



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

## **INFORME DE PROYECTO DE GRADUACIÓN**

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE CONTROL DE NIVEL Y  
CALEFACCIÓN DE AGUA UTILIZANDO PLATAFORMA LABVIEW DE  
NATIONAL INSTRUMENTS”**

**Previa a la obtención del título de:**

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y  
AUTOMATIZACIÓN INDUSTRIAL**

**PRESENTADA POR:**

**ROY JESUS MAIGUASCA NIEVEZ**

**GUAYAQUIL – ECUADOR**

**2013**

## AGRADECIMIENTO

A Dios, a mi madre Luz María, a mi esposa  
Alexandra, a mi amigo John Piyasagua.

A Dios por brindarme la oportunidad de  
terminar esta etapa de mi vida, a nuestro  
director de proyecto, el Ing. Holger Cevallos  
Ulloa por todo el apoyo.

## DEDICATORIA

Dedico este trabajo a todos quienes  
fueron artífices en mi formación.

A mi familia, al Rvdo. Padre Vicente  
Vivero.

## TRIBUNAL DE SUSTENTACIÓN

---

**Ing. Holger Cevallos Ulloa**

**DIRECTOR DEL PROYECTO**

---

**Ph.D. Boris Vintimilla B.**

**SUBDECANO DE LA FIEC**

---

**Ing. Alberto Larco Gómez**

**VOCAL PRINCIPAL**

---

**Ing. Carlos Salazar L.**

**VOCAL ALTERNO**

## **DECLARACIÓN EXPRESA**

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este Informe nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL).

---

**Roy Jesús Maiguasca Nieves**

## RESUMEN

El diseño y construcción de este prototipo académico se fundamenta en una aplicación específica de control, que es el control de dos variables físicas como son la temperatura y nivel de un líquido, en este caso agua que será depositada en un reservorio o tanque.

Aquí se hará uso de resistencias que serán sumergidas en el reservorio que contiene agua con el fin de calentarla, lo que nos permitirá tener el líquido a la temperatura deseada lista para ser usada en cualquier proceso posterior. Una vez que se tenga el líquido a la temperatura deseada, se abre la llave del tanque y baja el nivel del agua.

El diseño del controlador de este prototipo se lo realizara por medio de un computador provisto del software LabVIEW el mismo que le dará las órdenes a los actuadores en hardware, el cual gracias a su desempeño es ampliamente usado en la Instrumentación Virtual.

En el capítulo 1 se hace referencia a los antecedentes, justificación, a los objetivos y alcances o restricciones del desarrollo del proyecto que estamos planteando.

En el capítulo 2 hacemos referencia al contenido teórico, el cual comprende la introducción a la identificación de sistemas, los controladores PID, los actuadores, el microcontrolador PIC16F886, elemento importante para la ejecución del proyecto, la programación Serial en Circuito ICSP, comunicación serial UART y los dispositivos de medición de un líquido (nivel y temperatura).

En el capítulo 3 se describe el diseño, montaje e implementación de la planta en su totalidad, se va detallando cada uno de los elementos constitutivos de dicha planta y sus respectivas características, desde el hardware hasta el recipiente que contiene el líquido.

En el capítulo 4 se expone la Adquisición, Preprocesamiento de datos, Identificación y pruebas del sistema. Los gráficos, fotos, simulaciones del hardware, las conclusiones y recomendaciones.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIA .....	III
TRIBUNAL DE SUSTENTACION .....	IV
DECLARACION EXPRESA .....	V
RESUMEN.....	.VI
INDICE GENERAL .....	VIII
ABREVIATURAS .....	XII
ÍNDICE DE FIGURAS.....	XV
ÍNDICE DE TABLAS.....	XVIII
INTRODUCCIÓN.....	XIX

### **CAPITULO 1: ANTECEDENTES Y JUSTIFICACION**

1.1 Antecedentes.....	1
1.2 Justificación .....	2
1.3 Objetivos.....	4
1.4 Alcances y Restricciones.....	5



## **CAPITULO 2: CONCEPTOS FUNDAMENTALES**

2.0	Introducción a la Identificación de Sistemas.....	.7
2.1	Controladores PID .....	.8
2.1.1	Estructura de un Controlador PID.....	.9
2.1.1.1	Acción de control proporcional.....	.11
2.1.1.2	Acción de control integral.....	.12
2.1.1.3	Acción de control proporcional-integral.....	.13
2.1.1.4	Acción de control proporcional-derivativa .....	.14
2.1.1.5	Acción de control proporcional-integral-derivativa .....	15
2.2	Estrucutra de un controlador PID.....	16
2.3	Actuadores .....	19
2.4	Microcontrolador PIC 16F886.....	21
2.5	Programación Serial en Circuito ICSP.....	23
2.6	Programación Serial UART .....	28
2.7	Dispositivos o elementos de medición de un líquido (nivel,temp).....	34

## **CAPITULO 3: DISEÑO E IMPLEMENTACION DEL SISTEMA**

3.1	Diseño de la Planta.....	39
3.2	Descripción de los componentes de la Planta .....	40

3.2.1	Computador .....	41
3.2.2	Tanque o Reservoirio .....	42
3.2.3	Actuador Bomba por Módulo Disparador de Relé .....	44
3.2.4	Actuador Resistencia por Módulo Control AC carga Resistiva .....	45
3.2.5	Actuador Válvula de paso de agua .....	49
3.2.6	Sensor Ultrasónico de nivel SRF05 .....	50
3.3.7	Sensor de temperatura DS18B20 .....	57
3.3.8	Tarjeta de adquisición de datos con microcontroladores M.E I&T..	60
3.3.9	Interface de comunicación serial USB-UART I&T 02.....	61
3.3	Montaje de la Planta .....	62

## **CAPITULO 4: IDENTIFICACION Y PRUEBAS DEL SISTEMA**

4.1	Firmware.....	64
4.1.1	Simulaciones del Hardware .....	66
4.2	Estructura de la trama de Comunicación.....	68
4.3	Software LabVIEW .....	72
4.4	Descripción de los bloques de LabVIEW a utilizar.....	74
4.5	Adquisición de Datos .....	87
4.5.1	Señal estímulo aplicada al sistema.....	88

4.5.2 Señal de respuesta del sistema.....	89
4.6 Preprocesado de datos.....	90
4.7 Identificación de La Planta.....	93
4.8 Pruebas del sistema .....	113
4.9 Fotos y Gráficos del sistema .....	115
CONCLUSIONES .....	119
RECOMENDACIONES.....	122
ANEXOS.....	124
BIBLIOGRAFÍA.....	188

## ABREVIATURAS

<b>t:</b>	Tiempo.
<b>s:</b>	Variable compleja de Laplace.
<b>P:</b>	Controlador proporcional.
<b>PI:</b>	Controlador proporcional integral.
<b>PID:</b>	Controlador proporcional integral derivativo.
<b>PD:</b>	Controlador proporcional derivativo.
<b><math>\zeta</math></b>	Coefficiente de amortiguamiento.
<b><math>\omega_n</math>:</b>	Frecuencia natural.
<b>e(t):</b>	Error entre salida y referencia.
<b><math>K_P</math>:</b>	Ganancia proporcional.
<b><math>K_I</math>:</b>	Ganancia integral.
<b><math>K_D</math>:</b>	Ganancia derivativa.
<b>G(s):</b>	Función de transferencia de la planta.
<b>G<sub>c</sub>(s):</b>	Función de transferencia del controlador.
<b>G<sub>directa</sub>(s):</b>	Función de transferencia de trayectoria directa.
<b>y(t):</b>	Salida de la planta.
<b>r(t):</b>	Ruido presente en la planta.
<b>°C:</b>	Grado centígrado.
<b>Q:</b>	Cantidad de calor.
<b>P:</b>	Potencia.

<b>V:</b>	Voltios.
<b>mV:</b>	Milivoltios.
<b>R:</b>	Resistencia eléctrica.
<b>T:</b>	Temperatura.
<b><math>\Delta T</math>:</b>	Variación de temperatura.
<b>A:</b>	Área.
<b>DAQ:</b>	Data acquisition.
<b>Vdc:</b>	Fuente de voltaje.
<b>f:</b>	Frecuencia.
<b>CD:</b>	Corriente directa.
<b>CA:</b>	Corriente alterna.
<b>K:</b>	Ganancia constante.
<b><math>\Delta V_o</math>:</b>	Variación de voltajes de salida.
<b><math>\Delta V_i</math>:</b>	Variación de voltajes de entrada.
<b><math>t_{ss}</math>:</b>	Tiempo de estabilización.
<b><math>\tau</math>:</b>	Constante de tiempo.
<b><math>t_0</math>:</b>	Tiempo de retardo.
<b><math>\tau_{dom}</math>:</b>	Constante de tiempo dominante.
<b><math>Y_0</math>:</b>	Condición inicial de salida.
<b><math>\alpha</math>:</b>	Constante alfa de señales.
<b><math>\beta</math>:</b>	Constante beta de señales.
<b><math>T_s</math>:</b>	Tiempo de muestreo.

<b>T<sub>i</sub>:</b>	Tiempo integral.
<b>T<sub>d</sub>:</b>	Tiempo derivativo.
<b>C<sub>p</sub>:</b>	Capacidad calorífica a presión constante
<b>C<sub>v</sub>:</b>	Capacidad calorífica a volumen constante
<b>W:</b>	Calor
<b>h:</b>	Entalpía
<b>U<sub>D</sub>:</b>	Coeficiente de transferencia de calor
<b>Γ:</b>	Diferencia de temperatura
<b>ce:</b>	Se refiere al calentador eléctrico
<b>q:</b>	Flujo volumétrico,

## ÍNDICE DE FIGURAS

Figura 2.1 Diagrama de bloques de sistema con controlador PID .....	10
Figura 2.2 Regla del trapecio para la aproximación de la Integral .....	17
Figura 2.3 Regla rectangular hacia atrás .....	19
Figura 2.4 Microcontrolador PIC16F886 .....	22
Figura 2.5 Escritura de datos en ICSP .....	27
Figura 2.6 Lectura de datos ICSP .....	27
Figura 2.7 Comunicación serial.....	29
Figura 2.8 Marco para transmisión UART.....	31
Figura 2.9 Módulo de Temperatura.....	35
Figura 2.10 Módulo de Nivel .....	37
Figura 3.1 Diseño de la Planta.....	39
Figura 3.2 Computador con software de control .....	41
Figura 3.3 Tanque o Reservoirio de vidrio .....	43
Figura 3.4 Módulo Disparador de Relé .....	44
Figura 3.5 Módulo P.H.2A I&T 03 .....	45
Figura 3.6 Módulo Control AC Carga Resistiva .....	48
Figura 3.7 Válvula de Acción manual.....	49
Figura 3.8 Sensor SRF05 .....	50
Figura 3.9 SRF05 en modo 1 .....	51
Figura 3.10 Diagrama de tiempos del SRF05 en modo 1 .....	52

Figura 3.11	SRF05 en modo 2.....	52
Figura 3.12	Diagrama de tiempos del SRF05 en modo 2.....	54
Figura 3.13	Dimensiones del SRF05 .....	56
Figura 3.14	Patrón del haz del SRF05.....	57
Figura 3.15	Sensor de temperatura DS18B20.....	59
Figura 3.16	Módulo M.E. I&T 03 .....	60
Figura 3.17	Módulo M.E. I&T 04 .....	61
Figura 3.18	Interface USB-UART I&T 02 .....	62
Figura 3.19	Montaje de la Planta .....	63
Figura 4.1	Firmware en Mikrobasic Pro .....	65
Figura 4.2	Simulación del control ON-OFF .....	66
Figura 4.3	Señales de un control ON-OFF .....	67
Figura 4.4	Trama de datos de sensores .....	68
Figura 4.5	Trama de datos de actuadores .....	69
Figura 4.6	Inicialización en LabView.....	73
Figura 4.7	Panel frontal.....	75
Figura 4.8	Gráficas de nivel y temperatura .....	76
Figura 4.9	Diagrama de conexiones de indicadores .....	77
Figura 4.10	Diagrama de bloques .....	78
Figura. 4.11	Comunicación serial VISA .....	79
Figura 4.12	Configuración de comunicación VISA .....	80
Figura 4.13	Ícono de configuración para puerto serial .....	81



Figura 4.14	Íconos de funciones que configuran el puerto Serial .....	82
Figura. 4.15	Lectura serial de datos.....	83
Figura. 4.16	Indicador de porcentaje de potencia .....	83
Figura. 4.17	Indicador de nivel.....	85
Figura. 4.18	Delay 50ms.....	85
Figura. 4.19	Grabador de datos de nivel.....	85
Figura. 4.20	Grabador de datos de temperatura nivel .....	86
Figura. 4.21	Variables del Proceso .....	86
Figura. 4.22	Señal de estímulo .....	89
Figura. 4.23	Señal de respuesta de nivel.....	90
Figura. 4.24	Cargar datos desde ASCII .....	91
Figura 4.25	Cuadro de dialogo para Load from ASCII .....	92
Figura 4.26	Diagrama para control de nivel.....	96
Figura 4.27	Gráfica para aproximación lineal .....	98
Figura 4.28	Medición del voltaje aplicado a la Bomba de 12 Vdc.....	103
Figura 4.29	Medición de corriente en la Bomba de 12 Vdc .....	104
Figura 4.30	Calentamiento de agua.....	107
Figura 4.31	Mediciones de voltaje y corriente.....	115
Figura 4.32	Planta y PC.....	116
Figura 4.33	Módulos ideas&tecnologias .....	117
Figura 4.34	Llenado del tanque .....	118

## ÍNDICE DE TABLAS

Tabla 2.1 Pines para Programación serie .....	24
Tabla 2.2 Comandos de Programación .....	25
Tabla 2.3 Comandos de Programación .....	26
Tabla 2.4 Voltajes para RS-232.....	32
Tabla 4.1 Valores de caudal en llenado del tanque .....	101
Tabla 4.2 Valores de caudal en Vaciado del Tanque.....	104

## INTRODUCCIÓN

En el mundo de la ingeniería es de vital importancia el estudio del control automático [8]. Resulta imprescindible en sistemas robóticos o procesos de manufactura moderna, entre otras aplicaciones, se ha vuelto esencial en operaciones industriales como el control de presión, temperatura, humedad, viscosidad, nivel y flujo en las industrias de transformación y procesos.

En la época de hoy, un controlador automático es un dispositivo basado en hardware y software [8], que funciona mediante el monitoreo de una señal de error, que es la diferencia entre los valores establecidos (el valor de temperatura, nivel etc. que se requiere que el controlador mantenga) y los valores reales o medidos que el parámetro a controlar tiene. La combinación del sistema a controlar y el controlador empleado tienen generalmente una disposición de lazo cerrado y es el tipo de controlador que empleamos aquí.

En el punto de ajuste se establece el valor deseado del parámetro y se realiza la diferencia del valor real con el valor deseado, y este valor de error  $E$  es pasado al controlador, el cual responde de acuerdo a su implementación, (el nuestro es un controlador ON/OFF, que se explicará) el

controlador pasa su respuesta al elemento de control, que puede ser una válvula, una resistencia eléctrica, una fuente de poder, encargado de actuar para cambiar los valores del parámetro controlado, entonces, el sensor mide el valor del parámetro y dicho valor es enviado al punto "E" donde se realiza la diferencia o señal de error. El hecho de que la salida del controlador (considerando como controlador, el controlador mismo, el elemento de control y el parámetro) esté conectada con su misma entrada le confiere la naturaleza de sistema cerrado o retroalimentado. En nuestro caso en particular, el controlador es un programa que se ejecuta en una PC, los elementos de control son, en primer lugar una bomba de 12Vdc, y en segundo lugar una resistencia de 600W encargada de calentar el agua, el sensor electrónico de temperatura está en conexión con el Módulo de entrenamiento I&T 04 y el parámetro a controlar es la temperatura. El sensor ultrasónico SFR05 también está conectado al Módulo I&T 04 y el parámetro a controlar es el nivel de agua dentro del recipiente de vidrio.

Este proceso se ejecuta continuamente hasta obtener la temperatura y nivel deseados.

# CAPITULO 1

## 1. ANTECEDENTES Y JUSTIFICACION

### 1.1 Antecedentes

La construcción y el diseño de nuestro sistema se fundamentan en una aplicación para el control de dos variables físicas, como son temperatura y nivel.

El proyecto consiste en hacer uso de una resistencia que será sumergida en el líquido a ser calentado, lo que nos permitirá mantener al líquido a una temperatura deseada lista para ser utilizada en cualquier otro proceso posterior.

En cuanto se abre la llave del tanque que contiene al líquido a

temperatura deseada, baja el nivel. El sistema tendrá control también sobre el nivel del líquido en el tanque.

Todos los módulos como son: actuador puente H para la bomba, actuador de resistencia, tarjeta de adquisición de datos, módulos de comunicación, son de la marca nacional ecuatoriana Ideas&Technology (I&T).

## **1.2 Justificación**

El presente trabajo se justifica con la aplicación de tecnologías desarrollada con mano de obra ecuatoriana. Cabe recalcar que la materia prima como son los Microcontroladores, Jack, relé, y otros elementos electrónicos son comprados en almacenes de la localidad.

El control de las variables físicas (temperatura y nivel) sobre el líquido a utilizarse en procesos posteriores, se realizara a través de una PC que tendrá como software LabVIEW 2010 como controlador digital.

Casi todas las industrias pueden usar agua caliente para calentamiento de espacios, duchas y aplicaciones similares.

Pero las que más pueden beneficiarse son las que necesitan agua caliente continuamente en sus procesos. Entre las industrias típicas que utilizan agua caliente en sus procesos destacan:

- Alimentos y productos lácteos.
- Pasta y papel.
- Sector textil.
- Sector Farmacéutico.

### **1.3 Objetivos**

Como objetivos fundamentales establecidos previamente para el desarrollo de nuestro proyecto de graduación serán:

Verificar y controlar en cada instante el nivel del agua, el cual debe ser mantenido dentro de cierto rango preestablecido. Se desea que la bomba no sobrepase un nivel prefijado de agua.

Verificar y controlar el encendido o apagado de la bomba, así como, el control de potencia a través de la carga resistiva para calentar el agua a temperatura deseada.

Otro objetivo de este proyecto es diseñar e implementar un sistema de control de temperatura a través de la comunicación entre el sensor inteligente DS18B20 con su protocolo de comunicación One-wire y el microcontrolador 16F886 instalado en la tarjeta de adquisición de datos, el cual maneja los datos proporcionados por el sensor para mostrarlos por la pantalla del computador.

Mediante el uso de LabVIEW y la tarjeta de adquisición de datos se enviará y recibirá información del proceso con la ayuda de los respectivos sensores y actuadores.

Para llevar a cabo todo lo enunciado anteriormente, se requiere de los siguientes objetivos específicos:

- Desarrollo del firmware en lenguaje MIKRO BASIC PRO que se cargará en el microcontrolador para la adquisición y envío de información hacia y desde el computador.
- Utilizar la herramienta LabVIEW para el procesamiento de los datos que se obtienen desde los sensores.
- Desarrollar dos controladores digitales adecuados para obtener los resultados deseados de control usando LabVIEW.
- Desarrollo del firmware para el control de los actuadores.



## 1.4 Alcances y Restricciones

El alcance fundamental de nuestra obra es la construcción de instrumentos virtuales en LabVIEW para la aplicación del control de dos sistemas físicos, como son nivel y temperatura del agua.

Los modelos eléctricos pueden aplicar el 99% de la energía consumida al agua, mientras que los modelos a gas alcanzan entre un 80% y un 90% de eficiencia. En el caso de los calentadores a gas la energía no utilizada se libera en forma de aire caliente.

Los modelos eléctricos pueden instalarse en lugares cerrados pues no requieren ventilación, en cambio los de gas deben instalarse en lugares ventilados, o si se instalan en lugares cerrados, deben dirigir los gases que expelen a través de un ducto de ventilación al exterior.

Los calentadores eléctricos tienen ventajas de ahorro de espacio, ahorro de energía y agua caliente ilimitada pero sólo son prácticos en países de clima templado o caliente dada su baja capacidad de

calentamiento a grandes flujos de agua, o si la temperatura inicial del agua es muy baja.

Los sistemas eléctricos transportan energía que, si bien cada vez es más pequeña en las líneas de medición o control, en los sistemas de actuadores puede provocar incendios. Por otra parte ha habido casos que vapores ácidos han penetrado en los circuitos electrónicos destruyendo las plaquetas de circuito impreso y algunos componentes, incluyendo conductores, con la consiguiente salida de control del proceso.

Los sistemas neumáticos, tal vez menos precisos, son más tolerantes y pueden realizar prácticamente las mismas operaciones que los electrónicos (es una tecnología que tiene más de un siglo de constante innovación y perfeccionamiento). La diferencia está en el costo y el volumen ocupado. El factor crítico es la calidad del aire disponible (bien seco y sin partículas de aceite u otras) y el rango de presión (máxima y mínima).

## **CAPITULO 2**

### **2. CONCEPTOS FUNDAMENTALES**

#### **2.0 Introducción a la Identificación de Sistemas**

Se entiende por identificación de sistemas a la obtención de forma experimental de un modelo que reproduzca con suficiente exactitud, para los fines deseados, las características dinámicas del proceso objeto de estudio.

Identificar un sistema no es más que determinar modelos matemáticos para los sistemas dinámicos, todo ello basado en la observación y experimentación de los sistemas.

La experimentación es menos compleja que realizar modelos matemáticos basados en leyes físicas, para la identificación de los sistemas dinámicos.

En la experimentación solo nos valemos de señales de entrada y salida de los sistemas, las mismas que son analizadas, llegando a obtener un modelo de dichos sistemas.

Como personas que hemos estudiado ciencias experimentales, sabemos que la experimentación de un sistema dinámico esta sujeto a muchos factores, razón por la cual un modelo no da una descripción exacta de un sistema, pero podemos tomarlo como una aproximación.

## **2.1 Controladores PID**

Los controladores PID son suficientes para resolver el problema de control de muchas aplicaciones en la industria, particularmente cuando la dinámica del proceso lo permite (En general, procesos que pueden ser descritos por dinámicas de primer y segundo orden).

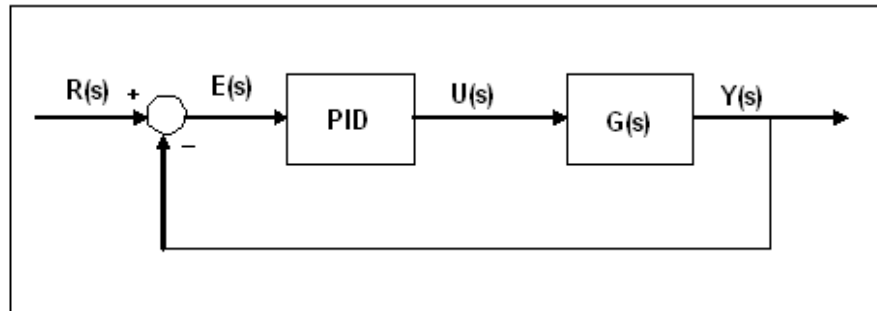
El controlador PID es un elemento importante en sistemas de control distribuido, de tal forma que un gran porcentaje de reguladores utilizados en la industria son del tipo PID, y más específicamente del tipo PI, demostrando que el usuario busca la simplicidad en los algoritmos de control.

El regulador PID se utiliza en el control de procesos industriales con mucha frecuencia como controlador en la implementación de esquemas de compensación en sistemas de control, por lo regular de un grado de libertad.

El esquema más utilizado en sistemas de control de procesos industriales es el de compensación en serie, o cascada. Es por ello que hoy en día gracias al desarrollo tecnológico de la teoría de control, el controlador PID se lo utiliza en el ámbito industrial de todo el mundo, con una gran particularidad para procesos industriales de tipo térmico o químico.

### **2.1.1 Estructura de un Controlador PID**

La figura a continuación representa un lazo de control simple, de una entrada  $R(s)$  y una salida  $Y(s)$ , y en donde  $G(s)$  es la función de transferencia del sistema a controlar, y el bloque PID, la función de transferencia del controlador.



**Figura 2.1 Diagrama de bloques de sistema con controlador PID**

La ecuación en el dominio del tiempo que describe el controlador PID es la siguiente:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt} \quad (2.1)$$

Sabemos que manejar una ecuación integro-diferencial es un trabajo tedioso, entonces pasaremos dicha ecuación al dominio de la frecuencia aplicando la transformada de Laplace:

$$U(s) = K_p E(s) + \frac{K_p}{T_i s} E(s) + \frac{K_p T_d s E(s)}{(\tau_d s + 1)} \quad (2.2)$$

$$\frac{U(s)}{E(s)} = K_p + \frac{K_p}{T_i s} + \frac{K_p T_d s}{(\tau_d s + 1)} \quad (2.3)$$

Si nos damos cuenta, la función de transferencia del controlador PID nos queda de la siguiente forma:

$$C(s)_{PID} = K_p \left[ 1 + \frac{1}{T_i s} + \frac{T_d s}{(\tau_d s + 1)} \right] \quad (2.4)$$

Las partes de la familia del controlador PID incluyen tres Acciones: proporcional (P), integral (I) y derivativa (D). Dichos controladores son los denominados P, PI, PD y PID.

Un controlador PID es la extensión natural del controlador ON/OFF. Este controlador es suficiente para muchos problemas de control. Más del 95% de los lazos de control utilizan el PID.

#### **2.1.1.1 Acción de control proporcional**

Esta acción produce una señal de control proporcional a la señal de error, es decir;

$$u(t) = K_p e(t) \quad (2.5)$$

En el dominio de la frecuencia la función de transferencia queda como:

$$C_p(s) = K_p \quad (2.6)$$

$K_p$  es la ganancia proporcional ajustable. Este tipo de control se aplica a cualquier sistema estable, pero tiene desempeño limitado.

Las características más importantes de esta acción de control son las que siguen:

- Simple.
- Fácil de sintonizar (un solo parámetro).
- Reduce, pero no elimina el error en estado estacionario.

### 2.1.1.2 Acción de control integral

La acción de control puramente integral da una salida del controlador proporcional al error acumulado, la ecuación viene dada por:

$$u(t) = \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (2.7)$$

La función de transferencia del controlador integral es:



$$C(s)_I = K_p \left[ \frac{1}{T_i s} \right] \quad (2.8)$$

Si la señal de error es cero, entonces la señal aplicada es diferente de cero. El error en estado estacionario siempre va a ser cero.

### 2.1.1.3 Acción de control proporcional-integral

La acción proporcional-integral nos da una corrección para compensar las perturbaciones y mantener la variable controlada en el punto que se desea.

Aquí damos las características más relevantes de la acción integral:

- Elimina los errores de estado estacionario.
- Más del 90% de los lazos de control usan PI.
- El sistema se vuelve inestable si  $T_i$  disminuye

Esta acción de control se expresa según la siguiente ecuación en el dominio del tiempo:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (2.9)$$

Y la expresión matemática de la función de transferencia del controlador PI es:

$$C(s)_{PI} = K_p \left[ 1 + \frac{1}{T_i s} \right] \quad (2.10)$$

Donde  $T_i$  es el tiempo integral y es el que se ajusta en la acción integral para reducir el error en estado estacionario.

#### 2.1.1.4 Acción de control proporcional-derivativa

El control proporcional-derivativo, anticipa el efecto de la acción proporcional para estabilizar en menor tiempo la variable controlada después de cualquier perturbación.

Esta acción de control viene definida por la ecuación a continuación:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.11)$$

La función de transferencia que corresponde al control PD es:

$$C(s)_{PD} = K_p [1 + T_d s] \quad (2.12)$$

### 2.1.1.5 Acción de control Proporcional-Integral-Derivativa

Como ya habíamos explicado anteriormente, la acción de control PID es la suma de las tres Acciones descritas anteriormente: Proporcional, Integral y Derivativa.

Es una estructura de control muy particular que es universalmente utilizada en la industria.

Los controladores PID es una estructura fija, solida, ya que se ha demostrado que son muy robustos y muy beneficiosos en el control de una gama de aplicaciones de importancia en la industria.

En la estructura del controlador PID se llego a obtener la función de transferencia del mismo:

$$C(s)_{PID} = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] \quad (2.13)$$

$K_p$ : Constante de Control proporcional.

$T_i$ : Tiempo de Control integral.

$T_d$ : Tiempo de Control derivativo.

## 2.2 Estructura de un Controlador PID Digital

En el análisis numérico se resuelven ecuaciones integro-diferenciales basadas en algoritmos. Estos algoritmos son llevados a programas de alto nivel para que sean resueltos por las computadoras modernas que poseen potentes microprocesadores digitales que resuelven estas ecuaciones con métodos recursivos.

Para el diseño de controladores digitales se lo realiza utilizando técnicas de los sistemas muestreados. Estas técnicas son muy poderosas las cuales nos permiten obtener controles versátiles iguales si utilizáramos compensadores analógicos.

La implementación digital del controlador PID analógico es una aproximación numérica de su ecuación de función de transferencia determinada anteriormente.

Los métodos numéricos que se emplean para resolver las ecuaciones diferenciales de un sistema analógico son las siguientes:

La regla rectangular de EULER hacia adelante.

La regla rectangular hacia atrás.

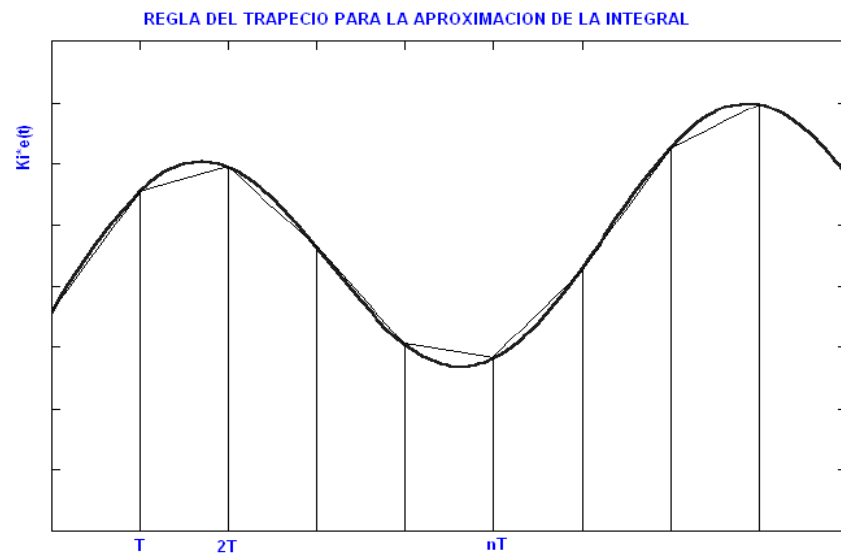
La regla del Trapecio o de Tustin.

