



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Electricidad y Electrónica

“DISEÑO Y CONSTRUCCIÓN DE UNA ESTACIÓN
METEOROLÓGICA”

Trabajo de Titulación para optar al
Título de Ingeniero en Electrónica

PROFESOR PATROCINANTE
Sr. Jorge Morales Vilugrón

RAÚL EDUARDO FICA JARAMILLO
VALDIVIA - CHILE
2008

Comisión de Titulación

Profesor Patrocinante:

Sr. Jorge Morales V.

Firma: _____

Profesores Informantes:

Sr. Franklin Castro.

Firma: _____

Sr. Pedro Rey C.

Firma: _____

Fecha de examen de titulación:

Agradecimientos y Dedicatorias

Agradezco a todos mis profesores de la comisión de titulación, Don Jorge Morales, Don Pedro Rey y Don Franklin Castro por su ayuda en este trabajo, al Sr. Director de escuela Don Raúl Urra por financiar este proyecto y a todas las personas que de alguna u otra forma me han ayudado a lo largo de mi carrera.

Y dedico este trabajo:

A mi Madre Ana Patricia, por su apoyo y amor incondicional a lo largo de toda mi vida.

A mi Padre Raúl Orlando, por su apoyo durante mi carrera y por su preocupación constante de que vaya avanzando en mi desarrollo profesional y no me quede dormido en el camino....

Y a todo el club de Toby, quiénes me han brindado innumerables momentos de entretención y alegría.

CONTENIDO

AGRADECIMIENTOS Y DEDICATORIAS	III
CONTENIDO	IV
RESUMEN	VI
SUMMARY	VII
INTRODUCCIÓN	VIII
Planteamiento del problema.....	VIII
Objetivos Generales.....	IX
Objetivos Específicos.....	IX
CAPÍTULO I. ESTADO DEL ARTE Y SELECCIÓN DE LA SOLUCIÓN	IX
I.1 Historia de la Meteorología.....	IX
I.2 Estaciones Meteorológicas.....	13
I.3 Análisis de soluciones.....	15
I.4 Selección de la solución.....	17
CAPÍTULO II. DESARROLLO DE LA SOLUCIÓN	20
II.1 Microcontrolador PIC usado.....	20
II.1.1 Arquitectura interna PIC 16F87X.....	25
II.1.2 Organización de la memoria.....	27
II.2 Pantalla LCD utilizada.....	29
II.2.1 Características principales.....	29
II.3 Herramientas utilizadas.....	31
II.3.1 Lenguaje Assembly.....	31
II.3.2 Mplab.....	33
II.3.3 Programa Proteus.....	34
II.3.4 Cargador MCP-USB.....	37
II.3.5 Compilador Visual Basic 6.0.....	38

II.4 Sensores utilizados y descripción de funcionamiento de las distintas etapas.....	42
II.4.1 Dirección del viento.....	42
II.4.2 Velocidad del viento.....	46
II.4.3 Sensor de Temperatura y Humedad SHT 15.....	50
II.4.3.1 Inicio de Transmisión.....	51
II.4.3.2 Secuencia de Medida	53
II.4.4 Nivel de agua caída.....	56
II.4.5 Revisión automática del nivel de batería.....	61
II.4.6 Programa realizado en Visual Basic 6.0.....	63
II.4.7 Programa cargado en microcontrolador PIC 16F877A.....	73
II.4.8 Objetivos Adicionales Logrados.....	76
II.4.8.1 Punto de Rocío.....	76
II.4.8.2 Nivel de luz.....	77
II.4.8.3 Sensación Térmica.....	77
II.4.8.3.1 Índice de calor.....	77
II.4.8.3.2 Índice de enfriamiento del aire.....	78
CAPÍTULO III CONSTRUCCIÓN, PUESTA EN MARCHA Y PRUEBAS.....	80
III.1 Ajustes Finales.....	86
CONCLUSIONES.....	87
GLOSARIO.....	89
REFERENCIAS BIBLIOGRÁFICAS.....	90

RESUMEN

Desde hace muchos años el hombre se ha interesado por los fenómenos climáticos, el interés fue tan grande que se desarrolló una rama de la geofísica que buscaba explicar y medir estos fenómenos, dicha rama fue llamada meteorología. Primero utilizó sensores análogos, pero con el pasar de los años fueron evolucionando hasta llegar a versiones electrónicas, dentro de las herramientas usadas por los meteorólogos, las estaciones meteorológicas juegan un rol demasiado importante, ellas son las encargadas de registrar en terreno la medición de las distintas variables meteorológicas.

El objetivo de este trabajo consiste en el diseño y construcción de uno de estos tan preciados instrumentos, para diseñarlo y construirlo se hizo uso de los conocimientos adquiridos por el autor a través de su viaje por la apasionante carrera de ingeniería electrónica, desarrollando de esta forma un sistema automático de adquisición de datos que además saca el máximo provecho a las virtudes de los microcontroladores PIC.

El proyecto es capaz de medir las variables ambientales de velocidad y dirección del viento, humedad relativa del aire, temperatura ambiental y el diseño teórico de la medición del nivel de agua caída. Además de cumplir con los objetivos propuestos el autor va más allá y logra registrar también la información del punto de rocío, sensación térmica y nivel de iluminación ambiental.

Además el proyecto no se limita a la sola adquisición de datos sino que se comunica con una aplicación en el PC para graficar y guardar toda la información proveniente de la estación en terreno.

En la primera parte del trabajo se analiza el problema, las opciones que lo solucionan y la alternativa elegida. A continuación se explica todo el trabajo realizado para desarrollar la estación, indicando los sensores utilizados, su forma de interacción con el sistema y el software realizado en Visual Basic para la visualización y almacenamiento de los datos.

Luego se presenta información de los objetivos adicionales logrados y por último se dan a conocer las pruebas, ajustes finales, el prototipo terminado y las conclusiones del trabajo.

SUMMARY

For many years the man has been interested in the weather, the interest was so great that developed a branch of the geophysics and measure that sought to explain these phenomena, this branch was called meteorology. First used analog sensors, but with the passing of the years were evolve until electronic versions within the tools used by meteorologists, weather stations play an important role too, they are responsible for registering land in the measurement of various variables.

The objective of this work is to design and build one of these very precious instruments, to design and build it was made use of the knowledge acquired by the author through his journey through the fascinating career of electronic engineering, thus developing automatic data acquisition also brings maximum benefit to the virtues of the PIC microcontroller. The project is capable of measuring environmental variables of wind speed and direction, relative humidity, air temperature and the design theory of the measurement of water level fall. In addition to complying with the proposed objectives, the author goes further and also achieved record information dew point, heat sensation and level of ambient lighting. In addition, the project is not limited to a simple data acquisition but communicates with an application on the PC for graphing and store, all information from the ground station. In the first part of the paper examines the problem, the options that solved and the alternative chosen. He then explained all the work done to develop the station, indicating the sensors used, how they interact with the system and software made in Visual Basic for display and storage of data.

It then presents information on the additional goals achieved and finally reveal the evidence, final adjustments, the prototype completed and the findings of the work.

INTRODUCCIÓN

Desde hace muchos años el hombre se ha interesado enormemente por los cambios climáticos, le ha llamado la atención predecir de alguna manera el estado del tiempo, esto último es útil para tripulantes de embarcaciones, militares, agrónomos y para la planificación diaria de personas en general, todos en algún momento de nuestras vidas nos hemos interesado por saber cómo estará el tiempo para ese día en que teníamos un reunión especial al aire libre, ante esta necesidad del hombre de saber el estado del tiempo, se desarrolló esta rama de la geofísica llamada meteorología, de la mano de ella se fueron inventando uno a uno los diferentes instrumentos meteorológicos. Con el pasar de los años estos sistemas han ido evolucionando hasta el punto de incluir imágenes satelitales con tal de tener herramientas adicionales para un pronóstico más fiable.

El objetivo del trabajo que se presenta a continuación es dar a conocer el desarrollo que se realizó desde su diseño hasta la construcción del prototipo, de una estación meteorológica automática, herramienta fundamental en meteorología, ésta es capaz de medir algunas variables ambientales y además tiene la capacidad de visualizar en forma gráfica los datos obtenidos mediante un software diseñado para tal propósito.

Planteamiento del problema:

El problema general abordado por este proyecto está orientado a buscar una manera eficiente de medir las variables meteorológicas de velocidad del viento y dirección del viento, humedad relativa del aire, temperatura ambiental y nivel de agua caída (esta última sólo teórica), además buscar una forma de guardar los datos a lo largo de los días y meses y poder visualizarlos de forma gráfica posteriormente.

Objetivos Generales:

El propósito general de este proyecto, es resolver el problema de medición de variables ambientales en forma automática, para ello es necesario utilizar las herramientas aprendidas durante mi formación para desarrollar una aplicación mediante el uso de un microcontrolador aprovechando al máximo sus capacidades, como por ejemplo el uso de distintas interrupciones, para lograr una adquisición armoniosa de las distintas variables medidas.

El trabajo no sólo se limita a la programación del microcontrolador, ya que, para dar una solución más completa al problema es necesario desarrollar un software que se comunicará con la estación para que las variables medidas queden guardadas en una base de datos y el usuario pueda visualizar posteriormente en forma gráfica las muestras medidas.

Objetivos Específicos:

- La estación deberá ser capaz de medir la velocidad del viento, la dirección del viento (Norte, Sur, Este, Oeste, Sureste, Suroeste, Noreste y Noroeste), la humedad relativa del aire y la temperatura ambiental.
- Se dejará diseñado en forma teórica un sistema de pluviometría el cual se podría agregar en forma práctica en una nueva versión de la estación.
- Estará equipada con una pantalla LCD, mediante la cual se visualizarán los datos medidos en terreno, éstos últimos se actualizarán cada dos minutos.
- La estación deberá guardar en el PC, un registro de las variables cada dos minutos, obteniendo por lo tanto 720 muestras diarias por cada una de ellas.

- Deberá ser capaz de chequear automáticamente el nivel de batería y avisar mediante una alarma visual y sonora cuando sea necesario cambiarla.
- Se comunicará en forma serial con el computador.
- Poseerá un software en el PC, amigable con el usuario, capaz de realizar gráficos de los datos guardados previamente.
- El software guardará los datos en una base de datos Access y además dará la opción al usuario de exportar los datos a Excel.

CAPÍTULO I. ESTADO DEL ARTE Y SELECCIÓN DE LA SOLUCIÓN

(*)I.1 Historia de la meteorología:

Desde la más remota antigüedad se tiene constancia de la observación de los cambios en el clima, asociando el movimiento de los astros con las estaciones del año y con los fenómenos atmosféricos. Los antiguos egipcios asociaban los ciclos de crecida del Nilo con los movimientos de las estrellas explicados por los movimientos de los dioses, mientras que los babilonios predecían el tiempo guiándose por el aspecto del cielo. Pero el término "meteorología" proviene de *Meteorológica*, título del libro escrito alrededor del año 340 A.C por Aristóteles, quien presenta observaciones mixtas y especulaciones sobre el origen de los fenómenos atmosféricos y celestes. Una obra similar, titulada "*Libro de las señas*", fue publicada por Teofrasto, un alumno de Aristóteles; se centraba más que en la previsión del tiempo, en la observación misma de los fenómenos.

Los progresos posteriores en el campo meteorológico se centraron en que nuevos instrumentos, más precisos, se desarrollaran y pusieran a disposición. Galileo construyó un termómetro en 1607, seguido de la invención del barómetro por parte de Evangelista Torricelli en 1643. El primer descubrimiento de la dependencia de la presión atmosférica en relación a la altitud fue realizado por Blaise Pascal y René Descartes; la idea fue profundizada luego por Edmund Halley. El anemómetro, que mide la velocidad del viento, fue construido en 1667 por Robert Hooke, mientras Horace de Saussure completa el elenco del desarrollo de los más importantes instrumentos meteorológicos en 1780 con el higrómetro a cabello, que mide la humedad del aire. Otros progresos tecnológicos, que son conocidos principalmente como parte del progreso de la física, fueron la investigación de la dependencia del volumen del gas sobre la presión, que conduce a la termodinámica, y el experimento de Benjamín Franklin con el volantín y el rayo. Franklin fue asimismo el primero en registrar de modo preciso y detallado las condiciones del tiempo en base diaria, así como de efectuar previsiones del tiempo sobre esa base.

El primero en definir de modo correcto la circulación atmosférica global fue George Hadley, con un estudio sobre los alisios efectuado en 1735. En los inicios, ésta fue una comprensión parcial de cómo la rotación terrestre influye en la cinemática de los flujos de aire. Más tarde (en el siglo XIX), fue comprendida la plena extensión de la interacción a larga escala tras la fuerza del gradiente de presión y la deflexión causada por la fuerza de Coriolis, que causa el movimiento de las masas de aire a lo largo de las isobaras. La fuerza de deflexión debe este nombre en los primeros años del siglo XIX, con referencia a una publicación de Gaspard-Gustave Coriolis en 1835, que describía los resultados de un estudio sobre la energía producida por la máquina con partes en rotación, como la ruta del agua de los molinos. En 1856, William Ferrel sugirió la existencia de una "célula de circulación" a latitudes intermedias, en las cuales el aire se defleca por la fuerza de Coriolis creando los principales vientos occidentales. La observación sinóptica del tiempo atmosférico era aún compleja por la dificultad de clasificar ciertas características climáticas como las nubes y los vientos. Este problema fue resuelto cuando Luke Howard y Francis Beaufort introdujeron un sistema de clasificación de las nubes (1802) y de la fuerza del viento (1806), respectivamente.

El real punto de cambio fue la invención del telégrafo en 1843 que permitía intercambiar información sobre el clima a velocidades inigualables.

A inicios del siglo XX, los progresos en la comprensión de la dinámica atmosférica llevaron a la creación de la moderna previsión del tiempo calculada en base matemática. En 1922, Lewis Fry Richardson publicó "*Weather prediction by numerical process*", que describía como eliminar las variantes menos importantes de las ecuaciones de la dinámica de fluidos que regulaban los fluidos atmosféricos para permitir encontrar fácilmente soluciones numéricas, pero sin embargo, el número de los cálculos necesarios era muy grande. En el mismo periodo, un grupo de meteorólogos noruegos conducido por Vilhelm Bjerknes desarrolló un modelo para explicar la generación, la intensificación y la disolución de los ciclones a media altura, introduciendo la idea del frente meteorológico y de las subdivisiones de las masas de aire. El grupo incluía a Carl-Gustaf Rossby (que fue el primero en explicar el flujo atmosférico a gran escala en términos de fluido dinámica) Tor Bergeron (el primero en comprender el mecanismo de formación de la lluvia) y Jacob Bjerknes.

En los años 1950, los experimentos de cálculo numérico con computador mostraron ser factibles. La primera previsión del tiempo realizada con este método usaba modelos baroscópicos (es decir, con un único nivel vertical), y podía prever con éxito los movimientos a gran escala de las ondas de Rossby, o sea, de las zonas de baja presión a alta presión. En los años 1960, la naturaleza caótica de la atmósfera fue comprendida por Edward Lorenz, fundador del campo de la teoría del caos. Los avances matemáticos obtenidos en este campo fueron retomados por la meteorología y contribuyeron a estabilizar el límite de predecibilidad del modelo atmosférico. Esto es conocido como *efecto mariposa*: la evolución de los disturbios del tiempo significa un efecto en otra zona. En 1960, el lanzamiento del TIROS-1, primer satélite meteorológico en funcionar, significó el inicio de una era de difusión global de las informaciones climáticas. Los satélites meteorológicos, junto a otros satélites de observación múltiple llegaron a ser instrumentos indispensables para el estudio de una gran variedad de fenómenos, incluyendo incendios forestales y el fenómeno de El Niño.

En los años recientes, se han estado desarrollando modelos climáticos a alta resolución, usados para estudiar los cambios a largo plazo, sobre todo el actual cambio climático.

I.2 Estaciones meteorológicas:

Una estación meteorológica es una instalación destinada a medir y registrar regularmente diversas variables meteorológicas. Estos datos se utilizan tanto para la elaboración de predicciones meteorológicas a partir de modelos numéricos como para estudios climáticos. Está equipada con los principales instrumentos de medición, entre los que se encuentran los siguientes:

- Anemómetro (que mide el viento)
- Barómetro (que mide la presión atmosférica)
- Heliógrafo (que mide la insolación del suelo)

(*) Historia de la meteorología: Extracto de “Meteorología “, Wikipedia.

- Higrómetro (que mide la humedad)
- Piranómetro (que mide la radiación solar)
- Pluviómetro (que mide el agua caída)
- Termómetro (que mide la temperatura)

Estos instrumentos se encuentran protegidos en una casilla ventilada, denominada abrigo meteorológico o pantalla de Stevenson, la cual mantiene la luz solar directa lejos del termómetro y al viento lejos del higrómetro, de modo de no alterar las mediciones de éstos.

Cuanto más numerosas son las estaciones meteorológicas, más detallada y exactamente se conoce la situación. La mayor parte de las estaciones meteorológicas están automatizadas requiriendo un mantenimiento ocasional. Además, existen observatorios meteorológicos sinópticos, que sí cuentan con personal (observadores de meteorología), de forma que además de los datos anteriormente señalados se pueden recoger aquellos relativos a nubes (cantidad, altura, tipo), visibilidad y tiempo presente y pasado. La recogida de estos datos se denomina observación sinóptica.

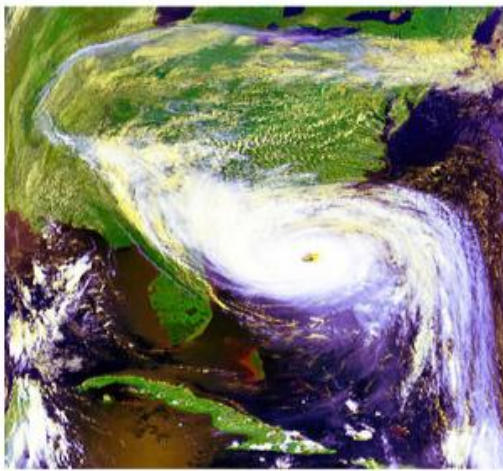
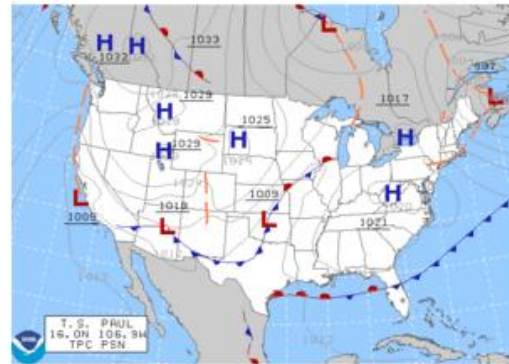


Imagen satelital del huracán Hugo



Mapa sinóptico de EE.UU para el 21/10/06

Fig. 1

Para la medida de variables en mares y océanos se utilizan sistemas especiales dispuestos en boyas meteorológicas. Además existen fragatas meteorológicas, barcos que contienen a bordo una estación meteorológica muy completa y a los cuales se asigna una posición determinada en pleno océano

Otras instalaciones meteorológicas menos comunes disponen de instrumental de sondeo remoto como radar meteorológico para medir la turbulencia atmosférica y la actividad de tormentas, perfiladores de viento y sistemas acústicos de sondeo de la estructura vertical de temperaturas. Alternativamente, estas y otras variables pueden obtenerse mediante el uso de globos sonda.

