

## **TRABAJO ESPECIAL DE GRADO**

# **ESTUDIO DE FACTIBILIDAD DEL USO DE TECNOLOGÍA INALÁMBRICA PARA TELEMETRÍA EN AMBIENTES INDUSTRIALES CON DISPOSITIVOS INALÁMBRICOS CONVENCIONALES.**

Presentado ante la Ilustre  
Universidad Central de Venezuela  
por el Br. Víctor R. Fernández A.  
para optar al Título de  
Ingeniero Electricista

Caracas, 2010

## **TRABAJO ESPECIAL DE GRADO**

# **ESTUDIO DE FACTIBILIDAD DEL USO DE TECNOLOGÍA INALÁMBRICA PARA TELEMETRÍA EN AMBIENTES INDUSTRIALES CON DISPOSITIVOS INALÁMBRICOS CONVENCIONALES.**

Profesor Guía: Ing. Lorena Núñez  
Tutor Industrial: Ing. Cesar Monagas

Presentado ante la Ilustre  
Universidad Central de Venezuela  
por el Br. Víctor R. Fernández A.  
para optar al Título de  
Ingeniero Electricista

Caracas, 2010

## CONSTANCIA DE APROBACIÓN

Caracas, 04 de mayo de 2010

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller, Víctor R. Fernández A., titulado:

### **“ESTUDIO DE FACTIBILIDAD DEL USO DE TECNOLOGÍA INALÁMBRICA PARA TELEMETRÍA EN AMBIENTES INDUSTRIALES CON DISPOSITIVOS INALÁMBRICOS CONVENCIONALES”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

  
Prof. Paolo Maragno  
Jurado

  
Prof. David Sirit  
Jurado

  
Prof. Lorena Núñez  
Prof. Guía

## **DEDICATORIA**

A mis padres, Carlos y Yolanda,  
mis hermanos Carlos y César,  
y a toda mi familia por  
apoyarme siempre.

## **AGRADECIMIENTOS**

A Dios todopoderoso, que en todos estos años de estudio, iluminó mi camino, y me dio energía para llegar a la meta.

Un especial agradecimiento a mi tutor industrial Ingeniero César Monagas, por haberme dado la oportunidad de desarrollar este proyecto, su disposición a orientarme y corregir el rumbo cuando fue necesario.

A mi profesora guía, Profesora Lorena Núñez, por su valiosa orientación y preocupación, así como su apoyo en la realización de este trabajo de grado.

A todo el personal de Corporación 80-60 C.A, por su solidaridad y colaboración cuando los necesité, en especial al profesor Luis Alvarado, de quien apreciaré siempre sus orientaciones y enseñanzas

A mi hermano Carlos que sin su ayuda nunca lo hubiese logrado.

A mis padres por su apoyo, mis hermanos, en general a toda mi familia que siempre estuvo pendiente de mi recorrido académico.

A Andryus Mijares por su compañía y amistad en estos años de carrera, así como a todos mis amigos.

Gracias a la Universidad Central de Venezuela por permitirme desarrollar como persona, profesional y como el ciudadano que el país requiere.

**Fernández A., Víctor R.**

**ESTUDIO DE FACTIBILIDAD DEL USO DE TECNOLOGIA INALAMBRICA  
PARA TELEMETRIA EN AMBIENTES INDUSTRIALES CON DISPOSITIVOS  
INALAMBRICOS CONVENCIONALES**

**Profesor Guía: Ing. Lorena Núñez. Tutor Industrial: Ing. Cesar Monagas.  
Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica.  
Ingeniero Electricista. Opción: Comunicaciones. Institución: Corporación 80-60  
C.A. Trabajo de Grado. 2010. 85h. + anexos.**

**Palabras Claves:** Wi-Fi, Bluetooth, ZigBee, subestación eléctrica, programación dsPIC, Tasa de error del bit.

**Resumen.** Este trabajo propone la implementación de una comunicación inalámbrica entre una unidad de procesamiento de analógicos (UPA) y una unidad terminal remota (UTR), ubicadas en subestaciones eléctricas. Se pretende entonces realizar pruebas en campo, a fin de determinar la factibilidad de implementar la comunicación inalámbrica. Para ello se estudiaron los estándares de las tecnologías Wi-Fi, Bluetooth y ZigBee con el propósito de seleccionar los módulos a usar en estos ambientes. Se realizó el diseño de un algoritmo de medición de tasa de error del Bit (BER) para ser implementado en la programación de un dsPIC. Se elaboró el diseño de un circuito impreso para el montaje de los módulos. Posteriormente se realiza las mediciones del BER en las pruebas ejecutadas en las subestaciones eléctricas.

# ÍNDICE GENERAL

	<b>Pág</b>
<b>CONSTANCIA DE APROBACIÓN</b> .....	iii
<b>DEDICATORIA</b> .....	iv
<b>RECONOCIMIENTOS Y AGRADECIMIENTOS</b> .....	v
<b>RESUMEN</b> .....	vi
<b>ÍNDICE GENERAL</b> .....	vii
<b>ÍNDICE DE TABLAS</b> .....	xiii
<b>ÍNDICE DE FIGURAS</b> .....	xiv
<b>ACRÓNIMOS</b> .....	xv
<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I</b>	
<b>DEFINICIÓN DEL PROBLEMA</b>	
1.1 PLANTEAMIENTO DEL PROBLEMA.....	5
1.2 OBJETIVOS DEL PROYECTO.....	6
1.2.1 Objetivo General.....	6
1.2.2 Objetivos Específicos.....	6
1.3 JUSTIFICACIÓN.....	7
1.4 ALCANCE.....	7
1.5 LIMITACIONES.....	7
<b>CAPÍTULO II</b>	
<b>MARCO TEÓRICO</b>	
2.1 Redes Inalámbricas de Área Local.....	9

2.2 Wi-Fi.....	10
2.2.1 Frecuencia de operación.....	10
2.2.2 Relación entre IEEE 802.11 y Wi-Fi.....	10
2.2.3 Estándar 802.11.....	11
2.2.3.1 Definiciones IEEE 802.11.....	12
2.2.4 Modos de red.....	13
2.2.4.1 Red Infraestructura.....	13
2.2.4.2 Red Ad Hoc.....	14
2.2.5 Arquitectura.....	15
2.2.6 Capa Control de Acceso al Medio.....	15
2.2.6.1 Trama MAC.....	16
2.2.7 Capa Física.....	18
2.2.7.1 Espectro expandido por secuencia directa (DSSS).....	19
2.2.7.2 Espectro expandido por salto de frecuencia (FSSS).....	19
2.2.7.3 Infrarrojo (IR).....	20
2.2.7.4 Multicanalización por división de frecuencias ortogonales (OFDM).....	20
2.2.7.5 Trama PPDU.....	21
2.2.8 Versiones IEEE 802.11.....	22
2.2.8.1 802.11a.....	22
2.2.8.2 802.11b.....	22
2.2.8.3 802.11c.....	22
2.2.8.4 802.11d.....	22
2.2.8.5 802.11e.....	23
2.2.8.6 802.11f.....	23
2.2.8.7 802.11g.....	23
2.2.8.8 802.11h.....	23
2.2.8.9 802.11i.....	23
2.2.9 Sensibilidad.....	23
2.2.10 Tasa de Error del Bits (BER).....	24

2.3 Redes Inalámbricas de Área Personal.....	25
2.4 Relación IEEE 802.15.1 y Bluetooth.....	26
2.5 Bluetooth.....	27
2.5.1 Banda de frecuencias y canales.....	27
2.5.2 Pila de protocolos.....	28
2.5.3 Capas.....	29
2.5.3.1 Capa de Radio.....	29
2.5.3.2 Capa Bandabase.....	29
2.5.3.3 Capa de gestión de enlace.....	30
2.5.4 Protocolos.....	31
2.5.4.1 Protocolo de gestión de enlace (LMP).....	31
2.5.4.2 Protocolo de adaptación y control de enlace lógico (L2CAP).....	31
2.5.4.3 Protocolo RFCOMM.....	31
2.5.4.4 Protocolo de descubrimiento de servicio (SDP).....	31
2.5.4.5 Protocolo Interfaz de Control (HCI).....	32
2.5.5 Potencia.....	32
2.5.6 Modulación.....	32
2.5.7 Perfiles.....	33
2.5.7.1 Perfil de acceso Genérico.....	33
2.5.7.2 Perfil de aplicación de descubrimiento de servicio.....	34
2.5.7.3 Perfil de puerto serie.....	34
2.5.7.4 Perfil de telefonía inalámbrica.....	34
2.5.7.5 Perfil de auricular.....	34
2.5.7.6 Perfil de Redes de Área Personal.....	34
2.5.7.7 Perfil de manos libres.....	35
2.5.7.8 Perfil de intercambio de objetos.....	35
2.5.7.9 Perfil de sincronismo.....	35
2.5.8 Estructura del Paquete.....	35
2.5.8.1 Código de acceso.....	36
2.5.8.2 Cabecera.....	36

2.5.8.3 Carga Útil.....	37
2.5.9 Redes.....	37
2.5.9.1 Piconet.....	37
2.5.9.2 Scatternet.....	38
2.5.10 Emparejamiento.....	39
2.5.11 Tipos de dispositivos.....	40
2.5.12 Sensibilidad y Tasa de Error del Bits (BER).....	41
2.6 Estándar 802.15.4.....	41
2.7 Relación IEEE 802.15.4 y ZigBee.....	43
2.7.1 Dispositivos IEEE 802.15.4.....	43
2.7.1.1 Dispositivos de función total.....	44
2.7.1.1 Dispositivos de función reducida.....	44
2.7.2 Dispositivos ZigBee.....	44
2.7.2.1 Dispositivo Coordinador ZigBee.....	44
2.7.2.2 Dispositivo Router ZigBee.....	45
2.7.2.3 Dispositivo Final.....	45
2.7.3 Topologías de Red.....	45
2.7.3.1 Topología estrella.....	46
2.7.3.2 Topología punto a punto.....	46
2.7.3.3 Topología Cluster-tree.....	46
2.7.4 Capa física.....	47
2.7.4.1 Detección de energía.....	48
2.7.4.2 Indicación de calidad de enlace.....	48
2.7.4.3 CCA.....	48
2.7.4.4 Servicio de datos PHY.....	49
2.7.4.5 Servicio de administración PHY.....	49
2.7.5.6 Trama PPDU.....	49
2.7.5 Modulación.....	50
2.7.6 Capa de control de acceso al medio (MAC).....	50
2.7.7 Estructura del paquete.....	51

2.7.7.1 Beacon.....	51
2.7.7.2 Datos.....	51
2.7.7.3 Acknowledgement.....	52
2.7.7.4 Control.....	52
2.7.8 Capa de aplicación.....	52
2.7.8.1 Perfiles.....	52
2.7.8.2 Clusters.....	53
2.7.9 Comunicación.....	53
2.7.10 Sensibilidad.....	54
2.8 Comparación de tecnologías.....	54
2.9 Microcontrolador.....	55
2.10 Comunicación Serial RS-232.....	55
2.11 Familia dsPIC.....	56

### **CAPÍTULO III**

#### **DESARROLLO DEL PROYECTO**

3.1 Selección de dispositivos.....	57
3.2 Selección del microcontrolador.....	59
3.3 Características de las antenas.....	60
3.4 Descripción de algoritmo de medición de BER.....	60
3.5 Diseño de los circuitos electrónicos.....	68

### **CAPÍTULO IV**

#### **REALIZACIÓN DE PRUEBAS**

4.1 Pruebas en Corporación 80-60.....	70
4.2 Pruebas en subestación de media tensión A.....	71
4.3 Pruebas en subestación de media tensión B.....	73
4.4 Pruebas en subestación de media tensión C.....	74
4.5 Pruebas en subestación de media tensión D.....	76
ANÁLISIS DE RESULTADOS.....	78

<b>CONCLUSIONES</b> .....	79
<b>RECOMENDACIONES</b> .....	81
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	82
<b>BIBLIOGRAFÍA</b> .....	84
<b>ANEXOS</b> .....	86

## ÍNDICE DE TABLAS

	Pág
1. Comparación de tecnologías.....	54
2. Módulos Wi-Fi.....	57
3. Módulos Bluetooth.....	58
4. Módulos ZigBee.....	59
5. Función de las variables.....	67
6. Resultados en sede de Corporación 80-60.....	70
7. Resultado con cambio de caracteres en sede de Corporación 80-60.....	71
8. Resultados en Subestación de media tensión A,.....	72
9. Resultados en Subestación de media tensión B.....	74
10. Resultados en Subestación de media tensión C, Celdas 1.....	75
11. Resultados en Subestación de media tensión C, Celdas 2.....	75
12. Resultados en Subestación de media tensión D, celdas.....	77
13. Resultados en Subestación de media tensión D.....	77
14. Resultados en Transformador 1,Subestación de media tensión D.....	77

## ÍNDICE DE FIGURAS

	Pág
1. Red Infraestructura.....	14
2. Red Ad Hoc.....	15
3. Formato de trama MAC 802.11.....	17
4. Técnica FHSS.....	20
5. Trama PDU 802.11.....	21
6. Clases de WPANs.....	26
7. Pila de Protocolos Bluetooth.....	28
8. Estructura de paquete Bluetooth.....	35
9. Estructura de paquete mejorado Bluetooth.....	37
10. Piconet.....	38
11. Scatternet.....	39
12. Topología de redes ZigBee.....	45
13. Trama PDU 802.15.4.....	49
14. Diagrama de flujo función principal.....	61
15. Diagrama de flujo función principal continuación... ..	62
16. Diagrama de flujo Función verifica trama.....	63
17. Diagrama de flujo Función calcula bits.....	64
18. Diagrama de flujo Función contbit.....	65
19. Estructura de Subestación de media tensión A.....	72
20. Estructura de Subestación de media tensión B.....	73
21. Estructura de Subestación de media tensión C.....	75
22. Estructura de Subestación de media tensión D.....	76

## ACRÓNIMOS

<b>ASCII</b>	American Standard Code for Information Interchange
<b>ACK</b>	Acknowledgment
<b>AP</b>	Access Point
<b>AT</b>	Attention
<b>ACL</b>	Asynchronous Connection-Less
<b>BER</b>	Bit Error Rate
<b>BSS</b>	Basic Service Set
<b>BSSID</b>	Basic Service Set Identifier
<b>CLH</b>	Cluster Head
<b>CID</b>	Cluster Identifier
<b>CSMA/CA</b>	Carrier Sense Multiple Access – Collision Avoidance
<b>CRC</b>	Cyclic Redundancy Check
<b>CTS</b>	Clear to Send
<b>CCK</b>	Complementary Code Keying
<b>CTP</b>	Cordless telephony profile
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>DS</b>	Distribution System
<b>DA</b>	Destination Address
<b>DLCI</b>	Data Link Connection Identifier
<b>ESS</b>	Extended Service Set
<b>ED</b>	Energy Detection
<b>FER</b>	Frame Error Rate
<b>FCS</b>	Frame Check Sequence
<b>FHSS</b>	Frequency Hopping Spread Spectrum
<b>FFD</b>	Full Function Device
<b>GAP</b>	Generis Access Profile

<b>GOEP</b>	Generis Object Exchange Profile
<b>GTS</b>	Guaranteed Time Slots
<b>HFP</b>	Hands-Free Profile
<b>HS</b>	Headset
<b>HR</b>	High Rate
<b>HCI</b>	Host Controller Interface
<b>HEC</b>	Header Error Check
<b>IEEE</b>	Institute of Electrical and Electronic Engineers
<b>ISP</b>	Internet Service Provider
<b>IBSS</b>	Independent Basic Service Set
<b>IR</b>	Infrared
<b>IP</b>	Internet Protocol
<b>ISM</b>	Industrial Scientific Medical
<b>LR-WPAN</b>	Low-rate Wireless Personal Area Network
<b>LAN</b>	Local Area Network
<b>LLC</b>	Logical Link Control
<b>LM</b>	Link Manager
<b>L2CAP</b>	Logical Link Control and Adaptation
<b>LQI</b>	Link Quality Indication
<b>MAC</b>	Medium Access Control
<b>OSI</b>	Open Systems Interconnection
<b>PAN</b>	Personal Area Network
<b>POS</b>	Personal Operating Space
<b>PDA</b>	Personal Digital Assistant
<b>PCB</b>	Printed Circuit Board
<b>PCF</b>	Point Coordination Function
<b>PPDU</b>	PHY Protocol Data Units
<b>PSDU</b>	PLCP Service Data Units
<b>PHY</b>	Physical
<b>PPP</b>	Point-to-Point Protocol

<b>QoS</b>	Quality of Service
<b>RA</b>	Receiver address
<b>RTS</b>	Request to Send
<b>RFD</b>	Reduced Function Device
<b>STA</b>	Station
<b>SSID</b>	Service Set Identity
<b>SA</b>	Source Address
<b>SCO</b>	Synchronous Connection-Oriented
<b>SIG</b>	Special Interest Group
<b>SDAP</b>	Service discovery application profile
<b>SP</b>	synchronization profile
<b>SEQN</b>	Sequence Number
<b>TA</b>	Transmitter Address
<b>TCP</b>	Transference Control Protocol
<b>TDD</b>	Time Division Duplex
<b>UDP</b>	User Datagram Protocol
<b>UPA</b>	Unidad de Procesamiento de Analogicos
<b>UTR</b>	Unidad Terminal Remota
<b>WPAN</b>	Wireless Personal Area Network
<b>WLAN</b>	Wireless Local Area Network
<b>Wi-Fi</b>	Wireless Fidelity
<b>WECA</b>	Wireless Ethernet Compatibility Alliance
<b>WM</b>	Wireless Medium
<b>ZC</b>	ZigBee Coordinator
<b>ZR</b>	ZigBee Router
<b>ZED</b>	ZigBee End Device

## INTRODUCCIÓN

La tecnología inalámbrica en los últimos años ha revolucionado el modo en que las personas se comunican en sus puestos de trabajo u hogares, esto ha permitido el surgimiento de una variedad de tecnologías, las cuales, están orientadas a satisfacer las necesidades de las personas en cuanto a movilidad y flexibilidad se refiere.

