



# Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería  
Escuela de Ingeniería Civil Electrónica

## DISEÑO, IMPLEMENTACIÓN Y OPTIMIZACIÓN DE UN MONITOR INALÁMBRICO EN UN INVERNADERO

Tesis para optar al título de:  
Ingeniero Electrónico.

Profesor Patrocinante:  
Sr. Néstor Fierro Morineaud.  
Ingeniero Electrónico,  
Licenciado en Ciencias de la Ingeniería,  
Diplomado en Ciencias de la Ingeniería.

LUIS HERNÁN ALARCÓN MELLADO

VALDIVIA - CHILE

2009

**COMISIÓN REVISADORA**

**Profesor patrocinante:**

Sr. Nestor Fierro M.

\_\_\_\_\_

**Profesores ayudantes**

Sr. Franklin Castro R.

\_\_\_\_\_

Sr. Alejandro Villegas M.

\_\_\_\_\_

## **AGRADECIMIENTOS**

A todos que de una u otra forma me acompañaron en este tramo de mi vida, en especial a mi familia, a mis amigos, a mis profesores y a todos los demás, igual.

## INDICE

<b>PORTADA.....</b>	<b>i</b>
<b>COMISIÓN REVISADORA.....</b>	<b>ii</b>
<b>AGRADECIMIENTOS.....</b>	<b>iii</b>
<b>INDICE.....</b>	<b>iv</b>
<b>RESUMEN.....</b>	<b>vii</b>
<b>SUMMARY.....</b>	<b>viii</b>
<b>INTRODUCCIÓN.....</b>	<b>ix</b>

## CAPITULO I

### PLANTEAMIENTO DEL PROBLEMA

<b>1 PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>1</b>
--	----------

## CAPITULO II

### MARCO TEÓRICO

<b>2.1 INVERNADEROS.....</b>	<b>2</b>
<b>2.11 Ventajas y desventajas del cultivo en invernaderos.....</b>	<b>3</b>
<b>2.12 Tipos de invernaderos.....</b>	<b>3</b>
<b>2.13 Control del clima en invernaderos.....</b>	<b>4</b>
<b>2.2 CARACTERÍSTICAS DE LAS VARIABLES A CONTROLAR.....</b>	<b>6</b>
<b>2.21 Temperatura ambiente.....</b>	<b>6</b>
<b>2.22 Humedad.....</b>	<b>6</b>
<b>2.3 INSTRUMENTACIÓN EXISTENTE.....</b>	<b>9</b>
<b>2.31 Sensores de temperatura.....</b>	<b>9</b>
<b>2.32 Sensores de humedad ambiental.....</b>	<b>9</b>
<b>2.33 Sensores de humedad de suelo.....</b>	<b>10</b>

### CAPITULO III

#### DISEÑO E IMPLEMENTACIÓN DEL MONITOR.

<b>3.1 ETAPA DE CAPTURA DE VARIABLES.....</b>	<b>13</b>
<b>3.2 ETAPA DE ORDENAR VARIABLES.....</b>	<b>15</b>
<b>3.3 ETAPA DE TRANSMISIÓN Y RECEPCIÓN.....</b>	<b>16</b>
<b>3.4 ETAPA DE PROCESAR VARIABLES.....</b>	<b>17</b>

### CAPITULO IV

#### DISPOSITIVOS UTILIZADOS

<b>4.1 SENSORES.....</b>	<b>18</b>
<b>4.11 Sensor SHT15.....</b>	<b>18</b>
<b>4.111 Comunicación con el sensor SHT15.....</b>	<b>20</b>
<b>4.112 Registro Status.....</b>	<b>22</b>
<b>4.113 Características eléctricas.....</b>	<b>23</b>
<b>4.114 Conversión a los valores físicos .....</b>	<b>24</b>
<b>4.12 Sensor de Humedad de suelo.....</b>	<b>25</b>
<b>4.2 MODULO TRANSMISOR Y RECEPTOR.....</b>	<b>27</b>
<b>4.21 Disposición de pines.....</b>	<b>28</b>
<b>4.22 Modulación GFSK.....</b>	<b>30</b>
<b>4.3 MICROCONTROLADORES.....</b>	<b>33</b>
<b>4.31 El microcontrolador 16f870.....</b>	<b>34</b>
<b>4.311 Conversión Análoga Digital.....</b>	<b>36</b>
<b>4.312 Modulo USART.....</b>	<b>41</b>
<b>4.32 El microcontrolador 16f628.....</b>	<b>45</b>
<b>4.4 INTERFAZ RS232 A USB.....</b>	<b>46</b>

## CAPITULO V

### SOFTWARE UTILIZADOS

<b>5.1 MPLAB.....</b>	<b>47</b>
<b>5.11Partes del MPLAB.....</b>	<b>47</b>
<b>5.2 VISUAL BASIC.....</b>	<b>51</b>
<b>5.21Entorno visual Basic.....</b>	<b>52</b>

## CAPITULO VI

### FUNCIONAMIENTO Y CARACTERISTICAS DEL EQUIPO

<b>6.1 CONFECCIÓN DE LOS CIRCUITOS.....</b>	<b>55</b>
<b>6.11 Diseño de las placas.....</b>	<b>55</b>
<b>6.12 Traspaso del circuito a cobre y soldado.....</b>	<b>56</b>
<b>6.2 FUNCIONAMIENTO DEL EQUIPO.....</b>	<b>57</b>
<b>6.3 EVALUACION FINANCIERA DEL DESARROLLO DEL MONITOR.....</b>	<b>59</b>
<b>CONCLUSIONES.....</b>	<b>61</b>
<b>BIBLIOGRAFÍA.....</b>	<b>63</b>
<b>ANEXOS.....</b>	<b>64</b>
<b>CÓDIGO DE PROGRAMACIÓN VISUAL BASIC.....</b>	<b>64</b>
<b>CÓDIGO DE PROGRAMACIÓN PIC 16F870.....</b>	<b>66</b>
<b>CÓDIGO DE PROGRAMACIÓN PIC 16F628.....</b>	<b>72</b>

## RESUMEN

La presente investigación, consiste en el diseño e implementación de un instrumento que monitoree las variables que son de relevancia en el funcionamiento un invernadero, para luego transmitirlos de forma inalámbrica, siendo almacenados los datos en un computador. De acuerdo a la investigación realizada se pudo detectar los diferentes requerimientos que debía tener el monitor, los cuales eran las variables a tomar y la forma de transmisión. Las variables que son en general importantes en los diferentes tipos de invernaderos, son la humedad ambiental, la temperatura ambiental y la humedad del suelo. Los invernaderos en su mayoría son estructuras fijas recubiertas con algún material transparente, que permite producir el efecto de invernadero.

El diseño del monitor quedo separado en 4 etapas principales, la captura de variables, ordenar variables, transmisión y recepción, y procesar variables. En la captura de variables en donde se encuentran los sensores para medir la humedad de suelo, la humedad ambiental y la temperatura; el sensor de humedad de suelo fue construido de yeso y placas de cobre, detectando los cambios en la resistencia del material en presencia de humedad, y lo demás esta en el sensor SHT15, que mide la humedad ambiental y la temperatura al mismo tiempo. En la segunda etapa se toman estos datos para ordenarlos para su posterior transmisión, para este efecto se ocupo el microcontrolador 16F870. Para la etapa de transmisión y recepción se utilizaron las tarjetas UM96, que tienen un alcance de 500mts con modulación GFSK (Gaussian Frequency shift keying), además de tener interfases para conexión con niveles TTL y RS232. Esto va conectado a un adaptador de RS232 a USB, el cual crea un puerto serial virtual en el computador, pudiendo trabajarse de esta forma para la recepción de los datos. Y como ultima etapa se reciben los datos en visual Basic, pasando estos a una plantilla Excel donde el usuario podría trabajar con estos.

## SUMMARY

The present investigation consists of the design and implementation of an instrument that monitors the variables relevant to the functioning of a greenhouse, an instrument that also transmits data wirelessly to store in a computer. According to the research realized, it was possible to detect the different requirements necessary for the monitor to have, these being the variables to document and the form of transmission. Generally, the variables most important to document in different types of greenhouses are environmental humidity, environmental temperature, and humidity of the soil. Greenhouses are principally fixed structures covered with some kind of transparent material that permits the production of the greenhouse effect.

The design of the monitor can be separated in four main steps: the capture of variables, the organization of variables, transmission and reception, and the processing of variables. The capture of variables is where we find the sensors for measuring the humidity of the soil, the humidity of the environment, and the temperature. The sensor for measuring the humidity of the soil was constructed of plaster and copper plates, detecting changes in the resistance of material in the presence of humidity; the rest was calculated by the sensor SHT15, which measures both environmental humidity and temperature at the same time. In the second step, these data are ordered for their subsequent transmission, for which we occupied the microcontroller 16F870. For the transmission and reception step, we used UM96 cards, which have a reach of 500m with GFSK modulation (Gaussian Frequency Shift Keying), as well as having an interface for connecting to levels TTL and RS232. This was in turn connected to an RS232 adaptor via USB, creating a virtual serial gateway in the computer, enabling it to work in this form for the reception of data. Finally, in the last step data were received in visual Basic, being passed to an Excel database where the user may work with them.

## INTRODUCCIÓN

La presente investigación, consiste en el diseño e implementación de un instrumento que permite el monitoreo constante y de forma automática de las variables consideradas de mayor relevancia en el funcionamiento de un invernadero, como son la humedad ambiental, humedad de suelo y temperatura ambiente, estas variables serán capturadas por medio de sensores especializados; uno de los cuales se encargara de medir específicamente la humedad de suelo mientras que el segundo medirá tanto la temperatura como humedad del ambiente. Una vez obtenidos los datos de los respectivos sensores estos serán ordenados por un software para su posterior transmisión, la cual debido a la distancia en que se encuentran las instalaciones, se realizara de forma inalámbrica a una estación, donde serán recepcionados y almacenados en un computador para su posterior análisis y estudio.

### **Objetivo general**

Realizar el diseño de un monitor para invernaderos, que considere las variables que son relevantes en este, transmitiendo esos datos de forma inalámbrica. Luego de esto hacer la implementación experimental, optimizando el trabajo del monitor.

### **Objetivos específicos**

- Estudiar las características que debiera tener el monitor.
- Diseñar el monitor contemplando las diferentes etapas que lo conforman.
- Realizar el estudio de cada una de las etapas.
- Implementar de manera experimental el monitor.
- Optimizar el funcionamiento de este, adecuándolo a los requerimientos reales.

## **1 CAPITULO I: PLANTEAMIENTO DEL PROBLEMA**

Hoy en día, en la mayoría de los invernaderos de nuestra zona, la toma de datos se realiza de forma manual. Si bien, éstos se ingresan a un computador, no es un monitoreo constante que indique los cambios bruscos. Por lo demás, no otorga la posibilidad de un análisis más profundo en relación a la variación en las condiciones dentro del invernadero.

El monitoreo en forma automática de un invernadero, apunta al control y estudio de éste. Las variables generalmente utilizadas en los invernaderos son los de temperatura, humedad ambiental y humedad del suelo. Instaurando nuevas herramientas, las variables podrían ser captadas mediante sensores especializados y los datos serían enviados a un microcontrolador el cual los prepararía para ser remitidos a través de una tarjeta de transmisión inalámbrica, luego ser recibidos por una tarjeta receptora de datos; conectada en un puerto USB de un computador.

## **2 CAPITULO II: MARCO TEÓRICO.**

### **2.1 INVERNADEROS**

“El invernadero podemos definirlo como una estructura de madera o metal y se encuentra recubierta por vidrio, PVC, polietileno transparente, que en su interior se puede cultivar hortalizas, flores o plantas, en época del año, donde dada las condiciones climáticas del lugar sería imposible de producirlas al aire libre”(Infante et al., 2004). Esta construcción aprovecha el efecto producido por la radiación solar, la cual calienta los objetos que hay detrás; éstos, a su vez, emiten radiación con una longitud de onda mayor que la solar, este efecto es el denominado de invernadero.

La función del material traslúcido es, atrapar energía dentro del invernadero ya que esto calienta el ambiente interior. En ausencia de un recubrimiento, el calor absorbido se eliminaría por corrientes convectivas y por la emisión de radiación infrarroja. La presencia del recubrimiento impide el transporte del calor acumulado hacia el exterior por convección y, obstruye la salida de una parte de la radiación infrarroja. El efecto neto es el de acumulación de calor y aumento de temperaturas del recinto. “El vidrio permite el paso de la mayor parte de la radiación solar (en especial, de poca longitud de onda), excepto si esta sucio o si su ángulo de incidencia de los rayos solares es demasiado agudo, lo que supone la pérdida de una parte de la energía. En cambio, el vidrio no es atravesado por las radiaciones de mayor longitud de onda procedentes del suelo, sino que, o bien refleja esta energía, que vuelve entonces al suelo, o la absorbe momentáneamente para volverla a emitir después. El cristal, por lo tanto actúa como “pared trampa” para el calor, pues permite el paso de la mayor parte de este al interior del bastidor o invernadero, impidiendo luego la salida del calor al exterior. El aire no se calienta directamente por radiación, sino eleva su temperatura (o la disminuye) tomando o cediendo calor de las superficies con las que contacta” (Esain, 1964).

## **2.11 Ventajas y desventajas del cultivo en invernaderos**

Los invernaderos principalmente ofrecen las siguientes ventajas:

- Los frutos se cosechan en menos tiempo que el normal
- Se aumenta la calidad y el rendimiento
- Se puede producir fuera de época
- Se ahorra en agua y fertilizantes
- Facilita el control de plagas
- Posibilidad de cultivar mas de una ves al año
- Mejor aprovechamiento del espacio del predio

Entre las desventajas están:

- Alta inversión inicial
- Los costos operativos pueden ser elevados (calefacción, aislamiento, etc.)
- Se requiere de personal capacitado para su operación.

## **2.12 Tipos de invernaderos**

“Pueden construirse invernaderos de casi cualquier forma y tamaño, que en cada ocasión vendrán determinadas principalmente por el emplazamiento, cosechas que se intenta cultivar, materiales utilizados par ala construcción y capital disponible. En general, los invernaderos pueden clasificarse de acuerdo con el revestimiento empleado en el tejado, en uno de los tres principales grupos principales que existen: contruidos de madera según el sistema tradicional, contruidos de aluminio y aquellos otros cuyo fundamento son las cristaleras holandesas. Las construcciones nuevas deben diseñarse, o elegirse entre los modelos prefabricados existentes, para conseguir la máxima flexibilidad en las cosechas. Se consigue el máximo conjunto de ventajas mediante un modulo de anchura de unos 1,5 metros y una altura mínima hasta el alero del tejado de 1,8 metros. Excepto en las construcciones niveladas, los edificios no deberán sobrepasar los 35

metros de longitud, ya que de otra manera resulta difícil mantener una distribución uniforme de la temperatura de un extremo a otro del local” (Toovey, 1959).

Según su uso y temperaturas, hay distintos tipos de invernaderos que se pueden clasificar como:

**2.121 Invernaderos fríos:** éstos consisten en una estructura que sólo recibe el calor del sol. Si es zona de inviernos fríos, la temperatura interior del invernadero será de aproximadamente 5°C por encima de la temperatura exterior. Este tipo de invernaderos se usa para, sembrar o almacenar plantas de semillero a finales de invierno o primavera.

**2.122 Invernaderos frescos:** este tipo de invernadero puede mantener una temperatura mínima de 5-7°C. Estos invernaderos se calentarían durante los meses de invierno en zonas de clima frío. Puede usarse para proteger a las plantas sensibles a las heladas, para cultivar plantas tres o cuatro semanas antes que en el invernadero frío, para cultivos de estación templada durante el verano o para cultivos de clima fresco durante el otoño e invierno.

**2.123 Invernaderos templados:** Este tipo de invernadero puede mantener una temperatura mínima de 13°C con calor adicional durante el día y la noche, dependiendo de su emplazamiento. Los costes de calefacción subirán a medida que bajen las temperaturas. Ofrece buenas condiciones para el cultivo de hortalizas y de muchas plantas anuales.

**2.124 Invernaderos cálidos:** Este tipo de invernadero resulta ser el más caro en cuanto a su mantenimiento porque, mantiene una temperatura mínima de 18°C con la ayuda de calor adicional. Aunque puede resultar demasiado sofocante para muchas hortalizas, puede destinarse al cultivo de plantas tropicales y subtropicales.

## **2.13 Control del clima en invernaderos**

**2.131 La Luminosidad:** básicamente es dar mayor o menor luminosidad al ambiente, utilizando mallas de sombreo en el caso de requerir menos luminosidad o por el contrario hacer uso de iluminación artificial.

### **2.132 La Temperatura:**

“De las variables climáticas la temperatura es la más importante de regular. El control de temperaturas es importante porque influye en la transpiración, respiración germinación fotosíntesis crecimiento y floración de las plantas” (Infante et al., 2004).

**Cómo subir la temperatura:** para mantener una temperatura elevada se puede mantener el invernadero bien cerrado o, utilizar calefacción externa; aire caliente o agua. Otra alternativa es, considerar el incremento de la temperatura, desde la construcción del invernadero, colocando doble techo el cual limita el enfriamiento nocturno. Si bien éste no afecta la temperatura durante el día, sí afecta la luminosidad

**Cómo bajar la temperatura:** generalmente se lleva a cabo mediante la ventilación del invernadero; mojar el suelo y oscurecimiento de las paredes, mediante pinturas o mallas. De hecho, los problemas de sobrecalentamiento son más comunes que los de un calentamiento deficiente.

**2.133 La ventilación:** “Los invernaderos precisan ventilación por tres razones: para abastecimiento del dióxido de carbono utilizado en la fotosíntesis, para limitar la elevación de la temperatura del aire producida por el calor solar y para reducir la humedad procedente de la transpiración de las plantas” (Toovey1967).

