

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

“SISTEMA INFORMÁTICO INTERACTIVO PARA EL CONTROL DE LOS NIVELES DE LÍQUIDOS EN TANQUES MEDIANTE LA UTILIZACIÓN DE SENSORES, CON TECNOLOGÍA DE LÓGICA DIFUSA”

INFORME DE PROYECTO DE GRADUACION

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN: SISTEMAS TECNOLÓGICOS.

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN: SISTEMAS MULTIMEDIA.

Presentada por:

ANDY GABRIEL GONZÁLEZ REYES

JORGE LUIS LÓPEZ HUAYAMABE

Guayaquil - Ecuador
2010

AGRADECIMIENTO

A Dios.

A nuestros padres, familiares y amigos.

A todas las personas que nos ayudaron a cumplir este objetivo.

Al Dr.Xavier Ochoa C., Director dela Tesis, por su colaboración y sabia orientación en la elaboración de la tesis.

DEDICATORIA

A Dios que ha sido nuestro guía, en especial en los momentos más difíciles.

A nuestros padres, que siempre estuvieron con nosotros y que gracias a sus sabios consejos y enseñanzas hicieron posible alcanzar esta meta.

TRIBUNAL DE SUSTENTACIÓN

Ing. Jorge Aragundi
Presidente del Tribunal

Dr. Xavier Ochoa C.
**Director de Proyecto de
grado**

Ing. Marcelo Loor.
Miembro Principal

Ing. Sixto García
Miembro Principal

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

ANDY GONZÁLEZ REYES

JORGE LÓPEZ HUAYAMABE

RESUMEN

En el trabajo presentado: “Sistema informático interactivo para el control de los niveles de líquidos en tanques mediante la utilización de sensores, con tecnología de lógica difusa, fue desarrollado con el fin de ayudar al campo industrial, en la manera de controlar, calibrar y monitorear sus niveles de tanques en el momento de llenado y experimentando una ayuda al operario o usuario con una visualización nueva en este campo que es la de 3D.

En el primer capítulo nos enfocamos a una introducción, justificación y el análisis de objetivos tanto generales como específicos del proyecto así como las grandes posibilidades de aplicar este proyecto a las industrias. En este capítulo haremos un detalle del diagrama general del proyecto haciendo énfasis en cada estructura que compone el Diseño.

En el segundo capítulo nos enfocamos en el análisis previo del diseño tomando en cuenta los requerimientos de hardware como de software para su correcta implementación y funcionamiento, hablaremos de sensores, microcontroladores, comunicación de datos, análisis de métodos de inteligencia artificial (lógica difusa), entorno de desarrollo de software, modelamiento en 3D.

En el tercer capítulo hablaremos sobre el diseño general de la solución, detallando cada parte desde la adquisición de datos con el sensor, la tarjeta de adquisición de datos, medios de comunicación de datos, interfaz con el computador, software de desarrollo a usar, bases de datos para guardar la información, modelamiento de los datos obtenidos

por el sensor en 3D, el análisis de las funciones de membresía para la lógica difusa, el análisis de tiempos de llenado para el funcionamiento automático del sistema.

En el cuarto capítulo es la etapa de implementación del diseño tanto de hardware como de software, integrando sensores, tarjeta de adquisición, interfaz con el PC, métodos de visualización como es el programa principal y a su vez el modelo en 3D del tanque, y resaltaremos lo importante que es tener muchas precauciones al momento de realizar los montajes eléctricos-electrónicos para evitar daños de Hardware incluso hasta fallas a nivel de software.

El último capítulo muestra las pruebas realizadas, datos obtenidos y análisis de resultados que respaldan este documento.

Finalmente en las conclusiones y recomendaciones damos a conocer las sugerencias a las que llegamos a partir de los resultados obtenidos en las pruebas del proyecto.

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
INDICE GENERAL	VIII
ABREVIATURAS	XI
INDICE DE GRÁFICOS	XIII
INDICE DE TABLAS	XVIII
INTRODUCCIÓN	1
CAPITULO 1 ANTECEDENTES	4
1.1 Control de Niveles de líquidos mediante sensor mediante Lógica difusa con representación grafica 3D en tiempo real	4
1.2.- Justificación	5
1.3.-Objetivo general	5
1.4- Objetivos específicos	5
1.5.- Aplicaciones a la Industria	6
CAPITULO 2 ANALISIS DEL SISTEMA	8
2.1.- Análisis de requerimientos del Sistema	8
2.2.- Características funcionales del sensor	12
2.3.- Características del microcontrolador para la adquisición de datos	13
2.4 Modelamiento en 3D del tanque utilizando frameworks Multimedia	13
2.5.-Análisis de funciones de membresía para el algoritmo de lógica difusa para el prendido y apagado de bombas para drenar y surtir líquidos	18

2.6. Protocolos de comunicación entre el PC y el microcontrolador	29
2.7. Especificaciones electrónicas para la seguridad entre el mundo físico y el hardware	33
CAPITULO 3 DISEÑO.....	35
3.1 Diseño de la arquitectura.....	35
3.1.1 Parámetros de entrada globales.....	36
3.1.2 Diseño de hardware.....	36
3.1.3 Diseño de software.....	38
3.2 Especificaciones de diseño del sensor en la tarjeta AD.....	38
3.3 Características del sensor ultrasónico maxzonar	40
3.4 Acondicionador de señal y Modelo lineal con amplificadores operacionales	43
3.5 Especificaciones de diseño usando MICRO 16F873A	44
3.5.1 Entradas analógicas	46
3.5.2 Convertidor analógico digital.....	47
3.5.3 Salida al LCD Hitachi	48
3.5.4 Comunicación serial mediante USART y norma RS232	50
3.6 Interfaz serial con MAX-232	60
3.7 Modelamiento 3D del tanque	60
3.7.1 Perspectiva de cámara.....	72
3.7.2 Luz ambiental	75
3.7.3 Luz difusa, material difuso.....	76
3.8 Representación de datos.....	79
3.8.1 Tiempo de llenado	79
3.8.2 Promedio de llenado	81
3.8.3 Altura y presión del líquido del tanque	83
3.9 Diseño de la lógica difusa.....	84
3.9.1 Funciones de membresía y Diseño de las funciones de membresía	86
3.9.2 Control on/off de bomba mediante esquemas de tiempo.....	90
3.9.3 Base de conocimiento motor de bases de datos.....	94
3.10 Interfaz interactiva del usuario.....	96
3.11 Diseño del plan de pruebas	97
CAPITULO 4 IMPLEMENTACIÓN DEL DISEÑO.....	105
4.1 Diagrama de conexiones generales	105
4.2 Montaje de Tanque Sensor	108
4.3 Esquema electrónico del sensor.....	110
4.4 Esquema electrónico de la etapa de acondicionamiento	112
4.5 Esquema electrónico del PIC16F873A	113

4.6 Esquema electrónico de la Interfaz de comunicación serial.....	114
4.7 Esquema grafico del tanque 3D	115
4.8 Interfaz.....	117
4.8.1 Pantalla principal	119
4.8.2 Controles principales	119
4.9 Esquema eléctrico –electrónico bomba de agua.....	122
4.10 Seguridad electrónica	125
4.11 Herramientas de desarrollo de hardware, software y Plataforma de programación	126
4.11.1 Herramientas de simulación de Hardware	126
4.11.2 Entorno de desarrollo para programación en pic.....	126
4.11.3 Entorno de desarrollo .net.....	126
CAPITULO 5 PRUEBAS	128
5.1 Pruebas del sensor.....	131
5.2 Pruebas del acondicionador de señal.....	135
5.3 Pruebas del PIC16F873A.....	136
5.3.1 Pruebas del convertidor A/D	136
5.3.2 Pruebas de comunicación serial con USART	140
5.4 Prueba de la interfaz serial con MAX232	143
5.5 Pruebas de entrada de datos al PC en tiempos determinados.....	145
5.6 Pruebas de modelado en 3D y salida de datos.....	148
5.7 Pruebas de las funciones de membresía para el control de la bomba.....	153
5.8 Análisis de los resultados de las pruebas.....	166
CONCLUSIONES Y RECOMENDACIONES.....	171
REFERENCIAS BIBLIOGRÁFICAS.....	176

ABREVIATURAS

A	Amperio
ADC	Convertidor Analógico Digital
C	Lenguaje C
CM	Centímetro
CD-ROM	Unidad de CD
DC	Corriente directa
DLL	Librería de Enlace Dinámica
DSP	Procesamiento Digital de Señales
DSPIC	Microcontrolador de Microchip con soporte para procesamiento de señales
EEPROM	Memoria de Lectura y Escritura Eléctricamente Borrable Programable
G	Lenguaje Grafico G
GPIO	Bus de Interfaz de Propósito General
IMAQ	Adquisición de Imágenes
I2C	Circuito Inter-Integrados.
JPEG	Grupo de Expertos Fotográficos Conjuntos
LABVIEW	Laboratorio de Instrumentación Virtual de Bancos de Trabajos de Ingeniería
LED	Diodo Emisor de Luz
LCD	Pantalla de Cristal Líquido
MB	Mega bites
MDF	Tablero de Fibra de Densidad Media
NI	National Instruments
OCR	Reconocimiento Óptico de Caracteres
PC	Computador Personal
PIC	Microcontrolador de Microchip

PNG	Red de Gráficos Portátiles
PWM	Modulación de ancho de pulso
PXI	PCI de extensiones para Instrumentación
RAM	Memoria de Acceso Aleatorio
ROI	Región de Interés
SPI	Interfaz Serial Periférica
TCP	Protocolo de Control de Transmisión
TTL	Lógica transistor - transistor
UART	Receptor-Transmisor Asíncrono Universal
UDP	Protocolo de Datagrama de Usuario
USB	Bus serial universal
UTP	Cable trenzado
V	Voltio
VI	Instrumento virtual
VAC	Voltio de corriente alterna
WMV	Formato de Video de Windows Media

INDICE DE GRÁFICOS

<i>Figura I Diagrama General</i>	1
<i>Figura II Tanque industrial</i>	3
<i>Figura 2.1 Tipo sensores</i>	9
<i>Figura 2.2 Tarjeta de adquisición de datos</i>	10
<i>Figura 2.3 Cable serial</i>	10
<i>Figura 2.4 Tecnología wpf</i>	12
<i>Figura 2.5 Microcontrolador</i>	13
<i>Figura 2.6 interfaz wpf</i>	15
<i>Figura 2.7 Esquema de la lógica difusa</i>	18
<i>Figura 2.8 Función de pertenencia</i>	20
<i>Figura 2.9 Función constante</i>	21
<i>Figura 2.10 Funciones variables</i>	21
<i>Figura 2.11 Variables lingüísticas</i>	22
<i>Figura 2.12 Modelo trapezoide</i>	23
<i>Figura 2.13 Entorno físico de sistema de control</i>	25
<i>Figura 2.14 Trama de datos</i>	30
<i>Figura 2.15 Max rs232</i>	32
<i>Figura 3.1 Esquema de la arquitectura general</i>	35
<i>Figura 3.2 Parámetros de entradas globales</i>	36
<i>Figura 3.3 Diseño de hardware</i>	36
<i>Figura 3.4 Diseño de software</i>	38
<i>Figura 3.5 Sensor max sonar</i>	39
<i>Figura 3.6 Diseño para el sensor</i>	42
<i>Figura 3.7 Diseño sin etapa amplificador</i>	44
<i>Figura 3.8 Diseño funcional del sensor</i>	44
<i>Figura 3.9 Especificaciones del microcontrolador</i>	45

<i>Figura 3.10 Entradas analógica del pic</i>	46
<i>Figura 3.11 LCD hitachi</i>	48
<i>Figura 3.12 Conexiones del LCD al pic</i>	49
<i>Figura 3.13 Formato de comunicación asincrónica</i>	51
<i>Figura 3.14 Transmisión asincrónica a máxima velocidad</i>	54
<i>Figura 3.15 Diagrama de estados de la tarjeta de Adquisición de Datos</i>	55
<i>Figura 3.16 Algoritmo general del pic</i>	56
<i>Figura 3.17a Código fuente del pic</i>	57
<i>Figura 3.17b Código fuente del pic</i>	58
<i>Figura 3.17c Código fuente del pic</i>	59
<i>Figura 3.18 Interfaz con max232</i>	60
<i>Figura 3.19 Sintaxis del viewport3D</i>	62
<i>Figura 3.20 modelado de un cubo</i>	65
<i>Figura 3.21 Código fuente del modelado</i>	68
<i>Figura 3.22a Código fuente del modelado</i>	69
<i>Figura 3.22b Código fuente del modelado</i>	70
<i>Figura 3.23 Sintaxis del viewport3D</i>	71
<i>Figura 3.24 Sintaxis de la perspectiva de cámara</i>	72
<i>Figura 3.25 Sintaxis de la perspectiva de cámara</i>	74
<i>Figura 3.26 Sintaxis de la perspectiva de cámara</i>	74
<i>Figura 3.27 Sintaxis de direccionamiento de luz</i>	75
<i>Figura 3.28 Código fuente de xaml del tanque</i>	77
<i>Figura 3.29 Diagrama general del modelado</i>	78
<i>Figura 3.30 Esquema de representación de los datos</i>	79
<i>Figura 3.31 Diagrama base de datos</i>	81
<i>Figura 3.32 Promedio llenado</i>	82
<i>Figura 3.33 Modelo de altura y presión</i>	84
<i>Figura 3.34 Diagrama de nivel</i>	85
<i>Figura 3.35 Función de membresía del sistema</i>	86

<i>Figura 3.36 Esquema fuzzificación / defuzzificación</i>	<i>86</i>
<i>Figura 3.37a Código fuente del motor de inferencia.....</i>	<i>89</i>
<i>Figura 2.37b Código fuente del motor de inferencia.....</i>	<i>90</i>
<i>Figura 3.38 Control on/off.....</i>	<i>91</i>
<i>Figura 3.39 Función de membresía del sistema on/off.....</i>	<i>92</i>
<i>Figura 3.40 Control por modo automático</i>	<i>93</i>
<i>Figura 3.41 Modo emergente.....</i>	<i>94</i>
<i>Figura 3.42 Base de conocimiento</i>	<i>95</i>
<i>Figura 3.43 Interfaz de usuario.....</i>	<i>96</i>
<i>Figura 4.1 Diagrama de conexiones generales.....</i>	<i>105</i>
<i>Figura 4.2 Pcb de la tarjeta de adquisición de datos</i>	<i>106</i>
<i>Figura 4.3 Tarjeta de adquisición de datos del proyecto</i>	<i>106</i>
<i>Figura 4.4 Tarjeta de adquisición de datos con sensor</i>	<i>107</i>
<i>Figura 4.5 Montaje de tanque sensor</i>	<i>108</i>
<i>Figura 4.6 Montaje de tanque sensor</i>	<i>109</i>
<i>Figura 4.7 Montaje de tanque sensor</i>	<i>109</i>
<i>Figura 4.8 Montaje de tanque sensor</i>	<i>110</i>
<i>Figura 4.9 Esquema electrónico del sensor</i>	<i>110</i>
<i>Figura 4.10 Esquema electrónico del sensor</i>	<i>111</i>
<i>Figura 4.11 Esquema electrónico del sensor</i>	<i>112</i>
<i>Figura 4.12 Esquema electrónico del PIC.....</i>	<i>113</i>
<i>Figura 4.13 Esquema electrónico del Pic.....</i>	<i>114</i>
<i>Figura 4.14a Esquema grafico del tanque.....</i>	<i>115</i>
<i>Figura 4.14b Esquema grafico del tanque.....</i>	<i>115</i>
<i>Figura 4.14c Esquema grafico del tanque</i>	<i>116</i>
<i>Figura 4.15 Interfaz inicial.....</i>	<i>117</i>
<i>Figura 4.16 Configuración inicial del sistema</i>	<i>117</i>
<i>Figura 4.17 Ingreso de datos del tanque.....</i>	<i>118</i>
<i>Figura 4.18 Pantalla principal</i>	<i>119</i>

Figura 4.19 Información del tanque.....	120
Figura 4.20 Indicadores del sistema	120
Figura 4.21 Notificaciones del sistema	121
Figura 4.22 Funcionamiento grafico del sensor.....	121
Figura 4.23 Esquema eléctrico electrónico.....	122
Figura 4.24a Esquema eléctrico electrónico.....	123
Figura 4.24b Esquema de conexiones de bomba.....	123
Figura 4.24c Conexiones de bomba Dc.....	124
Figura 4.24d Conexiones de bomba DC.....	124
Figura 4.25 Seguridad electrónica	125
Figura 4.26 Sección de Firmware de apagado de bomba	126
Figura 4.27 Ubicación de diodos.....	126
Figura 5.1 Pruebas del sensor.....	131
Figura 5.2 Grafica lineal Distancia en cm Vs Voltaje Analógico	134
Figura 5.3 Pruebas de acondicionador del Pic.....	135
Figura 5.4a Salida por Hypertemirnal con 0.5V.....	137
Figura 5.4bSalida por Hyperterminal con 0.75V.....	138
Figura 5.4cSalida por Hyperterminal con 1V.....	138
Figura 5.4d Salida por hyperterminal con 2V.....	139
Figura 5.4eSalida por hyperterminal con 2.55V	139
Figura 5.5 Resultado del osciloscopio a 1V	141
Figura 5.6Resultado del osciloscopio a 2.55V.....	142
Figura 5.7 Resultado de envío de trama \$I0F.....	144
Figura 5.8 Resultado de envío de trama \$I1F.....	144
Figura 5.9 Pruebas de entrada de datos al PC en tiempos determinados.....	146
Figura 5.10 Llenado en 3D para 1:30.720.....	150
Figura 5.11 Llenado en 3D para 3:00.412.....	151
Figura 5.12 Llenado en 3D para 3:57.891.....	151
Figura 5.13 Llenando en 3D para 5:24.324.....	152

Figura 5.14 Esquema de lógica Difusa	153
Figura 5.15 Grafica del nivel del llenado sin Lógica Difusa	157
Figura 5.16 Grafica del nivel de llenado con Lógica Difusa	157
Figura 5.17 Dato erróneo del sensor y ejecución del modo Automático.....	159
Figura 5.18 Dato erróneo del sensor y ejecución del modo Apagado de Emergencia	159
Figura 5.19 Histograma usando lógica difusa	165
Figura 5.20 Histograma sin usar lógica difusa	165
Figura 5.21 Distribución Normal (Con lógica difusa, sin lógica difusa).	166

INDICE DE TABLAS

<i>Tabla 1 Características del max sonar</i>	40
<i>Tabla 2 Características principales del microcontrolador</i>	45
<i>Tabla 3 Descripción de pines del LCD</i>	49
<i>Tabla 4 Pruebas y Casos de Usos asociados</i>	130
<i>Tabla 5 Resultados de voltajes medidos</i>	132
<i>Tabla 6 Resultados Voltaje vs Nivel</i>	133
<i>Tabla 7 Resultados de señal analógica de voltaje y salida por Hyperterminal..</i>	137
<i>Tabla 8 Resultados Señal analógica de voltaje y Salida digital a RS232</i>	141
<i>Tabla 9 Resultado en promedio de tiempos de llenado</i>	146
<i>Tabla 10 Resultado del tiempo de llenado vista en 3D</i>	150
<i>Tabla 11 Datos de la función de Membresía</i>	153
<i>Tabla 12 Resultados de la evaluación de las funciones de Membresía</i>	155
<i>Tabla 13 Resultados del sistema con Lógica difusa</i>	156
<i>Tabla 14 Resultados del sistema sin Lógica difusa</i>	156
<i>Tabla 15 Muestras de pruebas de la base de datos</i>	162
<i>Tabla 16 Resultados de las Pruebas de la Base de Datos usando Lógica Difusa ..</i>	163
<i>Tabla 17 Resultados de las pruebas con Lógica Difusa mostrando el porcentaje de error</i>	164
<i>Tabla 18 Resultados de las pruebas sin Lógica Difusa mostrando el porcentaje de error</i>	164

INTRODUCCIÓN

El proyecto que presentamos cumple con la función de censar líquidos, para este caso el agua como líquido de referencia, en depósitos (tanques de plástico), mediante una tarjeta de adquisición de datos la misma utiliza un software de computadora y permite visualizar el contenido del líquido en 3D. A continuación mostraremos un diagrama general del proyecto:

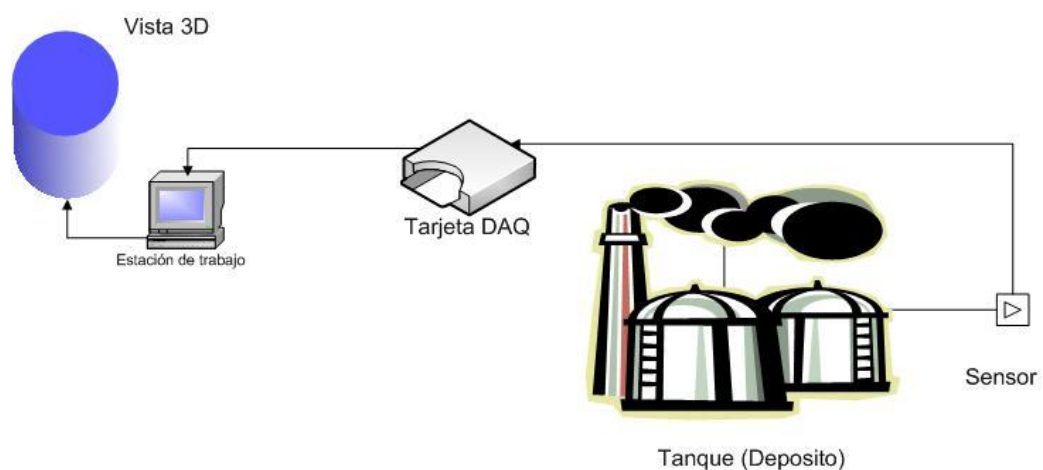


Figura I Diagrama General

En la actualidad, las empresas de carácter industrial, excluyen de su mecanismo de control y seguridad en los repositorios de sus materias primas, un medio donde se pueda visualizar el nivel de los líquidos, además de controlar su desalojo en cada descarga; y controlar el nivel mínimo y máximo hasta que se llenan los tanques.

Surge también la necesidad de estas empresas industriales, según un análisis estadístico que se realizó, que casi un 65 % de las mismas, tienen problemas de calibración y de fugas ocasionando pérdidas de materia prima innecesaria, restando eficiencia a los procesos de producción. Por estas razones se desea diseñar e implementar este sistema interactivo donde el operador pueda observar con mucha precisión y en cada momento los parámetros de control de cada tanque como su altura, densidad y presión.

Con esta problemática que tienen las industrias, solucionaremos esta deficiencia mostrándole al usuario/operador, calibración y exactitud de las medidas de los niveles de tanques en intervalos de tiempo, estabilidad del sensor, y características físicas del tanque como lo son presión, altura, densidad del líquido.

Además de evitar que los operadores corran el riesgo de sufrir accidentes por tener contacto con alguno de estos líquidos, el sistema ayudará al operador/usuario por medio de una interfaz interactiva y confiable, a evitar accidentes físicos por contacto directo con los líquidos.

Este proyecto representa un aporte muy importante para el tipo de industrias donde la materia prima principal reside en tanques como por ejemplo: laboratorios farmacéuticos, industrias químicas, las fábricas de bebidas, las industrias alimenticias, y de productos

lácteos etc., en donde el nivel de precisión son muy importantes en sus repositorios y por el nivel de seguridad. Ver figura II:



Figura II Tanque industrial

CAPITULO 1 ANTECEDENTES

1.1 Control de Niveles de líquidos mediante sensor mediante Lógica difusa con representación grafica 3D en tiempo real

Los requerimientos dividen al proyecto en varios componentes: Un sensor, que captura la señal analógica; una tarjeta de adquisición de datos, para el procesamiento de la señal, un medio de comunicación hacia el computador, y la visión en 3D. Un aspecto principal es la utilización de inteligencia artificial para el control del llenado, esto es con Lógica Difusa.

1.2.- Justificación

A lo largo de muchas investigaciones en las industrias, que guardan materia prima entanques, se observó que la mayoría poseen mecanismos de automatización en sus controles de niveles que son obsoletos, un ejemplo de estos son los medidores por boya o por simple inspección. Con el avance de la tecnología, las industrias son más exigentes en calibración, monitorización, y control en todos sus procesos. Es por esto que el proyecto muestra con un prototipo, como se puede utilizar inteligencia artificial para controlar los tiempos de llenados y mostrar el proceso en una gráfica 3D.

1.3.-Objetivo general

Medir el nivel de un líquido contenido en un tanque en intervalos pre-establecidos mediante un algoritmo de lógica difusa para el control automático de llenado de tanque y visualizarlo en 3D.

1.4- Objetivos específicos

- ✓ Mejorar el control de calidad del proceso de llenado de los Tanques, realizando auto calibración en cada llenado.
- ✓ Aplicar la inteligencia artificial para mejorar el proceso de llenado de los tanques y evitar errores en el proceso.
- ✓ Automatizar el proceso de llenado de tanques de plástico, utilizando como líquido de referencia el agua.

- ✓ Mostrar al usuario en intervalos pre-establecidos una nueva experiencia en visualización por computador 3D.

1.5.- Aplicaciones a la Industria

Este proyecto tiene como propósito mejorar la automatización de los sistemas de detección de nivel de tanques, depósitos, tolvas utilizados en la industria. Fomentará el uso de la adquisición de datos por computador mediante sistemas SCADA de los procesos de llenado, para realizar la verificación de la cota del líquido en el proceso de llenado.

El proyecto puede ser utilizado por pequeñas, medianas y grandes industrias dependiendo de los requerimientos que se obtengan, como son: altura, diámetro, volumen del tanque, tipo de líquido etc. El proyecto permite visualizar, calibrar, controlar y monitorear el proceso de llenado de tanques en el área industrial tomando el agua como líquido de referencia y el plástico como material del cual están hechos los tanques.

El sistema activa un sensor acústico acoplado al tanque que recepta la información del nivel del líquido y luego este es procesado por la tarjeta de adquisición de datos, con la ventaja de poder visualizar la información del nivel por un display LCD montado en la tarjeta de adquisición de datos, luego esos datos son enviados por interfaz serial al computador.

Esos datos son capturados por la computadora para procesarlos mediante el uso de funciones de lógica difusa las mismas que determinan el encendido y el apagado de la

bomba, luego esos datos, tanto la altura como la toma del tiempo, son guardados en la base de conocimientos (base de datos).

En el peor de los casos si el sensor llega a tener algún tipo de falla, el sistema automáticamente trabajará con su base de conocimiento, para el proceso de llenado.

Una alternativa que ayuda a las industrias cuando existe un mal manejo de recursos técnicos (hardware), es la opción de que el programa realiza un apagado de emergencia. Si el sensor o los recursos de hardware son ineficientes, el sistema se apagará automáticamente.

El programa también permite llevar un monitoreo o control del estado del sensor mostrando cuando está en buen estado o se encuentre en un mal funcionamiento, esto lo muestra mediante una gráfica lineal.

CAPITULO 2 ANALISIS DEL SISTEMA

2.1.- Análisis de requerimientos del Sistema

Para el análisis de la solución a este problema se pensó en la utilización de sensores de precisión como por ejemplo de presión, nivel o acústico para el registro del nivel del líquido; luego la señal analógica capturada por este sensor es recibida por una tarjeta de adquisición de datos con un microcontrolador 16f873, y convertida a forma digital para ser enviada a través de un cable serial a un computador.

Nuestra solución a nivel de hardware se desglosa en las siguientes partes:

- Sensores

- Tarjeta de adquisición de datos

- Medio de comunicación de datos

Sensores

El uso de sensores es indispensable para la captura de la señal analógica, la variedad de sensores en la actualidad nos permite poder elegir entre una gran gama desde los de precisión hasta los que se controlan por medio de un led.

Para este proyecto se podrá acoplar a cualquier sensor siempre y cuando se pueda medir la señal analógica de la medición del nivel del líquido en los tanques. Ver figura 2.1:



Figura 2.1 Tipo sensores

Tarjeta de adquisición de datos

El componente principal de la tarjeta, es un microcontrolador 16f873. Se recomienda el uso de este pic de propiedad de microchip por su variedad de aplicaciones, para este caso la más importante es la de funcionar como convertidor analógico-digital, que captura de la señal analógica censada y luego convertirla a formato digital,

permitiendo visualizar en un display LCD. La tarjeta nos sirve para el control de encendido y apagado de las bombas mediante un relay; la comunicación serie que tiene este pic, nos brinda la posibilidad de integrar lo obtenido y enviar a través de un cable serial al puerto respectivo del computador. Ver Figura 2.2:

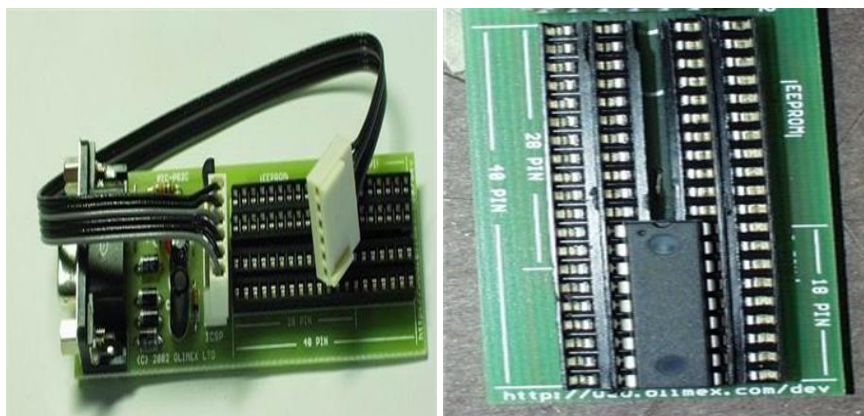


Figura 2.2 Tarjeta de adquisición de datos

Medio de Comunicación de Datos

Luego de que la señal es convertida a formato digital, un cable de tipo serial, es el medio por la cual se transmiten los datos hacia el puerto de un computador para poder ser interpretada por una interfaz de programación. Ver figura 2.3:



Figura 2.3 Cable serial

Nuestra solución a nivel de software se desglosa en las siguientes partes:

- Obtención de los datos
- Dibujar los datos
- Resultado final

Obtención de los datos

Luego de interpretar la señal por el puerto serial del computador, se registra el valor obtenido en una base de datos, para luego ser leído frecuentemente para graficar los datos.

Dibujar los datos

Capturada la señal por el puerto serial del computador, se dibujará en tiempos pre-establecidos cada centímetro de llenado para poder interpretar de manera gráfica el llenado del tanque.

Al recibir la señal de llenado del tanque, se reciben los datos y dibujamos cada escala por cada centímetro de llenado.

Cada escala que se dibuja, se lo realiza haciendo un tipo de sombreado en el tanque hasta llegar a su tope, luego de estar llenado completamente, obtenemos el tiempo de llenado y su altura específica.

Al desalojar el líquido por medio del tanque, tendremos un botón para ejecutar esa

opción, el cual se envía un pulso para activar una de las bombas, para que realice la tarea del desalojo de tanque.

Resultado final

Después de finalizar el llenado, el computador mostrará el gráfico con su respectivo llenado y su altura, mostrando de manera interactiva su proceso.

Componentes de software a utilizar:

Visual studio.net - C#.net “2005-2008”,Framework 3.5 para utilizar las librerías gráficas (Tecnología Wpf), librerías para el acceso a datos y comunicación con el serial por medio de la IO, y SQL server 2005 para el registro de los datos. Ver Figura 2.4:

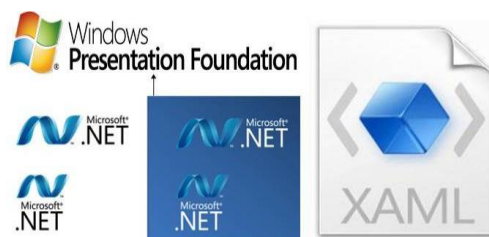


Figura 2.4 Tecnología wpf

2.2.- Características funcionales del sensor

El sensor seleccionado para este proyecto es de tipo ultrasónico MaxSonar de la casa Maxbotics. La información de la misma la podrá encontrar en la sección de referencia en: [1] Hoja de especificación de datos del sensor maxsonar, donde hallaremos toda la información detallada y técnica del sensor.

2.3.- Características del microcontrolador para la adquisición de datos

Para este proyecto se utiliza el pic16F873. Este microcontrolador forma parte de la familia de los pic16F87X. Revisar en [2] Hoja de especificación del PIC16F87XA. Ver figura 2.5:

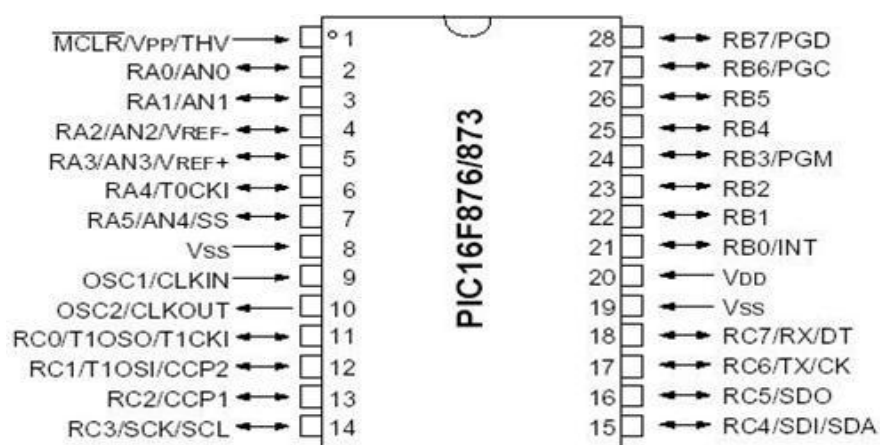


Figura 2.5 Microcontrolador

2.4 Modelamiento en 3D del tanque utilizando frameworks

Multimedia.

Para desarrollar el modelamiento del tanque en 3D utilizaremos el framework de .net “.net framework 3.5” que trae incluido en los paquetes de Visual Studio 2008.

.NET Framework. Es un marco de desarrollo que permite el desarrollo y ejecución de diversas aplicaciones hechas en los diversos lenguajes que soportan el .NET Framework, Es una Plataforma para construir, distribuir y ejecutar servicios web y aplicaciones. Sus principales componentes son:

- Common Language Runtime (CLR):
- Framework Class Libraries (FCL)

Common Language Runtime (CLR): es el que se encarga de la ejecución de las aplicaciones, así como de ofrecer un entorno de ejecución seguro para ellas. Ver link [14] información general.

Framework Class Libraries (FCL): son todas las clases que ofrece el .NET Framework, y las que pueden aprovechar desde los lenguajes soportados, usando un entorno de desarrollo (como Visual Studio). Ver link [13] información general.

Beneficios del .net framework:

- Está basado en estándares y practicas web.
- Extensible.
- Fácil de utilizar por los desarrolladores
- Es diseñado utilizando modelos de aplicaciones unificados.

Versiones del .NET Framework

- .Net framework 1.0
- .Net framework 1.1
- .Net framework 2.0
- .Net framework 3.0
- .Net framework 3.5
- Net framework 4.0

Tecnologías siguientes se introducen en .Net Framework 3.5:

- Windows presentation Foundation (WPF)
- Windows Communication (WCF)
- Windows Workflow Foundation (WF)

En las tecnologías mencionadas del .net framework 3.5 la que utiliza es la siguiente:

Windows Presentation Foundation (WPF). Es un sistema de presentación de la próxima generación, para crear aplicaciones cliente de Windows que proporcionen una experiencia impactante para el usuario desde el punto de vista visual. Con WPF, puede crear una amplia gama de aplicaciones independientes y hospedadas en explorador. En la siguiente figura se muestra un ejemplo de una aplicación WPF. Ver figura 2.6

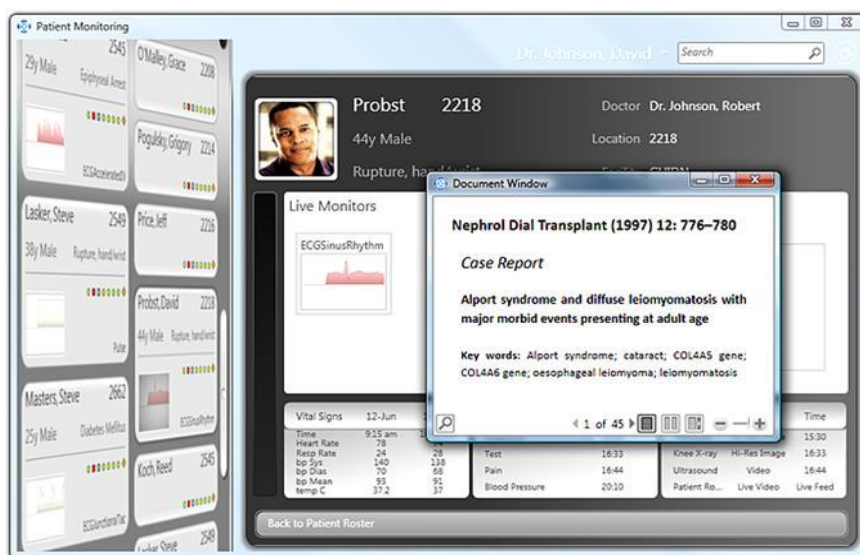


Figura 2.6 interfaz wpf

El núcleo de WPF es un motor de representación basado en vectores e independiente de la resolución que se crea para sacar partido del hardware de gráficos moderno. WPF extiende el núcleo con un conjunto completo de características de desarrollo de aplicaciones que incluye Lenguaje XAML, controles, enlace de datos, diseño, gráficos 2D y 3D, animación, estilos, plantillas, documentos, multimedia, texto y tipografía. WPF se incluye en Microsoft .NET Framework, de modo que es posible compilar aplicaciones que incorporen otros elementos de la biblioteca de clases de .NET Framework.

XAML: es un lenguaje “markup” (como HTML) para representar declarativamente la interfaz de una aplicación de Windows, mejorando la riqueza de las herramientas mediante las cuales developers y diseñadores gráficos pueden construir y modificar la interfaz de usuario. WPF utiliza XAML.