Por consiguiente muchas empresas han decidido estudiar distintas tecnologías inalámbricas a fin de buscar la más conveniente para sus propósitos empresariales. Una de estas empresas, es Corporación 80-60 C.A, la cual, brinda servicios de asesoría, apoyo técnico, evaluación de recursos, desarrollo de obras, diseño y elaboración de sistemas, equipos y dispositivos específicos y desarrollo de software especializado.

Corporación 80-60 C.A, basa su organización en cuatro departamentos coordinados por una Gerencia General. Estos departamentos son: Departamento de Ingeniería y Desarrollo, Departamento de Operaciones, Departamento de Administración y Finanzas, y Departamento de Producción. Esta empresa realiza servicio técnico y mantenimiento de equipos que se encuentran en las instalaciones del cliente, como también hace mantenimiento en subestaciones en donde la transferencia de datos en éstas, se efectúa mediante una Unidad de Procesamiento de Analógicos (UPA) instalada en las celdas de la subestación. La UPA es un sistema de adquisición de datos por telemetría, utilizado para la supervisión de procesos dentro de una subestación eléctrica. Este dispositivo posee capacidad para obtener datos analógicos calculados a partir del “principal” de una celda y sus respectivos circuitos, los cuales posteriormente son transmitidos a la Unidad Terminal Remota.

La unidad Terminal Remota (UTR) es un sistema de adquisición de datos por telemetría para la supervisión y control de procesos a distancia, especialmente diseñada para aplicaciones de la industria eléctrica, con el cual se pueden obtener datos analógicos y digitales, además de realizar controles en campo. La característica más resaltante de esta unidad es su capacidad de conectarse con las Unidades de Procesamiento de Analógicos (UPA) para la lectura de parámetros analógicos. La adquisición de puntos digitales y los controles se logra mediante tarjetas instaladas dentro de la UTR.

La UTR se encuentra instalada fuera de las celdas de la subestación, por consiguiente la transmisión de datos entre ella y las UPAs se realiza a través de fibra óptica colocada bajo suelo para su mayor seguridad, lo que trae como consecuencia una gran inversión económica y poca facilidad de instalación.

A causa de ello la empresa se planteó la necesidad de implementar tecnología inalámbrica en la subestación para la sustitución del cableado, a fin de reducir costos y permitir una fácil colocación e instalación.

La comunicación inalámbrica por medio de dispositivos convencionales se ha convertido en el mejor medio de comunicación debido a su bajo costo y fácil implementación en comparación con los medios de comunicación tradicional como son los cables de par trenzado no blindado (UTP), coaxial y fibra óptica, sin embargo, su mayor desventaja es la velocidad de transmisión y la interferencia producida por otros medios.

Los estándares más utilizados en la comunicación inalámbrica son Bluetooth, Wi-Fi y ZigBee; estos tres estándares surgen de la necesidad de las empresas de telecomunicaciones en conectar varios dispositivos sin la necesidad de cables, lo cual permite que trabajen en la banda destinada a equipos en el área industrial, científica y médica (ISM por sus siglas en Inglés), la cual está abierta a

todo el mundo sin la necesidad de licencia por parte del ente rector del espectro radioeléctrico.

La realización de este trabajo formula una propuesta de comunicación inalámbrica en un ambiente industrial a ser implementada por Corporación 80-60.

Para llevar a cabo esta propuesta, este trabajo está dividido en cuatro capítulos.

El **Capítulo 1**, plantea el problema, así como los objetivos del proyecto tanto el objetivo general como el específico, esto permite definir la magnitud de este trabajo, igualmente señala la justificación, alcance y limitaciones.

En el **Capítulo 2**, se documentan los aspectos teóricos de las tecnologías, además de detallar las características de cada estándar, señalado por el Instituto de Ingenieros Eléctricos y Electrónicos para la posterior selección de los dispositivos, Wi-Fi, Bluetooth y ZigBee.

El **Capítulo 3** Se realiza un estudio de los manuales técnicos de los dispositivos que existen en el mercado con el fin de determinar los módulos que mejor se adapten al ambiente industrial. Además se selecciona el microcontrolador que cumple ciertas características. Igualmente se aprende e implementa la programación de un microcontrolador a ser conectado al módulo Wi-Fi, ZigBee o Bluetooth para que realice la medición de la tasa de error.

En el **Capítulo 4**, se da a conocer las pruebas realizadas, incluso los resultados obtenidos en ellas. Así mismo, se comparan los resultados obtenidos con los recomendados por cada estándar, además de hacer un análisis de los resultados obtenidos.

Finalmente, se establecen conclusiones y recomendaciones que, basadas en la realización de pruebas de campo, refuerzan la implementación de sistemas de comunicación inalámbrica en ambientes industriales.

# **CAPÍTULO I**

## **DEFINICION DEL PROBLEMA**

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

La inversión del cableado de fibra óptica en la subestación es costosa y requiere de personal especializado para el manejo de la fibra, además de la poca facilidad de instalación del cableado en la subestación desde la UPA hacia la UTR. A pesar de que la fibra óptica no se ve afectada por alteraciones de voltaje o corriente en las líneas de potencia por interferencia electromagnética, proporciona un ancho de banda extremadamente grande y tiene una atenuación de potencia muy baja, de tal manera que pueden emplearse en ambientes industriales expuestos a condiciones muy severas en donde otro tipo de cable sería inadecuado. Para un estudio teórico de la fibra óptica se necesita de una habilidad que solo algunos ingenieros la tienen, este es un aspecto negativo en la implementación de la fibra en ambientes industriales.

El desarrollo de los estándares inalámbricos ha aumentado la implementación de estos en dispositivos que permiten la comunicación. Por tal motivo la empresa Corporación 80-60 está evaluando realizar la transferencia de datos vía aire, y para ello se realizó una comparación entre los estándares Bluetooth, Wi-Fi y ZigBee con la finalidad de obtener el mejor dispositivo que se adaptara al ambiente imperante en subestaciones eléctricas de alta y media tensión.

Además el principal problema que se planteó para la colocación de dispositivos inalámbricos, fueron las condiciones de trabajo que existen en la subestación, en donde por motivo de la ubicación de la celda puede haber campos electromagnéticos generados por las corrientes inducidas de los transformadores

ubicados en la cercanía de la celda, de modo que aparte de elevar la temperatura, generan ruido que puede causar un efecto negativo sobre la transferencia de datos vía comunicación inalámbrica.

## **1.2 OBJETIVOS DEL PROYECTO**

### **1.2.1 Objetivo General**

Determinar el estándar de comunicación inalámbrica, que mejor se adapte a los requerimientos técnicos que Corporación 80-60 necesita para la optimización del envío de datos.

### **1.2.2 Objetivos Específicos**

- Comprender la infraestructura de los estándares Wi-Fi, Bluetooth y ZigBee, a fin de conocer su funcionamiento a nivel de protocolos.
- Estudiar la información técnica de los dispositivos existentes en el mercado de cada estándar, a fin de seleccionar los módulos Wi-Fi, Bluetooth y ZigBee que tengan mejores características con respecto a los demás.
- Implementar un algoritmo para la medición de tasa de error del bit (BER) por medio de la programación de un circuito integrado programable (PIC).
- Realizar conexiones pertinentes a los módulos para su funcionamiento.
- Comprobar el funcionamiento de los dispositivos mediante la utilización de comandos AT (Attention command) para la realización de mediciones prácticas.
- Analizar los resultados para comprobar que los valores del BER práctico concuerden con el valor dado por el fabricante y a su vez realizar la comparación entre los diferentes estándares.

- Seleccionar, en base al estudio realizado, el mejor dispositivo que cumpla con los requerimientos propuestos.

### **1.3 JUSTIFICACIÓN**

La implementación de cables de fibra óptica en las subestaciones para su uso en el envío de datos desde la celda donde se encuentra la unidad de procesamiento de analógicos (UPA) hasta la unidad terminal remota (UTR), requiere que los conectores de la fibra estén bien acoplados, para ellos, se necesitan herramientas especializadas además de una elevada inversión, por tal motivo, la empresa Corporación 80-60 se planteó la necesidad de utilizar otro medio de transmisión con un costo menor y de fácil instalación.

### **1.4 ALCANCE**

El presente estudio no abarca la conexión de la Unidad de Procesamiento de Analógicos con la Unidad Terminal Remota a través de tecnología inalámbrica, sino, validar mediante pruebas prácticas la factibilidad de una comunicación inalámbrica en ambientes industriales como por ejemplo, una subestación eléctrica.

### **1.5 LIMITACIONES**

La ejecución de las pruebas de campo requiere los permisos por parte del cliente para ingresar a sus instalaciones, esto trae como consecuencia, la elaboración anticipada de una solicitud de acceso, esperar que esta sea coordinada por el personal de Operaciones en base a la disponibilidad del personal de mantenimiento del cliente y que no exista otro trabajo en esa subestación, aún cuando esta solicitud sea aprobada, está expuesta a su cancelación en cualquier momento debido a fallas en la

red eléctrica o emergencias operacionales, por lo que la disponibilidad de tiempo para realizar las pruebas en las subestaciones fue precaria.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1 REDES INALÁMBRICA DE ÁREA LOCAL**

Con la creciente demanda de servicios multimedia a través de internet mediante las redes de área local, se necesitan mejores técnicas de comunicación para proveer una mayor velocidad de transmisión, así como una baja latencia y baja pérdida de paquetes, por esta razón el Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronic Engineers, IEEE) presentó en el año 1997 el estándar 802.11 para Redes Inalámbricas de Área Local.

Una red inalámbrica de área local (WLAN) permite a los usuarios la conexión de sus computadoras portátiles, asistentes digitales personales (PDAs) y equipos de escritorio a la infraestructura de red que existe en sus ambientes de trabajo u hogar sin la necesidad de la utilización de cables, esto permite la extensión de la Red de área local (Local Area Network, LAN) minimizando la colocación de cables y consigo innumerables ventajas [1].

Las WLANs generalmente son usadas como una tecnología de acceso a la red cableada y puede ser fácilmente implementadas para permitir el acceso a sitios remotos donde el cableado puede ser difícil o costoso, además son consideradas como una tecnología de ultima milla por poder prestar servicio inalámbrico a sus usuarios si estas se asocian a un proveedor de servicios de internet (ISP). Una de las ventajas más atractivas de la WLAN es su movilidad y utilización de servicios mientras el usuario se mueve dentro del rango de la red. Esta comodidad junto a la disminución de costos en hardware inalámbrico está haciendo que las WLAN sean más populares hoy en día.

## **2.2 WI-FI**

La tecnología Wi-Fi se basa en los estándares 802.11 del Instituto de Ingenieros Eléctricos y Electrónicos. El IEEE desarrolla y publica normas, pero no prueba los equipos para que cumplan con dichas normas, por esta razón en 1999 surgió Wi-Fi Alliance para llenar este vacío y hacer cumplir las normas de interoperabilidad y compatibilidad, además de promover el uso de Redes Inalámbricas Área Local [2].

Las redes Wi-Fi pueden usarse para conectar dispositivos electrónicos entre sí, al internet y a las redes cableadas que usan la tecnología Ethernet, además pueden proporcionar un desempeño similar a las redes cableadas para comunicar dispositivos.

### **2.2.1 Frecuencia de operación**

Los productos Wi-Fi funcionan en la banda de 2.4GHz o 5GHz. Estas bandas son asignadas como de libre uso, es decir, no se necesita la licencia del ente rector del espectro radioeléctrico para la utilización de estas bandas. Debido a que la banda utilizada por Wi-Fi es de licencia libre, hace que los fabricantes de la alianza Wi-Fi deban garantizar que sus productos pasan los estándares de interoperabilidad establecidos por la certificación Wi-Fi. Uno de los aspectos más importantes por el cual la alianza Wi-Fi debe garantizar la comunicación entre los dispositivos, es que la banda de 2.4 GHz, es también utilizada por hornos microondas, teléfonos celulares, puertas de garaje, juguetes a control remoto, etc.

### **2.2.2 Relación entre IEEE 802.11 y Wi-Fi.**

En 1999 el IEEE ratificó el estándar 802.11 para redes inalámbricas de Área Local (WLAN). La primera versión se presentó con una tasa de transmisión de 1 a 2

Mbps, además definió las reglas fundamentales para la señalización y la prestación de servicios inalámbricos. La alianza Wi-Fi anteriormente llamada WECA (Wireless Ethernet Compatibility Alliance) tiene como misión certificar la interoperabilidad y la compatibilidad entre los equipos que se basan en el estándar 802.11, además de promover este estándar.

Debido a la estrecha relación y por razones de mercadeo, el término Wi-Fi se utiliza a menudo como sinónimo del estándar 802.11, sin embargo, una red Wi-Fi es en realidad una red basada en el estándar 802.11.

### **2.2.3 Estándar IEEE 802.11.**

Este estándar define el protocolo y la interconexión de datos compatibles con equipos de comunicaciones que existen en una Red de Área Local, en donde se comunican a través de radiofrecuencias o de infrarrojo usando el protocolo de acceso múltiple por detección de portadora con prevención de colisiones (Carrier Sense Multiple Access With Collision Avoidance CSMA/CA) como un mecanismo de acceso al medio. Este protocolo incluye la autenticación, asociación, servicio de re-asociación, un procedimiento opcional de encriptación y desencriptación, gestión de energía que permite reducir el consumo de energía en las estaciones móviles, función de coordinación de punto (Point Coordination Function, PCF) para la transferencia de datos [3]. Además también ofrece a los organismos reguladores un medio de normalización para el acceso a una o más bandas de frecuencias para la comunicación de una red de área local.

Así mismo describe las especificaciones de la capa física (Physical Layer, PHY) y la capa de enlace de datos específicamente las subcapa control de acceso al medio (Medium Access Control Layer, MAC) y control de enlace lógico (Logical Link Control, LLC) del modelo de referencia de interconexión de sistemas abiertos

(Open Systems Interconnection, OSI), sin embargo, las capas superiores no se describen, por lo cual, no se especifica ningún requisito que deba existir en ellas.

En general este estándar:

- Permite el funcionamiento de un dispositivo de conformidad con el estándar 802.11 del IEEE dentro de una red de área local inalámbrica (WLAN) que puede coexistir con la superposición de múltiples WLANs.
- Define varias técnicas de señalización en la capa física (PHY) y funciones de interfaz, que son controladas por la MAC.
- Define las funciones y servicios requeridos por los dispositivos del IEEE 802.11 para operar redes ad hoc o infraestructura.
- Describe los requerimientos y procedimientos para proporcionar confidencialidad en los datos que el usuario trasmite a través del medio inalámbrico (Wireless Medium, WM).
- Define los procedimientos MAC para el soporte a la red de área local (LAN), así como el requerimiento de calidad de servicio (Quality of Service, QoS) en aplicaciones, incluyendo el transporte de voz, audio y video.[3]

#### 2.2.3.1 Definiciones IEEE 802.11.

En este estándar se aplican varias definiciones y términos

- *Control de acceso* (Access control, AC): previene el uso de recursos no autorizados.
- *Estación* (Station, STA): dispositivo que cumple con el estándar 802.11 del IEEE en cuanto a la capa física y control de acceso al

medio; Esta dotado con una tarjeta que permite la interfaz con el medio inalámbrico.

- *Punto de acceso* (Access Point, AP): cualquier estación que tenga la funcionalidad de proveer acceso a los servicios de distribución a través del medio inalámbrico (Wireless Medium, WM) para estaciones asociadas.
- *Asociación* (Association): se utiliza para establecer la conexión entre un punto de acceso y una estación.
- *Sistema de distribución* (Distribution System, DS): usado para interconectar un conjunto de servicios básicos (Basic Service Sets, BSSs), e integrar redes de área local para crear un conjunto de servicios extendidos (Extended Service Set, ESS).
- *Conjunto de servicios extendidos* (Extended Service Set, ESS): conjunto de uno o más servicios básicos interconectados e integrados a redes de área local.
- *Conjunto de servicio básico independiente* (Independent Basic Service Set, IBSS): Un BSS que forma una red autónoma donde el acceso a un sistema de distribución no está disponible.

