B. Chang y X. Zhang, «Design of indoor temperature and humidity monitoring system based on CC2430 and fuzzy-PID,» de *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011, 2011.*
- [16] B. Chang y X. Zhang, «Design of indoor temperature and humidity monitoring system based on CC2430 and fuzzy-PID,» de *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011, 2011.*
- [17] R. Chen, H. Zhangxiang, Q. Jieting, H. Lanfang y C. Zixing, «Distributed data acquisition unit based on GPS and ZigBee for electromagnetic exploration,» de *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE, 2010.*
- [18] C. Gohiya, S. Sadistap, S. Akbar y B. Botre, «Design and development of digital PID controller for DC motor drive system using embedded platform for mobile robot,» de *Advance Computing Conference (IACC), 2013 IEEE 3rd International, 2013.*
- [19] Valdés Rodríguez, Durán Gómez, « Sistema de control didactico con graficación en tiempo real, » Congreso Interuniversitario de electrónica e intumentación, Morelia Mexico (1996)
- [20] Fu Jing-qi, Xu Cai-Xiang, Kan Bao-dong, Wang Hai-Kuan, «Desing and Simulation of Control Algorithms for WINCS, » de *Electrical Engineering and Automation Shanghai University, 2009 IEEE, 2009.*

# ANEXO A

## REGISTRO DE CONTROL (ADCON0)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** Bits de seleccion del reloj para la conversion A/D

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 **CHS2:CHS0:** Bits de seleccion del Canal

000 = channel 0, (AN0)  
 001 = channel 1, (AN1)  
 010 = channel 2, (AN2)  
 011 = channel 3, (AN3)  
 100 = channel 4, (AN4)  
 101 = channel 5, (AN5)  
 110 = channel 6, (AN6)  
 111 = channel 7, (AN7)

bit 2 **GO/DONE:** Bits de estado de la Conversión

When ADON = 1:

1 = La conversión A/D esta en curso (Si el bit es 1 la conversión A/D inicia automáticamente)

0 = La conversion ha terminado (el bit 0 es borrado por hardware al terminar la conversion)

bit 1 **No usado:** Su valor es '0'

bit 0 **ADON:** Activacion del conversor A/D

1 = Modulo conversor A/D esta activado

0 = Modulo conversor A/D esta desactivado

## REGISTRO DE CONTROL (ADCON1)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

bit 7 **ADFM:** Selección de formato del resultado  
 1 = Justificación a la derecha, los 6 bits más significativos de ADRESH son leídos como '0'  
 0 = Justificación a la izquierda, los 6 bits menos significativos de ADRESL son leídos como '0'

bit 6 **ADCS2:** selección de bits del reloj del Conversor A/D ( bits ADCON1)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Reloj del conversor
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	Frc (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	Frc (clock derived from the internal A/D RC oscillator)

bit 5-4 **No se usa:** Valor del registro '0'

bit 3-0 **PCFG3:PCFG0:** Configuración de las entradas del puerto A/D

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7 / 1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4 / 1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6 / 2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6 / 0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5 / 1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4 / 2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3 / 2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2 / 2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1 / 0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1 / 2

A = Analog input D = Digital I/O

C/R = # of analog input channels / # of A/D voltage references

## Calculo de tiempo de adquisición del conversor A/D

Para calcular el tiempo mínimo de adquisición, use la Ecuación 17-1. Esta ecuación supone que 1/2 LSB de error es usada (1024 pasos para la conversión A/D). El 1/2 LSB error es el error máximo permitido para que el A / D cumpla con su resolución especificada.

### ECUACIÓN 17-1: TIEMPO DE ADQUISICION

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \end{aligned}$$

### ECUACIÓN 17-2: TIEMPO DE CARGA MINIMO DEL A/D

$$\begin{aligned} V_{HOLD} &= (V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{-(T_c/CHOLD(R_{IC} + R_{SS} + R_s))}) \\ \text{or} \\ T_C &= -(120 \text{ pF})(1 \text{ k}\Omega + R_{SS} + R_s) \ln(1/2048) \end{aligned}$$

Ejemplo 17-1 muestra el calculo de el tiempo de adquisicon minimo  $T_{ACQ}$ . Este calculo es basado en los siguientes datos:

- $C_{HOLD}$  = 120 pF
- $R_s$  = 2.5 k $\Omega$
- Conversion Error  $\leq$  1/2 LSB
- $V_{DD}$  = 5V  $\rightarrow$   $R_{SS} = 7 \text{ k}\Omega$
- Temperature = 50°C (system max.)
- $V_{HOLD}$  = 0V @ time = 0

