Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

“Sistema de Adquisición de Datos de Humedad y Temperatura utilizando Tecnología 1 WIRE y Labview”

TESINA DE SEMINARIO

Previo a la obtención del título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

DIÓGENES GEOVANNY MERA VILLAVICENCIO

FERNANDO JAVIER VALDIVIESO FEIJOO

Guayaquil – Ecuador

2011

AGRADECIMIENTO

Nuestro agradecimiento va dirigido principalmente a Dios, ya que gracias a sus bendiciones y su iluminación nos permitió culminar nuestra carrera universitaria.

También agradecemos a esta prestigiosa institución, ESPOL, y a nuestros profesores de la carrera por compartir sus conocimientos; en especial a nuestro tutor de proyecto, Ing. Luis Fernando Vásquez Vera, por brindarnos amistad, comprensión, experiencia y apoyo incondicional durante el desarrollo de este proyecto.

Y no podemos dejar de agradecer a nuestros padres que con su amor han sido las bases fundamentales a lo largo de nuestras vidas

DEDICATORIA

Esta tesis va dedicada a mi madre Leonor Eufemia Feijoo Ocampo, mi amiga inseparable que me ha acompañado a lo largo de mi carrera estudiantil, brindándome su apoyo incondicional. Siendo la persona que me conduce por el buen camino. A mi hermano, Alfredo Francisco Valdivieso Feijoo, quien me ayudó y estuvo a mi lado durante mi carrera universitaria, guiándome y aconsejándome para no fallar en mis decisiones.

A mis amigos que de una forma u otra me ayudaron a seguir adelante, esperando poder ayudarlos para que lleguen pronto a la culminación de su carrera.

Fernando J. Valdivieso Feijoo

DEDICATORIA

Este trabajo va dedicado a mi madre Mariana Villavicencio, quien me guió por el camino del bien y confió en mí en todo momento. A mi hermana Beatriz Mera, quien me brindó su apoyo incondicional. A mi esposa Erica Gutiérrez, que siempre estuvo ahí en mis desveladas de estudios acompañándome, por la paciencia que me tiene y que la amo mucho. A mis hijos Kenjy y Erick, que son mi fortaleza día a día para seguir luchando, quienes con sus travesuras me sacan una sonrisa.

A mis familiares y amigos, que siempre me ayudaron a salir adelante, preocupándose y dándome ánimos en los momentos más difíciles.

Geovanny Mera Villavicencio

TRIBUNAL DE SUSTENTACIÓN

_____	_____
Luis Fernando Vásquez, Msc	Washington Medina Msc.
Profesor de Seminario de Graduación	**Delegado del Decano**

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL)

_____ _____

Diógenes Mera V. Fernando Valdivieso F.

RESUMEN

En este tiempo, son muy necesarios los sistemas de control y monitoreo, mediante adquisición de datos. Y qué mejor que la tecnología que va en ascenso, para ir mejorando estos sistemas.

En este proyecto se realiza un sistema de monitoreo y control de temperatura y humedad de un área específica. Manteniendo así los valores óptimos de los parámetros mencionados. Para implementarlo usaremos dos herramientas bases: La tecnología 1 wire que se basa en la programación de un micro-controlador y el software de lenguaje gráfico Labview con su tarjeta de adquisición de datos DAQ NI USB-6009.

En el primer capítulo explicaremos la parte teórica de los sensores de temperatura y humedad, el micro-controlador usado, los comandos del programa Labview que aplicamos y los procedimientos que se utilizarán en el proyecto. Todo esto lo hacemos de manera detallada con gráficos y protocolos que se deben seguir, para aprovechar el mejor rendimiento.

Para utilizar la tecnología 1-wire hacemos uso de un micro-controlador, el mismo que nos ayuda a manejar los tiempos y protocolos que se debe seguir para que exista una buena transmisión de datos con el sensor de temperatura en la tecnología 1-wire. Todo esto se explica en el segundo capítulo.

En el tercer capítulo detallamos la programación gráfica de Labview siguiendo cada proceso mediante un diagrama de bloque para un mejor entendimiento. También generamos una interfaz entre el computador y el usuario con el cual se podrá tener visualización del monitoreo de la temperatura y humedad en conjunto con la tarjeta DAQ NI USB-6009.

Finalmente, en el cuarto capítulo se trabaja en la creación de un prototipo para la demostración del sistema. En este tenemos varias etapas: La parte de adquisición compuesta por los sensores de temperatura y humedad junto con la tarjeta de adquisición de datos, la de control que es nuestro ordenador y la de fuerza la cual maneja la calefacción y el acondicionador de aire.

INDICE GENERAL

Agradecimiento……..*II*

Dedicatoria……*III*

Tribunal de Sustentación……*V*

Declaración Expresa……...*VI*

Resumen……..*VII*

Índice General……..*IX*

Índice de Figuras.……..*XIII*

Índice de Tablas……*XV*

Abreviaturas……...*XVI*

Introducción……...……...*XVII*

Capítulo 1……..……..……...*1*

1. Marco Teórico……...……...…....*1*

1.1. Sensores……...*1*

1.1.1. Sensor HIH-3610……..……...*3*

1.1.1.1. Características HIH-3610……...*4*

1.1.2. Sensor DS-2438……...*4*

1.1.2.1. Descripción de Pines DS-2438……..*5*

1.1.2.2. Descripción del DS-2438……....*6*

1.1.2.3. Características del DS-2438……..*7*

1.2. 1-Wire……....*8*

1.2.1. Ventajas del 1-Wire……..8

1.2.2. Diagrama de la Red 1-Wire……....9

1.2.3. Señales del Protocolo 1-Wire……...10

1.2.4. Reinicio del Bus……..11

1.2.5. Envío y Recepción de Datos……....11

1.3. Micro-controlador……...13

1.3.1. ¿Qué es un Micro-controlador?..……....13

1.3.2. Características del Micro-controlador……13

1.3.3. Partes que componen un Micro-controlador……..14

1.3.4. PIC16F886……..15

1.4. Labview 2010……18

1.4.1. Panel Frontal……...22

1.4.1.1. Indicadores……22

1.4.1.2. Controles y Switch……24

1.4.1.3. Gráficos y Leds……..27

1.4.2. Diagrama de Bloques……....28

1.4.2.1. DAQ-Assist……28

1.4.2.2. Fórmula……...32

1.4.2.3. Comparaciones Condicionales……....33

1.4.2.4. Lazos……...34

1.4.2.5. Funciones Especiales……...35

1.4.3. Tarjeta de Adquisición de Datos NI USB-6009……38

1.4.3.1. Características de Entradas y Salidas……...39

Capítulo 2……44

2. Programación del PIC……..44

2.1. Rutinas 1-Wire……...44

2.1.1. Rutina de Reinicio……..45

2.1.2. Rutina de Salto de Memoria (CCh)……46

2.1.3. Rutina de Escribir (4Eh)……....46

2.1.4. Rutina de Conversión a Temperatura (44h)..……...46

2.1.5. Rutina de Recuperación de Datos.(B8h)……...47

2.1.6. Rutina de Lectura (BEh)……...47

2.2. Creación de Proyecto……...48

2.2.1. Creación de Código Fuente……..49

Capítulo 3……55

3. Creación del Programa en Labview……...55

3.1. Diagrama de Bloque General……..55

3.1.1. Bloque de Adquisición de Datos……..56

3.1.2. Bloque Convertidor……....58

3.1.3. Bloque Mostrar Datos……....59

3.1.4. Bloque Comparador……...60

3.1.5. Bloque Generador de Alarmas……61

3.1.6. Bloque Generador y Envío de Informe……...62

3.1.7. Bloque de Envío de Señales de Control: Automático Y Manual...66

Capítulo 4……68

4. Creación del Prototipo……..68

Conclusiones

Recomendaciones

Anexo 1

Anexo 2

Anexo 3

Anexo 4

Anexo 5

Anexo 6

Anexo 7

Referencias Bibliográficas

INDICE FIGURAS

1. *Figura 1: Gráfica de la salida de tensión del Sensor vs humedad Relativa a una específica temperatura…….……3*
2. *Figura 2: Pines del DS2438……...5*
3. *Figura 3: Diagrama de una red 1-Wire……..9*
4. *Figura 4: PIC16f886……...18*
5. *Figura 5: Labview 2010……20*
6. *Figura 6: Proyecto con su panel frontal y diagrama de bloques……....21*
7. *Figura 7: Indicadores Numéricos……23*
8. *Figura 8:Obtención de Indicadores Numéricos……23*
9. *Figura 9: Indicador de Texto……24*
10. *Figura 10: Controles y Botones……...……25*
11. *Figura 11: Obtención de Controles y Botones……..……26*
12. *Figura 12: Gráfico y Leds……...…...27*
13. *Figura 13: Icono de la DAQ ASSIST……..….29*
14. *Figura 14: Configuración DAQ Assistant……....30*
15. *Figura 15: Canales por líneas……...31*
16. *Figura 16: Canales por puertos……....31*
17. *Figura 17: Función Fórmula……..32*
18. *Figura 18: Comparadores……..33*

19. *Figura 19: Condicionales……...34*
20. *Figura 20: Lazos……35*
21. *Figura 21: Format Date/Time String Function……...36*
22. *Figura 23: Build Path Function……36*
23. *Figura 24: Write To text File Function……36*
24. *Figura 25: Build Array……37*
25. *Figura 26: SMTP Email Send File……....38*
26. *Figura 27: DAQ NI USB-6009……...42*
27. *Figura 28: Nuevo proyecto en Micro Basic……48*
28. *Figura 29: Escribir código fuente……..49*
29. *Figura 30: Diagrama de Bloque General……....56*
30. *Figura 31: Ventana de Configuración de la DAQ……..57*
31. *Figura 32: Index Array Function……...58*
32. *Figura 33: Icono y configuración Convert from Dynamic Data……...59*
33. *Figura 34: Encabezado y Nombre del Archivo……..63*
34. *Figura 35: Archivo que se genera……....64*
35. *Figura 36: Proceso de Envío de Correos……...65*
36. *Figura 37: Campos del Mail Server……66*
37. *Figura 38: Control Sistema……...68*

INDICE TABLAS

1. *Tabla I: Rangos de tiempos para realizar la comunicación……...12*
2. *Tabla II: Terminales analógicas de la DAQ NI USB-6009……...42*
3. *Tabla III: Terminales digitales de la DAQ NI USB-6009…...……...43*
4. *Tabla IV: Secuencia de Rutinas……..……....45*
5. *Tabla V: Datos en Binario y Decimal (Temperatura Real)……...51*
6. *Tabla VI: Datos que se almacenan en DS2438…….……..53*

ABREVIATURAS

RH: Humedad Relativa

Vout: Voltaje de Salida

Gnd: Tierra

NC: No Conectado

DQ: Pin de comunicación 1 Wire del Ds2438

VDD: + 5 V

A/D: Analógico/Digital

Us: Microsegundos

CRC: Código de Detección de errores

E/S: Entrada/Salida

SSP: Puerto Serial Sincrónico

SPI: Interface de Periféricos Seriales

I2C: Entre Circuitos Integrados

PWM: Modulación por Ancho de Pulso

CCP: Comparación, Captura, Pwm

G: Lenguaje Gráfico de Programación

VI: Instrumentos Virtuales

NI: National Instruments

EP: Entradas Programables

INTRODUCCIÓN

El control de humedad y temperatura, es un sistema muy común en nuestro medio. Nosotros necesitamos mantener siempre en un área determinada un rango de temperatura y humedad óptimo para tener un ambiente estable.

En este trabajo nos hemos planteados, aprender conceptos básicos sobre programación gráfica y manejo de herramientas propias del software LABVIEW, aplicándolas a datos adquiridos en tiempo real.

Para ello implementaremos un sistema de control y monitoreo utilizando las aplicaciones de Labview en conjunto con la tecnología 1-wire y sensores de temperatura y humedad.

Además se realizará un informe detallado de los datos adquiridos por nuestro sistema y enviarlo a través de correos electrónicos a las personas que están realizando el monitoreo.

CAPITULO 1

1. MARCO TEÓRICO

En esta parte del informe se detalla el funcionamiento teórico de los elementos y herramientas a utilizar en este proyecto. Estos por separado cumplen un rol muy importante, y en conjunto hacen un sistema confiable y muy accesible a las personas.

1.1 Sensores

Un sensor es un dispositivo capaz de medir magnitudes físicas o químicas, llamadas variables de instrumentación y transformarlas en variables eléctricas.

Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, etc. Una capacidad eléctrica, como en un sensor de humedad, una tensión eléctrica, como en un termopar, una corriente eléctrica, como en un fototransistor, etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación con lo que se puede decir

también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como por ejemplo el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra. Áreas de aplicación de los sensores: industria automotriz, industria aeroespacial, medicina, industria de manufactura, robótica, etc.

Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a una base de datos, la toma de valores desde el sensor (1).

En este proyecto vamos hacer énfasis a los sensores de humedad y temperatura, ya que tendremos que estar monitoreando el ambiente de una área específica.

Dichos sensores son:

- Sensor de humedad *HIH3610.*
- Sensor de temperatura *DS2438.*

1.1.1. Sensor HIH3610

Este sensor convierte directamente humedad a tensión con circuito incorporado de acondicionamiento y viene encapsulado en un pequeño dispositivo de 3 pines. Dos de ellas se conectan a una fuente regulada de 5V, mientras que la tercera brinda una tensión de salida lineal que es proporcional a la humedad, como se puede observar en la ***figura 1***, en la misma está a una temperatura regulada de 25ºC, esa pendiente varía de acuerdo a la temperatura.

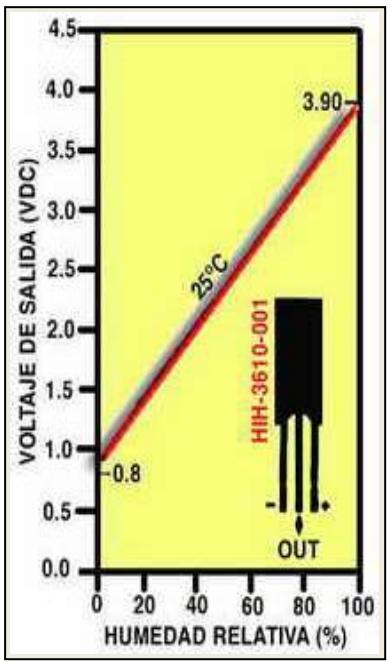


Figura 1: Gráfica de la salida de tensión del sensor vs humedad relativa a una específica temperatura (2).

1.1.1.1. Características HIH 3610:

Dicho sensor tiene diversas características, aquí mencionaremos las utilizables:

- Tiene una expresión matemática que depende del voltaje de salida del integrado y viene dada por: RH (%) = (Vout – 0.958)/0.0307; la misma que se ha tomado de una temperatura promedio 25ºC (2).
- Rango de temperatura operando es entre - 40ºC y 85ºC.
- Rango de humedad operando es entre 0% y 100%.
- Voltaje de salida depende de la temperatura en que esté operando el sensor. En este tomaremos temperatura ambiente y este oscila entre 0.8Vdc y 3.97Vdc.

1.1.2. Sensor DS2438

Este integrado tiene la función de medir la temperatura, mediante un sensor interno. En la ***figura 2*** se muestra la distribución de los pines de este dispositivo.

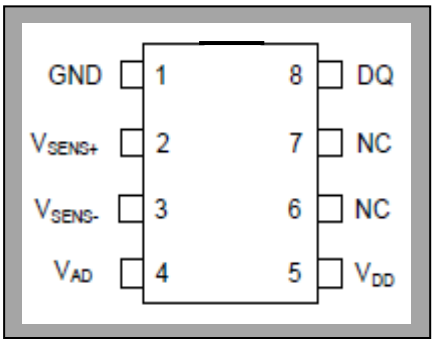


Figura 2: Pines del DS2438 (2).

1.1.2.1. Descripción de los Pines del Integrado

- Pin 1.- GND conectado a tierra.
- Pin 2.- VSENS+ conexión para corriente de la batería a controlar.
- Pin 3.- VSENS- conexión para corriente de la batería a controlar.
- Pin 4.- VAD entrada de convertidor analógico digital.
- Pin 5.- VDD entrada de voltaje de alimentación.
- Pin 6.- NC no conectado.
- Pin 7.- NC no conectado.
- Pin 8.- DQ entrada o salida de datos por medio 1-Wire.

1.1.2.2. Descripción del DS2438

El DS2438 proporciona varias funciones que son deseables para llevar en un paquete de baterías: una forma de etiquetar un paquete de baterías con un número de serie único, un sensor de temperatura directa a digital, que elimina la necesidad de termistores en la batería, un convertidor A/D, que mide el voltaje de la batería y la corriente, un acumulador de corriente integrada, que mantiene un total acumulado de todas las corrientes que entran y salen de la batería, un medidor de tiempo transcurrido, y 40 bytes de memoria EEPROM no volátil para el almacenamiento de los parámetros importantes tales como la química de la batería, capacidad de la batería, la metodología de carga y fecha de montaje. La información se envía desde y hasta el DS2438 más de una interfaz de 1-wire, por lo que sólo un cable debe estar conectado a un microprocesador central a una DS2438. Esto significa que los paquetes de batería de la necesidad sólo tiene tres conectores de salida: energía de la batería, el suelo, y la interfaz de 1-Wire.