La ventaja específica que lleva a XAML de WPF XAML es que es una lengua completamente declarativa. En un lenguaje de programación declarativa, el desarrollador (o diseñador) describe el comportamiento y la integración de los componentes sin el uso de procedimiento de programación. Aunque es raro que una aplicación completa se construya totalmente en XAML, la introducción de XAML permite que los diseñadores de aplicaciones que contribuyan más eficazmente con el ciclo de desarrollo de aplicaciones. Con XAML para desarrollar interfaces de usuario también permite la separación de la modelo y la vista, que se considera un buen principio arquitectónico. En XAML, los elementos y atributos de mapa para las clases y propiedades en el API subyacente. Ver [15]

WPF tiene las siguientes características:

- Enlace de datos
- Gráficos
- Animación
- Multimedia
- Texto y tipografía
- Documentos
- Personalizar las aplicaciones WPF
- Procedimientos recomendados en WPF

Ver más en el link [3] características Wpf

2.5.-Análisis de funciones de membresía para el algoritmo de lógica difusa para el prendido y apagado de bombas para drenar y surtir líquidos.

La lógica difusa es una extensión de la lógica tradicional (Booleana) que utiliza conceptos de pertenencia de conjuntos más parecidos a la manera de pensar humana

El concepto de un subconjuntos difuso fue introducido por L.A. Zadeh en 1965 como una generalización de un subconjuntos exacto (crisp subset) tradicional. Los subconjuntos exactos usan lógica booleana con valores exactos como por ejemplo la lógica binaria que usa valores de 1 o 0 para sus operaciones.

La lógica difusa no usa valores exactos como 1 o 0 pero usa valores entre 1 y 0 (inclusive) que pueden indicar valores intermedios (Ej. 0, 0.1, 0.2,...,0.9,1.0, 1.1, ...etc.)

La lógica difusa también incluye los valores 0 y 1 entonces se puede considerar como un superconjuntos o extensión de la lógica exacta (ver figura 2.7).



Figura 2.7 Esquema de la lógica difusa

Funciones de membresía

Subconjunto difuso

Asumiendo que X es un conjunto, un subconjunto difuso A de X es asociado con una función característica: $\mu_A(x)$.

$$\mu_A(x): X \rightarrow [0, 1]$$

La función característica es típicamente denominada función de pertenencia (membership function).

Si X es una colección de objetos en el cual $x \in X$, un subconjunto difuso es un mapa $\mu_F(x) : X \rightarrow [0, a]$, en el cual a cada valor x la función $\mu_F(x)$ le asigna un número entre los valores 0 a a .

El subconjunto difuso es el conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

Ejemplos discretos y continuos:

$$A = \{(0, 0.1), (1, 0.5), (2, 1), (3, 0.1), (4, 0.8)\}$$

$$B = \{0.1/0, 0.5/1, 1/2, 0.1/3, 0.8/4\}$$

$$C = \{(x, \mu_C(x)) \mid x \in X\}, \mu_C(x) = 1 / (1 + (x/10 - 5))$$

$X = \{0, 1, 2, 3, 4, 5, 6, 7\}$ es el conjunto de hijos que puede tener una familia, entonces el conjunto difuso D es “el número razonable de hijos que puede tener una familia”

$$D = \{(0, 0.1), (1, 0.3), (2, 0.7), (3, 1), (4, 0.7), (5, 0.3), (6, 0.2), (7, 0.1)\}$$

Función de pertenencia

El valor asignado por $\mu_F(x)$ corresponde al grado en el cual el valor x tiene el atributo F . Visto de otra manera la función $\mu_F(x)$ nos indica cual es el grado de pertenencia de x a el atributo F .

La función $\mu_F(x)$ se llama la función de pertenencia del atributo F . La función tiene que ver con un grado de ambigüedad sobre la característica de la variable que se está midiendo pero no es una probabilidad.

Ej.: $\mu_F(x)$ corresponde al nivel de frío medido en la variable x (ver figura 2.8):

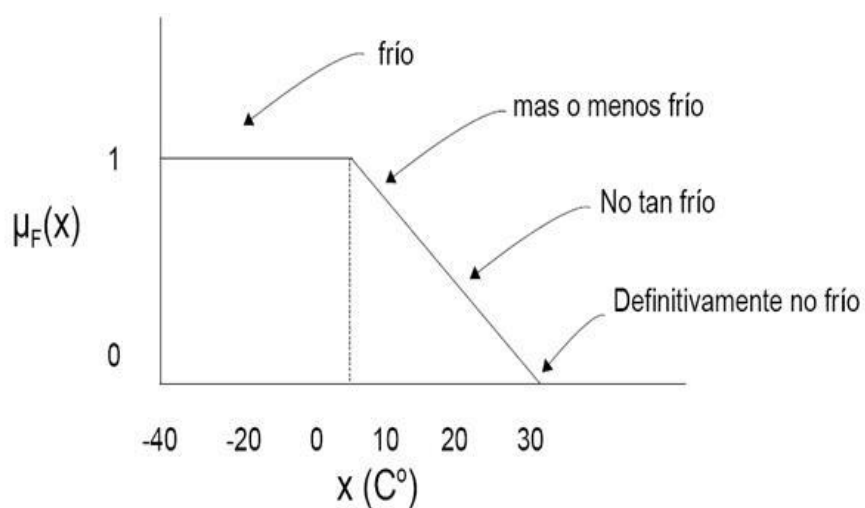


Figura 2.8 Función de pertenencia

Un conjunto exacto (ver figura 2.9-2.10)

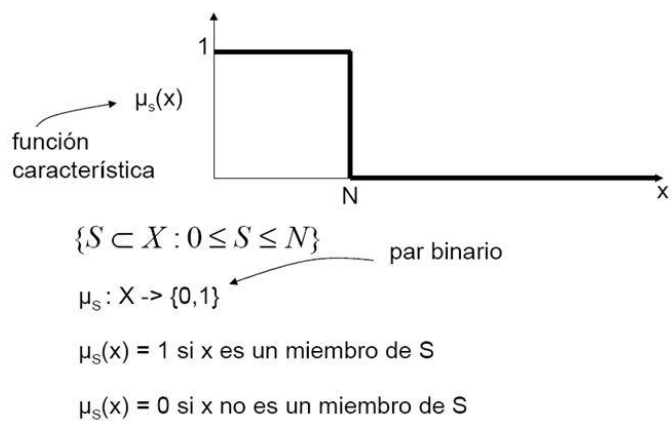


Figura 2.9 Función constante

Ejemplos de conjuntos difusos:

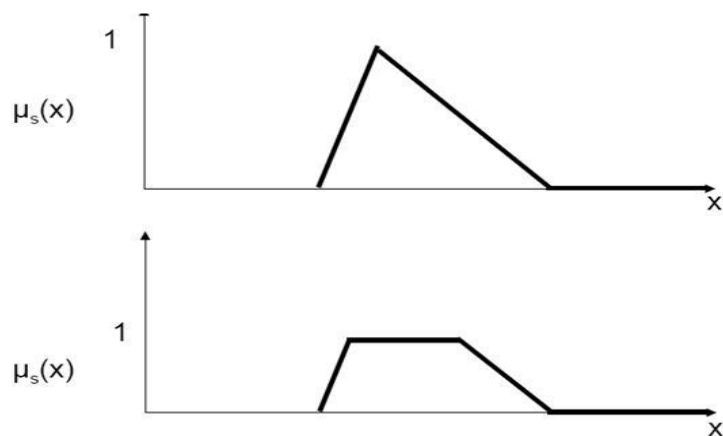


Figura 2.10 Funciones variables

Variables lingüísticas

Se usan variables lingüísticas para analizar y modelar un sistema bajo consideración:

Ej.: supongamos que $X = \text{“edad”}$, se pueden definir conjuntos difusos: “joven”,

“adulto”, “anciano” (ver figura 2.11)

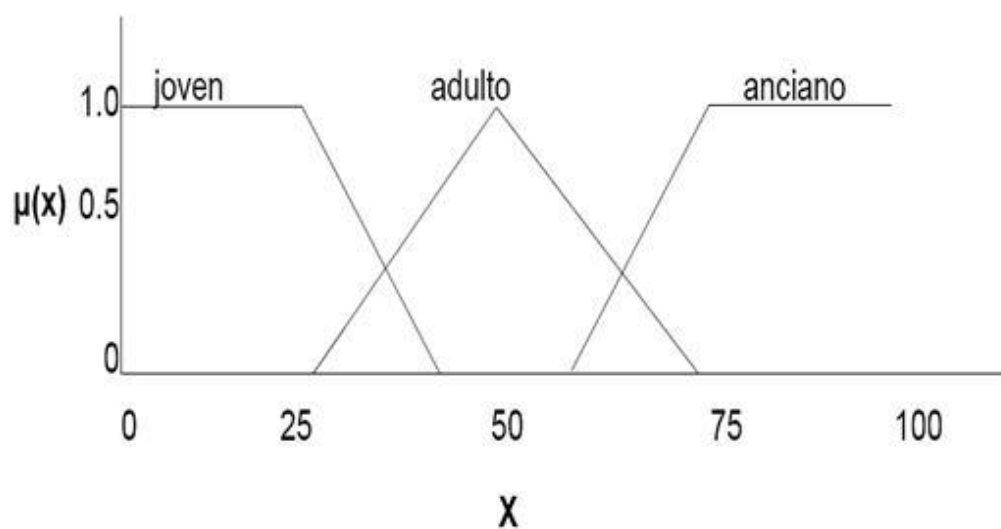


Figura 2.11 Variables lingüísticas

Operaciones en conjuntos difusos

Definición: asumiendo que A y B son dos subconjuntos difusos de X , la unión de A y B es un subconjunto difuso $C = A \cup B$, en el cual $C(x) = \text{Max} [A(x), B(x)]$

Definición: asumiendo que A y B son dos subconjuntos difusos de X , la intersección de A y B es un subconjunto difuso $C = A \cap B$ en el cual $C(x) = \text{Min} [A(x), B(x)]$

Definición: El complemento relativo de B con respecto a A es $E = A - B$ en el cual $E(x) = \text{Max} [0, A(x) - B(x)]$

Funciones de pertenencia

Función de pertenencia trapezoidal (ver figura 2.12).

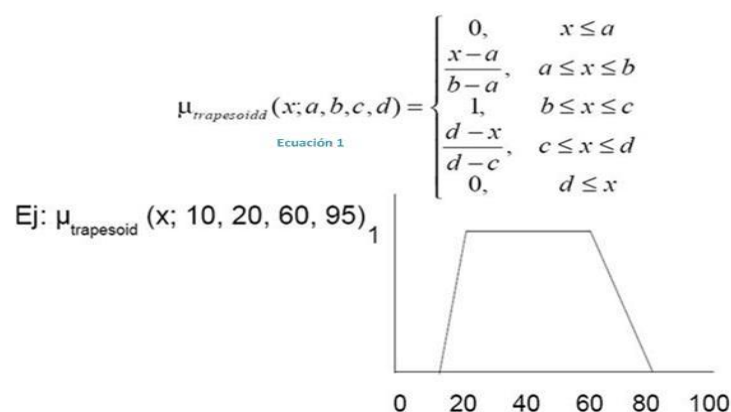


Figura 2.12 Modelo trapezoide

Funcionamiento

La lógica difusa se adapta mejor al mundo real en el que vivimos, e incluso puede comprender y funcionar con nuestras expresiones, del tipo "hace mucho calor", "no es muy alto", "el ritmo del corazón está un poco acelerado", etc.

La clave de esta adaptación al lenguaje, se basa en comprender los cuantificadores de nuestro lenguaje (en los ejemplos de arriba "mucho", "muy" y "un poco").

En la teoría de conjuntos difusos se definen también las operaciones de unión, intersección, diferencia, negación o complemento, y otras operaciones sobre conjuntos (ver también subconjunto difuso), en los que se basa esta lógica.

Para cada conjunto difuso, existe asociada una función de pertenencia para sus elementos, que indican en qué medida el elemento forma parte de ese conjunto difuso. Las formas de las funciones de pertenencia más típicas son trapezoidales, lineales y curvas.

Se basa en reglas heurísticas de la forma SI (antecedente) ENTONCES (consecuente), donde el antecedente y el consecuente son también conjuntos difusos, ya sea puros o resultado de operar con ellos. Sirvan como ejemplos de regla heurística para esta lógica (nótese la importancia de las palabras "muchísimo", "drásticamente", "un poco" y "levemente" para la lógica difusa):

SI hace muchísimo calor ENTONCES disminuyo drásticamente la temperatura. SI voy a llegar un poco tarde ENTONCES aumento levemente la velocidad.

Los métodos de inferencia para esta base de reglas deben ser simples, veloces y eficaces. Los resultados de dichos métodos son un área final, fruto de un conjunto de áreas solapadas entre sí (cada área es resultado de una regla de inferencia). Para escoger una salida concreta a partir de tanta premisa difusa, el método más usado es el del centroide, en el que la salida final será el centro de gravedad del área total resultante.

Las reglas de las que dispone el motor de inferencia de un sistema difuso pueden ser formuladas por expertos, o bien aprendidas por el propio sistema, haciendo uso en este caso de redes neuronales para fortalecer las futuras tomas de decisiones.

Los datos de entrada suelen ser recogidos por sensores, que miden las variables de entrada de un sistema. El motor de inferencias se basa en chips difusos, que están aumentando exponencialmente su capacidad de procesamiento de reglas año a año.

Un esquema de funcionamiento típico para un sistema difuso podría ser de la siguiente manera. Ver figura 2.13



Figura2.13 Entorno físico de sistema de control

En la figura 2.13, el sistema de control hace los cálculos con base en sus reglas heurísticas, comentadas anteriormente. La salida final actuaría sobre el entorno físico, y los valores sobre el entorno físico de las nuevas entradas (modificado por la salida del sistema de control) serían tomados por sensores del sistema.

Por ejemplo, imaginando que nuestro sistema difuso fuese el climatizador de un coche que se autor regula según las necesidades: Los chips difusos del climatizador recogen los datos de entrada, que en este caso bien podrían ser la temperatura y humedad simplemente. Estos datos se someten a las reglas del motor de inferencia (como se ha comentado antes, de la forma SI... ENTONCES...), resultando un área de resultados. De esa área se escogerá el centro de gravedad, proporcionándola como salida. Dependiendo del resultado, el climatizador podría aumentar la temperatura o disminuirla dependiendo del grado de la salida.

Aplicaciones generales

La lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos, para procesos altamente no lineales y cuando se envuelven definiciones y conocimiento no estrictamente definido (impreciso o subjetivo).

En cambio, no es una buena idea usarla cuando algún modelo matemático ya soluciona eficientemente el problema, cuando los problemas son lineales o cuando no tienen solución.

Esta técnica se ha empleado con bastante éxito en la industria, principalmente en Japón, y cada vez se está usando en gran multitud de campos. La primera vez que se usó de forma importante fue en el metro japonés, con excelentes resultados. A continuación se citan algunos ejemplos de su aplicación:

- Sistemas de control de acondicionadores de aire
- Sistemas de foco automático en cámaras fotográficas
- Electrodomésticos familiares (frigoríficos, lavadoras...)
- Optimización de sistemas de control industriales
- Sistemas de reconocimiento de escritura
- Mejora en la eficiencia del uso de combustible en motores

- Sistemas expertos del conocimiento (simular el comportamiento de un experto humano)
- Tecnología informática
- En general, en la gran mayoría de los sistemas de control que no dependen de un Sí/No.

Lógica difusa en inteligencia artificial

En inteligencia artificial, la lógica difusa, o lógica borrosa se utiliza para la resolución de una variedad de problemas, principalmente los relacionados con control de procesos industriales complejos y sistemas de decisión en general, la resolución la comprensión de datos. Los sistemas de lógica difusa están también muy extendidos en la tecnología cotidiana, por ejemplo en cámaras digitales, sistemas de aire acondicionado, lavarropas, etc. Los sistemas basados en lógica difusa imitan la forma en que toman decisiones los humanos, con la ventaja de ser mucho más rápidos. Estos sistemas son generalmente robustos y tolerantes a imprecisiones y ruidos en los datos de entrada. Algunos lenguajes de programación lógica que han incorporado la lógica difusa serían por ejemplo las diversas implementaciones de Fuzzy PROLOG.

Consiste en la aplicación de la lógica difusa con la intención de imitar el razonamiento humano en la programación de computadoras. Con la lógica convencional, las computadoras pueden manipular valores estrictamente duales, como verdadero/falso, sí/no o ligado/desligado. En la lógica difusa, se usan modelos matemáticos para

representar nociones subjetivas, como caliente/tibio/frío, para valores concretos que puedan ser manipuladas por los ordenadores.

En este paradigma, también tiene un especial valor la variable del tiempo, ya que los sistemas de control pueden necesitar retroalimentarse en un espacio concreto de tiempo, pueden necesitarse datos anteriores para hacer una evaluación media de la situación en un período anterior.

Ventajas e inconvenientes de la lógica difusa

Como principal ventaja, cabe destacar los excelentes resultados que brinda un sistema de control basado en lógica difusa: ofrece salidas de una forma veloz y precisa, disminuyendo así las transiciones de estados fundamentales en el entorno físico que controle. Por ejemplo, si el aire acondicionado se encendiese al llegar a la temperatura de 30°, y la temperatura actual oscilase entre los 29°-30°, nuestro sistema de aire acondicionado estaría encendiéndose y apagándose continuamente, con el gasto energético que ello conllevaría. Si estuviese regulado por lógica difusa, esos 30° no serían ningún umbral, y el sistema de control aprendería a mantener una temperatura estable sin continuos apagados y encendidos.

También está la indecisión de decantarse bien por los expertos o bien por la tecnología (principalmente mediante redes neuronales) para reforzar las reglas heurísticas iniciales de cualquier sistema de control basado en este tipo de lógica.

2.6. Protocolos de comunicación entre el PC y el microcontrolador

Comunicaciones seriales

El puerto serial de las computadoras es conocido como puerto RS-232, la ventaja de este puerto es que todas las computadoras traen al menos un puerto serial, este permite la comunicaciones entre otros dispositivos tales como otra computadora, el mouse, impresora y para nuestro caso con los microcontroladores.

Existen dos formas de intercambiar información binaria: la paralela y la serial.

La comunicación paralela transmite todos los bits de un dato de manera simultánea, por lo tanto la velocidad de transferencia es rápida, sin embargo tiene la desventaja de utilizar una gran cantidad de líneas, por lo tanto se vuelve mas costoso y tiene las desventaja de atenuarse a grandes distancias, por la capacitancia entre conductores así como sus parámetros distribuidos.

Tipos de comunicaciones seriales:

Existen dos tipos de comunicaciones seriales: la síncrona y asíncrona.

En la comunicación serial sincronía además de una línea sobre la cual se transmitirán los datos se necesita de una línea la cual contendrá los pulsos de reloj que indicarán cuando un datos es valido. Ejemplos de este tipo de comunicación son:

- I2C
- ONE WIRE
- SPI

En la comunicación serial asíncrona, no son necesarios los pulsos de reloj. La duración de cada bit esta determinada por la velocidad con la cual se realiza la transferencia

dedatos. A continuación mostramos la estructura de un carácter que se transmite en forma serial asíncrona. Ver figura 2.14:

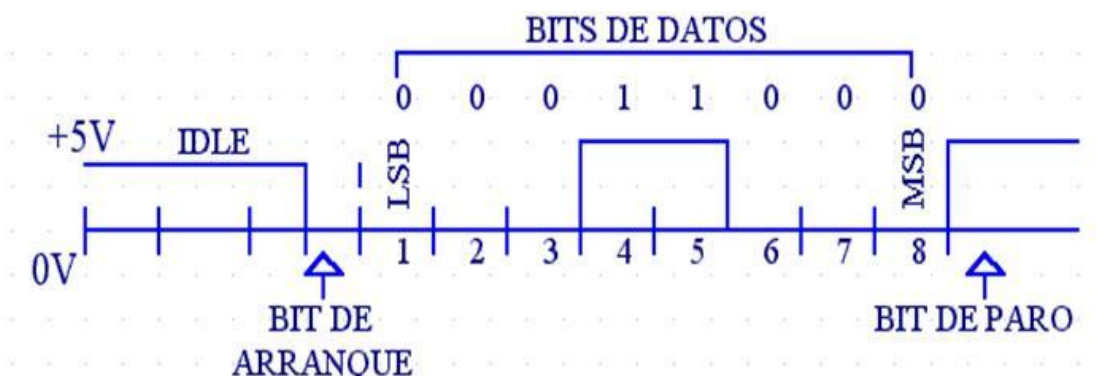


Figura 2.14 Trama de datos

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en estado de (idle) esto quiere decir en estado alto.

Para iniciar la transmisión de datos, el transmisor coloca esta línea en bajo durante determinado tiempo, lo cual se le conoce como bit de arranque (start bit) y a continuación empieza a transmitir con un intervalo de tiempo los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB), y terminando con el BIT más significativo. Si el receptor no está sincronizado con el transmisor, este desconocerá cuando se van a recibir los datos.

Por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y BIT de parada.

En los circuitos digitales, cuyas distancias son relativamente cortas, se pueden manejar transmisiones en niveles lógicos TTL (0-5V), pero cuando las distancias aumentan, estas señales tienden a distorsionarse debido al efecto capacitivo de los conductores y a su resistencia eléctrica. El efecto se incrementa a medida que se incrementa la velocidad de

la transmisión. Todo esto origina que los datos recibidos no sean igual a los datos transmitidos, por lo que no se puede permitir la transferencia de datos.

Una de las soluciones más lógicas es aumentar los márgenes de voltaje con que se transmiten los datos, de tal manera que las perturbaciones a causa de la línea se pueden corregir.

Protocolo RS232

Ante la gran variedad de equipos, sistemas y protocolos que existen surgió la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre sí. La EIA (Electronics Industry Association) elaboró la norma RS-232, la cual define la interfase mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial

Todas las normas RS-232 cumplen con los siguientes niveles de voltaje:- Un "1" lógico es un voltaje comprendido entre -5v y -15v en el transmisor y entre -3v y -25v en el receptor.

Un "0" lógico es un voltaje comprendido entre $+5\text{v}$ y $+15\text{v}$ en el transmisor y entre $+3\text{v}$ y $+25\text{v}$ en el receptor.

El envío de niveles lógicos (bits) a través de cables o líneas de transmisión necesita la conversión a voltajes apropiados. En los microcontroladores para representar un 0 lógico se trabaja con voltajes inferiores a 0.8v , y para un 1 lógico con voltajes mayores a 2.0V .

En general cuando se trabaja con familias TTL y CMOS se asume que un “0” lógico es igual a cero Volts y un “1” lógico es igual a cinco Volts.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el puerto serial del ordenador, ya que son diferentes a los que utilizan los microcontroladores y los demás circuitos integrados. Por lo tanto se necesita de una interfase que haga posible la conversión del niveles de voltaje a los estándares manejados por los CI TTL.

El Circuito MAX-232

Este circuito soluciona los problemas de niveles de voltaje cuando se requiere enviar unas señales digitales sobre una línea RS-232.

Este chip se utiliza en aquellas aplicaciones donde no se dispone de fuentes dobles de +12 y -12 Volts. El MAX 232 necesita solamente una fuente de +5V para su operación, internamente tiene un elevador de voltaje que convierte el voltaje de +5V al de doble polaridad de +12V y -12V. Cabe mencionar que existen una gran variedad de CI que cumplen con la norma RS-232 como lo son: MAX220, DS14C232, MAX233, LT1180A. Ver figura 2.15:

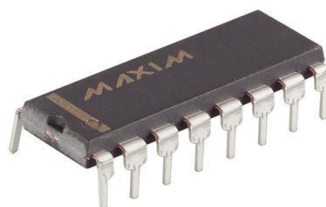


Figura 2.15 Max rs232

2.7. Especificaciones electrónicas para la seguridad entre el mundo físico y el hardware

La principal fuente de energía para la preparación de dispositivos y equipos electrónicos es la energía eléctrica, cuya frecuente utilización y difícil detección por los sentidos provoca cierta falta de previsión y, por consiguiente, los accidentes eléctricos.

El riesgo eléctrico, es aquel originado por la energía eléctrica, quedando específicamente incluidos los riesgos de:

- ✓ Choque eléctrico por contactos con elementos en tensión (contacto eléctrico directo) o con masas puestas accidentalmente en tensión (contacto eléctrico indirecto).
- ✓ Quemaduras por choque eléctrico o por arco eléctrico.
- ✓ Caídas o golpes como consecuencia de choque o arco eléctrico.
- ✓ Incendios o explosiones originados por la electricidad.

Aunque los accidentes eléctricos no son muy frecuentes, revisten una elevada gravedad como resultado de la electrización o, en su expresión más severa, de la electrocución.

[17]

Como en toda actividad, en el trabajo eléctrico, se debe tener precauciones y reducir los riesgos a "0". Cuando la electricidad se maneja inteligentemente, es segura. A continuación se muestra tips de seguridad para la elaboración de un proyecto como el que presentamos:

- ✓ Se debe de usar ropa adecuada para este trabajo.
- ✓ NO usar en el cuerpo piezas de metal, ejemplo, cadenas, relojes, anillos, etc. ya que podrían ocasionar un corto circuito.

- ✓ Cuando se trabaja cerca de partes con corriente o maquinaria, usar ropa ajustada y zapatos antideslizantes.
- ✓ De preferencia, trabajar sin energía.
- ✓ Al trabajar en líneas de alta tensión, aunque se haya desconectado el circuito, se debe de conectar (el electricista) a tierra con un buen conductor.
- ✓ Es conveniente trabajar con guantes adecuados cuando se trabaja cerca de líneas de alto voltaje y proteger los cables con un material aislante.
- ✓ Si no se tiene la seguridad del voltaje, o si esta desactivado, no correr riesgos.
- ✓ Deberan abrirse los interruptores completamente, no a la mitad y no cerrarlos hasta estar seguro de las condiciones del circuito.
- ✓ Si se desconoce el circuito o si es una conexión complicada, familiarizarse primero y que todo este correcto. hacer un diagrama del circuito y estudiarlo detenidamente, si hay otra persona, pedirle que verifique las conexiones o bien el diagrama.
- ✓ Hacer uso de herramientas adecuadas (barras aisladoras) para el manejo de interruptores de alta potencia.[18]

CAPITULO 3 DISEÑO

3.1 Diseño de la arquitectura

Para el diseño de la arquitectura de nuestro proyecto se basa en tres grandes partes. Ver figura 3.1:

- Parámetros de entrada globales
- Diseño en hardware
- Diseño de software



Figura 3.1 Esquema de la arquitectura general

3.1.1 Parámetros de entrada globales

Los parámetros de entrada para el diseño del proyecto se centran en los estímulos externos de los cuales, el sensor, captura el nivel del tanque, para luego ser transmitidos a la tarjeta de adquisición cuyo componente principal es un microcontrolador. Ver figura 3.2:

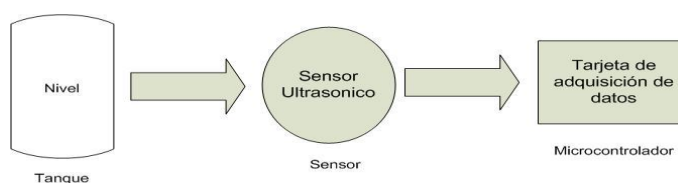


Figura 3.2 Parámetros de entradas globales

3.1.2 Diseño de hardware

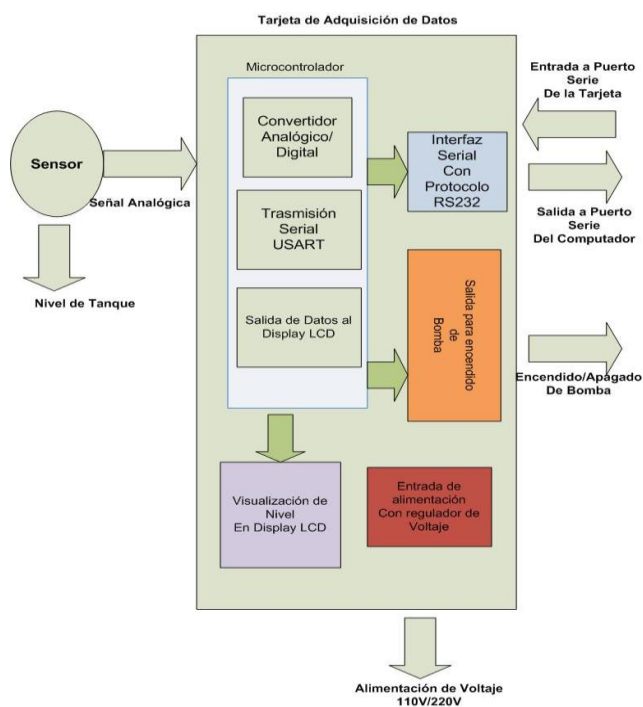


Figura 3.3 Diseño de hardware

El diseño del hardware lo componen específicamente el sensor y la tarjeta de adquisición de datos.

Dentro de la tarjeta de adquisición de datos como se observa en la figura 3.3 se ha dividido en bloques específicos para el diseño como es la parte del microcontrolador la cual consta de:

- Convertidor analógico digital
- Transmisión serial con USART
- Salida de datos al display LCD

También la interfaz serial para comunicación con el PC a través de la norma RS232, una salida para conectar una bomba, la salida de visualización a un display y la etapa de regulación de voltaje necesaria para trabajar a niveles de 5Vcc que es lo ideal para los componentes electrónicos.

3.1.3 Diseño de software

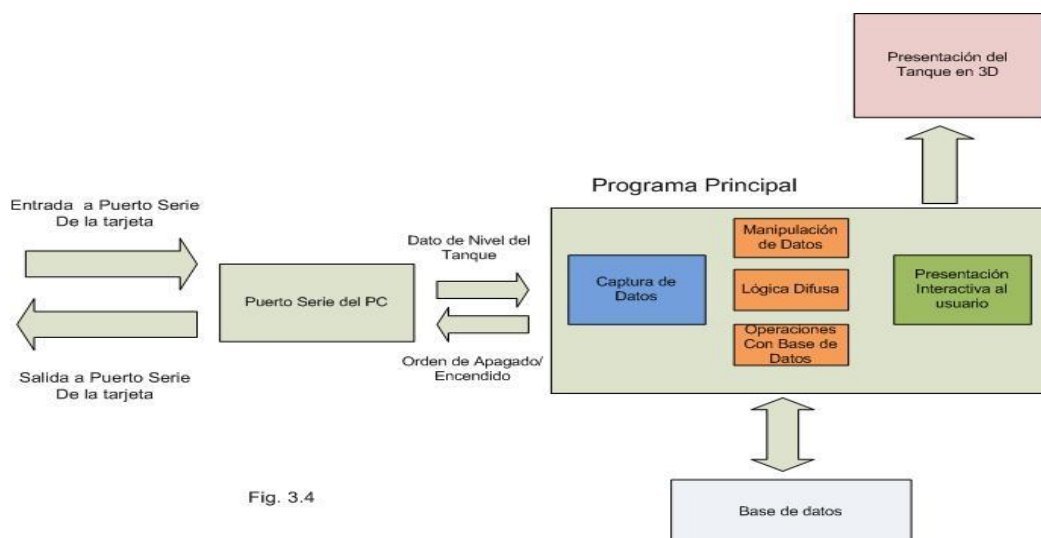


Fig. 3.4

Figura 3.4 Diseño de software

En la Figura 3.4 se muestra como está estructurado todo el diseño de la etapa de software en el proyecto. Se muestra la interacción con la entrada de datos, la manipulación y sus operaciones con la base de datos y su respectiva salida en 3D.

3.2 Especificaciones de diseño del sensor en la tarjeta AD

Una ráfaga de 40KHz es transmitida a través del sensor ultrasónico, el ultrasonido reflejado es recibido por el mismo. El cálculo de la distancia se hace con un PIC16F873A. El rango de distancia medido es de 15cm a 300cm



Figura 3.5 Sensor max sonar

Para una medición adecuada, sobre todo en distancias grandes se deben de cumplir las siguientes condiciones:

- El objeto debe estar perpendicular al medidor.
- La superficie del objeto debe ser plana.
- No debe haber objetos alrededor que puedan hacer una reflexión.
- El objeto no debe ser muy absorbente, como por ejemplo tela o una pared corrugada.

3.3 Características del sensor ultrasónico maxzonar

Tabla 1 Características del max sonar

Características	MaxSonar
Frecuencia de operación	38KHz a 42KHz
Nivel de presión sonora	117dB (0dB=0.2n bar)
Sensibilidad	-60dB (0dB 1V/u bar)
Máxima entrada de voltaje	20V RMS máximo (60V pico a pico máximo @ 10%)
Rango de temperatura	-40C hasta +65C (-40F hasta +150F)
Impedancia	1K Ohm
Capacitancia	2400 pF (+/- 20%)
Señal	menos de 1ms

Ultrasonido

Ultrasonido hace referencia a las frecuencias arriba de 20KHz (limite de sonido audible).

Altas frecuencias tienen longitudes de onda cortas lo que hace al reflejarse en objetos esta pueda ser leída. Desafortunadamente frecuencias muy altas son difíciles de generar y leer.

La generación y lectura de ultrasonido se hace a través de dos unidades piezoeléctricas en donde una de ellas es el emisor y la otra el receptor de ondas de presión ultrasónicas, en

nuestro caso utilizaremos el mismo para ambas funciones coordinando los tiempos. Para esto, la unidad emisora debe excitarse con una señal adecuada en amplitud y frecuencia. La unidad receptora translucirá todas aquellas ondas de presión ultrasónicas de 40KHz que lleguen a excitarla.[19]

El ultrasonido es aplicado comúnmente en detectores de movimiento, medidores de distancia, diagnóstico médico, limpieza, pruebas no destructivas (para detectar imperfecciones en materiales), soldadura entre otras más.

250mS después de conectar la alimentación el MaxSonar-EZ1 está listo para aceptar la señal RX. Si ésta está a nivel "1" se ejecuta un ciclo de calibración que dura 49mS y seguidamente comienzan los ciclos de trabajo. Estos consisten en realizar una medida y transmitir la distancia por los tres métodos disponibles: salida serie por TX, salida analógica por AN y salida de pulso por PW. Cada ciclo de trabajo consume 49mS, por lo que el primer ciclo válido tras conectar la alimentación se produce al de 100mS.

Cada ciclo de trabajo de 49mS comienza chequeando el estado de RX. Si está a "0" se da por finalizado ese ciclo. A continuación se transmite una señal ultrasónica de 42KHz y la señal PW se pone a "1". Cuando se detecta un obstáculo PW se pone a "0". Si el tiempo en que esta señal permanece a "1" es superior a 37.5mS significa que no se ha detectado obstáculo alguno. Durante los siguientes 4.7mS se transmite en serie el valor de la medida realizada. El resto del tiempo hasta llegar a los 49mS que dura el ciclo se emplea para ajustar la tensión analógica en AN a su nivel apropiado. Ver figura 3.6:

DISEÑO PARA EL SENSOR



Figura 3.6 Diseño para el sensor

Como la salida del sensor es analógica entre 0-2.55V, realizando hasta 20 medidas por segundo es decir una medición cada 50ms, según la especificación del fabricante se utilizara la siguiente formula de conversión:

$$\text{Voltaje} = 10 \frac{\text{mV}}{\text{pulgada}} \cdot * \text{Dist}$$

Ecuación 2

Por ejemplo una lectura de 1V en la salida analógica nos dará como resultado si aplicamos la ecuación 2:

$$1V = 10 \frac{mV}{pulgada} * Dist$$

$$Dist = \frac{1V}{10mV} * pulgada$$

$$Dist = 100pulgadas$$

$$1 Pulgada = 2.54cm$$

$$Dist = 254cm$$

3.4 Acondicionador de señal y Modelo lineal con amplificadores operacionales

Debido a que la señal analógica que emite el sensor opera en el rango de permitido del microcontrolador 16F873A, se descarta la creación de una etapa amplificadora; en el siguiente capítulo se detallará con más exactitud las características del PIC16F873A y su diseño en la tarjeta de adquisición de datos.

