



**I
N
A
O
E**

Instituto Nacional de Astrofísica, Óptica y Electrónica

Diseño e implementación de un instrumento con inteligencia local y comunicación inalámbrica: Monitorización y decisiones simples sobre variables

Por

Rogelio Hernández Hernández

Tesis sometida como requisito parcial para obtener el grado de
Maestro en Ciencias, en la especialidad de Electrónica en el
Instituto Nacional de Astrofísica, Óptica y Electrónica
.INAOE

Supervisada por:

Dr. Jorge Francisco Martínez Carballido

Investigador Titular de la Coordinación de Electrónica del
INAOE

Febrero, 2008

Tonanzintla, Puebla.

©INAOE 2008

Derechos Reservados

El autor otorga al INAOE el permiso de reproducir
y distribuir copias de esta tesis en su totalidad o en
partes.



RESUMEN

En diferentes ámbitos podemos encontrar procesos que requieren monitorización continua de variables físicas como presión, temperatura velocidad o volumen; además se requiere activar algún sistema que modifique las condiciones del proceso.

En este trabajo de tesis se implementó un sistema inalámbrico para monitorización y actuación de variables, con acceso WEB. La comunicación inalámbrica se da entre los nodos a los que se conectan a los sensores y el nodo que envía la información a la WEB. Se desarrollaron funciones y algoritmos para comunicar a los diferentes módulos que conforman el sistema tanto inalámbricos como los que se conectan por medio de un cable, a lo largo de la tesis se realizaron diferentes pruebas para comprobar el funcionamiento de algoritmos desarrollados para la transmisión y recepción de datos; pasando por diferentes etapas, desde una comunicación simple entre dos módulos, hasta la comunicación del sistema que se monitorea y/o hacen decisiones simples sobre variables que efectúen acciones en cualquier nodo remoto. Configuración y observación de datos se hace desde un nodo WEB.

AGRADECIMIENTOS

Al Dr Jorge Francisco Martínez Carballido por asesorarme a lo largo del desarrollo de la tesis.

Al Instituto Nacional de Astrofísica Óptica y Electrónica, por todas las facilidades que me brindó para desarrollar el trabajo,

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado a través de la beca para estudios de maestría

ÍNDICE

CAPÍTULO 1 INTRODUCCIÓN	1
1.1 Objetivos	3
1.1.1 Objetivos particulares	3
1.2 Justificación.....	4
1.3 Sistema propuesto y metodología a seguir	5
CAPÍTULO 2 REDES DE SENSORES Y/O ACTUADORES	7
2.1 Antecedentes	8
2.2 El protocolo ZIGBEE	9
2.3 Tipos de dispositivos ZIGBEE	9
2.4 Capa de Red de dispositivos ZIGBEE	10
2.4.1 Topología de estrella	11
2.4.2 Topología de anillo.....	12
2.4.3 Topología de malla	13
2.5 Estándar IEEE 802.15.4	14
2.5.1 Capa PHY	14
2.5.2 Capa MAC	15
2.5.3 Trama de datos.....	15
2.6 Servidor WEB.....	16
2.7 Comunicación RS232.....	17
2.8 Sumario	18
CAPÍTULO 3 DISEÑO DEL SISTEMA.....	19
3.1 Descripción general del sistema.....	20
3.2 Elementos usados en la comunicación inalámbrica	21
3.2.1 Transductor ZMD44102	21
3.2.2 Conexión entre el CI ZMD44102 y el PIC18F452	22
3.2.3 Elementos adicionales	22
3.3 Funcionamiento de la tarjeta	23
3.3.1 Envío de datos	24
3.3.2 Recepción de un dato	25
3.4 Funcionamiento general de los nodos de comunicación inalámbrica	26
3.5 Diseño del nodo remoto	28
3.5.1 Desarrollo de funciones para el nodo remoto	30
3.5.1.1 Función de conteo de tiempo.....	31
3.5.1.2 Funciones que realiza el nodo remoto	32
3.6 Diseño de Funciones inalámbricas del nodo coordinador	32
3.7 Diseño del Nodo WEB.....	35
3.7.1 Modificaciones hechas a las funciones.....	36
3.7.2 Diseño del programa principal de la tarjeta WEB	38
3.7.3 Diseño del algoritmo para la transmisión de paquetes	38
3.8 Diseño del programa del Nodo coordinador	40

3.9	Diseño de las páginas WEB	42
3.10	Sumario	46
CAPÍTULO 4 PRUEBAS Y RESULTADOS		47
4.1	Elementos usados y configuración de los módulos inalámbricos	48
4.2	Comunicación módulo a módulo	49
4.2.1	Pruebas de envío y recepción de datos usando diferentes ejemplos	50
4.2.2	Resultados de las pruebas módulo a módulo	54
4.3	Comunicación de una red de 4 módulos	55
4.3.1	Prueba de direccionamiento de los módulos	56
4.3.2	Comunicación inalámbrica entre un nodo coordinador y tres nodos remotos	57
4.3.3	Resultados de las pruebas de redes de 4 módulos	60
4.4	Prueba de nodos inalámbricos diseñados para el sistema.....	60
4.5	Prueba de comunicación entre el módulo WEB y el módulo coordinador	64
4.6	Prueba del sistema completo	69
4.7	Sumario.....	73
CAPÍTULO 5 CONCLUSIONES Y TRABAJO FUTURO.....		74
5.1	Conclusiones.....	74
5.2	Trabajo futuro.....	75
APÉNDICE A FUNCIONES DEL NODO REMOTO.....		76
A.1	Función switch del Programa del nodo remoto	76
APÉNDICE B PROGRAMAS Y FUNCIONES IMPLEMENTADAS PARA EL MÓDULO COORDINADOR.....		78
B.1	Funciones de comunicación con el nodo remoto	78
B.1.1	Función de envío de configuración.....	78
B.2	Función de envío escritura de puertos.....	79
B.1.3	Función envío de petición de tabla de valores.....	80
B.2	Programa de prueba de funciones del nodo coordinador	81
B.3	Programa cargado al nodo coordinador.....	81
APÉNDICE C PROGRAMA DEL MÓDULO SERVIDOR.....		84
APÉNDICE D FUNCIONES PRINCIPALES DEL ARCHIVO ZMD.H.....		87
REFERENCIAS		89

ÍNDICE DE FIGURAS Y TABLAS

Figura 1.1 Diagrama a bloques del sistema propuesto.....	5
Figura 2.1 Capas de dispositivos ZIGBEE	10
Figura 2.2 Topología de estrella	12
Figura 2.3 Topología de anillo.....	12
Figura 2.4 Topología de malla	13
Figura 2.5 Trama de datos.....	16
Figura 3.1 Diagrama general del sistema	20
Figura 3.2 Tarjeta inalámbrica prototipo edición ZMD	21
Figura 3.3 Trama de datos.....	25
Figura 3.4 Diagrama de flujo en donde se ilustra el envío de datos	26
Figura 3.5 Diagrama de flujo en donde se ilustra la recepción de datos.....	27
Figura 3.6 Topología inalámbrica tipo estrella, cada nodo remoto se comunica únicamente con el nodo coordinador.....	28
Figura 3.7 Diagrama de flujo del programa del nodo remoto.....	30
Figura 3.8 Diagrama de la función de tiempo	33
Figura 3.9 Diagrama del funcionamiento de las funciones que el nodo coordinador envía al nodo remoto	34
Figura 3.10 Tarjeta prototipo SBC44EC	36
Figura 3.11 Diagrama del algoritmo de transmisión de paquetes entre módulos	39
Figura 3.12 Diagrama de la función implementada en el nodo coordinador .	41
Figura 3.13 Página WEB de inicio, se carga cuando se accede al sitio WEB.	42
Figura 3.14 Página WEB que muestra los valores analógicas de un nodo remoto.....	43
Figura 3.15 Página WEB que muestra los valores digitales de los pines de los nodos remotos	44
Figura 3.16 Página WEB donde se configura los pines del puerto A en modo digital o analógico	45
Figura 3.17 Página WEB donde se configuran a los pines de los puertos como entradas o salidas	46
Figura 4.1 Descripción de los programas usados para la primer prueba, a) módulo trasmisor b) módulo receptor	51
Figura 4.2 Representación esquemática de la transmisión serial inalámbrica	51
Figura 4.3 Diagrama de flujo del programa.....	53
Figura 4.4 Razón de paquetes enviados y paquetes recibidos.....	55
Figura 4.5 Prueba de envío de datos analógicos.....	55
Figura 4.6 Red en topología de anillo	56

Figura 4.7 Representación del arreglo en estrella que se hizo con los 4 módulos inalámbricos	58
Figura 4.8 Diagrama de flujo del programa que se implementó para la figura 4.5.....	59
Figura 4.9 Conexión del nodo remoto para probar el funcionamiento de los pines analógicos y digitales	61
Figura 4.10 Esquema de comunicación inalámbrica entre s inalámbricos....	62
Figura 4.11 Esquema de conexión para la prueba de comunicación entre el módulo coordinador el módulo WEB y una PC	65
Figura 4.12 Diagrama de flujo de las funciones de prueba para recibir datos y mostrar la información a) módulo WEB b) módulo coordinador.....	67
Figura 4.13 Página WEB donde se muestran los valores analógicos leídos de los nodos remotos.....	68
Figura 4.14 Página WEB donde se muestran los valores digitales leídos de los nodos remotos.....	68
Figura 4.15 Esquema de conexión del sistema completo.....	69
Figura 4.16 Configuración de pines digitales, imagen extraída de la página WEB de configuración analógica	71
Figura 4.17 Tabla de configuración de pines, imagen extraída de la página WEB de configuración de puertos.....	71
Figura 4.18 Valores leídos en la página WEB.....	72
Figura 4.19 Imagen de tablas de resultados obtenidas de la página WEB...	73
Tabla 2.1 Frecuencias en que opera el estándar 802.15.4.....	14
Tabla 3.1 Conexión entre pines del CI ZMD44102 y el PIC18F452.....	23
Tabla 3.2 Registros definidos en el archivo Appcfg.h	37
Tabla 3.3 Modos de configuración de los nodos remotos	44
Tabla 4.1 Comparaciones que realiza el programa cargado en la tarjeta coordinador.....	57
Tabla 4.2 Valores leídos de la pantalla de la hyperterminal.....	64

CAPÍTULO 1

INTRODUCCIÓN

En diferentes ámbitos podemos encontrar procesos que requieren monitorización continua de variables físicas como presión temperatura velocidad o volumen, además se requiere activar algún sistema que modifique las condiciones del proceso.

Existen diferentes formas de llevar a cabo la monitorización de un proceso, que van desde las simples como por ejemplo las que se hacen manualmente donde el usuario tiene que estar presente en donde se lleva a cabo el proceso, hasta sistemas más complejos que están automatizados en cierta medida, sin embargo estos últimos presentan diferentes inconveniencias que los hacen tener un costo elevado por lo que su uso está limitado a un grupo de empresas que pueden solventarla.

Los sistemas de monitorización automática realizan la comunicación entre los diferentes puntos del sistema, generalmente de forma alámbrica, lo que

eleva el costo en cuanto a materiales, recursos de instalación, tiempo y mantenimiento, esto puede también limitar la colocación de los sensores a distancias cercanas al punto en donde se recoge la información, en algunos casos para aumentar la distancia se usan repetidores, en los cuales se recibe la información y se vuelve a enviar, elevando aun más el costo, otra característica que tienen este tipo de sistemas es que para poder consultar se debe de estar en algún lugar cercano.

Una forma de contribuir a que este tipo de sistemas este al alcance de un grupo mayor de empresas es crear un sistema de monitorización que realice un cierto número de funciones básicas, pero que tenga un costo menor y por lo tanto el acceso a este tipo de sistemas sea más amplio.

En este trabajo de tesis se propone implementar un sistema de monitorización y actuación, la transmisión entre nodos de comunicación sea inalámbrica y que la información se pueda acceder desde la WEB, permitiendo al usuario hacer algún tipo de actuación, se usaran tarjetas prototipo ZMD44102 para la comunicación inalámbrica y la tarjeta prototipo SBC44EC para usarla como servidor WEB .

1.1 Objetivos

Implementar un sistema de monitorización que tenga funciones de inteligencia local, la transmisión entre puntos de comunicación sea inalámbrica y que la información se pueda acceder desde la WEB, permitiendo al usuario hacer algún tipo de actuación.

1.1.1 Objetivos particulares

- Desarrollar un sistema de demostración, para probar la correcta transmisión de datos entre las tarjetas ZMD y la tarjeta WEB.
 - Probar la transmisión de datos entre dos tarjetas inalámbricas
 - Probar la transmisión de datos por ethernet con la tarjeta SBC44

1.2 Justificación

Aunque existen diferentes sistemas de monitorización, la mayoría son de costo elevado y solo un sector de empresas pueden tener acceso a este tipo de sistemas, por esta razón se tiene la oportunidad de implementar un sistema básico de monitorización y actuación el cual nos permitirá que el sistema tenga las siguientes facilidades:

- Ampliar el acceso a soluciones de monitoreo de variables para empresas/organizaciones por tiempo y costo de instalación
- localización dentro de límites, prácticamente inmediata y de bajo costo
- Sin mantenimiento de líneas de comunicación
- Existencia de módulos de costo reducido
- Puede ampliar el número de empresas/organización que tengan acceso a este tipo de sistemas y acelerar la mejora de su competitividad
- El sistema se implementará con módulos simples que reducen/limitan la complejidad/alcance del equipo

1.3 Sistema propuesto y metodología a seguir

El sistema que se implementará mandará información de sensores conectados a nodos remotos hacia un nodo central de forma inalámbrica, el nodo central recogerá la información y la pondrá disponible en un sitio WEB, para que pueda ser consultada por medio de una PC (computadora personal).

De acuerdo a la información, el usuario podrá hacer algún tipo de actuación desde el sitio WEB. El sistema prototipo se muestra en el diagrama a bloques de la figura 1.1

Los sensores deben dar una corriente de salida de 4 a 20 mA Para conectar los sensores con los nodos remotos, se debe usar una interfaz [1], Esta parte no se tratará en este trabajo de tesis.

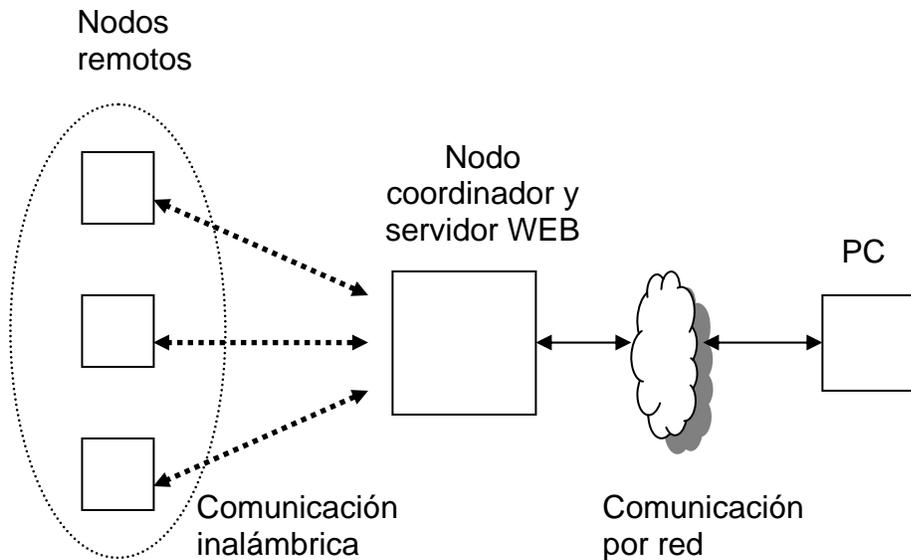


Figura 1.1 Diagrama a bloques del sistema propuesto

El primer paso será la selección de módulos que permitan transmitir datos de forma inalámbrica y un módulo que permita poner la información disponible en un sitio WEB a través de una red ethernet.

El siguiente paso consiste en diseñar e implementar algoritmos para poder enviar datos entre los módulos de forma inalámbrica, así como realizar diferentes pruebas para comprobar la correcta comunicación entre módulos.

Después diseñar, implementar y probar el módulo que funcionará como servidor WEB para asegurar que los datos se envíen y se reciban de forma correcta, así como diseñar páginas para ver los datos enviados.

Conectar el sistema completo y verificar que los datos se puedan visualizar de forma correcta, así como verificar que se pueden modificar el estado de una o más variables desde las páginas WEB.

En el capítulo dos se describen las principales características de la tecnología ZIGBEE así como los tipos de topologías inalámbricas usadas comúnmente, también se presenta el protocolo IEEE 802.15.4 que es un estándar para tecnologías inalámbricas, además se describe de forma breve lo que es un servidor WEB y el protocolo serial RS232.

En el capítulo tres se muestra el diseño de los algoritmos usados para la comunicación entre los módulos de forma inalámbrica, también se describe la forma en que se diseñaron las funciones para los nodos remotos y el nodo coordinador, también se muestra la conexión entre el módulo coordinador y el módulo servidor WEB, además se muestran las páginas en donde se pueden visualizar los datos.

En el capítulo cuatro se muestran las pruebas que se realizaron para la comunicación inalámbrica así como sus resultados, también se muestran las pruebas realizadas para comprobar la comunicación entre los módulos inalámbricos manipulando los nodos inalámbricos desde páginas WEB.

En el capítulo 5 se exponen las conclusiones y se habla acerca del trabajo futuro.

CAPÍTULO 2

REDES DE SENSORES Y/O ACTUADORES

En este capítulo se describen los tipos de redes de sensores inalámbricos que se están estudiando en la actualidad y sus antecedentes, se describen algunas topologías de red que se utilizan en este trabajo. También se describe el estandar RS-232 y sus características más importantes; además se presenta la definición de lo que es un servidor WEB, todos estos conceptos son de gran importancia para entender el contexto del trabajo desarrollado.

2.1 Antecedentes

Para el monitoreo de procesos se puede usar una red de sensores, sus funciones incluyen medir niveles de temperatura, líquido, humedad, etc. Muchos sensores tienen su propia red que se conecta con un ordenador o una caja de control a través de cables y, al detectar una anomalía, envían un aviso a la caja de controles. Se pueden clasificar en dos tipos: redes de sensores con cables y redes de sensores sin cables o inalámbricas.

Las redes de sensores inalámbricas tienen sus orígenes en aplicaciones militares, en la actualidad es un tema que se investiga ampliamente gracias a nuevos avances en la fabricación de microchips de radio, nuevas formas de ruteadores y nuevos programas informáticos relacionados con redes [2, 3].

Las redes de sensores pueden utilizar distintas tecnologías inalámbricas, incluyendo IEEE 802.11, LANs inalámbrico, Bluetooth entre otras. Actualmente se trabaja con radios de baja frecuencia con un alcance de hasta 80 metros y velocidades de hasta 300 Kb/segundo [4].

Esta clase de redes se caracterizan por su facilidad de despliegue y por ser autoconfigurables, pudiendo convertirse en todo momento en emisor, receptor, ofrecer servicios de encaminamiento (ruteo) entre nodos sin visión directa, así como registrar datos referentes a los sensores locales de cada nodo. Otra de sus características es su gestión eficiente de la energía, que les permite obtener una alta tasa de autonomía que las hacen plenamente operativas.

Entre sus principales aplicaciones se encuentran el ámbito de seguridad, al activar alarmas, también se utilizan en redes de sensores ambientales o Industriales; además de la automoción, en medicina y en la domótica.

Actualmente se desarrollan diferentes protocolos o estándares con respecto a las redes de sensores entre ellas encontramos al estándar IEEE 1451

(sensores inteligentes) [5] y al protocolo ZIGBEE [6], ambos utilizan el estándar IEEE 802.15 en sus 4 grupos, principalmente los grupos IEEE 802.15.1 [7] y IEEE 802.15.4 [8].

En este trabajo de tesis se usa el protocolo ZIGBEE como modelo de referencia para desarrollar el sistema planteado.

2.2 El protocolo ZIGBEE

ZIGBEE es el estándar basado en la tecnología necesaria para el control remoto de sensores/actuadores que se utilizan generalmente en aplicaciones de automatización de diversos procesos.

Fue creado para cubrir la necesidad del mercado de un sistema a bajo costo, un estándar para redes inalámbricas de pequeños paquetes de información, bajo consumo de energía, seguro y fiable [10].

En la figura 1, aparece la estructura de la arquitectura en capas. Las primeras dos capas, la física (PHY) y la de acceso al medio (MAC), son definidas por el estándar **IEEE 802.15.4**. Las capas superiores son definidas por la Alianza ZIGBEE [11].

La distancia de transmisión puede variar desde los 10 metros hasta los 75, dependiendo de la potencia de transmisión, antena y del entorno.