La regla del Trapecio nos sirve para resolver la acción de control integral, según esta regla, la integral es aproximada por la ecuación que sigue a continuación:

$$u_{nT} \cong \left(\frac{T}{2}\right) \sum_{i=0}^n [e_{(i-1)T} + e_{iT}] \quad (2.14)$$

Después:

$$u_{nT} \cong u_{(n-1)T} + \left(\frac{T}{2}\right) \sum_{i=0}^n [e_{nT} + e_{(n-1)T}] \quad (2.15)$$



**Figura 2.2 Regla del trapecio para la aproximación de la Integral**

El control de nivel y temperatura implementados en nuestro proyecto será desarrollado en el software LabVIEW.

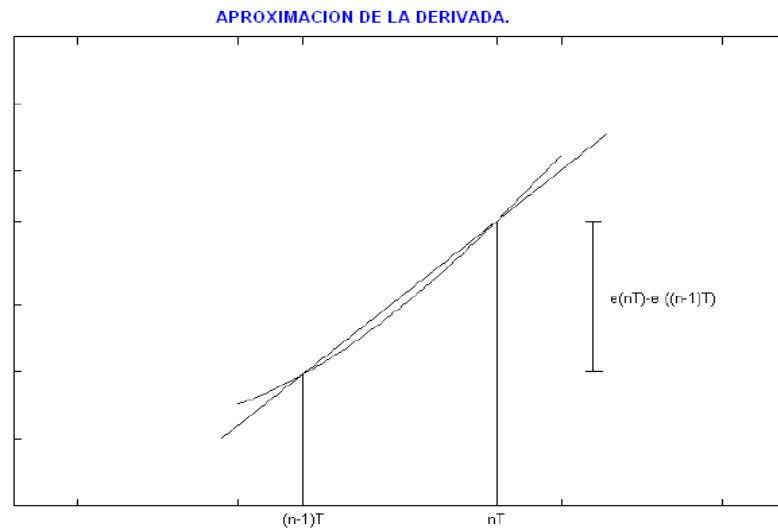
LabVIEW es un programa de alto nivel de entorno gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medidas y muestra de datos.

El programa de LabVIEW para el control diferencial utiliza la regla rectangular de EULER hacia atrás, la misma que viene dada por las ecuaciones (2.16) y ecuación (2.17):

$$\frac{de(t)}{dt} \cong \frac{e(t) - e(t-T)}{T} \quad (2.16)$$

$$\left. \frac{de(t)}{dt} \right|_{t=nT} \cong \frac{e_{nT} - e_{(n-1)T}}{T} \quad (2.17)$$

Las aproximaciones citadas anteriormente (Trapezio y Rectangular hacia atrás) son las más utilizadas por el software LabVIEW para resolver el problema del controlador PID digital.



**Figura 2.3 Regla rectangular hacia atrás**

### 2.3 Actuadores

El objetivo de un sistema de control es gobernar la respuesta del sistema controlado sin que deba intervenir directamente un operario sobre los elementos de salida. El operario manipula solamente las magnitudes de entrada deseadas de ese sistema, llamadas consignas, y el sistema de control se encarga de gobernarlas mediante actuadores.

Los actuadores son elementos que nos permiten controlar directa o indirectamente las variables del proceso. Tales elementos responden a la señal enviada por el controlador en este caso nuestra interface gráfica en

LabView. Por ejemplo en nuestro proyecto tenemos dos actuadores eléctricos:

- Un actuador para el encendido y apagado de la bomba.
- Un actuador para el encendido y apagado de la resistencia.

Los actuadores se clasifican básicamente en dos grandes grupos:

- Los SI/NO (todo o nada).
- Los proporcionales.

En los actuadores proporcionales la posición del elemento de control va a depender de la señal del controlador, lo que requiere muchas veces de un sistema propio de control que asegure la posición deseada. Este elemento se llama posicionador y es un controlador del actuador. Los actuadores también se clasifican según el tipo de energía empleada:

- Eléctricos
- Neumáticos
- Hidráulicos
- Térmicos

El diseño de actuadores es un tema complejo, y hay varios tipos en función de las necesidades y de los materiales que soportan las características del proceso y medio ambiente. Como ya lo habíamos



mencionado antes, en el desarrollo de nuestro proyecto utilizamos dos actuadores eléctricos:

### **Actuador Bomba**

Es un Módulo disparador de relé que sirve para el encendido o apagado de la bomba. Para mejoras en el funcionamiento y eliminación de ruidos, este Módulo fue cambiado por el Módulo P.H.2A I&T03 de mejores prestaciones el cual nos sirve para controlar la velocidad y dirección de giro de la bomba.

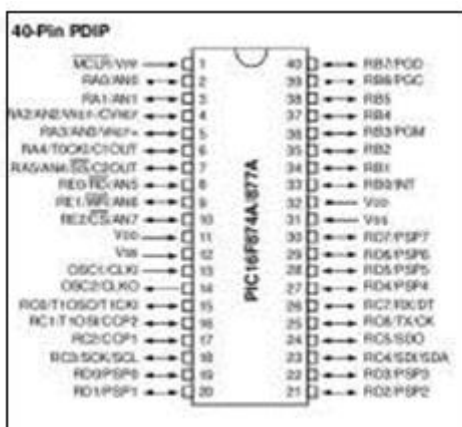
### **Actuador Resistencia**

El Módulo AC carga resistiva nos permite controlar la potencia suministrada a la resistencia. Se alimenta mediante 120 Vac.

## **2.4 Microcontrolador PIC16F886**

El módulo de control es el encargado de obtener datos de todos los sensores del sistema y de manejar los actuadores que mantienen el reservorio en su estado óptimo. El encargado de todo esto es un microcontrolador, en concreto el PIC16F886 de Microchip [9].

Es un microcontrolador RISC con 35 instrucciones diferentes que se ejecutan en cuatro ciclos de reloj, puede funcionar hasta a 20Mhz y tiene 4096 posiciones de memoria de 14 bits para el programa, 192 bytes de RAM y 128 bytes de memoria EEPROM para poder guardar datos permanentemente [9]. Tiene treinta y tres pines programables como entradas o salidas independientes, dos cronómetros/alarmas de 8 bits o uno de 16, puerto serie (USART) y ocho convertidores A/D de diez bits.



**Figura 2.4 Microcontrolador PIC16F886**

El programa grabado en el microcontrolador sirve para la transmisión y envío de información entre los diferentes dispositivos que intervienen en el proceso, como son los sensores, actuadores, PC, etc.

Dicho programa es el llamado FIRMWARE de la tarjeta de adquisición de datos.

En el Anexo A se muestra el programa en MIKRO BASIC PRO que se grabará en el Microcontrolador del módulo de entrenamiento M.E.I&T 04, la grabación se realiza en circuito.

## **2.5 Programación Serial en Circuito ICSP**

Una de las características que pueden inclinar la balanza hacia uno u otro fabricante, en el mercado de consumo de dispositivos electrónicos, es la facilidad de programación. Es decir que el dispositivo presente un sistema de programación sencillo y rápido que minimice los tiempos de desarrollo software.

Este es el caso de Microchip [9] y sus dispositivos PIC que presentan un protocolo de programación serie, que utiliza un número de señales mínimas y además es programable en circuito, pudiendo actualizar el software en la propia tarjeta. A este protocolo de programación se lo denomina ICSP o programación serie en circuito.

El ICSP es el sistema utilizado en los dispositivos PIC de Microchip para programarlo. Es válida para todos los PIC de la gama baja como puede ser el 12C508 y 12C509, los de la gama media desde el 16F84 hasta el 16F877 e incluso los de la gama alta.

El ICSP es un sistema de programación serie síncrona en el que intervienen dos señales, una de ENTRADA/SALIDA para la transmisión y recepción de datos y otra de ENTRADA para la sincronización de la transmisión y recepción de datos. Las líneas utilizadas se ubican en el pin RB6/PGC para la señal de reloj y el RB7/PGD para los datos.

El protocolo trabaja con dos tensiones, una de alimentación (VDD), cuyos valores están entre 4.5 y 5.5 voltios y otra de programación (VPP) cuyo rango oscila entre 12 y 14 voltios. La tabla 1 muestra una descripción de los pines necesarios para llevar a cabo una programación serie.

PIN	FUNCION	TIPO DE PIN	DESCRIPCION
RB6	CLOCK	ENTRADA	ENTRADA DE RELOJ
RB7	DATA	ENTRADA/SALIDA	ENTRADA Y SALIDA DE DATOS
MCLR	VPP	PROGRAMACION	TENSION DE PROGRAMACION
VDD	VDD	ALIMENTACION	TENSION POSITIVA
VSS	VSS	ALIMENTACION	TENSION NEGATIVA

**Tabla 2.1 Pines para Programación serie**

### **Programación del Dispositivo.**

Los Microcontroladores PIC tienen en la parte interna un protocolo de programación. Para Accionar este modo de programación la tensión del pin MCLR debe variar su valor desde VIL (0 voltios) hasta el valor de la tensión de programación VIH (Vpp). El chip entraría entonces en un

modo denominado por Microchip “Program/Verify”. Una vez en este modo se puede acceder a la memoria de programa, de datos y a la de configuración para programarla y verificarla. En este modo RB6 y RB7 son entradas Trigger Schmitt.

COMANDOS	MSB....LSB	Dato
Cargar Configuracion	000000	0,dato(14 bits),0
Cargar dato de Memoria de Pregroma	000010	0,dato(14 bits),0
Leer dato de memoria de Programa	000100	0,dato(14 bits),0
Incrementar direccion	000110	
Inicio de programacion	001000	
Cargar datos en memoria en memoria de datos	000011	0,dato(14 bits),0
Leer dato de memoria de datos	000101	0,dato(14 bits),0
Borrar memoria de programa	001001	
Borrar de datos	001011	

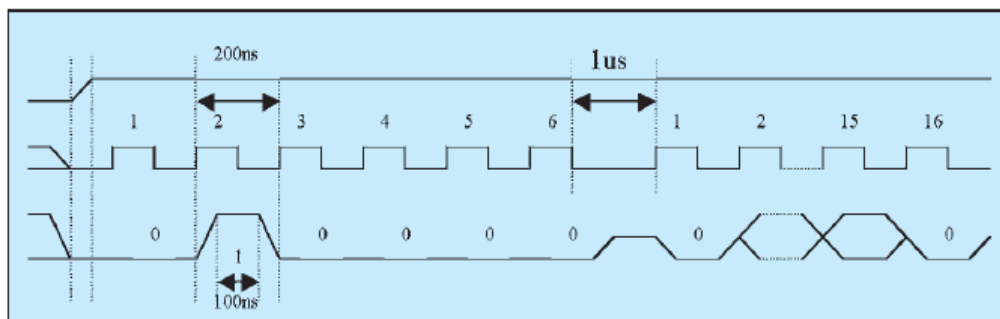
**Tabla 2.2 Comandos de Programación**

El tiempo para pasar del nivel VIL al VIH debe ser inferior a 72 ciclos de reloj para asegurar que el dispositivo ha entrado en este modo. La comunicación con el chip se realizará mediante una serie de comandos específicos, que aparecen en la Tabla 2 y la Tabla 3, y mediante una transmisión serie se podrá acceder a leer, borrar y programar cada uno de los bytes de memoria del dispositivo.

COMANDOS	MSB....LSB	Dato
Cargar Configuracion	XX0000	0,dato(14 bits),0
Cargar dato de Memoria de Pregroma	XX0010	0,dato(14 bits),0
Leer dato de memoria de Programa	XX0100	0,dato(14 bits),0
Incrementar direccion	XX0110	
Inicio de programacion	001000	
Cargar datos en memoria en memoria de datos	XX0011	0,dato(14 bits),0
Leer dato de memoria de datos	XX0101	0,dato(14 bits),0
Borrar memoria de programa	XX1001	
Borrar de datos	XX1011	

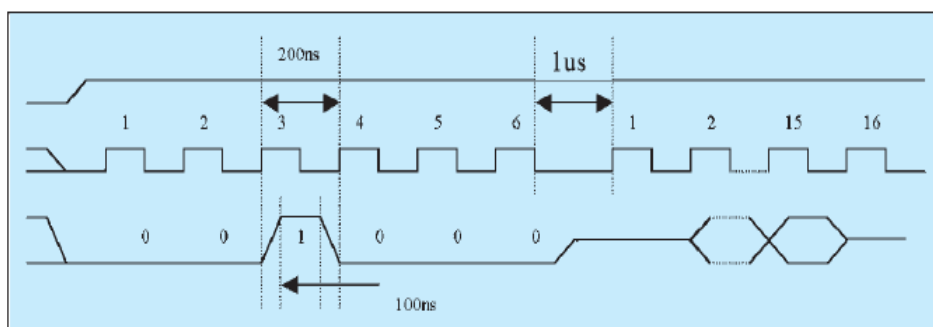
**Tabla 2.3 Comandos de Programación**

Los comandos presentan un valor distinto para que la circuitería interna del chip pueda discriminar uno de otro. Además, no todos los dispositivos pueden utilizar todos los comandos. La transmisión de un comando se realiza en 6 ciclos de reloj. Cada bit de un comando se registra durante el flanco de bajada del pulso de reloj siendo el bit menos significativo (LSB) el primero en enviarse. Los datos deben respetar un tiempo de setup y de hold (ver especificaciones AC/DC de cada dispositivo) anterior al flanco de bajada del pulso de reloj.



**Figura 2.5 Escritura de datos en ICSP**

Existen comandos de control para realizar las Comunicaciones y comandos de datos. Entre los comandos que trabajan con datos (carga y lectura) se debe respetar un tiempo de espera mínimo de  $1\ \mu\text{s}$  entre el comando y el dato. Una vez transcurrido este tiempo el dato se envía en 16 ciclos de reloj enviándose primero el bit menos significativo. En una operación de escritura de datos el bit es registrado en el flanco de bajada del segundo ciclo de reloj como se muestra en la gráfica de la Figura 2.5.



**Figura 2.6 Lectura de datos ICSP**

Durante la operación de lectura dicho bit es transmitido por el PIC en el flanco de subida del segundo ciclo de reloj (ver gráfica Figura 2.6). La escritura de un dato en memoria se realiza con un ciclo de programación formado por un comando de carga de, un comando de inicio de programación y una espera de 10 ms (Figura 2.6) para que la escritura del dato tenga lugar con éxito. Este tiempo de espera va a limitar la velocidad de programación siendo nunca mayor de 100

palabras por segundo. Para llegar a una posición de memoria concreta se utiliza el comando de incremento de dirección.

## **2.6 Comunicación Serial UART**

La USART o Universal Synchronous and Asynchronous serial Receiver and Transmitter es un dispositivo de comunicación serial altamente flexible.

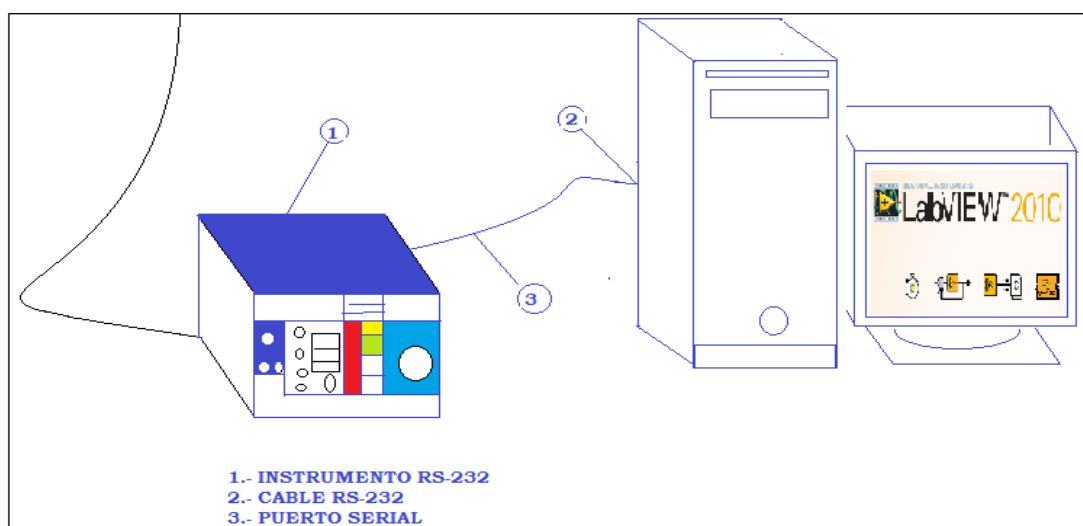
En nuestro proyecto, se realizara una comunicación entre la computadora y el microcontrolador del módulo de entrenamiento M.E. I&T 04. Para poder realizar la comunicación se necesita de varios componentes los cuales se enuncian a continuación:

- Max 232
- 4 capacitores de 10 uF
- 1 Microcontrolador 16F886
- Programador USB
- 1 Cable USB a Serial en caso de que la computadora no tenga puerto serial



La comunicación serial es una forma fácil de comunicar nuestro microcontrolador 16F886 hacia el exterior, ya sea con la PC o con otro microcontrolador.

La comunicación serial transmite datos entre un computador y un dispositivo periférico, tal como un instrumento programable u otro computador. La comunicación serial usa un transmisor para enviar datos a un receptor de un bit a la vez sobre una línea de comunicación. Use este método cuando las velocidades de transferencia de datos son bajas o deba transferir datos sobre distancias largas. La mayoría de los computadores poseen uno o dos puertos seriales, así usted no requiere ningún hardware extra más que un cable para conectar el instrumento al computador o para conectar dos computadores entre sí.



**Figura 2.7 Comunicación serial**

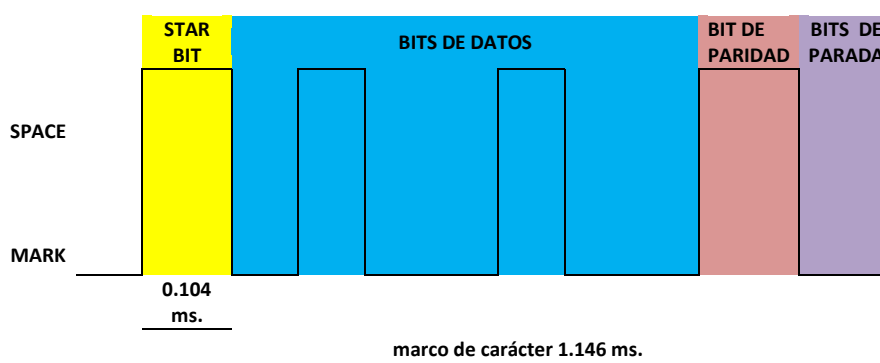
Usted debe especificar cuatro parámetros para una comunicación serial; La velocidad en baudios de la transmisión, el número de bits de datos que codifican un carácter, el sentido de un bit opcional de paridad y el número de bits de parada. Un marco de carácter empaqueta cada carácter transmitido como un único bit de inicio seguido por los bits de datos. La velocidad en Baudios mide que tan rápido se mueven los datos entre los instrumentos que usan la comunicación serial.

Los bits de datos son transmitidos a la inversa y hacia atrás, lo cual quiere decir que se usa lógica inversa y el orden de la transmisión es del bit menos significativo (LSB) hacia el bit más significativo (MSB). Para interpretar los bits de datos en un marco de carácter usted debe leerlos de derecha a izquierda y leer "1" para voltaje negativo y "0" para voltaje positivo.

Un bit opcional de paridad sigue a los bits de datos y el marco de carácter. El bit de paridad, si está presente, también sigue lógica inversa. Este bit se incluye como una forma de verificación de error. Usted especifica por adelantado en la transmisión la paridad para que esta sea par o impar. Si usted selecciona que la paridad sea impar, el bit de paridad se fija de tal forma que el número de "1s" sumen hasta completar un número impar en la cantidad de bits de datos incluyendo el bit de paridad.

La última parte del marco de carácter consiste de 1, 1.5 o de 2 bits de parada que siempre están representados por un voltaje negativo. Si no se van a transmitir más caracteres, la línea permanece en una condición negativa (MARK). La transmisión del siguiente marco de carácter, de existir, empieza con un bit de inicio de voltaje positivo (SPACE).

La siguiente figura muestra un marco típico de carácter codificando la letra "m".



**Figura 2.8 marco para transmisión UART**

RS-232 usa solo dos estados de voltaje, denominados MARK y SPACE. En un esquema tal de codificación de dos estados, la velocidad en baudios es idéntica al número máximo de bits de información, incluyendo los bits de control, que son transmitidos por segundos.

MARK es un voltaje negativo y SPACE es positivo. La ilustración anterior muestra como aparece la señal idealizada en un osciloscopio. La siguiente es la tabla de verdad para RS-232:

SEÑAL	> +3 V=0
SEÑAL	< -3 V=1

**Tabla 2.4 voltajes para RS-232**

El nivel de la señal de salida usualmente oscila entre +12 V y -12 V. En un área muerta entre +3 V y -3 V esta diseñada para absorber ruido de la línea.

Un bit de inicio significa el comienzo de cada marco de carácter. Esta es una transición desde un voltaje negativo (MARK) hacia uno positivo (SPACE). Su duración en segundos es el recíproco de la velocidad en Baudios. Si el instrumento está transmitiendo 9600 Baudios, la duración de bit de inicio y de cada subsiguiente bit es aproximadamente 0.104 ms. Todo el marco de carácter de 11 bits será transmitido en cerca de 1.146 ms.

La interpretación de los bits de datos para la transmisión produce “1101101” (binario) 0 6D (hex). Una tabla de conversión ASCII muestra que esta es la letra “m”.

La transmisión usa paridad impar. Existen cinco “1s” entre los bits de datos, lo cual ya es un número impar, así el bit de paridad se fija en cero.

### **Características de la USART del 16F886**

Sus principales características son:

- Operación Full Duplex
- Registros de transmisión y recepción independientes
- Operación síncrona o asíncrona
- Generador de Baud Rate de alta resolución
- Detección de error
- Filtro de ruido
- Modo de comunicación multiproceso
- Doble velocidad en modo de comunicación asíncrono

El manejo de la comunicación serial presenta muchos beneficios, entre los que destacan, el control de sistemas a través de la computadora realizando cálculos complejos, visualizando y gráficamente datos, entre otros. Es importante destacar que existen muchos programas, los cuales

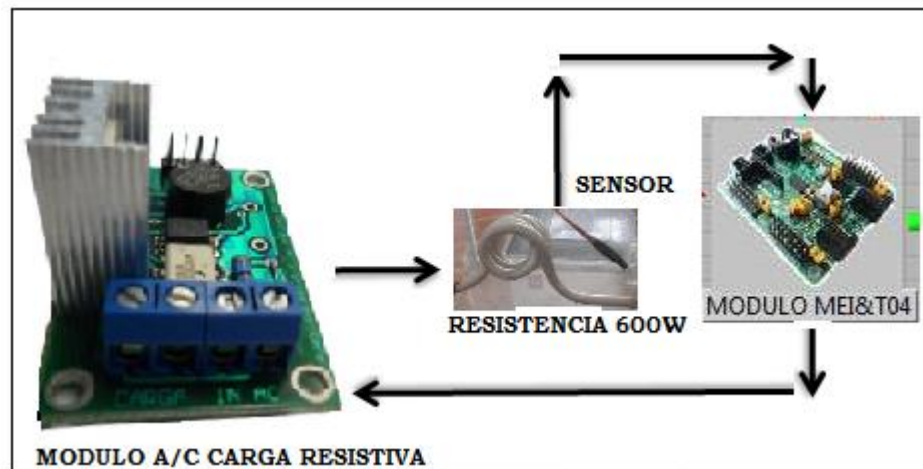
pueden entablar comunicación serial con el microcontrolador, programas como Matlab, LabVIEW, TeraTerm entre otros.

## **2.7 Dispositivos o elementos de medición de un líquido (nivel, temperatura)**

En el desarrollo de nuestro proyecto de graduación, para medir las variables físicas de nivel y temperatura hemos hecho uso dos dispositivos de medición, uno para temperatura que es el sensor DS18BS y otro para el nivel que es el sensor SRFO5. Ambos están descritos de la siguiente manera en dos módulos:

### **Módulo de Control de Temperatura**

Este Módulo se muestra en la figura 2.4 y contiene tanto el sensor de temperatura DS18S20, la resistencia de 600W y el actuador que maneja la resistencia calefactora. Este Módulo se activa de forma manual y automática.



**Figura 2.9 Módulo de Temperatura**

Tanto el sensor de temperatura DS18S20 de DALLAS/MAXIM el Módulo/actuador carga resistiva tienen comunicación con el Módulo de entrenamiento ME I&T04 para adquisición y envío de datos .

El DS18B20 es un termómetro digital de 9 bits que se conecta a través de un bus 1-wire. Tiene una precisión de  $\pm 0.5$  C en el rango de -10 C a +85 C y cada uno de ellos dispone de un identificador único de 64 bits, lo que permite que puedan coexistir varios sensores en un único bus 1-wire. El bus 1-wire puede ser de hasta 100 metros de largo por lo que la colocación de los sensores respecto del Módulo no deberá ser un problema.

El DS18B20 es un integrado con forma de transistor, tiene tres pines de los cuales dos sirven de alimentación (GND y Vdd) y el otro (DQ) es el que se conecta al bus. Este sensor puede funcionar también sin alimentación, con lo que se conoce como alimentación parasita. En este modo de funcionamiento la alimentación la saca directamente de la línea DQ por lo que solo se necesitan dos cables (DQ y GND).

Para proteger el sensor del agua lo montaremos dentro de una ampolla de vidrio o plástico rellena de silicona. Es importante que no entre humedad en el sensor para que no haya posibilidad de cortocircuito o de que se oxiden los pines.

El bus 1-wire es capaz de manejar varios dispositivos conectados a él por lo que con un bus es suficiente, este bus lo manejaremos por el pin RA4. Por el conector 1-wire conectaremos el sensor de temperatura DS18S20 y podemos colocar tantos como queramos repartidos por el tanque, en nuestro caso pondremos uno en el recipiente. Para controlar la temperatura utilizaremos un calefactor típicos conectados a T1 que se activaran cuando la temperatura del tanque suba al nivel preestablecido. Este calefactor disponen de un termostato que los enciende y apaga cuando es necesario, pero como el control de temperatura lo realizaremos nosotros, configuraremos los calefactores a una temperatura un poco mayor de la que necesitemos para que en

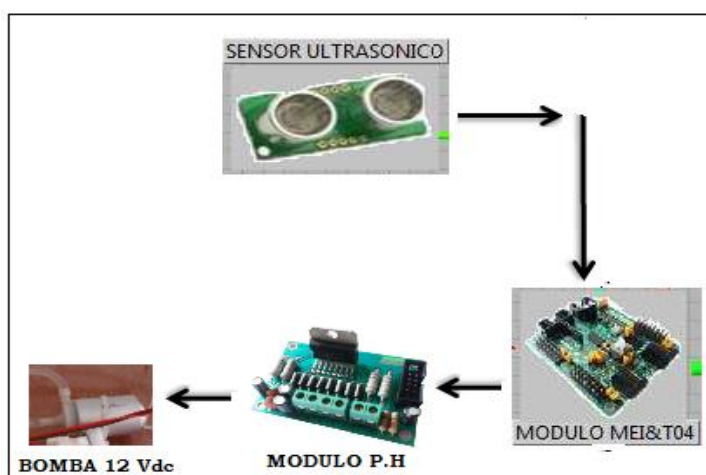


caso de mal funcionamiento del controlador no suba la temperatura excesivamente.

Habitualmente nunca necesitamos enfriar el agua ya que la temperatura ambiente suele ser menor que la necesaria para el tanque. Debemos tener en cuenta que cuantos más calefactores conectemos al Módulo/actuador carga resistiva más potencia necesita soportar por lo que tendremos que instalar uno adecuado a nuestras necesidades o conectar varios en paralelo.

### **Módulo controlador del nivel del agua**

Este Módulo consta de la bomba de 12 Vdc, el sensor de nivel SFRO5 y el actuador P.H.2A I&T 03, que es un módulo para el control de dirección y velocidad de motores DC.



**Figura 2.10 Módulo de Nivel**

Este Módulo se muestra en la Figura 2.5. Aquí el sensor SFRO5 envía señales de nivel al Módulo de entrenamiento, el mismo que envía señales al Módulo puente H para el encendido de la bomba de 12Vdc. En nuestro proyecto hemos previsto conectar un sensor de nivel, el SRF05 que nos indica exactamente la cantidad de agua del reservorio, las entradas a las que conectaremos los sensores son convertidores A/D por lo que simplemente con un sensor que nos proporcione una salida proporcional al nivel de agua tendremos un control mucho más preciso.

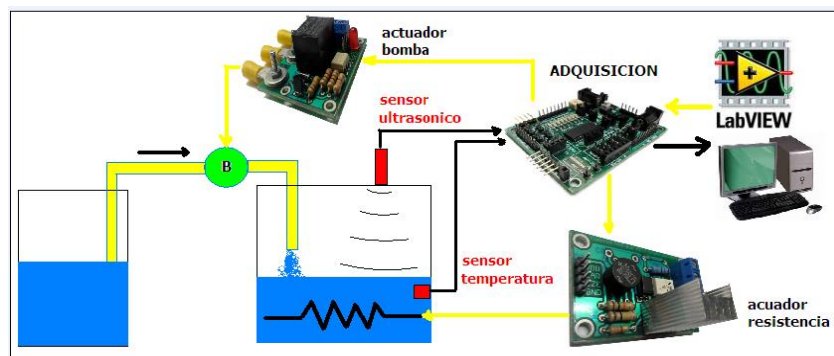
Disponemos de un Módulo/actuador para controlar el nivel de agua, una bomba de llenado, y un sensor de nivel que nos indicara si el tanque está lleno o vacío. Para introducir agua en el tanque usamos la bomba DC y cuando llega al nivel deseado procedemos a calentarla a temperatura preestablecida. Para el vaciado del tanque se lo realiza por gravedad abriendo la válvula de salida. Con esto podemos mantener constante el nivel de agua del tanque simplemente añadiendo un poco cuando baje el nivel.