En la siguiente figura se muestra el diseño de un sensor que utiliza el simulador de componentes electrónicos Proteus, el cual se muestra en detalle más adelante. Aquí se observa como una señal analógica puede ser conectada a través de una de las entradas del microcontrolador. Ver figura 3.7:

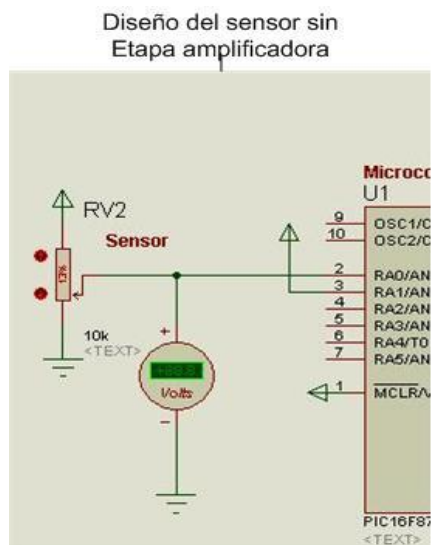


Figura 3.7 Diseño sin etapa amplificador

Para simular un sensor con salidas y tensión analógica deseada, se diseñó un potenciómetro con salida deseada de voltaje. Es decir que su salida de voltaje esté entre 0V-2.55V, utilizando una resistencia de 10K y un voltaje de alimentación regulado a 5V se pueden alcanzar los valores requeridos para la simulación. Ver figura 3.8:

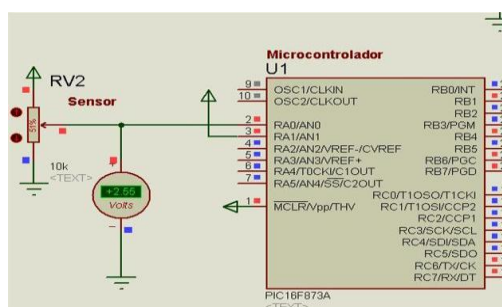


Figura 3.8 Diseño funcional del sensor

3.5 Especificaciones de diseño usando MICRO 16F873A

Como ya hemos mencionado para este proyecto el diseño que vamos a usar es para el pic16F873A a continuación en la figura vamos a mostrar el esquema completo del encapsulado del mismo con todas sus entradas salidas

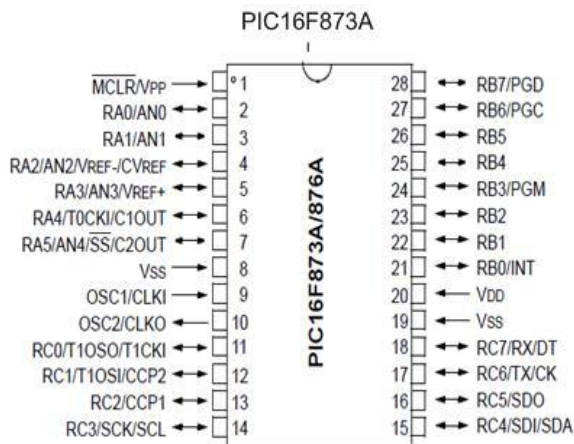


Figura3.9 Especificaciones del microcontrolador

Las características la resumiremos en la siguiente tabla:

Tabla 2 Características principales del microcontrolador

Características Principales
16F873A

Key Features	PIC16F873A
Operating Frequency	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K
Data Memory (bytes)	192
EEPROM Data Memory (bytes)	128
Interrupts	14
I/O Ports	Ports A, B, C
Timers	3
Capture/Compare/PWM modules	2
Serial Communications	MSSP, USART
Parallel Communications	—
10-bit Analog-to-Digital Module	5 input channels
Analog Comparators	2
Instruction Set	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN

En el simulador proteus, también contamos con un modelo encapsulado para el diseño de la tarjeta de adquisición de datos

3.5.1 Entradas analógicas

La patita RA0/AN0 actúa como pin de entrada de la señal analógica capturada por el sensor la cual luego va ser procesada por el conversor analógico digital y por los pines OSC1/CLKIN y OSC2/CLKOUT hemos colocado un circuito de reloj externo acoplado un cristal de cuarzo con frecuencia de 4MHZ y para estabilizar la frecuencia 2 capacitores de 15 picofaradios. Ver figura 3.10:

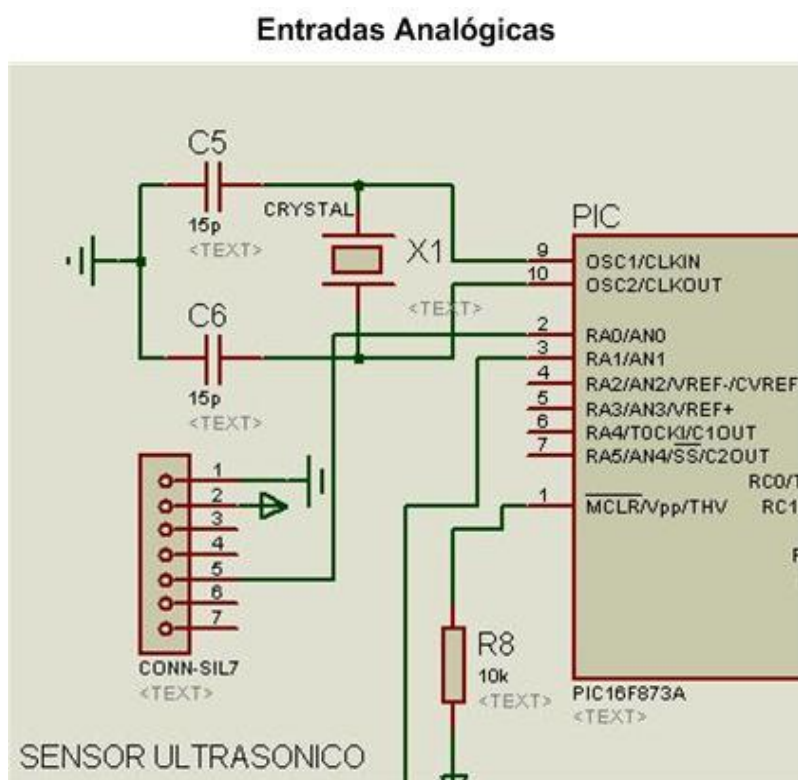


Figura 3.10 Entradas analógica del pic

3.5.2 Convertidor analógico digital

Los microcontroladores PIC16f873A poseen un conversor analógico digital de 10 bits de resolución y 5 canales de entrada. La resolución que tiene cada bit procedente de la conversión tiene un valor que es función de la tensión referencia V_{ref} , de acuerdo a la siguiente fórmula

$$resolucion = \frac{(V_{ref+} - V_{ref-})}{1024} = V_{ref} / 1024$$

Ecuación 3

Así por ejemplo, si la $V_{ref} = 5VDC$ y la V_{ref-} es tierra, la resolución es de 4.8mV/bit. Por tanto una entrada analógica de 0V le corresponde una digital de 00 0000 0000 y para 5V una de 11 1111 1111. La tensión de referencia determina los límites máximo y mínimo de la tensión analógica que puede convertir. El voltaje diferencial mínimo es de 2V.

A través del canal de entrada seleccionado, se aplica la señal analógica a un condensador de captura y mantenimiento y luego se introduce al conversor, el cual proporciona un resultado digital de 10 bits de longitud cuando la técnica de aproximaciones sucesivas.

Para las especificaciones del sensor maxsonar cuya salida analógica del sensor es 0a 2.55V tomando en cuenta los 10 bits de resolución se obtiene que para 0V, una salida digital de 00 0000 0000 y para los 2.55 V sería 011 1111 1111..

3.5.3 Salida al LCD Hitachi

Las pantallas de cristal líquido LCD o display LCD para mensajes (Liquid Cristal Display) tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica. La pantalla consta de una matriz de caracteres (normalmente de 5x7 o 5x8 puntos) distribuidos en una, dos, tres o cuatro líneas de 16 hasta 40 caracteres cada línea. El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado. Ver figura 3.11:

LCD HITACHI 16x2



Figura 3.11 LCD hitachi

A continuación se presenta la descripción de señales empleadas por el módulo LCD así como el número de patilla a la que corresponden. Además del respectivo diseño final en el simulador previo a la implementación. Ver figura 3.12 y la tabla 3:

Tabla 3 Descripción de pines del LCD

Descripción de Pines de LCD hitachi

PIN N°	SÍMBOLO	DESCRIPCIÓN
1	V _{SS}	Patilla de tierra de alimentación
2	V _{DD}	Patilla de alimentación de 5 V
3	V _O	Patilla de contraste del cristal líquido. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y +5V que permite regular el contraste del cristal líquido.
4	RS	Selección del registro de control/registro de datos: RS=0 Selección del registro de control RS=1 Selección del registro de datos
5	R/W	Señal de lectura/escritura R/W=0 El módulo LCD es escrito R/W=1 El módulo LCD es leído
6	E	Señal de activación del módulo LCD: E=0 Módulo desconectado E=1 Módulo conectado
7-14	D0-D7	Bus de datos bi-direccional. A través de estas líneas se realiza la transferencia de información entre el módulo LCD y el sistema informático que lo gestiona

Salida a LCD hitachi

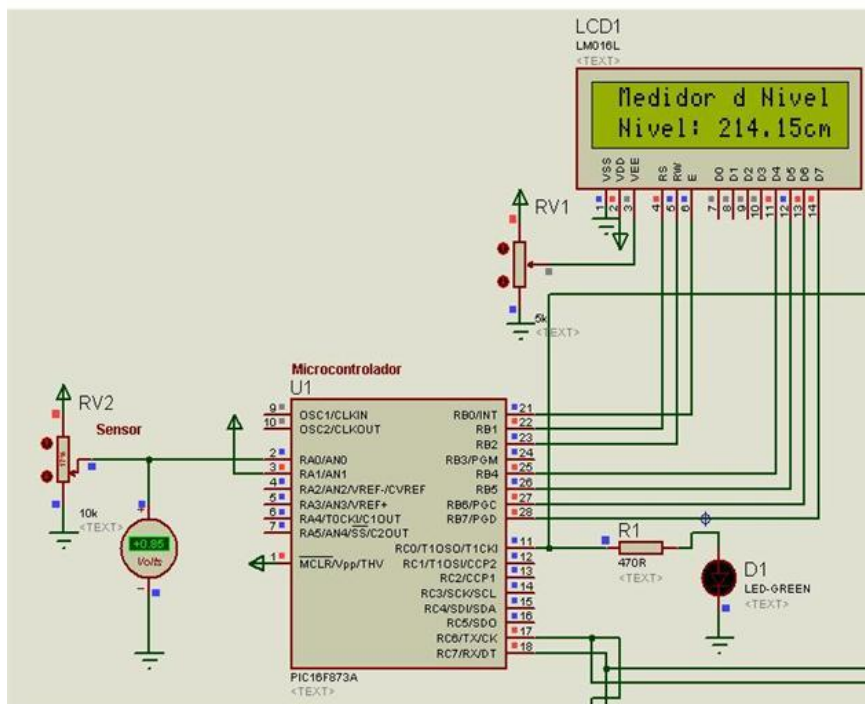


Figura 3.12 Conexiones del LCD al pic

3.5.4 Comunicación serial mediante USART y norma RS232

El módulo transmisor (USART) es otro de los dos puertos serie de los que dispone esta familia de microcontroladores.

Los módulos de I/O. (USART también se conocen como Serial Communications o Interface o SCI). El USART puede configurarse como asíncrono full dúplex que puede comunicar con los dispositivos periféricos como el terminales de CRT y los ordenadores personales, o puede configurarse como un sistema síncrono half duplex que puede comunicar con otros microcontroladores, con dispositivos periféricos como A/D o D/A circuitos integrados, EEPROMs serie etc.

El USART puede configurarse en los modos siguientes:

- Asíncrono (Full duplex)
- Síncrono-Master (half duplex)
- Síncrono-Slave (half duplex)

Bit de inicio y bit de parada

En la transmisión asíncrona un carácter a transmitir es encuadrado con un indicador de inicio y fin de carácter, de la misma forma que se separa una palabra con una letra mayúscula y un espacio en una oración. La forma estándar de encuadrar un carácter es a través de un bit de inicio y un bit de parada.

Durante el intervalo de tiempo en que no son transferidos caracteres, el canal debe poseer un "1" lógico. Al bit de parada se le asigna también un "1". Al bit de inicio del carácter a transmitir se le asigna un "0". Por todo lo anterior, un cambio de nivel de "1" a "0" lógico le indicará al receptor que un nuevo carácter será transmitido. Ver figura 3.13:

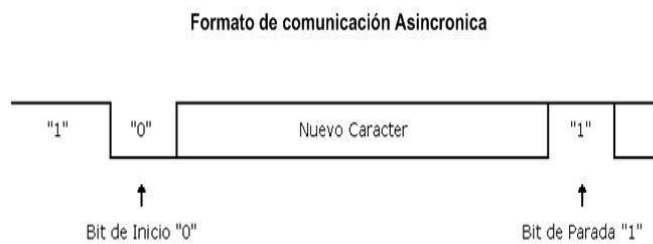


Figura 3.13 Formato de comunicación asincrónica

Reglas de transmisión asíncrona

La transmisión asíncrona que vamos a ver es la definida por la norma RS232, en la que profundizaremos más adelante y que se basa en las siguientes reglas:

1. Cuando no se envían datos por la línea, ésta se mantiene en estado alto (1).
2. Cuando se desea transmitir un carácter, se envía primero un bit de inicio que pone la línea a estado bajo (0) durante el tiempo de un bit.
3. Durante la transmisión, si la línea está a nivel bajo, se envía un 0 y si está a nivel alto se envía un 1.
4. A continuación se envían todos los bits del mensaje a transmitir con los intervalos que marca el reloj de transmisión. Por convenio se transmiten entre 5 y 8 bits.

5. Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse.
6. A continuación del último bit del mensaje se envía el bit (o los bits) del final que hace que la línea se ponga a 1 por lo menos durante el tiempo mínimo de un bit. Estos bits pueden ser un bit de paridad para detectar errores y el bit o bits de stop, que indican el fin de la transmisión de un carácter.

Los datos codificados por esta regla, pueden ser recibidos siguiendo los pasos siguientes:

1. Esperar la transición 1 a 0 en la señal recibida.
2. Activar el reloj con una frecuencia igual a la del transmisor.
3. Muestrear la señal recibida al ritmo de ese reloj para formar el mensaje.
4. Leer un bit más de la línea y comprobar si es 1 para confirmar que no ha habido error en la sincronización.

Generador de baudios

En el protocolo asíncrono RS-232 la frecuencia en baudios a la que se realiza la transferencia se debe efectuar a un valor normalizado: 330,600, 1.200, 2400, 4800, 9600, 19.200, 38.400, etc. Este generador sirve tanto para el modo síncrono como el asíncrono y consiste de un contador/divisor de frecuencia de 8 bits controlado por un registro de tal manera que la frecuencia de transmisión se calcula mediante la siguiente fórmula:

Modo asíncrono de alta velocidad:

$$Baud_rate = \frac{Fosc}{16(X + 1)}$$

Ecuación 4

Modo asíncrono de baja velocidad:

$$Baud_rate = \frac{Fosc}{64(X + 1)}$$

Ecuación 5

Fosc. Es la frecuencia de oscilación del pic y X es el valor que deberá ser cargado en el registro interno del pic para efectuar el control sobre el generador de baudios. Para nuestra aplicación seria:

Fosc=4MHZ con configuración a alta velocidad seria:

$$9600 = \frac{4000000}{16(X + 1)}$$

$$X = 25.04 = 25$$

$$Baud_rate = \frac{4000000}{16(25 + 1)} = 9615$$

$$\%error = \frac{9615 - 9600}{9600} * 100\% = 0.16\%$$

Velocidad de transmisión

En la transmisión asíncrona por cada carácter se envía al menos 1 bit de inicio y 1 bit de parada así como opcionalmente 1 bit de paridad. Esta es la razón de que los baudios no se correspondan con el número de bits de datos que son transmitidos. Ver figura 3.14:

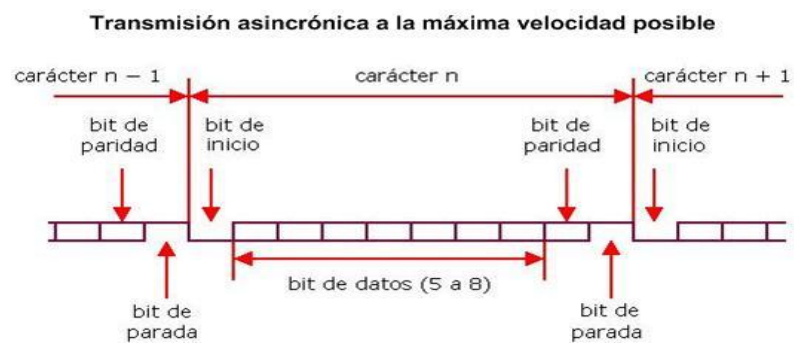


Figura 3.14 Transmisión asíncrona a máxima velocidad

Para transmitir un carácter en nuestra tarjeta de adquisición será:

$$1 \text{ bit inicio} + 8 \text{ bits datos} + 1 \text{ bit paridad} + 1 \text{ bits parada} = 11 \text{ bits.}$$

Como cada carácter posee 8 bits de datos serán transmitidos $874 * 8 = 6992$ bits de datos por segundo.

Diagrama de estados, firmware y algoritmo para el programa principal

Diagrama de estados para Tarjeta de Adquisición de Datos

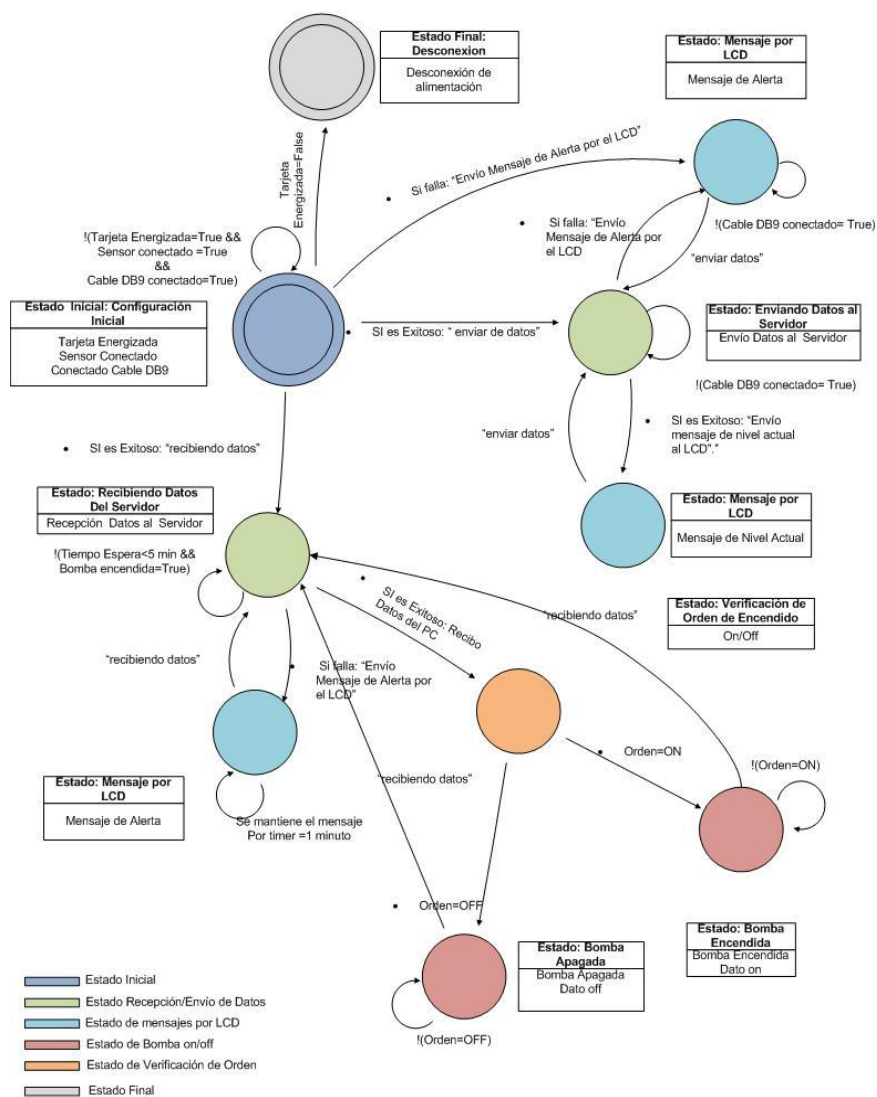


Figura 3.15 Diagrama de estados de la tarjeta de Adquisición de Datos

En la figura 3.15 se muestra, el diagrama de estados en la que el microcontrolador, empieza desde el estado inicial, hasta su estado final.

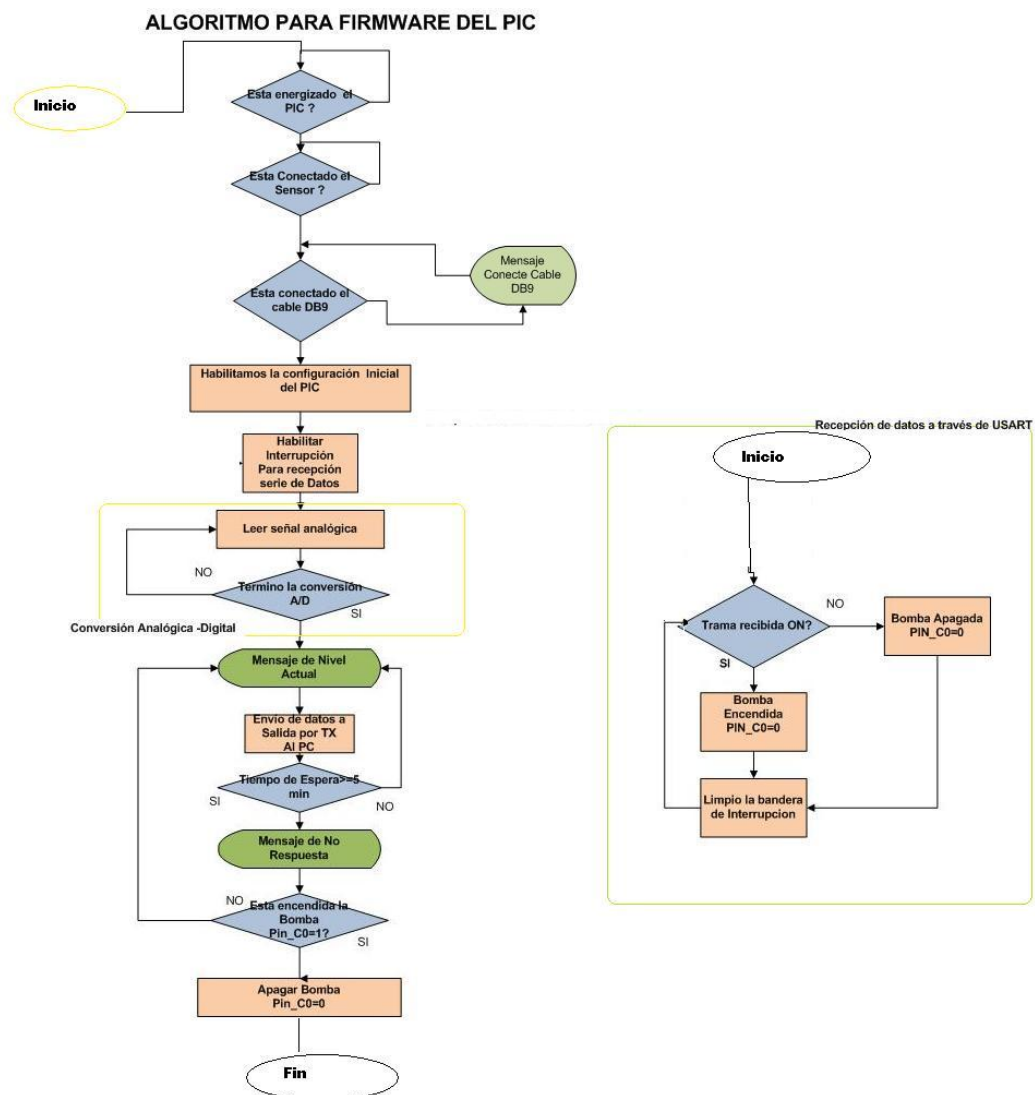


Figura 3.16 Algoritmo general del pic

En la figura 3.16 se observa el algoritmo del microcontrolador, donde se ha dividido en dos partes una para la secuencia de envío de datos, y la otra para la transferencia de datos.


```

case 3: { if (received_byte ==70) (//validamos el ultimo byte sea el caracter F
        if(dato==1)
        { output_high(PIN_C0); //Encendemos la bomba
          salto=0;
          break;
        }
        else(
          output_low(PIN_C0); //Apagamos la bomba
          salto=0;
          break;
        }
      }
    }
  else( salto=0; break; }
}
}
}
clear_interrupt(INI_RDA); //Limpio la bandera de interrupcion de la INI_RDA
enable_interrupts(INI_RDA); //Habilito la interrupcion INI_RDA
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void main()
{
  set_tris_a (0b00000011);
  set_tris_b (0b00000000);
  set_tris_c (0b10000000);

  disable_interrupts (INT_EXT); // Desabilito Interrupción INT_EXT
  disable_interrupts (INT_RB); // Desabilito Interrupción INT_RB
  enable_interrupts(global); // Habilita interrupciones
  enable_interrupts(int_rda); // Habilita Interrupción RDA
  disable_interrupts (INT_TIMER1); // Desabilito Interrupción TIMER1

  port_b_pullups(false); //Desabilito los resistores pullups del PORTB

  setup_adc(ADC_CLOCK_INTERNAL ); //Configuracion del ADC del PIC
  setup_adc_ports(RAO_ANALOG); //Configuracion el PORTA.0 como entrada analogica
  set_adc_channel(0); //Elijo el PORTA.0 como entrada para el convertidor ADC del PIC
  lcd_init(); //Configuracion de la LCD
  delay_ms(10);
}

```

Figura 3.17b Código fuente del pic

```

lcd_putc("\f Medidor d Nivel ");          //Imprimo una cadena de texto en la 1ERA Linea de la LCD
delay_ms(500);
lcd_putc("\n Nivel: ");                  //Imprimo una cadena de texto en la 2DA Linea de la LCD
done=adc_done();
output_low(PIN_CO);

while(TRUE){

while(!done){//Espero hasta q finalice la converscion del ADC del PIC
done=adc_done();
}
sensor= read_adc();//Leo el dato desde el convertidor ADC del PIC
voltaje=((sensor*5.00)/255.00); //Convierto el dato a voltaje
distancia=(voltaje*254.0) ;//Convierto de voltaje a distancia segun la especificacion del sensor
delay_ms(10);
cont++;//Incremento la variable cont2
cont2++;//Incremento la variable cont2
if(cont2>=20)//Pregunto si han transcurrido 200ms
{
lcd_gotoxy(9,2);//Ubico el cursor de la LCD en la (fila, columna)=(9,2)
printf(lcd_putc, "%.2f",distancia);//Imprimo el dato del sensor en la LCD
lcd_gotoxy(15,2);//Ubico el cursor de la LCD en la (fila, columna)=(15,2)
lcd_putc("cm");//Imprimo la cadena de texto en la LCD
cont2=0;
}

if(cont>=100)//Pregunto si han transcurrido 1seg
{
putc(36); //Envio caracter $
putc(73); //Envio caracter I
printf("%.2f",distancia);//Envio Dato del sensor
putc(70); //Envio caracter F
putc(10); //Envio caracter retorno de carro
putc(13); //Envio caracter Enter
cont=0;
}
if(input(PIN_A1)==0)//Pregunto si esta desconectado el cable serial o si esta cerrado el puerto COM
{
output_low(PIN_CO); //Apago la bomba
}
}

```

Figura 3.17c Código fuente del pic

3.6 Interfaz serial con MAX-232

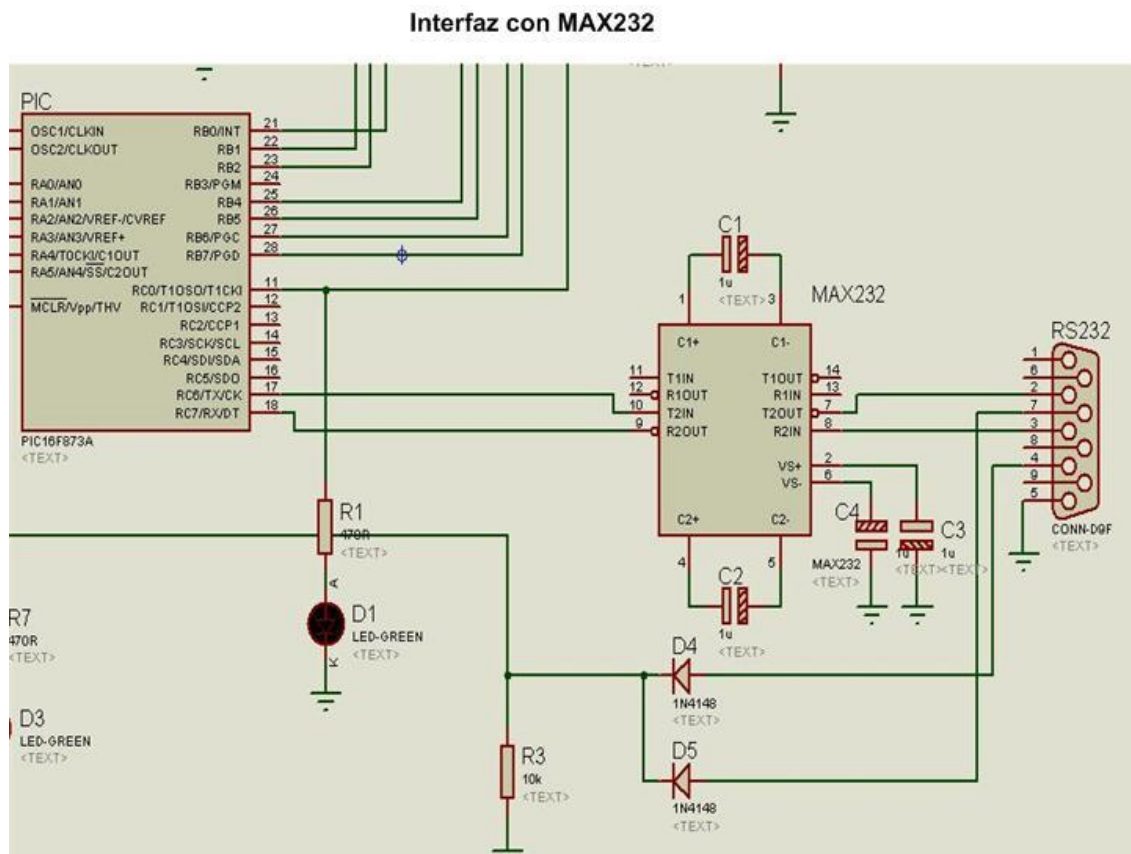


Figura 3.18 Interfaz con max232

En la figura 3.18 se muestra el dispositivo max232 conectado a la salida del puerto serial de la tarjeta de adquisición de datos en la etapa de diseño.

3.7 Modelamiento 3D del tanque

Para el modelamiento del tanque en 3D, nos enfocamos en las siguientes componentes de Visual Studio 2008 que emplean tecnología WPF:

- Librerías y clases multimedia.
- Elemento ViewPort3D.

- Elemento ModelVisual3D
- Clase MeshGeometry3D
- Tipos de materiales.
- Elemento ProjectionCamera.
- Iluminación de la escena
- Transformación.

Librerías y clases multimedia proporcionadas por Visual Studio 2008.

Las herramientas de Visual Studio 2008 nos proporcionan una gran cantidad y gama de librerías, y funcionalidades para trabajar con elementos visuales, gráficos, Audio y video y también nos ayuda a trabajar con elementos y objetos tridimensionales. Entre las clases que contribuyen a la Microsoft Windows Presentation Foundation, una debe hacer hincapié en que se encuentran en el espacio de nombres

System.Windows.Media.Media3D.

Las librerías utilizadas e investigadas para el modelamiento en 3d del tanque son las siguientes:

- System.Windows.Documents.
- System.Windows.Input.
- System.Windows.Media
- System.Windows.Media.Imaging.
- System.Windows.Navigation.

- System.Windows.Shapes.
- System.Windows.Media.Media3D.

Elemento ViewPort3D.

El elemento ViewPort3D lo contiene wpf, la utilidad de este elemento es de representar el contenido de un objeto en 3D, dentro de los límites de diseño que se generan en elementos de 2D en el ViewPort3D. El elemento ViewPort3D nos brinda la capacidad de dibujar una escena tridimensional en una superficie de dos dimensiones.

Veremos un ejemplo de declaración del Viewport en xaml. Ver la figura. 3.19

Espacio de nombres: [System.Windows.Controls](#)
Ensamblado: PresentationFramework (en PresentationFramework.dll)
XMLNS para XAML: <http://schemas.microsoft.com/winfx/2006/xaml/presentation>, <http://schemas.microsoft.com/netfx/2007/xaml/presentation>

Sintaxis

VB **C#** **C++** **F#** **JScript**

```
[ContentPropertyAttribute("Children")]  
[LocalizabilityAttribute(LocalizationCategory.NeverLocalize)]  
public class Viewport3D : FrameworkElement, IAddChild
```

XAML

```
<viewport3D>  
  Children  
</viewport3D>
```

Figura 3.19 Sintaxis del viewport3D

El elemento ViewPort3D expone la colección Children que es una colección de objetos del elemento ModelVisual3D dentro de los cuales se implementa las escenas tridimensionales.

Elemento ModelVisual3D

La clase ModelVisual3D sirve para colocar objetos en la escena, esta clase es muy importante porque construye una estructura de árbol ModelVisual3D, es decir se puede crear muchos objetos mediante este elemento.

Dentro del elemento ModelVisual3D podemos especificar cualquier dibujo que deseemos. Sin embargo a diferencia de los gráficos en dos dimensiones donde existen primitivas predefinidas para dibujar objetos como elipses o rectángulos, en el mundo de 3D únicamente contamos con una clase denominada MeshGeometry3D.

Clase MeshGeometry3D

La Clase MeshGeometry3D es básicamente una representación de una superficie. La malla representa la superficie a través de un sistema de puntos y líneas. Los puntos describen las áreas de alta y baja de la superficie, y las líneas conectan los puntos para establecer cómo llegar de un punto a otro.

Como mínimo, una superficie es un plano. Una superficie plana necesita tres puntos para definirla. Así, la superficie más simple que puede ser descrita en una malla es un único triángulo. Resulta que las mallas sólo pueden describirse con triángulos. Esto se debe que un triángulo es el más simple, es una manera fácil definir una superficie. Obviamente, no puede ser descrito con precisión por un triángulo. En su lugar, se puede aproximar por muchos triángulos más pequeños. Se puede argumentar que se puede usar un rectángulo para definir una superficie, pero no es tan granular como un triángulo. Un rectángulo se puede dividir en dos triángulos. Dos triángulos se pueden describir con mucha más

precisión en una superficie de un rectángulo único. El punto es que una malla representa una superficie a través de triángulos.

Una malla entera se compone de:

- Posiciones de malla
- Triángulo de índices
- Triángulo de las normales

Posiciones

Una posición de malla es la ubicación de un solo punto en una superficie. Cuanto más densos los puntos son, con mayor precisión describe la malla de la superficie.

Triángulo de índices

Un índice de triángulo es una posición de malla que define uno de los tres vértices de un triángulo en la malla. Las posiciones de malla por sí sola no pueden describir los triángulos de la malla. Después que las posiciones se hayan añadido, es necesario definir qué posiciones forman los triángulos.

Para más información sobre la Clase MeshGeometry3D ver link [15]

La clase MeshGeometry3D especifica un conjunto de vértices de triángulos mediante la propiedad `Positions` vista anteriormente. Esto se debe a que cualquier dibujo tridimensional es posible describirlo con triángulos. Recordemos lo que sucede en los juegos de video: cuando hacemos un acercamiento fuerte alguna escena tridimensional

podemos apreciar que las puntas y contornos de los objetos se tornan triangulares, esto se debe a que muchos objetos de los videojuegos, al igual que en WPF, son conjuntos de triángulos, que mediante su composición dan la sensación de un objeto tridimensional.

En la Siguiete Figura mostramos un ejemplo de cómo representar un cubo con la Clase MeshGeometry3D.

Un cubo es simplemente una extensión de la creación de un triángulo. La diferencia es:

Un cubo se compone de 12 triángulos en lugar de uno (seis lados, cada uno con dos triángulos). Ver figura 3.20

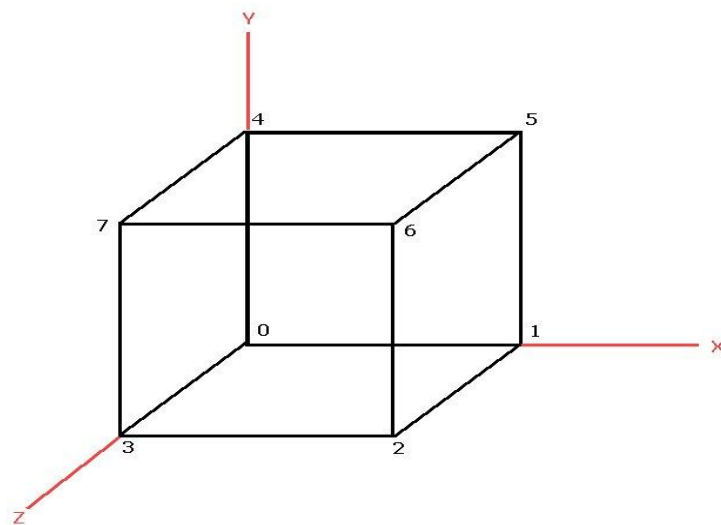


Figura 3.20 modelado de un cubo

Posiciones de malla

Las posiciones de los puntos que intervienen en el plano.

```
Point3D p0 = new Point3D (0, 0, 0);
```

```
Point3D p2 =new Point3D (5, 0, 5);
```

```
Point3D p3 =new Point3D (0, 0, 5);
```

```
Point3D p4 =new Point3D (0, 5, 0);
```

```
Point3D p5 =new Point3D (5, 5, 0);
```

```
Point3D p6 =new Point3D (5, 5, 5);
```

```
Point3D p7 = new Point3D (0, 5, 5);
```

Triángulo de índices

Indica que puntos se unen para formar el triangulo

```
cube.Children.Add(CreateTriangleModel(p3, p2, p6));
```

```
cube.Children.Add(CreateTriangleModel(p3, p6, p7));
```

```
cube.Children.Add(CreateTriangleModel(p2, p1, p5));
```

```
cube.Children.Add(CreateTriangleModel(p2, p5, p6));
```

```
cube.Children.Add(CreateTriangleModel(p1, p0, p4));
```

```
cube.Children.Add(CreateTriangleModel(p1, p4, p5));
```

```
cube.Children.Add(CreateTriangleModel(p0, p3, p7));
```

```
cube.Children.Add(CreateTriangleModel(p0, p7, p4));
```

```
cube.Children.Add(CreateTriangleModel(p7, p6, p5));
```

```
cube.Children.Add(CreateTriangleModel(p7, p5, p4));
```

```
cube.Children.Add(CreateTriangleModel(p2, p3, p0));
```

```
cube.Children.Add(CreateTriangleModel(p2, p0, p1));
```

Como hemos visto un ejemplo sencillo de cómo usar las propiedades fundamentales del MeshGeometry3D y obtener la figura deseada. Se tomo como modelo de un tanque en dibujar un cilindro ya que se asemeja a los tanques industriales de forma cilíndrica.

Nos encontramos con un gran problema al momento de modelar el tanque, el problema fue de cómo representar los puntos en el plano, ya que el cilindro tiene circunferencia, es decir una superficie curva si sabemos que las posiciones de los triángulos son rectas al aplicar las propiedades de la Clase MeshGeometry3D.

Cilindro3D y Primitive3D Estas clases fueron muy importante para modelar el cilindro, en la investigación que se realizo para modelar el objeto, encontramos una clase de mucha ayuda y jerarquía ver link [16]. Estas clases están implementadas por medio de MeshGeometry3D y sus propiedades position, Normals, TextureCoordinates, TriangleIndices. Ver link [15].