2.3 Tipos de dispositivos ZIGBEE

ZIGBEE tiene tres tipos de dispositivos: El *coordinador de red*, que mantiene en todo momento el control del sistema. Es el más sofisticado de los tipos de dispositivos, requiere memoria y capacidad de computación.

La trama general de operaciones (GOF) es una capa que existe entre la de aplicaciones y el resto de capas. La GOF suele cubrir varios elementos que son comunes a todos los dispositivos, como el subdireccionamiento y los modos de direccionamientos y la descripción de dispositivos, como el tipo de dispositivo, potencia, modos de 'dormir' y coordinadores de cada uno. Utilizando un modelo, la GOF especifica métodos, eventos, y formatos de datos que son utilizados para constituir comandos y las respuestas a los mismos.

La capa de red soporta múltiples configuraciones, a continuación se describen las principales.

2.4.1 Topología de estrella

Consiste en conectar varios nodos a un nodo central que funciona como coordinador, de esta forma, cuando un nodo envía información al nodo central, este la recibe y la puede reenviar a otro nodo, tiene la ventaja de que si un nodo externo falla la red no sufrirá gran cambio, pero si el nodo central falla entonces toda la red fallará, la configuración de estrella se muestra mediante la figura 2.2.

En la configuración en estrella, uno de los dispositivos tipo FFD asume el rol de coordinador de red y es responsable de inicializar y mantener los dispositivos en la red. Todos los demás dispositivos ZIGBEE, conocidos con el nombre de dispositivos finales, 'hablan' directamente con el coordinador.

Esta topología presenta una desventaja en el caso de redes conectadas con cables, ya que aumenta el costo de cableado debido a que cada dispositivo debe de estar conectado al dispositivo central, para el caso de redes inalámbricas, no se asume este costo.

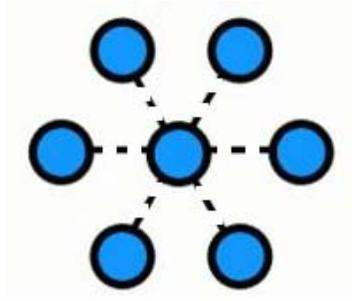


Figura 2.2 Topología de estrella

2.4.2 Topología de anillo

Topología de red en la que las estaciones se conectan formando un anillo. Cada estación está conectada a la siguiente y la última está conectada a la primera. Como se muestra en la figura 2.3

Cada estación tiene un receptor y un transmisor que hace la función de repetidor, pasando la señal a la siguiente estación del anillo. Cabe mencionar que si algún nodo de la red falla comunicación en todo el anillo se pierde, también si se hacen cambios en la red, como aumentar el número de nodos, también se puede causar que toda la red falle.

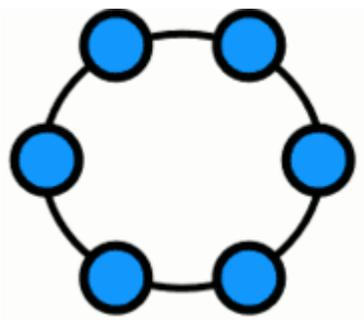


Figura 2.3 Topología de anillo

2.4.3 Topología de malla

Es una topología en la que cada nodo está conectado a uno o más de los otros nodos. Esta configuración ofrece caminos redundantes por toda la red, de modo que si falla una conexión, otra conexión se hará cargo del tráfico de datos. Esta topología, a diferencia de otras no requiere de un servidor o nodo central, con lo que se reduce el mantenimiento, un error en un nodo, sea importante o no, no implica la caída de toda la red. Las redes de malla son auto-regenerables, la red puede funcionar incluso cuando un nodo desaparece o la conexión falla, ya que el resto de nodos evitan el paso por ese punto. Las redes implementadas con cables tienen un costo alto de material, mano de obra y tiempo en su instalación, ya que utilizan mucho cableado. Por ello cobran mayor importancia el uso de redes inalámbricas, al no haber necesidad de cableado, aun que su complejidad aumenta, la topología de malla se muestra en la figura 2.4.

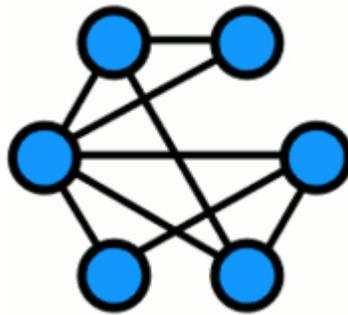


Figura 2.4 Topología de malla

2.5 Estándar IEEE 802.15.4

El estándar IEEE 802.15.4 [10] define un protocolo para la interconexión de dispositivos electrónicos en una red de área personal inalámbrica conocidas como WPAN por sus siglas en inglés, fue diseñado para bajo consumo de energía, y baja velocidad de transmisión en un rango de 20-250 kbps y corto alcance de 10 a 75 metros.

El estándar 802.15.4 usa tres bandas de frecuencias que se presentan en la tabla 2.1

Banda de frecuencias (MHz)	Canales	Región	Modulación	Velocidad de transmisión
868-868.6	1	Europa	BPSK	20
902-928	10	América, Japón	BPSK	40
2400-2483.5	16	Mundial	Q-QPSK	250

Tabla 2.1 Frecuencias en que opera el estándar 802.15.4

El estándar define el nivel físico (capa PHY) y el control de acceso al medio (capa MAC) de las redes inalámbricas

2.5.1 Capa PHY

La capa física contiene el transductor de radio frecuencia (RF) y se encarga de las siguientes tareas:

- Activación y desactivación del radio transductor
- Detección de energía (ED) dentro del canal actual

- Indicación de la calidad de enlace de paquetes recibidos
- Evaluación de limpieza del canal para CSMA-CA
- Selección de la frecuencia del canal

2.5.2 Capa MAC

La capa MAC maneja todos los accesos al canal de radio físico y habilita la transmisión y recepción de la MPDU a través del servicio de datos físicos.

Se encarga de las siguientes tareas:

- Genera redes “beacon” si el dispositivo es un coordinador
- Sincroniza los “beacons”
- Soporta la asociación y des-asociación PAN
- Soporta la seguridad del dispositivo
- Emplea el mecanismo CSMA-CA accesos de canal
- Maneja y mantiene el mecanismo GTS (Guaranted Time slot)
- Provee un nexo confiable entre dos entidades MAC conectadas en configuración punto-punto

2.5.3 Trama de datos

En la figura 2.5 se ilustra la trama completa de datos.

El campo donde se colocan los datos es referido como MSDU (MAC Service Data Unit), este campo se prefija con un encabezado llamado MHR (MAC Header) y se le añade un campo final conocido como MFR (MAC Footer), la MHR contiene el control de trama número de secuencia y campos de información de direccionamiento, la MFR se compone de un FCS (Frame Check Sequence) de 16 bits. La MHR, la MSDU y la MFR forman la trama

de datos MAC, esta pasa a ser la MPDU (MAC Protocol Data Unit) que se usa en la capa PHY.

La MPDU se pasa a la capa PHY como la trama física de carga de datos. La PSDU (PHY Service Data Unit) se prefija con un SHR (Synchronization Header), el cual contiene la secuencia de preámbulo y un PHR (PHY Header), el cual contiene la longitud de la PSDU en bytes de 8 bits, la secuencia de preámbulo y el delimitador de inicio de trama habilitan al receptor para lograr sincronización simbólica, la SHR, PHR y PSDU forman el paquete de datos físicos.

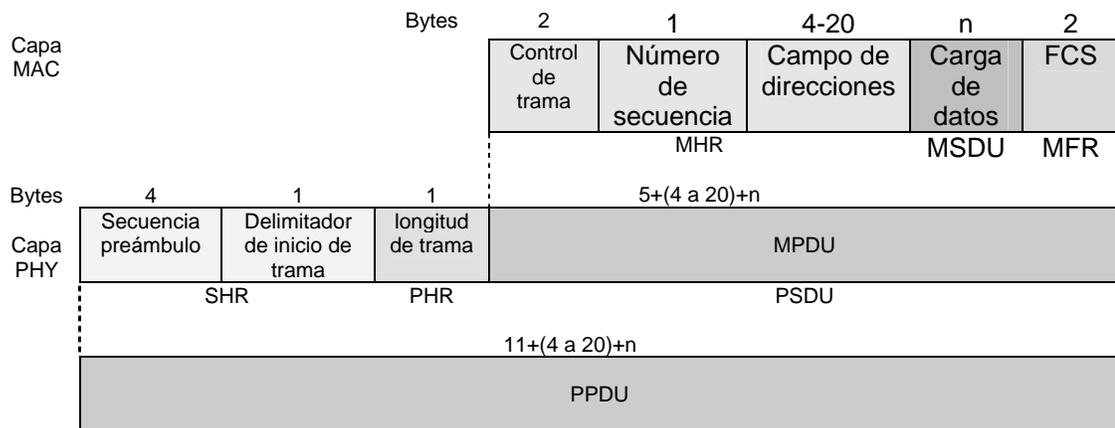


Figura 2.5 Trama de datos

2.6 Servidor WEB

El Servidor WEB es el dispositivo que pone la información disponible en la WEB.

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al dispositivo físico (como una computadora) en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos [12, 13].

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor WEB, este término podría referirse a la máquina que almacena y maneja los sitios WEB, y en este sentido es utilizada por las compañías que ofrecen hospedaje. Alternativamente, el servidor WEB podría referirse al software, como el servidor de http de Apache [14], que funciona en la máquina y maneja la entrega de los componentes de los páginas WEB como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor sirve información a las computadoras que se conecten a él, por medio de un cliente explorador http. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

En la WEB, un servidor WEB es un ordenador que usa el protocolo http para enviar páginas WEB a la PC de un usuario cuando el usuario las solicita.

Los servidores WEB, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet.

Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que una misma PC puede tener diferentes programas de servidor funcionando al mismo tiempo.

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica [12, 13, 14].

2.7 Comunicación RS232

Ante la gran variedad de equipos, sistemas y protocolos que existen surgió la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes

comunicarse entre si. La EIA (Electronics Industry Association) elaboró la norma RS-232, la cual define la interfaz mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

El envío de niveles lógicos (bits) a través de cables o líneas de transmisión necesita la conversión a voltajes apropiados. En los microcontroladores para representar un 0 lógico se trabaja con voltajes inferiores a 0.8v, y para un 1 lógico con voltajes mayores a 2.0V.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el puerto serial de la PC, ya que son diferentes a los que utilizan los microcontroladores y los demás circuitos integrados. Por lo tanto se necesita de una interfaz que haga posible la conversión de los niveles de voltaje a los estándares manejados por los CI TTL. Para mayor información consultar la norma TIA/EIA-232. [16].

El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas (normalmente, 9600 bits por segundo o más). Tanto el aparato a conectar como la PC (o el programa terminal) tienen que usar el mismo protocolo serie para comunicarse entre si. Puesto que el estándar RS-232 no permite indicar en qué modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes.

2.8 Sumario

En este capítulo se han descrito de forma breve la tecnología ZIGBEE, se describen algunas topologías de red que se utilizan en este trabajo. También se describe el protocolo RS-232 y sus características más importantes; además se presenta la definición de lo que es un servidor WEB.

En el capítulo 3 se presenta el diseño de algoritmos y funciones para lograr la comunicación entre los módulos inalámbricos y el servidor WEB.

CAPÍTULO 3

DISEÑO DEL SISTEMA

En este capítulo se presenta el diseño de cada una de las partes que conforman el sistema, se describe cada una de las tarjetas que se usaron para implementar los diferentes módulos, las funciones que se desarrollaron para hacer funcionar la comunicación entre los nodos inalámbricos, así como las funciones desarrolladas para el envío de datos a la red ethernet; además de las páginas que se usan para visualizar la información.

3.1 Descripción general del sistema

El sistema consta de un nodo central integrado por un *módulo coordinador* y un *módulo servidor WEB*; además de tres nodos remotos como se muestra en la figura 3.1

Los nodos remotos se comunican de forma inalámbrica con el módulo coordinador, mientras que el módulo coordinador y el módulo servidor se comunican entre ellos usando la interfaz RS232. El módulo servidor WEB se conecta a una red ethernet.

El sistema recaba información de sensores conectados a los nodos remotos, esta información se puede visualizar en una página WEB que se encuentra alojada en el módulo servidor.

Para esto los nodos remotos mandan la información de manera inalámbrica al módulo coordinador, este a su vez envía la información de forma serial al módulo servidor, el cual se conecta a una red ethernet para poder visualizar la información desde una PC conectada a la red a través de páginas WEB.

El sistema también puede manipular señales digitales de encendido y apagado de actuadores conectados a los nodos remotos, desde la página WEB.

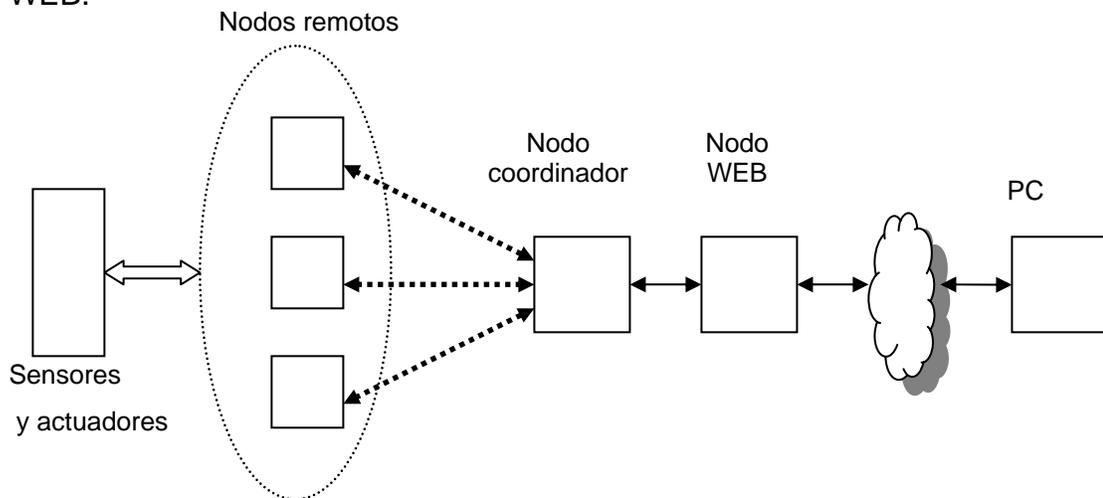


Figura 3.1 Diagrama general del sistema

3.2 Elementos usados en la comunicación inalámbrica

Para realizar la comunicación inalámbrica entre los diferentes módulos se utilizaron “tarjetas prototipo para comunicación inalámbrica edición ZMD” (figura 3.2).

Las tarjetas ZMD [17] permiten el desarrollo de dispositivos inalámbricos que cumplen con el estándar IEEE 802.15.4. El corazón de la tarjeta inalámbrica ZMD es un pequeño módulo que contiene el transductor ZMD44102. Para controlar y hacer funcionar al módulo ZMD44102, se incluye un microcontrolador PIC18LF452 [18], también se incluye en la tarjeta prototipo elementos adicionales que son controlados por el PIC18F452, con los que se puede realizar diversas pruebas.



Figura 3.2 Tarjeta inalámbrica prototipo edición ZMD

3.2.1 Transductor ZMD44102

El ZMD44102 es un transductor RF integrado en un chip, es un dispositivo que cumple con las especificaciones del protocolo IEEE 802.15.4, el cual se usa como base para aplicaciones ZIGBEE, opera bajo licencia libre europea de 868 MHz y en una banda multicanal Americana que va de 902 MHz hasta

928 MHz las características de gran rango y penetración de barrera del módulo ZMD44102 optimiza la transmisión de datos arriba de 40 kbps. El transductor asegura la transmisión de datos en ambientes difíciles, el nivel de integración del ZMD44102 incluye un controlador de acceso al medio (MAC), reduciendo el costo de componentes externos y sistemas de instalación [19]. Entre sus principales aplicaciones se encuentran la de redes de sensores inalámbricas, automatización de procesos industriales, así como el monitoreo de procesos que representan algún riesgo.

3.2.2 Conexión entre el CI ZMD44102 y el PIC18F452

La tarjeta prototipo ZMD contiene un Microcontrolador **PIC18F452** cuya función es la de configurar y controlar las diferentes funciones del módulo ZMD44102 para la transmisión/recepción inalámbrica de datos; además, sirve para controlar a los dispositivos adicionales incluidos en la tarjeta Prototipo edición ZMD y facilita la comunicación con módulos externos como PCs u otras tarjetas prototipos a través de su puerto serial RS232.

La conexión de los pines del módulo ZMD44102 con los del PIC18F452 se muestran en la tabla 3.1

3.2.3 Elementos adicionales

La tarjeta incluye un potenciómetro conectado en el pin RA0 (AN0), un botón momentáneo conectado en el pin RA4, tres leds conectados en los pines RB4, RB5 y RA5 respectivamente. La comunicación serial se hace con los pines RC6 para la recepción y RC7 para la transmisión a través de un controlador RS232.

Para poder desplegar valores numéricos, la tarjeta incluye 3 dígitos de 7 segmentos controlados por los pines RB2, RB4 y RB5, a través de 3 circuitos integrados 74HC595 conectados en cascada.

Señal	PIN CI ZMD 44102	PIN PIC18F452
Address Latch Enable	ALE	E2
Read Data	RD	E0
Write Data	WR	E1
Data	D0	D0
	D1	D1
	D2	D2
	D3	D3
	D4	D4
	D5	D5
	D6	D6
	D7	D7
Slave Select	SS	C2
Slave In	SI	C4
Slave Out	SO	C5
CLK	SCLK	C3
Interrupt Request	IRQ	B0
Global Power Down	GPD	B1
Chip Reset	RSN	C1

Tabla 3.1 Conexión entre pines del CI ZMD44102 y el PIC18F452

3.3 Funcionamiento de la tarjeta

El envío y recepción de paquetes de bytes de forma inalámbrica es un aspecto importante para tener idea de cómo se llevan a cabo los dos procesos y así poder tener más claro la forma de comunicar a los diferentes nodos.

3.3.1 Envío de datos

Para enviar una trama de datos se usan las funciones MAC implementadas en la librería ZMD44102.H [19].

Otras variables referentes al direccionamiento son:

- IntraPan
- Modo de dirección destino
- El bit de habilitación de seguridad FCS
- El bit de requerimiento de confirmación

El CI ZMD44102 tiene un registro TX FIFO que puede usarse en dos formas. En el primero, el método estándar usa al TX FIFO para guardar al MSDU y el ZMD44102 genera todo el resto del encabezado MAC dependiendo del encabezado de varios registros, como el mhrFc1Tx y el mhrDstAddr16.

La segunda forma requiere que el usuario genere un encabezado MAC y lo coloque dentro del TX FIFO. El FIFO a transmitir para el ZMD44102 es de 128 bytes.

Una vez que el FIFO a transmitir está lleno, la transmisión se inicializa poniendo el control de registro operacional TxOn a 0x03 si CSMA está habilitada, el ZMD44103 primero enlistará el tráfico de canal, por un tiempo corto para determinar si el canal está ocupado, y si el canal está ocupado, se pondrá a la radio en un estado de espera y se pondrá al registro MacTxStatus para denotar un error CSMA. Si el CSMA fue deshabilitado o no hay tráfico de canal, éste transmitirá la trama y pondrá al registro MacTxStatus como transmisión llevada a cabo. Si se hizo una petición de confirmación (Acknowledge, ACK), en el periodo de tiempo, el registro MacRxStatus será activado para esperar la confirmación por un tiempo corto [6, 19, 20].

El manejo de los registros del módulo ZMD44102 para la transmisión de datos se llevan a cabo con el PIC18F452, las funciones para realizar el proceso se encuentran en la referencia [19], En la figura 3.3 se da un ejemplo de cómo se llena una trama de datos que será enviada, mientras que en la figura 3.4 se ilustra con un diagrama de flujo el proceso.

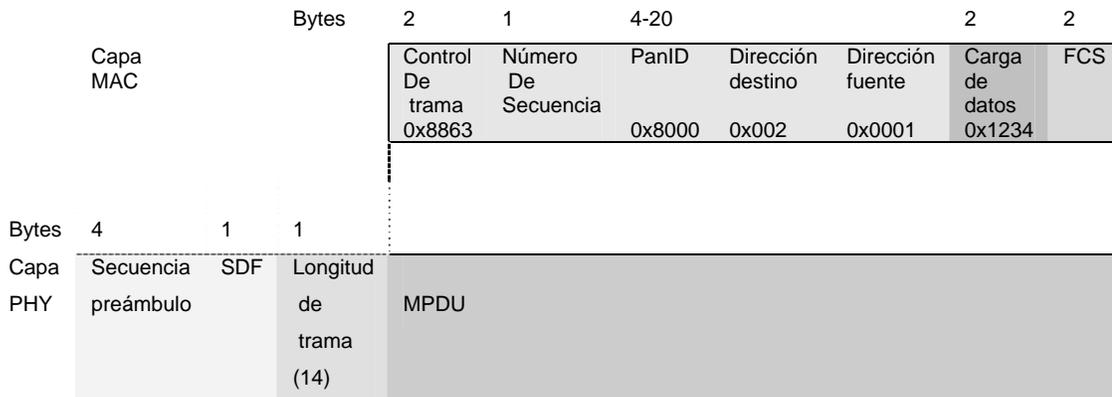


Figura 3.3 Trama de datos

3.3.2 Recepción de un dato

Las funciones para la recepción de datos se encuentran implementadas en la librería ZMD44102.H [19].