I.3 Análisis de soluciones:

Una manera de solucionar el problema es tener instrumentos tradicionales (análogos) en terreno que midan las distintas variables y tener una persona encargada de ir anotando varias veces al día con un cuaderno las mediciones realizadas, además esa persona tendría que tener la responsabilidad de graficar los datos obtenidos ya sea mediante un dibujo realizado por el mismo o mediante un software diseñado para ello.

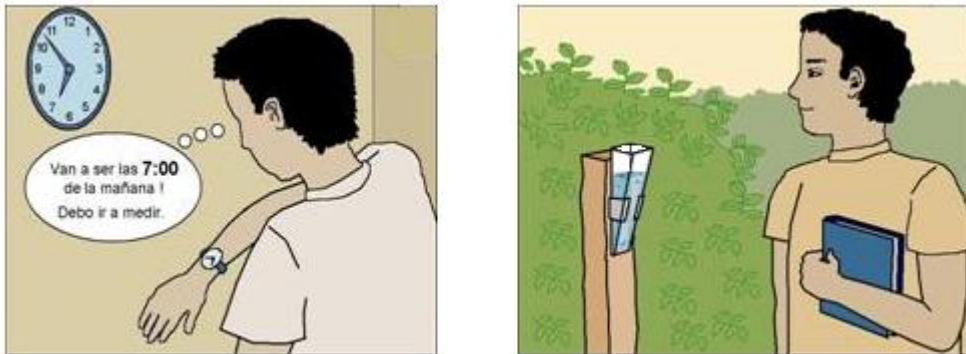


Fig. 2

Otra posible solución sería construir mediante sensores electrónicos un aparato capaz de medir, mediante la electrónica adecuada, en forma automática todas las variables. Además tendría que tener la capacidad de transmitir sin intervención humana los datos a un lugar donde pueda guardarse la información, un lugar razonable y eficiente podría ser un PC de escritorio, en el cual habría instalado una aplicación para el despliegue gráfico de los datos.

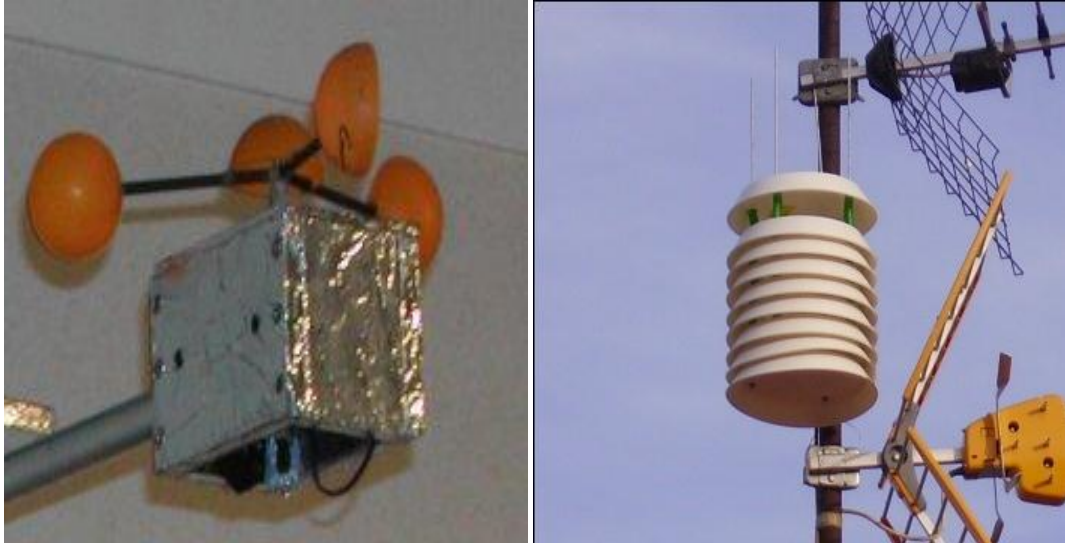


Fig. 3

Una última alternativa sería buscar en el mercado productos ya contruidos que tengan los requerimientos necesarios para solucionar el problema planteado, pero no se abordará ese camino, ya que, lo que se pretende con este proyecto es solucionar el problema poniendo en práctica los conocimientos del alumno y no comprar un producto terminado.

De todas formas algunos de estos productos ofrecidos en el mercado son muy costosos y otros productos podrían ser más económicos que diseñarlo y construirlo uno mismo, pero esto se debe a que los fabricantes de ellos ya han pasado su prototipo por las etapas posteriores que son reducir costos por volumen de producción y por acuerdos en lo referente a descuentos en la adquisición de insumos en grandes cantidades, esta misma etapa debería ser abordada por este proyecto que realizaré, si es que se quisiera pasar del prototipo a un producto comercial y competitivo en el mercado, tema que no será abordado en este trabajo.

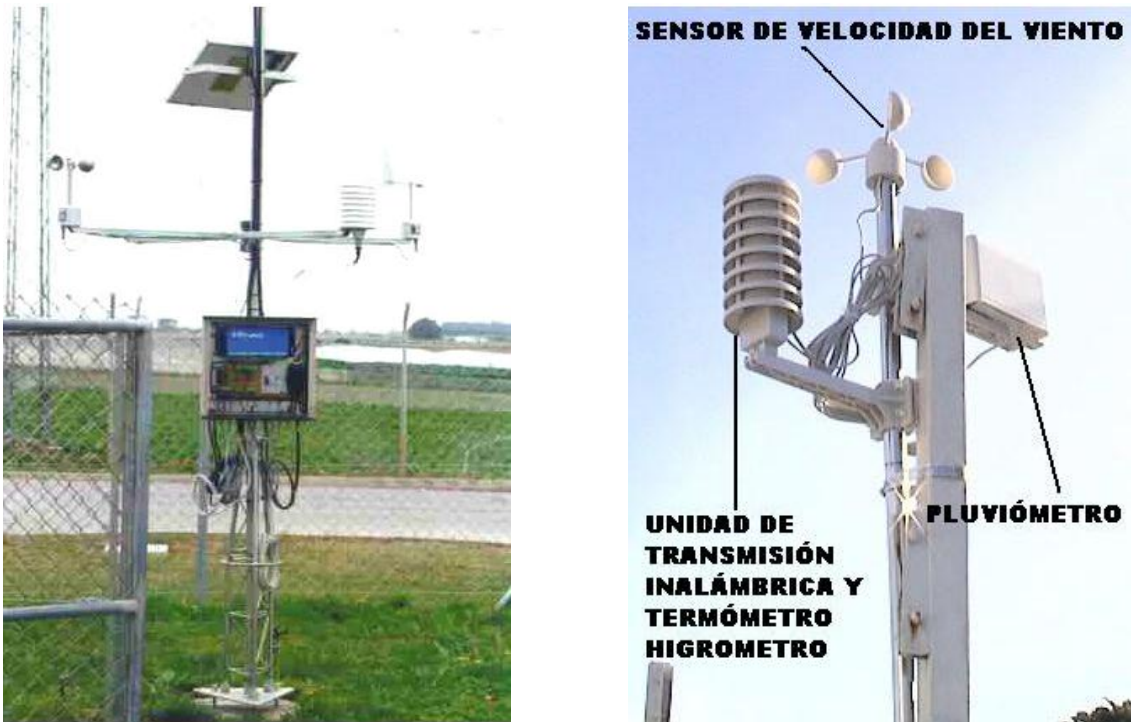


Fig. 4

I.4 Selección de la solución:

La primera solución mencionada es viable pero muy costosa a largo plazo, ya que, tendría un costo mensual permanente, por lo tanto el costo invertido en dicha solución sería extremadamente alto.

Por lo tanto de las tres alternativas mencionadas para dar solución al problema, se optó por la segunda, la de construir un dispositivo electrónico capaz de medir en forma automática las variables ambientales de velocidad y dirección del viento, humedad relativa del aire, temperatura ambiental y medición del nivel de agua caída (esta última sólo teórico) y capaz además de comunicarse con el PC para el registro automático de las variables medidas, debido a su superioridad con respecto a ahorros a largo plazo en comparación con la solución humana.

Ahora que ya se ha tomado la decisión de crear un sistema electrónico para resolver el problema, falta decidir que características tendrá este para que cumpla a cabalidad los requerimientos planteados inicialmente y a un costo razonable.

Con el fin de tener un registro bastante completo de cómo van cambiando las variables ambientales durante el día, el sistema se desarrollará pensando en un número de 720 muestras diarias por variable, o dicho de otro modo el sistema guardará cada dos minutos el valor medido en cada uno de los sensores.

Un buen camino es guardar los datos obtenidos directamente en el PC, así no necesitará de nadie que vaya a buscar la información a la estación y a traspasarla al PC cada vez que la capacidad de memoria de la estación este llena, si hay una conexión continua hacia el PC, los datos se pueden ir guardando automáticamente por meses o años dependiendo de la durabilidad de la batería utilizada, la cual una vez cambiada, el sistema seguirá funcionando correctamente.

Si pensamos ahora cual será la mejor forma para establecer esta conexión continua hacia el PC tenemos dos opciones.

La primera es comunicar la estación al PC mediante un enlace inalámbrico, como un sistema wifi por ejemplo.

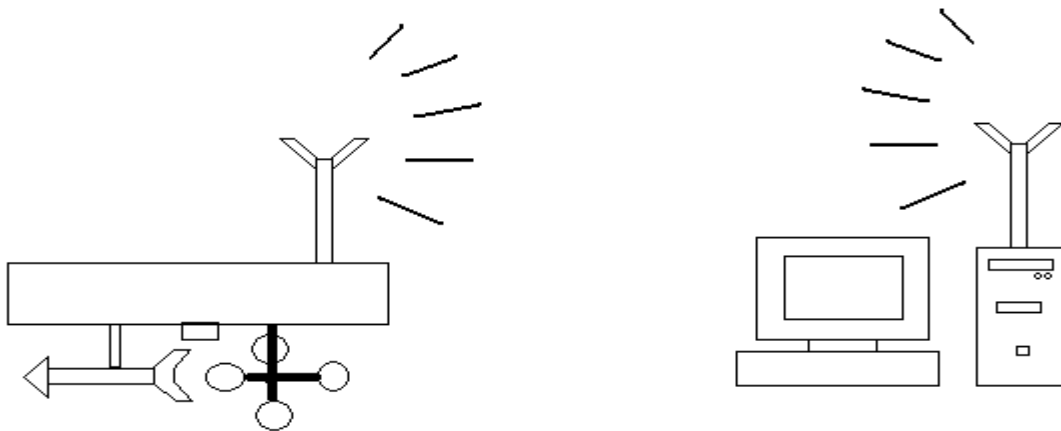


Fig. 5

La otra alternativa es usar una conexión continua mediante cables de cobre.

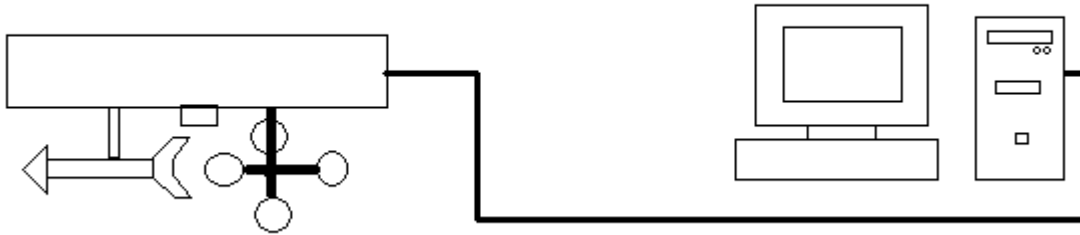


Fig. 6

De estas dos posibilidades la segunda es evidentemente más económica, pero para que cumpla con los requisitos, la forma de transmitir la señal debe tener la capacidad de que usando un cable la información pueda llegar correctamente por un tramo de aproximadamente 20 metros. Esta distancia es suficiente para poner la estación a una altura considerable para su mejor desempeño. Tomando esta última característica como requerimiento, un protocolo de comunicación que cumple con lo expuesto anteriormente es la comunicación de tipo serial, la cual por sus características puede transmitir en un tramo como el mencionado eficientemente, además este tipo de protocolo de comunicación es ampliamente usado en microcontroladores y lenguajes de programación, los cuáles serán necesarios para desarrollar este proyecto.

Dentro de los elementos necesarios para realizar este trabajo, se destaca el uso de un microcontrolador PIC para realizar el trabajo de manejo de la información capturada por los sensores y la transmisión hacia el PC, el microcontrolador PIC elegido es un 16F877A, ya que, cumple con todo lo necesario para un desempeño óptimo dentro del sistema.

Además del microcontrolador, es necesario elegir el lenguaje de programación para desarrollar la aplicación que desplegará los datos en forma gráfica en el PC. Lenguajes de programación hay varios, pero encuentro que Visual Basic 6.0 cumple con todo lo necesario para este trabajo, por lo tanto, la programación de la aplicación en el PC la realizaré con este lenguaje, más adelante mencionaré sus características más importantes.

CAPÍTULO II. DESARROLLO DE LA SOLUCIÓN

II.1 Microcontrolador PIC usado:

Casi todos los fabricantes de microprocesadores lo son también de microcontroladores, en el mercado existen una serie de marcas bastante conocidas y reconocidas como es el caso de Microchip, Motorola, Hitachi, etc. He seleccionado a Microchip y en particular la serie 16F87X, motivos para usar este dispositivo sobran, el principal de ellos es la abundante información y herramientas de diseño existente en el mercado (tanto local como internacional). También salta a la vista el hecho que es sencillo en el manejo y contiene un buen promedio elevado en los parámetros (velocidad, consumo, tamaño, alimentación).

Las principales características con que cuenta el 16F87X son:

- Procesador de arquitectura RISC avanzada
- Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción menos las de salto que tardan 2.
- Frecuencia de 20 Mhz
- Hasta 8K palabras de 14 bits para la memoria de código, tipo flash.
- Hasta 368 bytes de memoria de datos RAM
- Hasta 256 bytes de memoria de datos EEPROM
- Hasta 14 fuentes de interrupción internas y externas
- Pila con 8 niveles
- Modos de direccionamiento directo, indirecto y relativo
- Perro guardián (WDT)
- Código de protección programable
- Modo Sleep de bajo consumo
- Programación serie en circuito con 2 patitas
- Voltaje de alimentación comprendido entre 2 y 5.5 voltios
- Bajo consumo (menos de 2 mA a 5 V y 5 Mhz)

El siguiente diagrama da cuenta de los pines del PIC16F87X:

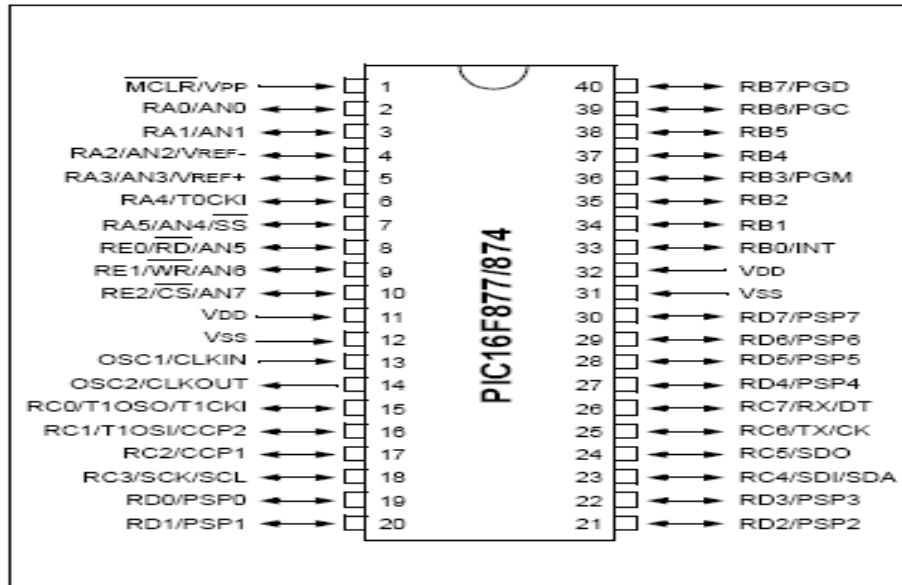


Fig. 7

En la siguiente tabla se muestra la descripción de sus pines:

OSC1/CLKIN(13)	Entrada para el oscilador o cristal externo.
OSC2/CLKOUT(14)	Salida del oscilador. Este pin debe conectarse al cristal o resonador. En caso de usar una red RC este pin se puede usar como tren de pulsos o reloj cuya frecuencia es 1/4 de OSC1.
MCLR/VPP/THV(1)	Este pin es el reset del microcontrolador, también se usa como entrada o pulso de grabación al momento de programar el dispositivo.
RA0/AN0(2)	Puede actuar como línea digital de E/S o como entrada analógica del convertor AD (canal 0).
RA1/AN1(3)	Similar a RA0/AN0.

RA2/AN2/VREF-(4)	Puede actuar como línea digital de E/S o como entrada analógica del conversor AD (canal 2) o entrada negativa de voltaje de referencia.
RA3/AN3/VREF+(5)	Puede actuar como línea digital de E/S o como entrada analógica del conversor AD (canal 3) o entrada positiva de voltaje de referencia.
RA4/T0CKI (6)	Línea digital de E/S o entrada del reloj del timer 0. Salida con colector abierto
RA5/AN4(7)	Línea digital de E/S, entrada analógica o selección como esclavo de la puerta serie síncrona.
RB0/INT(33)	Puerto B pin 0, bidireccional. Este pin puede ser la entrada para solicitar una interrupción.
RB1(34)	Puerto B pin 1, bidireccional.
RB2(35)	Puerto B pin 2, bidireccional.
RB3/PGM(36)	Puerto B pin 3, bidireccional o entrada del voltaje bajo para programación.
RB4(37)	Puerto B pin 4, bidireccional. Puede programarse como petición de interrupción cuando el pin cambia de estado.
RB5(38)	Puerto B pin 5, bidireccional. Puede programarse como petición de interrupción cuando el pin cambia de estado.
RB6/PGC(39)	Puerto B pin 6, bidireccional. Puede programarse como petición de interrupción cuando el pin cambia de estado. En la programación serie

	recibe las señales de reloj.
RB7/PGD(40)	Puerto B pin 7, bidireccional. Puede programarse como petición de interrupción cuando el pin cambia de estado. En la programación serie actua como entrada de datos
RC0/T1OSO/ T1CKI(15)	Línea digital de E/S o salida del oscilador del timer 1 o como entrada de reloj del timer 1
RC1/T1OSI/CCP2(16)	Línea digital de E/S o entrada al oscilador del timer 1 o entrada al módulo captura 2/salida comparación 2/ salida del PWM 2
RC2/CCP1(17)	E/S digital. También puede actuar como entrada captura 1,/salida comparación 1/ salida de PWM 1
RC3/SCK/SCL(18)	E/S digital o entrada de reloj serie síncrona /salida de los módulos SP1 e I2C.
RC4/SDI/SDA(23)	E/S digital o entrada de datos en modo SPI o I/O datos en modo I2C
RC5/SDO(24)	E/S digital o salida digital en modo SPI
RC6/TX/CK(25)	E/S digital o patita de transmisión de USART asíncrono o como reloj síncrono
RC7/RX/DT(26)	E/S digital o receptor del USART asíncrono o como datos síncrono
RD0/PSP0-D7/PSP7 (19-22, 27-30)	Las ocho terminales de esta puerta pueden actuar como E/S digitales o como líneas para la transferencia de información en la comunicación de la puerta paralela esclava. Solo están disponibles en los PIC 16F874/7.

