



**COLEGIO DE POSTGRADUADOS**  
INSTITUCION DE ENSEÑAZA E INVESTIGACION EN CIENCIAS AGRICOLAS

CAMPUS MONTECILLO  
POSTGRADO EN HIDROCIENCIAS

**DISEÑO DE UN SISTEMA DE TELEMETRÍA PARA  
VARIABLES CLIMATOLÓGICAS Y HUMEDAD DEL  
SUSTRATO EN INVERNADERO.**

**CARLOS ALBERTO GARDEA FLORES**

**TESIS**

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS**

MONTECILLO, TEXCOCO, EDO. DE MEXICO

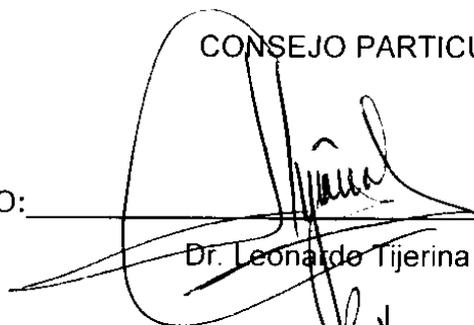
2010

La presente tesis, titulada **Diseño de un sistema de telemetría para variables climatológicas y humedad del sustrato en invernadero**, realizada por el alumno **Carlos Alberto Gardea Flores** bajo la dirección del Consejo Particular indicado, ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

**MAESTRO EN CIENCIAS  
HIDROCIENCIAS**

CONSEJO PARTICULAR:

CONSEJERO:



---

Dr. Leonardo Tijerina Chávez

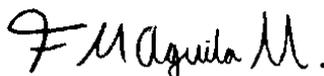
ASESOR:



---

Dr. Federico Hahn Schlam

ASESOR:



---

Dr. Francisco Miguel Águila Marín

ASESOR:



---

M.C. Rafael Acosta Hernández

Montecillo, Texcoco, Estado de México, Junio 2010

## RESUMEN

### DISEÑO DE UN SISTEMA DE TELEMETRÍA PARA VARIABLES CLIMATOLÓGICAS Y HUMEDAD DEL SUSTRATO EN INVERNADERO.

Carlos Alberto Gardea Flores

Colegio de Postgraduados, 2010

Se diseñó un sistema electrónico de telemetría, basado en microcontroladores PIC como centro de procesamiento y transceptores LINX de comunicación inalámbrica, para el monitoreo de humedad relativa, temperatura y humedad del sustrato en invernadero. El sistema cuenta con un módulo central que coordina las mediciones y almacena los datos de cinco unidades remotas que contienen los sensores. Como parte del software se implementó un protocolo de comunicación bidireccional que permite el control y flujo de información entre los dispositivos. La verificación de los sensores se realizó comparando las mediciones de éstos con las tomadas por la estación agrometeorológica del Colegio de Postgraduados Campus Montecillo. El funcionamiento del equipo se validó en el invernadero experimental del programa de Hidrociencias el cual tiene una superficie de 180m<sup>2</sup> con un cultivo de fresa (*Fragaria X ananassa* cv. aromas). Los resultados mostraron que es posible implementar sistemas de telemetría que proporcionan información confiable en tiempo real del estado ambiental de un invernadero.

**Palabras clave:** telemetría, comunicación inalámbrica, electrónica, microcontrolador, temperatura, humedad, invernadero.

## ABSTRACT

### DESIGN OF A TELEMETRY SYSTEM FOR CLIMATE VARIABLES AND SUBSTRATE HUMIDITY IN GREENHOUSE.

Carlos Alberto Gardea Flores  
Colegio de Postgraduados, 2010

An electronic telemetry system was designed, based on PIC microcontrollers as processing centers and LINX wireless transceivers for monitoring relative humidity, temperature and substrate humidity in greenhouse. The system comprises a central module to coordinate measurements and store data from five remote units containing sensors. As part of the software a bidirectional communication protocol was implemented. It allows control and information flux between devices. The verification of sensors was performed by comparing these measurements with those taken by the agrometeorological station of Colegio de Postgraduados Campus Montecillo. System validation was carried on in the experimental greenhouse for Water Science program which covers an area of 180m<sup>2</sup> with a strawberry crop (*Fragaria X ananassa* cv. Aromas). Results show that it is possible to implement telemetry systems to provide reliable real-time information about environmental conditions of a greenhouse.

**Keywords:** telemetry, wireless communication, electronics, microcontroller, temperature, humidity, greenhouse.

## **Dedicatoria**

Dedico esta tesis a mis padres Margarita y Jorge, por apoyarme en todas mis decisiones y estar siempre presentes sin importar donde me encuentre. Por formar a la persona que soy e inculcarme los valores en los cuales baso mi vida.

A mis hermanos Jorge e Iván por su apoyo y cariño. Por ser, ambos, un ejemplo para mí.

A mis primos Omar (†) y Ulises por su cariño. Por preguntarme siempre cuando iba a regresar.

A mi Tío Alberto por estar cerca y brindarme su ayuda siempre.

A Kiko, Dani, Chalo, Pedro, Irlanda, Yuri, Brisa por su amistad.

A Fabián y José por mostrarme que se puede hacer amigos en cualquier lugar.

A Luis, Pico, Esther, Sandra, Andrea, José Luis (†), Ernesto, Rosalba, Oziel, Martha, Eva, Elva, Julio, Carlos, Luz María, Pato, Cinthia, Nydia, Lucia, Miriam, Liz, Valeria, Selene, Brenda, Ángel, Uli, Omar, Mónica.... Y a todas las personas que estuvieron y están conmigo.

## **Agradecimientos**

Agradezco a **todos los mexicanos** que a través del Consejo Nacional de Ciencia y Tecnología (**CONACyT**) me concedieron el apoyo económico para realizar mis estudios de maestría.

Al **Colegio de Postgraduados** por brindarme un espacio para mi desarrollo académico.

Al **Dr. Leonardo Tijerina Chávez** por el apoyo y paciencia en la realización de este proyecto y por su dirección sin la cual esta investigación no habría finalizado.

Al **Dr. Federico Hahn Schlam** por su compromiso con este trabajo y por mostrarme como poner los pies en la tierra para resolver los problemas.

Al **Dr. Francisco Miguel Águila Marín** por su disposición y el entusiasmo que lleva más allá de las aulas.

Al **M.C. Rafael Acosta Hernández** por su interés y empeño en la revisión de mi trabajo.

Al **Dr. Abel Quevedo Nolasco** por su apoyo incondicional.

# Contenido

ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE CUADROS.....	XI
<b>1. INTRODUCCIÓN. ....</b>	<b>1</b>
<b>2. OBJETIVOS E HIPÓTESIS. ....</b>	<b>3</b>
2.1. OBJETIVO GENERAL.....	3
2.1.1.OBJETIVOS PARTICULARES. ....	3
2.2. HIPÓTESIS.....	3
<b>3. REVISIÓN DE LITERATURA.....</b>	<b>4</b>
3.1. VARIABLES CLIMÁTICAS EN INVERNADERO .....	4
3.1.1.TEMPERATURA .....	4
3.1.2.HUMEDAD RELATIVA .....	5
3.1.3.INTERACCIONES CLIMÁTICAS .....	6
3.1.4.TECNIFICACIÓN .....	12
3.2. MONITOREO .....	15
3.2.1.SISTEMA DE CONTROL .....	15
3.2.2.SISTEMA DE MEDICIÓN .....	15
3.2.3.SENSORES INTELIGENTES.....	19
3.3. COMUNICACIÓN ELECTRÓNICA .....	19
3.3.1.CODIFICACIÓN. ....	21
3.3.2.INTERFACES DE COMUNICACIÓN. ....	22
3.3.3.PROTOCOLOS DE COMUNICACIÓN.....	23
3.3.4.RUIDO. ....	24
3.3.5.TELEMETRÍA .....	25
3.3.6.ALCANCE .....	27
<b>4. MATERIALES Y MÉTODOS .....</b>	<b>29</b>
4.1. ESTRUCTURA DEL SISTEMA .....	29
4.2. UBICACIÓN DEL SITIO DE PRUEBA.....	29
4.3. DISPOSITIVO COORDINADOR.....	32
4.3.1.MICROCONTROLADOR.....	33
4.3.2.MEMORIA. ....	34
4.3.3.RELOJ. ....	34
4.3.4.PERIFÉRICOS.....	34
4.3.5.SOFTWARE COORDINADOR.....	34
4.4. DISPOSITIVO ESCLAVO .....	41
4.4.1.MICROCONTOLADOR .....	42
4.4.2.SENSOR HR/T .....	42
4.4.3.TENSIÓMETRO .....	45
4.4.4.VENTILADOR .....	46
4.4.5.SOFTWARE ESCLAVO .....	47
4.4.6.CALIBRACIÓN.....	49

4.5. MÓDULOS DE COMUNICACIÓN INALÁMBRICA.....	50
4.6. TRAMA DE COMUNICACIÓN .....	50
4.7. FABRICACIÓN.....	53
4.8. PROGRAMACIÓN .....	54
<b>5. RESULTADOS Y DISCUSIÓN .....</b>	<b>56</b>
5.1. INTERFAZ .....	56
5.2. DESCARGA DE DATOS. ....	58
5.3. PRESENTACIÓN DE DATOS. ....	59
5.4. ALCANCE .....	62
<b>6. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>63</b>
6.1. CONCLUSIONES .....	63
6.2. RECOMENDACIONES .....	64
<b>7. LITERATURA CITADA. ....</b>	<b>65</b>
<b>8. ANEXOS. ....</b>	<b>68</b>
8.1. CRC-8.....	68
8.2. LISTAS DE MATERIALES.....	69
8.3. SIMBOLOGÍA ELECTRÓNICA. ....	71
8.4. DIAGRAMAS ELÉCTRICOS.....	75
8.4.1.DISPOSITIVO COORDINADOR.....	75
8.4.2.DISPOSITIVO ESCLAVO .....	80
8.5. CIRCUITOS IMPRESOS .....	83
8.4.1 DISPOSITIVO COORDINADOR .....	83
8.4.2 DISPOSITIVO ESCLAVO .....	83
8.6. CÓDIGOS DE PROGRAMACIÓN C .....	84
8.6.1.DISPOSITIVO COORDINADOR: .....	84
8.6.2.DISPOSITIVO ESCLAVO: .....	100

# Índice de figuras

Figura 1. Ventilación cenital natural (Serrano, 2002). .....	8
Figura 2. Ventilación forzada por medio de extractores (Jiménez, 1999). .....	9
Figura 3. Estructura general de un sistema de adquisición y distribución de señales para medida y control. (Pallás, 1993). .....	17
Figura 4. Señal senoidal y su versión digital. ....	18
Figura 5. Transmisión de un mensaje. ....	19
Figura 6. Modulación en amplitud. ....	21
Figura 7. Estructura de un dato dentro de la interfaz RS232. ....	23
Figura 8. Estructura general de un sistema de telemedida (Pallas, 2003). ....	26
Figura 9. Zona de Fresnel. ....	27
Figura 10. Vista del invernadero experimental. ....	30
Figura 11. Posición de los sensores dentro del invernadero; los números indican cada dispositivo esclavo y la letra "C" el dispositivo coordinador. ....	31
Figura 12. Dispositivo esclavo colocado en una de las macetas de prueba. ....	31
Figura 13. Dispositivo coordinador. ....	32
Figura 14. Diagrama de bloques del dispositivo coordinador. ....	33
Figura 16. Diagrama de flujo del programa principal del dispositivo coordinador. ....	35
Figura 15. Diagrama de flujo del programa en la interrupción del temporizador uno del dispositivo coordinador. ....	36
Figura 17. Árbol de menús en el dispositivo coordinador. ....	40
Figura 18. Conjunto de módulos sin su cubierta. ....	41
Figura 19. Diagrama a bloques del dispositivo esclavo. ....	42
Figura 20. Sensor HHT02. ....	43
Figura 21. Diagrama a bloques del sensor HHT02. ....	43
Figura 22. Tensiómetro digital Blumat. ....	46
Figura 23. Módulo esclavo. ....	47
Figura 24. Diagrama de flujo del programa principal en el dispositivo esclavo. ....	48
Figura 25. Curvas de calibración en cada uno de los cinco tensiómetros (T). ....	49
Figura 26. Trama de comunicación, cada división representa un byte. ....	50
Figura 27. Petición de lectura enviada por el coordinador al esclavo 1. ....	52
Figura 28. Respuesta de lectura enviada del esclavo 1 al coordinador. ....	52
Figura 29. Diagrama de flujo de la interrupción del puerto RS232. ....	53
Figura 30. Compilador PCWHD. ....	55
Figura 31. Software de programación PICKit e Interfaz de grabación. ....	55
Figura 32. Elección del puerto de comunicación en la HyperTerminal. ....	56
Figura 33. Configuración del puerto en la HyperTerminal. ....	57
Figura 34. Comunicación en tiempo real por medio de la Hyperterminal. ....	57
Figura 35. Selección de captura de texto en la HyperTerminal. ....	58
Figura 36. Descarga de datos en la HyperTerminal. ....	58
Figura 37. Comparación de lecturas de humedad relativa entre el sensor HHT02 y la estación agrometeorológica de Montecillo. ....	59
Figura 38. Comparación de lecturas de temperatura entre el sensor HHT02 y la estación agrometeorológica de Montecillo. ....	59
Figura 39. Muestreo de humedad relativa, dispositivo esclavo número dos. ....	60
Figura 40. Muestreo de temperatura, dispositivo esclavo número dos. ....	60

<i>Figura 41. Muestreo de tensión, dispositivo esclavo número dos.....</i>	<i>61</i>
<i>Figura 42. Diagrama eléctrico de microcontroladores.....</i>	<i>75</i>
<i>Figura 43. Diagrama eléctrico del selector sobre el puerto RS232.....</i>	<i>76</i>
<i>Figura 44. Diagrama eléctrico de la etapa de memoria.....</i>	<i>77</i>
<i>Figura 45. Diagrama eléctrico del modulo de transmisión inalámbrica. ....</i>	<i>78</i>
<i>Figura 46. Diagrama eléctrico de conectores de teclado y pantalla LCD.....</i>	<i>79</i>
<i>Figura 47. Diagrama eléctrico de reguladores de voltaje internos. ....</i>	<i>79</i>
<i>Figura 48. Diagrama eléctrico del esclavo. ....</i>	<i>80</i>
<i>Figura 49. Diagrama eléctrico del modulo de comunicación inalámbrica. ....</i>	<i>81</i>
<i>Figura 50. Diagrama eléctrico del sensor HHT02, amplificador para la señal del tensiómetro.....</i>	<i>82</i>
<i>Figura 51. Diagrama eléctrico de reguladores de voltaje internos. ....</i>	<i>82</i>
<i>Figura 52. PCB del dispositivo coordinador.....</i>	<i>83</i>
<i>Figura 53. PCB del dispositivo esclavo.....</i>	<i>83</i>

## Índice de cuadros

Cuadro 1. Rangos de valores y espacio en memoria de los parámetros almacenados. ....	38
Cuadro 2. Estado de la comunicación al realizar una lectura. ....	39
Cuadro3. Medición de temperatura en el sensor HHT02.....	44
Cuadro 4. Códigos de dirección para cada dispositivo. ....	51
Cuadro 5. Códigos de funciones. ....	51
Cuadro 6. Valores de CRC-8 calculados usando el polinomio $X^8 + x^5 + x^4$ .....	68

## **1. Introducción.**

La producción agrícola, sea de cualquier tipo, siempre tiene como fin el consumo; este es el punto de partida que cualquier productor debe considerar a la hora de invertir en un proyecto.

En México, como en el resto del mundo, la superficie dedicada a la agricultura disminuye a causa del crecimiento de la mancha urbana, la contaminación y el cambio climático global. A causa de esto, la agricultura en los últimos años, ha cambiado su forma convencional para abrir paso a otras alternativas.

La creación de nuevas tecnologías ha permitido producir de manera intensiva algunos cultivos, como en el caso de la “agricultura protegida” que se origina al establecer una barrera entre el medio ambiente externo y el cultivo; dentro de las técnicas más comunes se encuentran la malla sombra, el acolchado, los invernaderos y la agroplasticultura, en general.

En presencia de una cubierta, como es el caso de los invernaderos, las condiciones del clima interno cambian en comparación con las que se presentan en el exterior. La radiación y la velocidad del aire se reducen, la temperatura y la presión de vapor del agua en el aire se incrementan y las fluctuaciones en la concentración de dióxido de carbono son más fuertes (Kempkes et al., 1998).

Una de las ventajas del uso de invernaderos es la adición de un microclima aislado del exterior, que puede controlarse para llevar el cultivo a las condiciones más adecuadas para desarrollar su potencial productivo al máximo posible, para esto es necesario conocer el estado de cada una de ellas en todo momento y así llevar a cabo acciones para mantenerlas estables, esto se logra por medio de sensores que permiten la observación de dichas condiciones. Al incluir sistemas automáticos en tareas que se realizan de forma manual se incrementa la precisión, velocidad y calidad en su ejecución, sin embargo, en algunos casos el tamaño de la aplicación genera nuevos problemas, como en el caso de sensores distribuidos en invernaderos de grandes dimensiones donde el costo de instalación se incrementa.

Los sistemas electrónicos son los más utilizados en procesos de monitoreo y control debido a su versatilidad; si la comunicación entre los sensores y alguna interfaz de interpretación es por medio de cables, el precio de estos aumenta a medida que cubren una mayor distancia, unido a la necesidad de cambiar la posición de alguno de ellos. Una solución para estos problemas es la telemetría; esta tecnología, permite la medición remota de magnitudes físicas y envío de información mediante comunicación inalámbrica; sin embargo, la elección del grado de control ambiental está limitada por la inversión de capital y los costos de operación que el usuario esté dispuesto a asumir (Rodríguez y Berenguer, 2004).

## **2. Objetivos e hipótesis.**

### **2.1. Objetivo general.**

Diseñar e implementar un sistema de telemetría para variables climatológicas y de humedad del sustrato en invernadero.

#### **2.1.1. Objetivos particulares.**

- Diseñar un sistema de monitoreo para variables climatológicas (temperatura, humedad relativa y humedad del sustrato).
- Establecer un modo de comunicación inalámbrica entre sensores.
- Desarrollar un algoritmo de control que permita regular y almacenar la información de los sensores.
- Elaborar módulos independientes basados en microcontroladores para la toma automática de mediciones.
- Incluir una interfaz de acceso a la información desde una computadora personal.

### **2.2. Hipótesis.**

Es posible implementar un sistema de telemetría que pueda proporcionar información confiable en tiempo real del estado ambiental de un invernadero.

## **3. Revisión de literatura.**

### **3.1. Variables climáticas en invernadero**

El clima en un invernadero, por lo general, está compuesto por la radiación solar, la temperatura, la humedad y la concentración de CO<sub>2</sub> en la atmósfera, así como varios estados definidos por las plantas. Los flujos controlables son la entrada de calor, el de ventilación y el de CO<sub>2</sub>. Por otro lado, los principales flujos en el cultivo son la evapotranspiración y la formación de carbohidratos en la fotosíntesis. Estos flujos son funciones dependientes de las condiciones ambientales en un invernadero (van Straten *et al.*, 2000). Los efectos combinados de temperatura y la humedad del aire solo pueden visualizarse de manera aproximada a partir de las graficas de mediciones.

Aun cuando se ha establecido que la temperatura y humedad del aire son determinantes para el desarrollo de las plantas, es difícil establecer con precisión su efecto en el crecimiento y calidad de los productos (Schmidt, 2004).

#### **3.1.1. Temperatura**

La temperatura ejerce gran influencia sobre el crecimiento y metabolismo de las plantas. Existen tres puntos de interés para estudios de desarrollo vegetal: la temperatura óptima en la cual la planta presenta su mayor crecimiento, la mínima por debajo de la cual se detiene a causa del cambio de fluidez en las membranas celulares y la máxima sobre la cual, la degradación de las proteínas vegetales lo reprimen (Batista y Ramírez, 1999).

A partir del valor de la temperatura mínima, el crecimiento de la planta aumenta de manera proporcional a ésta, hasta alcanzar el rango óptimo; al sobrepasarlo, el crecimiento disminuye. Morales *et al.* (2005) señalan que las temperaturas óptimas, máximas y mínimas cambian con cada etapa de desarrollo del cultivo, por lo que se debe tener en cuenta para el manejo de los métodos de calefacción y enfriamiento que se requieran; para esto, es necesario conocer el comportamiento térmico de los invernaderos.

Bastida y Ramírez (1999) indican que el calor se mueve en el interior del invernadero por procesos de convección, ya que la velocidad del viento es prácticamente nula. Esto implica que el aire caliente flota sobre el aire frío, y es el principio que permite extraer calor por medio de ventanas cenitales. Una diferencia de presión aparece permitiendo al aire más fresco entrar por las ventanas laterales. Los materiales plásticos que los cubren son relativamente opacos a la radiación térmica de onda larga y provocan el conocido efecto invernadero, que puede elevar la temperatura en su interior hasta llevarlo por encima del rango que la planta puede soportar (Jiménez, 1999); también reporta que el mayor costo de operación de un invernadero, una vez que se ha completado la inversión inicial, es el de mantener la temperatura en los rangos deseables para el desarrollo del cultivo. Es necesario recordar que uno de los principios básicos de funcionamiento es el movimiento de flotación del aire caliente (Morales *et al.*, 2005). Con esto y las diferencias de presión, se pueden generar ventajas gracias al flujo laminar del viento sobre la superficie de plástico y planear estrategias para calentar o refrescar de manera económica y eficiente los invernaderos, con lo que se podrían reducir los costos de calefacción y enfriamiento.

### **3.1.2. Humedad Relativa**

El vapor de agua es uno de los constituyentes más variables de la atmósfera. A pesar de no constituir una medida cuantitativa de este, la forma más común de medir la humedad ambiental es la llamada humedad relativa, la cual es el porcentaje del vapor de agua, que en un momento dado y a una determinada temperatura, contiene el aire, en relación con la máxima cantidad de vapor que puede alcanzar (Morales *et al.*, 2005).

(Villalobos *et al.*, 2003) explican otro término que ayuda a comprender la dinámica de la humedad ambiental, el contenido de humedad del aire le confiere cierta presión y esta característica se le llama presión de vapor de agua.

Dependiendo de la temperatura, el aire tiene cierta capacidad para retener una cantidad máxima de agua, y que se le conoce como presión máxima de vapor, o presión de saturación. La diferencia entre presión de saturación y de vapor actual es el déficit de presión de vapor y representa la succión del aire para la humedad.

Uno de los principales procesos de interés agrícola en el que se ve involucrado el vapor de agua en el aire es en la transpiración de las plantas. (Bastida y Ramírez, 1999) explican que la transpiración se da a través de los estomas y que constituye un fenómeno de intercambio gaseoso, al igual que el de CO<sub>2</sub>. Para llevar a cabo éste, la hoja debe mantener abiertos los estomas y perder agua. Tijerina (1992) menciona que el agua es transportada por un gradiente de energía que se establece entre los estomas y la cutícula de la planta con la atmósfera, y se transmite al interior de la planta como succión de agua desde la raíz. Este medio permite mover nutrimentos a los distintos órganos de la planta. Las plantas sometidas a humedades relativas cercanas a 100% muestran un escaso crecimiento.

Otro efecto negativo de la alta humedad relativa es que el exceso de agua en el aire se condensa fácilmente en las superficies, con frecuencia sobre el plástico del invernadero o sobre las plantas, favoreciendo diversas enfermedades (Bastida y Ramírez, 1999).

Los impactos combinados de temperatura y humedad del aire solo pueden ilustrarse con poca exactitud en los procesos de producción en invernaderos comerciales. Existe escasa información acerca de flujos de masa y energía de las plantas (Schmidt, 2004).

### **3.1.3. Interacciones climáticas**

La agricultura es una actividad muy compleja por tratar con seres vivos, la producción agrícola en invernadero, complica, aún más, el proceso por su objetivo de controlar y proveer las características climáticas para el óptimo desarrollo de los cultivos. Esto implica que además de las labores para un máximo crecimiento vegetal se implementa una serie de prácticas encaminada al manejo del clima.

Bastida y Ramírez (1999) señalan algunas consideraciones para el manejo de la radiación en invernadero. La luminosidad dentro de este depende de su orientación y la forma o diseño, sobre todo el ángulo de la cubierta con respecto al sol. Todos los materiales empleados para cubiertas de invernadero reflejan una parte de la luz que reciben del sol, en general de 20 a 30% para ángulos cercanos a 90°. Para ángulos mayores, se refleja una mayor cantidad de radiación.

(Muñoz, 2003) señala que el material de la cubierta del invernadero modifica la cantidad y la calidad de la radiación e influye en el balance de energía. Adecuar la geometría en la cubierta para controlar el ángulo de incidencia es una estrategia que permite un mejor aprovechamiento de la radiación. Con el fin de disminuir las temperaturas excesivas durante la primavera y el verano, se pueden encalar las cubiertas de los invernaderos, esta práctica reduce la radiación pero limita la producción de manera considerable.

Es esencial conocer las propiedades de la cubierta plástica empleada, ya que existen varias propuestas en el mercado. Su correcta elección simplifica el manejo del invernadero, así como el ahorro de energía. Dentro de los plásticos disponibles de manera comercial, (Muñoz, 2003) señala cubiertas térmicas para el uso en climas fríos, cubiertas antigoteo, antiadherentes al polvo, así como otras con propiedades radiativas, como mayor difusividad y opacidad en distintas bandas que evitan la proliferación de ciertas enfermedades.

Como ya se ha mencionado, uno de los principales efectos de la radiación es el incremento de la temperatura. Bastida y Ramírez (1999) afirman que la cantidad de calor que entra al invernadero está en función de la cantidad de energía luminosa que penetra a través de la cubierta plástica. Esta se puede regular mediante una malla-sombra colocada ya sea adentro o por fuera, cubiertas plásticas y el encalado. Considerando lo anterior, las longitudes de onda más importantes dentro de un invernadero son las infrarrojas cortas, las cuales pasan a través de los materiales de recubrimiento y son absorbidas por las plantas, el terreno y los materiales estructurales, convirtiéndolas en calor.

Existen varias maneras de disminuir la radiación en el interior del invernadero, como son las mallas sombra, las pantallas térmicas con cubiertas de aluminio y las cortinas negras. Estas estrategias se utilizan para reducir la temperatura; sin embargo, hay que hacer notar que cuando se coloca una malla-sombra corrediza, se evita que la radiación alcance el cultivo, pero continúa calentando el aire del interior.

Se han desarrollado mallas con cubierta superior reflectante y la inferior opaca, lo que reduce en gran medida la cantidad de energía calorífica aportada por la malla al invernadero y en cambio mantiene el aire caliente por las noches, cuando la radiación proveniente del suelo es mayor.

Para el caso de climas cálidos, las temperaturas elevadas se pueden convertir en el principal problema de la producción en invernadero. La opción más inteligente en términos económicos para atacar este fenómeno consiste en un adecuado diseño de las ventanas. Serrano (2002) propone que la superficie que ocupan los huecos de la ventilación lateral este comprendida entre 15 y 20% de la superficie total del invernadero, y 10% en el área de la ventana cenital. Para el caso de contar únicamente con ventanas laterales, su área debe sobrepasar 20% a la del invernadero. Además, señala la importancia de favorecer la ventilación natural, con base en la propiedad que tiene el aire caliente de flotar sobre el fresco, la figura 1 muestra el flujo de la ventilación cenital natural en un invernadero. Para favorecer este proceso, es necesario orientar el invernadero de acuerdo con la dirección de los vientos dominantes en la zona.

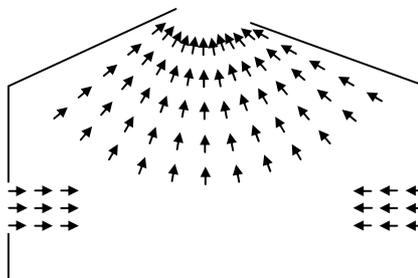


Figura 1. Ventilación cenital natural (Serrano, 2002).

Mediante una correcta ventilación se puede disminuir en gran medida la necesidad de utilizar otros métodos de enfriamiento.

La aireación forzada se utiliza como apoyo para la natural y emplea ventiladores y extractores para proporcionar el movimiento de circulación hacia afuera (Bastida y Ramírez, 1999; Jiménez, 1999) proponen utilizar varios ventiladores de pequeña o mediana capacidad en lugar de solo uno con mayor tamaño, para extraer el aire de los invernaderos a fin de lograr una mejor uniformidad, ver figura 2.

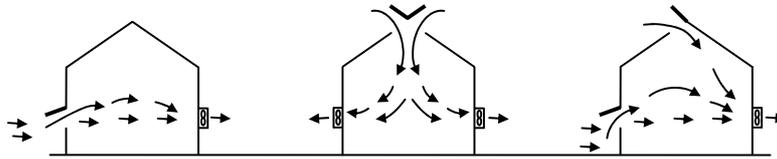


Figura 2. Ventilación forzada por medio de extractores (Jiménez, 1999).

Para los casos en que la ventilación resulta insuficiente para reducir la temperatura del invernadero, se recurre a prácticas que aprovechan la alta capacidad calorífica del agua para absorber calor. Estas son de gran importancia en regiones áridas y semiáridas, donde el objetivo es el de aumentar la humedad relativa del aire (Jiménez, 1999). El muro húmedo consiste en un cojincillo empapado que permite pasar una corriente de aire a través de él. Este se carga de humedad, y al inyectarlo al invernadero se consume calor al evaporarse disminuyendo la temperatura. Otra práctica semejante es el uso de nebulizadores, los cuales son microaspersores que atomizan el agua en pequeñas gotas, las cuales flotan en el aire sobre las plantas. Finalmente se puede mencionar un método común en invernaderos: hacer circular agua sobre su techo y cubierta, formando una película continua y disminuyendo la temperatura al evaporarse en el exterior. También es posible regar el piso de los pasillos con los mismos resultados.