La ventilación es muy importante, tanto para expulsar el aire caliente como para hacer que circule dentro del recinto, a la hora de evitar plagas y enfermedades. Los respiraderos deben estar situados en posiciones bajas y elevadas para establecer un flujo de aire adecuado. Unos extractores colocados en la parte alta del invernadero ayudan a expulsar el aire más caliente, mientras que permiten entrar aire fresco.

**2.134 Humedad de invernaderos:** “El contenido de humedad relativa en un invernadero es de gran importancia para el desarrollo normal del cultivo. Si la humedad es muy baja la productividad tiende a bajar, por el contrario la alta humedad y temperatura aceleran el

crecimiento de las plantas pero a su vez se desarrollan las condiciones para la proliferación de enfermedades” (Infante et al., 2004). La mayoría de las plantas prefieren una humedad relativa del aire entre el 45 y el 60%. La humedad alta favorece la transmisión de plagas y enfermedades y por el contrario la humedad baja podría secar las plantas.

## **2.2 CARACTERÍSTICAS DE LAS VARIABLES A CONTROLAR.**

Teniendo en cuenta el funcionamiento de un invernadero, podemos visualizar las variables que son relevantes en éstos: temperatura ambiente, humedad ambiente y la humedad del suelo. De las cuales se hablará a continuación:

**2.21 Temperatura ambiente:** Físicamente es una magnitud escalar dada por una función creciente del grado de agitación de las partículas de los materiales. A mayor agitación, mayor temperatura.

Existen varias formas de medir la temperatura; se dividen según el fenómeno físico que se registre, gracias a la variación de temperaturas, como lo son: la dilatación, la contracción, la resistencia, el potencial termoeléctrico y la radiación electromagnética.

Las escalas de medida que se utilizan son variadas, pero las que más se ocupan son los grados Celsius, los centígrados, kelvin y Fahrenheit. En Chile se ocupan generalmente los grados centígrados.

**2.22 Humedad:** la humedad es, un fenómeno natural que se presenta a nivel molecular y se encuentra básicamente relacionada con la cantidad de moléculas de agua presentes en una determinada sustancia, la cual puede estar en estado sólido o gaseoso.

**2.221 Humedad ambiente:** Se denomina humedad ambiental a la cantidad de vapor de agua presente en el aire. Esta se puede expresar de forma absoluta o relativa.

La humedad absoluta corresponde a la cantidad de vapor de agua presente en el aire. Esta se puede denotar de tres maneras, en gramos de agua por kilogramos de aire seco, gramos de agua por unidad de volumen o como presión de vapor.

La humedad relativa es la humedad que contiene una masa de aire, en relación con la máxima humedad absoluta que podría admitir sin producirse condensación, conservando las mismas condiciones de temperatura y presión atmosférica.

$$RH = \frac{P_{(H_2O)}}{P^*_{(H_2O)}} \times 100\%$$

$P_{(H_2O)}$  es la presión parcial de vapor de agua en el aire.

$P^*_{(H_2O)}$  es la presión de saturación en el aire

**2.222 Humedad de suelo:** Se denomina humedad de suelo a la cantidad de agua existente en este. El manejo de esta variable es muy importante en todo tipo de cultivos, ya que esta nos daría el punto en que la planta se puede marchitar o secar, además de tener en cuenta el ahorro de agua que significa no regar de más

La capacidad de almacenamiento del agua del suelo de acuerdo a su textura se puede apreciar en el siguiente recuadro.

Textura del suelo	Capacidad	agua disponible para la planta
Arena	10%	6,00%
Arena Margosa	16%	9%
Marga arenosa	21%	12%
Marga	27%	15%
Marga Limosa	30%	15%
Marga arcillo-arenosa	36%	16%
Arcilla arenosa	32%	14%
Marga arcillosa	29%	11%

Marga arcillo- limosa	28%	13%
Arcilla limosa	40%	20%
Arcilla	40%	18%

“Para el cálculo de la cantidad absoluta de agua en el suelo, en milímetros y para la determinación de la cantidad de agua en el suelo necesaria para las plantas, se precisan los siguientes datos agros hidrológicos:

1. Constantes importantes: peso por volumen, de suelo; humedad de marchitamiento (coeficiente de marchites); higroscopicidad máxima.
2. Otras constantes: capacidad máxima de agua del suelo; capacidad capilar de agua; capacidad mínima de agua, peso específico del suelo.

Peso por volumen del suelo: Se entiende por tal el peso de 1 cm cúbico de suelo, tal como existe en la naturaleza. Este dato es necesario para el calculo del contenido absoluto de agua en el suelo, en milímetros, ya que se medirán en esta unidad, todos los elementos de contenido de agua; la ganancia de humedad en forma de precipitación, la perdida por evaporación y escurrimiento

El contenido absoluto de agua en el suelo, en mm, esta dado por la formula:

$$V = p \cdot d \cdot h \cdot 10 / 100$$

V: existencia en mm, de la capa de agua en el suelo

P: contenido de agua en el suelo, en % de suelo seco

D: peso por volumen del suelo, en el horizonte dado

H, espesor de la capa de suelo, en mm (Golsov et al, 1975)

## 2.3 INSTRUMENTACIÓN EXISTENTE.

A continuación se explicará a modo general algunos de los instrumentos que existen para medir las variables que se van a utilizar, principalmente los que den sus datos en magnitud eléctrica.

### 2.31 Sensores de temperatura.

**Variaciones de Resistencia de un Conductor:** este tipo de termómetros aprovechan la dependencia de la resistencia eléctrica con la temperatura. Los materiales más utilizados son el platino, el níquel, el cobre y el tungsteno

**Diferencia de Potencial creada a partir de la unión de dos metales:** este fenómeno se conoce como efecto Seebeck, y consiste en la conversión de una diferencia de temperatura en electricidad. Si se tienen dos materiales juntos y se produce una diferencia de temperatura en su unión, esto induce una diferencia de potencial.

### 2.32 Sensores de humedad ambiental.

**Sensores mecánicos:** estos aprovechan las deformaciones que tienen algunas fibras en presencia de humedad, aumentando su tamaño. Esta deformación de alguna manera puede ser traducida a alguna variable eléctrica. Su rango de operación es de humedades relativas entre 15% y 95%, a una temperatura ambiente entre los 20[°C] y 70[°C].

**Sensores por bulbos secos y húmedos:** se colocan dos materiales termo resistivo, uno en seco y otro en contacto con algún material que absorba la humedad de un recipiente por capilaridad, esto hace que el registro de temperatura de estos sea diferente. Con este tipo de medición se logran valores de Hr con una precisión del 99.5%, sin embargo desventaja radica en que introduce vapor de agua al ambiente que se desea medir, lo cual en algunos procesos puede ser un inconveniente. Los rangos de operación van de 0% a 100% de humedad relativa, y de 0[°C] a 90[°C].

**Sensores electrolíticos:** este sensor aprovecha la particularidad del agua para descomponerse en electrolisis, lo cual produce corriente eléctrica. Se colocan dos filamentos cargados y entremedio una película higroscópica que absorbe el agua en el aire.

**Sensores por conductividad:** en este sensor se colocan dos hilos conductores no unidos el uno con el otro, alimentados con un voltaje continuo, al haber presencia de vapor de agua entre estos se produce una circulación de corriente proporcional a la humedad del aire.

**Sensores capacitivos:** estos funcionan bajo el principio de que la capacitancia de un condensador puede variar según la constante dieléctrica, y se usa como dieléctrico el aire entre las placas del condensador. Esto se puede aprovechar a través de un puente de wheastone, circuitos resonantes y otros. Son uno de los tipos más utilizados gracias a su fácil producción bajo costo y fidelidad. Son robustos y tienen excelente precisión, además de operar en rangos de temperaturas de  $-80^{\circ}\text{C}$  a  $60^{\circ}\text{C}$ .

**Sensores infrarrojos:** se aprovecha de la capacidad el agua de absorber radiaciones de  $1400\lambda$  y  $1900\lambda$ , que están en el espectro infrarrojo, así se puede hacer una comparación entre un rayo infrarrojo que llega a una fotorresistencia sin nada en el camino y otra que pasa a través del aire que contiene el vapor de agua. Los sensores que utilizan este método son muy sensibles y logran precisiones desde 0,05 a 30.000 PPM, en rangos de temperatura que comprenden de  $-85^{\circ}\text{C}$  a  $40^{\circ}\text{C}$ .

**Sensores piezoeléctricos:** estos utilizan cristales, los cuales tienen una frecuencia de oscilación muy constante y al cambiar la masa de este por deposiciones cambia su frecuencia de oscilación. Con este sistema se pueden detectar hasta variaciones de  $\pm 0.1\text{Hz}$ , lo cual equivale a humedades de alrededor de 0.1 PPM.

### 2.33 Sensores de humedad de suelo

Existen dos formas sencillas de realizar esta medida, la primera consiste en aprovechar la resistividad del suelo, el cual varía con la humedad del suelo, esto se puede capturar enterrando

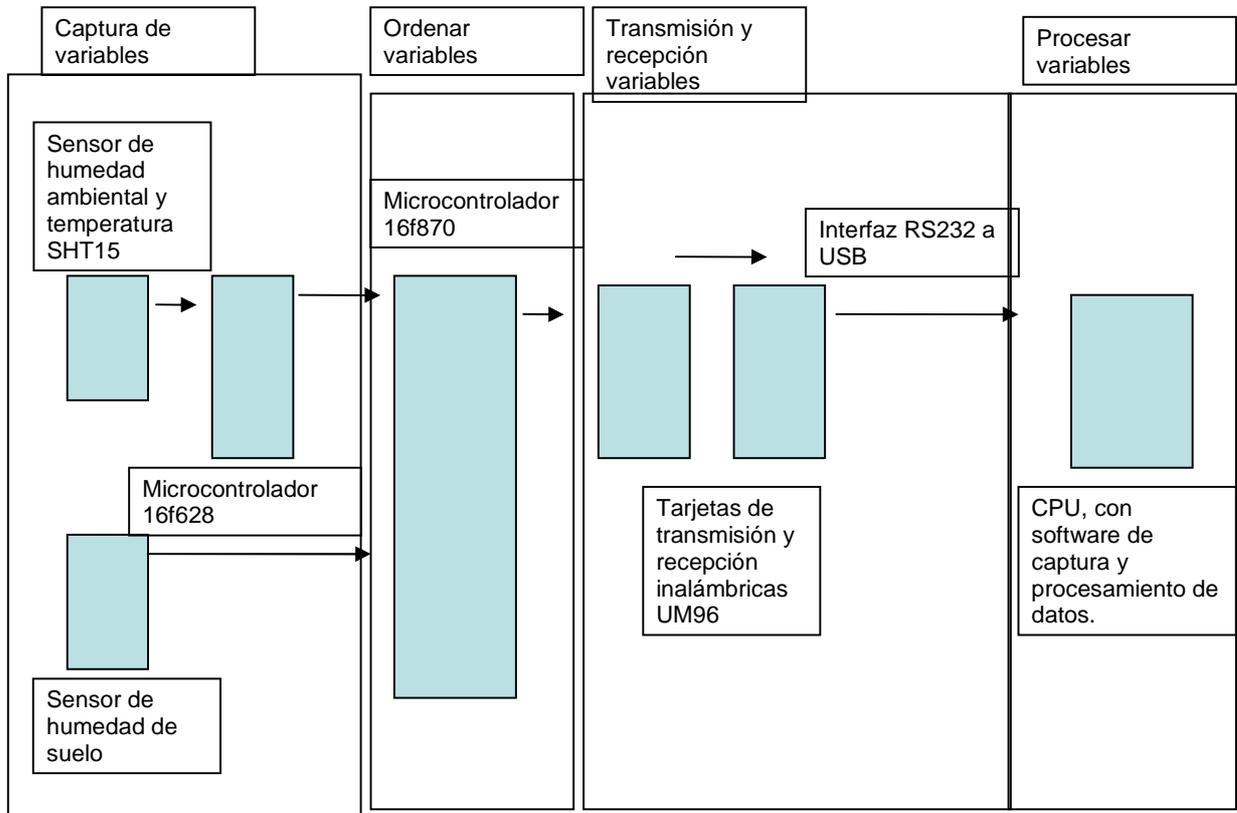
electrodos en la tierra y aplicándoles una diferencia de potencial, las distintas variaciones de corriente nos darían nuestra medida; hay que tener en consideración que es recomendable utilizar voltaje alterno en la tierra para evitar la electrolisis del terreno. Otra forma es utilizando el suelo como dieléctrico, colocando dos placas enterradas paralelamente, y la constante dieléctrica será proporcional a el agua que contenga el terreno. Es por esto que basta recubrirlos con un material higroscopico que absorba las moléculas de agua en el aire, y es esta variación de frecuencia la que daría la medición de humedad.

### **3 CAPITULO III: DISEÑO E IMPLEMENTACIÓN DEL MONITOR.**

El diseño e implementación del monitor esta determinado por la realización de una etapa de estudio, que dio como resultado que las variables más relevantes a controlar en un invernadero son la temperatura ambiente, la humedad ambiente y la humedad del suelo; al igual es importante que estas variables puedan ser transmitidas de forma inalámbrica por el tipo de emplazamiento que utilizan los invernaderos en la actualidad. Según estos requerimientos, con los que debería contar el monitor, los cuales serian principalmente la toma de variables, la transmisión de estas, para su posterior recepción y almacenamiento de los datos.

Según lo antes señalado, es que el monitor estaría dividido en 5 etapas principales:

- Captura de variables.
- Ordenar variables.
- Transmisión.
- Recepción.
- Procesar variables.



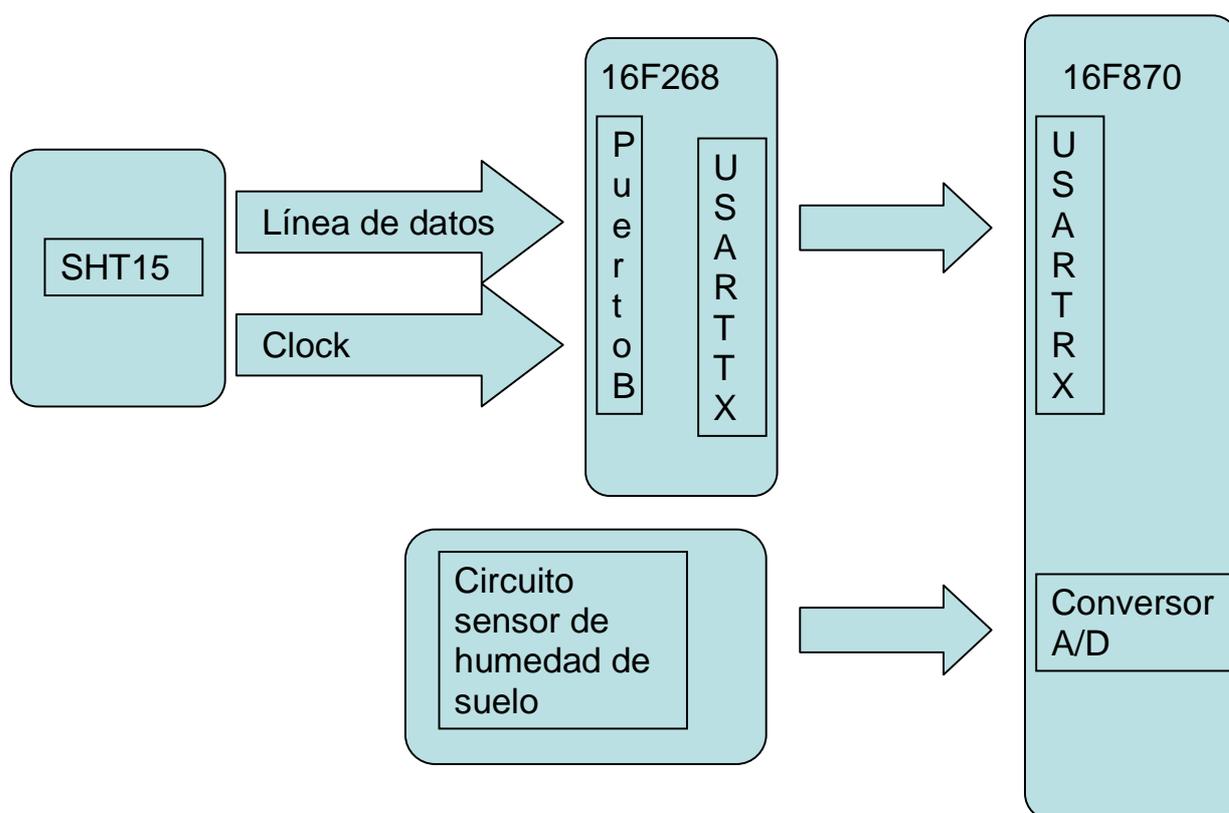
*Cuadro divisorio de las etapas del monitor*

### 3.1 ETAPA DE CAPTURA DE VARIABLES

Es aquí donde se van a tomar las variables relevantes para el funcionamiento del invernadero, las cuales son: la temperatura ambiente, la humedad ambiente y la humedad del suelo. Aquí se utilizarán sensores que capturen estas variables para posteriormente reunirlos en el siguiente módulo.

Se utilizó para la captura de temperatura y humedad ambiental el sensor SHT15, que para comunicarse utiliza comunicación serial, utilizando 2 pines, uno de clock y otro de datos. Para la comunicación con este sensor se utilizó el microcontrolador 16F628. Se utilizó este microcontrolador de forma específica para el sensor, para evitar problemas de sincronización y además de que se programó con lenguaje C, el cual es distinto al Assembler con el que se programó el 16F870.

