



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingenierías

**Entorno para la adquisición y visualización de
señales basado en instrumentación virtual**

**TESIS RECEPCIONAL
Para obtener el grado de**

Ingeniero en Redes

**Presenta
HUGO BASTIÁN HERNÁNDEZ**

**Director de Tesis
M.C. Víctor Manuel Sánchez Huerta**

Chetumal, Quintana Roo, Julio de 2007.



UNIVERSIDAD DE QUINTANA ROO División de Ciencias e Ingenierías

Tesis elaborada bajo la supervisión del Comité de Asesoría
y Aprobada como requisito parcial, para obtener el grado de:

INGENIERO EN REDES

COMITÉ DE TESIS

Director: M.C. Víctor Manuel Sánchez Huerta _____

Asesor: M.C. Javier Vázquez Castillo _____

Asesor: M.T.I. Melissa Blanqueto Estrada _____

Chetumal, Quintana Roo, México, julio de 2007

Agradecimientos

Este trabajo está dedicado a ti madre y a la memoria de tu esposo, que haz esperado pacientemente la culminación de mi formación profesional, tu cariño, consejos y apoyos durante toda mi existencia.

A mis hermanos y mi familia de los cuales he recibido la comprensión y apoyo, a lo largo de los últimos años.

A mis maestros que han soportado durante las horas de clases, mi comportamiento y mi mal carácter.

A mis compañeros de la carrera por estar siempre en mutuo apoyo, sin el cual no habría soportado la estancia en la institución

Al coordinador de la carrera de redes, por la atención siempre cordial, así como los apoyos recibidos de su persona y su esposa.

A mis asesores de este trabajo y muy especialmente a usted maestro Javier del cual he recibido innumerables horas de su vida dedicadas a mi formación y a la de Erendira, así como los apoyos en todos los sentidos, que usted nos ha ofrecido desinteresadamente.

Al patronato de la universidad por el apoyo recibido durante mi estancia en esta institución.

ÍNDICE

Resumen	7
1 Introducción	8
1.1 Problema de estudio.....	8
1.2 Objetivo general:	9
1.3 Objetivos particulares	9
2 Marco teórico	10
2.1 Adquisición de datos	11
2.1.1 Etapas que conforman un sistema de adquisición de datos... 11	
2.1.2 Características de los ADC	17
2.1.3 Especificaciones de los ADC.....	21
2.1.4 Hardware.....	23
2.1.5 Software de la aplicación.....	23
3 Metodología para el desarrollo del sistema de adquisición de datos.....	24
3.1 Etapa de adquisición	24
3.1.1 Sistema electrónico	24
3.1.2 Microcontrolador.....	25
3.1.3 Canales de adquisición	27
3.1.4 Sensor de temperatura LM35 AH.....	27
3.1.5 Sensor de humedad	27
3.1.6 Sensor de irradiancia solar.....	28
3.1.7 Comunicación RS-232 a PC.....	29
3.2 Lenguaje utilizado.....	30
3.2.1 Algoritmos de la etapa de adquisición	30
3.2.2 Interfaz Gráfica.....	31
3.2.3 Descripción de la interfaz gráfica	31
3.2.4 Componentes mínimos aceptables en la interfaz.....	32
3.2.5 Canales de la visualización gráfica	32
3.2.6 Tiempo de muestreo	32
3.2.7 Archivo de almacenamiento	32
3.2.8 Controles de arranque y paro.....	33
3.2.9 Lenguaje G	33
3.2.10 Ventajas de los lenguajes G.....	33
3.2.11 Lenguajes G existentes en el mercado.	33
3.3 Entorno de programación de LabView	37
3.3.1 Diseño del panel frontal.....	38
3.3.2 Gráficas.....	39
3.3.3 Interruptor.....	40
3.3.4 Algoritmo de la interfaz gráfica	41
3.3.5 Diagrama de bloques de la interfaz gráfica	42
4 Resultados experimentales	51
4.1 Monitoreo y registro de temperatura, humedad relativa e irradiancia solar	51
4.2 Visualización del funcionamiento interfaz.....	53
5 Conclusiones	54

6	Referencias	55
7	ANEXO A.....	57
7.1	Programación en PicBasic del microcontrolador	57
7.2	Descripción general de los comandos usados	57
7.3	Codificación.....	58
8	Anexo B.....	59
8.1	Hardware del sistema de adquisición de datos	59
8.2	Fotografías del sistema de adquisición	61

Índice de imágenes

Figura 1.	Diagrama de las etapas que conforman un sistema de adquisición de datos.....	12
Figura 2.	Diagrama de circuito básico de muestreo y retención	18
Figura 3.	Error de Offset	22
Figura 4.	Alias de la señal	22
Figura 5.	Diagrama electrónico del sistema de adquisición de datos por puerto serial	25
Figura 6.	Especificaciones del microcontrolador PIC16F873A	26
Figura 7.	Encapsulado del microcontrolador Pic16f873A	26
Figura 8.	Encapsulado del sensor de temperatura LM35	27
Figura 9.	Esquema del sensor de Humedad.....	28
Figura 10.	Esquema del sensor de Humedad.....	29
Figura 11.	Configuración de pines del puerto Serial	30
Figura 12.	Diagrama de flujo del proceso de adquisición de datos mediante el PIC16F873A	31
Figura 13.	Panel frontal.....	37
Figura 14.	Diagrama de bloques	37
Figura 15.	Elementos del panel frontal del sistema de adquisición de datos diseñado	38
Figura 16.	Gráfica de temperatura	39
Figura 17.	Gráfica del porcentaje de humedad.....	39
Figura 18.	Gráfica de la irradiancia solar	40
Figura 19.	Control de paro y disparo de la adquisición del puerto.....	40
Figura 20.	Diagrama de flujo y algoritmo de la interfaz gráfica	41
Figura 21.	Diseño del diagrama de bloques de la interfaz gráfica, While loop, Stacked sequence, Case sequence.....	42
Figura 22.	Estructura case “Case Structure”, que toma los datos y los procesa	43
Figura 23.	Control booleano(falso o verdadero).	43
Figura 24.	Configuración del puerto serial en el diagrama a bloques	44
Figura 25.	Configuración del periodo de lectura en el puerto serial	45
Figura 26.	Ejecución de la instrucción de apertura de puerto serial.....	45
Figura 27.	Función de lectura del puerto serial.....	45

Figura 28. Función de conversión de la cadena con los tres datos a un arreglo de Bytes	46
Figura 29. Función que toma un dato específico de un arreglo	46
Figura 30. Función que multiplica dos valores	46
Figura 31. Función que visualiza un valor numérico	47
Figura 32. Función que gráfica los valores	47
Figura 33. Función de conversión de un número a cadena.....	48
Figura 34. Función de Concatenación de cadenas	48
Figura 35. Función que regresa la hora y fecha del sistema.....	48
Figura 36. Función de apertura de archivo de escritura	49
Figura 37. Función de escritura de datos al archivo	49
Figura 38. Reloj para evaluar el lapso de escritura	49
Figura 39. Función de comparación de dos valores (menor o igual)	49
Figura 40. Función de cierra el puerto serial	50
Figura 41. Función de reporte de errores en el puerto serial	50
Figura 42. Gráfica de la Irradiancia solar registrada durante el día.....	51
Figura 43. Gráfica de la temperatura durante el día	52
Figura 44. Gráfica de humedad relativa registrada durante el día.....	52
Figura 45. Aplicación en un instante por la mañana	53
Figura 47. Diagrama esquemático del circuito electrónico del sistema de adquisición de datos	60

Fotografías del entorno de adquisición

Ilustración 1. Ventana de aplicación y sistema de adquisición de datos implementado	61
Ilustración 2. Circuito del sistema de adquisición de datos.....	62
Ilustración 3. Sistema de adquisición de datos en operación.....	63
Ilustración 4 Sistema de adquisición de datos en operación	64

Resumen

Existen diferentes configuraciones de hardware y software para implementar un sistema que adquiera, visualice y registre datos de una o más variables físicas de un proceso en una computadora personal. La utilización de la computadora personal como un medio de registro y visualización del comportamiento de una variable física en el tiempo se le conoce como instrumentación virtual. Las características de este sistema están en función de la exactitud que se requiera del registro de las variables, así como de la velocidad de cambio de la variable con respecto al tiempo. Una configuración clásica consiste en utilizar un microcontrolador (hardware) en la etapa de adquisición de datos de las variables a medir y una aplicación computacional (software) que los visualice en una computadora personal. Estos sistemas existen en el mercado, sin embargo son costosos, además de que no son configurables a las necesidades del usuario. Por otra parte, pueden ser diseñados y desarrollados con la tecnología apropiada; sin embargo, es necesario invertir un tiempo significativo en su elaboración ya que es necesario diseñar la etapa de adquisición, establecer el protocolo de comunicación entre el microcontrolador y la computadora personal, así como del diseño de la interfaz gráfica para la visualización de los datos. El objetivo de este trabajo de tesis es diseñar e implementar un sistema de adquisición y un entorno de instrumentación virtual utilizando un microcontrolador y un lenguaje gráfico que reduzcan el tiempo de desarrollo y costo de este tipo de sistemas. El sistema desarrollado servirá como base para la infraestructura de instrumentación de procesos del laboratorio de metrología de la Universidad de Quintana Roo, así como para sentar las bases para el desarrollo de un entorno de instrumentación virtual más sofisticado.

1 Introducción

1.1 Problema de estudio

En la universidad de Quintana Roo, existe la necesidad de sistemas de adquisición de datos e instrumentos de medición virtuales que automaticen las tareas rutinarias de obtención de datos en prácticas de laboratorio, prácticas de campo y apoyo a la investigación en las áreas de ingeniería en general. Sin embargo, el costo elevado de este tipo de sistemas en el mercado ha ocasionado que la tarea del registro y visualización de datos de variables físicas se realice de forma manual, trayendo consigo problemas de confiabilidad de las mediciones así como demora en el tiempo de experimentación.

A partir de esta problemática se determinó diseñar y construir un sistema de adquisición de datos que interactúe con un instrumento virtual para que registre y visualice el comportamiento de las variables físicas de un proceso en el tiempo. La solución está basada principalmente en la infraestructura de la universidad, ya que todos los laboratorios cuentan con computadoras personales en donde potencialmente se puede implementar un entorno de instrumentación virtual. Por otra parte, la mayoría de variables físicas de los procesos que es necesario instrumentar cambian lentamente su valor en el tiempo, por lo que no es necesario un sistema de adquisición de alta velocidad de muestreo. Esta característica permite que se utilice un microcontrolador para la etapa de adquisición de datos y que también establezca la comunicación con la computadora personal mediante una comunicación serial RS-232.

De esta forma, en este trabajo se presenta el desarrollo de un sistema electrónico basado en sensores y un microcontrolador que envíe de manera serial los datos a la computadora, donde mediante un entorno de instrumentación virtual, se visualizará el comportamiento de las variables físicas manera gráfica y numérica. Además, el instrumento virtual contará con la opción de publicación remota en un servidor WEB y un archivo de

salida de texto separado por comas que contenga los datos recogidos para análisis posteriores.

1.2 Objetivo general:

El objetivo de este trabajo consiste en diseñar e implementar un sistema de adquisición de datos y un entorno de instrumentación virtual utilizando un microcontrolador y un lenguaje gráfico que reduzcan el tiempo de desarrollo y costo para este tipo de sistemas.

1.3 Objetivos particulares

- Conocer el ambiente de programación gráfica LabView para el desarrollo de instrumentos virtuales.
- Caracterizar sensores de humedad relativa, temperatura e irradiancia solar.
- Diseñar el sistema de adquisición de datos a ser implementado, tanto en hardware como en software
- Implementar el sistema electrónico y el entorno de instrumentación virtual.

2 Marco teórico

El mundo físico está íntimamente ligado a las magnitudes analógicas con valores continuos como la temperatura, presión, humedad etc., mientras que los digitales emplean magnitudes con valores discretos, razón por la cual se requiere de sistemas que permitan llevar a cabo la transformación entre ambas magnitudes, con el fin de que los parámetros de sucesos físicos tengan mayor facilidad de tratamiento y poder dar a estos un potencial sistemático y eficiente a la hora de procesarlos y difundirlos [3].

Por otro lado, los sistemas digitales, en especial, los sistemas basados en microprocesadores y microcontroladores, han impulsado un cambio en las técnicas del tratamiento de la información, sustituyendo paulatinamente los sistemas analógicos por sistemas digitales. Esto fundamentalmente porque los sistemas digitales tienen un costo mas reducido, aumentan la potencia de cálculo y flexibilidad en el diseño. Además, la información digital puede ser procesada, transmitida y almacenada con gran facilidad [2].

La industria de la instrumentación está sufriendo importantes cambios como resultado del desarrollo de la PC. Estos cambios están ocurriendo tanto en el componente de hardware como en el software. Un elevado número de científicos e ingenieros en todo el mundo usan PC para automatizar sus tareas de investigación, diseño y fabricación [5].