En temporadas de invierno, es necesario incrementar la temperatura para lograr un adecuado desarrollo del cultivo, la primera y más sencilla práctica consiste en cerrar todas las ventanas del invernadero para evitar que el viento remueva el calor acumulado dentro de este. Es necesario asegurar un buen aislamiento de las uniones de plástico para lograr este objetivo.

Existen varios tipos de calefactores que se pueden utilizar en un invernadero. Debido al elevado costo de la energía es preciso considerar que el sistema de calefacción sea compatible con el combustible más económico disponible en la región. Jiménez (1990) así como Bastida y Ramírez (1999) reportan que los calentadores más comunes son los de combustión de gas.

También existen radiadores eléctricos y de rayos infrarrojos. Otro procedimiento consiste en la circulación de agua caliente por tuberías colocadas debajo del piso y en los laterales. En este caso, emplean calderas y algún combustible, o bien, por medio de paneles solares que aprovechan la radiación del día y la almacenan en el agua para su circulación por la noche.

Dado que el costo de la energía empleada para calentar un invernadero es alto, Jiménez (1999) reporta algunas prácticas que permiten disminuir su consumo.

Por ejemplo, menciona experimentos donde se observó una disminución en los requerimientos de calor del orden de 20 a 40% al usar doble cubierta de polietileno. Estas deben mantenerse separadas de la principal por medio de un colchón de aire inyectado con una bomba, además de ser materiales con alta transparencia para evitar que disminuya el flujo de radiación. Otra práctica consiste en la colocación de un techo dentro del invernadero, el cual puede ser permanente o temporal, de una sola pieza de plástico y colocado de tal forma que no interfiera con los tutores de los cultivos, cuando estén presentes, así como con otros elementos de las instalaciones.

Los techos móviles se pueden retirar de día para incrementar la radiación y colocarse de nuevo durante la noche. Esto se puede realizar por medio de plásticos como polietileno, o bien, por medio de mallas sombra de colores oscuros, las cuales absorben gran parte de la radiación infrarroja emitida por el suelo durante la noche. Estos techos funcionan gracias a que mantienen el aire caliente en la parte baja del invernadero, evitando que flote más allá de la altura de las plantas y se pierda calor en la parte superior.

Se ha mencionado que las prácticas comunes para disminuir las temperaturas incluyen el uso de agua por medio de nebulizadores, muro húmedo o riegos. Estas prácticas incrementan la humedad relativa al interior del invernadero, lo cual puede presentar problemas. Una atmósfera húmeda aumenta el riesgo de incidencia de enfermedades fungosas, además los frutos jugosos corren el riesgo de agrietarse. Por el contrario, la falta de humedad ambiental se convierte en un factor limitante que provoca deshidratación en los cultivos y aborto de flores y frutos (Morales *et al.*, 2005).

Las técnicas que se utilizan para el manejo de la humedad relativa se han mencionado, desde el punto de vista de control de temperatura, pues ambos aspectos se encuentran íntimamente relacionados. Se incrementa la humedad relativa por medio de muros húmedos, nebulizadores o riegos, y se puede abatir extrayendo el aire húmedo del invernadero o elevando la temperatura (lo que provoca que el aire pueda contener mayor cantidad de agua sin saturarse).

El manejo de la humedad relativa tiene una estrecha relación con la temperatura, pero también con el incremento de gases como el dióxido de carbono. El intercambio gaseoso se lleva a cabo en los estomas e incluye el transporte de vapor de agua, oxígeno y dióxido de carbono, principalmente.

En general, se busca que las concentraciones de CO<sub>2</sub> al interior de los invernaderos superen la concentración normal en la atmósfera, la cual es aproximadamente 387.27 ppm de acuerdo con el observatorio Mauna Loa en Hawaii para diciembre 2009. Bastida y Ramírez (1999) reportan que la mayoría de las plantas tienen su óptimo desarrollo entre 600 y 900 ppm de CO<sub>2</sub> en el ambiente. En los invernaderos donde no existe una ventilación apropiada, la concentración de CO<sub>2</sub> puede ser mayor que la del aire durante las primeras horas de la mañana, pero se reduce rápidamente conforme se incrementa el flujo de radiación y con ella la tasa de fotosíntesis. Cuando se nubla o disminuye la cantidad de luz, se incrementan gradualmente estos niveles. Si el invernadero permanece cerrado o la circulación del aire se dificulta dentro del mismo (situación que ocurre con los cultivos densos y altos) la concentración de CO<sub>2</sub> disminuye debido a que es consumido por las plantas y no se renueva desde el exterior. En estas condiciones, se convierte en una limitante para el desarrollo del cultivo y es necesario aportarlo.

El principal método para inyectar CO<sub>2</sub> al invernadero es la circulación continua del aire aunque, cuando ésta es reducida, es posible inyectarlo por medio de la quema de combustibles o desde tanques comerciales. La desventaja de estos es el fuerte incremento en los costos. La combustión debe ser eficiente para aportar solo CO<sub>2</sub> y agua, y evitar la acumulación de gases tóxicos, los cuales se deben remover para evitar fisiopatías (Jiménez, 1999).

#### **3.1.4. Tecnificación**

Una de las principales características de la producción en invernadero es la capacidad para mantener un control del clima interior. Para lograr esto, se están desarrollando sistemas de automatización que incorporan el uso de sensores y estrategias de control.

Dentro del diseño de estos sistemas existen varios enfoques, haciendo énfasis en aspectos como la economía, la exactitud y la precisión o sencillez del diseño. En cuanto a las estrategias de control, las tendencias son el dar mayor importancia a los modelos matemáticos del cultivo, del invernadero y el clima; o a las mediciones directas de variables físicas.

Prenger *et al.*, (2002), Zolnier *et al.*, (2004) y Fernández *et al.*,(2001), realizaron experimentos para calcular la evapotranspiración de árboles de maple, plantas de lechuga y melón. Sin embargo, estos valores solo son útiles para programar el riego, mas no para efectuar un control en tiempo real.

Un modelo de cultivo muy difundido es el TOMGRO, de crecimiento dinámico de tomate (Jones *et al.*, 1991). Los autores reportaron que éste describió de forma correcta las diferencias en crecimiento y producción, y lo recomiendan para estudiar los posibles efectos de los controladores de clima.

Los datos provenientes de modelos matemáticos, donde la teoría de ecuaciones en derivadas no lineales han sido una sus principales herramientas, son difíciles de usar para controlar el clima y riego en tiempo real, ya que dichas ecuaciones carecen de utilidad sin los datos necesarios para relacionarlas con el fenómeno de interés y estos no pueden ser más que parciales (imposible medir un medio continuo en todos sus puntos) y su precisión no es suficientemente alta como para garantizar una respuesta adecuada de control.

Challa (2002), toca el tema de los modelos matemáticos en comparación con sistemas que utilizan sensores. Menciona que el concepto de “planta que habla”, en el cual se confía en técnicas de fitomonitorio basadas en mediciones (flujo de savia, temperatura de los tejidos, intercambio de gases, etc.), permite usar información en línea para seguir algunos procesos de producción.

El desarrollo de modelos de crecimiento de los cultivos, integra diferentes disciplinas que proveen herramientas para un manejo más flexible, eficiente y efectivo de la cadena productiva.

A este respecto, Albright (2002), señala que los cálculos para los sistemas modelados con más de una magnitud medible son tan extensos que se vuelven prohibitivos para su aplicación en control óptimo en tiempo real. También sugiere el desarrollo de modelos simplificados más flexibles.

Un punto interesante a favor del control basado en sensores es la dificultad de calcular de manera precisa la evapotranspiración, según los métodos mostrados previamente, comparados con la sencillez de obtener el déficit de presión de vapor y relacionarlo con las necesidades hídricas para determinar el momento de riego. El déficit de presión de vapor ha demostrado ser un valioso indicador para la investigación y producción en invernadero. Van de Saden y Veen (1992), realizaron un experimento donde compararon el crecimiento de pepino respecto a la humedad del ambiente y encontraron que están fuertemente relacionados, aunque no de manera lineal. Por otro lado, (Schmidt, 2004) reporta que la transpiración de un cultivo de jitomate varía de manera proporcional al déficit de presión de vapor.

La necesidad de desarrollar sistemas de bajo costo, capaces de suministrar un máximo de información con un mínimo de instrumentos de medición, ha sido señalado por varios autores (Schmidt, 1998; Jiménez ,1999 y Almanza y García, 1997); sin embargo, es necesario conocer los sensores antes de evaluar la calidad de los algoritmos de control. Así mismo, hay que tener en claro los objetivos de las mediciones con respecto a los invernaderos, es decir, la finalidad de la información, los tipos de sensores disponibles y la manera que deben utilizarse (Coulon, 1998).

Otro enfoque, que considera la combinación de los dos anteriores, es el que se basa en métodos de control robustos no convencionales y que toman en cuenta el error en la medición, éste es un aspecto importante que puede deberse a la mala calibración de los instrumentos o a las condiciones cambiantes del clima y a la interacción del proceso y el ambiente.

El control adaptativo no lineal, que varia los parámetros ajustables en un sistema de modo que mantenga el desempeño óptimo independientemente de las modificaciones ambientales, ha sido usado con éxito para controlar sistemas dinámicos mediante sensores ordinarios y económicos a diferencia de los comúnmente usados en tratamientos analíticos de control que son especializados y de costo elevado, además de ser delicados y susceptibles a perder la calibración (Sigrimis y Rerras, 1998). Asimismo, se han presentado sistemas basados en redes neuronales para tareas de clasificación de plantas y semillas (Moshou *et al.*, 2001).

Además de la naturaleza intrínsecamente no-lineal de los sistemas de control climático, Gates *et al.*,(2001) mencionan que otra ventaja del control adaptativo (lógica difusa) es la factibilidad de implementar algoritmos de este tipo con microprocesadores estándar, incrementando en gran medida la versatilidad y robustez del control, además de facilitar su conceptualización gracias al uso de variables lingüísticas. Este tipo de sistemas de control está teniendo una aceptación cada vez mayor, gracias a que permiten disminuir los costos de instrumentación y cómputo; proporcionan además, una mayor flexibilidad para diferentes cultivos o variedades considerando el deterioro de los materiales o respuesta a los eventos imprevistos. Finalmente, el punto que resulta más difícil de superar es el de la transferencia de tecnología desarrollada en universidades y centros de investigación y su integración en la cadena productiva propiamente dicha. Kassler (2001), hace una revisión detallada de las razones que dificultan el proceso en la transferencia de tecnología de universidades a los sistemas de producción. Entre muchas otras, hace énfasis en la falta de capacitación en temas tecnológicos hacia los productores, así como la falta de desarrollos mecánicos y electrónicos especialmente diseñados para la producción agrícola. En estos puntos coincide con Baille (2001), quien menciona que la inversión en tecnología es mucho menor de lo que desearían los investigadores.

Atribuye esta limitada aportación a los altos costos de los equipos automáticos para los productores, y a la falta de una evaluación costo-beneficio que justifique su compra.

Sobre el tema de la capacitación, Van Straten *et al.* (2000), hacen varias recomendaciones para facilitar la transferencia tecnológica a los productores.

Entre ellas están cuantificar los beneficios, presentar un programa de transición gradual entre sistemas, diseñar los controladores con un enfoque general abarcando varios tipos de invernaderos y cultivos y algoritmos transparentes fáciles de comprender.

Además, es deseable que la interacción del productor con el dispositivo sea mínima, pues de otro modo pierde sentido la automatización.

## **3.2. Monitoreo**

### **3.2.1. Sistema de control**

Un sistema de control es un dispositivo o grupo de dispositivos que actúan conjuntamente para dirigir o regular su propio comportamiento o el de otros y alcanzar un objetivo específico (Ogata, 1998). Para lograr esto, algunos, mantienen una relación entre la entrada de referencia y la salida controlada, lo que les permite modificar las acciones correctivas y adecuarse a diferentes situaciones.

Dada la gran versatilidad de computadoras electrónicas, su amplia variedad, capacidad y métodos en el tratamiento de la información, hacen que la mayoría de los controladores se basen en ellas para realizar sus funciones.

Estas varían de acuerdo a su capacidad y comprenden desde microcontroladores, pequeños ordenadores en un simple circuito integrado para pequeñas aplicaciones, hasta computadoras personales de propósito general.

### **3.2.2. Sistema de medición**

La información es un conjunto de datos que proporcionan el significado o sentido de las cosas. Herrera (2004) describe una señal como la manifestación eléctrica de la información.

Cuando la fuente de información entrega mensajes no eléctricos, se requiere un transductor adecuado para convertir el mensaje en una señal.

Por definición un transductor es un dispositivo capaz de transformar o convertir un determinado tipo de energía a otra diferente, por ejemplo un transductor electromecánico transforma la energía eléctrica en energía mecánica; a su vez, los sensores transforman magnitudes físicas o químicas únicamente en magnitudes eléctricas, debido a esto los sensores también entran en la definición de transductores.

Las señales pueden describir una amplia variedad de fenómenos físicos y aunque pueden representarse en muchas formas, en todos los casos la información está contenida en un patrón de variaciones que presenta alguna forma determinada (Oppenheim *et al.*, 1998).

En un sistema de medición hay que adquirir información que se puede exhibir, registrar o utilizar como la señal de entrada en algún dispositivo o sistema secundario. Por ejemplo Hansman (1999) menciona un instrumento básico cuya señal se transmite a un visualizador o dispositivo de grabación donde se puede leer por un observador humano, otro ejemplo sería un sistema de control que compara el resultado de la medida con los objetivos establecidos y actúe en consecuencia sobre un sistema físico o proceso para modificar algún parámetro deseado (Pallás, 1993). En la figura 3 se aprecia la estructura general de un sistema de control que muestra el sentido de la información recopilada por un controlador, a través de un sensor, para realizar alguna tarea de regulación, por medio de un actuador, en base a los objetivos introducidos. La presentación, alarmas y supervisor pueden ser opcionales de acuerdo al tipo de control y la forma que muestra los datos que maneja, incluso encontrarse dentro del mismo controlador.

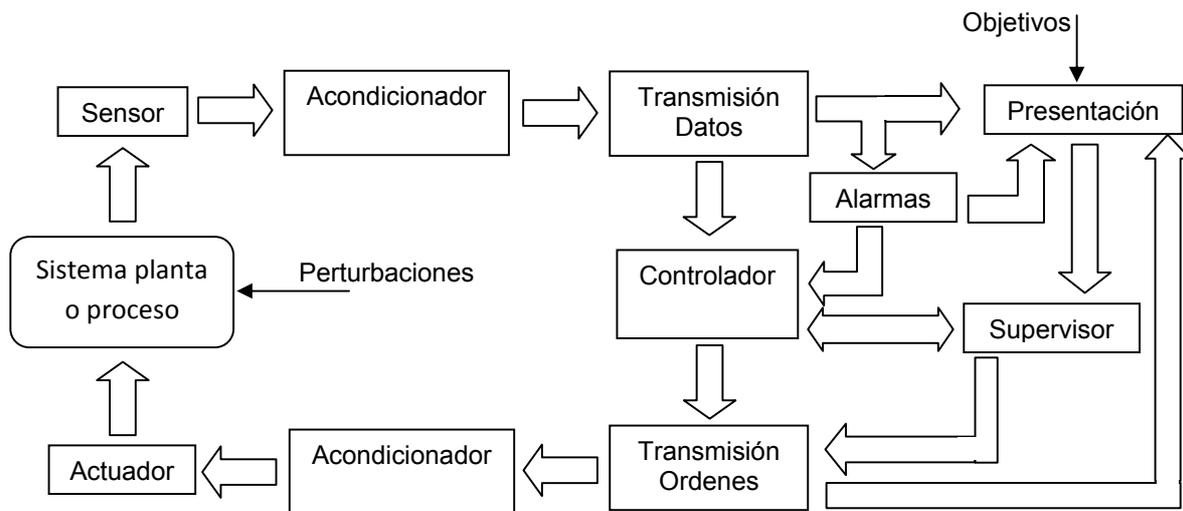


Figura 3. Estructura general de un sistema de adquisición y distribución de señales para medida y control. (Pallás, 1993).

La relación entre señal en la salida de un sensor en específico y los valores correspondientes a un patrón establecido es conocida como calibración, (Hansman, 1999). En muchos casos es necesario que el instrumento proporcione un valor numérico de la señal de modo que se pueda conectar a un sistema de adquisición de datos computarizado. Si el sensor intrínsecamente no proporciona una salida digital, la salida analógica del sensor es convertida por medio de un ADC (*analog to digital converter*) el cual cuantifica y genera un código del dato. Este rango lo define el número de bits del ADC, dado que un bit representa solo dos valores (“1” y “0”) las posibles combinaciones que se pueden hacer con “n” número de bits es  $2^n$ , por ejemplo, al tener un ADC de 3 bits, este generaría un código de  $2^3 = 8$  valores.

Cuanto más grande sea el número de bits del convertidor, mayor será el grado de aproximación que tengan con las señales analógicas, aunque nunca lleguen a ser idénticas debido a que estas últimas contienen infinidad de valores, en la figura 4 se muestra una señal analógica muestreada por un ADC, donde tamaño de los “escalones” está definido por el número de bits.

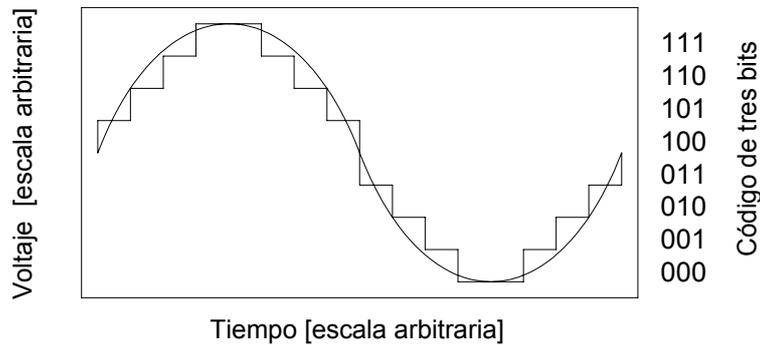


Figura 4. Señal senoidal y su versión digital.

La cantidad de energía usada por un transductor es pequeña y hay que acondicionarla para adaptarla a las etapas siguientes, para aprovechar el margen dinámico del ADC. La amplitud de la señal debe coincidir con la entrada del convertidor. Los circuitos de acondicionamiento se encargan de la amplificación, filtrado y adaptación de la señal del sensor al ADC u otro tipo de etapa posterior. Si los sensores necesitan una señal de alimentación externa, los circuitos de acondicionamiento deben proporcionársela, (Pallás, 1993).

Dado que los acondicionadores de señal manejan solo señales eléctricas sin convertirla en otras formas físicas, constituyen lo que se denomina interfaz. Otras interfaces del sistema son las necesarias para enviar información a través de los canales o líneas de comunicación. Su función y estructura dependen tanto del medio de comunicación como del código empleado para transmitir el mensaje.

Helfrick y Cooper (1991), mencionan que ninguna medición se puede realizar con una exactitud perfecta, siempre existe algún tipo de error y los clasifican en tres categorías:

- Errores gruesos: en gran parte de origen humano como malas lecturas, ajustes incorrectos y equivocaciones en cálculos.
- Errores sistemáticos: debidos a fallas en los instrumentos, como partes defectuosas o gastadas.
- Errores aleatorios: ocurren por causas que no se pueden establecer debido a variaciones en los parámetros o en el sistema de medición.

Existen técnicas que permiten reducir el margen de error en una medición, como determinar el promedio de varias lecturas en vez de efectuar una sola o realizar un análisis estadístico que permita estimar el error.

Field y Solie (2007), clasifican las mediciones más comunes usadas en la agricultura, de la siguiente manera:

Distancia, Área, Temperatura, Volumen, Peso o fuerza, Presión, Tiempo, Velocidad, Torque, Trabajo, Humedad, Temperatura.

### 3.2.3. Sensores inteligentes

Un sensor inteligente es aquel que combina funciones de detección con las de procesamiento de la señal y con las de comunicación. Dado que estas funciones adicionales suele realizarlas un microprocesador, cualquier combinación de sensor y microprocesador se denomina a veces sensor inteligente, estos incluyen, además del sensor primario, algún algoritmo de control, memoria y capacidad de comunicación digital. Los sensores inteligentes reducen la carga sobre controladores digitales ya que ejecutan funciones que estos realizarían si se tratara de sensores comunes. (Pallas, 2003).

### 3.3. Comunicación electrónica

Tomasi (2003), define las comunicaciones electrónicas como la transmisión, recepción y procesamiento de información usando dispositivos electrónicos y pueden ser de forma análoga (proporcional o continua) o digital (etapas discretas). En la figura 5 se muestran los componentes básicos de cualquier forma de comunicación.

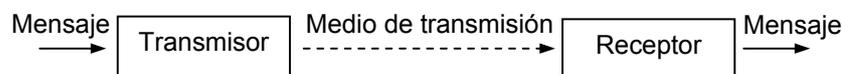


Figura 5. Transmisión de un mensaje.

Anttalainen (2003), se refiere al proceso de transmisión como el transporte de información entre puntos finales de un sistema o red, y lo clasifica de acuerdo al medio en que se transfiere información de uno a otro:

- Cables de cobre, usados en líneas de teléfono.
- Cables de fibra óptica, comúnmente empleados para altos rangos de transmisión.
- Ondas de radio, en teléfonos celulares y satélites.
- Espacio óptico libre, como en los controles remotos infrarrojos de televisión.

Los sistemas de comunicación electrónicos pueden diseñarse para manejar la información solamente en una dirección (modo-simplex), en ambas direcciones pero solo una a la vez (half-duplex), en ambas direcciones al mismo tiempo (modo full-duplex) o en ambas direcciones al mismo tiempo y entre diferentes dispositivos (modo full/full-duplex).

La información contenida en los mensajes que transmiten los sistemas de comunicación no siempre puede viajar tal cual es, por lo que necesita de un proceso que la transforme y adecúe al medio en donde se desplazará, es así como la voz humana se convierte en señales eléctricas al realizar una llamada telefónica, a este proceso se le conoce como modulación. Demodulación es el proceso inverso en el que una onda modulada regresa la información original.

En la radio, toda información debe convertirse a energía electromagnética antes de que pueda propagarse por un sistema de comunicaciones, ésta altera la amplitud, frecuencia o fase de una onda y varía en estos tres factores de acuerdo con el mensaje que se desea transmitir.

Estas técnicas de modulación, en los últimos años, están siendo desplazadas por técnicas digitales debido a que éstas presentan mayor inmunidad al ruido, un procesamiento más sencillo, alta seguridad en los datos y multicanalización. La figura 6 representa una señal digital modulada en amplitud por una onda portadora; cuando llega al receptor la presencia de la portadora muestra el nivel alto del mensaje y la ausencia de esta indica el nivel bajo.

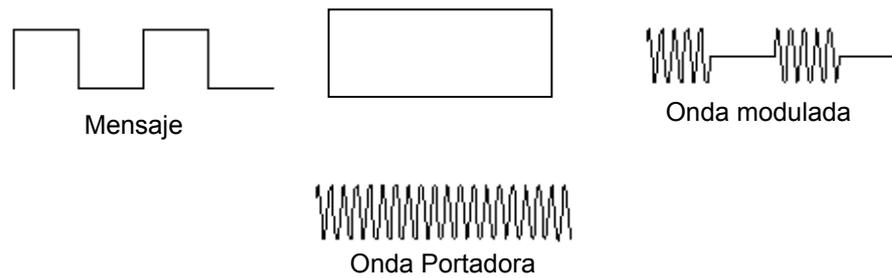


Figura 6. Modulación en amplitud.

En la radio digital, las señales de modulación y demodulación son pulsos digitales en lugar de formas de ondas continuas, sin embargo, usan una portadora analógica para ser transmitidas.

### 3.3.1. Codificación.

Los códigos de comunicación son secuencias prescritas de bits, usadas para representar caracteres y símbolos en las comunicaciones de datos (Tomasi, 2003), y también ayudan a sincronizar el receptor. En una transmisión digital, el receptor se sincroniza con el transmisor en el orden que recibe información cuando cada símbolo llega y de esta forma no necesita retransmitir información adicional como señales de tiempo.

Algunos códigos están diseñados para ofrecer mayor resistencia al ruido e interferencia, tal es el caso del código Barker también llamado código de dispersión o *PseudoNoise*, el cual funciona mezclando un patrón pseudoaleatorio de bits para extender los datos antes que se transmitan, es decir, cada bit de información se modula por medio de una secuencia de bits del código patrón de referencia.

Esto aumenta el tamaño del mensaje que se transmite y hace que la comunicación sea más lenta, sin embargo, cuando un fragmento del mensaje es alterado a causa del ruido o interferencia, éste se puede recuperar si la mayoría del mensaje concuerda con el código patrón.

Los sistemas que solo usan códigos, pero no modulación, son llamados sistemas de transmisión de banda base. Este término se refiere a la banda de frecuencias generadas por algún dispositivo antes de sufrir alguna modulación.

Para comunicaciones en distancias cortas y que viajan por un medio conductor no siempre es necesaria la modulación.

Otro de los códigos más utilizados es el ASCII (*American Standard Code for Information Interchange*) que se basa en el alfabeto latino tal y como se usa en el inglés moderno. El ASCII define códigos para 33 caracteres no imprimibles, de los cuales la mayoría son caracteres de control obsoletos, más otros 95 caracteres imprimibles que incluyen letras y números.

### **3.3.2. Interfaces de comunicación.**

Una interfaz es una conexión física entre dos sistemas o aparatos independientes a través de la cual envían o reciben información entre ellos. Existen diferentes estándares que establecen especificaciones para llevar a cabo dicha conexión, tal es el caso de la interfaz RS232, que define un paquete de ocho bits antecedido por un bit de inicio que sincroniza al receptor para la comunicación y dos de final que indica el término del bloque de datos. La comunicación es efectiva si el receptor está configurado a la velocidad en que se efectúa la transmisión. En la figura 7 se muestra la estructura de un mensaje dentro del estándar RS232 desde los niveles de +3 a +5 V para representar un espacio (cero lógico) y de -3 a -5 V para una marca (uno lógico), en ausencia de información la señal permanece dentro de los niveles de marca, en un mensaje el cambio de nivel en el primer bit señala el inicio de la transmisión, los siguientes siete bits representan los datos, otro indica la paridad de los bits de datos, y dos bits finales indican el término del mensaje; al usar solo un conductor para la transmisión, cada bit se sincroniza en un periodo de tiempo definido tanto en el emisor como en el receptor para realizar una comunicación efectiva.

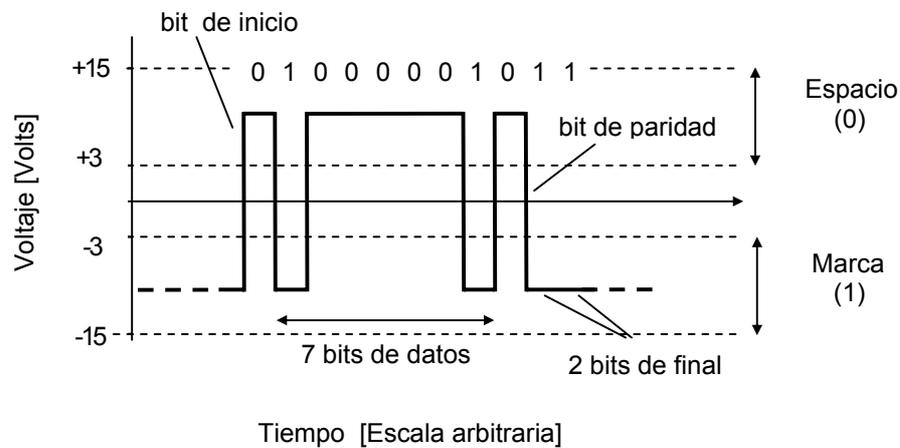


Figura 7. Estructura de un dato dentro de la interfaz RS232.

A diferencia de este tipo de interfaz o puertos que solo pueden comunicarse con un dispositivo, existen otras llamadas *buses* que permiten el tráfico de información ya sea entre componentes dentro de una computadora, o entre varias de ellas. Uno de los más conocidos es el USB (*Universal serial bus*) que define la comunicación entre una computadora y diversos periféricos, soportando hasta 127 aparatos a una velocidad de 5Gbit/s según la especificación del USB 3.0 publicada en 2008. El bus I<sup>2</sup>C (*Inter-Integrated Circuit*) desarrollado por la empresa Philips es muy utilizado en la industria, principalmente para comunicar diversos dispositivos de baja velocidad, 100 Kbit/s en modo estándar aunque también permite velocidades de 3.4 Mbit/s, dentro de sistemas integrados (*Embedded Systems*); este *bus*, utiliza dos líneas de comunicación, una de ellas maneja los datos y la otra una señal de reloj como sincronismo.

### 3.3.3. Protocolos de comunicación.

Las computadoras, deben hablar el mismo “lenguaje” para entenderse unas con otras. Este lenguaje común se define como protocolo y en él se detallan las especificaciones para comunicar dos sistemas diferentes. El protocolo se basa en varias reglas como por ejemplo: cual palabra en código binario representará la letra “A” y que voltaje representara el bit “1”, (Anttalainen,2003); así como en reglas para el direccionamiento, representación, autenticación, detección y corrección de errores.

Uno de estos protocolos es el Modbus, diseñado por Modicom para aplicaciones en controladores lógicos programables (PLC's).