RE0/RD#/AN5 (8)	E/S digital o señal de lectura para la puerta paralela esclava o entrada analógica canal 5.
RE1/WR#/AN6(9)	E/S digital o señal de escritura para la puerta paralela esclava o entrada analógica canal 6.
RE2/CS#/AN7 (10)	E/S digital o señal de activación/desactivación de la puerta paralela esclava o entrada analógica canal 7.
VSS(12,31)	Tierra.
VDD(11,32)	Fuente (5V).

II.1.1 Arquitectura interna 16F87x:

El microcontrolador posee varios elementos en forma interna: el procesador, memoria de programa, memoria de datos, periféricos, contadores. Observemos el siguiente diagrama de bloques del PIC16F87X:

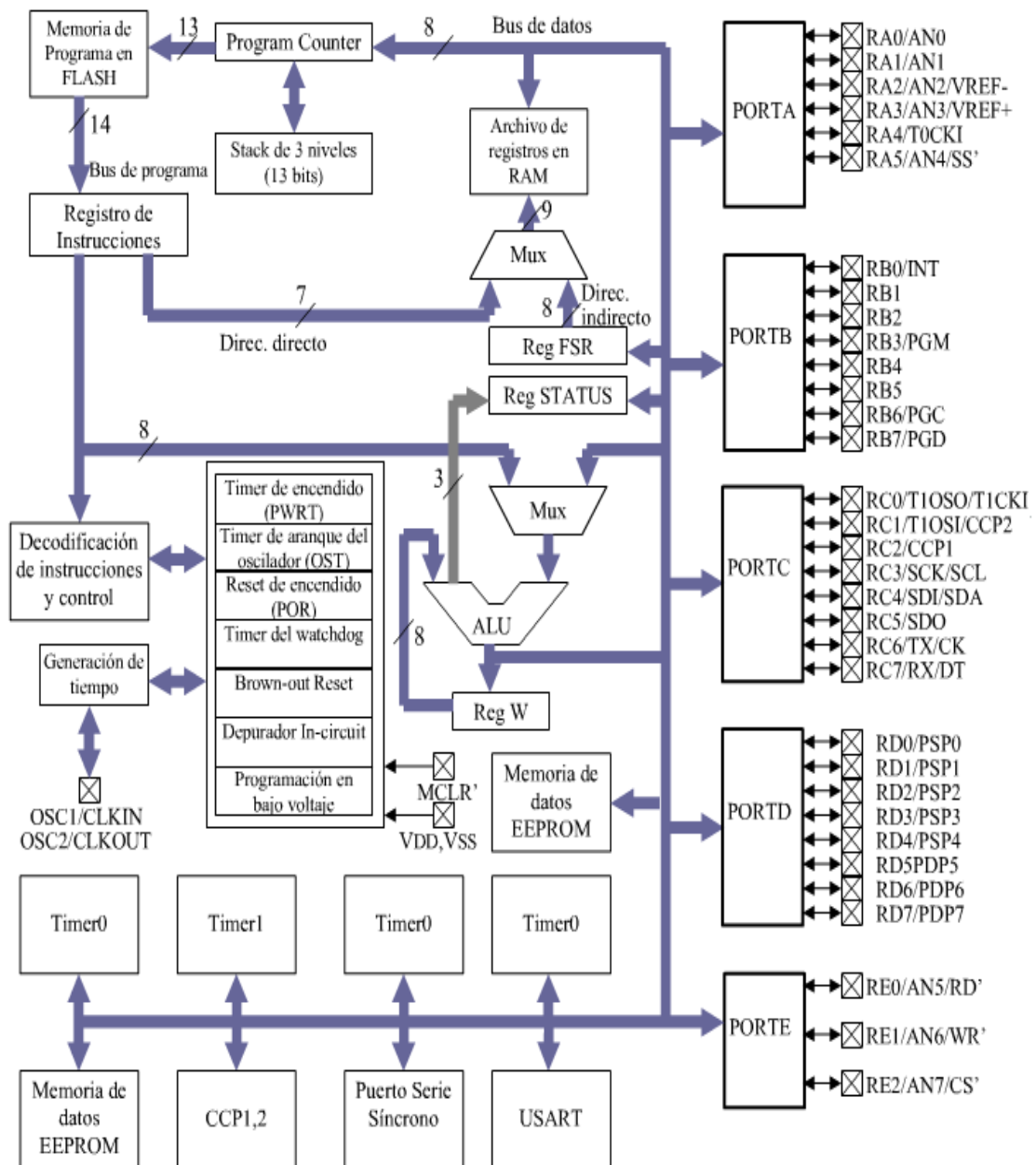


Fig. 8

En el diagrama podemos identificar la memoria del Programa en la parte superior izquierda con 8K posiciones por 14 bits, también esta presenta la memoria de datos (RAM) de 368 posiciones por 8 bits. La memoria EEPROM 256 posiciones x 8 bits. El procesador propiamente dicho esta formado por la ALU (unidad aritmética lógica) el registro de trabajo W. Tenemos los periféricos I/O puerto A, B, C, D, E, el TMR0

(Temporizador contador de eventos), TMR1 y TMR2 entre otros módulos. También contamos con un registro de instrucción que se carga cada vez que la ALU solicita una nueva instrucción a procesar. En la parte intermedia encontramos algunos bloques como son el Status Reg. que es el registro de estado encargado de anotar el estado actual del sistema, cada vez que se ejecuta una instrucción se llevan a cabo cambios dentro del microcontrolador como desborde, acarreo, etc. Cada uno de esos eventos esta asociado a un bit de este registro. Existe un registro de vital importancia que se llama el Program Counter o contador de programa este registro indica la dirección de la instrucción a ejecutar. El registro en cuestión no es necesariamente secuencial, esto es no se incrementa necesariamente de uno en uno ya que puede darse el caso en el que salte dependiendo si hay una instrucción de bifurcación de por medio o puede haber alguna

Instrucción de llamada a función y/o procedimiento. También se observa el bloque de la pila, la función de la pila es ser un buffer temporal en el que se guarda el contador de programa cada vez que se suscita una llamada a un procedimiento y/o función (incluyendo interrupciones). Por tanto el nivel de anidamiento es de hasta 8 llamadas. También esta presente el FSR reg. que es el registro que cumple una función similar a la del contador de programa direccionando en este caso la RAM, el FSR es un puntero a una dirección de la RAM, la aparición de multiplexores se debe a que los datos pueden tener diferentes fuentes. Mas adelante explicamos este punto.

Cuando programamos el microcontrolador debemos siempre tener en mente que es lo que el hace. Cuando lo prendemos asume un valor por defecto, el contador de programa asume la posición cero por tanto el microcontrolador toma la instrucción que se encuentra en esa posición en la memoria de programa y la ejecuta. Al momento de ejecutarla procede a informar si se ha llevado a cabo alguna operación en particular registrándola en el registro de estado (STATUS). Si la instrucción es de salto o bifurcación evaluará las condiciones para saber si continua o no con la siguiente instrucción, en caso que no sea así saltará a otra posición de memoria. En caso el programa haga un llamado a una función guardará en la pila el valor del contador de programa

ejecutará la rutina y al momento que termina restituirá el valor correspondiente para seguir con la siguiente instrucción.

II.1.2 Organización de la memoria:

La memoria se divide en memoria de datos y programa. La de datos a su vez se divide en:

- SFR (Special Function Register) Registros de propósito especial, son registros que ayudan a configurar el hardware interno así como sirven para escribir o leer valores de los diferentes componente que constituyen el microcontrolador. Por ejemplo el registro “TRIS A” que nos permite configurar el modo de trabajo de las líneas del puerto A

-GFR (General Function register) Registros de propósito general, son posiciones de memoria que podemos usar para almacenar valores que emplean nuestros programa

Debe señalarse también que la memoria de datos se divide en cuatro bancos (esto para el caso específico del 16F87X). Las posiciones bajas siempre están reservadas para los SFR en tanto que las altas para los GFR.

También tenemos una memoria EEPROM, con 256 posiciones, para acceder a la memoria no podemos leer o escribir directamente es decir colocar la dirección y obtener o dejar el valor. Para trabajarla debemos apoyarnos en registros adicionales de tal forma que la usamos indirectamente.

A continuación se muestra el mapa de memoria del pic usado:

Indirect addr. ⁽¹⁾	00h	Indirect addr. ⁽¹⁾	80h	Indirect addr. ⁽¹⁾	100h	Indirect addr. ⁽¹⁾	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPAD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
	20h		A0h		120h		1A0h
General Purpose Register		General Purpose Register		accesses			
96 Bytes		96 Bytes		20h-7Fh			
	7Fh		FFh		16Fh		1EFh
Bank 0		Bank 1			170h		1FDh
				Bank 2	17Fh	Bank 3	1FFh

Fig. 9

II.2 Pantalla LCD utilizada:

La pantalla de cristal liquido o LCD (Liquid Crystal Display) es un dispositivo μ Controlado de visualización gráfico para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos), en este caso dispone de 2 filas de 16 caracteres cada una y cada carácter dispone de una matriz de 5x7 puntos (píxeles), aunque los hay de otro número de filas y caracteres. Este dispositivo esta gobernado internamente por un microcontrolador Hitachi 44780 el cual regula todos los parámetros de presentación.

La información de las variables ambientales será visualizada en esta pantalla en la estación y será actualizada cada dos minutos, además la información se podrá visualizar en el software diseñado para eso que estará instalado en el PC.



Fig. 10

II.2.1 Características principales:

- Pantalla de caracteres ASCII, además de los caracteres Kanji y Griegos.
- Permite que el usuario pueda programar 8 caracteres.
- Desplazamiento de los caracteres hacia la izquierda o la derecha.
- Conexión a un procesador usando un interfaz de 4 u 8 bits.
- Movimiento del cursor y cambio de su aspecto.

Para realizar la comunicación entre la pantalla LCD y el microcontrolador es necesario elegir el modo de operación de la pantalla, se tienen dos opciones, una de ellas es usar comunicación a 8 bits y la otra es a 4 bits, la diferencia radica en la cantidad de pines necesarios en el microcontrolador para el control de la pantalla y en el tiempo que se demora en enviar los datos, el modo 8 bits usa mas pines pero es mas rápido que el modo de 4 bits. Se eligió el modo de 4 bits debido a que se ocupan menos pines del pic para la comunicación dejando de esta forma pines disponibles para agregar funciones a la estación en un diseño posterior, sin tener que cambiar el microcontrolador.

Las líneas de datos son triestado, esto indica que cuando el LCD no esta habilitado sus entradas y salidas pasan a alta impedancia. A continuación se muestra una tabla resumen con la función de cada pin de la pantalla:

Pin N°	Símbolo	Descripción
1	Vss	Tierra de alimentación GND.
2	Vdd	Alimentación de +5V CC.
3	Vo	Contraste del cristal líquido (0 a 5 V)
4	RS	Selección del registro de control / registro de datos RS = 0 Selección registro de control RS = 1 Selección registro de datos
5	R / W	Señal de lectura / escritura R/W = 0 Escritura (Write) R/W = 1 Lectura (Read)

6	E	Habilitación del módulo E = 0 Módulo desconectado E = 1 Módulo Conectado
7 - 14	D0 – D7	Bus de datos bidireccional

II.3 Herramientas utilizadas para la programación:

II.3.1 Lenguaje Assembly:

Para programar el microcontrolador existen varios lenguajes, entre los más utilizados están el assembly, el Basic, el Cc5x, el Ccs_c.

A pesar de que el assembly es uno de los más complicados para programar, se eligió esta opción debido a que se tiene un control completo de todo lo que está ocurriendo en la programación, debido a que se programan uno a uno todos los registros y funciones involucradas en los procesos.

En la siguiente tabla se resumen las 35 instrucciones que reconoce la cpu de los pic de medio rango, incluyendo su mnemónico, tiempo de ejecución, código de máquina, y afectación de banderas:

Mnemónico	Descripción	Ciclos	Código de Máquina	Banderas afectadas
Operaciones con el archivo de registros orientadas a bytes				
ADDWF f,d	Suma f + W	1	00 0111 dfff ffff	C,DC,Z
ANDWF f,d	W AND f	1	00 0101 dfff ffff	Z
CLRF f	Limpia f	1	00 0001 1fff ffff	Z
CLRWF	Limpia W	1	00 0001 0xxx xxxx	Z
COMF f,d	Complementa los bits de f	1	00 1001 dfff ffff	Z
DECF f,d	Decrementa f en 1	1	00 0011 dfff ffff	Z
DECFSZ f,d	Decrementa f, escapa si 0	1(2)	00 1011 dfff ffff	
INCF f,d	Incrementa f en 1	1	00 1010 dfff ffff	Z
INCFSZ f,d	Incrementa f, escapa si 0	1(2)	00 1111 dfff ffff	
IORWF f,d	W OR f	1	00 0100 dfff ffff	Z
MOVF f,d	Copia el contenido de f	1	00 1000 dfff ffff	Z
MOVWF f	Copia contenido de W en f	1	00 0000 1fff ffff	
NOP	No operación	1	00 0000 0xx0 0000	
RLF f,d	Rota f a la izquierda	1	00 1101 dfff ffff	C
RRF f,d	Rota f a la derecha	1	00 1100 dfff ffff	C
SUBWF f,d	Resta f – W	1	00 0010 dfff ffff	C,DC,Z
SWAPF f,d	Intercambia nibbles de f	1	00 1110 dfff ffff	
XORWF f,d	W EXOR f	1	00 0110 dfff ffff	Z
Operaciones con el archivo de registros orientadas a bits				
BCF f,b	Limpia bit b en f	1	01 00bb bfff ffff	
BSF f,b	Pone bit b en f	1	01 01bb bfff ffff	
BTFSC f,b	Prueba bit b en f, escapa si 0	1(2)	01 10bb bfff ffff	
BTFSS f,b	Prueba bit b en f, escapa si 1	1(2)	01 11bb bfff ffff	
Operaciones con literales y de control				
ADDLW k	Suma literal k + W	1	11 111x kkkk kkkk	C,DC,Z
ANDLW k	k AND W	1	11 1001 kkkk kkkk	Z
CALL k	Llamado a subrutina	2	10 0kkk kkkk kkkk	
CLRWDT	Limpia timer del watchdog	1	00 0000 0110 0100	TO',PD'
GOTO k	Salto a la dirección k	2	10 1kkk kkkk kkkk	
IORLW k	k OR W	1	11 0000 kkkk kkkk	Z
MOVLW k	Copia literal a W	1	11 00xx kkkk kkkk	
RETFIE	Retorna de interrupción	2	00 0000 0000 1001	
RETLW k	Retorna con literal k en W	2	11 01xx kkkk kkkk	
RETURN	Retorna de subrutina	2	00 0000 0000 1000	
SLEEP	Activa Modo standby	1	00 0000 0110 0011	TO'PD'
SUBLW k	Resta k - W	1	11 110x kkkk kkkk	C,CD,Z
XORLW k	k EXOR W	1	11 1010 kkkk kkkk	Z

Fig. 11

II.3.2 Programa Mplab:

El compilador mundialmente conocido y además gratuito de la empresa Microchip, Mplab fue el que se utilizó para programar en assembly. Mplab es una herramienta para escribir y desarrollar código en lenguaje Assembly para los microcontroladores PIC.

A continuación se muestra su escritorio de trabajo:

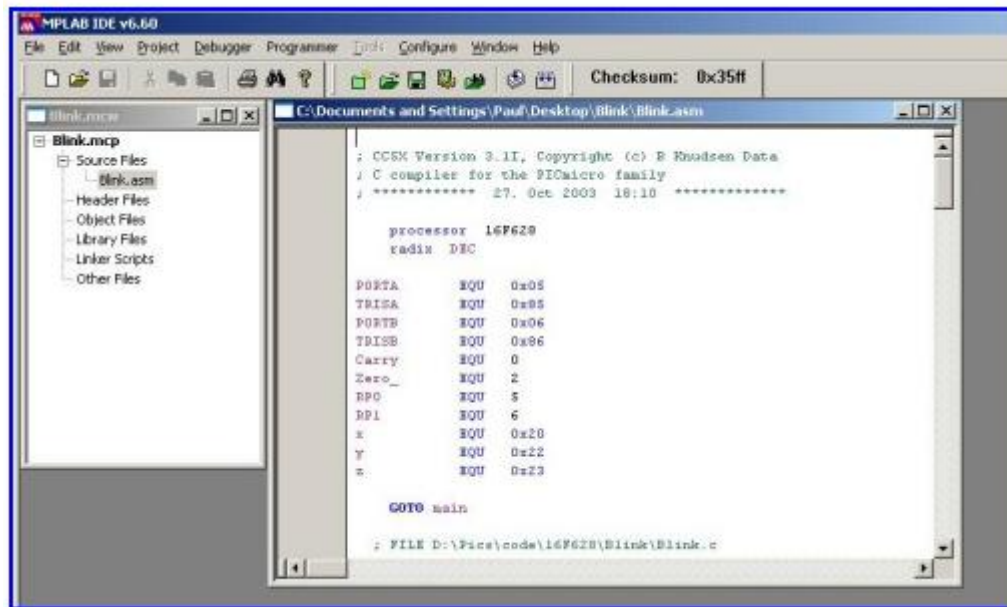


Fig. 12

El MPLAB incorpora todas las utilidades necesarias para la realización de cualquier proyecto, el programa permite editar el archivo fuente en lenguaje ensamblador de nuestro proyecto, además de ensamblarlo y simularlo en pantalla, pudiendo ejecutarlo posteriormente en modo paso a paso y ver como evolucionarían de forma real tanto sus registros internos, la memoria RAM y/o EEPROM de usuario, como la memoria de programa, según se fueran ejecutando las instrucciones.

Dentro de los cargadores que soporta para grabar el microcontrolador está el PIC START PLUS, el cual es equivalente al MCP- USB descrito más adelante, esto quiere decir que el Mplab reconoce al MCP – USB tal como si fuera el PIC START PLUS. El cargador MCP-USB es considerablemente más económico que el PIC STAR PLUS, teniendo las mismas características de funcionamiento.

II.3.3 Programa Proteus:

El software de diseño y simulación Proteus permite el diseño de circuitos empleando un entorno gráfico en el cual es posible colocar los símbolos representativos de los componentes y realizar la simulación de su funcionamiento sin el riesgo de ocasionar daños a los circuitos.

La simulación puede incluir instrumentos de medición y la inclusión de gráficas que representan las señales obtenidas en la simulación.