### EJEMPLO 17-1: CALCULANDO EL TIEMPO DE ADQUISICION MINIMO REQUERIDO

$$\begin{aligned} T_{ACQ} &= T_{AMP} + T_C + T_{COFF} \\ \text{Temperature coefficient is only required for temperatures} &> 25^\circ\text{C}. \\ T_{ACQ} &= 2 \mu\text{s} + T_C + [(Temp - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\ T_C &= -C_{HOLD} (R_{IC} + R_{SS} + R_s) \ln(1/2048) \\ &= -120 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \\ &= -120 \text{ pF} (10.5 \text{ k}\Omega) \ln(0.0004883) \\ &= -1.26 \mu\text{s} (-7.6246) \\ &= 9.61 \mu\text{s} \\ T_{ACQ} &= 2 \mu\text{s} + 9.61 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\ &= 11.61 \mu\text{s} + 1.25 \mu\text{s} \\ &= 12.86 \mu\text{s} \end{aligned}$$

## 17.2 SELECCION DEL RELOJ DE CONVERSIÓN

El tiempo de conversión del A/D por bit es definido como TAD. La conversión A/D requiere 12 TAD por conversión de 10 bits. La fuente del reloj de conversión A/D es un software seleccionable. Las siete posibles opciones para el TAD son:

- 2 T<sub>osc</sub>
- 4 T<sub>osc</sub>
- 8 T<sub>osc</sub>
- 16 T<sub>osc</sub>
- 32 T<sub>osc</sub>
- 64 T<sub>osc</sub>
- Internal A/D module RC oscillator (2-6 μs)

Para corregir las conversiones A/D, el reloj de conversión A/D (TAD) debe seleccionarse para asegurar un tiempo mínimo TAD de 1,6 ms.

## 17.3 CONFIGURACION DE PUERTOS ANALÓGICOS

El ADCON1, TRISA y TRISE controlan el funcionamiento de los pines del puerto A/D. Los pines del puerto que se desean como entradas analógicas, deben tener bit TRIS=1 (entrada). Si el bit de TRIS es 0 (salida), el nivel de salida digital (V<sub>OH</sub> o V<sub>OL</sub>) se convertirá.

La operación A/D es independiente del estado de los bits CHS2: CHS0 y los bits de TRIS.

La Tabla 17-1 muestra los tiempos de TAD resultantes derivados de las frecuencias de funcionamiento del dispositivo y el reloj de A/D de la fuente seleccionada.

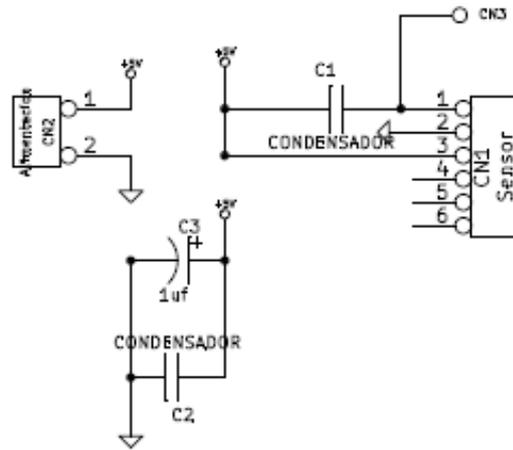
**TABLA 17-1: TAD vs. FRECUENCIAS DE FUNCIONAMIENTO DEL DISPOSITIVO**

AD Clock Source (TAD)		Maximum Device Frequency	
Operation	ADCS2:ADCS0	PIC18FXX2	PIC18LFXX2
2 T <sub>osc</sub>	000	1.25 MHz	666 kHz
4 T <sub>osc</sub>	100	2.50 MHz	1.33 MHz
8 T <sub>osc</sub>	001	5.00 MHz	2.67 MHz
16 T <sub>osc</sub>	101	10.00 MHz	5.33 MHz
32 T <sub>osc</sub>	010	20.00 MHz	10.67 MHz
64 T <sub>osc</sub>	110	40.00 MHz	21.33 MHz
RC	011	—	—