Las solicitudes para el monitor de baterías inteligentes incluyen computadoras portátiles, teléfonos móviles,

portátiles y paquetes de baterías de instrumentación en los que es fundamental para controlar el rendimiento de la batería en tiempo real (3).

1.1.2.3. Características del DS2438

- Interfaz única 1-Wire requiere sólo uno de los pines del puerto para la comunicación.
- Elimina termistores mediante la detección de temperatura de la batería en el chip.
- Contador de tiempo en formato binario.
- Memoria no volátil de usuarios de 40 bytes disponibles para el almacenamiento de baterías de datos específicos.
- Opera en el rango de -40ºC y 85ºC.

Como con todos los elementos electrónicos, con los sensores hay que tener cuidado en el momento de maniobrarlos, dado que la humedad o el sudor de la mano podría afectar su funcionamiento, o dañarlo.

1.2. 1-WIRE

1-Wire es un protocolo de comunicaciones en serie la misma que está basado en un bus de datos, más la referencia a tierra para poder establecer una comunicación entre los integrados.

Al realizar este diseño necesitamos de un maestro el cual inicia y controla la comunicación con varios dispositivos esclavos. Cada esclavo tiene un único número de identificación e inalterable, que vienen programadas desde fábrica y consta de 64 bits (4).

1.2.1. Ventaja del 1-Wire

La gran ventaja, es que solo se necesita de un cable para enviar y recibir diferentes tipos de datos de diferentes dispositivos y no tener un cable por cada integrado.

1.2.2. Diagrama de una red 1-Wire

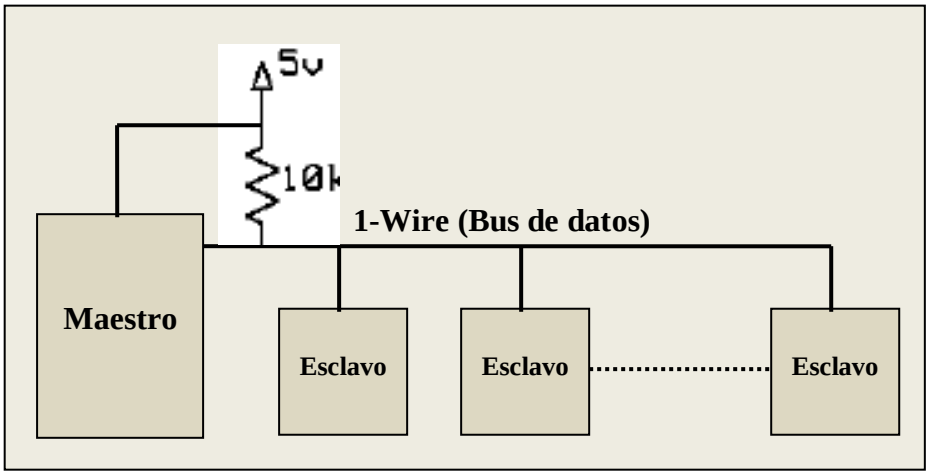


Figura 3: Diagrama de una red 1-Wire.

En el diagrama presentado en la ***figura 3*** nos muestra cómo se debe conectar una red 1-Wire. Cada dispositivo se conecta al bus de manera que cuando uno de ellos no esté transmitiendo, los otros puedan hacerlo, colocándose en un estado de alta impedancia.

Adicional podemos observar que el bus tiene adherida una resistencia de 10K o llamada pull-up conectada a 5 voltios, lo cual implica que el estado en reposo del bus es nivel alto.

1.2.3. Señales del Protocolo 1-Wire

Este sistema maneja varias señales, que en conjunto hacen que la red 1-Wire cumpla su objetivo:

- Pulso de presencia.
- Pulso de reinicio.
- Escritura de nivel lógico ‘0’.
- Escritura de nivel lógico ‘1’.
- Lectura de nivel lógico ‘0’.
- Escritura de nivel lógico ‘1’.

1.2.4. Reinicio del Bus

Se mantiene la señal de datos a 0 voltios durante 480 microsegundos. Se reinician todos los dispositivos conectados al bus. Los dispositivos reiniciados indican su presencia manteniendo la señal de datos a 0 voltios durante los 60 microsegundos.

1.2.5. Envío y Recepción de Datos

La lectura y escritura de datos en el bus 1-Wire se hace por medio de retardos, la generación de estos es responsabilidad del maestro. Cuando el maestro lee información del bus, debe forzar

la línea de datos a un estado bajo durante al menos 1 µs y esperar unos 15 µs para entonces leer el estado de la misma. El estado lógico de la línea en ese momento, estará determinado por el dispositivo esclavo.

Al momento de efectuar la escritura de un bit en el bus ocurre algo similar, el maestro produce un pulso de entre 1µs y 15µs de duración, para luego colocar en el bus el bit que se desea transmitir. Este bit deberá permanecer en el bus al menos 60µs.

Los datos se envían o reciben en grupos de 8 bits. Para iniciar una comunicación se reinicia el bus. El protocolo puede incluir detección de errores transmitiendo códigos de detección de errores (CRC).

Como en el bus puede haber muchos dispositivos, el protocolo incluye el direccionamiento de los mismos empleando un código único de 64 bits de los cuales el byte más significativo indica el tipo de dispositivo, y el último es un código de detección de errores (CRC) de 8 bits. Los comandos que pueden interpretar los dispositivos esclavos dependerán de estos.

Para encontrar los dispositivos presentes en el bus, el maestro puede enviar un comando de enumeración al cual responderán todos los dispositivos.

A continuación se muestran los tiempos que se deben seguir para que exista una comunicación 1-wire.

2.5 V <= VDD <= 5.5 V, TA= -20ºC a 70ºC		
Parámetros	**Mínimo(*us*)**	**Máximo(*us*)**
Espacio de tiempo	60	120
Tiempo de Recuperación	1	
Escribir 0 en bajo	60	120
Escribir 1 en bajo	1	15
Leer dato válido		15
Reinicio en alto	480	
Reinicio en bajo	480	960
Detección de presencia en alto	15	60
Detección de presencia en bajo	60	240

Tabla I: Rangos de tiempos para realizar la comunicación (4).

En el capítulo 2 entraremos en más detalles y explicaremos su funcionamiento y para ello se trabajará con el PIC16f886.

1.3. Micro-controlador

1.3.1. ¿Qué es un Micro-controlador?

Hace unos años, los sistemas de control se implementaban usando exclusivamente lógica de componentes, lo que hacía que fuesen dispositivos de gran tamaño y muy pesados. Para facilitar una velocidad más alta y mejorar la eficiencia de estos dispositivos de control, se trató de reducir su tamaño, apareciendo así los microprocesadores. Siguiendo con el proceso de miniaturización, el siguiente paso consistió en la fabricación de un controlador que integra todos sus componentes en un solo chip.

1.3.2. Características

- Son sistemas cerrados, ya que contiene todos los elementos de un computador en un solo chip.
- Son de propósito específico, es decir, son programados para realizar una única tarea.

Debido a que los micro-controladores sólo incluyen las características específicas para una tarea, su costo es relativamente bajo. Uno típico realiza funciones de manipulación

de instrucciones, posee E/S de accesos fáciles y directos, y un proceso de interrupciones rápido y eficiente. Además también reducen de manera notable los costos de diseño y hay una gran variedad de estos elementos en el mercado.

Dependiendo de la potencia y las características que se necesiten, se pueden elegir micro-controladores de 4, 8, 16 o 32 bits.

1.3.3. Partes que componen a un micro-controlador

- **CPU o procesador.-** Es el cerebro del sistema que procesa todos los datos.
- **Memoria.-** Está formada por una no volátil (ROM, EEPROM, FLASH) donde se almacenan los programas y una volátil (RAM) donde se almacenan los datos.
- **Reloj Principal.-** Normalmente todos los micro-controladores tienen incorporados circuitos osciladores para el funcionamiento de éstos.
- **Puertos E/S.-** Son los que permiten la comunicación con dispositivos externos.
- **Perro guardián o Watchdog.-** Contador que reinicia al micro-controlador para evitar fallos de funcionamiento.

- **Protección ante fallo de alimentación.-** Circuito que reinicia al micro-controlador cuando la tensión de alimentación baja de un cierto límite.
- **Temporizadores.-** Para controlar períodos de tiempo.

Además de convertidores A/D y D/A, comparadores análogos y control de interrupciones.

1.3.4. PIC16F886 (5).

En esta sección nos basaremos en las características principales de micro-controlador usado para nuestro proyecto. El mismo es el 16f886, y se lo muestra en la ***figura 4***. Este fue creado para reemplazar a su antecesor del 16F876A y 16F877A, la mejora más evidente es que el 16f886 posee un oscilador interno.

Vamos a describir las características principales de este micro-controlador:

- Velocidad máxima de hasta 20MHz.
- Posee 28 pines en su total de los cuales 24 de ellos pueden ser usados como entrada/salida independiente con controles individuales uno del otro.

- Tiene además un reloj interno que va desde los 31 kHZ hasta los 8 Mhz sin necesidad de un cristal de cuarzo externo.

- Memoria de programa de 8192 instrucciones, RAM de 368 bytes y EEPROM de datos de 256 bytes.

- Puerto serie SSP para comunicación SPI e I2C estas son marcas registradas por Motorola y Phillips respectivamente.

Entrando un poco más en detalle el I2C, es una interface síncrono de dos líneas ideales para la comunicación entre circuitos integrados como mencionamos anteriormente.

El SPI, es un bus serial estándar síncrono de enlace de datos que opera en modo full-dúplex. Los dispositivos se comunican en modo maestro/esclavo donde el dispositivo maestro inicia la trama de datos. El bus SPI posee cuatro señales lógicas:

- **SCLK:** Reloj serial. Señal proveniente desde el maestro.

- **MOSI, SIMO:** Salida maestro, entrada esclavo. Salida proveniente desde el maestro.

- **MISO, SOMI:** Entrada maestro, salida esclavo. Salida desde el esclavo.

- **SS**: Seleccionar esclavo. Señal activa en bajo, señal dada por el maestro.

Otra de las características que posee el micro-controlador 16f886 es que posee un Transmisor/Receptor asíncrono (USART), las siglas significan *Transmisor Receptor Asíncrono/Síncrono Universal*. El cual puede funcionar de forma síncrona de modo “half dúplex” y de forma asíncrona de forma “full dúplex”.

Podemos también recalcar que posee 2 módulos de captura, comparación y PWM.

Cada módulo CCP, tiene un registro de 16 bits que puede ser usado de 3 formas distintas:

- Como registro de 16 bits para captura de tiempo al producirse un evento.

- Como registro de 16 bits para compararlo con el valor de cuenta del temporizador Timer1, pudiendo provocar un

evento cuando se alcanza el valor contenido en este registro.

- Como registro de 10 bits del ciclo de trabajo de una señal PWM generada por el micro-controlador.

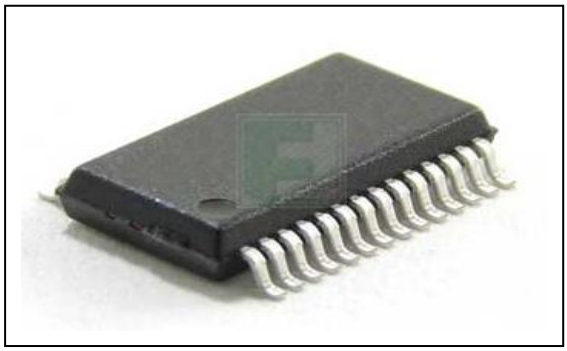


Figura 4: PIC16f886 (5).

1.4. LABVIEW 2010

Es una herramienta gráfica para realizar pruebas, monitoreo y diseño mediante la programación. El lenguaje que usa es G, que simboliza Lenguaje Gráfico.

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos

conocimientos en programación. Además es muy rápido hacer programas con LABVIEW y cualquier persona por muy experimentada que sea se puede beneficiar de éste. Estos programas son también llamados instrumentos virtuales (VIs).

Adicionalmente es una herramienta muy poderosa y versátil en lo que es el uso del ordenador para el monitoreo, evaluación, manejo y automatización de sistemas eléctricos y electrónicos en tiempo real.

National Instruments, creadora de este software a creado a sacado al mercado también tarjetas de adquisición de datos (dependiendo de su uso varían las tarjetas), estas nos ayudan a obtener datos de fuentes externas al ordenador y a su vez comandar circuitería externa por medio del mismo. En nuestro proyecto usamos la tarjeta 6009, la misma que explicaremos más adelante.

LABVIEW consta de dos etapas; la etapa de diagrama de bloques y la etapa de visualización al usuario o panel frontal. La primera es lo que realiza el programador, mientras que la segunda es la parte de ejecución del programa ya implementado, pero de cada etapa antes mencionada se utilizan opciones tanto del diagrama de bloque como del panel frontal.

Para empezar a tratar las funciones y opciones, primero abrimos el programa y creamos un proyecto en *Blank VI,* como se muestra en la ***figura 5***, en nuestro caso lo llamaremos ***proyecto.vi***, donde se nos abren dos ventanas como en la ***figura 6.***

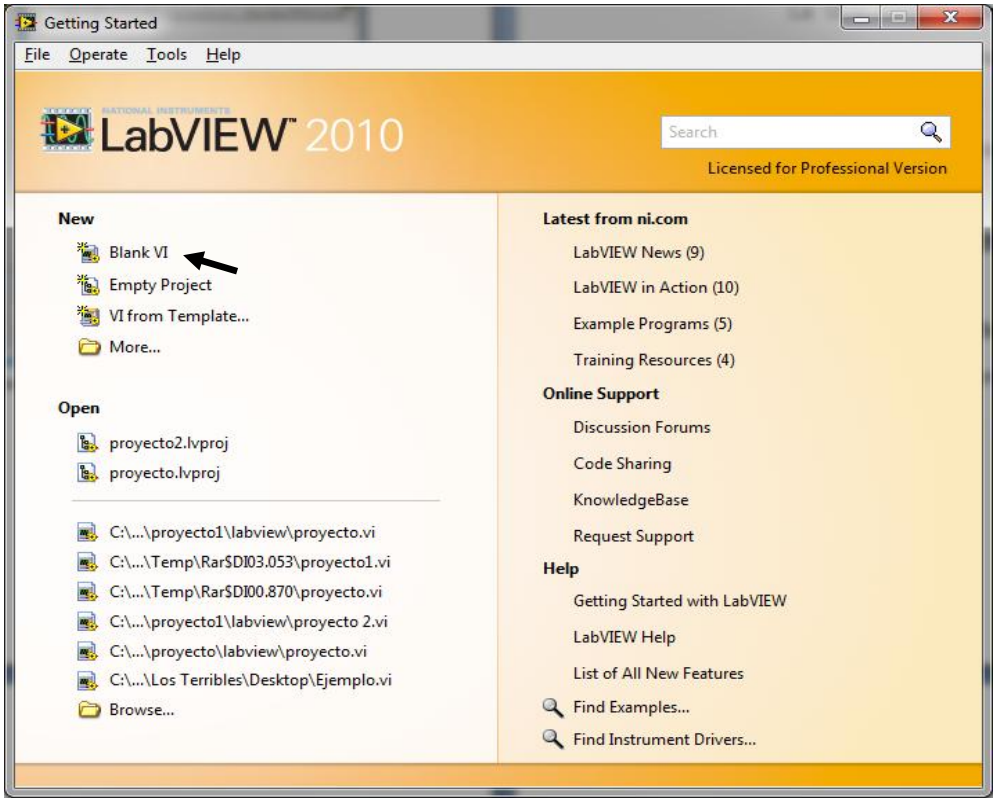


Figura 5: Labview 2010

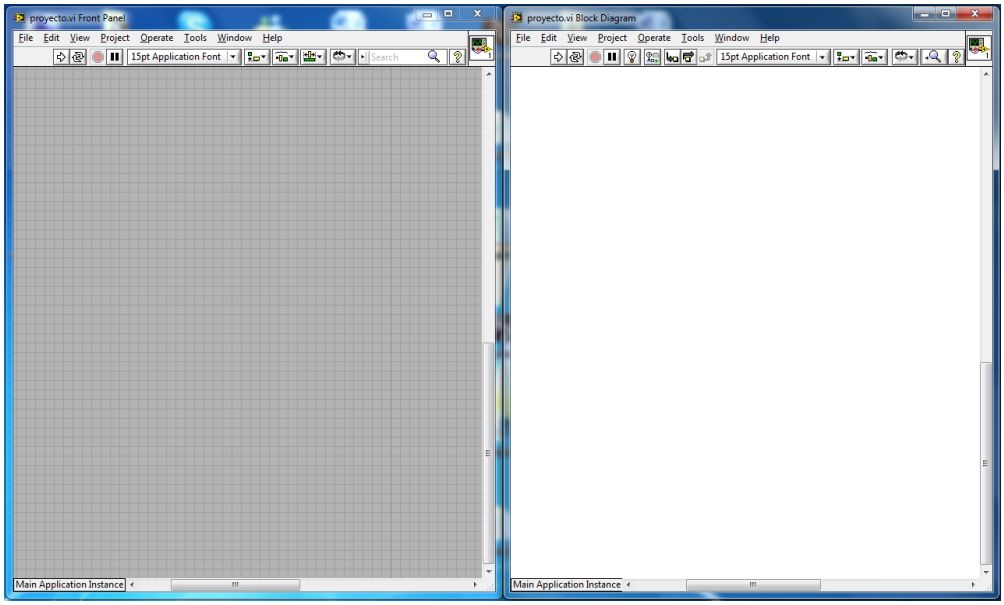


Figura 6: Proyecto con su panel frontal y diagrama de bloques.