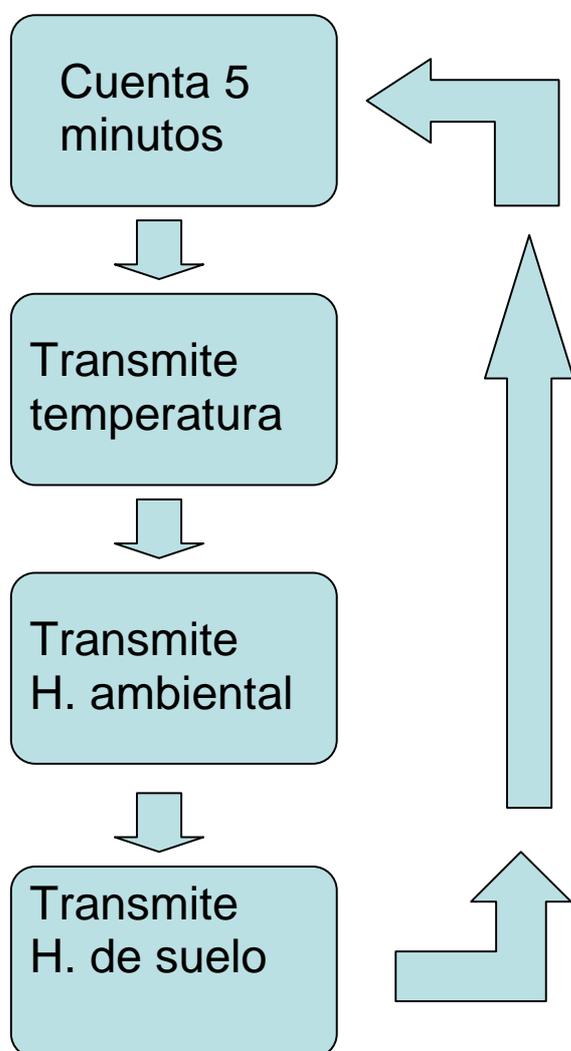
Para la obtención de la humedad del suelo se utilizó un sensor de yeso, que aprovecha la característica de este material para absorber la humedad, cambiando sus características eléctricas con esto, este sensor está puesto en un circuito simple con una resistencia fija. Al cambiar la resistencia en el sensor, provoca un cambio en la corriente en el circuito, lo que a su vez provoca un cambio en la tensión de la resistencia, y es esta tensión, la que se mide en el conversor analógico digital del microcontrolador 16F870.



*Desglose de la etapa de captura*

### 3.2 ETAPA DE ORDENAR VARIABLES

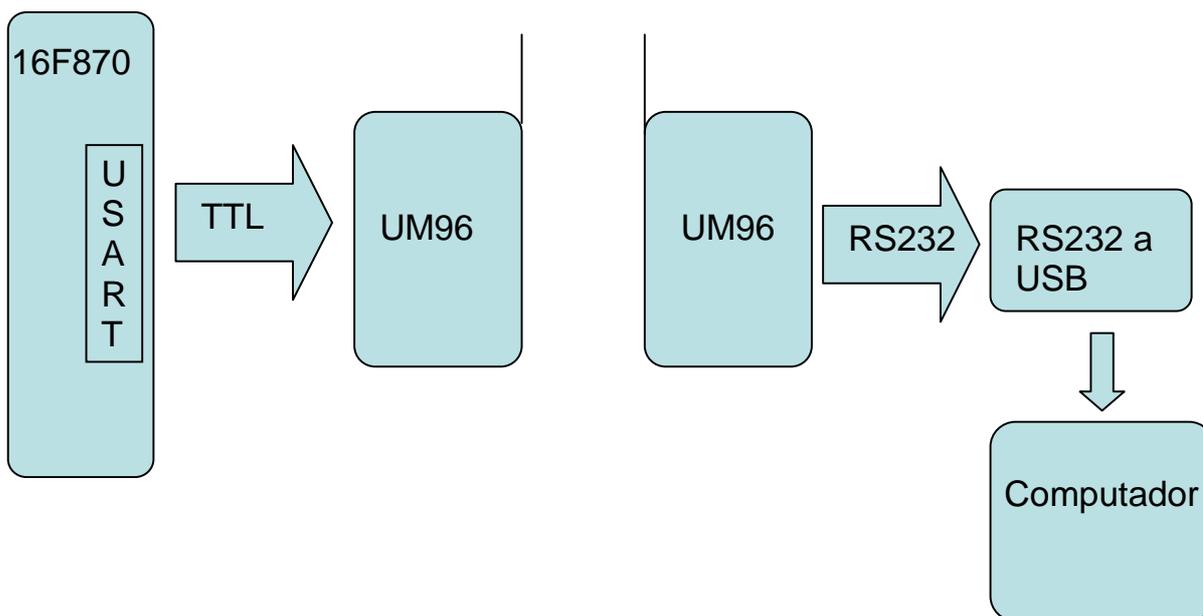
Es donde se toman las variables y ordenan para su posterior transmisión. Para esta función se utilizara el microcontrolador 16F870. Esta programado en Assembler, y el programa lo que hace es cada cierto tiempo activar la interrupción por recepción serial, toma los datos que envía el sensor SHT15 desde el microcontrolador 16F628, y los envía por el puerto serial asíncrono que posee o USART, luego prende el conversor análogo digital tomando el dato, y también lo envía por el mismo puerto.



*Desglose de la etapa de ordenar variables*

### 3.3 ETAPA DE TRANSMISIÓN Y RECEPCIÓN

Es donde se envían los datos de forma inalámbrica para su posterior recepción, con tarjetas diseñadas para este motivo considerando su flexibilidad, alcance y modulación. Para esto fueron utilizadas las tarjetas UM96, que cumplen con los requerimientos antes mencionados, como lo son la facilidad de conexión tanto al microcontrolador, como a la comunicación RS232; no necesitando hardware aparte para su conexión a estos. Tienen un alcance de 500 metros, que es distancia suficiente para llegar a su central. Tienen varios canales de comunicación, lo que facilita el hecho de que haya varios módulos funcionando a la par, con una modulación GFSK. La tarjeta de recepción se conecta a un adaptador de RS232 a USB, para garantizar la compatibilidad con computadores que no consten con el puerto RS232, como es el caso de los actuales computadores personales.



*Desglose de la etapa de transmisión y recepción.*

### **3.4 ETAPA DE PROCESAR VARIABLES**

En esta última etapa se recopilan los datos para su almacenamiento y posterior análisis por la gente interesada. Para esto se utilizara un computador, con un software que sea capaz de recopilar los datos y redirigirlos a un programa que sea capaz de realizar el análisis de estos. El software utilizado fue Visual Basic, debido a su entorno fácil de utilizar para el usuario, además de tener compatibilidad con Excel, que es el programa que mas se utiliza para a el análisis de datos de estas características.

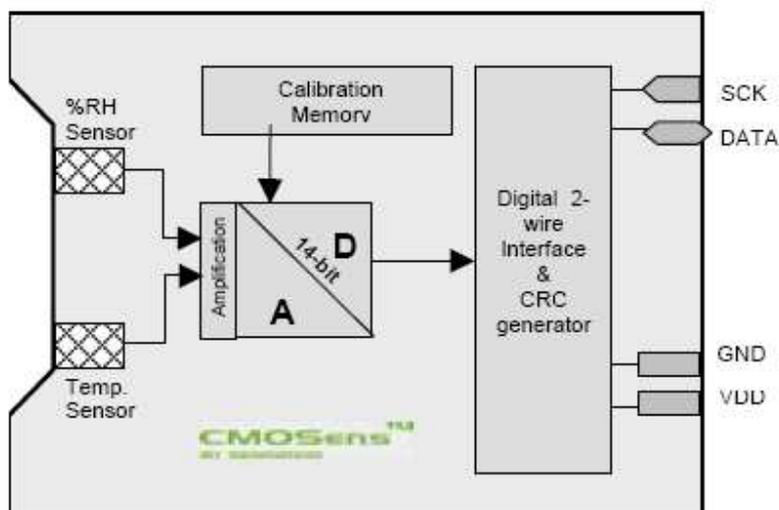
El programa lo que hace es abrir el puerto serial, que crea virtualmente el cable de adaptación de RS232 a USB, lee los datos que le están entrando, los reconoce y diferencia, para luego dejarlos en un formato entendible para la plantilla Excel.

## 4 CAPITULO IV: DISPOSITIVOS UTILIZADOS

### 4.1 SENSORES

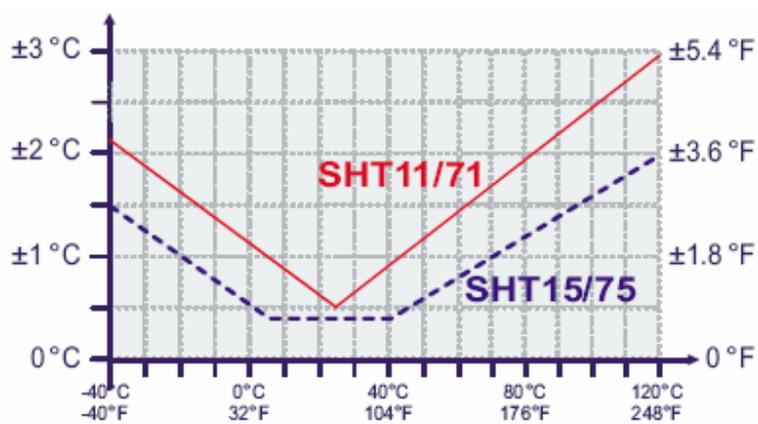
#### 4.1.1 Sensor SHT15

Para las variables de temperatura ambiente y humedad ambiente utilizare el sensor sht15, de la empresa Sensirion. Este dispositivo incluye un polímero capacitivo como elemento sensor de humedad y un sensor de temperatura bandgap, esto unido a un conversor análogo digital de 14 bits y como salida una interfaz serial con código de redundancia cíclica para la detección de errores.

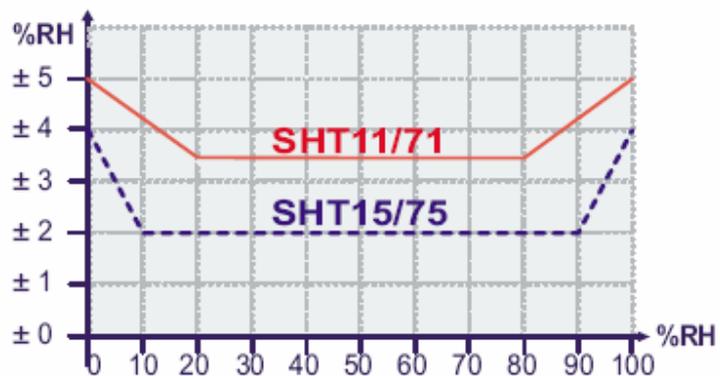


*Esquema interno del sensor SHT15*

En los siguientes gráficos se muestra la precisión que tiene este sensor.



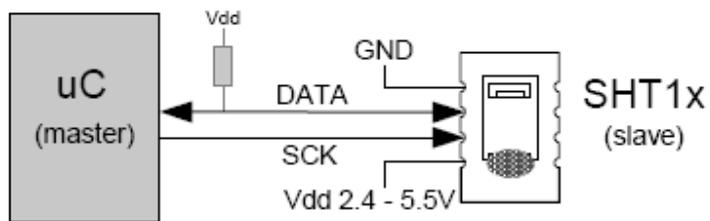
*Gráfico de comportamiento para la medición de temperatura.*



*Gráfico de comportamiento para la medición de humedad.*

#### 4.111 Comunicación con el sensor SHT15

Este sensor ocupa una interfase serial de dos hilos, uno que funciona de clock y otro para los datos.



*Conexión del sensor con el microcontrolador.*

Ocupando una comunicación muy similar al I2C, pero no es compatible con interfaces de este tipo de comunicación. La línea de clock se utiliza para sincronizar el microcontrolador y el sensor. La línea de datos se utiliza tanto para dar instrucciones al SHT como para obtener datos de este.

Para iniciar la comunicación con el sensor hay que ingresar una instrucción de la siguiente forma:

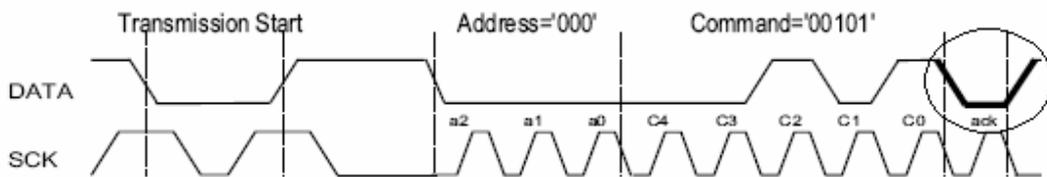


*Trama de inicio del sensor*

Luego se envían 3 ceros consecutivos seguidos de la instrucción que se desea aplicar:

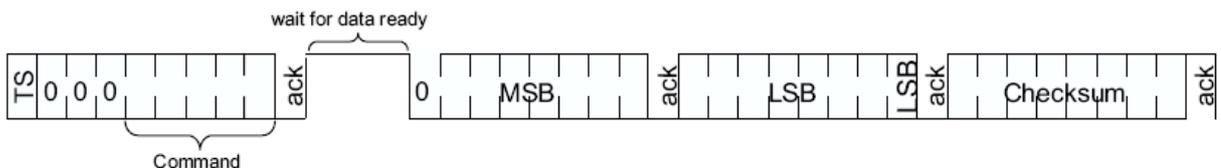
Medición de temperatura	0 0 0 1 1
Medición de humedad	0 0 1 0 1
Leer la condición del registro	0 0 1 1 1
Escribir la condición del registro	0 0 1 1 0

Para la recepción de la medida, sea de temperatura como de humedad, luego de enviar el código como se describió anteriormente, el controlador tiene que esperar a que se efectuó la medida, esto demora 11/55/210 ms para una medición de 8/12/14 bit, lo cual puede variar mas menos 15% según el oscilador interno. Con el fin de señalar la finalización de la medición el sht15 tira hacia abajo la línea de datos, por lo cual el controlador tiene que esperar esta señal para empezar a cambiar de nuevo el sck.



*Trama para la entrega de la medición de humedad.*

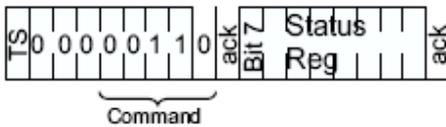
El sensor responde con tres bytes, 2 de datos y uno de corrección de errores o checksum. Esta corrección de errores se hace a través de código de redundancia cíclica. Si es que no se utiliza la comprobación de errores terminaría de transmitir luego de la medición. El sensor vuelve a reposo automáticamente luego de finalizada la medición y la comunicación.



*Trama completa de comunicación con el sensor.*

#### 4.112 Registro Status

Algunas de las funciones avanzadas del sensor se encuentran disponibles a través de este registro,



*Trama de configuración del registro status*

Bit	Tipo	Descripción	por defecto
7		Reservado	0
6	R	Fin de la batería( detección de bajo voltaje) 0 para Vdd > 2,47V 1 para Vdd < 2,47V	x
5		Reservado	0
4		Reservado	0
3		Solo para pruebas, no utilizar	0
2	R/W	Calentador	0 apagado
1	R/W	no reload from OTP	0 reload
0	R	1 = 8bit Humedad relativa/12bit Temperatura 0 = 12bit Humedad relativa/14bit Temperatura	0 12 bit humedad relativa 14 bit temperatura

*Cuadro detallado de los bit del registro status*

Resolución de la medición: El valor por defecto de la temperatura y la humedad es de 14 y 12 bit, lo cual se puede reducir a 12 y 8 bit, siendo especialmente útil para aplicaciones de alta velocidad o muy baja potencia.

Fin de batería: su función es detectar voltajes por debajo de los 2,47V con una precisión más o menos 0,05V.

Calentador: es un elemento de calefacción que puede aumentar la temperatura del sensor en aproximadamente 5°C ( 9F), para este efecto el consumo de energía aumentara en 8mA a 5V.

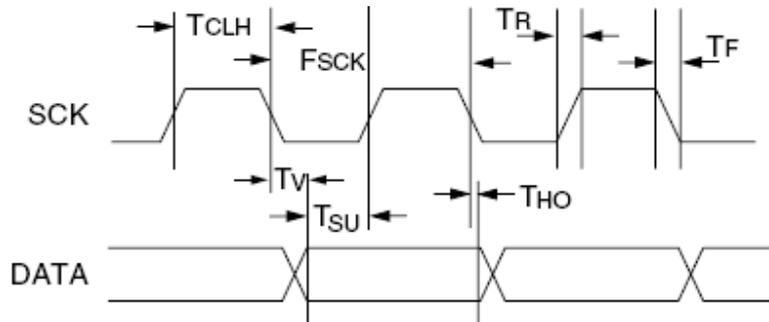
Esto se puede utilizar para humedades muy altas, evitando la condensación en el sensor, lo cual produciría fallas en su medida. Eso si que mientras se calienta el sensor se muestran temperaturas mas altas y humedad relativa mas baja que sin el calentador.