El modo de recepción puede ser activado poniendo el registro de control operacional MacControl a RxOn (0x04) o en el modo de transmisión para recibir un ACK, el contenido del registro MacRxStatus se adquiere si se detecta la entrada de una trama de datos, esta se guarda en el registro Rx FIFO, cuando la recepción es correcta se indica a través de una interrupción, a su vez se manda un ACK si este fue requerido por el módulo transmisor, para indicar que se recibió el paquete de datos. En la figura 3.5 se ilustra el proceso.

El manejo de los registros del módulo ZMD44102 para la recepción de datos se llevan a cabo con el PIC18F452, las funciones que se requieren se encuentran en la referencia [19, 20].

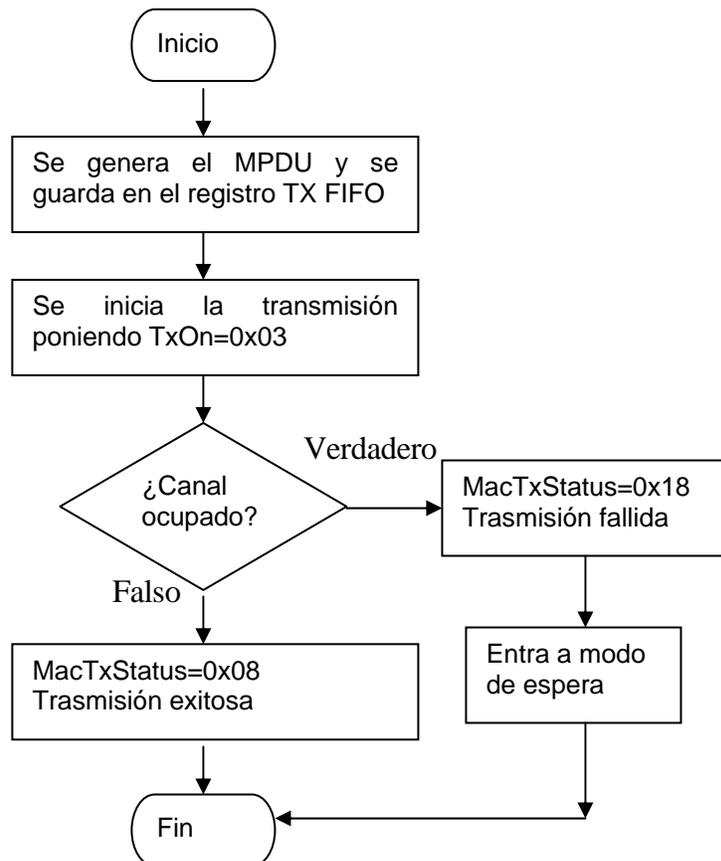


Figura 3.4 Diagrama de flujo en donde se ilustra el envío de datos

3.4 Funcionamiento general de los nodos de comunicación inalámbrica

La comunicación inalámbrica se establece entre un nodo coordinador y tres nodos remotos como se muestra en la figura 3.6, se establece una topología tipo estrella donde el nodo coordinador se comunica con cada uno de los nodos enviando y recibiendo información

Cada nodo tiene una dirección asignada, para este caso al nodo coordinador se le asignó la dirección 1 mientras que a los tres nodos remotos se les asignó las direcciones 2, 3, y 4 respectivamente.

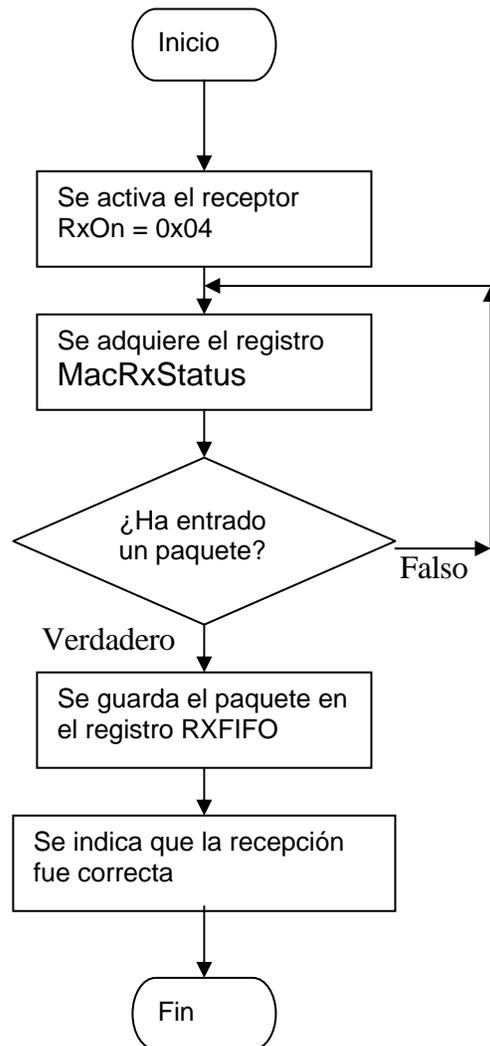


Figura 3.5 Diagrama de flujo en donde se ilustra la recepción de datos

Cada nodo se configura indicando la dirección del nodo coordinador, la dirección fuente, la dirección destino, e inicializando, además el canal que se va a usar para la comunicación.

Para tener una comunicación eficaz el nodo coordinador envía paquetes de bytes que sirven como instrucciones para el nodo remoto, este recibe los paquetes y ejecuta las funciones indicadas en los paquetes.

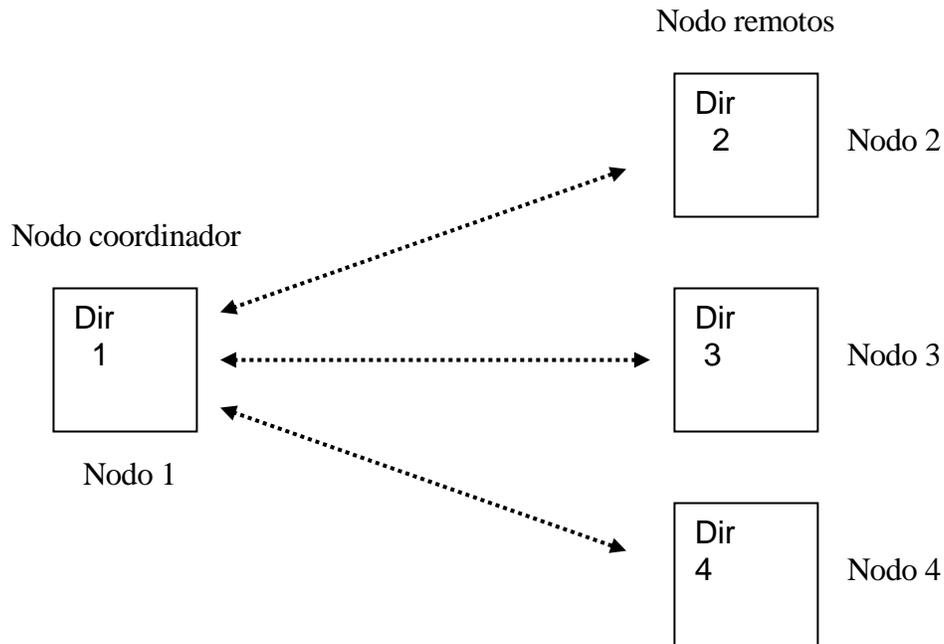


Figura 3.6 Topología inalámbrica tipo estrella, cada nodo remoto se comunica únicamente con el nodo coordinador

3.5 Diseño del nodo remoto

Debido a que se quiso un solo tipo de nodo remoto, se desarrolló un solo diseño para los tres nodos remotos.

El nodo remoto tiene la función de adquirir el estado de los pines definidos como entradas cada segundo y enviar la información al nodo coordinador, así como el de modificar el estado de los pines definidos como salidas de acuerdo a la información que reciba del nodo coordinador.

La tarjeta tiene 4 pines que se pueden usar para adquirir variables analógicas o digitales dependiendo de la configuración que tengan, estos pines son A0,

A1, A2, y A4, también tiene otros pines que únicamente funcionan de forma digital, estos son B2, B3, B4, B5, B6, B7

La tarjeta edición ZMD tiene 3 leds conectados a los pines B4, B5 y A5, para ser habilitados, por esa razón se configuraron a estos pines como salidas, mientras que el pin A4 se encuentra conectado a un botón momentáneo por lo que el pin A4 se configuró para que siempre esté como entrada, la tarjeta también cuenta con un potenciómetro conectado al pin A0 por lo que se configuró para que siempre esté como entrada.

El funcionamiento del nodo remoto inicia configurando los valores para el correcto funcionamiento de la rutina de interrupción, así como la configuración del valor de las direcciones que se usarán para la comunicación inalámbrica, también se establecen valores iniciales para los puertos y la configuración de estos además de configurar el ADC del PIC, también se inicializa la función para la recepción de paquetes de información.

Una vez configurada la tarjeta se inicia una función recurrente para detectar si hay algún paquetes en el buffer de recepción, si es así se usa una instrucción “switch case” para determinar la función que se debe ejecutar.

Si no se detecta la entrada de un paquete se determina si ha habido algún cambio en el contador de segundos, si es así se obtiene el estado de los pines que les corresponde en ese instante de tiempo.

Si se determina que no ha habido cambio en el contador de segundos se vuelve al inicio de la función recurrente para detectar la entrada de un paquete, este proceso se describe en el diagrama de flujo de la figura 3.7.

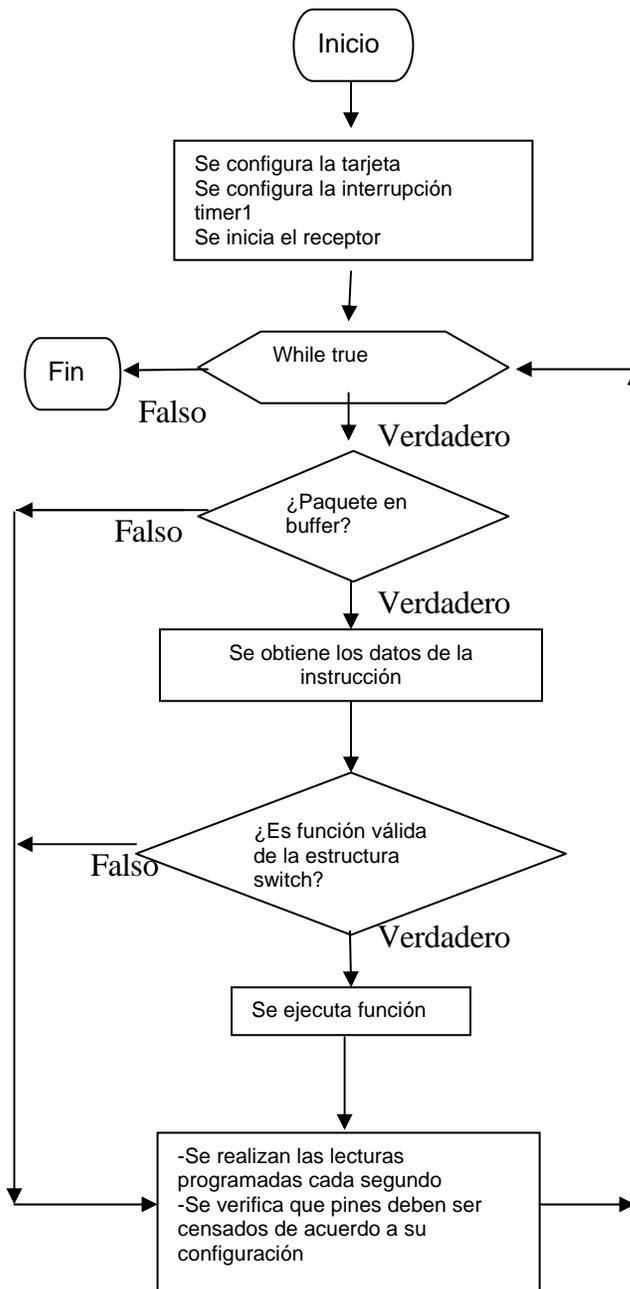


Figura 3.7 Diagrama de flujo del programa del nodo remoto

3.5.1 Desarrollo de funciones para el nodo remoto

El nodo remoto tiene la tarea de adquirir o cambiar el estado de los pines de acuerdo a la información que le llega desde el nodo coordinador, además de

contar el tiempo en segundos. Para esto se desarrollaron funciones específicas para cada tarea, el nodo remoto ejecuta diferentes funciones dependiendo del caso que se dé.

3.5.1.1 Función de conteo de tiempo

La función de conteo de tiempo se activa cuando hay una interrupción. Para implementar la rutina de interrupción de tiempo se usó el registro timer1 del PIC18F452 configurado en modo de 16 bits y pre-escala de 8.

Para obtener el máximo lapso posible de interrupción del timer1 se usó la fórmula [21] $Temporizacion = T_{cm} \cdot Preescaler \cdot V_{TMR1}$

Donde *Temporización* es el tiempo de interrupción deseado, T_{cm} es el período de un ciclo de máquina, igual al inverso de la frecuencia de oscilación sobre cuatro, *Preescaler* -Es el rango de divisor de frecuencia elegido, V_{TMR1} es el número total de impulsos a contar por el TMR1 antes de desbordarse en la cuenta ascendente.

Los valores que se usaron para obtener el valor de temporización máxima son

Para 10 MHz se tiene un $T_{cm} = 0.2 \text{ } \mu\text{S}$, el preescaler = 8 y $V_{TMR1} = 65536$

Con esto se obtuvo que la temporización máxima fuera de 0.2097152 segundos.

Para obtener el segundo exacto se determinó establecer el tiempo de interrupción en 0.2 segundos y contarlo 5 veces.

Estableciendo los valores en la fórmula

$$Temporizacion = 0.2 \text{ S} = 0.4 \text{ } \mu\text{S} \cdot 8 \cdot V_{TMR1}$$

Despejando V_{TMR1} se obtiene que el número de impulsos que el timer1 tiene que contar antes de desbordarse es de 62500, por lo que el valor que se estableció en el registro timer1 fue de $(65536 - 62500) = 3036$.

La función que se usa para contar el tiempo se describe en el diagrama de la figura 3.4, primero se espera a que se de una interrupción en un tiempo de 200 ms y si es la quinta que se da, se determina que ha pasado un segundo, por lo que el contador de segundos se incrementa. El diagrama de flujo de la función se ilustra en la figura 3.8.

3.5.1.2 Funciones que realiza el nodo remoto

Las funciones que el nodo remoto realiza para la configuración, lectura y escritura de puertos se encuentran dentro de una instrucción “switch case”. Cuando se detecta la entrada de un paquete se usa el primer byte del paquete de datos para usarlo como argumento de la instrucción “switch case” y de esa forma seleccionar la función que se debe de ejecutar.

A continuación se describe la función que corresponde a cada caso.

- Case 1: Esta función configura como entradas o salidas los pines de los puertos del PIC además de configurar como analógicos o digitales a los pines del puerto A
- Case 2: Esta función escribe el valor de cada pin de los puertos a y b
- Case 3: Esta función envía una tabla de datos al nodo remoto

Estas funciones se pueden consultar en el apéndice A.1

3.6 Diseño de Funciones inalámbricas del nodo coordinador

Para facilitar el intercambio de datos, se implementaron funciones a manera de instrucciones, el nodo coordinador envía las instrucciones y los nodos remotos las ejecutan regresando un paquete de datos para confirmar que la instrucción se llevó a cabo.

Las funciones tienen una forma general, primero se envía un paquete de 8 bytes de datos, el primer byte corresponde al número de la función que el nodo remoto debe ejecutar, los demás corresponden a valores necesarios

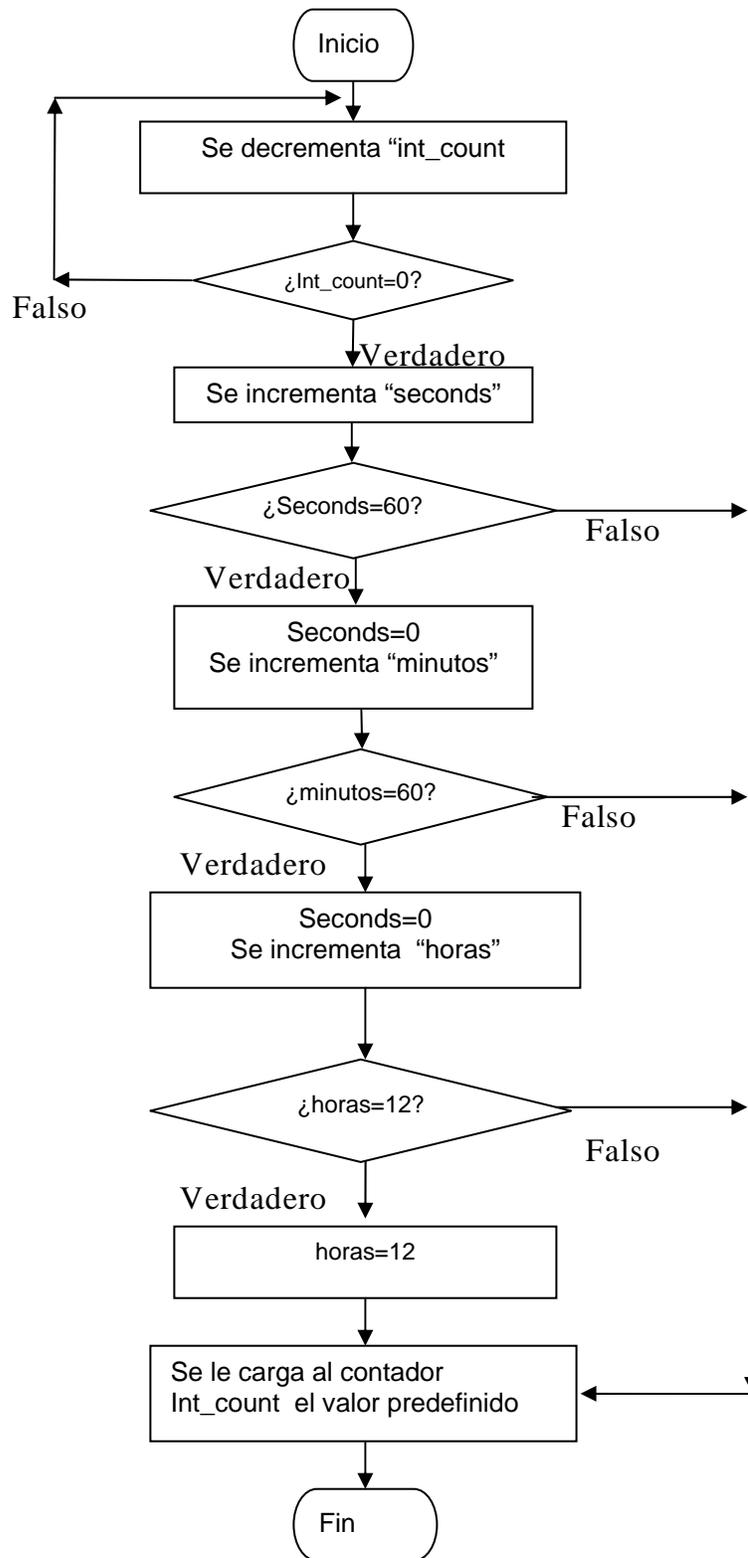


Figura 3.8 Diagrama de la función de tiempo

para ejecutar la función, después se recibe un paquete de datos para confirmar que la función se ejecutó correctamente, en el caso de que el paquete de datos no haya sido recibido por el nodo remoto entonces esta función regresa un 0 al programa principal, en el caso de que no se reciba la confirmación de que la función fue ejecutada, se regresa un 1 al programa principal, mientras que si toda la función se llevó a cabo correctamente se regresa un 2, este proceso se ilustra en el diagrama de flujo de la figura 3.9