En la ***figura 6,*** se pueden visualizar sus dos etapas; el panel frontal y el diagrama de bloques. Cada una de ellas tienen funciones y opciones diferentes, pero que en conjunto se pueden ir armando programas amigables, desde una simple suma hasta un sistema de automatización, que comprende: adquisición, monitoreo y control.

Empezaremos a revisar las funciones y opciones, que podemos utilizar, especialmente las que comprenden este proyecto. Y las trataremos por separado.

1.4.1. Panel Frontal

En este bloque, lo más importante es presentarle al usuario indicadores, controladores y switch amigables, también leds y gráficos de una manera sencilla de entender. Todas estas operaciones las sacamos haciendo clic derecho dentro del panel frontal, donde escogemos de acuerdo a nuestras necesidades.

1.4.1.1. Indicadores

Estos nos ayudan a visualizar cualquier tipo de medida sean estos; metros, grados, porcentajes, velocidad, tamaño, etc. Adicional a indicadores números, también tenemos indicadores en forma de texto.

Dentro de los indicadores números tenemos los siguientes:

- *Numeric.* [1]
- *Meter.* [2]
- *Thermometer.* [3]
- *Gauge.* [4]
- *Tank.* [5]

Los cuales son mostrados en la ***figura 7.*** En la ***figura 8*** podemos ver cómo obtenerlos, los cuales están dentro de *Express Controls*.

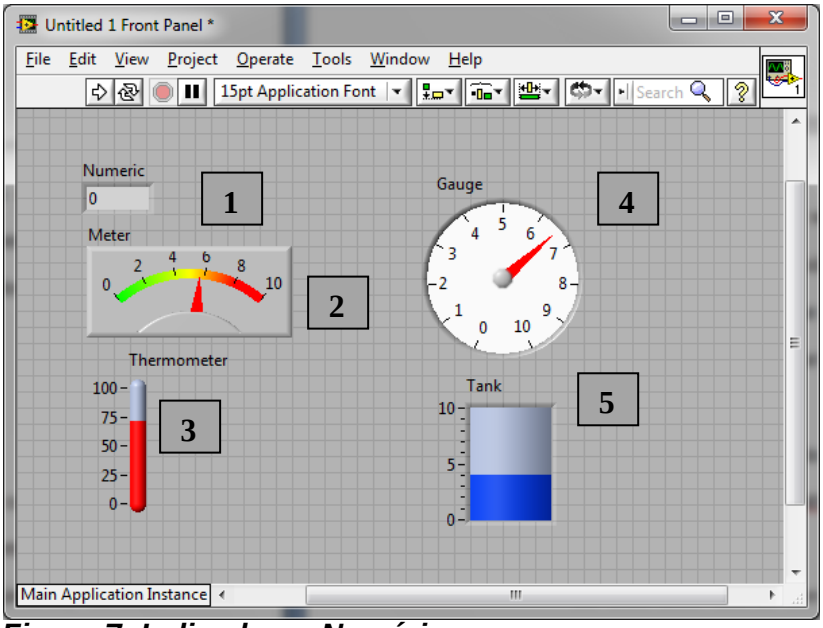


Figura 7: Indicadores Numéricos

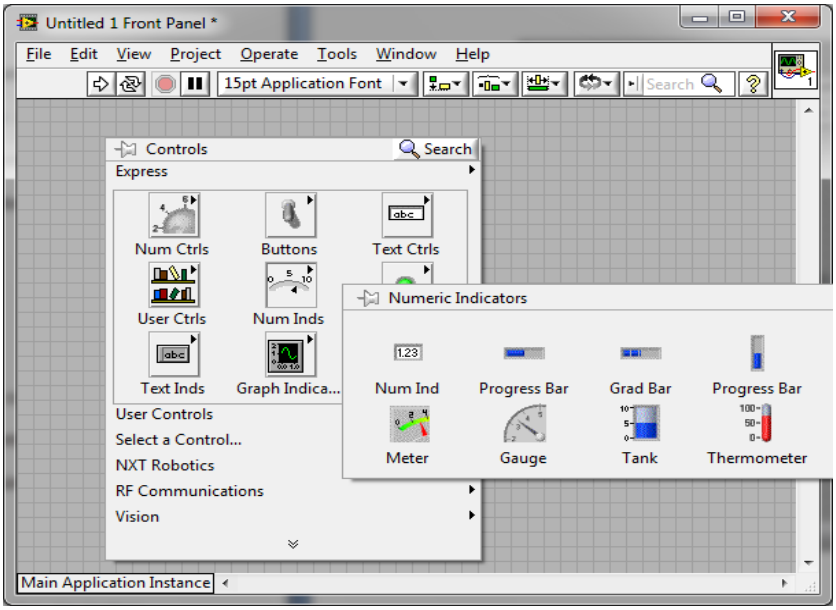


Figura 8: Obtención de Indicadores Numéricos.

De la misma manera obtenemos el indicador de texto. Se lo mostramos en la ***figura 9.***

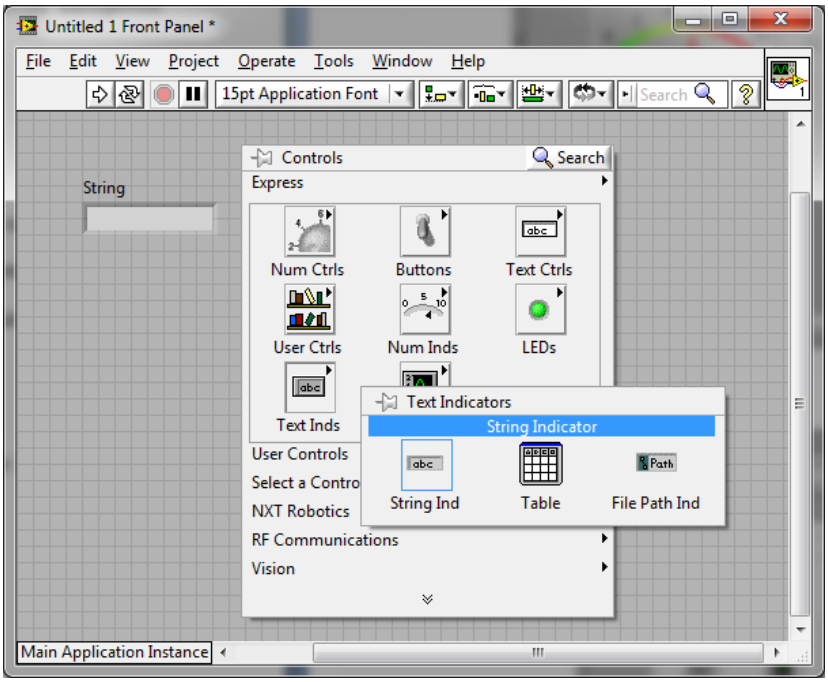


Figura 9: Indicador de Texto.

1.4.1.2. Controles y Switch

Estos nos ayudan a activar o desactivar señales, de acuerdo a la programación o las necesidades del usuario. De la misma manera tenemos numéricos, de texto y en forma de botones. Los botones son de tipo booleanos. A continuación le enumeramos las más utilizadas y especialmente las que utilizamos en este proyecto:

- *Numeric.* [1]
- *Slide.* [2]
- *Dial.* [3]
- *Boolean.* [4, 5, 6]

En la ***figura 10*** mostramos la parte simulada de las opciones antes mencionas. Así mismo, en la ***figura 11*** se indica cómo sacarlos.

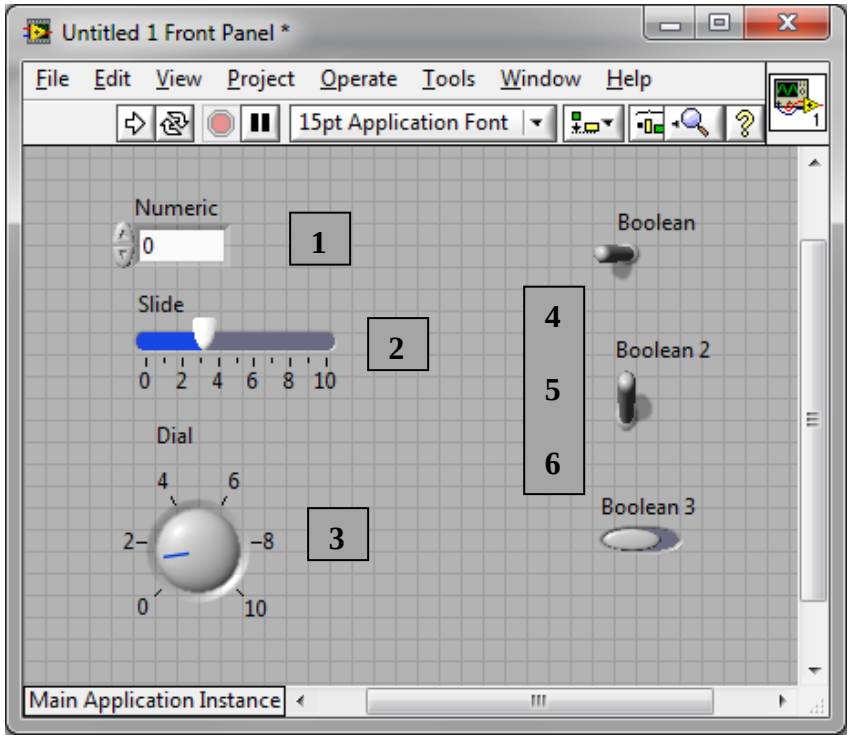


Figura 10: Controles y Botones.

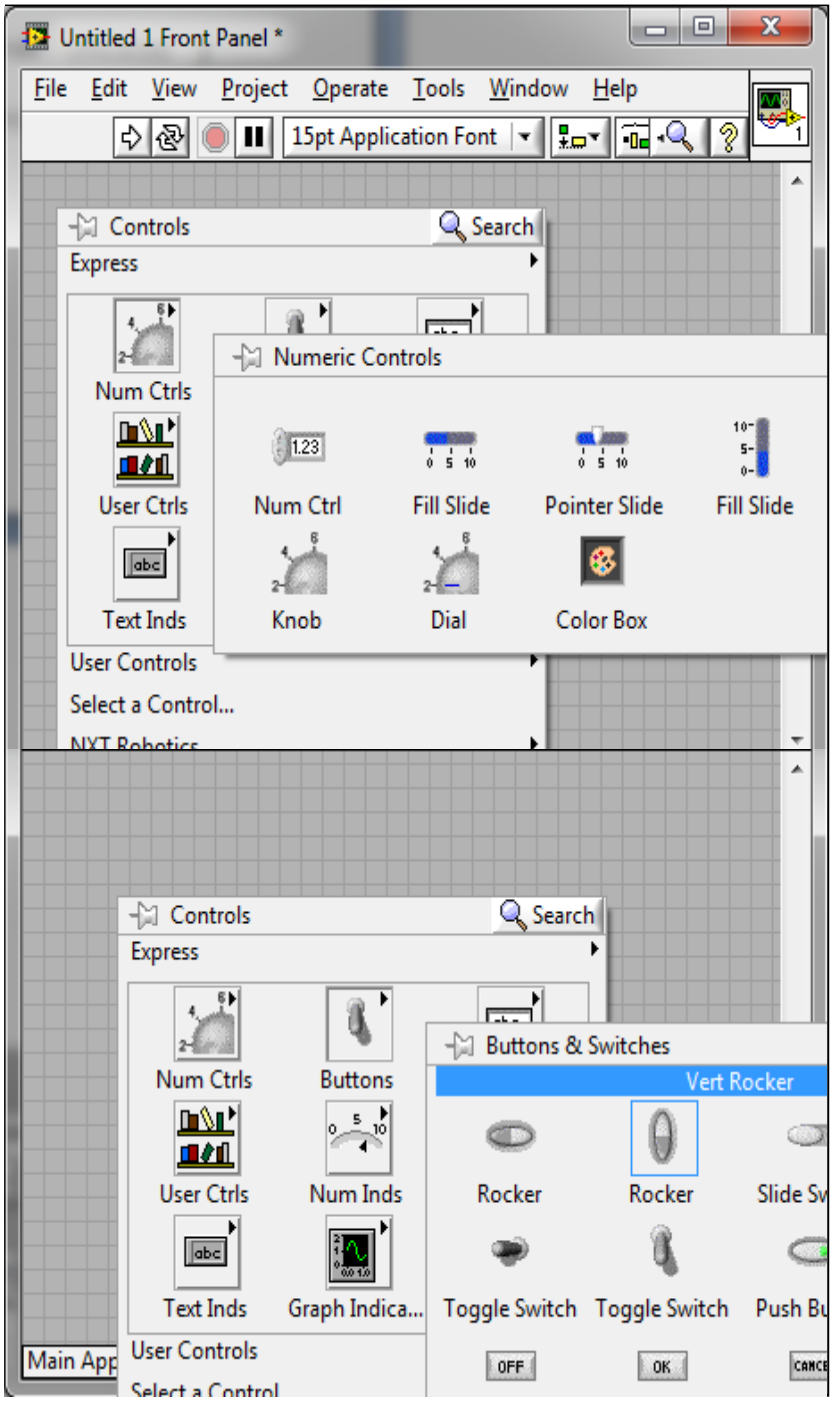


Figura 11: Obtención de Controles y Botones.

1.4.1.3. Gráficos y Leds

Este recurso nos permite graficar cualquier tipo de onda, o señal. Puede ser constante o variable. Los leds nos permiten visualizar el cambio de una señal, en dos tiempos.

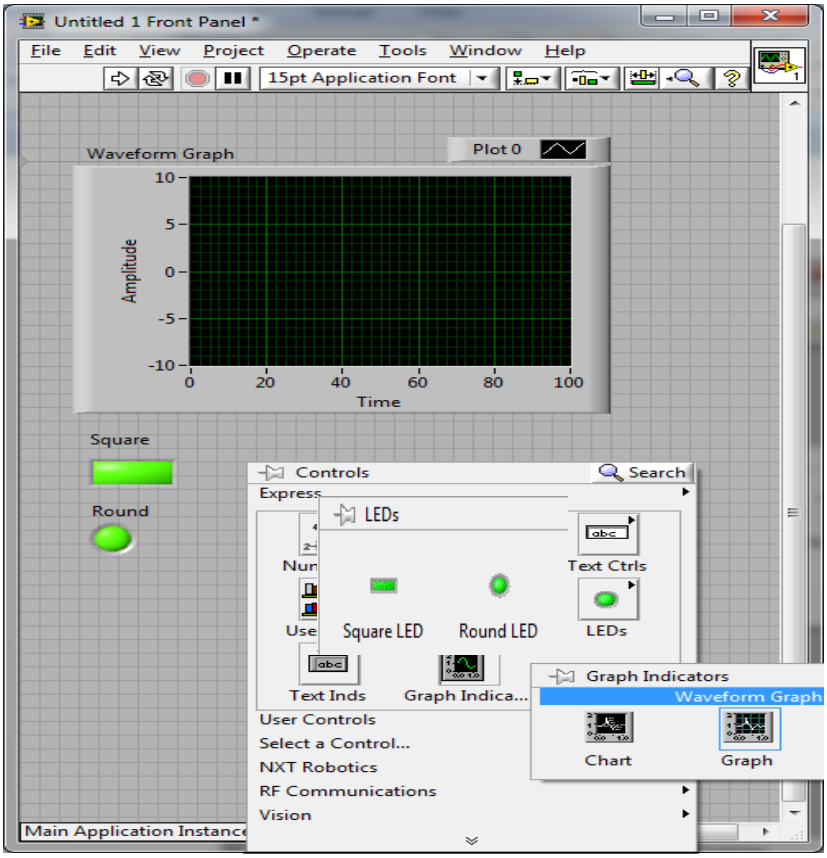


Figura 12: Gráfico y Leds.

En la ***figura 12*** podemos observar la forma de presentar el gráfico y los leds, además la rutina para poderlos obtener.

Ahora procederemos a explicar la parte del diagrama de bloque. La misma que es un poco más compleja de desarrollar, debido a que hay que ser muy minucioso con las señales al momento de tratar de unir, comparar o hacer cualquier proceso entre dos o más señales. Ya que no pueden ser del mismo tipo de dato, por lo que hay que hacer conversiones antes. Esto se lo detallará en los siguientes ítems.

1.4.2. Diagrama de Bloque

En esta parte del software, existen muchas aplicaciones, funciones y opciones. Tales como: de programación, de matemáticas, de procesamiento de señales, de conectividad. Pero la mayoría de estas las encontramos de manera resumida en ***Functions Express.*** Por tal motivo vamos a tratar, las específicas para este trabajo.