#### 4.113 Características eléctricas:

Parámetros	Condiciones	Mínimo	Típico	Máximo	Unidades
Alimentación		2,4	5	5.5	V
Corriente	Midiendo		550		$\mu$ A
	promedio	2	28		$\mu$ A
	sleep		0,3	1	$\mu$ A
Bajo nivel de voltaje de salida		0		20%	Vdd
Alto nivel de voltaje de salida		75%		100%	Vdd
Bajo nivel de voltaje de entrada		0		20%	Vdd
Alto nivel de voltaje de entrada		80%		100%	Vdd

	Parameter	Conditions	Min	Typ.	Max.	Unit
<b>F<sub>SCK</sub></b>	SCK frequency	VDD > 4.5 V			10	MHz
		VDD < 4.5 V			1	MHz
<b>T<sub>RFO</sub></b>	DATA fall time	Output load 5 pF	3.5	10	20	ns
		Output load 100 pF	30	40	200	ns
<b>T<sub>CLx</sub></b>	SCK hi/low time		100			ns
<b>T<sub>V</sub></b>	DATA valid time			250		ns
<b>T<sub>SU</sub></b>	DATA set up time		100			ns
<b>T<sub>HO</sub></b>	DATA hold time		0	10		ns
<b>T<sub>R</sub>/T<sub>F</sub></b>	SCK rise/fall time			200		ns

*Tabla de tiempos de respuesta.*



*Esquema de tiempos de respuesta.*

#### 4.114 Conversión a los valores físicos.

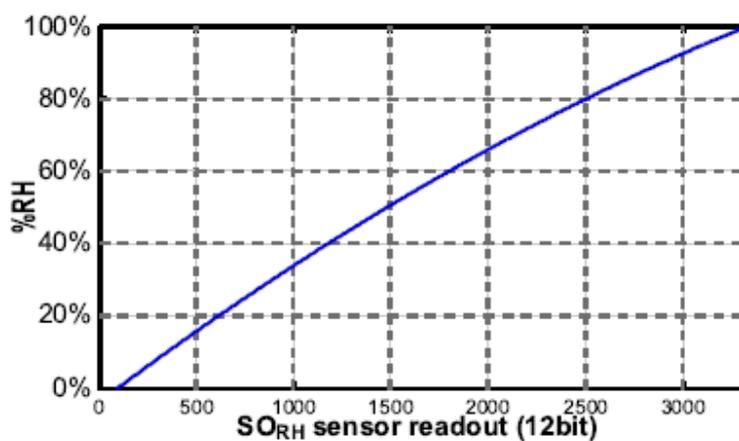
**Humedad Relativa:** para compensar la no linealidad del sensor de humedad y obtener la plena exactitud se debe aplicar la siguiente formula:

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

SO <sub>RH</sub>	c <sub>1</sub>	c <sub>2</sub>	c <sub>3</sub>
12 bit	-4	0.0405	-2.8 * 10 <sup>-6</sup>
8 bit	-4	0.648	-7.2 * 10 <sup>-4</sup>

*Formula para calcular la humedad relativa*

Estas medidas no tienen dependencia con el voltaje de entrada.



*Curva de respuesta de la humedad*

Para temperaturas notoriamente diferentes a los 25°C se debe aplicar esta otra formula:

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

SO <sub>RH</sub>	t <sub>1</sub>	t <sub>2</sub>
12 bit	0.01	0.00008
8 bit	0.01	0.00128

*Formula para calcular la humedad relativa a temperaturas superiores a los 25°C*

**Temperatura:** esta medida a diferencia de la anterior es mucho más lineal y basta con aplicar la siguiente formula:

$$Temperature = d_1 + d_2 \cdot SO_T$$

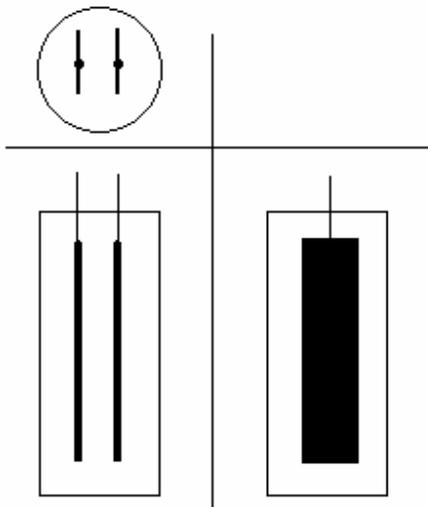
VDD	d <sub>1</sub> [°C]	d <sub>1</sub> [°f]
5V	-40.00	-40.00
4V	-39.75	-39.50
3.5V	-39.66	-39.35
3V	-39.60	-39.28
2.5V	-39.55	-39.23

SO <sub>T</sub>	d <sub>2</sub> [°C]	d <sub>2</sub> [°f]
14bit	0.01	0.018
12bit	0.04	0.072

*Formula para calcular la temperatura*

#### 4.12 Sensor de Humedad de suelo

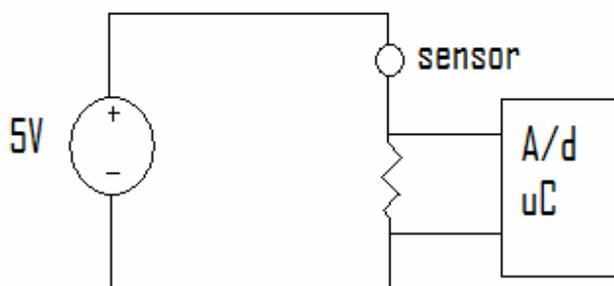
Este esta construido con yeso, el cual es un material higroscópico, este tipo de materiales tiene la característica de absorber o ceder humedad al medioambiente, característica que es aprovechada para medir el cambio en la resistencia del yeso por presencia de agua. Contra mas agua absorbe el yeso disminuye su resistencia y viceversa, contra menos agua tenga aumenta su resistencia. Para poder sensor esta variable se colocaron dentro del bloque de yeso dos placas de cobre. La forma en que se diseño se muestra en el siguiente dibujo:



*Esquema del sensor de humedad de suelo*

Sus dimensiones físicas son de 4 cm de alto por 1,5 cm de diámetro, como molde se uso un tubo de pvc. Las placas de cobre son de 3cm de alto por uno de ancho, tienen una separación entre ellas de 2mm.

Este sensor va unido a una resistencia en serie, y estos a una fuente de 5 v, produciendo un voltaje tanto en la resistencia como en el sensor, ya que el sensor varía su resistencia, esto influye en la caída de tensión de la resistencia fija, y es aquí el voltaje a testear con el conversor análogo digital del microcontrolador



*Circuito de conexión para el sensor de humedad de suelo*

Se realizaron pruebas para ver la variación entre el estado de tierra seca y tierra totalmente mojada, lo que correspondería a humedad de 0% y 100% respectivamente, colocando como resistencia de prueba una de  $1K\Omega$ , excitando el circuito con una fuente de 5V. Como resultado dio que para el suelo totalmente seco el voltaje en la resistencia era de 0,01 V y para suelo totalmente mojado de 4,99 V. Con estos datos podemos tomar como parámetros los 0V correspondientes a 0% de humedad, y los 5V como 100% de esta, todo esto aproximadamente.

#### **4.2 Modulo transmisor y receptor HAC-UM96.**

Se ha elegido el modulo HAC-UM96, ya que para las necesidades de alcance es el más adecuado, pudiendo alcanzar los 500mts en línea vista, distancia más que necesaria para el uso en invernaderos, además es de bajo consumo y económico.



*Modulo transmisor y receptor HAC-UM96*

Este dispositivo tiene una potencia de transmisión de 10mw, con modulación GFSK (Gaussian Frequency-Shift Keying), o Modulación por desplazamiento de frecuencia gaussiana.

#### 4.21 Disposición de pines

La disposición de pines que posee es la siguiente:

Pin N°	Nombre	Descripción	Nivel	Conectado al Terminal	Notas
1	GND	Tierra de la fuente		Tierra de la fuente	
2	Vcc	Voltaje de Alimentación	+3.3~5.5V		
3	RxD/TTL	Recepción serial de datos	TTL	TxD	COM1
4	TxD/TTL	Transmisión serial de datos	TTL	Rxd	
5	SGND	Tierra de la señal			
6	A(TxD)	A del RS-485 o TxD del RS-232		A(RxD)	COM2
7	B(RxD)	B del RS-485 o RxD del RS-232		B(TxD)	
8	<u>SLEEP</u>	Sleep control (input)	TTL	Señal para ahorro de energía	Baja eficiencia t>15ms
9	<u>RESET</u>	Reset (input)	TTL	Señal para reiniciar el sistema	Aplicar Impulso negativo para reiniciar

*Cuadro de disposición de los pines del modulo UM96*

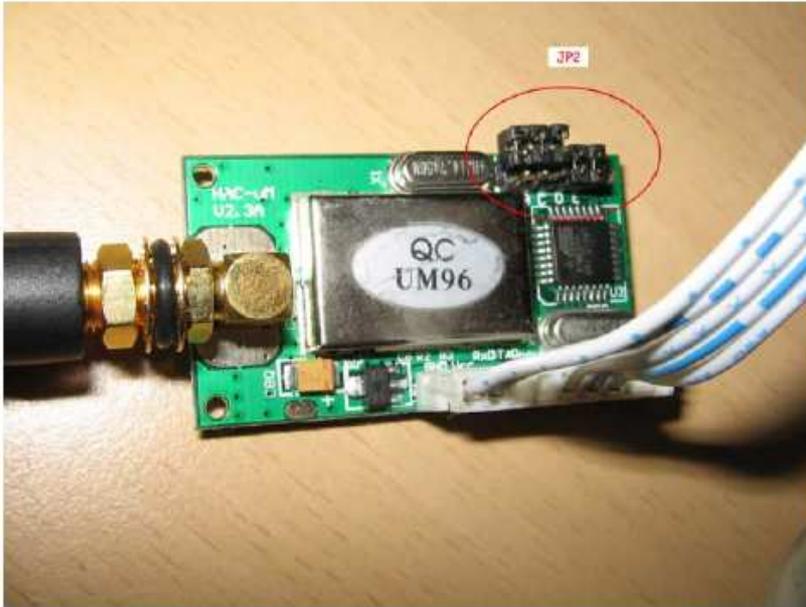
Como se puede apreciar, el UM-96 posee dos puertos, uno para niveles TTL y el segundo para protocolo RS232 o RS485; otorgando con esto una gran adaptabilidad en el uso que se le vaya a dar.

También tiene la posibilidad de elección de 8 canales diferentes, lo cual da flexibilidad a la hora de elegir la frecuencia a la cual funcione el dispositivo.

Canal N°	Frecuencia	Canal N°	Frecuencia
CBA=000(0)	430.2000 Mhz	CBA=100(4)	434.6940 Mhz
CBA=001(1)	431.4288 Mhz	CBA=101(5)	434.2332 Mhz
CBA=010(2)	431.7360 Mhz	CBA=110(6)	433.1580 Mhz
CBA=011(3)	430.5072 Mhz	CBA=111(7)	433.9260 Mhz

*Configuración de canales de comunicación.*

Para configurar el dispositivo, hace por medio de jumpers, estos van de la A a la E y están dispuestos de izquierda a derecha como se muestra en la siguiente figura:



*Pines de configuración.*

Los jumpers A, B y C son para la configuración de los canales de transmisión y recepción, el jumper D es para la elección entre el protocolo RS 232 y el RS 485.

- D = 1 (sin corto circuito). COM2 = RS-485
- D = 0 (con corto circuito). COM2 = RS-232

Y el E que se utiliza para ver si la comunicación es con o sin paridad.

- E=1 (sin corto circuito). Paridad: 8E1
- E=0 (con corto circuito). Paridad: 8N1

Estos módulos utilizan una alimentación entre 3,3 y 5,5V, este voltaje puede reducirse hasta 3V según las necesidades del usuario. No tiene problemas de conectarse a la misma fuente que otros

equipos, pero eso si hay que tener en cuenta el rizo de la señal, para lo que se recomienda colocar algún estabilizador de voltaje en la entrada.

En condiciones de trabajo, los consumos de corriente son: cuando transmite de menor o igual a 40mA, cuando recibe de menor o igual a 30mA, y en modo sleep de menor o igual a 20µA.

#### 4.22 Modulación GFSK

La modulación GFSK, viene de las siglas en ingles Gaussian Frequency shift keying, que traducido al español es modulación por desplazamiento de frecuencia gaussiana. La modulación GFSK es una versión mejorada de la FSK, ya que la información es pasada por un filtro gaussiano antes de modular la señal para hacer más suave el pulso de manera de limitar su ancho espectral.. Esto se traduce en un espectro de energía mas estrecho de la señal modulada, lo que permite mayores velocidades de transferencia sobre un mismo canal.

La modulación FSK (modulación por desplazamiento de frecuencia), es una forma de modulación angular d amplitud constante, es similar a la modulación de frecuencia tradicional, pero difiere en que la señal modulante es un flujo de pulsos binarios que varia entre dos niveles de voltaje discreto, en ves de una señal analógica que cambia en forma continua.

La expresión para la señal FSK viene dada por la siguiente formula:

$$v(t) = V_c \cos \left[ \left( \omega_c + \frac{\omega_m(t) \Delta \omega}{2} \right) t \right]$$

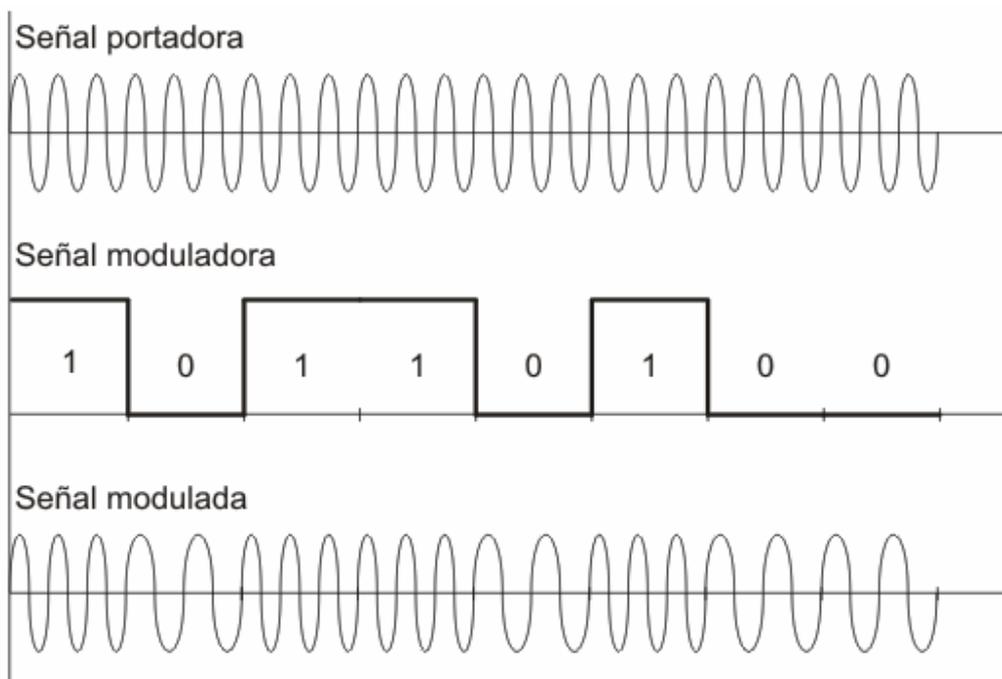
$V_c$ : Amplitud de la portadora no modulada

$\omega_c$ : frecuencia de la portadora en radianes

$\omega_m(t)$ : señal modulante digital binaria

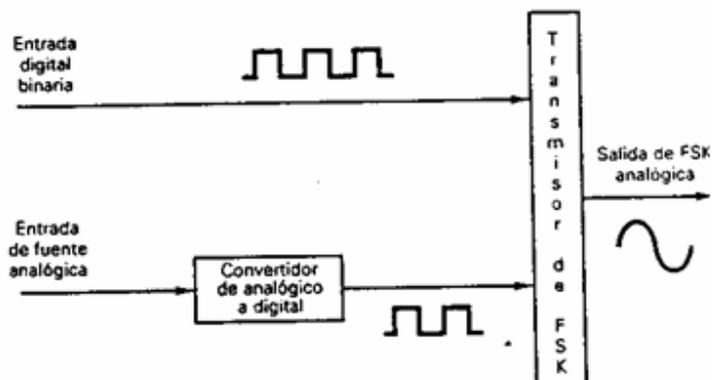
$\Delta \omega$ : cambio en frecuencia de salida en radianes

De esta ecuación se puede ver que la amplitud de la portadora es siempre constante, en cambio la frecuencia en radianes de la portadora cambia por una cantidad igual a  $\pm \Delta \omega/2$ . El cambio de la frecuencia es proporcional a la amplitud y polaridad de la señal de entrada binaria. Además la rapidez a la que cambia la frecuencia de la portadora es igual a la rapidez de cambio, de la entrada binaria  $V_m(t)$ .



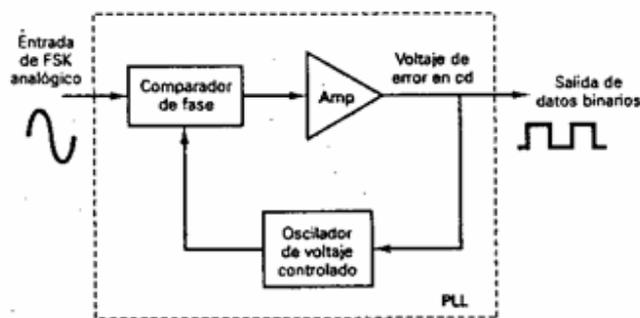
*Señal modulada FSK*

### **Transmisor FSK**



*Transmisor FSK*

La salida de un modulador FSK binario, es una función escalón en el dominio del tiempo. Conforme cambia la señal de entrada binaria de 0 lógico a 1 lógico y viceversa, la salida del FSK se desplaza entre 2 frecuencias, una frecuencia de marca o de 1 lógico y una frecuencia de espacio o de 0 lógico. Con el FSK binario, hay un cambio en la frecuencia de salida, cada vez que la condición lógica de la señal de entrada binaria cambia.