## **2.2.4 Modos de red**

### **2.2.4.1 Red Infraestructura**

Este modo consiste en un punto de acceso conectado a la red y a varias estaciones inalámbricas (Ver figura 1). Esta configuración está basada en la arquitectura celular, donde el sistema es dividido en un conjunto de servicio básico. Para extender el rango de cobertura, múltiples BSS son usadas donde los puntos de acceso son conectados a través de la red principal (backbone) llamada sistema de distribución (DS).

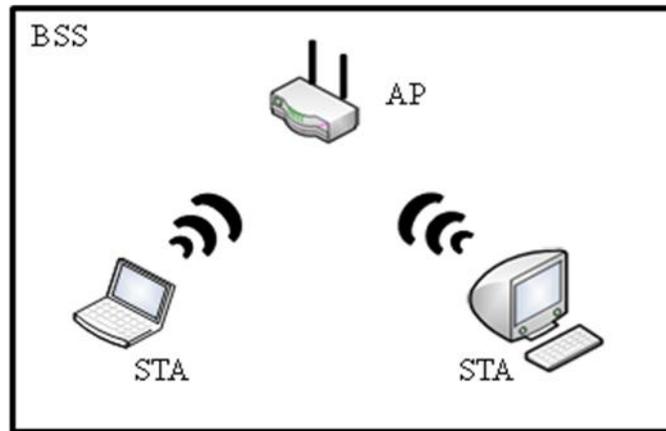


Figura 1. Red Infraestructura. Fuente: Autor

El estándar IEEE 802.11 adopta una dirección en particular. Cada celda es identificada por una dirección de 6 bytes (48 bits). Llamada identificador de conjunto de servicio básico (Basic Service Set Identifier, BSSID), la cual corresponde a la dirección MAC del Punto de Acceso de la celda.

Para evitar conexiones indeseadas, se puede configurar en el AP una información adicional llamada identificador de conjunto de servicio (Service Set Identity, SSID) que permite identificar el servicio de red, este es una cadena de caracteres de 32 bytes de un tamaño variable. El SSID es usado para garantizar la autenticación y la identificación entre el punto de acceso y una estación.

#### 2.2.4.2 Red Ad Hoc

Este modo representa un grupo de estaciones inalámbricas que se comunican directamente sin la necesidad de tener una conexión a un punto de acceso, como se observa en la figura 2. Permite crear una red rápidamente en donde no se requiera servicios, como los que proporcionaría una red infraestructura o cuando el acceso a la red fija no esté permitido o se hace difícil acceder a ella.

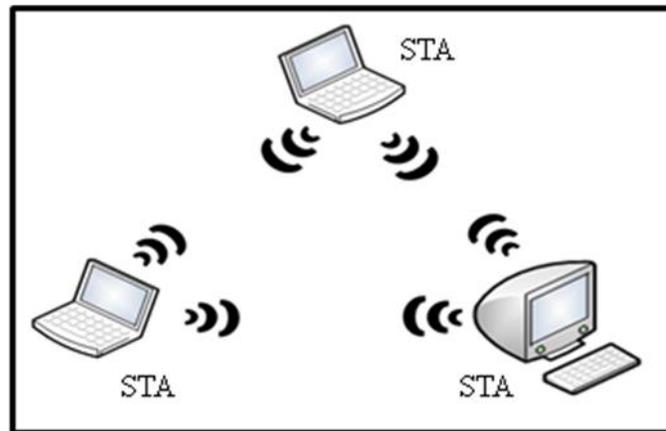


Figura 2. Red Ad Hoc, Fuente: Autor.

### 2.2.5 Arquitectura.

El estándar 802.11 contiene varios componentes que interactúan para proporcionar una red inalámbrica de área local, que favorece la movilidad de la estación de forma transparente en las capas superiores del modelo de referencia OSI.

Esta arquitectura está formada por una estación, un punto de acceso, un conjunto de servicio extendido, un conjunto de servicio básico, un conjunto de servicio básico independiente y un sistema de distribución [3].

### 2.2.6 Capa control de acceso al medio (Medium Access Control Layer)

Controla las transmisiones y previene las colisiones entre los paquetes mediante la utilización del protocolo de acceso múltiple por detección de portadora con prevención de colisiones (Carrier Sense Multiple Access with Collision Avoidance CSMA/CA) como mecanismo de acceso, esto permite a los dispositivos escuchar o detectar si en el canal se está transportando una señal; si se detecta la presencia en el canal de una portadora desde otro nodo, significa que en el canal se está transmitiendo a través de otro dispositivo, sino se detecta una señal portadora el dispositivo procede a enviar una notificación a través del medio, sobre su intención de utilizarlo, posteriormente transmite sus datos.

Cuando el dispositivo necesita transmitir y nota que el medio está ocupado, intentará enviar después de un corto período de tiempo llamado factor “backoff”, éste factor permite que el dispositivo espere un tiempo antes que pueda volver a escuchar el canal, para minimizar las colisiones entre paquetes. La probabilidad de que dos dispositivos posean el mismo factor es mínimo [4].

Hay un mecanismo de acceso opcional, el cual, es un procedimiento llamado RTS/CTS en el que un nodo que quiere transmitir, primero debe enviar al nodo receptor una solicitud de envío (Request To Send, RTS), la cual, contiene la información de la longitud del paquete, la dirección de la estación y el tiempo de duración en milisegundos requeridos para transmisión del paquete. Luego de que el nodo transmisor envía esta solicitud deberá obtener un paquete llamado “Listo para enviar” (Clear to Send, CTS) del nodo receptor. Después de este intercambio (handshake), el nodo transmisor procede a enviar el paquete. Cuando el nodo receptor recibe el paquete satisfactoriamente envía al nodo transmisor un paquete de acuse de recibo conocido como ACK (Acknowledgement). De esta manera se soluciona el problema de un nodo oculto, en el que dos dispositivos distantes no detectan las transmisiones y por la tanto envían paquetes al mismo tiempo pudiendo ocurrir colisiones.

#### 2.2.6.1 Trama MAC 802.11

La trama de la capa de control de acceso se compone de un encabezado MAC, cuerpo de trama y una secuencia de verificación de trama (Frame Check Sequence, FCS), estos campos son mostrados en la figura 3.

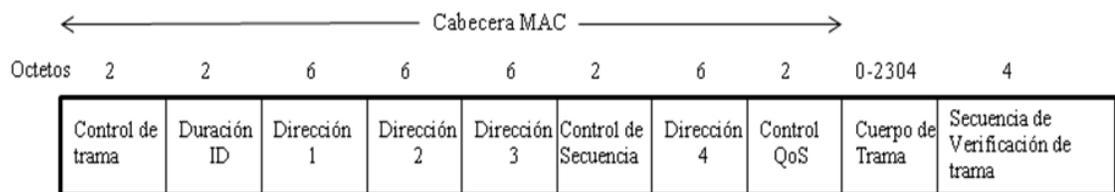


Figura 3. Formato de trama MAC 802.11. Fuente IEEE 802.11-2003: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Agosto 2005

Por otra parte la cabecera MAC está compuesta por campos, que permiten la identificación de la trama MAC.

- *Campo de control de trama:* contiene la información de control utilizado para definir el tipo de trama MAC 802.11, igualmente, provee información necesaria para los campos siguientes que permitan entender como procesar la trama MAC.
- *Campo de duración:* usado para todos los tipos de tramas, indica la duración restante para recibir la siguiente trama.
- *Campo de direcciones:* dependiendo del tipo de trama, el campo de direcciones puede ser una combinación de varias direcciones, las cuales definirán la ruta por la cual se enviarán los paquetes.
- *Campo de control de secuencia:* este campo tiene una longitud de 16 bits y consiste en dos subcampos: El primero es el número de secuencia y el segundo es el número de fragmentos, el primero es un campo de 12 bits, el cual, indica el número de secuencia de cada trama, el segundo, indica el número de cada trama enviada de una trama fragmentada.
- *Campo de trama de carga útil:* contiene los datos a ser enviados por la estación.

- *Campo de secuencia de verificación de trama:* campo de 32 bits que se utiliza para la verificación de redundancia cíclica (cyclic redundancy check, CRC) en todos los campos de la trama MAC.

El estándar define tres tipos de tramas: control, datos y de gestión, por consiguiente el campo de direcciones contendrá cuatro campos.

- *Identificador de conjunto de servicio básico (Basic set service identifier, BSSID):* este campo identifica de forma única cada conjunto de servicios básicos (BSS). Cuando la trama es desde una estación en el modo de red infraestructura, el BSSID es la dirección MAC del punto de acceso. En cambio cuando la trama es desde una red ad hoc, el BSSID es generada aleatoriamente. Este campo es de 48 bits del mismo formato que una dirección MAC del estándar 802 del IEEE.
- *Dirección de destino (Destination Address, DA):* indica la dirección MAC del destinatario final, el cual, recibirá la trama.
- *Dirección de origen (Source Address, SA):* indica la dirección MAC de la fuente origen que inicialmente creó y transmitió la trama.
- *Dirección de recepción (Receiver Address, RA):* identifica a la próxima estación a la cual le será enviado la trama.
- *Dirección de transmisor (Transmitter Address, TA):* indica la dirección MAC de la estación que transmitió la trama en el medio inalámbrico.

### **2.2.7 Capa física**

Se encarga de la transmisión de datos entre nodos. El IEEE define cinco técnicas de modulación para el estándar 802.11.

- Espectro expandido por secuencia directa (Direct Sequence Spread Spectrum, DSSS).

- Espectro expandido por salto de frecuencia (Frequency Hopping Spread Spectrum, FHSS).
- Infrarrojo (infrared, IR).
- Espectro expandido por secuencia directa de alta tasa (High Rate Direct Sequence Spread Spectrum HR/DSSS)
- Multicanalización por división de frecuencias ortogonales (Orthogonal Frequency Division Multiplexing, OFDM).

#### 2.2.7.1 Espectro expandido por secuencia directa (DSSS):

Esta técnica genera un patrón de 11 bits redundantes para cada bit de información que hay en la señal, esto permite que el receptor pueda reconstruir la información de la señal recibida aun cuando la transmisión se vea afectada por interferencias, en caso de haber una interferencia en el canal de transmisión, se reenvía varias veces la información codificada mediante la utilización de tres canales intercambiables. Usa la banda de frecuencias ISM con 14 canales de 22MHz de ancho de banda con una separación entre canal de 5MHz. Esta técnica se utiliza en los sistemas que tienen una alta velocidad de transmisión de datos.

#### 2.2.7.2 Espectro expandido por salto de frecuencia (FHSS):

Esta modulación consiste en dividir la banda de frecuencia en canales de salto con duración de 400ms cada uno permitiendo 1600 saltos por segundo, luego de haber transcurrido este tiempo se cambia la frecuencia de transmisión de manera pseudo-aleatoria a otra y se continua con la comunicación (Verfigura 4). Para la sincronización del transmisor y receptor se envían una secuencia de saltos durante la comunicación, esta secuencia de saltos está definida por la secuencia pseudo-aleatoria antes nombrada. Esta modulación ofrece 1 o 2Mbps de rendimiento. Utiliza la banda de frecuencia dedicada para la industria, ciencia y medicina (ISM 2.4 – 2.483GHz) con 79 canales de 1Mhz de ancho de banda cada uno.

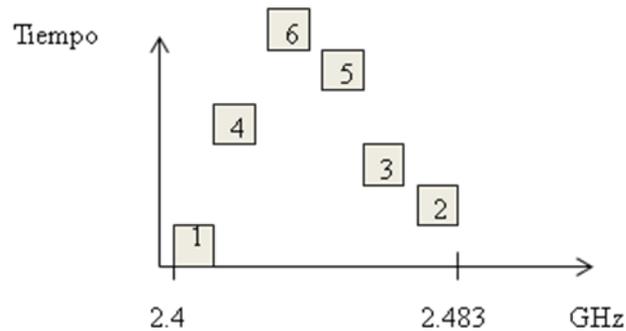


Figura 4. Técnica FHSS. Fuente: Labiod Houda, *Wi-Fi, Bluetooth, ZigBee and Wimax*, Springer, 2007

Algunas de las ventajas de FHSS son:

- Previene la pérdida total de la señal
- Proporciona una buena inmunidad contra interferencias y ruidos

#### 2.2.7.3 Infrarrojo (IR):

Es una técnica de modulación donde los datos son enviados mediante luces infrarrojas de longitudes de onda entre 850 a 950 nm requiriendo línea de vista.

#### 2.2.7.4 Multicanalización por división de frecuencias ortogonales (OFDM):

Esta técnica de modulación consiste en 52 subportadoras que sirven para enviar los símbolos paralelos unos con otros [5]. Utiliza la banda de frecuencias de 5Ghz con velocidades para la carga útil de 6, 9, 12, 18, 24, 36, 54 Mbps.

### 2.2.7.5 Trama PPDU.

La trama unidad de datos de protocolo PHY (The PHY Protocol Data Units, PPDU) contiene toda la información de sincronización necesaria para recibir los datos satisfactoriamente. Una trama PPDU consiste en un preámbulo PLCP, una cabecera PLCP, y un PSDU. (Ver figura 5)

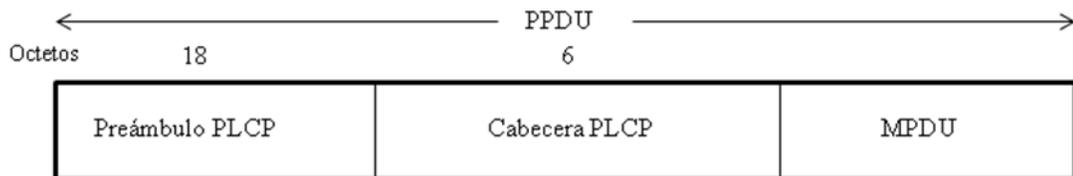


Figura 5. Trama PPDU 802.11. Fuente IEEE 802.11-2003: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Agosto 2005

En el estándar hay dos tipos diferentes de cabeceras y preámbulos:

- *Procedimiento amplio de convergencia de capa física (Long Physical Layer Convergence)*, el cual opera con la especificación DSSS de 1 y 2 Mbps.
- *Procedimiento corto de convergencia de capa física (Short Physical Layer Convergence)*, es opcional y destinado para aplicaciones donde el máximo rendimiento es deseado.
- *Unidades de servicio de datos de PLCP (The PLCP Service Data Units, PSDU)*: campo que contiene los datos que serán enviados, este campo puede ser modulado en cuatro diferentes velocidades 1, 2, 5.5 o 11 Mbps. Cuando se modula en 5.5 o 11Mbps se utiliza la modulación por cambio complementario (Complementary Code Keying, CCK).

## **2.2.8 Versiones IEEE 802.11.**

El estándar 802.11 fue creado por el IEEE en el año de 1997, este surgió con una velocidad de transmisión máxima de 2Mbps en la banda destinada para la industria, ciencia y medicina (industrial, scientific and medical, ISM) de 2.4Ghz, ahora bien, esta velocidad fue demasiado lenta para la mayoría de las aplicaciones. Por esta razón se han creado varias versiones a partir del estándar original 802.11 [6].

### 2.2.8.1 802.11a

Opera en la banda de 5Ghz con un rendimiento de 6 a 54 Mbps, utiliza OFDM, lo cual, permite una mayor resistencia a interferencias y mayor rendimiento. Por su alta velocidad permite la creación de WLANs para aplicaciones como videoconferencias.

### 2.2.8.2 802.11b

Versión más conocida a nivel mundial, comúnmente llamado Wi-Fi, sin embargo, Wi-Fi no se basa solo en una versión del 802.11 sino de todo el estándar 802.11, opera en la banda ISM de 2.4Ghz, utiliza DSSS para proporcionar mayor velocidad de transmisión, y posee un rendimiento hasta de 11Mbps.

### 2.2.8.3 802.11c

Provee procedimientos de operación de puentes.

### 2.2.8.4 802.11d

Señala nuevas extensiones para trabajar en otras bandas de frecuencia.

#### 2.2.8.5 802.11e

Versión para la calidad de servicios (QoS), en donde se definen las especificaciones de los mecanismos para permitir el uso de una infraestructura de múltiples proveedores, eliminando las normas propietarias.

#### 2.2.8.6 802.11f

Se ocupa de la normalización de los protocolos entre los puntos de acceso para permitir el uso de una infraestructura de múltiples proveedores

#### 2.2.8.7 802.11g

Permite velocidades hasta un máximo de 54Mbps en la banda de frecuencias de 2.4GHZ, utiliza OFDM y es compatible con el estándar 802.11b.

#### 2.2.8.8 802.11h

Gestiona el espectro para el estándar 802.11a para Europa, además de mecanismos para la selección dinámica de frecuencias y control de potencia en la transmisión.

#### 2.2.8.9 802.11i

Mejoras en MAC para una mayor calidad de seguridad en los datos.