## CAPITULO 3

### 3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

#### 3.1 Diseño de la Planta

La planta de nuestro sistema de control se muestra en la Figura 3.1 la cual se diseño con fines explicativos y didácticos.



**Fig. 3.1 Diseño de la Planta**

El reservorio en el cual se va a realizar el control de nivel y calefacción del agua es llenado con agua mediante el uso de una

pequeña bomba de 12 Vdc, la cual esta conectada con un actuador que es un driver puente H que permite controlar la velocidad de la bomba DC que a su vez permite variar el caudal que esta bomba entrega al tanque. La medición del nivel de líquido en el tanque es realizada a través de un sensor de sonido, la señal emitida por el sensor se realimenta al controlador que esta instalado en la PC.

El controlador al recibir la señal de medición de nivel compara el valor de esta señal con una señal de referencia establecida por el usuario mediante software, realiza la operación de control necesaria de acuerdo al resultado obtenido por la comparación y envía una señal de control hacia el actuador de la planta. El actuador de la planta esta representado por la combinación de la bomba con el controlador de velocidad, es así que el controlador actúa directamente sobre la bomba para de este modo variar el caudal de entrega al tanque y por consiguiente el nivel de líquido en el mismo.

### **3.2 Descripción de los componentes de la Planta**

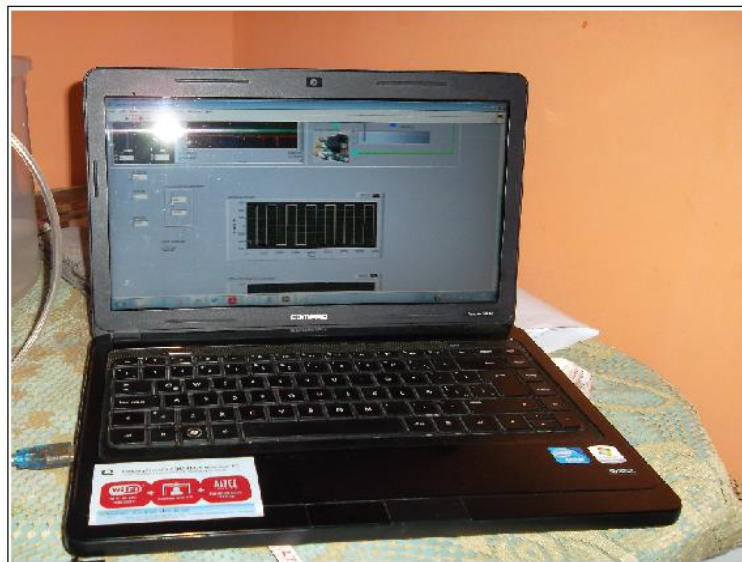
Los componentes principales que constituyen la planta son:

- El computador con el controlador en LabView

- El tanque o resorvorio de vidrio
- Actuador de la bomba
- Actuador de la resistencia
- El sensor de sonido SFR05
- La tarjeta de adquisición de datos
- El sensor de temperatura DS18B20

### 3.2.1 Computador

Nuestra computadora que utilizamos para el desarrollo de nuestro proyecto, es una PC Laptop de HP, Figura 3.2.



**Figura 3.2 Computador con software de control**

A la PC se le instaló el programa LabVIEW 2010, y los demás software para comunicación con la tarjeta de adquisición de datos M.E. I&T 04 de ideas&tecnologías, basada en Microcontroladores.

En esta computadora se ejecuta el programa de LabVIEW que contiene el controlador para nuestra planta modelo de control de temperatura y nivel.

Además se le instaló el software MIKRO BASIC PRO de Mikroelektronika, que sirve para la programación del PIC.

### **3.2.2 Tanque o Reservorio**

El reservorio que utilizamos en el proyecto es el que se muestra en la Figura 3.3, es un envase de vidrio realizado por un taller de la ciudad. El cristal tiene un espesor de 6mm, y sus lados son 15X15X25 cm dando un volumen total de 5625 cm<sup>3</sup>.



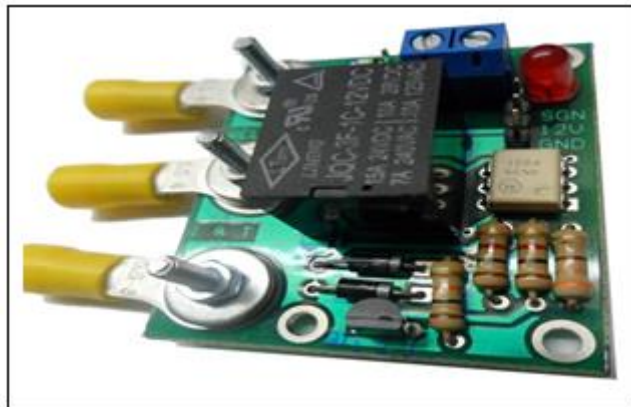
**Figura 3.3 Tanque o Reservorio de vidrio**

También se le realizó a dicho recipiente tres agujeros circulares de 1cm de diámetro, los mismos que sirven para instalar la resistencia en el fondo del tanque. El otro agujero sirve para evacuar el agua caliente a través de una manguera hacia otro tanque donde se colocará el agua caliente.

### 3.2.3 Actuador Bomba por Módulo Disparador de Relé

El módulo disparador de relé se muestra en la Figura 3.3 , sirve para el manejo de cargas de gran Potencia.

Pero en este proyecto la carga es un pequeño motor DC de 12 V.

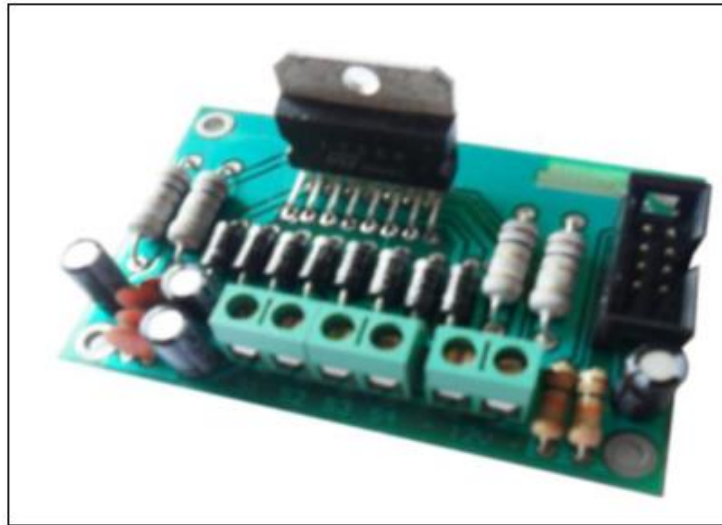


**Figura 3.4 Módulo Disparador de Relé**

Las especificaciones del Módulo Disparador de Relé están disponibles en el ANEXO C.

En las pruebas debimos cambiar el Módulo Disparador de Relé, por el Módulo P.H.2A I&T 03 que posee un PWM para control de dirección y velocidad de motores DC. Este Módulo se muestra en la figura 3.3. El ANEXO D se dan las especificaciones del Módulo puente H.





**Figura 3.5 Módulo P.H.2A I&T 03**

#### **3.2.4 Actuador resistencia por Módulo Control AC Carga Resistiva**

El módulo Control AC Carga Resistiva nos permite controlar la potencia suministrada a una carga AC.

Para el proyecto en cuestión, estamos tomando como carga AC una resistencia de 600W para calentar el agua del reservorio. Para poder elegir la resistencia adecuada que caliente una cierta cantidad de agua que se almacena en el tanque, se realizó un análisis para saber de cuantos watts debe ser la resistencia.

El cálculo que se realizó esta basado en los principios de transferencia de calor, ya que existen diferentes tipos de transferencia de calor. Como ya se decidió anteriormente cuanto es el volumen de agua, se puede realizar el siguiente cálculo.

La ecuación de energía calorífica es:

$$\Delta Q = mc\Delta T \quad (3.1)$$

$\Delta Q$  : variación de energía calorífica.

m: masa del agua.

c: calor específico del agua.

$\Delta T$ : variación de temperatura.

Por otro lado conocemos que:

$$\rho = \frac{m}{v} \quad (3.2)$$

Dónde:

m: masa del agua.

$\rho$ : densidad del agua.

v: volumen del agua.

Despejando m queda:

$$m = \rho v \quad (3.3)$$

$$m = \left(1 \frac{\text{gr}}{\text{cm}^3}\right) \times (3375 \text{cm}^3)$$

$$m = 3375 \text{ gr}$$

$$\Delta Q = (3375 \text{gr}) \times \left(1 \frac{\text{cal}}{\text{grK}}\right) \times (45^\circ\text{C} - 27^\circ\text{C})$$

$$\Delta Q = 60750 \text{ Calorias}$$

Convirtiendo de calorías a joule nos queda:

$$1 \text{ caloria} = 4.186 \text{ Joule}$$

$$\text{Energia} = (60750 \text{cal}) \times (4.186 \text{J}/1 \text{cal})$$

$$\text{Energia} = 254299.5 \text{ J}$$

Finalmente tenemos que:

$$\text{Potencia} = \text{Energia} / \text{Tiempo}$$

$$\text{Potencia} = 254299.5 \text{J} / 732 \text{ seg.}$$

$$\text{Potencia} = 347.166 \text{ watts.}$$

Como se observa la potencia de la resistencia salió aproximadamente de 350 W, el tiempo que se utilizó para llevar una temperatura de 27 grados a 47 grados fue de aproximadamente 12 minutos.

La resistencia es de tipo tubular y sumergible, en este proyecto esta siempre debe estar cubierta por el agua, porque si esta no se encuentra en esas condiciones se podría dañar.



**Figura 3.6 Módulo Control AC Carga Resistiva**

El Módulo AC se muestra en la figura 3.7 y sus especificaciones se indican en el ANEXO B

### 3.2.5 Actuador Válvula de paso de agua

El actuador para nuestra Válvula es manual, es decir, la válvula permanece cerrada cuando se llena el tanque hasta un nivel deseado. Luego de que se Accione el actuador de carga resistiva para calentar el agua a temperatura requerida, se abre la válvula. Se trata de una válvula proporcional Accionada manualmente de  $\frac{1}{4}$ ", Figura 3.5.



**Figura 3.7 Válvula de Acción manual**

Una de las aplicaciones de esta válvula es conectar a la parte inferior del tanque para así regular el caudal de salida del tanque. De esta manera simulamos las perturbaciones de la planta de control de nivel y temperatura. Teniendo un control manual sobre la apertura de la válvula

es difícil tener un conocimiento de que apertura tiene dicha válvula en un momento determinado.

### **3.2.6 Sensor Ultrasónico de nivel SRF05**

El SRF05 es un nuevo sensor de distancias pensado para ser una actualización del clásico SRF04 con el que es compatible, pero además añadiendo nuevas funciones y características. En el modo estándar, el SRF05 se comporta igual que el SRF04 con la diferencia de que el rango de trabajo se ha aumentado de 3 a 4 metros. Esto significa que todo el software que funciona con el SRF04, funciona con el SRF05. Por otro lado, el SRF05 cuenta con un nuevo modo de trabajo que emplea un solo pin para controlar el sensor y hacer la lectura de la medida. Lo que se hace es mandar un impulso para iniciar la lectura y luego poner el pin en modo entrada. Después basta con leer la longitud del pulso devuelto por el sensor, que es proporcional a la distancia medida por el sensor. El SRF05 es mecánicamente igual al SRF04, por lo que puede ser un sustituto de este.

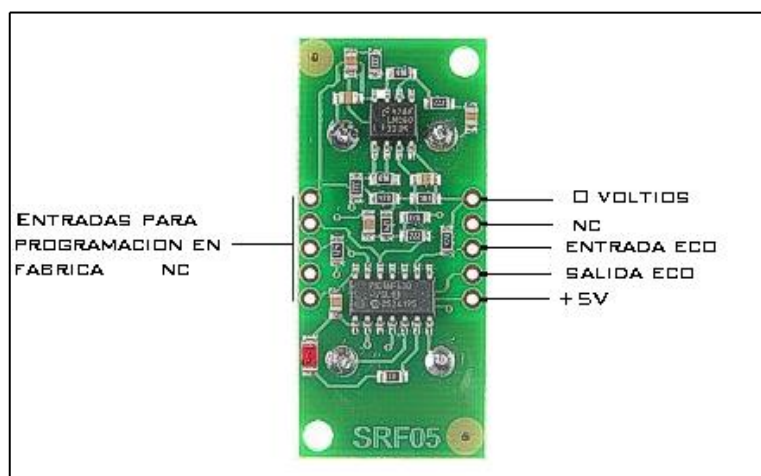


**Figura 3.8 Sensor SRF05**

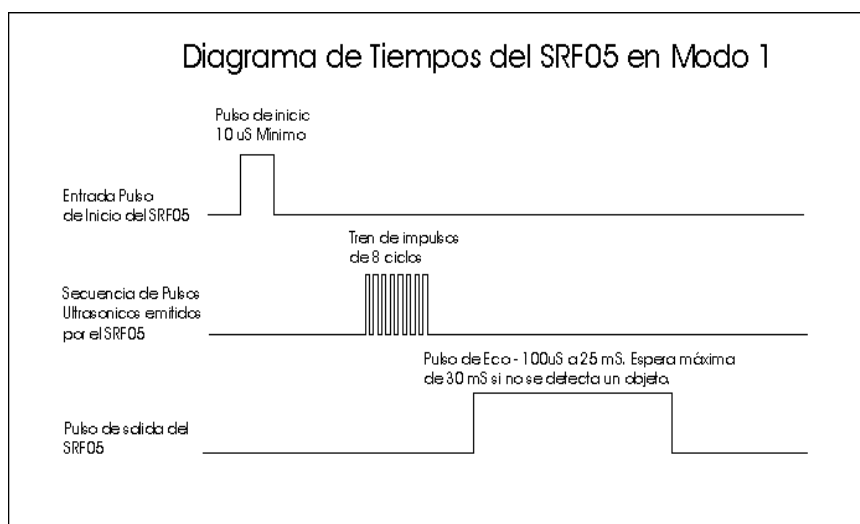
El sensor SRF05 tiene dos modos de funcionamiento, según como se realicen las conexiones.

**Modo1- Compatible con SFR04- Señal de activacion y eco independientes.**

Este modo utiliza pines independientes para la señal de inicio de la medición y para retorno del eco, siendo el modo más sencillo de utilizar. Todos los ejemplos de códigos para el sensor SRF04 funcionarán para SRF05 en este modo. Para utilizar este modo, simplemente deberá dejar sin conectar el pin de modo - el SRF05 integra una resistencia pull-up en este pin.



**Figura 3.9 SRF05 en modo 1**



**Figura 3.10 Diagrama de tiempos del SRF05 en modo 1**



## Modo 2 - Pin único para la señal de activación y eco.

Este modo utiliza un solo pin para las señales de activación y eco, está diseñado para reducir el número de pines en los microcontroladores.

Para utilizar este modo, conecte el pin de modo al pin de tierra de 0v. La señal de eco aparecerá en el mismo pin que la señal de activación. El SRF05 no elevará el nivel lógico de la línea del eco hasta 700uS después del final de la señal de activación. Dispone de ese tiempo para cambiar el pin del disparador y convertirlo en una entrada para preparar el código de medición de pulsos. El comando PULSIN integrado en la mayor parte de los controladores del mercado lo hace automáticamente.

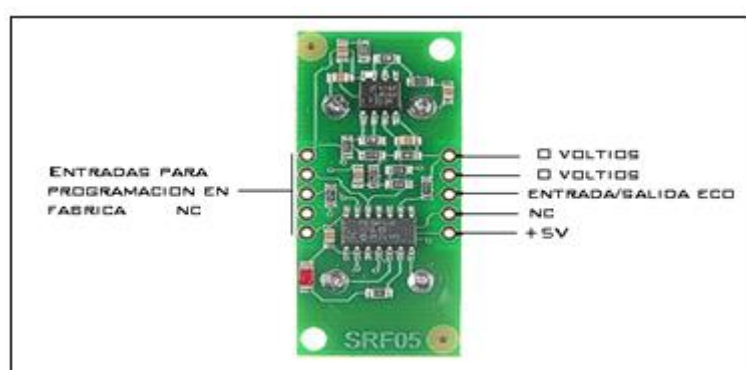
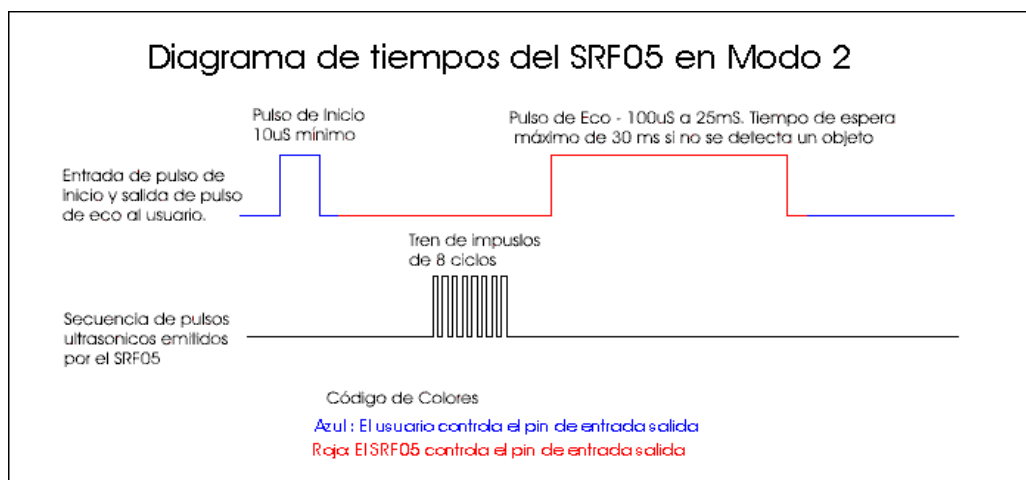


Figura 3.11 SRF05 en modo 2



**Figura 3.12 Diagrama de tiempos del SRF05 en modo 2**

### Determinacion de la distancia

A continuación, se muestran todos los diagramas de tiempo para el sensor de distancias por ultrasonido SRF05 para cada modo. Deberá suministrar un breve pulso de al menos 10µs para disparar la entrada de comienzo del cálculo de distancia. El SRF05 transmitirá una ráfaga de 8 ciclos de ultrasonidos a 40kHz elevando el nivel lógico de la señal del eco (o la línea de activación en el modo 2). Entonces el sensor "escucha" un eco, y en cuanto lo detecta, vuelve a bajar el nivel lógico de la línea de eco. La línea de eco es por lo tanto un pulso, cuyo ancho es proporcional a la distancia respecto al objeto. Registrando la duración del pulso es posible calcular la

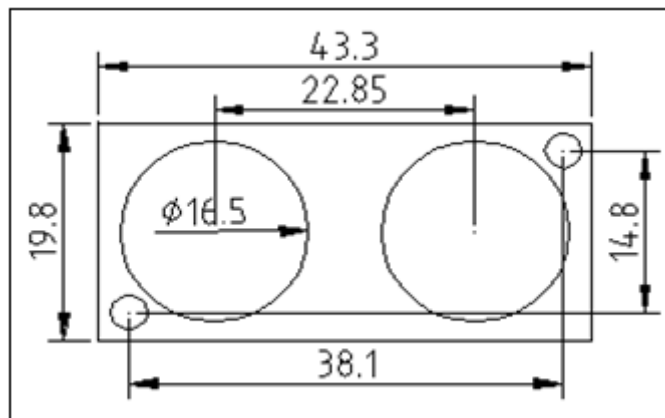
distancia en pulgadas/centímetros o en cualquier otra unidad de medida. Si no se detectase nada, entonces el SRF05 baja el nivel lógico de su línea de eco después de 30ms.

El SRF05 proporciona un pulso de eco proporcional a la distancia. Si el ancho del pulso se mide en  $\mu\text{S}$ , el resultado se debe dividir entre 58 para saber el equivalente en centímetros, y entre 148 para saber el equivalente en pulgadas.  $\mu\text{S}/58=\text{cm}$  o  $\mu\text{S}/148=\text{pulgadas}$ .

El SRF05 puede activarse cada 50mS, o 20 veces por segundo. Debería esperar 50ms antes de la siguiente activación, incluso si el SRF05 detecta un objeto cerca y el pulso del eco es más corto. De esta manera se asegura que el "bip" ultrasónico ha desaparecido completamente y no provocará un falso eco en la siguiente medición de distancia.

### **Terminales de Programación**

Las 5 terminales marcados como "programming pins" (pines de programación) se muestran en las Figuras 3.9 y Figura 3.11, se utilizan sólo una vez durante el proceso de fabricación para programar la memoria Flash en el chip del PIC16F630.



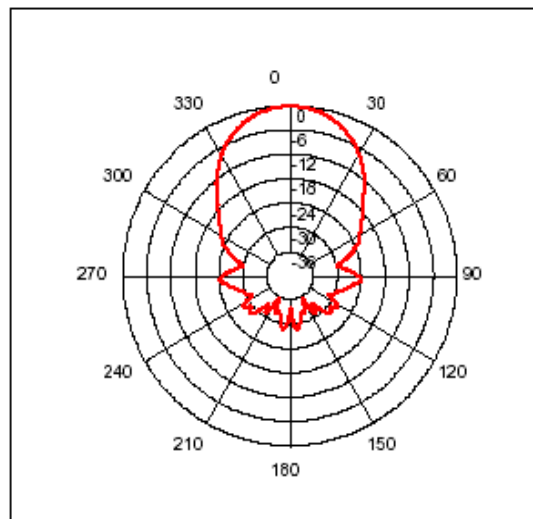
**Figura 3.13 Dimensiones del SRF05**

Los pines de programación de PIC16F630 se utilizan también para realizar otras funciones en el SRF05, por lo que deberá asegurarse de que nada esté conectado a ellos o se interrumpirá el funcionamiento de los módulos. En la Figura 3.13 se muestran además las dimensiones del sensor SRF05, dadas por la hoja de datos del fabricante.

### **Patrón y ancho del haz**

No se puede cambiar el patrón y el ancho del haz. Los usuarios de este sensor se plantean siempre esta pregunta; sin embargo no existe ninguna manera sencilla de reducir o cambiar el ancho del haz. El patrón del haz del sensor SRF05 es cónico mientras que el

ancho del haz es una función del área de la superficie de los transductores y es fijo. El patrón del haz de los transductores utilizados en el SRF05, según la hoja de datos de los fabricantes, es la que se muestra en la Figura 3.14:



**Figura 3.14 Patrón del haz del SRF05**

**Las observaciones, características, figuras y diagramas del SRF05 aquí descritos fueron obtenidos de la hoja de datos en español del mismo sensor.**

### **3.2.7 Sensor de temperatura DS18B20**

Para realizar el proyecto utilizamos el sensor inteligente DS18B20 el cual nos va a permitir obtener las lecturas de la temperatura del

sistema, mediante su protocolo ONE-WIRE, este protocolo de comunicación es en un bus, un maestro y varios esclavos de una sola línea de datos en la que se alimentan.

El sensor de temperatura DS18B20 impermeabilizado. Es bastante utilizado para medir en sitios lejanos, o en condiciones húmedas. Mientras que el sensor es apto hasta los 125 °C, el cable tiene una cubierta de PVC por lo que se sugiere mantenerlo debajo de 100 °C. Como es digital, no hay ninguna degradación de la señal incluso a largas distancias. El DS18B20 proporciona lecturas de la temperatura de 9 a 12 bits (configurable) sobre una interfaz "1-Wire", con sólo un cable de señal (y tierra) estando conectado a un microprocesador central. Utilizable con voltajes de 3.0-5.5V.

Como cada DS18B20 contiene un número de serie único, pueden existir varios DS18B20s en el mismo bus 1-Wire. Esto permite colocar sensores de temperatura en diferentes lugares. Entre las aplicaciones donde esta característica es útil, se incluyen controles ambientales de HVAC, sensor de temperatura interior de edificios, equipo o maquinaria, monitoreo de procesos y control.



**Figura 3.15 Sensor de temperatura DS18B20**

Como sabemos que se debe sensor la temperatura del agua, el DS18B20 puede ser sumergido siempre y cuando sus pines no estén en contacto con el agua, por esta razón al sensor se lo cubrió con silicón y con masilla, tapado con cinta scott.

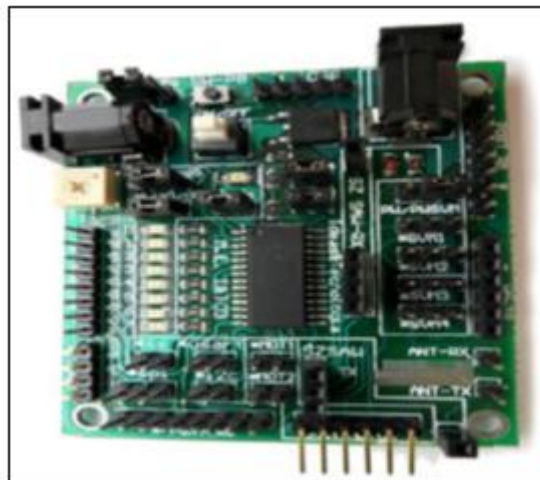
Las características mas importantes del DS18B20 son:

- Interfaz de 1-Wire® para comunicación.
- Cada dispositivo tiene un código serial único de 64-bit guardado en una memoria ROM interna.
- No requiere componentes externos.
- Puede ser alimentado a través de la línea de datos. El rango de voltajes de alimentación es de 3V a 5.5V
- Realiza mediciones desde  $-55^{\circ}\text{C}$  hasta  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  hasta  $+257^{\circ}\text{F}$ )
- Tiene  $\pm 0.5^{\circ}\text{C}$  de exactitud desde  $-10^{\circ}\text{C}$  hasta  $+85^{\circ}\text{C}$
- Resolución seleccionable de 9 a 12 bits.

- Convierte la temperatura a una palabra digital de 12 bits en 750ms (max.)
- Alarmas configurables por el usuario en memoria no volatil
- Aplicaciones de control térmico, sistemas industriales, productos finales, termómetros y cualquier otro sistema que sea sensible térmicamente.

### 3.2.8 Tarjeta de adquisición de datos con microcontroladores M.E.I&T03

El M.E. I&T03 es un módulo de entrenamiento y desarrollo que nos permiten realizar múltiples tareas con el microcontrolador 16F886.



**Figura 3.16 Módulo M.E. I&T 03**

Las especificaciones técnicas de la Tarjeta de adquisición de datos M.E. I&T 03 se presentan en el ANEXO E.



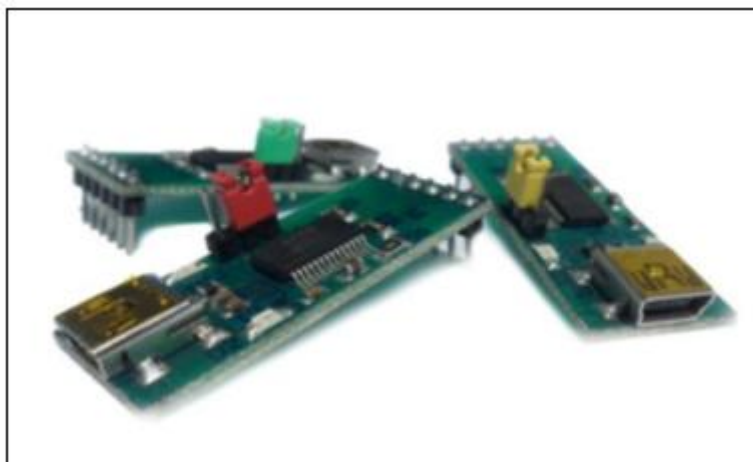


**Figura 3.17 Módulo M.E. I&T 04**

En las pruebas debimos cambiar el Módulo M.E. I&T 03, por el Módulo M.E. I&T 04 que posee un PWM para control de dirección y velocidad de motores DC, además, se le incorporó un Módulo de comunicación UART-USB. Este Módulo se muestra en la figura 3.8. El ANEXO F se dan las especificaciones del Módulo M.E. I&T 04.

### **3.2.9 Interface de comunicación serial USB-UART I&T 02.**

El USB - UART I&T 02 es un Módulo de comunicación serial por puerto USB para hacer interface con un computador o PC, permite alimentar circuitos de 5V.



**Figura 3.18 Interface USB-UART I&T 02.**

Las especificaciones del USB-UART I&T 02 se indican en el ANEXO D

### **3.3 Montaje de la Planta**

Este proyecto fue construido con varios tipos de materiales, desde vidrio, plástico, hasta silicon, etc. El mismo que consta primeramente en un reservorio que fue armado con 5 piezas de vidrio de 10 mm de espesor cuyas piezas fueron ensambladas con silicon y montado sobre una base de acrílico para así lograr ubicar los equipos de control, adquisición, actuadores tanto en la parte superior, inferior e interior del tanque.

En la parte superior se instaló el sensor SRFO5 para medir el nivel del agua sostenido firmemente por un brazo soporte de acrílico. Por este mismo brazo se adhiere la manguera que proviene de la bomba de 12 Vdc, que nos sirve para el llenado del tanque.

En la parte inferior del tanque tenemos la pequeña bomba de 12Vdc, el depósito de reserva, los actuadores y la tarjeta de adquisición de datos. En el interior del tanque hemos colocado la resistencia que nos servirá para calentar el agua. En la figura 3.19 podemos observar la ubicación física de los equipos de control y demás accesorios de la planta.