#### *Desmodulación señal FSK*

El circuito que más se utiliza para demodular las señales de FSK binarias es el circuito de fase cerrada (PLL). Conforme cambia la entrada de PLL entre las frecuencias de marca y espacio, el voltaje de error de cc a la salida del comparador de fase sigue el desplazamiento de frecuencia. Debido a que sólo hay dos frecuencias de entrada, también hay sólo dos voltajes de error de salida. Uno representa un 1 lógico y el otro un 0 lógico. En consecuencia, la salida es una representación de dos niveles (binaria) de la entrada de FSK. Por lo regular, la frecuencia natural del PLL se hace igual a la frecuencia central del modulador de FSK. Como resultado, los cambios en el voltaje de error cc, siguen a los cambios en la frecuencia de entrada analógica y son simétricos alrededor de 0 V.

### 4.3 MICROCONTROLADORES.

Los microcontroladores son circuitos integrados programables, capaces de ejecutar las instrucciones que existen en su memoria. Principalmente constan de un microprocesador, líneas de entrada y salida, memorias RAM y ROM. Para su funcionamiento requiere de alimentación, de un oscilador, y de un programa de instrucciones.

La función principal del microcontrolador es interpretar combinaciones de bits y generar señales digitales internas y o externas, para ejecutar de manera continua una secuencia de instrucciones que permita controlar un sistema o subsistema electrónico. Estos dispositivos vienen con un juego de instrucciones reducido, además de su pequeño encapsulado con pocos pines y poco consumo, lo cual los hacen muy utilizables.

La mayoría de los microcontroladores constan con las siguientes características:

Procesador o CPU: es quien procesa todos los datos que pasan por el bus.

Memoria ROM: es la memoria no volátil, que es donde se guardan los programas

Memoria RAM: o memoria volátil, que es donde se guardan los datos.

Oscilador: que sincroniza todo el funcionamiento del sistema.

Puertos de entrada y salida: Es por donde se comunica el microcontrolador con los periféricos externos.

Convertidores análogo digital(A/D), digital análogo (D/A): como su nombre los dice convierten señales de análogas a digital y viceversa.

Temporizadores: sirven para controlar periodos de tiempo.

Comparadores: como su nombre lo indica comparan señales analógicas.

Moduladores de ancho de pulso: esta función modula en PWM.

Puerto USART: comunicación serie de transmisión y recepción.

Controladores de interrupciones

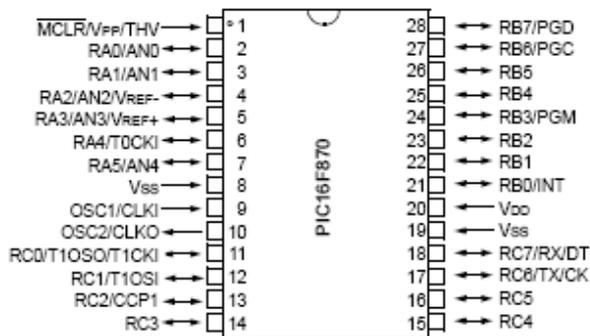
Perro guardián o watch dog: contador que resetea el microcontrolador cada vez que se desborda el stack.

Protección ante fallo de alimentación: resetea el microcontrolador cada vez que la alimentación baja de un cierto límite.

#### **4.31 El microcontrolador 16f870**

Los microcontroladores de la familia 16f87x, al igual que el resto de los microcontroladores de la actualidad, están diseñados con la arquitectura Harvard. La arquitectura Harvard dispone de dos memorias independientes; una que contiene solo instrucciones, y otra donde se almacenan los datos.

El integrado 16f870 construido por la empresa microchip, es una buena elección en cuanto a microcontroladores se habla, ya que posee tres puertos de E/S, El puerto A con 6 pines, y el puerto B y C con 8, en total tiene 28 pines, 22 de estas configurables como entrada o salidas.



### *Microcontrolador 16F870*

A continuación se darán a conocer la mayoría de características generales de este microcontrolador:

Voltaje de alimentación de 5V

Frecuencia de operación máxima de 20Mhz

Memoria flash de 4Kbytes

Memoria de datos de 192Bytes

Memoria de datos EEPROM de 128Bytes

2 temporizadores internos de 8 bites

1 temporizador interno de 16 bites

1 modulo de captura, comparación y PWM

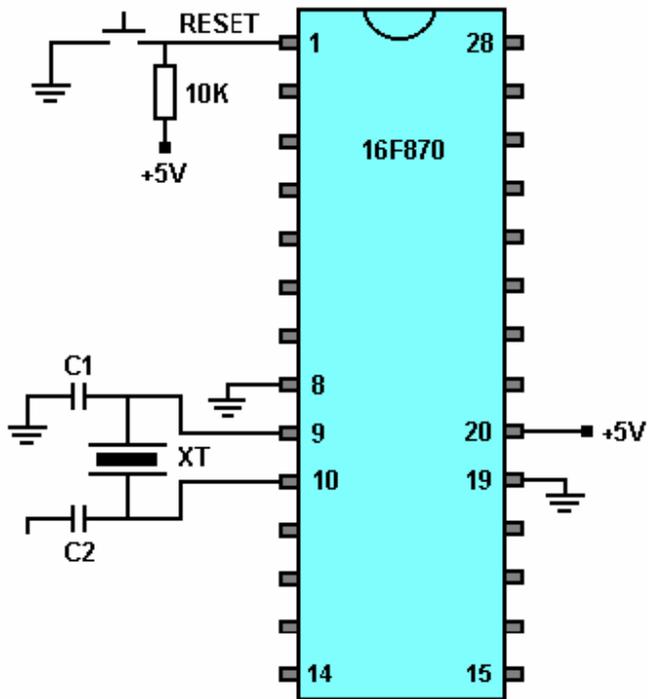
1 modulo de comunicaciones USART

5 canales de conversión analógica digital

13 interrupciones

Como se dijo anteriormente para que el microcontrolador funcione es necesario grabarle un programa en su interior. Este programa será leído línea a línea por la CPU, y para que este funcione tiene que contar con un generador de pulsos de reloj externos, el cual sincroniza el funcionamiento del sistema.

Como fuente de la señal de reloj externa generalmente se utiliza un cristal, el microcontrolador ejecuta cada instrucción (un ciclo de programa) a un cuarto de la velocidad del cristal utilizado. En general un ciclo de programa es una línea de código assembler.

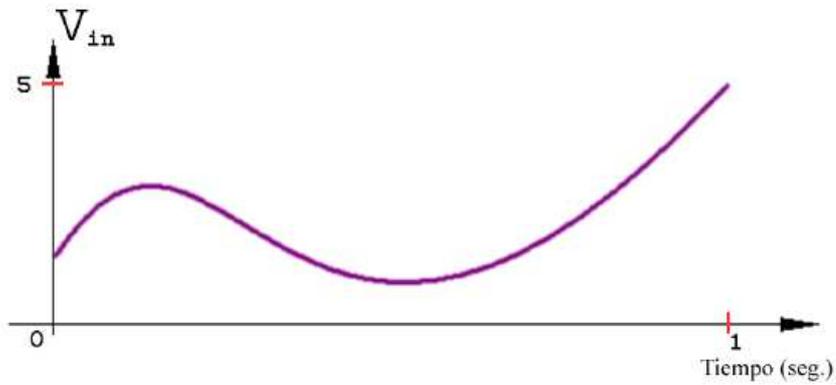


#### 4.311 Conversión Análoga Digital

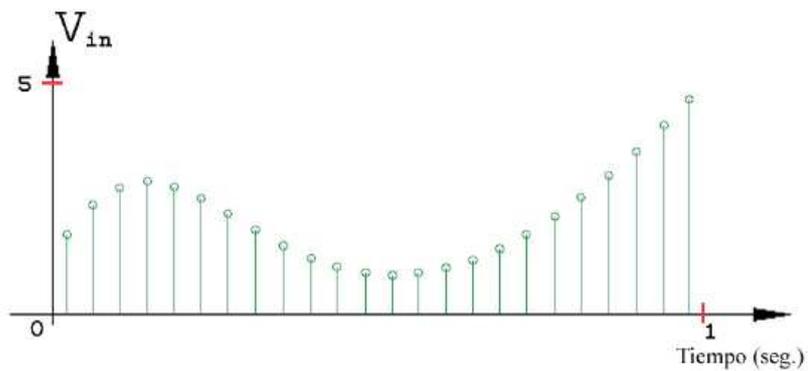
La conversión análoga digital(A/D) se realiza en el microcontrolador 16f870, y este toma la señal que viene del sensor, en este caso un voltaje continuo, y lo trasforma a un numero digital.

La conversión A/D consiste en la trascripción de una señal analógica en una señal digital, para su tratamiento como tal. Las señales análogas son señales continuas que tienen una frecuencia y amplitud, en cambio una señal digital es discreta, que tiene tiempo y amplitud, esta señal toma un

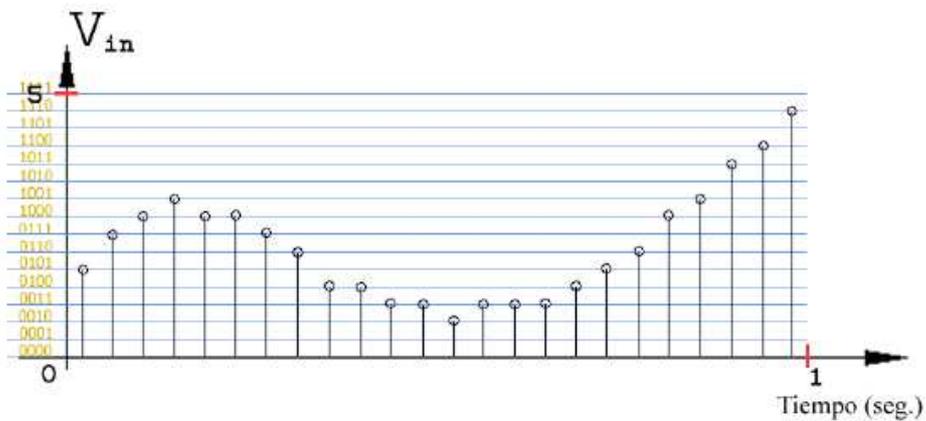
valor fijo en un tiempo determinado. Estos valores fijos se toman del sistema binario, lo que significa que la señal va a ser convertida en una combinación de ceros y unos.



*Señal analógica*



*Señal discreta*



*Cuantificación de la señal discreta*

En el proceso de conversión intervienen 4 etapas:

**Muestreo:** consiste en tomar muestras periódicas de la amplitud de la señal de entrada. La velocidad con que se realiza este proceso, el número de muestras por segundo, se le denomina frecuencia de muestreo.

**Retención:** las muestras han de ser retenidas durante un instante de tiempo para evaluar su nivel. Este más que un proceso matemático es un recurso técnico realizado por algún circuito de retención que cumple esta labor.

**Cuantificación:** en este proceso es donde se mide el nivel de voltaje de cada una de las muestras, consiste en asignar un margen de valor de la señal de entrada a un único valor de salida.

**Codificación:** consiste en traducir los valores obtenidos durante la cuantificación a código binario, Hay que tener presente que el código binario es el comúnmente utilizado pero igual existen otro tipo de codificaciones.

Cabe destacar que en los dos primeros la señal aun es analógica, solo en los dos últimos se puede catalogar como señal digital.

### **Registros del modulo A/D del 16f870**

El modulo A/D del microcontrolador tiene 4 registros, los cuales son:

ADRESH

ADRESL

ADCON0

ADCON1

ADCON0:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

Bit 7-6: ADCS1, ADCS0: es en estos bits en los que se hace la elección de la frecuencia de muestreo.

0 0 Fosc/2

0 1 Fosc/8

1 0 Foasc/32

1 1 Frc ( oscilador RC interno)

Bit 5-3, CHS2-CHS0: en estos se selecciona el canal analógico por donde entra la señal.

0 0 0 = Canal 0, (RA0/AN0)

0 0 1 = Canal 1, (RA1/AN1)

0 1 0 = Canal 2, (RA2/AN2)

0 1 1 = Canal 3, (RA3/AN3)

1 0 0 = Canal 4, (RA4/AN4)

1 0 1 = Canal 5, (RA5/AN5)

1 1 0 = Canal 6, (RA6/AN6)

1 1 1 = Canal 7, (RA7/AN7)

Bit 2, GO/DONE: bit de estado del conversor.

Bit 1, no implementado.

Bit 0, ADON: bit de puesta en marcha.

#### ADCON1

Este registro configura los pines del puerto.

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Bit 7, ADFM: Selecciona el formato del resultado de la conversión A/D

1 = >Pone en el registro **ARDESH** los seis bits de mayor peso a “0”

0 =>Pone los 6 bits de menor peso del registro **ADRESL** a “0”

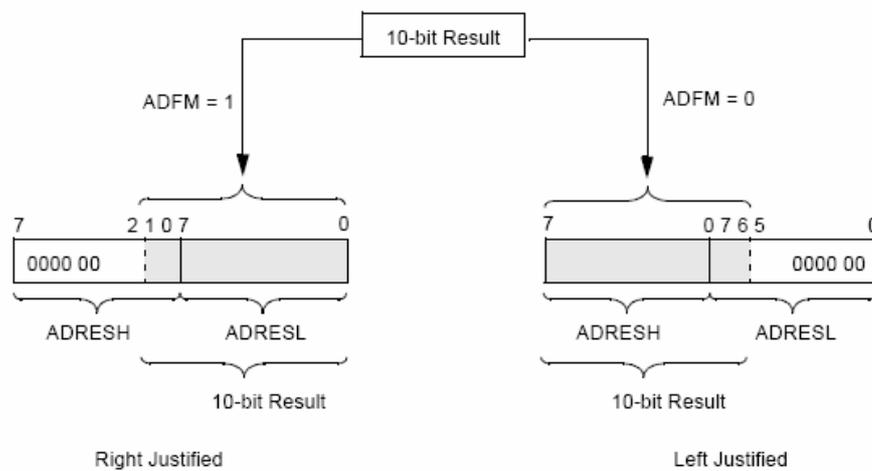
Bits 6-4: No implementados

Bit 3-0: **PCFG3:PCFG0**: bits de configuración de los canales de entrada del convertidor A/D. Se utilizan para configurar las patillas como E/S digital o como entrada analógica de acuerdo con la siguiente tabla:

PCFG3: PCFG0	AN7 <sup>(1)</sup> RE2	AN6 <sup>(1)</sup> RE1	AN5 <sup>(1)</sup> RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs <sup>(2)</sup>
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Los registros **ADRESH** y **ADRESL**

Este par de registros se cargan con 10 bit del la conversión A/D. La forma en que se distribuyen esos 10 bit en los dos registros están dados por el bit **ADFM** del registro **ADCON1**.



### 4.312 Modulo USART

El modulo USART, viene de las siglas en ingles universal synchronous asynchronous receiver transmitter, o en español transmisor-receptor serie síncrono-asíncrono universal. Como su nombre lo dice puede ser configurado de modo asíncrono, principalmente utilizado para la comunicación con computadores, y síncrono que se utiliza con dispositivos tales como memorias EEPROM u conversores análogo digitales. En este caso se utilizo la comunicación asíncrona, los datos son del porte de un byte, acompañados de un bit de inicio y otro de parada, además se puede añadir un bit de paridad.



Registros utilizados.

Los registros que usa el modulo USART para su funcionamiento tanto en el microcontrolador 16f870 como en el 16f628, son los siguientes:

Direcc.	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en: POR, BOR	Valor en otros RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	FX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	<i>Baud Rate Generator Register</i>								0000 0000	0000 0000

SPBRG: este registro es utilizado para configurar la selección de baudios, o en otras palabras la velocidad con que transmite.

Para calcular los baudios de utiliza la siguiente formula.

SYNC	BRGH = 0 (Baja velocidad)	BRGH = 1 (Alta velocidad)
0	(Asíncrono) Baudios = $F_{OSC}/(64(X+1))$	Baudios = $F_{OSC}/(16(X+1))$
1	(Síncrono) Baudios = $F_{OSC}/(4(X+1))$	N/A

X = valor en SPBRG (0 a 255)

Tabla para modo asíncrono (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	62.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

Tabla para modo asíncrono (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

TXSTA: registro utilizado para el control y estado de la transmisión.

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

Bit 7, CSRC: Selección de la fuente de la señal del reloj

Modo asincrono: no importa

Modo síncrono

1= modo maestro (utiliza el reloj generado internamente por BRG)

0= modo esclavo (utiliza un reloj externo)

Bit 6, TX9: habilitación de la transmisión de 9 bits

1= selecciona transmisión de 9-bit

0= selecciona transmisión de 8-bit

Bit 5, TXEN: Habilita transmisión

1= transmisión activada

0= transmisión desactivada

Bit 4, SYNC: selección del modo USART

1= modo síncrono

0= modo asíncrono

Bit 3, no implementado

Bit 2, BRGH: selección de velocidad.

Modo asíncrono:

1= velocidad alta

0= velocidad baja

Modo síncrono: no se utiliza en este modo

Bit 1, TRMT: bit de estado del registro de desplazamiento de transmisión.

1= TSR vacío

0= TSR lleno

Bit 0, TX9D: noveno bit de transmisión, puede ser de paridad.