En las Figuras 3.21 ,3.22 a, 3.22 b mostraremos las clases implementadas por Visual Studio 2008

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Media.Media3D;

namespaceCodigo_3D
{
    public abstract class Primitive3D : ModelVisual3D
    {
        public Primitive3D()
        {
            Content = _content;
            _content.Geometry = Tessellate();
        }

        public static DependencyProperty MaterialProperty =
            DependencyProperty.Register(
                "Material",
                typeof(Material),
                typeof(Primitive3D), new PropertyMetadata(
                    null, new PropertyChangedCallback(OnMaterialChanged)));

        public Material Material
        {
            get { return (Material)GetValue(MaterialProperty); }
            set { SetValue(MaterialProperty, value); }
        }

        internal static void OnMaterialChanged(Object sender,
            DependencyPropertyChangedEventArgs e)
        {
            Primitive3D p = ((Primitive3D)sender);

            p._content.Material = p.Material;
        }

        internal static void OnGeometryChanged(DependencyObject d)
        {
            Primitive3D p = ((Primitive3D)d);

            p._content.Geometry = p.Tessellate();
        }

        internal double DegToRad(double degrees)
        {
            return (degrees / 180.0) * Math.PI;
        }
    }
}

```

Figura 3.21 Código fuente del modelado

```

using System.Windows;
using System.Windows.Media;
using System.Windows.Media.Media3D;
using System;

namespaceCodigo_3D
{
    publicsealedclassCilindro3D : Primitive3D
    {
        internalPoint3DGetPosition(doublet, doubley)
        {
            doublex = Math.Cos(t);
            doublez = Math.Sin(t);

            returnnewPoint3D(x, y, z);
        }
        privateVector3DGetNormal(doublet, doubley)
        {
            doublex = Math.Cos(t);
            doublez = Math.Sin(t);

            returnnewVector3D(x, 0, z);
        }
        privatePointGetTextureCoordinate(doublet, doubley)
        {
            Matrixm = newMatrix();
            // m.Scale(200, 200.0);
            m.Scale(1 / (2 * Math.PI), -0.5);

            Pointp = newPoint(t, y);
            p = p * m;

            returnp;
        }
    }
}

```

Figura 3.22a Código fuente del modelado

```

internal override Geometry3D Tessellate()
{
    // Parametros jj = new Parametros();
    int tDiv = 32;
    int yDiv = 32;
    double maxTheta = DegToRad(360.0);
    double minY = -1.0;
    double maxY = 1;//jj.ValorParametroM();//1.0;

    double dt = maxTheta / tDiv;
    double dy = (maxY - minY) / yDiv;

    MeshGeometry3D mesh = new MeshGeometry3D();

    for (int yi = 0; yi <= yDiv; yi++)
    {
        double y = minY + yi * dy;

        for (int ti = 0; ti <= tDiv; ti++)
        {
            double t = ti * dt;

            mesh.Positions.Add(GetPosition(t, y));
            meshNormals.Add(GetNormal(t, y));
            mesh.TextureCoordinates.Add(GetTextureCoordinate(t, y));
        }
    }

    for (int yi = 0; yi < yDiv; yi++)
    {
        for (int ti = 0; ti < tDiv; ti++)
        {
            int x0 = ti;
            int x1 = (ti + 1);
            int y0 = yi * (tDiv + 1);
            int y1 = (yi + 1) * (tDiv + 1);

            mesh.TriangleIndices.Add(x0 + y0);
            mesh.TriangleIndices.Add(x0 + y1);
            mesh.TriangleIndices.Add(x1 + y0);

            mesh.TriangleIndices.Add(x1 + y0);
            mesh.TriangleIndices.Add(x0 + y1);
            mesh.TriangleIndices.Add(x1 + y1);
        }
    }

    mesh.Freeze();
    return mesh;
}

```

Figura 3.22b Código fuente del modelado

Como vemos en las Figuras 2.21, 2.22 a y 2.22 b se muestra la Clase MeshGeometry3D aplicando sus propiedades para modelar el Cilindro.

Una vez que hemos definido nuestra superficie de dibujo procederemos a utilizar la propiedad ModelVisual3D.Content para dentro de la misma especificar un MeshGeometry3D, en este caso con las posiciones establecidas estaremos generando un cilindro.

Tipos de materiales proporcionado por Wpf.

WPF especifica tres tipos de materiales:

- **SpecularMaterial._** que refleja la luz de tal manera que el objeto parece brillar.
- **EmissiveMaterial._** que refleja la luz.
- **DiffuseMaterial._** que no refleja la luz.

Para el proyecto nosotros usamos DiffuseMaterial porque se relaciona con la luz difusa de acuerdo al material que se le implante al objeto del modelado. Para especificar el tipo de material del cual se compone nuestro Mesh hacemos uso del siguiente código dentro de GeometryModel3D: ver figura 3.23.

```
<DiffuseMaterial>  
  <DiffuseMaterial.Brush>  
    <ImageBrush ImageSource="C:\Windows\Web\Wallpaper\Material.jpg" Opacity="10" />  
  </DiffuseMaterial.Brush>  
</DiffuseMaterial>
```

Figura 3.23 Sintaxis del viewport3D

3.7.1 Perspectiva de cámara

Elemento ProjectionCamera.

Dentro del mundo de tercera dimensión los objetos cambian de apariencia desde el punto de vista de quien observa al objeto. En WPF se especifica este punto de vista haciendo uso de los objetos que heredan de ProjectionCamera.

Controlamos las cámaras como si en verdad la misma estuviera posicionada en el espacio, la propiedad Position es del tipo Point3D especifica las coordenadas en la cual se encuentra la cámara. A continuación en el grafico siguientes se mostrará su sintaxis y librerías. Ver Figura. 3.24.

Espacio de nombres: [System.Windows.Media.Media3D](#)
Ensamblado: PresentationCore (en PresentationCore.dll)
XMLNS para XAML: <http://schemas.microsoft.com/winfx/2006/xaml/presentation>, <http://schemas.microsoft.com/netfx/2007/xaml/presentation>

Sintaxis

VB C# C++ F# JScript

```
public sealed class PerspectiveCamera : ProjectionCamera
```

XAML

```
<PerspectiveCamera .../>
```

Figura 3.24 Sintaxis de la perspectiva de cámara

Luego de haber declarado la Clase PerspectiveCamera se debe agregar parámetros, para un buen manejo de dirección y posición del objeto, en la investigación realizada se encuentra los siguientes parámetros de esa clase, dieron un buen resultado al momento de observar el objeto tridimensional.

```
Position="x,y,z"
```

```
UpDirection="x,y,z"
```

```
LookDirection="x,y,z"
```

```
FieldOfView="45"
```

```
NearPlaneDistance="0.125"/>
```

En la figura 3.25 se muestra la sintaxis, librerías al cual pertenecen y la definición de cada parámetro en la construcción de una `PerspectiveCamera`.

Espacio de nombres: System.Windows.Media.Media3D
Asamblea: PresentationCore (en PresentationCore.dll)

Sintaxis

```

VB C# C++ F# JScript
"Declaración
Public Sub Nuevo ( _
    Posición de Point3D, _
    Como lookDirection Vector3D, _
    Como upDirection Vector3D, _
    fieldOfView As Double _
)

```

Parámetros

posicion

Tipo: System.Windows.Media.Media3D.Point3D
 Point3D que especifica la posición de la cámara.

lookDirection

Tipo: System.Windows.Media.Media3D.Vector3D
 Vector3D que especifica la dirección de la proyección de la cámara.

upDirection

Tipo: System.Windows.Media.Media3D.Vector3D
 Vector3D que especifica la dirección hacia arriba según la perspectiva del espectador.

fieldOfView

Tipo: System.Double
 Ancho de la cámara el ángulo de proyección, se especifica en grados.

Figura 3.25 Sintaxis de la perspectiva de cámara

En la figura 3.26 mostraremos los valores de cada parámetro utilizados en la clase PerspectiveCamera, dichos valores son cantidades principales de utilidad para el proyecto que aporta a una mejor visualización eficaz y eficiente en el modelado 3D.

```

<Viewport3D Canvas.Left="815" Canvas.Top="97" Name="viewport3D1" Margin="417,0,0,336.5">
  <Viewport3D.Camera>
    <PerspectiveCamera Position="0,0,-8" UpDirection="0,1,0" LookDirection="0,0,1" FieldOfView="45" NearPlaneDistance="0.125"/>
  </Viewport3D.Camera>

```

Figura 3.26 Sintaxis de la perspectiva de cámara

Iluminación de la escena

A diferencia de lo que sucede con los objetos 2D donde el concepto de luz no existe, en 3D es indispensable especificar una fuente de luz debido a que si no existe ésta simplemente nuestro objeto no será visible de la misma manera que sucede en la vida real. Esto significa debemos de especificar el lugar que estará irradiando la luz.

Las luces siguientes se derivan de la clase base Light:

AmbientLight: proporciona iluminación de ambiente que ilumina uniformemente todos los objetos sin tener en cuenta su ubicación u orientación.

DirectionalLight: ilumina como una fuente de luz distante. Las luces direccionales tienen Direction, que se especifica como Vector3D, pero ninguna ubicación concreta.

Más información de luces esta en [7] Luces en WPF. En le figura 2.27 mostraremos los parámetros colocados de la Clase DirectionalLight.

```
<ModelVisual3D>  
  <ModelVisual3D.Content>  
    <DirectionalLight Color="White" Direction="0,0,1" />  
  </ModelVisual3D.Content>  
</ModelVisual3D>
```

Figura 3.27 Sintaxis de direccionamiento de luz

3.7.2 Luz ambiental

Corresponde al modelo en el cual cada objeto se presenta con una intensidad intrínseca. Se puede considerar este modelo, que no tiene una fuente de luz externa, como la

descripción de un mundo ligeramente irreal de objetos no reflejantes y auto luminosos. En este caso cada objeto aparece como una silueta monocromática, a menos que sus partes individuales, como los polígonos de un poliedro, reciban matices diferentes al crear el objeto. Ver más en [4] Características de iluminación en el proyecto no usamos luz ambiental, usamos librerías de Wpf. que ayudaron aplicar modelos de iluminación en la escena 3D del modelado del tanque.

3.7.3 Luz difusa, material difuso

La luz que llega al objeto rebota pero con menor intensidad que la especular y se difumina por su superficie. Para modelar la reflexión difusa se usa la fórmula de Lambert (mucho más sencilla que la de Phong!) ver más en:[5] Características de Iluminación 2

Reflexión difusa

Consideramos ahora un tipo de reflexión causada por una fuente de luz puntual cuyos rayos emanan en todas direcciones de modo uniforme. Se trata de una buena aproximación para aquellas fuentes cuyo tamaño sea pequeño comparado con el del resto de los objetos del entorno.

Más información vaya al link: [6] Características de Iluminación 3

Transformación.

Como aún no hacemos ninguna transformación de nuestro objeto no podemos ver la profundidad que tiene. Añadiremos una transformación posteriormente. Otro aspecto que

podemos resaltar es que a pesar de que el material especificado con DiffuseMaterial fue una imagen escogida por el sistema.

En la figura 2.28 se muestra el código del modelo del tanque en 3D, realizado por librerías y clases de Visual Studio 2008, que ofrece un gran aporte y beneficio para el ámbito multimedia y escenas tridimensionales.

```
<Viewport3D Canvas.Left="815" Canvas.Top="97" Name="viewport3D1" Margin="417,0,0,336.5">
  <Viewport3D.Camera>
    <PerspectiveCamera Position="0,0,-8" UpDirection="0,1,0" LookDirection="0,0,1" FieldOfView="45"
      NearPlaneDistance="0.125"/>
  </Viewport3D.Camera>
  <Viewport3D.Children>
    <ModelVisual3D>
      <ModelVisual3D.Content>
        <DirectionalLight Color="White" Direction="0,0,1" />
      </ModelVisual3D.Content>
    </ModelVisual3D>

    <ModelVisual3D>
      <ModelVisual3D.Transform>
        <RotateTransform3D>
          <RotateTransform3D.Rotation>
            <AxisAngleRotation3D x:Name="rotation1" Angle="0" Axis="0,0,0" />
          </RotateTransform3D.Rotation>
        </RotateTransform3D>
      </ModelVisual3D.Transform>
      <!--TranslateTransform3D OffsetX="-1" OffsetY="0" OffsetZ="-1.73" /-->
      <my:Cilindro3D>
        <ModelVisual3D.Transform>
          <TranslateTransform3D OffsetX="-2" OffsetY="1.0" OffsetZ="0" />
        </ModelVisual3D.Transform>
        <my:Cilindro3D.Material>
          <DiffuseMaterial>
            <DiffuseMaterial.Brush>
              <ImageBrush ImageSource="C:\Windows\Web\Wallpaper\Material.jpg" Opacity="10" />
            </DiffuseMaterial.Brush>
          </DiffuseMaterial>
        </my:Cilindro3D.Material>
      </my:Cilindro3D>
    </ModelVisual3D>
  </Viewport3D.Children>
</Viewport3D>
```

Figura 3.28 Código fuente de xaml del tanque

Estas características importantes para modelar un cilindro o tanque en visual studio 2008, es de gran importancia, adquirir conocimientos en programación xaml es de gran proporción y evolución para diseños tridimensionales e industriales.

Como veremos en un esquema en la figura 3.29:

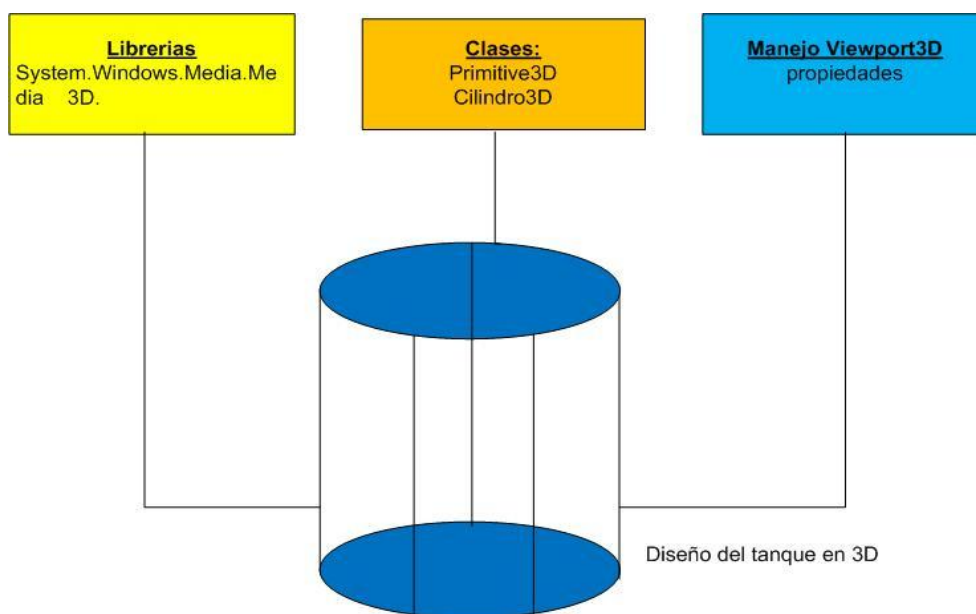


Figura 3.29 Diagrama general del modelado

3.8 Representación de datos

En este capítulo se enfoca en la representación de los datos, enviados por el sensor, en la figura 3.30 muestra un esquema de la representación de recibimientos de los dato en el computador.

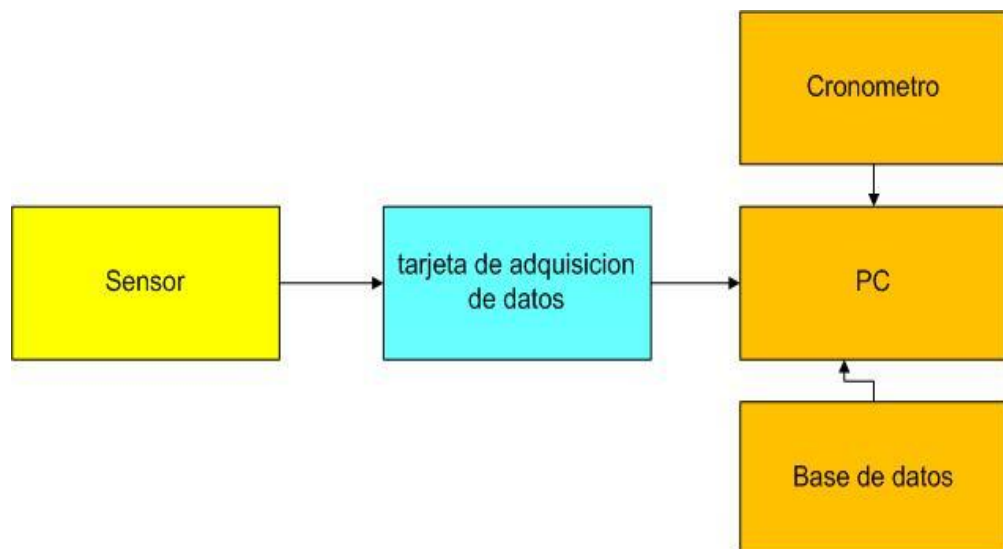


Figura 3.30 Esquema de representación de los datos

3.8.1 Tiempo de llenado

Para calcular el tiempo de llenado del tanque hemos realizado dos soluciones importantes que son factibles en el momento de llenar un tanque específico, la primera solución fue de implementar en código un cronómetro, este cronómetro se encarga de mostrarle al usuario el tiempo que dura el transcurso del llenado de un tanque, mostrándole las horas, los minutos, los segundos y los milisegundos en el proceso de llenado, este cronómetro es independiente a la hora del computador.

Uno de estos beneficios de crear el cronometro es indicarle al usuario el proceso de llenado desde el inicio hasta el final, el llenado completo del tanque. El tiempo de un computador normal no hemos tomado en cuenta, porque las computadoras saben tener las fallas de la des configuración de la hora, y así evitamos un fallo en el sistema.

La segunda solución fue crear una base de datos, para el almacenamiento de los datos del sensor, datos del tiempo o cronometro y datos de la lógica difusa que emplea su función de membresía. Este objetivo se realizó para tener el control de cada intervalo de tiempo de llenado del tanque.

Otras de las características importantes de esta solución es de insertar en cada instante de tiempo la altura, que son datos que envía el sensor a la tarjeta de adquisición y su vez es leído por la computadora para guardarlo en la base de datos.

En la siguiente formula mostramos la ecuación del tiempo de llenado que tiene nuestro proyecto.

$$\sum_{i=1}^{i=Th} (t_1 + t_2 + t_3 + \dots + t_n)$$

t = tiempo_Cronometro
Th = tiempo_altura_limite
t = hh : mm : ss : ms

Ecuación 6

Como en la segunda solución hablamos de la base de datos en la figura 3.31 mostramos el esquema lógico de la base de datos.

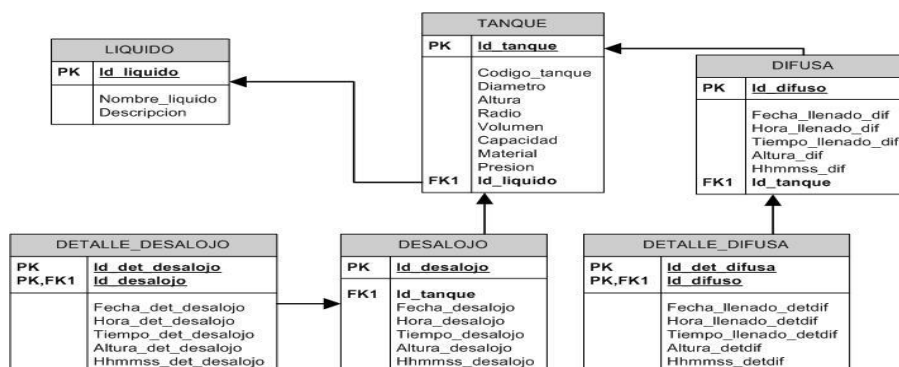


Figura 3.31 Diagrama base de datos

3.8.2 Promedio de llenado

Para automatizar el proceso de llenado de los tanques, sin necesidad que el usuario lo apague manualmente y se auto calibre por cada llenado de tanque, se analizó y se pensó como solución, obtener el promedio de llenado de cada tiempo del proceso culminado.

En este proceso nos ayuda los historiales de la base de datos, en la cual por cada llenado guardamos su tiempo completo de culminación.

En la figura 3.32 se muestra un esquema general del manejo del promedio de llenado Manipulando la base de datos.

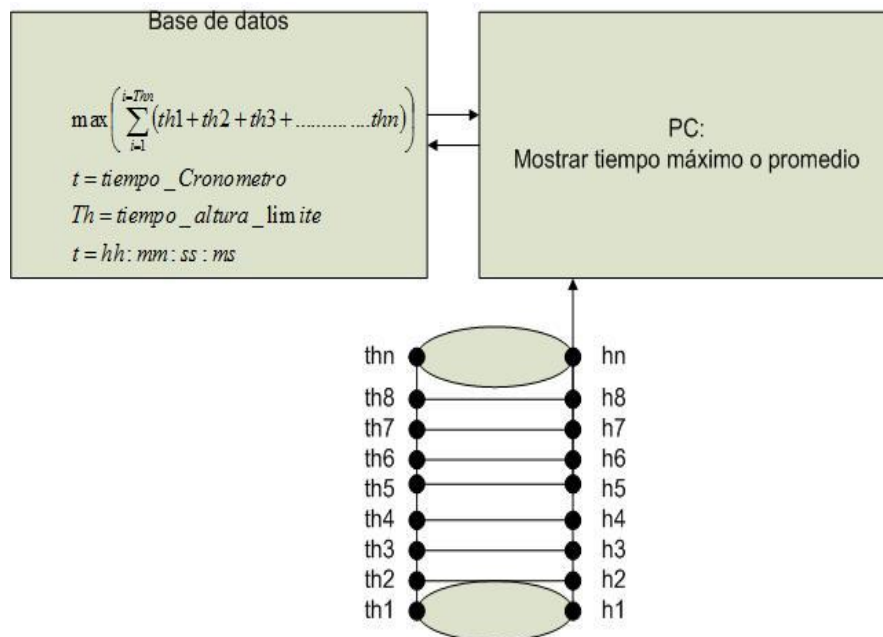


Figura 3.32 Promedio llenado

Como observamos en la fórmula de la fig. 3.32 tenemos en la fórmula como tiempo máximo “Max”, escogimos como tiempo máximo al conjunto de tiempos que se encuentra en el historial del detalle de la base de datos, ya que el motor de la base de datos SQL server 2005, no cumple con las funciones de promedio, analizando el caso obtuvimos como conclusión que por cada llenado, ya que cada llenado no varía mucho con su tiempo final, decidimos optar por el tiempo máximo, este tiempo auto calibra el llenado del tanque y aprende históricamente en cada día de llenado.

3.8.3 Altura y presión del líquido del tanque

Medir la altura del tanque en cada instante de tiempo, dependemos del sensor acústico y también por la función de membresía, aquella función facilita el apagado automático de la bomba.

Mientras el sensor este en lectura y funcionando con normalidad esos datos capturados por la tarjeta de adquisición de datos pasa al PC y directamente se guarda en la base de datos la altura capturada y el instante de tiempo para esa altura específica.

Para determinar la presión de nuestro proyecto nos enfocamos en la presión total máxima que puede soportar el tanque, es decir su punto más bajo, lo obtenemos mediante el ingreso de los datos del tanque en la interfaz gráfica.

Por ejemplo en nuestros tanques tiene una altura de 30 cm es decir 0.3 mts. Y de acuerdo como la formula física $Presion = \rho * g * h$ entonces el resultado, asumiendo la densidad del agua $\rho_{agua} = 1000 \text{ kg/m}^3$.

Presión = 2940 pascales.

En la siguiente figura 3.33 mostraremos un diseño de la altura y presión del tanque.

Nosotros calculamos la presión como dato extra para registrar algunas características que las industrias requieren como objetivo, nuestro proyecto incluso tiene la ventaja para añadir nuevos requerimientos de acuerdo a como se integre en la industria

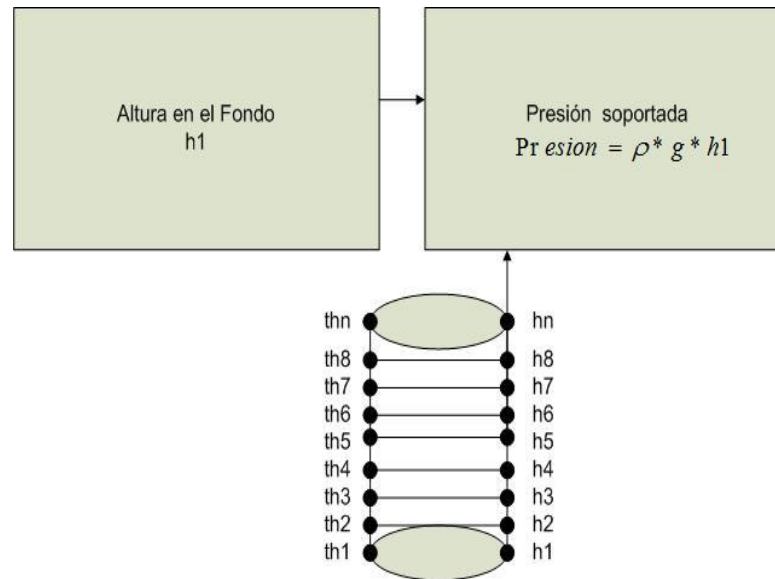


Figura 3.33 Modelo de altura y presión

3.9 Diseño de la lógica difusa

En muchos procesos complejos, el control que ejerce un operador humano es más efectivo que el que proporciona un controlador automático convencional. Para esto el operador se basa en su experiencia (heurística). Usualmente, el operador expresa sus estrategias de control lingüísticamente como un conjunto de reglas de toma de decisiones. Para nuestro proyecto un esquema similar sería. Ver figura 3.34

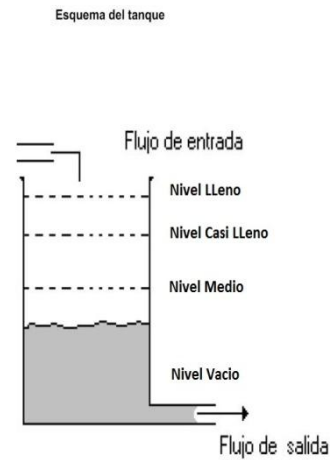


Figura 3.34 Diagrama de nivel

Y traducido en lenguaje de reglas difusas sería

**SI NIVEL es VACIO o NIVEL es MEDIO entonces MANTENER
PRENDIDA BOMBA**

**SI NIVEL es MEDIO o NIVEL es CASILLENO entonces MANTENER
PRENDIDA BOMBA**

**SI NIVEL es CASILLENO o NIVEL es LLENO entonces MANTENER
PRENDIDA BOMBA**

SI NIVEL es LLENO entonces APAGAR LA BOMBA

3.9.1 Funciones de membresía y Diseño de las funciones de membresía

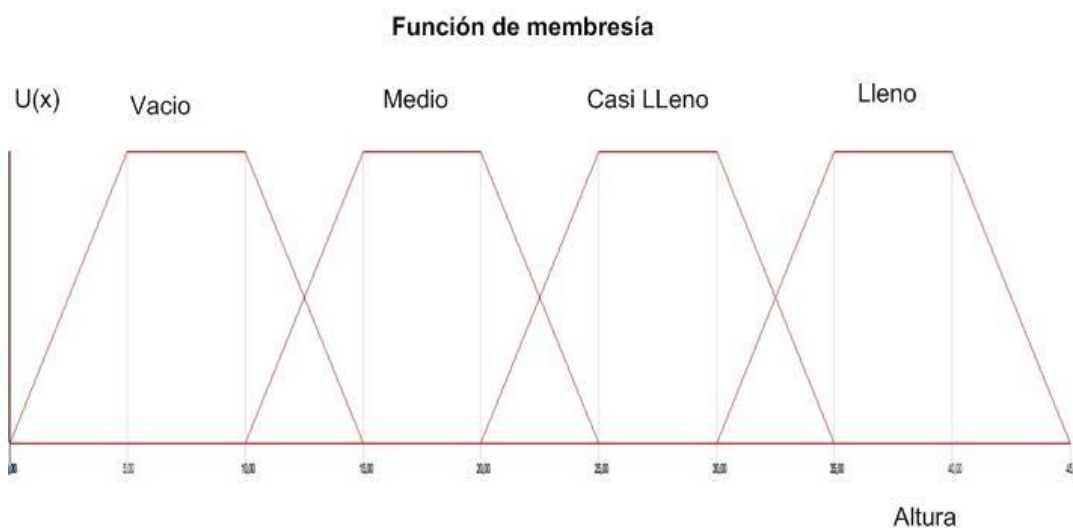


Figura 3.35 Función de membresía del sistema

Para el proceso de inteligencia artificial de nuestro proyecto usamos el siguiente esquema. Ver figura 3.36

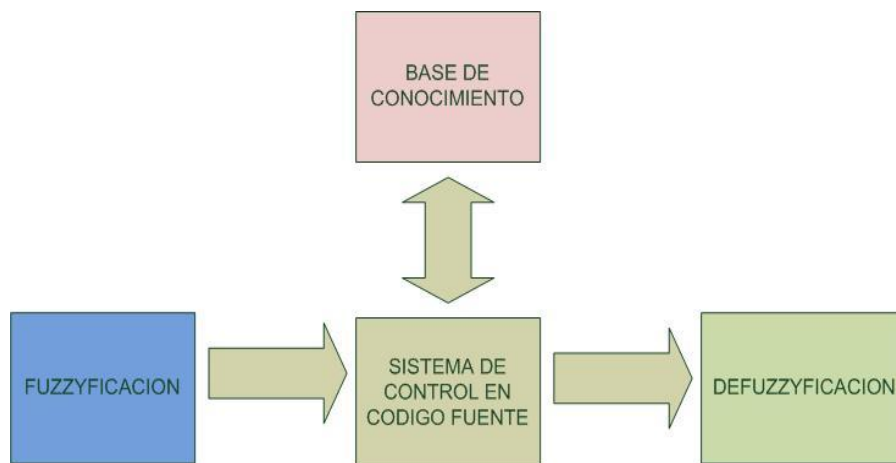


Figura 3.36 Esquema fuzzificación / defuzzificación

Con esto debemos considerar las funciones de membresía expresadas en términos del trapecoide y su respectiva defuzzificación , con esto las

funciones quedarían:

$$f \text{ (vacío, } 0, 5, 10, 15 \text{)} \\ \mu \text{ (vacío)} = \begin{cases} 0; & X \leq 0 \\ \frac{x-0}{5}; & 0 \leq x \leq 5 \\ 1; & 5 \leq x \leq 10 \\ \frac{15-x}{15-10}; & 10 \leq x \leq 15 \\ 0; & x \geq 15 \end{cases}$$

$$\text{vacío} = \frac{\int_0^5 \frac{x}{5} * x . dx + \int_5^{10} x . dx + \int_{10}^{15} \frac{15-x}{15-10} * x . dx}{\int_0^5 \frac{x-0}{5} . dx + \int_5^{10} 1 . dx + \int_{10}^{15} \frac{15-x}{15-10} . dx}$$

$$f \text{ (medio, } 10, 15, 20, 25 \text{)} \\ \mu \text{ (medio)} = \begin{cases} 0; & X \leq 10 \\ \frac{x-10}{15}; & 10 \leq x \leq 15 \\ 1; & 15 \leq x \leq 20 \\ \frac{25-x}{25-20}; & 20 \leq x \leq 25 \\ 0; & x \geq 25 \end{cases}$$

$$\text{medio} = \frac{\int_{10}^{15} \frac{x-10}{15} * x . dx + \int_{15}^{20} x . dx + \int_{20}^{25} \frac{25-x}{25-20} * x . dx}{\int_{10}^{15} \frac{x-10}{15} . dx + \int_{15}^{20} 1 . dx + \int_{20}^{25} \frac{25-x}{25-20} . dx}$$

$$f(\text{casi_lleno}, 20, 25, 30, 35) = \begin{cases} 0; & X \leq 20 \\ \frac{x-20}{25}; & 20 \leq x \leq 25 \\ 1; & 25 \leq x \leq 30 \\ \frac{35-x}{35-30}; & 30 \leq x \leq 35 \\ 0; & x \geq 35 \end{cases}$$

$$\text{casi_lleno} = \frac{\int_{20}^{25} \frac{x-20}{25} * x.dx + \int_{25}^{30} x.dx + \int_{30}^{35} \frac{35-x}{35-30} * x.dx}{\int_{20}^{25} \frac{x-20}{25}.dx + \int_{25}^{30} 1.dx + \int_{30}^{35} \frac{35-x}{35-30}.dx}$$

$$f(\text{lleno}, 30, 35, 40, 45) = \begin{cases} 0; & X \leq 30 \\ \frac{x-30}{35}; & 30 \leq x \leq 35 \\ 1; & 35 \leq x \leq 40 \\ \frac{45-x}{45-40}; & 40 \leq x \leq 45 \\ 0; & x \geq 45 \end{cases}$$

$$\text{lleno} = \frac{\int_{30}^{35} \frac{x-30}{35} * x.dx + \int_{35}^{40} x.dx + \int_{40}^{45} \frac{45-x}{45-40} * x.dx}{\int_{30}^{35} \frac{x-30}{35}.dx + \int_{35}^{40} 1.dx + \int_{40}^{45} \frac{45-x}{45-40}.dx}$$

Dotfuzzy

DotFuzzy es un código abierto independiente de biblioteca de clases de lógica difusa. La biblioteca se construye en C # y por lo tanto pueden ser utilizados por todas las lenguas del entorno. NET. Debido a implementación orientada a objetos de la biblioteca es fácil de usar e implementar. DotFuzzy está diseñado para ser flexible

DotFuzzy implementa fuzzificación, las reglas de validación / evaluación y defuzzificación con el método de centroide.

Motor de inferencias

Código Fuente del Motor de Inferencia

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using DotFuzzy;

namespace SerialPuerto
{
    public class FuzzyClass
    {
        public string ImplementarReglasFuzzy(double valor,int medio ) {

            LinguisticVariable nivel = new LinguisticVariable("Nivel");
            nivel.MembershipFunctionCollection.Add(new MembershipFunction("vacio", 0, 5, 10,15));
            nivel.MembershipFunctionCollection.Add(new MembershipFunction("medio", 10, 15, 20, 25));
            nivel.MembershipFunctionCollection.Add(new MembershipFunction("casi_lleno",20,25,30,35));
            nivel.MembershipFunctionCollection.Add(new MembershipFunction("lleno",30,35,40,45));

            LinguisticVariable power = new LinguisticVariable("Power");
            power.MembershipFunctionCollection.Add(new MembershipFunction("on", 0, 20, 28, 40));
            power.MembershipFunctionCollection.Add(new MembershipFunction("off", 41, 42, 44, 45));

            FuzzyEngine fuzzyEngine = new FuzzyEngine();
            fuzzyEngine.LinguisticVariableCollection.Add(nivel);
            fuzzyEngine.LinguisticVariableCollection.Add(power);
            fuzzyEngine.Consequent = "Power";
            if (medio == 1)
            {
                fuzzyEngine.FuzzyRuleCollection.Add(new FuzzyRule("IF (Nivel IS medio) THEN Power IS off"));
            }
            else
            {
                fuzzyEngine.FuzzyRuleCollection.Add(new FuzzyRule("IF (Nivel IS vacio) OR (Nivel IS medio) THEN Power IS on"));
                fuzzyEngine.FuzzyRuleCollection.Add(new FuzzyRule("IF (Nivel IS medio) OR (Nivel IS casi lleno) THEN Power IS on"));
            }
        }
    }
}

```

Figura 3.37a Código fuente del motor de inferencia

Mediante el código fuente que adquirimos de esta librería el código fuente principal que realiza la manipulación de la lógica difusa para el proyecto es el siguiente:

Código del Motor de inferencias

```

fuzzyEngine.FuzzyRuleCollection.Add(new FuzzyRule("IF (Nivel IS medio) OR (Nivel IS casi_lleno) THEN Power IS on"));
//fuzzyEngine.FuzzyRuleCollection.Add(new FuzzyRule("IF (Nivel IS casi_lleno) OR (Nivel IS lleno) THEN Power IS on"));
fuzzyEngine.FuzzyRuleCollection.Add(new FuzzyRule("IF (Nivel IS lleno) THEN Power IS off"));
}

nivel.InputValue = valor;

try
{
    //System.Windows.MessageBox.Show(fuzzyEngine.Defuzzify().ToString());
    return fuzzyEngine.Defuzzify().ToString();
}
catch (Exception e)
{
    //System.Windows.MessageBox.Show(e.Message);
    return null;
}

```

Figura 2.37b Código fuente del motor de inferencia

3.9.2 Control on/off de bomba mediante esquemas de tiempo

Nuestro proyecto maneja situaciones de apagado automáticos mediante ciertos eventos que se dan en las industrias y casos especiales que suceden día a día y las hemos tomado en cuenta.

En la siguiente figura 3.38 mostraremos un esquema de las funcionalidades de cada situación.

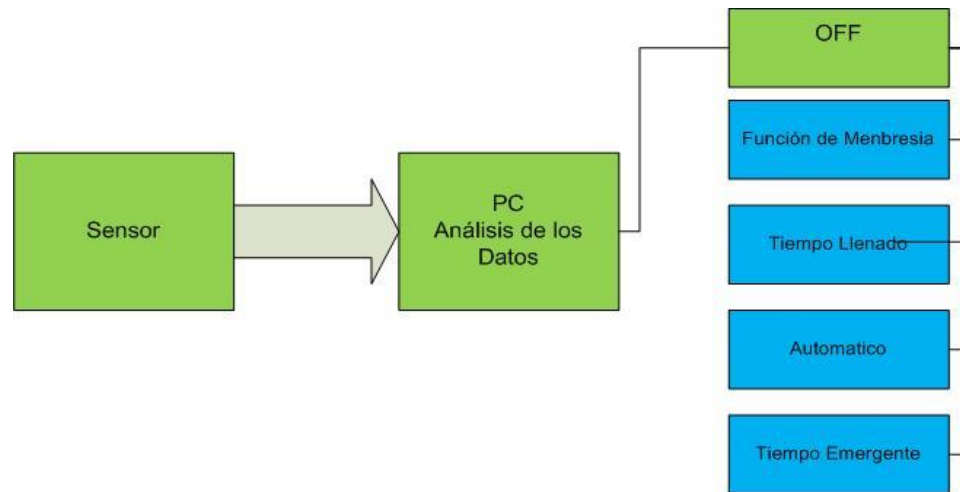


Figura 3.38 Control on/off

Como vemos en la figura 3.38 las opciones para el apagado automático de la bomba son las siguientes:

- Función de membresía
- Tiempo de llenado.
- Modo automático.
- Tiempo emergente.