El Modbus no está legitimado por un organismo de estandarización pero es una norma aceptada y usada para muchas aplicaciones donde interviene la conexión de una computadora con una unidad remota (RTU). Existen dos formas de representar los datos en este protocolo: el Modbus RTU, el cual es una representación compacta de datos y finaliza la trama con un CRC (control de redundancia cíclica), y el Modbus ASCII que representa la información por medio de códigos legibles y finaliza con un LRC (control de redundancia longitudinal).

Mientras que el LRC utiliza la integridad del bit de paridad (bit agregado de forma que el número de bits "1" contenidos en un mensaje resulte par) para detectar, y en algunos casos corregir errores en la transmisión de un grupo de caracteres; el CRC asigna un segmento de datos cuyo valor es el resultado de una división binaria de campo finito entre un polinomio generador y el conjunto de datos que se transmiten, donde el cociente es descartado y el residuo se convierte en el resultado.

A este conjunto de datos se le conoce como trama de comunicación y normalmente consta de: inicio, datos y final los cuales permiten un manejo ordenado de información. La trama de comunicación puede presentar anomalías al transmitirse, causadas por ruido eléctrico.

#### **3.3.4. Ruido.**

Lathi (2001) define el ruido eléctrico como una señal indeseable sin relación alguna con la señal deseada y lo clasifica de acuerdo a la fuente que lo genera como:

a) Ruido producido por el hombre. Es la recepción de señales indeseables provenientes de otras fuentes como contactos defectuosos, artefactos eléctricos, radiación por ignición y alumbrado fluorescente.

b) Perturbaciones naturales y erráticas. Ocurren irregularmente y son producidas por relámpagos, tormentas eléctricas en la atmósfera, ruido intergaláctico o disturbios atmosféricos.

c) Ruido de fluctuación. Se presenta en el interior de los sistemas físicos y es producido por la generación aleatoria, recombinación y difusión de portadores (electrones y espacios no ocupados) en semiconductores.

El ruido producido por el hombre se puede evitar eliminando la fuente que lo produce, mientras que las otras dos son inherentes de los sistemas eléctricos y electrónicos, éste puede manifestarse en las comunicaciones en forma de interferencia.

Se entiende por interferencia cualquier señal ajena o no al sistema, que se mezcla con la señal que se transmite provocando perturbaciones en ella, modificando el contenido de la información (Herrera, 2004) por lo cual se hace tan esencial la detección y corrección de errores.

### **3.3.5. Telemetría**

Pallas, (2003) plantea que para distancias grandes, cuando el emisor o el receptor son inaccesibles o interesa que se puedan mover libremente uno respecto al otro, se emplea telemetría vía radio. Para esto, en primer lugar, la señal del sensor, una vez acondicionada, se modula (si su salida es analógica) o se codifica (si su salida es digital), para poder ser combinada con otras que vayan a compartir el mismo medio, y se emplea una comunicación dúplex para poder verificar las mediciones. Si la información se transmite en forma de tensión continua proporcional a la medición, la distancia debe ser pequeña debido a que las tensiones parásitas pueden producir ruido y la medida podría alterarse. La telemetría frecuentemente es inmune a las interferencias, pero no hay normas que permitan utilizar, en un mismo sistema, transmisores de distintos fabricantes. Además, las señales de frecuencia transmitidas pueden ser fuente de interferencias en circuitos próximos.

Según Stojmenović, (2005) una red de sensores inalámbricos consiste en un número de dispositivos inalámbricos capaces de registrar medidas ambientales tales como temperatura, luz, sonido y humedad. Las lecturas de los sensores se transmiten hacia una aplicación que tome decisiones basadas en éstas.

En la figura 8 se muestra la estructura general de un sistema de telemetría: la información viaja, ya sea a través de un medio físico (línea) o por el espacio libre desde algún sensor a una etapa de presentación donde se visualiza o procesa. La primera línea representa a los sistemas simples donde la señal obtenida del sensor no se modifica para transmitirse.

En la segunda interviene la codificación y modulación, pero en ambos casos el alcance se encuentra dentro de la línea de vista (Line-of-Sight, LoS), microondas que operan entre 2 y 6 Ghz puede transmitir a distancias entre 20 y 30 millas; para sistemas que necesitan cubrir una distancia mayor es necesario el uso de repetidores para regenerar la señal. Los sistemas con modo de transmisión dúplex integran algún tipo de MODEM (modulador-demodulador).

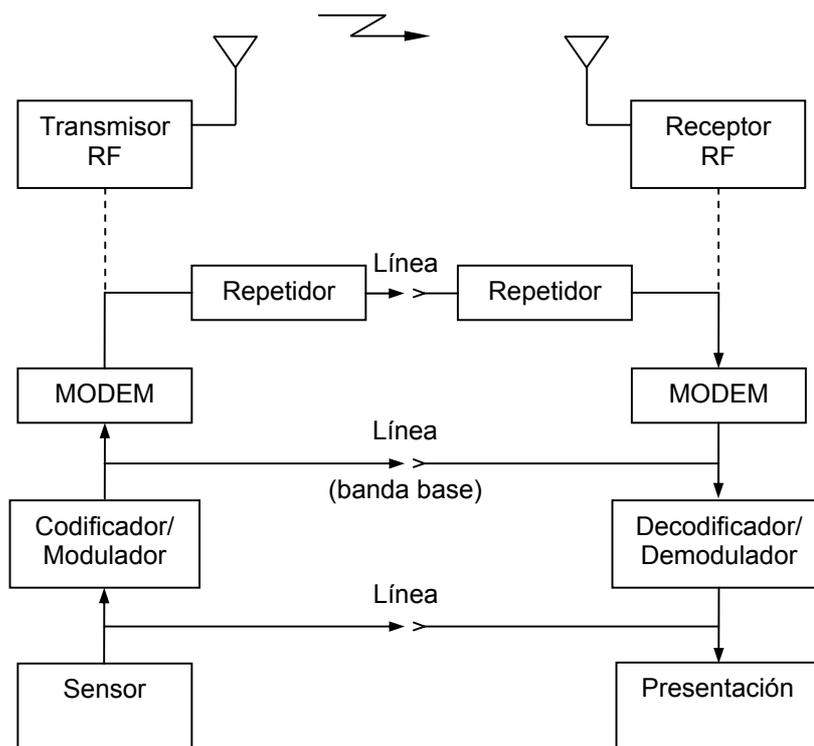


Figura 8. Estructura general de un sistema de telemetría (Pallas, 2003).

### 3.3.6. Alcance

Uno de los puntos más importante en los módulos de comunicación inalámbrica es el alcance que son capaces de cubrir, este es un dato preciso dentro sus hojas de especificación, sin embargo, está calculado de acuerdo a condiciones ideales como el tipo de antena, su altura y una distancia libre de obstáculos entre emisor y receptor conocida como RF LoS (*Radio Frequency Line of Sight*) lo cual resulta lógico al considerar que las obstrucciones reducen el alcance de la comunicación.

Cualquier situación que involucre radiación de ondas implica al fenómeno de difracción provocando interferencia. El espacio donde ocurre esto es uno de los teóricamente infinitos elipsoides concéntricos que definen los volúmenes del patrón de radiación y se le conocen como Zona Fresnel propuesta por primera vez por el ingeniero francés Agustin Fresnel en 1818 en un intento de explicar fenómenos de difracción usando el principio de Huygens (Hristov, 2000).

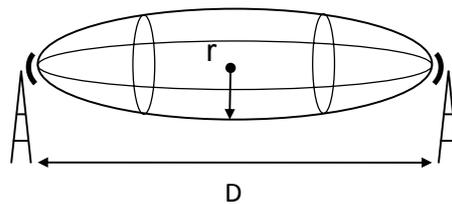


Figura 9. Zona de Fresnel.

Las señales mas fuertes están en la línea directa entre el transmisor y el receptor, y siempre se encuentran en la zona 1 de Fresnel, definida por la siguiente fórmula:

$$r = 17.32 \sqrt{\frac{D}{4f}}$$

Donde:

$r$  = radio en metros

$D$  = distancia total en kilómetros

$f$  = frecuencia transmitida en gigahertz.

Cualquier obstrucción en la zona de Fresnel reduce el alcance de las comunicaciones, incluyendo edificios, vegetación, el suelo, etc. Con el fin de mantener ésto libre es necesario elevar las antenas.

## **4. Materiales y métodos**

### **4.1. Estructura del sistema**

Para el desarrollo del sistema de telemetría se diseñó un dispositivo coordinador encargado de dirigir el tráfico de información y cinco dispositivos esclavos los cuales contienen un sensor HHT02 para medir la temperatura del aire y humedad relativa, además de un tensiómetro digital Blumat para conocer el potencial de agua en el suelo.

Ambos dispositivos están basados en microcontroladores como centro de procesamiento y módulos transceptores de la serie LT, marca LINX technologies para la comunicación inalámbrica, que permiten el intercambio bidireccional de información y con esto es la introducción de una señal de control entre los dispositivos; el suministro de energía para este equipo lo proporcionaron convertidores de voltaje de 127 Vac a 9 Vdc conectados a la red de energía eléctrica.

Como parte del software se implementó un protocolo de comunicación derivado del MODBUS RTU y el MODBUS ASCII, sobre la interfaz RS-232 y una codificación Barker de ocho bits. Con este software se logró el enlace de los dispositivos y la recopilación de información.

Para verificar el sistema se realizó una comparación de los datos obtenidos por éste y los de una estación agrometeorológica usada como referencia.

### **4.2. Ubicación del sitio de prueba**

La validación del sistema de telemetría desarrollado en este trabajo se realizó en el invernadero experimental del Programa en Hidrociencias del Colegio de Postgraduados Campus Montecillo, Texcoco, Estado de México. Localizado a 19° 21' latitud norte y 98° 54' longitud oeste y a una altitud de 2240 m. El clima del lugar es C(Wo)(W)b(i')y corresponde a una categoría templado subhúmedo con lluvias en verano, con una precipitación y temperatura anual de 625 mm y 15.5°C, respectivamente (García, 1987).

El invernadero es de tipo cenital con cubierta de polietileno con dimensiones de 9 m de ancho por 20 m de largo y cubre un área total de 180 m<sup>2</sup>, posee cortinas en tres de sus costados con una altura 3 m y una cortina en la parte superior de 90 cm de ancho. La altura total del invernadero es de 6 m y una distancia del suelo al desplante del arco de 3 m.

Se maneja un cultivo de fresa (*Fragaria x ananassa* cultivar aromas) en bolsas de plástico individuales de 30 por 30 cm y una mezcla de 66% piedra volcánica (tezontle rojo) con una granulometría de 3 a 5 mm y 33% turba (promix) como sustrato. La solución nutritiva utilizada es de tipo *Steiner*. La figura 10 muestra el invernadero en donde se estableció el sistema telemétrico diseñado en este estudio; en la figura 11 se observa la posición donde fueron colocados los sensores dentro del invernadero y en la figura 12 una maceta de prueba donde se desvió el riego a un contenedor, con el propósito de desecar la maceta y lograr lecturas con una mayor tensión de humedad en el sustrato en un tiempo más corto de prueba.



Figura 10. Vista del invernadero experimental

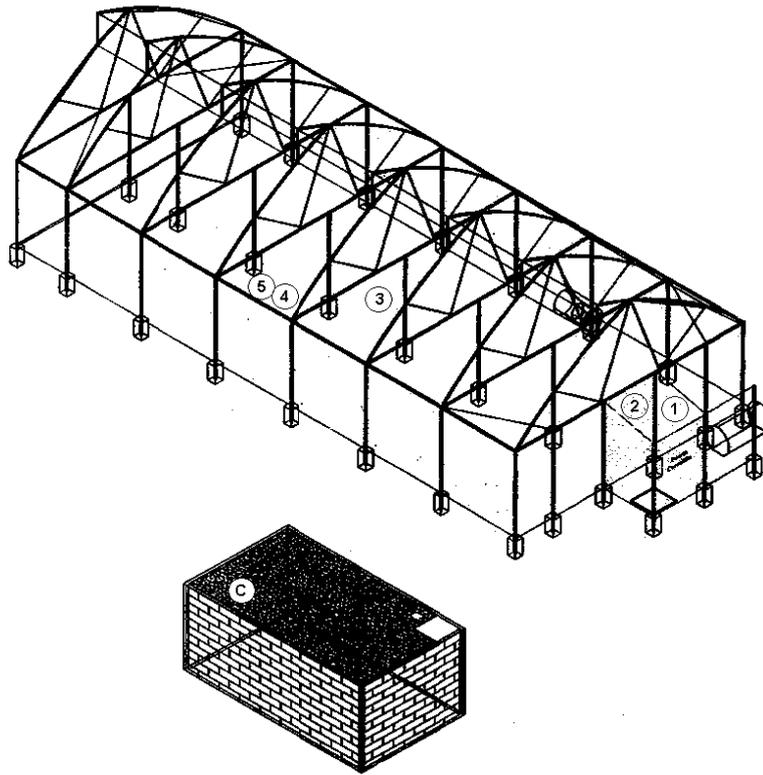


Figura 11. Posición de los sensores dentro del invernadero; los números indican cada dispositivo esclavo y la letra "C" el dispositivo coordinador.



Figura 12. Dispositivo esclavo colocado en una de las macetas de prueba.

### 4.3. Dispositivo coordinador

Este equipo se diseñó sobre un PCB (del inglés *Printed Circuit Board*) de 10 cm de ancho por 15 cm de largo colocado dentro de un gabinete SERPAC A-42, que cuenta con un teclado numérico y una pantalla LCD (*Liquid Crystal Display*) del modelo JHD-5391 como interfaz con el usuario y conector de la norma RS-232 para el intercambio de datos binarios. La alimentación puede variar de 9-35 V ya que cuenta con un regulador interno para adaptarse a baterías y transformadores de corriente directa dentro de este rango.



Figura 13. Dispositivo coordinador.

En la figura 14 se muestra el diagrama a bloques del dispositivo coordinador, las flechas indican el flujo de información. El bloque principal lo ocupa el microcontrolador PIC18F4680 el cual recibe datos provenientes del teclado matricial TECLADO-TEL ya sean de configuración u opciones que se visualizan en la pantalla LCD JHD-5391, controla el reloj/calendario DS1302 para consultar o establecer la fecha y hora, guarda datos en la memoria 24LC1025 y los extrae para descargarlos a una computadora personal por medio de la interfaz RS232; se comunica con los dispositivos esclavos por medio del módulo inalámbrico TRM-433-LT. Dado que la comunicación con los sensores y la PC utilizan la misma interfaz, un selector 74LS126 dirige la información al destino pertinente. Por último, el microcontrolador principal envía una señal que indica su buen funcionamiento, si el microcontrolador supervisor PIC12F683 no recibe dicha señal, reiniciará todo el sistema evitando que el dispositivo quede bloqueado por alguna falla.

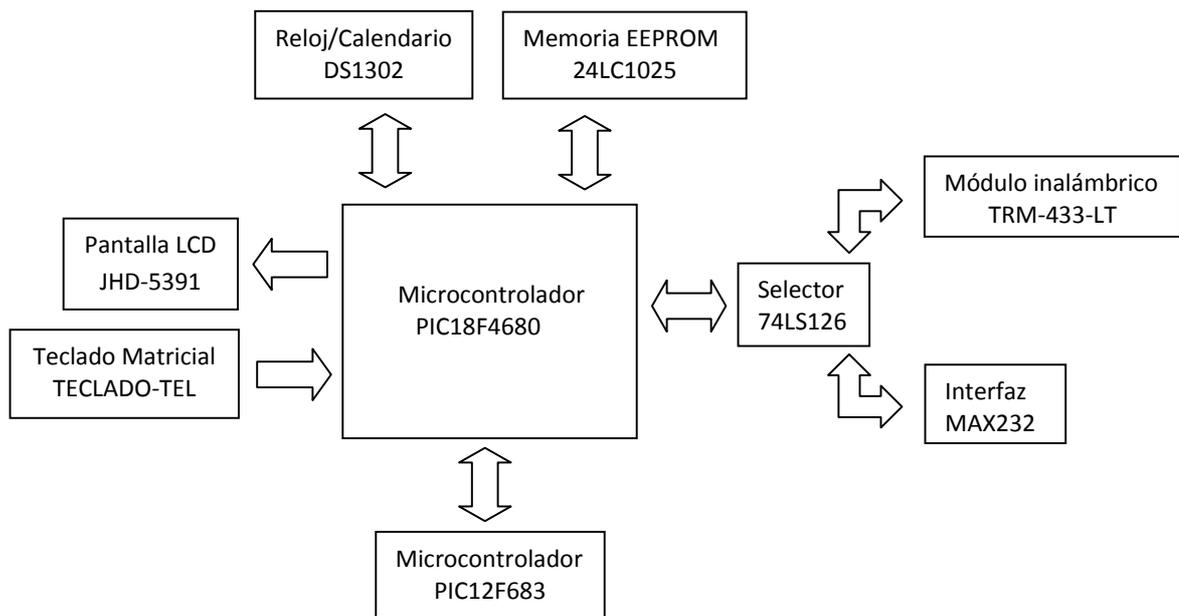


Figura 14. Diagrama de bloques del dispositivo coordinador.

#### 4.3.1. Microcontrolador

El núcleo de este dispositivo es un microcontrolador PIC18F4680 de la empresa Microchip, encargado de realizar funciones de acuerdo a su programación, como son:

- Monitorear la entrada de datos por medio del teclado.
- Desplegar mensajes en la pantalla.
- Configurar tiempos de lectura.
- Monitoreo del tiempo de muestra.
- Coordinar el tráfico de información.
- Almacenamiento de datos.
- Lectura de memoria interna por medio del puerto RS232

El dispositivo cuenta además con otro pequeño microcontrolador de baja capacidad PIC12F683 encargado de monitorear la ejecución del programa principal del PIC18F4680 mediante la retroalimentación de una señal con la cual se recarga un intervalo de tiempo en la que el sistema se reiniciará, es decir, si el PIC12F683 no recibe esta retroalimentación por parte del PIC18F4680 a causa de un mal funcionamiento, después de un tiempo definido, reiniciará todo el sistema.

#### **4.3.2. Memoria.**

Tres memorias EEPROM 24LC1025 con capacidad de 128 Kbytes cada una son las encargadas de almacenar los datos obtenidos a partir de las mediciones; se comunican con el microcontrolador a través del bus I<sup>2</sup>C.

. Estas memorias se dividen en seis bloques de 64 kbytes, donde cada bloque está definido para un dispositivo esclavo.

#### **4.3.3. Reloj.**

Como base de tiempo se empleó un reloj/calendario DS1302, el cual cuenta con un oscilador estándar de 32.768 KHz exclusivo para su funcionamiento y una batería de respaldo para mantenerse funcionando aun sin la alimentación principal. La comunicación por medio de tres vías con el microcontrolador, permite acceder a la fecha y hora o configurarlas según sea conveniente.

#### **4.3.4. Periféricos.**

El puerto RS232 se utiliza para transferir datos a una computadora para su posterior análisis, un selector 74LS126 relaciona las vías de comunicación entre la interfaz con computadora y el módulo de comunicación inalámbrica, ya que comparten este puerto.

#### **4.3.5. Software coordinador**

El microcontrolador se apoya en un programa que siempre se encuentra en ejecución (figura 16), de ahí pueden enlazarse otras funciones dedicadas a realizar operaciones fuera de él, estas funciones son comúnmente usadas en algoritmos que se repiten dentro del programa principal y donde su desarrollo resultaría redundante. Otras fracciones de programa dedicadas a eventos especiales son las peticiones de interrupción y surgen ante la necesidad de los periféricos en ser atendidos, cuando una llamada a interrupción es generada el programa actual detiene la ejecución para realizar las actividades definidas en ésta, luego regresa a la parte donde se encontraba.

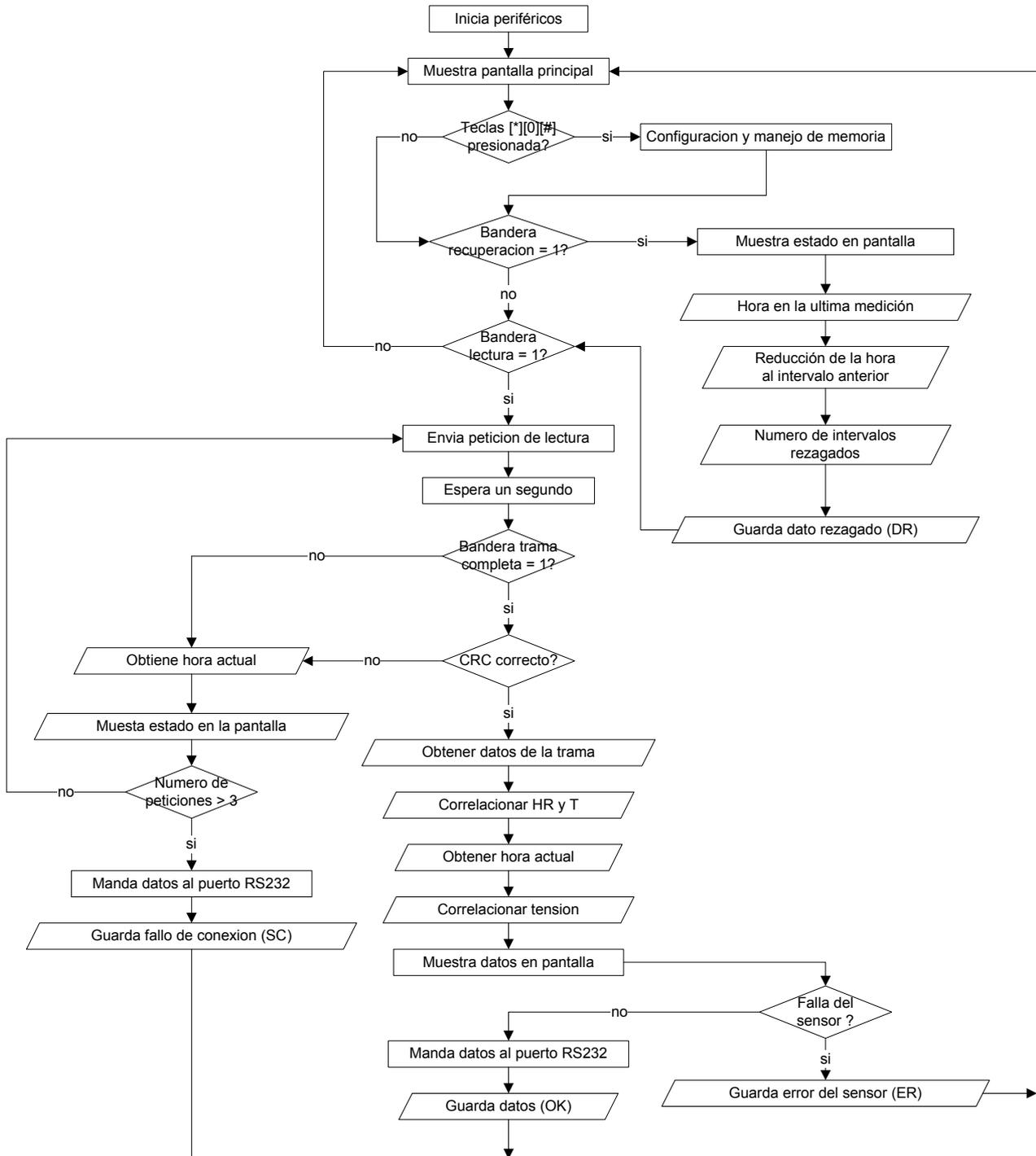


Figura16. Diagrama de flujo del programa principal del dispositivo coordinador.

Dentro del coordinador, el programa principal se limita a mostrar mensajes de fecha y hora en la pantalla junto con el monitoreo del teclado, a su vez cada 250 ms una interrupción es activada y lleva al WDT (*watchdog timer*) a cero, también le envía una señal al microcontrolador PIC12F683; ambos cuentan un intervalo de tiempo definido después del cual reinician todo el sistema, evitando así que éste se bloquee.

La interrupción del temporizador uno se activa cada segundo y se encarga de revisar el reloj para llevar a cabo las lecturas de los sensores en los tiempos configurados como lo muestra el diagrama de flujo de la figura 15, en esta parte se lee la memoria y se compara el tiempo actual con el de la última medición, si el tiempo es mayor a dos intervalos de medición se ejecuta un programa de actualización de datos y llena los espacios que corresponden a cada medición que no se efectuó, dejando los valores en blanco, esto facilita la interpretación de la base de datos al no contener saltos en la hora en que se realizaron las mediciones .

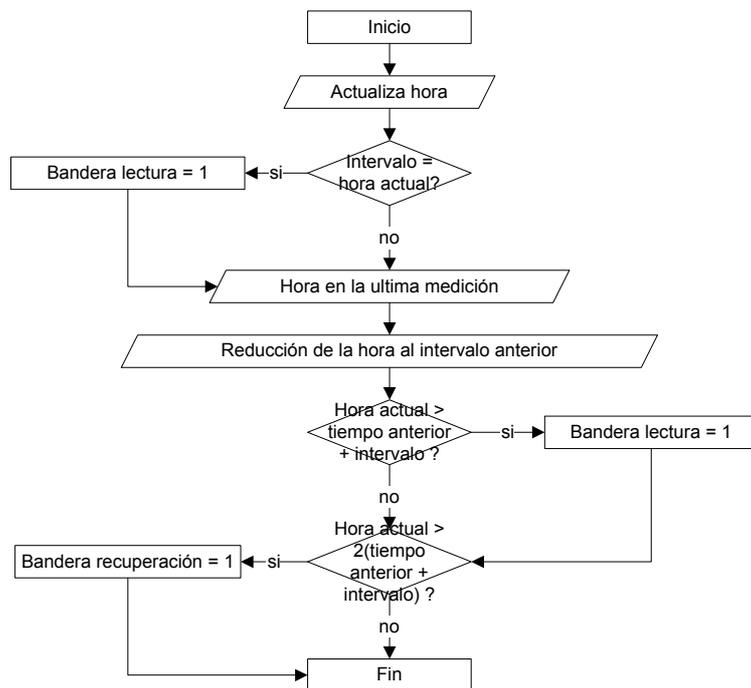


Figura 15. Diagrama de flujo del programa en la interrupción del temporizador uno del dispositivo coordinador.

Por otra parte, si el tiempo actual es igual o mayor al intervalo de medición se realiza la lectura de los sensores, se envía una petición de lectura a cada sensor y se habilita la interrupción del puerto RS232 para recibir información, la tolerancia de espera para cada respuesta es de un segundo, concluido este tiempo se repite la petición en un total de tres intentos, si no hay respuesta del sensor se registra el estado de no conexión dejando los valores en blanco.

Una interrupción, del oscilador, se genera cuando ocurre un mal funcionamiento en la base de tiempo en el sistema, al inicio se utiliza un resonador externo de 8 Mhz como tal, el microcontrolador sincroniza su oscilador interno para monitorearlo y si detecta variaciones en los intervalos de tiempo lo descarta y continúa trabajando con el interno.

Por su parte el subprograma que lleva a cabo las rutinas de configuración y manejo de memoria deshabilita las interrupciones para delegar el control del sistema al usuario, la comunicación con los sensores no se realiza dentro de esta parte. Para acceder al menú principal se presiona un botón en la última fila del teclado ([#] [0] [\*]), siempre y cuando no se esté comunicando, las opciones aparecerán en la pantalla y se eligen oprimiendo el número que le corresponde a cada una de ellas.

Dentro del menú se puede configurar la fecha y hora actual, así como el tiempo de muestra de los sensores en horas, minutos y segundos de cada periodo; para evitar que se desborde la cuenta, el mínimo para las lecturas se define como un segundo.

La respuesta de cada sensor es aproximadamente un segundo desde que una petición es detectada (tres mediciones con duración de 300 ms cada una) sin embargo al no detectarse una conexión, el coordinador repite la petición tres veces; en el caso que ninguno de los cinco sensores responda, el tiempo invertido será de 25 segundos para esta rutina, este valor se puede considerar como el intervalo mínimo para las lecturas en el peor de los casos

En otra sección del menú se puede comprobar la comunicación de cada sensor ya sea por medio de una solicitud lectura, donde el sensor realiza una medición que no se registra en la memoria o mediante una solicitud de eco (PING) en la cual la respuesta es inmediata al no realizar la lectura del sensor y solo muestra si la comunicación que se realizó fue exitosa o no.

En la opción de memoria se envían los datos guardados al puerto RS-232 en forma de texto. Estos códigos binarios pasan por un proceso de compresión lógica después de ser recibidos por los sensores, con esto se utilizan los espacios vacíos dentro del formato de ocho bits que tienen los registros de memoria. Así se hace posible almacenar 9,362 lecturas en cada bloque. En el cuadro 1 se muestra el número de bits que son necesarios para almacenar cada parámetro.

Cuadro 1. Rangos de valores y espacio en memoria de los parámetros almacenados.

Parámetro	Rango de valores	Bits
Mes	1-12	4
Día	1-31	5
Hora	0-23	5
Minuto	0-59	6
Segundo	0-59	6
Tensión [mb]	0-750	10
Humedad relativa [%]	0-100	8
Temperatura [°C]	-40 – 123.8	12

Al momento de leerlos se descomprimen, luego se aplican las ecuaciones de correlación para obtener los valores reales y se transmiten por el puerto. El programa brinca los espacios en blanco y éstos no se transmiten para disminuir el tiempo, aun así la lectura de toda una sección puede durar algunos minutos, a su vez en la pantalla aparece el porcentaje leído de la sección y una opción para cancelar la lectura.

La opción de borrar limpia los datos de todas las secciones en la memoria mostrando también su avance en la pantalla y permitiendo cancelar esta operación, sin embargo, una vez que se ha iniciado, los datos se empiezan a guardar en la primera localidad, ya que el apuntador de memoria toma el valor inicial.