RCSTA: Este registro se utiliza para el control y estado de la recepción.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Bit 7, SPEN: Habilita el puerto serie

1= habilita el puerto serie

0= deshabilita el Puerto serie

Bit 6, RX9: Habilita la recepción del bit 9

1= selecciona recepción con 9-bit

0= selecciona recepción con 8-bit

Bit 5, SREN: Habilita la recepción sencilla

Modo asíncrono: no influye

Modo síncrono-maestro:

1= habilita la recepción sencilla

0= deshabilita la recepción sencilla

Este bit se desactiva después de la recepción

Modo síncrono-esclavo: no influye

Bit 4, CREN: Habilita la recepción continua

Modo asíncrono:

1= habilita la recepción continua

0= deshabilita la recepción continua

Modo síncrono:

1= habilita la recepción continua

0= deshabilita la recepción continua

Bit 3, ADDEN: Habilita la detección de la dirección

Modo asíncrono con 9-bit (RX9=1)

1= habilita la detección de la dirección, solo recibe el dato y produce interrupción de recepción

0= deshabilita la detección de la dirección, se reciben todos los bytes y un noveno bit se puede usar para paridad.

Bit 2, FERR: error de encuadre.

1= error de encuadre (puede actualizarse usando RCREG y recibiendo el próximo byte válido)

0= no hay error de encuadre.

Bit 1, OERR: error de sobre escritura.

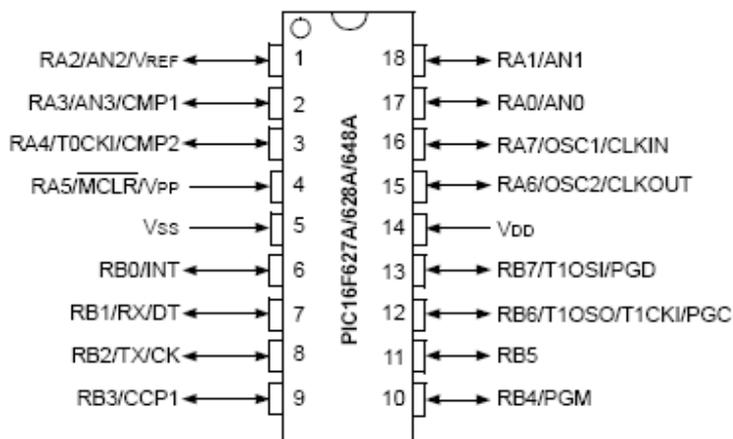
1= error de sobre escritura (puede ser borrado escribiendo un cero en CREN)

0= no hay error de sobre escritura

Bit 0, RX9D: noveno bit del dato recibido (la paridad debe ser calculada por el software de usuario).

### 4.32 El microcontrolador 16f628

Este microcontrolador consta de 2 puertos de entrada – salida. Además tiene la peculiaridad de tener completa compatibilidad, pin a pin con el 16f84, habiendo muchos programas para este ultimo por la cantidad de años en las cual se a ocupado, la única diferencia es que la memoria RAM se encuentra en otra dirección, para lo cual no ocupa mucho trabajo provocar ese cambio de configuración en el programa.



A continuación se darán a conocer la mayoría de características generales de este microcontrolador:

Voltaje de alimentación de 5V

Frecuencia de operación máxima de 20Mhz

Oscilador interno de 37Khz - 4Mhz

Memoria de datos de 224 Bytes

Memoria de datos EEPROM de 128 Bytes

2 temporizadores internos de 8 bites

1 temporizador interno de 16 bites

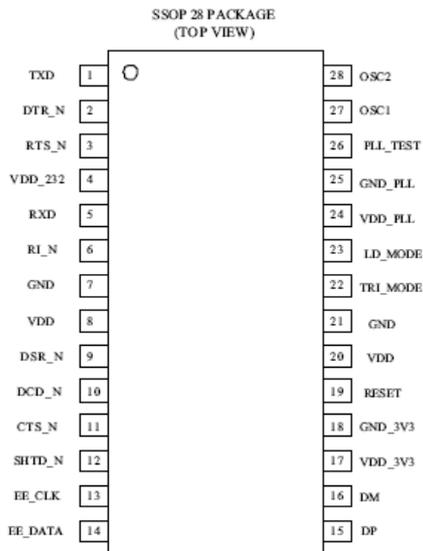
1 modulo de captura, comparación y PWM

1 modulo de comunicaciones USART

#### 4.4 INTERFAZ RS232 A USB

Ya que las tarjetas inalámbricas um96, se comunican o con niveles TTL (0 lógico a 0V y 1 lógico de 4,5 a 6 V), RS232 (0 lógico de 3 a 25V y 1 lógico de -3 a -25V) y RS485, se aprovecha la disponibilidad en el mercado de un adaptador de RS232 a USB, basado en el dispositivo PL 2030 de la empresa Prolific, este al conectarlo al puerto USB crea un puerto COM virtual. Esto facilita el trabajo de recepción de datos en el Pc, ya que la bibliografía disponible para recepción a través de puerto USB, además de ser más compleja es también mucho más escasa su información.

Este dispositivo tiene una tasa de transferencia del orden de 1Mbps, doble buffer para flujo de datos de bajada y subida y oscilador interno de 12Mhz.



## **5 CAPITULO V SOFTWARE UTILIZADOS**

### **5.1 MPLAB**

Este es un software de la empresa Arizona Microchip Technology, este software permite escribir el programa para los microcontroladores, sea en lenguaje assembler como en C, además de crear proyectos, compilar, simular el programa, para finalmente programar el dispositivo, contando con un programador para estos.

Este programa contiene todo lo necesario para la realización de cualquier proyecto, por ejemplo permite editar el archivo fuente en lenguaje ensamblador, además de ensamblarlo y simularlo en pantalla, pudiendo ejecutarlo posteriormente en modo paso a paso y ver como evolucionarían los registros internos, la memoria RAM la EEPROM y la memoria del programa, según se fueran ejecutando las instrucciones. Además este entorno es similar a estar utilizando cualquier emulador.

#### **5.11Partes del MPLAB**

Editor: incorporado para permitir escribir y editar programas u otros archivos de texto.

Project Manager: Organiza los distintos archivos relacionados con un programa en un proyecto. Permite crear un proyecto, editar y simular un programa. Además crea archivos objeto y permite bajar archivos hacia emuladores o simuladores de hardware.

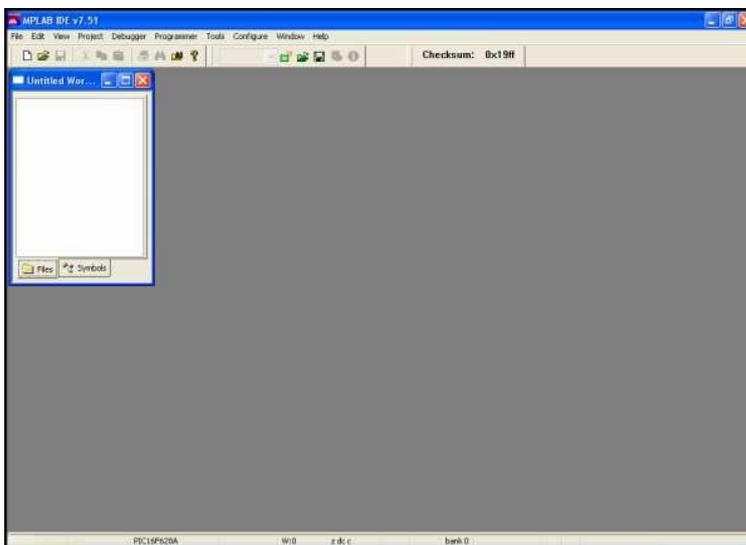
Simulador: simulador de eventos discretos que permite simular programas con ilimitados breakpoint( puntos de parada), examinar o modificar registros, observar variables, tiempos y simular estímulos externos.

Ensamblador: Genera varios tipos de archivos objetos y relacionados, para programadores microchip y universales.

Linker: Permite unir varios archivos objetos en uno solo, generados por el ensamblador o compiladores C.

Programador: El MPLAB puede trabajar con varios tipos de programadores, el usuario debe seleccionar con cual va a trabajar, haciendo clic en la opción Programmer/ Select Programmer.

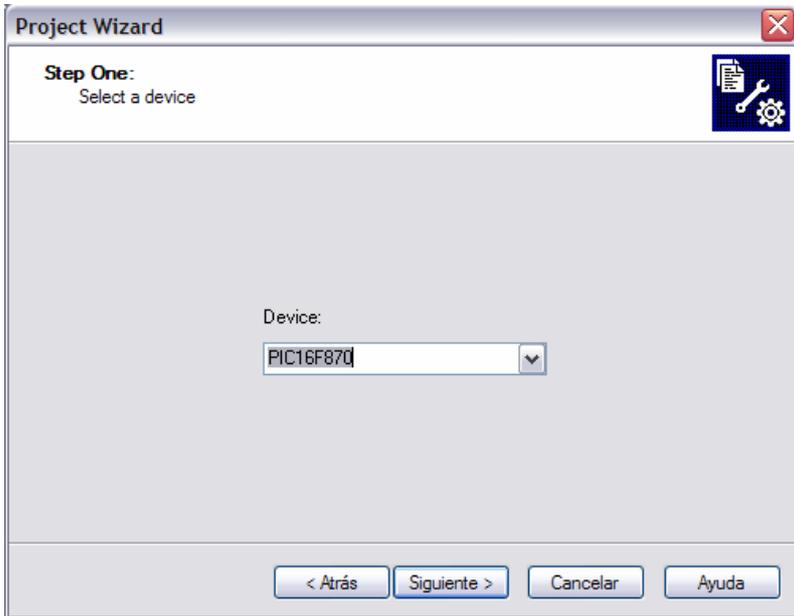
Para crear un nuevo proyecto hay que seguir los siguientes pasos, teniendo al principio la siguiente pantalla:



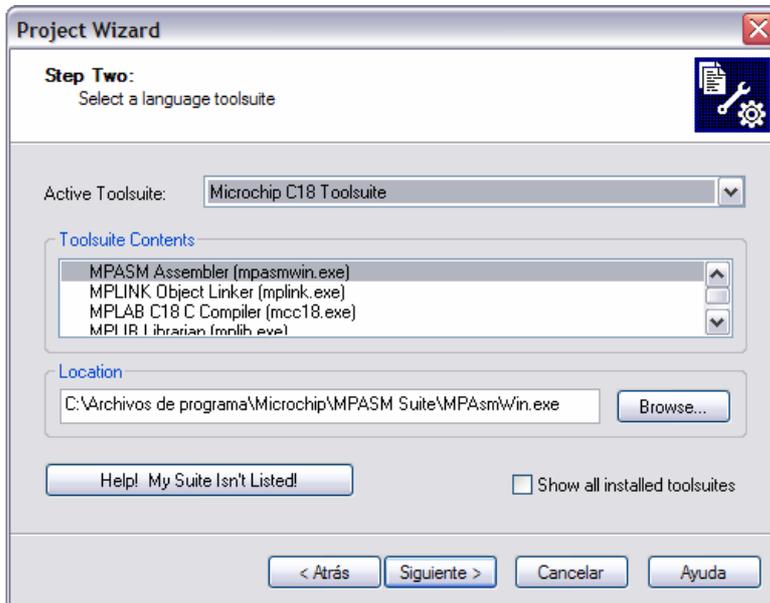
Vamos en el menú a Project/ project wizard



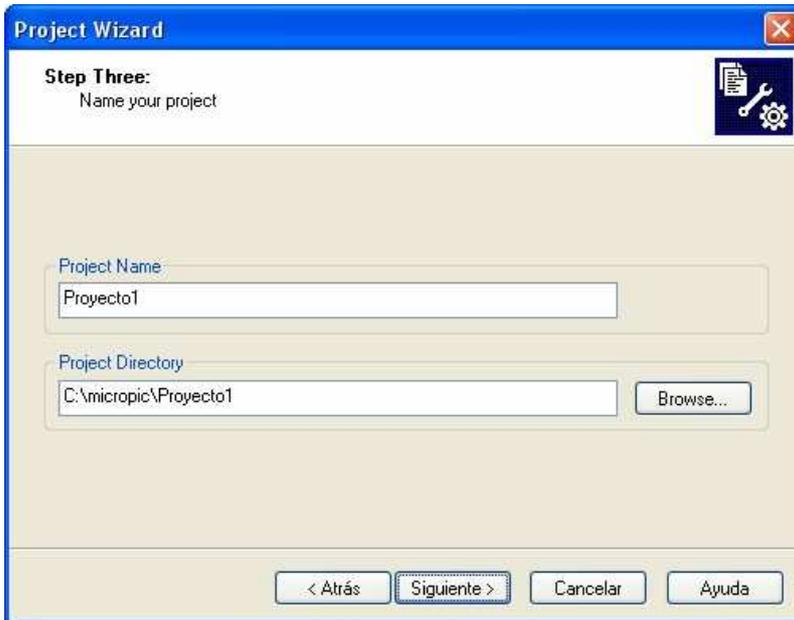
Hacemos clic en siguiente, y seleccionamos el dispositivo a utilizar, en este caso el pic 16f870:



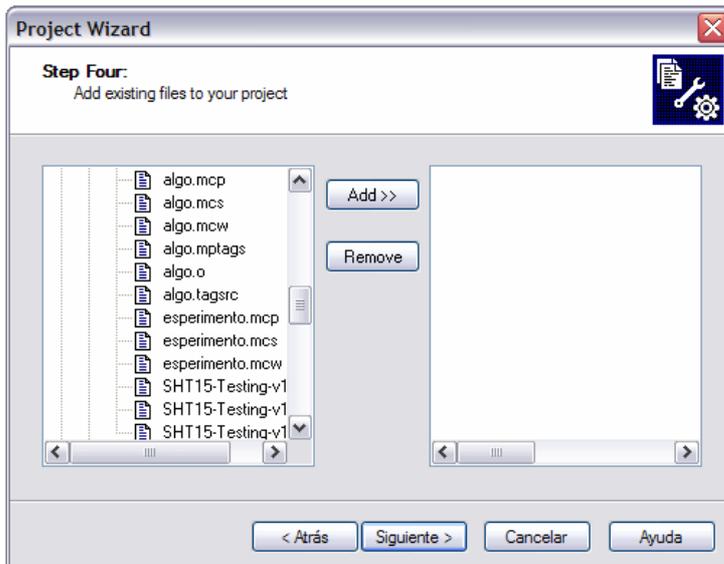
Luego se define el lenguaje a usar en la programación



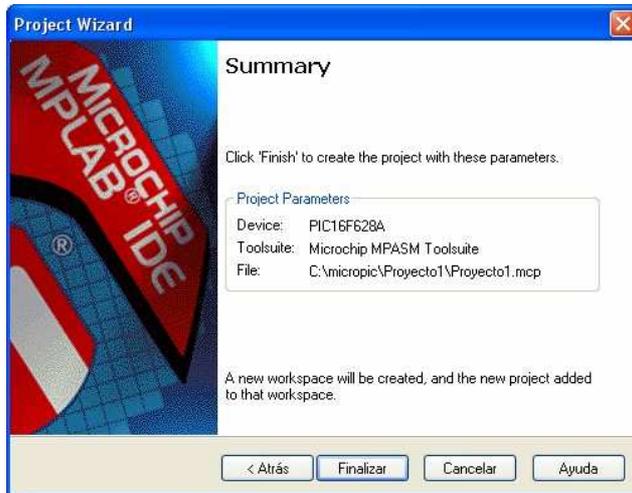
En la siguiente ventana, se escoge el nombre del proyecto y la dirección en la cual va a quedar guardada. Se recomienda que quede lo mas cerca de C:, ya que si la ruta es muy larga el compilador puede arrojar errores:



En la siguiente ventana permite agregar archivos existentes al proyecto, si no existe ninguno simplemente se coloca siguiente:



Y para finalizar aparece un resumen de lo que se hizo:



Luego si es que no se tenía un archivo previo para escribir el código se crea uno con extensión .asm o .c según el lenguaje de programación que se este utilizando.

## 5.2 VISUAL BASIC

Visual Basic es un lenguaje de programación que ha sido diseñado para facilitar el desarrollo de aplicaciones en el entorno grafico que nos presenta

Windows. Visual Basic, es una excelente herramienta para aquellos que quieran iniciarse en la programación, ya que es un lenguaje guiado por eventos, y centrado en un motor de formularios que facilita el rápido desarrollo de aplicaciones gráficas.

Visual Basic constituye un entorno de desarrollo integrado que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

## 5.21 Entorno visual Basic

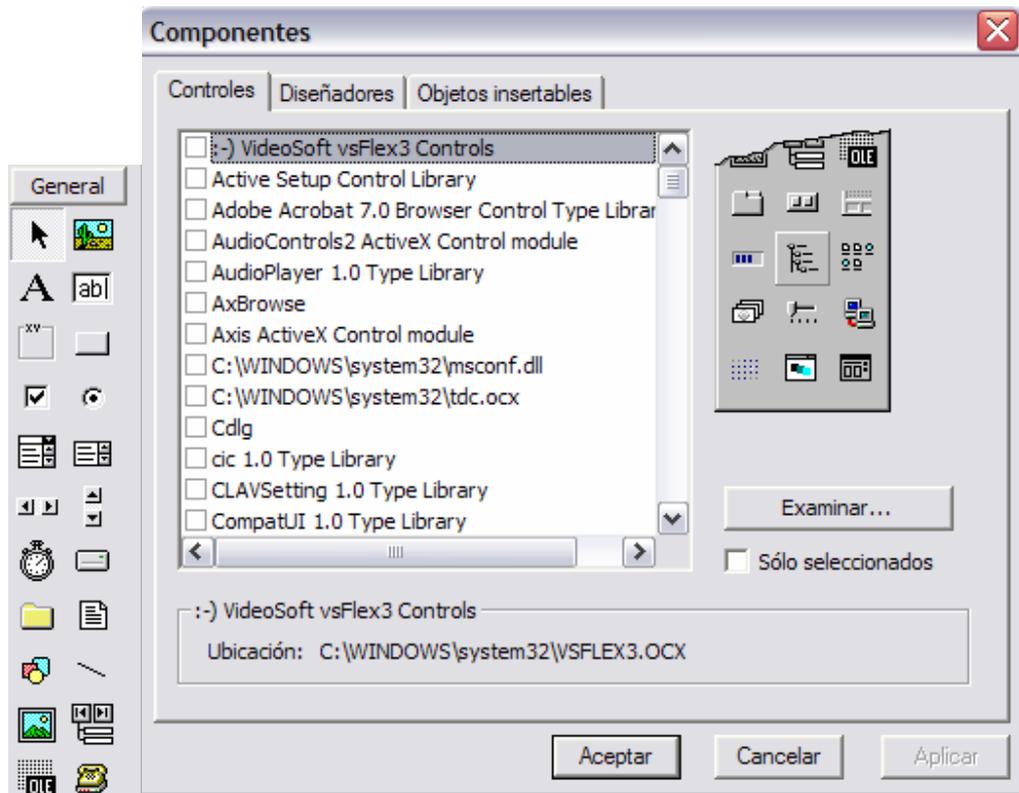
Barra de menú y barra de herramientas:

La barra de menú es similar a cualquiera de las aplicaciones de Windows, debajo de esta se encuentra la barra de herramientas, con una serie de botones que permiten acceder fácilmente a las aplicaciones más importantes de los menús. Existen 4 barras de herramientas, depuración, edición, edición de formularios y estándar.