Además, el desarrollo de redes de PC's y la aparición de la Internet están revolucionando la arquitectura fundamental de los sistemas de medición basados en PC [5].

Al estudiar la configuración de los sistemas de adquisición de datos modernos DAQs (Data Acquisition System), basados en equipos PC, se aprecia que una de las partes principales que componen dichos sistemas, es el software quien controla y administra los recursos de la PC, presenta los datos, y participa en el análisis[5]

Programas y lenguajes de programación que cumplan las tareas de desarrollo existen en gran número en el mercado actual, como por ejemplo

visual basic, C, C++, visual C++, plataforma .NET, pascal, LabWindows CVI, LabView, entre otros desarrollados para programar aplicaciones específicas [5].

2.1 Adquisición de datos

La adquisición de datos consiste básicamente en captar una señal física y llevarla a la PC, esto significa tomar un conjunto de variables mesurables en forma física y convertirlas en tensiones eléctricas, de tal manera que se puedan utilizar o puedan ser leídas en la PC [6].

Es necesario que la señal física pase por una serie de etapas que le permitan a la PC, ser capaz de interpretar la señal enviada. Una vez que las señales eléctricas se transformaron en digitales dentro de la memoria de la PC, se les puede procesar con un programa de aplicación adecuado dándole el uso que el usuario desea [6].

De la misma manera que se puede tomar una señal eléctrica y transformarla dentro de la PC en digital, también se puede convertir una señal digital en señal eléctrica y ser enviada a algún dispositivo externo para realizar el control de un proceso industrial o experimental [6].

2.1.1 Etapas que conforman un sistema de adquisición de datos

La señal física pasa por varias etapas para poder ser leída por la computadora, éstas son: etapa de transductores, etapa de acondicionamiento de la señal, etapa de adquisición y etapa de procesamiento, como lo muestra la figura 1 [6].

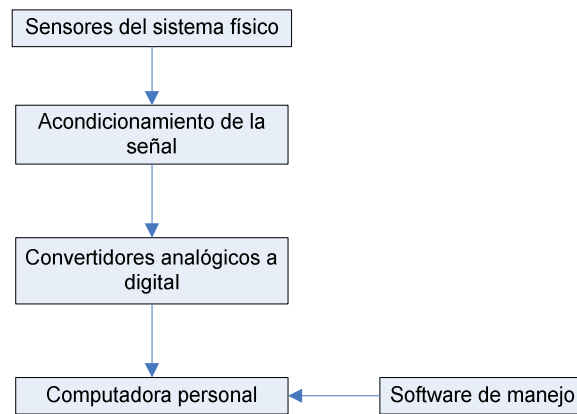


Figura 1. Diagrama de las etapas que conforman un sistema de adquisición de datos

2.1.1.1 Etapa de sensores

En esta etapa la señal física (temperatura, presión, luz, etc.) de la variable a medir es transformada en señales analógicas de voltaje o corriente, esta transformación se lleva mediante un circuito sensor, el cual asigna un voltaje o corriente de acuerdo al valor de la señal física en una escala determinada; como ejemplo un sensor de temperatura comercial como el LM35AH asigna un voltaje de 10 mV por cada grado centígrado de temperatura, es decir si tenemos 30 grados de temperatura en el ambiente, entonces el sensor a su salida nos proporcionará un voltaje de 300 mV [6].

Tipos de sensores

El **sensor** es el dispositivo que se encarga de transformar la variación de la magnitud de la señal física a medir en una señal eléctrica. Los sensores se pueden clasificar de la siguiente forma en:

- **Pasivos:** los que necesitan un aporte de energía externa.
- **Resistivos:** son los que transforman la variación de la magnitud a medir en una variación de su resistencia eléctrica. Un ejemplo puede ser un termistor que sirve para medir temperaturas.
- **Capacitivos:** son los que transforman la variación de la magnitud a medir en una variación de la capacidad de un condensador. Un

ejemplo es un condensador con un material en el dieléctrico que cambie su conductividad ante la presencia de ciertas sustancias.

- **Inductivos:** son los que transforman la variación de la magnitud a medir en una variación de la inductancia de una bobina. Un ejemplo puede ser una bobina con el núcleo móvil, que puede servir para medir desplazamientos.
- **Activos:** los que son capaces de generar su propia energía. A veces también se les llama *sensores generadores*. Un ejemplo puede ser un transistor en el que la puerta se sustituye por una membrana permeable sólo a algunas sustancias (IsFET), que puede servir para medir concentraciones [6].

2.1.1.2 Etapa de acondicionamiento

Por lo general las señales eléctricas producida por los sensores vienen contaminadas de ruido, armónicos, caídas de voltaje o corriente y otros fenómenos que distorsionan la señal original. Por lo que es necesario realizar un acondicionamiento de la señal proveniente de los sensores, de tal forma que la información original no sea perdida en el proceso de transmisión o conversión a una señal digital.

El acondicionamiento de señal es opcional, porque dependiendo de cada señal o aplicación, se puede o no requerir amplificación, atenuación, filtraje, aislamiento o algún otro tratamiento de señal [2].

En una etapa de acondicionamiento se puede encontrar estas subetapas, aunque no todas están siempre presentes:

Amplificación: Para conseguir la mayor precisión posible la señal de entrada deber ser amplificada de modo que su máximo nivel coincida con el máximo nivel que el convertidor o tarjeta pueda leer, de este modo se aprovecha todo el rango del dispositivo.

Aislamiento: El aislamiento eléctrico entre el sensor y la PC, es importante para proteger el sistema de adquisición de altas tensiones que puedan dañarlos. Un motivo adicional para usar aislamiento es el garantizar

que las lecturas del convertidor no sean afectadas por diferencias en el potencial de tierra o por tensiones en modo común. Cuando el sistema de adquisición y la señal a medir están ambas referidas a tierra pueden aparecer problemas si hay una diferencia de potencial entre ambas tierras, apareciendo un "bucle de tierra", que puede devolver resultados erróneos.

Multiplexado: El multiplexado es la conmutación de las entradas a la etapa de adquisición que esta conformada por un convertidor analógico-digital (ADC, por sus siglas en inglés, Analogic Digital Converter). Con esta técnica se utiliza un sólo ADC para medir los datos de diferentes canales de entrada. Puesto que el mismo convertidor está midiendo diferentes canales, su frecuencia máxima de conversión será la original dividida por el número de canales muestreados.

Filtrado: La finalidad de esta etapa es eliminar las señales no deseadas de la señal que proviene del sensor, esta etapa esta construida por un filtro electrónico que puede ser pasabajas, pasaaltas o pasabanda.

Linealización: Muchos sensores presentan una respuesta no lineal ante cambios lineales en los parámetros que están siendo medidos. Aunque la linealización puede realizarse mediante métodos numéricos implementados mediante software en la PC, esta etapa tiene el inconveniente de reducir la capacidad del sistema de adquisición de datos.

2.1.1.3 Etapa de adquisición

En esta etapa la señal de voltaje o corriente entregada por el sensor y la etapa de acondicionamiento, es transformada a señal digital mediante el ADC.

Los ADC representan los enlaces entre los sistemas analógicos y digitales. Los ADC, proporcionan un valor codificado digitalmente en función de la amplitud de la señal analógica, es decir producen un número binario que es directamente proporcional a un voltaje análogo de entrada o corriente.

Los ADC son sistemas electrónicos que reciben a su entrada una señal analógica, en general en forma de tensión variable y proporcionan a su salida una combinación en un código binario cuyo equivalente decimal, es igual a la magnitud de la variable analógica introducida [1].

2.1.1.4 Tarjetas de adquisición de datos (DAQs)

Existen alternativas de solución en el mercado para la tarea de convertir señales analógicas a digitales. Una de ellas son las tarjetas de adquisición de datos, en donde se integran las etapas de multiplexado, acondicionamiento y el ADC. La tarjeta de adquisición de datos permite que la PC reciba y envíe datos, así como comunicarse con otros dispositivos inteligentes.

Cuando se desea elegir una tarjeta DAQ se debe tomar en cuenta:

- Numero de canales entrada y salida de la tarjeta
- Características digitales y analógicas de los canales
- Contadores, relojes, cantidad de memoria
- Resolución de sus ADC(8,12,16 bits)
- Rango de lectura en voltaje y corriente
- Ganancia de entrada, impedancia y frecuencia
- Características y arquitectura modular
- Capacidad de expansión de canales
- Bus de comunicación con la PC
- Software de procesamiento de datos
- Compatibilidad con otros fabricantes y otros protocolos
- Características industriales (robustez, temperatura, filtros, etc.)
- Velocidad de muestreo [6]

2.1.1.5 Tipos de tarjetas DAQ

Las tarjetas DAQ se pueden clasificar de acuerdo a la función que realizan en:

- Tarjeta A/D convierte la señal analógica en digital. Los rangos del voltaje de entrada comúnmente utilizados son: 10v,-5v y 5v, 0 a 5v, 0 a 10v, también las hay que miden corriente entre 4 a 20 mA.
- Tarjetas D/A: Convierten una señal digital dada por la PC en una señal analógica. Los rangos normales de salida de tensión que otorgan estas tarjetas son de 5V, 10V, 0 a 5V y de 0 a 10V, también hay tarjetas que generan corriente de 4 a 20 mA.
- Tarjetas I/O: Son tarjetas de entradas y salidas digitales. Mediante estas tarjetas se pueden realizar tareas de encendido y apagado de dispositivos, o señalar el inicio o fin de un proceso. Por lo general se tiene un nivel bajo de 0 a 0.8 V, y un nivel alto de 2 a 5 V, dependiendo de cada fabricante.
- Tarjetas con relevadores: Son tarjetas que poseen un relevador de salida digital que se emplea para accionar un determinado componente del proceso. Este relevador cumple con las funciones de un interruptor.
- Tarjetas con acopladores de aislamiento eléctrico: Son tarjetas que poseen circuitos optoacopladores en las entradas digitales que permiten separar la electrónica del proceso con la electrónica de la computadora. Esto se emplea con la finalidad de proteger a la PC de un eventual cortocircuito.
- Tarjetas de comunicación: Permiten comunicar la PC con el medio exterior.
- Tarjetas inteligentes: Estas tarjetas cuentan con un microprocesador que les permite realizar cálculos y operaciones autónomamente [6].

2.1.2 Características de los ADC

A continuación se presenta una breve descripción de las principales características de los ADC, las cuales tienen que ver con velocidad de conversión, código binario de salida, forma de transmitir el valor digital obtenido.

Existen dos formas de transmitir a la PC el dato digital obtenido por el ADC, las cuales son:

Paralelo: suministran en terminales independientes simultáneamente una combinación binaria equivalente al valor de la variable analógica de entrada. Consiste en un circuito electrónico que recibe la variable analógica y actúa sobre un conjunto transistores bipolares o unipolares colocados en estado de saturación a corte correspondiente a los valores binarios cero y uno.

Serie: codifica la variable analógica mediante una combinación binaria en serie a través de una sola línea de transmisión, es decir los valores binarios correspondientes se presentan de manera secuencial por un solo canal de comunicación.

Para convertir una señal analógica en digital, se requiere de un proceso basado en cuatro fases que son: muestreo, retención, cuantificación y codificación.

Muestreo: Consiste en tomar una serie de muestras de la amplitud instantánea de la señal a intervalos regulares de tiempo, obteniendo de este modo una serie de pulsos modulados en amplitud; la frecuencia de muestreo f_c debe permitir la reconstrucción completa a partir de las muestras tomadas. Esto determina que dicha frecuencia sea al menos el doble de la frecuencia máxima $f_{max.}(x(t))$ de la señal a muestrear.

Retención: La tensión instantánea muestreada debe ser retenida entre dos impulsos consecutivos de muestreo para permitir que el circuito Análogo-Digital realice la conversión en el tiempo que precisa T_c (tiempo de conversión).

Las operaciones de muestreo y retención se obtienen a partir de circuitos denominados de muestreo y retención, gobernados a partir de la terminal $V_{s/h}$, como se muestra en la figura 2.

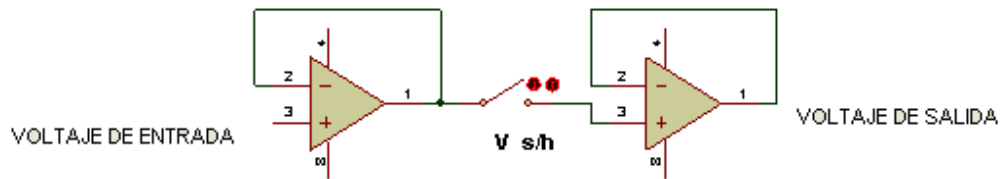


Figura 2. Diagrama de circuito básico de muestreo y retención

Cuantificación: El proceso de cuantificación consiste en asignar una función que determina que margen de valores de la tensión de entrada deberán proporcionar un único nivel de la función de salida. Si dicha función es lineal, entonces el intervalo de cuantificación es constante y se determinara por la ecuación 1:

$$q = \frac{V_{EntradaMax} - V_{EntradaMin}}{N} \quad (1)$$

Donde N es el número de niveles del cuantificador.