Lo que más interés ha despertado es la capacidad de simular adecuadamente el funcionamiento de los microcontroladores más populares (PICS, ATMEL-AVR, MOTOROLA, 8051, etc.)

También tiene la capacidad de pasar el diseño a un programa integrado llamado ARES en el cual se puede llevar a cabo el desarrollo de placas de circuitos impresos.

A continuación se muestra el escritorio de trabajo de este programa:

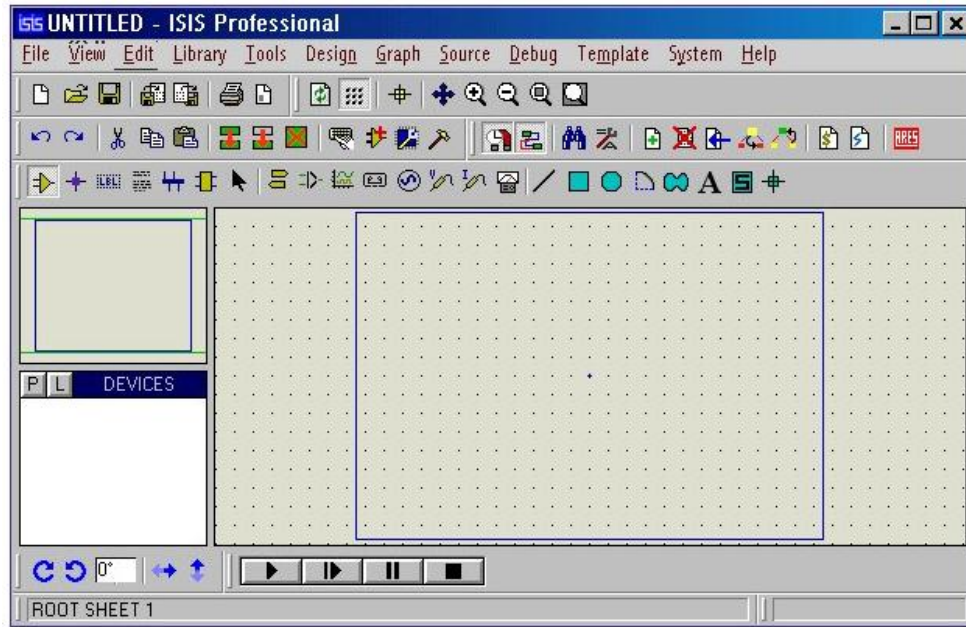


Fig. 13

Los componentes del circuito virtual que se desea armar, están disponibles en las librerías del programa, por lo tanto es necesario indicarle al programa cuáles son los componentes que se necesitan cargar desde la librería.

Para llegar a la librería es necesario dar un click en el botón **Pick Devices** localizado en la parte izquierda de la pantalla debajo de la pantalla de exploración del diagrama para abrir la pantalla del mismo nombre.

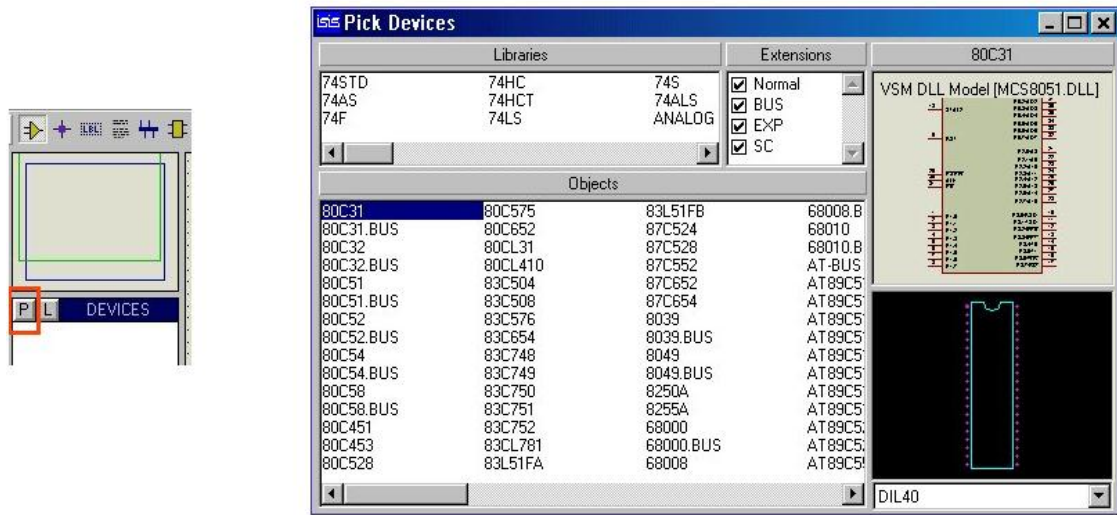


Fig. 14

Una vez que se han buscado y unido todos los componentes necesarios para armar el circuito, se puede apretar play y ver como se comporta el circuito simulado.

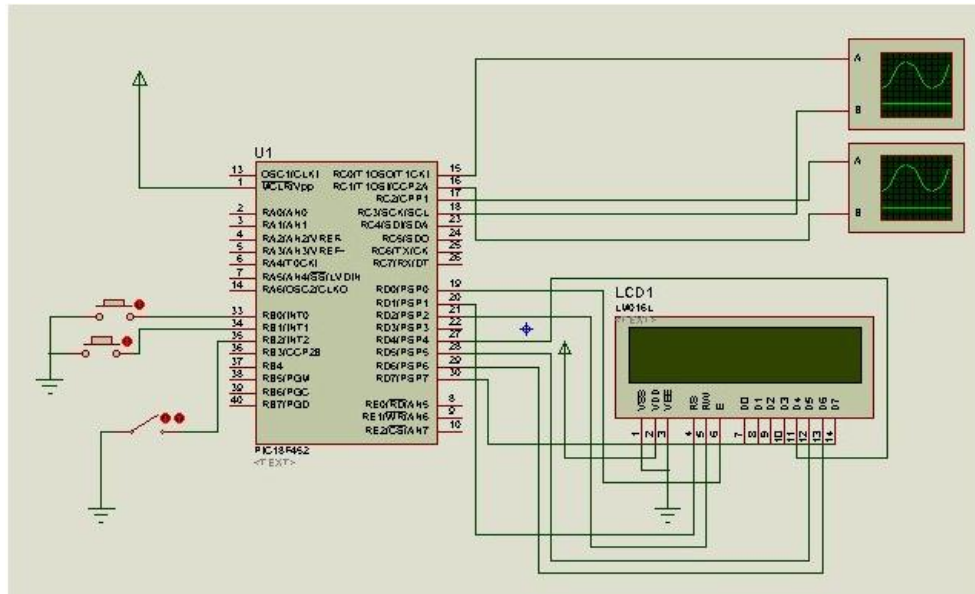


Fig. 15

Debido a que el programa Proteus dentro de sus componentes tiene el pic 16F877A y además tiene pantallas LCD como la usada en el proyecto, fue de gran utilidad en la etapa de diseño, ya que, cada vez que se realizaba un cambio en la programación mediante Mplab, en vez de tener que cargar el programa al circuito real, corriendo el riesgo de cometer un error y dañar algún componente, usaba Proteus para simularlo.

II.3.4 Cargador MCP – USB:

Para lograr traspasar el programa assembler escrito en Mplab, es necesario un hardware el cual cumple la función de grabar el programa al microcontrolador. Existen varios cargadores en el mercado que cumplen esta función pero en este trabajo utilicé el cargador MCP- USB de la empresa Olimex, este cargador no necesita fuente externa, ya que, se alimenta por el puerto USB.



Fig. 16

II.3.5 Compilador Visual Basic:

Para realizar el software en el PC encargado de guardar automáticamente los datos que envía el microcontrolador PIC, utilice el compilador Visual Basic 6.0, el cual es una excelente herramienta de programación que permite crear programas para entorno windows, se puede crear desde aplicaciones sencillas como el caso de una calculadora, hasta programas de alto nivel para ingenierías o cualquier otra área, permite crear ventanas, botones, menús y cualquier elemento de windows de una manera fácil. El lenguaje de programación que utiliza es el Basic.

El programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfase gráfica, es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos.

Dentro de los elementos del entorno de desarrollo de Visual Basic 6.0 se destacan los siguientes:

Barra de Herramientas:

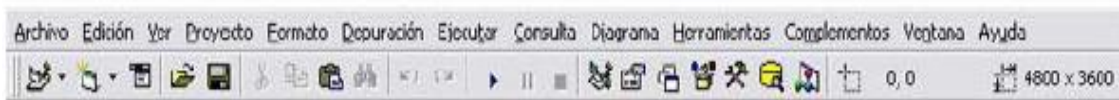


Fig. 17

La barra de herramientas permite un acceso rápido a las funciones más utilizadas del programa.

Diseñador de formularios:

Es una ventana en la cual se diseña la interfaz del programa que uno crea, en esta se pueden agregar controles para la aplicación o incluso imágenes, cada pantalla de la aplicación o formulario posee su propia ventana de diseño.

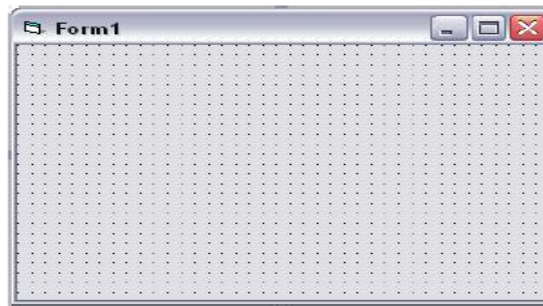


Fig. 18

Cuadro de Herramientas:

En el cuadro de herramientas se puede encontrar un conjunto de herramientas que permiten insertar los objetos o controles en el formulario durante el tiempo de diseño, los objetos más comunes son botones (command), etiqueta (label), cuadros de imagen (Picture), imágenes (Image), etc...

Si se hace doble click con el ratón sobre alguno de ellos se insertará automáticamente en la ventana del formulario.



Fig. 19

Ventana de Propiedades:

Cada objeto usado para la programación tiene propiedades que describen sus atributos tales como los valores, su comportamiento y la apariencia que tiene el objeto dentro de la aplicación.

Dentro de las propiedades más comunes están nombre, backcolor, caption, enable, font entre muchas otras.

La ventana de propiedades se puede visualizar de dos formas, por categoría o alfabéticamente.

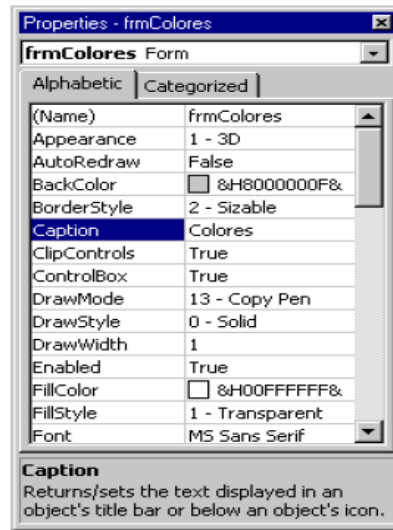


Fig. 20

Ventana de Proyectos:

Contiene la lista de los archivos que forman parte de la aplicación.

Los tipos de archivos que se pueden incluir en un proyecto son:

Archivo de Proyecto: Es el que realiza el seguimiento de todos los ficheros que forman parte de la aplicación. Se guarda en un fichero con la extensión `.VBP`

Archivo de Recursos: Aquí se guardan cadenas de texto, mapas de bits, y demás datos que puedan modificarse sin tener que volver a modificar el código. Se guardan con una extensión `.RES`

Módulo de Formulario: Contiene controles y código, sólo hay uno por formulario. Se guardan con extensión `.FRM`

Módulo de Clase: Son similares a los módulos de formulario. Se guardan con la extensión `.CLS`

Módulo Estándar: Sólo pueden contener código. Tienen una extensión `.BAS`

Controles ActiveX: Controles que se pueden añadir al cuadro de herramientas e incluirlos en un formulario.

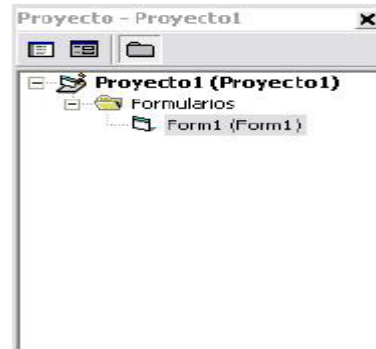


Fig. 21

Ventana Editor de Código:

En esta ventana se escribe el código en Basic, existe una de estas ventanas para cada formulario o módulo de la aplicación.

Para acceder a esta ventana la forma más fácil es hacer doble clic sobre el formulario o sobre el objeto al cual quiera incluir código.

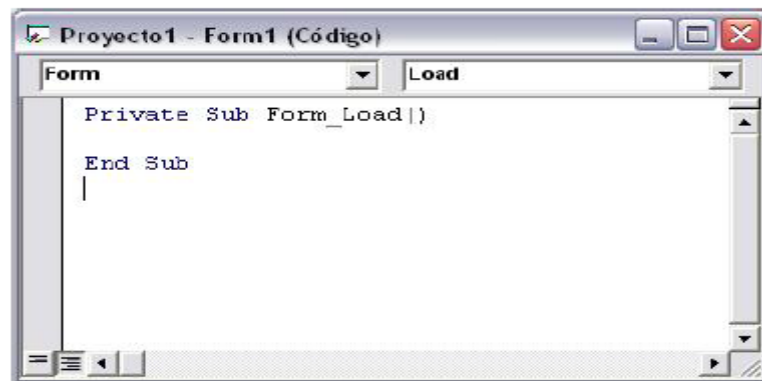


Fig. 22

II.4 Sensores utilizados y descripción de funcionamiento de las distintas etapas:

II.4.1 Dirección del viento:

Para medir la dirección del viento, decidí que una manera conveniente era utilizando sensores infrarrojos. Para lograr medir las ocho direcciones comúnmente utilizadas en meteorología (norte, sur, este, oeste, noreste, sureste, noroeste y suroeste) fue necesario utilizar tres sensores de este tipo.

A continuación se muestra el sensor infrarrojo utilizado y sus principales características.



Fig. 23

- Su salida es un fototransistor.
- No necesita contacto con la superficie sensada (hasta 1 cm de separación).
- Desenfocado para sensar superficies difusas.
- Empaquetado compacto.
- Posee un filtro para la luz de día.

Este sensor utiliza un diodo emisor infrarrojo en combinación con un fototransistor infrarrojo para detectar las señales infrarrojas que rebotan desde la superficie sensada.

Muy efectivo para detectar transiciones negro a blanco o para detectar objetos cercanos (0.5 a 1 cm).

Su característica de poseer un filtro para la luz de día lo hace ideal para esta aplicación ya que el ingreso de luz del exterior no afectará su correcto funcionamiento.

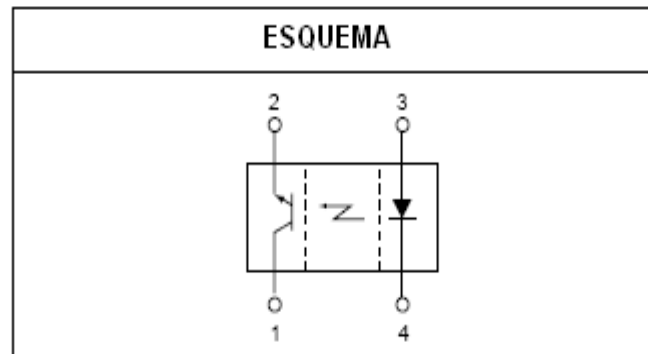


Fig. 24

Su distribución de pines es la siguiente:

Pin 1	Colector
Pin 2	Emisor
Pin 3	Ánodo
Pin 4	Cátodo

El principio de funcionamiento de esta etapa encargada de medir la dirección del viento es que los sensores infrarrojos detectaran la posición de una veleta que estará unida a un eje que girará solidaria a un disco codificado, sobre el cual a una distancia de 0,3 cms aproximadamente, estarán ubicados los tres sensores, los cuales detectaran si bajo a cada uno de ellos se encuentra una zona blanca o negra del disco codificado.

Para que el microcontrolador PIC interprete que lo que esta bajo el sensor es un color blanco o negro, o dicho de otra manera para que lea un nivel lógico alto o bajo proveniente de cada sensor infrarrojo fue necesario agregar un par de resistencias para obtener un voltaje de salida alrededor

de 5 volts para una lectura de color blanco en el disco y un voltaje de aproximadamente 0 volts para una lectura de color negro en el disco.

En la figura inferior se muestra el circuito utilizado para cada sensor infrarrojo.

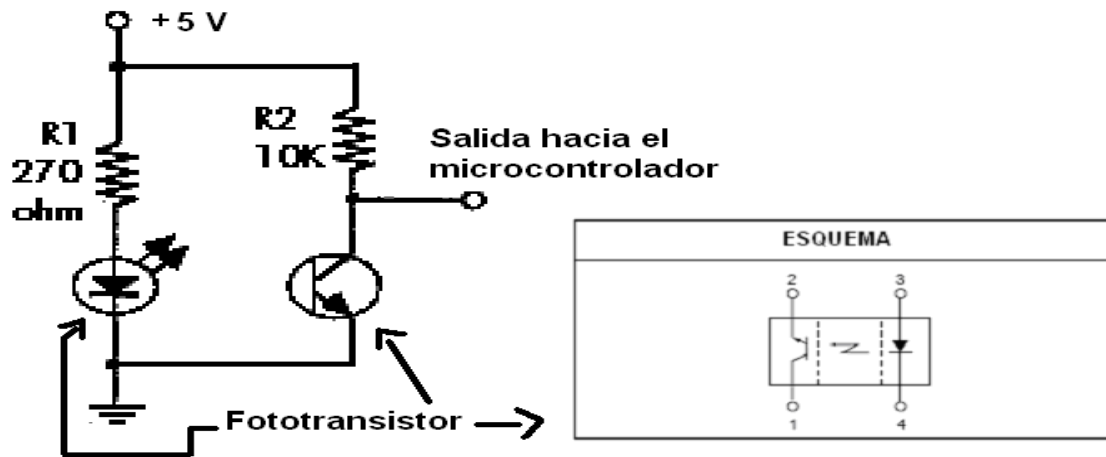
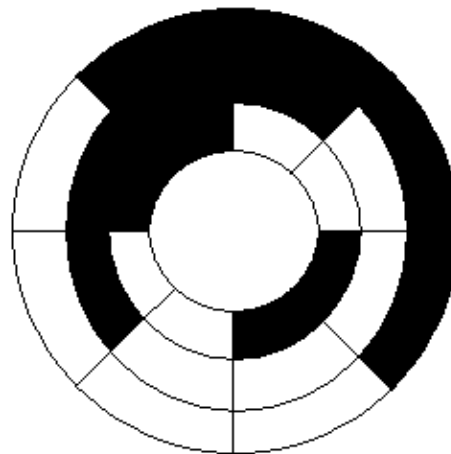


Fig. 25

En la figura inferior se puede apreciar el código de colores blanco y negro en el disco mencionado anteriormente:



Disco Codificador

Fig. 26

En la tabla que se muestra a continuación se observa el código correspondiente a cada dirección del viento.