### **2.2.9 Sensibilidad**

El estándar señala que un dispositivo WLAN basado en el IEEE 802.11, deberá poseer al menos una sensibilidad de -76 dBm y su tasa de error de trama

(Frame Error Rate, FER) debe ser de  $8 \cdot 10^{-2}$ . Las tramas a enviar deben tener 1024 octetos en el campo PSDU y la modulación deberá ser CCK con velocidad de 11Mbps [3].

### 2.2.10 Tasa de error del bit.

Como el estándar señala la tasa de error de trama (FER) y lo que se busca es la tasa de error del bit (BER), mediante las siguientes ecuaciones se convertirá el FER a BER [4].

La probabilidad de que una trama sea detectada correctamente es:

$$P_{\text{trama correcta}} = (P_c)_1 * (P_c)_2 * \dots * (P_c)_N = (P_c)^N \quad (1)$$

$P_c$  = probabilidad de bit correcto

$N$  = Número total de bits transmitidos

Probabilidad de que la trama sea incorrecta:

$$P_{\text{trama incorrecta}} = 1 - (P_c)^N \quad (2)$$

Despejando  $P_c$  de la ecuación 1.

$$P_c = (1 - P_{\text{trama incorrecta}})^{1/N} \quad (3)$$

Probabilidad de que un bit sea incorrecto:

$$P_i = 1 - P_c \quad (4)$$

$P_i$  = probabilidad de bit incorrecto

Sustituyendo la ecuación (2) en (4).

$$P_i = 1 - (1 - P_{\text{trama incorrecta}})^{1/N} \quad (5)$$

El estándar especifica que las tramas poseen 1024 bytes de carga útil, esto son 8192 bits a ser transmitidos por cada trama, el preámbulo y la cabecera son 192 bits, estos no se toman en consideración para el cálculo del FER [5], por lo tanto no se tomarán en el cálculo del BER.

Sustituyendo valores en la ecuación 5.

$$P_i = 1 - (1 - 8 \cdot 10^{-2})^{1/8192} = 1.0178 \cdot 10^{-5} \quad (6)$$

Al redondear la ecuación 6 se obtiene una tasa de error de  $1 \cdot 10^{-5}$ .

### **2.3 REDES INALÁMBRICAS DE ÁREA PERSONAL.**

El estándar 802.15 fue desarrollado por Instituto de Ingenieros Eléctricos y Electrónicos en el año de 1999 para desarrollar una red de corto alcance, baja potencia y bajo costo, que permitiera la creación de pequeñas redes con el fin de comunicar dispositivos periféricos como impresoras, celulares o asistentes digitales personales a una computadora, estas redes se definieron como redes inalámbricas de Área Personal (Wireless Personal Area Network, WPAN).

El grupo de trabajo del estándar 802.15 define en la actualidad tres clases de WPAN como se muestra en la figura 6. Estas se diferencian por la velocidad de transmisión, duración de la batería de alimentación y la calidad de servicios [7].

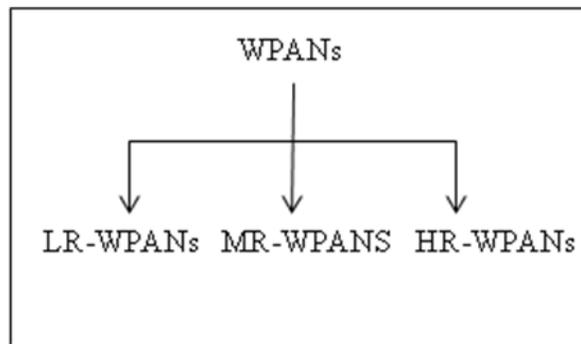


Figura 6. Clases de WPANs. Fuente: Autor

La alta tasa (High Rate, HR) es adecuada para aplicaciones multimedia que requieran alta calidad de servicios (QoS). Soporta velocidades entre 11 y 55Mbps.

Media tasa (Medium Rate, MR), puede manejar una variedad de tareas como por ejemplo la transferencia de archivos entre un teléfono celular y una computadora, además de poseer calidad de servicio adecuada para la comunicación de voz.

Baja tasa (Low Rate, LR), está destinada a ser utilizada en aplicaciones industriales, residenciales y médicas que no requieran de un alto consumo de energía, así como gran velocidad de transmisión. Las aplicaciones potenciales para utilizar esta clase son: sensores, juguetes, tarjetas inteligentes y automatización del hogar.

## **2.4 RELACIÓN IEEE 802.15.1 Y BLUETOOTH.**

En el año 2002 el IEEE publica el estándar 802.15.1, derivado de una red inalámbrica de área personal basada en Bluetooth [8], este estándar es un recurso adicional para los fabricantes de dispositivos, donde se especifican las normas que deben poseer estos equipos en cuanto a capa física y capa de enlace de datos.

## 2.5 BLUETOOTH

Es una tecnología inalámbrica de radiofrecuencia de baja potencia y de bajo costo para comunicar dispositivos personales, esta tecnología surgió en el año 1994 cuando la compañía de telecomunicaciones Ericsson ubicada en Lound, Suecia, realizó una investigación que le permitiera comunicar sus equipos celulares con sus accesorios; a medida que avanzaba la investigación se determinó que las aplicaciones que poseería esta tecnología sería ilimitada debido a su versatilidad y ventajas que ofrece la propagación en RF. En el año 1997 Ericsson decidió reunirse con otros fabricantes de equipos portátiles para incentivar su interés en esta tecnología, por tal motivo surge en febrero de 1998, un grupo de interés especial (SIG siglas en ingles) conformado por grandes fabricantes como Nokia, IBM, Toshiba e Intel en donde se proponen establecer una tecnología para conectividad sin hilos de corto alcance.

En Mayo de 1998 se anuncia la creación del consorcio Bluetooth, en donde, reúne a más de dos mil compañías cuyo propósito era crear un estándar, el cual, permitiera la interoperabilidad entre dispositivos de distintos fabricantes.

### 2.5.1 Banda de frecuencia y canales.

Bluetooth utiliza la banda de frecuencia ISM, la cual, está destinada para la industria, ciencia, medicina y cuyos dispositivos son de baja potencia, por esta razón Bluetooth opera en la banda de frecuencia de 2.4-2.4835GHz. Para poder transmitir en esta banda, la modulación usada deber Spread Spectrum [7].

Son 79 canales que utiliza esta tecnología, con un ancho de banda de 1 MHz por canal.

## 2.5.2 Pila de protocolos.

La especificación Bluetooth divide la pila de protocolo como se muestra en la figura 7, en tres grupos lógicos para que todas las aplicaciones sean capaces de operar entre sí.

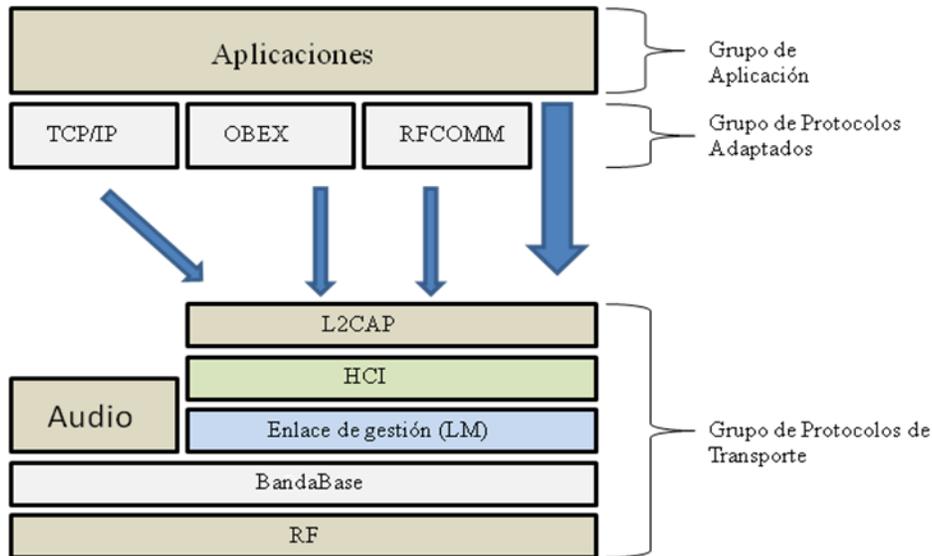


Figura 7. Pila de Protocolos Bluetooth. Fuente: Autor.

- *Grupo de protocolo de transporte:* permite a los dispositivos Bluetooth ser localizados por otros, además gestiona los enlaces físicos y lógicos con los protocolos de la capa superior. Este grupo no concuerda con la capa de transporte de modelo de referencia OSI, sino más bien correspondería a las capas de enlace de datos y física del modelo de referencia.

Los protocolos que están contenido en este grupo de transporte son: el radio (The Radio), Bandabase (Baseband), gestión de enlace (Link Manager, LM), adaptación y control enlace lógico (logical Link Control and Adaptation, L2CAP), interfaz controladora de maquina (Host Controller Interface, HCI). Estos protocolos

soportan transmisiones asincrónicas y sincrónicas, además de ser requeridos para la comunicación entre dispositivos.

- *Grupo de Protocolos adaptados:* permite que las aplicaciones existentes y nuevas operen a través de vínculos Bluetooth. Algunos de los protocolos que están en este grupo y que no son específicos de Bluetooth son: Protocolo de datagrama usuario (User Datagram Protocol, UDP), protocolo de control de transmisión (Transmission Control Protocol, TCP), protocolo de internet (Internet Protocol, IP), Protocolo Punto a punto (Point-to-Point Protocol, PPP) y Protocolo de intercambio de objetos (Objects Exchange Protocol).
- *Grupo de aplicaciones:* consiste en las aplicaciones actuales que utiliza los enlaces Bluetooth para permitir la conectividad entre dispositivos.

### **2.5.3 Capas**

#### 2.5.3.1 Capa de Radio

Es la responsable de transmitir y recibir paquetes de información sobre el canal físico, además es la encargada de transformar una cadena de datos para y desde el canal físico hacia la capa de banda base. Especifica el diseño de los transceptores Bluetooth.

#### 2.5.3.2 Capa de Bandabase

Define como los dispositivos escanean a los demás dispositivos Bluetooth que estén a su alrededor. Las reglas que deben seguir los dispositivos maestro (Master) y esclavo (Slave), se definen en esta capa como son: la secuencia de salto de frecuencia usada por los dispositivos, los tipos de paquete, procedimientos para

procesar los paquetes, las estrategias para la detección y corrección de errores, codificación de la señal y la transmisión y retransmisión de paquetes.

Esta capa soporta dos tipos de enlaces para aplicaciones multimedia: enlaces orientados a conexión sincrónica (Synchronous Connection-Oriented, SCO) y enlaces sin conexión asincrónicos (Asynchronous Connection-Less, ACL). Los SCO se caracterizan por ser de conmutación de circuito, poseer una sola ranura de tiempo para la asignación de paquetes, además de ser usados principalmente para la transmisión de voz cuyas velocidades oscilan alrededor de 64Kbps.

Un dispositivo que tiene establecido un enlace SCO tiene reservado unas determinadas ranuras de tiempo (time slots), además, los paquetes serán tratado como prioridad y por lo tanto serán atendidos antes que los paquetes ACL.

Los dispositivos con un enlace ACL puede enviar paquetes de longitud variable de 1, 3 o 5 ranuras de tiempo, sin embargo, no tiene ranuras de tiempo reservadas. Puede transmitir a 721 kbps en un sentido y 57,6 kbps en sentido contrario. Los enlaces ACL usan conmutación de paquetes.

#### 2.5.3.3 Capa de gestión de enlace (Link Manager, LM)

Se encarga de la implementación del protocolo de gestión de enlace (LMP), como el establecimiento de la conexión entre los dispositivos, incluso de la administración de las propiedades de la interfaz de aire entre los dispositivos.

## **2.5.4 Protocolos.**

### 2.5.4.1 Protocolo de gestión de enlace (LMP)

Define la asignación del ancho de banda necesario para los datos, igualmente, reserva ancho de banda para el tráfico de paquetes de audio, además, se encarga de la encriptación de datos y control de potencia de los dispositivos.

### 2.5.4.2 Protocolo de adaptación y control de enlace lógico (L2CAP)

Provee la interfaz entre los protocolos de las capas superiores con los protocolos de las capas inferiores. Soporta multicanalización de varios Protocolos de nivel superior, tal como el RFCOMM y SDP. También es responsable de la segmentación y re-ensamblaje de paquetes.

### 2.5.4.3 Protocolo RFCOMM

Proporciona la emulación de un puerto serie RS-232 mediante el protocolo L2CAP, permite hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth. Un identificador conocido como identificador de conexión de enlace de datos (data link connection identifier, DLCI) identifica el enlace entre el cliente y el servidor de aplicación. Los comandos AT (AT command) son aplicados directamente sobre el RFCOMM.

### 2.5.4.4 Protocolo de descubrimiento de Servicios (SDP)

Es de vital importancia para los dispositivos Bluetooth, ya que, permite el descubrimiento de diversos servicios proporcionados por los servidores de aplicaciones.

#### 2.5.4.5 Interface de control de maquina (HCI):

Define una interfaz estándar a las aplicaciones de nivel superior para poder acceder a las capas inferiores. Tiene propósito permitir la interoperabilidad entre los dispositivos y el uso de protocolos y aplicaciones existente en el nivel superior.

### 2.5.5 Potencia

La especificación Bluetooth señala 3 clases según el nivel de potencia de emisión:

- Clase 1. Diseñado para dispositivos de alto alcance tal como un punto de acceso Bluetooth, cuyo rango es de hasta 100 m, con un nivel de potencia de salida de 100mW (20 dBm)
- Clase 2. Para computadoras personales portátiles o de escritorio. Pueden alcanzar una distancia de 10 m con nivel de potencia de 2.5mW (4dBm).
- Clase 3. Para dispositivos de bajo consumo de potencia, con rango de 1 m cuya potencia de salida es de un 1mW (0dBm).

### 2.5.6 Modulación.

La tecnología Bluetooth dispone de dos modos de modulación. Uno obligatorio para la transferencia básica de datos, que permite minimizar la complejidad entre el transmisor y receptor, mediante la utilización de la modulación de frecuencias binarias y permite una velocidad de transmisión de 1Mbps.

El otro modo es opcional llamado transferencia de datos mejorada en donde se utiliza la modulación PSK y posee dos variables:  $\pi/4$ - DQPSK, la cual, permite una tasa de transferencia de 2Mbps y 8DPSK que permite 3Mbps.

Los canales de transmisión usan un programa de dúplex con división de tiempo (Time Division Duplex, TDD) con salto de frecuencia (Frequency Hopping), esto permite que el canal este dividido en intervalos de 625us al que le corresponde una frecuencia distinta de salto, con lo cual se logran 1600 saltos por segundo.

### **2.5.7 Perfiles**

Definen un conjunto de componentes de protocolo en SDP o RFCOMM, necesarios para la correcta instalación de aplicaciones de comunicaciones. Actualmente decenas de perfiles son definidos por el grupo de interés especial (Special Interest Group, SIG)

En general un perfil específica como utilizar la pila de protocolos para desarrollar una solución que permita que dos dispositivos de distintos fabricantes se puedan comunicar estableciendo opciones y parámetros, en donde se detalle cómo usar los procedimientos de los estándares que se encuentran implicados.

#### **2.5.7.1 Perfil de acceso genérico (Generis Access Profile, GAP):**

Describe el comportamiento de los dispositivos cuando están a la espera de una conexión, además de garantizar que los enlaces y canales se puedan establecer entre los nodos.

#### 2.5.7.2 Perfil de aplicación de descubrimiento de servicio (Service Discovery Application Profile, SDAP)

Define los protocolos y procedimientos usados por la aplicación de descubrimiento de servicio para encontrar servicios en otros dispositivos que soporten SDP.

#### 2.5.7.3 Perfil de puerto serie (Serial Port Profile):

Se basa en la especificación ETSI TS07.10, la cual, tiene por objeto definir un protocolo que puede utilizarse para emular un puerto serie [9]. Describe como configurar puerto serie y conectar dos dispositivos con tecnología Bluetooth.

#### 2.5.7.4 Perfil de telefonía inalámbrica (Cordless Telephony Profile, CTP)

Permite la comunicación inalámbrica entre un teléfono celular y una computadora.

#### 2.5.7.5 Perfil de Auricular (Headset, HS)

Este perfil es el más usado en la telefonía celular, ya que, permite que un auricular con tecnología Bluetooth se comunique con un ordenador u otro dispositivo compatible. Actúa como una interfaz de entrada y salida de audio.

#### 2.5.7.6 Perfil de redes de área personal (PANP):

Describe como dos dispositivos pueden formar una red ad hoc, así mismo, diferencia la red ad hoc del punto de acceso y del usuario de la red de área personal.

#### 2.5.7.7 Perfil de manos libres (Hands-Free profile, HFP)

Describe como un dispositivo utilizado como puerta de enlace puede realizar y recibir llamadas a través de un manos libres.