Siempre que se cuantifica una magnitud, a través de una función, se producirá un error, conocido como **error de cuantificación** que dependerá del intervalo de cuantificación y la función de salida del cuantificador, este valor del error esta dado por la ecuación 2:

$$\text{Error de cuantificación} = V_{salida} - V_{entrada} \quad (2)$$

Codificación: el proceso de codificación consiste en asignar un código digital a cada uno de los niveles de la salida del cuantificador.

Puesto que “q” representa el intervalo mínimo de cuantificación, este valor debe ser asignado a la variación mínima entre dos códigos consecutivos, equivalente al valor del bit menos significativo **LSB** del código empleado en la codificación. De este modo se puede establecer la relación por medio de la ecuación 3:

$$q = \frac{V_{EntradaMax} - V_{EntradaMin}}{N} = \text{LSB} \quad (3)$$

Cuando se aplica una cuantificación binaria a la salida de un cuantificador, se hace preciso que el número N de intervalos de cuantificación coincida con una potencia de dos, con el fin de obtener todas las combinaciones posibles en el código de salida formado a partir de **n** bit, como se ejemplifica en la Ecuación 4.

$$q = \frac{V_{EntradaMax} - V_{EntradaMin}}{2^n} \quad (4)$$

Los códigos digitales más utilizados son:

Códigos unipolares: Son aquellos destinados a representar únicamente valores positivos o negativos, correspondientes a la tensión de entrada en el cuantificador.

- **Binario natural (BN):** Es uno de los códigos de salida mas utilizados en los ADC, donde cada palabra binaria representa una fracción del rango a fondo de plena escala.
- **Binario codificado en decimal (BCD):** Este tiene agrupada la palabra binaria en grupos de cuatro dígitos, cada grupo representa un número decimal del 0-9, donde cada palabra binaria representa una fracción del rango a fondo de escala FS.

Códigos bipolares: Son aquellos destinados a representar valores positivos y negativos, correspondientes a la tensión de entrada del cuantificador.

- **Binario natural en complemento a dos:** Es el código de salida que mejor se adapta a las representaciones con variaciones de signo y simétricas con respecto a cero.
- **Binario natural desplazado (OB):** Este código aporta la solución a la aplicación del código Binario Natural a un ADC, agregar un offset para conseguir que a la palabra binaria 100...000 le corresponda un valor 0 V de entrada, quedando la mitad superior de la tabla asociada a las tensiones positivas y la mitad inferior a las negativas de forma continua.

2.1.3 Especificaciones de los ADC

Resolución: Especifica el número de bits del código de salida en relación con los intervalos de cuantificación q y el valor del rango de fondo de plena escala.

Tiempo de conversión: es el tiempo que ocurre entre la orden de inicio de conversión y la salida estable del código equivalente a la señal de entrada, generalmente asociada a una señal de fin de conversión.

Margen de tensión analógica de entrada: Define el rango admisible de tensión de entrada al ADC. Es importante saber que valores mínimos y máximos aceptará un convertidor, para que tenga una respuesta óptima y sus márgenes de errores se reduzcan al mínimo.

Código de salida: Especifica el tipo o los tipos de código de salida del convertidor. Lo más común es que la salida se codifique en binario natural o binario natural en complemento a dos, recordemos que los complementos de un número en cualquier base, es otro número que sumados permitan obtener el complemento pedido.

Salida digital: En este caso se refiere a los elementos tecnológicos digitales empleados para hacer la conversión (TTL, CMOS, ECL, etc.) y al tipo de salida (colector abierto, Tótem-Pole, Triestado, etc.).

Errores de conversión

Los errores que pueden ocurrir cuando se convierte una señal analógica a digital son los siguientes:

- **Error de ganancia:** Produce un valor de fondo de escala incorrecto. Un error de ganancia positivo hace que el valor de plena escala analógico se obtenga con un código digital menor que la mayor combinación de dígitos binarios posibles. Un error de ganancia negativo hace que el código obtenido sea producido por un valor menor que el valor de plena escala.
- **Error diferencial no-lineal:** Es la máxima diferencia entre dos valores de entrada que producen códigos de salida consecutivos.

- **Error integral no-lineal:** Es la integral del área limitada por la curva característica del convertidor y la curva ideal.
- **Error de apertura:** Es el error debido a la variación de la señal de entrada mientras se está realizando la conversión.
- **Error de offset:** Es un desplazamiento constante para todos los valores de la curva obtenida (real) y la curva original (ideal). Este efecto se muestra en la figura 3.

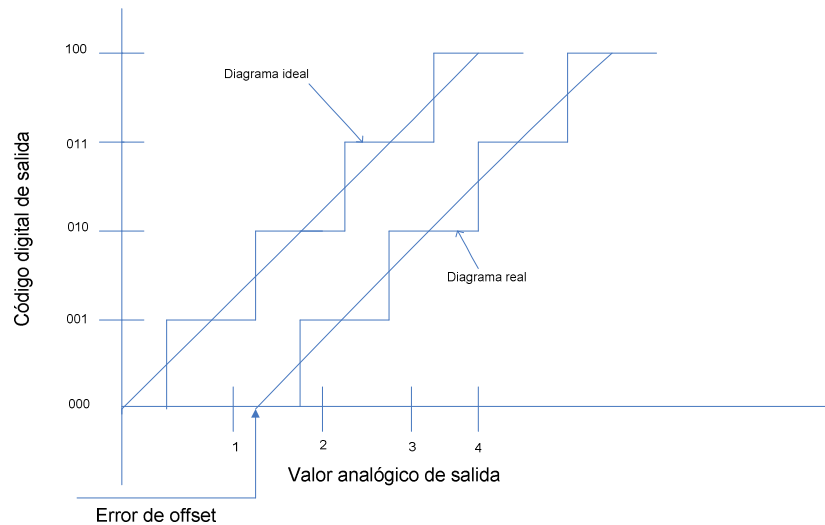


Figura 3. Error de Offset

Efecto aliasing: El aliasing se produce cuando la frecuencia de muestreo es menor que la de la señal que se muestrea, y se refiere al hecho de que podemos interpretar de una manera no exacta la señal original, apareciendo un "alias" de la señal [4]. Este efecto se muestra en la figura 4.

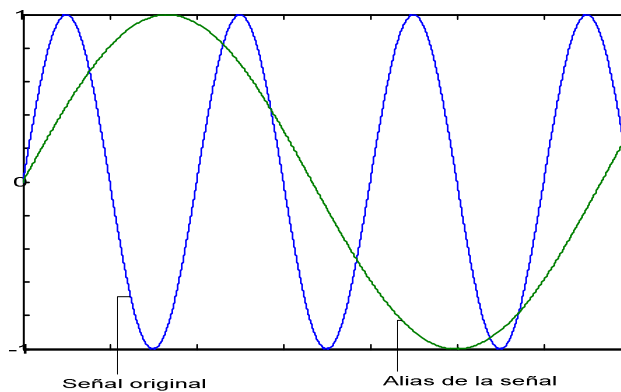


Figura 4. Alias de la señal

2.1.4 Hardware

Los sistemas de adquisición de datos para ser desarrollados necesitan de sensores que recojan la variable física y la transformen en señales eléctricas. Normalmente están en encapsulados y son desarrollados por diversas compañías del área de la electrónica. Los sensores pueden contar o no, con un sistema de acondicionamiento de señal que nos permita tener una tensión o corriente en niveles aceptables. Normalmente son desarrollados con componentes electrónicos dependiendo de cada caso particular de adquisición y un convertidor analógico a digital que dependiendo de las necesidades particulares se deberá seleccionar. Es muy común utilizar como circuito integrado a microcontroladores los cuales ya cuentan con salidas de comunicación (paralela, serial etc.) y ADC's ya integrados [5] [6].

2.1.5 Software de la aplicación

La aplicación que se ejecutara en la computadora, contendrá un ambiente grafico, amigable y con controles para manipular el curso normal de la ejecución; esta deberá contener configuraciones de arranque en un lenguaje de alto nivel, por lo que se propone un lenguaje especializado en el área de adquisición, control automático y instrumentación virtual.

Los lenguajes de programación de quinta generación o también conocidos por lenguajes G, son aquellos que no necesitan código escrito para desarrollar aplicaciones, pues su código se basa en iconos interconectados en un diagrama de bloques, los cuales cada uno tiene sus propiedades y métodos que les permiten interactuar para realizar la tarea para la cual fueron creados, reduciendo así la complejidad de la programación y acelerando los tiempos de desarrollo de las aplicaciones [5].

3 Metodología para el desarrollo del sistema de adquisición de datos

La programación de la aplicación desarrollada para la PC, esta basada en un lenguaje de programación de quinta generación, ya que estos lenguajes están desarrollados específicamente para aplicaciones de adquisición de datos e instrumentación virtual. El tiempo de desarrollo se reduce notablemente en relación a los lenguajes tradicionales, pues no necesitan líneas de código para su programación, ya que están basados en objetos e iconos que tienen propiedades y métodos preprogramados. Estos elementos sólo necesitan interconectarse mediante un diagrama de bloques para interactuar y generar una aplicación, reduciendo notablemente el tiempo de diseño y desarrollo. Así de esta forma, este trabajo está pensado como un sistema de adquisición de datos utilizando un microcontrolador y un lenguaje G, que simplifique y reduzca los tiempos de diseño y desarrollo. La utilización de un microcontrolador con ADC's integrados facilitará y reducirá los tiempos del desarrollo de la parte electrónica del proyecto y permitirá la adición de funciones posteriores, como por ejemplo la manipulación de dispositivos y maquinas desde la PC o mediante un cliente remoto.

3.1 Etapa de adquisición

3.1.1 Sistema electrónico

Es la etapa del componente de hardware que entregará de forma digital los datos de las variables físicas a la computadora. El diseño de esta etapa se basa en sensores de temperatura, humedad e irradiancia solar. Solo para la variable de irradiancia solar fue necesario diseñar e implementar una etapa de acondicionamiento de señal. La parte principal de la etapa de adquisición la compone el microcontrolador. El microcontrolador utilizado fue el PIC16F873 debido a que ya integra el ADC en el mismo chip. En la figura 5 se muestra el diagrama general del sistema electrónico desarrollado en este proyecto.

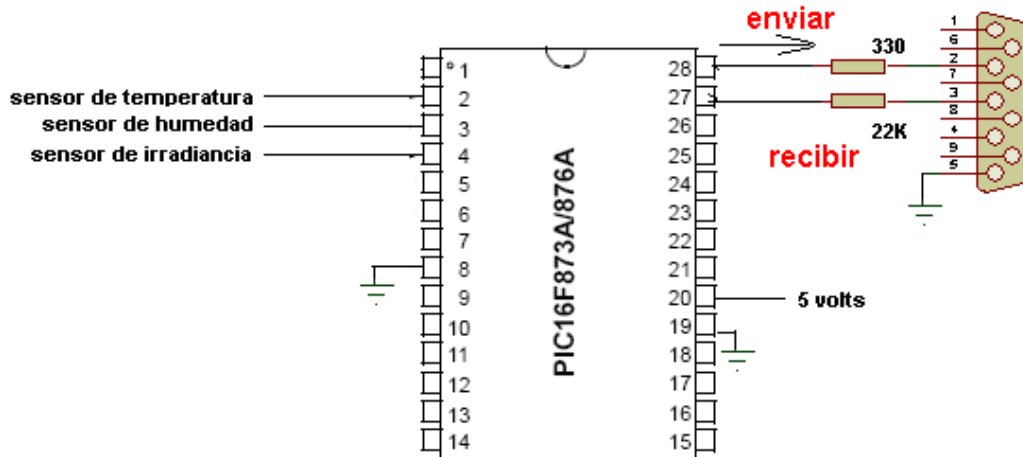


Figura 5. Diagrama electrónico del sistema de adquisición de datos por puerto serial

En general el microcontrolador fue programado para adquirir en los tres canales de datos y enviarlos a la PC de forma serial.

La comunicación serial, se realiza con dos líneas y tierra común, una de las líneas para recibir y la otra para transmitir, la transmisión serial se realiza a 9600 baudios, con un esquema (8, n, 1, n), es decir, datos de 8 bits, sin paridad, un bit de parada y sin control de flujo. Los datos son enviados en una cadena que contiene tres valores ASCII correspondientes a los valores capturados por los ADC del microcontrolador.

3.1.2 Microcontrolador

El microcontrolador fue seleccionado con base en los requerimientos necesarios para el proyecto, principalmente en lo que se refiere al ADC y que incluyera el módulo de comunicación serial asíncrona. Tener un microcontrolador con estas características integradas puede permitir en el futuro crecimiento en las características del sistema como son memorias de respaldo y comparadores.

Se seleccionó el microcontrolador PIC16F873A de la compañía Microchip. Este microcontrolador tiene 28 pines de conexión, 22 canales de entrada/salida y puede trabajar hasta con oscilador de 20 MHz. El PIC16F873A se alimenta con una tensión de 5 V, cuenta con memoria Flash

para programas de 7.2 kbytes, memoria para datos SRAM de 192 bytes y una memoria EEPROM de 128 bytes, esto garantiza memoria suficiente para un crecimiento del 100 % en códigos futuros[11]. La figura 6 muestra un resumen de otras funciones integrada en el PIC16F873A.

- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
 - Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
 - Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
 - Brown-out detection circuitry for Brown-out Reset (BOR)
 - In-Circuit Debug (ICD) via two pins
- CMOS Technology:**
- Low-power, high-speed Flash/EEPROM technology
 - Fully static design
 - Wide operating voltage range (2.0V to 5.5V)
 - Commercial and Industrial temperature ranges
 - Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

© 2003 Microchip Technology Inc.

DS39582B-page 1

Figura 6. Especificaciones del microcontrolador PIC16F873A

En la figura 7 se muestra el encapsulado y la descripción de la función de cada pin del microcontrolador.

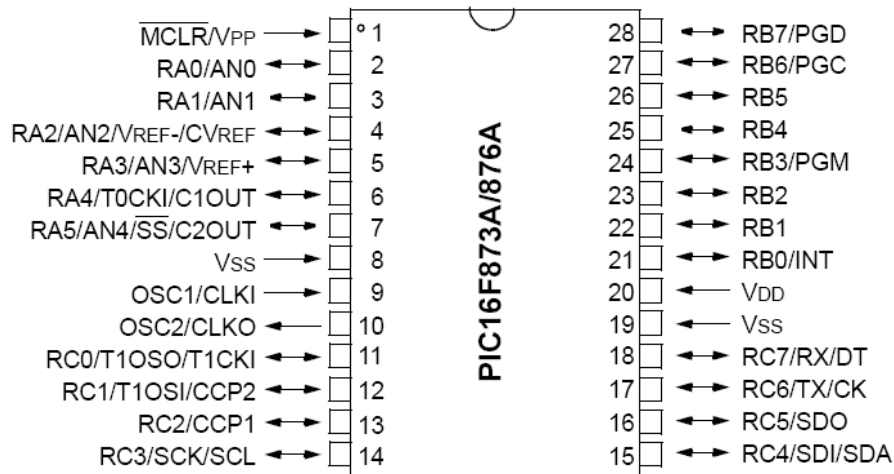


Figura 7. Encapsulado del microcontrolador PIC16f873A

3.1.3 Canales de adquisición

Se seleccionaron 3 canales de adquisición de los 5 con que cuenta el microcontrolador, uno para cada variable física a medir (humedad, temperatura e irradiancia solar). Los canales de adquisición están asignados al puerto A del PIC16F873A y son designados como AN0, AN1, AN2, AN3, AN4 y que corresponden a los pines 2, 3, 4, 5 y 7 respectivamente.

3.1.4 Sensor de temperatura LM35 AH

Para la medición de temperatura se utilizó el sensor de circuito integrado LM35 AH, Este sensor solamente requiere su voltaje de alimentación y genera un voltaje de salida proporcional a la temperatura con una relación de 10 mV/°C. La figura 8 muestra la conexión eléctrica del LM35AH. El sensor LM35AH tiene un rango de medición de -55 °C a +150 °C con una precisión de ± 0.5 °C [10].

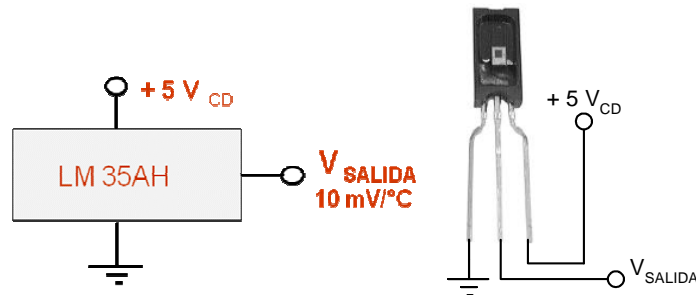


Figura 8. Encapsulado del sensor de temperatura LM35

El valor de la temperatura medida por el sensor está dado por la ecuación 5.

$$\text{Temperatura} = V_{\text{salida}} * (5/255) * 100 \quad (5)$$

3.1.5 Sensor de humedad

En la medición de humedad se utilizó un sensor de la línea Honeywell de la serie HIH-3610 [9]. Este tipo de sensores tienen la ventaja de que se energizan con 5 VCD y proporciona un voltaje de salida que es proporcional

a la humedad relativa evitando con ello la necesidad de acondicionamiento de señal. La señal del voltaje de salida está en el rango del voltaje de alimentación del sensor. En la figura 9, se muestra la conexión eléctrica del sensor de humedad relativa.

Con base en la hoja de datos del fabricante, este tipo de sensores tienen un rango de medición de 0 – 100% HR con una precisión de $\pm 2\%$ HR. El valor actual de humedad relativa se determina mediante la ecuación 6.

$$\%HR = \frac{V_{salida} * 0.86}{0.0320} \quad (6)$$

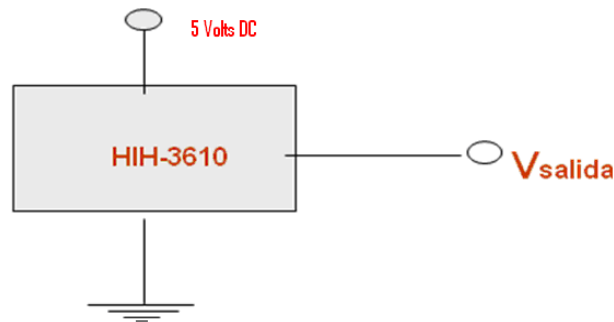


Figura 9. Esquema del sensor de Humedad.

3.1.6 Sensor de irradiancia solar

La medición de la irradiancia solar se realizó con un piranómetro Eppley [9] y un circuito acondicionador conectado a él, la función de la etapa de acondicionamiento de la señal es amplificar el milivoltaje proporcionado por el piranómetro, para poder introducirlo al ADC del microcontrolador como se ejemplifica en la figura 10.

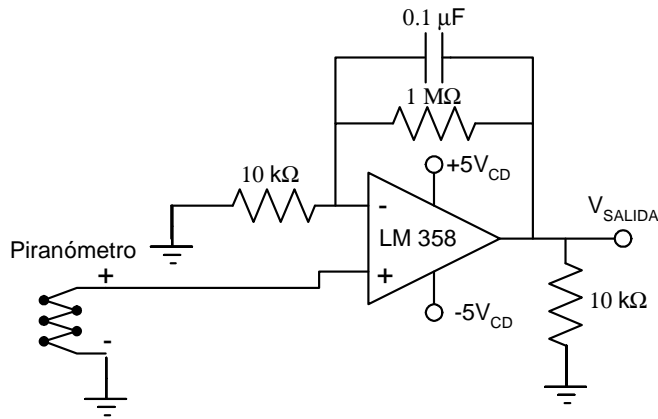


Figura 10. Esquema del sensor de Humedad.

La irradiancia solar cuya unidad es el W/m^2 , se calcula a partir de la ecuación 7.

$$V_{salida} = Irradiancia \times 9.88 \times 10^{-6} \frac{volts}{W \cdot m^2} \quad (7)$$

3.1.7 Comunicación RS-232 a PC

Para la comunicación entre la PC y el microcontrolador se seleccionó el protocolo serial RS-232 por ser uno de los más comunes y fáciles de implementar, además de que esta función ya viene integrada en el PIC16F873A.

El puerto serial es una interfaz de comunicación para ser utilizada entre la PC y dispositivos periféricos, en donde la información es transmitida bit por bit, enviando un solo bit a la vez en forma secuencial, siguiendo un protocolo de Transmisión/recepción, el protocolo que se usará para recibir y transmitir los datos seriales a una velocidad de 9600 baudios es el 8N1, que establece el envío y recepción de datos a 8 bits, sin paridad, un bit de parada y sin control de flujo.

El RS-232, utilizado para el envío de datos binarios consta de dos hilos en comunicación half duplex. La conexión entre la PC y el microcontrolador se realizó mediante un conector DB-9. La figura 11 muestra la configuración eléctrica del conector DB-9.

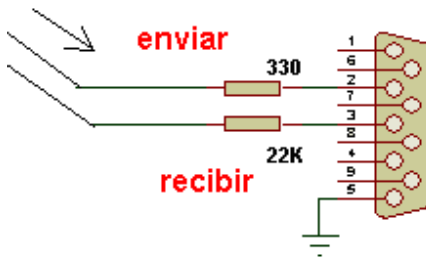


Figura 11. Configuración de pines del puerto Serial

3.2 Lenguaje utilizado

El lenguaje elegido para programar el microcontrolador es PicBasic de la empresa Mecanique inc., debido a que es un lenguaje de alto nivel, fácil de usar, genera código ensamblador y hexadecimal, además de programar el microcontrolador PIC directamente mediante el programador de microchip llamado PICStart Plus, la generación de programas es muy rápido y sin necesidad de utilizar un lenguaje ensamblador.

La herramienta PicBasic, tiene funciones implementadas de comunicación entre dispositivos, escritura en pantalla de cristal líquido (LCD, por sus siglas en ingles, Liquid, Cristal, Display), generadoras de pulsos y señales diversas para recepción y envío por diferentes dispositivos inalámbricos como infrarrojos, ultrasónicos etc. Es muy versátil y además las estructuras de datos de un lenguaje de alto nivel [8].

3.2.1 Algoritmos de la etapa de adquisición

El algoritmo del programa del PIC esta dado en un ciclo infinito que se ejecuta cada segundo. El programa inicia primero recogiendo los datos de los tres canales del microcontrolador correspondientes a las tres variables a medir, cuando el ADC del microcontrolador termina el proceso de conversión de los tres canales, se envían los tres datos en una sola cadena en valores ASCII por comunicación serial a 9600 baudios y regresa de nuevo al principio del ciclo. La figura 12 muestra el diagrama de flujo de este proceso.

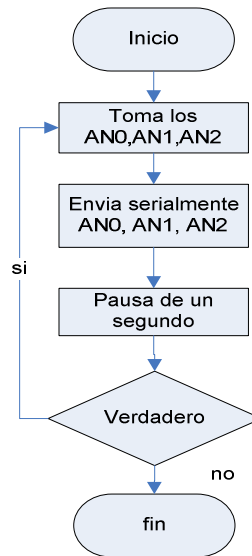


Figura 12. Diagrama de flujo del proceso de adquisición de datos mediante el PIC16F873A

La implementación del diagrama de flujo de la figura 12 en el lenguaje PicBasic puede ser consultada a mayor detalle en el anexo A. El hardware implementado para el sistema de adquisición de datos y los materiales utilizados se detallan en el Anexo B.

3.2.2 Interfaz Gráfica

3.2.3 Descripción de la interfaz gráfica

La interfaz gráfica es el elemento de interacción entre la PC y el usuario final, por lo que debe ser amigable, fácil de utilizar y contener solo los elementos necesarios para su correcto funcionamiento. Al tener una interfaz de este tipo, el usuario solo tiene que seleccionar el elemento (icono) en la pantalla y pulsar en él para iniciar una acción o un proceso, en este tipo de ambientes los elementos pueden personalizarse, para que el usuario trabaje con figuras o iconos acordes a su área de trabajo.

3.2.4 Componentes mínimos aceptables en la interfaz

Este entorno le proporciona al usuario la configuración del puerto de comunicación, la posibilidad de iniciar y detener la aplicación, la visualización numérica de los datos, la visualización gráfica de los datos y la capacidad de generar un archivo separado por comas de los datos recogidos, incluyendo la hora del suceso.

3.2.5 Canales de la visualización gráfica

La aplicación cuenta con tres gráficas continuas de la magnitud con respecto al tiempo de la adquisición, están personalizadas y corresponden a cada canal adquirido, con letreros de especificación de la variable a la que se hace referencia.

También, contiene tres cuadros de texto que visualizan numéricamente los datos de cada canal, para que el usuario, tenga mayor referencia de lo sucedido.

3.2.6 Tiempo de muestreo

El tiempo de muestreo es ajustable, como mínimo se deberá configurar a un segundo, ya que es el tiempo de respuesta de la adquisición en hardware, la aplicación proporciona un cuadro de texto donde se puede modificar el tiempo de muestreo dado en segundos.

3.2.7 Archivo de almacenamiento

La aplicación genera un archivo separado por comas, el cual contiene los datos adquiridos de los tres canales y el tiempo de la adquisición de dicho dato, este tipo de archivos son leídos por Excel y algunos manejadores de datos para tratamientos posteriores.

3.2.8 Controles de arranque y paro

Se tienen dos controles de encendido y apagado de la aplicación, el primero toma las configuraciones de inicio del puerto serial, tiempo de adquisición, nombre del archivo de salida, etc. El segundo, inicia y detiene la adquisición de los datos del puerto serial.

3.2.9 Lenguaje G

Los lenguajes de quinta generación están basados en elementos visuales (iconos) interconectados, los cuales ejecutan acciones de forma secuencial. Cada elemento de la aplicación contiene propiedades y métodos seleccionables para personalizar su funcionamiento, por lo que no necesitan código para generar un programa [12].