**Figura 3.19 Montaje de la Planta**

## **CAPÍTULO 4**

### **4. IDENTIFICACION Y PRUEBAS DEL SISTEMA**

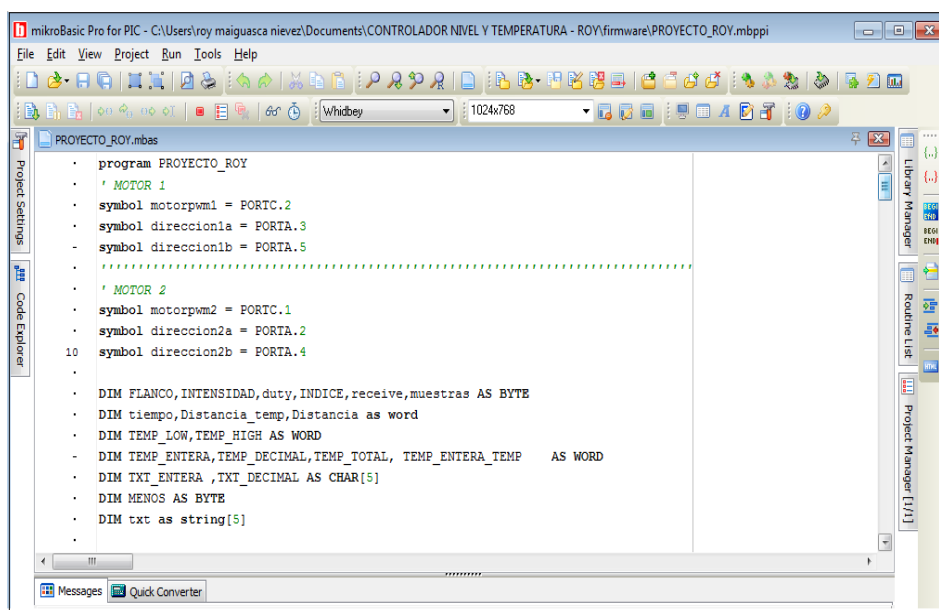
#### **4.1 Firmware**

El Firmware es un bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria de tipo de solo lectura (ROM, EEPROM, flash, etc), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Está fuertemente integrado con la electrónica del dispositivo siendo el software que tiene directa interacción con el hardware: es el encargado de controlarlo para ejecutar correctamente las instrucciones externas.

## MikroC Pro for Pic

El Firmware de nuestro proyecto esta desarrollado en MicroC, perteneciente a MIKROELECTRONICA, muy formal y estructurado con un entorno de trabajo más elaborado, en este lenguaje podemos destacar el uso de la librería del protocolo one-wire para nuestro proyecto.

MikroC PRO for PIC organiza aplicaciones en los proyectos que consisten en un solo fichero de proyecto (fichero con extensión .mcppi) o en uno o más ficheros fuentes (ficheros con extensión .c).



```
program PROYECTO_ROY
·
· ' MOTOR 1
· symbol motorpwm1 = PORTC.2
· symbol direccion1a = PORTA.3
· symbol direccion1b = PORTA.5
·
·
· ' MOTOR 2
· symbol motorpwm2 = PORTC.1
· symbol direccion2a = PORTA.2
10 symbol direccion2b = PORTA.4
·
·
· DIM FLANCO, INTENSIDAD, duty, INDICE, receive, muestras AS BYTE
· DIM tiempo, Distancia_temp, Distancia as word
· DIM TEMP_LOW, TEMP_HIGH AS WORD
· DIM TEMP_ENTERA, TEMP_DECIMAL, TEMP_TOTAL, TEMP_ENTERA_TEMP AS WORD
· DIM TXT_ENTERA ,TXT_DECIMAL AS CHAR[5]
· DIM MENOS AS BYTE
· DIM txt as string[5]
·
·
```

**Figura 4.1 Firmware en Mikrobasic Pro**

En el ANEXO A se muestra el software de comunicación en Mikrobasic.

### 4.1.1 Simulaciones del Hardware

En la simulación se trata de introducir al modelo un conjunto de entradas y verificar el resultado de la respuesta del modelo ante dichas entradas. Si el resultado es similar al conjunto de salidas real de la planta, es decir, que su diferencia tiene un error pequeño, entonces podemos concluir que tenemos un buen modelado.

Para la simulación de nuestro controlador ON\_OFF, haremos uso de LabVIEW. La Figura 4.2 muestra la simulación de la planta de nivel y temperatura.

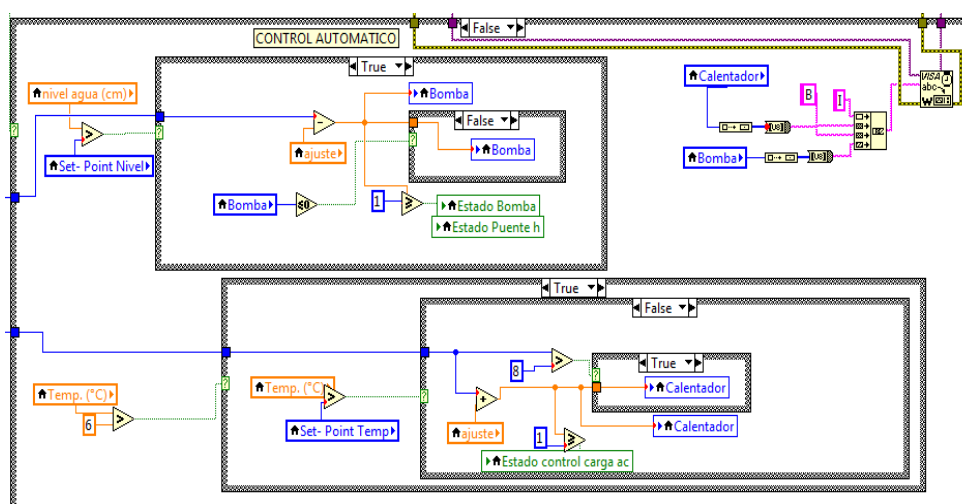


Figura 4.2 Simulación del control ON-OFF

La Figura 4.3 muestra el comportamiento de las señales en un sistema de dos posiciones(ON-OFF), donde el gráfico superior representa la variable controlada  $y$ , para la cual se fija un punto de consigna  $Y_{sp}$ , y el gráfico inferior muestra el comportamiento de la variable manipulada  $m$ , o lo que es lo mismo la salida del controlador.

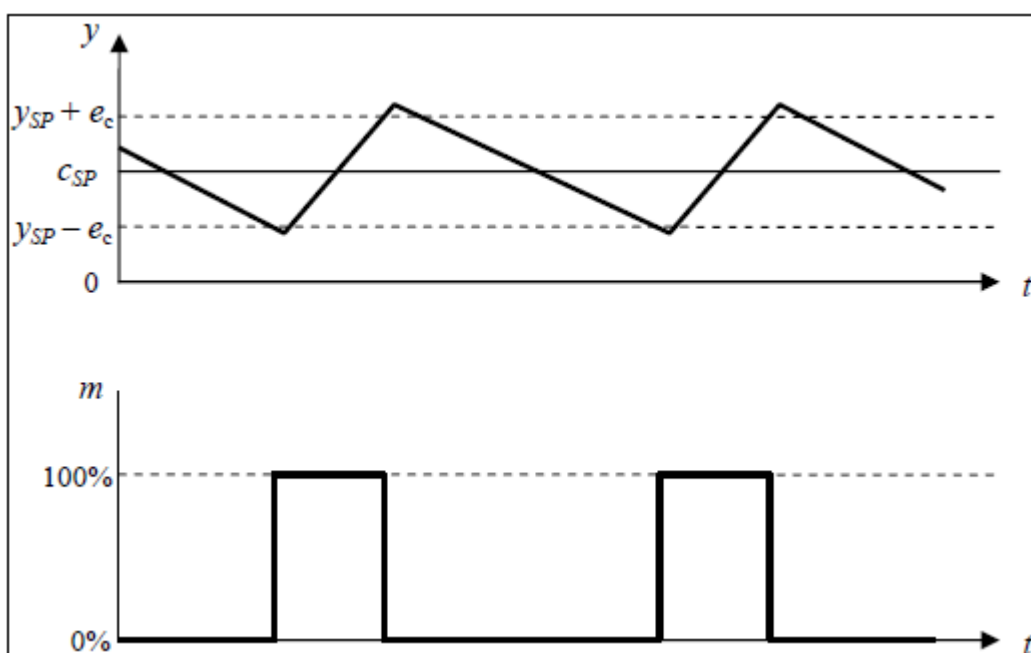


Figura 4.3 Señales de un control ON-OFF

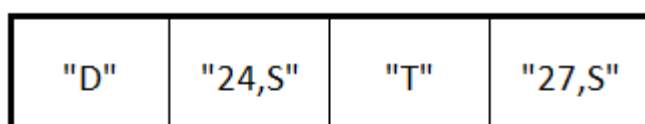
Las dos gráficas se representan sobre el eje del tiempo  $t$ . Como puede verse, cuando el valor absoluto del error sobrepasa un valor crítico predeterminado  $e_c$ , la salida cambia de 0% a 100% o viceversa, por lo que existe una banda muerta alrededor del error cero dentro de la cual no ocurre un cambio en la salida del controlador.

En este sistema la variable controlada siempre oscilará, con una frecuencia que aumenta conforme se disminuye la banda muerta [15].

## 4.2 Estructura de la Trama de comunicación

En la transferencias de datos serie en modo asíncrono, el dispositivo transmisor envía una señal que se conoce como “marca”, usualmente de nivel alto, véase la figura 3, mientras no contenga un dato para transmitir, para indicar que comienza a transmitir un dato válido, el transmisor envía un bit 0, el cual se conoce como señal ó “bit de arranque” ó “bit de inicio”, después del bit de arranque el transmisor envía un dato compuesto de una cantidad predefinida de bits.

Para indicar que se termina la transmisión de un dato, el transmisor envía una señal que se conoce como señal de paro, ésta señal de paro consiste de uno, uno y medio o dos “bits de paro” con nivel alto. Cada dato en la transmisión asíncrona tiene los formatos que se muestran en las figuras 4.3 y figura 4.4.



**Figura 4.4 Trama de datos de sensores**



"I"	BYTE	"B"	BYTE
	0-8		0-255

**Figura 4.5 Trama de datos de actuadores**

### **Arquitectura de Software**

La arquitectura de software para el control de instrumentos usando LabView es similar a la arquitectura para DAQ. Las interfaces de instrumentos tales como GPIB incluye un conjunto de controladores. Use MAX para configurar la interfaz. VISA, Virtual Instruments Software Architecture, es un API comun para comunicarse con los controladores de interfaz y es el metodo preferido que se usa cuando se programa el control de instrumentos en LabView, ya que VISA abstraee el tipo de interfaz usada. Muchos Vis de LabView usados para control de instrumentos usa el API de VISA. Por ejemplo, el Instrument I/O Assistant es un VI Express de LabView que puede usar VISA para comunicarse con instrumento basados en mensajes y convierte la respuesta desde datos en bruto a una representacion ASCII. Use el Instrument I/O Assistant cuando un controlador de instrumentos no este disponible. En LabView, un controlador de

instrumentos es un conjunto de Vis especialmente escrito para comunicarse con un instrumento.

## **VISA**

La Arquitectura de Software para Instrumento Virtual (VISA) es la capa más baja de funciones en los Vis controladores de instrumentos de LabView que se comunica con el software controlador. VISA en sí misma no provee capacidades de Programación de instrumentos. VISA es un API de alto nivel que llama controladores de bajo nivel. VISA puede controlar el VIX, GPIB, Serial o instrumentos basados en computador y realiza los llamados apropiados de controlador dependiendo del tipo de instrumento usado. Cuando se depuran problemas con VISA, recuerde que un aparente problema en VISA puede ser un problema de instalación con uno de los controladores que VISA llama.

En LabView, VISA es una sola librería de funciones que usted usa para comunicación GPIB, serial, VIX e instrumentos basados en computador. Usted no requiere usar paletas separadas de E/S para programar un instrumento. Por ejemplo, algunos instrumentos le dan la opción de tipo de interfaz. Si el controlador de instrumentos LabView estuviera escrito

con funciones de la paleta Functions>>All Functions>>Instrument I/O>>GPIB, esos VIs controladores de instrumentos no trabajarían para el instrumento con interfaz de puerto serial. VISA soluciona este problema entregando un único conjunto de funciones que trabajan para cualquier tipo de interfaz. Por tanto muchos controladores de instrumentos LabView usan VISA como lenguajes E/S.

### **Terminología de Programación VISA**

La siguiente terminología es similar a la usada para Vis controladores de instrumentos:

Resource.- Cualquier instrumento en el sistema, incluyendo puertos seriales y paralelos.

Session.- Usted debe abrir una sesión VISA para un recurso con el fin de comunicarse con él, similar a un canal de comunicación. Cuando abre una sesión para un recurso, LabView retorna un número de sesión VISA, el cual es un refnum único para el instrumento. Debe usar el número de sesión en todas las funciones subsecuentes de VISA.

Instrument Descriptor.- Nombre exacto de un recurso. El descriptor especifica el tipo de interfaz (GPIB, VXI, ASRL), la dirección del dispositivo (dirección lógica o dirección primaria) y el tipo de sesión VISA (INSTR o Event).

Para la comunicación con dispositivos 1-Wire las diferentes interfaces de programa de aplicación (Application Program Interface, API) tienen características comunes que reflejan los patrones fundamentales de la comunicación que emerge del protocolo. API es la plataforma que proporciona el fabricante para acceder a los dispositivos y a la red 1-Wire desde un ordenador. Básicamente se definen como un conjunto de subprogramas o funciones de bajo nivel programadas en un entorno que depende tanto del lenguaje de programación como del sistema operativo utilizado.

### **4.3 Software LabVIEW**

LabVIEW es un software suministrado por la empresa National Instruments orientado a la programación de instrumentos virtuales en un entorno gráfico [4].

A estos instrumentos virtuales los podemos definir como módulos de software que simulan el funcionamiento de uno o varios instrumentos físicos.

Este software debe poseer un hardware controlado por una computadora que le permita acceder a los datos externos al instrumento. Este

hardware en nuestro proyecto son la tarjeta de adquisición de datos, los actuadores, sensores de nivel y temperatura, bomba DC y resistencia de calefacción. En otros pueden ser una tarjeta DSP o un instrumento controlado mediante GPID (Global Purpose Interface Bus) [4], mediante RS- 232 o mediante el bus XVI.



**Figura 4.6 Inicialización en LabView**

De este modo el usuario del instrumento virtual puede observar en la pantalla de un computador los datos recibidos por el instrumento. Al mismo tiempo puede analizarlos con un programa realizado por el mismo LabVIEW. De los resultados del análisis de los datos y mediante el

hardware el programador de LabVIEW puede llegar a controlar un sistema, en este caso, nuestro calentador de agua.

#### **4.4 Descripción de los bloques de LabVIEW a Utilizar**

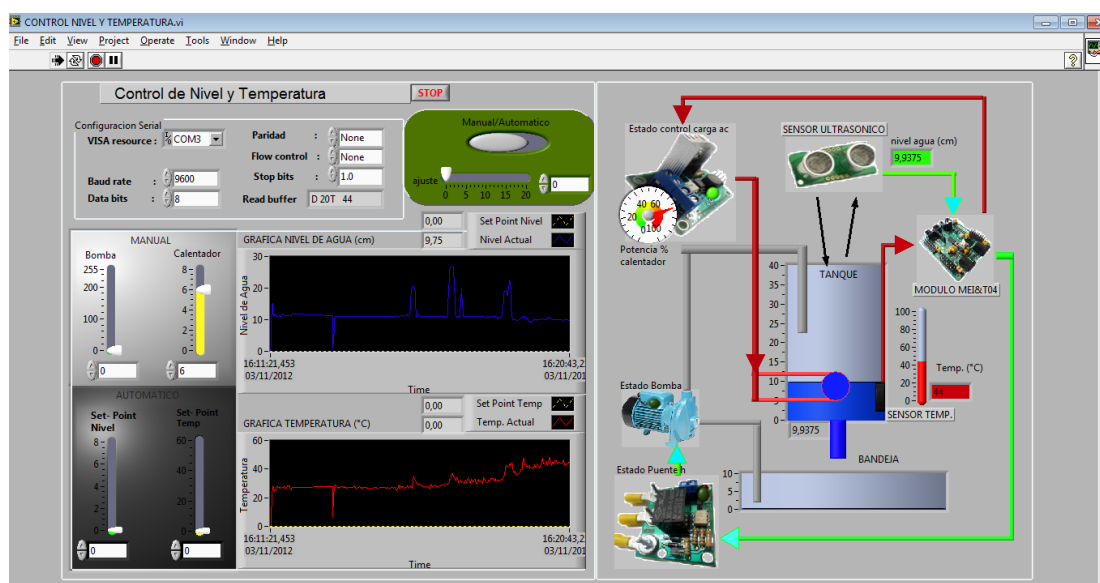
La Descripción de todos y cada uno de los bloques que conforman nuestro sistema la realizaremos desde el panel de frontal hasta el diagrama de bloques.

##### **Panel Frontal**

El panel frontal de nuestro proyecto se muestra en la Figura 4.3 el mismo que contiene en primer plano la configuración de la Comunicación serial, es decir, por que puerto de la PC se va a comunicar el Software con el Hardware, la velocidad de comunicación y los datos.

A continuación tenemos en la parte izquierda todos los controles, los botones de STOP y Manual/Automático, los gráficos de nivel y temperatura, los controles de la bomba y el calentador. También están los controles para colocar los set point de las variables en estudio.

En la parte derecha hemos colocado los indicadores como son el llenado del tanque, indicador de encendido o apagado de la bomba, de carga resistiva. Además figuras de los sensores, del Módulo de entrenamiento, y

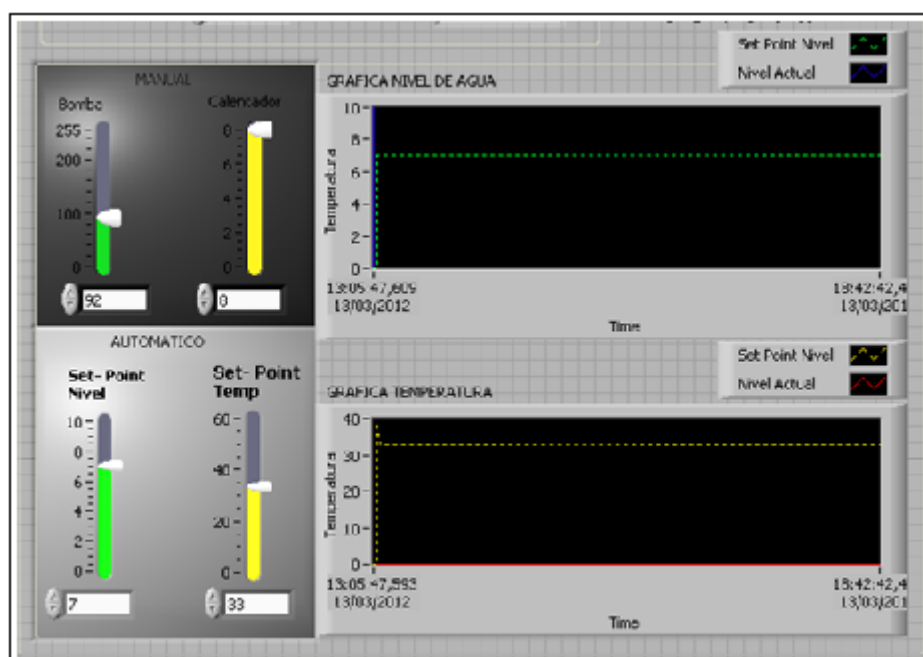


**Figura 4.7 Panel frontal**

por último los indicadores de nivel y temperatura durante la operación del sistema. Además se ha colocado un control de ajuste de encendido de la bomba durante la operación en modo automático del sistema.

## Gráficas de Nivel y Temperatura

Las gráficas de nivel y temperatura se muestran en la Figura 4.7 las mismas que nos muestran las variables en cada instante de tiempo.



**Figura 4.8 Gráficas de Nivel y Temperatura**

SET POINT NIVEL.- Nivel deseado de agua.

SET POINT TEMP.- Nivel deseado de temperatura.

NIVEL ACTUAL.- Este indicador nos muestra el nivel de agua en cada instante.

TEMPERATURA ACTUAL.- Este indicador nos muestra el nivel de agua en cada instante.



BOMBA.- Control manual de encendido y apagado de la bomba.

CALENTADOR.- Control manual de encendido y apagado de la resistencia.

### Indicadores

En la Figura 4.9 se muestra las conexiones de los indicadores en el panel frontal.

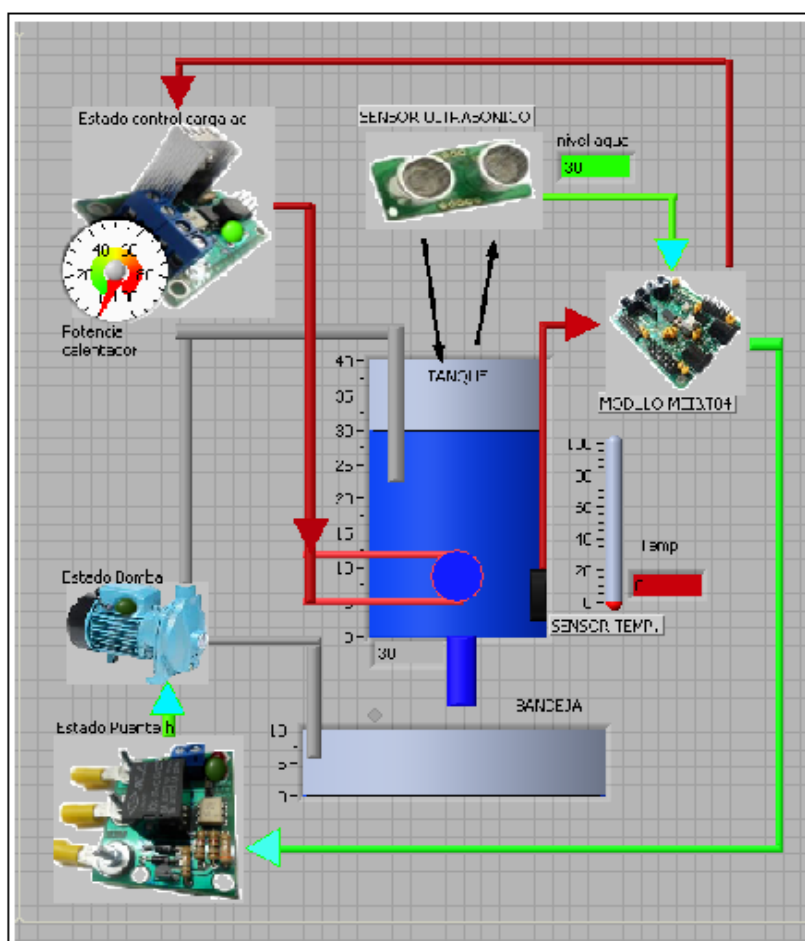
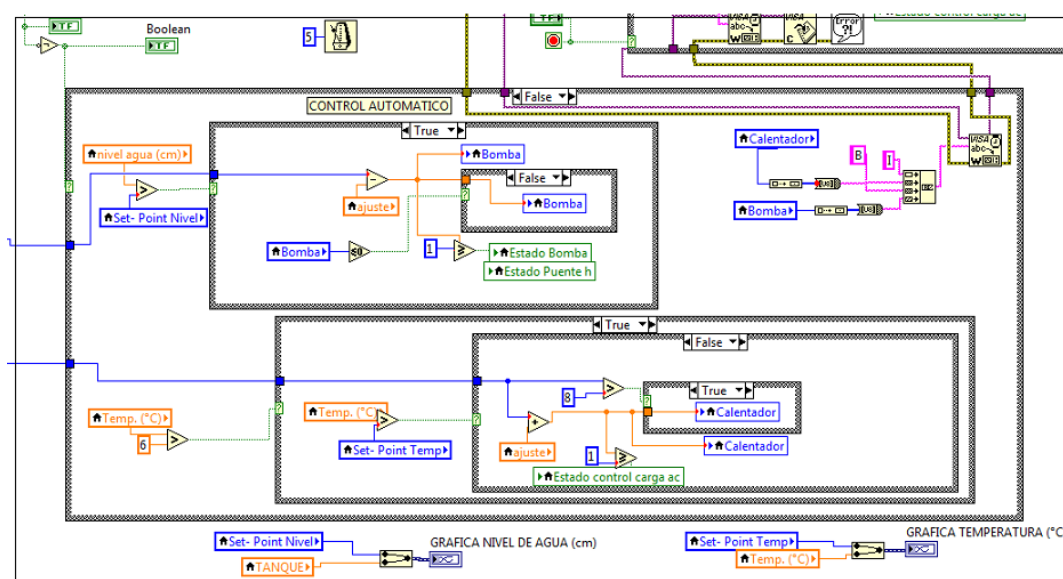


Figura 4.9 Diagrama de conexiones de indicadores

Aquí se muestran las partes más importantes de la planta de nivel y temperatura, como son la bomba, actuadores, sensores, tanques, etc.

### Diagrama de Bloques

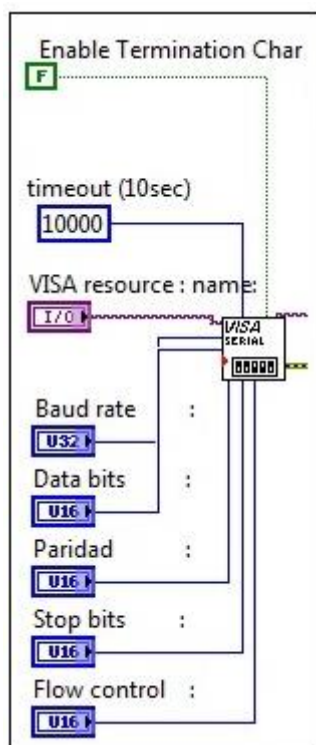


**Figura 4.10 Diagrama de bloques**

En el diagrama se muestra el programa en sí, del proceso a controlar, este es un programa gráfico como sabemos que es LabVIEW. A continuación se detalla su funcionamiento.

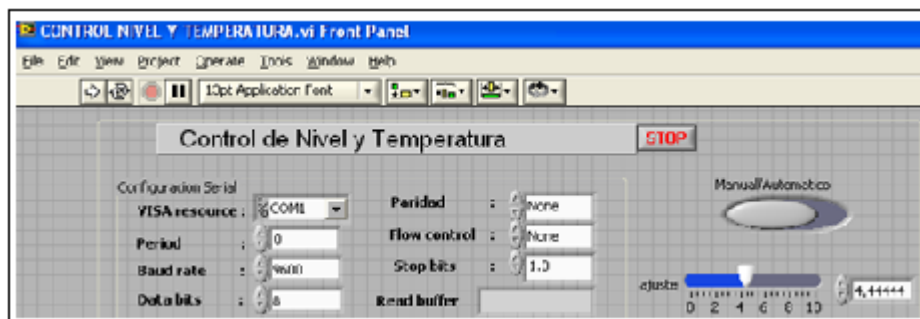
## Configuración de comunicación serial

Configura los parámetros necesarios para establecer la comunicación con los módulos de hardware.



**Figura. 4.11 Comunicación serial VISA**

Como mencionamos anteriormente la configuración de la comunicación serial se realiza mediante el control del VISA resource name, mediante el cual indicamos con un string el puerto de comunicación entre la PC y el Hardware. Esta parte se encuentra en la parte superior del panel frontal y se muestra claramente su funcionalidad en el digrama de bloques de la Figura 4.11.



**Figura 4.12 Configuración de comunicación VISA**

VISA RESOURCE.- Es un string que únicamente identifica la fuente desde donde se va a leer y escribir los datos entre la PC y el Hardware.

BAUD RATE.- Velocidad de transmisión de información en Baudios,9600bits/s.

DATA BITS.- Bits de datos a ser transmitido en una trama, son8 bits

PARIDAD.- No hay bit de paridad.

STOP.- Para el proceso en cualquier momento.

FLOW CONTROL.- No hay flujo de control de datos.

STOP BITS.- Bit de parada en la transmisión de datos.

READ BUFFER.- Lee el buffer de datos

AJUSTE.- Ajusta el encendido de la bomba en modo automático.

MANUAL/AUTOMATICO.- selecciona para operar el proceso en modo manual o automático.

## Comunicación serial con LabVIEW

La comunicación serial es un medio muy usado para la transmisión de datos entre la computadora y un dispositivo periférico. Lab VIEW dispone de una librería denominada Instrument I/O >>Serial que contiene funciones utilizadas para las operaciones del puerto serial. Estas funciones llaman al driver del puerto serial instalado por el sistema operativo de la computadora, en la Figura 4.13 se observa el icono de la función VISA serial que es la que configura los parámetros requeridos para la transmisión de datos.

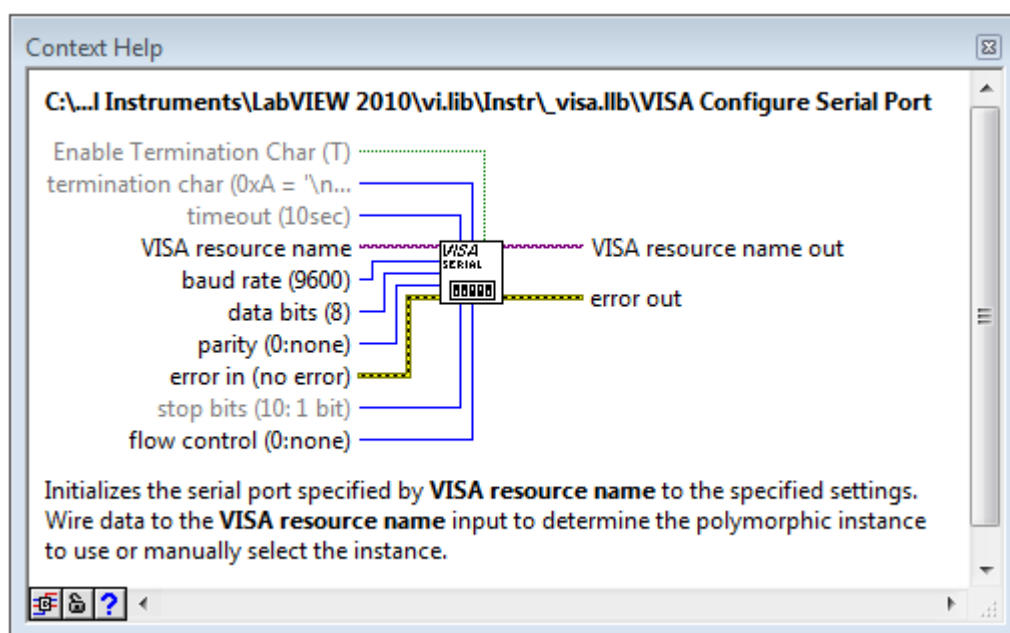
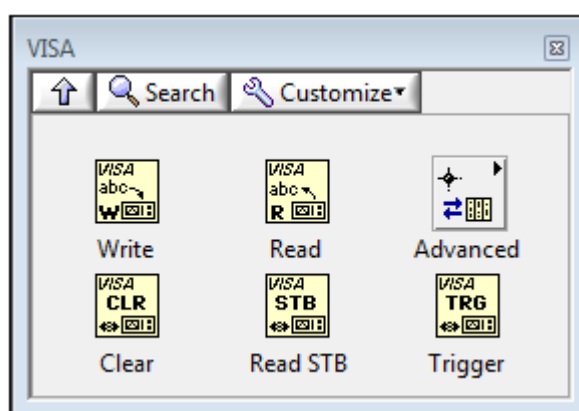


Figura 4.13 Icono de configuración para puerto serial

El comando grafico VISA configura el puerto serial a utilizar en los valores estándar de la velocidad de transmisión, el número de bits por dato, dato de paridad y el bit de parada. En la figura 3.15 se puede observar varios iconos de instrumentación que son funciones para abrir, cerrar, escribir o leer datos a través del puerto serial.