<i>ANILLO 1</i> <i>(mayor tamaño)</i>	<i>ANILLO 2</i>	<i>ANILLO 3</i> <i>(menor tamaño)</i>	<i>Nº DECIMAL</i>	<i>DIRECCIÓN</i>
0	0	0	0	NORDESTE
0	0	1	1	NORTE
0	1	0	2	OESTE
0	1	1	3	NOROESTE
1	0	0	4	ESTE
1	0	1	5	SUDESTE
1	1	0	6	SUROESTE
1	1	1	7	SUR

Distribución de zonas blancas y negras

Fig. 27

Hay un código único para cada dirección del viento, en la figura siguiente se muestra un dibujo de la forma del mecanismo.

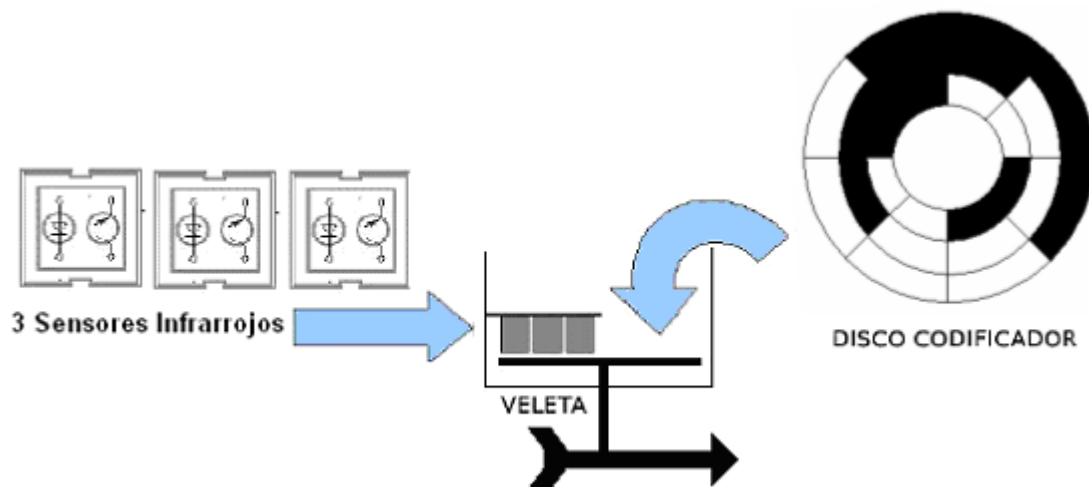


Fig. 28

El microcontrolador continuamente estará actualizando la medida presente en cada uno de los tres sensores infrarrojos y actualizando la dirección del viento en la pantalla LCD, además enviará esa información al PC cada dos minutos cuando éste la solicite junto a los datos de las otras variables.

II.4.2 Velocidad del viento:

Para lograr medir el viento construí un anemómetro con cuatro semiesferas adheridas a los extremos de cuatro brazos metálicos, los cuales estaban unidos a un eje que gira libremente cuando el viento incide sobre el mecanismo sin importar de donde provenga.

La separación angular entre cada brazo con una semiesfera en su extremo es de 90° como se aprecia en la figura inferior:

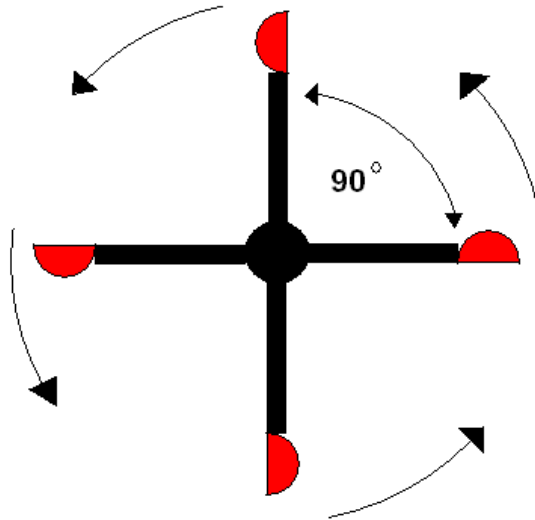


Fig. 29

El principio de funcionamiento del anemómetro consiste en medir el tiempo que demora los brazos en recorrer 90°.

Al obtener la medida de dicho tiempo, se puede calcular la velocidad angular que lograr alcanzar el mecanismo cuando el viento incide sobre él.

Si tomamos en cuenta que la distancia radial entre el centro del dispositivo giratorio y el centro de la semiesfera es de 11 cms , podemos tomar estos datos y calcular la velocidad lineal en metros por segundo a la que logra girar la semiesfera.

Tenemos que velocidad angular w está definida por:

$$w = 2\pi / t$$

Donde t es el tiempo que se demora en dar una vuelta en 360° o $2 * \pi$ pero tomando en cuenta que necesitamos calcularlo solo para 90° la fórmula se transforma en:

$$w = \pi / 2t$$

Y ahora t corresponde al tiempo necesario en recorrer 90°.

Ahora, si queremos transformarla a velocidad lineal basta con multiplicar lo anterior por el radio de la circunferencia, o sea, los 11 cms (0.11 mts) mencionados anteriormente

$$\text{Velocidad} = (\pi * 0.11) / 2t$$

Por ejemplo, si el tiempo que demora en recorrer 90° es de 0.1 segundos su velocidad en metros por segundo es de :

$$\text{Velocidad} = (3.14 * 0.11) / (2 * 0.1) = 1,728 \text{ mts / seg}$$

Si sabemos que

1 Km ----- 1000 metros

1 Hora ----- 3600 segundos

Al pasar de metros por segundo a kilómetro por hora se tiene un factor de 3.6, por lo tanto:

Velocidad (kilómetros / hora) = 3.6 * Velocidad (metros / segundo)

Así que en el ejemplo anterior la velocidad en kilómetros / hora sería :

Velocidad (kilómetros / hora) = 3.6 * 1.728 (metros/segundo) = 6,22 kilómetros / hora

Para medir el tiempo que demoran los brazos del anemómetro en girar 90° se utilizó un sensor infrarrojo del mismo tipo utilizado para la etapa que medía la dirección del viento, con su circuito respectivo para acondicionar la señal de salida.

De esta forma obtenemos a la salida del sensor un nivel lógico alto cada vez que los brazos completan un giro de 90° debido a que justo en ese momento bajo el sensor se encuentra una superficie reflectante que indica el avance de 90° de las semiesferas.

Las semiesferas giran solidarias a un eje que esta unido a un disco como se muestra en la figura siguiente:

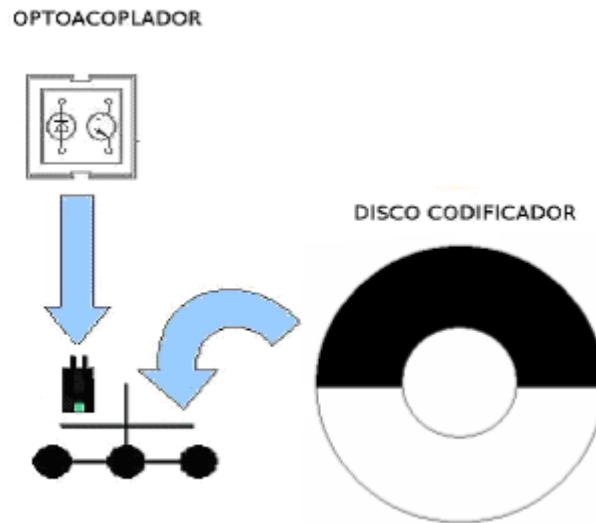


Fig. 30

Esta etapa a diferencia de la de medición de la dirección del viento funciona mediante interrupciones. Cada vez que hay un cruce de un brazo bajo el sensor se manda una interrupción al PIC mediante el pin habilitado para interrupciones externas mencionado en la primera parte.

Esta interrupción le indica al microcontrolador que comience la cuenta de tiempo hasta que cruce el siguiente brazo con el cual el microcontrolador guardara el tiempo medido y comenzará una nueva medida de tiempo.

Cada dos minutos el PC solicitará el envío de datos desde el microcontrolador y se enviará el último tiempo medido en el anemómetro, con este dato el programa en el PC realizado en visual basic 6.0 calculará la velocidad del viento en kilómetros / hora con las fórmulas explicadas anteriormente y mandará esta información al microcontrolador para mostrarla en el LCD.

Se realiza de esta forma, ya que, desde el punto de vista de la programación es mucho más práctico hacer cálculos complejos haciendo uso de lenguaje de alto nivel en Visual Basic que realizarlo en lenguaje ASM en el microcontrolador.

II.4.3 Sensor de Temperatura y Humedad SHT 15:

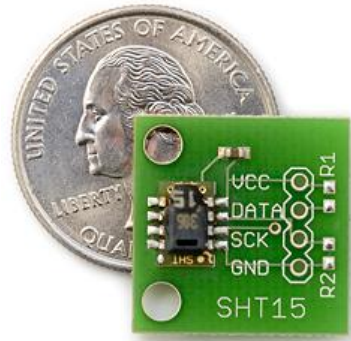


Fig. 31

El SHT de la casa www.sensirion.com es un sensor integrado de humedad calibrado en fábrica con salida digital mediante un bus serie síncrono y protocolo específico. El dispositivo también dispone de un sensor de Temperatura integrado para compensar la medida de humedad dependiendo de la temperatura, en casos extremos. Cuenta también en su interior con un calefactor para evitar condensación en el interior de la cápsula de medida para condiciones de niebla o similar donde existe condensación.

El SHT se puede alimentar con un rango de tensión continua comprendido entre 2,4 a 5V y es necesario proveer lo mas cerca posible del chip un condensador de desacoplo de 100nF entre GND y VCC, la empresa distribuidora en la cual compré el sensor, lo vende ya montado en un circuito impreso, en la placa que aloja el chip viene montado el condensador de desacoplo

Este chip dispone de 10 pines aunque solo se usan 4 de ellos y los demás sirven para sujetarlo de mejor forma a la placa.

El pin DATA corresponde a la salida/entrada de datos para comandar y leer el sensor, éste es un pin triestado por lo que necesita de una resistencia de polarización a Vcc (push-up), elegí una resistencia de 10 Kohm para esto.

El pin de SCK se utiliza para sincronizar la transmisión y no dispone de frecuencia mínima.

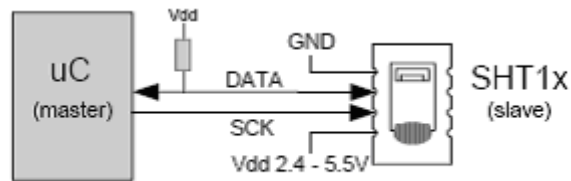


Fig. 32

A continuación se muestra el circuito impreso de este sensor, las resistencias que aparecen remarcadas son las que se agregan adicionales a la placa.

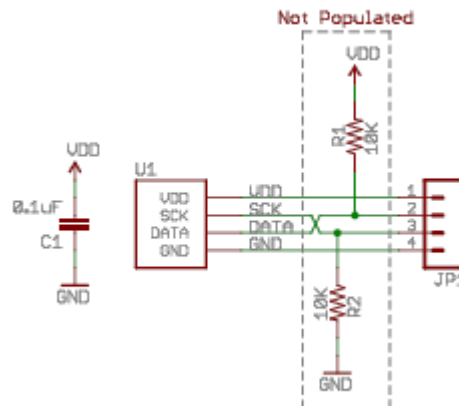


Fig. 33

II.4.3.1 Inicio de la transmisión:

Para comunicarse con el SHT lo primero que hay que hacer es mandar una secuencia de Inicio de Transmisión "Transmission Start". Esta consiste en poner a cero lógico la línea de datos mientras SCK esta a uno, seguidamente se genera un pulso bajo en SCK mientras la línea DATA sigue a cero y para finalizar se pone a 1 DATA mientras SCK esta a 1, seguidamente se baja a cero también SCK y finaliza la secuencia dejando la línea de datos y Clock en los estados lógicos por defecto, DATA en estado alto y SCK en estado bajo.

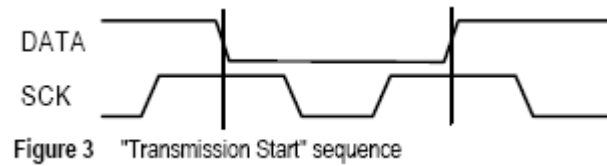


Fig. 34

Lo siguiente será mandar un comando de los siguientes que acepta:

Comando	Código binario
Reservado	0000x
Medida de Temperatura	00011
Medida de Humedad	00101
Leer el registro de estado interno	00111
Escribir el registro de estado interno	00110
Reservado	0101x – 1110x
Genera un Reset al software interno, reponiendo a los valores por defecto el registro de estado. Hay que esperar 11mS al menos después de mandar este comando.	11110

El protocolo de transmisión de comandos esta basado en un byte completo o lo que es lo mismo 8 bits, como los comandos son de 5 bits tan solo el resto de bits a la izquierda siempre serán 0. Por lo que se mandaran primero los 3 ceros seguido del comando para completar el byte.

El SHT indicará una recepción valida con un pulso de ACK en la línea de datos que es bidireccional y lo hará en el siguiente pulso de CLK después de haber completado los 8 bits del comando, por lo tanto acto seguido después de enviar el comando hay que configurar como entrada el puerto del microcontrolador conectado a DATA del SHT y generar una señal de

CLOCK para que el sensor nos mande su respuesta ACK y lo hará poniendo la línea a nivel bajo ya que como se dijo mas arriba la línea de datos esta polarizada a VCC por lo tanto siempre hay un 1 lógico en ella y lo que hace el SHT es forzar a 0 lógico con su salida a colector abierto.

II.4.3.2 Secuencia de medida:

Después de mandar un comando para que el sensor nos entregue la medición de humedad o temperatura, u otro comando es necesario esperar un tiempo de adquisición, este tiempo suele ser de unos 55mS para una resolución de salida de 12bits o bien, unos 11mS para una resolución de 8 bits. Para evitar esperas innecesarias el sensor genera un pulso bajo en la línea de datos y así se sabe que lo siguiente ya serán datos validos, esto se hace comprobando la línea de datos cada x tiempo hasta que esta pase a nivel bajo.

El paso siguiente es generar 8 pulsos de reloj en la línea CLK y se guarda el estado de cada bit que nos retornara el sensor. La trama que retorna el sensor se compone de 3 bytes, el primero corresponde a MSB el segundo a LSB y el tercero es el CRC-8 Checksum para comprobar que el dato a llegado correcto, pero lo podemos ignorar generando NACK después de la llegada del segundo byte (LSB). Como puede observarse el dato de la medida se compone de 2 bytes aunque como mucho el dato valido será de 12 bits de resolución, pero aunque configuremos el sensor para una resolución de 8 bits siempre leeremos 2 bytes ignorando el MSB en este caso.

Después de cada 8 bits recibidos el microcontrolador a de generar un ACK para que el sensor sepa que ya a recibido los datos. Esto se hace generando un flanco de subida en CLK mientras DATA esta a nivel bajo. Después de esto se continúa con la lectura de otro byte.

Para finalizar la trama se genera desde el microcontrolador un NACK (NoACK) esto es lo mismo que el ACK pero en vez de mantener la línea DATA a nivel bajo la tendremos que poner a 1 generando mientras esto pasa un flanco de subida en CLK igual que antes.

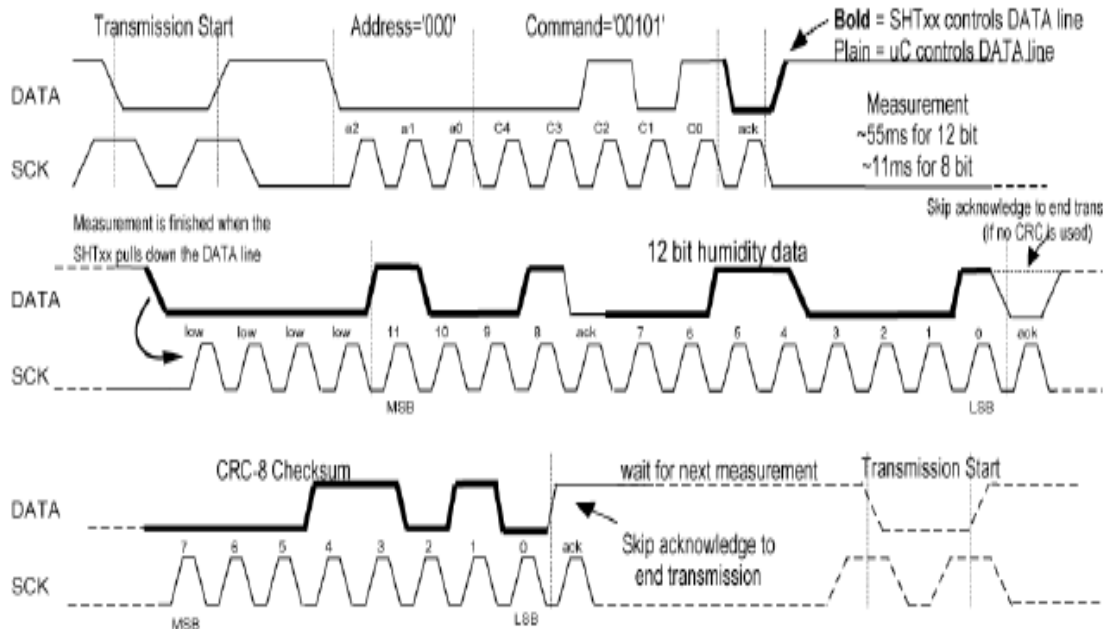


Figure 5 Example RH measurement sequence for value "0000'1001' 0011'0001" = 2353 = 75.79 %RH (without temperature compensation)

Fig. 35

El sensor admite unas dos adquisiciones o medidas por segundo

En la tabla que se muestra a continuación se muestran los porcentajes de error que tiene cada modelo de la familia SHT, el utilizado es el SHT15, porque cumple con los requisitos de esta aplicación:

Part Number	Humidity accuracy [%RH]	Temperature accuracy [K] @ 25 °C
SHT10	±4.5	±0.5
SHT11	±3.0	±0.4
SHT15	±2.0	±0.3
SHT71	±3.0	±0.4
SHT75	±1.8	±0.3

Fig. 36

Los valores mostrados en la tabla son pertenecientes a la zona típica de trabajo, ya que, en otras zonas el error es diferente, en gráfico siguiente se puede apreciar como varía el error dependiendo de la temperatura.