Control por función de membresía

La función de membresía se encarga de indicar el estado de la altura correspondiente al envío de datos del sensor. Si del sensor se obtiene valores próximos a llenarse el tanque, la función de membresía indicará el estado on/off de acuerdo a la función defuzzificadora analizada para nuestro proyecto.

En la fig. 3.39 mostraremos el grafico de la función de membresía que indica el estado y activa las opciones on/off.

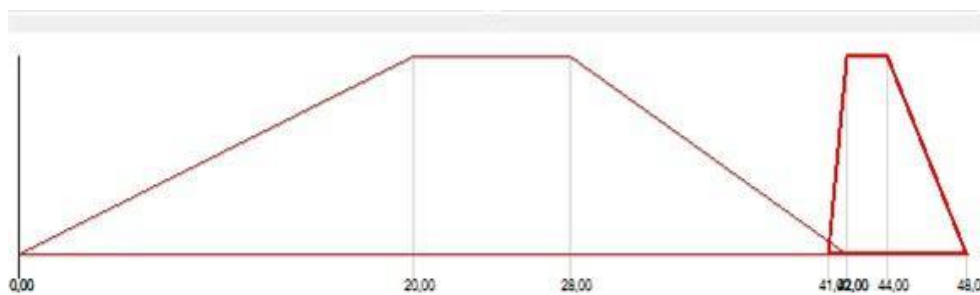


Figura 3.39 Función de membresía del sistema on/off

Tiempo de llenado. El tiempo de llenado se obtiene mediante la base de datos (“base de conocimientos”) en el siguiente capítulo nos profundizaremos en detalle todo su proceso. El tiempo de llenado cumple un papel importante para el apagado automático de la bomba, porque en la base de conocimientos “base de datos” obtenemos todos los registros de los tiempos de llenados anteriores.

Control por modo automático.

El modo automático se activa, cuando el sensor emite valores erróneos o el sensor esta averiado, cuando sucede alguna de estas opciones nuestro proyecto toma la opción a modo automático.

El modo automático consiste en consultar el histórico de la base de conocimientos “base de datos” de los tiempos de llenado de los procesos anteriores realizados.

En la Fig. 3.40 mostraremos un esquema del modo automático.

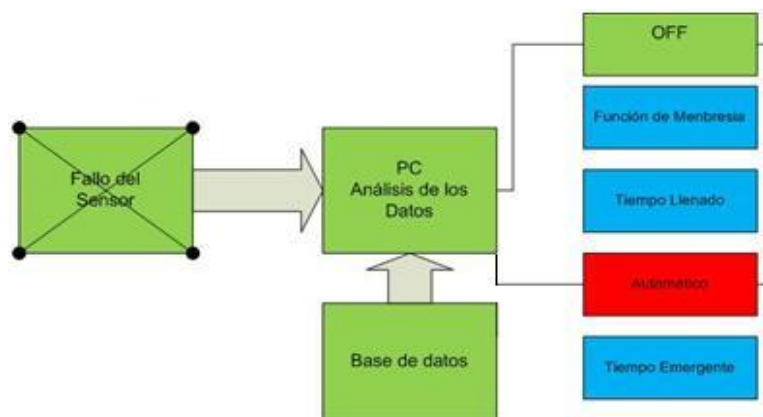


Figura 3.40 Control por modo automático

Tiempo emergente.

El modo emergente surge de casos especiales que suceden en las computadoras, cuando se sobrecarga los procesos o conflictos que hacen que la maquina se inactive por cualquier proceso o subprocesso colgado.

En el proyecto se lo denomino apagado emergente “modo catástrofe”, cumple la función, cuando el sensor está inactivo y procesos sobrecargados. Internamente creamos un cronometro y ese método del cronometro lo enlazamos con un thread, que facilito manejar los procesos en paralelo y no deja que el computador colapse. El cronometro compara el tiempo de llenado consultado en la base, con el cronometro interno creado, si el tiempo del cronometro interno es mucho mayor al tiempo obtenido por la base de datos, inmediatamente por emergencia mandamos apagar el motor.

En la fig. 3.41 mostraremos el esquema del apagado emergente “modo catástrofe”.

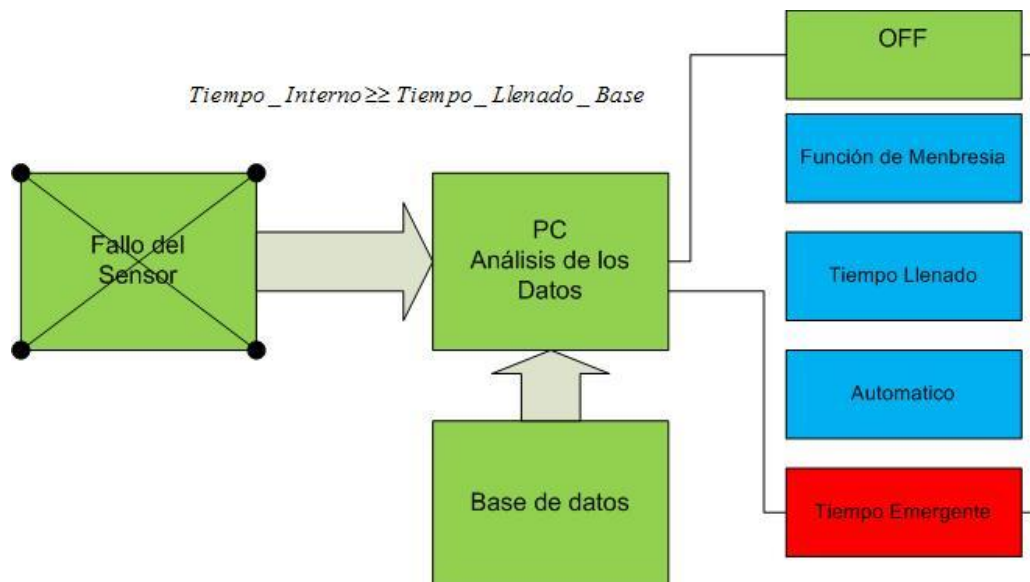


Figura 3.41 Modo emergente

3.9.3 Base de conocimiento motor de bases de datos

En nuestro proyecto creamos la base de datos o llamada Base de conocimientos, fue por llevar un registro de los tiempos de llenado y el detalle de cada llenado que se realiza día a día en las empresas. La base de datos cumple un papel importante a la hora de comenzar y finalizar el proceso de llenado.

En la figura 3.42 mostraremos el esquema de la base de datos hecha en SQL server 2005

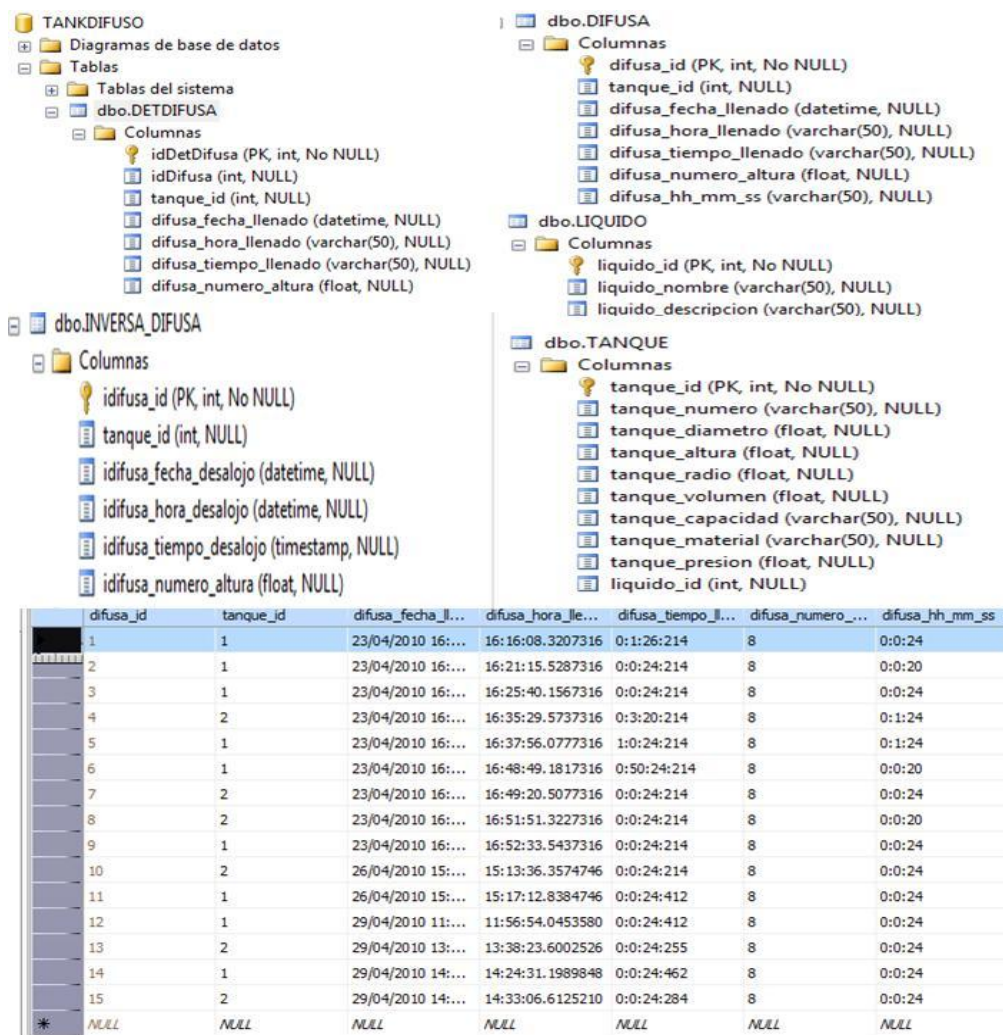


Figura 3.42 Base de conocimiento

3.10 Interfaz interactiva del usuario

En nuestro proyecto creamos nuestra interfaz de usuario separando ciertos bloques de control para una mejor comprensión e interacción con el usuario y aproximándonos a un sistema scada. Ver figura 3.43

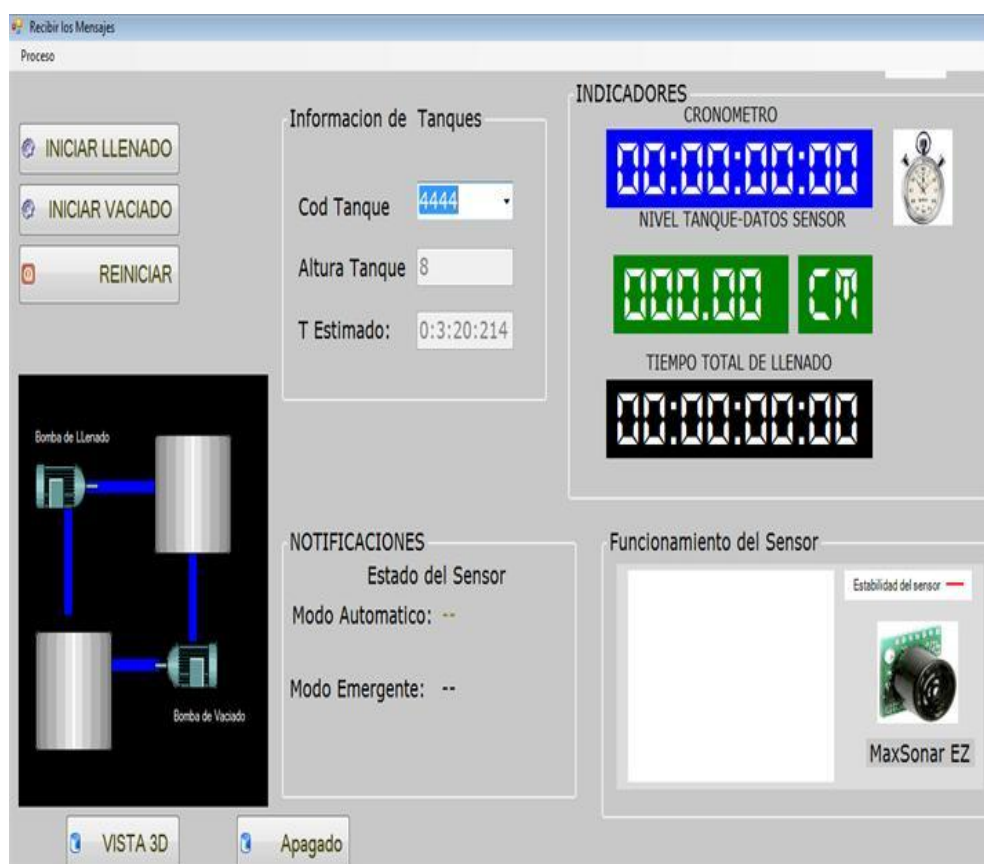


Figura 3.43 Interfaz de usuario

3.11 Diseño del plan de pruebas

Plan de Pruebas

Para lograr nuestro objetivo, que el llenado del tanque cumpla un 98% de exactitud en calibración, visualización, control y monitoreo hemos realizado un plan de pruebas evaluando los casos de uso que intervienen en el proyecto.

Lista de pruebas a evaluar:

- ✓ Pruebas del Sensor
- ✓ Pruebas del Acondicionador de Señal
- ✓ Pruebas del PIC16f873
- ✓ Prueba de la interfaz serial con MAX232
- ✓ Pruebas de entrada de datos al PC
- ✓ Pruebas de modelado en 3D y salida de datos
- ✓ Pruebas de las funciones de membresía para el control de la bomba.

Para las pruebas de nuestro proyecto, utilizamos los siguientes materiales:

- ✓ 2 tanques plásticos de aproximadamente 50 cm de alto. Para demostrar el nivel de líquido ingresado.
- ✓ 2 llaves de paso (3/4 pulgadas). Para controlar el paso del líquido al tanque.
- ✓ 2 Bombas DC 12v, 3A. especificación de los motores para trabajar de manera normal.
- ✓ 1 Tarjeta de Adquisición de datos. Para el envío y recepción de información.

Las pruebas realizadas de nuestro proyecto, el llenado del tanque se realizó con agua y colorantes para demostrar el nivel. No se trabajó con otros tipos de líquidos con densidades mayores o menores a 1.000 Kg/m^3 .

La descripción del plan de prueba y de los casos de uso se especifica con más detalle a continuación en la tabla 4.

Tabla 4 Pruebas y Casos de Usos asociados

Nombre de Prueba	Caso de USO
Pruebas del Sensor	✓ Trasmitiendo información Sensor -Tarjeta
Pruebas del Acondicionador de Señal	✓ Ningún caso de Uso Asociado
Pruebas del PIC16f873	✓ Conversión de señal analógica a digital ✓ Comunicación serial por modulo USART
Prueba de la interfaz serial con MAX232	✓ Recepción de datos a la Tarjeta de Adquisición
Pruebas de entrada de datos al PC	✓ Registrar de datos de llenado de tanque
Pruebas de modelado en 3D y salida de datos	✓ Iniciar llenado del tanque ✓ Visualización tanque 3D

Pruebas de las funciones de membresía para el control de la bomba	<ul style="list-style-type: none"> ✓ Ejecución de las Funciones de Membresía. ✓ Ejecución de Modo Automático ✓ Ejecución de Modo Emergente
---	--

Caso de Uso:	Trasmitiendo información Sensor –Tarjeta
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la transmisión de datos analógicos entre el sensor y la tarjeta	
Actores: Operador.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ol style="list-style-type: none"> 1. Se transmite con éxito la señal analógica 2. Los voltajes de la señal analógica del sensor coincide con el nivel deseado. 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ol style="list-style-type: none"> 1. No se transmitieron los datos exitosamente por sensor dañado 2. No se transmitieron los datos porque no se energizó la tarjeta 3. No se transmitieron los datos por falla del microcontrolador 4. No se transmitieron los datos debido a un componente defectuoso 5. No se transmitieron los datos debido a un ruido externo o interferencia electromagnética 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

Caso de Uso:	Conversión de señal analógica a digital
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la conversión de señal analógica a digital	
Actores: Tarjeta de Adquisición de datos.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ul style="list-style-type: none"> • Se transmite con éxito la señal digital a analógica 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ul style="list-style-type: none"> • No se transmitieron los datos exitosamente por sensor dañado 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

Caso de Uso:	Comunicación serial por módulo USART
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la comunicación serial a través del módulo USART en modo transmisión por el pin TX del PIC	
Actores: Tarjeta de Adquisición de datos.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ul style="list-style-type: none"> • Se transmite con éxito los datos digitales hacia la interfaz RS232 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ul style="list-style-type: none"> • No se transmitieron los datos exitosamente por sensor dañado • No se transmitieron los datos exitosamente por mala configuración del 	

módulo USART en el firmware
Resultados: Los datos son enviados al PIC y convertidos a formato digital

Caso de Uso:	Recepción de datos a la Tarjeta de Adquisición
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la comunicación serial a través del módulo USART en modo recepción por el pin RX del PIC	
Actores: Tarjeta de Adquisición de datos.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ul style="list-style-type: none"> • Se recibe con éxito los datos digitales en sus dos únicas tramas \$IF , y \$OF 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ul style="list-style-type: none"> • No se recibieron los datos exitosamente por desconexión de cable serial • No se recibieron los datos exitosamente por mala configuración del módulo USART en el firmware. 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

Nombre:	Registrar de datos de llenado de tanque
Autor:	Andy González/Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza el registro en la base de datos la información referente a: tiempo estimado de llenado, fecha de llenado, hora, minuto y segundo de llenado, el código del tanque, y la altura a la que llegó el líquido.	
Actores: Operador.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenarios Exitosos: 1. Se registró con éxito la información sobre el llenado del tanque	
Escenario Alternativo: No hay	
Escenarios no Exitoso 1. No se pudo registrar la información debido a problemas de conexión a la base de datos 2. No se pudo registrar la información debido a falta de datos en los campos de uno de los formularios de ingreso de información. 3. No se pudo registrar la información debido a desconexión del PC o sobrecarga de energía. 4. No se pudo registrar la información debido a un colapso de procesos en el PC	
Resultados: Los datos son enviados de formato digital al puerto serie del Computador	

Caso de Uso	Iniciar llenado del tanque
Autor:	Andy González /Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza el llenado del tanque, cuando el operador ejecuta la orden de iniciar el proceso de llenado.	
Actores: Operador.	
Precondiciones: El operador debe haber conectado la tarjeta , haber conectado el cable serial de la tarjeta al PC y además haber configurado el puerto com. Llenar los datos de configuración inicial.	
Escenario Exitoso: Se transmite con éxito los datos a intervalos de tiempo Se inició correctamente el llenado del tanque.	
Escenario Alternativo: No hay	
Escenarios no Exitoso No se inició el llenado del tanque por falta de datos en el formulario. No se inició el llenado del tanque por mala configuración del puerto com No se inició el llenado del tanque porque se desconectó el puerto com No se inició el llenado del tanque, no está energizado el pc. No se inició el llenado del tanque debido a un ruido externo o interferencia electromagnética. No se inició el llenado del tanque debido a no encontrar conexión a la base de datos.	
Resultados: Los datos enviados por PIC los recibe el computador con éxito.	

Caso de Uso:	Visualización tanque 3D
Autor:	Andy González – Jorge López
Fecha:	Agosto/2010
Descripción: Se visualiza el tanque en 3D cuando el operador de la orden de mostrar la gráficas.	
Actores: Operador.	
Precondiciones: <ol style="list-style-type: none"> 1. El operador debe haber conectado la tarjeta, haber conectado el cable serial de la Tarjeta al PC y además haber configurado el puerto com. 2. Haber establecido la conexión a la base. 3. Llenar datos de configuración inicial. 4. Tiempos determinados correctos. 5. Sensor en óptimas condiciones sin ruido o interferencias. 	
Escenario Exitoso: <ol style="list-style-type: none"> 1. Se transmite con éxito los datos a intervalos de tiempo 2. Se visualiza el tanque en 3D con éxito. 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ol style="list-style-type: none"> 1. No se visualizó el tanque en 3d debido a un ruido externo o interferencia electromagnéticas que sufrió el sensor. 2. No se visualizó el tanque en 3d debido a procesos colapsados por el Pc. 3. No se visualizó el tanque en 3d debido a conexión de la base falle. 4. No se visualizó el tanque en 3d debido a los tiempos no determinados. 	
Resultados: Se visualiza el tanque en 3D mostrando un sombreado de la altura de tanque.	

CAPITULO 4 IMPLEMENTACIÓN DEL DISEÑO

4.1 Diagrama de conexiones generales

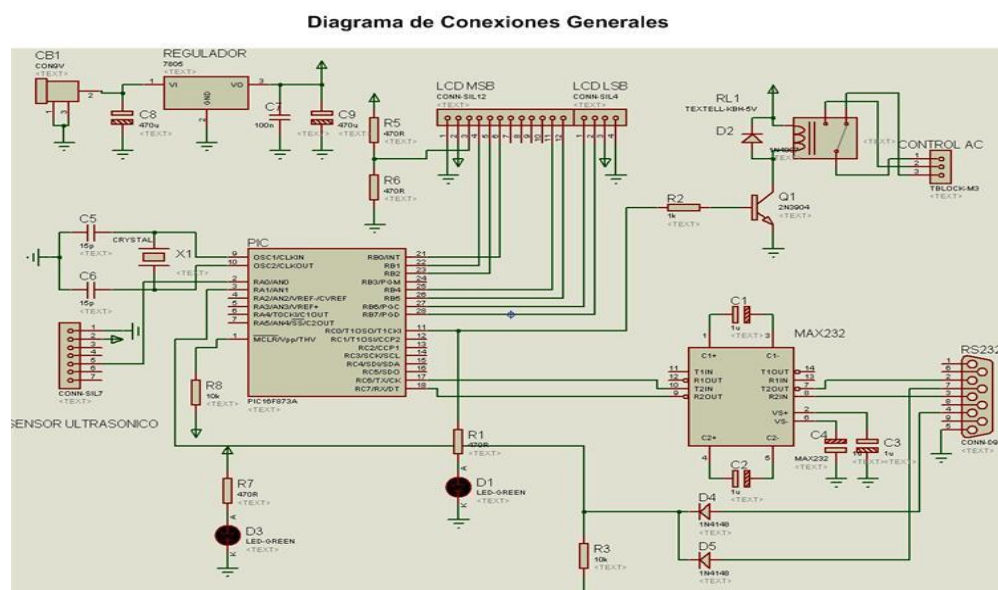


Figura 4.1 Diagrama de conexiones generales

En la figura 4.1 se observa el diseño de la tarjeta de adquisición de datos donde resalta el pic 16F873A el MAX232 la etapa reguladora de tensión y las conexiones con el relay.

PCB de la Tarjeta de Adquisición de Datos

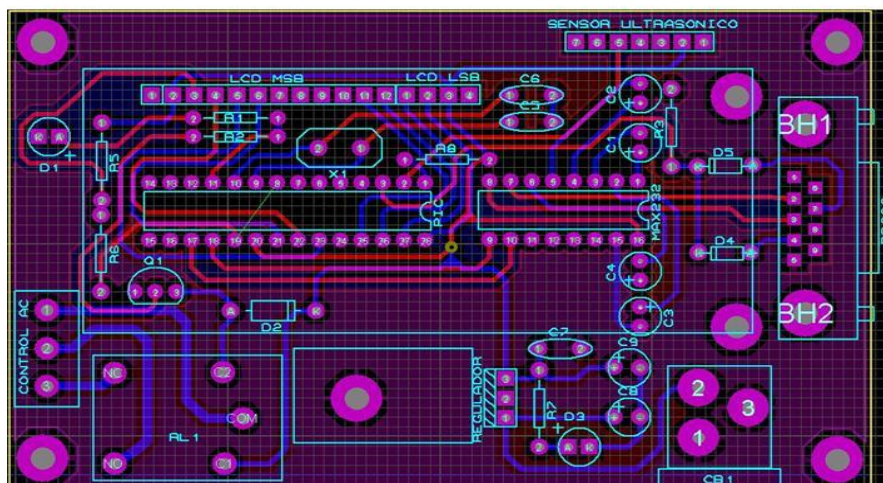


Figura 4.2 Pcb de la tarjeta de adquisición de datos

En la figura 4.2 el simulador Proteus presenta el diseño de la placa para la tarjeta de adquisición de datos cuyas dimensiones son 10 cm de ancho y 18 cm de largo.

Tarjeta de Adquisición de Datos

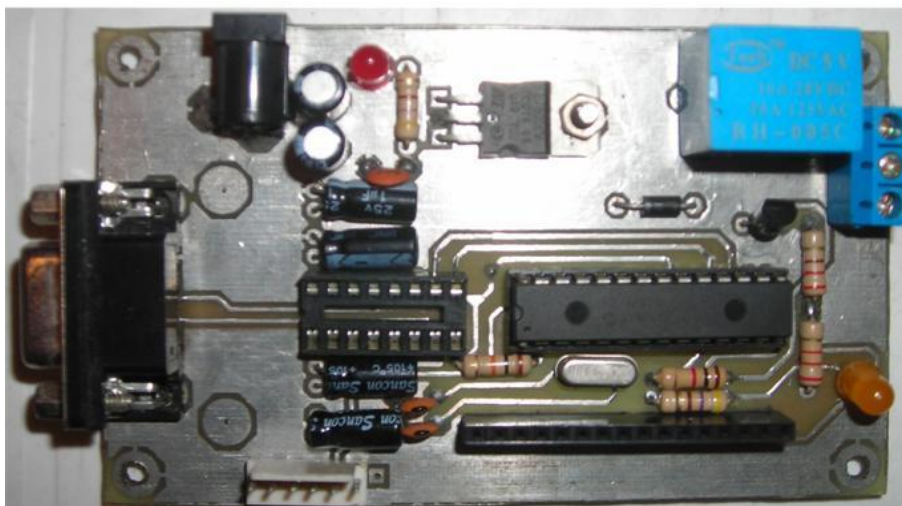


Figura 4.3 Tarjeta de adquisición de datos del proyecto

En la figura 4.3 se muestra la placa ya finalizada con sus componentes integrados resistencias, capacitores, diodos sin la instalación de la pantalla LCD y sin el sensor ultrasónico

Tarjeta de Adquisición con Sensor Instalado y LCD

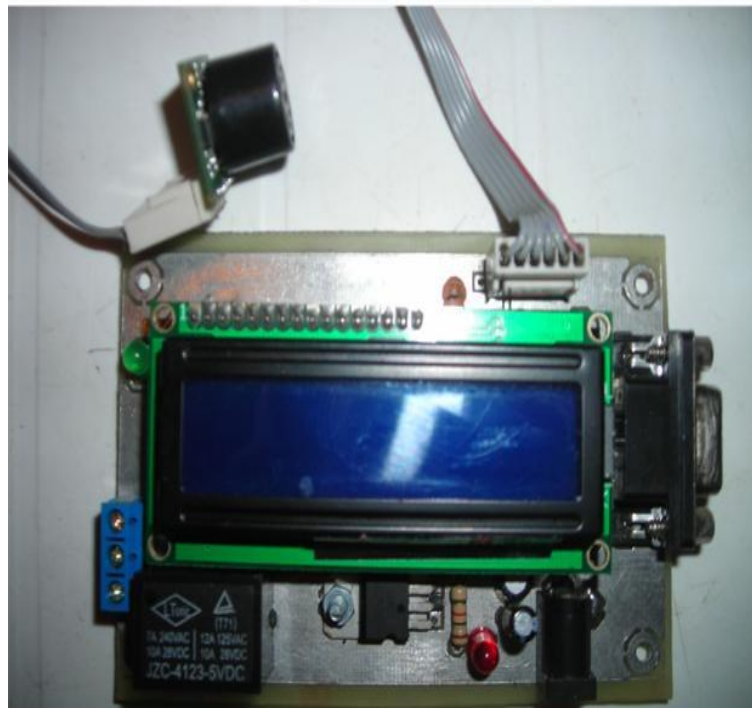


Figura 4.4 Tarjeta de adquisición de datos con sensor

En la figura 4.4 se observa la pantalla de cristal líquido LCD ya instalada en la tarjeta de adquisición de datos y el sensor ultrasónico en esta figura la tarjeta de adquisición de datos se encuentra sin alimentación de energía

4.2 Montaje de Tanque Sensor

Montajes de Tanque Sensor



Figura 4.5 Montaje de tanque sensor

Para el montaje del prototipo se necesitaron

- 2 tanques de plástico de capacidad para 20 litros
- 2 llaves de paso de $\frac{3}{4}$ de pulgadas
- 1 metro de manguera de $\frac{1}{4}$ de pulgada para las llaves de paso
- 2 bombas DC de limpiaparabrisas
- 1 fuente DC a 3 amperios

En la figuras 4.6 ,4.7, 4.8 se muestra diversas perspectivas del prototipo con el sensor

Montaje Tanque Sensor

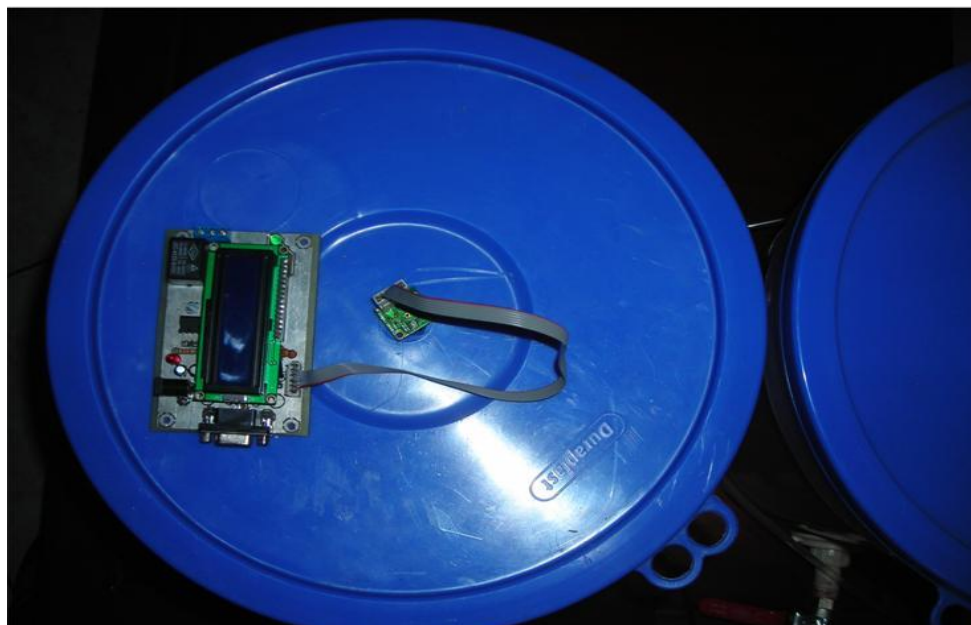


Figura 4.6 Montaje de tanque sensor

Montaje Tanque-Sensor



Figura 4.7 Montaje de tanque sensor

Montaje Tanque-Sensor

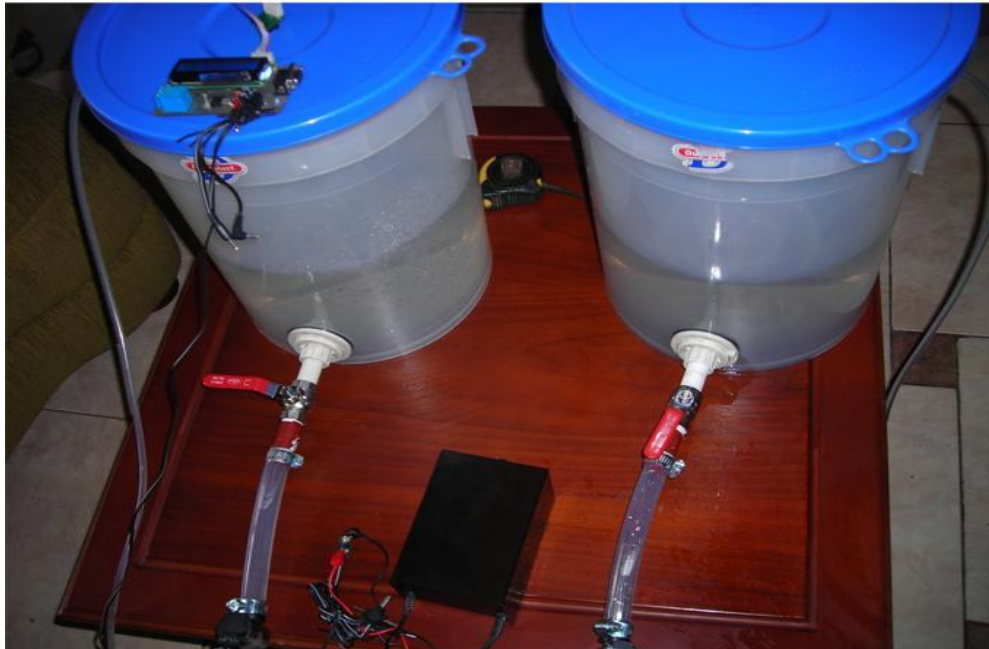


Figura 4.8 Montaje de tanque sensor

4.3 Esquema electrónico del sensor

Esquema Electrónico del Sensor

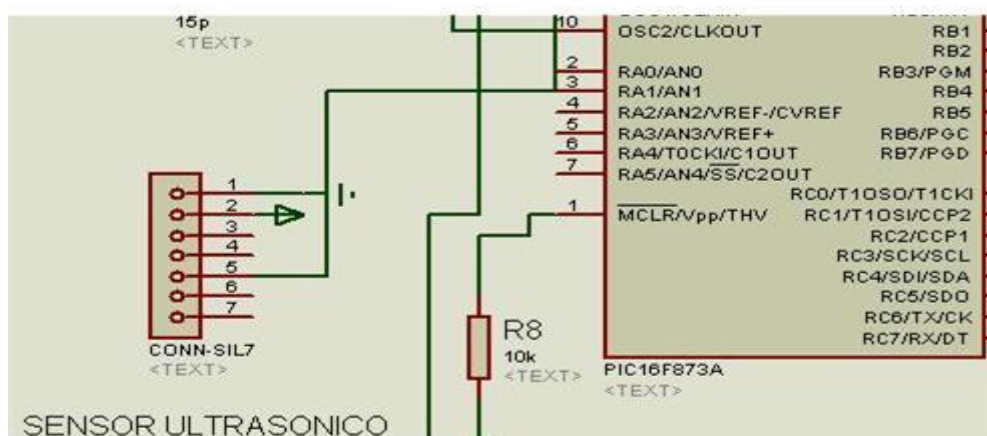


Figura 4.9 Esquema electrónico del sensor

En la figura 4.9 se muestra la etapa donde se instala el sensor; la salida analógica de la misma se conecta directamente a la entrada AN0 del microcontrolador. El sensor se alimenta con 5V y una conexión a tierra

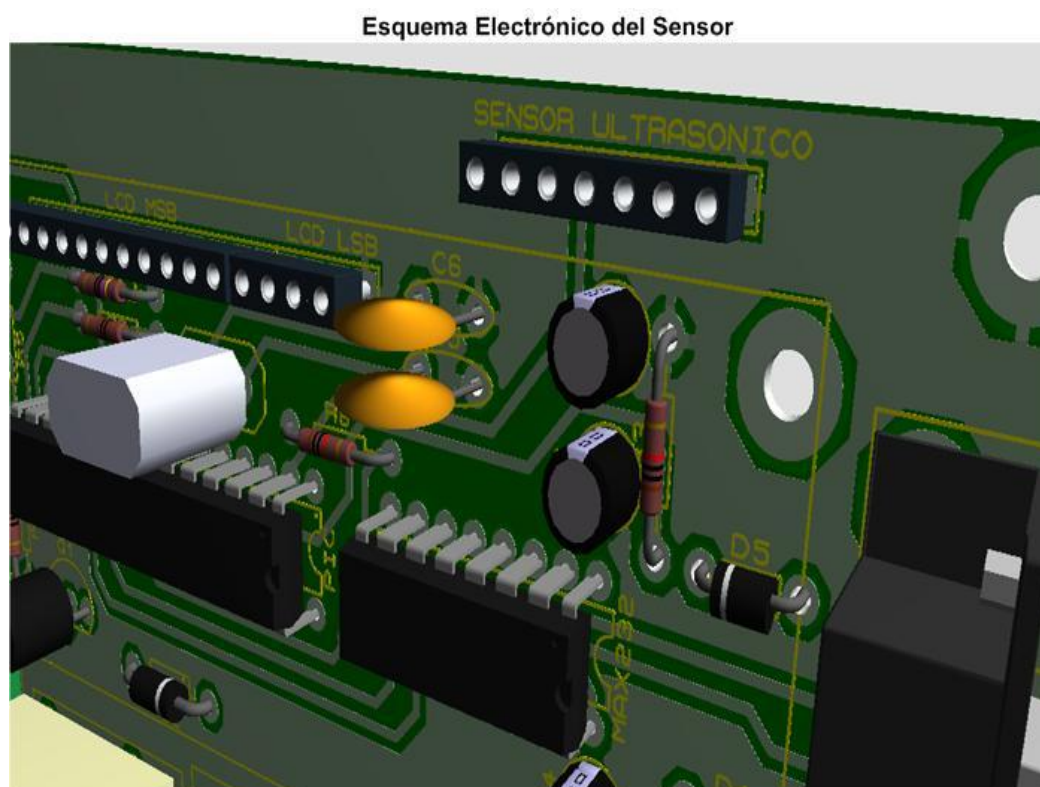


Figura 4.10 Esquema electrónico del sensor

Antes de llevarlo a placa, se realizó un diseño en vista previa en 3D en Proteus. En la vista se muestra las pistas que conectan con los integrados y en donde estará ubicado el sensor.

Ver figura 4.10.

En la figura 4.11 queda ya instalado el sensor, para la conexión del mismo se usó un pequeño bus de datos de 6 líneas para transmitir las señales.