**Figura 4.14 Iconos de funciones que configuran el puerto Serial.**

### **Lectura de datos serial**

Lee la trama de datos completa que en via el hardware hacia el software Labview.

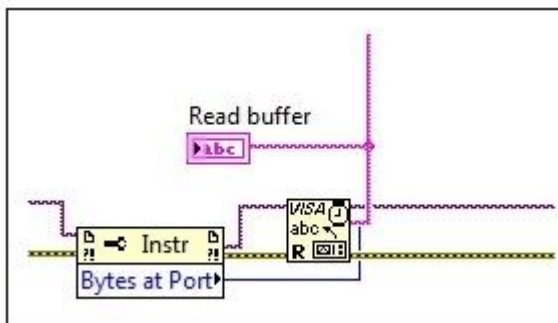


Figura. 4.15 Lectura serial de datos

### Indicador de Porcentaje de potencia en el calentador

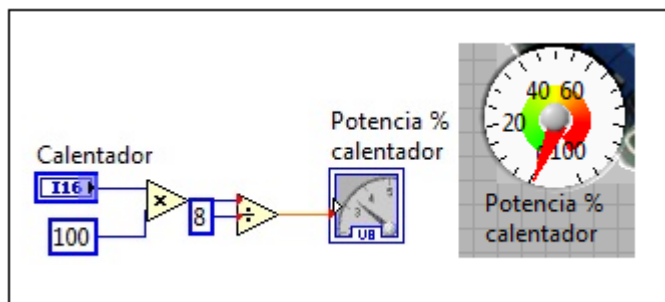
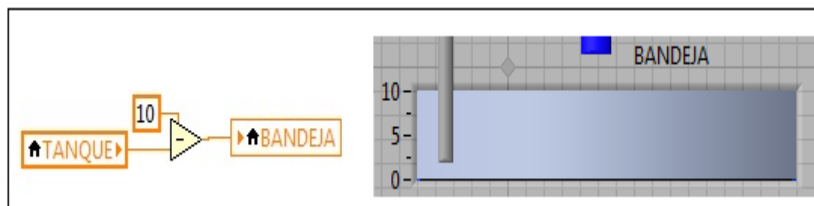


Figura. 4.16 Indicador de porcentaje de potencia

### Indicador de nivel en la bandeja que recibe el agua caliente:

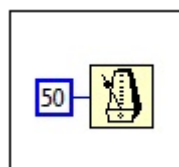
Este indicador se muestra en la Figura 4.17, el cual esta formado en el diagrama de bloques por las variables locales de TANQUE y BANDEJA, conjuntamente con el operador de resta.



**Figura. 4.17** Indicador de nivel

### Delay

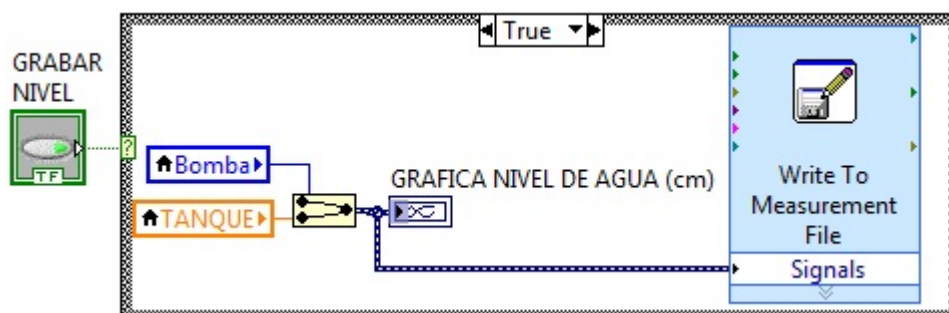
Temporizador de 50 milisegundos por cada iteración:



**Figura. 4.18** Delay 50ms

Bloque que graba los datos de Nivel

Bloque que permite grabar los datos obtenidos en la adquisición de datos de nivel de agua.



**Figura. 4.19** Grabador de datos de nivel



## Bloque que graba los datos de Temperatura

Bloque que permite grabar los datos obtenidos en la adquisición de datos de temperatura de agua:

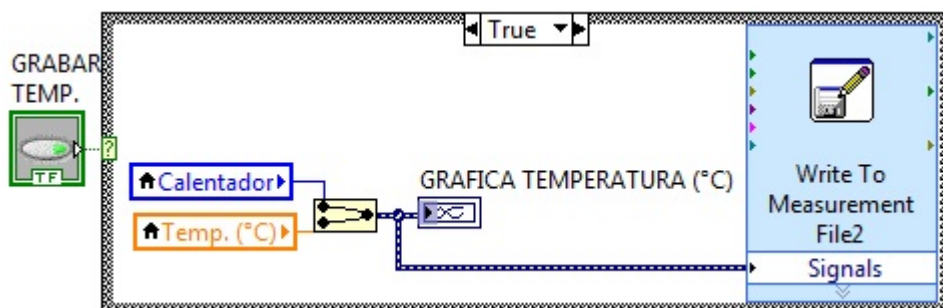


Figura. 4.20 Grabador de datos de temperatura

## Variables utilizadas en el programa



Figura. 4.21 Variables del Proceso

#### 4.5 Adquisición de datos

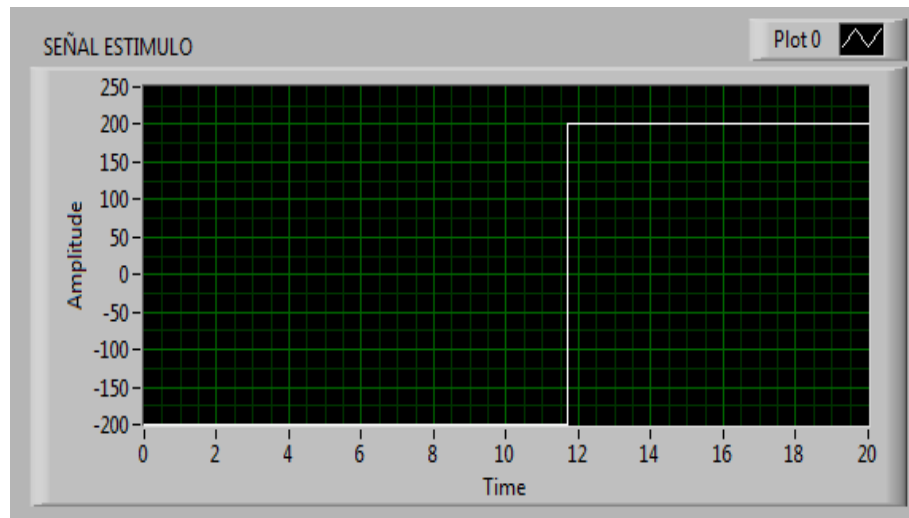
Un sistema de adquisición de datos DAQ usa una tarjeta de adquisición de datos para pasar una señal eléctrica acondicionada a un computador para análisis y registro de datos vía software. Nosotros podemos seleccionar una tarjeta de adquisición de datos que emplee un bus PCI, un bus PXI, el puerto USB o IEEE1394 del computador.

Este proyecto utiliza una tarjeta de adquisición de datos M.E. I&T 04 de Ideas&Tecnologías.

Para llevar a cabo la identificación de la planta debemos primeramente realizar la adquisición de los datos provenientes del experimento. El experimento consiste en generar una señal estímulo al sistema, y adquirir la respuesta del mismo ante dicho estímulo. Una buena identificación de la planta depende básicamente de la correcta aplicación de una señal estímulo para de esta forma obtener una señal de respuesta que nos proporcione la información más útil posible de la misma. Vale decir que a partir de una buena adquisición de datos, lo que significa tener una buena representación del sistema, tenemos casi asegurada una buena identificación de la planta. En este proyecto no realizaremos esta tarea de adquisición-identificación, ya que nuestro proyecto es un simple control todo-nada.

#### 4.5.1 Señal estímulo aplicada al sistema

Nuestro proyecto de nivel y temperatura es un sistema no lineal, lo práctico es aplicar una señal estímulo dentro de un rango de operación cuyos resultados se aproximen a un sistema lineal. Aplicaremos un valor fijo tipo string de 250 que corresponde un voltaje fijo de VDC para encontrar un punto de estabilización de la planta. Desde este punto de estabilización se aplica una señal cuadrada aplicando hacia arriba y hacia abajo el mismo nivel de voltaje. Este nivel de voltaje debe ser aplicado de modo tal que se logre obtener a la salida un comportamiento casi lineal. Es decir que el aumento de voltaje a la señal de estímulo nos provoque un incremento en igual magnitud al decremento que se producirá en la señal de respuesta al disminuir el voltaje de estímulo aplicado. Es así que luego de realizar varias pruebas sobre la planta se determinó que se aplicaría un voltaje fijo DC de 8.24 voltios durante un tiempo estimado de varios segundos durante el cual se lleva la planta a la estabilización.



**Figura. 4.22 Señal de estímulo**

#### **4.5.2 Señal de respuesta del sistema**

Después de aplicar la señal de estímulo a la bomba DC debemos comprobar que la señal que obtenemos por respuesta proporciona la información necesaria para la identificación. Se debe confirmar que esta señal está dentro de un rango muy cercano al punto de operación. En la figura 4.13 podemos observar la señal de respuesta en cm de la planta de nivel y temperatura.



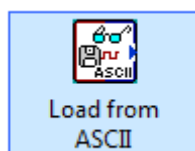
**Figura. 4.23** señal de respuesta de nivel

#### 4.6 Pre procesado de Datos

Después de que se adquirieron los datos de la experimentación se realiza el preprocesado de esos datos, pues debido a fallas de hardware o software se pueden presentar imperfecciones en los datos las cuales conllevan a inexactitudes en la identificación. Estas fallas pueden ser debidas a ruidos externos, problemas en los equipos de medición.

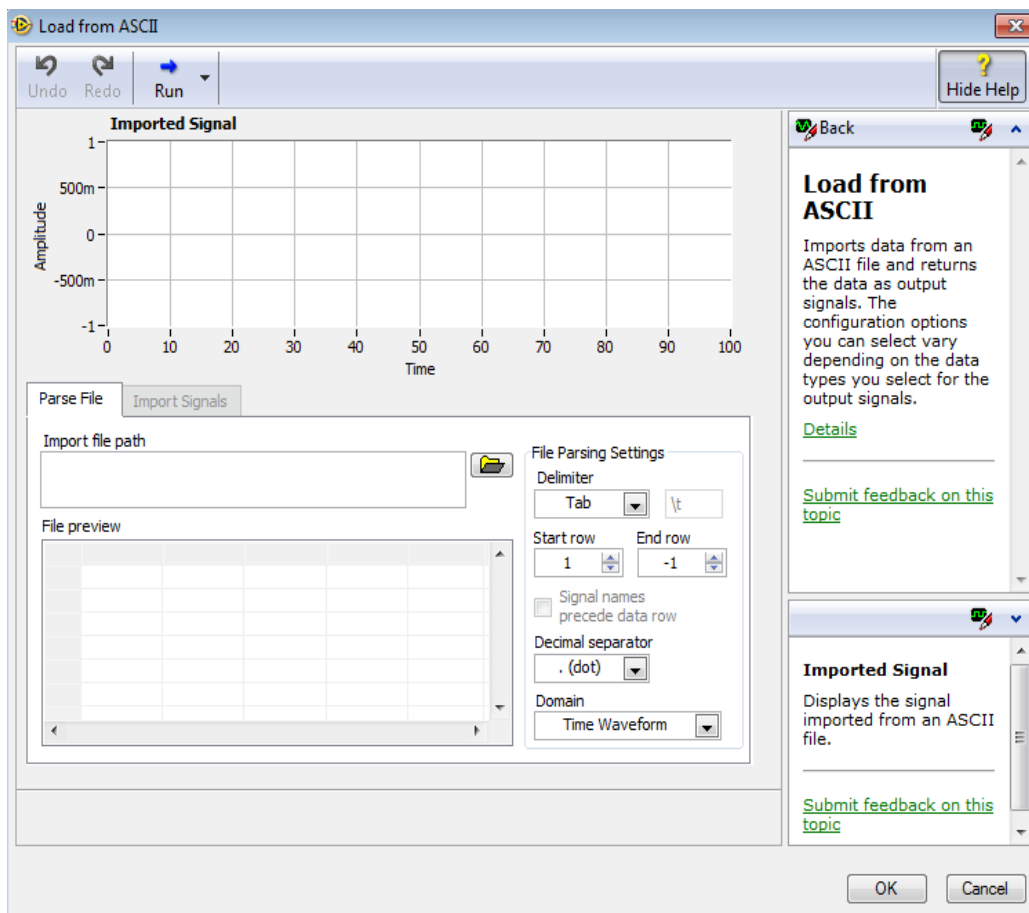
Lo primero que hacemos es inspeccionar visualmente los datos, en busca de datos erróneos o algo que demuestre la existencia de alguna anomalía en el conjunto de datos. Para llevar a cabo este procedimiento abrimos un nuevo VI en el programa LabVIEW. En la ventana de diagrama de bloques colocamos la función cargar desde ASCII (Load

from ASCII), figura 4.28, la cual se encuentra en la paleta de funciones en Signal Express - Analog - Signal I/O - Load ASCII.



**Figura 4.24 Cargar datos desde ASCII**

Cuando colocamos la función cargar desde ASCII automáticamente se abre un cuadro de diálogo como el que se muestra en la figura 4.29. En la primera pestaña denominada Parse File tenemos la opciones: Import file path, que sirve para especificar la ruta en donde se encuentra el archivo que contiene el set de datos usado para la identificación; File preview, donde podemos observar los datos que han sido cargados; File Parsing Settings, donde especificamos el delimitador de los datos, punto decimal a usarse y desde que fila del set de datos empieza la lectura de los datos.



**Figura 4.25 Cuadro de diálogo para Load from ASCII**

En la segunda pestaña denominada Import Signals, seleccionamos cuales son las columnas que deseamos importar desde el archivo que contiene el set de datos dando un clic sobre las columnas. Aquí seleccionamos las columnas 2 y 3 que son las que contienen la señal respuesta y estímulo respectivamente. Una vez especificada la configuración damos clic en OK para guardar dicha configuración.

#### 4.7 Identificación de la Planta

En control automático existen dos métodos principales para obtener el modelo de un sistema:

- Modelado teórico.- Se trata de un método analítico, en el que se recurre a leyes básicas de la física para describir el comportamiento dinámico de un fenómeno o proceso.
- Identificación del sistema.- Se trata de un método experimental que permite obtener el modelo de un sistema a partir de datos reales recogidos de la planta bajo estudio.

El modelado teórico tiene un campo de aplicación restringido a procesos sencillos de modelar, o a aplicaciones en que no se requiera gran exactitud en el modelo obtenido. En muchos casos, además, la estructura del modelo obtenido a partir del conocimiento físico de la planta posee un conjunto de parámetros desconocidos y que sólo se pueden determinar experimentando sobre el sistema real. De ahí la necesidad de recurrir a los métodos de identificación de sistemas.

En la construcción y desarrollo de nuestro trabajo hemos realizado la identificación de la planta mediante software LabView, y la experimentación, además de un modelado teórico analizando el



comportamiento de la misma. Al estar la identificación de la planta ligada con la experimentación, se hace uso de las herramientas de adquisición de datos, acondicionamiento de señales, uso de equipos de medición, y aplicación de teoría de control moderno.

Para la identificación de la planta de control de nivel y calefacción de agua se tuvo que llevar a cabo el diseño de una etapa de experimentación, en la cual se definió las variables a medirse, la señal estímulo a ser aplicada al sistema, el periodo de experimentación, las asunciones a realizarse, en fin las condiciones bajo las cuales se realizaría el experimento.

Nosotros esperamos con la identificación de la planta es obtener un modelo matemático que represente las características de la misma para las condiciones en las cuales esta va a operar. Una vez hallado un modelo matemático se procederá a trabajar con este para así predecir el comportamiento de la planta y en el futuro desarrollar un control más robusto como un PID.

Para la experimentación utilizaremos una señal de estímulo que se aplicará a la planta en lazo abierto, y la lectura de la señal de respuesta de la planta hacia esa señal estímulo. La señal estímulo se aplica al

actuador de la bomba de 12 Vdc y luego de igual forma al actuador de la resistencia, y la respuesta de la planta se mide en la salida del sensor de temperatura. Además observaremos el efecto de la no linealidad de la planta de control de nivel y temperatura. La etapa del software consiste en la programación adecuada en LabVIEW para la realización de la generación de datos, y adquisición de los datos, y la etapa de hardware se refiere a la planta en si.

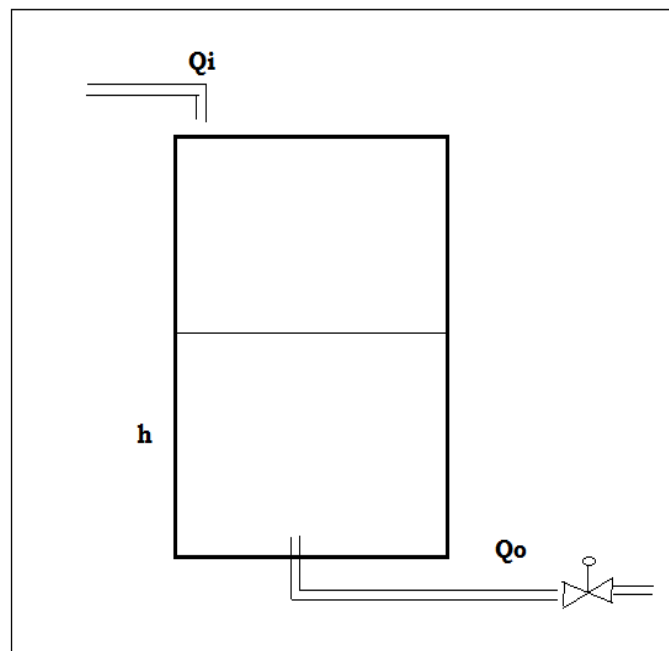
Debemos tener en cuenta el efecto de la no linealidad de la planta de control de nivel y temperatura. Sin embargo dentro de las asunciones realizadas para la implementación de este experimento tenemos que:

Se escogerá un rango alrededor de un punto de operación, rango sobre el cual se tomaran las mediciones. La apertura de la válvula de salida se mantendrá constante a lo largo de la realización del experimento.

El análisis del modelo teórico de la planta de nivel y temperatura es muy importante ya que nos sirve de apoyo o guía dentro del proceso de la identificación de la planta para en el futuro realizar un control PID. Además nos da un conocimiento previo de las leyes que rigen el comportamiento de dicha planta, lo que nos ayuda a mejorar el proceso de identificación. Por este motivo, a continuación revisaremos el modelo teórico de la planta de control de nivel y temperatura.

**Modelo teórico de planta de control de nivel y calefacción de agua.**

Observe el esquema que se muestra en la figura 4.26, en el cual se tiene interés en saber como varía el nivel,  $h$ , altura del agua en el tanque con respecto a los cambios del caudal de líquido de entrada,  $Q_i$ , y de la apertura  $A_p$  de la válvula de salida.



**Figura 4.26 Diagrama para control de nivel**

No importa la geometría que tenga el tanque, el caudal a través de la válvula de salida será proporcional a la raíz cuadrada del nivel del líquido directamente sobre la válvula [14]. De este modo el caudal de salida del tanque,  $Q_o$ , será:

$$Q_o = d\sqrt{h} \quad (4.1)$$

Donde:

$d$  es una constante que involucra el coeficiente de la válvula que esta relacionado con la apertura y cierre de la misma,  $h$  es el nivel de líquido en el tanque.

De esta manera tenemos que el modelo del proceso de nivel en un tanque es como se expresa a continuación:

$$\frac{\partial V(t)}{\partial t} = Q_i(t) - Q_o(t) \quad (4.2)$$

$$V(t) = Ah(t) \quad (4.3)$$

$$A \frac{\partial h(t)}{\partial t} = Q_i(t) - d\sqrt{h} \quad (4.4)$$

Donde:

$A$  es el área del tanque y es uniforme.

$Q_i$  es el Caudal de entrada de líquido al tanque.

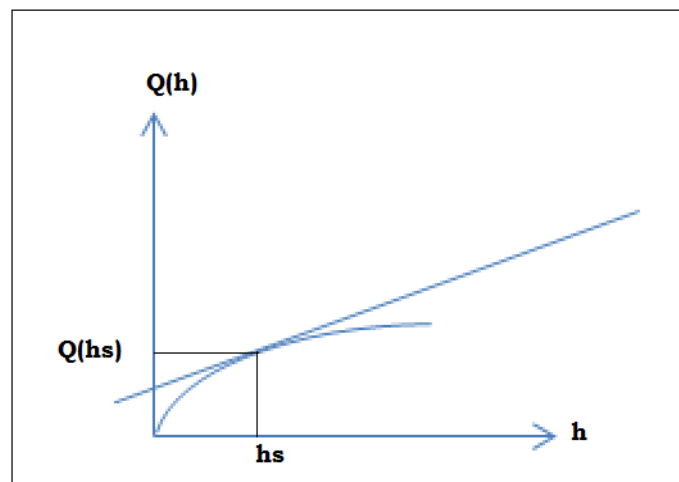
El primer término de la ecuación 4.2 representa el diferencial de caudal en el tanque, considerando que el área del tanque es uniforme, a lo largo de todo el tanque. Observando la ecuación 4.4 podemos ya decir que el modelo representa a un proceso no lineal. La no linealidad es debido a la presencia

del término de raíz cuadrada en la ecuación 4.4. Una opción para poder trabajar con este modelo es linealizar el término no lineal de dicha ecuación. La función no lineal queda definida como:

$$Q(h) = \sqrt{h} \quad (4.5)$$

$$Q(h_s) = h_s^{1/2} \quad (4.6)$$

Para linealizar esta función utilizaremos series de Taylor alrededor de un punto, en este caso el punto de estado estable,  $h_s$ , como muestra la figura a continuación:



**Figura 4.27 Gráfica para aproximación lineal**

Aplicamos primeramente la derivada a la ecuación 4.5, entonces obtenemos la ecuación 4.7 que se muestra a continuación:

$$\frac{dQ(h)}{dh} = \frac{1}{2}h^{-1/2} \quad (4.7)$$

La ecuacion 4.8 indica el valor de la derivada en el punto de estado estable:

$$Q'(h_s) = \frac{1}{2}h_s^{-\frac{1}{2}} \quad (4.8)$$

Observando la figura 4.18 obtenemos una aproximacion de la pendiente de la curva en el punto de interes, la cual es la ecuacion 4.9:

$$Q'(h_s) = \frac{Q(h) - Q(h_s)}{h - h_s} \quad (4.9)$$

$$Q(h) - Q(h_s) = Q'(h_s)(h - h_s) \quad (4.10)$$

$$Q(h) = Q(h_s) + Q'(h_s)(h - h_s) \quad (4.11)$$

$$Q(h) = h_s^{1/2} + \frac{1}{2}h_s^{-\frac{1}{2}}(h - h_s) \quad (4.12)$$

$$h^{1/2} = h_s^{1/2} + \frac{1}{2}h_s^{-\frac{1}{2}}(h - h_s) \quad (4.13)$$

Donde:

$h_s$  es el valor de  $h$  en estado estable.

Entonces reemplazando la ecuación 4.13 en la ecuación 4.4 obtenemos:

$$A \frac{\partial h}{\partial t} = Q_i - d \left[ h_s^{\frac{1}{2}} + \frac{1}{2} h_s^{-\frac{1}{2}} (h - h_s) \right] \quad (4.15)$$

Así en la ecuación 4.15 tenemos la ecuación del modelo de nivel del tanque linealizada en un punto de estado estable  $h_s$ . Ahora definiremos la variable

$y = h - h_s$  e introduciremos la variable  $x = Q_i - Q_{is}$ .

Así:

$Q_{is}$  es el valor de caudal de entrada en estado estable.

Adicionalmente debemos recordar que en estado estable  $\frac{dh}{dt} = 0$ , entonces aplicando esta condición a la ecuación 4.4 tenemos que:

$$Q_{is} = d\sqrt{h_s} \quad (4.16)$$

De esta forma la ecuación 4.15 se convierte en:

$$\tau \frac{\partial y}{\partial t} = Kx - y \quad (4.17)$$

Así  $\tau$  y  $K$  tienen los siguientes valores:

$$\tau = \frac{2Ah_s^{1/2}}{d} \quad (4.18)$$

$$K = \frac{2h_s^{1/2}}{d} \quad (4.19)$$

La constante  $\tau$  es denominada constante de tiempo mientras que la constante  $K$  es la ganancia de estado estable. Aplicando a la ecuación 4.17 la transformada de La Place obtenemos la siguiente función de transferencia, la cual representa teóricamente el modelo del nivel del tanque en un punto de estado estable  $h_s$ .

$$Y(S) = \frac{K}{(\tau S + 1)} X(S) \quad (4.20)$$

Debemos recordar que la ecuación 4.20 es válida solo para el punto de estabilización  $h_s$ , para un nuevo punto de estabilización  $h_{s2}$  los valores de las constantes  $\tau$  y  $K$  serán diferentes.

### **Calculo del valor de constante de tiempo $\tau$ para control de Nivel.**

En este punto realizaremos el cálculo del valor de la constante de tiempo,  $\tau$ . Consideramos un valor de estado estable  $h_s$  de 16,5cm, conocemos el área



del tanque A la cual es  $15 \times 15 \text{ cm}^2$ , un valor desconocido es el valor de la constante “d” que depende de las características de la válvula colocada a la salida del tanque. Sin embargo podemos calcular el valor de esta constante experimentalmente y calcular un valor aproximado de  $\tau$  [14].

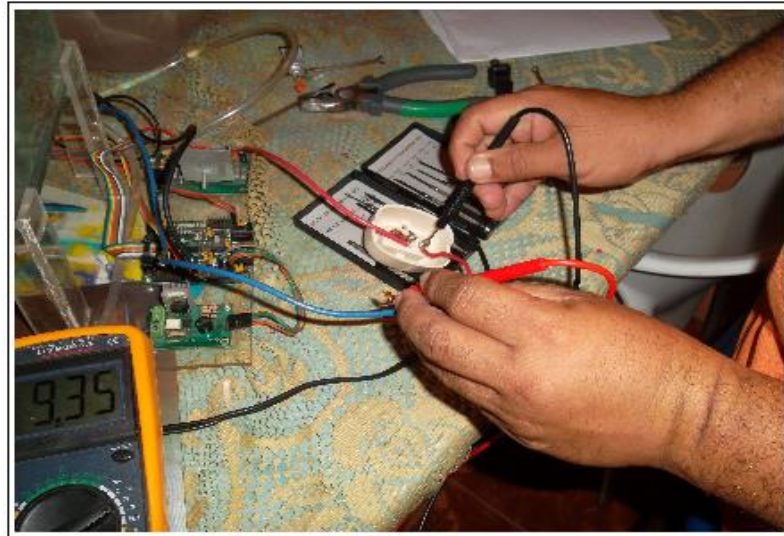
El procedimiento que llevaremos a cabo es el siguiente, necesitamos medir el caudal a la salida de la válvula conectada a la parte inferior del tanque, esta medición se realizará haciendo uso de un cronometro, una fuente de  $V_{dc}$ , un multímetro. Las limitaciones de este método es el error humano en la medición del caudal de entrada y salida. Encendemos la bomba con un interruptor y al mismo tiempo inicializamos el cronometro. La tabla 4.1 muestra los valores obtenidos para el caso de llenado del tanque de forma directa.

Una vez que se alcance el estado estable, registraremos los valores de tiempo, altura del tanque. Estos valores servirán para calcular el valor de la constante d descrita con anterioridad en la ecuación 4.1.

TIEMPO(S)	AREA( $\text{cm}^2$ )	ALTURA DE LLENADO(cm)	GROSOR DE VIDRIO(cm)	VOLUMEN( $\text{cm}^3$ )	CAUDAL DE ENTRADA CALCULADO( $\text{cm}^3/\text{s}$ )
109	225	16,5	0,6	3577,5	32,82

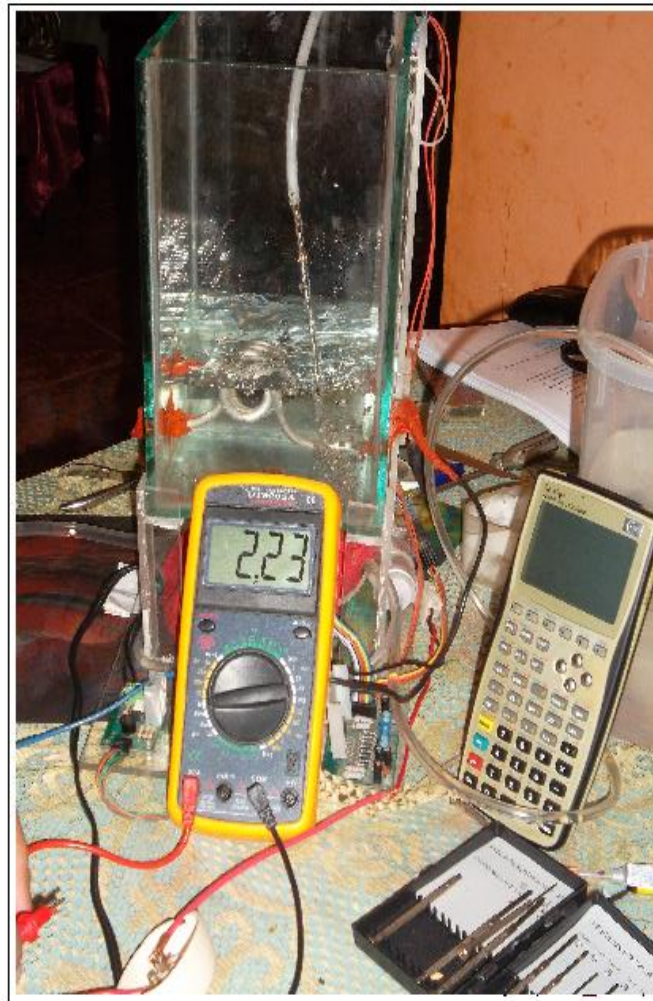
**Tabla 4.1 Valores de caudal en llenado del tanque**

La bomba se enciende con un valor de 9,35 V y una corriente de 2.23 A como se muestran las Figuras 4.28 y 4.29, que son fotografías tomadas durante el experimento.