## Anexo B

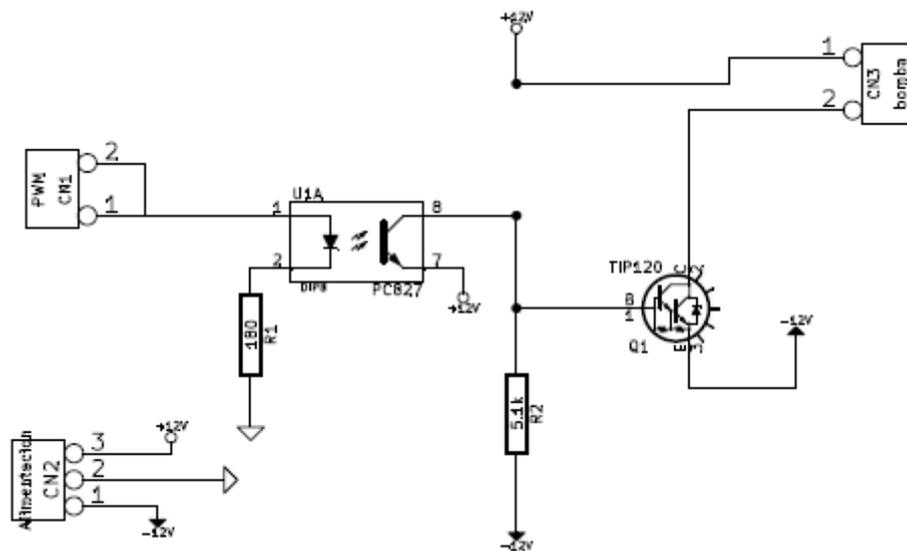
### Circuitos Esquemáticos de Tarjeta Sensor y potencia