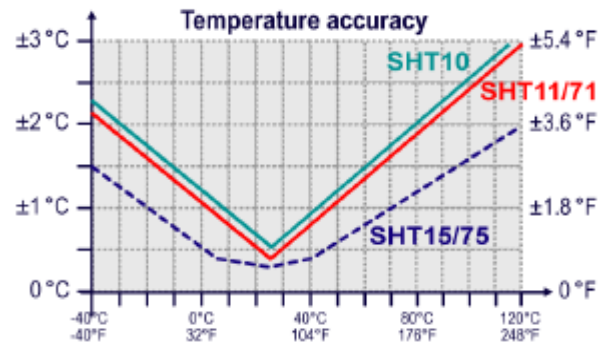


Fig. 37

La lectura que se tiene del sensor no es directamente la temperatura o la humedad, es un valor que debe introducirse en una fórmula para obtener las medidas de humedad y temperatura respectivas, a continuación se presenta la fórmula usada para calcular la humedad relativa, donde SO_{RH} es la lectura entregada por el sensor.

$$RH_{linear} = c_1 + c_2 \cdot SO_{RH} + c_3 \cdot SO_{RH}^2$$

SO_{RH}	c_1	c_2	c_3
12 bit	-4	0.0405	$-2.8 \cdot 10^{-6}$
8 bit	-4	0.648	$-7.2 \cdot 10^{-4}$

Table 6 Humidity conversion coefficients

Fig. 38

De igual forma es necesario introducir la lectura entregada por el sensor a una fórmula para obtener la temperatura, donde SO_T es el valor entregado por el sensor:

$$\text{Temperature} = d_1 + d_2 \cdot SO_T$$

VDD	d_1 [°C]	d_1 [°F]
5V	-40.00	-40.00
4V	-39.75	-39.55
3.5V ⁽³⁾	-39.66	-39.39
3V ⁽³⁾	-39.60	-39.28
2.5V ⁽³⁾	-39.55	-39.19

SO_T	d_2 [°C]	d_2 [°F]
14bit	0.01	0.018
12bit	0.04	0.072

Para minimizar el error mostrado para la humedad relativa a otras temperaturas que no sean de la zona típica de trabajo, hay una fórmula que involucra la medida entregada por el sensor en lo referente a humedad y además involucra la medida de temperatura, para mejorar la calidad del trabajo esta fue la medición utilizada, previniendo errores innecesarios a otras temperaturas que no sean la zona típica de trabajo, la fórmula es la que se presenta a continuación:

$$RH_{true} = (T_c - 25) \cdot (t_1 + t_2 \cdot SO_{RH}) + RH_{linear}$$

SO _{RH}	t ₁	t ₂
12 bit	0.01	0.00008
8 bit	0.01	0.00128

Fig. 39

II.4.4 Nivel de agua caída

El pluviómetro es un instrumento que se emplea en las estaciones meteorológicas para la recogida y medición de la precipitación.

El nivel de agua caída es una variable muy importante a tomar en cuenta en meteorología. Por un asunto de no seguir aumentando los costos de este proyecto no se incluyó físicamente pero se dejará explicada la forma de abordar el problema de la creación de un sistema de pluviometría, el cual puede ser añadido en una versión futura de este proyecto.

Una manera tradicional de medir el nivel de agua caída durante el día es usar un envase graduado el cual es revisado por una persona normalmente cada 10 horas, este envase debe ser vaciado a la medianoche para comenzar a acumular nuevamente agua en el recipiente para nuevas mediciones del día siguiente.

Para lograr mantener la estación meteorológica con su característica de automática se decidió usar, además del sensor encargado de medir el nivel de agua caída, un servomotor cuya función principal es dar en la medianoche un giro de 180° al recipiente receptor del agua para de esta

forma vaciarlo, una vez terminada esta operación inmediatamente se vuelve el recipiente a su posición original para continuar con las mediciones del siguiente día.

A continuación se dará una descripción de los dos elementos más importante para la creación del sistema de pluviometría, el servomotor y posteriormente el sensor de presión.

Los servomotores juegan un papel importante en los microbots ya que proporcionan un elevado par de salida y son relativamente fáciles de controlar, pudiendo ser conectados directamente a los microcontroladores

Un servomotor es básicamente un motor eléctrico de corriente continua con un gran par, que nos permite situar su eje de salida en una determinada posición angular, mediante una señal externa de control.

Los servomotores son usados principalmente en modelismo de radio control, pero también pueden usarse para dotar de movimiento a los microbots.



Fig. 40

Un servomotor está formado por:

- Carcasa: aloja todos los elementos.
- Motor: es un motor de corriente continua.
- Engranajes: transfieren el par del motor al eje de salida, además reducen la velocidad del motor.
- Circuito de control: controla la posición del eje de salida.
- Potenciómetro: se utiliza como sensor para conocer la posición del eje de salida.

La conexión se realiza mediante 3 cables, el cable rojo es la alimentación positiva, el cable negro es la masa, el cable de control es de color blanco o a veces amarillo.

Por defecto un servomotor sólo puede girar aproximadamente 180 grados, para conseguir que gire de forma continua es necesario realizar una modificación interna.

El circuito de control, situado dentro de la carcasa, posiciona el eje de salida comparando la señal proporcionada por el potenciómetro, que está unido al eje de salida, con la señal de control externa. Al girar el eje de salida también gira el potenciómetro.

La señal de control es una onda cuadrada con una frecuencia de 50 Hz y un valor peak entre 3 y 5 V, conocida como PWM (Pulse Width Modulation). La frecuencia de control puede variar dependiendo del fabricante del servomotor, pero debe ser una señal estable para poder conseguir un buen posicionamiento del eje de salida.

El cable de control se usa para comunicar el ángulo. El ángulo está determinado por la duración de un pulso que se aplica al cable de control. El servo espera ver un pulso cada 20 milisegundos. La longitud del pulso determinará los giros de motor. Un pulso de 1.5 ms., por ejemplo, hará que el motor se torne a la posición de 90 grados (llamado la posición neutra). Si el pulso es menor de 1.5 ms., entonces el motor se acercará a los 0 grados. Si el pulso es mayor de 1.5ms, el eje se acercará a los 180 grados.

En la figura inferior se aprecia esto en forma gráfica.

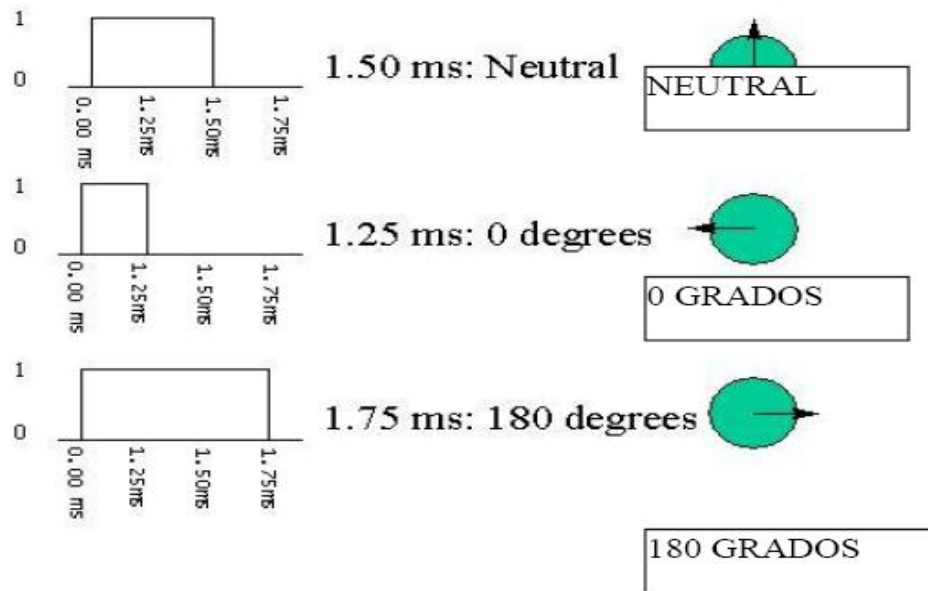


Fig. 41

El segundo elemento elegido, el cual sirve para medir el nivel de agua caída es un sensor de presión modelo MPXV5004G ofrecido por el sitio web www.freescale.com.

A continuación se muestra el empaquetado de este sensor:

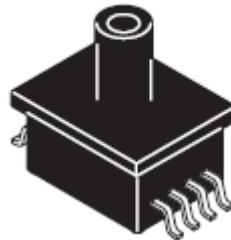


Fig. 42

Alimentación, tierra y salida son los tres pines útiles que posee, el resto solo sirven para fijación.

Se puede alimentar con 5 V, lo cual lo hace útil para esta aplicación.

En V out (salida) se tiene una tensión de salida directamente proporcional a la presión del aire, como se aprecia en el empaquetado mostrado anteriormente, el sensor tiene una especie de bombilla pequeña por la cual entra el aire. Para poder medir el nivel de agua se uniría una bombilla al sensor y se ubicaría en el recipiente que recibe el agua caída como se muestra en la figura siguiente:

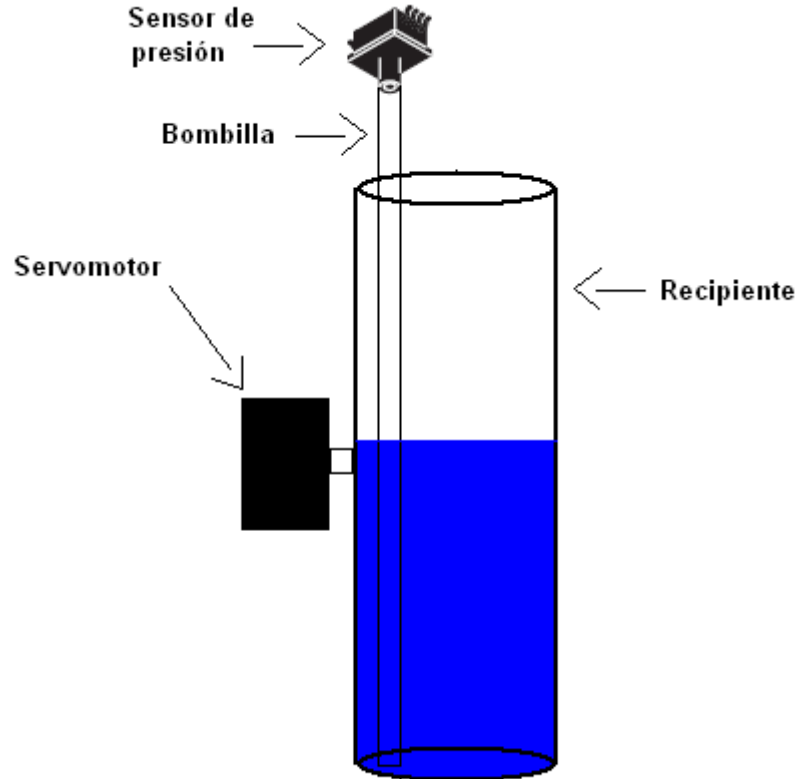


Fig. 43

El esquema mostrado muestra sólo los elementos principales, al momento de construirlo deberá incluirse una protección para aislar los pines del sensor del agua, además deberá incluirse algún tipo de abrazadera que rodee todo el envase y lo fije al eje del servomotor; además la mayor parte del servomotor estará introducida dentro de la estación para darle una buena fijación.

El principio de funcionamiento es que a medida que el nivel del agua suba, la presión del aire comprimido dentro de la bombilla también aumentará de forma directamente proporcional según lo indicado por el fabricante.

La relación entre el voltaje de salida del sensor y el nivel de agua está dada por una recta, de forma que el cálculo resultaría simple para el programa en Visual Basic, pero la calibración y elección de las dimensiones del recipiente deberán hacerse una vez comprado el sensor tomando en cuenta los diferentes niveles de voltaje de salida para distintos niveles de agua.

Para que la estación sepa cuando dar vuelta el recipiente con agua con ayuda del servomotor se podría enviar de forma automática un comando desde la aplicación en Visual Basic por medio del cable serial a la medianoche, el microcontrolador interpretaría este comando como una orden de vaciado y después de enviar al PC los datos de las muestras correspondientes a las 00:00 horas, procedería a girar 180° el recipiente para vaciarlo e inmediatamente lo volvería a su posición normal. Después de esta operación el programa seguiría funcionando de la forma diseñada e implementada actualmente.

Cabe destacar que el pin de salida del sensor irá conectado a uno de los pines del microcontrolador que posee la función de conversor análogo digital.

II.4.5 Revisión automática del nivel de batería:

Para que el sistema funcione correctamente es necesario que el microcontrolador PIC sea alimentado con una fuente de voltaje continuo de 5 V, para lograr esta tensión se decidió usar una batería de 9 V conectada a la entrada de un regulador de tensión 7805, el cual para este nivel de entrada proporciona en su salida un nivel continuo de 5 V.

Se pensó que es de utilidad que el sistema tuviera la capacidad de avisar al usuario cuando sea necesario cambiar la batería. Para lograr esto se utilizó una de las entradas análogo digital que posee el PIC 16F877A, conectándola directamente a la salida de 5 V proporcionada por el regulador de tensión, de esta forma se incluyó en el programa principal guardado en el PIC una secuencia de instrucciones que se aseguren que el PIC está siendo alimentado con una tensión entre 4.7 y 5 V, las cuales son adecuadas para el correcto funcionamiento del sistema y en el caso de que la tensión de entrada fuera menor a 4.7 V entonces el PIC desplegaría en la pantalla LCD el siguiente mensaje:



Fig. 44

Además, el PIC enviaría un aviso al PC para que mediante la interfaz realizada en Visual Basic el sistema le avise al usuario que es necesario cambiar la batería, el programa realiza esto mediante una señal sonora y a través de un mensaje en pantalla.

Para logra el correcto funcionamiento de esta etapa fue necesario usar otra batería de 9 V regulada a 5 V para ser usada como voltaje de referencia del PIC.

La batería para el voltaje de referencia no es necesario sustituirla al cambiar la batería de alimentación del PIC debido a que el consumo de corriente del circuito desgasta solamente la batería de alimentación y no la de referencia.

II.4.6 Programa realizado en Visual Basic 6.0:

Con el fin de visualizar los datos que está enviando la estación meteorológica al computador y para visualizar además los datos guardados en días y meses anteriores se desarrolló una aplicación realizada en Visual Basic 6.0.

Al hacer doble clic sobre el icono de programa en el escritorio del PC o donde el usuario lo haya situado, se abre la pantalla de presentación del programa como se muestra a continuación:



Fig. 45

Como se puede observar en esta pantalla, se muestra el nombre del proyecto, el nombre del alumno tesista y el nombre de los profesores involucrados en el trabajo.

Después de un momento se carga la siguiente pantalla de bienvenida:



Fig. 46

En ella se muestran dos botones que llevan a dos caminos diferentes.

Al hacer clic en el primero de ellos denominado “**Tomar Muestras**”, se carga la siguiente pantalla:



Fig. 47

En ésta se dan las instrucciones necesarias para comenzar a tomar muestras.

La primera de las instrucciones como se aprecia en la imagen hace referencia a realizar la conexión entre el computador y la estación meteorológica mediante un cable serial, sin el cual sería imposible la transmisión de los datos.

La segunda instrucción recuerda al usuario encender la estación para que el proceso de adquisición de datos se realice correctamente.

Una vez conectada la estación al puerto serie del PC se hace click en el botón aceptar y se llega a la siguiente pantalla:

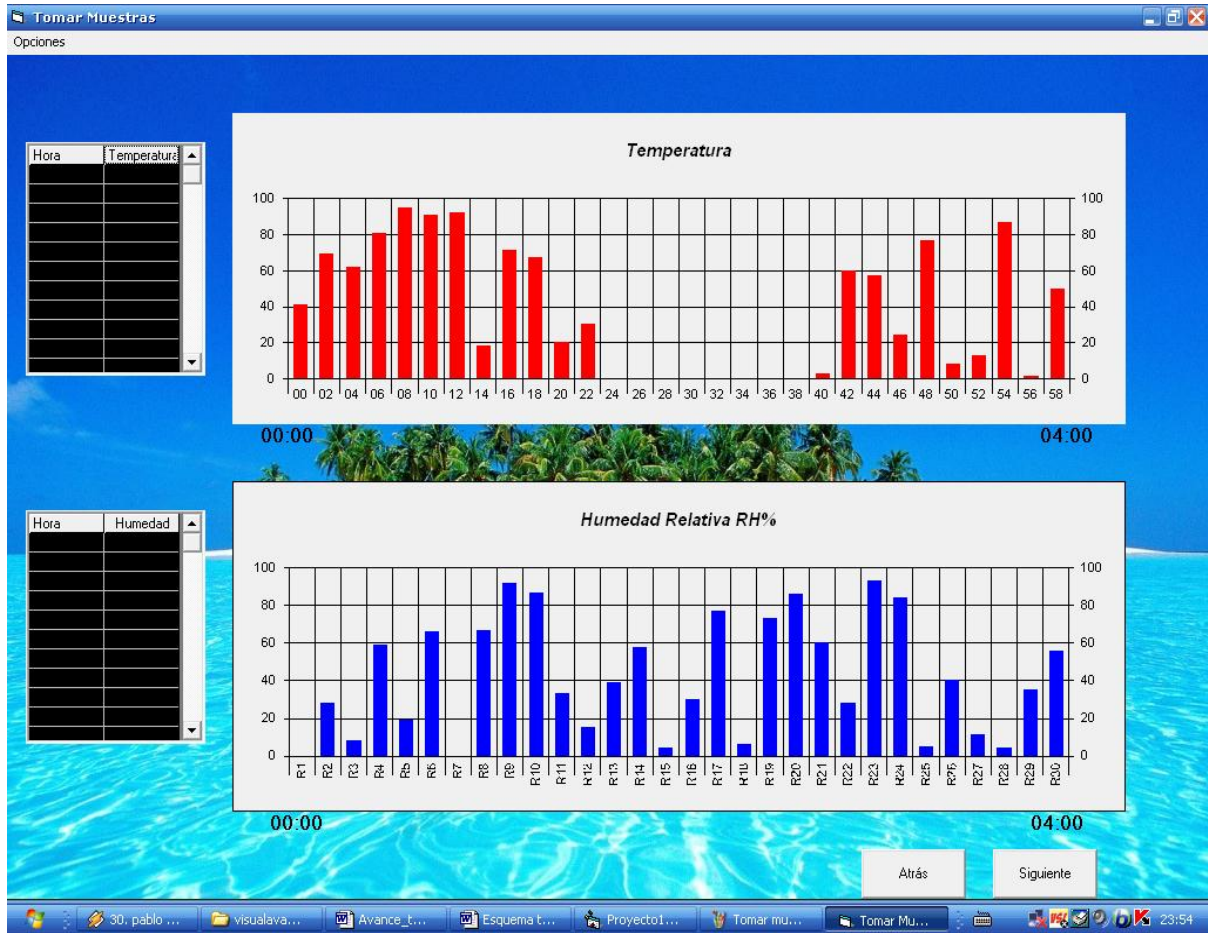


Fig. 48

Aquí se observan dos gráficos en los cuales visualizarán los datos que vayan llegando de la estación, además al lado de cada uno de los gráficos se encuentra un cuadro en el que irá apareciendo el valor medido y la hora de la muestra, lo cual permite saber el valor exacto que muestra el gráfico para una hora determinada.

Las variables que se grafican son la velocidad del viento, dirección del viento, temperatura ambiental, humedad relativa del aire y además se grafican tres datos adicionales a los objetivos iniciales del proyecto, estos datos adicionales son el punto de rocío, la sensación térmica y el nivel de luz ambiental.