**Figura 4.28 Medición del voltaje aplicado a la Bomba de 12 Vdc**

El tanque inicialmente está vacío y la Válvula cerrada, haremos trabajar la bomba a voltaje constante para que de esta manera el caudal de entrada sea constante. Debemos tener en cuenta que en estado estable el caudal de entrada será igual al caudal de salida por lo tanto una vez que se alcance el estado estable el caudal que medimos a la entrada será igual al caudal que existe a la salida de la válvula.



**Figura 4.29 Medición de corriente en la Bomba de 12 Vdc**

Ahora para determinar el caudal de salida del tanque lo calculamos de manera analoga al llenado, pero ahora apagamos la bomba y abrimos la Válvula de salida, vaciando el agua por gravedad. Los valores experimentales se muestran en la tabla 4.2.

TIEMPO(S)	AREA(cm <sup>2</sup> )	ALTURA DE VACIADO(cm)	GROSOR DE VIDRIO(cm)	VOLUMEN(cm <sup>3</sup> )	CAUDAL DE SALIDA CALCULADO(cm <sup>3</sup> /s)
83	225	12,5	0,6	2677,5	32,26

**Tabla 4.2 Valores de caudal en Vaciado del Tanque**

Observando los valores determinados en la Tabla 4.1 con los de la Tabla 4.2, vemos que el caudal de entrada es igual a de salida, como lo mencionamos anteriormente.

Con los datos obtenidos del experimento y la ecuación 4.1 realizamos el cálculo de la constante  $d$ , con el cual los valores necesarios para calcular la constante  $\tau$  ya están completos. Reemplazando los valores de  $A$ ,  $h_s$  y  $d$  en la ecuación 4.18 tenemos:

$$\tau = 15 \times 15 \frac{2 \times 16.5^{1/2}}{8.08}$$

$$\tau = 226.22$$

El valor obtenido para  $\tau$  es de 226.22s, este valor puede compararse mas adelante con los resultados de la identificación de la planta.

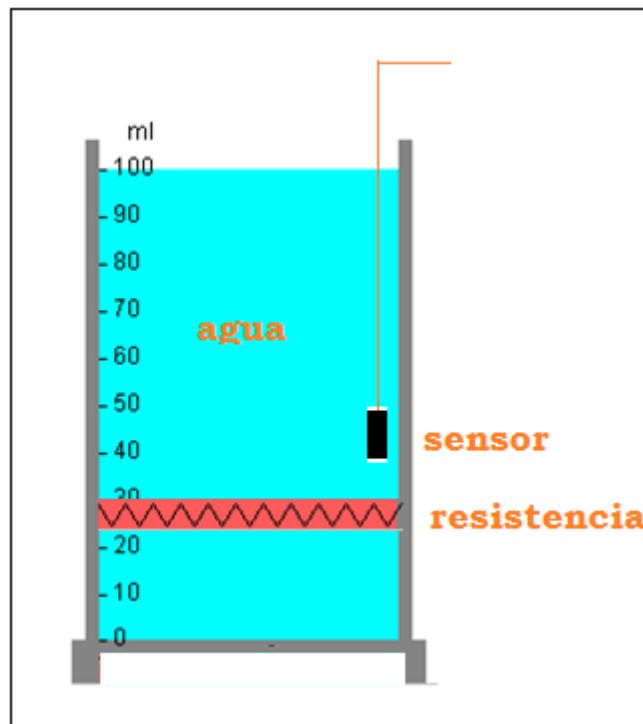
## Modelado teórico para el Control de Temperatura

En este proyecto se calienta un recipiente de agua con una resistencia eléctrica de potencia  $P$  constante [11]. Se observa el incremento de la temperatura del agua. Si la potencia de la resistencia eléctrica es suficiente o las pérdidas no son grandes, el agua alcanza la temperatura de ebullición.

En el caso de que la potencia de la resistencia eléctrica no fuese suficiente o las pérdidas fuesen grandes, el agua no alcanzaría la temperatura máxima de ebullición.

En este proyecto, vamos a analizar los cambios de temperatura del líquido contenido en un recipiente, Figura 4.30, cuando por la resistencia de la resistencia eléctrica circula una corriente alterna que hace que el líquido se caliente de forma periódica. Supondremos que el recipiente pierde calor y lo transfiere a la atmósfera y que estas pérdidas obedecen a la ley de enfriamiento de Newton. El régimen de temperaturas del líquido se puede dividir en dos partes:

- Un estado transitorio que dura un determinado tiempo (teóricamente infinito) en la práctica viene determinado por la denominada constante de tiempo.
- Después, el estado estacionario caracterizado por una oscilación de temperaturas de amplitud constante, alrededor de un valor medio.



**Figura 4.30 Calentamiento de agua**

La potencia  $P$  suministrada por el resistencia eléctrica de resistencia  $R$  por el cual circula una corriente alterna de amplitud  $i_0$  y de frecuencia angular  $\omega$  es

$$P = Ri^2 = Ri_0^2 \cos^2(\omega t) \quad (4.21)$$

El calor  $dQ=P \cdot dt$  suministrado por el calentador eléctrico en el intervalo de tiempo entre  $t$  y  $t+dt$  se invierte en elevar la temperatura del líquido  $mc \cdot dT$  se transfiere a la atmósfera  $\alpha S (T-T_a)$ , de acuerdo con la ley del enfriamiento de Newton[12]. Donde  $\alpha$  es el coeficiente de intercambio de calor y  $S$  es el área

del cuerpo en contacto con la atmósfera.  $T$  es la temperatura del líquido y  $T_a$  es la temperatura ambiente.

$$P = \frac{dQ}{dt} = mc \frac{dT}{dt} + \alpha S (T - T_a) \quad (4.22)$$

$$\frac{dT}{dt} = \frac{Ri_0^2 \cos^2(\omega t)}{mc} - \frac{\alpha S}{mc} (T - T_a) \quad (4.23)$$

$$\frac{dT}{dt} + \frac{\alpha S}{mc} T = \frac{\alpha S}{mc} T_a + \frac{Ri_0^2 \cos^2(\omega t)}{mc} \quad (4.24)$$

$$\frac{dT}{dt} + \frac{\alpha S}{mc} T = \frac{\alpha S}{mc} T_a + \frac{Ri_0^2}{2mc} + \frac{Ri_0^2}{2mc} \cos(2\omega t) \quad (4.25)$$

La solución de esta ecuación diferencial es la suma de la solución particular de la forma

$$T_1 = A + B \cdot \cos(2\omega t) + C \cdot \sin(2\omega t) \quad (4.26)$$

y de la solución de la ecuación diferencial homogénea

Introducimos esta solución en la ecuación diferencial para determinar los coeficientes  $A$ ,  $B$  y  $C$ .

$$A = T_a + \frac{Ri_0^2}{2\alpha S} \quad B = \frac{Ri_0^2}{4\omega^2 m^2 c^2 + \alpha^2 S^2} mc \omega \quad C = \frac{Ri_0^2}{4\omega^2 m^2 c^2 + \alpha^2 S^2} \frac{\alpha S}{2}$$

La solución particular se escribe

$$T_1 = T_a + \frac{Ri_0^2}{2\alpha S} + \frac{Ri_0^2}{4\omega^2 m^2 c^2 + \alpha^2 S^2} \left( mc \omega \sin(2\omega t) + \frac{\alpha S}{2} \cos(2\omega t) \right) \quad (4.27)$$

La solución de la ecuación diferencial homogénea

$$\frac{dT}{dt} + \frac{aS}{mc} T = 0 \quad T_2 = D \cdot \exp\left(-\frac{aS}{mc} t\right) \quad (4.28)$$

La solución completa de la ecuación diferencial es

$$T = T_1 + T_2 = T_a + \frac{R_0^2}{2aS} + \frac{R_0^2}{4\omega^2 m^2 c^2 + a^2 S^2} \left( mc\omega \sin(2\omega t) + \frac{aS}{2} \cos(2\omega t) \right) + D \exp\left(-\frac{aS}{mc} t\right) \quad (4.29)$$

El coeficiente D se determina a partir de las condiciones iniciales. En el instante  $t=0$ , la temperatura del líquido contenido en el recipiente es la temperatura ambiente  $T_a$ , del ambiente.

$$T = T_a + \frac{R_0^2}{2aS} + \frac{R_0^2}{2\sqrt{4\omega^2 m^2 c^2 + a^2 S^2}} \cos(2\omega t - \phi) + D \exp\left(-\frac{aS}{mc} t\right) \quad (4.30)$$

$$D = -\frac{R_0^2}{2aS} - \frac{R_0^2 aS}{2(4\omega^2 m^2 c^2 + a^2 S^2)} \quad \tan \phi = 2\omega \frac{mc}{aS}$$

Definimos los parámetros

$$\tau = \frac{mc}{aS} \quad K = \frac{R_0^2}{mc}$$

El primero  $\tau$  se denomina constante de tiempo. La temperatura T del líquido en función del tiempo t se expresa de forma más simple.



$$T = T_a + \frac{K\tau}{2} + \frac{K\tau}{2\sqrt{1+4\omega^2\tau^2}} \cos(2\omega t - \phi) + D \exp(-t/\tau) \quad (4.31)$$

$$D = -\frac{K\tau}{2} - \frac{K\tau}{2(1+4\omega^2\tau^2)} \quad \tan \phi = 2\omega\tau$$

Después de haber explicado anteriormente el fenómeno termodinámico, es necesario obtener la función de transferencia de la planta en lo que a la temperatura se refiere [11].

Realizamos primeramente un balance de energía para el líquido dentro del tanque del diagrama mostrado en la Figura 4.7 bajo las suposiciones de temperatura de entrada  $T_i$  constante, volumen de líquido  $V$  dentro del tanque constante, despreciables pérdidas de calor al ambiente, e igual flujo de entrada y de salida se llega a:

$$\frac{d(\rho V \mu(t))}{dt} = \rho_i h_i q(t) - \rho h(t) q(t) + w(t) \quad (4.32)$$

$\mu(t)$ : Energía interna del líquido (agua) en el tanque

$h(t)$ : Entalpía del líquido (agua)

$w(t)$ : Calor suministrado por el calentador eléctrico a potencia máxima

$$\mu(t) = C_v T(t) \quad (4.33)$$

$$h(t) = C_p T(t) \quad (4.34)$$

$$w(t) = U_D A_{ce} (T_{ce}(t) - T(t)) \quad (4.35)$$

Si consideramos despreciable la variación de la densidad y de la capacidad calorífica dentro del ámbito limitado de temperatura, y se supone la temperatura del calentador como una constante, tenemos:

$$\frac{\rho V C_v d(T(t))}{dt} = \rho_i C_p T_i q(t) - \rho C_p T(t) q(t) + U_D A_{ce} (T_{ce}(t) - T(t)) \quad (4.36)$$

Si  $t=0$  es el estado estable inicial, entonces tenemos lo siguiente:

$$\frac{\rho V C_v d(T(0))}{dt} = \rho_i C_p T_i q(0) - \rho C_p T(0) q(0) + U_D A_{ce} (T_{ce} - T(0)) \quad (4.37)$$

Si restamos las ecuaciones (4.37) de (4.36) se tiene:

$$\frac{\rho V C_v d(T(t))}{dt} = \rho C_p T_i (q(t) - q(0)) - \rho C_p (T(t) q(t) - q(0) T(0)) + U_D A_{ce} (T_{ce} - T(0)) \quad (4.38)$$

Utilizamos las siguientes variables de desviación:

$$F(t) = q(t) - q(0) \quad (4.39)$$

$$\Gamma(t) = T(t) - T(0) \quad (4.40)$$

Al linealizar la función dependiente del tiempo en el segundo término de la derecha de la ecuación (4.38) se tiene lo siguiente:

$$q(t)T(t) \approx q(0)T(0) + T(0)(q(t) - q(0)) + q(0)(T(t) - T(0)) \quad (4.41)$$

$$q(t)T(t) - q(0)T(0) \approx T(0)F(t) + q(0)\Gamma(t) \quad (4.42)$$

Sustituyendo tenemos:

$$\frac{\rho V C_v d\Gamma(t)}{dt} = \rho C_p T_i F(t) - \rho C_p [T(0)F(t) + q(0)\Gamma(t)] - U_D A_{ce} \Gamma(t) \quad (4.43)$$

$$\frac{\rho V C_v d\Gamma(t)}{dt} = \rho C_p [T_i - T(0)] F(t) - [\rho C_p q(0) - U_D A_{ce}] \Gamma(t) \quad (4.44)$$

$$\frac{\rho V C_v}{\rho C_p q(0) - U_D A_{ce}} \frac{d\Gamma(t)}{dt} = \frac{\rho C_p [T_i - T(0)]}{\rho C_p q(0) - U_D A_{ce}} F(t) - \Gamma(t) \quad (4.45)$$

$$\tau = \frac{\rho V C_v}{\rho C_p q(0) - U_D A_{ce}} \quad (4.46)$$

$$K = \frac{\rho C_p [T_i - T(0)]}{\rho C_p q(0) - U_D A_{ce}} \quad (4.47)$$

$$\tau \frac{d\Gamma(t)}{dt} + \Gamma(t) = KF(t) \quad (4.48)$$

Aplicamos la transformada de Laplace a la ecuación (4.43) y simplificando se llega finalmente al modelo de primer orden:

$$\Gamma(s) = \frac{K}{\tau s + 1} F(s) \quad (4.49)$$

Se puede dar un tiempo muerto en la respuesta se corrige quedando tal como sigue:

$$\Gamma(s) = \frac{Ke^{-t_0s}}{\tau s + 1} F(s) \quad (4.50)$$

#### 4.8 Pruebas del sistema

Las pruebas de medición para la calibración del sistema se realizarán mediante los siguientes pasos:

1. Realizar cinco mediciones de temperatura con el sensor de temperatura de líquidos DS18B20 en varios puntos del tanque y sacar el promedio por cada prueba realizada.
2. Medir altura de vacío con flexómetro desde borde superior del tanque.
3. Calcular manualmente el volumen de agua almacenada.

4. Verificar los valores del software en el display del panel frontal.

En las primeras pruebas se presentaron las siguientes novedades:

- Existía demasiado ruido en las dos señales medidas, sus formas eran similares a dientes de sierra con una variación de dos grados en temperatura y cuatro en altura de agua.
- Se observa en la gráfica de temperatura y nivel medido del software de la pantalla de supervisión que la oscilación tiene una tendencia, es decir que el valor se mantiene constante por mayor tiempo, se desvía momentáneamente pero retorna al valor medido original.
- En las pruebas finales se logró realizar una calibración con un margen de error aceptable, considerando que el proyecto es únicamente un prototipo didactico de prueba.

#### 4.9 Fotos y Gráficos del sistema

A continuación se muestran varias fotografías tomadas durante el experimento de la planta de control de nivel y temperatura, además de otros gráficos del programa en LabVIEW 2010.



**Figura 4.31 Mediciones de voltaje y corriente**



**Figura 4.32 Planta y PC**



**Figura 4.33 Módulos ideas&tecnologías**





**Fig. 4.34** Llenado del tanque

## CONCLUSIONES

1. Hemos desarrollado el FIRMWARE en lenguaje Microbasic Pro, el cual nos sirve para la comunicación entre el Hardware y Software del proyecto desarrollado.
2. Se diseñó en LabView el programa de control para nivel y temperatura. Para este programa se utilizó VISA, que establece la comunicación entre el módulo de entrenamiento, los sensores, actuadores y PC.
3. Las rutinas del protocolo one-wire proporcionadas por el programa Mikroc pro for pic nos permiten convertir los datos proporcionados por el sensor DS18B20 de bits a valores tipo char, para estos poder enviar a las funciones que permiten la visualización de los datos en pantalla.
4. Logramos construir un sistema que permite el control de la temperatura de un líquido gracias a la utilización del sensor inteligente

DS18B20, a través de dispositivos como los microcontroladores y software para mostrar los datos obtenidos.

5. El protocolo de programación ICSP se trata de un sistema con un set de comandos muy reducido y sencillo de utilizar. Se puede acceder indistintamente a cualquiera de los tipos de memoria de los dispositivos y modificar una parte concreta de los mismos sin variar el resto. Esto es muy útil a la hora de depurar programas con gran cantidad de líneas de código, reduciendo el tiempo de programación del dispositivo.
6. El sistema posee un único comando de incremento de posición por lo que si queremos acceder a un byte de memoria situado en la última posición debemos recorrer toda la memoria hasta llegar a esa posición. Mejoraría el sistema un comando de acceso a memoria que permitiese mandar la posición de la memoria a la que se desea acceder.
7. Una desventaja es la utilización de un voltaje de programación distinto al de funcionamiento del dispositivo. Esto provoca el tener que utilizar

un sistema de alimentación algo más complejo para generar dos tensiones de 5 y 14 voltios.

## RECOMENDACIONES

1. Crear un modelo adecuado de comandos para que la comunicación entre el sensor y el microcontrolador 16F886 sea eficiente, esto es respetando el tiempo que el sensor necesita para la captura de datos.
2. Cuando se ingresa los valores de los rangos máximos y mínimo de temperatura por el panel frontal de LabView, se debe procurar que estos valores sean acordes a los parámetros del sensor de temperatura DS18B20 que solo soporta valores de temperatura entre -55 °C y +125 °C para que el sistema tenga un perfecto funcionamiento.
3. Verificar que el microcontrolador trabaje con una frecuencia de al menos 4Mhz, debido que las rutinas de la librería one-wire requieren ese parámetro para la utilización de termómetros digitales.

4. Se recomienda en este proyecto instalar un sistema agitador para calentar el agua en el menor tiempo posible, ya que así se obtiene una temperatura igual en toda la masa de agua.
  
5. Como última recomendación, sería que, en base a los conocimientos de adquisición de datos, identificación de sistemas y procesamiento de datos, realizar y diseñar un control más robusto y preciso como un PID para el control de nivel y calefacción de agua utilizando el software LabVIEW.

## **ANEXOS**

## ANEXO A

### SOFTWARE DE COMUNICACIÓN

#### PROYECTO\_ROY\_interrupt:

```

MOVWF  R15+0
SWAPF  STATUS+0, 0
CLRF   STATUS+0
MOVWF  ___saveSTATUS+0
MOVF   PCLATH+0, 0
MOVWF  ___savePCLATH+0
CLRF   PCLATH+0

```

```

;PROYECTO_ROY.mbas,20 ::      sub procedure interrupt()
;PROYECTO_ROY.mbas,22 ::      if(PIR1.RCIF=1)then
    BTFSS  PIR1+0, 5
    GOTO   L_PROYECTO_ROY_interrupt2
;PROYECTO_ROY.mbas,23 ::      if UART1_Data_Ready = 1 then
    CALL  _UART1_Data_Ready+0
    MOVF  R0+0, 0
    XORLW 1

```



```

    BTFSS    STATUS+0, 2
    GOTO     L_PROYECTO_ROY_interrupt5

;PROYECTO_ROY.mbas,24 ::      receive = Uart1_Read

    CALL    _UART1_Read+0
    MOVF    R0+0, 0
    MOVWF   _receive+0

;PROYECTO_ROY.mbas,26 ::      CASE 0

    MOVF    _INDICE+0, 0
    XORLW   0
    BTFSS   STATUS+0, 2
    GOTO    L_PROYECTO_ROY_interrupt10

;PROYECTO_ROY.mbas,27 ::      if(receive="I")then

    MOVF    _receive+0, 0
    XORLW   73
    BTFSS   STATUS+0, 2
    GOTO    L_PROYECTO_ROY_interrupt12

;PROYECTO_ROY.mbas,28 ::      INDICE=1

    MOVLW   1
    MOVWF   _INDICE+0
    GOTO    L_PROYECTO_ROY_interrupt13

;PROYECTO_ROY.mbas,29 ::      else

L_PROYECTO_ROY_interrupt12:

```

```

;PROYECTO_ROY.mbas,30 ::      INDICE=0
        CLRF    _INDICE+0

;PROYECTO_ROY.mbas,31 ::      end if

L_PROYECTO_ROY_interrupt13:
        GOTO    L_PROYECTO_ROY_interrupt7

L_PROYECTO_ROY_interrupt10:

;PROYECTO_ROY.mbas,32 ::      CASE 1
        MOVF    _INDICE+0, 0
        XORLW   1
        BTFSS   STATUS+0, 2
        GOTO    L_PROYECTO_ROY_interrupt16

;PROYECTO_ROY.mbas,33 ::      INTENSIDAD=receive
        MOVF    _receive+0, 0
        MOVWF   _INTENSIDAD+0

;PROYECTO_ROY.mbas,34 ::      INDICE=2
        MOVLW   2
        MOVWF   _INDICE+0
        GOTO    L_PROYECTO_ROY_interrupt7

L_PROYECTO_ROY_interrupt16:

;PROYECTO_ROY.mbas,35 ::      case 2
        MOVF    _INDICE+0, 0
        XORLW   2

```

```

    BTFSS    STATUS+0, 2
    GOTO     L_PROYECTO_ROY_interrupt19
;PROYECTO_ROY.mbas,36 ::      if(receive="B")then
    MOVF     _receive+0, 0
    XORLW    66
    BTFSS    STATUS+0, 2
    GOTO     L_PROYECTO_ROY_interrupt21
;PROYECTO_ROY.mbas,37 ::      INDICE=3
    MOVLW    3
    MOVWF    _INDICE+0
    GOTO     L_PROYECTO_ROY_interrupt22
;PROYECTO_ROY.mbas,38 ::      else
L_PROYECTO_ROY_interrupt21:
;PROYECTO_ROY.mbas,39 ::      INDICE=0
    CLRF     _INDICE+0
;PROYECTO_ROY.mbas,40 ::      end if
L_PROYECTO_ROY_interrupt22:
    GOTO     L_PROYECTO_ROY_interrupt7
L_PROYECTO_ROY_interrupt19:
;PROYECTO_ROY.mbas,41 ::      CASE 3
    MOVF     _INDICE+0, 0
    XORLW    3

```

```

    BTFSS    STATUS+0, 2
    GOTO     L_PROYECTO_ROY_interrupt25
;PROYECTO_ROY.mbas,42 ::      duty=receive
    MOVF     _receive+0, 0
    MOVWF    _duty+0
;PROYECTO_ROY.mbas,43 ::      INDICE=0
    CLRF     _INDICE+0
    GOTO     L_PROYECTO_ROY_interrupt7
L_PROYECTO_ROY_interrupt25:
L_PROYECTO_ROY_interrupt7:
;PROYECTO_ROY.mbas,44 ::      end select
L_PROYECTO_ROY_interrupt5:
;PROYECTO_ROY.mbas,46 ::      PIR1.RCIF=0 ' Si el dato a llegado
limpio la bandera de recepcion
    BCF      PIR1+0, 5
;PROYECTO_ROY.mbas,47 ::      PIE1.RCIE=1      ' Habilitar
nuevamente la interrupcion por USART
    BSF      PIE1+0, 5
L_PROYECTO_ROY_interrupt2:
;PROYECTO_ROY.mbas,50 ::      if(INTCON.INTF=1)then
    BTFSS    INTCON+0, 1
    GOTO     L_PROYECTO_ROY_interrupt27

```

```

;PROYECTO_ROY.mbas,52 ::      if(intensidad > 0)then
    MOVF    _INTENSIDAD+0, 0
    SUBLW   0
    BTFSC   STATUS+0, 0
    GOTO    L_PROYECTO_ROY_interrupt30
;PROYECTO_ROY.mbas,53 ::      PORTB.4=1
    BSF     PORTB+0, 4
;PROYECTO_ROY.mbas,55 ::      case 1
    MOVF    _INTENSIDAD+0, 0
    XORLW   1
    BTFSS   STATUS+0, 2
    GOTO    L_PROYECTO_ROY_interrupt35
;PROYECTO_ROY.mbas,56 ::      TMR0=240
    MOVLW   240
    MOVWF   TMR0+0
    GOTO    L_PROYECTO_ROY_interrupt32
L_PROYECTO_ROY_interrupt35:
;PROYECTO_ROY.mbas,57 ::      case 2
    MOVF    _INTENSIDAD+0, 0
    XORLW   2
    BTFSS   STATUS+0, 2
    GOTO    L_PROYECTO_ROY_interrupt38

```

```
;PROYECTO_ROY.mbas,58 ::      TMR0=225  
  
    MOVLW    225  
  
    MOVWF    TMR0+0  
  
    GOTO     L_PROYECTO_ROY_interrupt32
```

```
L_PROYECTO_ROY_interrupt38:
```

```
;PROYECTO_ROY.mbas,59 ::      case 3  
  
    MOVF     _INTENSIDAD+0, 0  
  
    XORLW    3  
  
    BTFSS    STATUS+0, 2  
  
    GOTO     L_PROYECTO_ROY_interrupt41
```

```
;PROYECTO_ROY.mbas,60 ::      TMR0=209  
  
    MOVLW    209  
  
    MOVWF    TMR0+0  
  
    GOTO     L_PROYECTO_ROY_interrupt32
```

```
L_PROYECTO_ROY_interrupt41:
```

```
;PROYECTO_ROY.mbas,61 ::      case 4  
  
    MOVF     _INTENSIDAD+0, 0  
  
    XORLW    4  
  
    BTFSS    STATUS+0, 2  
  
    GOTO     L_PROYECTO_ROY_interrupt44
```

```
;PROYECTO_ROY.mbas,62 ::      TMR0=193  
  
    MOVLW    193
```

```
MOVWF  TMR0+0
GOTO   L_PROYECTO_ROY_interrupt32
L_PROYECTO_ROY_interrupt44:
;PROYECTO_ROY.mbas,63 ::      case 5
MOVF   _INTENSIDAD+0, 0
XORLW  5
BTFSS  STATUS+0, 2
GOTO   L_PROYECTO_ROY_interrupt47
;PROYECTO_ROY.mbas,64 ::      TMR0=178
MOVLW  178
MOVWF  TMR0+0
GOTO   L_PROYECTO_ROY_interrupt32
L_PROYECTO_ROY_interrupt47:
;PROYECTO_ROY.mbas,65 ::      case 6
MOVF   _INTENSIDAD+0, 0
XORLW  6
BTFSS  STATUS+0, 2
GOTO   L_PROYECTO_ROY_interrupt50
;PROYECTO_ROY.mbas,66 ::      TMR0=162
MOVLW  162
MOVWF  TMR0+0
GOTO   L_PROYECTO_ROY_interrupt32
```

```
L_PROYECTO_ROY_interrupt50:
;PROYECTO_ROY.mbas,67 ::      case 7
    MOVF    _INTENSIDAD+0, 0
    XORLW   7
    BTFSS   STATUS+0, 2
    GOTO    L_PROYECTO_ROY_interrupt53
;PROYECTO_ROY.mbas,68 ::      TMR0=146
    MOVLW   146
    MOVWF   TMR0+0
    GOTO    L_PROYECTO_ROY_interrupt32
L_PROYECTO_ROY_interrupt53:
;PROYECTO_ROY.mbas,69 ::      case 8
    MOVF    _INTENSIDAD+0, 0
    XORLW   8
    BTFSS   STATUS+0, 2
    GOTO    L_PROYECTO_ROY_interrupt56
;PROYECTO_ROY.mbas,70 ::      TMR0=131
    MOVLW   131
    MOVWF   TMR0+0
    GOTO    L_PROYECTO_ROY_interrupt32
L_PROYECTO_ROY_interrupt56:
L_PROYECTO_ROY_interrupt32:
```



```

;PROYECTO_ROY.mbas,72 ::      INTCON.T0IF=0
      BCF    INTCON+0, 2

;PROYECTO_ROY.mbas,73 ::      INTCON.T0IE=1
      BSF    INTCON+0, 5
      GOTO   L_PROYECTO_ROY_interrupt31

;PROYECTO_ROY.mbas,74 ::      else

L_PROYECTO_ROY_interrupt30:

;PROYECTO_ROY.mbas,75 ::      PORTB.4=0
      BCF    PORTB+0, 4

;PROYECTO_ROY.mbas,76 ::      end if

L_PROYECTO_ROY_interrupt31:

;PROYECTO_ROY.mbas,77 ::      INTCON.INTF=0
      BCF    INTCON+0, 1

;PROYECTO_ROY.mbas,78 ::      INTCON.INTE =1
      BSF    INTCON+0, 4

L_PROYECTO_ROY_interrupt27:

;PROYECTO_ROY.mbas,81 ::      if(INTCON.T0IF =1)then
      BTFSS  INTCON+0, 2
      GOTO   L_PROYECTO_ROY_interrupt58

;PROYECTO_ROY.mbas,82 ::      PORTB.4=0
      BCF    PORTB+0, 4

;PROYECTO_ROY.mbas,83 ::      INTCON.T0IF=0