Al salir del subprograma de configuración el programa principal hace un llamado a otro subprograma, este se encarga de actualizar los datos cuyo tiempo en el que debían efectuarse las mediciones ha pasado, para generar una base de datos uniforme al momento de descargarla; este subprograma, también es llamado en el encendido del sistema y es continuamente monitoreado. Los datos se almacenan con la fecha y hora en que se realizaron y al igual que donde no se ha logrado establecer una comunicación con los sensores los valores se dejan en blanco, mostrándose un mensaje en la columna de estado como se muestra en el siguiente cuadro.

Cuadro 2. Estado de la comunicación al realizar una lectura.

Condición	Estado
Correcto	OK
Sin conexión	SC
Dato rezagado	DR
Error del sensor	ER

En la figura 17 se muestra el árbol de menús con las opciones de configuración y manejo de datos, los números entre corchetes indican las teclas que se deben presionar para llevar a la opción seleccionada

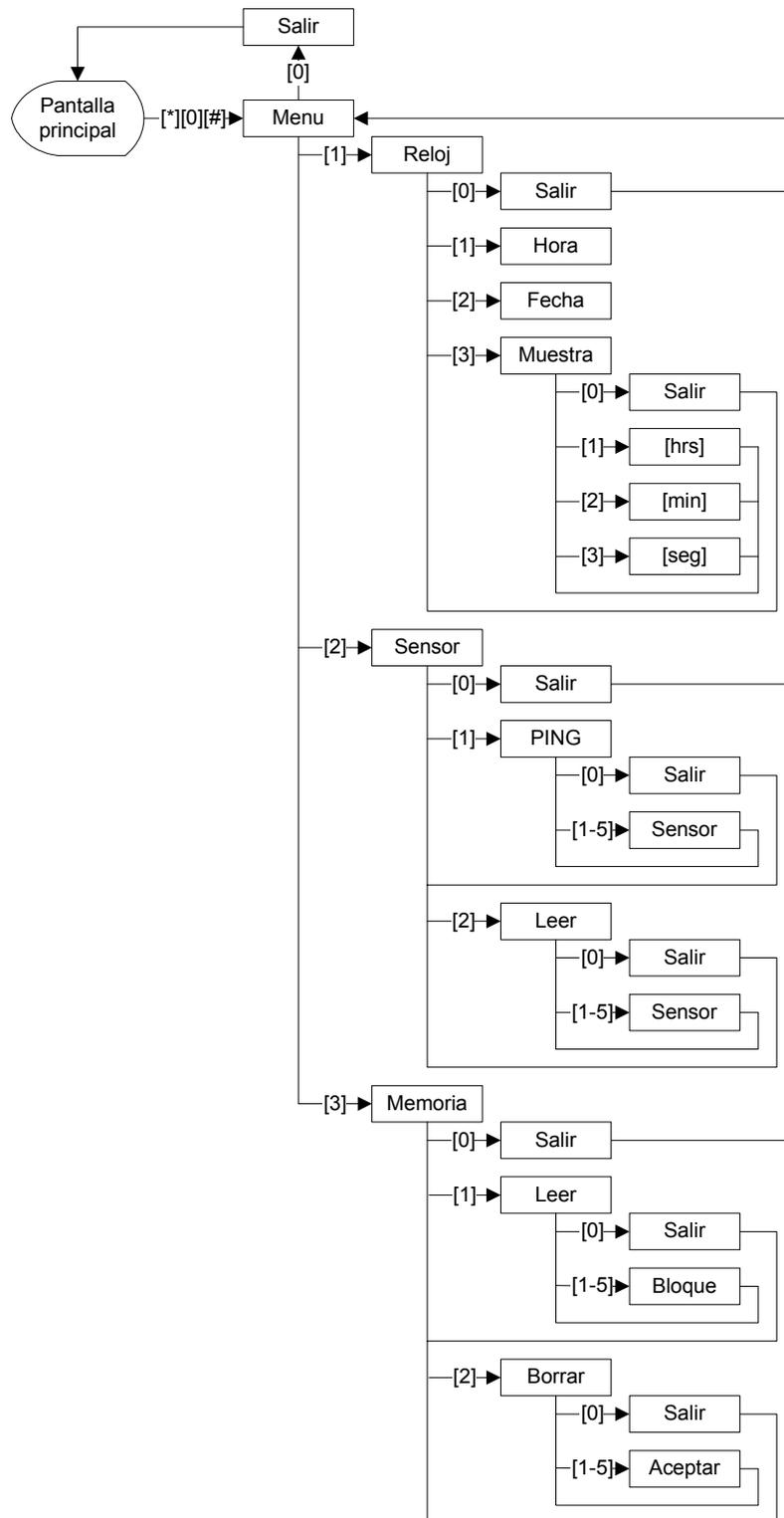


Figura 17. Árbol de menús en el dispositivo coordinador.

#### 4.4. Dispositivo esclavo

Cada uno de los cinco sensores está montado sobre una placa de 5 cm de ancho por 10 cm de largo dentro de un gabinete SERPAC A-20, cuenta con un motor AFB0412MD para ventilación y un conector para el transductor de presión en el tensiómetro. Al igual que el coordinador, la alimentación puede variar de 9-35 V por su regulador interno.

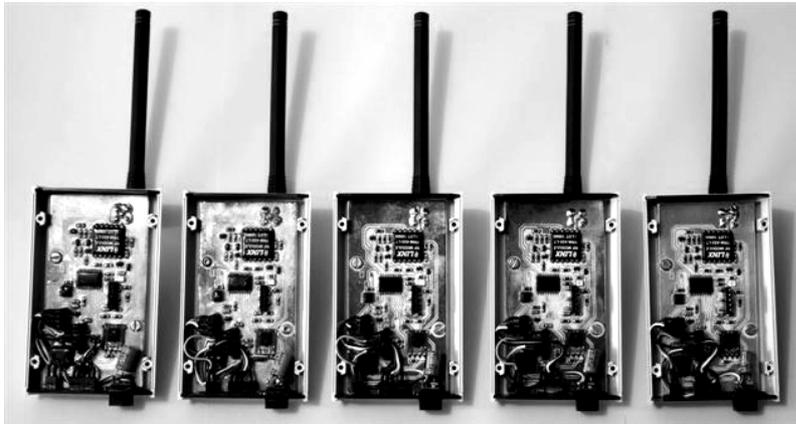


Figura 18. Conjunto de módulos sin su cubierta.

En la figura 19 se muestra el diagrama a bloques del dispositivo esclavo, las flechas indican el flujo de información. El bloque principal muestra el microcontrolador PIC16F88 que monitorea constantemente el modulo de comunicación inalámbrica en espera de alguna petición por parte del coordinador, configura y realiza la lectura del sensor HHT02, así como la lectura del transductor dentro del tensiómetro cuya señal pasa a través de una etapa de amplificación INA122 y transmite la información recopilada. La circulación de aire en el interior del módulo se realiza por el ventilador, su encendido y apagado está a cargo del microcontrolador.

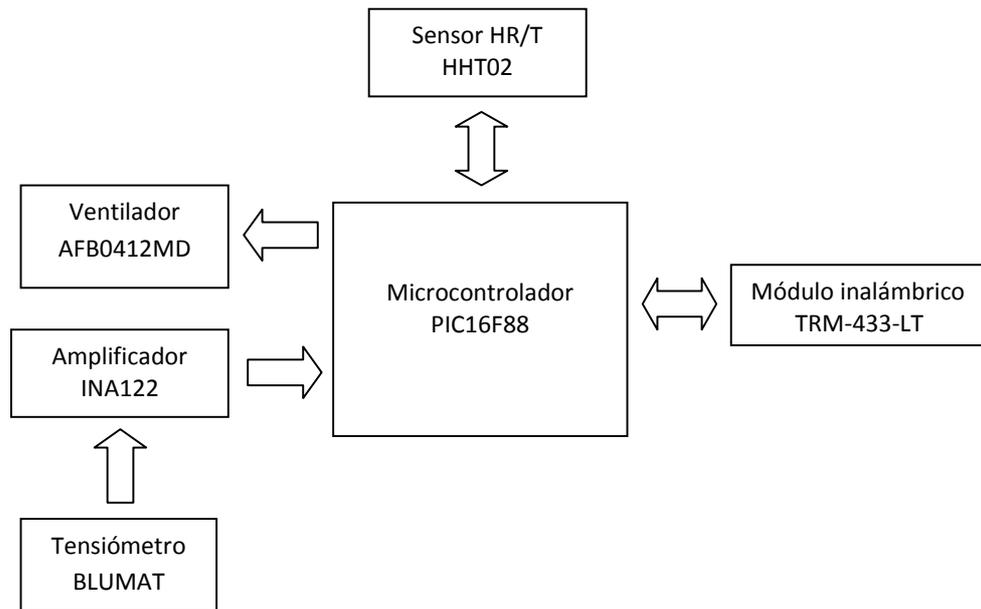


Figura 19. Diagrama a bloques del dispositivo esclavo.

#### 4.4.1. Microcontrolador

El microcontrolador encargado de realizar las funciones de este dispositivo es el PIC16F88, de menor capacidad pero se ajusta a funciones que se realizan en esta aplicación, como son:

- Monitorear peticiones del coordinador.
- Establecer la comunicación con los sensores.
- Generar respuestas a dichas peticiones.

#### 4.4.2. Sensor HR/T

La parte principal del módulo es un sensor de temperatura y humedad HHT02 (figura 20) que integra los dos en un solo encapsulado y proporciona una salida digital.

Este chip incluye un polímero capacitivo como elemento sensor de humedad relativa y un sensor de temperatura en su interior, ambos están acoplados a un ADC y una interfaz serial como se observa en la figura 21.

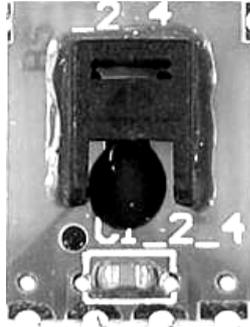


Figura 20. Sensor HHT02

La comunicación se lleva a cabo por medio de una interfaz de dos vías y un protocolo propio similar al I<sup>2</sup>C, cada medición es seguida de un CRC de ocho bits que permite detectar y eliminar datos incorrectos. El tiempo de respuesta de este sensor es de 11, 55 y 210 ms en conversiones de 8, 12 y 14 bits respectivamente.

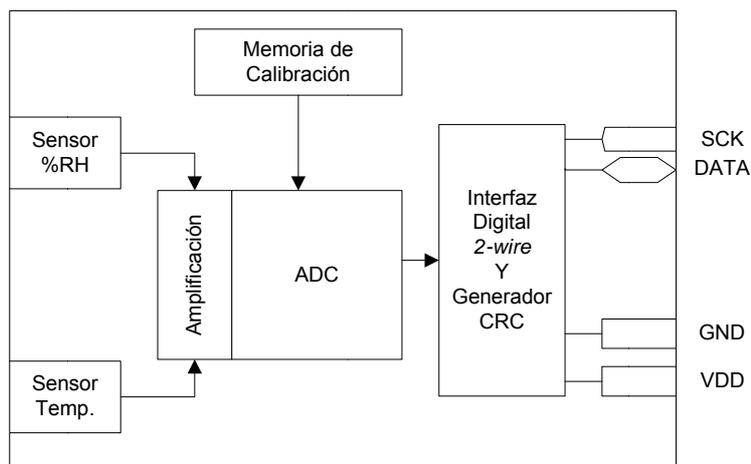


Figura 21. Diagrama a bloques del sensor HHT02.

El siguiente ejemplo muestra la comunicación sensor HHT02 para la lectura de temperatura:

Cuadro3. Medición de temperatura en el sensor HHT02

Descripción	Código	Comentario
Inicio de transmisión		Secuencia de señales entre el DATA y el SCK
Comando	0x06	Escribir en el registro de estatus
Dato escrito	0x01	Configuración 8 bits RH / 12 bits T
Inicio de transmisión		Secuencia de señales entre el DATA y el SCK
Comando	0x03	Lectura de temperatura
Tiempo de respuesta		55 ms
Dato leído	0x06	Byte más significativo de la medición
Dato leído	0x52	Byte menos significativo de la medición
Dato leído	0x9a	CRC

Ya que el algoritmo del sensor invierte un *nibble* (conjunto de cuatro bits) del registro de estatus para generar el CRC, el resultado generado es también el inverso, por lo tanto  $CRC = 0x59$  y se puede calcular de acuerdo al cuadro 6 del anexo como se observa:

1. Se inicializa el CRC con el *nibble* bajo del registro de estatus (invertido ( $s_0, s_1, s_2, s_3'0000$ )), en el ejemplo anterior se utilizó 0x01 para configurar una resolución de 8 bits para humedad y 12 bits para temperatura por lo tanto

$$CRC = 0x80.$$

2. El primer dato es el comando para medir la temperatura 0x03 y realizando la operación lógica de suma exclusiva con el valor anterior tenemos:

$CRC = 0x03 \text{ XOR } 0x80 = 0x83$ , donde el valor correspondiente, según el cuadro 6, sería:

$$CRC = 0x29$$

3. El segundo dato es el byte más significativo de la lectura de temperatura, 0x06 repitiendo lo del punto anterior tenemos:  $CRC = 0x06 \text{ XOR } 0x29 = 0x2f$ , y según el cuadro 6:

$$CRC = 0xa8.$$

4. Para el byte menos significativo 0x52 tenemos:

$CRC = 0x52 \text{ XOR } 0xa8 = 0xfa$  que según el cuadro 6

$$CRC = 0x59$$

Al corresponder el CRC calculado con el obtenido del sensor se asegura que la comunicación esté libre de errores. A continuación se realiza una conversión del valor hexadecimal a decimal ( $0x0652 = 0d1618$ ) para obtener el valor físico de acuerdo con la función de correlación.

$$\text{Temperatura} = (0.04 * 1618) - 40 = 24.72^{\circ}\text{C}.$$

Esta función es proporcionada por el fabricante ya que cada sensor es calibrado individualmente dentro de una cámara de humedad controlada y sus coeficientes se almacenan en la memoria interna para ser utilizados en cada medición.

#### **4.4.3. Tensiómetro**

Como complemento se anexó un tensiómetro en cada módulo esclavo para conocer el potencial de agua en el suelo o tensión, el cual es una medición de la cantidad de energía con que el agua es retenida a la matriz del suelo.

De esta manera los tensiómetros pueden ser usados para fijar el momento de riego cuando la tensión en el suelo es baja, es decir, antes de que la planta entre en estado de estrés hídrico.

Un tensiómetro es un tubo lleno de agua con un medidor de vacío en la parte superior y una cápsula de cerámica en la inferior; cuando es colocado en el suelo, el agua del instrumento entra en equilibrio con el agua del suelo fluyendo a través de la capsula de cerámica. En equilibrio, la tensión del agua en el instrumento es igual a la tensión del agua en el suelo. Cuando se usa un tensiómetro en el campo, la cápsula de cerámica está localizada dentro de la zona de raíces de la planta, entonces se puede medir la tensión de agua en el suelo donde éstas la experimentan. (Smajstrla y Pitts, 1997).

En la lectura del tensiómetro interviene un amplificador de instrumentación INA122 en el que se pueden adaptar transductores que utilizan un puente de Wheatstone para adecuar el elemento sensor; la salida pasa a través de un convertidor analógico a digital con una resolución de 10 bits en el interior del microprocesador para generar una lectura. En este caso se utilizó un tensiómetro de la marca BLUMAT, aunque no se ofrecen especificaciones por ser un sistema cerrado.



Figura 22. Tensiómetro digital Blumat.

#### 4.4.4. Ventilador

Para extraer el aire dentro del gabinete que contiene los sensores se instaló un ventilador modelo AFB0412MD que proporciona un flujo de aire de  $0.20 \text{ m}^3/\text{min}$  con una presión de  $4.10 \text{ mmH}_2\text{O}$ , en la parte superior se realizaron dos perforaciones para facilitar la entrada de aire y permitir su circulación.

Cada apertura fue cubierta por una malla antiáfidos para evitar la introducción de objetos extraños a los módulos



Figura 23. Módulo esclavo.

#### **4.4.5. Software esclavo**

El programa principal del dispositivo esclavo permanece en espera de alguna solicitud por parte del coordinador, esto lo hace a través de un programa en la interrupción del puerto RS232, igual que el dispositivo coordinador cuando espera una respuesta. Una vez que una trama de comunicación se confirma, se evalúa la dirección y la función requerida para generar una respuesta; si la dirección a la que se envía una petición no concuerda con la definida en cada esclavo, el sistema permanece en espera de una nueva solicitud. En el caso de una lectura, el microcontrolador se comunica con el HHT02, registra los datos y verifica el CRC para comprobar que no contiene errores, si no fuera así se genera una respuesta de valor nulo y se reinicia todo el sistema. Para la medición de la tensión se efectúan 500 lecturas y se obtiene el promedio para filtrar posibles variaciones en la señal.

Una vez que son hechas todas las mediciones, se realiza una compresión lógica para montarlas en la trama de comunicación y esta se transmite al coordinador, como se muestra en el diagrama de flujo de la siguiente figura.

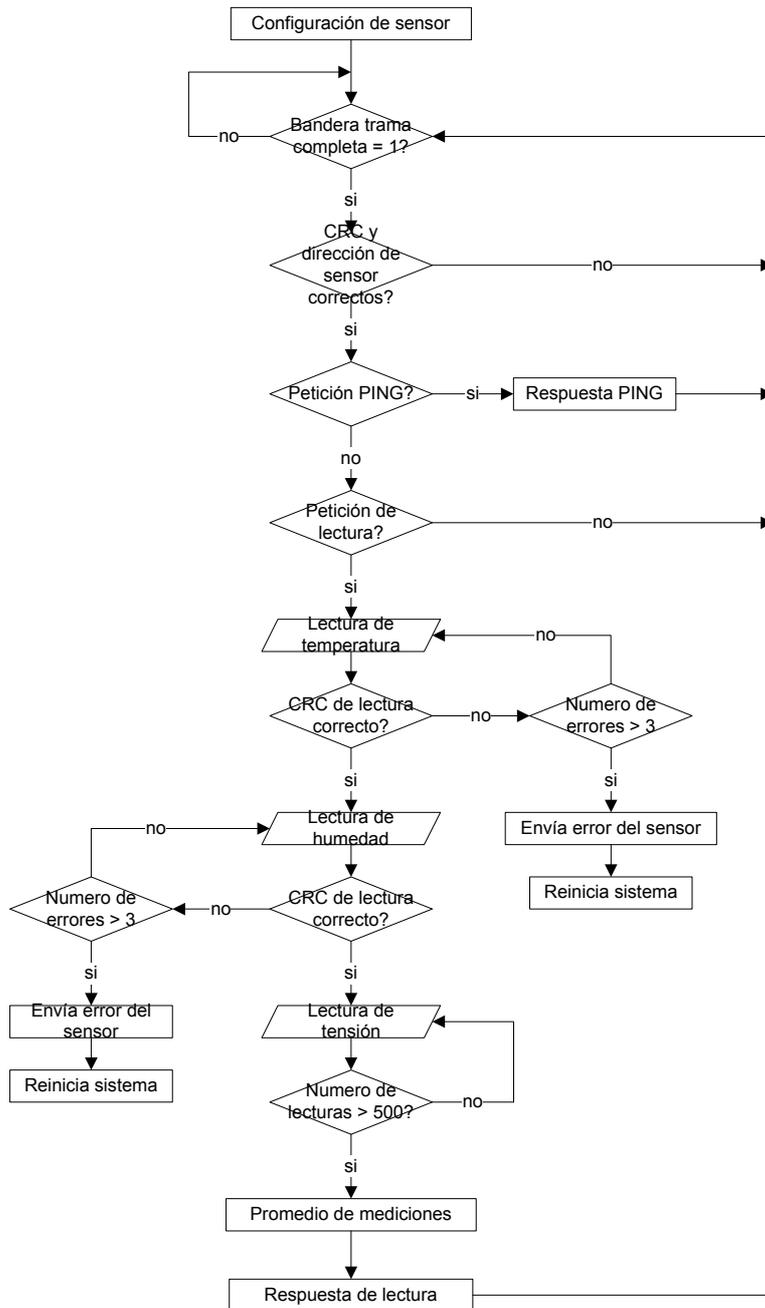


Figura 24. Diagrama de flujo del programa principal en el dispositivo esclavo.

#### 4.4.6. Calibración

Dado que cada HHT02 viene con una calibración de fábrica realizar alguna otra resultaría redundante. En el caso de los tensiómetros se llevo a cabo la correlación del valor mostrado en la pantalla de cada uno y la señal registrada en su respectivo módulo, para variar la tensión se dejo desecar su capsula porosa a temperatura ambiente. En las siguientes graficas se observan los valores obtenidos y la ecuación utilizada para mostrar la lectura en milibares.

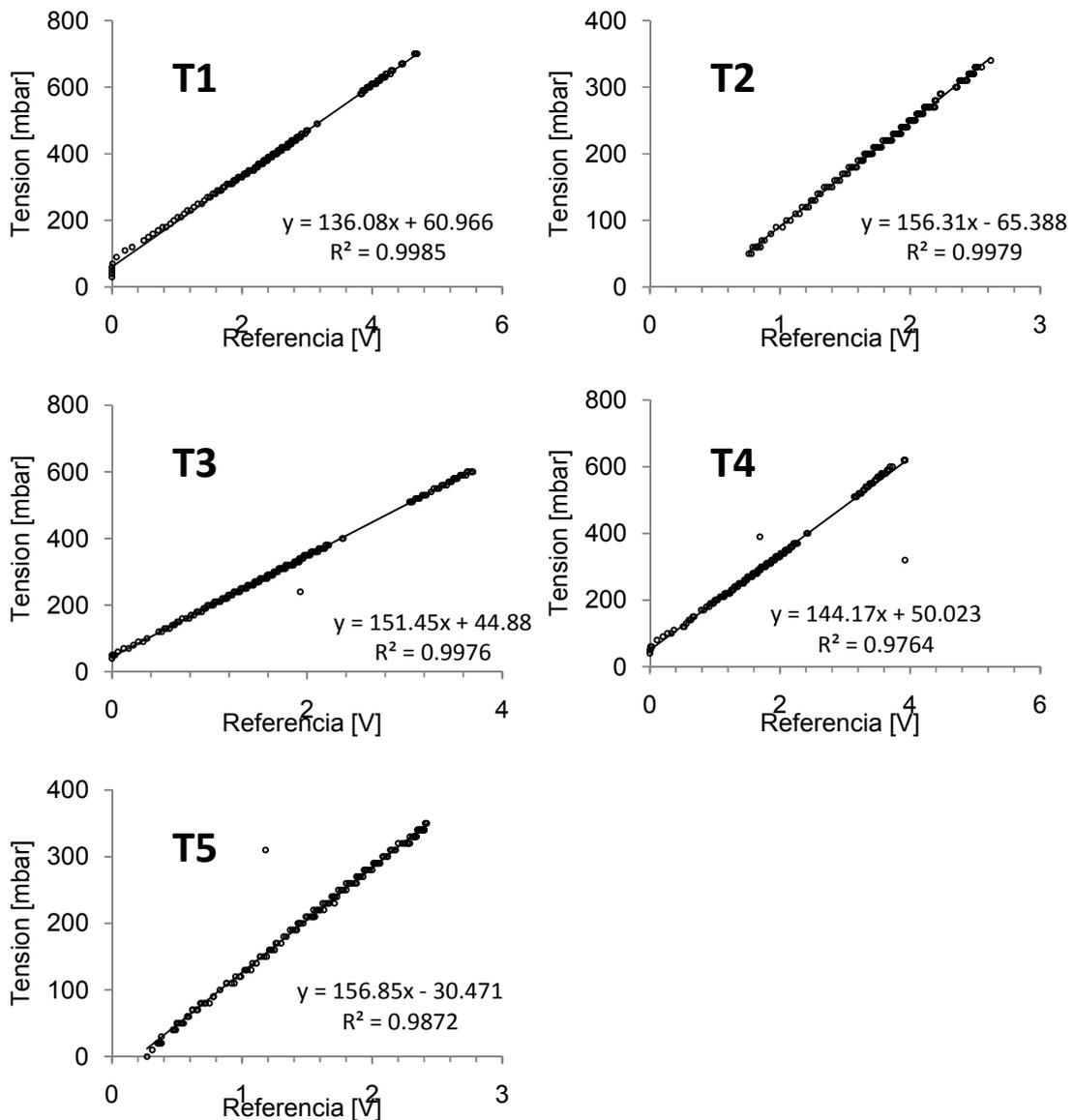


Figura 25. Curvas de calibración en cada uno de los cinco tensiómetros (T).

#### 4.5. Módulos de comunicación inalámbrica

El centro de comunicación, para los dispositivos desarrollados, es un módulo transceptor (dispositivo que realiza funciones tanto de transmisión como de recepción) de la serie LT, marca *LINX technologies* que ofrece transferencia inalámbrica bidireccional de datos en la banda de 433 MHz con una modulación en AM del tipo OOK (*On-off keying*).

Entre sus ventajas se encuentran, entre otras: su largo alcance, bajo costo, bajo consumo de energía y no requerir más dispositivos externos (a excepción de una antena) para funcionar.

#### 4.6. Trama de comunicación

Esta sección se definió usando un protocolo derivado del MODBUS RTU y el MODBUS ASCII, sobre la interfaz de comunicación RS-232 y una codificación Barker de ocho bits, (0x86 = 0, 0x76=1).

Se define el inicio de los mensajes por el carácter ':'=0x3A y el final con los caracteres "\r\n"=0x0D0A (MODBUS ASCII) y el CRC16-CCITT para control de errores de transmisión (MODBUS RTU), con lo que la trama de datos queda integrada de la siguiente manera:

Inicio	Dirección	Función	Dato 0	Dato 1	Dato 2	Dato 3	CRC 1	CRC 2	Fin 1	Fin 2
--------	-----------	---------	--------	--------	--------	--------	-------	-------	-------	-------

Figura 26. Trama de comunicación, cada división representa un byte.

El CRC se calcula con el polinomio 0x1021, según el estándar de la CCITT (Consultative Committee for International Telegraphy and Telephony), ahora renombrada UIT-T (International Telecommunication Union - Telecommunication Standardization Sector).

De acuerdo con la configuración de la trama, un byte guarda la información de la dirección del dispositivo al que corresponde el mensaje transmitido, esto permite tener 256 direcciones posibles de sensores a monitorear; en este trabajo, solo se usaron cinco de ellas para los dispositivos esclavos y una más en el coordinador; el siguiente cuadro muestra las direcciones utilizadas.

Cuadro 4. Códigos de dirección para cada dispositivo.

Dirección	Código
Coordinador	0x30
Esclavo 1	0x31
Esclavo 2	0x32
Esclavo 3	0x33
Esclavo 4	0x34
Esclavo 5	0x35

Al igual que las direcciones, la función está contenida dentro de un byte, abarcando un total de 256 funciones solicitadas a los diferentes dispositivos, siendo de uso solo cuatro de ellas como se muestra en el cuadro siguiente:

Cuadro 5. Códigos de funciones.

Función	Código
Solicitud de eco	0x01
Respuesta de eco	0x02
Solicitud de lectura	0x03
Respuesta de lectura	0x04
Respuesta de error	0x05

La solicitud de eco se utiliza para mostrar si un dispositivo es accesible al sistema sin necesidad de esperar el tiempo de medición.

El siguiente ejemplo muestra la trama generada en una solicitud de lectura (figura 27) enviada al esclavo 1 y la respuesta que éste genera para el coordinador (figura 28).

0x3a	0x31	0x03	0x00	0x00	0x00	0x00	0x84	0xfe	0x0d	0x0a
------	------	------	------	------	------	------	------	------	------	------

Figura 27. Petición de lectura enviada por el coordinador al esclavo 1.

0x3a	0x30	0x04	0x1a	0xa2	0x66	0x90	0x24	0x4f	0x0d	0x0a
------	------	------	------	------	------	------	------	------	------	------

Figura 28. Respuesta de lectura enviada del esclavo 1 al coordinador.

Al llegar al coordinador estos datos son correlacionados para proporcionar los siguientes valores:

Tensión:	0x01aa	343.85 mBar
Temperatura:	0x0026	27.19 °C
Humedad:	0x0690	19.71 %

Ambos dispositivos utilizan la interrupción del puerto RS232 para recibir información, en la figura 29 se observa el diagrama de flujo del programa. Cuando se transmite una trama se antecede de un conjunto de 10 bytes para “limpiar” el canal de posible ruido que modifique el contador de datos entrantes en el receptor y cada byte enviado es acompañado del dato 0x99 que indica su inicio. Cuando un byte es recibido se decodifica y evalúa para reconstruir la trama de comunicación original, al ocurrir esto, una bandera es activada.