1.4.2.1. DAQ Assist

Dentro del software se encuentra la función *DAQ Assist*, que trabaja en conjunto con la tarjeta de adquisición NI USB-6009 (se explica más adelante). Esta opción sirve como entrada o salida al programa, específicamente datos reales. La misma se encuentra

siguiendo estos pasos; En la parte de Diagrama de Bloque hacemos clic derecho, donde nos dirigimos en *Input,* en el caso de entrada y en *Output* en el caso de salida de *Funtions Express,* ahí encontramos la *DAQ Assist.* El ícono y como llegamos a esta opción se la mostramos en la ***figura 13.*** A continuación se debe configurar la tarjeta según nuestras necesidades.

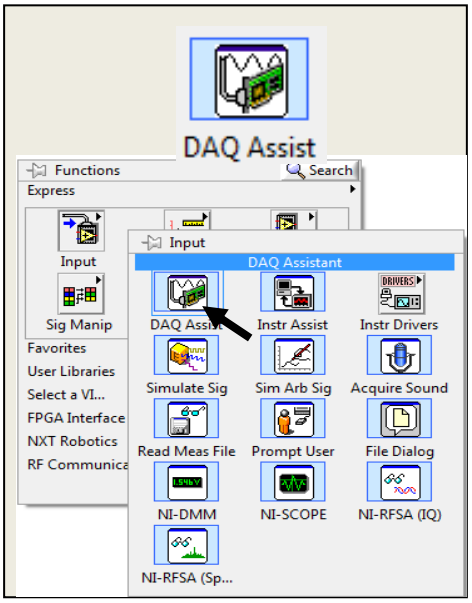


Figura 13: Icono de la DAQ ASSIST

La entrada o salida a definir la decimos, si la queremos de forma análoga o de forma digital.

Podemos configurar de forma análoga valores como: En la ***Figura 14*** mostramos los datos posibles:

- Voltaje
- Corriente
- Temperatura
- Sonido
- Frecuencia. Etc.

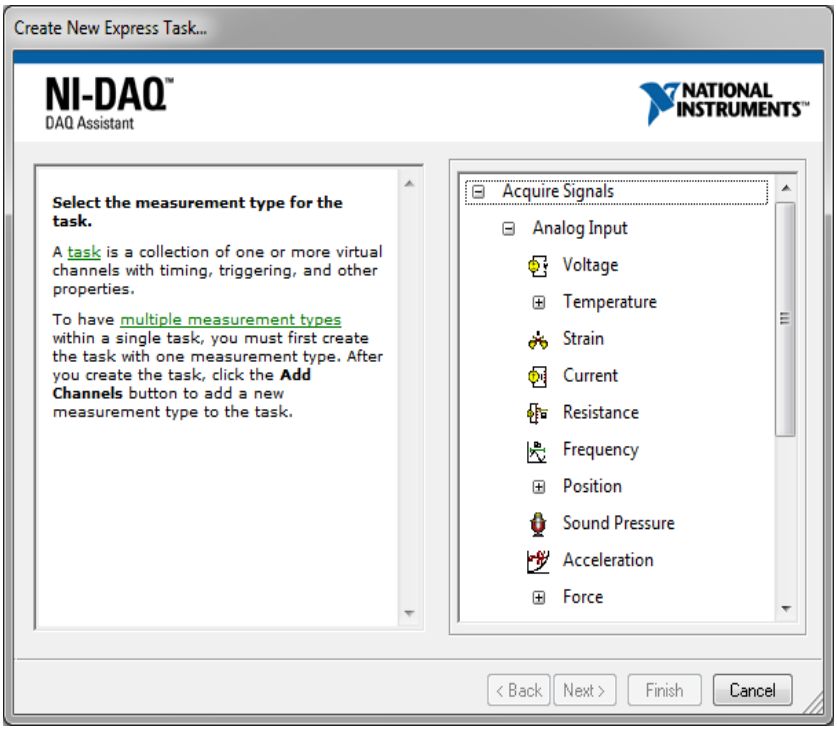


Figura 14: Configuración DAQ Assistant.

Ahora lo que corresponde a la parte digital, se puede configurar como línea como en la ***figura 15*** o por puerto como en la ***figura 16***, según nuestra petición.

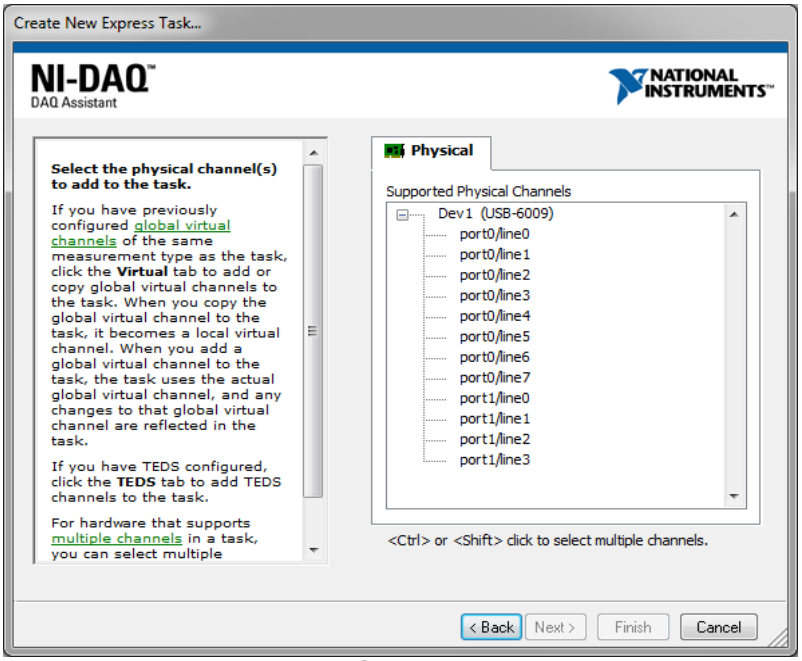


Figura 15: Canales por líneas

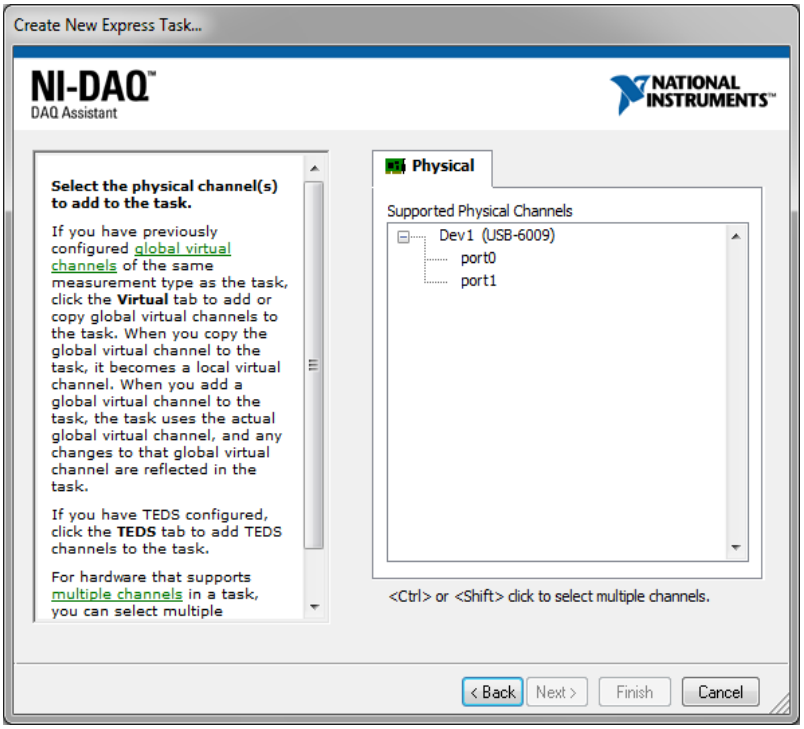


Figura 16: Canales por puertos.

1.4.2.2. Fórmula

Esto nos ayuda a evitar sumas, restas, multiplicación y algunas operaciones más. Debido a que con un solo llamado, podemos resolver procedimientos como si fuera una calculadora, con diferentes señales, pero siendo el mismo tipo de dato, como se muestra en la ***figura 17.***

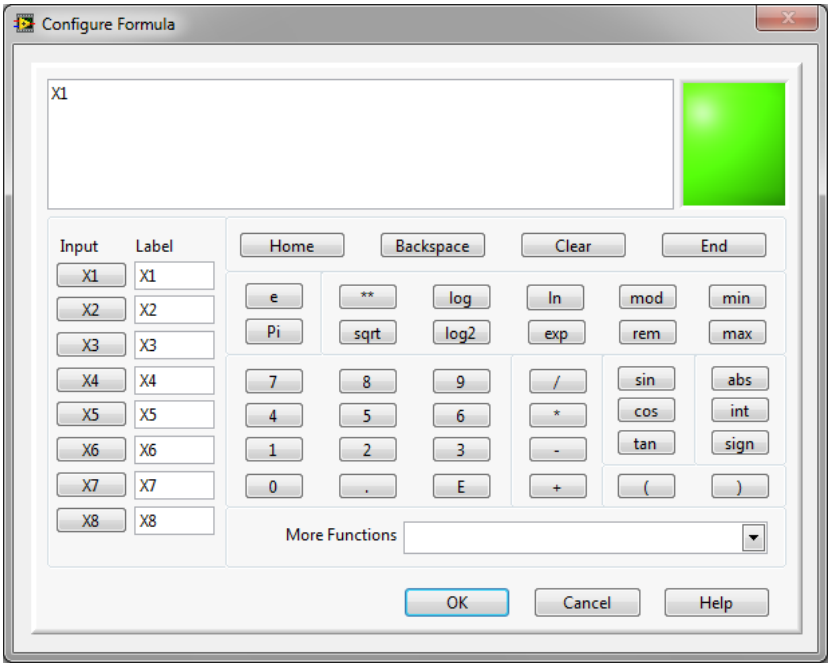


Figura 17: Función Fórmula.

1.4.2.3. Comparaciones y Condicionales

Nos permite controlar, y realizar operaciones lógicas, entre señales o valores constantes dependiendo de nuestro criterio. Aquí encontramos funciones como:

- Mayor que.
- Menor que.
- Igual que, o Igual a 0.
- Seleccionador de señal.

El seleccionar de señal, nos permite el paso de una señal, dependiendo del selector. Estas y algunas opciones más se las mostramos a continuación.

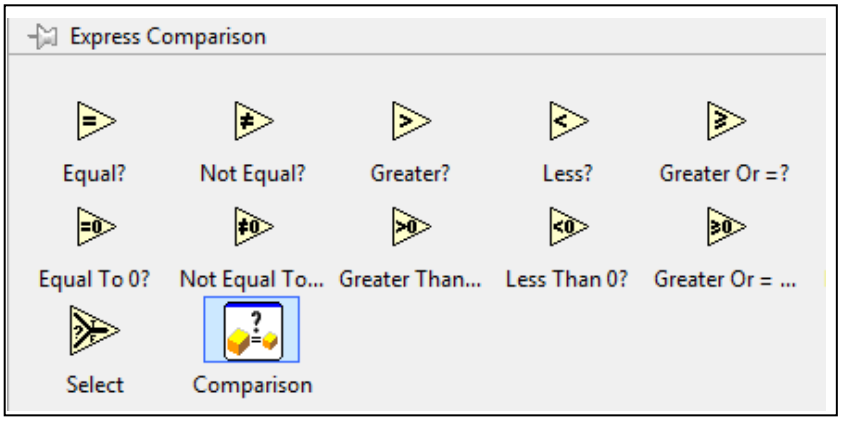


Figura 18: Comparadores.

Dentro de los condicionales tenemos puertas lógicas como las siguientes: La mostramos en la ***Figura 19.***

- And y Not And.
- Or y Not Or .
- Not.

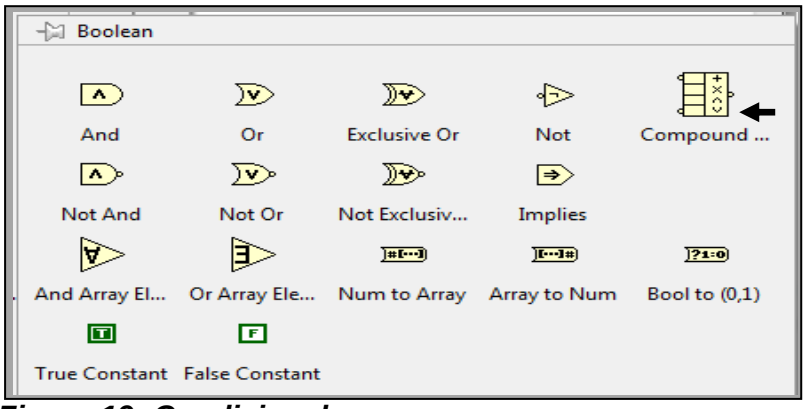


Figura 19: Condicionales.

Adicionalmente tenemos una opción que utilizamos en nuestro proyecto, que es la que está marcada con la flecha, llamada *Compound Arithmetic*. Esta nos ayuda a realizar la operación lógica de más de dos señales a la vez.

1.4.2.4. Lazos

Nos permiten realizar operaciones repetitivamente, tantos infinitos como finitos. Los infinitos duran mientras está en ejecución el programa, y el finito

solo funciona si cumple una condición planteada. Entre los más usados tenemos:

- While Loop.
- Flat Sequence.
- Case Structure.

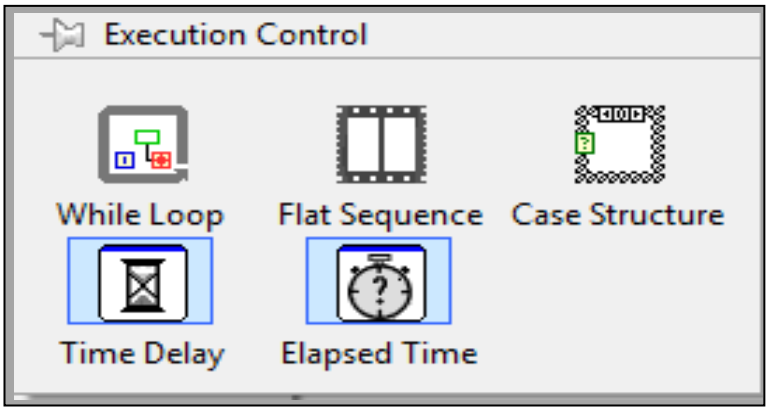


Figura 20: Lazos

1.4.2.5. Funciones Especiales (7)

Vamos a explicar algunas de estas funciones, por motivo que son utilizadas en este trabajo. Estas son:

Format Date/Time String Function.- Nos permite configurar la fecha y la hora actual obtenida de la computadora, siguiendo una secuencia propia de esta función, como se indica en la ***Figura 21.***

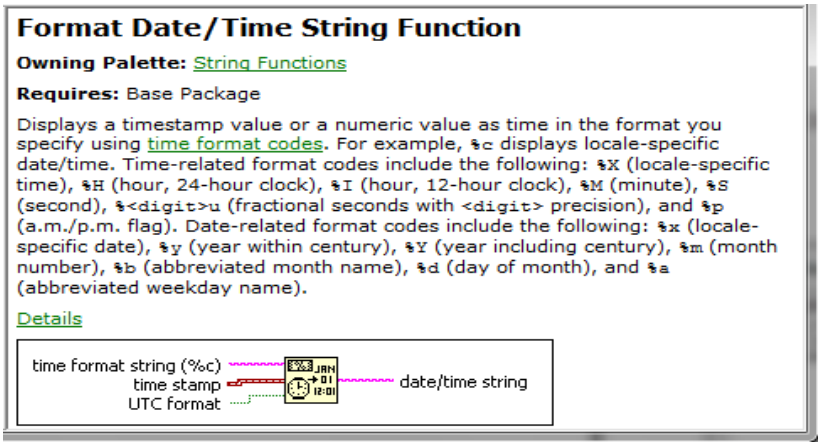


Figura 21: Format Date/Time String Function.

Build Path Function.- Dicha función concatena señales y construye una nueva. Su ícono se muestra en la siguiente figura.

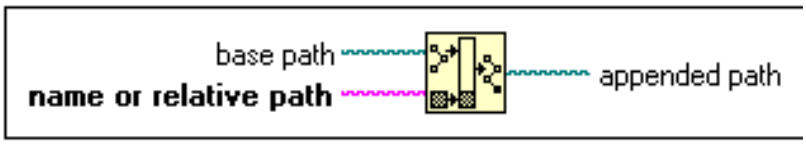


Figura 23: Build Path Function.

Write to text File Function.- Nos permite escribir texto dentro de un archivo.

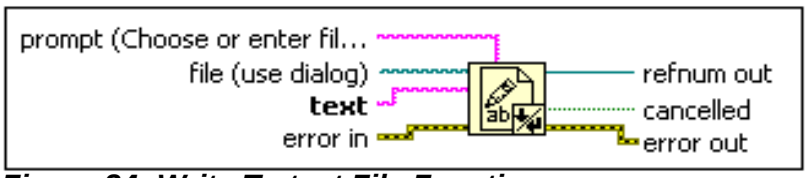


Figura 24: Write To text File Function.

Write To Spreadsheet File.- Con esta función, unimos las tres funciones especiales antes mencionadas, y nos ayudan

a crear archivos, estos pueden ser de cualquier programa que realicemos y lo ponemos como reporte. Todas las entradas son indicadores de texto, que a gusto del usuario los puede ir llenando. Puede ser de cómo quiere llamar al archivo, qué requiere como encabezado, y los datos a llenar.