Esquema Electrónico del Sensor

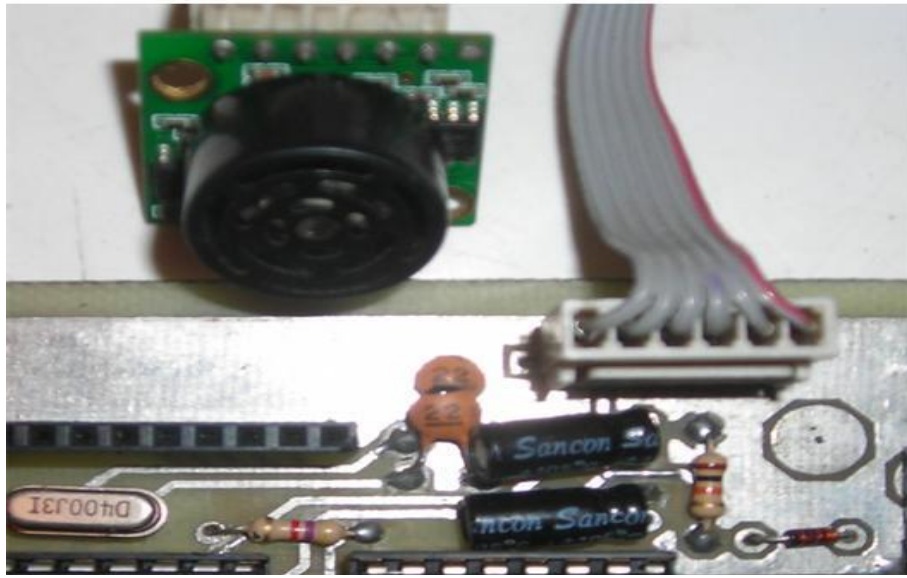


Figura 4.11 Esquema electrónico del sensor

4.4 Esquema electrónico de la etapa de acondicionamiento

Puesto que las prestaciones del PIC y la del sensor permitieron la no utilización de ninguna etapa de acondicionamiento, no hubo necesidad de emplear op-amps para este proyecto.

4.5 Esquema electrónico del PIC16F873A

Esquema electrónico del PIC16F873A

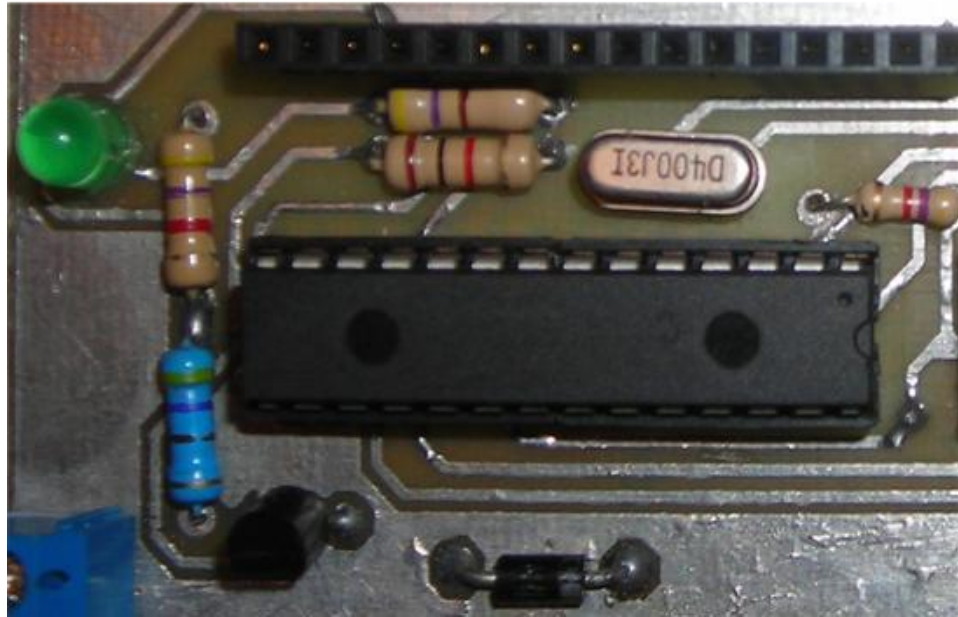


Figura 4.12 Esquema electrónico del PIC

En la figura 4.12 se muestra más de cerca la etapa donde se encuentra el pic los componentes que se hayan acoplados al pic son:

- 1 cristal de cuarzo de 4 MHz
- 2 Capacitores de 15 pico Faradios
- 1 resistencia de 10kOhm

4.6 Esquema electrónico de la Interfaz de comunicación serial

Esquema de la Interfaz Serial

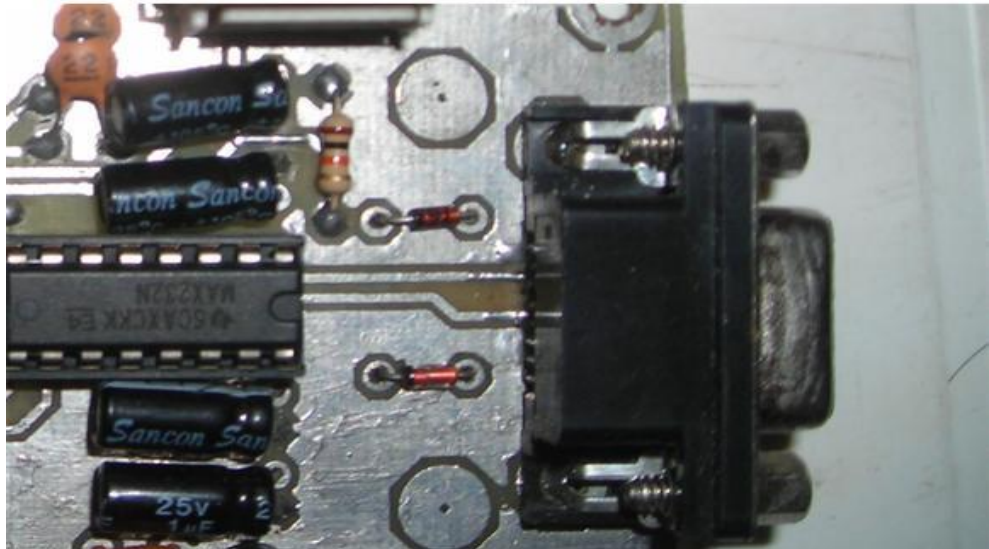


Figura 4.13 Esquema electrónico del Pic

En la figura 4.13 se muestra el esquema electrónico, ya implementado en la tarjeta de adquisición, donde se destaca la presencia de microcontrolador 16f873.

4.7 Esquema grafico del tanque 3D

En este esquema mostaremos las imágenes de cómo el tanque 2 llena el tanque 1 en la siguiente figura 4.14 a) b) y c)

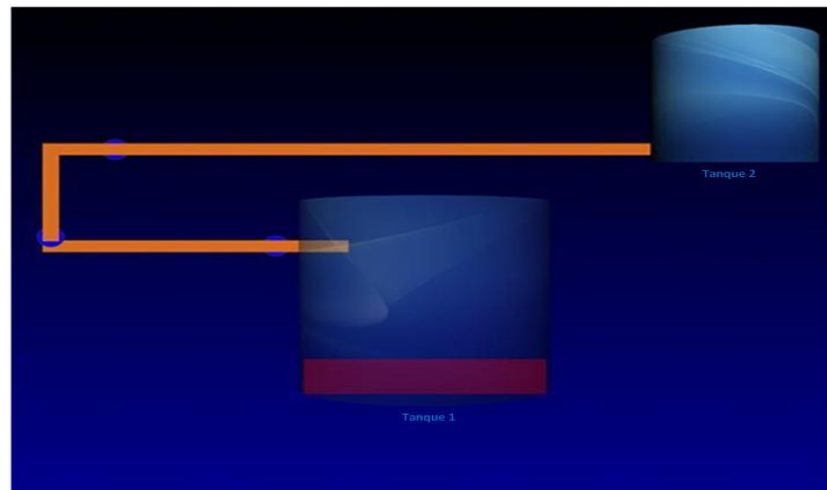


Figura 4.14a Esquema grafico del tanque

En la figura 4.14a se observa como el tanque 2 llena el tanque 1 en $t= 52$ segundos, y vemos el sombreado que se muestra en el tanque 1, demostrando el llenado hasta el determinado tiempo.

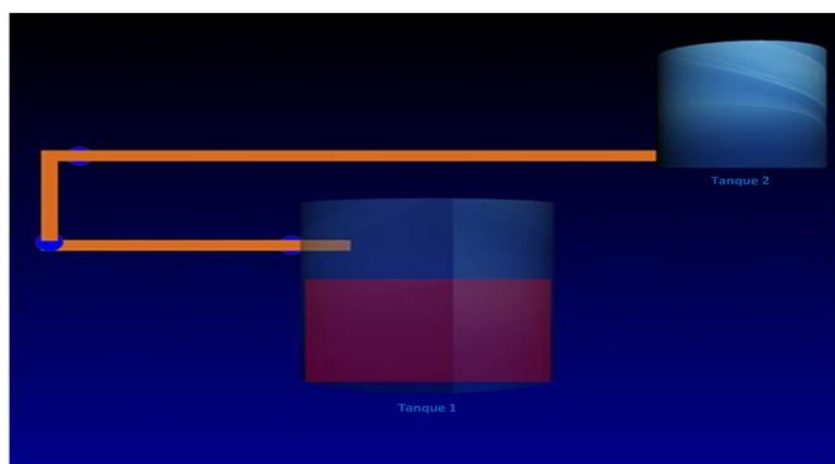


Figura 4.14b Esquema grafico del tanque

En la figura 4.14b se observa como el tanque 2 llena el tanque 1 en $t= 2,48$ minutos, y vemos el sombreado que se muestra en el tanque 1, demostrando su llenado hasta ese determinado tiempo.

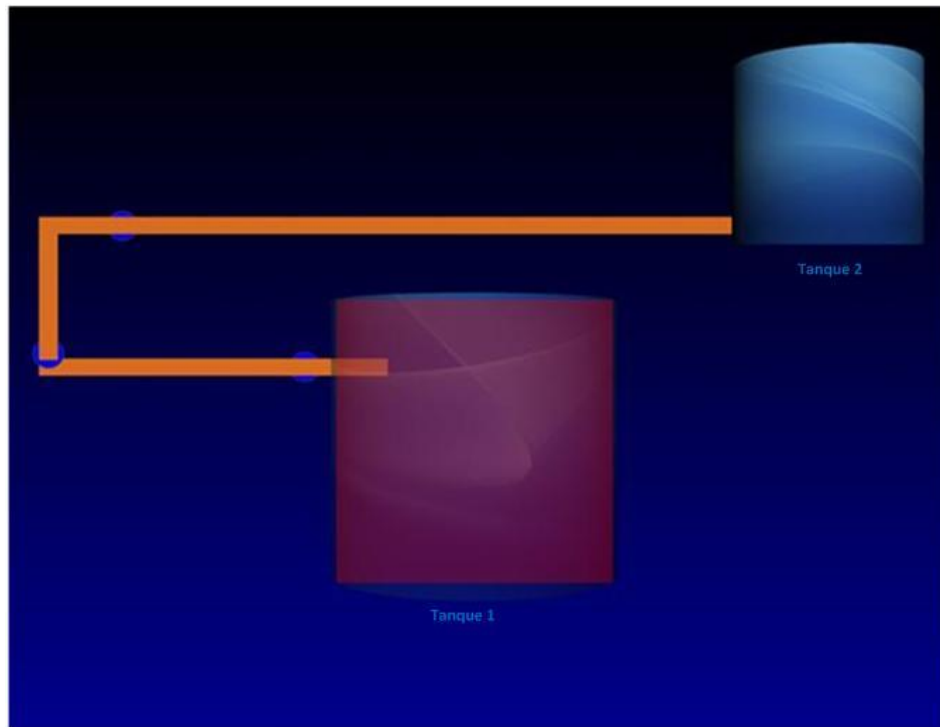


Figura 4.14c Esquema grafico del tanque

En la figura 4.14c se observa como el tanque 2 a llenado por completo el tanque 1 en $t= 5,18$ minutos, y también el sombreado que se muestra en el tanque 1, demostrando el llenado total en ese determinado tiempo.

4.8 Interfaz

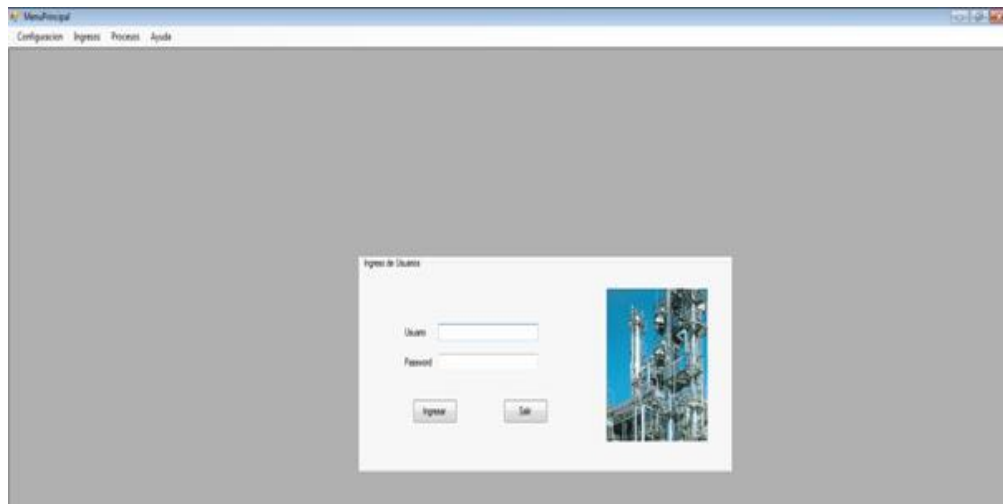


Figura 4.15 Interfaz inicial

En la figura 4.15 mostramos la interfaz inicial, el ingreso de usuario que permite el acceso aquellos operadores que tengan permiso de manejar el sistema.



Figura 4.16 Configuración inicial del sistema

En la figura 4.16 se muestra la configuración inicial del sensor, como su nombre lo indica, configura el puerto COM con la cual trabajara la tarjeta de adquisición y el envío de datos al PC.



The screenshot shows a software window titled "RegistroTanque" with a sub-header "Ingreso de Datos". The interface includes several input fields and controls:

- NoTanque**: A text input field.
- Diametro**: A text input field with the unit "Cm" to its right.
- Altura**: A text input field with the unit "Cm" to its right.
- Radio**: A text input field with the unit "Cm" to its right.
- Volumen**: A text input field with the unit "Cm" to its right.
- Material**: A text input field with the unit "Cm^3" to its right.
- Presion MAX**: A text input field with the unit "Cm" to its right.
- Tipo de Liquido**: A dropdown menu currently showing "Agua" with the unit "Dina/Cm^2" to its right.
- ID**: A small text label below the "Tipo de Liquido" dropdown.
- Tiempo Aprox de llenado**: A text input field at the top right.
- Unidades Medidas**: A section containing a radio button labeled "Metros".
- Registrar** and **Cancelar**: Two buttons at the bottom right.
- Image**: An illustration of three cylindrical tanks of different sizes and orientations.

Figura 4.17 Ingreso de datos del tanque

En la figura 4.17 se observa el ingreso de los datos del tanque, en esta pantalla se obtienen todas las características importantes del tanque como lo son: altura, diámetro y volumen que van hacer registradas en la base de conocimiento ("base de datos").

4.8.1 Pantalla principal

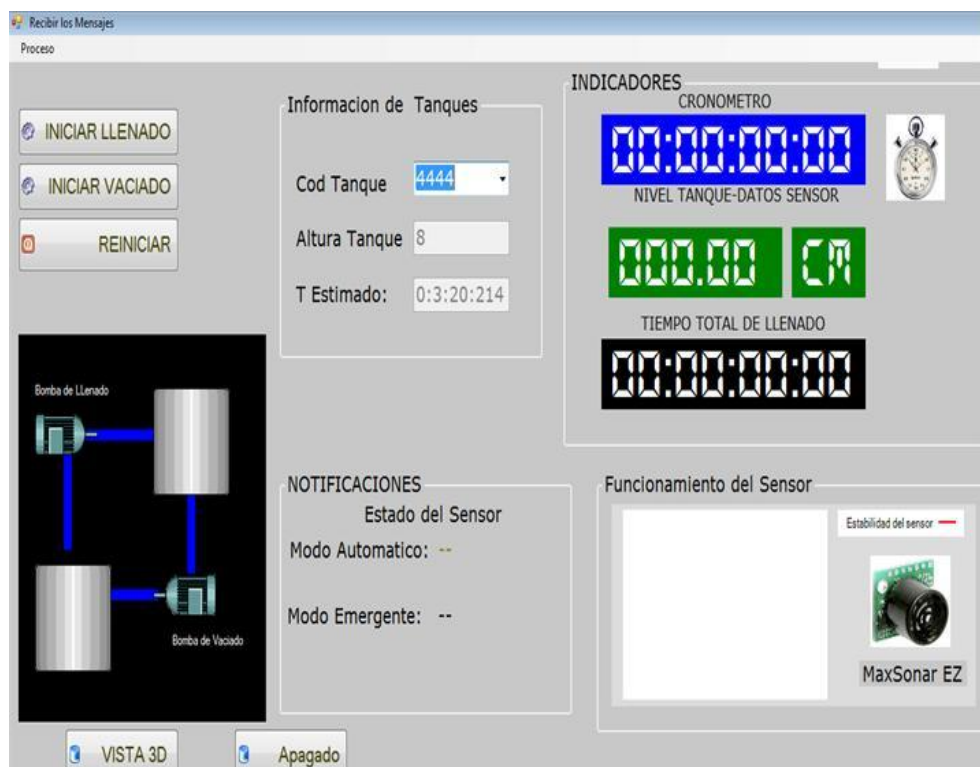


Figura 4.18 Pantalla principal

4.8.2 Controles principales

En la fig. 4.19 se muestra el bloque de la información del tanque, se realiza una búsqueda del código del tanque, mostrando su altura y el tiempo estimado de llenado.

Información de Tanques

Cod Tanque 4444

Altura Tanque 8

T Estimado: 0:3:20:214

Figura 4.19 Información del tanque

En la figura 4.20 se muestra el bloque Indicadores, el cronómetro interno, el nivel del tanque por medio del sensor, y el tiempo de llenado total que se toma al terminar este proceso.



Figura 4.20 Indicadores del sistema

En la figura 4.21 se muestra el bloque de las notificaciones, este bloque muestra los tipos de modos que se pueden cambiar en la ejecución del proyecto. Los modos hablados en el capítulo 2, si en algún momento el sensor falla o proceso se queda inactivo estos se activarán indicando que opción cogió.

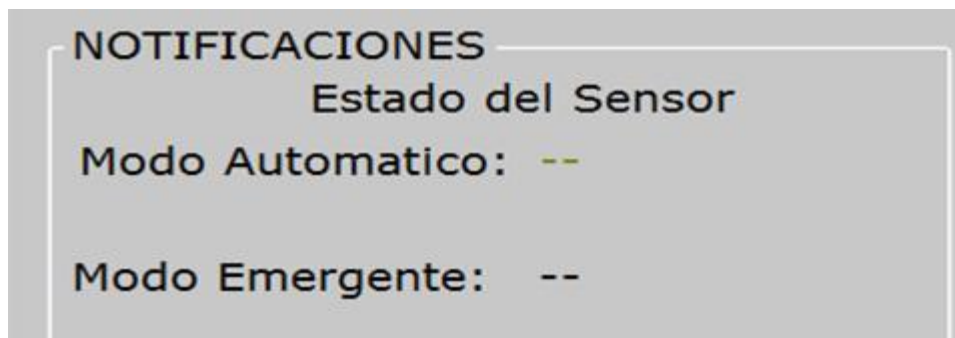


Figura 4.21 Notificaciones del sistema

En la figura 4.22 se muestra el bloque del funcionamiento del sensor, este bloque nos indica como está trabajando el sensor por medio de una gráfica, indicando si el sensor está en buenas condiciones o no al momento de obtener datos del nivel.



Figura 4.22 Funcionamiento grafico del sensor

4.9 Esquema eléctrico -electrónico bomba de agua

Esquema Eléctrico Electrónico

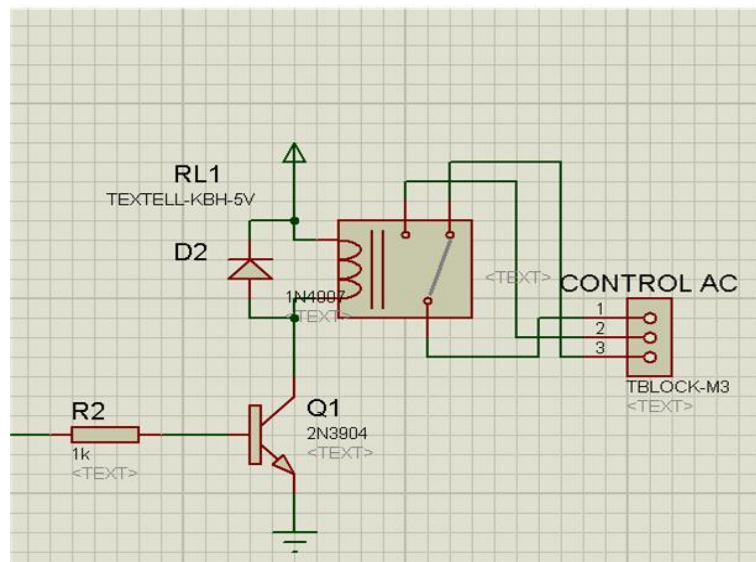


Figura 4.23 Esquema eléctrico electrónico

Para el esquema electrónico se implementó el envío de un pulso enviado a través del pic para active en un nivel alto un Relé de 5V de alimentación para que funcione de paso con las fuente de corriente AC.

Esquema Electrico-Electronico

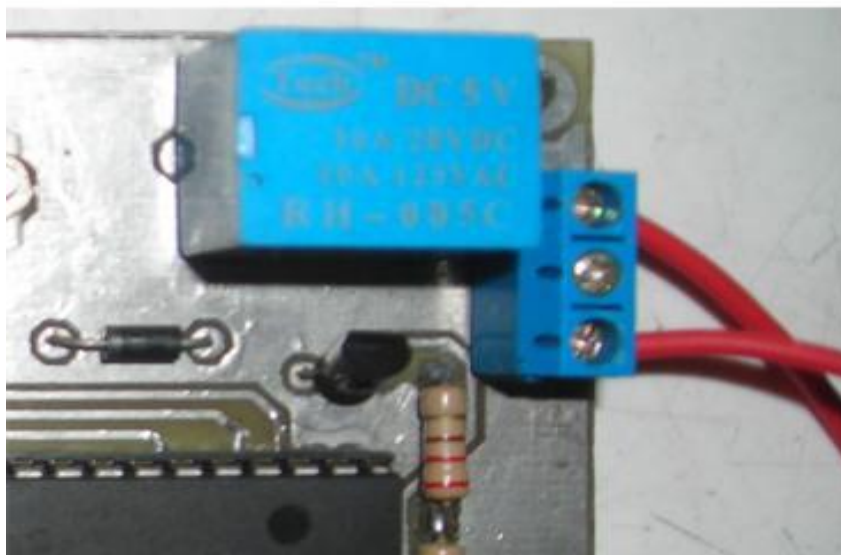


Figura 4.24a Esquema eléctrico electrónico

Esquema de Conexiones con Bomba

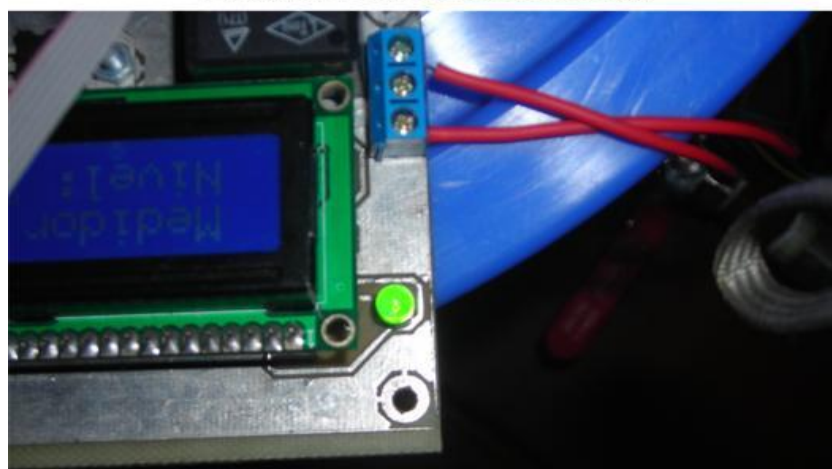


Figura 4.24b Esquema de conexiones de bomba

Conexiones con Bomba DC

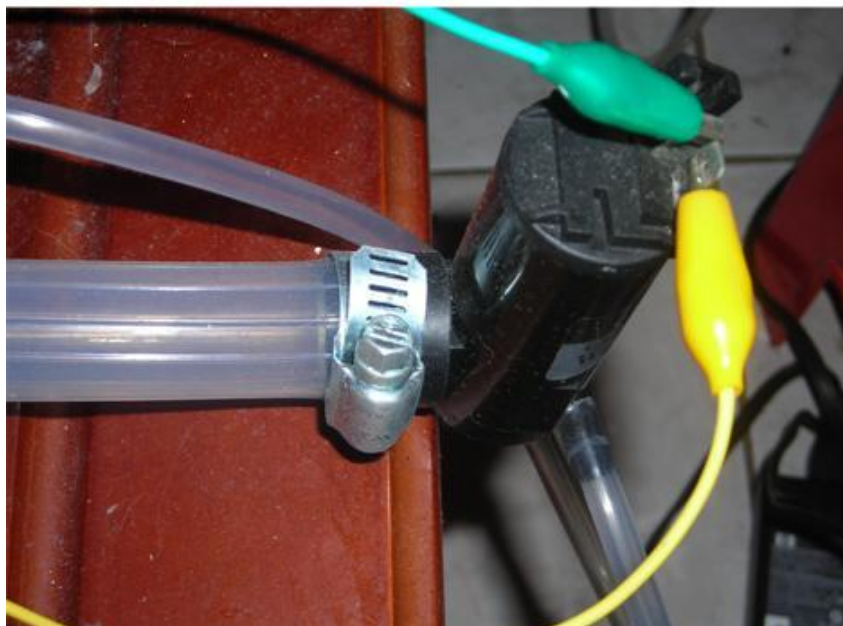


Figura 4.24c Conexiones de bomba Dc

Conexiones de Bombas



Figura 4.24d Conexiones de bomba DC

4.10 Seguridad electrónica

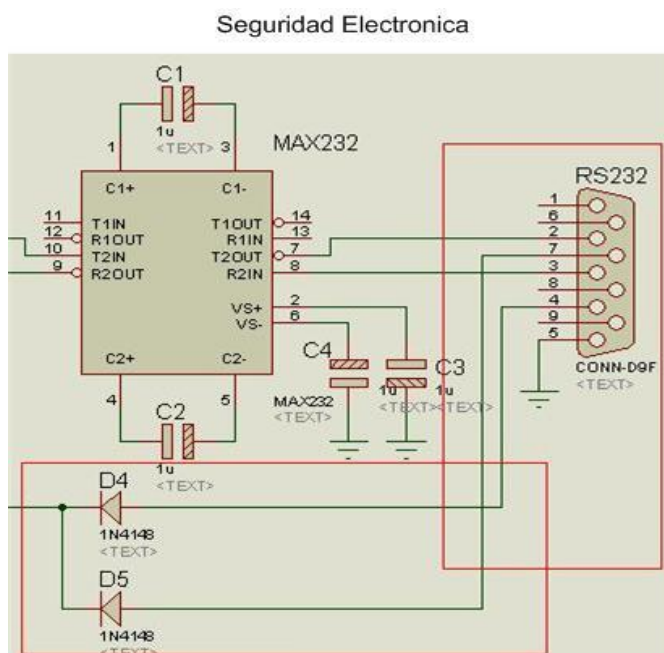


Figura 4.25 Seguridad electrónica

Cuando hablamos de la seguridad electrónica nos referimos al hecho de tomar las precauciones al momento de estar conectada la bomba a la tarjeta de adquisición de datos, hemos diseñado un mecanismo mediante el cual si por circunstancias extremas llegase a ocurrir una desconexión a nivel físico por ejemplo un tropiezo con el cable, colocamos en los pines de alimentación del RS232, 2 diodos que están cargados como circuito cerrado si se desconecta la alimentación hace que cambie la polaridad y actúe como circuito abierto esto hará mediante el firmware una desconexión o apagado automático del sistema

Sección del Firmware que apaga la Bomba en caso de Emergencia

```
}  
if (input (PIN_A1)==0)//Pregunto si esta desconectado el cable serial o si esta cerrado el puerto COM  
{  
  output_low (PIN_CO); //Apago la bomba  
}  
}
```

Figura 4.26 Sección de Firmware de apagado de bomba

Ubicación de Diodos que apagan la Bomba

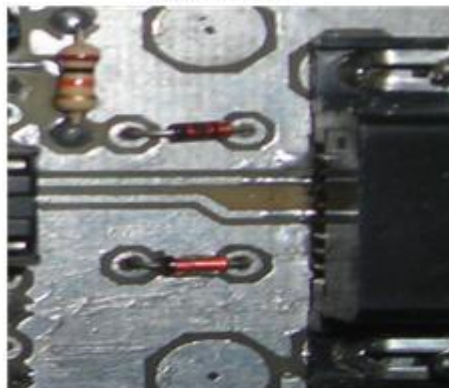


Figura 4.27 Ubicación de diodos

4.11 Herramientas de desarrollo de hardware, software y Plataforma de programación

4.11.1 Herramientas de simulación de Hardware

4.11.2 Entorno de desarrollo para programación en pic

4.11.3 Entorno de desarrollo .net

Para el desarrollo del proyecto se necesitaron de tres entornos de desarrollo y son

Ver referencia en

- [8]Proteus de Labcenter Electronics
- [9]PCHWD COMPILER de CCS inc.
- [10]Visual Studio. Net 2008 con SQL Server 2005 de Microsoft

CAPITULO 5 PRUEBAS

Plan de Pruebas

Para lograr nuestro objetivo, que el llenado del tanque cumpla un 98% de exactitud en calibración, visualización, control y monitoreo hemos realizado un plan de pruebas evaluandolos casos de uso que intervienen en el proyecto.

Lista de pruebas a evaluar:

- ✓ Pruebas del Sensor
- ✓ Pruebas del Acondicionador de Señal
- ✓ Pruebas del PIC16f873
- ✓ Prueba de la interfaz serial con MAX232
- ✓ Pruebas de entrada de datos al PC
- ✓ Pruebas de modelado en 3D y salida de datos
- ✓ Pruebas de las funciones de membresía para el control de la bomba.

Para las pruebas de nuestro proyecto, utilizamos los siguientes materiales:

- ✓ 2 tanques plásticos de aproximadamente 50 cm de alto. Para demostrar el nivel de líquido ingresado.
- ✓ 2 llaves de paso (3/4 pulgadas). Para controlar el paso del líquido al tanque.
- ✓ 2 Bombas DC 12v, 3A. especificación de los motores para trabajar de manera normal.
- ✓ 1 Tarjeta de Adquisición de datos. Para el envío y recepción de información.

Las pruebas realizadas de nuestro proyecto, el llenado del tanque se realizó con agua y colorantes para demostrar el nivel. No se trabajó con otros tipos de líquidos con densidades mayores o menores a 1.000 Kg/m^3 .

La descripción del plan de prueba y de los casos de uso se especifica con más detalle a continuación en la tabla 4.

Tabla 4 Pruebas y Casos de Usos asociados

Nombre de Prueba	Caso de USO
Pruebas del Sensor	✓ Trasmitiendo información Sensor –Tarjeta
Pruebas del Acondicionador de Señal	✓ Ningún caso de Uso Asociado
Pruebas del PIC16f873	<ul style="list-style-type: none"> ✓ Conversión de señal analógica a digital ✓ Comunicación serial por modulo USART
Prueba de la interfaz serial con MAX232	✓ Recepción de datos a la Tarjeta de Adquisición
Pruebas de entrada de datos al PC	✓ Registrar de datos de llenado de tanque
Pruebas de modelado en 3D y salida de datos	<ul style="list-style-type: none"> ✓ Iniciar llenado del tanque ✓ Visualización tanque 3D
Pruebas de las funciones de membresía para el control de la bomba	<ul style="list-style-type: none"> ✓ Ejecución de las Funciones de Membresía. ✓ Ejecución de Modo Automático ✓ Ejecución de Modo Emergente

5.1 Pruebas del sensor



Figura 5.1 Pruebas del sensor

El caso de uso relacionado a esta prueba es el de Transmitiendo información Sensor – Tarjeta.

Caso de Uso:	Trasmitiendo información Sensor –Tarjeta
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la transmisión de datos analógicos entre el sensor y la tarjeta	
Actores: Operador.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ol style="list-style-type: none"> 3. Se transmite con éxito la señal analógica 4. Los voltajes de la señal analógica del sensor coincide con el nivel deseado. 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ol style="list-style-type: none"> 6. No se transmitieron los datos exitosamente por sensor dañado 7. No se transmitieron los datos porque no se energizo la tarjeta 8. No se transmitieron los datos por falla del microcontrolador 9. No se transmitieron los datos debido a un componente defectuoso 10. No se transmitieron los datos debido a un ruido externo o interferencia electromagnética 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

Tabla 5 Resultados de voltajes medidos

	Voltajes Medidos (Voltios)							
	0.05	0.50	0.75	1	1.25	1.85	2.55	
Resultados En CM	9.90	124.98	189.27	254.3	318.87	468.63	647.32	
	9.93	124.89	188.63	254.36	318.23	468.52	647.53	
	9.95	124.69	189.36	255	318.96	468.56	647.63	
	9.90	124.96	190.36	256.36	319.63	468.99	647.35	
	9.95	124.96	190.25	257.36	319.00	468.96	647.32	
	9.97	124.77	189.56	255.36	318.36	468.63	647.33	
	9.98	124.85	189.54	254.26	319.63	468.63	647.36	
	9.50	124.96	189.36	254.96	318.23	468.63	647.55	
	9.30	124.96	189.36	254.96	318.96	468.99	647.32	
	9.90	124.99	189.99	254.32	318.96	468.33	647.32	
	9.95	124.36	189.89	255.36	318.63	468.56	647.52	
	9.96	124.70	188.99	254.33	318.12	468.51	647.11	
	9.97	124.96	189.63	254.96	318.23	468.25	647.01	
	9.98	124.99	189.23	255.36	318.45	468.23	647.21	
	9.99	125.00	189.35	254.96	318.63	468.25	647.25	
	9.20	129.65	189.36	255.36	318.96	468.25	647.32	
	9.30	128.33	189.45	254.96	318.56	468.32	647.32	
	9.96	129.65	189.56	254.36	318.63	468.56	647.23	
	Promedio	9,81	125,5	189,5	255,09	318,72	468,5	647,33
	Desviación Standard	0,2	1,69	0,41	0,79	0,44	0,24	0,15

Hemos realizado 18 muestras para estimar el promedio de nivel en cm. de acuerdo a cada medición de voltaje con su respectiva desviación estándar, el resultado de esta prueba se detalla en la tabla 5 y 6.

Tabla 6 Resultados Voltaje vs Nivel

Voltaje analógico (Voltios)	Nivel Observado(cm)
0.05	9.81 ± 0.2
0.50	125.5 ± 1.69
0.75	189.5 ± 0.41
1	255.09 ± 0.79
1.25	318.72 ± 0.44
1.85	468.5 ± 0.24
2.55	647.33 ± 0.15

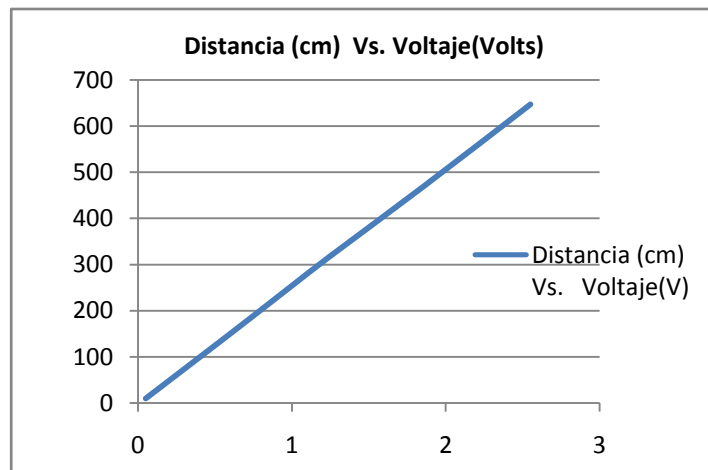


Figura 5.2 Grafica lineal Distancia en cm Vs Voltaje Analógico

Los resultados de esta pruebas demuestran la linealidad que existe entre el voltaje y el nivel medido esto se da por lo descrito en el capítulo de diseño. (Ver ecuación 2 página 43).