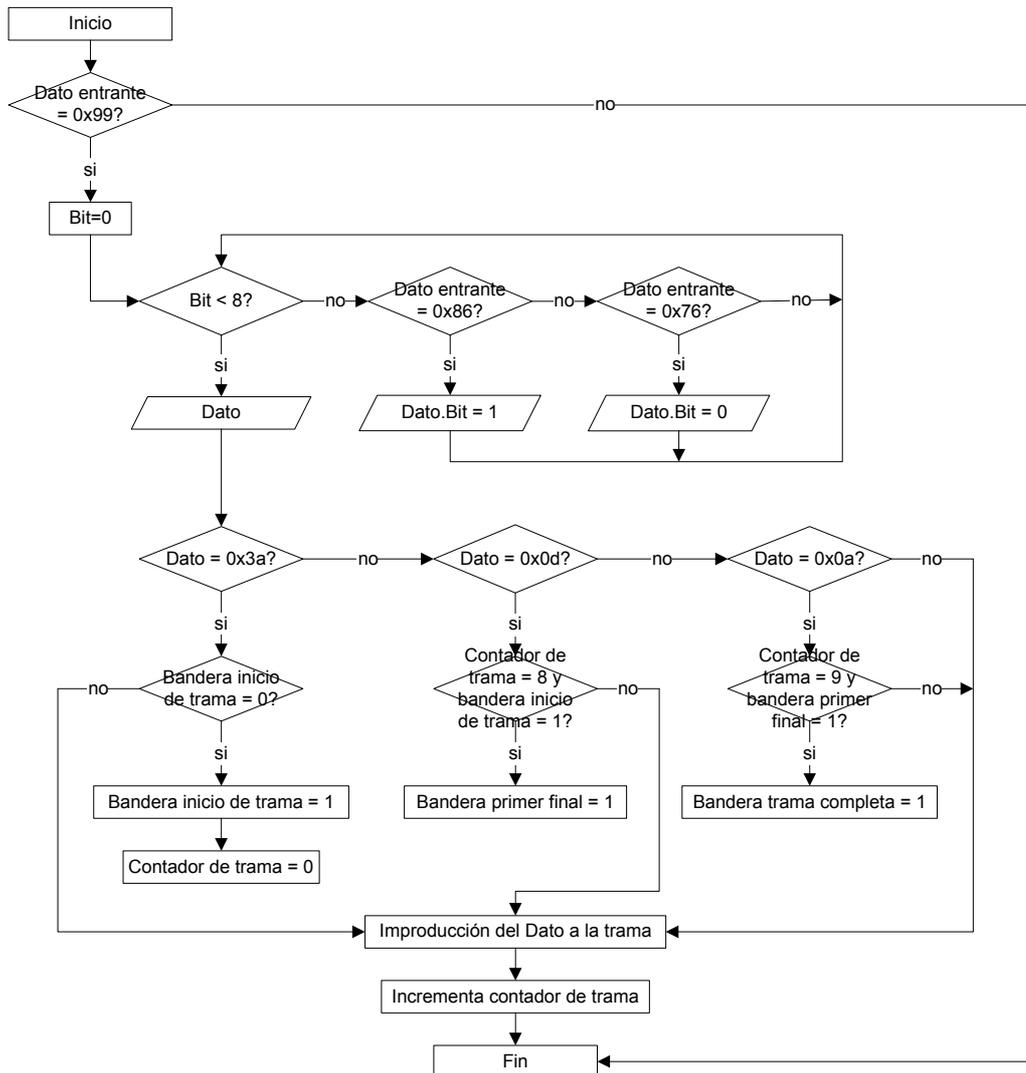


Figura 29. Diagrama de flujo de la interrupción del puerto RS232.

#### 4.7. Fabricación

La fabricación de estos dispositivos se llevó a cabo utilizando como apoyo el software de diseño electrónico OrCAD que permite el dibujo de diagramas esquemáticos y circuitos impresos o PCB (ver anexo).

El PCB es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de *rutas* o *pistas* de material conductor, grabados desde placas de cobre laminadas sobre un sustrato no conductor.

Este grabado se realizó con uso de hojas *press and peel* de transferencia térmica y una impresora laser. El tóner depositado en la hoja de transferencia pasa, mediante la aplicación de calor, a una placa fenólica y queda impreso sobre la cara de cobre, este brinda protección a las pistas contra el ataque químico de percloruro férrico utilizado para disolver el excedente de cobre del PCB, el resultado final, al remover el tóner, es la impresión de nuestro diseño.

Esta técnica, aunque presenta una calidad inferior comparada con la fabricación comercial, resulta adecuada para prototipos de bajo costo.

Una vez terminada la elaboración del PCB se continúa con el montaje de los elementos electrónicos. Los dispositivos se diseñaron para emplear componentes de montaje superficial, y aunque requieren un esfuerzo mayor para su posicionamiento, esta técnica ofrece un espacio menor que el uso de componentes de inserción.

#### **4.8. Programación**

El conjunto de instrucciones que ejecuta un microcontrolador al realizar una tarea determinada se graba dentro de su memoria física, esas instrucciones son interpretadas como niveles de voltaje que se representan, por abstracción, como unos y ceros. Esta forma de representar instrucciones se conoce como *lenguaje maquina* y el uso directo de éste permite la programación de circuitos secuenciales y redes de conmutación.

Sin embargo, dado que las tareas que maneja un microcontrolador son más complejas, el uso de lenguajes de bajo y mediano nivel facilita la programación. Los lenguajes de bajo nivel constituyen la representación más directa del código maquina, sustituyendo palabras (mnemónicos) por códigos de instrucción, este tipo de programación aunque ofrece mayor rapidez en su ejecución y un menor uso de memoria, tiene el inconveniente de ser un lenguaje orientado a la maquina y para su aplicación se requiere el conocimiento de un conjunto de instrucciones propias así como la arquitectura interna del procesador. Los lenguajes de mediano y alto nivel se acercan un poco más al lenguaje humano y contienen instrucciones fáciles de aprender e interpretar. Programas generados con este tipo de lenguaje tienen un mayor uso de memoria aunque las operaciones con números de punto flotante resultan más sencillas.

El programa de los microcontroladores se realizó con un compilador de mediano nivel, el *PCWHD Compiler* (figura 30) de la empresa *Custom Computer Services Inc.* que genera un código máquina a partir del lenguaje C de programación (ver anexo).

```

125 int8 hh,mm,ss;
126 int32 a=0,s=0,div_s=0,res_s=0;
127 set_timer1(3036); //1seg
128 rtc_get_time(hh,mm,ss);
129
130 a=(int32)hh*0x0e10;
131 a+=(int32)mm*0x003c;
132 a+=(int32)ss;
133
134 s=(int32)read_eeeprom(mem_hrs)*0x0e10;
135 s+=(int32)read_eeeprom(mem_min)*0x003c;
136 s+=(int32)read_eeeprom(mem_seg);
137
138 div_s = a/s;
139 res_s = a - (div_s * s);
140 if(res_s==0) flag_t=1;
141
142 if(flag_ini==0) flag_enc=1;
143
144 flag_tt=1;
145
146
147
148 ----- timer 1
149 #int_timer1
150 void timer1(void){
151   set_timer1(3036); //250ms
152   restart_wdt();
153   flag_led=flag_led;
154   bit_test(flag_led,0)?output_high(tr_led):output_low(tr_led);
155   output_high(wdt_ext);
156   delay_us(1);
157   output_low(wdt_ext);
158 }

```

Figura 30. Compilador PCWHD.

Posteriormente el programa fue grabado en la memoria física de cada microprocesador por medio del programa *PICKit 2 v2.55* de la empresa *Microchip* y una interfaz compatible con éste mostrados en la siguiente figura.

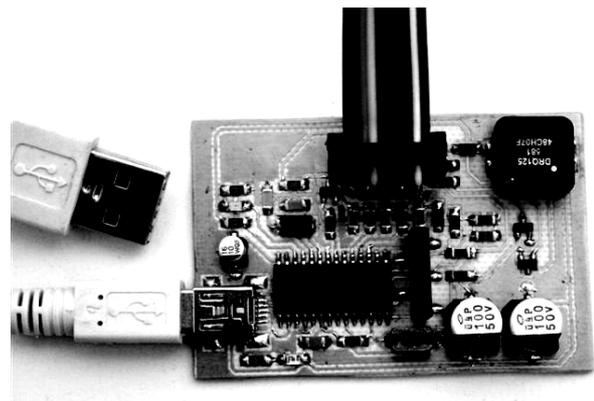
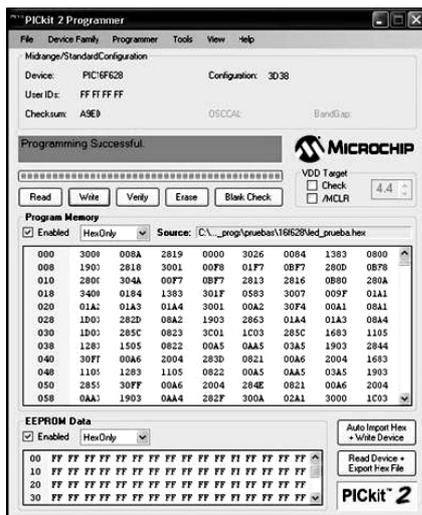


Figura 31. Software de programación PICKit e Interfaz de grabación.

## 5. Resultados y discusión

### 5.1. Interfaz.

Este sistema se desarrolló con el propósito de comunicarse con una computadora personal; al no contar con un programa de cómputo propio para esto, la transferencia de datos se realiza por medio del puerto RS232 y la *HyperTerminal* dentro de los programas accesorios de *Windows* o algún otro programa similar.

Para entablar una comunicación adecuada se tiene que crear una conexión nueva dentro de la *HyperTerminal* eligiendo el puerto que se usará como se muestra en la siguiente figura; en este ejemplo, se llamó a la conexión “COM”.



Figura 32. Elección del puerto de comunicación en la *HyperTerminal*.

En el siguiente cuadro de dialogo se configura la velocidad, tamaño de los datos y de flujo del puerto, los cuales deben ser (9600, 8, N, 1, N) como los que se muestran en la figura 33.



Figura 33. Configuración del puerto en la *HyperTerminal*.

En la figura 34 se observa la operación normal del sistema cuando transmite datos actuales de tensión [mBar], humedad [%] y temperatura [°C] en cada intervalo de tiempo que es configurado por el dispositivo coordinador.

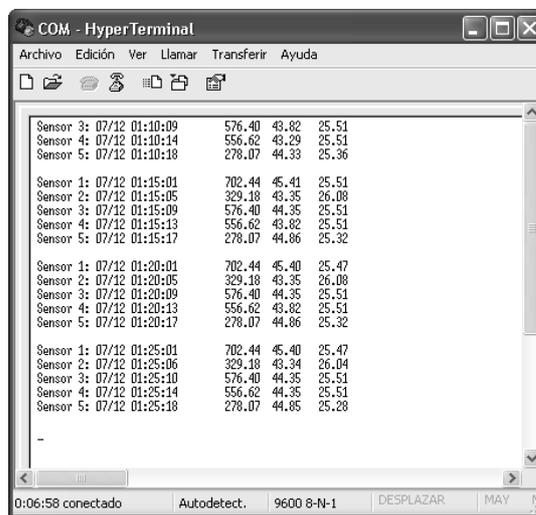


Figura 34. Comunicación en tiempo real por medio de la *Hyperterminal*.

## 5.2. Descarga de datos.

Al descargar los datos se crea un archivo de texto como se muestra en la figura siguiente, después de seleccionar “Capturar texto...” se elige el nombre y ubicación del archivo; por otro lado se configura el dispositivo coordinador en “Menu-> Memoria -> Leer -> Bloque [1-5]” lo cual iniciará la descarga del bloque seleccionado. Este archivo puede ser procesado como uno de texto delimitado por tabulaciones en el programa *Office Excel* o algún otro similar (figura 36).

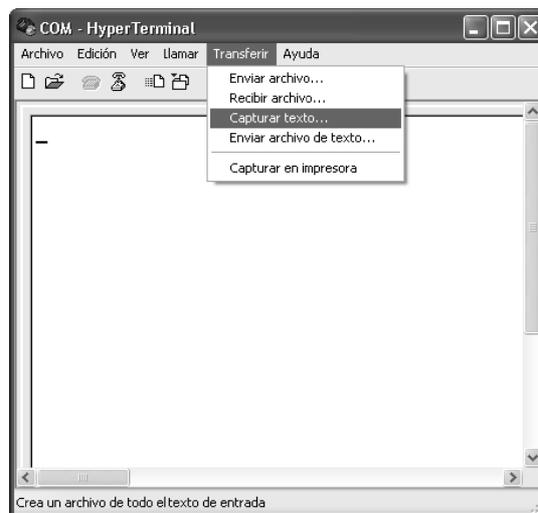


Figura 35. Selección de captura de texto en la *HyperTerminal*.

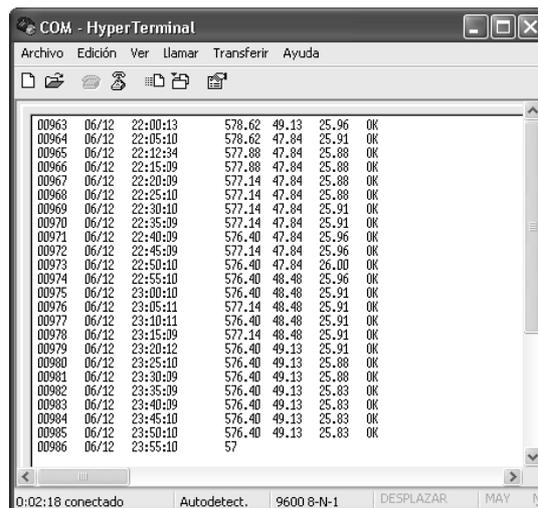


Figura 36. Descarga de datos en la *HyperTerminal*.

### 5.3. Presentación de datos.

Las siguientes figuras muestran los datos de humedad relativa (figura 37) y temperatura del aire (figura 38) obtenidos por uno de los sensores HHT02 en comparación con los registrados en la estación agrometeorológica del Postgrado en Hidrociencias del Campus Montecillo del 16 al 18 de julio de 2009, a los cuales se les puede acceder por medio de la dirección electrónica: <http://www.cm.colpos.mx/meteoro/> con el propósito de evaluar el correcto funcionamiento de dichos sensores.

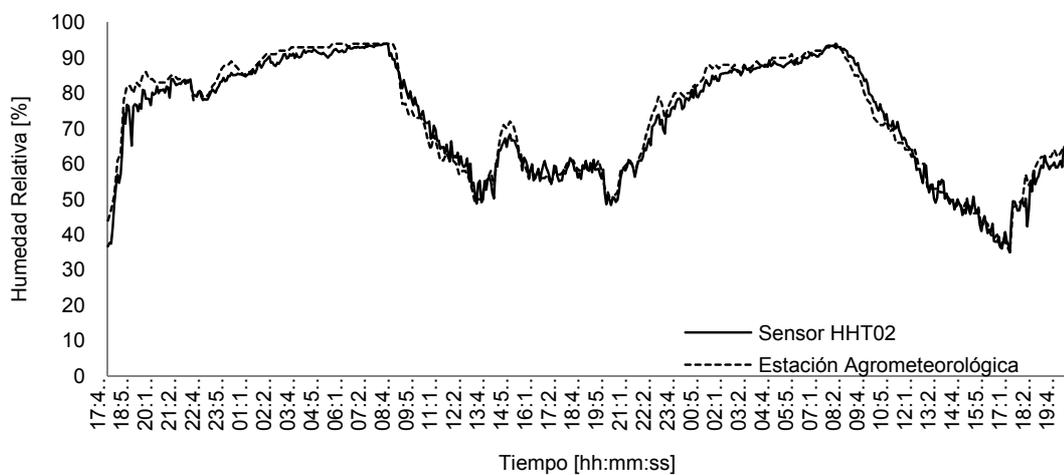


Figura 37. Comparación de lecturas de humedad relativa entre el sensor HHT02 y la estación agrometeorológica de Montecillo

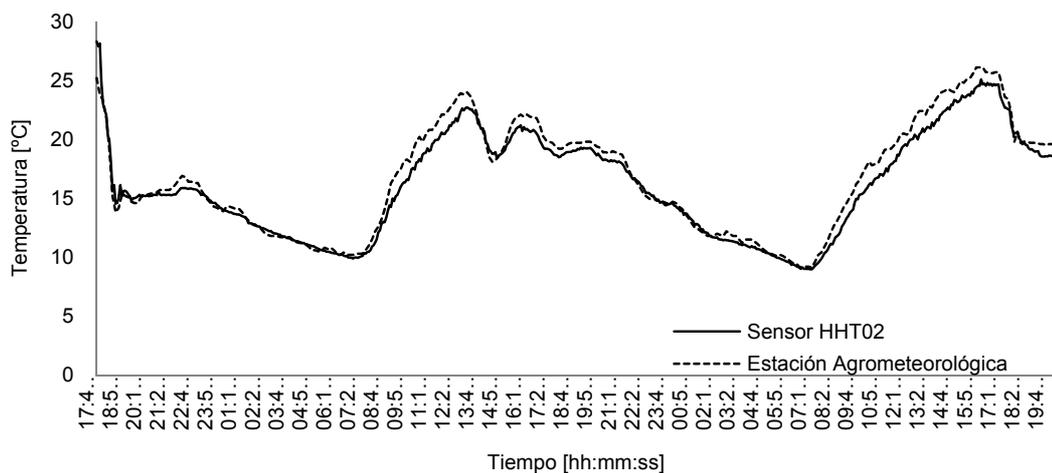


Figura 38. Comparación de lecturas de temperatura entre el sensor HHT02 y la estación agrometeorológica de Montecillo

Ambas gráficas poseen una correlación del 97% entre las lecturas tomadas por el sensor HHT02 y la estación agrometeorológica del Postgrado en Hidrociencias del Colegio de Postgraduados, Campus Montecillo, Edo. de Mexico.

Las figuras siguientes muestran los datos de humedad relativa (figura 39), temperatura (figura 40) y tensión (figura 41) obtenidos en el dispositivo esclavo número dos del día 7 al 14 de diciembre del 2009 en la etapa de muestreo.

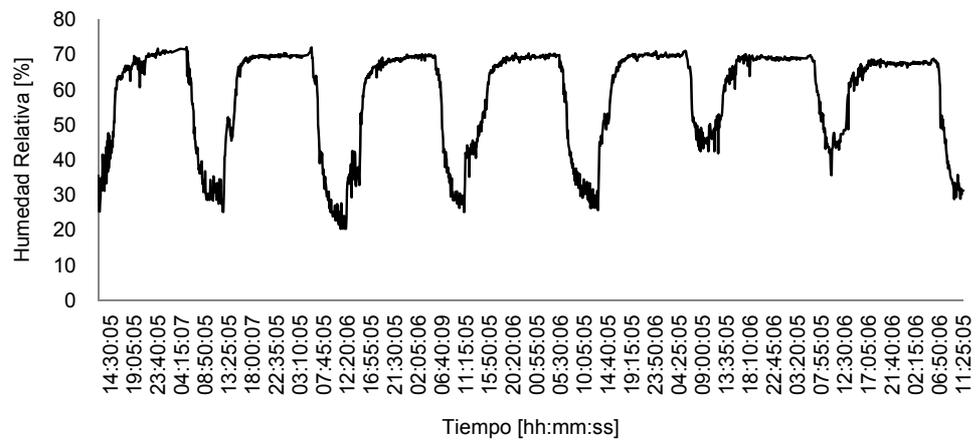


Figura 39. Muestreo de humedad relativa, dispositivo esclavo número dos.

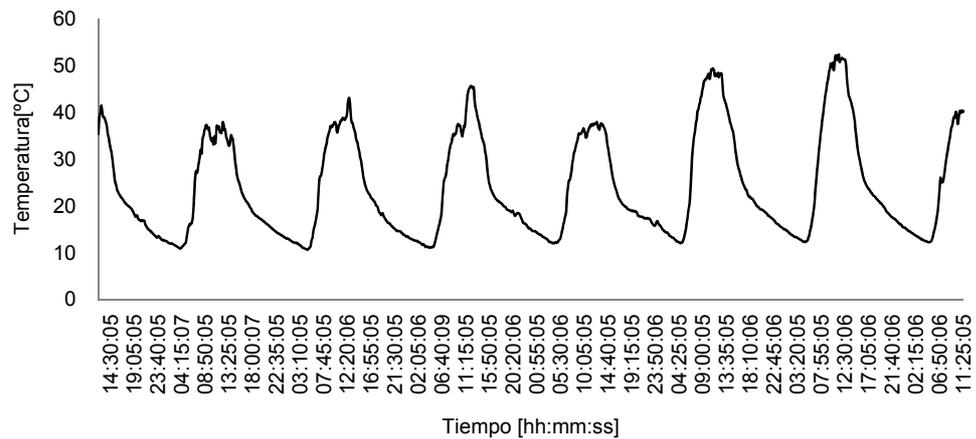


Figura 40. Muestreo de temperatura, dispositivo esclavo número dos.

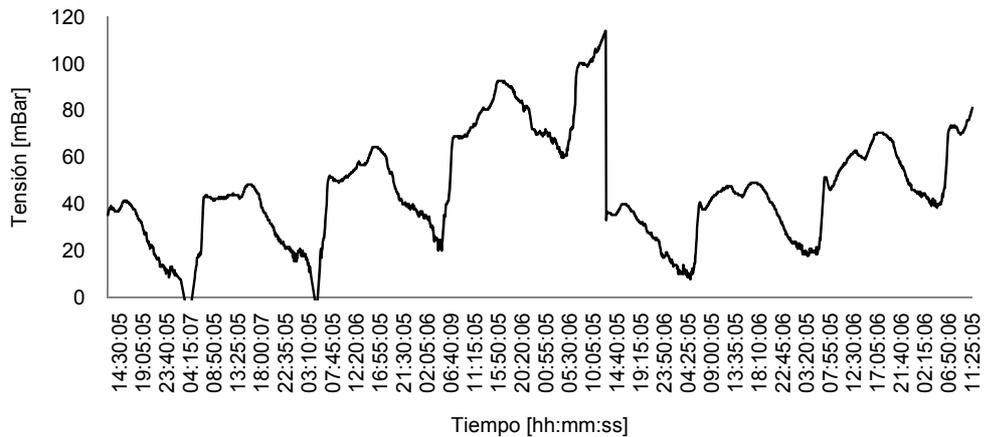


Figura 41. Muestreo de tensión, dispositivo esclavo número dos.

En la grafica de tensión podemos observar fluctuaciones producidas por la hidratación del sustrato que se refleja en una reducción de tensión en el instrumento, esta hidratación aumenta durante la noche y muestra una relación directa, al no llevarse a cabo el riego, con la humedad relativa del ambiente; sin embargo la tendencia general en esto se incrementa indicando la pérdida de humedad en horario diurno. Otra situación que corrobora la observación anterior son las variaciones en las crestas de la grafica de tensión, más sutiles en las de humedad relativa y temperatura, las cuales indican la reducción de tensión alrededor de las 9:00 horas (apertura de las ventanas del invernadero) y de nuevo a las 15:00 horas (cierre de las ventanas), horario en el cual se llevan a cabo estas operaciones manualmente y se realizan en presencia de radiación solar la cual aumenta la temperatura e influye en la humedad relativa como se vió en la revisión de literatura.

Al tomar en cuenta el riego diario, en el manejo de este invernadero, que se realiza alrededor de las 9:00 horas; el agua se derivó de la maceta a un depósito independiente, para luego aplicarla de forma manual y en su totalidad en sus respectivas macetas el 11 de diciembre como lo muestra la gráfica.

#### 5.4. Alcance

De acuerdo con la fórmula para calcular la zona de Fresnel aplicándola en nuestro sistema los transmisores se mantendrían libres de obstrucciones a una altura de:

$$r = 17.32 \sqrt{\frac{D}{4f}}$$

$$r = 17.32 \sqrt{\frac{1}{4*0.433}}$$

$$r = 13.16 \text{ m}$$

Sin embargo, en la práctica se colocaron a unos cuantos centímetros del suelo y cerca del cultivo, en estas condiciones el alcance máximo fue 170 m.

## **6. Conclusiones y recomendaciones**

### **6.1. Conclusiones**

El hardware y software descritos en la parte de materiales y métodos cumplió fielmente con las tareas para las que fueron diseñados.

El sistema desarrollado hizo posible el monitoreo de temperatura y humedad relativa por medio del sensor HHT02, el cual mostró un funcionamiento adecuado en esta aplicación tomando lecturas de buena calidad y una correcta comunicación.

En relación a cinco sensores de humedad en el sustrato al enlazarlos al sistema, uno de ellos funciono correctamente mientras que los cuatro restantes, en ocasiones, presentaron problemas al interferir con los módulos de comunicación.

El modo de transmisión bidireccional permitió el enlace y control inalámbrico a través de la trama de comunicación, la cual funcionó adecuadamente sin presentar ningún problema en su procesamiento y verificación. Sin embargo, los módulos de comunicación en modo receptor presentaron problemas de ruido eléctrico en los circuitos o de alguna otra fuente transmisora ocasionando que se captaran datos de manera continua sin que se supiera su procedencia. Este problema fue resuelto por medio de la codificación Barker de ocho bits de los datos, a causa de esto, la velocidad de comunicación bajó de 9600 a 1200 bits/s, aun así, ésto no fue una limitante en el tiempo de lectura requerido por los sensores, pero ocasionó que los módulos siempre se mantuvieran encendidos con lo que su consumo de energía se incrementa.

Los algoritmos de compresión en la etapa de memoria permitieron un manejo eficiente del espacio destinado al almacenamiento de datos.

## **6.2. Recomendaciones**

Aumentar la velocidad de transmisión en comunicación, así como modificar el tipo de petición y orden para evitar demoras innecesarias.

Utilizar módulos de comunicación que permitan un bajo consumo de energía durante la espera de una petición por parte del coordinador para optimizar su funcionamiento.

Combinar fuentes de alimentación de la línea de 110v con baterías de respaldo para evitar problemas con fallas en el suministro de energía eléctrica.

Agregar subrutinas en el software que permitan a los módulos comportarse como repetidores de señal y aumentar el alcance de la transmisión, para optimizar su uso.

Cambiar la pantalla LCD del dispositivo coordinador a una de tipo gráfico que permita la visualización directa en las graficas de cada sensor, así como desarrollar un menú más amigable para facilitar la interacción con el usuario.

Cambiar la cubierta plástica de los módulos por una que permita la ventilación natural de los sensores y con ésto eliminar los ventiladores que se agregaron para evitar tener lecturas alteradas por el calentamiento.

Delegar la elaboración de las tarjetas electrónicas a un fabricante especializado para lograr una mayor calidad.

## 7. Literatura citada.

- Albright, L. D. 2002. *Controlling greenhouse environments*. Acta Horticulturae 578: 47-60.
- Almanza, M. y P. García 1997. *La electrónica en la agricultura*. Universidad Autónoma Chapingo. Texcoco, estado de México. 97pp.
- Anttalainen, T. 2003. *Introduction to Telecommunications Network Engineering*. Artech House, Inc. USA. 377pp.
- Baille, A. 2001. *Trends in greenhouse technology for improved climate control in mild winter climates*. Acta Horticulturae 559: 161-167
- Bastida T., A. y J. A. Ramírez 1999. *Invernaderos en México. Diseño, construcción y manejo*. Universidad Autónoma Chapingo. Departamento de Preparatoria Agrícola. Texcoco, estado de México. 105 pp.
- Challa H, 2002. *Crop models for greenhouse production systems*. Acta Horticulturae 593: 47-53.
- Coulon, G. 1998. *Sensors for horticultural climate control computers*. Acta Horticulturae 421: 285.
- Creus S,A. 1998. *Instrumentacion industrial*. Alfaomega. México. 644 pp.
- Field , H. L. y Soile, J.B. 2007. *Introduction to Agricultural Engineering Technology A Problem Solving Approach*, Third Edition, Oklahoma State University, Stillwater, OK, USA. 389 pp.
- Fernández, M. D., S. Bonachela, M. Gallard, F. Orgaz, y E. Fereres. 2001. *Irrigation scheduling for plastic greenhouse melon crops based on historical reference evapotranspiration data*. Acta Horticulturae 559: 383-389.
- García M., E. 1987. *Modificaciones al Sistema de Clasificación Climática de Köppen*. Universidad Nacional Autónoma de México. México. 245 pp.
- Gates, R. S., K. Chao y N. Sigrimis. 2001. *Identifying design parameters for fuzzy control of staged ventilation control systems*. Computers and Electronics in Agriculture 31: 61-74.
- Hansman, R.J.Jr. 1999. *Characteristics of Instrumentation*. En: *Measurement, Instrumentation, and Sensors Handbook CRCnetBASE 1999*. CRC Press LLC. USA. 1.1 – 1.4
- Helfrick, A.D. y Cooper, W.D. 1991. *Instrumentación Electrónica Moderna y Técnicas de Medición*. Person Educación. México. 450 pp.
- Herrera P., E. 2004. *Comunicaciones I Señales, Modulación y Transmisión*. Limusa. México. 281 pp.
- Hristov, H. D. 2000. *Fresnel Zones in Wireless Links, Zone Plate Lenses and Antennas*. Artech House Publishers. Norwood , MA. 323pp.
- Jiménez M., O. 1999. *Invernaderos (Diseño, establecimiento y manejo)*. *Antología*. SEP/SEIT/GTA. México. 203 pp.
- Jones J. W., E. Dayan, H.L. Allen, Van Keulen, H. Challa. 1991. *A dynamic tomato growth and yield model (TOMGRO)*. Transactions of the ASAE 32: 663-672.
- Kassler, M. 2001. *Agricultural automation in the new millenium*. Computers and Electronics in Agriculture 30: 237-240.
- Kempkes, F.L.K, Bakker y N.J.van de Braak. 1998. *Control and modelling of vertical temperature distribution in greenhouse crops*. Acta Horticulturae 456: 363.