Build Array.- Nos permite unir todo tipo de dato para luego agregarlo a la función *Write To Spreadsheet File,* pero hay que tener en cuenta que los datos sean del mismo tipo. En la ***figura 25*** se les muestra el ícono.

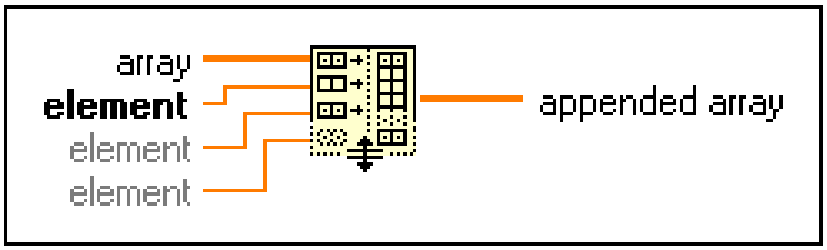


Figura 25: Build Array

Adicionalmente se explica una función de cómo enviar correos electrónicos. Esto nos facilitaría en caso de no encontrarnos en el área que estemos monitoreando, poder saber lo que está sucediendo con el sistema desarrollado. Esta aplicación se llama *SMTP Email Send File VI.* Este funciona como cualquier servidor de correo, que tiene su

destinatario, su remitente, su asunto, también se le puede adjuntar archivos. Lo que se necesita es tener contratada una cuenta de servidor de correo. En este caso las entradas son configuradas con controles de texto, para así poder definir todos los parámetros antes mencionados.

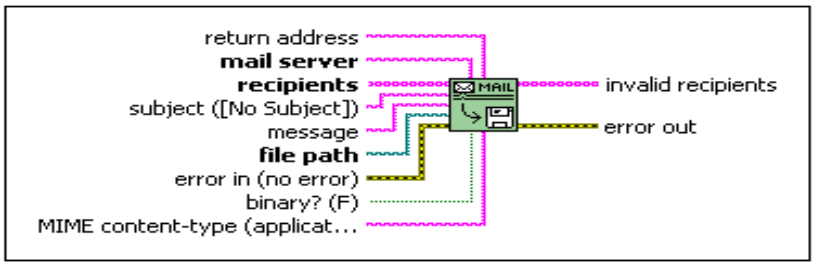


Figura 26: SMTP Email Send File VI.

1.4.3. Tarjeta de Adquisición de Datos NI USB-6009

En esta sección veremos la tarjeta de adquisición que usamos para hacer la comunicación entre nuestros sensores y el ordenador, y a su vez es usada para que la computadora controle las salidas de alarmas y de control para estabilizar la temperatura y humedad.

Esta tarjeta es producida por *National Instruments*, la misma empresa creadora del programa LabView.

La tarjeta NI USB-6009 cuenta con 8 entradas análogas, 2 salidas análogas, 12 entradas/salidas digitales programables, un contador ascendente de 32 bit, se conecta al ordenador por vía USB para transmitir datos y alimentación de voltaje.

1.4.3.1. Características de las Entradas y Salidas

EA (Entrada Analógica).- Estos terminales se pueden usar como entradas individuales de voltaje analógico o como combinaciones para entradas de voltaje diferencial como nos muestra la tabla. Tenemos 8 entradas terminales individuales, las cuales pueden ser usadas de dos formas:

- **Como terminales simples**, en este caso se usa como referencia las salidas GND de la 6009, el voltaje máximo para esta configuración es de ±10V.

- **Como terminales de voltaje diferencial**, se usan dos terminales simples. En la ***Tabla 2*** se especifica cuáles son las parejas para usar la configuración como voltaje diferencial, lo valores de voltajes que se pueden utilizar para este tipo de configuración son ±20 V, ±10 V, ±5 V, ±4 V, ±2.5 V, ±2 V, ±1.25 V, ±1 V.

SA (Salidas Analógicas).- Son las dos únicas salidas analógicas que posee la tarjeta de adquisición de datos, las cuales no son individuales y se usan junto con la tierra. El rango de voltaje es de 0 a +5V con una impedancia de salida de 50Ω y una corriente de 5mA.

GND (Tierra).- Es la referencia para todas las entradas y salidas analógicas y la referencia para los voltajes diferenciales, así como de todos los voltajes que da la tarjeta 6009.

EP (Entradas Digitales Programables).- Son todas las entradas y salidas digitales, estas se pueden programar para que funcionen como entrada o salida, individualmente sin depender una de la otra. En total la tarjeta posee 12 entradas/salidas digitales, todas ellas compatibles con tecnología TTL, LVTTL, CMOS. Además posee un rango de voltaje con respecto al terminal GND desde -0.5V hasta 5.8V.

Para nuestros terminales digitales tenemos cuatro opciones posibles de configuración.

En el caso de las salidas:

- Configurarlo como colector abierto.

- Configurarlo como unidad de salida digital de conducción activa.

En caso de las entradas:

- Configurarlo como receptor de señal de voltaje por medio de un switch.
- Configurarlo como receptor de señal TTL.

EPC (Entrada Programable Contador).- Esta entrada podemos programarla para que se use como una entrada para un contador de eventos, o como disparador digital.

+5V.- Voltaje entregado por la tarjeta de adquisición de datos para usarlo con fuente de poder.

+2.5V.- Voltaje entregado por la tarjeta, es usado para pruebas de envolventes.

Para nuestro proyecto usamos 8 entradas digitales configuradas como entradas digitales para recibir una señal TTL debido a que el microprocesador envía señales de máximo +5V en paralelo, en un intervalo de tiempo en us. También usamos 2 salidas digitales para activar las alarmas y el sistema de enfriamiento. A continuación se muestra la tarjeta NI USB-6009 físicamente, la ***tabla II,*** donde

mostramos los terminales analógicos con sus respectivas señales y la ***tabla III*** las terminales digitales.

Figura 27: DAQ NI USB-6009. [8]

Número de terminal	Señal (modo terminal simple)	Señal (modo diferencial)
1	GND	GND
2	EA0	EA0+
3	EA4	EA0-
4	GND	GND
5	EA1	EA1+
6	EA5	EA1-
7	GND	GND
8	EA2	EA2+
9	EA6	EA2-
10	GND	GND
11	EA3	EA3+
12	EA7	EA3-
13	GND	GND
14	SA0	SA0
15	SA1	SA1
16	GND	GND

Tabla II: Terminales analógicas de la DAQ NI USB-6009 (6).

Número de terminal	Señal (modo terminal simple)
17	EP00
18	EP01
19	EP02
20	EP03
21	EP04
22	EP05
23	EP06
24	EP07
25	EP10
26	EP11
27	EP12
28	EP13
29	EPC0
30	+2.5V
31	+5V
32	GND

Tabla III: Terminales digitales de la DAQ NI USB-6009 (6).

CAPITULO 2

2. Programación del PIC

En este trabajo investigativo como primera parte programaremos el PIC 16f886, el mismo que servirá como máster en relación al sensor de temperatura que es el esclavo. Se realiza de esta manera ya que así se transmite mediante la tecnología 1-Wire, para ello nos basamos en rutinas, cada una de las mismas tiene su tiempo de duración en *us*, como se mencionó en el capítulo anterior.

Estas rutinas se las van a programar en código c, para esto utilizaremos el programa Micro-Basic de micro-controladores, con el cual simulamos el código y a su vez programamos el PIC16f886.

2.1. Rutinas 1-wire

Como ya sabemos para que se cumplan los requerimientos, para obtener una buena comunicación 1-Wire, debemos seguir un conjunto de rutinas. Las cuales son propias del integrado DS2438 para que trabaje con 1-wire. El protocolo a seguir de forma general es el siguiente:

- Inicialización.
- Funciones de Comandos de la ROM.
- Funciones de Comandos de la Memoria.
- Transmisión y Recepción de Datos.

En la ***tabla IV*** se muestra la secuencia a seguir.

Modo del Máster	Dato (Rutina)
Tx	Reset
Rx	Presencia
Tx	CCh
Tx	4Eh
Tx	Reinicio
Rx	Presencia
Tx	CCh
Tx	44h
Tx	Reinicio
Rx	Presencia
Tx	CCh
Tx	B8h
Tx	Reinicio
Rx	Presencia
Tx	BEh
Rx	Datos en Bytes

Tabla IV: Secuencia de Rutinas (4).

Una vez que ya tenemos la secuencia, procedemos a explicar brevemente cada una de las rutinas:

2.1.1. Rutina de Reinicio

Para iniciar alguna transmisión 1-Wire, la iniciamos con un pulso de reinicio enviado desde el dispositivo maestro, en nuestro

caso desde el micro-controlador, seguido de un pulso de presencia que envía el dispositivo esclavo, el cual es el sensor DS2438.

2.1.2. Rutina Salto de Memoria (CCh)

Este comando le permite al maestro acceder a las funciones de memoria de los dispositivos esclavos, y así hacerlos transmitir simultáneamente sin causar colisión entre ellos. De esta manera el sistema ahorra tiempo.

2.1.3. Rutina de Escribir (4Eh)

El dispositivo maestro produce un pulso que dura entre 1 y 15 us, como lo indica la ***tabla I,*** dentro del cual realiza la escritura de 8 bits en la memoria del dispositivo esclavo, el cual lo coloca en el bus de comunicación, y el proceso termina cuando el maestro envía un nuevo reinicio.

2.1.4. Rutina de Conversión a Temperatura (44h)

Con el siguiente comando se inicia la conversión a temperatura. El mismo configura una bandera del registro de memoria, colocándolo en ‘1’ durante la conversión y cuando ya se

ha realizado el proceso la bandera se limpia y pasa a ‘0’. Si el dispositivo maestro tiene problemas con los intervalos de tiempos, envía un ‘0’ a la salida del bus de comunicación, hasta cuando se ha completado correctamente la conversión.

2.1.5. Rutina de Recuperación de Datos (B8h)

Este comando recupera los datos almacenados en la EEPROM del dispositivo DS2438, para que puedan ser leídos por el dispositivo maestro en el siguiente proceso.

2.1.6. Rutina de Lectura (BEh)

Este nos ayuda a leer los datos recuperados anteriormente, siempre comenzando en la dirección 0, el maestro puede leer hasta el final los datos y también la detección de errores (CRC) de los datos. O si el maestro no va a leer todos los datos puede emitir un reinicio para terminar la lectura en cualquier momento.

Una vez ya teniendo claro los procesos, se procede a la creación del código fuente, para luego programarlo en el PIC16F886. Para hacerlo vamos a crear un proyecto en Micro-Basic.

2.2. Creación de Proyecto

Creamos un proyecto para realizar el código, en nuestro caso se llama *One-Wire*, en el cual configuramos qué micro-controlador se va a utilizar y las configuraciones respectivas del mismo, como el oscilador, el perro guardián etc. Como se lo muestra en la ***figura 28.***

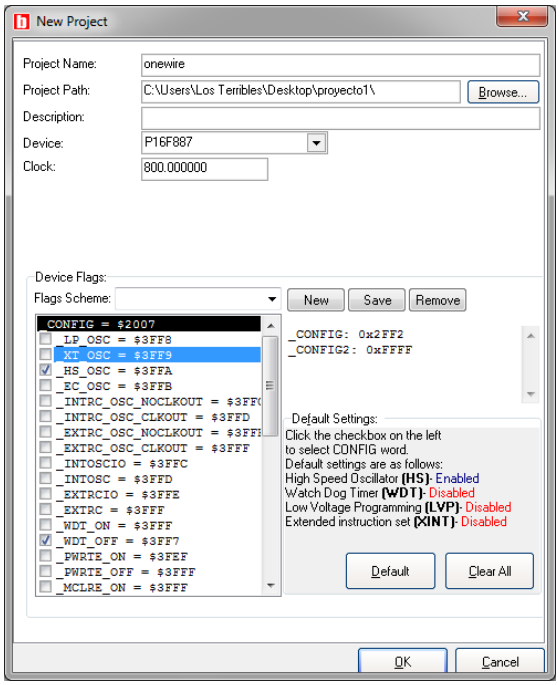


Figura 28: Nuevo proyecto en Micro Basic

Luego se nos presenta la ventana donde procedemos a escribir el código. La ventaja de utilizar este compilador es que posee la librería que incluye el proceso de comunicación 1-wire, para ello se usarán los comandos de la misma, siguiendo la secuencia con las rutinas explicadas con anterioridad, para así garantizar una buena transmisión. En la ***figura 29*** se muestra la ventana donde se escribirá el código fuente.

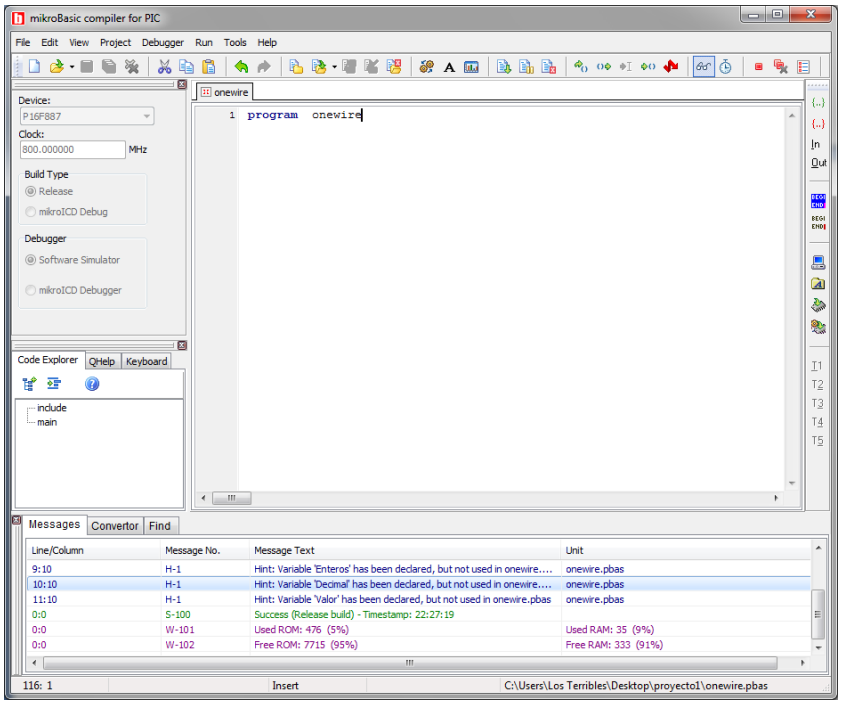


Figura 29: Escribir código fuente

2.2.1. Creación del Código Fuente

Cuando se programa un micro-controlador, se configuran ciertos parámetros propios. Pero lo que nos interesa es configurar qué puerto queremos como salida y cual como entrada. En nuestro trabajo necesitamos que un pin nos trabaje como entrada, el cual es por donde se comunica con el sensor de temperatura y será el puerto a0. Y a su vez configuramos el puerto B como salidas, en este caso utilizaremos todos los pines, porque vamos a tener 8 salidas, las mismas que serán configuradas como salidas digitales. A continuación se muestra la parte del código antes mencionado.

```
sub procedure conf_16f886

OSCCON = 0x75          ' configuración oscilador interno 8mhz

OPTION_REG = 0X80   ' desactiva el pull up en portb

TRISA = 0x01     ' Puerto A, primer pin  como entrada 1-wire

PORTA = 0x00

TRISB = 0x00          ' Todo puerto B como salida

PORTB = 0X00


TRISC = 0X01

PORTC = 0X00


ANSEL = 0X00          ' salidas digitales

ANSELH = 0X00         ' digitales

end sub
```

Externamente, el pin de entrada debe conectarse a una fuente de alimentación a través de una resistencia pullup (4,7K), para que se realice la comunicación 1-wire correctamente con el dispositivo esclavo.

Adicionalmente configuramos una variable que utilizaremos, el cual es *Dato[],* con su respectivo índice *(i),* el cual es un byte

donde se almacena toda la información durante el proceso 1-wire. Del cual cogeremos el dato de temperatura de las 8 posibles opciones; como corriente, voltaje, etc., el mismo se lo puede revisar con más detalle en (4).

La temperatura la tenemos en binario, y su transformación a decimal es el valor real de la temperatura. En la siguiente tabla se mostrara lo antes expuesto.

BINARIO	TEMPERATURA (Decimal)
00011110	+30 ºC
00011001	+25 ºC
00000000	0 ºC
10011110	-25 ºC
10011001	-30 ºC

Tabla V: Datos en Binario y Decimal (Temperatura Real) (4).

Como se podrá notar en la ***tabla V*** que tiene signos positivos y negativos, esto es gracias al último bit de la palabra en binario, el cual da el signo. Si es 0 entonces es positiva y si es 1 es negativa la temperatura.