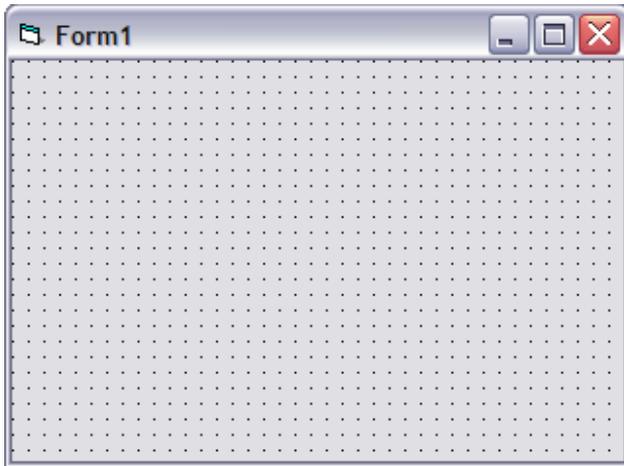
Las herramientas o toolbox.

Incluye los controles con los que se puede diseñar la pantalla de la aplicación. Estos pueden ser botones, etiquetas, cajas de texto, zonas gráficas, etc. Para pasar uno de estos al formulario solo se debe hacer clic y colocarlo en este. Además el número de estas aplicaciones se puede modificar a gusto del usuario.



## Formularios.

Los formularios son las zonas de la pantalla sobre las que se diseña el programa y sobre las que se sitúan los controles. Al ejecutar el programa, el formulario o form, se convertirá en la ventana de la aplicación, donde aparecerán los botones, el texto, los gráficos, etc.



Resumiendo, cuando se vaya a crear un programa en *Visual Basic 6.0* habrá que dar dos pasos:

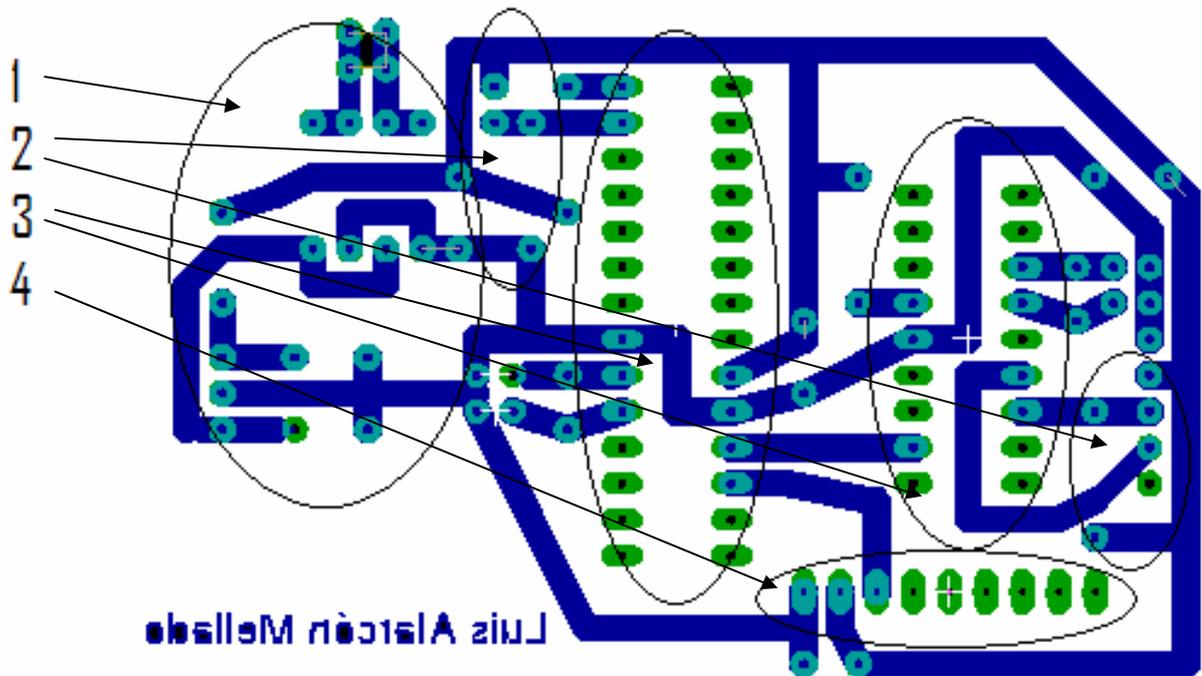
1. Diseñar y preparar la parte gráfica (formularios, botones, menús, etc.)
2. Realizar la programación que gestione la respuesta del programa ante los distintos eventos.

## 6 CAPITULO VI: FUNCIONAMIENTO Y CARACTERISTICAS DEL EQUIPO

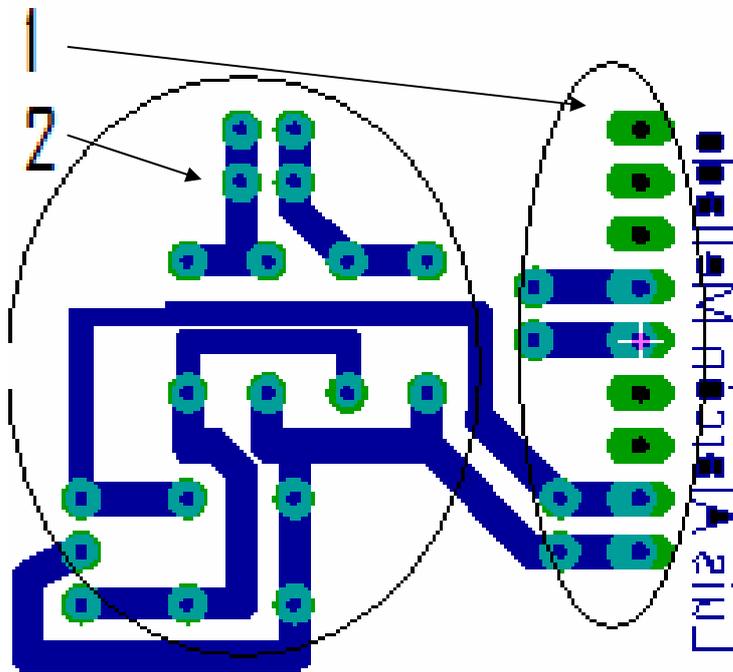
### 6.1 CONFECCIÓN DE LOS CIRCUITOS.

#### 6.1.1 Diseño de las placas

Todo el monitor consta de dos circuitos, el primero y principal, es el que contiene una etapa de alimentación (1), toda la conexión hacia los sensores (2), los microcontroladores (3) y la conexión con la tarjeta emisora (4).



En la segunda placa, se encuentra la etapa de alimentación (1), y la tarjeta receptora con la conexión al adaptador de RS232 a USB (2).



Estos esquemas fueron realizados con el programa EAGLE 4.09r2, el cual es fácil de ocupar, además de tener librerías bastante completas.

### 6.12 Traspaso del circuito a cobre y soldado.

Esto fue realizado con la técnica de impresión del circuito en papel de transparencia, para luego pasar la tinta con calor. Estando ya impregnada la tinta en la placa de cobre, se sumerge en una solución corrosiva para placas impresas. Posteriormente se realizan los agujeros con una broca de 1mm. Para el soldado se utiliza estaño, colocando primero en la placa los elementos de menor altura, como las resistencias y bases para microcontroladores, y luego los demás elementos.

## 6.2 FUNCIONAMIENTO DEL EQUIPO

Para el funcionamiento óptimo del equipo se tienen que tener algunas consideraciones, tales como:

Se tiene que conectar a una fuente continua de 9V, puede ser con transformador o con batería.

El sensor de humedad de suelo tiene que ser enterrado a una profundidad igual al de las raíces de las plantas que se quieren monitorear.

La distancia entre el monitor y la recepción de los datos no tiene que superar los 500mts.

Al conectar el adaptador de USB a RS232, en el software de recepción se tiene que colocar en que puerto serial esta conectado, a que dependiendo de la cantidad de estos asigna el numero de puerto, y eso varia de un computador a otro.



Como se puede apreciar en la figura, el programa consta de 3 botones, el primero “Conectar”, que es para iniciar la toma de datos, previamente seleccionando el puerto Com en el que esta instalado el monitor, el segundo “Desconectar” finaliza con la toma de datos, y el ultimo es el que crea el archivo Excel con los datos tomados.

El equipo entrega al computador los resultados en bruto, o sea se necesita de una formula para calcular el valor real. Esto lo hace automáticamente la planilla Excel donde se tienen las formulas previamente guardadas. Los datos en la planilla van quedando por columnas, la primera para la temperatura, la segunda para humedad ambiental y la ultima para humedad de suelo. La temperatura es convertida a grados Celsius, y las dos humedades en porcentaje de humedad relativa. Además hay una cuarta columna la que nos indica la fecha y hora en que fueron tomados los datos.

	A	B	C	D
1	Temperatura	Humedad	Humedad de suelo	Fecha y Hora
2	20,3	70,4731168	43,55932203	"04-07-2009 12:37:57"
3	20,28	70,5015068	43,55932203	"04-07-2009 12:38:05"
4	20,26	70,4447212	43,55932203	"04-07-2009 12:38:21"
5	20,26	70,2457952	43,55932203	"04-07-2009 12:38:32"
6	20,26	70,3595008	43,55932203	"04-07-2009 12:38:41"
7	20,24	70,5015068	43,55932203	"04-07-2009 12:38:50"
8	20,24	70,5866432	43,55932203	"04-07-2009 12:38:59"
9	20,21	70,6433728	43,55932203	"04-07-2009 12:39:07"
10	20,19	70,27423	46,27118644	"04-07-2009 12:39:30"
11	20,17	70,1889088	43,55932203	"04-07-2009 12:39:46"
12	20,14	70,6150108	46,27118644	"04-07-2009 12:39:57"
13	20,12	70,7284252	46,27118644	"04-07-2009 12:40:06"
14	20,12	70,6150108	46,27118644	"04-07-2009 12:40:13"
15	20,12	70,41632	46,27118644	"04-07-2009 12:40:22"
16	20,14	70,5298912	46,27118644	"04-07-2009 12:40:31"
17	20,14	70,3310828	48,98305085	"04-07-2009 12:40:42"
18	20,1	70,3879132	48,98305085	"04-07-2009 12:40:51"
19	20,12	70,5015068	57,11864407	"04-07-2009 12:41:00"
20	20,12	70,5015068	57,11864407	"04-07-2009 12:41:09"
21	20,1	70,3595008	59,83050847	"04-07-2009 12:41:23"
22	20,08	70,4447212	65,25423729	"04-07-2009 12:41:34"
23	20,08	70,84175	65,25423729	"04-07-2009 12:41:43"
24	20,08	70,5298912	65,25423729	"04-07-2009 12:42:00"
25	20,08	70,5015068	65,25423729	"04-07-2009 12:42:12"
26	20,08	70,5015068	59,83050847	"04-07-2009 12:42:21"

### 6.3 EVALUACION FINANCIERA DEL DESARROLLO DEL MONITOR.

El desarrollo e investigación del monitor de invernaderos, tiene un costo asociado, que va desde los materiales a la mano de obra. A continuación se detallaran de mejor forma los gastos asociados a estos:

Materiales		
	Modulo UM96 x2	\$ 87.356
	Sensor SHT15	\$ 36.649
	Cable RS232/USB	\$ 13.977
	PIC 16F628	\$ 2.368
	PIC 16F870	\$ 3.819
	Cristal 10Mhz x 2	\$ 1.520
	Conector batería 9V x2	\$ 2.094
	Pines de conexión	\$ 1.747
	Conector hembra RS232	\$ 664
	Regulador de voltaje x2	\$ 2.140
	batería 9V x2	\$ 2.000
	base para PIC 28 pines	\$ 873
	base para PIC 18 pines	\$ 804
	conector DC x2	\$ 1.608
	diodos 1n4002 x8	\$ 1.360
	yeso	\$ 400
	plaquetas de cobre	\$ 1.000
	resistencias 0,25W x20	\$ 500
	Condensadores x15	\$ 600
	Placas de cobre x2	\$ 2.000
Ingeniería		
	mano de obra	\$ 300.000
	investigación y desarrollo	\$ 500.000
Investigación		
	viaje a invernaderos	\$ 2.400
	viaje a UCT*	\$ 6.400
	fotocopias	\$ 5.400
<b>TOTAL</b>		<b>\$ 977.679</b>

*Tabla de costos (\*Universidad Católica de Temuco)*

En el Caso de producirlos en forma comercial, tendría que considerarse el costo de los materiales que asciende a los 163.479 pesos. Lo cual podría bajar al comprar los elementos al por mayor. Este costo de materiales tendría que ser sumado a costos de mano de obra en

producirlo, que por equipo debería ser unos 30.000 pesos, más las ganancias alcanzaría un total de 250000 pesos por monitor terminado.

## CONCLUSIONES

Para el diseño de un monitor o cualquier otro dispositivo, es esencial realizar un estudio previo, para determinar las necesidades que tendrá el dispositivo, las variables a considerar, rangos, metodología, etc. para precisar los objetivos que debe cumplir el artefacto y diseñarlo de la manera más adecuada para cumplir estos objetivos, considerando las exigencias que tendrá el dispositivo durante su funcionamiento.

La optimización que deberá efectuarse en el diseño e implementación del instrumento, es un proceso que es realizado durante el trabajo con el mismo, debido a que es necesario un ensayo del funcionamiento “in situ” del dispositivo. La optimización de un dispositivo de este tipo, requiere del análisis de variables del medio externo, que solo pueden ser determinadas cuando el monitor ya esté operando en su ubicación definitiva.

El monitoreo de registros de variables tan fundamentales como son humedad y temperatura en invernaderos, dan paso a que se puedan implementar en base a estos, elementos de control automático en el proceso. Este tipo de registros, permite la optimización de los recursos que posee el agricultor para lograr las variables óptimas para el desarrollo de las plantas, al menor costo posible.

Las tarjetas UM96 son de gran utilidad en dispositivos de transmisión inalámbrica, ya que su flexibilidad nos permite un gran ámbito de uso de estas, además de tener en la misma familia, una gran gama de distancias a alcanzar, lo cual puede bajar costos para quien no necesite tanta distancia para alcanzar sus objetivos, además de poseer varios canales que permiten el uso de varios módulos funcionando de forma de paralela.

Los sensores de humedad de suelo, están sometidos a un deterioro que se hace notable después del mes de estar en utilización aproximadamente, pero estos al ser de fácil construcción y bajo costo no debería significar mayores complicaciones en su reemplazo.

La utilización del monitor, permite un estudio continuo en las variables del invernadero, lo cual da la posibilidad de un análisis profundo al como se esta llevando a cabo el proceso, realizar curvas de comportamiento diarios, mensuales y anuales. Al tener este tipo de información a disposición siempre va en post del desarrollo y mejora de un proceso como lo es el cultivo de plantas, hortalizas y flores dentro de invernaderos.

## BIBLIOGRAFÍA

Infante A, K San Martin. 2004. Manual de agroecología. Centro de educación y tecnología, Chile

Escobar J. 1964. Cosechas bajo cubierto. Editorial Acribia. Zaragoza España

Toovey F. 1967. Invernaderos Comerciales: construcción y calefacción. Editorial Acribia. Zaragoza España

Golzov M, C Maximov, V Iaroshevski. 1975. Agro meteorología practica. Dirección general de aeronáutica civil, dirección meteorológica de chile. Santiago chile.