#### 2.5.7.8 Perfil genérico de intercambio de objetos (Generic Object Exchange Profile, GOEP)

Es muy utilizado para el intercambio de objetos entre dos teléfonos, se utiliza para transferir imágenes, tarjetas, etc. Señala la diferencia entre el servidor y el cliente, además este perfil depende del perfil de puerto serie.

#### 2.5.7.9 Perfil de Sincronización (Synchronization Profile SP)

Se utiliza para la sincronización de elementos del administrador de información personal, como directorios, agendas. También describe como las aplicaciones se puede sincronizar automáticamente entre dispositivos Bluetooth.

### 2.5.8 Estructura de paquete.

Los paquetes se dividen en dos formatos:

Paquete de tasa básica (Basic Rate packet) como se muestra en la figura 8, el cual se divide en 3 entidades: código de acceso, cabecera y carga útil (payload) y Paquete de tasa de datos mejorado (Enhanced Data Rate packet).

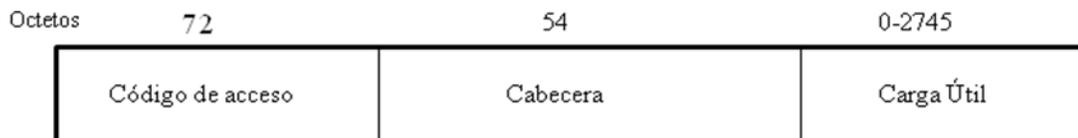


Figura 8. Estructura de paquete Bluetooth. Fuente IEEE 802.15.1: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Mayo 2005.

### 2.5.8.1 Código de acceso

Campo formado por 72 bits, en general contiene la dirección de la piconet y la sincronización del receptor. Este código es sumamente robusto y resistente a interferencias; todos los paquetes de una piconet usa el mismo código de acceso, si este código es distinto en el paquete, el receptor desechará el paquete.

### 2.5.8.2 Cabecera

Campo de 54 bits que contiene la información de la trama de datos, está compuesta de 6 campos de información.

Dentro de la cabecera hay 6 campos los cuales están protegidos por un código de corrección de error hacia adelante. Estos campos se pueden definir de la siguiente manera:

- *Campo de dirección de miembro activo (Active Member Address, AM\_ADDR)*, es la dirección de dispositivo activo en la piconet.
- *Campo de tipo (Type)*: define 16 tipos posibles de paquete, ayuda a decodificar la información en diferentes situaciones como SCO y ACL.
- *Campo de flujo (The Flow)*: es usado para el control de flujo de los paquetes ACL. Ayuda al esclavo indicarle al maestro que su buffer está lleno.
- *Campo de petición de retransmisión automática (Automatic Retransmisi3n Query, ARQ)*. Informa a la fuente sobre la recepci3n de un paquete correcto o incorrecto mediante la comprobaci3n de redundancia c3clica (CRC). Si la recepci3n de un paquete es correcta se enviará a la fuente un acuse de recibo (ACK), de lo contrario se enviara un NACK.

- *Campo de número de secuencia (Sequence Number, SEQN):* indica la numeración de los paquetes a transmitir, cada paquete posee un número que lo diferencia de los demás. Con esto el receptor ordena los paquetes según el número.
- *Campo verificación de error en cabecera (Header Error Check, HEC):* detecta los errores en la cabecera para su posterior corrección.

### 2.5.8.3 Carga útil (Payload)

Campo que puede contener hasta un máximo de 2745 bytes. Contiene los datos del cliente.

### Paquete de tasa de datos mejorado (Enhanced Data Rate packet)

Consiste de 6 entidades (Ver figura 9): código de acceso, cabecera, periodo de guarda, secuencia de sincronización, carga útil de tasa de datos mejorada.

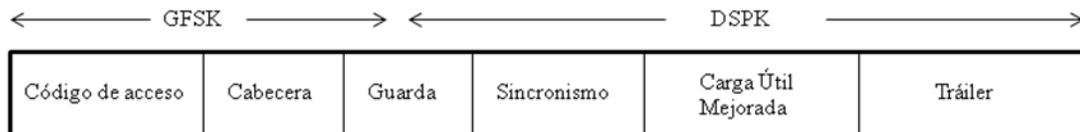


Figura 9. Estructura de paquete mejorado Bluetooth. Fuente: Autor

## 2.5.9 Redes

### 2.5.9.1 Piconet.

La especificación Bluetooth define a la piconet como una red ad hoc que reúne en un mismo radio de cobertura a varios dispositivos Bluetooth (Ver figura 10), estos dispositivos al unirse a la piconet compartirán el mismo canal de comunicaciones.

Esta red se crea a partir de dos dispositivos que se quieren comunicar, en donde uno de ellos tomara el rol de maestro y el otro de esclavo, la cantidad máxima de dispositivos que puede contener la piconet es de 8, es decir, un maestro y los demás serán esclavos. No se permiten dos maestros por piconet. Si hay más de 7 esclavos, los dispositivos que no lograron ser parte de la red deberán ser “estacionados” hasta que sea liberado un cupo en la piconet

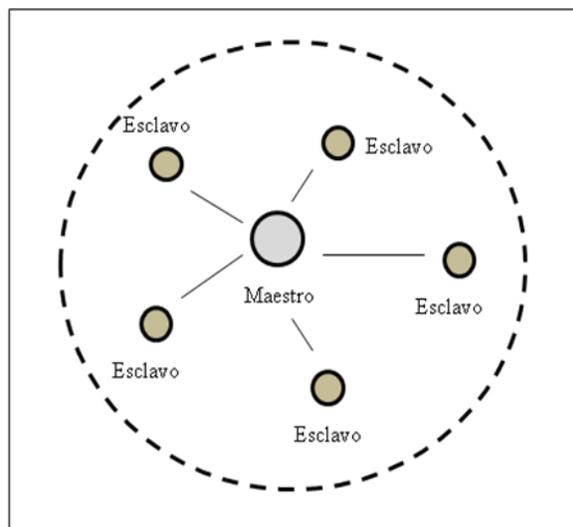


Figura 10. Piconet. Fuente: Autor

#### 2.5.9.2 Scatternet

Esta red está formada por la interconexión de dos o más piconet. Esta red se creó para solucionar los problemas de ancho de banda que se originaban en la piconet, ya que, cuando más usuarios se incorporen a la piconet, menor será el ancho de banda asignado para cada usuario debido a que todos los usuarios comparten el mismo canal, el cual, tiene ancho de banda de 1Mhz..

En la scatternet un dispositivo que es maestro en una piconet puede ser esclavo en otra piconet, al igual que un dispositivo esclavo en una piconet puede ser maestro en otra red. (Ver Figura 11)

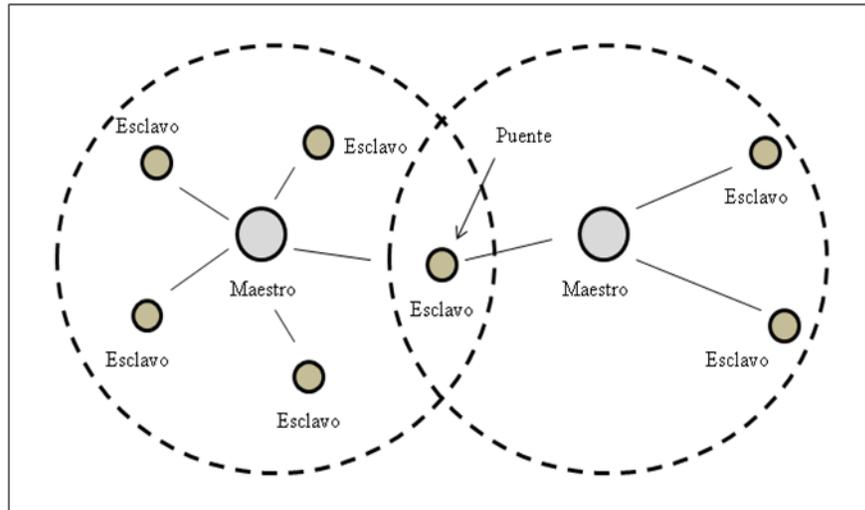


Figura 11. Scatternet. Fuente: Autor.

### 2.5.10 Emparejamiento

Cuando dos dispositivos establecen comunicación y no están asociados a una piconet, un proceso de negociación se producirá y uno de ellos se convertirá en maestro y el otro en esclavo. Las piconet son redes ad-hoc por lo tanto la comunicación se llevará a cabo entre un maestro y un esclavo y nunca entre dos esclavos, ya que, el maestro es el que selecciona la frecuencia, la secuencia de salto de frecuencia y el orden de selección de los dispositivos esclavos para la posterior comunicación.

Las unidades esclavas esperan en modo Stand by mientras no se les haga una solicitud de envío por parte del maestro, sin embargo, en todo momento reciben periódicamente señales de control por si el maestro les solicita “despertar”. Esto se realiza cuando el esclavo toma la señal de control recibida y hace una correlación

entre ella y el código de acceso derivado de su propia identidad, en caso de haber una correlación, es decir, si los bits coinciden, el esclavo se acciona corriendo un algoritmo de conexión, si no es así, actualiza el código que necesita para cambiar a modo de conexión con su respectiva frecuencia y fase.

El maestro transmite de forma continua el código de acceso a los dispositivos esclavos que se encuentran en stand by, de modo que estos no pierdan la información a cerca de la red.

Si el maestro desea comunicarse con mas esclavos, éstos enviarán un paquete que contendrá su identidad y reloj, esto permite al maestro conocer las unidades a las que le va a transmitir el paquete

#### **2.5.11 Tipos de dispositivos.**

Existen tres tipos de dispositivos Bluetooth, estos son: maestro, esclavo y puente. Los dispositivos puente interconectan dos o más piconet para comunicaciones multi-salto.

Los dispositivos puente solo pueden comunicarse a una piconet a la vez, debido a que deben cambiar de reloj para sincronizarse con la piconet a la que se van a unir, esto trae como consecuencia la generación de posibles cuellos de botellas. Un dispositivo puente puede ser un esclavo o maestro de una piconet.

Los dispositivos pueden estar en uno de los siguientes estados:

- Standby: modo de encendido pero no está conectado a ninguna piconet

- Inquiry: el dispositivo entra a este estado cuando no está unido a ninguna piconet y empieza a enviar solicitudes para encontrar otros dispositivos a los que se pudiera conectar.
- Page: en este estado el maestro envía mensajes en busca de dispositivos alrededor que se puedan unir a su piconet.
- Connected: cuando solicitud de conexión es exitosa entre el maestro y un esclavo, los dispositivos entran a este estado.
- Transmit: cuando existe una comunicación entre los dispositivos se dice que están en estado de transmitir. Cuando se termina el envío de datos los dispositivos vuelven al estado de conectado.
- Sniff: en este estado el dispositivo consume baja potencia mientras espera un número predeterminado de intervalos de tiempo para ir al estado Inquiry.
- Hold: también este estado es de bajo consumo de potencia donde el esclavo no tiene actividad por un determinado tiempo.
- Park: cuando el esclavo no está recibiendo o transmitiendo datos, el maestro puede decidir si ordena al esclavo entrar en este estado, esto significa que el esclavo dejara la dirección de la piconet, la cual, se le dará a otro esclavo.

### **2.5.12 Sensibilidad y tasa de error del Bit.**

El dispositivo receptor debe tener una sensibilidad de al menos de -70 dbm y la tasa de error de bit deber ser menor o igual a 0,1% [10].

## **2.6 ESTÁNDAR IEEE 802.15.4**

El estándar IEEE 802.15.4 define el protocolo y la interconexión compatible de datos entre dispositivos, utilizando una baja tasa de datos con baja potencia, baja complejidad y un radio de corto alcance dentro de una red inalámbrica de área

personal. Los objetivos principales de una WPAN de baja tasa (Low Rate, LR-WPAN) son la facilidad de instalación, transferencia confiable de datos, operación de corto alcance, costo extremadamente bajo y una razonable vida de la batería mientras mantiene un protocolo simple y flexible [11].

Por tanto define la capa física y la subcapa, control de acceso al medio de la capa de enlace de datos, del modelo de referencia OSI para la conectividad inalámbrica con baja velocidad de transmisión en aplicaciones fijas, portátiles o dispositivos móviles que no requieran un alto consumo de energía en el espacio de operación personal (Personal Operation Space, POS), éste espacio, es típicamente de 10 m que envuelve a un dispositivo fijo en todas las direcciones.

Las características más resaltantes son [9]:

- Velocidad de transmisión de 250 Kbps, 40kbps y 20kbps
- Existen dos modos de dirección: 16 bits y direccionamiento del IEEE de 64 bits
- Acceso al canal mediante CSMA/CA
- 16 canales en la banda ISM de 2.4 GHZ, 10 canales en la de 915 MHZ y un canal en la banda de 868MHZ
- Establecimiento de una red automática por parte de un coordinador
- Administración de energía para garantizar el consumo de baja potencia
- Indicación de calidad de enlace (Link quality indication, LQI)
- Detección de energía (Energy detection, ED)
- Asignación de ranuras de tiempo garantizadas (Guaranteed Time slots, GTS).

## **2.7 RELACIÓN IEEE 802.15.4 Y ZIGBEE**

ZigBee es una red inalámbrica de área personal dirigida hacia la automatización y aplicaciones de control remoto con una tasa baja de datos, bajo consumo de potencia y de bajo costo. Es diseñada para satisfacer las necesidades del mercado en concepto de bajo costo, además de proveer servicios como seguridad, descubrimiento de servicios y perfiles basados en la parte superior del estándar 802.15.4 del IEEE.

El estándar 802.15.4 solo cubre las capas inferiores del modelo de referencia OSI [12], por esta razón las capas superiores deben estandarizarse para lograr la interoperabilidad en una amplia gama de aplicaciones en el hogar, la industria y en la automatización de edificios. De ahí que, la alianza ZigBee y la IEEE han estado trabajando estrechamente para especificar toda la pila de protocolos.

El estándar 802.15.4 se centra en la especificación de las dos capas inferiores del protocolo, capa física y capa de enlace de datos, mientras la alianza ZigBee tiene como objetivo proporcionar la interoperabilidad de los dispositivos en las capas superiores de la pila de protocolos para asegurar a los clientes, la compra productos de diferentes fabricantes con la confianza de que estos trabajan sin mayor dificultad.

### **2.7.1 Dispositivos 802.15.4**

Este estándar define dos tipos de dispositivos que pueden participar en una red LR-WPAN; dispositivos de función total (Full Function Device, FFD) y dispositivos de función reducida (reduced function device, RFD).

### 2.7.1.1 Dispositivo de función Total

Este dispositivo puede operar en tres modos de servicio; como coordinador de una red área personal (PAN), coordinador o como dispositivo. Un dispositivo FFD puede hablar con un RFD u otros FFD, mientras que un dispositivo RFD solo puede hablar con un FFD.

### 2.7.1.2 Dispositivo de Función reducida.

Está diseñado para aplicaciones que son extremadamente simples, como un interruptor de luz, que no tienen la necesidad de enviar grandes cantidades de datos y solo pueden asociarse con un FFD, en consecuencia, un RFD puede ser implementado utilizando un mínimo de recursos y capacidad de memoria.

## **2.7.2 Dispositivos ZigBee.**

Basados en los dispositivos FFD y RFD del estándar 802.15.4 del IEEE, se definen tres tipos de dispositivos ZigBee.

### 2.7.2.1 Coordinador ZigBee (ZigBee Coordinator, ZC)

Este es el dispositivo más completo en la red, su función es controlar la red y las rutas que deben seguir los dispositivos para permitir conectarse entre ellos. Es capaz de almacenar información acerca de la red, inclusive de actuar como puente entre las demás redes. Debe existir uno en la red. Este dispositivo se considera de función total (FFD).

### 2.7.2.2 Router ZigBee ( ZigBee Route, ZRr)

Puede actuar como un enrutador intermedio, pasando los datos de otros dispositivos por él; esto permite la interconexión de dispositivos separados en la red. Considerado como FFD.

### 2.7.2.3 Dispositivo final (ZigBee End Device, ZED)

Se puede comunicar con un router o coordinador pero no puede enviar la información de otros dispositivos. Por ello requiere menor cantidad de memoria y por lo tanto es menos costoso.

## 2.7.3 Topología de redes

ZigBee soporta tres tipos de topología (Ver figura 12); estrella (Star Topology), punto a punto (peer-to-peer Topology) y grupo de árbol (Cluster tree).