- Lathi, B.P. 2001. *Introducción a la teoría y sistemas de comunicación*. Limusa. México. 409 pp.
- Moshou, D., Vrindts, E., De Ketelaere, B., De Baerdemaeker, J. y Ramon, H. 2001. *A neural network based plant classifier*. Computers and Electronics in Agriculture 31: 5-16.
- Morales P., J.T., I. Miranda V., I., I. Gil V., I., A. Bastida T., J. Hernández O., D.S. Reyes R., D. S., G. Flores E. y E. Navarro L. 2005. *Introducción a la hidroponía*. Universidad Autónoma Chapingo. Departamento de Preparatoria Agrícola. Texcoco, estado de México. 201 pp.
- Muñoz R., J.J. 2003. *El cabezal de riego*. En: Muñoz R.,J.J y J. Z. Castellanos. *Manual de producción hortícola en invernadero*. INCAPA. México. 74-85.
- Ogata, K. 1998, ingeniería de control moderna. Pearson Educación. México. 997
- Oppenheim, A.V., A.S. Willsky y S.H. Nawab. 1998. *Señales y sistemas*. Person Education. Mexico. 941 pp.
- Pallás A.,R. 1993. *Adquisición y distribución de señales*. Marcombo. España. 421 pp.
- Pallás A.,R. 2003. *Sensores y acondicionadores de señal*. Marcombo. España. 480 pp.
- Prenger, J. J., R.P. Fynn y R.C. Hansen. 2002. *A comparison of four Evapotranspiration models in a greenhouse environment*. Transactions of the ASAE 45: 1779-1788.
- Rodríguez D., F. y M. Berenguer S. 2004. *Control y robótica en la agricultura*. Universidad de Almería. Almería, España. 433pp.
- Schmidt, U. 1998. *Low-cost system for on-line measurement of plant transpiration and photosynthesis in greenhouse production*. Acta Horticulturae 421: 249-253
- Schmidt, U. 2004. *Decision support for greenhouse climate control using a computerised Mollier diagram*. Acta Horticulturae 654: 187-202.
- Serrano C., Z. 2002. *Construcción de invernaderos*. Mundi Prensa. España.
- Sigrimis, N. A. y N. Rerras. 1998. *Enhancement of the intelligent leaf sensor*. Acta Horticulturae 421: 231-239.
- Smajstrla, A.G y D.J. Pitts. 1997. Bulletin 319, Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida. 5 pp.
- Stojmenović , I. 2005. *Handbook of sensor networks Algorithms and Architectures*. Wiley & Sons, Inc. USA. 532 pp.
- Tijerina Ch., L. 1992. *Uso eficiente del agua en unidades de riego para el desarrollo rural. Diplomado*. Colegio de Postgraduados. Montecillo, México.
- Tomasi, W. 2003. *Sistemas de comunicaciones electrónicas*. Pearson Educación. México. 841 pp.
- Van de Sanden, P. y B. Veen. 1992. *Effects of air humidity and nutrient solution concentration on growth, water potential and stomatal conductance of cucumber seedlings*. Scientia Horticulturae 50: 173-186.
- Van Straten, G., H. Challa y F. Buwalda. 2000. *Towards user accepted optimal control of greenhouse climate*. Computers and Electronics in Agriculture 26: 221-238.

Villalobos R., S., J.Z. Castellanos y J.L Ojodeagua. 2003. *Manejo del riego en el invernadero*. En: Muñoz R.,J.J y J. Z. Castellanos. *Manual de producción hortícola en invernadero*. INCAPA. México. 86-108

Zolnier, S., G.B. Lyra, y R.S. Gates. 2004. *Evapotranspiration estimates form greenhouse lettuce using an intermittent nutrient film technique*. Transactions of the ASAE 47: 271-282.

## 8. Anexos.

### 8.1. CRC-8

Cuadro 6. Valores de CRC-8 calculados usando el polinomio  $X^8 + x^5 + x^4$

<b>00</b>	<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>	<b>06</b>	<b>07</b>	<b>08</b>	<b>09</b>	<b>0A</b>	<b>0B</b>	<b>0C</b>	<b>0D</b>	<b>0E</b>	<b>0F</b>
00	31	62	53	C4	F5	A6	97	B9	88	DB	EA	7D	4C	1F	2E
<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>1A</b>	<b>1B</b>	<b>1C</b>	<b>1D</b>	<b>1E</b>	<b>1F</b>
43	72	21	10	87	B6	E5	D4	FA	CB	98	A9	3E	0F	5C	6D
<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>2A</b>	<b>2B</b>	<b>2C</b>	<b>2D</b>	<b>2E</b>	<b>2F</b>
86	B7	E4	D5	42	73	20	11	3F	0E	5D	6C	FB	CA	99	A8
<b>30</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>3A</b>	<b>3B</b>	<b>3C</b>	<b>3D</b>	<b>3E</b>	<b>3F</b>
C5	F4	A7	96	01	30	63	52	7C	4D	1E	2F	B8	89	DA	EB
<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>4A</b>	<b>4B</b>	<b>4C</b>	<b>4D</b>	<b>4E</b>	<b>4F</b>
3D	0C	5F	6E	F9	C8	9B	AA	84	B5	E6	D7	40	71	22	13
<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>5A</b>	<b>5B</b>	<b>5C</b>	<b>5D</b>	<b>5E</b>	<b>5F</b>
7E	4F	1C	2D	BA	8B	D8	E9	C7	F6	A5	94	03	32	61	50
<b>60</b>	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>	<b>66</b>	<b>67</b>	<b>68</b>	<b>69</b>	<b>6A</b>	<b>6B</b>	<b>6C</b>	<b>6D</b>	<b>6E</b>	<b>6F</b>
BB	8A	D9	E8	7F	4E	1D	2C	02	33	60	51	C6	F7	A4	95
<b>70</b>	<b>71</b>	<b>72</b>	<b>73</b>	<b>74</b>	<b>75</b>	<b>76</b>	<b>77</b>	<b>78</b>	<b>79</b>	<b>7A</b>	<b>7B</b>	<b>7C</b>	<b>7D</b>	<b>7E</b>	<b>7F</b>
F8	C9	9A	AB	3C	0D	5E	6F	41	70	23	12	85	B4	E7	D6
<b>80</b>	<b>81</b>	<b>82</b>	<b>83</b>	<b>84</b>	<b>85</b>	<b>86</b>	<b>87</b>	<b>88</b>	<b>89</b>	<b>8A</b>	<b>8B</b>	<b>8C</b>	<b>8D</b>	<b>8E</b>	<b>8F</b>
7A	4B	18	29	BE	8F	DC	ED	C3	F2	A1	90	07	36	65	54
<b>90</b>	<b>91</b>	<b>92</b>	<b>93</b>	<b>94</b>	<b>95</b>	<b>96</b>	<b>97</b>	<b>98</b>	<b>99</b>	<b>9A</b>	<b>9B</b>	<b>9C</b>	<b>9D</b>	<b>9E</b>	<b>9F</b>
39	08	5B	6A	FD	CC	9F	AE	80	B1	E2	D3	44	75	26	17
<b>A0</b>	<b>A1</b>	<b>A2</b>	<b>A3</b>	<b>A4</b>	<b>A5</b>	<b>A6</b>	<b>A7</b>	<b>A8</b>	<b>A9</b>	<b>AA</b>	<b>AB</b>	<b>AC</b>	<b>AD</b>	<b>AE</b>	<b>AF</b>
FC	CD	9E	AF	38	09	5A	6B	45	74	27	16	81	B0	E3	D2
<b>B0</b>	<b>B1</b>	<b>B2</b>	<b>B3</b>	<b>B4</b>	<b>B5</b>	<b>B6</b>	<b>B7</b>	<b>B8</b>	<b>B9</b>	<b>BA</b>	<b>BB</b>	<b>BC</b>	<b>BD</b>	<b>BE</b>	<b>BF</b>
BF	8E	DD	EC	7B	4A	19	28	06	37	64	55	C2	F3	A0	91
<b>C0</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>	<b>C9</b>	<b>CA</b>	<b>CB</b>	<b>CC</b>	<b>CD</b>	<b>CE</b>	<b>CF</b>
47	76	25	14	83	B2	E1	D0	FE	CF	9C	AD	3A	0B	58	69
<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>D5</b>	<b>D6</b>	<b>D7</b>	<b>D8</b>	<b>D9</b>	<b>DA</b>	<b>DB</b>	<b>DC</b>	<b>DD</b>	<b>DE</b>	<b>DF</b>
04	35	66	57	C0	F1	A2	93	BD	8C	DF	EE	79	48	1B	2A
<b>E0</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>E6</b>	<b>E7</b>	<b>E8</b>	<b>E9</b>	<b>EA</b>	<b>EB</b>	<b>EC</b>	<b>ED</b>	<b>EE</b>	<b>EF</b>
C1	F0	A3	92	05	34	67	56	78	49	1A	2B	BC	8D	DE	EF
<b>F0</b>	<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>	<b>F5</b>	<b>F6</b>	<b>F7</b>	<b>F8</b>	<b>F9</b>	<b>FA</b>	<b>FB</b>	<b>FC</b>	<b>FD</b>	<b>FE</b>	<b>FF</b>
82	B3	E0	D1	46	77	24	15	3B	0A	59	68	FF	CE	9D	AC

## 8.2. Listas de materiales

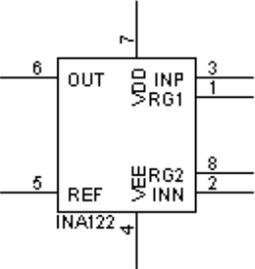
Componentes utilizados en el dispositivo coordinador:

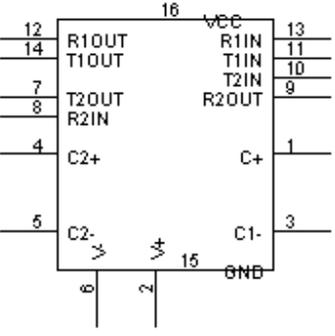
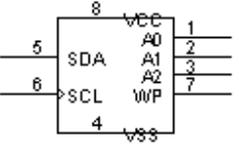
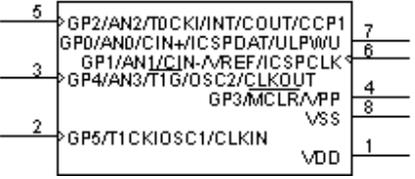
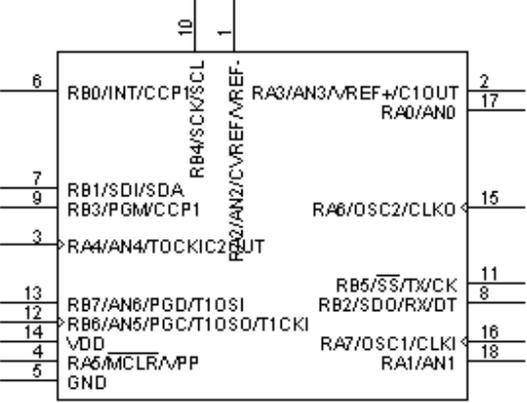
1	Antena
1	Batería CR2032
	Cable plano y conectores de diez vías
5	Capacitor 1 $\mu$ F
5	Capacitor 0.1 $\mu$ F
1	Conector Alimentación
1	Conector Antena
1	Conector DB9
1	Conector LCD
2	Conector Programación
1	Conector RS232
1	Conector Teclado
1	Cristal 32,768KHz
2	Diodo Led
1	Diodo 1N4001
1	Gabinete SERPAC A-42
1	Interfaz MAX232
3	Memoria 24LC1025
1	Microcontrolador PIC12F683
1	Microcontrolador PIC18F4680
1	Módulo inalámbrico TRM-433-LT
1	Oscilador 8Mhz
1	Pantalla LCD de 16 columnas por 4 renglones
1	Placa fenólica de 10x15cm
1	Teclado matricial de 4 filas por 4 columnas
6	Transistor Q2N3904
36	Resistencia 10k $\Omega$
4	Resistencia 1k $\Omega$
1	Resistencia 10 $\Omega$
25	Resistencia 0
1	Reloj/Calendario DS1302
1	Regulador LM7805
1	Regulador MCP1825
1	Selector 74LS126

Componentes utilizados en el dispositivo esclavo:

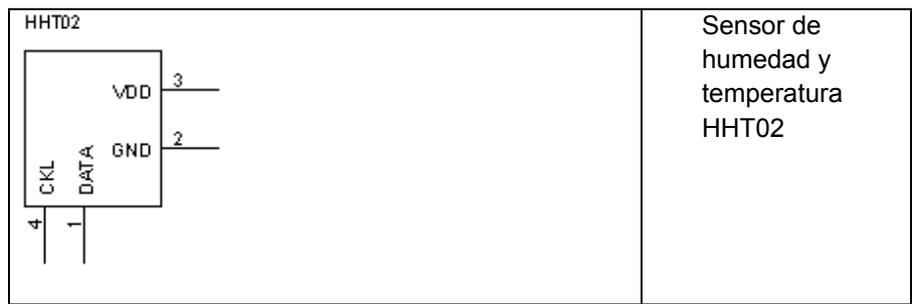
1	Amplificador INA122
1	Antena
5	Capacitor 0.1 $\mu$ F
2	Conector Alimentación
1	Conector de Programación
1	Conector Tensiómetro
2	Diodo 1N4001
1	Diodo Led
1	Gabinete SERPAC A-20
1	Microcontrolador PIC16F88
1	Modulo inalámbrico TRM-433-LT
6	Transistor 2N3904
1	Transistor 2N3906
1	Placa fenolica de 5x10 cm
1	Regulador LM7805
1	Regulador MCP1825
5	Resistencia 0 $\Omega$
1	Resistencia 10 $\Omega$
2	Resistencia 1 k $\Omega$
18	Resistencia 10 k $\Omega$
1	Sensor HHT02

### 8.3. Simbología Electrónica.

	Antena
	Batería
	Capacitor
	Cristal
	Diodo
	Diodo emisor de luz (LED)
	Transistor NPN
	Transistor PNP
	Resistencia
	Amplificador INA122

 <p>MAX232</p> <p>12 R1OUT 14 T1OUT 7 T2OUT 8 R2IN 4 C2+ 5 C2- 16 VCC 18 VCC 13 R1IN 11 T1IN 10 T2IN 9 R2OUT 1 C+ 3 C- 15 GND 1 2</p>	<p>Interfaz RS232</p>
 <p>24LC1025</p> <p>8 VCC 5 SDA 6 SCL 4 VSS 1 A0 2 A1 3 A2 7 W/P</p>	<p>Memoria EEPROM</p>
<p>PIC12F683/SO</p>  <p>5 GP2/AN2/T0CKI/INT/COUT/CCP1 GP0/AND/CIN+/ICSPDAT/ULPWU 3 GP1/AN1/CIN-/VREF/ICSPCLK GP4/AN3/T1G/OSC2/CLKOUT 2 GP5/T1CKI/OSC1/CLKIN 7 6 4 8 1 VSS VDD</p>	<p>Microcontrolador PIC12F683</p>
 <p>PIC16F88</p> <p>6 RB0/INT/CCP1/SCL 7 RB1/SDI/SDA 9 RB3/PGM/CCP1 3 RA4/AN4/T0CKI/C2OUT 13 RB7/AN6/PGD/T1OSI 12 RB6/AN5/PGC/T1OSO/T1CKI 4 VDD 5 RA6/MCLR/VPP GND 10 RB4/SCK/SCL 1 RB2/AN2/CVREF/VREF- 2 RA3/AN3/VREF+/C1OUT 17 RA0/AN0 15 RA6/OSC2/CLKO 11 RB5/SS/TX/CK 8 RB2/SDO/RX/DT 16 RA7/OSC1/CLKI 18 RA1/AN1</p>	<p>Microcontrolador PIC16F88</p>

	<p>Microcontrolador PIC18F4680</p>
	<p>Modulo de comunicación inalámbrica LINX</p>
	<p>Regulador de voltaje</p>
	<p>Reloj/calendario</p>
	<p>Selector 74LS126</p>



## 8.4. Diagramas eléctricos

### 8.4.1. Dispositivo coordinador

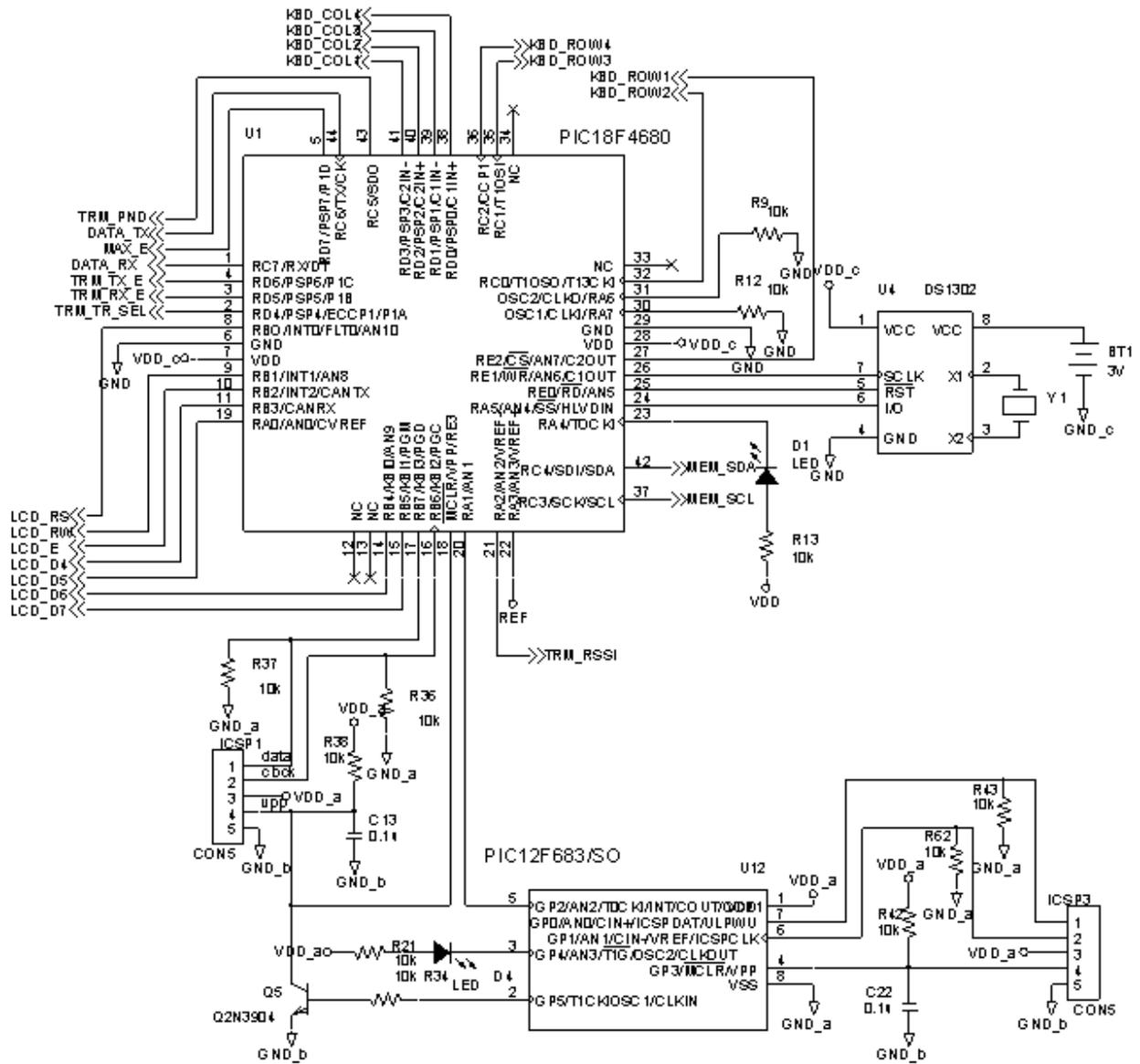


Figura 42. Diagrama eléctrico de microcontroladores.

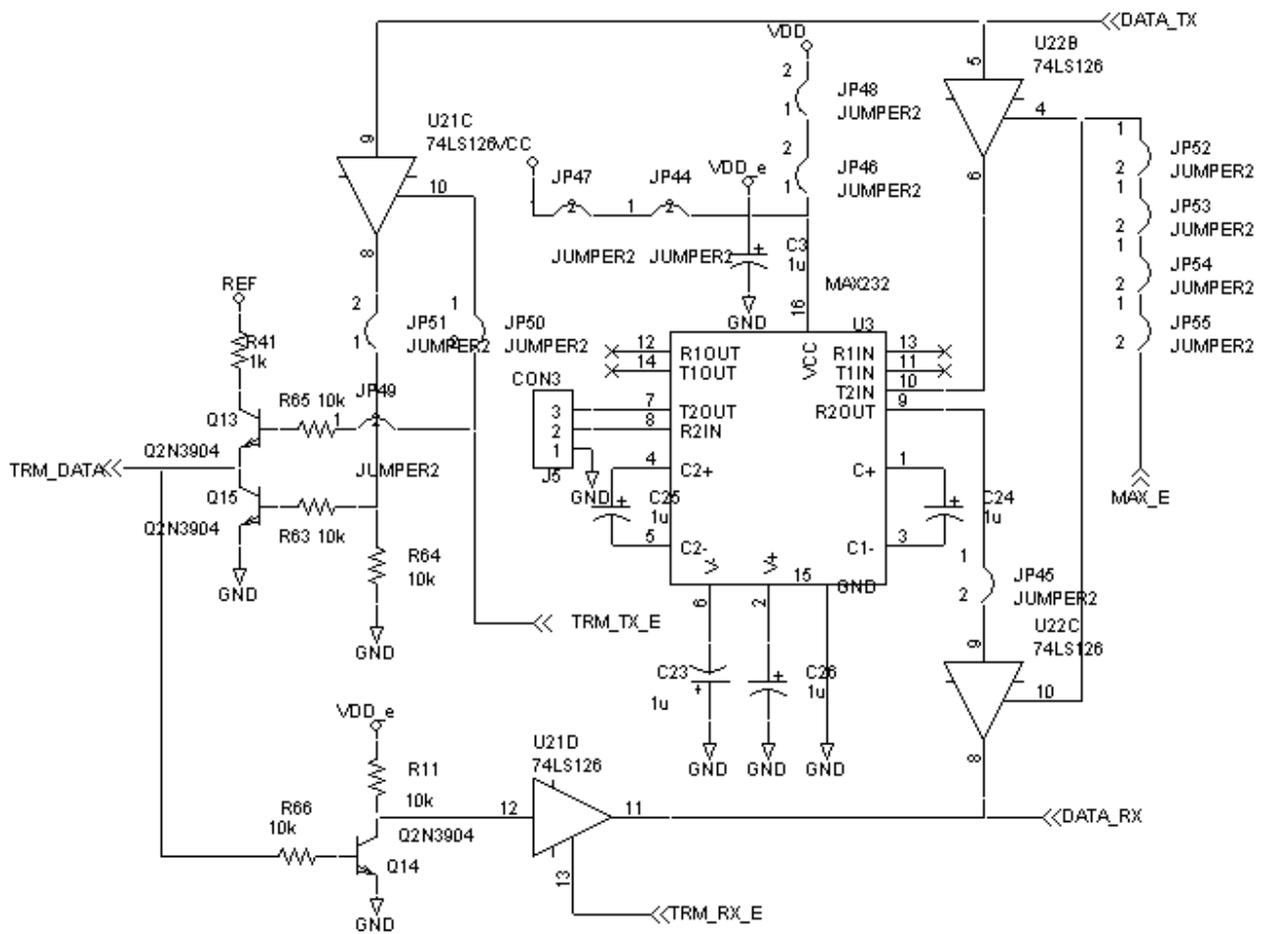


Figura 43. Diagrama eléctrico del selector sobre el puerto RS232

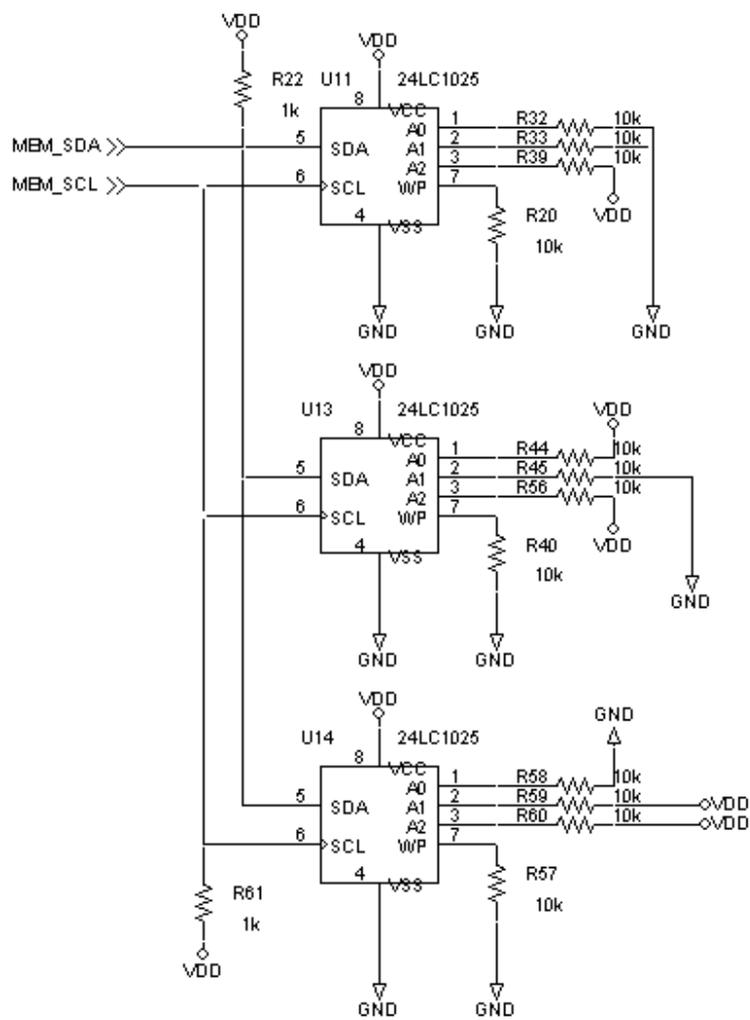


Figura 44. Diagrama eléctrico de la etapa de memoria.



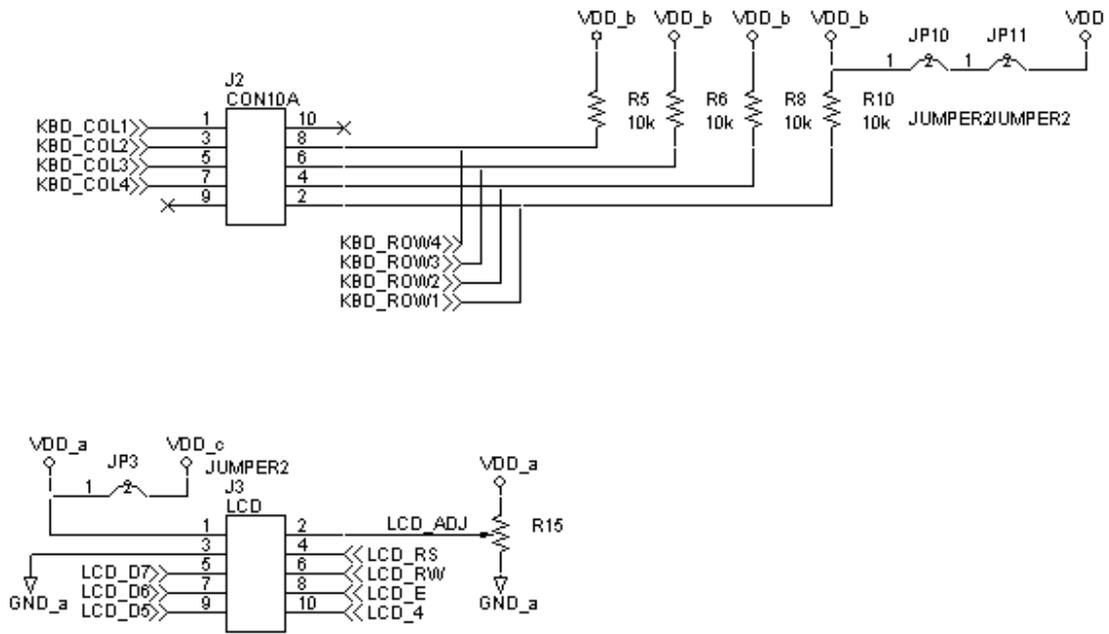


Figura 46. Diagrama eléctrico de conectores de teclado y pantalla LCD.

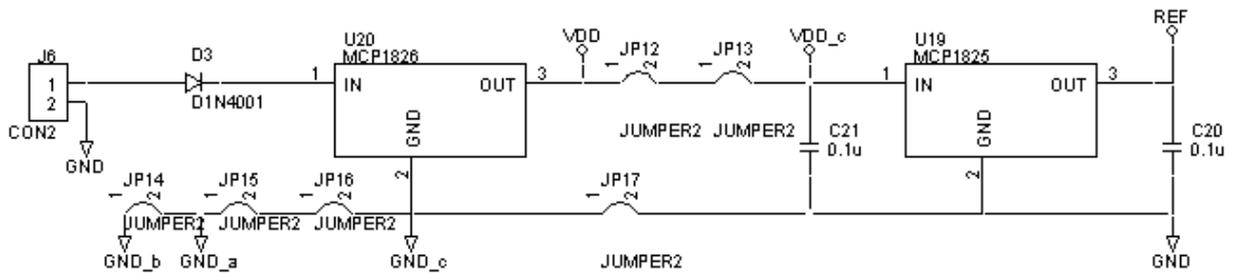


Figura 47. Diagrama eléctrico de reguladores de voltaje internos.