Microchip. 2003. Datasheet Pic16F62X. (Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/40300C.pdf>)

Microchip. 2003. Datasheet Pic16F870/871. (Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/30569b.pdf>)

Sensirion. 2008. DatasheetSHT1x. (Disponible en: [http://www.sensirion.com/en/pdf/product\\_information/Datasheet-humidity-sensor-SHT1x.pdf](http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT1x.pdf))

## ANEXOS

### CÓDIGO DE PROGRAMACIÓN VISUAL BASIC

Option Explicit

Dim Cadena As String

Private Sub Conectar\_Click()

'comprueba que el puerto este cerrado para poder abrirlo

If MSComm1.PortOpen = False Then

'determina el puerto que hemos seleccionado

If Combo1.ListIndex = 0 Then

MSComm1.CommPort = 1

Else

If Combo1.ListIndex = 1 Then

MSComm1.CommPort = 2

Else

MSComm1.CommPort = 5

End If

End If

'determina: 9600-Velocidad en Baudios, N-No utiliza ninguna paridad,

'8-Cantidad de bits de envio y recepcion por paquete,

'1-Determina los bits de parada

MSComm1.Settings = "9600,N,8,1"

'lee todo el buffer de entrada para que quede vacio

MSComm1.InputLen = 0

'Abre el puerto seleccionado

MSComm1.PortOpen = True

Me.Caption = "Conectado por el puerto " & MSComm1.CommPort

End If

End Sub

Private Sub Desconectar\_Click()

If MSComm1.PortOpen Then

```

'cierra el puerto
MSComm1.PortOpen = False
Me.Caption = "Desconectado"
Close #1
End If
End Sub
Private Sub Enviar_Click()
Shell ("asdf.bat")
Me.Caption = "Excel creado"
End Sub
Private Sub Form_Load()
Combo1.AddItem "Com1"
Combo1.AddItem "Com2"
Combo1.AddItem "Com5"
Combo1.ListIndex = 0
End Sub
Private Sub Form_Unload(Cancel As Integer)
Desconectar_Click
End Sub
'El evento OnComm se genera siempre que cambia el valor de la propiedad
'CommEvent e indica que se ha producido un evento o un error en la comunicación.
Private Sub MSComm1_OnComm()
Dim i As Integer
Dim Valor As String
Dim largo As Integer
'recoge el valor de entrada
Valor = MSComm1.Input
'MsgBox (Valor)
'busca la posicion del caracter de salto de linea
Open "datos.txt" For Append As #1
Write #1, Valor

```

```
If (InStr(Valor, "X") <> 0) Then
    Write #1, Date$ + " " + Time$
End If
Close #1
End Sub
Private Sub Texto_Change()
End Sub
Private Sub TextoEnviar_Change()
End Sub
Private Sub OLE1_Updated(Code As Integer)
End Sub
```

### **CÓDIGO DE PROGRAMACIÓN PIC 16F870**

```
LIST P=16F870
INCLUDE <P16F870.INC>
cblock 0x20

cont
msnib
lsnib
PDe10
PDe11
PDe12
PDe13
dato
leer
contar
repcion
    endc
ORG 0
```

```

GOTO INICIO
ORG 4
GOTO INTER
asc ADDWF PCL,1      ;Calcula el código a retornar
                                ;Saltando W instrucciones adelante
DT "0123456789ABCDEF"
INICIO:
BSF STATUS, RP0
movlw b'10111111';MOVLW b'10111111'
MOVWF TRISC
call initransm
call inianalog
bsf  recepcion,0
MOVLW b'11000000'
MOVWF INTCON ;//Habilitamos la interrupción
;*****
atras  btfsc recepcion,0
        goto atras
        MOVLW    0x01      ;Selecciona el canal AN0, reloj de conversión Fosc/2
        MOVWF    ADCON0    ;y enciende el convertidor
ciclo  CALL  pausa      ;espera 30 µseg a que pase el tiempo de adquisición
        BSF    ADCON0,2    ;inicia conversión
espera BTFSC    ADCON0,2
        GOTO    espera     ;Espera a que termine la conversión
        MOVF    ADRESH,W   ;Carga en W el Byte alto del resultado
        CALL  Envbyte     ;envía el byte por el puerto serie
        BSF    STATUS,RP0 ;banco 1
        MOVF    ADRESL,W   ;Carga en W el Byte bajo del resultado
        CALL  Envbyte     ;envía el byte por el puerto serie
        MOVLW    0x0D      ;carga código de retorno de línea <CR>
        CALL  envia      ;lo envía

```

```

    MOVLW    0x0A        ;carga código de avance de línea <LF>
    CALL envia          ;lo envía
    movlw    'X'
    BCF     STATUS,RP0
    BCF     PIR1, TXIF
    MOVWF   TXREG ;//Cargamos el dato recibido en RCREG en TXREG para
volver a transmitirlo
    BSF     STATUS,RP0
TX_W:BTFFS    TXSTA, TRMT;//Miramos si ha finalizado la transmisión
    GOTO    TX_W
    BCF     STATUS,RP0
;*****
    CALL DEMORA1
    CALL DEMORA1
    CALL DEMORA1
    CALL DEMORA1
    CALL DEMORA1
    bsf     recepcion,0
    Bsf     PIE1, RCIE
    bsf     INTCON,PEIE
    bsf     INTCON,GIE
    GOTO    atras ;repite
Envbyte:
    MOVWF   msnib ;pone byte en msnib
    MOVWF   ls nib ;y una copia en ls nib
    SWAPF   msnib,1 ;intercambia nibbles en ls nib
    MOVLW   0x0F ;máscara para limpiar el nibble alto
    ANDWF   msnib,1 ;limpia parte alta de msnib
    ANDWF   ls nib,1 ;limpia parte alta de ls nib
    MOVF    msnib,W ;carga msnib en W
    CALL    asc ;obtiene código ASCII equivalente

```

```

CALL          envia ;lo envía por el puerto serie
MOVF         lsnib,W ;carga lsnib en W
CALL        asc  ;obtiene código ASCII equivalente
CALL        envia ;lo envía por el puerto serie
CALL        DEMORA
RETURN

;*****
; inicializacion
;*****
initransm
    MOVLW b'00100100'
    MOVWF TXSTA ;// csrc(maestro), txen(transmission on),sync(síncrono) a 1
    MOVLW .64
    MOVWF     SPBRG           ;//Comunicación a 9600
baudios&#61664;BRG=Fosc/(4*(X+1))&#61664;X=103
    BSF PIE1, RCIE
    BCF STATUS, RP0
    MOVLW b'10010000'
    MOVWF RCSTA ;//Recepción continua y activamos el módulo SCI(spen=1)
    return
inianalog
    BSF STATUS,RP0 ;Banco 1
    CLRF ADCON1 ;configura 8 canales analógicos, VREF+=VDD y VREF-=VSS
    BSF ADCON1,ADFM ;Elije resultado con justificación a la derecha
    BSF TRISA,0 ;configura como entrada el canal digital RA0
    BCF STATUS,RP0 ;Banco 0
    return
;*****
;envio serial
;*****
envia

```

```

                BSF  STATUS,RP0 ;banco 1
esp   BTFSS     TXSTA,TRMT ;checa si el buffer de transmisión
                GOTO     esp      ;si está ocupado espera
                BCF  STATUS,RP0 ;regresa al banco 0
                MOVWF    TXREG     ;envía dato guardado en W
                RETURN

,*****
; INTERRUPCION
,*****
INTER:          BCF  PIR1, RCIF
                MOVF   RCREG, W
                movwf dato
                movlw   'T'
                xorwf  dato,W
                btfss  STATUS,Z
                goto   gg
                movlw  .13
                movwf  contar
                bsf    leer,0
gg         btfss  leer,0
                retfie
                DECFSZ  contar
                goto   hh
                clrf   recepcion
                bcf    leer,0
                BCF  PIE1, RCIE
                bcf    INTCON,PEIE
                retfie
hh         movf   RCREG,W
                BCF  PIR1, TXIF

```

```

MOVWF TXREG ;//Cargamos el dato recibido en RCREG en
TXREG para volver a transmitirlo

```

```

BSF STATUS,RP0
TX_WAIT: BTFSS TXSTA,TRMT;//Miramos si ha finalizado la transmisión
GOTO TX_WAIT
BCF STATUS,RP0
RETFIE

```

```

;*****
;

```

```

;rutinas fijas de tiempo

```

```

;*****
;

```

```

DEMORA movlw .14 ; 1 set numero de repeticion (C)

```

```

movwf PDel0 ; 1 |

```

```

PLoop0 movlw .72 ; 1 set numero de repeticion (B)

```

```

movwf PDel1 ; 1 |

```

```

PLoop1 movlw .247 ; 1 set numero de repeticion (A)

```

```

movwf PDel2 ; 1 |

```

```

PLoop2 clrwdt ; 1 clear watchdog

```

```

decfsz PDel2, 1 ; 1 + (1) es el tiempo 0 ? (A)

```

```

goto PLoop2 ; 2 no, loop

```

```

decfsz PDel1, 1 ; 1 + (1) es el tiempo 0 ? (B)

```

```

goto PLoop1 ; 2 no, loop

```

```

decfsz PDel0, 1 ; 1 + (1) es el tiempo 0 ? (C)

```

```

goto PLoop0 ; 2 no, loop

```

```

PDelL1 goto PDelL2 ; 2 ciclos delay

```

```

PDelL2 clrwdt ; 1 ciclo delay

```

```

return ; 2+2 Fin.

```

```

;-----

```

```

; Generado con PDEL ver SP r 1.0 el 18/01/2009 Hs 22:26:28

```

```

; Descripcion: Delay 2500000 ciclos

```

```

;-----

```

```

DEMORA1 movlw .22 ; 1 set numero de repeticion (C)

```

```

    movwf   PDel0   ; 1 |
PLoop01 movlw   .134   ; 1 set numero de repeticion (B)
    movwf   PDel1   ; 1 |
PLoop11 movlw   .211   ; 1 set numero de repeticion (A)
    movwf   PDel2   ; 1 |
PLoop21 clrwdt      ; 1 clear watchdog
    decfsz  PDel2, 1 ; 1 + (1) es el tiempo 0 ? (A)
    goto    PLoop21 ; 2 no, loop
    decfsz  PDel1, 1 ; 1 + (1) es el tiempo 0 ? (B)
    goto    PLoop11 ; 2 no, loop
    decfsz  PDel0, 1 ; 1 + (1) es el tiempo 0 ? (C)
    goto    PLoop01 ; 2 no, loop
PDelL11 goto PDelL21 ; 2 ciclos delay
PDelL21 clrwdt      ; 1 ciclo delay
    return   ; 2+2 Fin.

pausa MOVLW    0x23 ;Carga dato para 30 µseg.
        MOVWF   cont ;inicializa contador con el dato
rep    DECFSZ   cont,1 ;Decrementa contador y escapa si cero
        GOTO    rep ;si no es cero, repite
esc    RETURN   ;regresa de esta subrutina
        END

```

### **CÓDIGO DE PROGRAMACIÓN PIC 16F628**

```

#define Clock_8MHz
#define Baud_9600
#include "c:\CC5X\16F628a.h" // device dependent interrupt definitions
#pragma origin 4
// #define STATUS_LED PORTB.3
#define WRITE_sda() TRISB = TRISB & 0b.1011.1111 //SDA must be output when writing

```

```

#define READ_sda() TRISB = TRISB | 0b.0100.0000 //SDA must be input when reading - don't
forget the resistor on SDA!!
#define SCK PORTB.7
#define SDA PORTB.6
#define espera PORTA.0
#define CHECK_TEMP 0b.0000.0011
#define CHECK_HUMD 0b.0000.0101
#define CHECK_STAT 0b.0000.0111
#define WRITE_STAT 0b.0000.0110
void boot_up(void);
void sht15_start(void);
void sht15_read(void);
void enviar(void);
void sht15_send_byte(uns8 sht15_command);
uns16 sht15_read_byte16(void);
void delay_ms(uns16);
void printf(const char *nate, int16 my_byte);
uns16 responset;
uns16 responseh;
void main(void)
{
    uns8 choice;
    boot_up();
    delay_ms(100);
    //printf("\nSHT15 Testing", 0);
    sht15_read();
    enviar();
    //printf("\nDone", 0);
    while(1);
} //End Main
void boot_up(void)

```

```

{
//OSCCON = 0b.0111.0000; //Setup internal oscillator for 8MHz
//while(OSCCON.2 == 0); //Wait for frequency to stabilize
//Setup Ports
//ANSEL = 0b.0000.0000; //Turn off A/D
PORTA = 0b.0000.0000;
TRISA = 0b.0000.0011;
PORTB = 0b.0000.0000;
TRISB = 0b.0000.0100; //0 = Output, 1 = Input RX on RB2
//Setup the hardware UART module
//=====
SPBRG = 64; //10MHz for 9600 inital communication baud rat brgh1
//SPBRG = 129; //20MHz for 9600 inital communication baud rate
TXSTA = 0b.0010.0100; //8-bit asych mode, high speed uart enabled
RCSTA = 0b.1001.0000; //Serial port enable, 8-bit asych continous receive mode
//=====
}
//Init the sensor and read out the humidity and temperature data
void sht15_read(void)
{
//Issue command start
sht15_start();
sht15_send_byte(CHECK_TEMP);
responset = sht15_read_byte16(); //Listen for response from SHT15
sht15_start();
//Now send command code
sht15_send_byte(CHECK_HUMD);
responseh = sht15_read_byte16();
}
void enviar(void)
{

```

```

    printf("T%d ", responset);
    delay_ms(100);
    printf("H%d ", responseh);
}
void sht15_send_byte(uns8 sht15_command)
{
    uns8 i;
    WRITE_sda();
    for(i = 0 ; i < 8 ; i++)
    {
        sht15_command = rl(sht15_command);
        SCK = 0;
        SDA = Carry;
        SCK = 1;
    }
    //Wait for SHT15 to acknowledge.
    SCK = 0;
    READ_sda();
    while (SDA == 1); //Wait for SHT to pull line low
    SCK = 1;
    SCK = 0; //Falling edge of 9th clock
    while (SDA == 0); //Wait for SHT to release line
    //Wait for SHT15 to pull SDA low to signal measurement completion.
    //This can take up to 210ms for 14 bit measurements
    i = 0;
    while (SDA == 1) //Wait for SHT to pull line low
    {
        i++;
        if (i == 255) break;
        delay_ms(10);
    }
}

```

```

        clrwdt();
    }
    //Debug info
    // i *= 10; //Convert to ms
    // printf("\nResponse time = %dms", i); //Debug to see how long response took
}
//Specific SHT start command
void sht15_start(void)
{
    WRITE_sda();
    SDA = 1;
    SCK = 1;
    SDA = 0;
    SCK = 0;
    SCK = 1;
    SDA = 1;
    SCK = 0;
}
//Read 16 bits from the SHT sensor
uns16 sht15_read_byte16(void)
{
    uns8 j;
    uns16 in_byte;
    SCK = 0;
    READ_sda();
    /***** CHANGE *****/
    for(j = 0 ; j < 17 ; j++)
    {
        if(j!=8)
        {
            nop();nop();nop();nop();

```

```

    SCK = 1;
    in_byte = rl(in_byte);
    in_byte.0 = SDA;
    SCK = 0;
}
else
{
    WRITE_sda();
    SDA = 0;
    nop();
    SCK = 1; //clk #9 for acknowledge
    nop();nop();nop();
    SCK = 0;
    READ_sda();
}
}
/***** END CHANGE *****/
return(in_byte);
}
//Low-level system routines
//General short delay
void delay_ms(uns16 x)
{
    //Clocks out at 1006us per 1ms
    uns8 y, z;
    for ( ; x > 0 ; x--)
        for ( y = 0 ; y < 4 ; y++)
            for ( z = 0 ; z < 69 ; z++);
}
//Sends nate to the Transmit Register
void putc(uns8 nate)

```

```

{
    while(TXIF == 0);
    TXREG = nate;
}
uns8 getc(void)
{
    while(RCIF == 0);
    return (RCREG);
}
uns8 scanc(void)
{
    uns16 counter = 0;
    //CREN = 0;
    //CREN = 1;
    RCIF = 0;
    while(RCIF == 0)
    {
        counter++;
        if(counter == 1000) return 0;
    }
    return (RCREG);
}
//Returns ASCII Decimal and Hex values
uns8 bin2Hex(char x)
{
    skip(x);
    #pragma return[16] = "0123456789ABCDEF"
}
//Prints a string including variables
void printf(const char *nate, int16 my_byte)
{

```

```

uns8 i, k, m, temp;
uns8 high_byte = 0, low_byte = 0;
uns8 y, z;

uns8 decimal_output[5];
for(i = 0 ; ; i++)
{
    //delay_ms(3);
    k = nate[i];
    if (k == '\0')
        break;
    else if (k == '%') //Print var
    {
        i++;
        k = nate[i];
        if (k == '\0')
            break;
        else if (k == '\\') //Print special characters
        {
            i++;
            k = nate[i];
            putc(k);
        } //End Special Characters
    }
    else if (k == 'b') //Print Binary
    {
        for( m = 0 ; m < 8 ; m++ )
        {
            if (my_byte.7 == 1) putc('1');
            if (my_byte.7 == 0) putc('0');
            if (m == 3) putc(' ');
            my_byte = my_byte << 1;
        }
    }
}

```

```

    }
} //End Binary
else if (k == 'd') //Print Decimal
{
    //Print negative sign and take 2's compliment
    if(my_byte < 0)
    {
        putc('-');
        my_byte *= -1;
    }
    if (my_byte == 0)
        putc('0');
    else
    {
        //Divide number by a series of 10s
        for(m = 4 ; my_byte > 0 ; m--)
        {
            temp = my_byte % (uns16)10;
            decimal_output[m] = temp;
            my_byte = my_byte / (uns16)10;
        }

        for(m++ ; m < 5 ; m++)
            putc(bin2Hex(decimal_output[m]));
    }
} //End Decimal
else if (k == 'h') //Print Hex
{
    //New trick 3-15-04
    putc('0');
    putc('x');

```

```
    if(my_byte > 0x00FF)
    {
        putc(bin2Hex(my_byte.high8 >> 4));
        putc(bin2Hex(my_byte.high8 & 0b.0000.1111));
    }
    putc(bin2Hex(my_byte.low8 >> 4));
    putc(bin2Hex(my_byte.low8 & 0b.0000.1111));
} //End Hex
else if (k == 'f') //Print Float
{
    putc('!');
} //End Float
else if (k == 'u') //Print Direct Character
{
    //All ascii characters below 20 are special and screwy characters
    //if(my_byte > 20)
    putc(my_byte);
} //End Direct

} //End Special Chars

else
    putc(k);
}
}
```