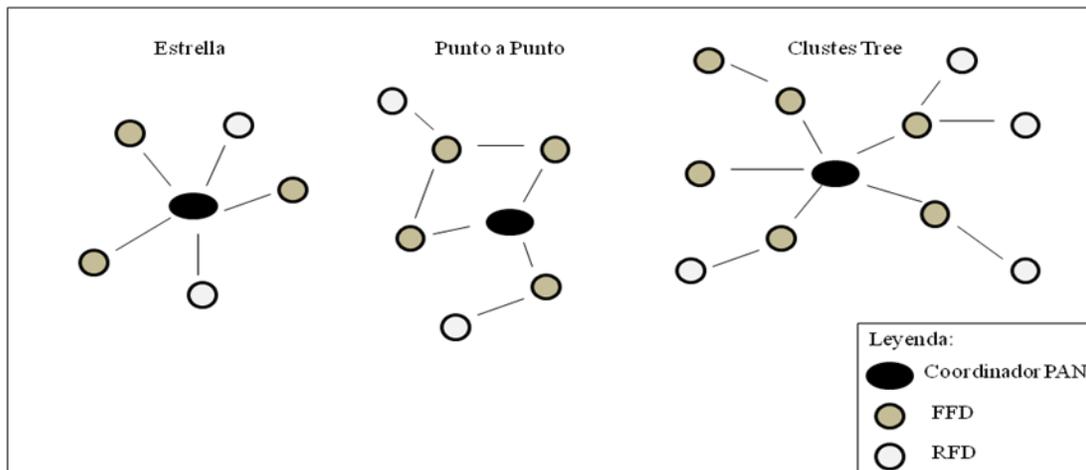


Figura 12. Topología de redes ZigBee. Fuente: Autor.

#### 2.7.3.1 Topología estrella.

La comunicación se establece entre dispositivos mediante un coordinador central, llamado coordinador PAN. El coordinador PAN puede ser conectado a un toma corriente, mientras que los otros dispositivos pueden funcionar con baterías. Las aplicaciones que se benefician de esta topología son: la automatización del hogar, periféricos de una computadora, juguetes, etc. Un coordinador PAN es un FFD.

Si el dispositivo FFD es activado por primera vez, podrá crear su propia red y convertirse en el coordinador de la red de área personal. Cada PAN elige un identificador, el cual, será diferente a los demás usados por otras redes permitiendo que cada red pueda operar de forma independiente.

#### 2.7.3.2 Topología punto a punto.

En esta topología también existe un coordinador PAN y a diferencia de la topología estrella, cualquier dispositivo puede comunicarse con otro, siempre y cuando estén en rango de comunicación. Permite múltiples saltos para enrutar los mensajes desde cualquier dispositivo a cualquier otro dentro de la red.

Una de las ventajas que posee esta topología es su uso en el monitoreo de controles industriales y redes de sensores inalámbricos.

#### 2.7.3.3 Topología Cluster Tree

Este es un caso especial de la topología punto a punto en el que la mayoría de los dispositivos son FFD. Cualquiera de los FFD puede actuar como coordinador y proveer servicio de sincronización con otros dispositivos y coordinadores. A pesar de que en la red haya varios coordinadores, solo uno será coordinador PAN.

El coordinador PAN forma el primer grupo mediante el establecimiento de sí mismo como la cabecera de grupo (Cluster Head, CLH), con un identificador de grupo (Cluster Identifier, CID), en donde, no es usado por las demás redes. El coordinador PAN envía un beacon, el cual, es un paquete que señala la disponibilidad y presencia de un dispositivo, éste beacon se envía broadcasting para todos los dispositivos que puedan estar alrededor. Un dispositivo al recibir un beacon puede pedir unirse a la red del CLH. Si el coordinador PAN permite la incorporación de este dispositivo, lo añadirá a su lista de vecinos. El nuevo dispositivo agregado a la red añadirá a su lista de vecinos el CLH del coordinador PAN y empezará a transmitir beacon periódicamente para que otros dispositivos puedan encontrar la red y así unirse a ella.

#### **2.7.4 Capa física (Physical Layer, PHY)**

La PHY provee una interfaz entre la subcapa MAC y el canal físico de radio mediante RF del hardware y firmware. La PHY conceptualmente incluye una entidad de gestión llamada PLME. Esta entidad provee la interfaz entre la capa de gestión de servicios, a través de la cual puede invocar funciones de administración de la capa física. El PLME también es responsable de mantener una base de datos en la administración de objetos perteneciente a la PHY. Esta base de datos se conoce como base de información PAN (PAN Information Base, PIB).

La capa física es responsable de las siguientes tareas:

- Recepción y transmisión de datos
- Selección de frecuencia del canal
- LQI para paquetes recibidos
- Detección de energía en el canal

#### 2.7.4.1 Detección de energía (Energy Detection, ED)

Se trata de una estimación de la potencia de la señal recibida dentro del ancho de banda en un canal IEEE 802.15.4. Esta medida se analiza con respecto a un valor umbral predeterminado y se destina para ser usada por la capa de red como parte de un algoritmo de selección del canal. La detección de energía no intenta identificar o descifrar las señales en el canal.

#### 2.7.4.2 Indicación de calidad de enlace (Link Quality Indication, LQI)

Esta medida es una caracterización de la fuerza y/o calidad del paquete recibido. El LQI puede ser implementado a través de la detección de energía

#### 2.7.4.3 CCA (Clear Channel Assessment).

Este análisis se realiza de acuerdo a uno de los siguientes métodos:

- Si la energía está por encima del umbral, el CCA reportará que el canal está ocupado.
- Solo detección de portadora; el CCA reportará que el medio está ocupado solo si la señal detectada, cumple con la modulación y ensanchamiento que especifica el estándar IEEE 802.15.4. Esta señal puede estar por encima o por debajo del umbral ED.
- Detección de portadora con energía por encima del umbral. CCA reportará que el medio está ocupado.

La capa física provee dos servicios: el servicio de datos PHY (PHY data service) y servicio de administración PHY (PHY management service).

#### 2.7.4.4 Servicios de datos PHY

Permite la transmisión y recepción de las unidades de datos de protocolo PHY (PPDU) a través del canal físico de radio.

#### 2.7.4.5 Servicio de administración PHY.

Controla la configuración de comunicaciones y las funciones del control de radio transmisor.

#### 2.7.4.6 Trama PPDU

La estructura de la trama PPDU se observa en la figura 13. Este formato de trama se presenta para que el campo de la izquierda sea escrito y transmitido o recibido según sea el caso en primer lugar. Los demás bits de cada octetos deberán ser transmitido o recibidos según el bit menos significativo (least significant bit, LSB).

Octetos	4	1	1	Variable	
	SHR	PHR		Carga útil PHY	
	Preámbulo	SFD	Longitud de Trama 7 bits	reservado 1 bit	PSDU

Figura 13. Trama PPDU 802.15.4. Fuente: IEEE 802.15.4: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Mayo 2003

Cada paquete PPDU consiste en los siguientes componentes básicos:

- Encabezado de sincronización (Synchronization header, SHR), permite a un dispositivo de recepción sincronizarse y bloquear una cadena de bits.

- Encabezado PHY, contiene la información de la longitud de la trama
- Longitud variable en carga útil, en el que lleva la trama de la subcapa MAC.

### **2.7.5 Modulación**

El estándar ofrece para las diferentes bandas de frecuencias de operación, 868, 915 y 2.4 GHz la técnica DSSS. La tasa de datos alta en 2.4 GHz es atribuida a una modulación de orden superior, O-QPSK, esto permite que los datos sean primero asignados en símbolos, para luego estos símbolos asignarlos a chips de 32 bits de largo y después modulados.

### **2.7.6 Capa de Control de Acceso al Medio.**

La subcapa MAC provee una interfaz entre la subcapa de convergencia de servicios específicos (the service-specific convergence sublayer. SSCS) y la PHY. Incluye una entidad de gestión llamada MLME. Esta entidad proporciona el servicio de interfaz a través de la cual se pueden gestionar funciones de la capa MAC. También es responsable del mantenimiento de la administración de objetos para la subcapa MAC.

Esta capa controla todo el acceso al canal físico y es responsable de las siguientes tareas:

- Sincronización de beacons en las redes.
- Proporciona un vínculo fiable entre dos entidades MAC.
- Apoya la asociación o disociación entre PAN
- Emplea el mecanismo CSMA/CA para el acceso al canal.
- Proporciona seguridad en los dispositivos.

### **2.7.7 Estructura de la trama MAC.**

El estándar 802.15.4 divide la trama en cuatro categorías; beacon, data, acknowledgement y control [6].

#### **2.7.7.1 Beacon.**

Permite a los dispositivos “despertar” cuando una solicitud de transmisión es recibida, en caso contrario seguirán en modo “dormido”. Los beacons son importantes en topologías de malla y de árbol para mantener a todos los nodos sincronizados, sin necesidad que los nodos deban escuchar por largos periodos reduciendo el consumo de energía.

#### **2.7.7.2 Datos**

La trama de datos provee una carga útil (payload) de hasta 104 bytes. La trama es numerada para asegurarse un seguimiento de todos los paquetes además una secuencia de control de trama (SCF) asegura que todos los paquetes sean recibidos sin errores.

Después de recibir el paquete, el receptor lleva a cabo una verificación de redundancia cíclica para verificar si el paquete no fue dañado durante la transmisión. Dependiendo de la aplicación y las necesidades de la red, si el CRC es correcto, el receptor automáticamente puede transmitir un paquete de acuse de recibo (ACK), permitiendo así saber al dispositivo transmisor que los datos fueron recibidos de forma aceptable. Por otra parte si el CRC es incorrecto, se descarta el paquete y el receptor transmite un NACK.

Cuando se configura la red para que espere un ACK, el dispositivo transmisor retransmitirá los paquetes cuyos ACK no se hayan recibido para asegurar la entrega de los paquete con éxito.

#### 2.7.7.3 Acknowledgement

Como se describió anteriormente, esta categoría proporciona retroalimentación (feedback) del receptor con el transmisor mediante un paquete acuse de recibo. Esto permite conocer si el paquete fue recibido exitosamente o no.

#### 2.7.7.4 Control.

Proporciona el mecanismo para control remoto y configuración de los nodos cliente. Un administrador de red ZigBee utiliza la dirección MAC para configurar los nodos clientes.

### **2.7.8 Capa de aplicación**

En la capa de aplicación existe dos fundamentos de comunicación, los cuales permiten que los dispositivos se comuniquen unos con otros, estos son: perfiles y Clusters,

#### 2.7.8.1 Perfiles

Al igual que en los perfiles de Bluetooth, los perfiles de ZigBee permiten a las aplicaciones de cada dispositivo, de distinto fabricante, crear una aplicación que permita la interoperabilidad de estos dispositivos.

### 2.7.8.2 Clusters

Los Clusters son únicos en un perfil dado, y se asocian a un flujo de datos de entrada y salida en un dispositivo. Estos son identificados por un identificador de Cluster (Cluster ID).

### 2.7.9 Comunicación

Para que los dispositivos se puedan comunicar, deben utilizar un protocolo de aplicación compartido conocido como perfil. Este perfil permite que los dispositivos a asociar deban tener una coincidencia entre los identificadores de cluster de entrada y salida.

El descubrimiento de dispositivos se hace mediante dos métodos distintos. El primero, Cuando se conoce la dirección de red se pedirá la dirección IEEE mediante un mensaje unicast, de lo contrario se enviará un mensaje broadcast, y la dirección IEEE formará parte de la respuesta. Los dispositivos finales (end device) responden con la dirección propia solicitada, mientras que los routers y coordinadores envían también las direcciones de todos los dispositivos asociados.

A causa de ello, permite conocer los dispositivos que conforman la red sus servicios ofrecidos a nodos externos de la misma. Los endpoints pueden informar acerca de estos servicios cuando el protocolo de descubrimiento dirige mensajes a ellos.

Una vez establecida la asociación, los dispositivos pueden iniciar la comunicación. Esta se puede realizar mediante el direccionamiento directo, el cual utiliza la dirección de radio y el número de endpoint, en cambio, el indirecto necesita toda la información relevante y la envía al coordinador de la red.

### 2.7.10 Sensibilidad

El IEEE 802.15.4 señala que un dispositivo que cumpla con una tasa de error de paquete (PER) de 1%, longitud PSDU de 20 bytes, deberá ser capaz de alcanzar una sensibilidad de -86dbm [8].

## 2.8 COMPARACIÓN DE TECNOLOGÍAS

Tabla 1. Comparación de tecnologías.

	Wi-Fi	Bluetooth	ZigBee
Frecuencia de operación	2.4 GHz	2.4 GHz	2.4 GHz 868/915 MHz
Tipo de Modulación	DSSS	FHSS	DSSS
Tamaño de pila	1Mb	1Mb	20kb
Tasa de transferencia	11-54Mbps	1-3Mbps	250kbps (2.4 GHz) 40 Kbps (915 MHz) 20 kbps (868 MHz)
Tipos de Datos	Digital	Digital, Audio	Digital (Texto)
Rango	100 m	10m-100m	10m-100m
Nº de Canales	11 - 14	79	16 (2.4GHz) 10 (915MHz) 1 (868 MHz)
Nº de dispositivos	32	8	255/65535
Requisito de alimentación	Media-Horas de Batería	Media- Días de Batería	Muy Baja- Años de Batería
Introducción al mercado	Alta	Media	Baja

Arquitecturas	Estrella	Estrella, punto a punto	Estrella, Árbol, punto a punto y malla
Aplicaciones	Edificio con internet	Computadoras y teléfonos	Control de Bajo Costo y Monitoreo
Precio	Costoso	Accesible	Bajo
Complejidad	Complejo	Complejo	Simple

Fuente: Autor.

## 2.11 MICROCONTROLADOR

El microcontrolador se define como un circuito integrado o chip que incluye en su interior las unidades funcionales fundamentales de un ordenador: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado programable. Aunque sus prestaciones son limitadas, además de dicha integración, su característica principal es su alto nivel de especialización.

## 2.12 COMUNICACIÓN SERIAL RS-232

Es un protocolo de comunicación serial asincrónica creado por la Asociación de Industriales Electronicos (Electronic Industries Alliance EIA) en el año de 1962 [14]. La EIA decidió crear este Standard para tratar de normalizar la comunicación que se lleva acabo entre un equipo de comunicación de datos (Data Communication Equipment DCE) y un equipo Terminal de datos (Data Terminal Equipment, DTE).

Este estándar especifica las descripciones mecánicas, eléctricas, funcionales y procedimientos para la interfaz entre el DTE y el DCE [13]. La norma no define la manera como serán enviados los datos, la codificación, algoritmo de compresión, algoritmo de detección de errores ni cantidad de bits para cada dato. Tampoco define la velocidad a la que debe transmitirse los datos, aunque limita el máximo en 20Kbps

(En la actualidad son comunes los dispositivos que superan esa velocidad en transmisión serial alcanzando 192Kbps).

Básicamente se pueden comunicar dos dispositivos RS232 utilizando tres señales: entrada de datos, salida de datos y tierra, pero en la norma se definen otras señales para asegurar la integridad de los datos transmitidos y reducir errores.

### **2.13 FAMILIA DSPIC**

Entre los microcontroladores más populares hoy en día se encuentran los fabricados por la empresa Microchip. Esta empresa tiene una amplia gama de productos que van desde los microcontroladores de 8 bits hasta los más recientes de 32 bits. Dentro de su oferta de productos se encuentran los microcontroladores de 16 bits series PIC24 y dsPIC.

La familia PIC24 está formada por dispositivos que conjugan altas prestaciones de velocidad y memoria con una amplia variedad de módulos de periféricos. La familia dsPIC agregan a lo anterior capacidades de procesadores digitales de señal (DSP por sus siglas en inglés) como por ejemplo acumuladores de 40 bits, operación de multiplicación de un solo ciclo, modos de redondeo y soporte de librerías especializadas como la transformada rápida de Fourier (FFT).

Ambas series de microcontroladores son adecuadas para el estudio que se propone, más sin embargo, se ha decidido usar la serie dsPIC por su capacidad para realizar cálculos matemáticos más eficientemente que la serie PIC24 puesto que cuenta con hardware especializado para la realización de dichas tareas.

## CAPÍTULO III

### DESARROLLO DEL PROYECTO.

#### 3.1 SELECCIÓN DE DISPOSITIVOS.

Se realizó un estudio comparativo de los manuales técnicos de los dispositivos que existen en el mercado, de tal modo de determinar los módulos de cada tecnología que cumplieran con una serie de características. Estas fueron: potencia, distancia, sensibilidad, costo y la utilización de comandos AT mediante la comunicación de un transmisor y receptor Asíncrono Universal (Universal Asynchronous Receiver Transmitter, UART).

Una vez definido las características que debían tener los dispositivos a utilizar en el proyecto, se realizó la búsqueda en la página proveedora de equipos a Corporación 80-60 obteniendo los módulos Wi-Fi, Bluetooth y ZigBee que se muestran a continuación en las Tablas 2, 3 y 4 respectivamente.

Tabla 2. Módulos Wi-Fi.

Módulo	Ganancia	Distancia(m)	Sensibilidad	Potencia	Precio(\$)
PRM112	2dbi	120	-99dbm a 280kbps	20db	99
DC-WMW- O1T-JT	2dbi	100	-82dbm a 11Mbps	16dbm	249
X09-019NSI	2.1dbi	100	-105 db a 9600 bps	17dBm	165
DC-WME- O1T-S	1.8dbi	100	-28 dBm a 11 Mbps	16dBm	130

WISMC01BI	2dbi	100	-84 dBm a 11 Mbps	15 dBm	70
MT800SWM	2dbi	90	-82 dBm a 11 Mbps	16 dBm	84.64
IW- SM2128MW- US	2dbi	100	-88 dBm a 11 Mbps	15dBm	56

Fuente: Autor, con los datos proporcionados por mouser.com

Al realizar la comparación entre los distintos módulos Wi-Fi se seleccionó el IW-SM2128MW-US (Ver anexo 1), por una mejor relación parámetro costo.