Si se busca en la parte superior izquierda de la pantalla se encuentra el botón de opciones, al presionarlo se puede ver la opción denominada ‘**Orden Datos**’, al presionarla se accede al siguiente cuadro:

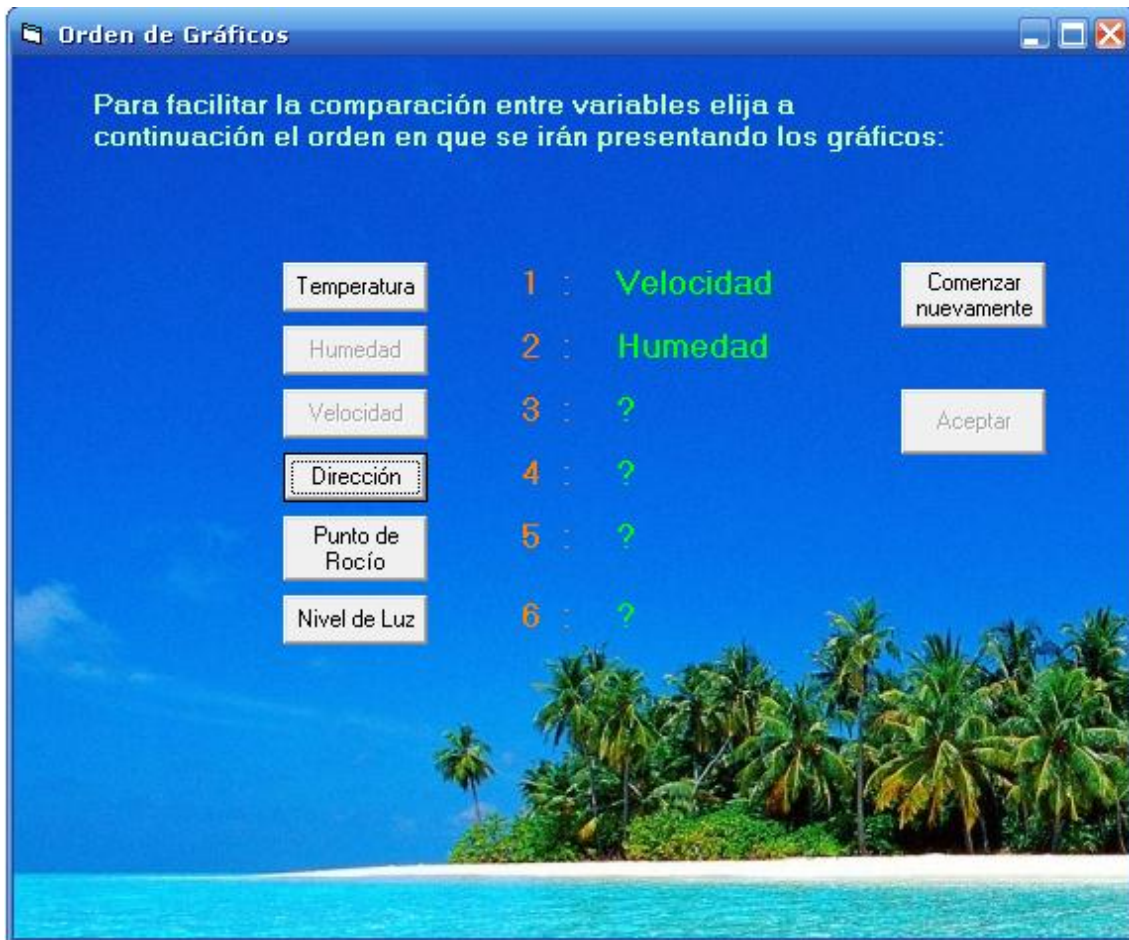


Fig. 49

En esta pantalla se muestran como botones todas las variables que se pueden graficar, al ir presionando sobre cada una de ellas se va definiendo el orden en el que se mostrarán los gráficos. Además posee un botón que permite comenzar a definir el orden nuevamente en caso de que el usuario haya cambiado de parecer en último minuto. Presionando **Aceptar** se vuelve a la pantalla anterior.

Volviendo nuevamente a la pantalla en la que están los gráficos y sus cuadros a los costados cabe destacar que cada dos minutos se guarda una muestra de cada variable medida y se visualiza en pantalla.

Una vez completada una hora, el gráfico se borra y comienza a graficar la hora siguiente.

En la parte inferior derecha de la pantalla hay dos botones, uno que dice **Siguiente** y otro que dice **Atrás**, si se presiona siguiente se puede ver que en los dos gráficos se actualiza la información con los datos de las dos variables siguientes respetando el orden definido por el usuario en la opción **Orden Datos** y si se presiona **Atrás** se vuelven a graficar las variables anteriores.

En el caso de que el usuario no intervenga en la opción **Orden Datos**, el programa tiene un orden de despliegue de datos establecido que se seguirá.

En el menú opciones esta la opción **Tipo de Gráfico**, la cual permite elegir que tipo de gráfico se utilizará para graficar las variables, se tiene el gráfico de barras, de líneas y de 3 D, entre otros.

Como se menciono anteriormente, al presentarse la pantalla de bienvenida hay dos botones, el botón **Tomar Muestras** explicado anteriormente y un botón llamado **Graficar Muestras Guardadas**, este último al ser presionado haciendo clic con el mouse sobre él presenta la siguiente pantalla:

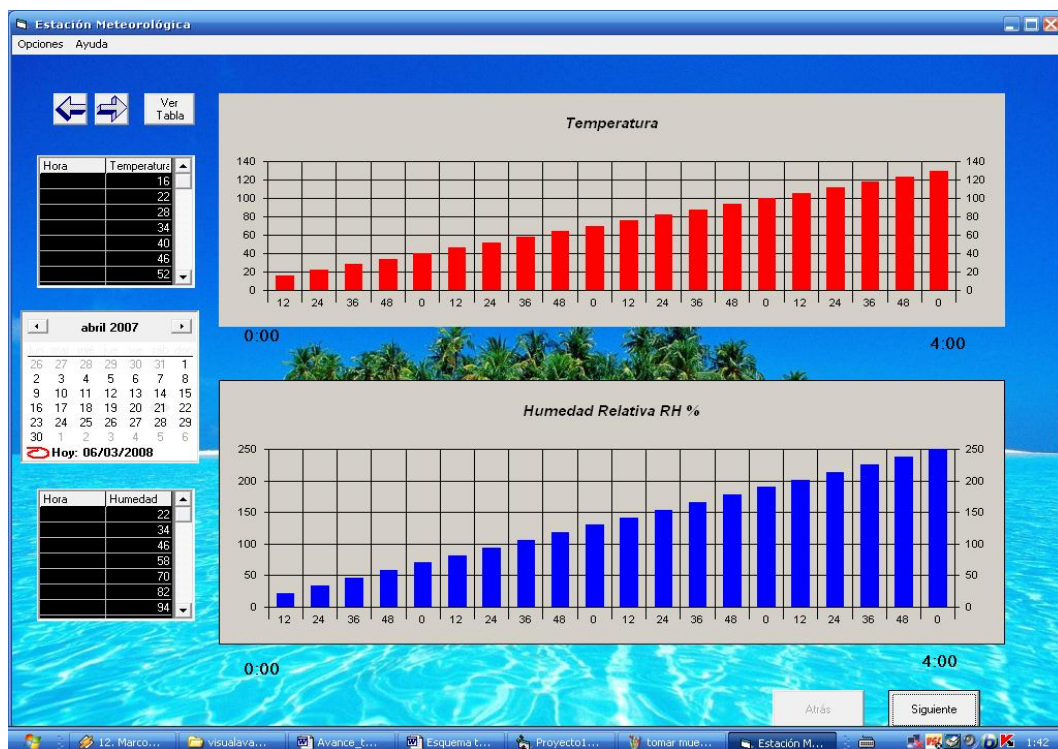


Fig. 50

La función de esta parte de la aplicación es graficar los datos guardados anteriormente con el botón **Guardar Muestras**.

Los datos guardados se encuentran grabados en una base de datos Access.

Graficar los datos guardados no afecta a que la estación este tomando muestras y guardándolas en la base de datos, así que perfectamente se puede dejar el programa guardando los datos provenientes de la estación y al mismo tiempo observar gráficos de días y meses anteriores.

De forma similar a la pantalla de **Guardar Muestras**, esta parte de la aplicación muestra de manera simultánea dos gráficos en los cuales se presentan los datos guardados. En la parte inferior derecha de la pantalla hay un botón **Adelante** y uno **Atrás** para cargar en los dos gráficos las siguientes dos variables o las anteriores dos variables guardadas respectivamente.

En el menú opciones ubicado en la esquina superior izquierda de la pantalla, presionando **Orden Datos** se puede definir el orden en que aparecerán los datos en formar similar a lo explicado en la etapa de **Guardar Muestras**. Además en la opción configuración se tiene la opción de elegir el tipo de gráfico usado, pero también tiene una opción que es muy importante para realizar los gráficos, en la opción configuración se puede elegir el número de muestras que se graficarán por pantalla y el lapso de tiempo en horas que se graficarán por pantalla.

A continuación se muestra la pantalla de configuración mencionada:



Fig. 51

Debido a que hay varias opciones para la cantidad de muestras por pantalla y para la cantidad de horas por pantalla, el usuario dispone de 24 posibles combinaciones.

De esta forma es bastante adaptable el sistema a los requerimientos que tenga el usuario.

Para ir desplazándose de una pantalla a otra en los gráficos, se dispone de dos flechas en la parte superior izquierda de la pantalla, una hacia delante y otra hacia atrás, así por ejemplo en el caso de tener un número de 20 muestras por pantalla y un número de 4 horas por pantalla al presionar la flecha hacia delante se graficarán las 4 horas siguientes con sus respectivas 20 muestras y si se presiona la flecha hacia atrás se volverá a graficar las 4 horas anteriores. Opera de igual forma para cualquiera de las otras 23 combinaciones disponibles.

Con el fin de hacer más amigable al usuario la forma de elegir el día que se desea graficar, se incluyó un calendario en el lado izquierdo de la pantalla como el que se muestra a continuación:



Fig. 52

En la parte superior del calendario se puede elegir el mes del día que se busca y en el momento de hacer doble clic en el número de la fecha deseada, automáticamente el sistema cargará en los gráficos los datos del día elegido.

Otra opción que le da el programa al usuario es la posibilidad de visualizar en una sola tabla todas las muestras guardadas de todas las variables para el día elegido, esto se logra presionando el botón **Ver Tabla** que se encuentra en la parte superior izquierda de la pantalla.

Hora	Velocidad
	22
	34
	46
	58
	70
	82
	94

Fig. 53

A continuación se muestra la tabla cargada al presionar **Ver Tabla**:

Fecha Seleccionada: 21/4/2007

	Temperatura	Humedad	Velocidad	Direccion	Nivel
	11	12	13	14	15
	12	14	16	18	20
	13	16	19	22	25
	14	18	22	26	30
	15	20	25	30	35
	16	22	28	34	40
	17	24	31	38	45
	18	26	34	42	50
	19	28	37	46	55
	20	30	40	50	60
	21	32	43	54	65
	22	34	46	58	70
	23	36	49	62	75
	24	38	52	66	80
	25	40	55	70	85
	26	42	58	74	90
	27	44	61	78	95
	28	46	64	82	100
	29	48	67	86	105
	30	50	70	90	110
	31	52	73	94	115
	32	54	76	98	120

Exportar a Excel

Todos los datos del día

Solo los datos del modo seleccionado

Aceptar

Volver

Fig. 54

Además el programa permite exportar estos datos a Microsoft Excel permitiendo de esta forma que el usuario haga con ellos cualquier otro tipo de gráficos o cálculos.

El programa da dos opciones para exportar los datos, la primera es exportar todos los datos guardados durante el día, o sea, muestras cada dos minutos teniendo un total de 720 muestras por cada variable. Si al usuario le parece demasiadas también puede optar por la opción de exportar

solo la cantidad de muestras que eligió en la configuración; por ejemplo, si se eligió graficar 20 muestras cada 4 horas se tendría un total de:

$$4 * 20 * 6 = 480 \text{ muestras por variable}$$

El 6 se incluyó para completar las 24 horas del día. Hay otras configuraciones en que este número es mucho menor.

II.4.7 Programa cargado en microcontrolador PIC:

La programación de la estación meteorológica se hizo en dos lenguajes de programación diferentes, para el programa realizado en el microcontrolador PIC 16f877A se utilizó el lenguaje assembly y se utilizó el lenguaje BASIC para la aplicación desarrollada en Visual Basic 6.0.

A continuación se presenta a grandes rasgos la secuencia de tareas que realizaba el programa del microcontrolador:

- (1) Definición de variables en ram del pic.**
- (2) Configuración de puertos análogos y digitales.**
- (3) Habilitación de las interrupciones que se usarán.**
- (4) Configuración del Timer 1.**
- (5) Configuración de transmisión y recepción.**
- (6) Inicialización de pantalla LCD.**
- (7) Lee sensores infrarrojos de dirección.**
- (8) Lee nivel de luz ambiental.**
- (9) Lee nivel de batería.**
- (10) Determina si el nivel de batería está bajo el límite.**
- (11) Si está bajo el límite escribe Batería Baja en LCD y salta al punto (16)**
- (12) Si esta bien la batería escribe en LCD la dirección leída.**
- (13) Escribe en LCD la velocidad medida.**
- (14) Escribe en LCD la humedad relativa medida.**

(15) Escribe en LCD la temperatura medida.

(16) Si llego petición desde PC llama rutina ‘Envía muestras’ y después vuelve al punto (7).

(17) Si no llegó petición desde el PC, vuelve al punto (7) sin llamar rutina ‘Envía Muestras’.

La rutina ‘Envía Muestras’, envía al PC la información necesaria para obtener las medidas de dirección, velocidad, temperatura, humedad relativa, batería y nivel de luz. Si se observa todos los pasos mostrados anteriormente no se observa en ninguna parte que se haya realizado la medición de temperatura, humedad y velocidad; se hizo de esta forma porque el sensor de temperatura y humedad SHT 15 permite una velocidad máxima entre consultas de dos segundos aproximadamente, entonces no es conveniente forzarlo con innecesarias consultas continuas. La forma en que se resolvió esta limitación consiste en consultarle al sensor SHT 15 solamente cuando el PC pida que le envíen datos, por este motivo la lectura de temperatura y humedad se hace en la rutina de ‘Enviar Muestras’ y no en el programa principal mostrado anteriormente.

Debido a que las mediciones hechas por el sensor SHT 15 no entrega directamente los valores de temperatura y humedad, sino que entrega un valor el cual debe ingresarse a una fórmula para calcular la temperatura y humedad correspondiente se decidió que este cálculo lo haría la aplicación en Visual Basic para evitar complicar la programación en el microcontrolador y una vez realizado el cálculo se enviaría al PIC por medio del puerto serie los valores ya calculados de temperatura y humedad para ser presentados en la pantalla LCD.

De igual forma en el caso de la velocidad, lo que mide el microcontrolador es el tiempo entre interrupciones hechas por el anemómetro y para obtener la velocidad este tiempo debe ingresarse a una fórmula.

Con el mismo fin de simplificar la programación se decidió que el PIC enviaría el tiempo al PC y la aplicación Visual Basic calcularía la velocidad y enviaría esa información al microcontrolador para ser mostrada en la pantalla LCD. Esta forma de abordar el problema funcionó perfectamente.

La rutina de interrupción que no se aprecia en la lista de pasos mostrados anteriormente tiene funciones importantes, primero que nada debe reconocer de que tipo de interrupción se trata, puede ser producida por el anemómetro, por el timer 1 o por recepción de datos.

En el caso de que la interrupción sea provocada por el anemómetro. El programa guarda el tiempo medido entre la interrupción actual hecha por el anemómetro y la interrupción anterior que también haya sido realizada por el anemómetro, con esta medida de tiempo se puede calcular la velocidad del anemómetro pero este cálculo lo hace la aplicación en Visual Basic.

El timer 1 es un módulo interno del microcontrolador que sirve para medir tiempo o puede usarse como contador. En este caso se usó como instrumento de medición del tiempo. La interrupción provocada por este módulo se realiza cuando el registro interno del timer 1 llega a su máximo; si es que se desea poder medir un tiempo mayor sólo es necesario resetear el timer 1 e ir contando con otra variable cuantas veces ha sido peseteado, esto es lo que se hace en la rutina de interrupción cuando es el timer 1 quién la provoca.

Cuando llegan datos al PIC a través del puerto serie también se provoca una interrupción, en este caso la rutina de interrupción analiza si lo que llegó es un dato o un aviso del dato que llegará.

Como se mencionó anteriormente la aplicación hecha en Visual Basic hace los cálculos de temperatura, humedad y velocidad; la forma en que envía estos datos al microcontrolador es enviando primero el primer decimal, luego la unidad, decena y centena de cada dato. Antes de enviar un dígito, envía primero una letra de control que permite saber a la rutina de interrupción a que número de que dato pertenece, después de repetir el proceso con distintas letras de control la rutina de interrupción tiene la información que se desplegará en la pantalla LCD.

II.4.8 Objetivos adicionales logrados:

II.4.8.1 Punto de rocío:

El punto de rocío es la temperatura a la cual el aire debe ser enfriado para que ocurra la saturación, siempre que no haya un cambio en el agua contenida. El punto de rocío es una medida importante utilizada para predecir la formación de rocío, hielo y niebla. Si la temperatura y el punto de rocío están cercanos en la misma tarde cuando el aire empieza a volverse frío, es probable que se forme niebla durante la noche. El punto de rocío es también un buen indicador del vapor de agua contenido en el aire en el momento, al contrario que la humedad relativa que considera la temperatura del aire. Un punto de rocío alto significa que hay mucho vapor de agua contenido. Un valor bajo significa poco vapor de agua contenido. Además un alto punto de rocío indica probabilidad de lluvia y tormentas. Puede usar el punto de rocío para predecir la temperatura mínima de la noche. Siempre que no se esperen nuevos frentes por la noche y la humedad relativa por la tarde sea $\geq 50\%$, el punto de rocío por la tarde le dará una idea de que temperatura mínima debe esperar por la noche, puesto que al aire es probable que no se enfríe más que el punto de rocío en ningún momento de la noche.

La aplicación desarrollada en Visual Basic 6.0 calcula automáticamente el punto de rocío usando los datos de temperatura y humedad que la estación haya registrado, se grafica de igual forma que las otras variables.

La fórmula que relaciona la temperatura real y la humedad relativa para calcular la temperatura del punto de rocío es la siguiente:

$$\text{Temperatura del punto de rocío} = 243.12 \cdot H / (17.62 - H)$$

Donde:

$$H = (\log_{10}(RH) - 2) / 0.4343 + (17.62 \cdot T) / (243.12 + T)$$

RH = humedad relativa

T = temperatura real en °Celsius

II.4.8.2 Nivel de luz:

La LDR (Light Dependent Resistor) o resistencia dependiente de la luz, como su propio nombre lo indica es una resistencia que varía su valor en función de la luz que incide sobre su superficie. Contra mas sea la intensidad de luz que incida en la superficie de la LDR menor será su resistencia y contra menos luz incida mayor será la resistencia.

Se utilizó uno de los pines conversores análogo digitales del microcontrolador PIC para medir la tensión de salida en un circuito divisor de voltaje en el cual se incluía una LDR, este circuito se protegió mediante una cápsula de plástico transparente y se situó en la parte superior de la estación para que le llegara sin problemas la luz ambiental; utilizando este sensor y con ayuda de la aplicación realizada en Visual Basic se tiene un registro de las horas de sol para la localidad en la que esté instalada la estación.