3.2.10 Ventajas de los lenguajes G

Un lenguaje G nos permite desarrollar aplicaciones mas rápidamente que los lenguajes de alto nivel, además que cualquier profesionista o investigador puede pasar sus anotaciones o algoritmos gráficos directamente a un programa en este tipo de lenguaje; el estudiante sin muchos conocimientos en programación puede desarrollar aplicaciones basándose en los diagramas de proyectos y acelerar la generación de resultados de trabajos de investigación, por lo que es recomendable para estudiantes, profesionistas e investigadores del área técnica con pocos conocimientos de programación para desarrollar aplicaciones profesionales en las áreas de adquisición de datos e instrumentación [12].

3.2.11 Lenguajes G existentes en el mercado.

En el mercado existen una variedad de lenguajes de este tipo pero los más relevantes por su especialización en desarrollos de instrumentación son:

LabVIEW

Desarrollado por National Instruments, es un lenguaje de programación gráfica, su entorno visual, le permite la programación sin necesidad de código, para la programación se cuenta con dos ventanas, la primera consta de los elementos visuales que aparecerán en la aplicación llamado panel frontal, en el cual estarán los indicadores, gráficas, botones, cuadros de texto de entrada-salida etc., la segunda ventana llamada diagrama de bloques, contendrá el flujo de la información a través de los dispositivos de adquisición, dispositivos periféricos diversos, operadores, flujos remotos etc. y de cómo interactúan entre si para proporcionar las salidas adecuadas, así como las interconexiones con otras aplicaciones, así como elementos de publicación local o a través de la red [12].

LabVIEW ofrece poderosas características que facilitan la conexión a una gran variedad de hardware y otros software. Ofrece a los ingenieros y científicos diseñar instrumentos virtuales a medida, creando interfaces gráficas de usuario en la pantalla de la PC con la cual puede:

- Operar el programa de instrumentación
- Controlar el hardware seleccionado
- Analizar datos adquiridos
- Visualizar los resultados

Se puede personalizar paneles frontales con perillas, botones, visualizadores y gráficos a fin de emular paneles de control de instrumentos tradicionales, crear paneles de ensayo personalizados o representar visualmente el control y operación de procesos. La similitud entre los diagramas de flujo y los programas gráficos acorta la curva de aprendizaje asociada con los lenguajes tradicionales basados en texto [12].

HP-VEE de Hewlett Packard

HP VEE de Hewlett Packard, es un lenguaje de programación gráfica que ofrece un completo ambiente gráfico, también ha elaborado funciones interconstruidas para crear interfaces de usuario, controlar los puertos seriales, etc. Lo cual hace mucho mas fácil la escritura de programas puesto

que toda la programación se hace mediante iconos y recuadros que representan los controladores de instrumentos de medición y los flujos de la información, trabaja exactamente como un diagrama de flujo. El flujo del programa se ejecuta de arriba hacia abajo mientras que los datos pueden viajar de derecha a izquierda o de izquierda a derecha [13].

MATLAB SIMULINK

Simulink es un paquete desarrollado por The MathWorks Inc., que se ejecuta acompañando a Matlab (parte gráfica) para modelar, simular y analizar sistemas dinámicos. Es soportado en ambientes como Unix, Macintosh y Windows.

En simulink los sistemas son dibujados en pantalla como diagramas de bloques, tales como funciones de transferencia, sumadores, uniones, etc., así como entradas y salidas virtuales de aparatos tales como generadores, voltímetros, osciloscopios.

Posee una interfaz de usuario gráfica (GUI, por sus siglas en inglés, Graphical User Interface), con diagramas de bloques para construir los modelos utilizando operaciones de arrastre. De esta forma, se puede dibujar los modelos de la misma forma que se haría con lápiz y papel. Una vez construidos los diagramas de bloques, se puede ejecutar simulaciones y analizar los resultados, también de forma gráfica.

Cuenta con dos fases de uso: La definición del modelo, que significa construir el modelo a partir de elementos básicos construidos previamente, tal como, integradores, bloques de ganancia, etc. El análisis del modelo, que significa realizar la simulación, linealización y determinar el punto de equilibrio de un modelo previamente definido [14].

Agilent VEE

Es un lenguaje de programación desarrollado por Agilent Technologies, para aplicaciones de adquisición de datos para generar dispositivos tales como voltímetros digitales, osciloscopios, generadores de onda, fuentes de alimentación, etc.

Puede representar funciones matemáticas y permite elaborar algoritmos en base a los datos recogidos, todo dentro de un ambiente de programación gráfica, usando bloques tipo diagrama seleccionados desde menús específicos [15].

DAQFactory

Programa desarrollado por AzeoTech. Entre las características principales tenemos: Permite una rápida conmutación entre el modo de ejecución y el modo de programación. Con capacidad para combinar datos leídos en línea y datos almacenados en una base de datos, contiene el algoritmo PID integrado como función, permite comunicaciones de tipo ethernet, modbus, y otras, permite captar eventos del ratón, teclado y otros periféricos, permite importar y exportar texto, imágenes, etc., tiene varias presentaciones o versiones de acuerdo a la potencia, funcionalidad y requerimientos particulares [16].

3.3 Entorno de programación de LabView

El Ambiente del lenguaje de programación gráfico LabView, se compone de dos secciones, la primera llamada panel frontal, como lo muestra la figura 13, consta de los elementos de visualización de resultados y elementos de control de la aplicación. La segunda es llamada diagrama de bloques, como lo muestra la figura 14, la cual consta de los elementos de programación interconectados por líneas de flujo, donde cada uno de estos elementos cuenta con las entradas y salidas necesarias para su funcionamiento, de acuerdo al algoritmo previamente diseñado [7].

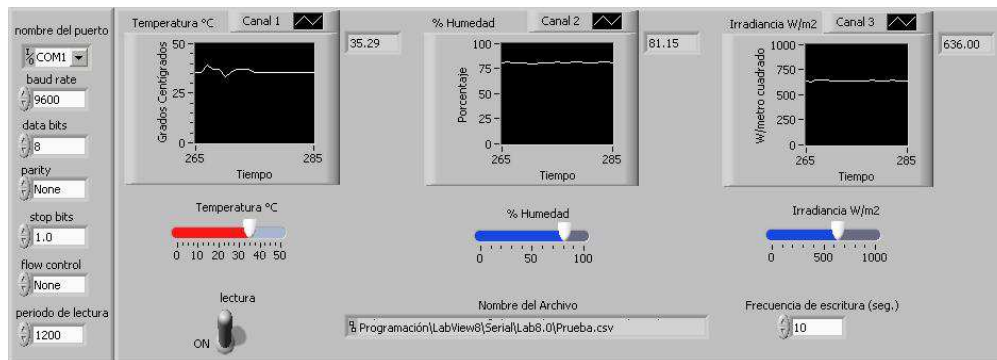


Figura 13. Panel frontal

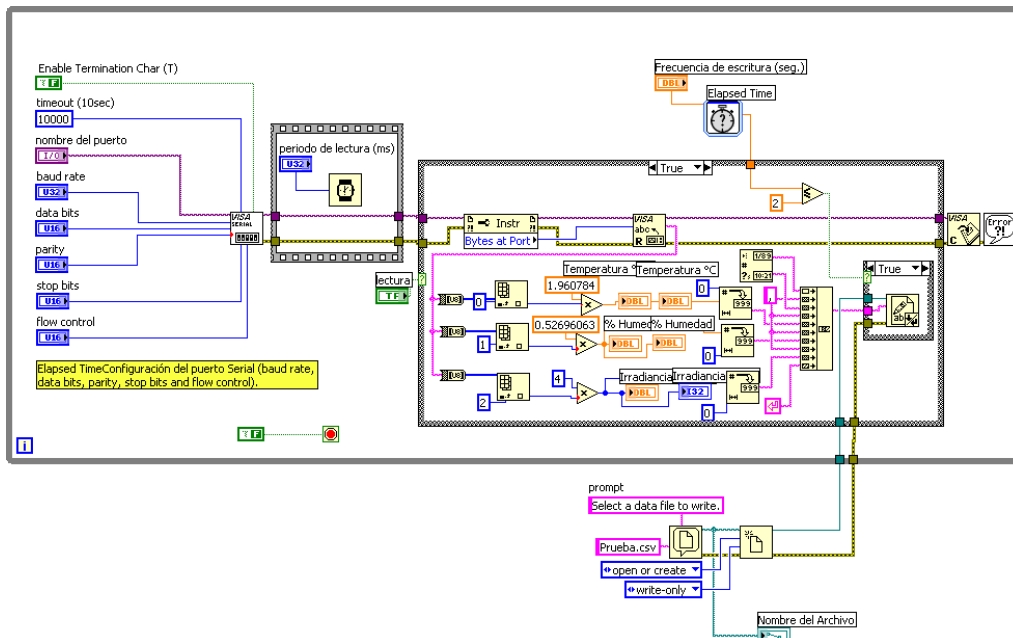


Figura 14. Diagrama de bloques

3.3.1 Diseño del panel frontal

El diseño del entono con el que interactúa el usuario final, consta de cuadros de texto llamados combo box, para seleccionar los parámetros iniciales de la configuración del puerto serial, periodo del tiempo de lectura (presentada en milisegundos) y frecuencia de escritura en el archivo de salida (en segundos). El tiempo mínimo de lectura del puerto es de un segundo, debido a que es el tiempo de respuesta del microcontrolador. Cuenta con gráficas de los datos, valores numéricos e indicadores deslizables de los 3 canales adquiridos. También, el panel frontal cuenta con un control de paro y disparo de la adquisición del puerto, un cuadro de texto que muestra la ruta y nombre del archivo donde se almacenan los datos [7].

La aplicación tiene parámetros de inicio predefinidos, pero pueden cambiarse en modo ejecución para adquirir por otro puerto o reconfigurar el puerto serial, si es que el hardware cambia sus parámetros de envío de datos. Al inicio la aplicación propone un nombre de archivo y ruta, los cuales pueden modificarse, con la observación que si se cambia el nombre del archivo, éste deberá contener su extensión .CSV.

Todos estos elementos están especificados en la figura 15, para su mejor identificación.

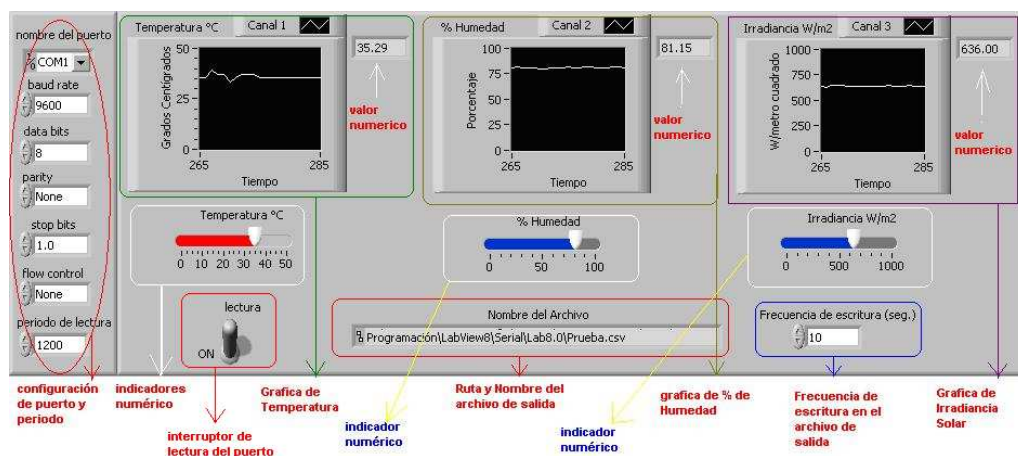


Figura 15. Elementos del panel frontal del sistema de adquisición de datos diseñado

3.3.2 Gráficas

La gráfica continua de temperatura, muestra los datos evaluados en grados centígrados y tiene un registro de 20 segundos, también presenta el valor numérico del valor actual adquirido, como lo muestra la figura 16.

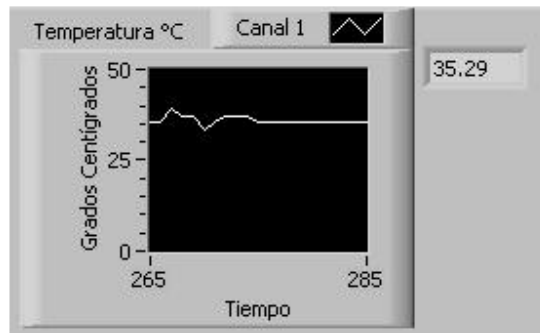


Figura 16. Gráfica de temperatura

La gráfica continua de humedad, muestra los datos evaluados en porcentaje y tiene un registro de 20 segundos, también presenta el valor numérico del valor actual adquirido [7], como lo muestra la figura 17.