```

```

        BCF    INTCON+0, 2
;PROYECTO_ROY.mbas,84 ::      INTCON.T0IE=0

        BCF    INTCON+0, 5
L_PROYECTO_ROY_interrupt58:
;PROYECTO_ROY.mbas,85 ::      end if

L_PROYECTO_ROY_interrupt108:

        MOVF   ___savePCLATH+0, 0
        MOVWF  PCLATH+0
        SWAPF  ___saveSTATUS+0, 0
        MOVWF  STATUS+0
        SWAPF  R15+0, 1
        SWAPF  R15+0, 0

        RETFIE

; end of PROYECTO_ROY_interrupt

PROYECTO_ROY_PULSO:

;PROYECTO_ROY.mbas,90 ::      SUB    PROCEDURE    PULSO()
'Procedimiento o subruina para generar pulso trigger para
;PROYECTO_ROY.mbas,92 ::      PORTC.0=1

        BSF    PORTC+0, 0
;PROYECTO_ROY.mbas,93 ::      Delay_us(30)

```

```

    MOVLW    9
    MOVWF    R13+0

L_PROYECTO_ROY_PULSO61:
    DECFSZ   R13+0, 1
    GOTO     L_PROYECTO_ROY_PULSO61
    NOP
    NOP

;PROYECTO_ROY.mbas,94 ::      PORTC.0=0
    BCF     PORTC+0, 0
    RETURN

; end of PROYECTO_ROY_PULSO

PROYECTO_ROY_set_motors:

;PROYECTO_ROY.mbas,97 ::      sub  procedure  set_motors(dim
motor2, motor1 as integer)

;PROYECTO_ROY.mbas,99 ::      if motor1 < 0 then
    MOVLW   128
    XORWF   FARG_set_motors_motor1+1, 0
    MOVWF   R0+0
    MOVLW   128
    SUBWF   R0+0, 0

```

```

    BTFSS    STATUS+0, 2
    GOTO     L_PROYECTO_ROY_set_motors109
    MOVLW    0
    SUBWF    FARG_set_motors_motor1+0, 0
L_PROYECTO_ROY_set_motors109:
    BTFSC    STATUS+0, 0
    GOTO     L_PROYECTO_ROY_set_motors64
;PROYECTO_ROY.mbas,100 ::      motor1= -motor1
    MOVF     FARG_set_motors_motor1+0, 0
    SUBLW    0
    MOVWF    FARG_set_motors_motor1+0
    MOVF     FARG_set_motors_motor1+1, 0
    BTFSS    STATUS+0, 0
    ADDLW    1
    CLRF     FARG_set_motors_motor1+1
    SUBWF    FARG_set_motors_motor1+1, 1
;PROYECTO_ROY.mbas,101 ::      direccion2a = 0
    BCF      PORTA+0, 2
;PROYECTO_ROY.mbas,102 ::      direccion2b = 1
    BSF      PORTA+0, 4
    GOTO     L_PROYECTO_ROY_set_motors65
;PROYECTO_ROY.mbas,103 ::      else

```

**L\_PROYECTO\_ROY\_set\_motors64:**

**;PROYECTO\_ROY.mbas,104 ::        direccion2a = 1**

**BSF    PORTA+0, 2**

**;PROYECTO\_ROY.mbas,105 ::        direccion2b = 0**

**BCF    PORTA+0, 4**

**;PROYECTO\_ROY.mbas,106 ::        end if**

**L\_PROYECTO\_ROY\_set\_motors65:**

**;PROYECTO\_ROY.mbas,108 ::        if motor2 < 0 then**

**MOVLW    128**

**XORWF    FARG\_set\_motors\_motor2+1, 0**

**MOVWF    R0+0**

**MOVLW    128**

**SUBWF    R0+0, 0**

**BTFSS    STATUS+0, 2**

**GOTO     L\_PROYECTO\_ROY\_set\_motors110**

**MOVLW    0**

**SUBWF    FARG\_set\_motors\_motor2+0, 0**

**L\_PROYECTO\_ROY\_set\_motors110:**

**BTFSC    STATUS+0, 0**

**GOTO     L\_PROYECTO\_ROY\_set\_motors67**

**;PROYECTO\_ROY.mbas,109 ::        motor2= -motor2**

**MOVF     FARG\_set\_motors\_motor2+0, 0**

```

    SUBLW    0

    MOVWF   FARG_set_motors_motor2+0

    MOVF    FARG_set_motors_motor2+1, 0

    BTFSS   STATUS+0, 0

    ADDLW   1

    CLRF    FARG_set_motors_motor2+1

    SUBWF   FARG_set_motors_motor2+1, 1

;PROYECTO_ROY.mbas,110 ::      direccion1a = 1

    BSF     PORTA+0, 3

;PROYECTO_ROY.mbas,111 ::      direccion1b = 0

    BCF     PORTA+0, 5

    GOTO    L_PROYECTO_ROY_set_motors68

;PROYECTO_ROY.mbas,112 ::      else

L_PROYECTO_ROY_set_motors67:

;PROYECTO_ROY.mbas,113 ::      direccion1a = 0

    BCF     PORTA+0, 3

;PROYECTO_ROY.mbas,114 ::      direccion1b = 1

    BSF     PORTA+0, 5

;PROYECTO_ROY.mbas,115 ::      end if

L_PROYECTO_ROY_set_motors68:

;PROYECTO_ROY.mbas,117 ::      if motor1 > 0xFF then

    MOVLW   128

```

```

MOVWF    R0+0
MOVLW    128
XORWF    FARG_set_motors_motor1+1, 0
SUBWF    R0+0, 0
BTFSS    STATUS+0, 2
GOTO     L_PROYECTO_ROY_set_motors111
MOVF     FARG_set_motors_motor1+0, 0
SUBLW    255

```

L\_PROYECTO\_ROY\_set\_motors111:

```

BTFSC    STATUS+0, 0
GOTO     L_PROYECTO_ROY_set_motors70

```

;PROYECTO\_ROY.mbas,118 :: motor1 = 0xFF

```

MOVLW    255
MOVWF    FARG_set_motors_motor1+0
CLRF     FARG_set_motors_motor1+1

```

L\_PROYECTO\_ROY\_set\_motors70:

;PROYECTO\_ROY.mbas,121 :: if motor2 > 0xFF then

```

MOVLW    128
MOVWF    R0+0
MOVLW    128
XORWF    FARG_set_motors_motor2+1, 0
SUBWF    R0+0, 0

```

```

    BTFSS    STATUS+0, 2
    GOTO     L_PROYECTO_ROY_set_motors112
    MOVF     FARG_set_motors_motor2+0, 0
    SUBLW    255

L_PROYECTO_ROY_set_motors112:
    BTFSC    STATUS+0, 0
    GOTO     L_PROYECTO_ROY_set_motors73
;PROYECTO_ROY.mbas,122 ::      motor2 = 0xFF
    MOVLW    255
    MOVWF    FARG_set_motors_motor2+0
    CLRF     FARG_set_motors_motor2+1

L_PROYECTO_ROY_set_motors73:
;PROYECTO_ROY.mbas,125 ::      PWM2_Set_Duty(motor2)
    MOVF     FARG_set_motors_motor2+0, 0
    MOVWF    FARG_PWM2_Set_Duty_new_duty+0
    CALL     _PWM2_Set_Duty+0
    RETURN

; end of PROYECTO_ROY_set_motors

_main:

;PROYECTO_ROY.mbas,129 ::      main:

```



```

;PROYECTO_ROY.mbas,130 ::      OPTION_REG=%11000101
'CONFIGURACION PARA EL TEMPORIZADOR TMR0 FOSC=4MHz
PRESCALADOR 128
TIEMPO=(4/FOSC)*PRESCALADOR*(256-TMR0)
    MOVLW    197
    MOVWF   OPTION_REG+0
;PROYECTO_ROY.mbas,131 ::      T1CON=%00000001
    MOVLW    1
    MOVWF   T1CON+0
;PROYECTO_ROY.mbas,132 ::      INTCON=0
    CLRF    INTCON+0
;PROYECTO_ROY.mbas,133 ::      ANSEL=0 'Todos los pines ANX
como tipo digital
    CLRF    ANSEL+0
;PROYECTO_ROY.mbas,134 ::      ANSELH=0 'Todos los pines ANX
como tipo digital
    CLRF    ANSELH+0
;PROYECTO_ROY.mbas,135 ::      TRISA=%00000000
    CLRF    TRISA+0
;PROYECTO_ROY.mbas,136 ::      TRISB=%00000111 ' 1-> ENTRADA
0 -> SALIDA
    MOVLW    7

```

```
MOVWF TRISB+0
;PROYECTO_ROY.mbas,137 :: TRISC=%10001000
MOVLW 136
MOVWF TRISC+0
;PROYECTO_ROY.mbas,138 :: PORTA=0
CLRF PORTA+0
;PROYECTO_ROY.mbas,139 :: PORTB=0
CLRF PORTB+0
;PROYECTO_ROY.mbas,140 :: PORTC=0
CLRF PORTC+0
;PROYECTO_ROY.mbas,141 :: FLANCO=0
CLRF _FLANCO+0
;PROYECTO_ROY.mbas,142 :: DUTY=0
CLRF _duty+0
;PROYECTO_ROY.mbas,143 :: Distancia=0
CLRF _Distancia+0
CLRF _Distancia+1
;PROYECTO_ROY.mbas,144 :: Distancia_temp=0
CLRF _Distancia_temp+0
CLRF _Distancia_temp+1
;PROYECTO_ROY.mbas,145 :: muestras=0
CLRF _muestras+0
```

```
;PROYECTO_ROY.mbas,146 :: tiempo =0  
    CLRFB _tiempo+0  
    CLRFB _tiempo+1  
;PROYECTO_ROY.mbas,147 :: INDICE=0  
    CLRFB _INDICE+0  
;PROYECTO_ROY.mbas,148 :: TEMP_LOW=0  
    CLRFB _TEMP_LOW+0  
    CLRFB _TEMP_LOW+1  
;PROYECTO_ROY.mbas,149 :: TEMP_HIGH =0  
    CLRFB _TEMP_HIGH+0  
    CLRFB _TEMP_HIGH+1  
;PROYECTO_ROY.mbas,150 :: TEMP_TOTAL =0  
    CLRFB _TEMP_TOTAL+0  
    CLRFB _TEMP_TOTAL+1  
;PROYECTO_ROY.mbas,151 :: TEMP_ENTERA=0  
    CLRFB _TEMP_ENTERA+0  
    CLRFB _TEMP_ENTERA+1  
;PROYECTO_ROY.mbas,152 :: TEMP_ENTERA_TEMP=0  
    CLRFB _TEMP_ENTERA_TEMP+0  
    CLRFB _TEMP_ENTERA_TEMP+1  
;PROYECTO_ROY.mbas,153 :: TEMP_DECIMAL=0  
    CLRFB _TEMP_DECIMAL+0
```

```
        CLRF    _TEMP_DECIMAL+1
;PROYECTO_ROY.mbas,154 ::      MENOS=0
        CLRF    _MENOS+0
;PROYECTO_ROY.mbas,155 ::      INTENSIDAD=0
        CLRF    _INTENSIDAD+0
;PROYECTO_ROY.mbas,156 ::      UART1_Init(9600)
        MOVLW   25
        MOVWF   SPBRG+0
        BSF     TXSTA+0, 2
        CALL    _UART1_Init+0
;PROYECTO_ROY.mbas,157 ::      Delay_ms(100)
        MOVLW   130
        MOVWF   R12+0
        MOVLW   221
        MOVWF   R13+0
L__main76:
        DECFSZ  R13+0, 1
        GOTO    L__main76
        DECFSZ  R12+0, 1
        GOTO    L__main76
        NOP
        NOP
```

```

;PROYECTO_ROY.mbas,158 ::      INTCON =%11010000
    MOVLW    208
    MOVWF    INTCON+0
;PROYECTO_ROY.mbas,159 ::      PIE1 = %00100000
    MOVLW    32
    MOVWF    PIE1+0
;PROYECTO_ROY.mbas,160 ::      PIR1=0
    CLRF     PIR1+0
;PROYECTO_ROY.mbas,161 ::      Delay_ms(100)
    MOVLW    130
    MOVWF    R12+0
    MOVLW    221
    MOVWF    R13+0
L__main77:
    DECFSZ   R13+0, 1
    GOTO     L__main77
    DECFSZ   R12+0, 1
    GOTO     L__main77
    NOP
    NOP
;PROYECTO_ROY.mbas,164 ::      PWM2_Init (1000)
    BCF      T2CON+0, 0

```

```
BCF    T2CON+0, 1
BSF    T2CON+0, 0
BCF    T2CON+0, 1
MOVLW  249
MOVWF  PR2+0
CALL   _PWM2_Init+0
;PROYECTO_ROY.mbas,165 ::      PWM2_Start ()
CALL   _PWM2_Start+0
;PROYECTO_ROY.mbas,166 ::      PWM2_Set_Duty (0)
CLRF   FARG_PWM2_Set_Duty_new_duty+0
CALL   _PWM2_Set_Duty+0
;PROYECTO_ROY.mbas,167 ::      delay_ms(100)
MOVLW  130
MOVWF  R12+0
MOVLW  221
MOVWF  R13+0
L__main78:
DECFSZ R13+0, 1
GOTO   L__main78
DECFSZ R12+0, 1
GOTO   L__main78
NOP
```

```

      NOP

;PROYECTO_ROY.mbas,168 ::      WHILE(1)

L__main80:

;PROYECTO_ROY.mbas,169 ::      PULSO()      'GENERAR      PULSO

TRIGGER

      CALL      PROYECTO_ROY_PULSO+0

;PROYECTO_ROY.mbas,172 ::      DO

L__main84:

;PROYECTO_ROY.mbas,173 ::      IF((PORTC.3=1)      AND

(FLANCO=0))THEN

      BTFSC      PORTC+0, 3

      GOTO      L__main113

      BCF      115, 0

      GOTO      L__main114

L__main113:

      BSF      115, 0

L__main114:

      MOVF      _FLANCO+0, 0

      XORLW      0

      MOVLW      255

      BTFSS      STATUS+0, 2

      MOVLW      0

```

```
MOVWF  R1+0
CLRF   R0+0
BTFSC  115, 0
INCF   R0+0, 1
MOVF   R1+0, 0
ANDWF  R0+0, 1
BTFSC  STATUS+0, 2
GOTO   L__main90
;PROYECTO_ROY.mbas,174 ::      TMR1L=0
CLRF   TMR1L+0
;PROYECTO_ROY.mbas,175 ::      TMR1H=0
CLRF   TMR1H+0
;PROYECTO_ROY.mbas,176 ::      FLANCO=1
MOVLW  1
MOVWF  _FLANCO+0
L__main90:
;PROYECTO_ROY.mbas,178 ::      LOOP UNTIL (FLANCO=1)
MOVF   _FLANCO+0, 0
XORLW  1
BTFSC  STATUS+0, 2
GOTO   L__main87
GOTO   L__main84
```



L\_\_main87:

;PROYECTO\_ROY.mbas,179 :: DO

L\_\_main92:

;PROYECTO\_ROY.mbas,180 :: IF((PORTC.3=0) AND

(FLANCO=1))THEN

BTFSC PORTC+0, 3

GOTO L\_\_main115

BSF 115, 0

GOTO L\_\_main116

L\_\_main115:

BCF 115, 0

L\_\_main116:

MOVF \_FLANCO+0, 0

XORLW 1

MOVLW 255

BTFSS STATUS+0, 2

MOVLW 0

MOVWF R1+0

CLRF R0+0

BTFSC 115, 0

INCF R0+0, 1

MOVF R1+0, 0

```

    ANDWF    R0+0, 1
    BTFSC   STATUS+0, 2
    GOTO    L__main98
;PROYECTO_ROY.mbas,181 ::      tiempo=TMR1H
    MOVF    TMR1H+0, 0
    MOVWF   _tiempo+0
    CLRF    _tiempo+1
;PROYECTO_ROY.mbas,182 ::      tiempo=tiempo<<8
    MOVF    _tiempo+0, 0
    MOVWF   _tiempo+1
    CLRF    _tiempo+0
;PROYECTO_ROY.mbas,183 ::      tiempo=((tiempo) or (TMR1L))
    MOVF    TMR1L+0, 0
    IORWF   _tiempo+0, 1
    MOVLW   0
    IORWF   _tiempo+1, 1
;PROYECTO_ROY.mbas,184 ::      FLANCO=0
    CLRF    _FLANCO+0
L__main98:
;PROYECTO_ROY.mbas,186 ::      LOOP UNTIL (FLANCO=0)
    MOVF    _FLANCO+0, 0
    XORLW   0

```

**BTFSC STATUS+0, 2**

**GOTO L\_\_main95**

**GOTO L\_\_main92**

**L\_\_main95:**

**;PROYECTO\_ROY.mbas,188 :: Distancia\_temp=(tiempo)/58**

**'ECUACION PARA CALCULAR DISTANCIA**

**MOVLW 58**

**MOVWF R4+0**

**CLRF R4+1**

**MOVF \_tiempo+0, 0**

**MOVWF R0+0**

**MOVF \_tiempo+1, 0**

**MOVWF R0+1**

**CALL \_Div\_16x16\_U+0**

**MOVF R0+0, 0**

**MOVWF \_Distancia\_temp+0**

**MOVF R0+1, 0**

**MOVWF \_Distancia\_temp+1**

**;PROYECTO\_ROY.mbas,189 :: Ow\_Reset(PORTA,1)**

**MOVLW PORTA+0**

**MOVWF FARG\_Ow\_Reset\_port+0**

**MOVLW 1**

```
MOVWF FARG_Ow_Reset_pin+0
CALL _Ow_Reset+0
;PROYECTO_ROY.mbas,190 :: Ow_Write (PORTA,1,$CC)
MOVLW PORTA+0
MOVWF FARG_Ow_Write_port+0
MOVLW 1
MOVWF FARG_Ow_Write_pin+0
MOVLW 204
MOVWF FARG_Ow_Write_data_+0
CALL _Ow_Write+0
;PROYECTO_ROY.mbas,191 :: Ow_Write (PORTA,1,$44)
MOVLW PORTA+0
MOVWF FARG_Ow_Write_port+0
MOVLW 1
MOVWF FARG_Ow_Write_pin+0
MOVLW 68
MOVWF FARG_Ow_Write_data_+0
CALL _Ow_Write+0
;PROYECTO_ROY.mbas,192 :: Delay_us(120)
MOVLW 39
MOVWF R13+0
L__main100:
```

```
DECFSZ  R13+0, 1
GOTO    L__main100
NOP
NOP
;PROYECTO_ROY.mbas,193 ::      Ow_Reset (PORTA,1)
MOVLW   PORTA+0
MOVWF   FARG_Ow_Reset_port+0
MOVLW   1
MOVWF   FARG_Ow_Reset_pin+0
CALL    _Ow_Reset+0
;PROYECTO_ROY.mbas,194 ::      Ow_Write(PORTA,1,$CC)
MOVLW   PORTA+0
MOVWF   FARG_Ow_Write_port+0
MOVLW   1
MOVWF   FARG_Ow_Write_pin+0
MOVLW   204
MOVWF   FARG_Ow_Write_data_+0
CALL    _Ow_Write+0
;PROYECTO_ROY.mbas,195 ::      Ow_Write (PORTA,1,$BE)
MOVLW   PORTA+0
MOVWF   FARG_Ow_Write_port+0
MOVLW   1
```

```
MOVWF  FARG_Ow_Write_pin+0
MOVLW  190
MOVWF  FARG_Ow_Write_data_+0
CALL   _Ow_Write+0
;PROYECTO_ROY.mbas,197 ::      TEMP_LOW=Ow_Read(PORTA,1)
MOVLW  PORTA+0
MOVWF  FARG_Ow_Read_port+0
MOVLW  1
MOVWF  FARG_Ow_Read_pin+0
CALL   _Ow_Read+0
MOVF   R0+0, 0
MOVWF  _TEMP_LOW+0
CLRF   _TEMP_LOW+1
;PROYECTO_ROY.mbas,198 ::      TEMP_HIGH=Ow_Read(PORTA,1)
MOVLW  PORTA+0
MOVWF  FARG_Ow_Read_port+0
MOVLW  1
MOVWF  FARG_Ow_Read_pin+0
CALL   _Ow_Read+0
MOVF   R0+0, 0
MOVWF  _TEMP_HIGH+0
CLRF   _TEMP_HIGH+1
```

**;PROYECTO\_ROY.mbas,199 ::           MENOS= TEMP\_HIGH >> 3**

**MOVF    \_TEMP\_HIGH+0, 0**

**MOVWF   R1+0**

**MOVF    \_TEMP\_HIGH+1, 0**

**MOVWF   R1+1**

**RRF     R1+1, 1**

**RRF     R1+0, 1**

**BCF     R1+1, 7**

**RRF     R1+1, 1**

**RRF     R1+0, 1**

**BCF     R1+1, 7**

**RRF     R1+1, 1**

**RRF     R1+0, 1**

**BCF     R1+1, 7**

**MOVF    R1+0, 0**

**MOVWF   \_MENOS+0**

**;PROYECTO\_ROY.mbas,200 ::           TEMP\_TOTAL= (TEMP\_HIGH << 8)**

**OR TEMP\_LOW**

**MOVF    \_TEMP\_HIGH+0, 0**

**MOVWF   \_TEMP\_TOTAL+1**

**CLRF    \_TEMP\_TOTAL+0**

**MOVF    \_TEMP\_LOW+0, 0**

```

    IORWF  _TEMP_TOTAL+0, 1
    MOVF   _TEMP_LOW+1, 0
    IORWF  _TEMP_TOTAL+1, 1
;PROYECTO_ROY.mbas,201 ::      IF(MENOS=%000011111)THEN
    MOVF   R1+0, 0
    XORLW  31
    BTFSS  STATUS+0, 2
    GOTO   L__main102
;PROYECTO_ROY.mbas,202 ::      TEMP_TOTAL= NOT TEMP_TOTAL
    COMF   _TEMP_TOTAL+0, 1
    COMF   _TEMP_TOTAL+1, 1
;PROYECTO_ROY.mbas,203 ::      TEMP_TOTAL =TEMP_TOTAL + 1
    INCF   _TEMP_TOTAL+0, 1
    BTFSC  STATUS+0, 2
    INCF   _TEMP_TOTAL+1, 1
L__main102:
;PROYECTO_ROY.mbas,206 ::      TEMP_ENTERA_TEMP          =
(TEMP_TOTAL AND $0FF0) >> 4
    MOVLW  240
    ANDWF  _TEMP_TOTAL+0, 0
    MOVWF  _TEMP_ENTERA_TEMP+0
    MOVF   _TEMP_TOTAL+1, 0

```



```

ANDLW    15
MOVWF    _TEMP_ENTERA_TEMP+1
RRF      _TEMP_ENTERA_TEMP+1, 1
RRF      _TEMP_ENTERA_TEMP+0, 1
BCF      _TEMP_ENTERA_TEMP+1, 7
RRF      _TEMP_ENTERA_TEMP+1, 1
RRF      _TEMP_ENTERA_TEMP+0, 1
BCF      _TEMP_ENTERA_TEMP+1, 7
RRF      _TEMP_ENTERA_TEMP+1, 1
RRF      _TEMP_ENTERA_TEMP+0, 1
BCF      _TEMP_ENTERA_TEMP+1, 7
RRF      _TEMP_ENTERA_TEMP+1, 1
RRF      _TEMP_ENTERA_TEMP+0, 1
BCF      _TEMP_ENTERA_TEMP+1, 7

```

```

;PROYECTO_ROY.mbas,207 ::      TEMP_DECIMAL = (TEMP_LOW
AND $000F)*625

```

```

MOVLW    15
ANDWF    _TEMP_LOW+0, 0
MOVWF    R0+0
MOVF     _TEMP_LOW+1, 0
MOVWF    R0+1
MOVLW    0

```

```
ANDWF    R0+1, 1
MOVLW    113
MOVWF    R4+0
MOVLW    2
MOVWF    R4+1
CALL     _Mul_16x16_U+0
MOVF     R0+0, 0
MOVWF    _TEMP_DECIMAL+0
MOVF     R0+1, 0
MOVWF    _TEMP_DECIMAL+1
;PROYECTO_ROY.mbas,208 ::      set_motors(DUTY,0)
MOVF     _duty+0, 0
MOVWF    FARG_set_motors_motor2+0
CLRF     FARG_set_motors_motor2+1
CLRF     FARG_set_motors_motor1+0
CLRF     FARG_set_motors_motor1+1
CALL     PROYECTO_ROY_set_motors+0
;PROYECTO_ROY.mbas,209 ::      Delay_ms(10)
MOVLW    13
MOVWF    R12+0
MOVLW    251
MOVWF    R13+0
```

**L\_\_main104:**

**DECFSZ R13+0, 1**

**GOTO L\_\_main104**

**DECFSZ R12+0, 1**

**GOTO L\_\_main104**

**NOP**

**NOP**

**;PROYECTO\_ROY.mbas,210 :: inc(muestras)**

**INCF \_muestras+0, 1**

**;PROYECTO\_ROY.mbas,211 :: Distancia= Distancia +**

**Distancia\_temp**

**MOVF \_Distancia\_temp+0, 0**

**ADDWF \_Distancia+0, 1**

**MOVF \_Distancia\_temp+1, 0**

**BTFSC STATUS+0, 0**

**ADDLW 1**

**ADDWF \_Distancia+1, 1**

**;PROYECTO\_ROY.mbas,212 :: TEMP\_ENTERA = TEMP\_ENTERA**

**+ TEMP\_ENTERA\_TEMP**

**MOVF \_TEMP\_ENTERA\_TEMP+0, 0**

**ADDWF \_TEMP\_ENTERA+0, 1**

**MOVF \_TEMP\_ENTERA\_TEMP+1, 0**

```
BTFSC STATUS+0, 0
ADDLW 1
ADDWF _TEMP_ENTERA+1, 1
;PROYECTO_ROY.mbas,213 :: if(muestras>=20)then
MOVLW 20
SUBWF _muestras+0, 0
BTFSS STATUS+0, 0
GOTO L__main106
;PROYECTO_ROY.mbas,215 :: Distancia= Distancia/20
MOVLW 20
MOVWF R4+0
CLRF R4+1
MOVF _Distancia+0, 0
MOVWF R0+0
MOVF _Distancia+1, 0
MOVWF R0+1
CALL _Div_16x16_U+0
MOVF R0+0, 0
MOVWF _Distancia+0
MOVF R0+1, 0
MOVWF _Distancia+1
```

**;PROYECTO\_ROY.mbas,216 ::**

**TEMP\_ENTERA=TEMP\_ENTERA/20**

**MOVLW 20**

**MOVWF R4+0**

**CLRF R4+1**

**MOVF \_TEMP\_ENTERA+0, 0**

**MOVWF R0+0**

**MOVF \_TEMP\_ENTERA+1, 0**

**MOVWF R0+1**

**CALL \_Div\_16x16\_U+0**

**MOVF R0+0, 0**

**MOVWF \_TEMP\_ENTERA+0**

**MOVF R0+1, 0**

**MOVWF \_TEMP\_ENTERA+1**

**;PROYECTO\_ROY.mbas,217 ::**

**WordToStr(TEMP\_ENTERA,TXT\_ENTERA)**

**MOVF R0+0, 0**

**MOVWF FARG\_WordToStr\_input+0**

**MOVF R0+1, 0**

**MOVWF FARG\_WordToStr\_input+1**

**MOVLW \_TXT\_ENTERA+0**

**MOVWF FARG\_WordToStr\_output+0**

```
CALL    _WordToStr+0
;PROYECTO_ROY.mbas,218 ::      ByteToStr(DISTANCIA,txt)
MOVWF   _Distancia+0, 0
MOVWF   FARG_ByteToStr_input+0
MOVLW   _txt+0
MOVWF   FARG_ByteToStr_output+0
CALL    _ByteToStr+0
;PROYECTO_ROY.mbas,219 ::      Uart1_Write("D")
MOVLW   68
MOVWF   FARG_UART1_Write_data_+0
CALL    _UART1_Write+0
;PROYECTO_ROY.mbas,220 ::      Uart1_Write_Text(txt)
MOVLW   _txt+0
MOVWF   FARG_UART1_Write_Text_uart_text+0
CALL    _UART1_Write_Text+0
;PROYECTO_ROY.mbas,221 ::      Uart1_Write("T")
MOVLW   84
MOVWF   FARG_UART1_Write_data_+0
CALL    _UART1_Write+0
;PROYECTO_ROY.mbas,222 ::      Uart1_Write_Text(TXT_ENTERA)
MOVLW   _TXT_ENTERA+0
MOVWF   FARG_UART1_Write_Text_uart_text+0
```

```
CALL    _UART1_Write_Text+0
;PROYECTO_ROY.mbas,223 ::      Distancia=0
      CLRf    _Distancia+0
      CLRf    _Distancia+1
;PROYECTO_ROY.mbas,224 ::      TEMP_ENTERA=0
      CLRf    _TEMP_ENTERA+0
      CLRf    _TEMP_ENTERA+1
;PROYECTO_ROY.mbas,225 ::      muestras=0
      CLRf    _muestras+0
;PROYECTO_ROY.mbas,226 ::      Distancia_temp=0
      CLRf    _Distancia_temp+0
      CLRf    _Distancia_temp+1
;PROYECTO_ROY.mbas,227 ::      TEMP_ENTERA_TEMP=0
      CLRf    _TEMP_ENTERA_TEMP+0
      CLRf    _TEMP_ENTERA_TEMP+1
L__main106:
;PROYECTO_ROY.mbas,229 ::      WEND
      GOTO    L__main80
      GOTO    $+0
; end of _main
```

## **ANEXO B**

### **Módulo Control AC Carga Resistiva[10]**

El módulo Control AC Carga Resistiva nos permite controlar la potencia suministrada a una carga AC.

### **Especificaciones:**

- Bus de señales para control y detección cruce por cero
- Conectores para entrada y salida 110VAC
- Potencia máxima para carga AC de 600w

### **Aplicaciones:**

Control de potencia para resistencia Térmica

### **Características**

#### **Fuente de Alimentación**

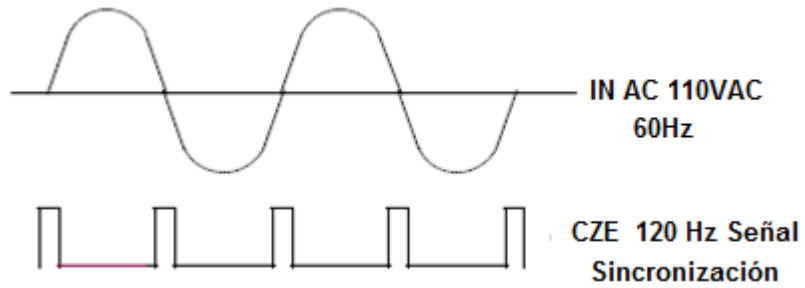
- **IN AC** : Alimentación 110VAC
- **VDD**: 5VDC
- **GND**:0V

#### **Señales de Control**

- **CAR**: Entrada disparadora para triac



- **CZE:** Salida señal de sincronización de cruce por cero



### Salidas

**CARGA:** Salida para carga AC Resistiva



## ANEXO C

### **Módulo Disparador de Relé[10]**

Módulo Disparador de Relé sirve para el manejo de cargas de gran Potencia.