5.2 Pruebas del acondicionador de señal

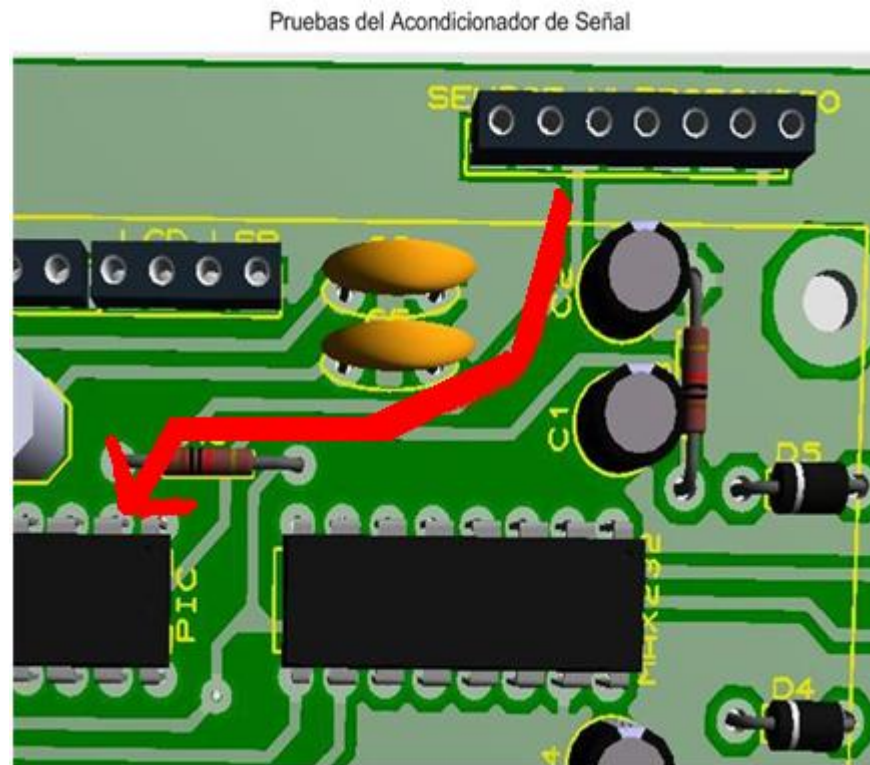


Figura 5.3 Pruebas de acondicionador del Pic

Como podemos ver en la figura 5.3 se ha conectado directamente la salida analógica del sensor a la entrada del PIC, como se explicó en la etapa de diseño no hubo la necesidad de acoplar una etapa de acondicionamiento de señal para amplificarla. Las condiciones eléctricas del PIC en sus entradas analógicas, permiten un rango de voltaje de 0 a 5V.

5.3 Pruebas del PIC16F873A

5.3.1 Pruebas del convertidor A/D

Para las pruebas del convertidor analógico digital y probar su efectividad, haremos uso del Hypertemirnal Para los niveles de tensión mostrados. También podemos acotar que para cada dato digital convertido, se ajustó una trama de caracteres por seguridad y protección, al momento de enviar un alto =1 o un bajo = 0 a cada línea de caracteres leídos del sensor se acopló por código en firmware los caracteres \$IdatoF. El caso de uso relacionado a esta prueba es la de convertir señal analógica a digital.

Caso de Uso:	Conversión de señal analógica a digital
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la conversión de señal analógica a digital	
Actores: Tarjeta de Adquisición de datos.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ul style="list-style-type: none"> • Se transmite con éxito la señal digital a analógica 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ul style="list-style-type: none"> • No se transmitieron los datos exitosamente por sensor dañado 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

Tabla7 Resultados de señal analógica de voltaje y salida por Hyperterminal

Voltaje analógico (Voltios)	Salida por Hyperterminal(cm)
0.50	\$I124.50F
0.75	\$I189.25F
1	\$I254.0F
2	\$I508.0F
2.55	\$I647.45F

Se ha considerado el rango de valores de la tabla 7 para las pruebas del convertidor A/D puesto que, la máxima distancia del sensor puede aceptar por especificaciones de fábrica hasta 6.4 metros máximo de distancia, con los respectivos voltajes medidos.

0.50V de entrada analógica

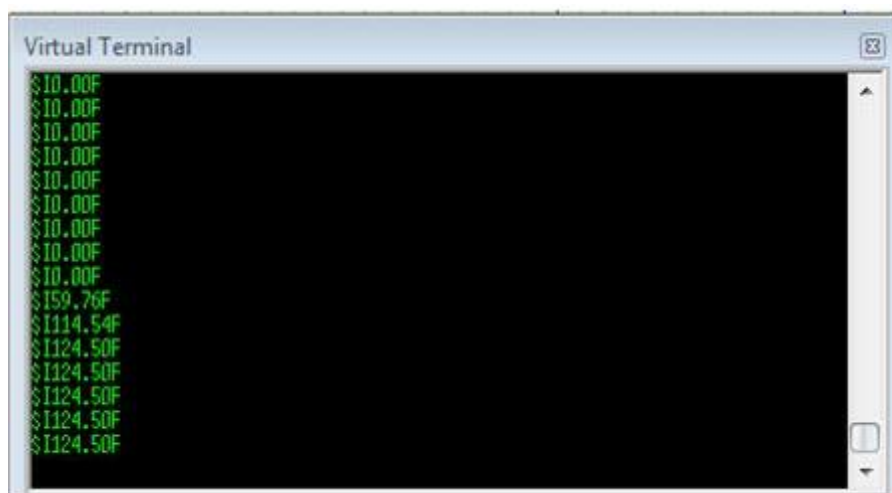


Figura 5.4a Salida por Hypertemirnal con 0.5V

0.75V de entrada analógica

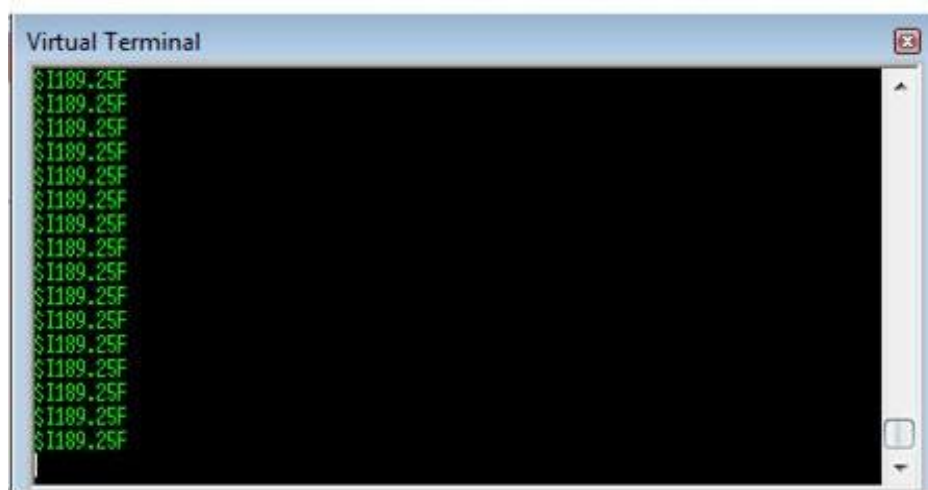


Figura 5.4b Salida por Hyperterminal con 0.75V

1V de entrada analógica

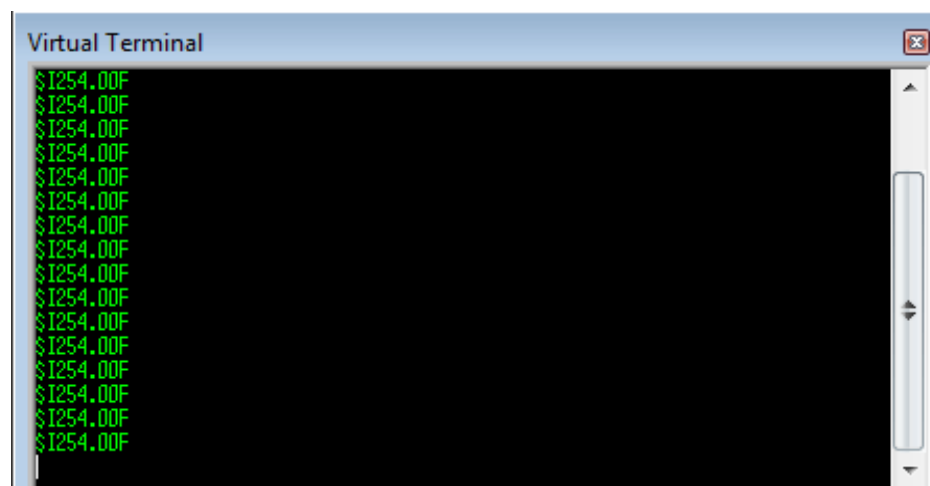
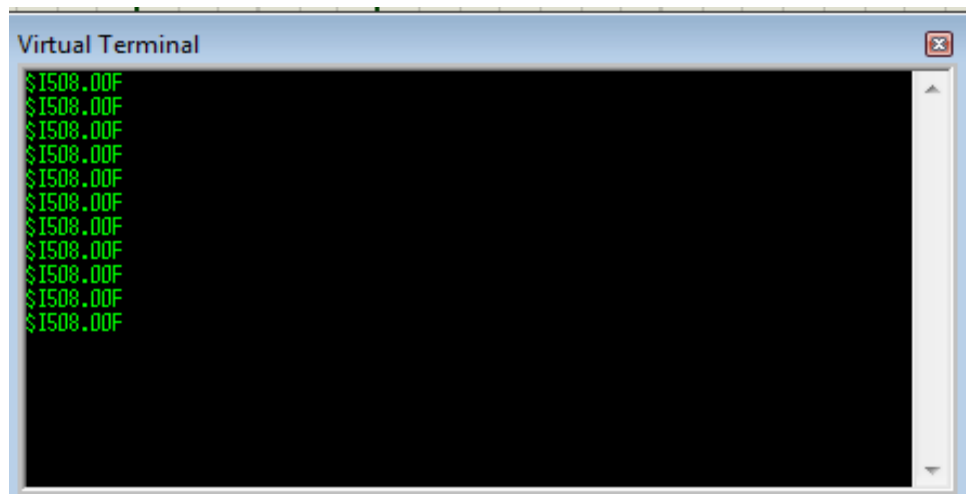


Figura 5.4c Salida por Hyperterminal con 1V

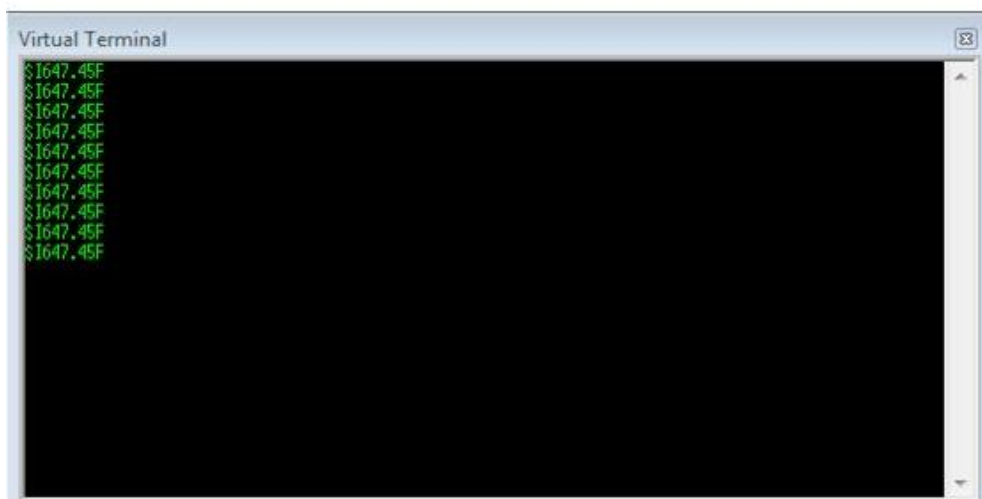
2V de entrada analógica



```
Virtual Terminal
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
$I508.00F
```

Figura 5.4d Salida por hyperterminal con 2V

2.55V de entrada analógica



```
Virtual Terminal
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
$I647.45F
```

Figura 5.4e Salida por hyperterminal con 2.55V

5.3.2 Pruebas de comunicación serial con USART

Se realizó a través de un osciloscopio virtual examinando las salidas Tx y Rx del PIC se pudo capturar la información sobre la trama de datos mostrada en el osciloscopio.

Caso de Uso:	Comunicación serial por módulo USART
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la comunicación serial a través del módulo USART en modo transmisión por el pin TX del PIC	
Actores: Tarjeta de Adquisición de datos.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ul style="list-style-type: none"> • Se transmite con éxito los datos digitales hacia la interfaz RS232 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ul style="list-style-type: none"> • No se transmitieron los datos exitosamente por sensor dañado • No se transmitieron los datos exitosamente por mala configuración del módulo USART en el firmware 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

Tabla 8 Resultados Señal analógica de voltaje y Salida digital a RS232

Voltaje analógico (Voltios)	Salida a RS232 en Hexadecimal	Distancia (cm)
0.50	24 49 31 32 34 2e 35 30 46	124.5
0.75	24 49 31 38 39 2e 32 35 46 0d 0a 0d 0a	189.2
1	24 49 32 35 34 2e 30 46 0d 0a 0d 0a	254.0
2	24 49 35 30 38 2e 30 46	508.0
2.55	24 49 36 34 37 2e 34 35 46	647.4

Por ejemplo para una lectura de tensión analógica de 1V el osciloscopio muestra en la señal una trama compuesta como se ve en la gráfica 5.5, para otros valores de voltaje podemos tomar de referencia la tabla 8.

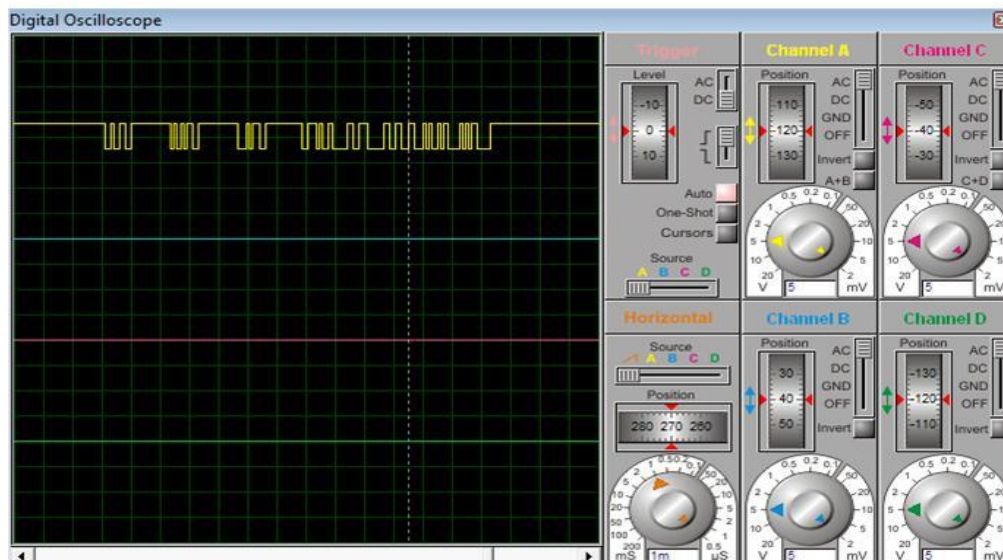


Figura 5.5 Resultado del osciloscopio a 1V

Salida por TX a 1 voltio su salida en el osciloscopio es Dato analógico: 1V convertido a $cm = 254$ Original \rightarrow 254 F . Codificado: 00100100 01001001 00110010 00110101 00110100 00101110 00110000 00110000 01000110

Salida por TX a 2.55 voltio su salida en el osciloscopio es:

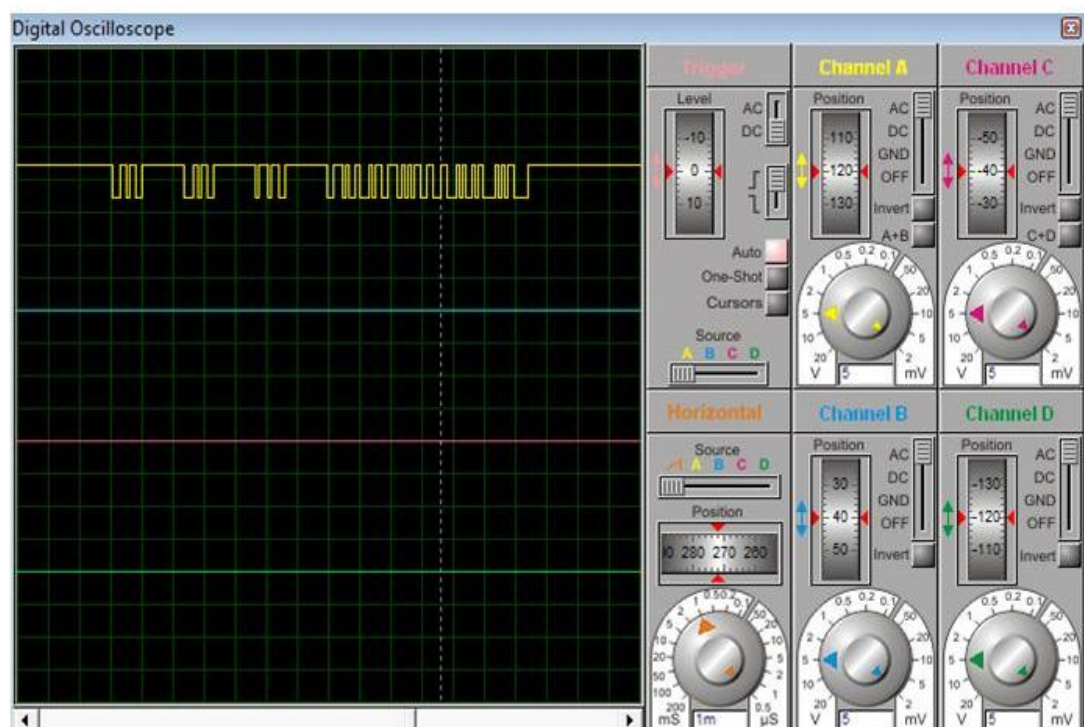


Figura 5.6 Resultado del osciloscopio a 2.55V

Dato analógico: 2.55V, convertido a $Cm = 647.45$. Dato original $= 647.45 \text{ F}$. Codificado: 00100100 01001001 00110110 00110100 00110111 00101110 00110100 00110101 01000110

5.4 Prueba de la interfaz serial con MAX232

Para las pruebas de la interfaz serial con el MAX232 se puso el osciloscopio de modo que se pueda capturar los dos tipos de trama que hacen que la bomba encienda o apague como son \$I1F y \$I0F, mostradas en las figuras 5.7 y 5.8.

Caso de Uso:	Recepción de datos a la Tarjeta de Adquisición
Autor:	Andy González-Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza la comunicación serial a través del módulo USART en modo recepción por el pin RX del PIC	
Actores: Tarjeta de Adquisición de datos.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenario Exitoso: <ul style="list-style-type: none"> • Se recibe con éxito los datos digitales en sus dos únicas tramas \$IF , y \$0F 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ul style="list-style-type: none"> • No se recibieron los datos exitosamente por desconexión de cable serial • No se recibieron los datos exitosamente por mala configuración del módulo USART en el firmware . 	
Resultados: Los datos son enviados al PIC y convertidos a formato digital	

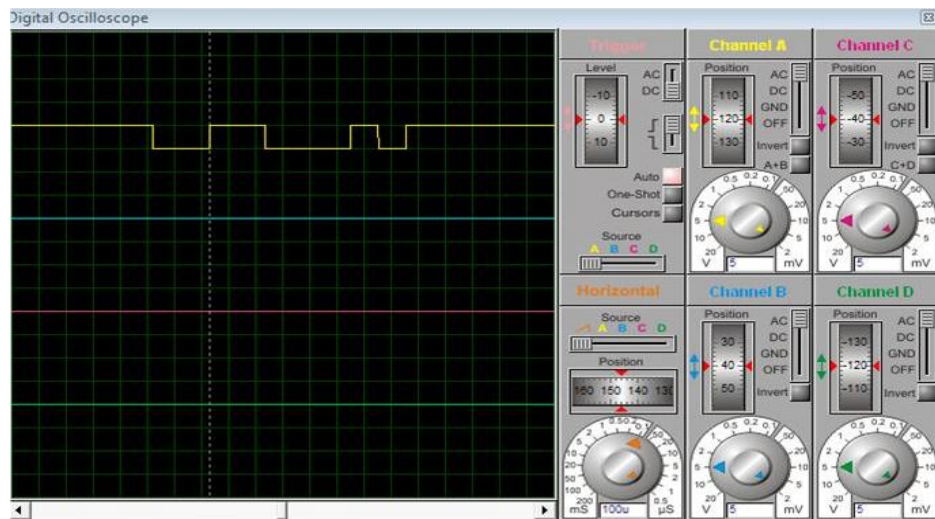


Figura 5.7 Resultado de envío de trama \$10F

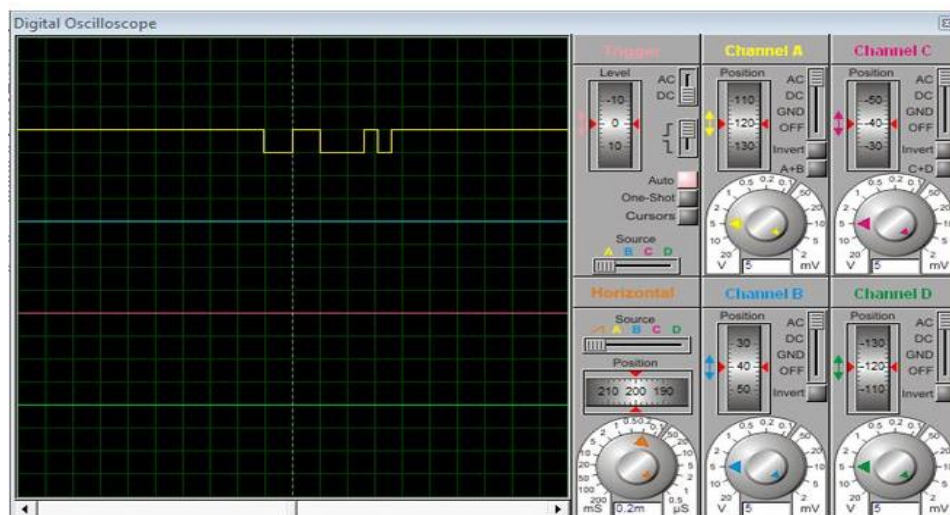


Figura 5.8 Resultado de envío de trama \$11F

5.5 Pruebas de entrada de datos al PC en tiempos determinados

Para las pruebas de llenado de tanques tomamos un total de 15 muestras, para estimar un promedio aproximado y queden registrados en la base de conocimientos. Los casos de uso asociado a esta prueba son :

Nombre:	Registrar de datos de llenado de tanque
Autor:	Andy González/Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza el registro en la base de datos la información referente a: tiempo estimado de llenado, fecha de llenado, hora, minuto y segundo de llenado, el código del tanque, y la altura a la que llegó el líquido.	
Actores: Operador.	
Precondiciones: El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC	
Escenarios Exitosos: 2. Se registró con éxito la información sobre el llenado del tanque	
Escenario Alternativo: No hay	
Escenarios no Exitoso 5. No se pudo registrar la información debido a problemas de conexión a la base de datos 6. No se pudo registrar la información debido a falta de datos en los campos de uno de los formularios de ingreso de información. 7. No se pudo registrar la información debido a desconexión del PC o sobrecarga de energía. 8. No se pudo registrar la información debido a un colapso de procesos en el PC	
Resultados: Los datos son enviados de formato digital al puerto serie del Computador	

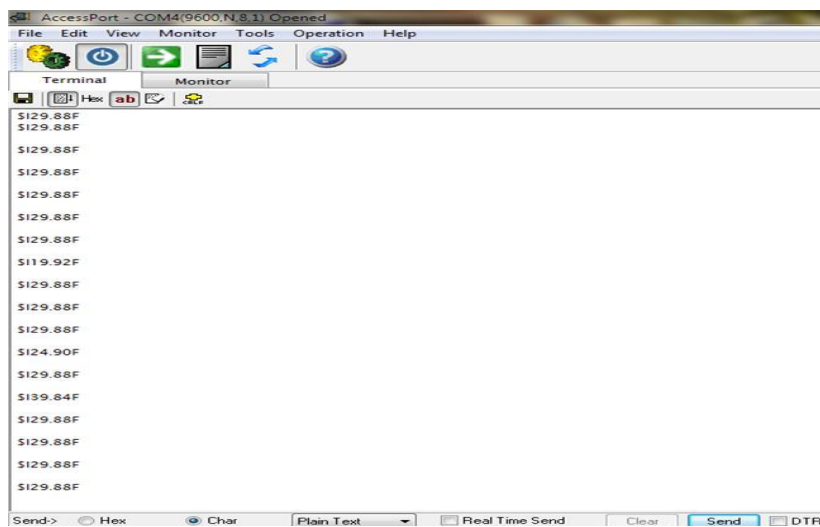


Figura 5.9 Pruebas de entrada de datos al PC en tiempos determinados

En la figura 5.9 mostramos por pantalla los datos que ingresan al PC mediante el software Access Port

Tabla9 Resultado en promedio de tiempos de llenado

Voltaje bomba (V)	Tiempo llenado (mm:hh:ss)	Altura tanque(cm)	Datos sensor(cm)
12	5:36:61	30	29.90
12	5:24:07	30	29.90
12	5:23:50	30	29.88
12	5:18:68	30	29.88
12	5:28:47	30	29.89
12	5:14:13	30	29.95
12	5:13:15	30	29.90
12	5:15:25	30	29.85
12	5:26:25	30	29.75
12	5:28:25	30	29.66
12	5:20:21	30	29.70
12	5:21:23	30	29.90
12	5:20:29	30	29.80
12	5:28:29	30	29.65
12	5:28:30	30	29.05

Las pruebas de la recepción de los datos al computador, fue un arduo trabajo por la series de fallos que encontramos en todo el transcurso de la implementación del proyecto, uno de ellos fue la mala manipulación de la entrada de voltaje al micro controlador y la bomba por la cual el puerto COM se cerraba y no transmitía la información necesaria para manipularlas en el computador.

Solucionamos el problema poniendo y calculando el voltaje ideal en cual podía funcionar normalmente. En cada prueba de llenado de tanque se midió el voltaje de la bomba, por lo cual en la tabla indicamos el voltaje ideal que maneja una bomba DC.

Luego de poder recibir los datos correctamente al computador por medio del sensor, tuvimos fallos de transmisión de envío del computador a la tarjeta de adquisición de datos, para indicarle el prendido y apagado de la bomba que lo maneja automáticamente. El motivo fue un colapso debido a que el puerto COM maneja las 2 líneas de transmisión: tx y rx que entran a la tarjeta de adquisición, es decir cuando tx transmite rx no puede recibir datos al mismo tiempo y viceversa, entonces se tuvo que aplicar un retardo de un segundo en cada proceso de transmisión y recepción de los datos. Consecuentemente obtuvimos los tiempos de llenado en el instante de que cumple la altura especificada como se muestra en la tabla 9.

5.6 Pruebas de modelado en 3D y salida de datos

Para esta prueba del modelado en 3D, evaluamos en determinados tiempos su proceso y visualización, asociado con dos casos de uso que son:

Caso de Uso	Iniciar llenado del tanque
Autor:	Andy González /Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza el llenado del tanque, cuando el operador ejecuta la orden de iniciar el proceso de llenado.	
Actores: Operador.	
Precondiciones: El operador debe haber conectado la tarjeta , haber conectado el cable serial de la tarjeta al PC y además haber configurado el puerto com. Llenar los datos de configuración inicial.	
Escenario Exitoso: Se transmite con éxito los datos a intervalos de tiempo Se inició correctamente el llenado del tanque.	
Escenario Alternativo: No hay	
Escenarios no Exitoso No se inició el llenado del tanque por falta de datos en el formulario. No se inició el llenado del tanque por mala configuración del puerto com No se inició el llenado del tanque porque se desconectó el puerto com No se inició el llenado del tanque, no está energizado el pc. No se inició el llenado del tanque debido a un ruido externo o interferencia electromagnética. No se inició el llenado del tanque debido a no encontrar conexión a la base de datos.	
Resultados: Los datos enviados por PIC los recibe el computador con éxito.	

Caso de Uso:	Visualización tanque 3D
Autor:	Andy González – Jorge López
Fecha:	Agosto/2010
Descripción: Se visualiza el tanque en 3D cuando el operador de la orden de mostrar la gráfica.	
Actores: Operador.	
Precondiciones: <ol style="list-style-type: none"> 6. El operador debe haber conectado la tarjeta, haber conectado el cable serial de la Tarjeta al PC y además haber configurado el puerto com. 7. Haber establecido la conexión a la base. 8. Llenar datos de configuración inicial. 9. Tiempos determinados correctos. 10. Sensor en óptimas condiciones sin ruido o interferencias. 	
Escenario Exitoso: <ol style="list-style-type: none"> 3. Se transmite con éxito los datos a intervalos de tiempo 4. Se visualiza el tanque en 3D con éxito. 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ol style="list-style-type: none"> 5. No se visualizó el tanque en 3d debido a un ruido externo o interferencia electromagnéticas que sufrió el sensor. 6. No se visualizó el tanque en 3d debido a procesos colapsados por el Pc. 7. No se visualizó el tanque en 3d debido a conexión de la base falle. 8. No se visualizó el tanque en 3d debido a los tiempos no determinados. 	
Resultados: Se visualiza el tanque en 3D mostrando un sombreado de la altura de tanque.	

Tabla 10 Resultado del tiempo de llenado vista en 3D

Altura Tanque Real (cm)	Propiedad Rectángulo (cuadrillas - Height)	Tiempo llenado (mm:ss.:ms)
0	0	0:00.000
5	63.33	1:30.720
10	126.07	2:20.412
15	190	3:00.891
20	253.33	3:57.987
30	380	5:24:324

En la prueba del llenado del modelado 3D del tanque, tuvimos un inconveniente en mostrar la visión en tiempo real, en la cual resolvimos este problema con un rectángulo utilizando los controles de wpf, donde se manipula sus propiedades de altura y anchura, para poder apreciar el llenado del tanque en tiempos determinados. Por consiguiente podemos apreciar este llenado del tanque en tiempo determinados. En la figuras 5.10 - 5.11 - 5.12 - 5.13 se muestra el proceso de llenado y el tiempo promedio de cada intervalo de altura, para los tiempos indicados en la Tabla 10.

Los datos obtenidos, a una altura de 5 cm el tiempo promedio es 1:30.720 minutos

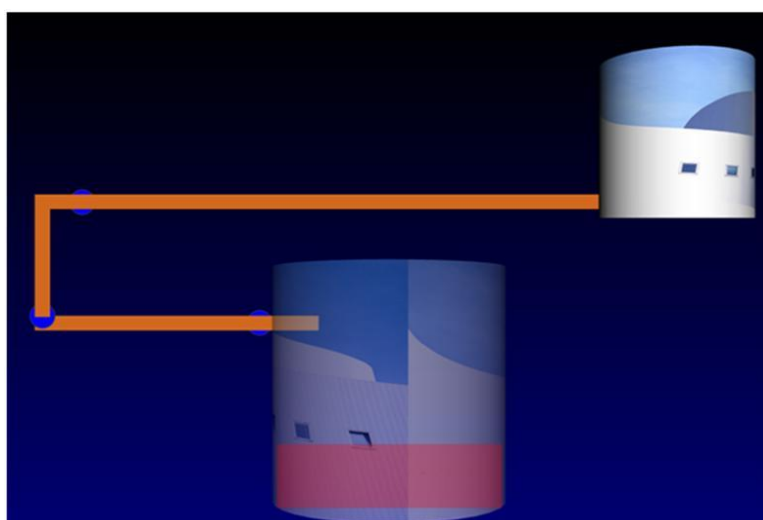


Figura 5.10 Llenado en 3D para 1:30.720

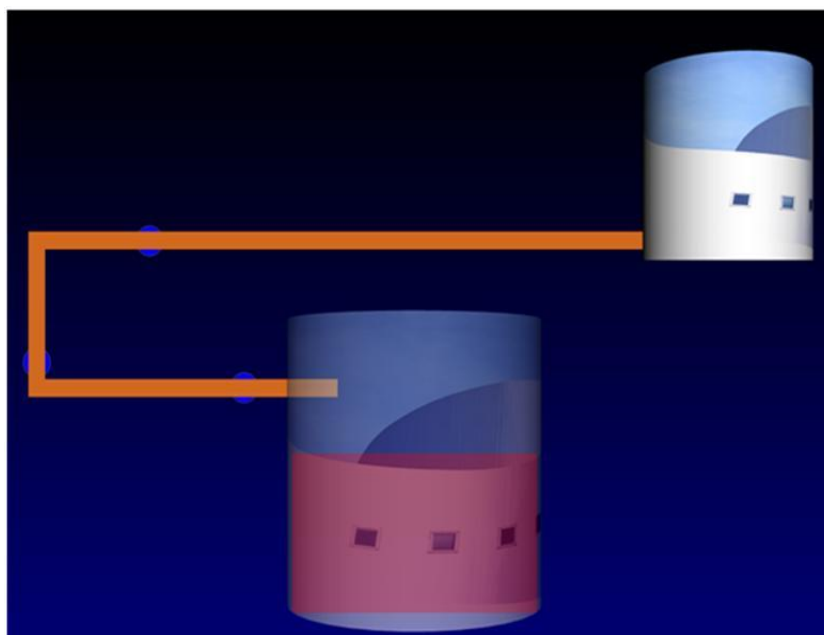


Figura 5.11 Llenado en 3D para 3:00.412

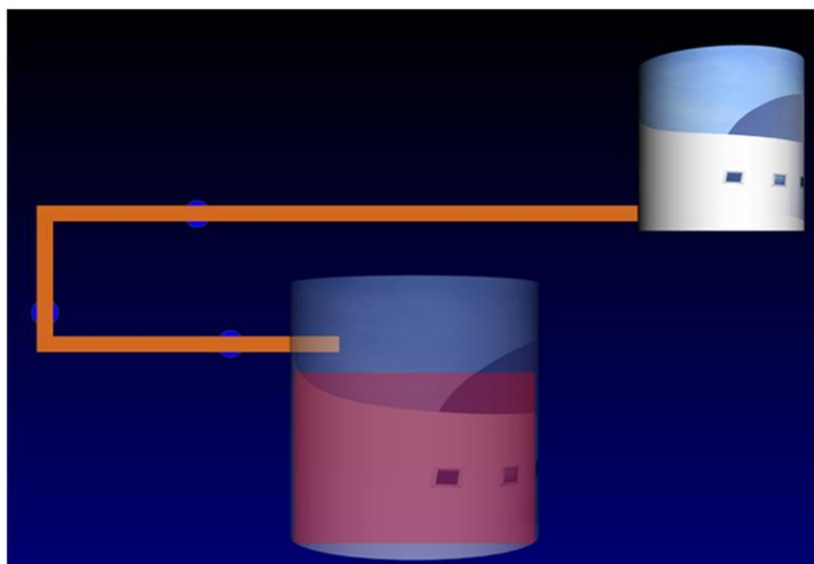


Figura 5.12 Llenado en 3D para 3:57.891

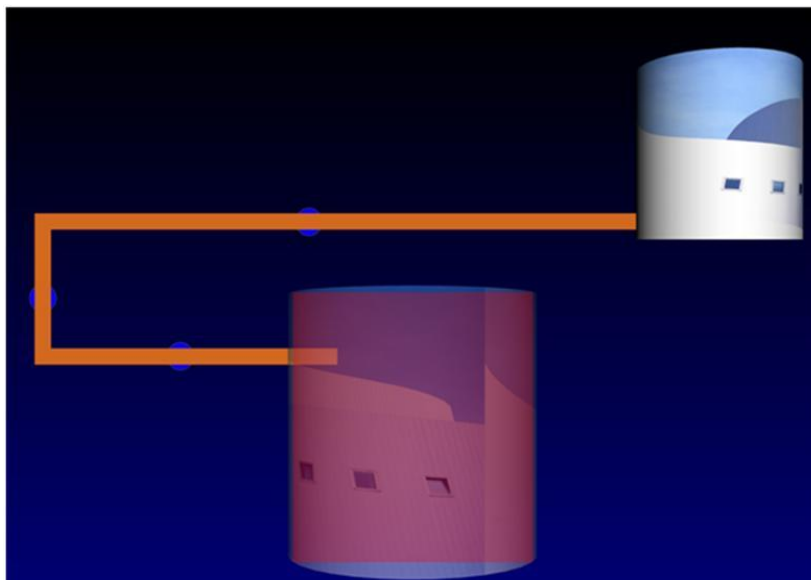


Figura 5.13 Llenando en 3D para 5:24.324

5.7 Pruebas de las funciones de membresía para el control de la bomba

El esquema de la lógica difusa se muestra en la figura 5.14 donde el eje de las x representalos diversos intervalos de altura del tanque, para la cual como máximo hemos colocado un tope de 45 cm y en la tabla 11 se especifican los rangos de nivel por cada función de membresía.

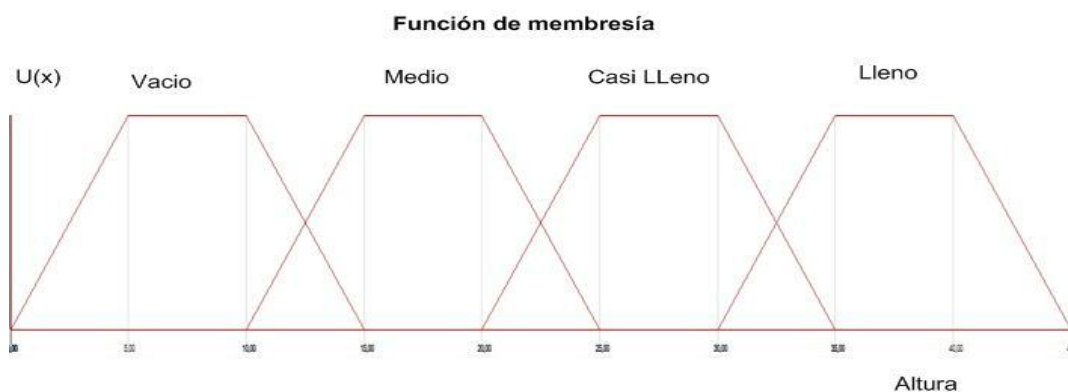


Figura 5.14 Esquema de lógica Difusa

Tabla 11 Datos de la función de Membresía

Rango de niveles de tanques en centímetros (cm)				
Vacio	0	5	10	15
Medio	10	15	20	25
Casi lleno	20	25	30	35
Lleno	30	35	40	45

Estado de bomba	Rango de niveles en centímetros (cm)			
on	0	20	28	40
off	41	42	44	45

Los casos de uso especificado para esta prueba son:

Nombre:	Ejecución de las Funciones de Membresía.
Autor:	Andy González /Jorge López
Fecha:	Agosto/2010
Descripción: Se ejecuta la función de membresía, cuando el sensor esta en óptimas condiciones evaluando cada intervalo de altura y determinando si continua encendida o apagada la bomba (on/off)	
Actores: Operador – Sistema.	
Precondiciones: <ol style="list-style-type: none"> 1. El operador debe haber conectado la tarjeta haber conectado el cable serial de la Tarjeta al PC y además haber configurado el puerto com. 2. Llenar datos del formulario y configuraciones iniciales. 3. Haber Comenzado el llenado del tanque. 	
Escenario Exitoso: <ol style="list-style-type: none"> 1. Se transmite con éxito los datos a intervalos de tiempo 2. Se inició correctamente la función de membresía. 	
Escenario Alternativo: No hay	
Escenarios no Exitoso <ol style="list-style-type: none"> 1. No se ejecutó la función de membresía debido a un ruido externo o interferencia electromagnéticas que sufrió el sensor. 2. No se ejecutó la función de membresía debido a datos erróneos por el sensor. 3. No se ejecutó la función de membresía, no está energizado el pc. 	
Resultados: Se evalúa la función de membresía indicando el prendido o apago de la bomba.	