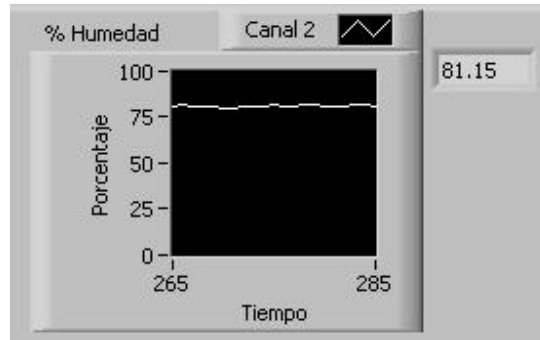


Figura 17. Gráfica del porcentaje de humedad

La gráfica relacionada a la Irradiancia Solar, muestra los datos evaluados en Watts sobre metro cuadrado y tiene un registro de 20 segundos, también presenta el valor numérico del valor actual adquirido, como lo muestra la figura 18.

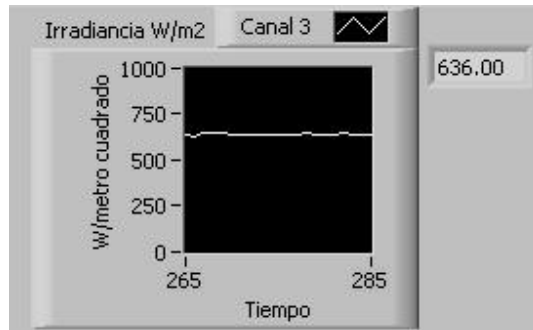


Figura 18. Gráfica de la irradiancia solar

3.3.3 Interruptor

El interruptor de inicio y paro de la adquisición, es un control de dos polos y un tiro [7], el cual se acciona con el ratón sobre él, invirtiendo su estado como se muestra en la figura 19.



Figura 19. Control de paro y disparo de la adquisición del puerto

3.3.4 Algoritmo de la interfaz gráfica

La programación de la interfaz se compone de un ciclo infinito, que realiza la adquisición del puerto serial, procesa los datos y los presenta en los diferentes elementos de la aplicación, dependiendo si la adquisición del puerto esta activa, en caso contrario, espera a que se active la adquisición del puerto como lo muestra el diagrama de flujo de la figura 20.

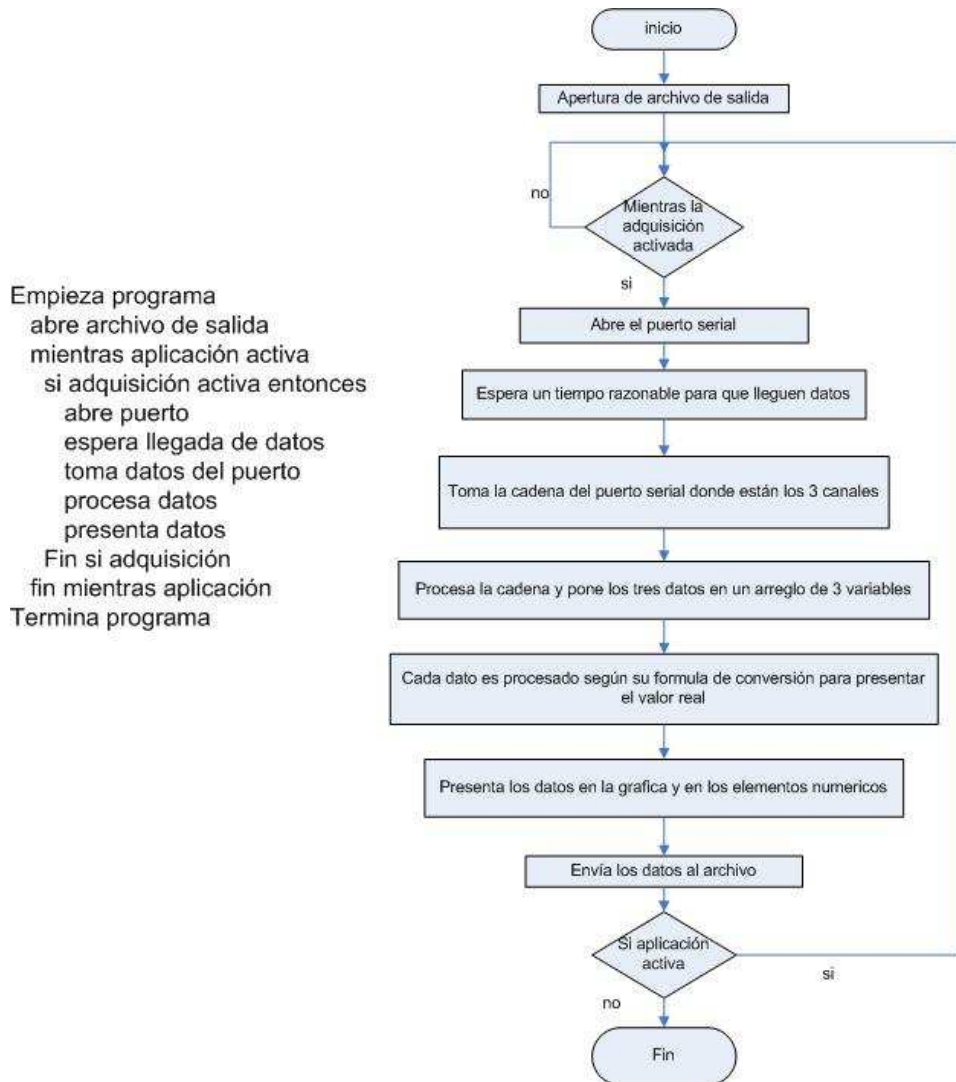


Figura 20. Diagrama de flujo y algoritmo de la interfaz gráfica

3.3.5 Diagrama de bloques de la interfaz gráfica

La programación de la interfaz se compone de un ciclo infinito, que realiza la adquisición del puerto serial, procesa los datos y los presenta en los diferentes elementos de la aplicación (gráfica, numérica y visual), dependiendo si la adquisición del puerto esta activa. En caso contrario, espera a que se active la adquisición del puerto. En la figura 21 se muestra el diagrama a bloques de la interfaz gráfica. La ejecución del diagrama de bloques se realiza de izquierda a derecha, si se observa existen algunos elementos fuera del ciclo infinito por la parte inferior, que son los que configuran el archivo de salida separado por comas (.CSV), por lo que esta parte se ejecutará antes que el resto del programa [7].

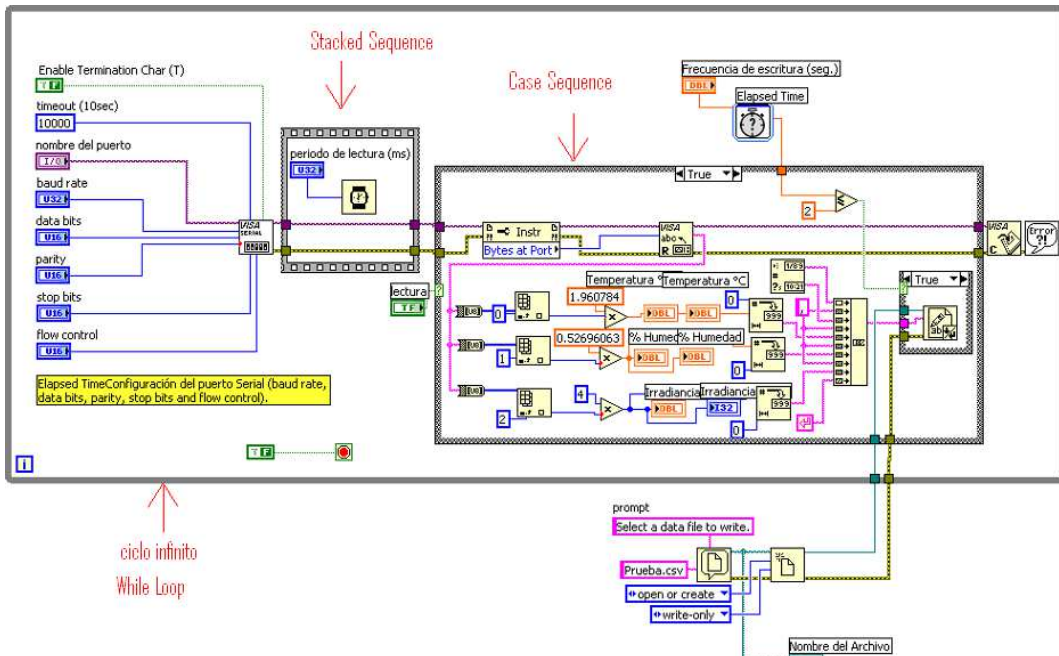


Figura 21. Diseño del diagrama de bloques de la interfaz gráfica, While loop, Stacked sequence, Case sequence.

3.3.5.1 Estructuras de control

La estructura más externa del diagrama a bloques es un ciclo infinito While Loop [7], que necesita una condición verdadera para empezar a ejecutarse como lo muestra la figura 21.

La estructura Stacked Sequence, ejecuta una secuencia controlada por un reloj, que es la que espera un tiempo razonable para asegurar la existencia de datos en el registro del puerto como lo muestra la figura 21.

La estructura Case Sequence [7], es utilizada para la ejecución del resto del algoritmo, pues controla la activación de la adquisición del puerto serial como lo ejemplifica la figura 22, ésta se activa desde el panel frontal con el interruptor descrito en 3.3.2, este interruptor tiene una representación en el diagrama a bloques mostrado en la figura 23.

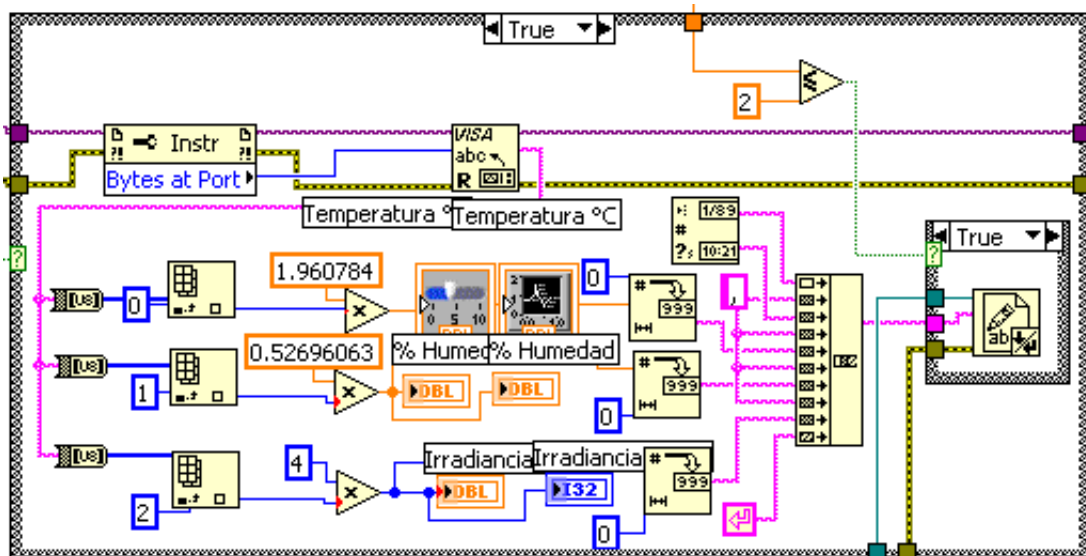


Figura 22. Estructura case "Case Structure", que toma los datos y los procesa

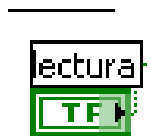


Figura 23. Control booleano(falso o verdadero).

3.3.5.2 Configuración del puerto serial

La configuración del puerto serial necesita tiempo de espera, caracter de terminación, nombre, velocidad, bits de datos, paridad, bits de parada y control de flujo [7]. Estos valores se introducen mediante cajas de texto que tienen la propiedad de contener varias opciones seleccionables. La aplicación desarrollada trabaja en Com1, 9200 baudios, 8 bits, paridad ninguna, un bit de parada y sin control de flujo. Los valores descritos pueden modificarse para cumplir otra configuración de estos parámetros. La gráfica de configuración y sus parámetros se ejemplifican en la figura 24.

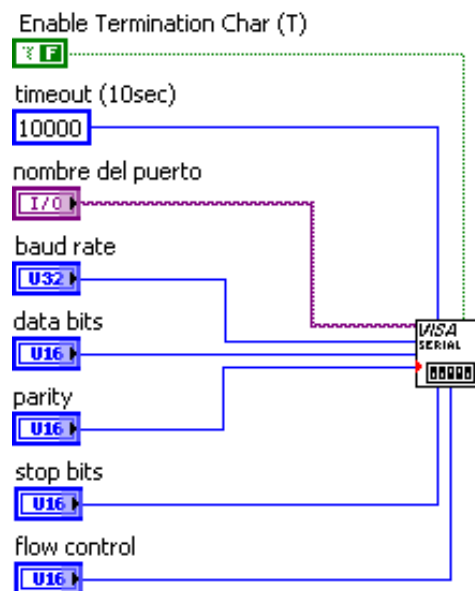


Figura 24. Configuración del puerto serial en el diagrama a bloques

3.3.5.3 Periodo de lectura

Para recoger los datos del puerto serial, se tiene un periodo de espera antes de tomar los datos, esta función esta construido por un icono de secuencia (Stacked Sequence) como lo muestra en la figura 21 y figura 25, un reloj que detiene la ejecución hasta obtener su valor límite como lo ejemplifica la figura 25.