Continuando con el código, la siguiente parte, son las líneas de las rutinas del proceso 1-wire, la misma que es una función y la llamamos *sub function get_temperatura().*

```
sub function get_temperatura() as byte
    dim Dato as byte[9]
    dim i as byte

    Delay_ms(1)

    ow_reset(PORTA,0)           'Reinicio
    Ow_Write(PORTA,0,0xCC)      ' Salto de memoria
    Ow_Write(PORTA,0,0x4E)      'Escritura

    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC)
    Ow_Write(PORTA,0,0x44)      ' Conversión a temperatura
    Delay_ms(1)

    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC)
    Ow_Write(PORTA,0,0xB8)      ' Recuperar Dato
    Ow_Write(PORTA,0,0x00)
    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC)
    Ow_Write(PORTA,0,0xBE)
    Ow_Write(PORTA,0,0x00)
```

```
for i=0 to 8
    Dato[i] = Ow_Read(PORTA,0)        'Lectura de Datos
next i
```

Ya obtenidos los datos, procedemos a elegir solo el de temperatura, el cual está en la posición 2, respecto a la ***tabla VI.***

Byte	Contenido
0	Estatus-Configuración
1	Temperatura LSB
2	Temperatura MSB
3	Voltaje LSB
4	Voltaje MSB
5	Corriente LSB
6	Corriente MSB
7	Umbral

Tabla VI: Datos que se almacenan en DS2438 (4).

Adicionalmente, como vimos en la ***tabla 5,*** dependiendo del último bit, la temperatura es positiva o negativa. Esta representación la tenemos en las siguientes líneas de código.

```
if TestBit(Dato[2],7) then
    Dato[2]= Dato[2] xor 0xff
    setbit(Dato[2],7)
end if
```

Para terminar el código, los datos obtenidos en la función *get_temperatura()* se los envía por el puerto B del micro-controlador, el cual fue configurado en un principio como salida. Y como es un dato que lo requerimos constantemente, lo colocamos dentro de un lazo while. Como se muestra a continuación.

```
main:
    conf_16f886()

    while 1
        PORTB = get_temperatura()
        delay_ms(200)
    wend
end
```

Una vez ya terminado el código, se procede a compilarlo y a programar el micro-controlador.

CAPITULO 3

3. Creación del Programa en Labview

En este capítulo vamos a crear la interfaz entre el sistema de adquisición y control de datos y el usuario. Para llegar a dicha comunicación, se ha realizado una programación gráfica donde se explica cada etapa de los procesos y para ello lo dividiremos en bloques con funciones específicas. Para esto utilizaremos todas las herramientas u opciones del programa LABVIEW.

Cabe acotar que todos los comandos y funciones que se utilicen en este capítulo fueron explicados detalladamente en la sección 1. Por lo que unicamente lo nombraremos.

3.1. Diagrama de Bloque General

Aquí presentamos de forma general como se va a ir armando el programa de acuerdo a las necesidades y las opciones que nos presenta

el software. A continuación mostramos en la ***figura 30*** el diagrama de bloque antes mencionado y por consiguiente se va a ir tratando cada etapa.

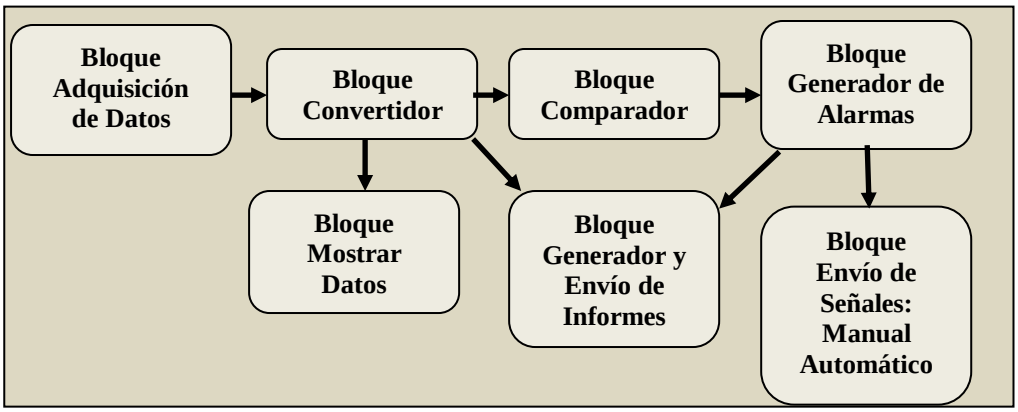


Figura 30: Diagrama de Bloque General

3.1.1. Bloque adquisición de datos

Después de la realización de la parte electrónica y programación del PIC, utilizamos la tarjeta de adquisición de datos de *National Instruments USB-6009* para obtener valores de la parte exterior, para así llevar un registro y poder controlar dichos valores. Logramos esto conectando la parte electrónica a los puertos de entradas y salidas de la tarjeta NI USB-6009. En

nuestro caso nos ayudará a obtener datos de temperatura y humedad de un área específica.

Dentro del software utilizamos la opción *DAQ Assist.* A continuación configuramos la tarjeta, para que funcione como entrada de datos con los pines que se van a utilizar, en nuestro caso utilizaremos entradas digitales para los datos de temperatura y para humedad una entrada analógica. En la ***figura 31*** se muestra la configuración de los puertos seleccionados.

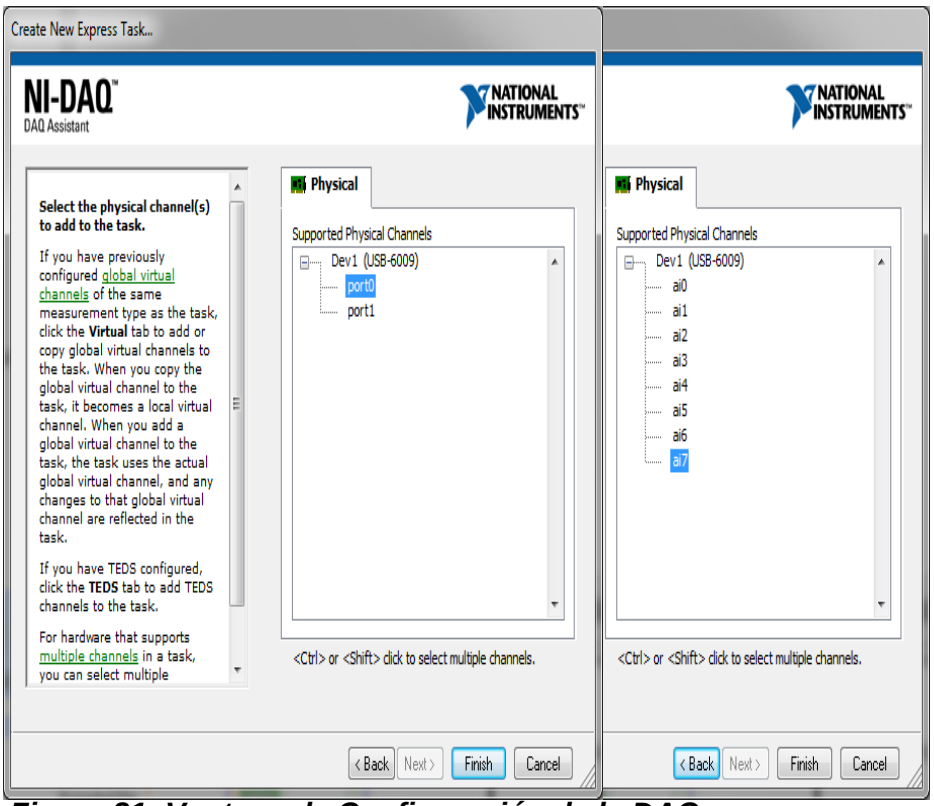


Figura 31: Ventana de Configuración de la DAQ.

Como lo habíamos mencionado, para la temperatura vamos a utilizar entradas digitales, por lo que configuramos el puerto 0 de la DAQ, para los datos que los obtenemos del micro-controlador de forma paralela. En este caso son 8 [0…7]. Y para el dato de humedad, seleccionamos la entrada análoga AI7, ya que es voltaje que estamos adquiriendo.

3.1.2. Bloque Convertidor

En el momento que adquirimos los datos, tenemos que realizarle una conversión mediante funciones propias de Labview, dado que la temperatura nos las presenta en un arreglo en binario, y la humedad en un valor dinámico análogo.

Para el caso de la temperatura. El arreglo de binario automáticamente lo transforma en un arreglo decimal, del cual con la función *Index Array* escogemos el primer número de dicho arreglo. De esta manera obtenemos un escalar simple para así poderlo manipular dentro del programa sin ningún inconveniente. La función antes mencionada la encontramos en *Array,* dentro de *Programming.* El respectivo ícono se la mostramos en la ***figura 32.***

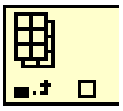


Figura 32: Index Array Function

De la misma manera, hacemos el proceso para la humedad, en este caso se utiliza la función *Convert from Dynamic Data,* dentro de la cual configuramos, que el dato a convertir sea un simple escalar como se muestra en la ***figura 33,*** con su respectivo ícono.

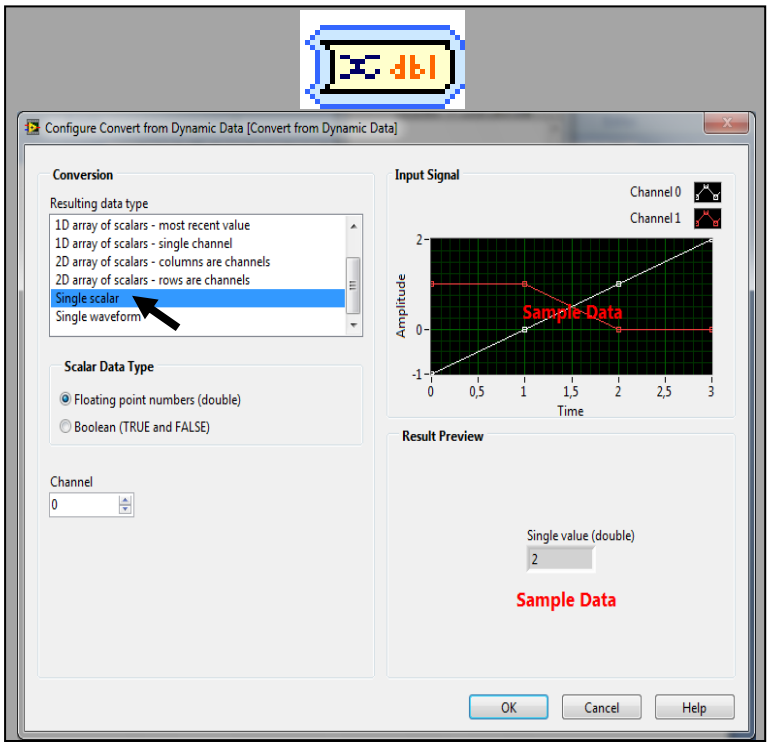


Figura 33: Icono y configuración Convert from Dynamic Data

3.1.3. Bloque mostrar datos

Una vez obtenidos los datos, utilizamos de la opción de los indicadores el *Thermometer* para mostrar la temperatura y *Meter* para la humedad.

De esta manera llegamos al usuario de una manera amigable, ya que los datos adquiridos en números se la mostramos de una manera gráfica.

3.1.4. Bloque Comparador

Este bloque nos ayuda a que las medidas de las condiciones ambientales sean las óptimas, para que los equipos trabajen sin que se presente ningún defecto o se sobrecaliente, por el aumento o disminución de temperatura y humedad. Para ello nos hemos planteado valores constantes máximos y mínimos. Y también permitirle al usuario que pueda definir dichos valores. Para este trabajo y por los valores promedios en Guayaquil. Los valores planteados son:

- Para la temperatura máxima y mínima de 20ºC y 25ºC respectivamente.
- Para la humedad relativa entre 50% y 75%.

La manera sencilla de escoger que valores límites deseamos, si automático o manual, es con un *Switch* en conjunto con la función *Select,* la misma que tiene tres entradas, un selector a la cual le conectamos el *switch* y las otras dos entradas son los valores límites, como constante el automático, y el manual con un control

numérico del panel frontal. El control tiene una ventaja, ya que el usuario tiene la posibilidad de cambiar los valores máximos y mínimos de acuerdo a sus necesidades. Luego escogemos *Num Ctrl* de los elementos en los controles numéricos.

Este proceso lo realizamos para todos los valores máximos y mínimos, tanto de temperatura y humedad.

Ahora se procede a comparar con los valores topes definidos, manuales o automáticos, dependiendo del usuario. En el cual usaremos a la opción *Greater* (mayor que), para los valores máximos y la opción *Less* (menor que), para los mínimos. Esto lo utilizamos tanto para temperatura como para humedad.

3.1.5. Bloque Generador de Alarmas

Cuando los datos adquiridos no se encuentran entre sus valores límites, se procede a activar una alarma, representado con *Leds*, y así pueda ser visible para el usuario. En nuestro proyecto hemos separado todas las señales de alarmas. De esa manera sabemos exactamente, que medición hay que controlar y así proceder de la mejor forma.

3.1.6. Bloque Generador y Envío de Informe

En esta sección se creará un archivo de tipo Excel, y luego enviarlo a diferentes correos electrónicos, para así siempre estar alerta, cuando sufra cambios el área a controlar. El cual es un informe detallado de todo el sistema de adquisición.

Para generar el informe utilizaremos diversas funciones *Format Date/Time String Function, Build Path Function, Write to text File Function*, para en conjunto crear el encabezado del archivo, el nombre del archivo y la dirección donde lo guardaremos.

Además se utiliza la función *Flat Sequence Structure* y *Write to File,* para junto con indicadores de texto ir armando lo que va a componer el informe.

El *Flat Sequence Structure* nos ayuda a que primero se escriba el encabezado del archivo, y una vez ya hecho ese proceso como se muestra la función *Write to File* unido a *Flat Sequence Structure*, darle paso a lo escrito en los indicadores de texto. Todo este proceso se lo muestra en la ***figura 34.***

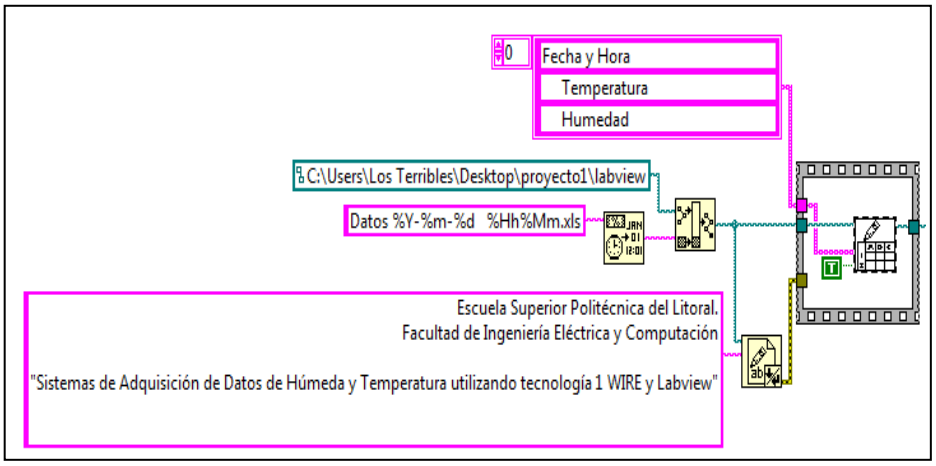
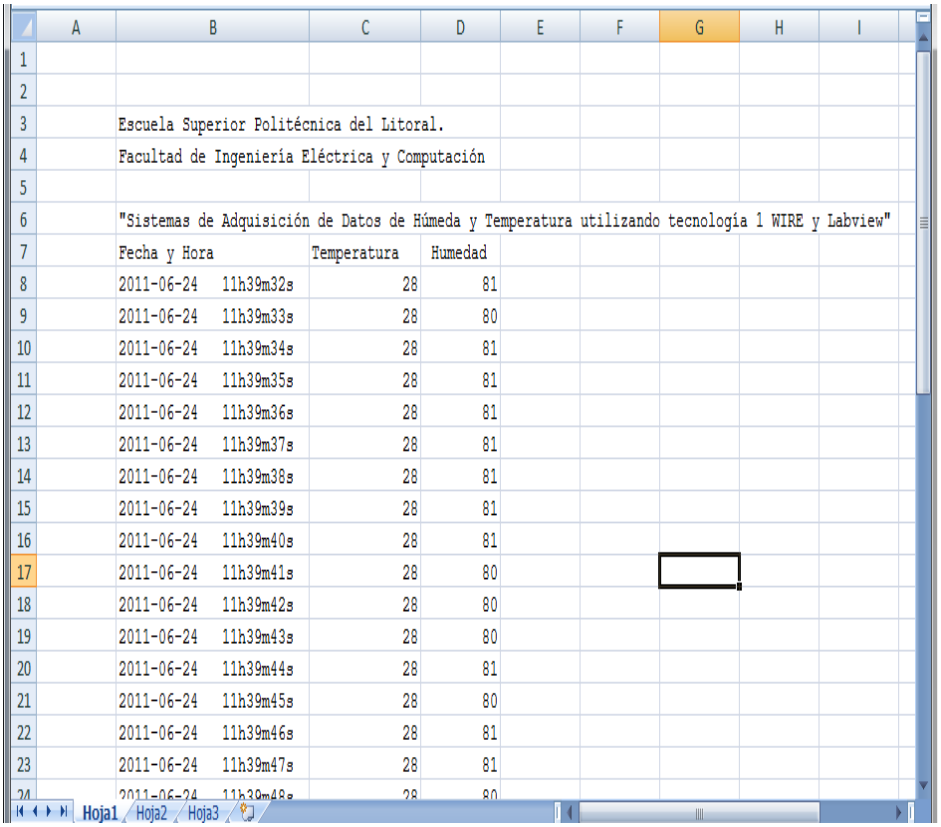


Figura 34: Encabezado y Nombre del Archivo.