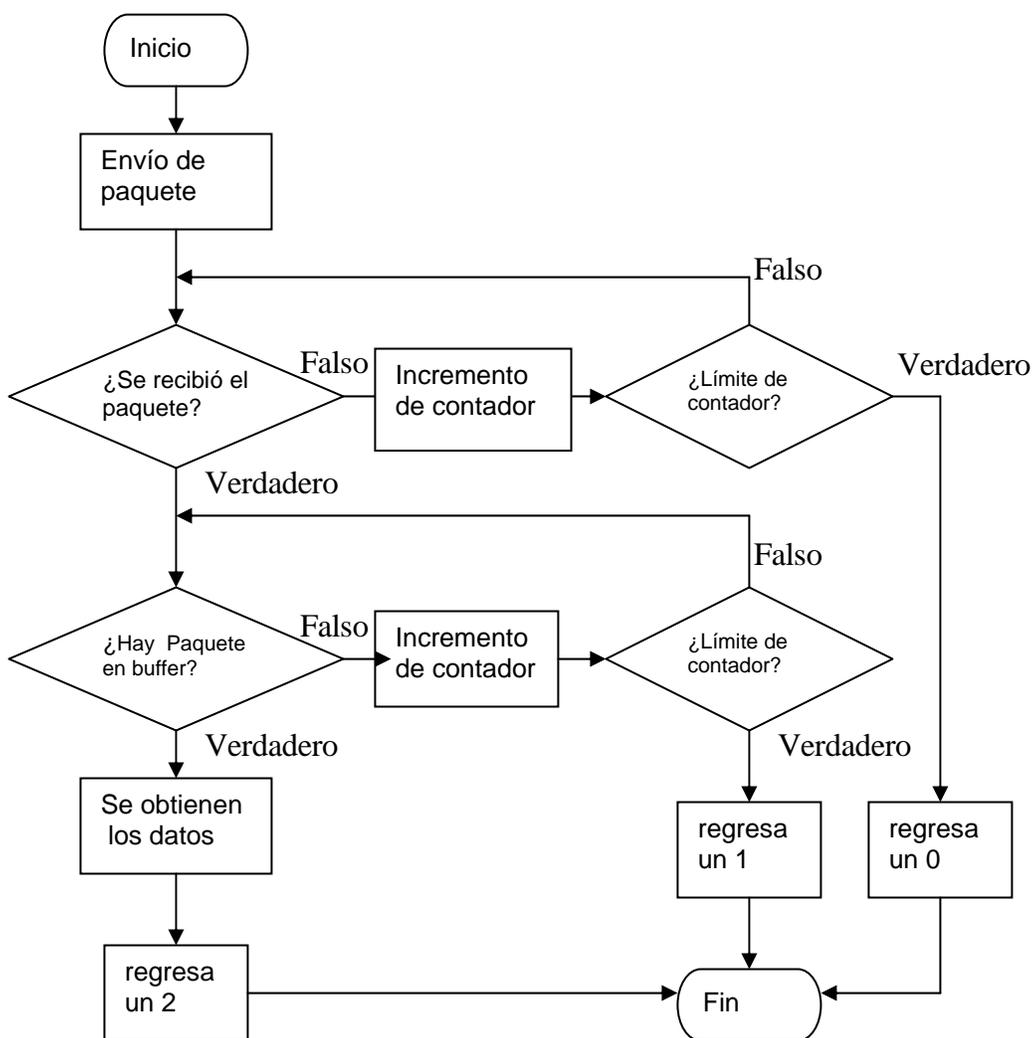


Figura 3.9 Diagrama del funcionamiento de las funciones que el nodo coordinador envía al nodo remoto

Las funciones se muestran en los apartados del apéndice B.1

Se hizo una instrucción para enviar los datos necesarios para configurar los pines del nodo remoto, el nombre y los argumentos de la función son

“conf_nr(int tris_a,int tris_b,int adcon1_r,ZMAC_NODE_INFO txNode)”.

Donde los 3 primeros argumentos son los que usa el nodo remoto y el último argumento es el nodo remoto al que se le envía el paquete de datos.

También se hizo una función para enviar el valor que deben de tenerlos pines del nodo remoto configurados como salidas.

Su sintaxis y sus argumentos son

“write_nr(int port_a,int port_b,ZMAC_NODE_INFO txNode)”

Donde los primeros dos argumentos son para escribir los pines de los puertos del nodo remoto y el tercer argumento es el nodo al que se le envía el paquete de datos.

Se hizo una función para obtener una tabla del estado de los pines del nodo remoto, su nombre y sus argumentos son

“read_nr(ZMAC_NODE_INFO txNode,int dir_ini)”

Su único argumento es número de nodo del que se obtendrán los datos.

3.7 Diseño del Nodo WEB

El nodo WEB se implementó usando la tarjeta prototipo SBC44EC [22] (figura 3.10) cuenta con una interfaz ethernet de 10 Mbs, y una interfase serial rs232, sus funciones son llevadas a cabo por medio de un microcontrolador PIC18F452, también se incluye una memoria externa EEPROM 24LC256 [23] de 32 Kbytes que permite guardar las páginas WEB del sistema

La función del módulo WEB es la de comunicarse con el nodo coordinador y con una PC a través de una red Ethernet.



Figura 3.10 Tarjeta prototipo SBC44EC

Las funciones del PIC para controlar el envío y recepción de datos hacia la red ethernet se encuentran desarrolladas en la referencia [22].

Para poder usar la tarjeta como módulo servidor del sistema propuesto se hicieron algunas modificaciones a las funciones del programa del PIC de la tarjeta.

Las funciones fueron diseñadas para la lectura y escritura de los puertos de la tarjeta SBC44EC, y mostrar los valores en una página WEB alojada en la memoria eeprom, se hizo una adaptación para que la tarjeta mostrara los datos leídos de los puertos de los nodos remotos.

El sitio WEB definido para el sistema consta principalmente de 4 páginas, dos para configurar a los puertos de los nodos remotos, una para leer el estado de los puertos digitales y una para leer el valor de los pines definidos como analógicos.

3.7.1 Modificaciones hechas a las funciones

Las funciones que se encuentran en los archivos de la referencia [4], acceden los registros relacionados a los puertos de la tarjeta para leerlos o

modificarlos y enviar el estado de los registros a la red ethernet para ser visualizados en una página WEB.

En el archivo Appcfg.h de la referencia 22, se definieron tres grupos de bytes para guardar la información de los puertos de los nodos remotos, el primer grupo se definió para contener la configuración de los pines de los puertos de los nodos remotos, el segundo se definió para guardar el estado de los pines de los puertos definidos como entradas, el tercer grupo se definió para guardar el estado de los pines definidos como salidas.

Como se muestra en la tabla 3.2

Grupo 1	Grupo 2	Grupo 3
AppConfig.PortCfg.trisa	AppConfig.PortRw.porta	AppConfig.LatRw.lata
AppConfig.PortCfg.trisb	AppConfig.PortRw.portb	AppConfig.LatRw.latb
AppConfig.PortCfg.trisc	AppConfig.PortRw.portc	AppConfig.LatRw.late
AppConfig.PortCfg.trisd	AppConfig.PortRw.portd	AppConfig.LatRw.latd
AppConfig.PortCfg.trise	AppConfig.PortRw.porte	AppConfig.LatRw.late
AppConfig.PortCfg.trisf	AppConfig.PortRw.portf	AppConfig.LatRw.latf

Tabla 3.2 Registros definidos en el archivo Appcfg.h

En el archivo Appcfg.c se definió la configuración del registro TRISB correspondiente al PIC de la tarjeta WEB con un valor constante de tal forma que los pines B0, B1, B2, B3y B4 siempre sean salida y el pin B5 siempre sea entrada.

En el archivo Htttexec.c se sustituyeron los nombres de los registros relacionados a los puertos por el nombre de los registros definidos en la tabla 4.3, con el fin de poder enviar el estado de los puertos de los nodos remotos a la página WEB.

3.7.2 Diseño del programa principal de la tarjeta WEB

En el archivo mxwebsrvr.c se implementó el programa que controla el flujo de datos entre la tarjeta WEB. El envío y recepción de datos entre el módulo WEB y el módulo coordinador se hizo de forma serial RS232.

Se implementó un algoritmo para realizar de forma correcta la transmisión serial entre los dos módulos.

Primero se definieron paquetes de ocho bytes en cada módulo, donde el primero y el último definen el inicio y el final del paquete respectivamente, el segundo byte indica el número del paquete y los bytes restantes son datos que se envían.

El módulo WEB envía los paquetes de acuerdo a la página WEB que se haya accedido desde una PC. De esta forma cuando se accede la página de valores analógicos se envía el paquete denominado como insta, cuando se accede la página de valores digitales se envía el paquete instb y cuando se accede alguna de las páginas de configuración se envía el paquete instc.

3.7.3 Diseño del algoritmo para la transmisión de paquetes

Cuando el módulo WEB envía un paquete al módulo coordinador, este lo recibe y le regresa un paquete de datos referentes al estado de los pines de los nodos remotos al módulo WEB, este los guarda en los registros especificados anteriormente.

Para enviar los paquetes entre módulos se creo un algoritmo de recepción de paquetes.

Cuando un módulo envía los bytes del paquete, el otro módulo los recibe, usando una función de interrupción y los guarda en un buffer.

El algoritmo busca el byte inicial para guardar al paquete y determina si es un paquete válido usando el último byte, si es valido se incrementa una

variable cuya función es indicar que un paquete valido ha llegado, si no es valido la variable permanece en cero, esto se describe con el diagrama de flujo de la figura 3.11, el programa se puede consultar en el apéndice c.

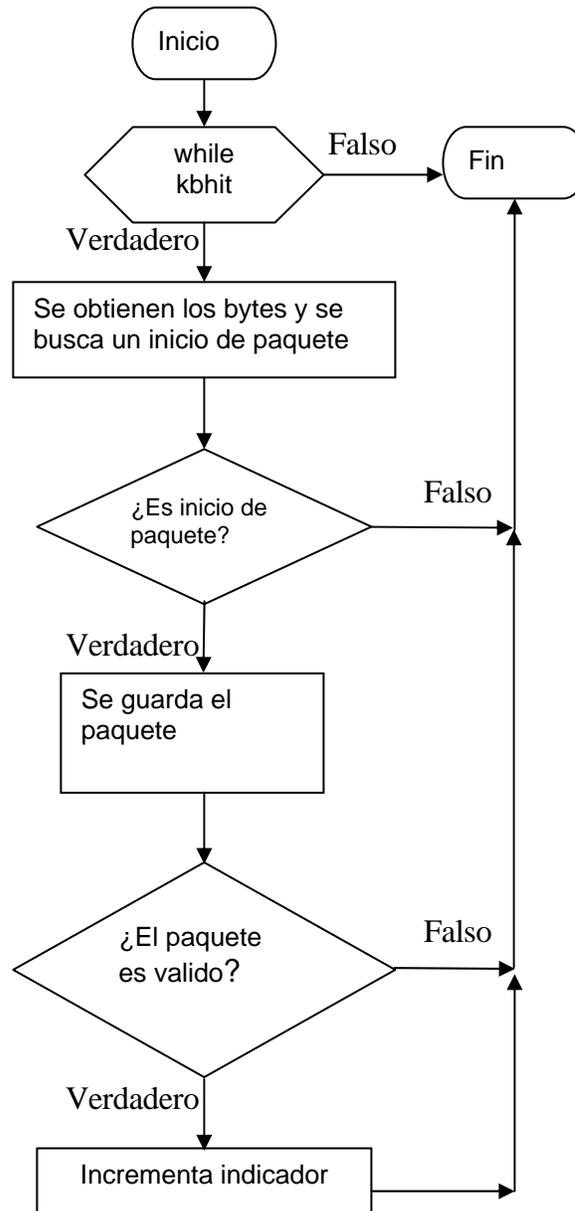


Figura 3.11 Diagrama del algoritmo de transmisión de paquetes entre módulos

3.8 Diseño del programa del Nodo coordinador

El módulo coordinador maneja el flujo de datos entre los nodos remotos y el módulo WEB.

Cuando se accede a una página WEB el módulo servidor envía un paquete de datos al nodo coordinador, este recibe el paquete y guarda los datos necesarios, los bytes del paquete tienen una función específica, el primero y el último define si es un paquete válido, el segundo byte indica el número de paquete, con el cual se indica que página se está accediendo y de acuerdo a esto se determina qué funciones de comunicación inalámbrica se deben de ejecutar. Además también se determina qué datos se deben de enviar al módulo WEB. El tercer byte define el número del nodo con el que establece la comunicación inalámbrica.

Los bytes restantes del paquete son datos útiles para las funciones inalámbricas. El programa se describe en el diagrama de flujo de la figura 3.12, y se puede consultar en el apéndice B.3.

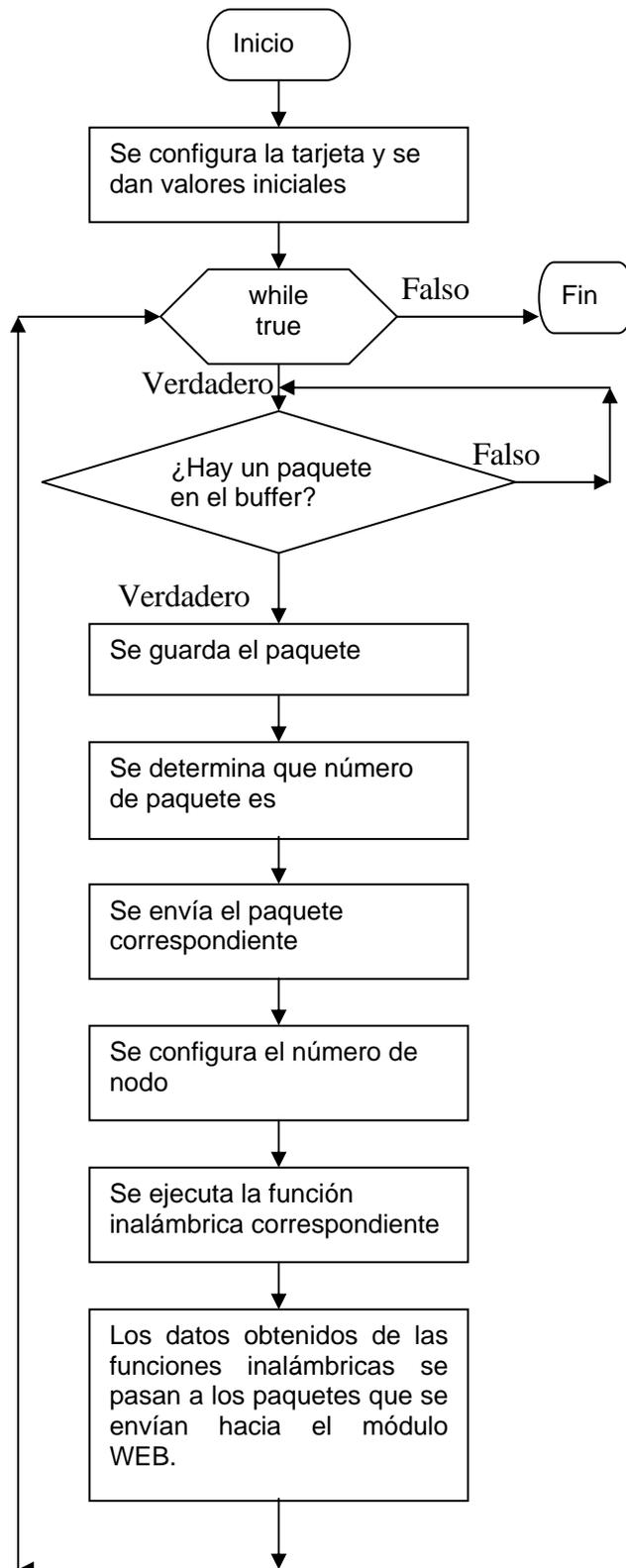


Figura 3.12 Diagrama de la función implementada en el nodo coordinador

3.9 Diseño de las páginas WEB

Para poder visualizar el estado de los puertos y poder configurarlos además de poder cambiar la configuración de red se definieron 7 páginas WEB para el sistema.

La primera es la página inicial, es cargada cuando se accede al servidor desde una PC conectada a la red ethernet, como se observa en la figura 3.13.

La página contiene el índice de enlaces a las páginas que muestran el valor de lo puertos, así como el enlace a las páginas de configuración.

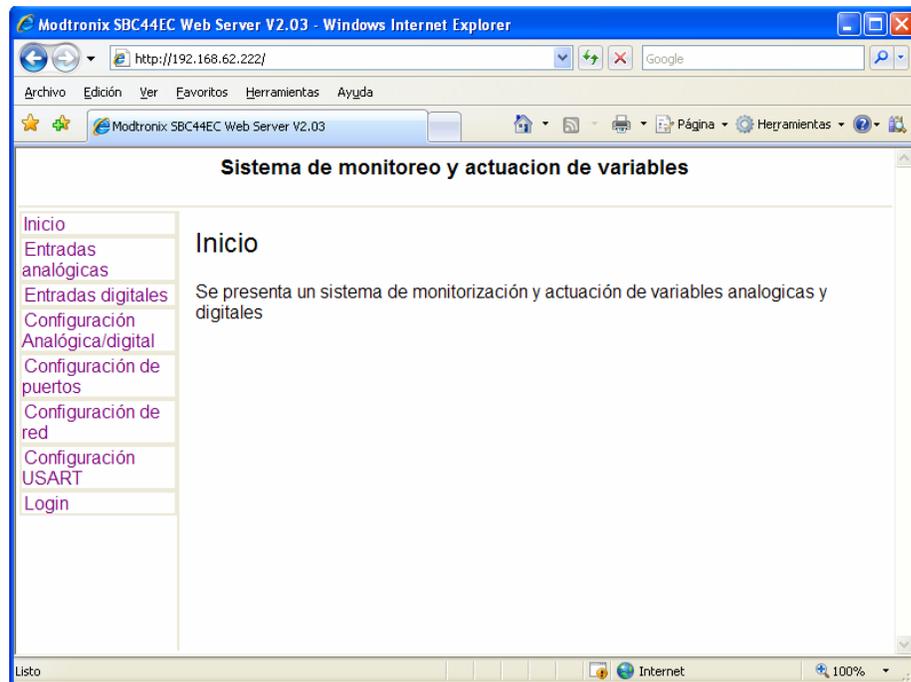


Figura 3.13 Página WEB de inicio, se carga cuando se accede al sitio WEB.

Los archivos de las páginas restantes se guardan con extensión CGI para poder cargar los datos en área que se encuentra al lado derecho del recuadro de índice de enlaces.

La segunda página muestra los valores analógicos en un rango de voltaje que de 0V a 5V con dos decimales de precisión, como se muestra en la figura 3.14.

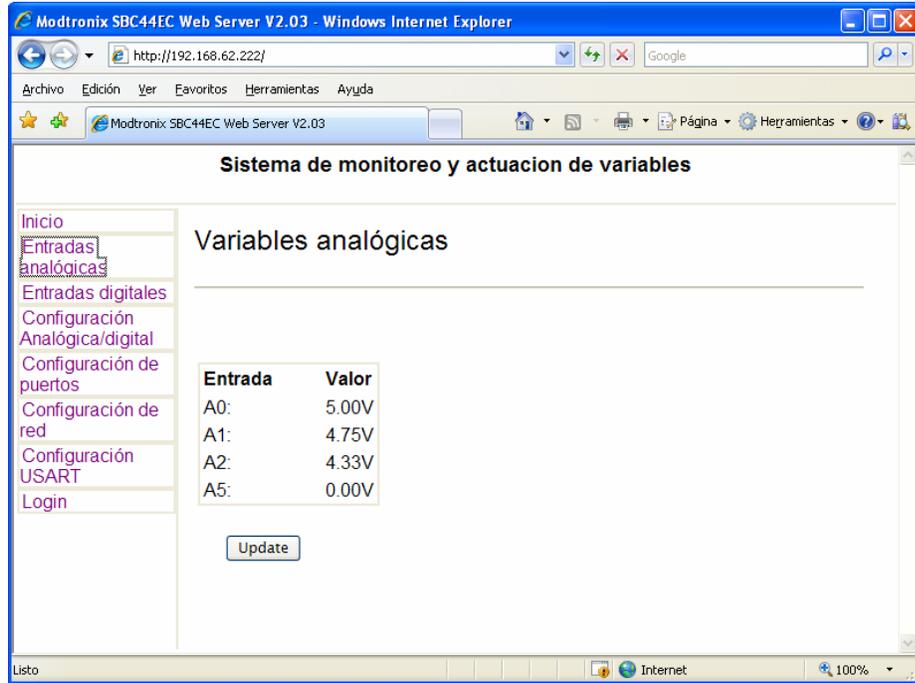


Figura 3.14 Página WEB que muestra los valores analógicos de un nodo remoto

La tercer página muestra el valor de los pines de los puertos A y B de los nodos remotos, además puede cambiarse el estado de los pines que están configurados como salidas, como se muestra en la figura 3.15.