### 8.4.2. Dispositivo esclavo

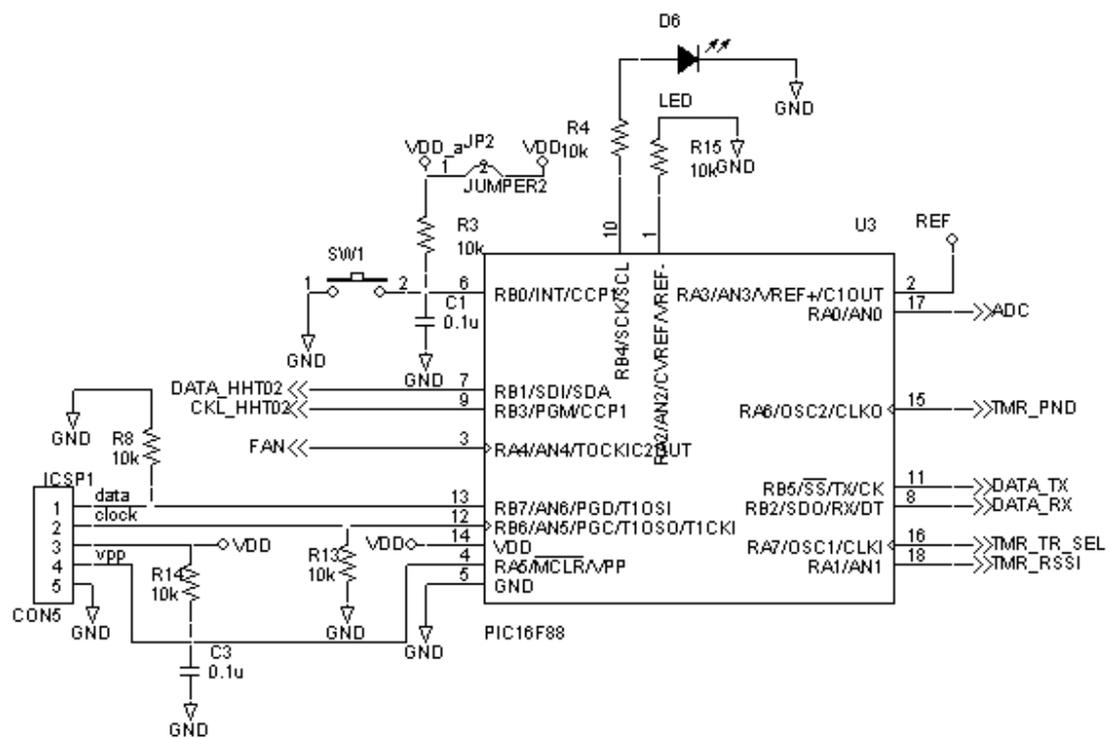


Figura 48. Diagrama eléctrico del esclavo.

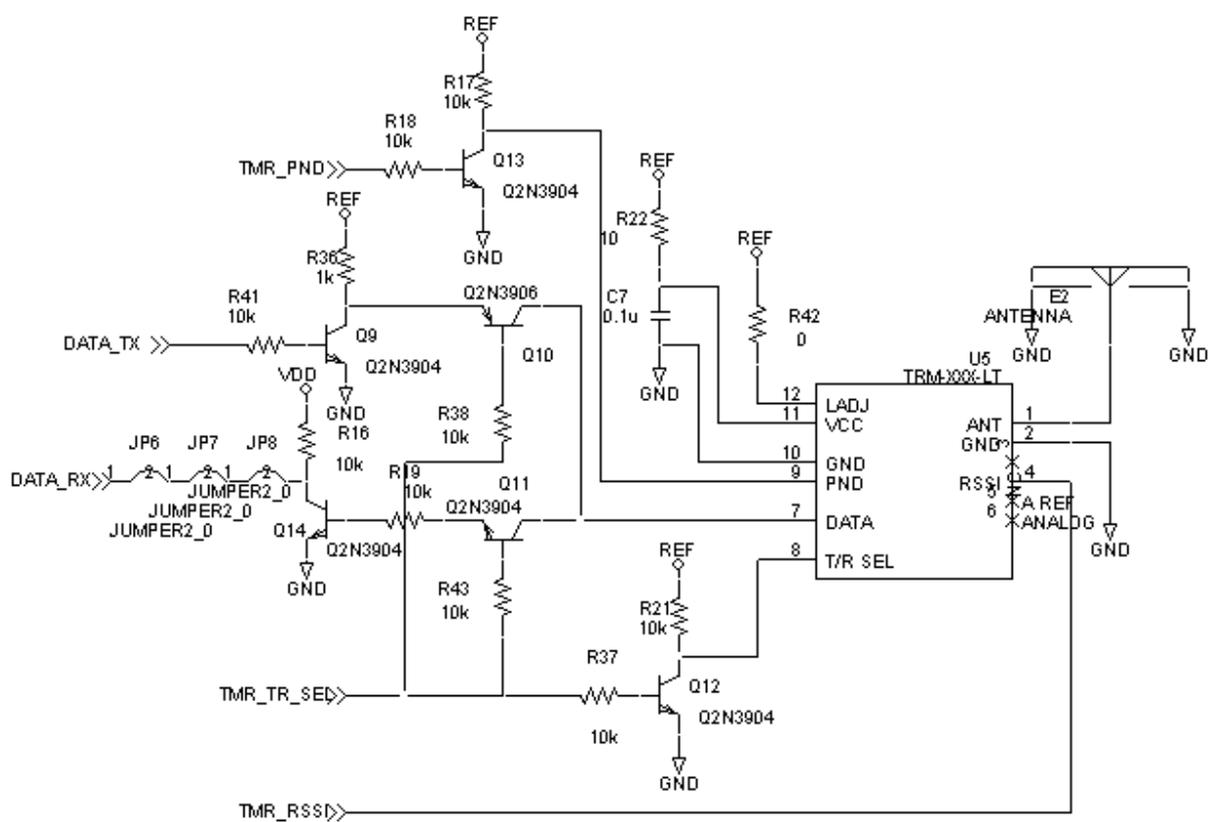


Figura 49. Diagrama eléctrico del módulo de comunicación inalámbrica.

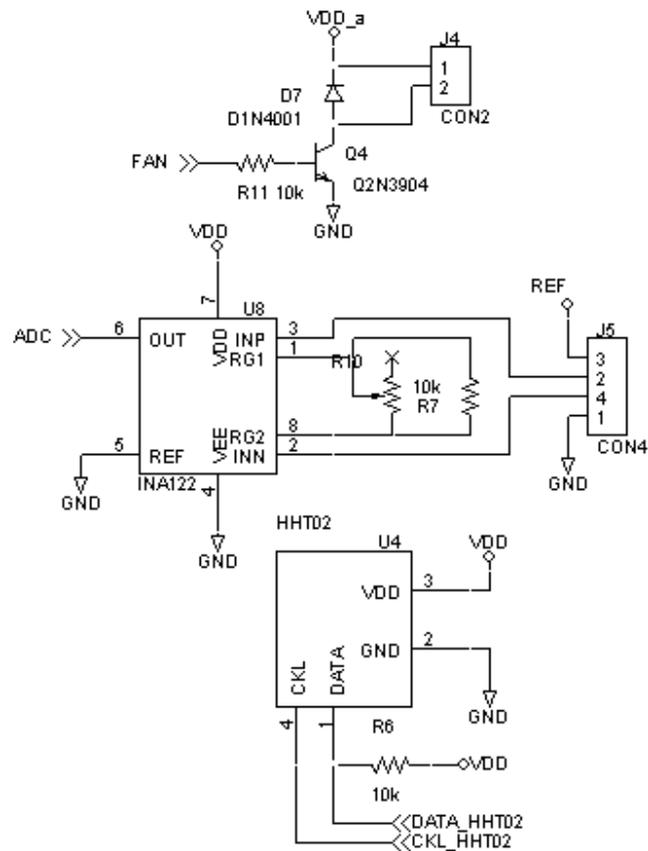


Figura 50. Diagrama eléctrico del sensor HHT02, amplificador para la señal del tensiómetro.

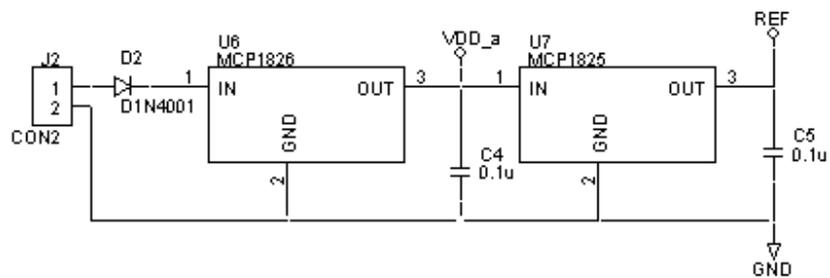


Figura 51. Diagrama eléctrico de reguladores de voltaje internos.

## 8.5. Circuitos impresos

### 8.5.1 Dispositivo coordinador

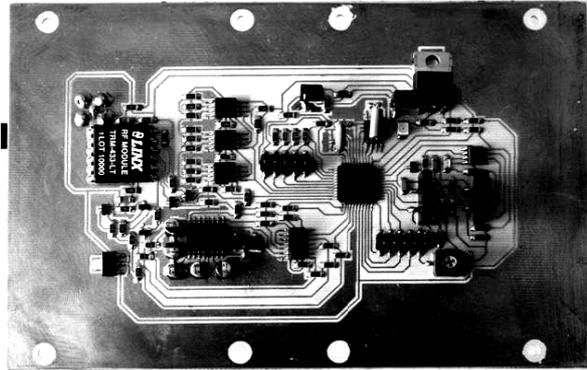


Figura 52. PCB del dispositivo coordinador.

### 8.5.2 Dispositivo esclavo

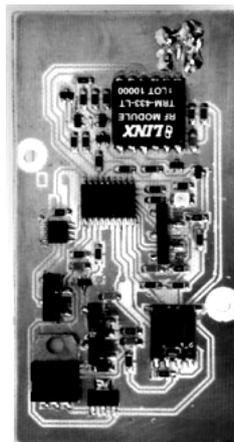


Figura 53. PCB del dispositivo esclavo.

## 8.6. Códigos de programación C

### 8.6.1. Dispositivo Coordinador:

```
#include <18F4680.h>
#fuses
hs,fcmen,ieso,put,nobrownout,borv43,wdt256,nopbaden,nolpt1osc,mclr,nostvren,nolvp,noxinst,nodebug,noprotect,nocpb,nocpd,nowrt,nowrtc,nowrtb,nowrtd,noebr,noebrb
#use delay(clock=8000000)
#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7,restart_wdt,timeout=10)
#use I2C(master,sda=pin_c4,scl=pin_c3)
#include "c:\documents and settings\hero\escritorio\pic_progs\header\crc.h"
//-----lcd
#define LCD_DB4 pin_b3
#define LCD_DB5 pin_a0
#define LCD_DB6 pin_b4
#define LCD_DB7 pin_b5
#define LCD_RS pin_b0
#define LCD_RW pin_b1
#define LCD_E pin_b2
#include "c:\documents and settings\hero\escritorio\pic_progs\header\flex_lcd.h"
//-----teclado
#define COL1 pin_d3
#define COL2 pin_d2
#define COL3 pin_d1
#define COL4 pin_d0
#define ROW1 pin_e2
#define ROW2 pin_c0
#define ROW3 pin_c1
#define ROW4 pin_c2
#include "c:\documents and settings\hero\escritorio\pic_progs\header\kbd.h"
//-----reloj
#define rtc_sclk pin_e1
#define rtc_io pin_a5
#define rtc_rst pin_e0
#include <ds1302.c>
//-----registros
#bit oerr = 0xfab.1
#bit cren = 0xfab.4
#bit scs0 = 0xfd3.0
#bit scs1 = 0xfd3.1
#bit osts = 0xfd3.3
//-----apuntadores

#define mem_hrs 0x0020
#define mem_min 0x0022
#define mem_seg 0x0024

#define p00 0x0026
#define p01 0x0028
#rom 0xf00020={0,5,0,0,0}
//-----manejadores
#define tr_pnd pin_c5
#define tr_sel pin_d4
#define tr_led pin_a4
#define tr_tx pin_d6
#define tr_rx pin_d5
#define max_on pin_d7
#define wdt_ext pin_a1
//-----variables globales
int1 flag_i=0,flag_r=0,flag_n=0,flag_led=0,flag_ini=0,flag_enc=0,flag_tt=0,flag_t=0,flag_recu=0;
int1 f1=0,f2=0,f3=0,f4=0,f5=0;
char str[17];
int8 msn[8],n_msn=0;

int8 s_t,h_t,m_t,d_t,mes_t;
int8 mes=0,dia=0,hora=0,min=0,seg=0;
int8 temp_h=0,temp_l=0;
```

```

int8 tension_h=0,tension_l=0;
int8 hrel=0;
float hum=0,tmp=0,ten=0;

void enviar_msn(int8,int8,int32);
int1 gettr(void);

void teclado(void);
void guardar (int8,int8);
void borrar(void);
void leer(void);

void reloj(void);
void reloj_hora(void);
void reloj_fecha(void);
void reloj_sens(void);

void sensor(void);
void sensor_ping(void);
void sensor_leer(void);
int1 sensor_gral(int8);

void memoria(void);
void e_eeprom(void);

void delay_s(int8);

void modo_tx(int1);
void modo_rx(int1);
void modo_pc(int1);

void recupera(void);

//-----rs232
#include <stdio.h>
void rda_isr(void)
{
    int8 n=0,dato=0,bit=0;
    if(!oerr){cren=0;delay_ms(10);cren=1;}
    if(getc()!=0x99) goto end_rda;
    for(bit=0;bit<8;bit++){
        n=getc();
        switch(n){
            case 0x86: bit_set(dato,bit); break;
            case 0x76: bit_clear(dato,bit); break;
        }
    }

    switch(dato){
        case 0x3a: if(!flag_i) {flag_i=1; n_msn=0;} else {msn[n_msn]=dato; n_msn++;} break;
        case 0x0d: if(n_msn==8 && flag_i==1) flag_r=1; else msn[n_msn]=dato; n_msn++; break;
        case 0x0a: if(n_msn==9 && flag_r==1) flag_n=1; else msn[n_msn]=dato; n_msn++; break;
        default: if(flag_i==1) {msn[n_msn]=dato; n_msn++;} break;
    }

    if(n_msn>10){n_msn=0;flag_i=0;flag_r=0;flag_n=0;}

    if(flag_n==1){
        modo_rx(0);
    }
end_rda:
    rs232_errors=0;
}
//-----timer 0
#include <time.h>
void timer0(void){
    int8 dd, hh, mm, ss, p, n, data[7], s_t, h_t, m_t;
    int16 dir;

```

```

int32 a=0,s=0,res_s=0,b=0;

set_timer1(3036); //1seg
rtc_get_time(hh,mm,ss);
rtc_get_date(dd,n,n,n);

a=(int32)hh*0x0e10;
a+=(int32)mm*0x003c;
a+=(int32)ss;

s=(int32)read_eeprom(mem_hrs)*0x0e10;
s+=(int32)read_eeprom(mem_min)*0x003c;
s+=(int32)read_eeprom(mem_seg);

res_s = a - ((a/s) * s);
if(res_s==0) flag_t=1;

if(flag_ini==0) flag_enc=1;
flag_tt=1;

a+=(int32)dd*0x15180;

p=0;
timer0_mem_1:
p++; if(p>5) e_eeprom();
dir=make16(read_eeprom(p01),read_eeprom(p00));
if(dir!=make16(read_eeprom(p01),read_eeprom(p00))) goto timer0_mem_1;

dir-=0x07;

i2c_start();
i2c_start();
i2c_write(0xa0);
i2c_write(make8(dir,1));
i2c_write(make8(dir,0));
i2c_start();
i2c_write(0xa1);

for(n=0;n<7;n++){
if(n==5) data[n]=i2c_read(0);
else data[n]=i2c_read(1);
}
i2c_stop();

s_t=((data[3]&0xc0)>>6) | ((data[2]&0x0f)<<2);
m_t=((data[2]&0xf0)>>4) | ((data[1]&0x03)<<4);
h_t=(data[1]&0x7c)>>2;
d_t=((data[1]&0x80)>>7) | ((data[0]&0x0f)<<1);
mes_t=(data[0]&0xf0)>>4;

b=(int32)d_t*0x15180;
b+=(int32)h_t*0x0e10;
b+=(int32)m_t*0x003c;
b+=(int32)s_t;

res_s = b - ((b/s) * s);

if(res_s!=0){
while(res_s>0x15180) {res_s-=0x15180; b-=0x15180; restart_wdt();}
while(res_s>0x0e10) {res_s-=0x0e10; b-=0x0e10; restart_wdt();}
while(res_s>0x003c) {res_s-=0x003c; b-=0x003c; restart_wdt();}
while(res_s>0) {res_s--; b--; restart_wdt();}
}

if(a>=(s+b)) flag_t=1;

```

```

if(a>=(2*s+b)) {flag_recu=1; flag_t=0;}
}
//-----timer 1
#int_timer1
void timer1(void){
  set_timer1(3036); //250ms
  restart_wdt();
  flag_led=~flag_led;
  bit_test(flag_led,0)?output_high(tr_led):output_low(tr_led);
  output_high(wdt_ext);
  delay_us(1);
  output_low(wdt_ext);
}
//-----osc fail
#int_oscf
void osc_fail(void){
  printf("\nfalla osc primario%1x \r\n",osts);
  //delay_ms(500); //tiempo de estabilizacion del osc primario
  //scs0=0;scs1=0; //regresa a uso de oscilador primario
}
//-----main
void main(void){
  int8 n=0,m=0;

  setup_oscillator(osc_8mhz);
  setup_timer_0(rtcc_internal|rtcc_div_32);
  setup_timer_1(t1_internal|t1_div_by_8);
  set_timer0(3036);
  set_timer1(3036);
  setup_wdt(wdt_on);
  setup_adc(adc_clock_internal);
  setup_adc_ports(an0|vss_vref);
  set_adc_channel(0);

  rtc_init();
  kbd_init();
  lcd_init();
  modo_pc(0);

  enable_interrupts(int_timer0);
  enable_interrupts(int_timer1);
  enable_interrupts(int_rda);
  enable_interrupts(int_oscf);
  enable_interrupts(global);

  modo_pc(1);
  printf("\r\n");
  delay_ms(10);
  modo_pc(0);
  while(true){
    if(flag_enc==1){
      rtc_get_date(dia,mes,n,n);
      rtc_get_time(hora,min,seg);
      sprintf(str,"HIDROCIENCIAS"); lcd_gotoxy(1,0); lcd_send_string(str);
      sprintf(str,"%02d-",dia); lcd_gotoxy(0,1); lcd_send_string(str);
      switch(mes){
        case 1: sprintf(str,"ENE "); break;
        case 2: sprintf(str,"FEB "); break;
        case 3: sprintf(str,"MAR "); break;
        case 4: sprintf(str,"ABR "); break;
        case 5: sprintf(str,"MAY "); break;
        case 6: sprintf(str,"JUN "); break;
        case 7: sprintf(str,"JUL "); break;
        case 8: sprintf(str,"AGO "); break;
        case 9: sprintf(str,"SEP "); break;
        case 10: sprintf(str,"OCT "); break;
        case 11: sprintf(str,"NOV "); break;
        case 12: sprintf(str,"DIC "); break;
      }
    }
  }
}

```

```

lcd_send_string(str);
sprintf(str,"%02d:%02d:%02d",hora,min,seg); lcd_send_string(str);
sprintf(str,"0-Menu"); lcd_gotoxy(10,3); lcd_send_string(str);
flag_enc=0;
}
if(!input(pin_c2)){
teclado();
}

if(flag_recu) {recupera();flag_t=0;flag_recu=0;}

if(flag_t){
lcd_clear();
for(n=1;n<=5;n++){
for(m=0;m<3;m++){
sprintf(str,"Sensor %d:",n); lcd_gotoxy(0,0);lcd_send_string(str);
if(sensor_gral(n+'0')){
switch(n){
case 1: ten = 136.0*ten + 60.96; break;
case 2: ten = 156.3*ten - 65.38; break;
case 3: ten = 151.4*ten + 44.88; break;
case 4: ten = 144.1*ten + 50.02; break;
case 5: ten = 156.8*ten - 30.47; break;
}
if(ten>715) {sprintf(str,"%c: ",0xe5); lcd_gotoxy(2,1);lcd_send_string(str);}
else {sprintf(str,"%c: %03.2fmb",0xe5,ten); lcd_gotoxy(2,1);lcd_send_string(str);}

sprintf(str,"HR: %02.2f%% ",hum); lcd_gotoxy(2,2);lcd_send_string(str);
sprintf(str,"T: %02.2f%cC",tmp,0xdf);lcd_gotoxy(2,3);lcd_send_string(str);
modo_pc(1);

if(msn[2]==0x05){
printf("Sensor %d: %02d/%02dt%02d:%02d:%02d\t\t\t\t\r\n",n,dia,mes,hora,min,seg);
guardar(n,3);
}
else{
printf("Sensor %d: %02d/%02dt%02d:%02d:%02d\t",n,dia,mes,hora,min,seg);
if(ten>715) printf("\t");
else printf("\t\t",ten);
printf("\t\t\t\t\r\n",hum,tmp);
guardar(n,1);
}

delay_ms(1000);
lcd_clear();
break;
}
else{
lcd_clear();
sprintf(str,"Sensor %d:",n); lcd_gotoxy(0,0);lcd_send_string(str);
switch(m){
case 0: sprintf(str,"Conectando. ");lcd_gotoxy(2,2);lcd_send_string(str); break;
case 1: sprintf(str,"Conectando.. ");lcd_gotoxy(2,2);lcd_send_string(str); break;
case 2: sprintf(str,"Sin conexion"); lcd_gotoxy(2,2);lcd_send_string(str);
delay_ms(500);
lcd_clear();
modo_pc(1);
printf("Sensor %d: %02d/%02dt%02d:%02d:%02d\t\t\t\t\r\n",n,dia,mes,hora,min,seg);
guardar(n,0);
break;
}
}
}
}
modo_pc(1);
printf("\r\n");
delay_ms(10);
lcd_clear();
modo_pc(0);
flag_t=0;

```

```

    }
  }
}
//-----
void enviar_msn(int8 dir, int8 fun, int32 d32){
  int8 msn[11],n,i;
  int16 crc_d=0x0000;

  msn[0]= 0x3a;
  msn[1]= dir; //direccion
  msn[2]= fun; //funcion
  msn[3]= make8(d32,3);
  msn[4]= make8(d32,2);
  msn[5]= make8(d32,1);
  msn[6]= make8(d32,0);
  for(n=1;n<7;n++) crc_d=crc16(msn[n],crc_d);
  msn[7]= make8(crc_d,1);
  msn[8]= make8(crc_d,0);
  msn[9]= 0x0d;
  msn[10]=0x0a;

  modo_tx(1);
  for(n=0;n<10;n++) putc(0); //limpiar canal
  for(n=0;n<11;n++){
    putc(0x99);
    for(i=0;i<8;i++){
      if(bit_test(msn[n],i)) putc(0x86); //1
      else putc(0x76); //0
    }
  }
}
//-----guardar
void guardar(int8 mem,int8 nul){
  int8 dato[7];
  int8 n,p;
  int16 dir;

  if(nul==0) {temp_l=0xff;temp_h=0xff;hrel=0xff;tension_l=0xff;tension_h=0xff;}

  if(nul==3) {temp_l=0xfe;temp_h=0xfe;hrel=0xfe;tension_l=0xfe;tension_h=0xfe;}

  if(nul==2){
    dato[6] = 0x00;
    dato[5] = 0x00 | 0x00<<4;
    dato[4] = 0x00>>4 | 0x00<<4;
    dato[3] = 0x00>>4 | 0x00<<4 | s_t<<6 ;
    dato[2] = s_t>>2 | m_t<<4;
    dato[1] = d_t<<7 | h_t<<2 | m_t>>4;
    dato[0] = d_t>>1 | mes_t<<4;
  }
  else{
    dato[6] = temp_l;
    dato[5] = temp_h | hrel<<4;
    dato[4] = hrel>>4 | tension_l<<4;
    dato[3] = tension_l>>4 | tension_h<<4 | seg<<6 ;
    dato[2] = seg>>2 | min<<4;
    dato[1] = dia<<7 | hora<<2 | min>>4;
    dato[0] = dia>>1 | mes<<4;
  }

  p=0;
  guardar_mem_1:
  p++; if(p>5) e_eeprom();
  dir=make16(read_eeprom(p01),read_eeprom(p00));
  if(dir!=make16(read_eeprom(p01),read_eeprom(p00))) goto guardar_mem_1;

  if(dir>=0xffff) dir=0x0000;

```

```

for(n=0;n<7;n++){
  i2c_start();
  i2c_start();
  switch(mem){
    case 1: i2c_write(0xa0); f1=1; break;
    case 2: i2c_write(0xa2); f2=1; break;
    case 3: i2c_write(0xa4); f3=1; break;
    case 4: i2c_write(0xa8); f4=1; break;
    case 5: i2c_write(0xaa); f5=1; break;
  }
  i2c_write(make8(dir,1));
  i2c_write(make8(dir,0));
  i2c_write(dato[n]);
  i2c_stop();

  delay_ms(6);
  dir++;
}
if(f1&&f2&&f3&&f4&&f5){
  f1=0;f2=0;f3=0;f4=0;f5=0;
  p=0;
guardar_mem_2:
  p++; if(p>5) e_eeprom();
  write_eeprom(p00,make8(dir,0));
  write_eeprom(p01,make8(dir,1));
  if(dir!=make16(read_eeprom(p01),read_eeprom(p00))) goto guardar_mem_2;
}
}
//-----borrar
void borrar(void){
  int8 sec=0,b=0;
  int32 n=0,m=1,k=0;
  int1 bAck=0;

  modo_pc(1);
borrar_ini:
  lcd_clear();
  sprintf(str,"Borrar memoria:"); lcd_gotoxy(0,0); lcd_send_string(str);
  //sprintf(str,"[1-5]:"); lcd_gotoxy(3,1); lcd_send_string(str);
  sprintf(str,"1-Aceptar"); lcd_gotoxy(7,2); lcd_send_string(str);
  sprintf(str,"0-Salir"); lcd_gotoxy(7,3); lcd_send_string(str);
  sec=kbd()-'0';
  //sprintf(str,"%d",sec); lcd_gotoxy(10,1);lcd_send_string(str);
  if(sec==0) goto borrar_fin;
  if(sec>1){
    sprintf(str,"error opcion"); lcd_gotoxy(0,1); lcd_send_string(str);
    delay_ms(1000);
    goto borrar_ini;
  }

  write_eeprom(p00,0);
  write_eeprom(p01,0);

  for(sec=1;sec<=5;sec++){
    for(n=0;n<0xffff;n+=128){
      i2c_start();
      i2c_start();
      switch(sec){
        case 1: i2c_write(0xa0); break;
        case 2: i2c_write(0xa2); break;
        case 3: i2c_write(0xa4); break;
        case 4: i2c_write(0xa8); break;
        case 5: i2c_write(0xaa); break;
      }
      i2c_write(make8(n,1));
      i2c_write(make8(n,0));
      for(b=0;b<128;b++){
        i2c_write(0xff);
      }
    }
  }
}

```

```

i2c_stop();

do{
  i2c_start();
  i2c_start();
  switch(sec){
    case 1: bAck=i2c_write(0xa0); break;
    case 2: bAck=i2c_write(0xa2); break;
    case 3: bAck=i2c_write(0xa4); break;
    case 4: bAck=i2c_write(0xa8); break;
    case 5: bAck=i2c_write(0xaa); break;
  }
  i2c_stop();
}while(bAck);

k=m;
m=((n+(((int32)sec-1)*0xffff))/0x0ccc);

if(k!=m){
  if(m<10) {lcd_gotoxy(0,1); sprintf(str,"|-----|"); lcd_send_string(str);}
  for(k=1;k<=m/10;k++) {lcd_gotoxy(k ,1);sprintf(str,"%c",0xff); lcd_send_string(str);}
  lcd_gotoxy(12,1); sprintf(str,"%3lu%%",m); lcd_send_string(str);
}

//delay_ms(6);

if(!input(pin_c2)){
  sec=6;
  //lcd_gotoxy(0,3);
  //sprintf(str,"cancelado");
  //lcd_send_string(str);
  break;
}
}
}
borrar_fin:
  lcd_clear();
  modo_pc(0);
}
//-----leer
void leer(void){
  int1 flag_sc=0,flag_dr=0,flag_er=0;
  int8 n=0,dato[7],j=0,i=1,sec=0;
  int16 m=0,a=0,b=0;

leer_ini:
  lcd_clear();
  sprintf(str,"Leer bloque:");   lcd_gotoxy(0,0); lcd_send_string(str);
  sprintf(str,"[1-5]:");        lcd_gotoxy(3,1); lcd_send_string(str);
  sprintf(str,"0-Salir");        lcd_gotoxy(9,3); lcd_send_string(str);
  sec=kbd()-'0';
  sprintf(str,"%d",sec);         lcd_gotoxy(10,1);lcd_send_string(str);
  if(sec==0) goto leer_fin;
  if(sec>5){
    sprintf(str,"error bloque");  lcd_gotoxy(0,2); lcd_send_string(str);
    delay_ms(1000);
    goto leer_ini;
  }
  modo_pc(1);
  printf("\n\nBloque %d\n",sec);
  printf("\td/mm\thh:mm:ss\tt[mb]\tHR[%%]\tT[C]\r\n");
  i2c_start();
  switch(sec){
    case 1: i2c_write(0xa0); break;
    case 2: i2c_write(0xa2); break;
    case 3: i2c_write(0xa4); break;
    case 4: i2c_write(0xa8); break;
    case 5: i2c_write(0xaa); break;
  }
  i2c_write(0x00);