Figura 25. Configuración del periodo de lectura en el puerto serial

3.3.5.4 Instrucción de ejecución de apertura de puerto serial

Después de que se configura el puerto, se necesita ejecutar la operación real de apertura de puerto. Para ello, se emplea el icono de ejecución de instrucción que se muestra la figura 26.



Figura 26. Ejecución de la instrucción de apertura de puerto serial

3.3.5.5 Lectura de datos del puerto serial

La función de lectura de puerto es realizada por un icono de lectura llamado Visa. La figura 27 muestra el icono que realiza esta función en el diagrama a bloques.



Figura 27. Función de lectura del puerto serial

3.3.5.6 Conversión de la cadena en valores numéricos

Los datos leídos del puerto, están contenidos en una cadena, por lo que es necesario convertirlos a formato numérico. Para realizar esta función se utiliza el icono que se muestra en la figura 28, el cual toma la cadena y la convierte en un arreglo numérico.



Figura 28. Función de conversión de la cadena con los tres datos a un arreglo de Bytes

3.3.5.7 Toma de datos del arreglo numérico

Los datos leídos desde el puerto y convertidos a un formato de byte, están contenidos dentro de un arreglo, por lo que es necesario leer cada uno individualmente. Para la lectura del arreglo de datos se utiliza el icono que se muestra en la figura 29. Para la lectura de un dato en particular se especifica su índice, que en el caso del ejemplo que se muestra en la figura 30 es el elemento 0 del arreglo.

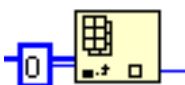


Figura 29. Función que toma un dato específico de un arreglo

3.3.5.8 Tratamiento de datos

Los datos son tratados para su presentación, puesto que para obtener su valor real, cada dato tiene su fórmula de conversión. Para realizar esta acción se utilizan operadores matemáticos como el que se muestra en la figura 30.

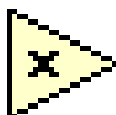


Figura 30. Función que multiplica dos valores

3.3.5.9 Reporte numérico en control deslizable

La presentación de los datos se hace de manera numérica real. Esta función es realizada por el icono que se muestra en la figura 31. Este objeto contiene un método de representación visual de datos. El icono de este

objeto puede minimizar su representación para ahorrar espacio en el diagrama a bloques.

Temperatura °C



Figura 31. Función que visualiza un valor numérico

3.3.5.10 Reporte gráfico y numérico

La presentación de los datos en una gráfica, esta realizada con el objeto representado en la figura 32. Este objeto grafica los datos adquiridos del puerto serial contra el tiempo de adquisición. Además de graficar, este objeto tiene otras propiedades como el de reportar también el valor numérico actual, presentar el valor numérico de la variable física adquirida. También estos objetos pueden minimizar su representación para ahorrar espacio en el diagrama a bloques como se muestra en la figura 21 y su representación normal es apreciada en la figura 22.

Temperatura °C

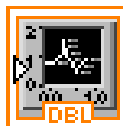


Figura 32. Función que gráfica los valores

3.3.5.11 Conversión de número en cadena

El envío de datos a un archivo de texto se hace enviando una cadena que contiene los tres datos separados por comas. Cada dato debe ser convertido de formato byte a cadena. Esta conversión se realiza con la función ejemplificada con el icono que se muestra en la figura 33. Una vez que los datos estén en formato cadenas se concatenan en una línea de

texto que contiene los datos separados por comas, para ello se utiliza la función representada por el icono de la figura 34. La función de concatenación toma varias cadenas y las integra en una sola; por lo que en la figura 22 se aprecia que a esta función que concatena, entra primero el icono de la figura 35 que regresa la fecha y la hora, después una coma, seguido de los tres datos separados por coma.

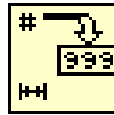


Figura 33. Función de conversión de un número a cadena

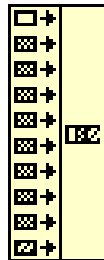


Figura 34. Función de Concatenación de cadenas



Figura 35. Función que regresa la hora y fecha del sistema

3.3.5.12 Escritura de datos en el archivo de texto

La escritura de datos a un archivo se realiza desde el inicio de la aplicación. El icono de la figura 36 representa la función que crea el archivo donde van a ser registrados los datos medidos a un archivo de texto. Posterior a la apertura del archivo, los datos a registrar son enviados al archivo mediante la función que se muestra en la figura 37. La función de escritura está condicionada en una estructura controlada por un reloj que controla la frecuencia de escrituración. Esta función de reloj esta representada en el diagrama a bloques por el icono de la figura 38. La

frecuencia de escritura es definida por el usuario, que posteriormente es comparada con la frecuencia del reloj. Dicha comparación es realizada por la función de la figura 39.

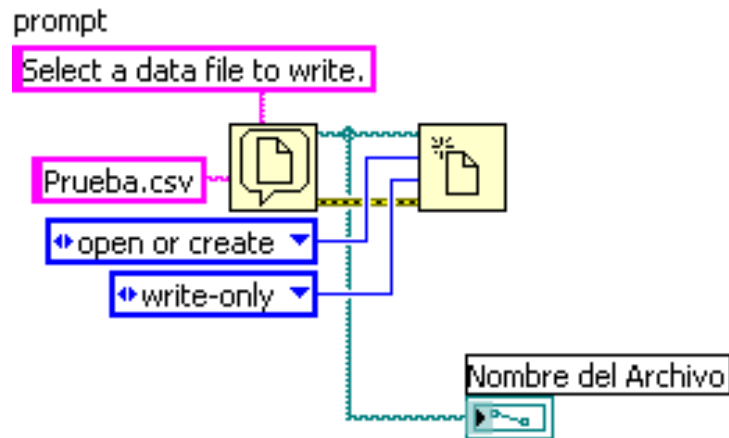


Figura 36. Función de apertura de archivo de escritura

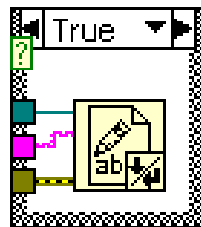


Figura 37. Función de escritura de datos al archivo

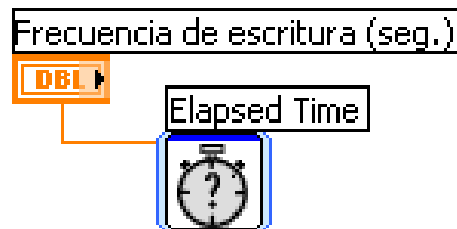


Figura 38. Reloj para evaluar el lapso de escritura

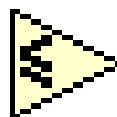


Figura 39. Función de comparación de dos valores (menor o igual)

3.3.5.13 Final del ciclo infinito

Cuando se termina el ciclo de la aplicación, se cierra el archivo de escritura con la función de la figura 40 y se reporta cualquier error en la toma de datos con la función representada por la figura 41.

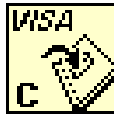


Figura 40. Función de cierra el puerto serial



Figura 41. Función de reporte de errores en el puerto serial

4 Resultados experimentales

4.1 Monitoreo y registro de temperatura, humedad relativa e irradiancia solar

El sistema de adquisición y visualización fue configurado para registrar los datos de humedad relativa, temperatura e irradiancia solar con un intervalo de 1 minuto. Estos datos son almacenados y presentados por la aplicación.

En las figuras siguientes se presentan las gráficas del comportamiento de las variables durante un día de muestreo. El muestreo inicia a las 5:00 AM y finaliza a las 00:00 del día siguiente, con un intervalo de muestreo de 1 minuto. De esta forma, la figura 42 muestra los datos registrados por el sistema de adquisición de datos para la irradiancia solar. En la figura 43 se muestran los datos de la temperatura y por ultimo en la figura 44 se muestran los datos de la humedad relativa.

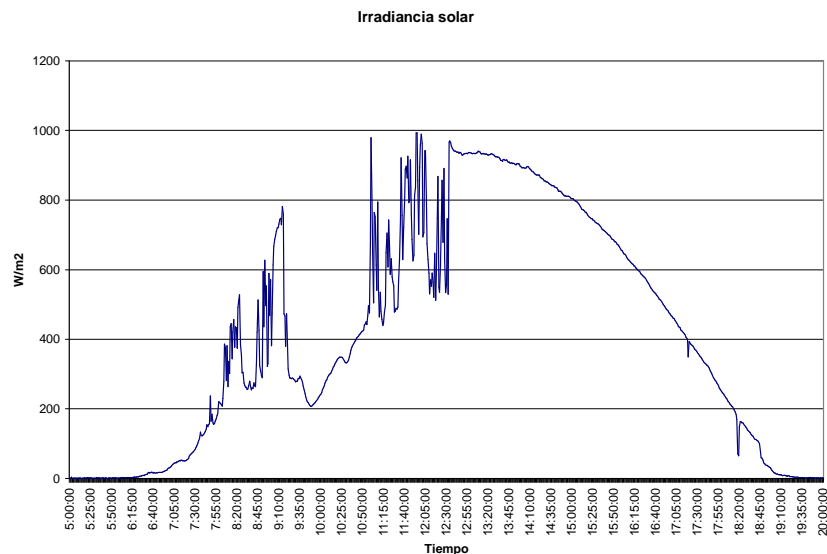


Figura 42. Gráfica de la Irradiancia solar registrada durante el día

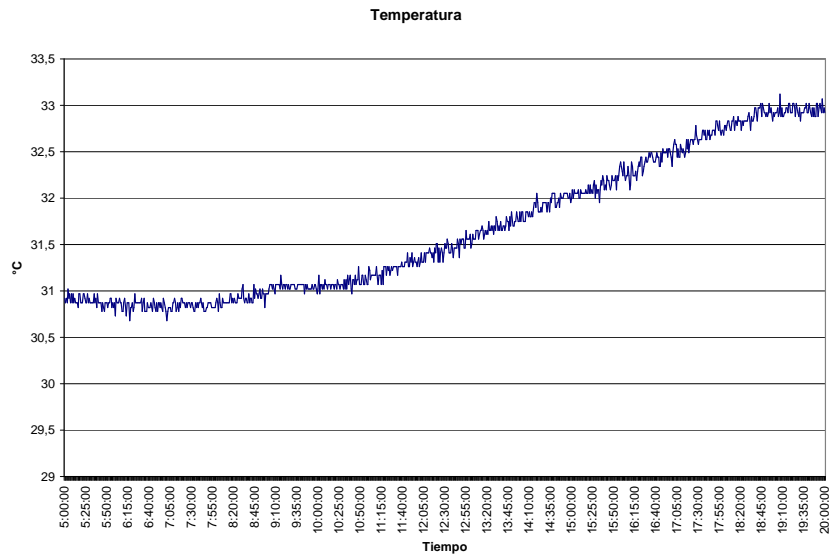


Figura 43. Gráfica de la temperatura durante el día

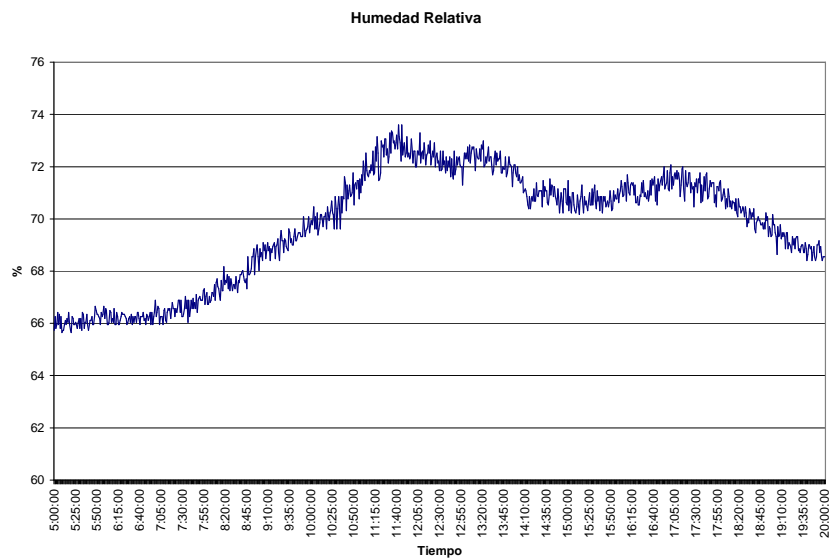


Figura 44. Gráfica de humedad relativa registrada durante el día

4.2 Visualización del funcionamiento interfaz

En la figura 45 se muestra la interfaz gráfica del sistema de adquisición y visualización. En ella se aprecian las 3 gráficas cartesianas correspondientes a cada una de las variables medidas. En la parte izquierda de la interfaz se aprecian los parámetros para la comunicación con el microcontrolador por el puerto serial. En la parte inferior de la gráfica se muestra la caja de dialogo del archivo y la ruta en el disco duro en donde se localiza; así como la caja de dialogo donde se configura la frecuencia de escritura a archivo.