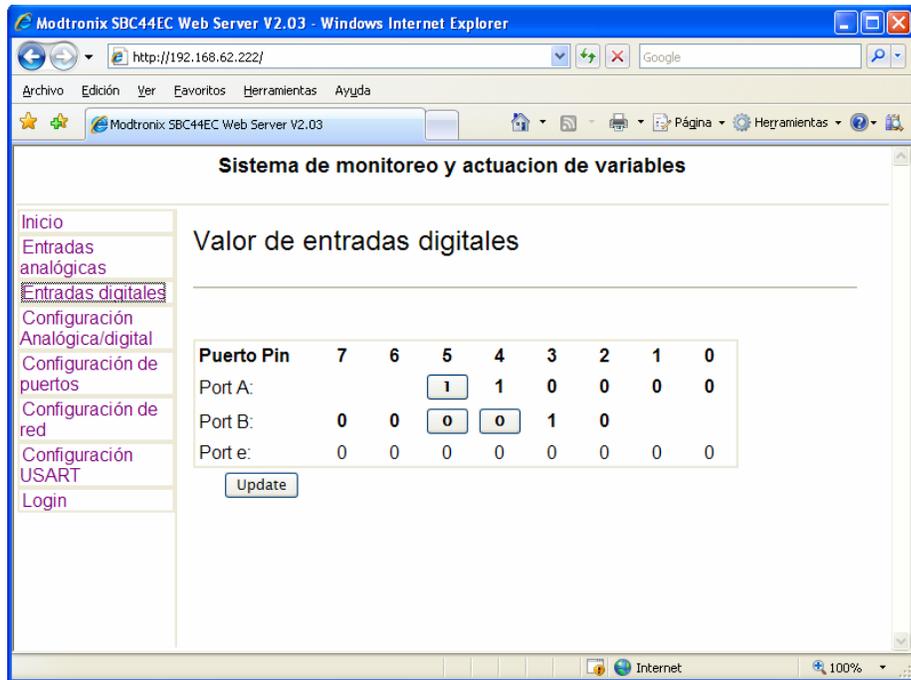


Figura 3.15 Página WEB que muestra los valores digitales de los pines de los nodos remotos

La cuarta página, se muestra en la figura 3.16. Fue diseñada para configurar a los pines del puerto A como digitales o analógicos. Además se puede seleccionar cual de los tres nodos se va a configurar, debido a que el pin A3 no está disponible para conectar solo se pueden hacer 4 modos de configuración, como se muestra en la tabla 3.3

Modo	Pines Analógicos	Pines digitales
1	A ₀	
2	A ₀ , A ₁	A ₂ , A ₅
3		A ₀ , A ₁ , A ₂ y A ₅
4	A ₀ , A ₁ , A ₂ y A ₅	

Tabla 3.3 Modos de configuración de los nodos remotos

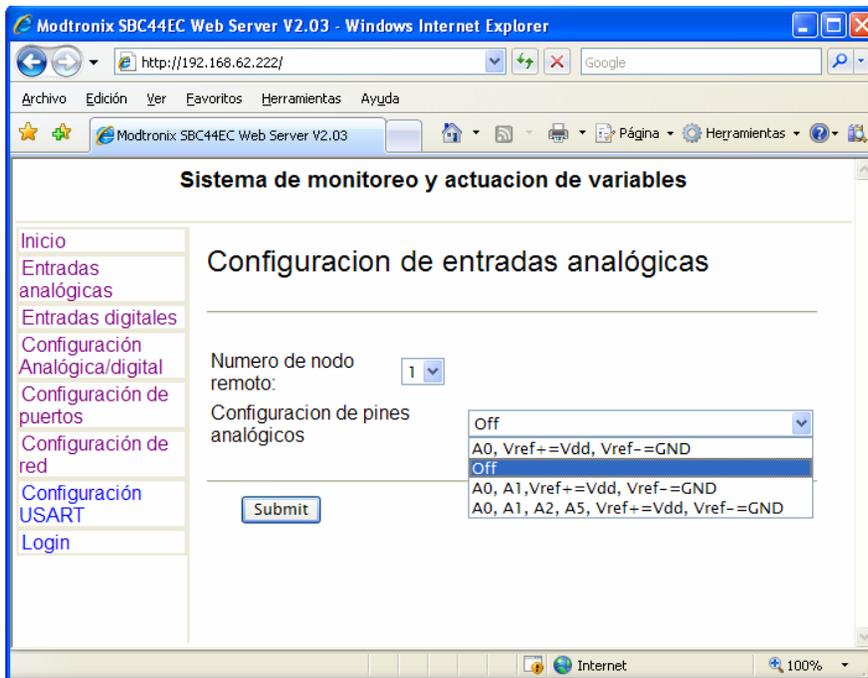


Figura 3.16 Página WEB donde se configura los pines del puerto A en modo digital o analógico

La quinta página se diseñó para poder configurar a los pines de los puertos A y B correspondientes a los nodos remotos como entradas o como salidas. También se puede seleccionar a que nodo se va a configurar, como se muestra en la figura 3.17.

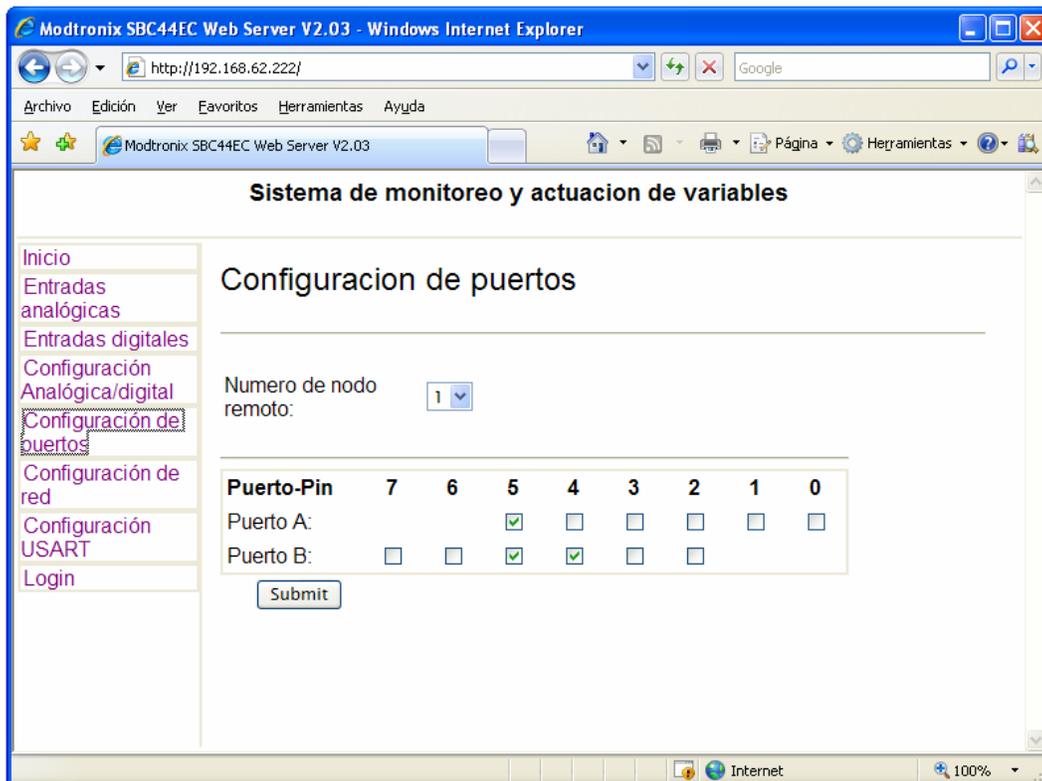


Figura 3.17 Página WEB donde se configuran a los pines de los puertos como entradas o salidas

3.10 Sumario

En este capítulo se presentó la forma en que se diseñó cada una de las partes que conforman el sistema. Primero se mostró un esquema general en donde se describen todas las partes de forma global, después se mostró como se fueron desarrollando las funciones y algoritmos para cada uno de los módulos, también se describe brevemente las características de las tarjetas que se usaron para implementar los diferentes módulos; además de las páginas que se usan para visualizar la información. En el capítulo siguiente se describen las pruebas que se hicieron, así como los resultados obtenidos.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En este capítulo se describen las pruebas que se realizaron a lo largo de la tesis y así como las conclusiones a las que se llegó después de hacerlas, estas pruebas sirvieron como base para desarrollar los diseños de las funciones que se presentaron en el capítulo 3, primero se presentan las pruebas que se refieren a la comunicación inalámbrica, comenzando con dos módulos y después aumentando a cuatro módulos, se sigue con la comunicación entre el módulo servidor y un módulo inalámbrico, al final se prueba el sistema completo.

4.1 Elementos usados y configuración de los módulos inalámbricos

La comunicación inalámbrica entre módulos ZMD, se realizó utilizando las bibliotecas, cuyas interfaces de funciones están en el archivo, zmd44102.h, la cual contiene funciones específicas para la transmisión y recepción de paquetes de datos.

La programación del PIC16F452 se hizo usando el compilador PIC-C [24], el cual se hace a través de lenguaje C.

Para facilitar el manejo de algunos elementos de la tarjeta prototipo, se creó el archivo ZMD.H, donde se guardaron funciones que se crearon para manejar al sistema inalámbrico, ver anexo 3, en donde se definen funciones y variables que controlaran a los leds y a los dígitos de 7 segmentos, así como la comunicación serial RS232.

Todas las pruebas se hicieron colocando las tarjetas con un metro de distancia entre ellas.

Para configurar las tarjetas inalámbricas, primero se define el identificador PAN del coordinador, la dirección de la tarjeta origen de donde se envían los datos y la dirección de las tarjetas destino a las que se les enviarán los datos. Los datos y el número de canal a utilizar, también se indican en el archivo cabecera “zmd.H” como se muestra en seguida:

```
#define MyPanId          1
#define MyShortAddress  1
#define DestShortAddress_1 2
#define DestShortAddress_2 2
#define MyRFChannel     6

#include "zmd.h"
```

Se declara la estructura de tipo “ZMAC_HEADER” en la cual, se guardará la trama de datos recibidos provenientes de alguna otra tarjeta, también se

declara una estructura tipo ZMAC_NODE_INFO la cual tendrá los datos del nodo al cual se le transmita la información

```
ZMAC_HEADER header;  
ZMAC_NODE_INFO txNode;
```

Se activa la tarjeta así como la MAC y la API, se establece el identificador PAN y se establece que las direcciones serán de tipo corto de 16 bits

```
ZMac_Init();  
ZMac_SetMyPanId(MyPanId);  
ZMac_UseMyShortAddress();  
ZMac_SetMyShortAddress(MyShortAddress)  
:
```

Se establecen los datos del nodo al que se le enviarán los datos

```
txNode.panId=MyPanId;  
txNode.addressMode=ZMAC_ADDRESS_MODE_SHORT;  
txNode.shortAddress=DestShortAddress;
```

Se establece el canal a usar

```
ZPhy_SetChannel (MyRFChannel );
```

4.2 Comunicación módulo a módulo

Se hicieron tres pruebas de comunicación entre dos tarjetas, enviando datos de tipo analógico y digital, variando las condiciones para poder mostrar el funcionamiento correcto de las tarjetas.

4.2.1 Pruebas de envío y recepción de datos usando diferentes ejemplos

La primera que se realizó fue la de enviar datos desde un módulo y recibirlos en otro. Para este propósito se definió a una tarjeta como transmisor y a otra tarjeta como receptor.

La función del transmisor consistió en enviar el valor de un contador hacia el receptor con retraso de un segundo, la función del receptor fue la de recibir el valor enviado por el transmisor y mostrar el valor en sus respectivos dígitos de 7 segmentos.

Se usó la función “ZMac_PutPacket [20]” Para enviar el mensaje en el módulo transmisor y la función “ZMac_GetPacket [20]” Para recibir el mensaje en el módulo receptor, además la función regresa un código de error o un código confirmación de que el paquete se recibió.

.También se mostró en el nodo transmisor la relación de datos enviados y datos recibidos en forma de porcentaje, de tal forma que si todos los datos se envían correctamente, se tendrá un razón de 100 y si algún dato no llega correctamente la razón será menor de 100.

Los programas se describen mediante los diagramas de flujo de la figura 4.1

La segunda prueba consistió en transmitir datos de PC a PC de forma inalámbrica, para esto cada módulo se conectó al puerto serial de una PC, usando el programa de “hyperterminal”, como se ilustra en la figura 4.2.

Esta prueba consistió en escribir letras con el teclado en una de las PC y visualizar los caracteres en la otra PC. Usando las mismas funciones para enviar y recibir paquetes.

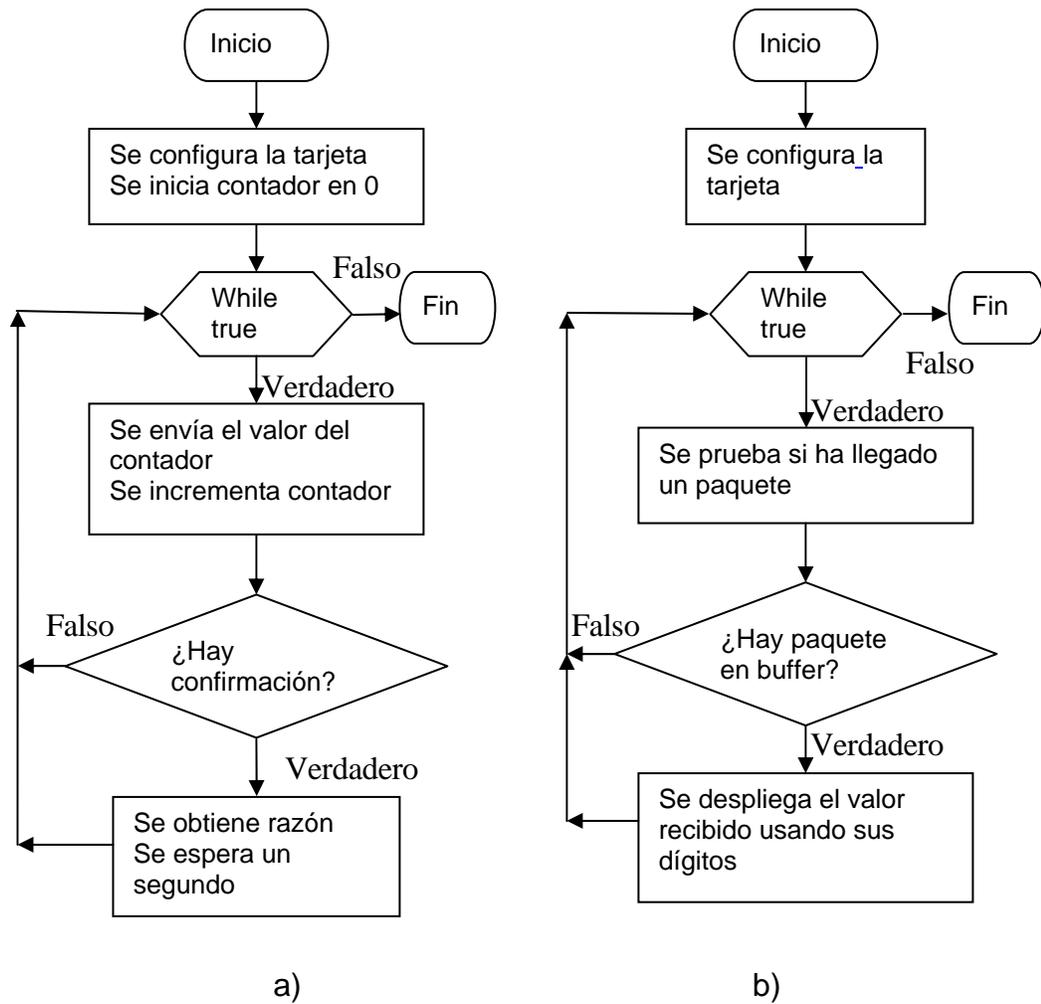


Figura 4.1 Descripción de los programas usados para la primer prueba, a) módulo trasmisor
b) módulo receptor

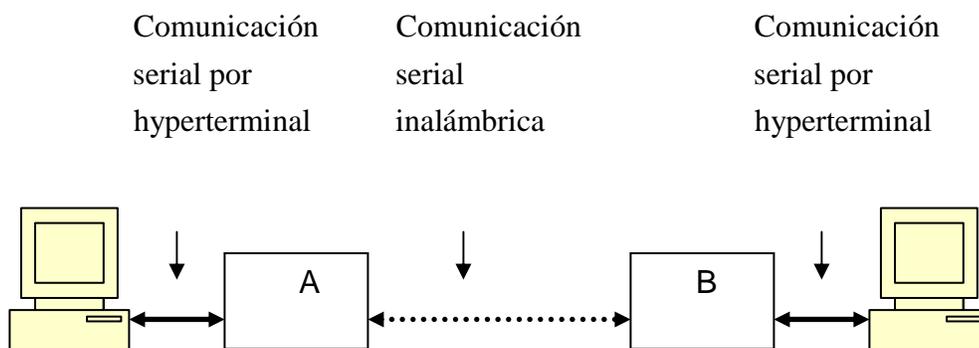


Figura 4.2 Representación esquemática de la transmisión serial inalámbrica

La hyperterminal de cada PC se configuró de la siguiente forma:

Bits por segundo	9600
Bits de datos	8
Paridad	Ninguno
Bits de parada	1
Control de flujo	Ninguno

Para este propósito se usó un solo programa para las dos tarjetas, en el programa se prueba si ha sido presionado una tecla, si se ha presionado se envía de forma inalámbrica, también se prueba si ha llegado un paquete de forma inalámbrica si esto ha ocurrido se envía por el puerto serial hacia la PC.

La tercer prueba consistió en leer el valor analógico del pin A_0 conectado al potenciómetro de la tarjeta y obtener el estado del pin A_4 conectado a un botón momentáneo, y enviar los valores obtenidos al otro módulo, de tal forma que los valores leídos en una tarjeta se mostrarán en la otra tarjeta a través de sus dígitos de 7 segmentos para el caso del valor analógico, y a través de un led para el caso del pin conectado al botón. El diagrama de flujo del programa usado para esta prueba se muestra mediante la figura 4.3.

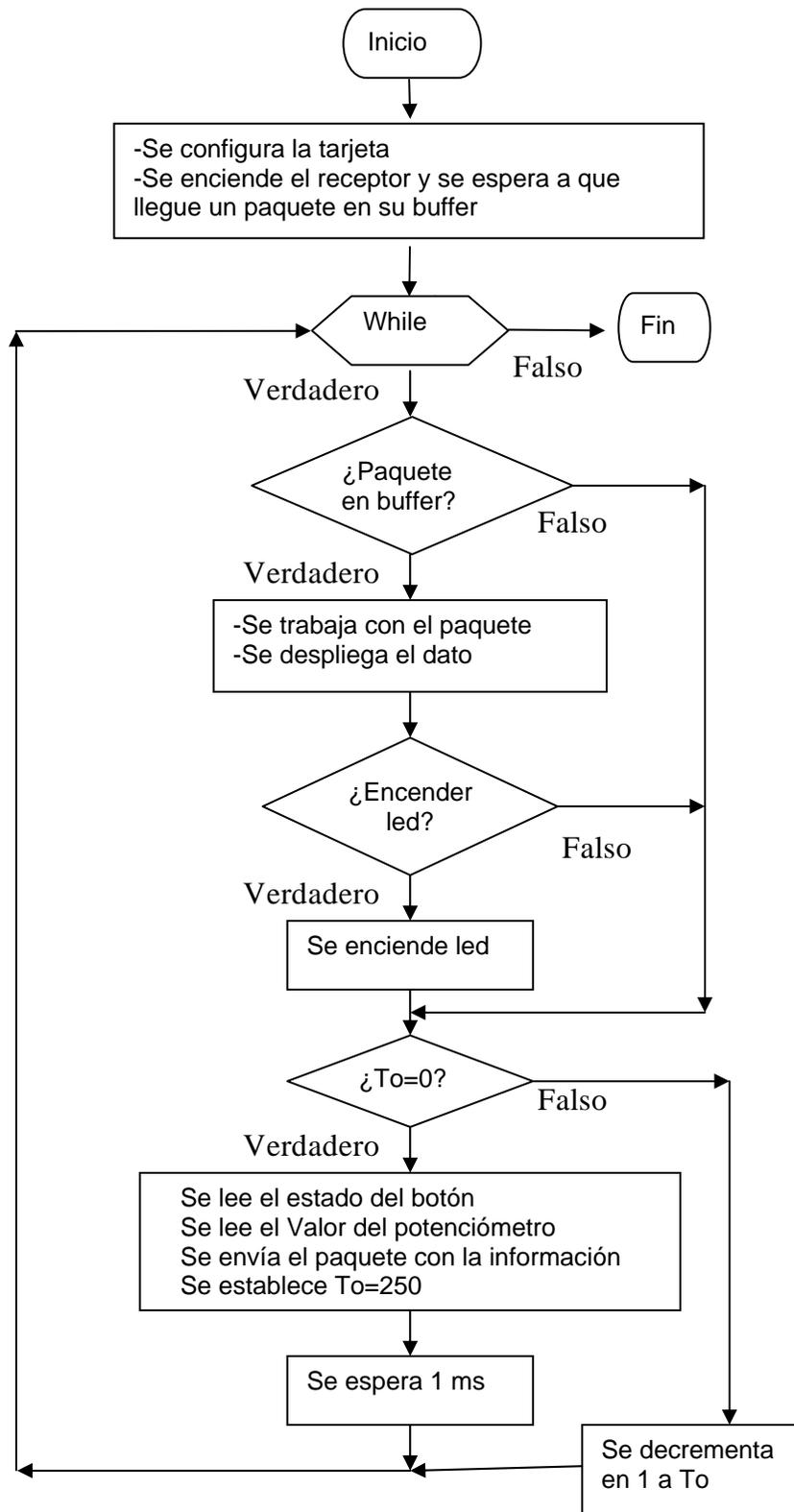


Figura 4.3 Diagrama de flujo del programa

4.2.2 Resultados de las pruebas módulo a módulo

Para la primera prueba se pudo observar que el valor del contador ubicado en la tarjeta transmisora se mostraba en la tarjeta receptora de forma correcta, En el transmisor se pudo observar a través de sus respectivos dígitos de 7 segmentos, la relación de datos enviados y datos recibidos. Como se muestra en la figura 4.4.