Tabla 12 Resultados de la evaluación de las funciones de Membresía

Valor del sensor (cm)	Evaluador de las funciones de membresía	Estado Bomba (on/off)	Estado funcionamiento
10.10	22.16	ON	OK
12.15	22.16	ON	OK
12.50	22.16	ON	OK
15.35	22.16	ON	OK
20.12	22.16	ON	OK
22.98	22.16	ON	OK
35.25	22.16	ON	OK
40.00	22.16	ON	OK
43.00	44.33	OFF	OK
45.00	44.33	OFF	OK
60.76	NEUN	ON	Automático

Valor del Sensor: Son los datos obtenidos del sensor mediante la tarjeta de adquisición de datos.

Evaluador de las funciones de membresía: En esta función obtenemos dos valores 22.16 es un valor constante que significa que la Bomba esta prendida y 44.33 es otra constante que significa que la bomba está apagada, como se mostró en el capítulo de diseño estos valores son los datos defuzzyficados. (Ver página 85).

Estado de Bomba: Este estado significa que si está en ON la bomba continúa prendida o en funcionamiento, si está en OFF la bomba está apagada o se apaga.

Estado de Funcionamiento: Este resultado representa el modo en el que se encuentra el sistema, que son tres: OK, Automático, Emergente. (Ver página 91).

Tabla 13 Resultados del sistema con Lógica difusa

Desfase del Sensor (cm)	Sensor (cm)	Tiempo Promedio (mm:ss:ms)	Estado de la Bomba
15	44,13	5:24:10	OFF
15	44,01	5:24:18	OFF
15	43,25	5:23:12	OFF
15	44,23	5:23:10	OFF
15	44,59	5:24:20	OFF
15	45,00	5:24:59	OFF
15	44,11	5:25:00	OFF
15	43,55	5:23:50	OFF
15	44,17	5:23:59	OFF
15	44,33	5:24:58	OFF
15	44,30	5:24:45	OFF

Tabla 14 Resultados del sistema sin Lógica difusa

Desfase del Sensor (cm)	Sensor (cm)	Tiempo Promedio (mm:ss:ms)	Estado de la Bomba
15	46,00	00:05:40	OFF
15	47.32	00:05:52	OFF
15	46.25	00:05:35	OFF
15	48,36	00:06:00	OFF
15	50.32	00:06:20	OFF
15	49.32	00:05:59	OFF
15	47.25	00:05:45	OFF
15	48.32	00:05:59	OFF
15	46.32	00:05:45	OFF
15	46.55	00:05:55	OFF
15	49.36	00:06:05	OFF

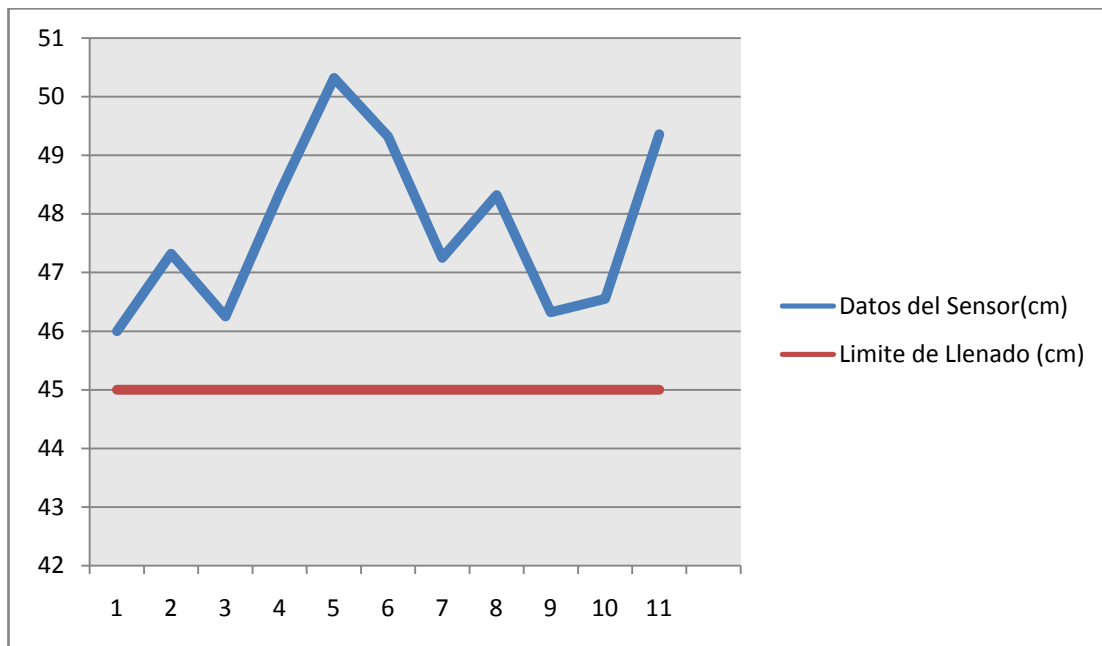


Figura 5.15 Grafica del nivel del llenado sin Lógica Difusa

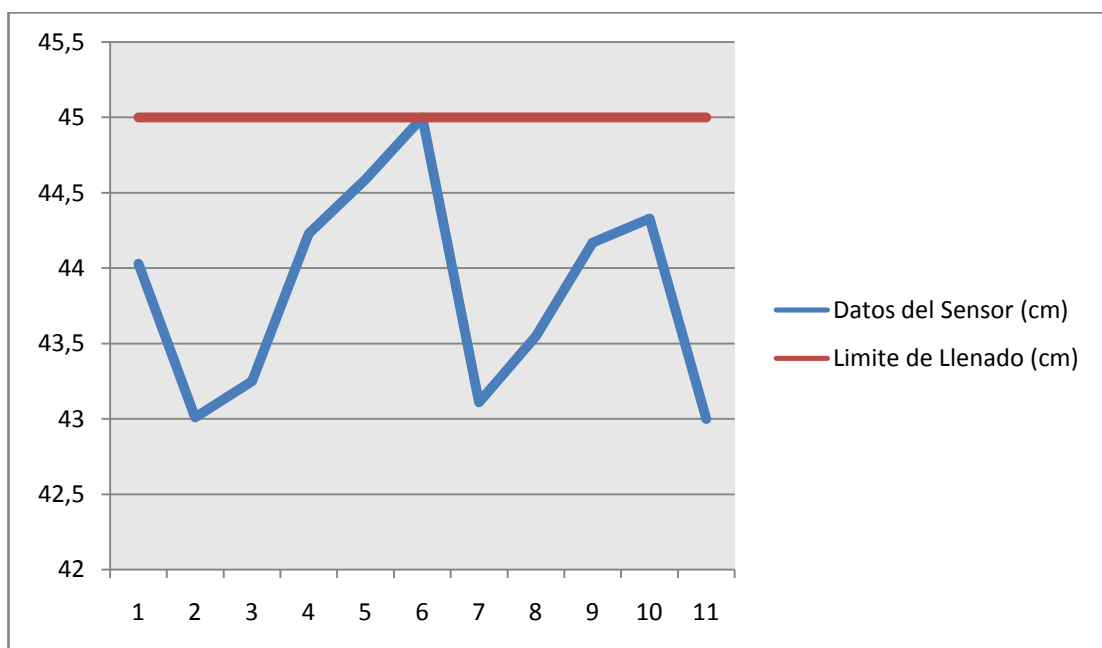


Figura 5.16 Grafica del nivel de llenado con Lógica Difusa

Como se observó en las tablas 13 y 14, usando lógica difusa y sin ella los resultados que se muestran indican que el método de lógica difusa es indispensable para la automatización de este proyecto que permite controlar y calibrar el llenado del tanque, ya que sin usar lógica difusa los niveles se pasan del límite de llenado como se muestran en la Tabla 14 y las figuras 5.15 y 5.16

Las pruebas de las funciones de membresía fueron extensas, por motivo que el sensor acústico es sensible al ruido y fuentes cercanas a un rango, las desventajas del sensor acústico están especificadas en el capítulo 3. Como vemos en la tabla tomamos como funciones de membresía para indicar que el sensor está en estado activo entre las cuales son: vacío, medio, casi lleno, lleno. Estas variables están en un rango de 0 – 45 cm que es el límite para llenar los tanques del proyecto realizado.

Las funciones de membresía nos muestran valores de acuerdo al rango en que se encuentre, el problema que tuvimos en las funciones de membresía fue cuando se pasaba del rango (mayor a 45 cm) por motivo que el sensor es sensible a los ruidos.

La solución obtenida fue, como las librerías de .net Dotfuzzy facilita el manejo de las funciones de membresía y habladas en el capítulo 2 y 3. Cuando se pasa del rango el sistema muestra una constante llamada NEUN, Neun es un valor incierto cuando esta pasada del rango y no lo conoce de acuerdo a las variables lingüísticas de nivel mostradas.

Cada vez que el sensor esté en lectura y este expuesto a ruido, falla la lectura del sensor, se activará la constante NEUN, esta constante se la utiliza para controlar automáticamente el llenado del tanque.

En la figura 5.17 se muestra un dato erróneo del sensor y el cambio del estado del sensor a estado automático.



Figura 5.17 Dato erróneo del sensor y ejecución del modo Automático



Figura 5.18 Dato erróneo del sensor y ejecución del modo Apagado de Emergencia

El caso de uso asociado a estos modos son dos, el control por Modo Automático y el modo Apagado de Emergencia, ambos se ejecutan en diferentes instancias, para referencia ir al capítulo de diseño.

Caso de Uso:	Ejecución de Modo Automático
Autor:	Andy González/Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza cuando el sistema detecta que el sensor esta averiado, este modo recoge el histórico de información que se encuentra en la base de datos.	
Actores: Operador.	
Precondiciones: <ol style="list-style-type: none"> 1. El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC 2. Haber ingresado la información del tanque a llenar 3. Sensor dañado 	
Escenarios Exitosos: <ol style="list-style-type: none"> 3. Se ejecutó el proceso automático exitosamente 	
Escenario Alternativo: Se visualiza estado del sensor averiado en una gráfica lineal	
Escenarios no Exitoso <ol style="list-style-type: none"> 9. No se ejecutó el proceso automático debido a que el sensor está funcionando correctamente. 10. No se ejecutó el proceso automático debido a que se desconectó la alimentación de la tarjeta de adquisición de datos. 11. No se ejecutó el proceso automático debido a un colapso de los procesos del sistema. 12. No se ejecutó el Modo Automático por alguno de los otros modos inteligentes del sistema esté en ejecución. 	
Resultados: El sistema toma la información del tiempo promedio de llenado asociado al tanque de la base de datos y ejecuta el Modo Automático	

Caso de Uso:	Ejecución de Modo Emergente
Autor:	Andy González/Jorge López
Fecha:	Agosto/2010
Descripción: Se realiza cuando el sistema detecta que el sensor esta averiado, y el modo automático no se ejecutó correctamente	
Actores: Operador.	
Precondiciones: <ol style="list-style-type: none"> 1. El operador debe haber conectado la tarjeta y haber conectado el cable serial de la tarjeta al PC 2. Haber ingresado la información del tanque a llenar. 3. Sensor dañado. 4. Modo automático no se ejecutó. 	
Escenarios Exitosos: Se ejecutó el Modo Emergente exitosamente	
Escenario Alternativo: Se visualiza estado del sensor averiado en una gráfica lineal	
Escenarios no Exitoso <ol style="list-style-type: none"> 1. No se ejecutó el modo emergente debido a que el sensor está funcionando correctamente 2. No se ejecutó el modo emergente debido a que se desconectó de alimentación la tarjeta de adquisición de datos. 3. No se ejecutó el modo emergente por alguno de los otros modos inteligentes del sistema este en ejecución 	
Resultados: Apenas se detecte colapso de algún proceso se manda a apagar el sistema en el tiempo más cercano al ideal del llenado para ese tanque.	

En la tabla 15, Se muestran los datos obtenidos del proyecto, de las diferentes pruebas realizadas (Con lógica difusa, sin lógica difusa).

Tabla 15 Muestras de pruebas de la base de datos

Número de pruebas	Altura de llenado (cm)	Tiempo de llenado (mm:ss.ms)	Modo
1	43,27	5:25:23	Difusa(ok)
2	43,60	5:25:25	Difusa(ok)
3	44,45	5:24:30	Difusa(ok)
4	44,15	5:24:25	Difusa(ok)
5	44,28	5:24:22	Difusa(ok)
6	44,22	5:24:22	Difusa(ok)
7	44,26	5:24:59	Difusa(ok)
8	44,82	5:24:58	Difusa(ok)
9	44,24	5:24:01	Difusa(ok)
10	43,95	5:23:50	Difusa(ok)
11	44,73	5:24:45	Difusa(ok)
12	43,45	5:23:10	Difusa(ok)
13	44,13	5:24:10	Difusa(ok)
14	44,01	5:24:18	Difusa(ok)
15	43,25	5:23:12	Difusa(ok)
16	44,23	5:23:10	Difusa(ok)
17	44,59	5:24:20	Difusa(ok)
18	45,00	5:24:59	Difusa(ok)
19	44,11	5:25:00	Difusa(ok)
20	43,55	5:23:50	Difusa(ok)
21	44,17	5:23:59	Difusa(ok)
22	44,33	5:24:58	Difusa(ok)
23	44,30	5:24:45	Difusa(ok)
24	43,50	5:23:30	Difusa(ok)
25	45,26	5:25:00	Automático
26	46,50	5:25:00	Automático
27	68,56	5:25:00	Automático
28	69,05	5:25:00	Automático
29	46,00	5:40:00	Sin usar Difusa
30	47.32	5:52: 00	Sin usar Difusa

31	46.25	5:35: 00	Sin usar Difusa
32	48,36	6:00: 00	Sin usar Difusa
33	50.32	6:20: 00	Sin usar Difusa
34	49.32	5:59: 00	Sin usar Difusa
35	47.25	5:45: 00	Sin usar Difusa
36	48.32	5:59: 00	Sin usar Difusa
37	46.32	5:45: 00	Sin usar Difusa
38	46.55	5:55: 00	Sin usar Difusa
39	49.36	6:05: 00	Sin usar Difusa
40	81,97	5:25:00	Automático
41	82,17	5:25:00	Automático
42	83,06	5:25:00	Automático

En la tabla 16, se muestran los resultados de las pruebas usando lógica difusa, dando un porcentaje de eficiencia del 98,01%.

Tabla 16 Resultados de las Pruebas de la Base de Datos usando Lógica Difusa

Nivel Promedio	Tiempo Promedio	Nivel Ideal	Tiempo Ideal
44,105	5:24:21	45	5:25:00

$\% \text{ de Eficiencia en Nivel} = (\text{Nivel Promedio del Sensor} / \text{Nivel Ideal}) * 100$

%Eficiencia en Nivel = 98,01 %

En la tabla 17 se muestran los resultados de las pruebas usando lógica difusa, dando un porcentaje de error de 1.98%

Tabla 17 Resultados de las pruebas con Lógica Difusa mostrando el porcentaje de error

Nivel Promedio (cm)	Nivel ideal (cm)	Desviación Estándar(cm)	%Error
44,11	45	0.445	1.98 ± 1.04

%Erróneo en Nivel = 1.98 %

En la tabla 18, se muestran los resultados de las pruebas sin lógica difusa, dando un porcentaje erróneo del 6.13%.

Tabla 18 Resultados de las pruebas sin Lógica Difusa mostrando el porcentaje de error

Nivel Promedio (cm)	Nivel ideal (cm)	Desviación Estándar(cm)	%Error
47,76	45	1.41	6.13 ± 3.26

%Erróneo en Nivel = 6.13 %

Los resultados obtenidos por las pruebas, indican que el uso de la tecnología difusa es fundamental para el control y calibración del llenado de los tanques, y sin usar dicha tecnología habrá errores en el proceso de llenado.

En las siguientes figuras (5.19-5.20-5.21) se muestra la distribución normal e histogramas, usando lógica difusa y sin usar lógica difusa.

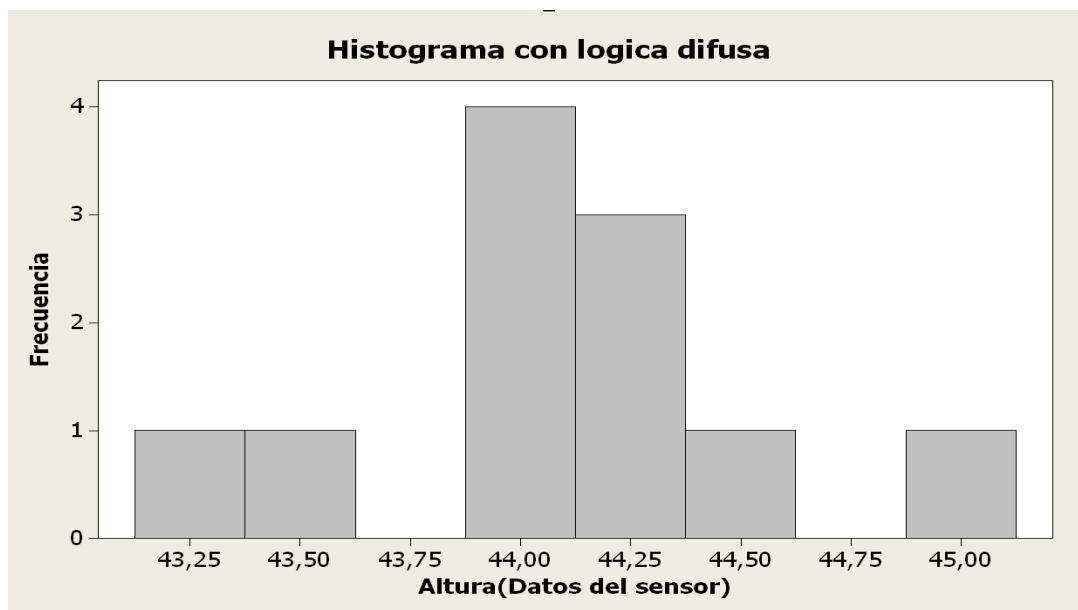


Figura 5.19 Histograma usando lógica difusa

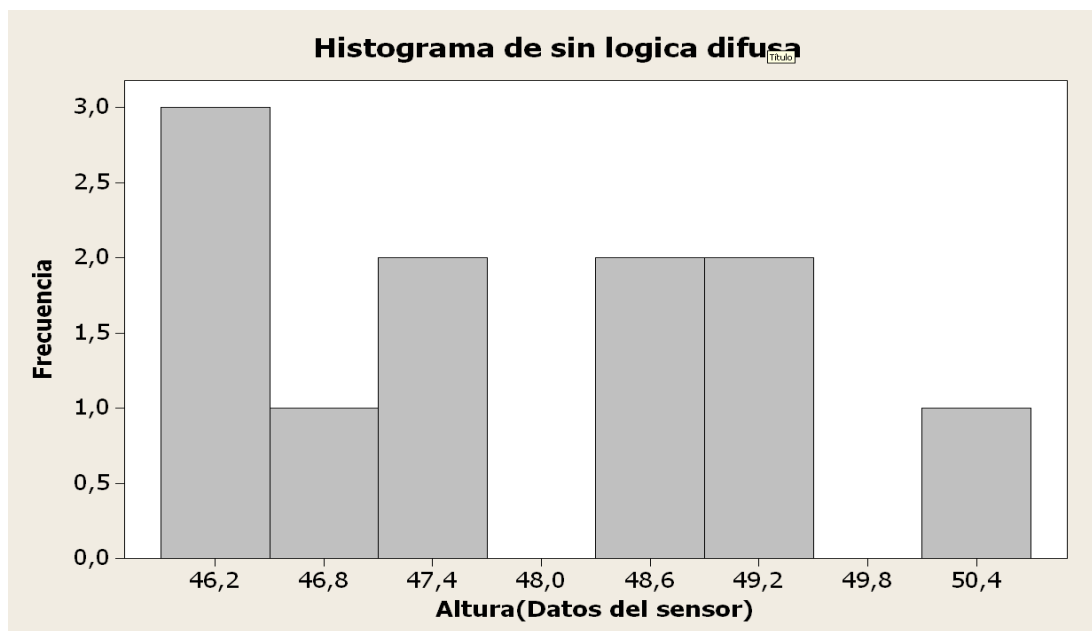


Figura 5.20 Histograma sin usar lógica difusa

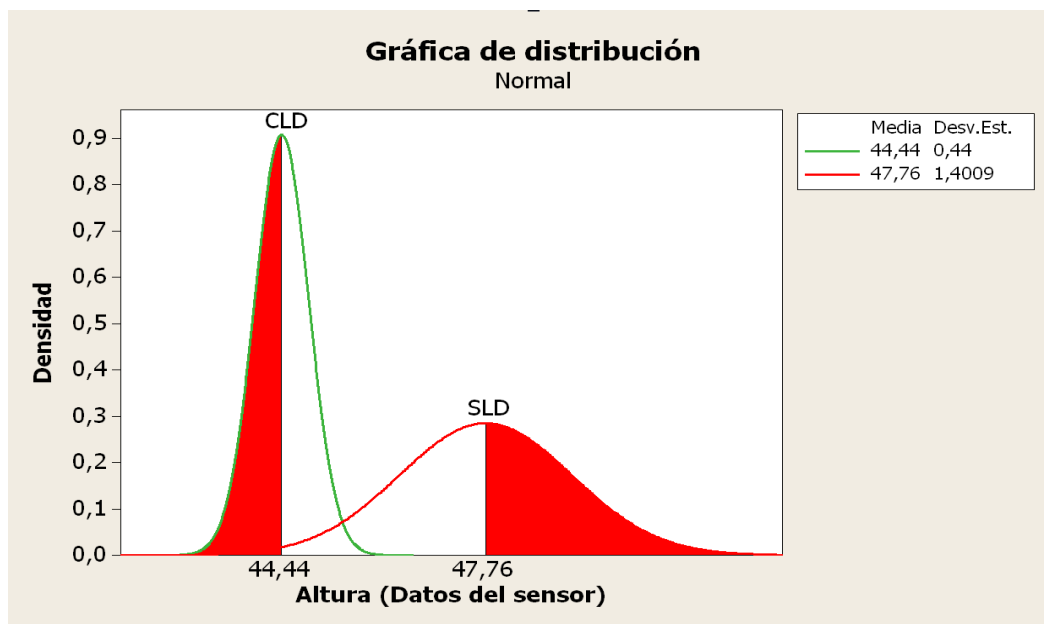


Figura 5.21 Distribución Normal (Con lógica difusa, sin lógica difusa).

5.8 Análisis de los resultados de las pruebas

En las pruebas realizadas, 42 muestras de la base nos dicen lo siguiente:

- ✓ 24 pruebas fueron realizadas, usando el algoritmo de lógica difusa.
- ✓ 11 pruebas fueron realizadas sin ningún tipo de algoritmo controlador.
- ✓ 7 pruebas extras fueron realizadas para el análisis del estado del sistema.

Estas pruebas se realizaron en el momento que el sensor comenzó a fallar, por exceso de manejo en sus circuitos electrónicos. Hay que recalcar, que las 11 de las 42 pruebas que se realizó sin usar lógica difusa fueron forzadas para verificar el suceso de llenado del tanque y los resultados fueron los siguientes:

- ✓ 24 de las 24 muestras que se tomaron, usando lógica difusa, se llegó al tiempo estimado de llenado en un promedio de 5 minutos con 24 segundos, para un tanque de 45 cm. Concluimos un 98.01% de efectividad usando lógica difusa
- ✓ 11 de 11 pruebas realizadas sin lógica difusa se obtuvo un error de 6.13%
- ✓ 7 de las 7 pruebas realizadas demuestran que el sistema entra en Modo Automático, cumpliendo con el tiempo ideal 5:25:00

Estos datos fueron obtenidos en toda la etapa de las pruebas y realización del proyecto.

Para realizar las pruebas es vital constar con un prototipo, sin él no podríamos realizar ningún tipo de pruebas ni verificar la veracidad de los datos, entonces podemos decir que para la realización de un proyecto de tal magnitud es importante constar con un modelo a escala de la solución. El usar tanques, bombas DC, reguladores de tensión permitió acercarse más a la realidad a la hora de la realización de pruebas.

Pudimos determinar que si no tenemos regulada la tensión para el nivel ideal indicado en las bombas DC se producirá interferencias a tal punto de obstruir la comunicación en los puertos de comunicación COM.

Se tuvo que implementar un retardo de un segundo para darle tiempo a los procesos programados en el pic, como lo es el proceso de envío de datos, y el proceso de recepción de datos, ya que ambos no se pueden ejecutar al mismo tiempo.

De los niveles de tensión registrados por la salida analógica del sensor (0-2.5V) se determinó en las pruebas un porcentaje de precisión del 99% de exactitud, es decir cada

dato leído por el sensor era el correcto según el nivel actual del tanque en ese instante de tiempo.

En algunas pruebas, el sensor fue sometido a fallos con fuentes de ruido externos, eso ocasionó que el sistema opte por un llenado en modo automático, el sensor indicaba datos fuera de los límites especificados por las funciones de membresía, y optó por tomar el máximo de los promedios registrados en la base de datos para el llenado de un tanque con esa altura específica.

Se tuvo que manejar procesos implementados en paralelos (Threads) para la visualización de todos los datos posibles tanto como el estado del sensor y la visión 3D, sin esto el PC colapsa y el sistema deja de funcionar.

Luego de realizar las pruebas se puede recomendar que para evitarla implementación de retardos usar PICS de gama ALTA como los de la serie 18F ellos manejan dos módulos USART por separados pudiendo utilizar uno para transmisión y otro para recepción

Invertir en sensores de mayor precisión para la obtención de resultado que esté libre de exposición a ruidos

Proteger de la mejor forma los componentes acoplando un fusible en caso de un corto en la bomba DC

Para el modelado en 3d del tanque utilizar herramientas de modelado como 3d studio max y maya, ya que visual studio.net y 3d studio max versión 2010 trae un plugin para

convertir el modelado .Max a lenguaje xaml y puede utilizarse con facilidad los modelados de cualquier tipo.

Utilizar un firmware para controlar y manipular tanto hardware como software para la manipulación de los datos, tanto del computador como la tarjeta de adquisición de datos.

Hacer las pruebas del llenado del tanque con otros tipos de líquidos, con densidades mayores o menores a 1. Ya que las pruebas anteriores fueron realizadas con agua.

Hacer las pruebas con otro tipo de material de tanque (“no plástico”), ya que si utilizan líquidos a altas temperaturas puede causar averías.

Implementar otros métodos de algoritmo difuso, por ejemplo usando el Método Trapezoidal, este método es excelente para manejos de lógica difusa ver link [4].

Para la utilización de sensores acústico se recomienda utilizar un medidor de ruido (decibeles) para calcular los ruidos externos, ya que estos pueden causar daños al sensor y a la vez emitir datos erróneos, si se pasa del límite establecido por las especificaciones del sensor.

Si existe una fuente de ruido externa trabajando con el sensor, se debe indicar: la distancia que puede trabajar dicha fuente y su nivel de ruido aproximado.

En los cálculos realizados se consideró las bombas de limpiaparabrisas de carros, operan a una corriente de 3mA y dependiendo del voltaje que se le suministre esta pueden variar en potencia y en tiempo de llenado para eso se definió un voltaje fijo de 12V es decir 36mW de energía, el llenado de un tanque de 20 litros aproximadamente se llenó en 5 minutos aproximadamente.

CONCLUSIONES Y RECOMENDACIONES

Recapitulando desde los objetivos planteados en el primer capítulo, en la presente tesis se desarrolló un Sistema en .net que permite la detección de niveles de llenado de tanques por medio de un sensor ultrasónico. A lo largo de los capítulos se plantearon estructuradamente las motivaciones, problemas y soluciones. A modo de síntesis, se concluye y se recomienda:

Conclusiones

- 1) Las mayorías de las industrias no manejan sistemas que integran tecnologías entre hardware + software para censar los datos de niveles en tanques, con esta combinación de nuestro proyecto mejoramos un 98 % de exactitud en el llenado y calibrado para los tanques con lógica difusa.
- 2) Concluimos que mostrar el llenado del tanque modelado en 3D, realizando el proceso en tiempo real, ayuda al operador a controlar el nivel de llenado del

tanque, ya que las mayorías de sistemas solo manejan los sensores de nivel es decir solo el hardware.

- 3) La utilización de PIC de gama media resulto ineficiente para la toma de datos y emplearlo como corazón de la tarjeta de adquisición de datos, para analizar proyectos a gran escala se debe pensar en el uso de PICS más avanzado y con mayores prestaciones.
- 4) Las funciones de membresía son fundamentales para automatizar el apagado de las bombas DC, ayudan a pronosticar por medio de sus variables lingüísticas y además obteniendo los datos del sensor, a obtener en qué nivel esta llenado el tanque y saber en qué momento apagar las bombas DC, estas funciones de membresía son ideales al momento de automatizar algún proceso industrial que requieran algún control.
- 5) El apagado emergente es prioritario, si el proceso de cualquier maquina colapse, si esto pasase puede ocurrir cualquier catástrofe industrial, concluimos que la creación de un tiempo emergente histórico es fundamental para evitar cualquier eventualidad o emergencias industriales.
- 6) Llevar un registro de los tiempos de llenado, fecha de llenado, hora de llenado, el fin es de optimizar las consultas de resultado, ya que el sistema trabaja por tiempos promedios de los ingresos anteriores, estos historiales de registro permiten visualizar ítem a ítem su proceso, si fállase algún evento, se consultaría su historial, donde se guarda en la base de datos.

Recomendaciones

- 1) Se recomienda el uso de dispositivos de hardware especializados como son los PLC (Controladores lógicos Programables), estos dispositivos pueden mejorar la adquisición de datos , puesto que en su arquitectura ellos manejan los datos en tiempo real y son aparatos exclusivamente dedicados pero todo bajo su entorno de programación.
- 2) Otra opción sería utilizar un sistema operativo en tiempo real como lo es RTlinux cuyas características se pueden ver en [11], este tipo de SO optimizan las capturas en tiempo real de los datos, evitando así los retrasos que pueden haber por prioridad de procesos.
- 3) Una aplicación útil en tiempo real recomendada es Wonderware InTouch HMI. Esta es una aplicación SCADA en tiempo real para revisar más referencias ver en [12] Intouch
- 4) Para el modelado en 3d del tanque podemos utilizar herramientas de modelado como 3d Studio Max y maya, ya que Visual Studio.net y 3d Studio Max versión 2010 trae un pluggin para convertir el modelado .max a lenguaje xaml y puede utilizarse con facilidad los modelados de cualquier tipo.
- 5) Utilizar un firmware para controlar y manipular tanto hardware como software para la manipulación de los datos, tanto del computador como la tarjeta de adquisición de datos.

- 6) Se recomienda utilizar valores de voltaje en un rango entre 9v – 12 v, para que la tarjeta de adquisición de datos funcione de manera ideal y no tenga inconvenientes al transmitir la información al computador.
- 7) Recomendamos la utilización de un PIC de GAMA alta como los de la serie 18FXXX poseen varios módulos de comunicación mayor capacidad en memoria y para uno de los problemas que tuvimos sería ideal a la hora de solucionar el problema de la comunicación.
- 8) Para censar niveles de líquido se recomienda utilizar un sensor de mayor jerarquía, estos sensores pueden ser: sensor de presión, sensor de boya, sensor de radar, ellos tienen características importantes como es la precisión de los datos de los niveles de altura, pero se requiere de una inversión considerable, el motivo por la cual nuestro proyecto va orientado a medianas y grandes industrias que deseen optimizar procesos como lo especificado aquí en nuestro proyecto.
- 9) Se recomienda para la adquisición de los datos al PC, usar Ethernet, con este estándar de red se utilizaría para la transmisión y recepción de los datos a gran escala, es una de las grandes ventajas para enviar y recibir información a la tarjeta de adquisición de datos.
- 10) Cuando se selecciona el tipo o tamaño de bomba, se debe tener en cuenta que la bomba por si sola debe ser capaz de abastecer la demanda máxima dentro de los rangos de presiones y caudales, existiendo siempre una bomba adicional para alternancia con la (o las) otra (u otras) y cubrir entre todas, por lo menos el

140% de la demanda máxima probable. Además debe trabajar por lo menos contra una carga igual a la presión máxima del tanque. Ver en [20]

11) Cuando se dimensiona un tanque se debe considerar la frecuencia del número de arranques del motor en la bomba, llamados Ciclos de Bombeo. Si el tanque es demasiado pequeño, la demanda de distribución normal extraerá el agua útil del tanque rápidamente y los arranques de las bombas serán demasiado frecuentes, lo que causaría un desgaste innecesario de la bomba y un consumo excesivo de potencia. Ver en [20]

12) El punto en que ocurre el número máximo de arranques, es cuando el caudal de demanda de la red alcanza el 50% de la capacidad de la bomba. En este punto el tiempo que funcionan las bombas iguala al tiempo en que están detenidas. Si la demanda es mayor del 50%, el tiempo de funcionamiento será más largo; cuando la bomba se detenga, la demanda aumentada extraerá el agua útil del tanque más rápidamente. Ver en [20].

REFERENCIAS BIBLIOGRÁFICAS

- [1]<http://www.maxbotix.com/uploads/LV-MaxSonar-EZ1-Datasheet.pdf>
- [2]<http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>
- [3]<http://msdn.microsoft.com/es-es/library/ms747437.aspx#materials>
- [4]http://asignaturas.inf.udec.cl/cgraf/public_html/iluminacion.pdf
- [5]http://asignaturas.inf.udec.cl/cgraf/public_html/iluminacion.pdf
- [6]http://asignaturas.inf.udec.cl/cgraf/public_html/iluminacion.pdf
- [7]<http://msdn.microsoft.com/es-es/library/ms747437.aspx#lights>
- [8]<http://www.labcenter.com/index.cfm>
- [9]http://www.ccsinfo.com/newsdesk_info.php?newsdesk_id=120
- [10]<http://msdn.microsoft.com/es-es/vstudio/default.aspx>
- [11] <http://es.wikipedia.org/wiki/RTLinux>
- [12]<http://global.wonderware.com/EN/Pages/WonderwareInTouchHMI.aspx>
- [13] <http://msdn.microsoft.com/es-es/library/ms229335%28v=VS.90%29.aspx>
- [14] <http://msdn.microsoft.com/es-es/library/zw4w595w%28v=vs.90%29.aspx>

- [15] http://translate.googleusercontent.com/translate_c?hl=es&sl=en&u=http://www.kindohm.com/technical/wpfdtutorial.htm&prev=/search%3Fq%3DMeshGeometry3D%2B%252B%2Bwpf%26hl%3Des&rurl=translate.google.com.ec&usg=ALkJrhi7es5IDVw5pHpuV7jhEFTRcQYPEA#note01
- [16] http://translate.googleusercontent.com/translate_c?hl=es&sl=en&u=http://code.msdn.microsoft.com/Project/Download/FileDownload.aspx%3FProjectName%3Dwpfsamples%26DownloadId%3D7726&prev=/search%3Fq%3DMeshGeometry3D%26hl%3Des%26sa%3DG&rurl=translate.google.com.ec&usg=ALkJrhEI Cl6dP6LGBO9IfXninTWesGLyQ
- [17] <http://www.spri.upv.es/mselectronica1.htm#p2>
- [18] <http://www.electricidadbasica.net/>
- [19] <http://pacific.fi-p.unam.mx/cursos/sensor%20ultra/sensor.html>
- [20] <http://fluidos.eia.edu.co/hidraulica/articulos/maquinashidraulicas/hidroneumaticos/paginas/hidroneumaticos.htm>

Referencias de figuras con Derechos de autor:

- Figura II Fuente de las imágenes www.silosytolvas.com
- Figura 1.1 Fuente de las imágenes www.directindustry.com
- Figura 1.3 www.labomecanica.tol.itsem.mx
- Figura 1.5 www.articulo.mercadolibre.com.pe
- Figura 1.7 www.blogs.msdn.com
- Figura 1.17 <http://code.msdn.microsoft.com/wpfsamples>
- Figura 1.18 <http://msdn.microsoft.com/es-es/library/zw4w595w.aspx>
- Figura 1.19 <http://msdn.microsoft.com/es-es/library/aa970268.aspx>
- Figura 1.20 <http://msdn.microsoft.com/es-es/library/aa970268.aspx#Graphics>
- Figura 1.21 <http://msdn.microsoft.com/es-es/library/aa970268.aspx>
- Figura 2.5 www.adafruit.com
- Figura 2.11 www.articulo.mercadolibre.com.mx

Referencias de Internet para consultas en la parte Multimedia:

- www.grupomegabyte.com
- www.editorialmegabyte.com
- <http://code.msdn.microsoft.com/wpfsamples>
- http://es.wikipedia.org/wiki/Microsoft_.NET
- http://es.wikipedia.org/wiki/Archivo:Diagrama_CLR.jpg
- <http://es.wikipedia.org/wiki/Framework>
- <http://msdn.microsoft.com/es-es/library/zw4w595w.aspx>
- <http://msdn.microsoft.com/es-es/library/aa970268.aspx>
- <http://msdn.microsoft.com/es-es/library/aa970268.aspx#Graphics>
- <http://msdn.microsoft.com/es-es/library/ms747437.aspx>
- <http://blogs.microsoft.co.il/blogs/maxim/archive/2009/03/31/daily-tip-how-to-import-3d-model-to-wpf-from-3d-max.aspx>
- <http://social.msdn.microsoft.com/Search/en-us?query=MAXScript+>