```



```

//printf("%02d/%02d/%02d:%02d:%02d\t%lu\t%u\t%lu\r\n", dia, mes, hora, min, seg, make16(tension_h, tension_l), hrel, make16(temp_h
, temp_l));
}
j=i;
i=a/0x28f;

if(j!=i){
if(i<10) {lcd_gotoxy(0,2); sprintf(str,"|-----|"); lcd_send_string(str);}
for(j=1; j<=i/10; j++) {lcd_gotoxy(j,2); sprintf(str,"%c",0xff); lcd_send_string(str);}
lcd_gotoxy(12,2); sprintf(str,"%3d%%",i); lcd_send_string(str);
}

if(linput(pin_c2)){
n=8; m=0x2495;
i2c_read(1);
i2c_stop();
i2c_start();
switch(sec){
case 1: i2c_write(0xa0); break;
case 2: i2c_write(0xa2); break;
case 3: i2c_write(0xa4); break;
case 4: i2c_write(0xa8); break;
case 5: i2c_write(0xaa); break;
}
i2c_write(0x00);
i2c_write(0x00);
i2c_stop();
//lcd_gotoxy(0,3);
//sprintf(str,"cancelado");
//lcd_send_string(str);
}
}
i2c_stop();
goto leer_ini;

leer_fin:
modo_pc(0);
lcd_clear();
}
//-----reloj_fecha
void reloj_fecha(void){
int8 err=0;
lcd_clear();
sprintf(str,"Fecha:"); lcd_gotoxy(0,0); lcd_send_string(str);

sprintf(str,"[01-31dia]"); lcd_gotoxy(3,2); lcd_send_string(str);
dia=(kbd()-'0')*10; if(dia>30) {err=1; goto reloj_fecha_fin;}
sprintf(str,"%d", dia/10); lcd_gotoxy(7,0); lcd_send_string(str);
dia+=(kbd()-'0'); if(dia>31) {err=1; goto reloj_fecha_fin;}
sprintf(str,"%02d", dia); lcd_gotoxy(7,0); lcd_send_string(str);

sprintf(str,"[01-12mes]"); lcd_gotoxy(3,2); lcd_send_string(str);
mes=(kbd()-'0')*10; if(mes>10) {err=2; goto reloj_fecha_fin;}
sprintf(str,"%d", mes/10); lcd_gotoxy(10,0); lcd_send_string(str);
mes+=(kbd()-'0'); if(mes>12) {err=2; goto reloj_fecha_fin;}
sprintf(str,"%02d", mes); lcd_gotoxy(10,0); lcd_send_string(str);

reloj_fecha_fin:
if(err==0){
rtc_get_time(hora, min, seg);
rtc_set_datetime(dia, mes, 0x00, 0x00, hora, min);
}
else{
sprintf(str,"error "); lcd_gotoxy(0,3); lcd_send_string(str);
switch(err){
case 1: sprintf(str,"dia"); lcd_send_string(str); break;
case 2: sprintf(str,"mes"); lcd_send_string(str); break;
}
}
}

```

```

    delay_ms(500);
}
//-----reloj_hora
void reloj_hora(void){
    int8 err=0;
    lcd_clear();
    sprintf(str,"Hora:");lcd_gotoxy(0,0);lcd_send_string(str);
    sprintf(str,"[00-23hrs]");lcd_gotoxy(3,2);lcd_send_string(str);
    hora=(kbd()-'0')*10;    if(hora>20) {err=1; goto reloj_hora_fin;}
    sprintf(str,"%d",hora/10);lcd_gotoxy(7,0);lcd_send_string(str);
    hora+=(kbd()-'0');    if(hora>23) {err=1; goto reloj_hora_fin;}
    sprintf(str,"%02d:",hora); lcd_gotoxy(7,0);lcd_send_string(str);

    sprintf(str,"[00-59min]");lcd_gotoxy(3,2);lcd_send_string(str);
    min=(kbd()-'0')*10;    if(min>50) {err=2; goto reloj_hora_fin;}
    sprintf(str,"%d",min/10); lcd_gotoxy(10,0);lcd_send_string(str);
    min+=(kbd()-'0');    if(min>59) {err=2; goto reloj_hora_fin;}
    sprintf(str,"%02d",min); lcd_gotoxy(10,0);lcd_send_string(str);
reloj_hora_fin:
    if(err==0){
        rtc_get_date(dia,mes,err,err);
        rtc_set_datetime(dia,mes,0x00,0x00,hora,min);
    }
    else{
        sprintf(str,"error ");lcd_gotoxy(0,3);lcd_send_string(str);
        switch(err){
            case 1: sprintf(str,"hora"); lcd_send_string(str); break;
            case 2: sprintf(str,"min"); lcd_send_string(str); break;
        }
    }
    delay_ms(500);
    flag_t=0;
}
//-----
void teclado(void){
    int8 n=0;
    flag_ini=1;
teclado_ini:
    lcd_clear();
    sprintf(str,"Menu:"); lcd_gotoxy(0,0); lcd_send_string(str);
    sprintf(str,"0-Salir"); lcd_gotoxy(6,0); lcd_send_string(str);
    sprintf(str,"1-Reloj"); lcd_gotoxy(6,1); lcd_send_string(str);
    sprintf(str,"2-Sensor"); lcd_gotoxy(6,2); lcd_send_string(str);
    sprintf(str,"3-Memoria"); lcd_gotoxy(6,3); lcd_send_string(str);
    while(!input(pin_c2)); // row4=esc;
    delay_ms(10);

    n=kbd();
    if(n=='0') goto teclado_fin;
    if(n=='1') reloj();
    if(n=='2') sensor();
    if(n=='3'){
        disable_interrupts(int_timer0);
        memoria();
        enable_interrupts(int_timer0);
    }
    goto teclado_ini;

teclado_fin:
    lcd_clear();
    flag_ini=0;
}
//-----
void reloj(void){
    int8 n=0;
reloj_ini:
    lcd_clear();
    sprintf(str,"Reloj:"); lcd_gotoxy(0,0);lcd_send_string(str);
    sprintf(str,"0-Salir"); lcd_gotoxy(7,0);lcd_send_string(str);
    sprintf(str,"1-Hora"); lcd_gotoxy(7,1);lcd_send_string(str);

```

```

sprintf(str,"2-Fecha"); lcd_gotoxy(7,2);lcd_send_string(str);
sprintf(str,"3-Muestra"); lcd_gotoxy(7,3);lcd_send_string(str);
while(!input(pin_c2));
delay_ms(10);

n=kbd();
if(n=='0') goto reloj_fin;
if(n=='1') reloj_hora(); break;
if(n=='2') reloj_fecha();break;
if(n=='3') reloj_sens(); break;
goto reloj_ini;

reloj_fin:
lcd_clear();
}
//-----
void reloj_sens(void){
int8 t=0,err=0,n=0;

reloj_sens_ini:
lcd_clear();
sprintf(str,"Muestra:"); lcd_gotoxy(0,0);lcd_send_string(str);
sprintf(str,"0-Salir"); lcd_gotoxy(9,0);lcd_send_string(str);
sprintf(str,"1-[hrs]"); lcd_gotoxy(9,1);lcd_send_string(str);
sprintf(str,"2-[min]"); lcd_gotoxy(9,2);lcd_send_string(str);
sprintf(str,"3-[seg]"); lcd_gotoxy(9,3);lcd_send_string(str);
sprintf(str,"tiempo"); lcd_gotoxy(0,2);lcd_send_string(str);
sprintf(str,"%02d:%02d:%02d",read_eeprom(mem_hrs),read_eeprom(mem_min),read_eeprom(mem_seg));
lcd_gotoxy(0,3);lcd_send_string(str);

while(!input(pin_c2));
delay_ms(10);

n=kbd();
if(n=='0') goto reloj_sens_fin2;
if(n=='1'){
lcd_clear();
sprintf(str,"[hrs]:"); lcd_gotoxy(0,0);lcd_send_string(str);
sprintf(str,"[00-23hrs]"); lcd_gotoxy(3,2);lcd_send_string(str);
t=(kbd()-'0')*10; if(t>20) {err=1; goto reloj_sens_fin;}
sprintf(str,"%d",t/10); lcd_gotoxy(7,0);lcd_send_string(str);
t+=(kbd()-'0'); if(t>23) {err=1; goto reloj_sens_fin;}
sprintf(str,"%02d",t); lcd_gotoxy(7,0);lcd_send_string(str);
write_eeprom(mem_hrs,t);
delay_ms(1000);
}
if(n=='2'){
lcd_clear();
sprintf(str,"[min]:"); lcd_gotoxy(0,0);lcd_send_string(str);
sprintf(str,"[00-59min]");lcd_gotoxy(3,2);lcd_send_string(str);
t=(kbd()-'0')*10; if(t>50) {err=2; goto reloj_sens_fin;}
sprintf(str,"%d",t/10); lcd_gotoxy(7,0);lcd_send_string(str);
t+=(kbd()-'0'); if(t>59) {err=2; goto reloj_sens_fin;}
sprintf(str,"%02d",t); lcd_gotoxy(7,0);lcd_send_string(str);
write_eeprom(mem_min,t);
delay_ms(1000);
}
if(n=='3'){
lcd_clear();
sprintf(str,"[seg]:"); lcd_gotoxy(0,0);lcd_send_string(str);
sprintf(str,"[00-59seg]");lcd_gotoxy(3,2);lcd_send_string(str);
t=(kbd()-'0')*10; if(t>50) {err=3; goto reloj_sens_fin;}
sprintf(str,"%d",t/10); lcd_gotoxy(7,0);lcd_send_string(str);
t+=(kbd()-'0'); if(t>59) {err=3; goto reloj_sens_fin;}
sprintf(str,"%02d",t); lcd_gotoxy(7,0);lcd_send_string(str);
write_eeprom(mem_seg,t);
delay_ms(1000);
}
}

```

```

reloj_sens_fin:
if(err!=0){
    sprintf(str,"error ");lcd_gotoxy(0,3);lcd_send_string(str);
    switch(err){
        case 1: sprintf(str,"hora"); lcd_send_string(str); break;
        case 2: sprintf(str,"min"); lcd_send_string(str); break;
        case 3: sprintf(str,"seg"); lcd_send_string(str); break;
    }
}
if(read_eeprom(mem_min)==0 && read_eeprom(mem_hrs)==0 && read_eeprom(mem_seg)==0) write_eeprom(mem_seg,1);
goto reloj_sens_ini;
reloj_sens_fin2:
lcd_clear();

}
//-----
void sensor(void){
    int8 n=0;
sensor_ini:
    lcd_clear();
    sprintf(str,"Sensor:"); lcd_gotoxy(0,0);lcd_send_string(str);
    sprintf(str,"0-Salir"); lcd_gotoxy(8,0);lcd_send_string(str);
    sprintf(str,"1-PING"); lcd_gotoxy(8,1);lcd_send_string(str);
    sprintf(str,"2-Leer"); lcd_gotoxy(8,2);lcd_send_string(str);
    while(!input(pin_c2));
    delay_ms(10);

    n=kbd();
    if(n=='0') goto sensor_fin;
    if(n=='1') sensor_ping();
    if(n=='2') sensor_leer();

    goto sensor_ini;
sensor_fin:
    lcd_clear();
}
//-----
void sensor_ping(void){
    int8 sensor=0;
sensor_ping_ini:
    lcd_clear();
    sprintf(str,"Sensor PING:"); lcd_gotoxy(0,0);lcd_send_string(str);
    sprintf(str,"[1-5]"); lcd_gotoxy(5,1);lcd_send_string(str);
    sprintf(str,"0-Salir"); lcd_gotoxy(9,3);lcd_send_string(str);
    sensor=kbd();
    sprintf(str,"%c",sensor); lcd_gotoxy(13,0);lcd_send_string(str);

    if(sensor=='0') goto sensor_ping_fin;
    if(sensor>='1' && sensor<='5'){
        modo_tx(1);
        delay_ms(20);
        enviar_msn(sensor,0x01,make32(0x00,0x00,0x00,0x00));

        if(gettr()){
            sprintf(str,"Sensor %c OK",sensor); lcd_gotoxy(2,2);lcd_send_string(str);
        }
        else{
            sprintf(str,"Sensor %c X",sensor); lcd_gotoxy(3,2);lcd_send_string(str);
        }
    }
    else{
        sprintf(str,"error sensor");lcd_gotoxy(0,2);lcd_send_string(str);
    }

    delay_ms(2500);
    goto sensor_ping_ini;
sensor_ping_fin:
    delay_ms(1);
}

```

```

//-----
void sensor_leer(void){
    int8 sensor=0;
sensor_leer_ini:
    lcd_clear();
    sprintf(str,"Sensor LEER:"); lcd_gotoxy(0,0);lcd_send_string(str);
    sprintf(str,"[1-5]");    lcd_gotoxy(5,1);lcd_send_string(str);
    sprintf(str,"0-Salir");    lcd_gotoxy(9,3);lcd_send_string(str);
    sensor=kbd();
    sprintf(str,"%c",sensor); lcd_gotoxy(13,0);lcd_send_string(str);

if(sensor=='0') goto sensor_leer_fin;
if(sensor>='1' && sensor<='5'){
    if(sensor_gral(sensor)){
        lcd_clear();
        sprintf(str,"Sensor %c:",sensor);    lcd_gotoxy(0,0);lcd_send_string(str);
        switch(sensor){
            case '1': ten = 136.0*ten + 60.96; break;
            case '2': ten = 156.3*ten - 65.38; break;
            case '3': ten = 151.4*ten + 44.88; break;
            case '4': ten = 144.1*ten + 50.02; break;
            case '5': ten = 156.8*ten - 30.47; break;
        }
        if(ten>715) {sprintf(str,"%c: -.0xe5); lcd_gotoxy(2,1);lcd_send_string(str);}
        else {sprintf(str,"%c: %03.2fmb",0xe5,ten); lcd_gotoxy(2,1);lcd_send_string(str);}
        sprintf(str,"HR: %02.2f%%",hum);    lcd_gotoxy(2,2);lcd_send_string(str);
        sprintf(str,"T: %02.2f%C",tmp,0xdf);lcd_gotoxy(2,3);lcd_send_string(str);
    }
    else{
        sprintf(str,"Sensor %c X",sensor); lcd_gotoxy(3,2);lcd_send_string(str);
    }
}
else{
    sprintf(str,"error sensor");lcd_gotoxy(0,2);lcd_send_string(str);
}

    delay_ms(5000);
    goto sensor_leer_ini;
sensor_leer_fin:
    delay_ms(1);
}
//-----
int1 gettr(void){
    int8 r=0,n=0;
    int16 crc_d=0;
    modo_rx(1);
    while(flag_n==0 && r<=1){ //3s
        delay_s(1);
        r++;
    }
    if(flag_n==1){
        flag_n=0;flag_j=0;flag_r=0;crc_d=0;
        for(n=0;n<6;n++) crc_d=crc16(msn[n],crc_d);
        if(make8(crc_d,1)==msn[6] && make8(crc_d,0)==msn[7]) n=1;//reviion CRC
        else n=0;
    }
    else n=0;
    return n;
}
//-----
void delay_s(int8 seg){
    int n=0;
    for(n=0;n<seg;n++){
        flag_tt=0;
        while(flag_tt==0);
    }
}
//-----
int1 sensor_gral(int8 sensor){

```

```

int8 n=0;
modo_tx(1);
delay_ms(20);
enviar_msn(sensor,0x03,make32(0x00,0x00,0x00,0x00));

if(gettr()){

temp_l=msn[5];
temp_h=msn[4]&0x0f;
hrel=((msn[4]&0xf0)>>4) | ((msn[3]&0x0f)<<4);
tension_l=((msn[3]&0xf0)>>4) | ((msn[2]&0x0f)<<4);
tension_h=(msn[2]&0x30)>>4;

ten=(float)make16(tension_h,tension_l)/204.8; //tension 10bits!
tmp=(0.04*(float)make16(temp_h,temp_l))-40;
hum= -4 + (0.648*(float)hrel) - (0.00072 * (float)hrel * (float)hrel);
hum= (tmp - 25) * (0.01 + (0.00128 * (float)hrel)) + hum;

rtc_get_date(dia,mes,n,n);
rtc_get_time(hora,min,seg);
return 1;
}
else{
rtc_get_date(dia,mes,n,n);
rtc_get_time(hora,min,seg);
return 0;
}
}
//-----
void memoria(void){
int8 n=0;
memoria_ini:
lcd_clear();
sprintf(str,"Memoria:"); lcd_gotoxy(0,0);lcd_send_string(str);
sprintf(str,"0-Salir"); lcd_gotoxy(8,0);lcd_send_string(str);
sprintf(str,"1-Leer"); lcd_gotoxy(8,1);lcd_send_string(str);
sprintf(str,"2-Borrar"); lcd_gotoxy(8,2);lcd_send_string(str);
while(!input(pin_c2));
delay_ms(10);

n=kbd();
if(n=='0') goto memoria_fin;
if(n=='1') leer();
if(n=='2') borrar();

goto memoria_ini;
memoria_fin:
lcd_clear();
}
//-----
void modo_tx(int1 f){
output_low(tr_tx);
output_low(tr_rx);
output_high(tr_pnd);
output_low(max_on);
output_low(tr_sel); //modo transmisor

output_high(tr_tx);
if(f) output_low(tr_pnd);
}
//-----
void modo_rx(int1 f){
output_low(tr_tx);
output_low(tr_rx);
output_high(tr_pnd);
output_low(max_on);
output_high(tr_sel); //modo receptor

```

```

output_high(tr_rx);
if(f)output_low(tr_pnd);
}
//-----
void modo_pc(int1 f){
output_low(tr_tx);
output_low(tr_rx);
output_high(tr_pnd);
output_low(max_on);

if(f)output_high(max_on);
}
//-----
void e_eeeprom(void){
lcd_clear();
sprintf(str,"ERROR EEPROM!!"); lcd_gotoxy(1,1);lcd_send_string(str);
modo_pc(1);
printf("ERROR EEPROM!!\n\n");
delay_ms(2000);
reset_cpu();
}
//-----
void recupera(void){
int8 dd,hh,mm,ss,p,n,data[7],j=0,i=1;
int16 dir;
int32 a=0,s=0,res_s=0,b=0,c=0;
disable_interrupts(int_timer0);

lcd_clear();
sprintf(str,"Actualizando"); lcd_gotoxy(2,0); lcd_send_string(str);
sprintf(str,"Datos"); lcd_gotoxy(5,1); lcd_send_string(str);

p=0;
recupera_mem_1:
p++; if(p>5) e_eeeprom();
dir=make16(read_eeeprom(p01),read_eeeprom(p00));
if(dir!=make16(read_eeeprom(p01),read_eeeprom(p00))) goto recupera_mem_1;

if(dir==0) goto recupera_fin;
dir-=0x07;

i2c_start();
i2c_start();
i2c_write(0xa0);
i2c_write(make8(dir,1));
i2c_write(make8(dir,0));
i2c_start();
i2c_write(0xa1);

for(n=0;n<7;n++){
if(n==5 ) data[n]=i2c_read(0);
else data[n]=i2c_read(1);
}
i2c_stop();

s_t=((data[3]&0xc0)>>6) | ((data[2]&0x0f)<<2);
m_t=((data[2]&0xf0)>>4) | ((data[1]&0x03)<<4);
h_t=(data[1]&0x7c)>>2;
d_t=((data[1]&0x80)>>7) | ((data[0]&0x0f)<<1);
mes_t=(data[0]&0xf0)>>4;

b =(int32)h_t*0x00e10;
b+=(int32)m_t*0x0003c;
b+=(int32)s_t;

s =(int32)read_eeeprom(mem_hrs)*0x0e10;

```

```

s+=(int32)read_eeprom(mem_min)*0x003c;
s+=(int32)read_eeprom(mem_seg);

res_s = b - ((b/s) * s);

if(res_s!=0){
  while(res_s>0x15180) {res_s-=0x15180; b-=0x15180; restart_wdt();}
  while(res_s>0x0e10) {res_s-=0x0e10; b-=0x0e10; restart_wdt();}
  while(res_s>0x003c) {res_s-=0x003c; b-=0x003c; restart_wdt();}
  while(res_s>0) {res_s--; b--; restart_wdt();}
}

b+=(int32)d_t*0x15180;

rtc_get_time(hh,mm,ss);
rtc_get_date(dd,n,n,n);

a =(int32)dd*0x15180;
a+=(int32)hh*0x00e10;
a+=(int32)mm*0x0003c;
a+=(int32)ss;

c=(a-b)/s;

while(b<=a){

  j=i;
  i=make8(100-((100*(a-b))/(s*c)),0);

  if(j!=i){
    if(i<10) {lcd_gotoxy(0,3); sprintf(str,"|-----|"); lcd_send_string(str);}
    for(j=1;j<=i/10;j++) {lcd_gotoxy(j ,3);sprintf(str,"%c",0xff); lcd_send_string(str);}
    lcd_gotoxy(12,3); sprintf(str,"%3d%%",i); lcd_send_string(str);
  }

  b+=s;

  if(b<=a){
    //printf(">%lu\t%lu\r\n",a,b);
    d_t=make8( b/(int32)0x15180 ,0);
    h_t=make8((b-(int32)d_t*0x15180)/(int32)0xe10,0);
    m_t=make8((b-(int32)d_t*0x15180-(int32)h_t*0xe10)/(int32)0x3c,0);
    s_t=make8((b-(int32)d_t*0x15180-(int32)h_t*0xe10-(int32)m_t*0x3c),0);

    //printf("%02d/%02d %02d:%02d:%02d\r\n",mes_t,d_t,h_t,m_t,s_t);
    for(n=1;n<=5;n++) guardar(n,2);
  }
}

recupera_fin:
  lcd_clear();
  enable_interrupts(int_timer0);
}

```

## 8.6.2. Dispositivo esclavo:

```

#include <16F88.h>
#device adc=10
#fuses intrc_io,wdt,noput,mclr,brownout,nolvp,nocpd,nowrt,nodebug,noprotect,nofcmem,noieso
#use delay(clock=8000000)
#use rs232(baud=9600,parity=n,xmit=pin_b5,rcv=pin_b2,restart_wdt,timeout=10)
#include "c:\documents and settings\hero\escritorio\pic_progs\header\crc.h"

```

```

#define sensor 0x35

#define bit_oerr = 0x18.1
#define bit_cren = 0x18.4

#define trpnd pin_a6
#define trsel pin_a7
#define trled pin_b4
#define trfan pin_a4

#define fost_data pin_b1
#define fost_sclk pin_b3
#include "c:\documents and settings\hero\escritorio\pic_progs\header\ fost_02.h"

int1 flag_i=0,flag_r=0,flag_n=0;
int8 msn[8],n_msn=0;

void enviar_msn(int8,int8,int32);
void tx_rf (int8);
//-----rs232
#define int_rda
void rda_isr(void)
{
    int8 n=0,dato=0,bit=0;
    if(oerr){cren=0;delay_ms(10);cren=1;}

    if(getc()!=0x99) goto end_rda;
    for(bit=0;bit<8;bit++){
        n=getc();
        switch(n){
            case 0x86: bit_set(dato,bit); break;
            case 0x76: bit_clear(dato,bit); break;
        }
    }
    switch(dato){
        case 0x3a: if(!flag_i) {flag_i=1; n_msn=0;} else {msn[n_msn]=dato; n_msn++;} break;
        case 0x0d: if(n_msn==8 && flag_i==1) flag_r=1; else msn[n_msn]=dato; n_msn++; break;
        case 0x0a: if(n_msn==9 && flag_r==1) flag_n=1; else msn[n_msn]=dato; n_msn++; break;
        default: if(flag_i==1) {msn[n_msn]=dato; n_msn++;} break;
    }

    if(n_msn>10){n_msn=0;flag_i=0;flag_r=0;flag_n=0;}

    if(flag_n==1){
        output_high(trpnd); //pnd
        disable_interrupts(int_rda);
    }
end_rda:
    rs232_errors=0;
}
//-----timer 1
#define int_timer1
void timer1(void)
{
    set_timer1(3036); //250ms
    restart_wdt();
}
//-----main
void main(void){
    int8 temp_h=0,temp_l=0;
    int8 tension_h=0,tension_l=0;
    int8 hrel_h=0,hrel_l=0;
    int8 n=0,err=0;
    int8 crc_f=0,crc_p=0;
    int8 reg_f=0,crc_i=0;
    int16 crc_d=0,nn=0;
    int32 tension=0;

    setup_wdt(wdt_576ms);

```

```

setup_timer_1(T1_internal | T1_div_by_8);
set_timer1(3036);

setup_adc_ports(san1|san1|vss_vref);
setup_adc(adc_clock_internal);
set_adc_channel(0);

enable_interrupts(int_timer1);
enable_interrupts(int_rda);
enable_interrupts(global);

output_high(trsel); //modo receptor (high)
output_low(trpnd); //pnd
output_high(trled); //led
output_high(trfan);

start_fost02();
write_fost02(0x06); //registro status
write_fost02(0x01); // 8bits RH / 12bits T

while(true){
  if(flag_n==1){
    flag_i=0;
    flag_r=0;
    flag_n=0;
    crc_d=0;

    for(n=0;n<6;n++) crc_d=crc16(msn[n],crc_d); //obtener CRC

    if(make8(crc_d,1)==msn[6] && make8(crc_d,0)==msn[7] && msn[0]==sensor){ //revision CRC

      if(msn[1]==0x01){ //ping
        output_low(trsel); //modo transmisor
        output_low(trpnd); //pnd
        output_high(trled); //led
        delay_ms(20);
        enviar_msn('0',0x02,make32(0x00,0x00,0x00,0x00));
        output_low(trled); //led
        output_high(trsel); //modo receptor high

        output_low(trpnd); //pnd
        enable_interrupts(int_rda);
      }

      if(msn[1]==0x03){ // leer
        err=0;
registro:
        start_fost02();
        write_fost02(0x07); //registro
        reg_f= read_fost02(0);
        crc_f= read_fost02(1);
        crc_f= invcrc_fost02(crc_f);

        crc_i= invcrc_fost02(reg_f) & 0xf0;

        crc_p=crc_fost02(0x07,crc_i);
        crc_p=crc_fost02(reg_f,crc_p);

        if(crc_p!=crc_f){
          err++;
          if(err>3){
            output_low(trsel); //modo transmisor
            output_low(trpnd); //pnd
            output_high(trled); //led
            delay_ms(20);
            enviar_msn('0',0x05,make32(0x00,0x00,0x00,0x00));
            output_low(trled); //led
            output_high(trsel); //modo receptor high
          }
        }
      }
    }
  }
}

```

```

        output_low(trpnd); //pnd
        enable_interrupts(int_rda);
        reset_cpu();
    }
    start_fost02();
    write_fost02(0x1e); //soft reset
    delay_ms(20);
    goto registro;
}

err=0;
temperatura:
start_fost02();
write_fost02(0x03); //temperatura
delay_ms(300);
temp_h=read_fost02(0);
temp_l=read_fost02(0);
crc_f= read_fost02(1);
crc_f= invcrc_fost02(crc_f);

crc_p=crc_fost02(0x03,crc_i);
crc_p=crc_fost02(temp_h,crc_p);
crc_p=crc_fost02(temp_l,crc_p);

if(crc_p!=crc_f){
err++;
if(err>3){
    output_low(trsel); //modo transmisor
    output_low(trpnd); //pnd
    output_high(trled); //led
    delay_ms(20);
    enviar_msn('0',0x05,make32(0x00,0x00,0x00,0x00));
    output_low(trled); //led
    output_high(trsel); //modo receptor high

    output_low(trpnd); //pnd
    enable_interrupts(int_rda);
    reset_cpu();
}
start_fost02();
write_fost02(0x1e); //soft reset
delay_ms(20);
goto temperatura;
}
err=0;
humedad:
start_fost02();
write_fost02(0x05); //humedad
delay_ms(300);
hrel_h=read_fost02(0);
hrel_l=read_fost02(0);
crc_f= read_fost02(1);
crc_f= invcrc_fost02(crc_f);

crc_p=crc_fost02(0x05,crc_i);
crc_p=crc_fost02(hrel_h,crc_p);
crc_p=crc_fost02(hrel_l,crc_p);

if(crc_p!=crc_f){
err++;
if(err>3){
    output_low(trsel); //modo transmisor
    output_low(trpnd); //pnd
    output_high(trled); //led
    delay_ms(20);
    enviar_msn('0',0x05,make32(0x00,0x00,0x00,0x00));
    output_low(trled); //led
    output_high(trsel); //modo receptor high

```

```

        output_low(trpnd); //pnd
        enable_interrupts(int_rda);
        reset_cpu();
    }
    start_fost02();
    write_fost02(0x1e); //soft reset
    delay_ms(20);
    goto humedad;
}

tension:
    delay_ms(50);
    tension=0;

    for(nn=0;nn<500;nn++)
        tension+=(int32)read_adc(); //tension

    tension/=(int32)500;

    tension_h=make8(tension,1);
    tension_l=make8(tension,0);

    output_low(trsel); //modo transmisor
    output_low(trpnd); //pnd
    output_high(trled); //led
    delay_ms(20);
    enviar_msn('0',0x04,make32(tension_l>>4|tension_h<<4,hrel_l>>4|tension_l<<4,temp_h|hrel_l<<4,temp_l));
    output_low(trled); //led
    output_high(trsel); //modo receptor high

    output_low(trpnd); //pnd
    enable_interrupts(int_rda);
}
}
else{
    output_low(trpnd); //pnd
    enable_interrupts(int_rda);
}
}
}
}
}
//-----
void enviar_msn(int8 dir, int8 fun, int32 d32){
    int8 msn[11],n;
    int16 crc_d=0x0000;

    msn[0]= 0x3a;
    msn[1]= dir; //direccion
    msn[2]= fun; //funcion
    msn[3]= make8(d32,3);
    msn[4]= make8(d32,2);
    msn[5]= make8(d32,1);
    msn[6]= make8(d32,0);
    for(n=1;n<7;n++) crc_d=crc16(msn[n],crc_d);
    msn[7]= make8(crc_d,1);
    msn[8]= make8(crc_d,0);
    msn[9]= 0x0d;
    msn[10]=0x0a;

    for(n=0;n<10;n++) putc(0); //limpiar canal
    for(n=0;n<11;n++) tx_rf(msn[n]);
}
//-----
void tx_rf (int8 dato){
    int8 i;
    putc(0x99);
    for(i=0;i<8;i++){
        if(bit_test(dato,i)) putc(0x86); //1
        else putc(0x76); //0
    }
}

```

}