Se observó que mientras no se activara la tarjeta receptora la razón de envió de datos enviados y datos recibidos era de cero, pero cuando se conectaba, la razón se mostraba disminuida a causa de que no todos los datos fueron recibidos (figura 4.4 b).

Sin embargo si ambos se activaban al mismo tiempo la razón siempre era de 100 (figura 4.4 a).

En la prueba de comunicación de PC a PC se pudo observar que al presionar una tecla en el teclado de una PC la letra correspondiente se visualizaba en la pantalla de la otra PC de forma correcta.

Para la tercera prueba se varió la posición del potenciómetro para cambiar el valor analógico en un módulo y se observó que el valor se mostraba en el otro módulo (figura 4.5). También se observó que al presionar el botón en un módulo el led correspondiente en el otro módulo se encendía



a)



b)

Figura 4.4 Razón de paquetes enviados y paquetes recibidos

Módulo 1

Módulo 2



Figura 4.5 Prueba de envío de datos analógicos

4.3 Comunicación de una red de 4 módulos

Se hicieron dos pruebas para comprobar el funcionamiento correcto de los módulos al comunicarse con más de un módulo, se probó el uso de direcciones y el uso de un nodo que coordinara el flujo de datos entre los otros módulos.

4.3.1 Prueba de direccionamiento de los módulos

Primero se probó el direccionamiento de los módulos para construir una red, para este propósito se hizo la comunicación inalámbrica de 4 tarjetas ZMD, a cada tarjeta se le asignó una dirección física diferente. A las cuatro tarjetas se les asignaron las direcciones 1, 2, 3 y 4 respectivamente, como se muestra en la figura 4.6, en esta prueba se hizo la comunicación usando un solo canal.

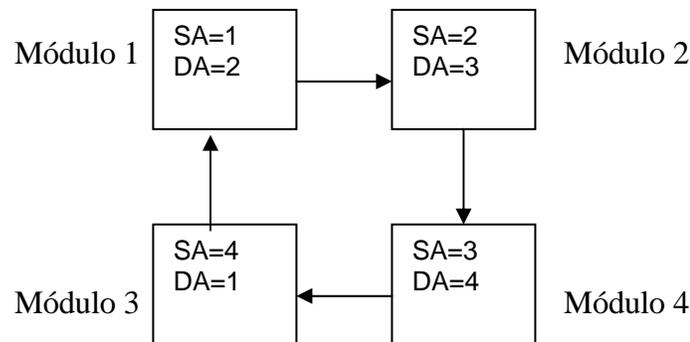


Figura 4.6 Red en topología de anillo

Donde SA es la dirección fuente (Source Address) y DA es la dirección destino (Destination Address).

En esta configuración se lee el voltaje que hay en el pin A_0 , conectado al potenciómetro y el estado del botón del nodo 1, los valores se envían al nodo 2 y ahí se visualiza el valor de voltaje en los dígitos de 7 segmentos y si el botón del nodo 1, se encuentra presionado se enciende un led del nodo 2, siguiendo la misma configuración, los valores que se leen en el nodo 2 se visualizan en el nodo 3, los valores que se leen en el nodo 3 se visualizan en el nodo 4 y los valores que se leen en el nodo 4 se visualizan en el nodo 1, formando una red de anillo.

El programa que se usó fue el mismo que el usado para la prueba de envío de valores analógicos descrito por el diagrama de flujo de la figura 4.3.

4.3.2 Comunicación inalámbrica entre un nodo coordinador y tres nodos remotos

En esta prueba se implementó una red en estrella usando tres nodos remotos y un nodo coordinador conectado a su vez a una PC de forma serial usando sus respectivos puertos seriales. La prueba consistió en hacer que el nodo1 definido como coordinador, reciba datos enviados por otros nodos y realice con ellos un análisis sencillo; además de coordinar el flujo de datos entre nodos de la red.

Los nodos 2 y 3 leen el valor analógico que hay en su respectivo pin A0 conectado a un potenciómetro y envían la información hacia el nodo 1.

El nodo coordinador recibe los valores, los compara y envía un par de valores al nodo 4 de acuerdo al resultado de la comparación.

El nodo coordinador compara los niveles de voltaje enviado por los nodos 2 y 3, con valores determinados. A continuación se muestra la comparación en la tabla 4.1:

Casos		Valor 1	Valor 2
$N_2 > 78$	$N_3 < 78$	1	1
$N_2 < 78$	$N_3 > 78$	1	0
$N_2 > 78$	$N_3 > 78$	0	1
Cualquier otro caso		0	0

Tabla 4.1 Comparaciones que realiza el programa cargado en la tarjeta coordinador. Otra función del nodo coordinador es recibir un valor enviado por la PC a través del puerto serial usando el programa de hyperterminal y enviar el valor al nodo 2, este recibe el dato y lo muestra mediante sus dígitos de 7 segmentos, el arreglo de los módulos se ilustra en la figura 4.7

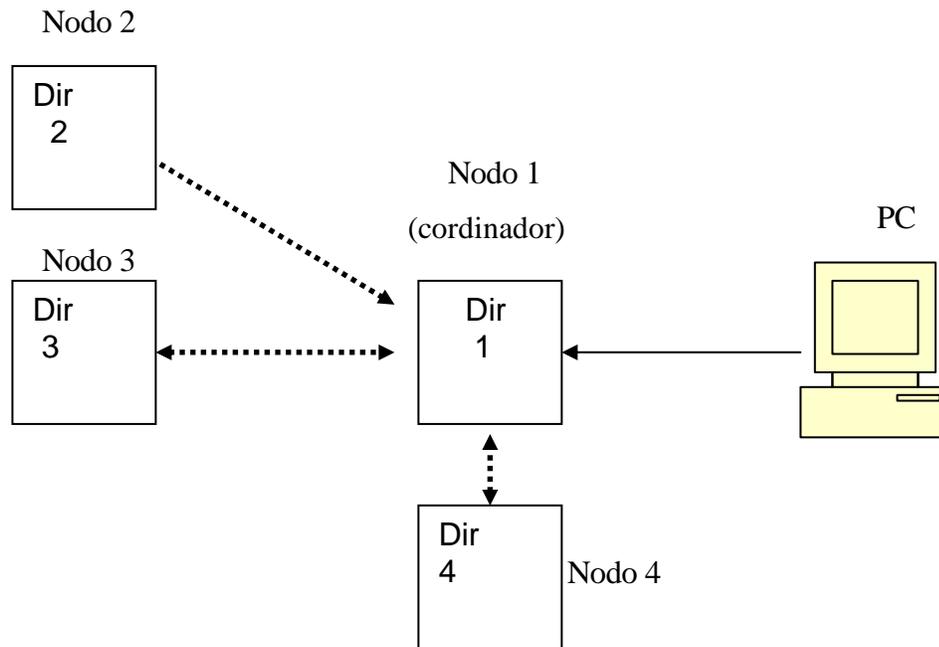


Figura 4.7 Representación del arreglo en estrella que se hizo con los 4 módulos inalámbricos

Para cada uno de los nodos remotos se implementó un programa diferente, basado en el programa explicado mediante el diagrama de flujo de la figura 4.3. Para el nodo coordinador se implementó un programa capaz de comunicarse con los tres nodos y con la PC, este programa se describe en el diagrama de la figura 4.8

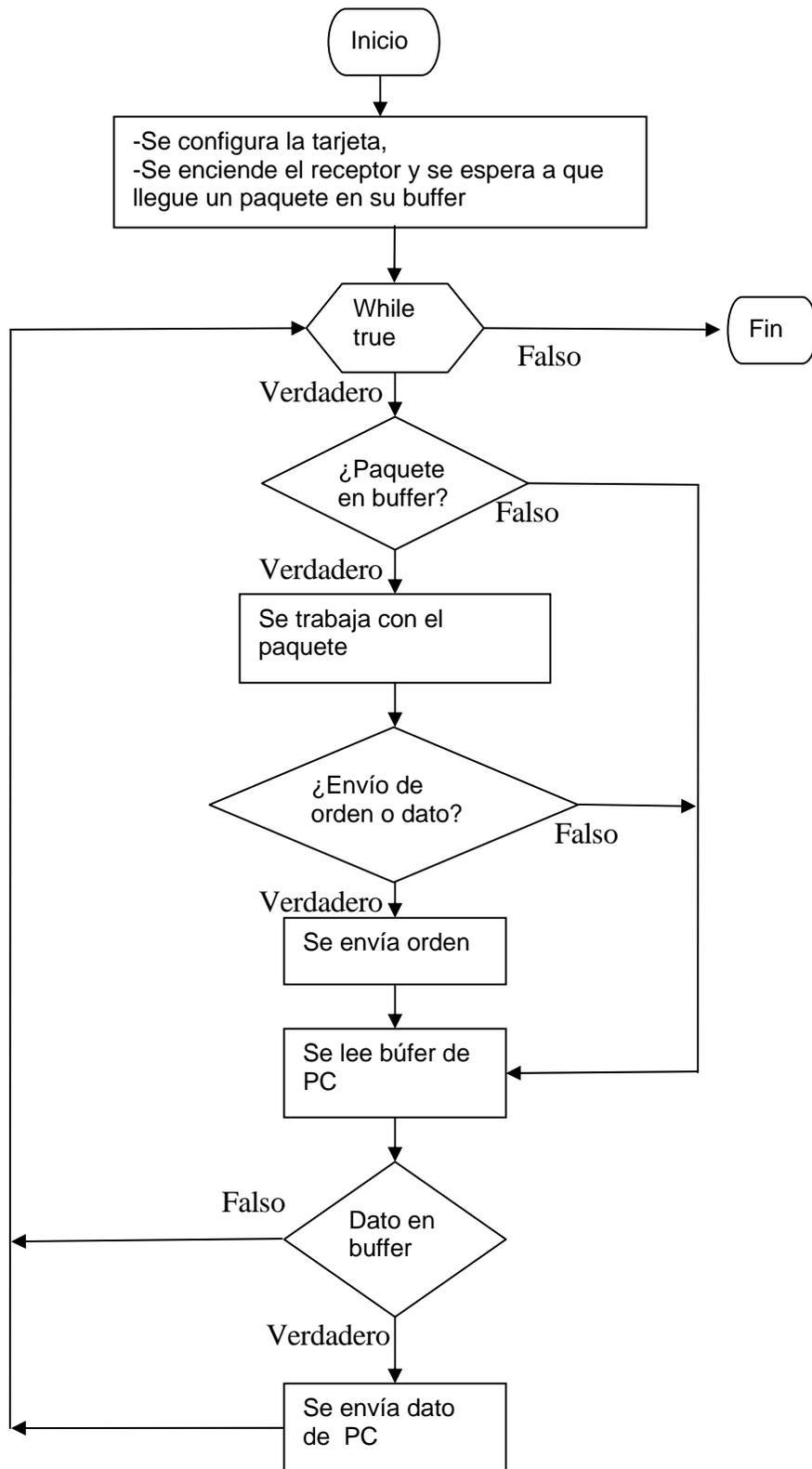


Figura 4.8 Diagrama de flujo del programa que se implementó para la figura 4.5

4.3.3 Resultados de las pruebas de redes de 4 módulos

Para la primera prueba de red en anillo, se hizo girar la perilla de los potenciómetros a diferente posición para variar el valor de voltaje, en cada uno de los módulos, se observó que los valores desplegados variaban de acuerdo a la posición de los potenciómetros. También se comprobó que al presionar el botón de un módulo el led del módulo que le correspondía de acuerdo a la dirección asignada, se encendía.

Para el caso de la red en estrella con un módulo coordinador se observó que los datos se enviaban y se recibían de forma correcta.

Al hacer rotar los potenciómetros hasta determinado nivel de los nodos 2 y 3, de acuerdo a la tabla de comparación, se modificaba el estado del led del nodo 4 y se cambiaba el valor mostrado por los dígitos, también al presionar una letra en el teclado de la PC, el código correspondiente a la letra se mostraba en el nodo 2.

Con estas dos pruebas se pudo mostrar que un nodo podía manejar el flujo de datos entre los otros nodos, además de que el nodo coordinador también puede procesar los datos.

4.4 Prueba de nodos inalámbricos diseñados para el sistema

Una vez que se hicieron las pruebas iniciales se pudo diseñar las funciones específicas para el manejo de los nodos remotos por medio del nodo coordinador de forma inalámbrica.

Para probar a los nodos remotos, se conectaron interruptores en los pines definidos como digitales y potenciómetros en los pines definidos como analógicos, de acuerdo al diagrama de la figura 4.9.

Para probar el sistema inalámbrico y visualizar la correcta comunicación entre los diferentes nodos se implementó una red inalámbrica tipo estrella

con un nodo coordinador y tres nodos remotos. Para poder visualizar la información se conecto al nodo coordinador a una PC de forma serial RS232. El esquema se muestra en la figura 4.10.

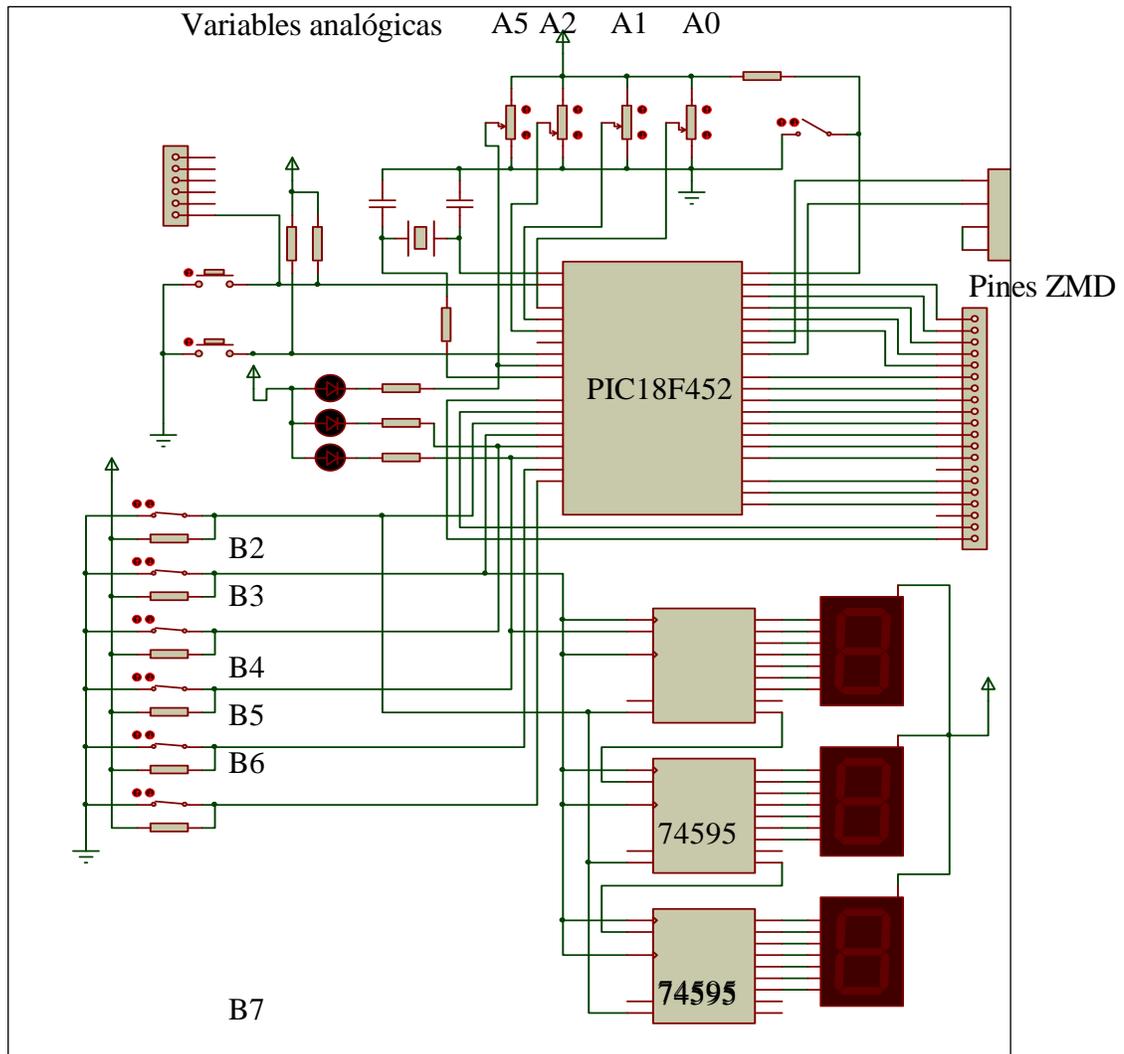


Figura 4.9 Conexión del nodo remoto para probar el funcionamiento de los pines analógicos y digitales

Donde la comunicación de los nodos remotos con el coordinador es bidireccional.

La prueba consistió en leer el valor de los pines de los nodos remotos y visualizar los valores en la PC, para esta prueba solo se conectaron tres

potenciómetros al nodo remoto número 1, para los otros nodos remotos se hizo la prueba con el potenciómetro, el botón y los leds incluidos en la tarjeta prototipo.

Al nodo coordinador se le asignó la dirección 1, y al nodos remotos 1, 2, 3 se les asignó las direcciones 2, 3 y 4 respectivamente.

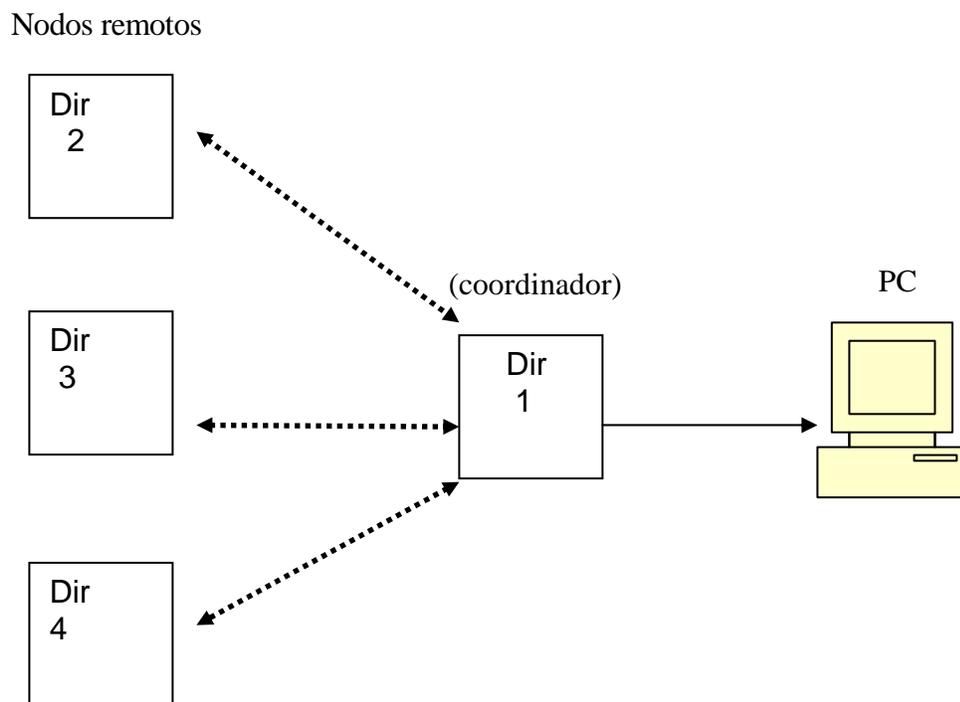


Figura 4.10 Esquema de comunicación inalámbrica entre s inalámbricos

A cada uno de los nodos remotos se les cargó las funciones desarrolladas en el apartado 3.5 del capítulo 3.

Para el nodo coordinador se implementó un programa de prueba, este se puede ver en el apéndice B.2.