#### Diseño Esquemático de la tarjeta del Sensor



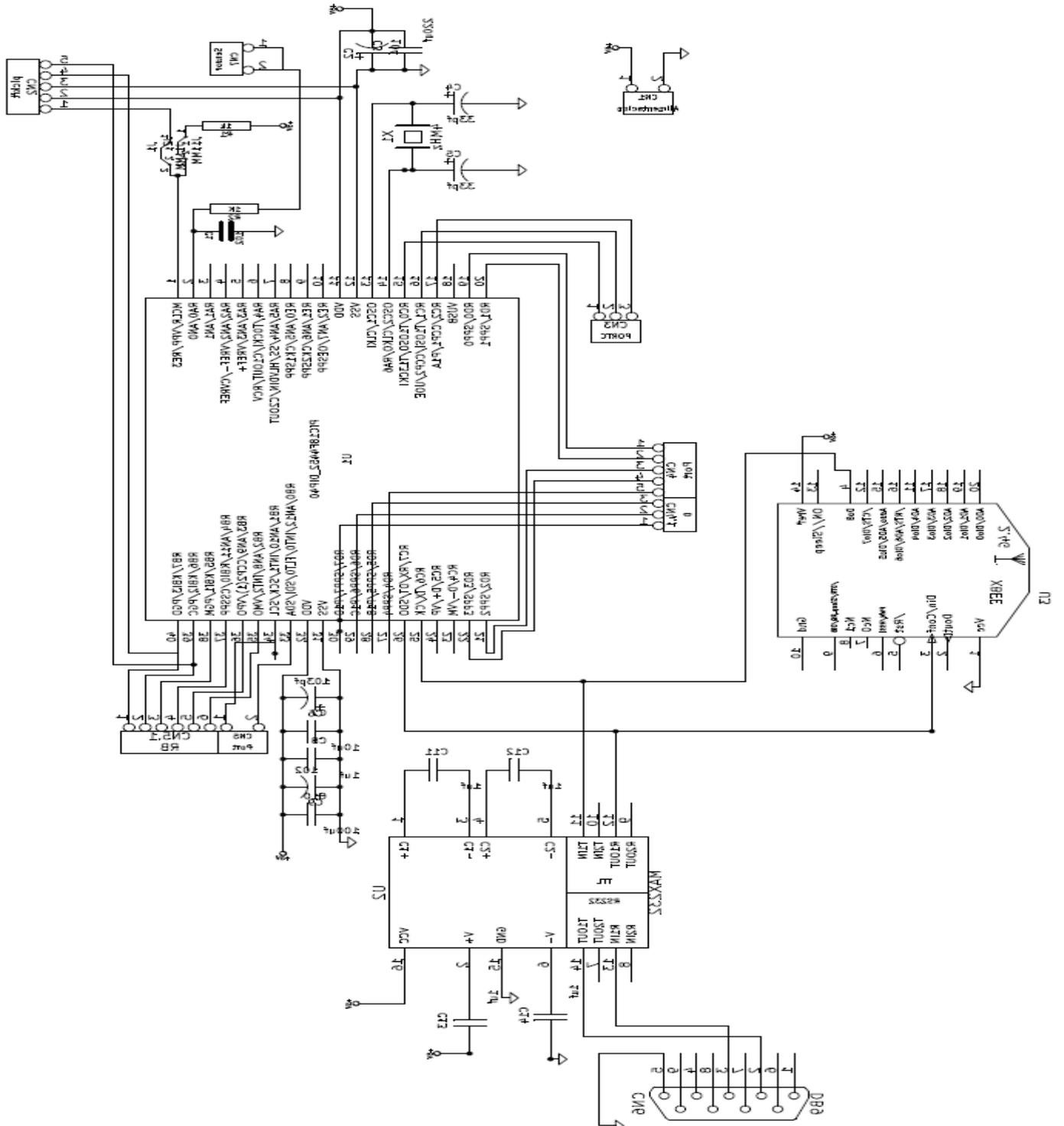
Fuente: Autores

#### Diseño Esquemático de la tarjeta de Potencia

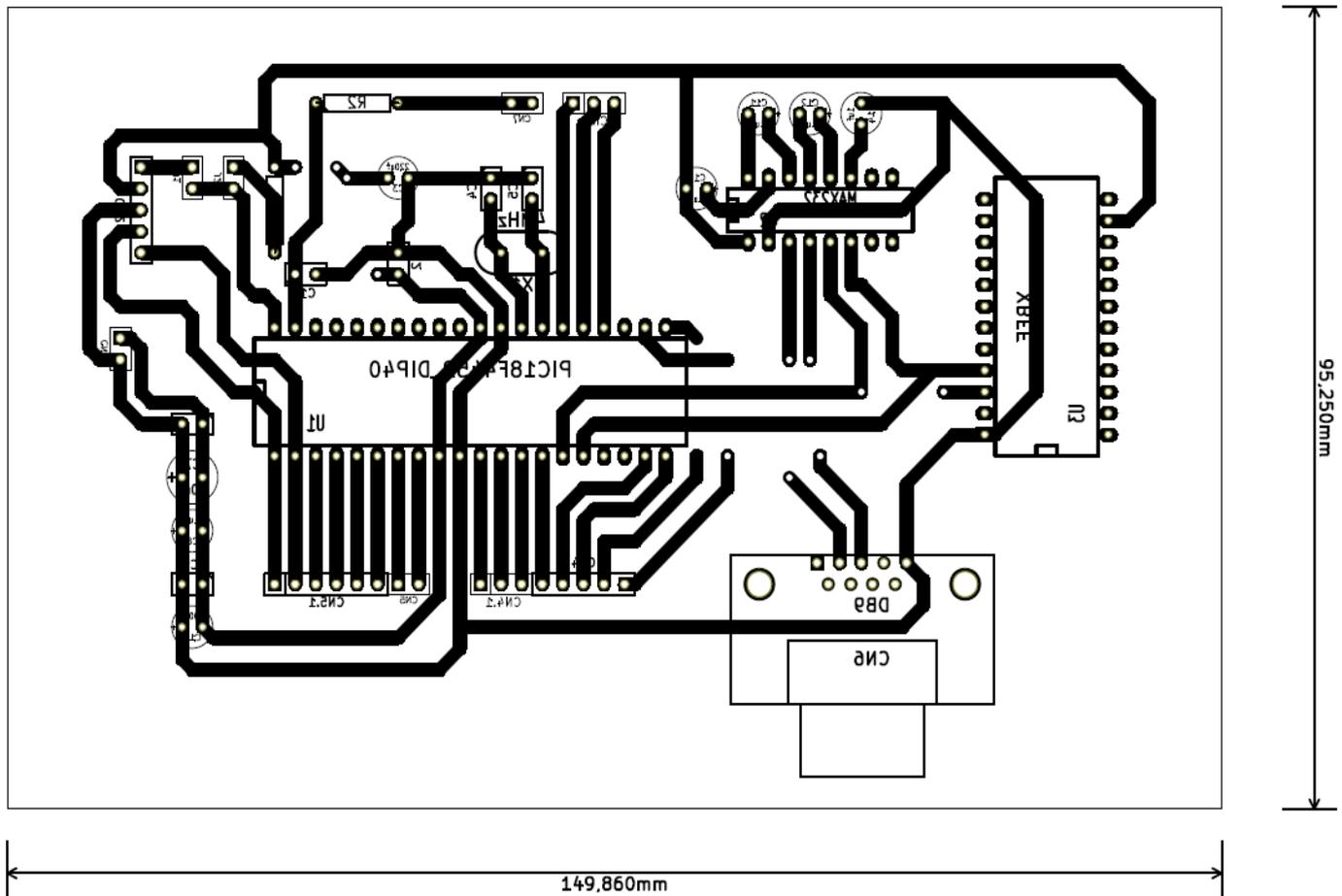


Fuente: Autores

# Diseño Esquemático de la Tarjeta de Control

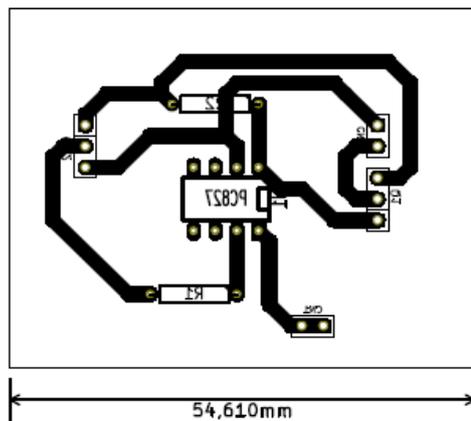


Diseños de placas de circuito impreso  
PCB de tarjeta programable y de comunicación