#### **Especificaciones:**

- Led indicador de activación de relé
- Alimentación independiente para relé
- Salidas mediante terminales de potencia

#### **Aplicaciones:**

- Encendido o Apagado de motores AC/DC.
- Control ON/OFF de luces.

#### **Características**

#### **Fuente de Alimentación**

La alimentación se la puede realizar por dos maneras

- 12V de alimentación directa (1)
- 12V de alimentacion por la señales de control (2)

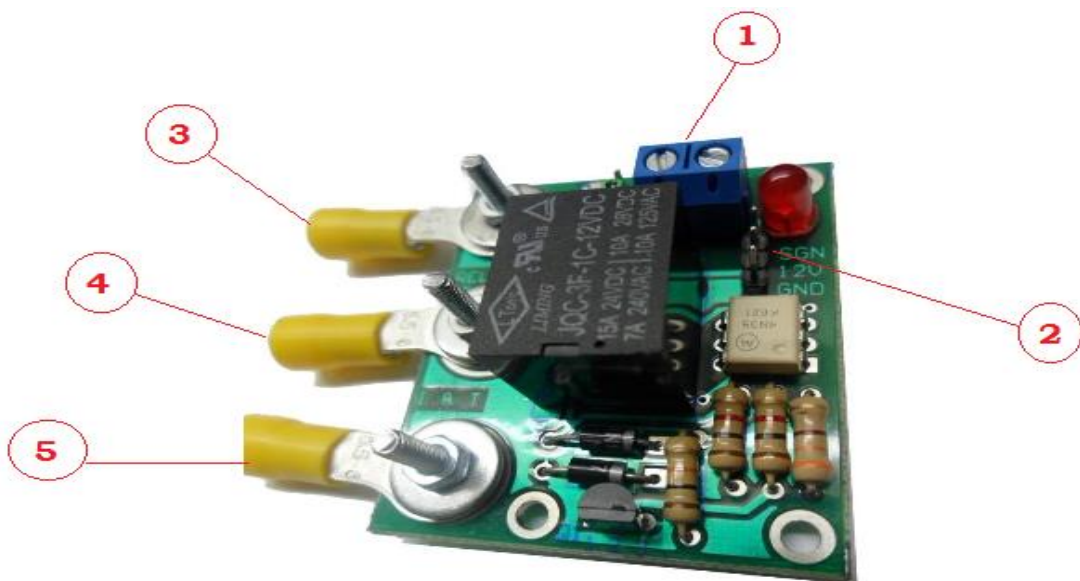
### Señales de Control

- SGN: Señal TTL/CMOS para controlar la conmutación de los relés (2)
- EN/DS: AL seleccionar EN escogemos que la alimentación de 12V sea por el control (2) o DS por el control (1)

### Salidas

- Normalmente Abierto: Cuando el Relé está en estado abierto (3).
- Normalmente Cerrado: Cuando el Relé está en estado cerrado (4).
- Común: Pin común para la referencia en la conmutación del relé (5).

La carga en la salida puede ser con los siguientes parámetros:  
110V/10A; 28V/10A; 24V/15A; 240V/7A.



## **ANEXO D**

### **Módulo P.H.2A I&T 03[10]**

P.H.2A I&T 03 es un módulo para el control de dirección y velocidad de dos motores DC totalmente in dependientes.

#### **Especificaciones:**

- Driver L298
- Control de 2 motores DC.
- Conexión a fuentes para motores.
- Conexión con M.E I&T 04 con bus datos IDC
- Señales de control dirección y velocidad (PWM) por cada motor.
- 2A por canal, 3 A Pico

#### **Aplicaciones:**

- Aplicaciones de control de motores DC.
- Control de intensidad de cargas por PWM.

#### **Características**

#### **Fuente de Alimentación**

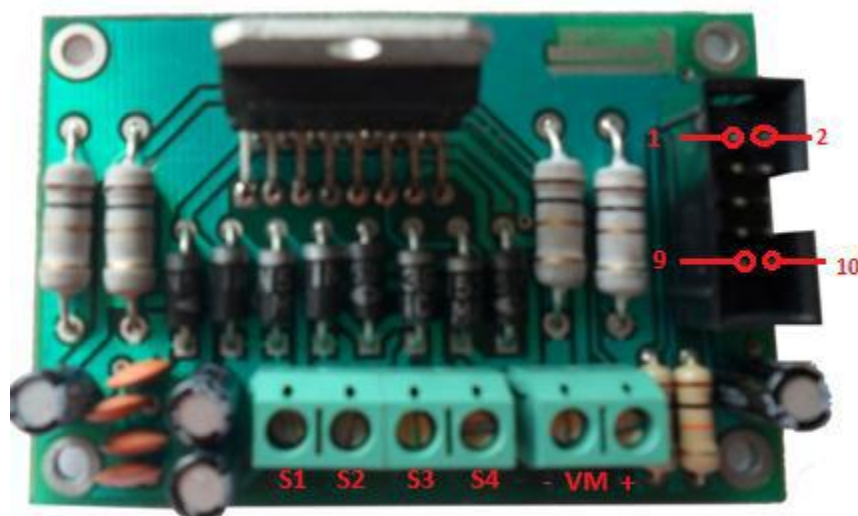
- Alimenntación de Motor VM (5-46)VDC

### Señales de Control

- IN1->PIN1: Entrada de control 1
- IN2->PIN3: Entrada de control 2
- IN3->PIN5: Entrada de control 3
- IN4->PIN7: Entrada de control 4
- Vcc-> PIN 9:Votaje entre( 4.5 – 7) VDC
- ENA-> PIN 4: Señal ENA de habilitación motor A
- ENB-> PIN 2: Señal ENB de habilitación motor B
- GND-> PIN10 : Referencia 0 VDC

### Salidas

- S1,S2-> MOTOR A
- S3,S4 -> MOTOR B



## ANEXO E

### Módulo USB-UART I&T 02[10]

USB - UART I&T 02 es un Módulo de comunicación serial por puerto USB para hacer interface con un computador o PC, permite alimentar circuitos de 5V.

#### Especificaciones:

- Cable mini -USB
- Leds indicadores de TX y RX
- Transferencia de datos 300 Baud hasta 3MBaud
- Soporta 7 a 8 bits datos, 1 o 2 bits stop, y odd/even/mark/space/no parity
- Datos serial con amplitud seleccionables de 5V/3.3V

#### Aplicaciones:

- Interface de comunicación serial con el computador
- Adquisición y envío de datos desde circuitos con microcontroladores hacia el computador.



## ANEXO F

### **Módulo M.E.I&T 03[10]**

M.E. I&T03 es un módulo de entrenamiento y desarrollo que nos permiten realizar múltiples tareas con el microcontrolador 16F886.

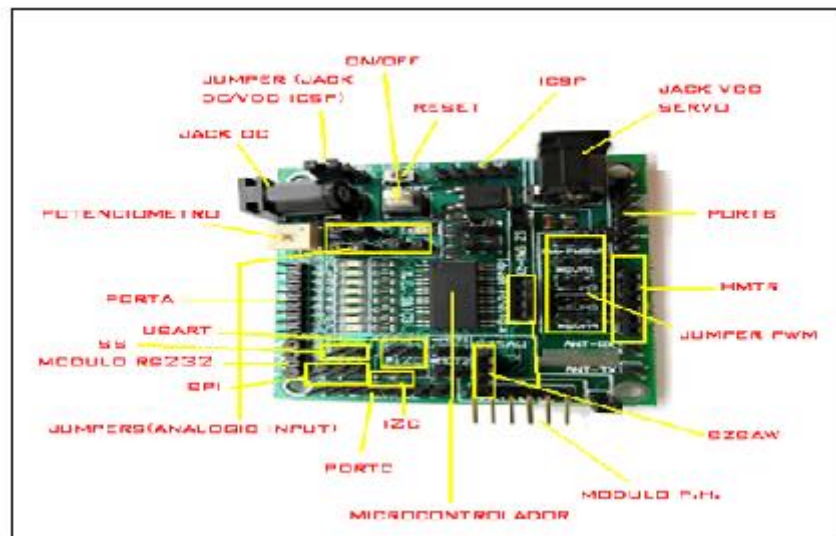
#### **Especificaciones:**

- Comunicación serial asíncrona UART
- Comunicación serial síncrona SPI e I2C
- Comunicación ONE WIRE y USART
- Comunicación inalámbrica RX y TX con módulos FSK y ASK
- Potenciómetro integrado
- 10 entradas analógicas
- 24 entradas y salidas digitales
- 8 leds indicadores de salidas digitales
- Control para 4 servomotores
- Control para 2 motores DC (Dirección y Velocidad)
- Programación ICSP in circuit
- Reset manual
- Switch de ON/OFF
- Led indicador de power
- Regulador integrado

### Aplicaciones:

- Construcción de robots (seguidores de líneas, sumobot, teleoperados, exploradores, soccer, etc.)
- Aplicaciones de Telemetría y radio control
- Implementación de sistemas de control
- Tarjeta de adquisición de datos
- Placa de desarrollo de ejercicios de programación con microcontroladores

### Características



### Fuente de Alimentación

- Alimentación desde (5 - 35) VDC en el Jack DC.
- Interruptor ON/OFF para energizar o desenergizar al módulo de entrenamiento.



- Jumper (Jack DC / VDD ICSP) permite seleccionar la fuente que va a alimentar al módulo la cual puede ser mediante el JACK DC o VDD ICSP que es la energía que el programador de PIC puede suministrar.

### **Microcontrolador PIC16F886**

- 25/24 Pines Entradas/Salidas.
- Puertos (PORT A, B, C, E).
- Oscilador interno seleccionable entre (31KHZ – 8MHZ).
- Rango de voltaje de Operación (2 – 5.5) VDC
  
- 11 entradas analógicas con 10 bit de resolución.
- 3 Timers (Timer0 8bits, Timer1, 2 16bits).
- 2 PWM (CCP) de 10bits, frecuencia máx. 20KHZ
- Comunicaciones seriales sincrónicas MSSP (SPI (4 modos), I2C)
- Módulo USART (RS-485, RS-232 and LIN 2.0)

### **Reset**

Este botón posee un resistor pull up y está conectado al PIN MCLR.

Para utilizar este botón es necesario que se lo habilite mediante software.

ICSP (Programación serial en circuito). Este conector se lo utiliza para cargar el código en el microcontrolador usando cualquier programador que tenga terminales ICSP: Pickit 2, 3, etc.

El jumper Jack DC/VDD ICSP permite alimentar el Módulo de entrenamiento con el programador, habilitando en power target del programador.

Si se quiere programar el microcontrolador mientras se está utilizando JACK DC, se debe de deshabilitar power target del programador.

Entradas/Salidas en PORT A, B, C.

Al no colocar los jumpers que se indica en la figura mediante cuadros de color amarillo nos permite dejar libres los pines del PORTA, B y C. Esto nos permite hacer cualquier tipo de configuración adicional para utilizarlos con I/O de datos.

Cada PORT tiene 8 pines correspondientes a los 8 bits para datos.

- PORTA (0,1,2,3,4,5,6,7)
- PORTB (0,1,2,3,4,5,6,7)
- PORTC (0,1,2,3,4,5,6,7)

### **Entradas Analógicas**

AN0=RA0, AN1=RA1, AN2=RA2, AN3=RA3, AN4=RA5, AN8=RB2,  
AN9=RB3, AN10=RB1

AN11=RB4, AN12=RB0, AN13=RB5

### **Leds en PORTA**

Para utilizar los leds del PORTA, debemos setear los pines de este puerto como salidas, además debemos colocar los jumpers a0/d0, a5/d5 y Leds. De esta manera quedan habilitados todos los indicadores led del módulo de entrenamiento.

Jumper Leds: Habilita todos los led conectándolos a GND.

Jumper a0/d0: Conecta el led D0 con el pin A0, habilitando el LED0.

Jumper a5/d5: Conecta el led D5 con el pin A5, habilitando el LED5.

### **Analogic PORTA**

Para utilizar el potenciómetro del módulo de entrenamiento se debe colocar el Jumper P0/a0 tal como se indica en la figura de esta manera, el cual permite conectar el Pin A0 del PIC con el potenciómetro.

Mediante la variación del POT se producirá un voltaje analógico entre (0 – VCC) el cual será enviado al PIN A0.

Control de Velocidad y motores DC.

El módulo de entrenamiento le permite controlar la dirección y velocidad de motores DC.

En el módulo hay dos terminales Mot1, Mot2 cada terminal tiene un Pin DIR y PWM.

DIR: Señal digital (1 delante – 0 atrás) pin

PWM: Señal de modulación por ancho de pulso.

Módulo P.H.I&T 03: Es un terminal para utilizar un módulo puente H ya diseñado.

Ejemplo: Módulo Puente H motores DC P.H. I&T 03

Control de Servomotor

Podemos controlar 4 servomotores, los cuales poseen 3 señales.

SVM: Este pin necesita una señal PPM

5VDD: Alimentación de 5VDC para el servomotor

GND: Tierra (0 VDC)

Jumper PW/PWSVM: Permite seleccionar la fuente de alimentación para los servomotores la cual puede ser la del módulo de entrenamiento o una externa a través de JACK VDD SERVO.

Comunicación Serial UART

Este módulo nos permite realizar varias aplicaciones con comunicación serial

PIC TX: Pin de transmisión de datos seriales UART

PIC RX: Pin de transmisión de datos seriales UART

Módulo TTL I&T 03: Es un terminal para utilizar un módulo para comunicación RS232 ya diseñado.

Ejemplo: Módulo RS232 - TTL I&T 03

### **Módulo SZSAW de Radiofrecuencia**

Este módulo de entrenamiento le permite hacer el uso de comunicación inalámbrica SZSAW (Simplex, ASK), estos módulos tienen una comunicación simplex por ello debemos hacer uso del módulo TX y RX.

#### **Pines del módulo TX SZSAW: SZSAW:**

1. - Data (PIC\_TX)

2.-Vcc (5V)

#### **Pines del módulo RX**

1. - GND

2 y 3. - Data (PIC\_RX)

3.-GND

4.-Vcc (5V)

### **Módulo HMTR de Radiofrecuencia**

Podemos hacer el uso de estos excelentes módulos HMTR (Half dúplex, FSK)

En nuestro módulo de entrenamiento, permitiéndonos hacer múltiples aplicaciones.

- Radiocontrol y radio monitoreo.
- Telemetría
- Aplicaciones inalámbricas en robots.
- Aplicaciones controladas por computador, etc.

### **Comunicación serial síncrona SPI**

- Para utilizar el PIN RA5 como SS se debe colocar el jumper ss/a5 tal como se indica.
- **Recordar:** Cuando se utiliza las Comunicaciones I2C y SPI se debe configurarlo como máster o esclavo.
- **SS:** Señal de entrada para selección del esclavo.
- **SPI:** Este conector es para realizar la comunicación SPI.

- **SCK:** Serial clock.
- **SDI:** Entrada de datos seriales.
- **SDO:** Salida de datos seriales.

### **Comunicación serial síncrona I2C**

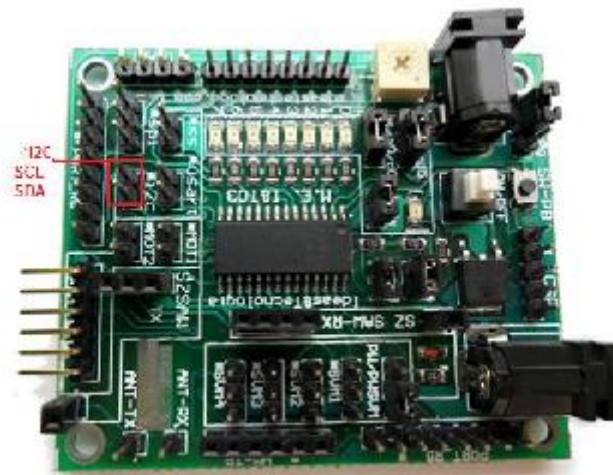
- **I2C:** Este conector sirve para para realizar la comunicación I2C.
- **SCL (system clock):** Señal de clock
- **SDA (system data):** Salida/Entrada (maestro/esclavo) de datos serial.

Los pines de SCL y SDA están como I/O con pull up para la comunicación I2C.

### **Aplicaciones de la Tarjeta de adquisición de datos M.E. I&T 03 son:**

- Construcción de robots (seguidores de líneas, sumobot, teleoperados, exploradores, soccer, etc.).
- Aplicaciones de Telemetría y radio control.
- Implementación de sistemas de control.
- Tarjeta de adquisición de datos.

- Placa de desarrollo de ejercicios de programación con microcontrolador.



## ANEXO G

### Módulo M.E.I&T 04[10]

M.E. I&T04 es un módulo de entrenamiento y desarrollo que nos permite realizar múltiples tareas con el microcontrolador 16F886.

#### Especificaciones:

- Comunicación serial asíncrona UART
- Comunicación serial síncrona SPI e I2C
- Comunicación ONE WIRE y USART
- Comunicación inalámbrica RX y TX con módulos FSK y ASK
- Potenciómetro integrado
- 10 entradas analógicas
- 24 entradas y salidas digitales
- 8 leds indicadores de salidas digitales
- Control para 4 servomotores
- Control para 2 motores DC (Dirección y Velocidad)
- Programación ICSP in circuit
- Reset manual
- Switch de ON/OFF
- Led indicador de power
- Regulador integrado



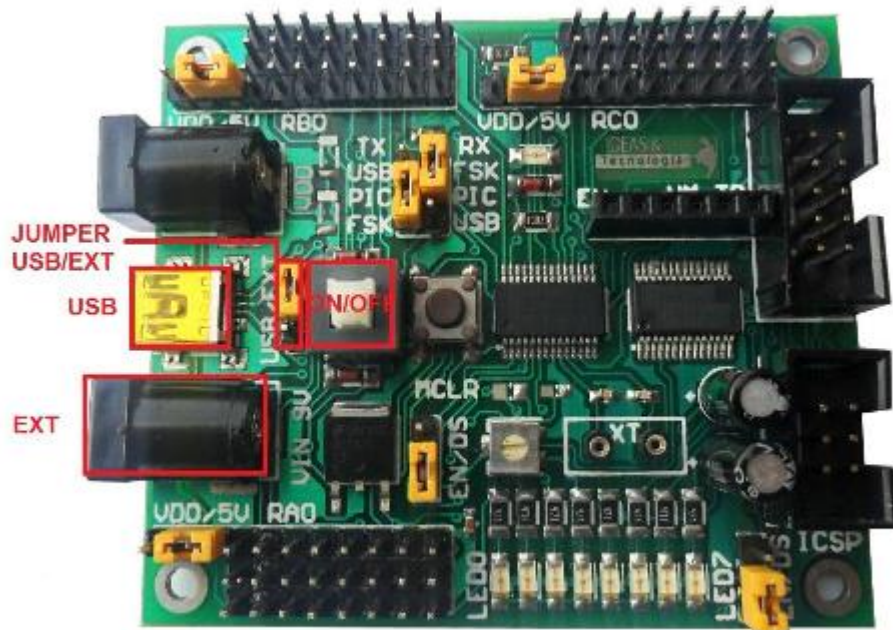
**Aplicaciones:**

- Construcción de robots (seguidores de líneas, sumobot, teleoperados, exploradores, soccer, etc.)
- Aplicaciones de Telemetría y radio control
- Implementación de sistemas de control
- Tarjeta de adquisición de datos

**Características**

## Fuente de Alimentación

- Alimentación desde (5 - 25) VDC en el EXT (VIN 9V)
- Interruptor ON/OFF para energizar o desenergizar al módulo de entrenamiento.
- Jumper (USB/ EXT) permite seleccionar la fuente que va alimentar al módulo la cual puede ser mediante el EXT (VIN 9V) o USB que es la energía que se obtiene a través del puerto USB.



### Microcontrolador PIC16F886

- 25/24 Pines Entradas/Salidas.
- Puertos (PORT A, B, C, E).
- Oscilador interno seleccionable entre (31KHZ – 8MHZ).
- Rango de voltaje de Operación (2 – 5.5) VDC
- 11 entradas analógicas con 10 bit de resolución.
- 3 Timers (Timer0 8bits, Timer1, 2 16bits).
- 2 PWM (CCP) de 10bits, frecuencia máx. 20KHZ
- Comunicaciones seriales sincrónicas MSSP (SPI (4 modos), I2C)
- Módulo USART (RS-485, RS-232 and LIN 2.0)

### **Reset (MCLR)**

Este botón posee un resistor pull up y está conectado al PIN MCLR.

Para utilizar este botón es necesario que se lo habilite mediante software.

### **ICSP (Programación serial en circuito)**

Este conector IDC 3X2 se lo utiliza para cargar el código en el microcontrolador usando cualquier programador con terminales ICSP.

El módulo de entrenamiento puede ser alimentado con el programador, habilitando en power target en el software.

### **Entradas/Salidas en Puertos A, B, C**

Cada puerto tiene 8 pines correspondientes a los 8 bits, a cada bit se denomina Señal I/O acompañada de pines de +Vcc y Gnd, donde Vcc puede ser seleccionable es decir utilizar el voltaje interno del Módulo entrenamiento o externo mediante el Jack VDD.

- PORTA (0,1,2,3,4,5,6,7)
- PORTB (0,1,2,3,4,5,6,7)
- PORTC (0,1,2,3,4,5,6,7)22,4+3,95+0,8

### **Entradas Analógicas**

AN0=RA0, AN1=RA1, AN2=RA2, AN3=RA3, AN4=RA5, AN8=RB2,  
AN9=RB3, AN10=RB1

AN11=RB4, AN12=RB0, AN13=RB5

### Led en PORTB

Para utilizar los led del PORTB, debemos setear los pines de este puerto como salidas, además debemos de habilitar el jumper LED (**EN/DS**). De esta manera quedan habilitados todos los indicadores led del módulo de entrenamiento.

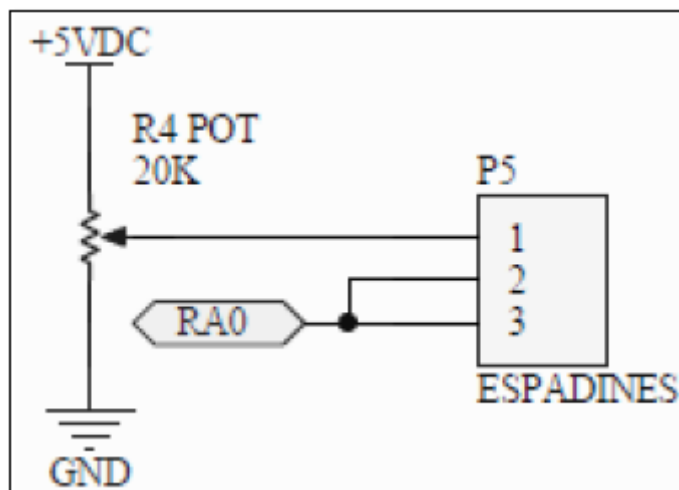
**Conecciones: LED0-> RB0 LED7-> RB7**

**Jumper Led: EN** Habilita todos los led conectándolos a GND.

**Jumper Led: DS** Deshabilita todos los led.

### Potenciómetro en PORTA

Para utilizar el potenciómetro del módulo de entrenamiento se debe colocar el Jumper **POT (EN/DS)** tal como se indica en la figura de esta manera, el cual permite conectar el Pin RA0 del PIC con el potenciómetro. Mediante la variación del POT se producirá un voltaje analógico entre (0 – 5VDC) el cual será enviado al PIN A0.



### Control de motor DC

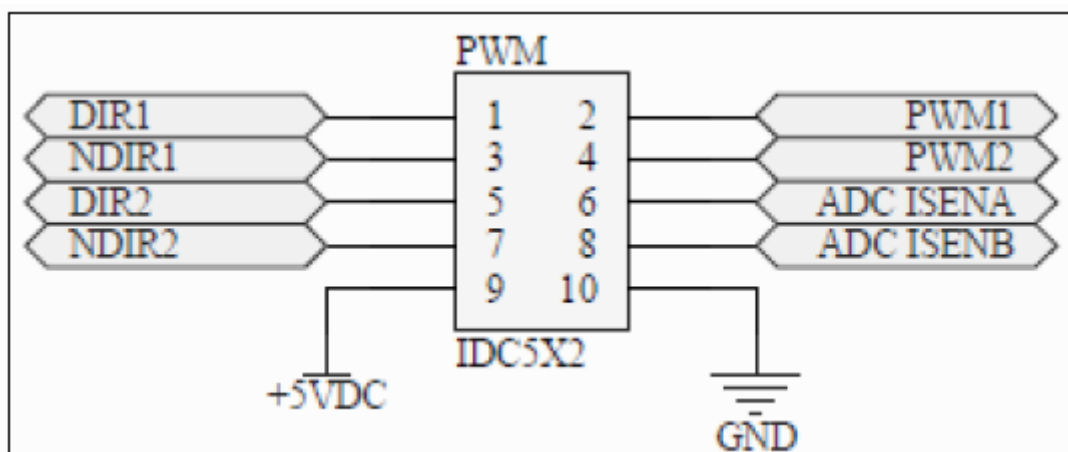
El módulo de entrenamiento M.E I&T 04 permite controlar la dirección y velocidad de motores DC.

Para aquello se agregó un conector IDC de 5X2 compatible eléctricamente con los módulos P. H I&T04, P.H.2A I&T (Puente H para motores DC),y otros módulos desarrollados por IDEAS&TECNOLOGIA.

### Control de motor DC

El módulo de entrenamiento M.E I&T 04 permite controlar la dirección y velocidad de motores DC.

Para aquello se agregó un conector IDC de 5X2 compatible eléctricamente con los módulos P. H I&T04, P.H.2A I&T (Puente H para motores DC),y otros módulos desarrollados por IDEAS&TECNOLOGIA.



**DIR1:** RA2

**NDIR1:** RA4

**DIR2:** RA3

**NDIR2:** RA5

**PWM1:** RC1

**PWM2:** RC2

**ADCISENA:** RB1

**ADCISENB:** RB2

### Control de Servomotor

Podemos conectar hasta 24 servomotores, debido a que a cada puerto se le agrego pines de +Vcc y Gnd según el estándar de los servomotores los cuales poseen 3 señales.

**Señal:** Este pin necesita una señal PPM

**VDD/5V:** Alimentación a través del jack **VDD** si se requiere más corriente o se utiliza el voltaje del Módulo de entrenamiento **5V**

**GND:** Tierra (0 VDC)

**Jumper VDD/5V:** Permite seleccionar la fuente de alimentación para los servomotores la cual puede ser la del módulo de entrenamiento o una externa a través de JACK **VDD**.

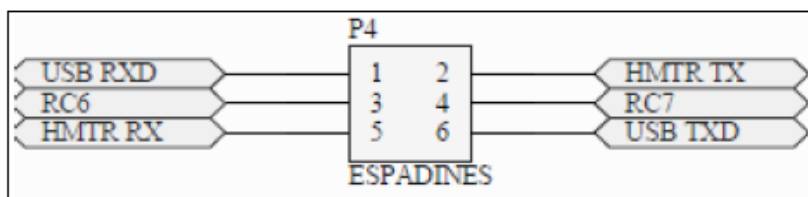
### Comunicación Serial UART/USB/FSK

Este módulo nos permite realizar varias aplicaciones con comunicación serial por este motivo se le incorporo un Módulo de comunicación UART-USB y de radiofrecuencia UART-FSK.

Mediante el Jumper USB-PIC-FSK podemos realizar varias selecciones para diferentes configuraciones .

**PIC TX:** Pin RC6 de transmisión de datos seriales UART

**PIC RX:** Pin RC7 de recepción de datos seriales UART



## BIBLIOGRAFÍA

[1] COSS BU, RAÚL. (2003) Simulación. Un Enfoque Práctico. ISBN: 968-18-1506-8. Editorial: LIMUSA, 11 - 12.

[2] HIMMELBLAU DAVID Y KENNETH BISCHOFF (2004). Análisis y Simulación de Procesos, ISBN 84-291-7235-1. Editorial: Reverté S.A., 2 - 5.

[4] LABVIEW, GETTING STARTED WITH LABVIEW, gtstrtlv.pdf, (2003) Ed., U. S.

[6] LÁZARO, ANTONIO M. (2001). LabVIEW Programación Gráfica para el Control de Instrumentación, Paraninfo, España.

[8] Dorf-Bishop (2005) Sistemas de control moderno, Editorial Pearson.

[9] [www.microchip.com](http://www.microchip.com)

Fecha de Consulta Diciembre del 2011

[10] [www.ideastechnology.com](http://www.ideastechnology.com)

Fecha de Consulta Noviembre del 2011

[11] O'Connell J. *Heating water: Rate correction due to Newtonian cooling.* The Physics Teacher Vol 37, December 1999, pp. 551-552.

[12] Krupska A., Krupski M., Konarski J. *Phase delay effect in a thermally homogeneous system heated periodically.* Eur. J. Phys. 22 (2001) pp. 133-138.

[13] [www.nationalinstruments.com](http://www.nationalinstruments.com)

Fecha de Consulta Abril del 2012

[14] [www.dspace.espol.edu.ec](http://www.dspace.espol.edu.ec) TESIS DE GRADO “DISEÑO Y CONSTRUCCIÓN DE UN MODULO ORIENTADO AL DESARROLLO DE PRÁCTICAS DE INSTRUMENTACIÓN INDUSTRIAL BAJO LA PLATAFORMA DE LABVIEW” Leyes-Navarrete 2007.

[15] Johnson, C. D. (1984). *Microprocessor-based process control.* Nueva Jersey: Prentice-Hall.