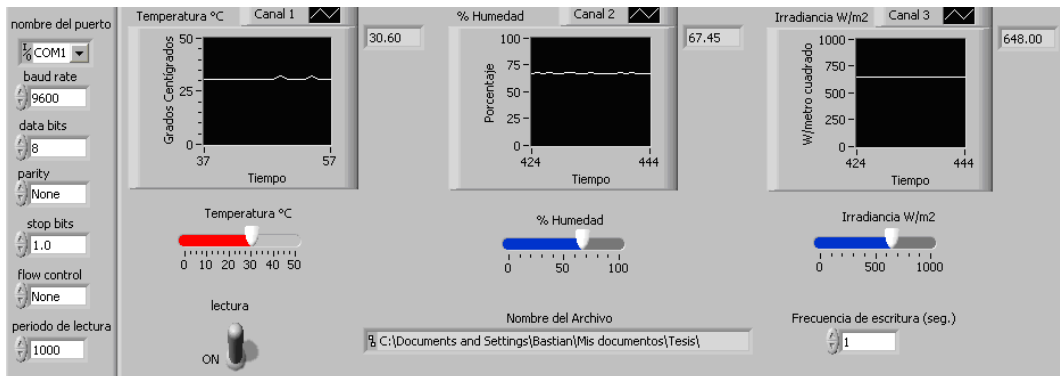


Figura 45. Aplicación en un instante por la mañana

Como se observa en las figura 45, el sistema de visualización y adquisición de datos fue capaz de cumplir con la razón de muestreo seleccionada para las tres variables. La temperatura y la humedad fueron tomadas en un laboratorio con clima y la temperatura empezó a registrarse con el clima encendido y después apagado para ver el comportamiento. De esta forma, vemos que la temperatura tiene un incremento en su valor debido al calentamiento gradual del ambiente y la humedad relativa empieza en un valor que incrementa para finalizar de la manera que empezó el registro.

Para la irradiancia solar el día estuvo un poco nublado por la mañana, por lo que se notan interrupciones durante las primeras horas, un máximo al medio día y un decremento gradual conforme avanzaba el reloj, teniendo un comportamiento aceptable.

5 Conclusiones

Se ha presentado el desarrollo e implementación de un sistema de adquisición de datos desarrollado con un microcontrolador y LabView en una PC, la cual recibe los datos convertidos de analógico a digital de manera serial por parte de un microcontrolador. La principal característica de este sistema de adquisición es que opera de manera ininterrumpida almacenando y presentando los datos de manera numérica y gráfica, con la posibilidad de modificar los parámetros de operación. Dicho sistema de manera opcional y ya probada, puede presentar los datos de manera remota, ya que LabView puede publicar la aplicación en un servidor WEB, el cliente remoto solo necesitara acceder al servidor y pedir la operatividad de la aplicación y tener instalada la maquina virtual LVRunTimeEng 8.0 de National Instruments, para visualizar y operar la aplicación, lo que facilitará al estudiante o investigador tener un seguimiento constante del registro de sus datos desde Internet. Por otra parte, el desarrollo de este tipo de sistema permite generar tecnología propia que puede ser configurada para aplicaciones de monitoreo de forma remota o automatización.

6 Referencias

- [1] Santiago Acha Alegre, Julio Pérez Martínez, Manuel Alonso Castro Gil. "Electrónica Digital: Introducción a la Lógica Digital". Primera edición, 2003, Editorial Alfaomega.
- [2] James W. Bignell y Robert L. Donovan. "Electrónica Digital". Segunda reimpresión, 1999. Editorial Continental.
- [3] Enrique Mandano. "Sistemas Electrónicos Digitales". Séptima edición. Editorial Alfaomega.
- [4] Tomas L. Floyd. "Fundamentos de Sistemas Digitales". Editorial Prentice Hall.
- [5] Palacios Euskalduna. "Mejora de Laboratorios Tradicionales mediante el uso de instrumentos virtuales", Junio 2006. Sitio Web: www.virtualeduca.org
- [6] Henry Antonio Mendiburu Díaz. "Instrumentación Virtual Industrial". Julio 2006. INDECOPI-PERÚ. Sitio Web: <http://hamb.galeon.com>.
- [7] LabView Basics I Course Manual. National Instruments Corporation. Septiembre 2000. Sitio Web: www.ni.com
- [8] Compilador de PicBasic Pro. MicroEngineering Labs, Inc. Traducción: Luís Frino. Sitio web: www.frino.com.ar
- [9] Honeywell
<http://content.honeywell.com/sensing/prodinfo/humiditymoisture/>
- [10] National Semiconductor
<http://www.national.com/pf/LM/LM35.html>
- [11] Microchip technology Inc.
<http://www.microchip.com>
- [12] National Instruments Inc.
<http://www.ni.com>
- [13] HP-VEE
<http://www.evaluationengineering.com/expo/hpvee.htm>

[14] The MathWorks.
www.mathworks.com/products/simulink/

[15] Agilent Laboratories.
www.home.agilent.com

[16] AzeoTech.
www.audon.co.uk/daqfactory.html

7 Anexo A

7.1 Programación en PicBasic del microcontrolador

Para programar en PicBasic se puede hacer un archivo con el código desde cualquier editor de texto y después compilar con el MicroCode o bien editar desde el ambiente integrado de MicroCode Studio. Este ambiente genera tres archivos, el código en lenguaje ensamblador, el código en lenguaje hexadecimal y el código en lenguaje binario.

7.2 Descripción general de los comandos usados

Los primeros comandos configuran un LCD de la marca Hitachi, que fue utilizado para verificar los datos enviados al puerto, la segunda sección tiene la configuración con la que operaran los convertidores, después tenemos la incorporación de librerías de comunicación serial y variables utilizadas. La parte final del código está implementado por un ciclo (Loop) que adquiere los tres datos, los presenta en el LCD, los envía serialmente a la PC y por último espera un segundo para regresar a ejecutar de nuevo la adquisición como se puede ver en el código de la sección 7.3 que está documentado para mejor comprensión [8].

7.3 Codificación

Este programa toma los valores de temperatura, humedad e irradiancia solar provenientes de los sensores en los primeros 3 ADC y los envía de manera serial a 9600 baudios por el pin Portb.7, también los pone en LCD de 4 líneas, el canal 0 en pin RA0, canal 1 en pin RA1, canal 2 en pin RA2

```
-----'Definiciones de puertos y datos para el LCD-----
DEFINE LCD_DREG PORTC      'datos del LCD en el puertoC
DEFINE LCD_DBIT 4          'LCD de cuatro bits
DEFINE LCD_RSREG PORTC    'puerto de Reset para LCD
DEFINE LCD_RSBIT 2        'Pin para el reset portB.2
DEFINE LCD_EREG PORTC     'puerto para Activar el LCD
DEFINE LCD_EBIT 3         'Pin de Activación PortB.3
DEFINE CHAR_PACING 1000   ' Tiempo de espaciado entre cada carácter del LCD

-----'Definiciones para los convertidores internos-----

Define ADC_BITS      8      ' numero de bit del dato
Define ADC_CLOCK     3      ' tiempo de trabajo (3=rc)
Define ADC_SAMPLEUS  50    ' tiempo de conversión

-----'Se incluyen las librerías de configuración del puerto serial-----

Include "modedefs.bas"      ' biblioteca de funciones seriales

-----Declaración de variables-----

adval0 var    byte          ' variable para dato convertido
adval1 var    byte          ' variable para dato convertido
adval2 var    byte          ' variable para dato convertido
enviar var    PORTC.0       'envío de datos por Pin 0 del puerto C
recibir var   PORTC.1       'recibo de datos por Pin 1 del puerto C

-----Configuración de modo de trabajo de los puertos-----

        TRISA = %11111111   ' pone PORTA como entrada
        ADCON1 = %0000001   ' pone PORTA como análogo

-----Programa principal-----

loop:                                'ciclo infinito programa principal

        ADCIN 0, adval0      ' lectura del canal 0 del convertidor
        ADCIN 1, adval1      ' lectura del canal 1 del convertidor
        ADCIN 2, adval2      ' lectura del canal 2 del convertidor
        Lcdout $fe, 1        ' Activa el LCD
        Lcdout DEC adval0, adval1, adval2      ' pone los datos en el LCD
        Serout enviar, N9600, [adval0, adval1, adval2] ' envía los datos por serial
        Pause 1000          ' pausa de 1 segundo
        Goto loop           ' termina ciclo

End                                  ' fin de programa
```

8 Anexo B

8.1 Hardware del sistema de adquisición de datos

La implementación del sistema electrónico se basa principalmente en un microcontrolador Pic16F873A, un sensor de temperatura (LM35), un sensor de humedad Honeywell (HIH-3610) y un piranómetro Eppley , como lo muestra el diagrama de la figura 47. Las fuentes de poder necesarias para el circuito y el protoboard, se tomaron del NI ELVIS mostrado en la Ilustración 1.

Los materiales usados fueron:

Materiales:

1. Microcontrolador Pic 16F873A.
2. Sensor de temperatura LM 35 H.
3. Piranómetro Eppley 8-48.
4. Sensor de humedad Honeywell HIH-3610.
5. Resistencias: 22K, 320Ω, 10K, 10M.
6. Capacitares: 0.1μF, 0.022 pF.
7. Conector serial (cable).
8. Botón de presión.
9. Oscilador 20 Mhz.
10. Amplificador Operacional LM 358.
11. Tarjeta para prototipos (ProtoBoard).
12. Una pantalla de cristal líquido LCD TM162AD.

8.2 Fotografías del sistema de adquisición

En la siguiente fotografía de la ilustración 1, se muestra la ejecución de la aplicación en la PC, así como el dispositivo de hardware del sistema de adquisición, armado en el protoboard de un sistema integrado de National Instruments, del cual se tomaron las fuentes de poder.



Ilustración 1. Ventana de aplicación y sistema de adquisición de datos implementado

En la siguiente fotografía de la ilustración 2, se muestra el dispositivo de hardware del sistema de adquisición, mostrando el cable de comunicación serial hacia la PC.

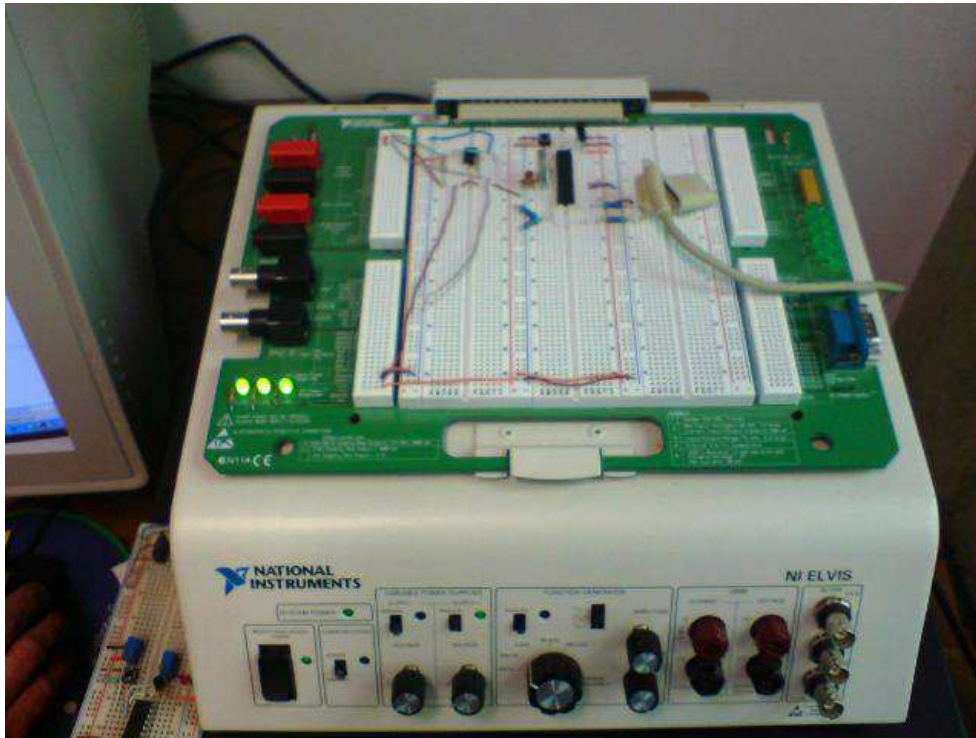


Ilustración 2. Circuito del sistema de adquisición de datos

En la siguiente fotografía de la ilustración 3, se muestra la ejecución de la aplicación en la PC.



Ilustración 3. Sistema de adquisición de datos en operación

En la siguiente fotografía de la ilustración 4, se muestra la ejecución de la aplicación en la PC desde diferente ángulo, para apreciar con mayor énfasis el cable serial.



Ilustración 4 Sistema de adquisición de datos en operación