Hay que tomar en cuenta que esta parte se la realiza fuera del lazo *While*, porque este proceso se lo realiza una vez por ejecución.

Después de terminada la primera parte, incluiremos los datos adquiridos dentro del archivo. En el mismo vamos a tener también la fecha y hora real, los datos de la temperatura y humedad.

Para esto vamos a utilizar la función *Built Array.* Es la que se encarga de unir todos los datos antes mencionados. Para luego hacer uso otra vez de la función *Write to File*, pero ahora sí dentro del lazo, debido a que se va a estar tomando los datos constantemente hasta que el usuario lo requiera. En la siguiente imagen podemos observar como se ve el archivo que se genera.



	A	B	C	D	E	F	G	H	I
1									
2									
3		Escuela Superior Politécnica del Litoral.							
4		Facultad de Ingeniería Eléctrica y Computación							
5									
6		"Sistemas de Adquisición de Datos de Húmeda y Temperatura utilizando tecnología 1 WIRE y Labview"							
7		Fecha y Hora	Temperatura	Humedad					
8		2011-06-24 11h39m32s	28	81					
9		2011-06-24 11h39m33s	28	80					
10		2011-06-24 11h39m34s	28	81					
11		2011-06-24 11h39m35s	28	81					
12		2011-06-24 11h39m36s	28	81					
13		2011-06-24 11h39m37s	28	81					
14		2011-06-24 11h39m38s	28	81					
15		2011-06-24 11h39m39s	28	81					
16		2011-06-24 11h39m40s	28	81					
17		2011-06-24 11h39m41s	28	80					
18		2011-06-24 11h39m42s	28	80					
19		2011-06-24 11h39m43s	28	80					
20		2011-06-24 11h39m44s	28	81					
21		2011-06-24 11h39m45s	28	80					
22		2011-06-24 11h39m46s	28	81					
23		2011-06-24 11h39m47s	28	81					
24		2011-06-24 11h39m48s	28	80					

Hoja1 Hoja2 Hoja3

Figura 35: Archivo que se genera.

Por último vamos a enviar el informe a correos electrónicos. Esto lo vamos a hacer cada vez que los datos sobrepasen los valores máximos o mínimos. Y como sabemos son 4 valores, utilizaremos la función *Compound Arithmetic* en forma de *Or,* para que en el momento que una de las 4 señales envíen alerta, se active la función *Case Structure,* y a su vez se envíe el o los correos electrónicos mediante la función *SMTP Email Send File.* Y como se muestra en la ***figura 36*** que el archivo creado se une con la

función *SMTP Email Send File,* para ser enviado como archivo adjunto.

Cabe recalcar que los correos son configurados por afinidad del usuario, y que para poderlo realizar se debe tener una cuenta de servidor de correos. Para este caso demostrativo hemos obtenido una cuenta de prueba.

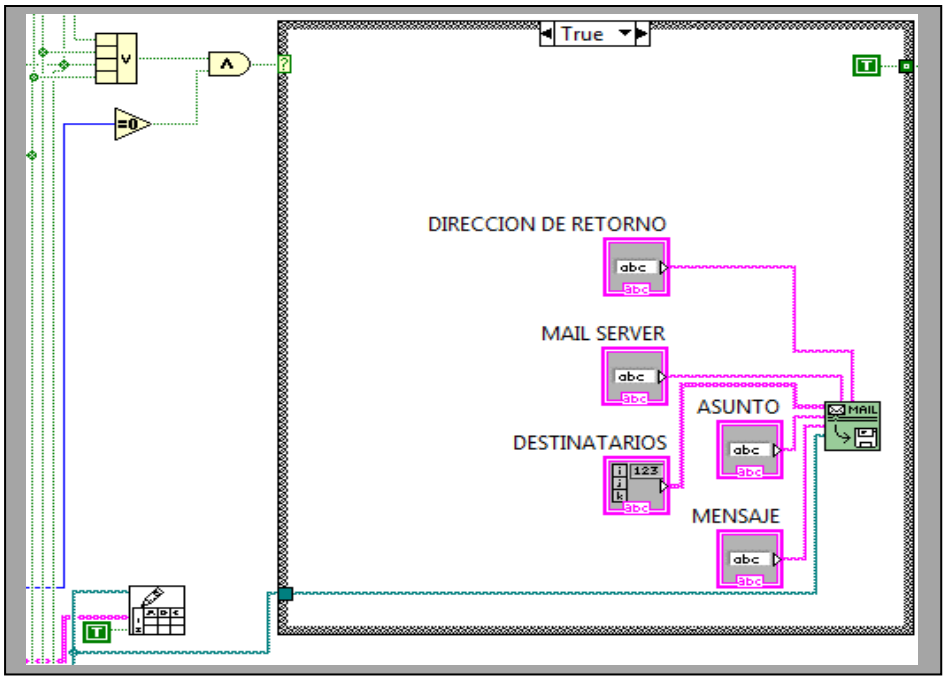


Figura 36: Proceso de Envío de Correos.

Y en la ***figura 37*** se muestran los campos que se deben llenar, para enviar el correo correctamente. Y cuando suceda esto se encenderá el indicador *led.*

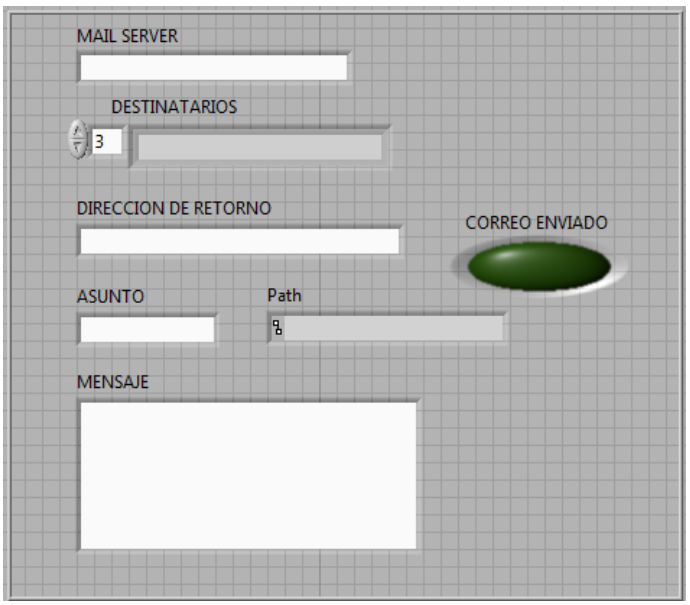


Figura 37: Campos del Mail Server.

3.1.7 Bloque de Envío de Señales de Control: Automática y Manual.

Como todo sistema de control y monitoreo, se maneja de dos maneras: Automáticamente y manualmente.

La forma automática se activa, cuando uno de los valores adquiridos sobrepasa los límites. Al mismo tiempo se enciende un *led* o los *leds*, alertando así al usuario.

La manera manual, no necesita que los datos estén sobre o bajo sus límites, para que el operador pueda activar estas señales.

Luego de que se activen las señales de control, utilizamos la función *DAQ Assist,* para así mandar a activar los dispositivos de control del exterior. Y como vamos a enviar señales análogas, se transmitirá por los puertos *Ao0 y Ao1.*

En la siguiente gráfica le mostramos este bloque.

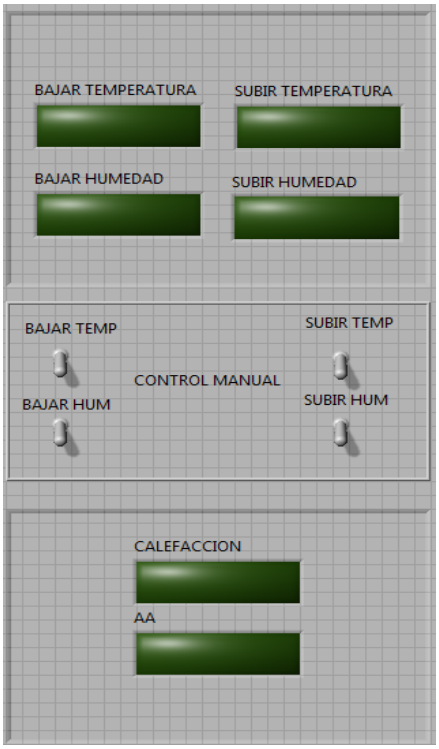


Figura 38: Control Sistema.

Una vez ya terminado todo el proceso de la programación, en **Anexo 1 y Anexo 2** le mostraremos el código gráfico y la interacción de la computadora con el usuario.

CAPITULO 4

4. Creación del Prototipo

En este capítulo se construyó una maqueta como se muestra en **Anexo 6** para implementar nuestro sistema, y probar como funcionaría en la parte práctica. Nuestro prototipo cuenta con tres etapas:

- Parte de adquisición de datos. **Anexo 2**.
- Parte de control. **Anexo 3**.
- Parte de fuerza. **Anexo 3**.

La parte de control está dada por el ordenador con el programa diseñado en LabView en conjunto con la tarjeta de adquisición, que ya se explicaron en los capítulos anteriores.

La parte de adquisición de datos está compuesta por la tarjeta de adquisición de datos y por la tarjeta con el sensor de calor y el sensor de humedad. Estas dos tarjetas están conectadas entre sí y a su vez la DAQ está conectada a la

computadora para monitorear los datos y enviar las señales para controlar los sistemas de calefacción y aire acondicionado.
Las señales para controlar la calefacción y A/A son también proporcionadas por la DAQ, pero esta señal de control va hacia la tarjeta que maneja la etapa de fuerza.

La parte de fuerza es como explicamos un poco anteriormente la que controla la calefacción y el A/A. La misma que está formada básicamente por transistores, relés, resistencia y diodos. La llamamos la parte de fuerza debido a que esta puede controlar voltajes de hasta 220 Vac, esto se debe a los interruptores internos de los relés.

Debemos aclarar que a lo que nosotros llamamos calefacción es un pequeño sistema formado por un ventilador y una resistencia térmica la cual al ser activado envía aire caliente hacia el cubo de acrílico donde está en su interior la tarjeta con los sensores.

Y a lo que nosotros llamamos A/A es un sistema de un ventilador con hielo seco, usamos esto debido a que no podemos usar un sistema de compresor para enfriar nuestro pequeño cuarto ya que es un caso demostrativo y no hay un A/A de esas dimensiones.

Ahora explicaremos el funcionamiento de la maqueta, todo el sistema vamos a manejarlo desde la computadora, al encender el sistema. Este toma los datos de los sensores y nos muestra los datos obtenidos del ambiente. Ahora activamos manualmente por un minuto la calefacción y observamos en la pantalla que va subiendo la temperatura interior del cubo, en el momento en que el sensor tome una medida mayor al límite establecido por nosotros, nuestro sistema activará una alarma de medición de temperatura elevada y enviará en ese momento una señal para activar el A/A, previamente hemos desactivado la calefacción manualmente.

Un proceso similar es el que para cuando se baja la temperatura menor que el límite establecido. Pero en este caso se activa la calefacción para calentar nuestro ambiente. Tenemos que acotar que cada vez que se activa manualmente las señales de calefacción y A/A se envía una señal desde el ordenador a la DAQ y de esta hacia la tarjeta de fuerza.

CONCLUSIONES

1. Podemos concluir que Labview es una de las herramientas más poderosas existentes en simulación y monitoreo, con su ayuda se puede automatizar todo un sistema para una empresa. Pero para implementaciones a mayor escala se necesita una tarjeta de adquisición de datos que tenga mas entradas y mas salidas, y así tener un control total.

2. Se logró implementar el sistema de control y monitoreo con el uso de las herramientas y protocolos planteados inicialmente. No importó que tipo de dato ingresaba o salía del sistema, que utilizando funciones de Labview nos ayudaron a no tener problemas de conexión.

3. Además se realizó un informe detallado del sistema en Excel, donde se incluyó un encabezado, la fecha y hora de cada dato obtenido y los parámetros monitoreados.

4. Logramos enviar correos electrónicos adjuntando el informe del sistema cuando los parámetros controlados no se encuentran entre sus valores límites, y así siempre monitorear sin la necesidad de la presencia humana.

RECOMENDACIONES

1. En el momento de hacer las conexiones dentro del programa Labview verificar que las señales sean del mismo tipo, sino se tendrá errores de conexión.

2. Tener mucho cuidado en la configuración de la DAQ, cuando se quiera hacer una adquisición tener siempre claro cuáles de los puertos son entradas y cuales son salidas.

3. Con respecto a la programación del micro-controlador utilizar el software Micro Basic, ya que posee librerías de protocolos 1 wire, que ayudan a un mejor desarrollo. Al igual de escoger el micro controlador más adecuado, según las necesidades requeridas.

4. Para todo sistema de control y monitoreo siempre tener la opción de poderlo controlar manualmente. Como sabemos nunca estamos a expensas de una falla automática.

ANEXOS

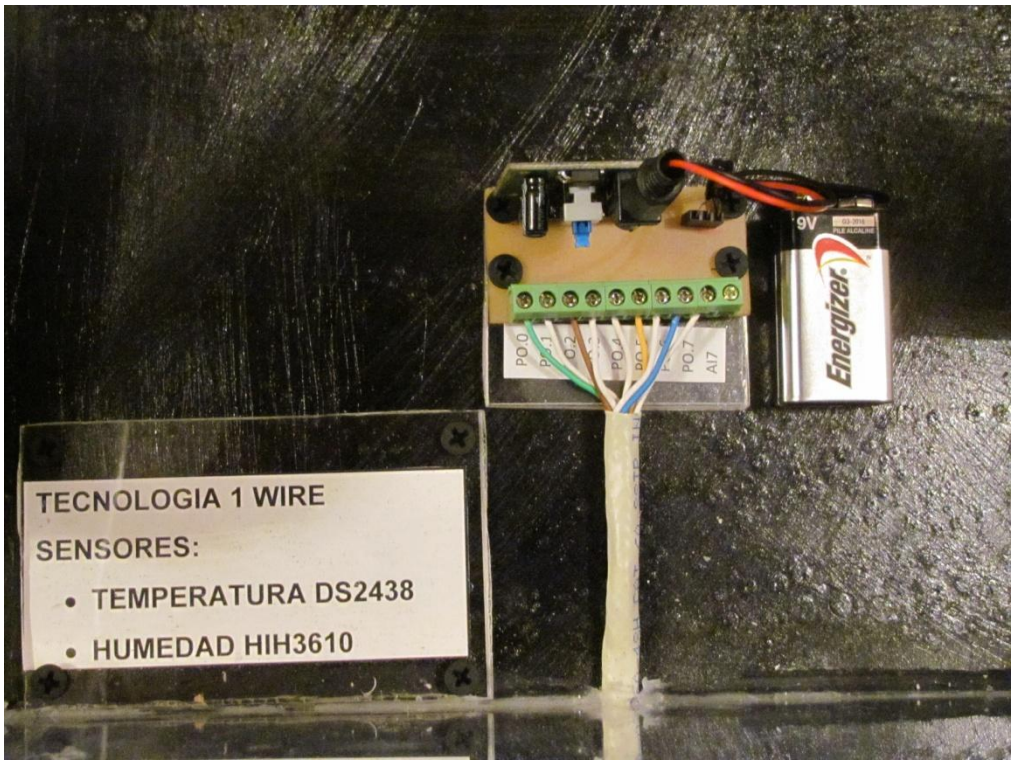
Anexo 1: Código en C del programa para el micro-controlador

```
program codigo_ds2438


dim txt as string[23]
dim temperatura as float


sub function get_temperatura() as byte
    dim Dato as byte[9]
    dim i as byte
    dim Enteros as float
    dim Decimal as float
    dim Valor as float


    Delay_ms(1)


    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC) ' skip rom
    Ow_Write(PORTA,0,0x4E)


    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC) ' skip rom
```