En el programa de prueba se hace una serie de llamadas a las funciones diseñadas para la comunicación con el nodo remoto, estas se pueden consultar en el anexo 2, la información que se obtiene se despliega en la pantalla de la PC usando el programa de hyperterminal.

Primero se envía la configuración de los puertos; después, se envía el valor que deben de tener los pines definidos como salidas y después se pide una tabla de datos.

La configuración de los puertos de los nodos remotos mediante el nodo coordinador se explica a continuación: Para el nodo remoto 1, los pines del puerto A se configuraron como entradas analógicas, los pines del puerto B: B1, B3 y B4 se configuraron como salidas y los demás como entradas.

Para los nodos remotos 2 y 3, el pin A1 se configuró como entrada analógica y A4 se configuró como entrada digital, los demás pines del puerto A se configuraron como salidas, mientras que para el puerto B solo el pin B0 se configuró como entrada y los demás pines del puerto se configuraron como salidas.

Durante las pruebas se modificó el estado de los potenciómetros así como el de los botones colocados en los pines de los nodos remotos.

Se observó que los datos se desplegaron de forma correcta en la pantalla de la hyperterminal En la siguiente tabla se muestran los valores obtenidos en la pantalla. En la tabla 4.2 se muestra un ejemplo de los datos obtenidos.

De la tabla se puede observar que para el caso del nodo remoto 1 se obtuvieron los valores analógicos en los pines del puerto A.

Los valores de los pines definidos como entradas de los puertos A de los nodos remotos se observan en el registro PORTA, en la tabla se observa que para el nodo remoto 1 encuentra un valor de 16, en el nodo remoto se observa un valor de 0 y para el tercer nodo un valor de 16, esto se explica

debido a que cuando no se presiona el botón de la tarjeta el pin A4 tiene un valor de uno, pero cuando se presiona el botón el pin tiene un valor de cero. Los valores de salida se encuentran en los registros LATA y LATB.

Pin o puerto correspondiente	Nodo remoto 1	Nodo remoto 2	Nodo remoto 3
A ₀	120	92	10
A ₁	30	0	0
A ₂	230	0	0
A ₃	0	0	0
A ₅	255	0	0
PORTA	16	0	16
PORTB	136	0	0
LATA	0	0	32
LATB	48	0	32

Tabla 4.2 Valores leídos de la pantalla de la hyperterminal

De la tabla de resultados se pudo observar que los datos se desplegaban correctamente.

Con estos resultados se concluyó que la recepción de datos funcionó correctamente.

4.5 Prueba de comunicación entre el módulo WEB y el módulo coordinador

En esta parte se probó la comunicación entre el WEB y el coordinador, además de la página WEB diseñada para la interacción con el usuario.

La primer parte consistió en probar la carga de la página WEB, desde una PC, usando el programa “Internet Explorer 6”.

Para realizar las pruebas se usó un cable cruzado ethernet conocido generalmente como “crossover”, el cual se conecta directamente del puerto ethernet [26] de la computadora al puerto ethernet de la tarjeta WEB.

Para poder comunicar al WEB con la PC se configuraron los puertos para operar correctamente en Internet de las dos partes.

Las propiedades de protocolo se encuentran dentro de las propiedades de la conexión de área local, la modificación quedó de la siguiente forma:

- Dirección IP: 192.168.92.221
- Puerta de enlace: 192.168.92.222
- Máscara de subred 255. 255. 255.0
- DHCP Deshabilitada

La modificación de las propiedades del protocolo TCP/IP correspondientes a la tarjeta WEB se hizo de forma serial [27], quedando de la siguiente forma:

- Dirección IP: 192.168.92.222
- Puerta de enlace: 192.168.92.221
- Máscara de subred: 255. 255. 255.0
- DHCP Deshabilitada

Las páginas diseñadas se guardaron en la memoria externa vía tcp/ip [22, 27]

Para acceder a la página se escribió en la barra de direcciones de Internet Explorer la dirección 192.168.62.222 y se accedió a las páginas diseñadas.

Con esto se comprobó que la página se acceda de forma correcta.

La segunda parte de la prueba consistió en probar la comunicación del WEB con el coordinador inalámbrico, para esto se conectaron los módulos de acuerdo a como se muestra en la figura 4.11.

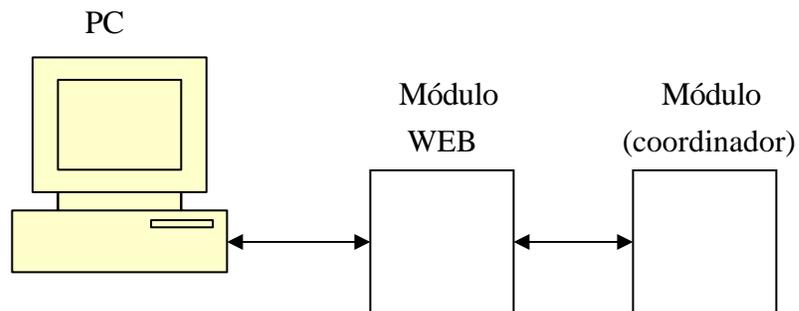


Figura 4.11 Esquema de conexión para la prueba de comunicación entre el módulo coordinador el módulo WEB y una PC

Esta prueba consistió en enviar valores constantes del coordinador al servidor y visualizar los valores en la página WEB, además de modificar variables definidas en el servidor y enviar los valores al nodo coordinador para ser mostrados en sus dígitos de 7 segmentos.

Para este propósito se creó un programa de prueba para cada módulo donde se usa principalmente el algoritmo presentado en el capítulo 3 sección 3.6.3

El programa para el módulo WEB se escribió dentro de la función llamada como While(1) que se encuentra dentro de la función principal del archivo mxwebsrvr [22].

Este programa de prueba se describe mediante el diagrama de flujo de la figura 4.12.

El programa del servidor, en lenguaje c, se compiló usando MPLAB.

Para realizar la prueba se accedió a la página de la figura 4.13 para verificar que los datos enviados desde el nodo coordinador llegaban correctamente.

Después se acceso a la página WEB de la figura 4.14 para verificar que al modificar el estado de los botones virtuales de la página, se modificaba el registro correspondiente y esta modificación se visualizaba en los dígitos de 7 4segmentos del módulo coordinador.

Con esta prueba se pudo concluir que el flujo de datos entre los dos módulos era correcto.

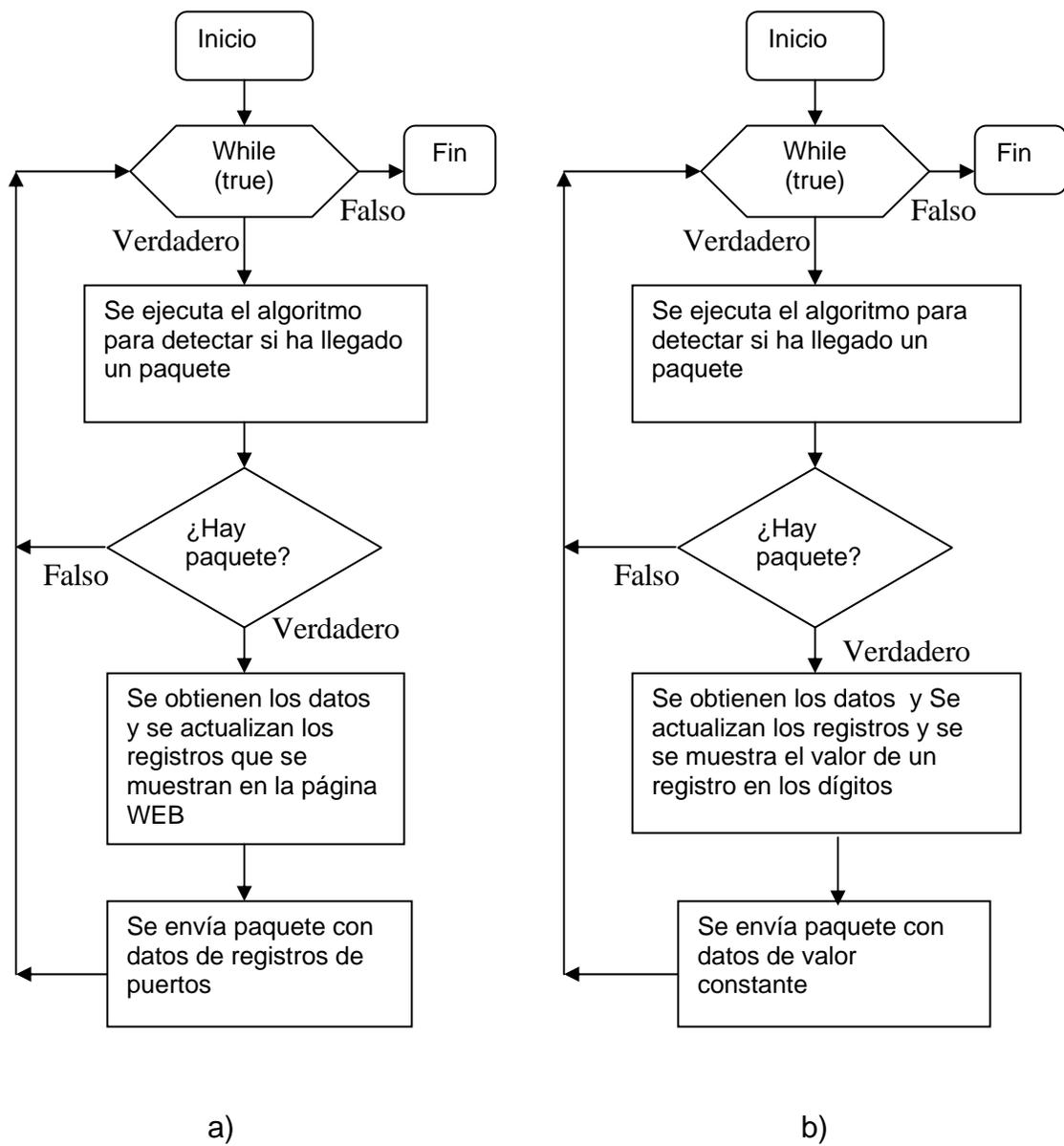


Figura 4.12 Diagrama de flujo de las funciones de prueba para recibir datos y mostrar la información a) módulo WEB b) módulo coordinador



Figura 4.13 Página WEB donde se muestran los valores analógicos leídos de los nodos remotos

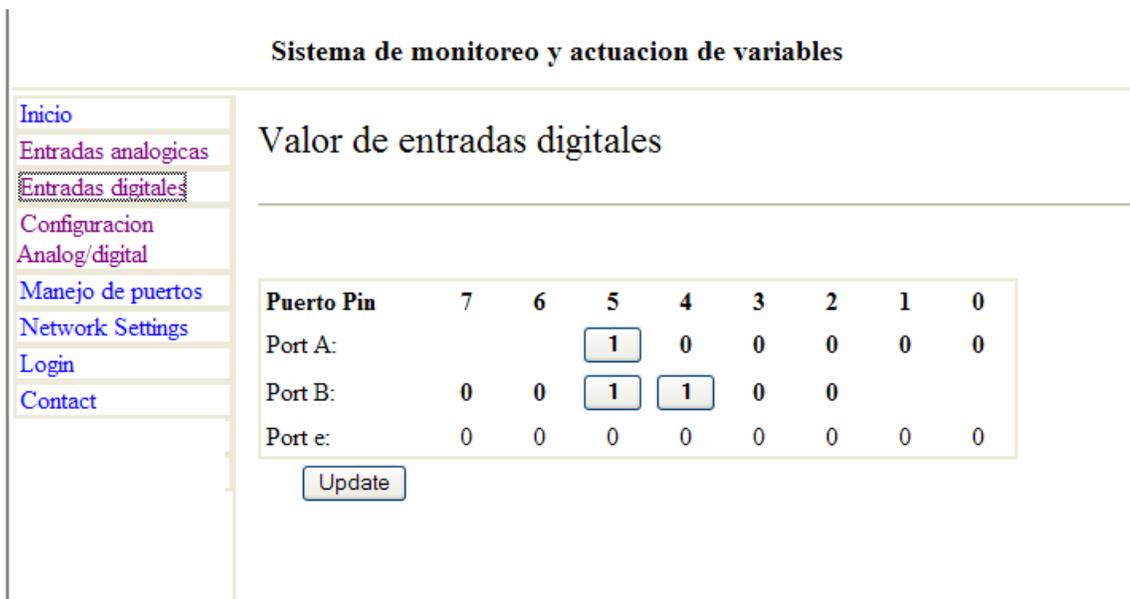


Figura 4.14 Página WEB donde se muestran los valores digitales leídos de los nodos remotos

4.6 Prueba del sistema completo

Una vez que se hicieron funcionar el sistema inalámbrico y el sistema WEB por separado, se hizo la prueba de todo el sistema, como se muestra en la figura 4.15

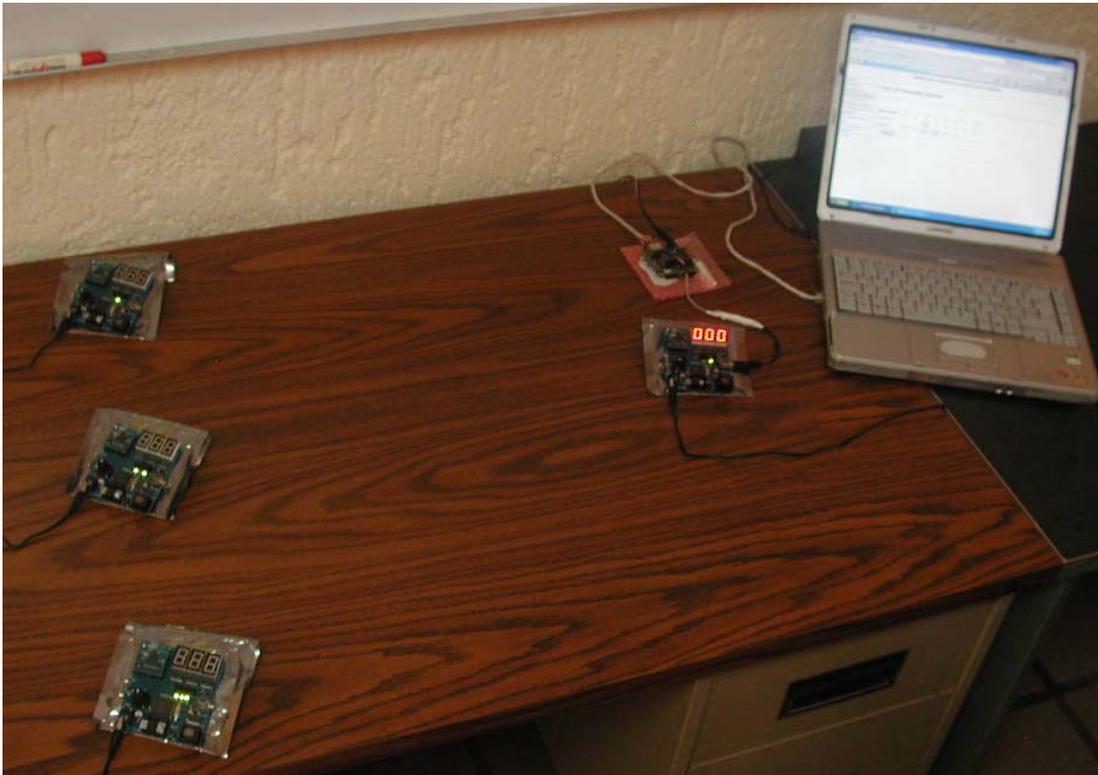


Figura 4.15 Esquema de conexión del sistema completo

Se probó la configuración de los puertos, la modificación del estado de los pines configurados como salidas además de leer el estado de los pines configurados como entradas.

Como se hizo anteriormente al nodo coordinador se le asignó la dirección 1, y al nodos remotos 1, 2, 3 se les asigno las direcciones 2, 3 y 4 respectivamente.

A cada nodo remoto se le cargaron las funciones descritas en el apartado 3.5 del capítulo 3.

Al nodo coordinador se le cargó el programa descrito en el apartado 3.6 y 3.8, del capítulo 3

Al nodo Web se le cargó con las funciones descritas en el apartado 3.7 del capítulo 3 y se grabó en la eeprom del nodo WEB las páginas descritas en el apartado 3.9 del capítulo 3

Para el nodo remoto 1 se hizo la siguiente conexión de sus pines

El pin A0 se encuentra conectado a un potenciómetro

El pin A1 se conecto a Vcc (5 V) de forma directa

El pin A2 se conectó a un interruptor para poder cambiar su conexión de Vcc a Vdd (0 V)

El pin A4 Se encuentra conectado un botón

El pin A5 se encuentra conectado a un led

Los pines B6 y B7 y B3 se conectaron a Vcc de forma directa

Los pines B5 y B4 se encuentran conectados a leds

El pin B2 se conecto a un interruptor para poder cambiar su conexión de Vcc a Vdd

Se configuró a los pines A0 y A2 como entradas analógicas, a los pines A5 B5 y B3 se configuraron como salidas y todos los demás pines como entradas.

La configuración analógica se muestra en la figura 4.16, mientras que la configuración de los pines como entradas o salidas se muestra en la figura 4.17.

Configuración de entradas analógicas

Numero de nodo remoto:

Configuración de pines analógicos:

Figura 4.16 Configuración de pines digitales, imagen extraída de la página WEB de configuración analógica

Puerto-Pin	7	6	5	4	3	2	1	0
Puerto A:			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Puerto B:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Figura 4.17 Tabla de configuración de pines, imagen extraída de la página WEB de configuración de puertos

Una vez hecha la configuración se pasó a leer el estado de los pines en la página de valores digitales.

Los resultados se muestran en la figura 4.18

Puerto-Pin	7	6	5	4	3	2	1	0
Puerto A:			<input type="text" value="0"/>	1	0	0	0	0
Puerto B:	1	1	<input type="text" value="0"/>	<input type="text" value="0"/>	1	1		

a)

Puerto-Pin	7	6	5	4	3	2	1	0
Puerto A:			<input type="text" value="1"/>	0	0	0	0	0
Puerto B:	1	1	<input type="text" value="1"/>	<input type="text" value="0"/>	1	1		

b)

Puerto-Pin	7	6	5	4	3	2	1	0
Puerto A:			<input type="text" value="1"/>	1	0	1	0	0
Puerto B:	1	1	<input type="text" value="0"/>	<input type="text" value="1"/>	1	0		

c)

Figura 4.18 Valores leídos en la página WEB

De las figuras se puede ver que los pines conectados directamente a Vcc (B7, B6, B3) se mantuvieron con un valor de 1 los pines conectados a un interruptor (A4, A2, B2) cambiaron de valor conforme se cambiaba el estado del interruptores, mientras que los pines definidos como salida (A5, B5 y B4) se pudo cambiar su estado para encender o apagar los leds de los nodos remotos.

Los pines A0 y A1 al estar configurados como analógicos en los casos mostrados permanecieron con un valor de cero.

Por último se observó la página donde se pueden observar los valores de los pines analógicos, en la figura 4.19 se muestran los valores presentados. De la figura se puede observar que A1 presenta un valor de 5 V debido a que se conectó a Vcc, mientras que el valor del pin A0 varió desde 0 a 5 V.

Variables analógicas	
Entrada	Valor
A0:	2.97V
A1:	5.00V
A2:	0.00V
A5:	0.00V

Variables analógicas	
Entrada	Valor
A0:	0.00V
A1:	5.00V
A2:	0.00V
A5:	0.00V

Variables analógicas	
Entrada	Valor
A0:	5.00V
A1:	5.00V
A2:	0.00V
A5:	0.00V

Figura 4.19 Imagen de tablas de resultados obtenidas de la página WEB

4.7 Sumario

En este capítulo se describieron las pruebas que se hicieron a lo largo de la tesis también se presentan los resultados, estas pruebas sirvieron como base para desarrollar los diseños de las funciones que se presentaron en el capítulo 3, primero se describieron las pruebas que se refieren a la comunicación inalámbrica, comenzando con dos módulos y después aumentando a cuatro módulos, se sigue con la comunicación entre el módulo servidor y un módulo inalámbrico, al final se prueba el sistema completo.

En el capítulo siguiente se presentan las conclusiones a las que se llegó, con el trabajo presentado.

CAPÍTULO 5

CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan las conclusiones derivadas del trabajo realizado, también se presentan las propuestas para un trabajo futuro, que se puede llevar a cabo en el sistema, de tal forma que se puede agregar funciones para mejorar la funcionalidad del sistema, también se propone aumentar algunos elementos físicos como la antena y módulos inalámbricos adicionales.

5.1 Conclusiones

- En este trabajo de tesis se desarrolló un sistema capaz de monitorear variables digitales y analógicas.
- Usando el sistema se pueden manipular señales digitales para manejar algún tipo de actuador.
- Se desarrollaron funciones para lograr el envío de datos e instrucciones entre el módulo coordinador y los módulos remotos.
- Se desarrolló un algoritmo para enviar datos e instrucciones del módulo coordinador al módulo WEB.