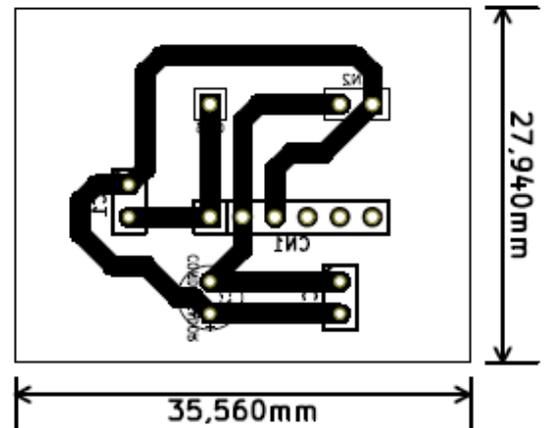


Fuente: Autores

PCB Tarjeta de Potencia



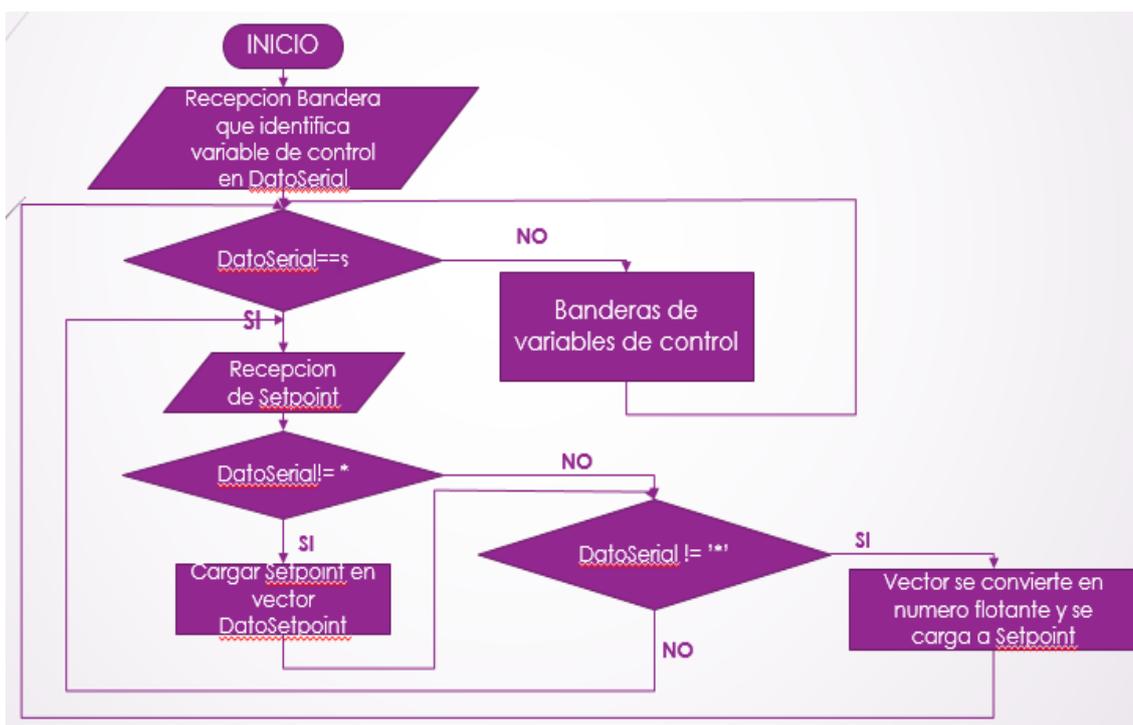
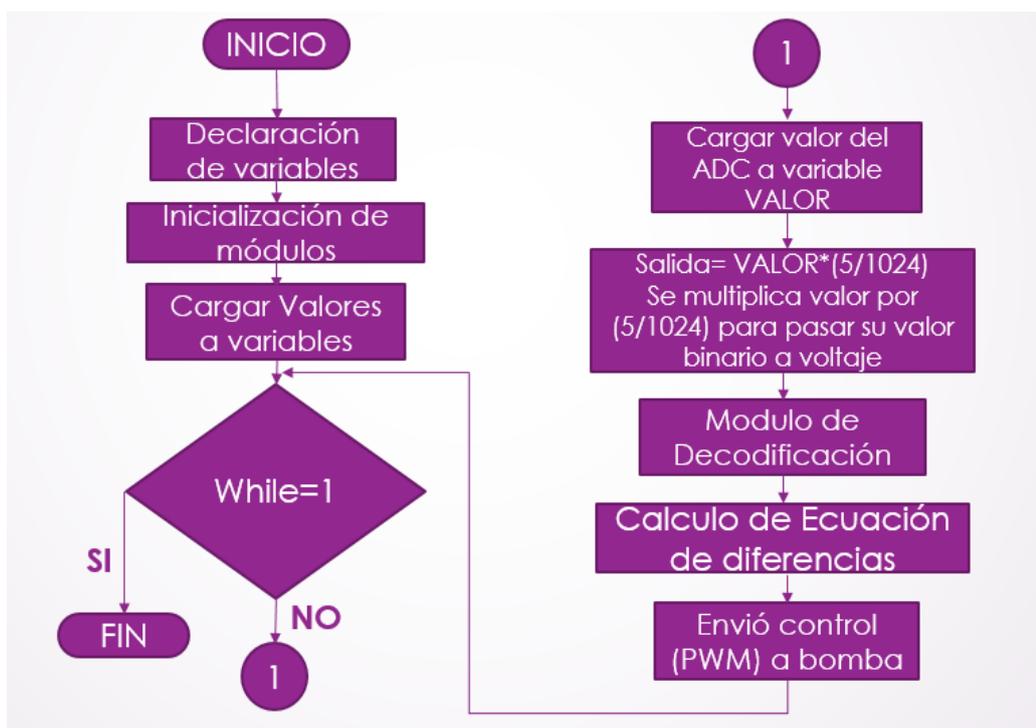
PCB tarjeta sensor



## Anexo C

### Controlador por software y modulo serial

El proceso del controlador por software y modulo serial se explican en los siguientes diagramas de flujo.



## Anexo D

### Lista de componentes y relación de precios del proyecto

COMPONENTES	REFERENCIA	CANTIDAD	PRECIO EN PESOS
Sensor de presión	MPX5100	1	90000
MóduloXbeeExplorer	Xbee	2	100000
PIC	PIC18F452	1	12000
Antena tipo cable 100m	XBeeTranseiver Series 2	2	174000
Transistor NPN de potencia	TIP41C	1	1200
Cristal de cuarzo 12MHz	XTAL 12MHz	1	600
Conversor RS232	Max323D	1	2700
Condensadores	Condensadores	10	4000
Resistencias	Resistencias	6	1000
Cable USB	Cable mini USB	1	3000
Tee inox	Tee inox ¼	1	62500
Tee inox	Tee inox 1/8	1	99000
Tubing inox	Tubing inox ¼	1	37000
Unión Inox	Unión Inox ¼	1	37700
Tuercas inox-felulas y contrafelulas	Tuercas inoxidables	7	147000
Tuercas inox-felulas y contrafelulas	Tuercas inoxidables	3	72000
Válvula de aguja	Válvula A inox de 1/4	1	195000
Soporte de 1	Soporte vía aluminio	1	6000
Recipiente Plástico	Recipiente plástico	1	6000
Registro de Bola	Registro inox de 1/4	1	35000
Bomba de agua 24V de parabrisas	Bomba de agua 24V de parabrisas	1	25000
Tabla soporte	Tabla soporte	1	7000
Fungibles y Consumibles	Fungibles y Consumibles	1	15000
		<b>TOTAL</b>	<b>1123700</b>