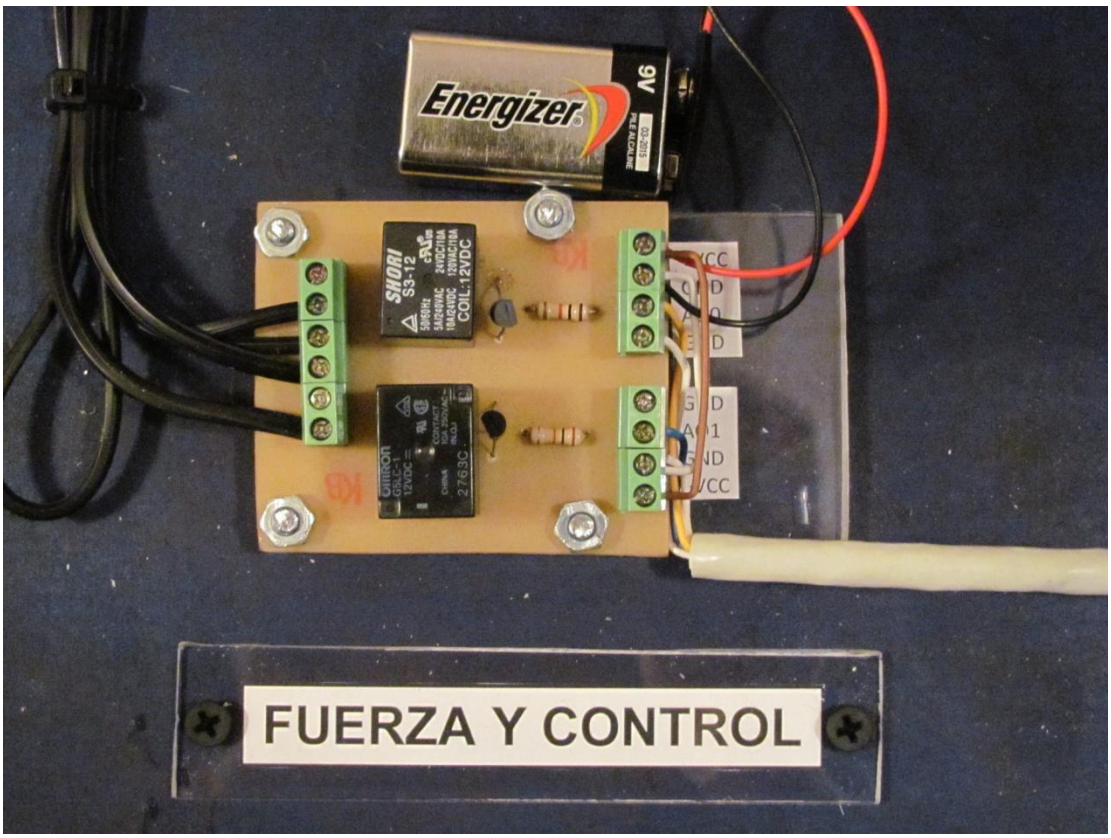
```
    Ow_Write(PORTA,0,0x44) ' temperature
    Delay_ms(1)

    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC) ' skip rom
    Ow_Write(PORTA,0,0xB8) ' recall memory
    Ow_Write(PORTA,0,0x00)

    ow_reset(PORTA,0)
    Ow_Write(PORTA,0,0xCC) ' skip rom
    Ow_Write(PORTA,0,0xBE) ' recall memory
    Ow_Write(PORTA,0,0x00)

    for i=0 to 8
        Dato[i] = Ow_Read(PORTA,0)
    next i

    if TestBit(Dato[2],7) then
        Dato[2]= Dato[2] xor 0xff
        setbit(Dato[2],7)
    end if
    result = Dato[2]
end sub
```

```
sub procedure conf_16f886
    OSCCON = 0x75 ' config osc int 8mhz
    OPTION_REG = 0X80 ' desactv el pull up en portb

    TRISA = 0x01 ' porta.0 como entrada
    PORTA = 0x00

    TRISB = 0x00 ' todoc como salida
    PORTB = 0X00

    TRISC = 0X01 ' nnnn
    PORTC = 0X00
    ANSEL = 0X00 ' salidas digitales
    ANSELH = 0X00 ' digitales
end sub

main:
    conf_16f886()
    while 1
        PORTB = get_temperatura()
        delay_ms(200)
    wend
end.
```

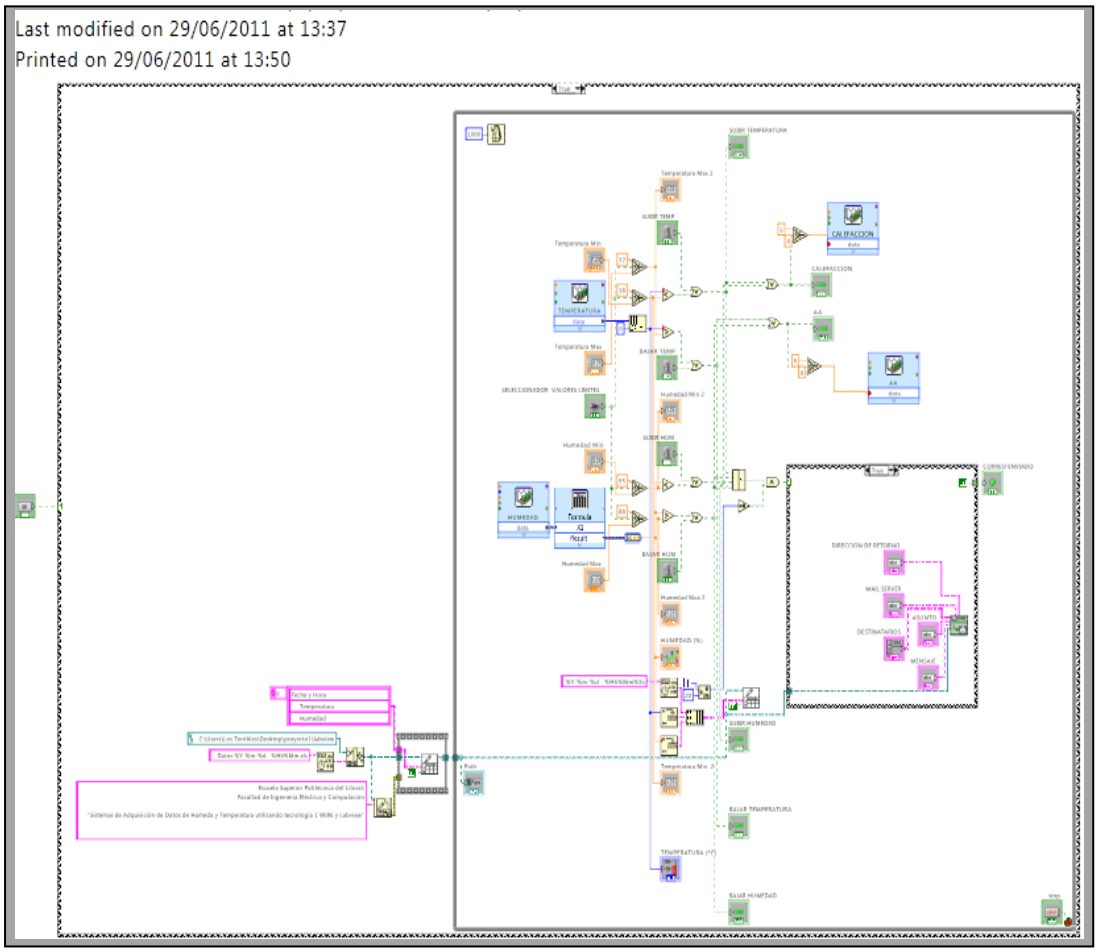
ANEXO 2: Parte de Adquisición de Datos.



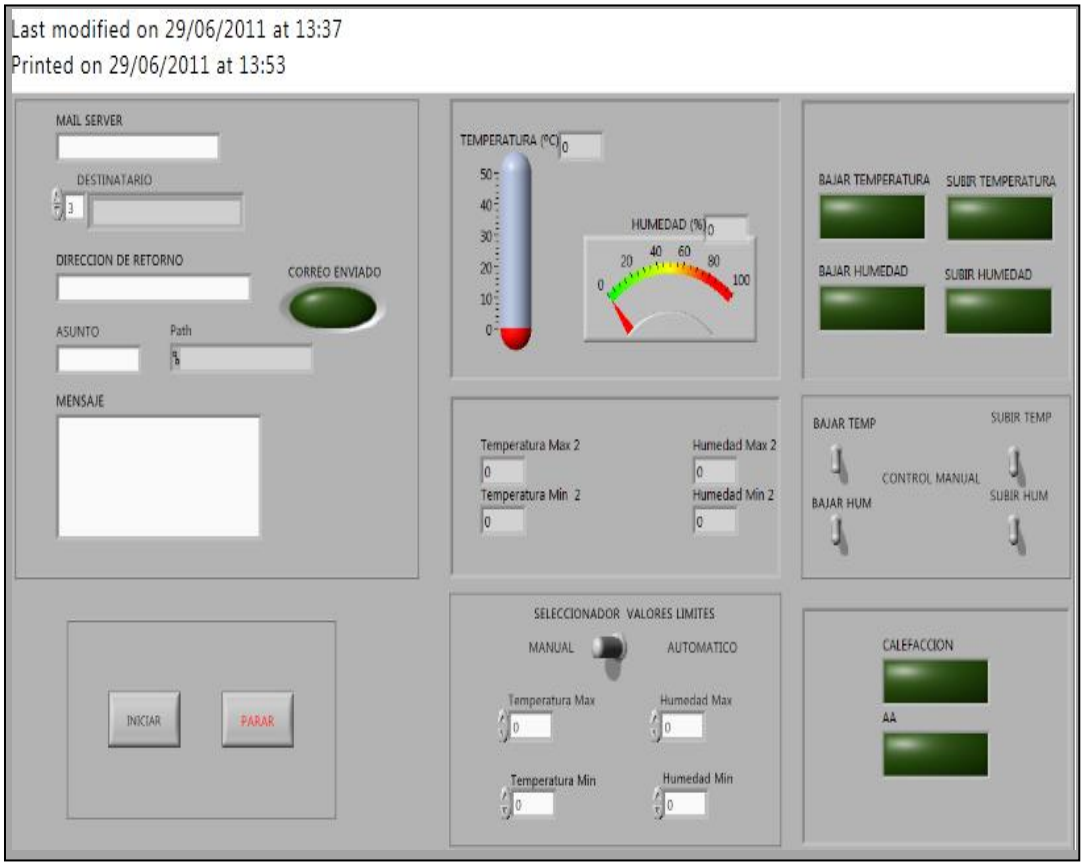
ANEXO 3: Parte de Control y Fuerza



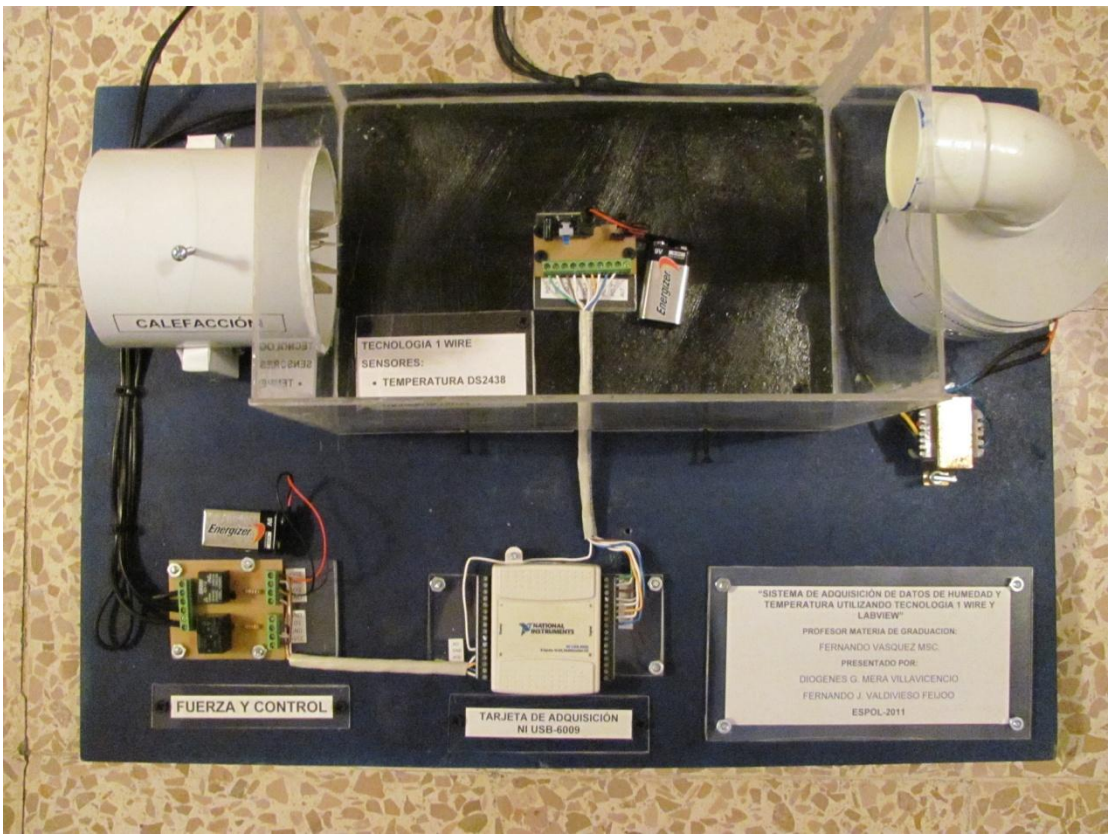
ANEXO 4: Diagrama de Bloque



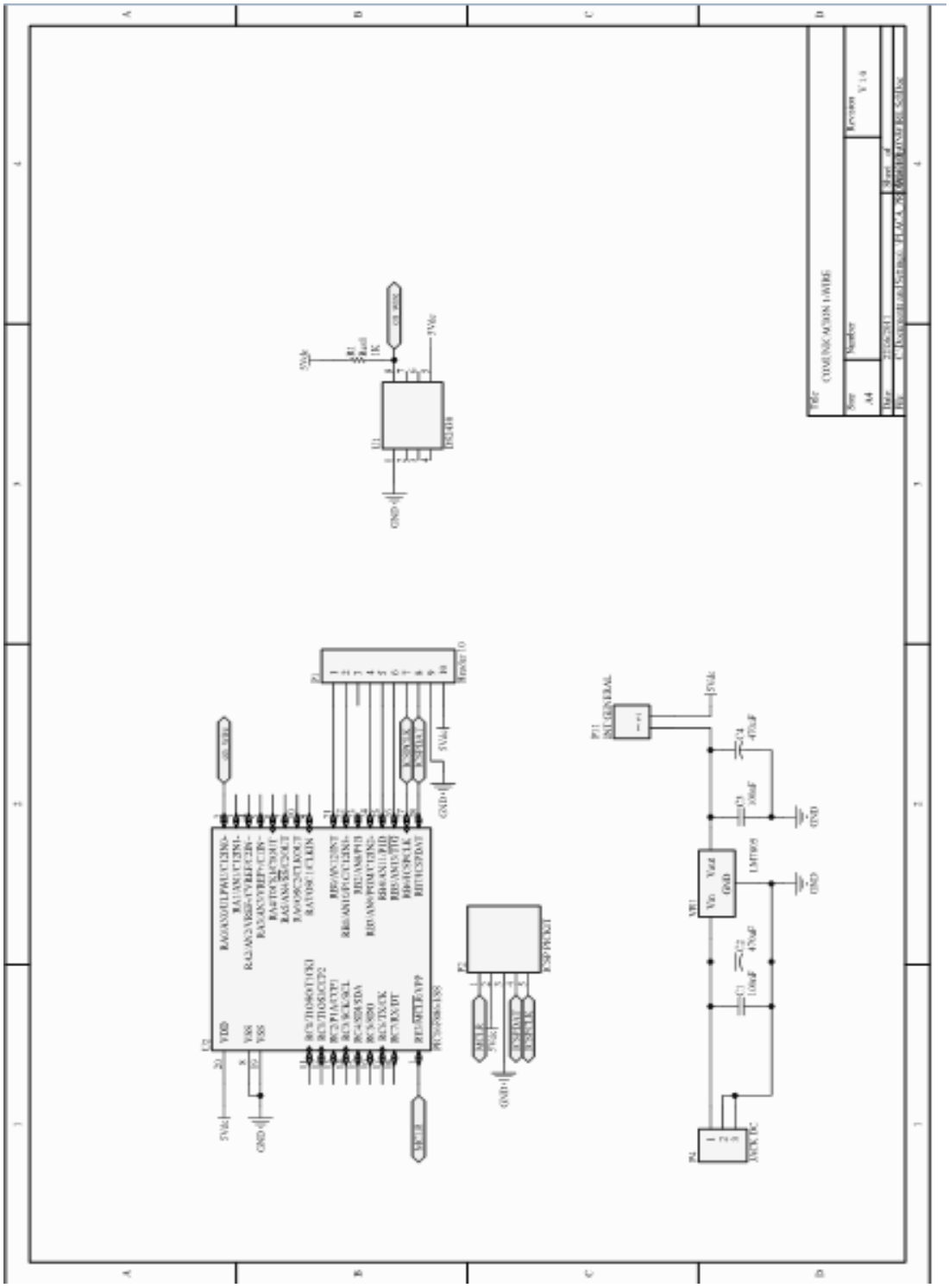
ANEXO 5: Panel Frontal



ANEXO 6: Presentación de Maqueta



Anexo 7: Diagrama Esquemático de la placa de los sensores



REFERENCIAS BIBLIOGRÁFICAS

[1] Wikipedia, Sensor, http://es.wikipedia.org/wiki/Sensor, fecha de consulta noviembre 2010.

[2] Honeywell, Sensor de Humedad, http://content.honeywell.com/sensing/prodinfo/humiditymoisture/009012_2.pdf, fecha de consulta noviembre 2010.

[3] Maxim-Ic, Sensor DS2438, http://pdfserv.maxim-ic.com/en/ds/DS2438.pdf, fecha de consulta enero 2011.

[4] Maxim-Ic, 1-Wire, http://www.maximic.com/products/1-wire, fecha de consulta marzo 2011.

[5] Future Electronics, PIC16F886, http://www.futureelectronics.com, fecha de consulta marzo 2011.

[6] National Instrument, NI USB-6009, www.ni.com/pdf/manuals/371303l.pdf, fecha de consulta marzo 2011.

[7] Bishop Robert H., LABVIEW 2009 Student Edition, Pretince Hall 2010.

[8] Logic Electronic, NI USB-6009, http://www.logicelectronic.com/productos/Adquisicion%20NI/NI%20USB-6009.html, fecha de consulta mayo 2011.