Tabla 3. Módulos Bluetooth.

Módulo	Ganancia	Distancia(m)	Sensibilidad	Potencia	Precio(\$)
BISMS02BI- NA	2dBi	250	-87dBm	6dBm	32.86
BTM402	2dBi	1000	-87dBm	16dBm	45.92
TRBLU23- 00300	2dBi	250	-86dBm	6dBm	42.31
RN-24S	2dBi	100	-80dBm	12dBm	69

Fuente: Autor

Al realizar la comparación se observó que el módulo BTM402 poseía mejores parámetros con respecto a los demás, no obstante, se consideró que estos eran excesivos para ser usados en el proyecto, por lo cual se optó por el módulo BISMS02BI-NA por tener la mejor relación parámetro costo.

Tabla 4. Módulos ZigBee

Módulo	Ganancia	Distancia(m)	Sensibilidad	Potencia	Precio(\$)
XB24-ACI-001	2.1dbi	100m	-92dbm	0dBm	18\$

Fuente: Autor

Para la fecha de búsqueda de los módulos ZigBee, solo se encontró el modulo XB24-ACI-001, por lo cual se seleccionó dicho módulo. (Ver anexo 3).

### 3.2 SELECCIÓN DE MICROCONTROLADOR.

Luego de haber seleccionado los módulos se procedió a la selección del microcontrolador. Para ello, se tuvo en cuenta los siguientes requerimientos:

- Capacidad de manejo de datos de al menos 12 bits
- Poseer al menos un UART
- Capacidad de memoria superior a los 1024 KB
- Programable en lenguaje C.
- Bajo costo
- Posibilidad de uso en futuros proyectos

El microcontrolador dsPIC33FJ64GP202 fue el seleccionado para implementación de la medición de la tasa de error del Bit. Este microcontrolador tiene 64KB de memoria flash, y 8 KB de memoria RAM, además, es capaz de ejecutar una instrucción en 25 ns. (Ver anexo 4).

Corporación 80-60 utiliza el software MPLAB para la programación de los microcontroladores. Este software permite la programación del dsPIC mediante el

lenguaje Assembler o C. Por esta razón se estudiaron varios tutoriales de lenguaje C para poder programar el dsPIC.

### **3.3 CARACTERISTICAS DE LAS ANTENAS**

Las antenas seleccionadas para los módulos poseen las siguientes características:

- Frecuencia 2.4 Ghz
- 2dbi
- 50  $\Omega$
- Omni-direccional
- 11mm
- VSWR<2.
- Conector U.FL

### **3.4 DESCRIPCIÓN DEL ALGORITMO DE MEDICIÓN DE BER**

Para la medición de la tasa de error del bit de cada tecnología, fue necesario el diseño de un algoritmo que permitiera tal tarea, este fue programado en el microcontrolador utilizando lenguaje c

A continuación se muestran los diagramas de flujo con su respectiva explicación.

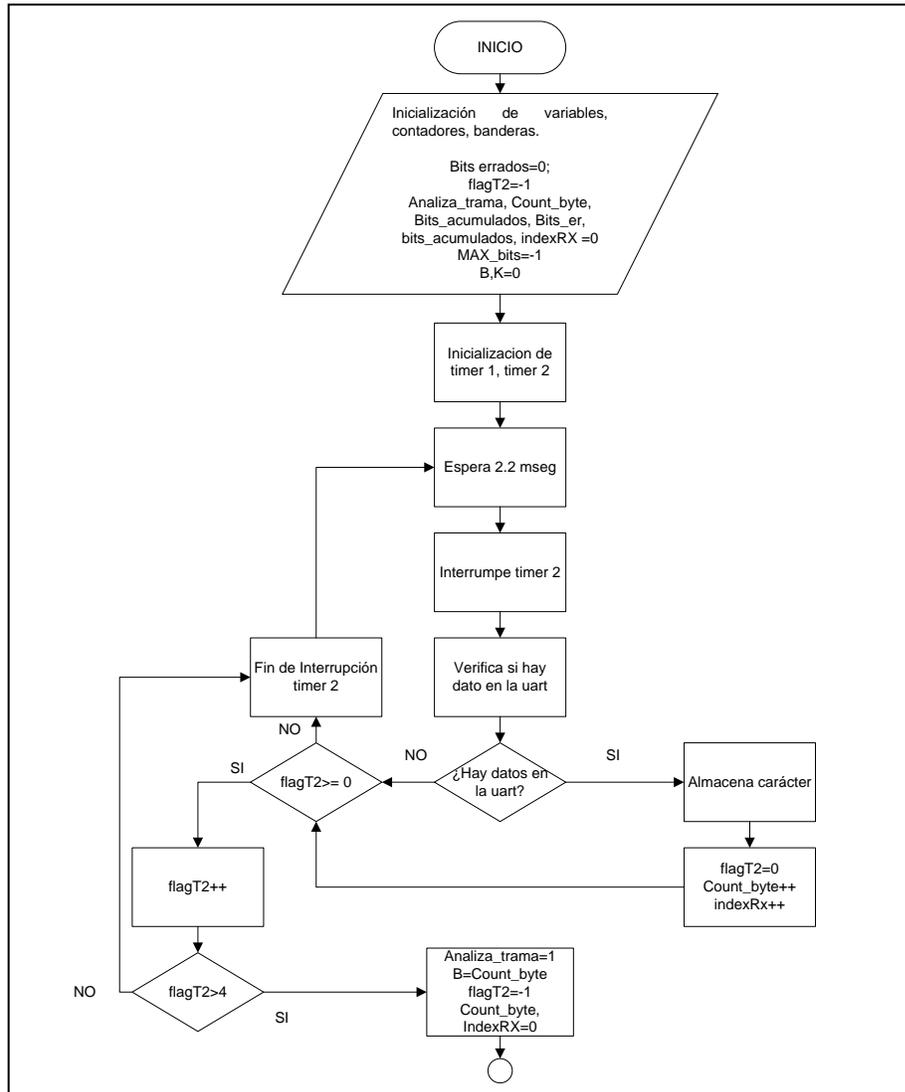


Figura 14. Diagrama de flujo función principal.

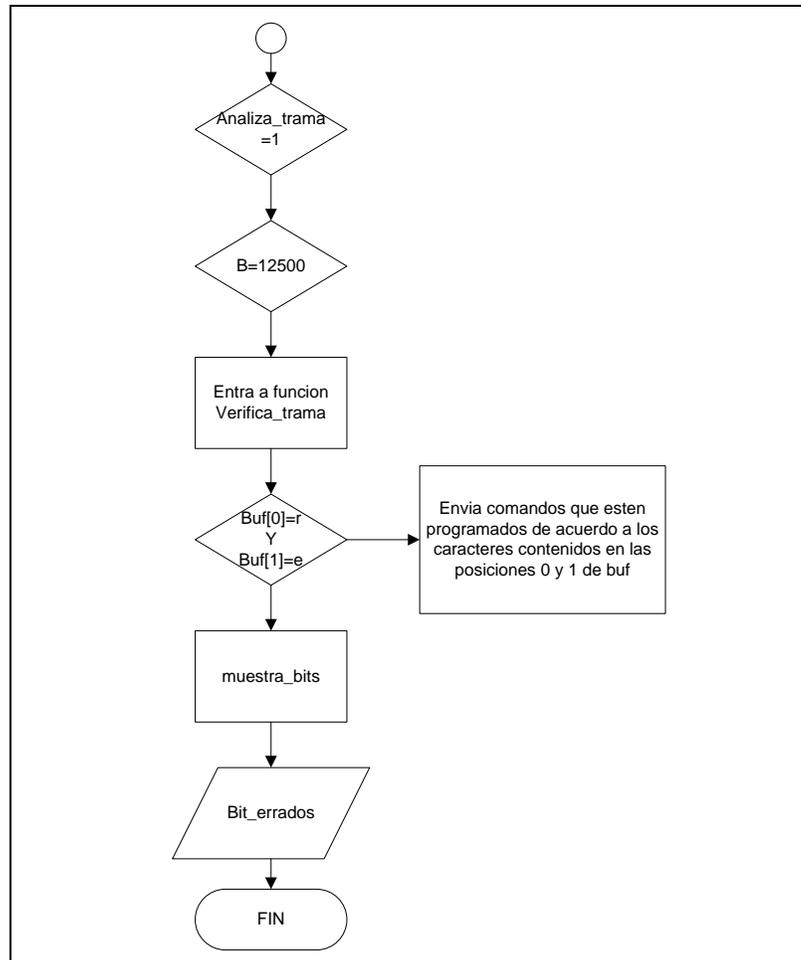


Figura 15. Diagrama de flujo Función Principal continuación.

1. En el programa del dsPIC del esclavo Wi-Fi y Bluetooth o end device para ZigBee, esta contenido la trama que el Maestro o coordinador envía. Esto con el fin de comparar los caracteres.
2. La recepción de los caracteres se programó a través de interrupción de timer, esto quiere decir que el timer interrumpe cada 2.2 ms, este tiempo es la duración de un carácter a velocidad de 4800 baudios. Al interrumpir el timer se verifica si en el UART está contenido un carácter, de ser afirmativo, el carácter se almacena en el vector “buf” y se termina la interrupción.

3. Luego de haber pasado 4 tiempos de carácter sin recibir dato en el UART, los caracteres guardados en el vector “buf” se envían a la función “verifica trama”. (Ver figura 16).

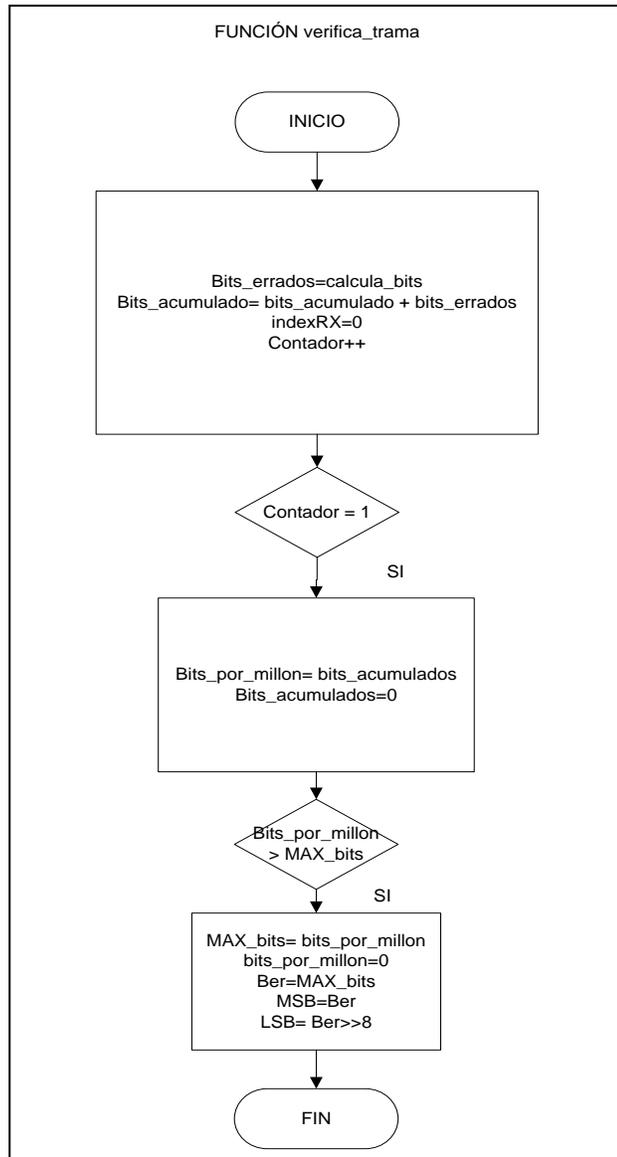


Figura 16. Diagrama de flujo Función verifica trama.

4. Una vez obtenido la trama almacenada en buf, se realiza la comparación con los caracteres guardados en el programa esclavo, esto permite determinar si hubo cambios o no con los caracteres enviados por el maestro.

5. Estos caracteres se comparan mediante la utilización de un XOR, el cual permite determinar a nivel de bits, los bits errados entre los dos caracteres, este proceso se lleva a cabo en la función “Calcula\_bits” mostrada a continuación.

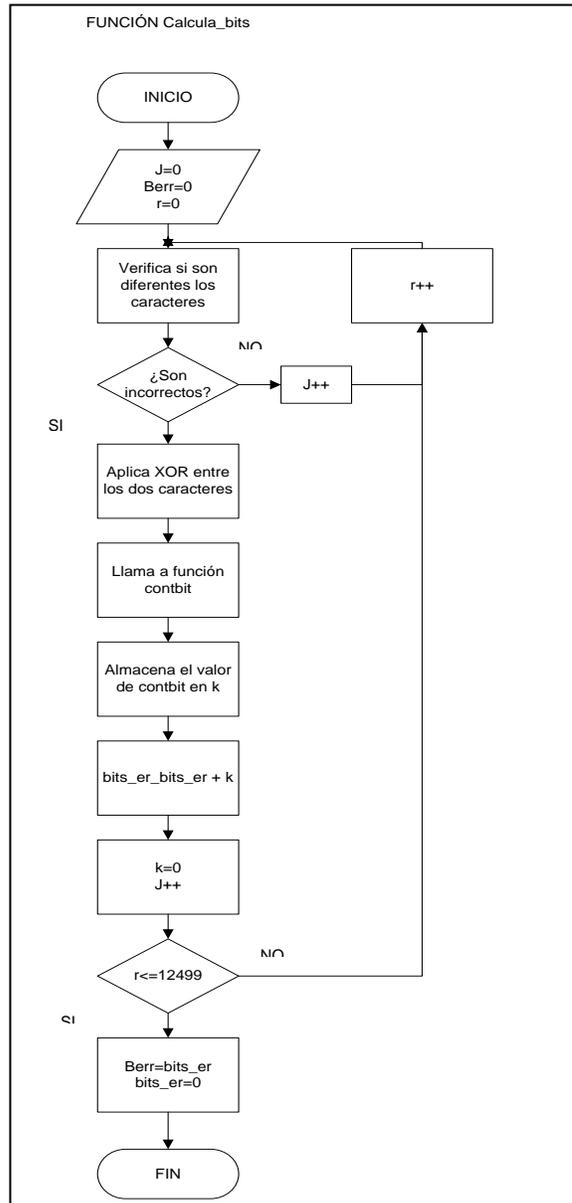


Figura 17. Diagrama de flujo Función calcula bits.

6. Luego de haber realizado el XOR, este valor se almacena en la variable “x” y se envía a la función “contbit” como se muestra en la figura 18, esta función

compara los 8 bits del carácter con un bits puesto en 1, si al comparar se determina que el bits del carácter es 1, este se cuenta mediante “contador”, una vez comparado los 8 bits se procede a salir de la función.

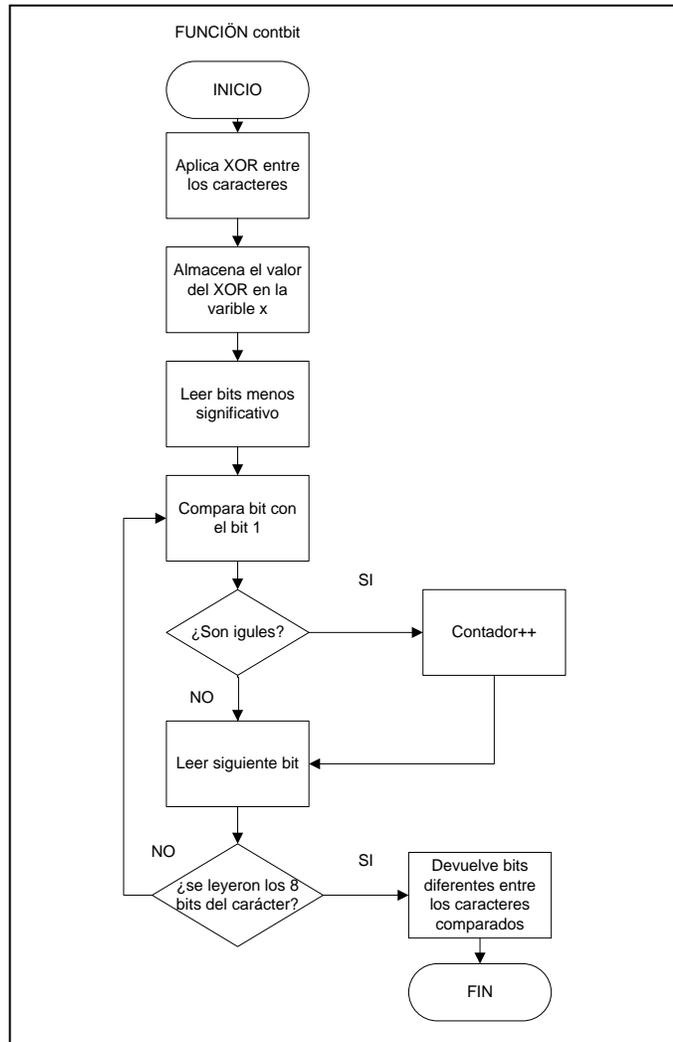


Figura 18. Diagrama de flujo Función contbit

7. Todos los caracteres siguen el mismo procedimiento mencionado en los pasos 4,5 y 7.
8. Una vez contados los bits errados entre los caracteres, se devuelve a la función “verifica\_trama”, estos bits se almacenan en la variable

“bits\_errados”, y si se reciben más caracteres, los bits errados, hallados anteriormente se guardan en “bits\_acumulados”.