II.4.8.3 Sensación Térmica:

Todos hemos apreciado que la sensación térmica de calor o frío no sólo depende de la temperatura sino que intervienen otros factores como son la humedad relativa del aire o el viento. Así, si hace frío y además sopla viento la sensación de frío es mayor, a su vez la combinación de calor y humedad puede provocar una sensación agobiante.

Dos índices creados para evaluar y cualificar la sensación térmica son el **índice de calor "heat index"** y el **índice del enfriamiento del aire "windchill index"**.

II.4.8.3.1 Índice de calor

El índice de calor es la combinación de la temperatura del aire y la humedad que proporciona una descripción de la manera en que se percibe la temperatura. Expresado en grados Celsius ó Fahrenheit indica el nivel del calor que se siente cuando la humedad relativa se suma a la temperatura real. Para su cálculo emplea la temperatura real y nos da como resultado una "temperatura aparente".

Se calcula por medio de la siguiente fórmula:

$$\text{IC} = - 42.379 + 2.04901523 \cdot T + 10.14333127 \cdot \text{HR} - 0.22475541 \cdot T \cdot \text{HR} - 6.83783 \times 10^{-3} \cdot T^2 - 5.481717 \times 10^{-2} \cdot \text{HR}^2 + 1.22874 \times 10^{-3} \cdot T^2 \cdot \text{HR} + 8.5282 \times 10^{-4} \cdot T \cdot \text{HR}^2 - 1.99 \times 10^{-6} \cdot T^2 \cdot \text{HR}^2$$

Donde:

IC: índice de calor

HR: humedad relativa en %

T : temperatura del termómetro en grados Fahrenheit

II.4.8.3.2 Índice del enfriamiento del aire

El índice del enfriamiento del aire es un cálculo de temperatura no real que toma en consideración los efectos que conjuntamente tienen el viento y la temperatura real en el cuerpo humano. Cuando el viento sopla sobre la piel, arrastra la capa aisladora de aire cálido adyacente a la piel. Cuando todos los factores se dan por igual, mientras más sopla el viento, más se quita el calor, lo cual resulta en una sensación más fría.

El índice de enfriamiento se puede calcular mediante la expresión:

$$\text{Indice de enfriamiento } ^\circ\text{F} = 35.74 + 0.6215 \cdot T - 35.75 \cdot V^{0.16} + 0.4275 \cdot T \cdot V^{0.162}$$

Donde:

T = Temperatura del aire o°F

V = velocidad del viento mph

Las conversiones de unidades se realizan automáticamente en la aplicación y se grafica de igual forma que las otras variables.

Cabe destacar que dentro de las opciones del programa esta la elección por parte del usuario de si se usará el índice de calor o el índice de enfriamiento del aire para calcular la sensación térmica, ya que esta elección debe hacerla el usuario meteorólogo dependiendo del clima típico de la región y de la estación del año.

CAPÍTULO III CONSTRUCCIÓN, PUESTA EN MARCHA Y PRUEBAS

Al terminar el diseño y programación del microcontrolador y de la aplicación en Visual Basic, se procedió a la etapa de traspaso del montaje hecho en protoboard a una placa y a la construcción del prototipo.

Para alojar al sensor de temperatura y humedad SHT 15, se eligió una tapa de plástico como se muestra en la foto:



Esto se decidió debido a que las paredes de la tapa protegen al sensor del viento y de la luz solar directa logrando de esta forma que las medidas no sean afectadas por estos factores

Para construir el anemómetro se utilizó como mecanismo giratorio un pequeño ventilador en desuso, las aspas del ventilador se fijaron firmemente a un tubo de PVC y la otra parte del ventilador quedó girando libremente debido a que el ventilador poseía 4 brazos metálicos, éstos se utilizaron para fijar en cada esquina una semiesfera de plástico.

A continuación se muestra una foto de los brazos metálicos, las semiesferas y el disco con zonas blancas y negras hecho con un disco compacto fijado a la estructura.

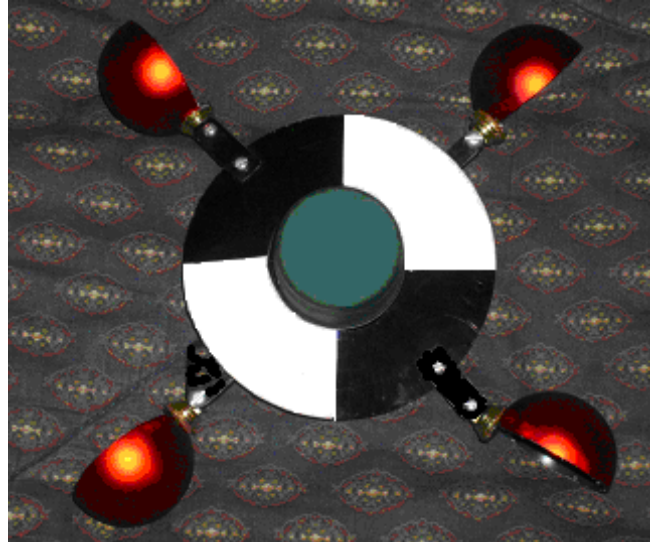


Fig. 56

Para que el anemómetro funcione fue necesario ubicar entre el tubo de PVC y el disco el sensor infrarrojo mencionado en capítulos anteriores.

A continuación se muestra una foto del anemómetro montado:



Fig. 57

En forma similar para construir el sensor encargado de medir la dirección del viento se utilizó un tubo de PVC, un pequeño ventilador para el mecanismo giratorio con sus aspas unidas firmemente al tubo de PVC, un disco con zonas blancas y negras y tres sensores infrarrojos los cuáles sirven para interpretar el código del disco.

Además se construyó una veleta de aluminio la cual se unió al conjunto giratorio.

A continuación se muestran dos fotos, una foto del disco con el código de colores y la otra de todo el sensor montado:

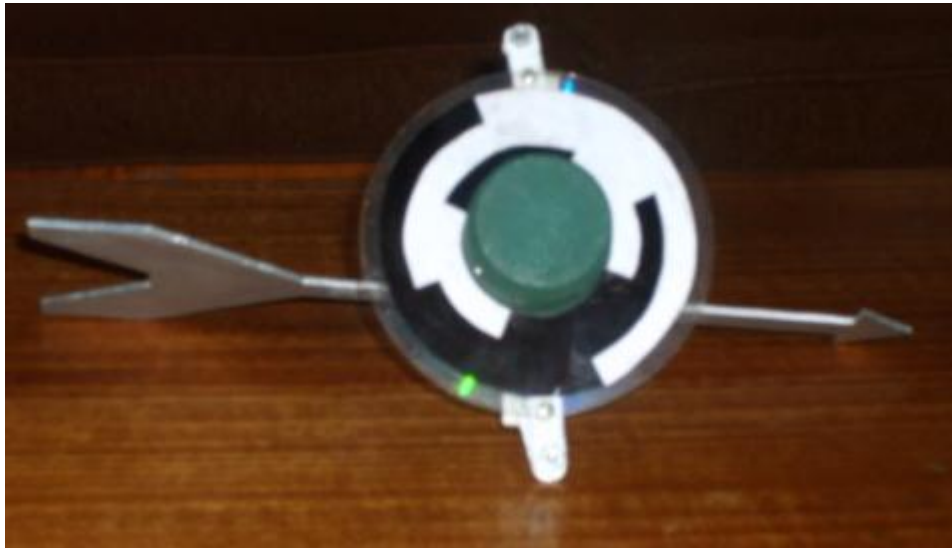


Fig. 58

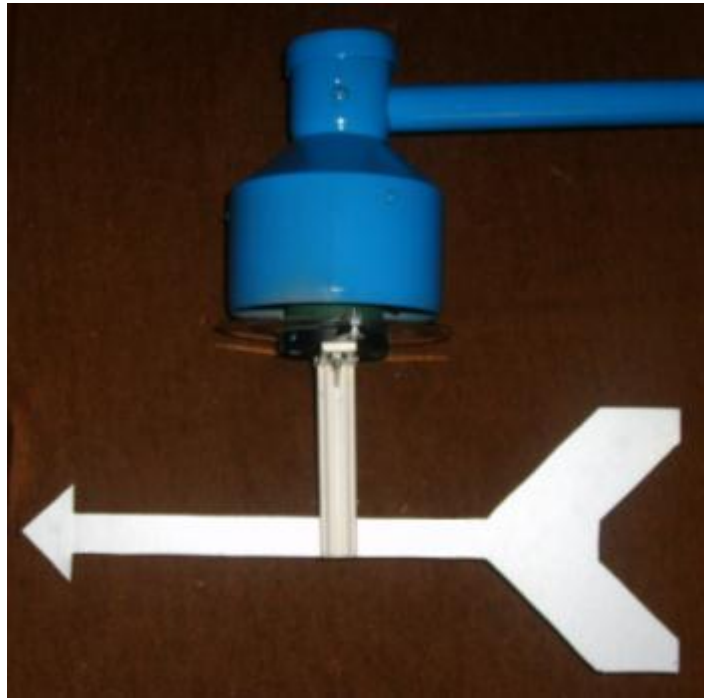


Fig. 59

Para determinar el nivel de iluminación se montó una fotorresistencia sobre una placa de cobre y se protegió por una cápsula de plástico transparente como se muestra en la siguiente foto:



Fig. 60

La pantalla LCD, los led indicadores, y la placa con el circuito principal en la que está alojado el microcontrolador y llegan las conexiones de la pantalla y todos los sensores; se alojó en una caja metálica como se muestra a continuación:



Fig. 61

Los sensores de iluminación, temperatura y humedad se fijó directamente a esta caja metálica y el anemómetro y la veleta se unieron a esta mediante un tubo de aluminio.

A continuación se muestra una foto de toda la estación meteorológica armada:

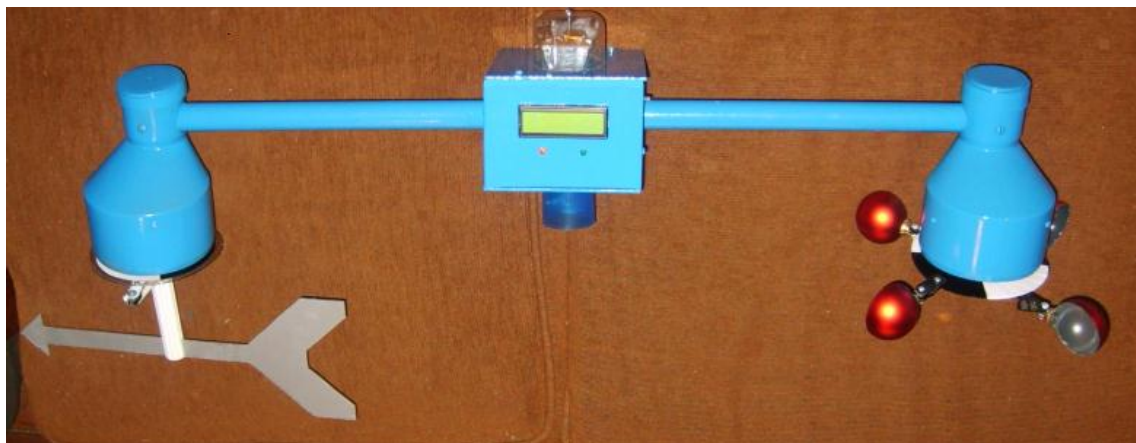


Fig. 62

Con la estación armada se procedió a realizar algunas pruebas. Antes de hacer las pruebas fue necesario orientar la estación con su parte frontal mirando hacia el norte.

Para comprobar el funcionamiento del sensor de temperatura y humedad SHT 15, se midió la temperatura y humedad en distintos momentos del día. En el caso del sensor de temperatura se comparó con la medida entregada por un termómetro de mercurio y siempre se obtuvo una diferencia menor a 1 grado, esta diferencia coincidía con el margen de error que mencionaba el fabricante.

En el caso de la humedad se utilizó como patrón el conocido y respetado sitio web del clima www.theweatherchannel.com, se comparó la humedad relativa medida por la estación con la humedad relativa anunciada para ese horario por el sitio web, de igual forma a la temperatura la diferencia entre las dos medidas era muy pequeña; haciendo esta comparación varias veces en el día se pudo afirmar que el sensor de temperatura y humedad SHT 15 funcionaba correctamente.

Para comprobar la correcta medición del anemómetro no fue necesario compararlo con otro anemómetro, ya que, la base de esta etapa radica en la precisión con la que se mide el tiempo entre una interrupción y la siguiente o dicho de otra forma entre el paso una zona blanca y la siguiente; así que se utilizó un cronómetro y se comparó con el tiempo medido por el microcontrolador al detectar dos interrupciones, los tiempos medidos fueron iguales en todas las pruebas que se hicieron.

Cabe señalar que la precisión que tiene la para medir el tiempo entre interrupciones es 10 mil veces más exacto que la precisión de un cronómetro común debido a que el cronómetro mide centésimas de segundo y el microcontrolador es capaz de medir millonésimas de segundo.

Para comprobar el funcionamiento de la veleta fue mucho más fácil, ya que, con ayuda de una brújula una vez orientada la estación hacia el norte, operación que debe realizarse al montar la estación en terreno, bastaba con girar la veleta con la mano a otra dirección y ver como coincidía tanto la medida vista en la pantalla LCD como en la aplicación en Visual Basic con la lectura esperada.

Para la comprobación del funcionamiento del sensor de iluminación fue muy fácil debido a que se puso como referencia la propia vista para comprobar el nivel de iluminación indicado por la estación, el cual coincidía con distintas iluminaciones testeadas.

El punto de rocío y la temperatura aparente no necesitaba ser comprobados debido a que estos datos se calculan con fórmulas que dependen de las variables ya analizadas.

III.1 Ajustes finales:

Para finalizar el desarrollo de este proyecto fue necesario volver a realizar el circuito impreso de la placa principal, ya que, la placa ya construida presentaba algunas pistas muy finamente unidas debido a una sobre exposición de la placa al ácido, lo cual puede producir problemas posteriormente.

Otro detalle que fue necesario corregir fue la distancia que sobresalían los sensores infrarrojos del tubo de PVC, fue necesario reducir un centímetro aproximado esa distancia, ya que, sino se reducía el mecanismo giratorio no quedaba firmemente ajustado.

CONCLUSIONES

- La forma en que se abordó el problema en sus diferentes etapas fue muy adecuada, ya que, se logró cumplir con todos los objetivos propuestos.
- Además, cabe destacar que además de lograr todos los objetivos propuestos, el programa registra datos adicionales que le dan valor agregado al proyecto. Estos datos adicionales son el registro del punto de rocío, el nivel de iluminación ambiente y la temperatura aparente.
- La forma en que se realizó el software en Visual Basic 6.0 permite en un futuro aumentar el número de variables medidas sin realizar grandes cambios en el código de programa.
- El diseño del programa cargado al microcontrolador permite en un futuro aumentar el número de sensores para lograr medir más variables ambientales.
- A pesar de que la medición del agua caída se hizo solamente teórico como se había propuesto en los objetivos iniciales, el microcontrolador pic elegido permite una futura incorporación de el sensor necesario, ya que se dejaron sin usar entradas que permiten la conversión análogo digital y además pines digitales disponibles.
- Se comprobó que el sensor de temperatura y humedad SHT 15 venía calibrado de fábrica como lo aseguraba el fabricante en el certificado de calibración, ya que, la medida de temperatura tenía sólo una pequeña diferencia en la zona de trabajo comparándolo con un termómetro de mercurio.
- La página del clima www.theweatherchannel.com ofrece en su pronóstico las distintas humedades relativas, entre otros datos, para cada hora del día en la ciudad, de esta forma esta página se tomó como instrumento patrón para comprobar que el valor de humedad

relativa medido por la estación era correcto. Las medidas de humedad relativa realizadas por la estación y las proporcionadas por el sitio web eran muy similares, por lo tanto se concluye que la estación funciona correctamente en la medición de esta variable.

- Para asegurarse de que la velocidad era medida correctamente no fue necesario compararlo con otro anemómetro, sino que se comprobó la exactitud con la cual el anemómetro construido medía el tiempo en que el sensor infrarrojo detectaba las zonas blancas y negras del disco explicado en el apartado correspondiente. El cristal utilizado para establecer la frecuencia de trabajo del microcontrolador es de 4 megas, con esta frecuencia se logra un control del tiempo por parte del microcontrolador de 1 microsegundo, por lo tanto, 1 microsegundo es la exactitud con la cual el anemómetro mide el tiempo, el buen funcionamiento de esta medición se comprobó comparando el tiempo medido por el anemómetro entre dos interrupciones, con el tiempo medido por un cronómetro; cabe destacar que el tiempo medido por el anemómetro es 10 mil veces más preciso que un cronómetro que mide centésimas de segundo, en repetidas mediciones se observó que los tiempos medidos eran iguales, logrando de esta forma asegurar el correcto funcionamiento del anemómetro.

- Se comprobó el buen funcionamiento del sensor que mide la dirección del viento con el sólo hecho de girar la flecha de la veleta en una dirección determinada conocida. Esta se verificó que medía correctamente tanto en la pantalla LCD como en el programa en Visual Basic.

GLOSARIO

SHT15: Sensor de temperatura y humedad calibrado en fábrica.

E/S: Pin del microcontrolador que puede funcionar como entrada o salida.

AD: Análogo digital.

Timer 0: Módulo perteneciente al microcontrolador pic que se utiliza como contador o como temporizador.

USART: Módulo del microcontrolador pic utilizado para transmisión y recepción de dato de forma serial.

Assembly: Lenguaje de programación de tipo ensamblador utilizado para programar el microcontrolador pic.

Mplab : Software de diseño y simulación de programas para microcontroladores.

MCP – USB: Dispositivo utilizado para grabar microcontroladores pic.

PROTEUS: Software de diseño y simulación de circuitos electrónicos.

DATA: Terminal del sensor SHT 15 utilizado para enviar y recibir datos.

SCK: Terminal del sensor SHT 15 utilizado para enviar pulsos de reloj.

SOrh: Coeficiente entregado por el sensor SHT 15 usado para calcular la humedad relativa del aire.

SOt: Coeficiente entregado por el sensor SHT 15 usado para calcular la temperatura ambiental.

REFERENCIAS BIBLIOGRÁFICAS

LIBROS

Enrique Palacios, Fernando Remiro, Lucas López. Microcontrolador PIC 16F84 Desarrollo de Proyectos. 1º Ed., Editorial Alfaomega Ra-ma, México, 2004.

Anónimo, Microcontrolador 16F877

Microchip , PIC 16F87X Data sheet.

WEBS

www.x-robotics.com/sensores	10/01/08 17:00 pm
http://www.reconnsworld.com/ir_ultrasonic_basiciirdetectemit.html	15/01/08 19:00 pm
http://www.freescale.com/webapp/sps/site/overview.jsp?nodeId=0112699036	05/02/08 16:00 pm
http://es.wikipedia.org/wiki/Estaci%C3%B3n_meteorol%C3%B3gica	07/02/08 20:00 pm