- Se construyó un sitio WEB en donde se puede consultar el valor de las variables.
- Se comprobó que los datos adquiridos por los nodos remotos se visualizaron de forma correcta en el sitio WEB.
- Se comprobó que la manipulación de variables en los nodos remotos desde la página WEB fue correcta.
- Se desarrollaron funciones lógicas para toma de decisiones en nodos remotos.
- Con esto se logró mostrar en operación un sistema de monitorización y actuación de conexión inalámbrica con recursos sencillos.

5.2 Trabajo futuro

Como trabajo futuro se propone

- Continuar con el desarrollo de funciones que le den un mayor rango de operaciones para enviar y recibir datos.
- Debido a que en los módulos inalámbricos necesitan una antena para transmitir a las distancias mencionadas en el apartado 2.4 del capítulo 2, se debe de probar y seleccionar un tipo de antena para utilizarse en cada módulo inalámbrico.
- Al aumentar el número de funciones para el sistema, se debe de complementar el diseño de las páginas WEB, para manejar cada función que se desarrolle.
- Se propone aumentar en la medida de lo posible el número de tarjetas y probar el funcionamiento del sistema.
- Trabajar sobre el diseño de un módulo que incluya mayor número de periféricos.

APÉNDICE A FUNCIONES DEL NODO REMOTO

Aquí se presentan las funciones y programas reprobados desarrollados para la implementación de los nodos remotos diseñado en este trabajo, los programas fueron escritos en lenguaje C.

A.1 Función switch del Programa del nodo remoto

```
case 1 :TRISA=packetb.instruccion[1];
        TRISA &= 0b11011111;
        TRISA |= 0b00010001;

        TRISB=packetb.instruccion[2];

        TRISB &= 0b11001110;
        TRISB |= 0b00000010;
        ADCON1=packetb.instruccion[3];

        if(ADCON1==2)
        {
            setup_adc_ports(AN0_AN1_AN2_AN3_AN4);
        }
        else if(ADCON1==4)
        {
            setup_adc_ports(AN0_AN1_AN3);
        }
        else if(ADCON1==0X0E)
        {
            setup_adc_ports(AN0);
        }
        else
        {
            setup_adc_ports(NO_ANALOGS);
        }

        to=0;
        ACK=10;
        i=1;
        while(i!=0) //Se envia el paquete y se espera a que se reciba
        {
            i= ZMac_PutPacket(&txNode,&ACK,1);//Si i=0 el paquete se recibio

            to++;
            if(to>=30)
            break;
        }
```

```

break;

case 2 : PORTA=packetb.instruccion[1];
        PORTB=packetb.instruccion[2];

        for(j=0;j<=5;++j)
        {

            if(bit_test(TRISB,2+j)==0)
            output_bit( PIN_B2+j, bit_test(PORTB,2+j));
        }

        if((ADCON1!=2)&(ADCON1!=4)&(ADCON1!=0X0E))
        {
            if(bit_test(TRISA,0)==0)
            output_bit( PIN_A0, bit_test(PORTA,0));
        }

        if((ADCON1!=2)&(ADCON1!=4))
        {
            if( bit_test( TRI SA, 1)==0)
            output_bit( PIN_A1, bit_test(PORTA, 1));
        }
        if(ADCON1!=2)
        {
            if( bit_test( TRI SA, 2)==0)
            output_bit( PIN_A2, bit_test(PORTA, 2));
            if( bit_test( TRI SA, 5)==0)
            output_bit( PIN_A5, bit_test(PORTA, 5));
        }

        to=0;
        ACK=10;
        i=1;
        while(i!=0) //Se envía el paquete y se espera a que sea
reci bi do
        {
reci bi ó
            i = ZMac_PutPacket(&txNode, &ACK, 1); //Si i=0 el paquete se
            to++;
            if(to>=30)
            break;
        }

        break;

case 3 : for (j=0;j<=0x09; ++j)
        tabla.tag[j]=memo[j+1];
        to=0;
        i=1;
        ACK=10;
        while(i!=0) //Se envía el paquete y se espera a que sea
reci bi do
        {
            i = ZMac_PutPacket(&txNode, &tabla, 10);
            //Si i=0 el paquete se reci bi ó
            to++;
            if(to>=30)
            break;
        }

        break;
    }
    ZMac_DiscardRXAIL ();
}}

```

APÉNDICE B PROGRAMAS Y FUNCIONES IMPLEMENTADAS PARA EL MÓDULO COORDINADOR

B.1 Funciones de comunicación con el nodo remoto

Las funciones que aquí se presentan sirven a manera de instrucciones hacia los nodos remotos, se usan para enviarles los valores para configurar sus puertos, además de enviarle los valores que deben de tener los pines definidos como salidas y solicitarle que envíe una tabla de datos acerca de los valores actuales que tienen los puertos de entrada.

B.1.1 Función de envío de configuración

```
int conf_nr(int tris_a,int tris_b,int adcon1_r,ZMAC_NODE_INFO txNode)
{
//ZMAC_NODE_INFO txNode;
ZMAC_HEADER header;
int i=1, to=0, ACK, count;
struct
{
int instruccion[4];
} packet;
packet.instruccion[0]=1;
packet.instruccion[1]=tris_a;
packet.instruccion[2]=tris_b;
packet.instruccion[3]=adcon1_r;

ZMac_DiscardRXALL();
ZMac_StartListen();
to=0;
i=1;
while(i!=0)
{
i= ZMac_PutPacket(&txNode, &packet, 4);
to++;
if(to>=50)
return(0);
}
to=0;
```

```

        ACK=1;

        while (!ZMac_IsRxReady())
        {
            to++;

            if(to>=200)
                return(1);
        }
        count=ZMac_GetHeader(&header);
        ZMac_Getd(&ACK, 1);
        ZMac_StopListen();
        ZMac_DiscardRX();

    return(2);
}

```

B.1.2 Función de envío escritura de puertos

```

int write_nr(int port_a, int port_b, ZMAC_NODE_INFO txNode)
{
    ZMAC_HEADER header;
    int i=1, to=0, ACK, count;

    struct
    {
        int instruccion[4];
    } packet;

    packet.instruccion[0]=2;
    packet.instruccion[1]=port_a;
    packet.instruccion[2]=Port_b;
    packet.instruccion[3]=0;

    ZMac_DiscardRXAll();
    ZMac_StartListen();
    to=0;
    i=1;
    while(i!=0)
    {
        i = ZMac_PutPacket(&txNode, &packet, 4);
        to++;
        if(to>=50)
            return(0);
    }

    to=0;

    ACK=1;

    while (!ZMac_IsRxReady())
    {
        to++;

        if(to>=200)
            return(1);
    }
    count=ZMac_GetHeader(&header);
    ZMac_Getd(&ACK, 1);
    ZMac_StopListen();
    ZMac_DiscardRX();
    return(2);
}

```

B 1.3 Función envió de petición de tabla de valores

```
int read_nr(ZMAC_NODE_INFO txNode,int dir_ini)
{
    ZMAC_HEADER header;
    int16 to=0;
    int i=1,j,tag[10];
    int *ptr;
    struct
    {
        int instruccion[4];
    } packet;

    struct
    {
        int tag[10];
    } tabla;

    int8 count;

    ZMac_DiscardRXAll();
    ZMac_StartListen();

    packet.instruccion[0]=3;
    packet.instruccion[1]=0;
    packet.instruccion[2]=0;
    packet.instruccion[3]=0;
    packet.instruccion[4]=0;
    packet.instruccion[5]=0;
    packet.instruccion[6]=0;
    packet.instruccion[7]=0;
    to=0;
    i=1;
    while(i!=0)
    {
        i= ZMac_PutPacket(&txNode,&packet,4);
        to++;
        if(to>=50)
            return(0);
    }

    //delay_ms(1);
    to=0;

    while (!ZMac_IsRxReady())
    {
        to++;
        // lcd_putd(to);
        if(to>=300)
            return(1);
    }
    count=ZMac_GetHeader(&header);
    ZMac_Getd(&tabla,10);
    for (j=0;j<=9;++)
        memo_cor[j+dir_ini]=tabla.tag[j];
    ptr=&tabla.tag[0];
    ZMac_StopListen();
    ZMac_DiscardRX();
    return(2);
}
```

B.2 Programa de prueba de funciones del nodo coordinador

```
//////////
//Configuración de la tarjeta coordinador
//////////

a=conf_nr(0b11111111,0b11001101,0x02,txNode); //Se envía la configuración a
b=write_nr(0b00000000,00111000,txNode); //Los nodos remotos y el valor
a=conf_nr(0b00010001,0b00000001,0x0e,txNodeb); //de los puertos definidos como
b=write_nr(0b00010000,0b11100000,txNodeb); //salidas

a=conf_nr(0b00010001,0b00000001,0x06,txNodec);
b=write_nr(0b00110000,0b00010000,txNodec);

While(true)
{
    c=read_nr(txNode,0); // Se obtienen los valores de los puertos
    c=read_nr(txNodeb,10);
    c=read_nr(txNodec,20);
    delay_ms(1000);
    //Se envían los datos a la PC
    printf("tag1[%d]-->%d %d %d \n\r",j,memo_cor[1],memo_cor[10],memo_cor[20]);
}
}
```

B.3 Programa cargado al nodo coordinador

```
while(1)
{
    while(bkbhit)
    {
        x=bgetc();
        if((x==13)&(next_out>=8))
        {
            for(j=0;j<=7;++j)
            {
                paquete[7-j]=buffer[next_out-1-j];
            }
            if(paquete[0]==1)
            {
                vp++;
                //break;
            }
            else
                vp=0;
        }
    }

    if(vp!=0)
    {
        LED_OFF(PIN_LED3);
        if(paquete[1]==2)
        {
            serPutString(Valor);
        }
        else if(paquete[1]==3)
        {
            serPutString( val_ports);
        }
    }
}
```

```

else if((paquete[1]==4) || (paquete[1]==5))
{
    serPutString(cadena);
}

    vp=0;
    funcion=paquete[1];
    nodo=paquete[2];
    if(temp!=paquete[1])
    {
        for(k=0; k<=31; ++k);
        buffer[k]=0;

        temp=paquete[1];
    }
    break;
}

}

//lcd_putd(cont);

// delay_ms(10);

if(nodo==1)
txNode.shortAddress=DestShortAddress_a;
else if(nodo==2)
txNode.shortAddress=DestShortAddress_b;
else if(nodo==3)
txNode.shortAddress=DestShortAddress_c;

LED_ON(PIN_LED3);
if (funcion==2 || funcion==3)
{
    c=read_nr(txNode, 0);
}
if(funcion==3)
{
    b=write_nr(paquete[3], paquete[4], txNode);
}
if(funcion==4 || funcion==5)
{
    a=conf_nr(paquete[3], paquete[4], paquete[5], txNode);
}

if(a==0 || b==0 || c==0)
reset_cpu();

switch (funcion)
{
    case 2:
        Val or[2]=memo_cor[0];
        Val or[3]=memo_cor[1];
        Val or[4]=memo_cor[2];
        Val or[5]=memo_cor[4];

        break;
    case 3: val_ports[2]=memo_cor[5]; //portA
        val_ports[3]=memo_cor[6]; //portb
        val_ports[4]=memo_cor[7]; //lata
        val_ports[5]=memo_cor[8]; //latb

        break;
}
}

```

```
lcd_putd(paquete[6]);  
//lcd_putd(8);
```

```
LED_OFF(PIN_LED3);  
}
```

APÉNDICE C PROGRAMA DEL MÓDULO SERVIDOR

```
if(AppConfig.PortCfg_n.pag==2)
{
    serPutStringb(insta);

    while(serIsGetReady())
    {
        x=serGetByte();
        btemp[cont]=x;
        ++cont;
        if (cont>=32)
            cont=0;
        if((x==13)&(cont>=8))
        {
            for(j=0;j<=7;++j)
            {
                paquete[7-j]=btemp[cont-1-j];
            }
            if(paquete[0]==1)
            {
                ++vp;
                break;
            }
            else
                vp=0;
        }
    }

    if(vp!=0)
    {
        if(paquete[1]==2)
        {
            for(j=0;j<=4;++j)
            {
                tag[j]=paquete[j+2];
            }
        }
        vp=0;
        cont=0;
    }
}
else if(AppConfig.PortCfg_n.pag==3)
{
    serPutStringb(instb);

    while(serIsGetReady())
    {
        x=serGetByte();
        btemp[cont]=x;
        ++cont;
```

```

        if (cont>=32)
        cont=0;
        if((x==13)&(cont>=8))
        {
            for(j=0;j<=7;++j)
            paquete[7-j]=btemp[cont-1-j];

            if(paquete[0]==1)
            {
                ++vp;
                break;
            }
            else
            vp=0;
        }
    }

    if(vp!=0)
    {
        if(paquete[1]==3)
        {
            AppConfig.PortRw.porta=paquete[2];
            AppConfig.PortRw.portb=paquete[3];
            AppConfig.LatRw.lata=paquete[4];
            AppConfig.LatRw.latb=paquete[5];
        }

        vp=0;
        cont=0;
    }
}

else if((AppConfig.PortCfg_n.pag==4)||((AppConfig.PortCfg_n.pag==5))

{
    serPutStringb(instc);

    while(serIsGetReady())
    {
        x=serGetByte();
        btemp[cont]=x;
        ++cont;
        if (cont>=32)
        cont=0;
        if((x==13)&(cont>=8))
        {
            for(j=0;j<=7;++j)
            paquete[7-j]=btemp[cont-1-j];

            if(paquete[0]==1)
            {
                ++vp;
                break;
            }
            else
            vp=0;
        }
    }

    if(vp!=0)
    {

```

```
vp=0;  
cont=0;  
}  
  
}
```

APÉNDICE D FUNCIONES PRINCIPALES DEL ARCHIVO ZMD.H

```
#include <18F452.h>
#fuses HS,NOWDT,NOLVP
#use delay(clock=1000000)
#use rs232(baud=57600, xmit=PIN_C6,rcv=PIN_C7)
#define BUFFER_SIZE 32

BYTE buffer[BUFFER_SIZE];
BYTE next_in = 0;
BYTE next_out = 0;
#include <ZMD44102.H>

#define BUTTON_PRESSED() (!input(PIN_A4))
#define PIN_LED1 PIN_B4
#define PIN_LED2 PIN_B5
#define PIN_LED3 PIN_A5
#define LED_ON output_low
#define LED_OFF output_high

#define EXP_OUT_ENABLE PIN_B2
#define EXP_OUT_CLOCK PIN_B4
#define EXP_OUT_DO PIN_B5
#define NUMBER_OF_74595 3
#include <74595.c>

#define bkbhit (next_in!=next_out)

BYTE memo_cor[20];
const char digit_format[10]={
0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90
};

void leds_off(void) {
    LED_OFF(PIN_LED1);
    LED_OFF(PIN_LED2);
    LED_OFF(PIN_LED3);
}

void lcd_clear(void) {
    int8 digits[3]={0xFF,0xFF,0xFF};
    write_expanded_outputs(&digits[0]);
    output_low(EXP_OUT_ENABLE);
    leds_off();
}

void lcd_putd(int16 num) {
    int8 digits[3];
    if (num>999) {num=999;}

    digits[0]=num / 100;
    digits[1]=(num % 100) / 10;
    digits[2]=num % 10;
```

```

digits[0]=digit_format[digits[0]];
digits[1]=digit_format[digits[1]];
digits[2]=digit_format[digits[2]];

write_expanded_outputs(&digits[0]);
output_low(EXP_OUT_ENABLE);

LED_OFF(PIN_LED1);
LED_OFF(PIN_LED2);
}

void lcd_putdc(int16 numa,int16 numb,int16 numc) {
    int8 digits[3];
    if (numa>999) {numa=999;}
    if (numb>999) {numb=999;}
    if (numc>999) {numc=999;}

    digits[0]=numa%10;
    digits[1]=numb%10;
    digits[2]=numc%10;

    digits[0]=digit_format[digits[0]];
    digits[1]=digit_format[digits[1]];
    digits[2]=digit_format[digits[2]];

    write_expanded_outputs(&digits[0]);
    output_low(EXP_OUT_ENABLE);

    LED_OFF(PIN_LED1);
    LED_OFF(PIN_LED2);
}

BYTE bgetc() {
    BYTE c;

    while(!bkbhit) ;
    c=buffer[next_out];
    next_out=(next_out+1) % BUFFER_SIZE;
    return(c);
}

void serPutString(int *s) {
    int j;

    for(j=0;j<=7;++j)
    {
        Putc(*(s+j));
    }
}

void myZMacInit(void)
{
    ZMac_Init();

    ZMac_SetMyShortAddress(MyShortAddress);
    ZMac_SetMyPanId(MyPanId);
    ZMac_UseMyShortAddress();
    //ZPhy_SetChannel(MyRFChannel);
}

```

REFERENCIAS

- [1] Gómez del Villar Víctor Enrique, Diseño y construcción de sistema para medición de Ph, integrando señales para la actuación de válvulas/actuadores, Tesis de Maestría. INAOE 2005.
- [2] Dacfe Dzong, Jan Endresen, Christoffer Apneseth, Jan-Erik Frey: 'Design and Implementation of a Real-Time Wireless Sensor/Actuator Communication System' In: 'ETFA05, Catania, 19–22 September 2005.
- [3] Feng Xia , Yu-Chu Tian, Yanjun Li and Youxian Sun ,Wireless Sensor/Actuator Network Design for Mobile Control Applications, Sensors 2007, 7, 2157-2173.
- [4] Alberto Requena, Luis Manuel Tomas, Inalámbricos, autónomos y redes de sensores, Apuntes ciencia y tecnología.
<http://www.um.es/actualidad/hemerotecapdf/ef/ef-20070311-4.pdf>
- [5] IEEE Std 1451.4, "Draft Standard for a Smart Transducer Interface for Sensors and Actuators – Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats." Institute of Electrical and Electronics Engineers.
- [6] <http://www.zigbee.org/>, 2006.
- [7] ZigBee Alliance, Network Specification, Version 1.0, Dec. 2004.
- [8] IEEE Std. 802.15.1-2003: "IEEE Standard for Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Based on Bluetooth.
- [9] IEEE Std. 802.15.4-2003: "IEEE Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)".

- [10] Gang Ding; Sahinoglu, Z.; Orlik, P.; Jinyun Zhang; Bhargava, B. Tree-Based Data Broadcast in IEEE 802.15.4 and ZigBee Networks, Transactions on Mobile Computing Volume 5, Issue 11, Nov. 2006 Page(s):1561 – 1574.
- [11] Dan Strassberg, Simple Networks will free many sensors from wires, EDN, April 2006.
- [12] David M. Chandler, Tobin Anthony, and Rick Darnell, Running a Perfect Web Site With Windows, Que; Pap/Cdr edition (May 1996).
- [13] Hutchison, David, Local area network architectures, 1988 Addison-Wesley.
- [14] David M. Chandler, Tobin Anthony, and Rick Darnell, Running a Perfect Web Site With Apache Que; Pap/Cdr Su edition (May 1996)
- [15] Morgan Denis, Revised data interface standard, 1984, Data application, Motorola.
- [16] EIA/TIA-232 3 Driver x 5 Receiver, Data sheet, Nacional semiconductor, Rev 1996.
- [17] Wireless Development Kit - ZMD Edition.
<http://www.ccsinfo.com>
- [18] PIC18FXX2 Data Sheet, Microchip Technology Incorporated, 2002.
- [19] ZMD44102 Data Sheet and User Manual.
<http://www.zmd.de/zigbee.php?content=zig&product=zmd44102>
- [20] Wireless ZMD Exercise Book, CCSINFO, 2006.
- [21] Dogan Ibrahim, PIC Basic Projects: 30 Projects using PIC BASIC and PIC BASIC PRO (Paperback), Newness.
- [22] “Modtronix SBC44EC Web” Server V2.03. Modular Electronic Solutions (MODTRONIX Engineering). Nota de aplicación.
http://www.modtronix.com/product_info.php?products_id=73. Enero del 2006.
- [23] 24AA256/24LC256/24FC256256K I2C™ CMOS Serial EEPROM, Data

Sheet, Microchip Technology Inc, 2002.

- [24] Manual de usuario del compilador pcw de ccs, <http://www.ccsinfo.com>
- [25] MPLAB® C18 C COMPILER, user guide, Microchip Technology Incorporated, 2002.
- [26] D. E. Comer, D.L. Stevens, Internetworking with TC/IP, Prentice Hall Vol II, 1993.
- [27] Nilesh Rajbharti. "The Microchip TCP/IP Stack" AN833. Microchip Technology Inc. Nota de Aplicación.
<http://ww1.microchip.com/downloads/en/AppNotes/00833b.pdf>.