9. Después que se haya transmitido todos los caracteres y verificado si coinciden o no, el BER se almacena en “bits\_por\_millon”
10. EL acumulador “bits\_por\_millon”, se compara con MAX\_bits y si es mayor, ahora MAX\_bits almacenara el dato del acumulador, esto se realiza con el fin de buscar la tasa de error máxima que pudiera haber en las demás transmisiones de la trama.
11. Al obtener los 12500 bytes en el esclavo, se envía al maestro o coordinador una palabra de “CORRECTO”, la cual, indica que se vuelvan a enviar los caracteres dentro de 40 seg para luego repetir el ciclo.
12. Realizado el cálculo de los 12500, se procede a enviar a través del programa Docklight de la computadora la palabra “resultado”, la cual permitirá obtener en pantalla los bits errados.

Es importante destacar que este algoritmo se diseñó para que pudiera comprobar 12500 bytes (100000 bits) enviados por el maestro o coordinador en una sola ventana de tiempo, a fin de hacer una similitud entre la tasa de error teórica y práctica. Pero el dsPIC no puede almacenar dicha cantidad de bytes, sin embargo, el buffer del módulo si lo puede hacer, por esta razón se hace la comparación en grupos de 6 tramas (2500 bytes).

En la siguiente tabla se muestra la función de cada variable (Ver tabla 5).

Tabla 5. Función de las variables.

Variable	Función
Bits_errados	Acumula el total de bits errados en una trama
flagT2	Contador que indica las interrupciones sin recibir carácter.
analiza_trama	bandera
count_byte	Cuenta los bits recibidos en la UART
bits_acumulados	Acumula todos los bits errados existentes en los 12500 bytes.
bits_er	Acumula los bits errados de un trama
indexRX	Puntero del vector “buf”
buf	Vector que almacena los caracteres recibidos
Max_bits	Acumula la tasa de error máxima.
B	Contador de caracteres
k	Acumula los bits errados entre dos caracteres
bits_por_millon	Acumula los bits contenidos en la variable “bits_acumulado”
MSB	8 bits más significativos de la variable “Ber”
LSB	8 bits menos significativos de la variable “Ber”
BERR	Acumula los bits de la variable “bits_er”,

	para ser enviada al finalizar la función “calcula_bits”
J	contador
Contador2500	Cuenta los 6 grupos de tramas

Fuente: Autor.

El código de programación se muestra en el anexo 7.

### **3.5 DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS.**

Una vez configurado el dsPIC, se procedió a estudiar las especificaciones eléctricas y de comunicaciones (Ver anexo 5), para la puesta en funcionamiento de los módulos. El montaje de los módulos se realizó en dos Protoboards (Ver anexo 6), y se comprobó la correcta comunicación del módulo con el dsPIC, así como la comunicación hacia la computadora.

Luego de haber obtenido satisfactoriamente una comunicación entre el módulo y el computador, se procedió a estudiar los manuales de comandos AT de cada dispositivo para su posterior implementación en el código de programación. (Ver anexo 7).

Una vez ejecutado los comandos de configuración se procedió a verificar la comunicación entre los dos módulos, obteniendo una respuesta satisfactoria. Además se comprobó el correcto funcionamiento del algoritmo realizado para la medición del BER.

Seguidamente, después de verificar, el correcto funcionamiento del módulo, la comunicación entre los módulos y la implementación del algoritmo en los protoboard, se realizó un diagrama de circuito impreso de las conexiones en el

programa Altium Designer (Ver anexo 8). Posteriormente, este circuito se transfirió a una baquelita mediante una serie de instrucciones recomendadas por los fabricantes de circuitos electrónicos.

Para la construcción del circuito en la baquelita, se utilizaron componentes superficiales para una mejor conexión y acabado; todos los componentes electrónicos estaban en el inventario Corporación 80-60, por lo que fue fácil la adquisición de estos. Los componentes superficiales para su colocación en la baquelita se requirió de soldadura superficial, por ello, el personal de Corporación, dio una inducción de soldadura superficial.

Una vez que se aprendió como soldar superficialmente, se procedió a soldar los componentes en la baquelita (Ver anexo 9), es importante destacar que esta baquelita se adaptó para que permitiera la colocación de los tres módulos en ella.

Después de elaborar el circuito en la baquelita, se conectaron los módulos a ella (Ver anexo 10); para la conexión del módulo Bluetooth se realizó una baquelita extra (Ver Anexo 2) para que calzara en los socket del circuito realizado anteriormente.

## CAPÍTULO IV

### REALICACIÓN DE PRUEBAS

Con la finalidad de comprobar el funcionamiento de los dispositivos de las tres tecnologías, tanto en un ambiente ideal como en subestaciones eléctricas de distintos niveles de tensión (4.8 y 12.4 KV), denominadas subestaciones de media tensión se realizaron pruebas en distintas ubicaciones de las subestaciones incluyendo casos donde el transmisor se encontrara en las adyacencias de un transformador, con la finalidad de verificar el efecto de estos sobre la comunicación. A continuación se muestra los resultados de cada prueba.

#### 4.1 PRUEBA EN CORPORACIÓN 80-60

Antes de realizar las pruebas en la sede de corporación 80-60, se hizo un escaneo de los diversos puntos de acceso de diferentes empresas que existen en el área, a fin de conocer los canales utilizados por estas, para así poder configurar el canal de comunicación del módulo con el propósito de evitar interferencias co-canal. Una vez logrado esto, se procedió a realizar las pruebas (Ver anexo 12 y 13) con un tiempo de duración de 12 horas obteniendo los siguientes resultados: (Ver tabla 6)

Tabla 6. Resultados en sede de Corporación 80-60

Distancia(m)	Módulo	BER
5	Wi-Fi	0
5	Bluetooth	0
5	ZigBee	0

Fuente: autor.

Además para la verificación del algoritmo se cambió en la configuración del código, la posición de los caracteres en el maestro, a fin de obtener bits errados en la comparación de los caracteres entre la trama del Maestro y el esclavo. En esta prueba se obtuvieron los siguientes resultados (Ver tabla 7).

Tabla 7. Resultados con cambio de caracteres en sede de Corporación 80-60

Distancia(m)	Módulo	BER
5	Wi-Fi	$96 \times 10^{-5}$
5	Bluetooth	$96 \times 10^{-5}$
5	ZigBee	$96 \times 10^{-5}$

Fuente: autor.

AL comprobar el correcto funcionamiento del algoritmo, se llevaron los dispositivos a varias subestaciones eléctricas de media tensión, con el propósito de medir la tasa de bits errados, en esos ambientes industriales.

#### **4.2 PRUEBA EN SUBESTACIÓN DE MEDIA TENSIÓN A.**

En esta Subestación de 4.8 kV, el Maestro se colocó en la UTR y el Esclavo en la caja de control del transformador 1 como se muestra en el anexo 15, esto debido a que la celda 1 el tomacorriente no tenía tensión. La duración de estas pruebas fue de 30 minutos por tecnología. En la tabla 8 se muestran los resultados obtenidos. El inicio de esta prueba fue a las 10 AM.

A continuación se muestra la disposición de celdas y UTR, (Vista superior):

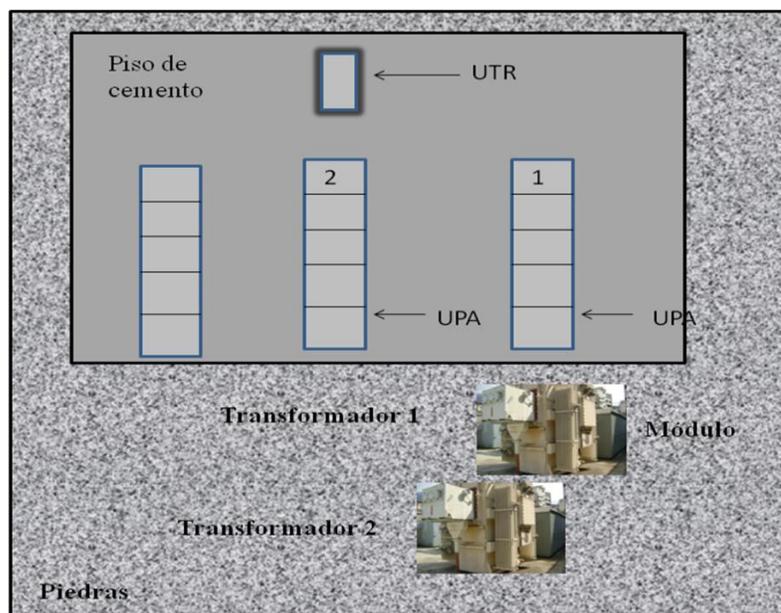


Figura 19. Estructura de Subestación de media tensión A. Fuente: Autor

Tabla 8. Resultados en subestación de media tensión A.

	Distancia	UTR	Celda	BLUETOOTH	Wi-Fi	ZigBee
PUERTA	16 m	Abierta	Abierta	BER= 0	Reinicio	BER = $8 \cdot 10^{-5}$
PUERTA	16 m	Cerrada	Cerrada	n/c	Reinicio	n/c
PUERTA	16 m	Cerrada	Abierta	n/c	Reinicio	n/c
PUERTA	16 m	Abierta	Cerrada	n/c	Reinicio	n/c

Fuente: Autor

n/c significó que no se logro la conexión entre los dispositivos.

Cuando se señala un BER de  $8 \cdot 10^{-5}$  en ZigBee, se está refiriendo a que en el maestro llega ruido y por lo tanto no hay una comunicación exitosa.

Reinicio significa que los comandos configurados en el programa del dsPIC se envían una y otra vez al módulo. Este problema es debido a un consumo excesivo potencia del módulo Wi-Fi, el cual, hace que el dsPIC se reinicie. Este problema fue solucionado para la realización de futuras pruebas.

#### 4.3 PRUEBA EN SUBESTACIÓN DE MEDIA TENSIÓN B.

En esta Subestación de 12.4 kV (Ver anexo 16). El Maestro se colocó adentro del la caseta y Esclavo en la celda 1 donde se encontraba la UPA (Ver anexo 17), con un tiempo de duración de 30 minutos para cada tecnología. En la tabla 9 se dan a conocer los resultados obtenidos. En la figura 20 se observa la disposición de celdas y UTR (Vista superior). El inicio de las pruebas fue a las 11 AM.

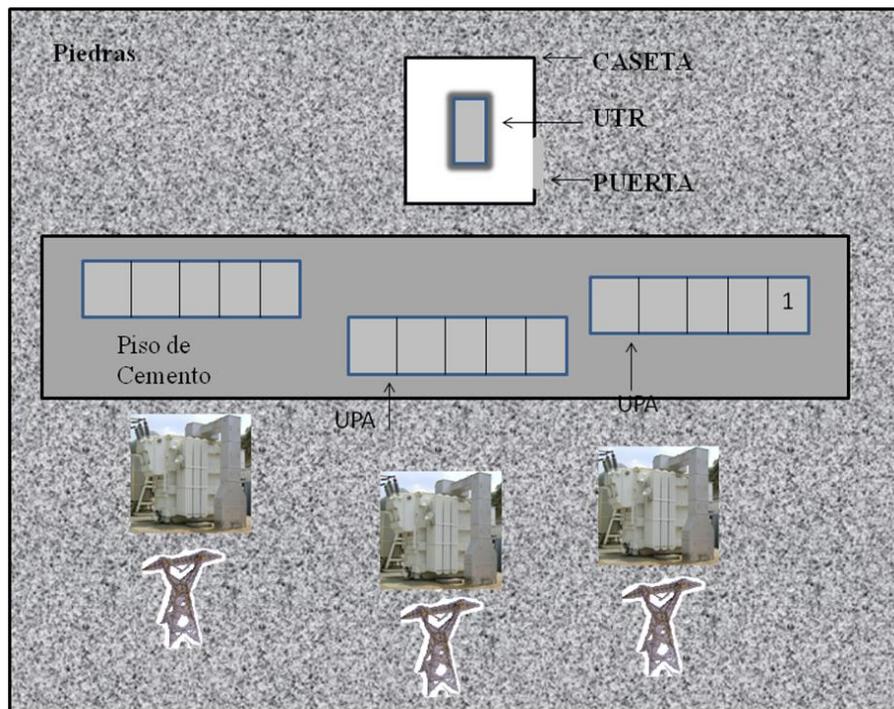


Figura 20. Estructura de Subestación de media tensión B. Fuente: Autor

Tabla 9. Resultados en subestación de media tensión B.

	Distancia	Caseta	Celda	BLUETOOTH	Wi-Fi	ZigBee
PUERTA	7 m	Abierta	Abierta	BER= 0	BER= 0	BER 8*10-5
PUERTA	7 m	Cerrada	Cerrada	n/c	n/c	n/c
PUERTA	7 m	Cerrada	Abierta	BER= 0	BER= 0	BER 8*10-5
PUERTA	7 m	Abierta	Cerrada	n/c	n/c	BER 8*10-5

Fuente: Autor

#### 4.4 PRUEBA EN SUBESTACIÓN DE MEDIA TENSIÓN C.

En esta subestación de 4.8 KV (Ver anexo 18) se realizaron dos pruebas. En la primera, el Maestro se colocó en una UTR y Esclavo en la celda 1 donde se encontraba la UPA, con un tiempo de duración de 1 hora cada equipo, los resultados obtenidos se muestran en la tabla 10. En la segunda prueba el maestro se dejó adentro de la caseta y el esclavo se colocó en la celda 2 con un tiempo de duración de 30 minutos, los resultados obtenidos se muestran en la tabla 11. El inicio de las pruebas fue a la 1 PM.

La disposición de celdas y UTR se muestra en la figura 21 con una vista superior de la estructura de la subestación.

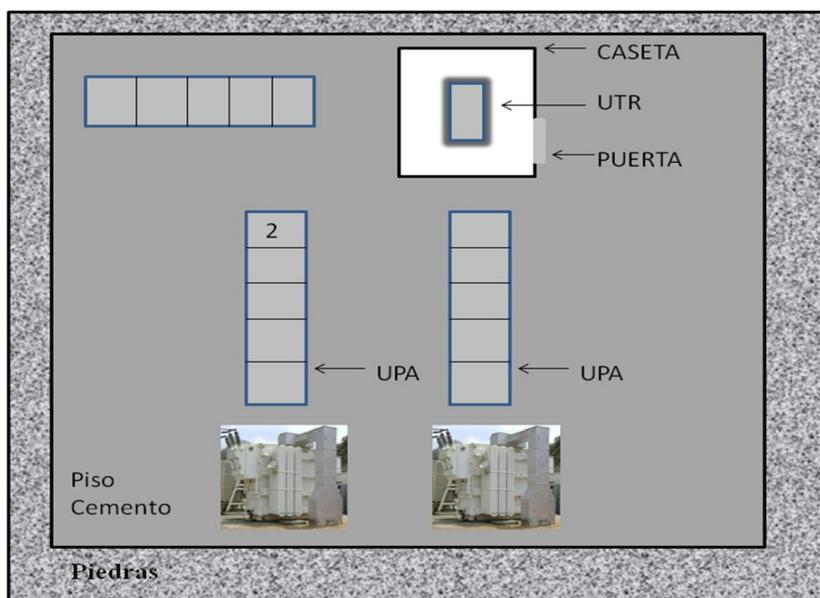


Figura 21. Estructura de Subestación de media tensión C. Fuente: Autor

Tabla 10. Resultados en subestación de media tensión C, Celdas 1

	Distancia(m)	Caseta	Caseta	BLUETOOTH	Wi-Fi
PUERTA	12	Abierta	Abierta	BER= 0	BER=0
PUERTA	12	cerrada	Cerrada	n/c	n/c
PUERTA	12	Cerrada	Abierta	BER= 0	BER=0
PUERTA	12	Abierta	Cerrada	n/c	n/c

Fuente: Autor

Tabla 11. Resultados en subestación de media tensión C, Celdas 2

	Distancia(m)	Caseta	Caseta	BLUETOOTH	Wi-Fi
PUERTA	14	Abierta	Abierta	BER= 0	BER=0
PUERTA	14	cerrada	Cerrada	n/c	n/c
PUERTA	14	Cerrada	Abierta	BER= 0	BER=0
PUERTA	14	Abierta	Cerrada	n/c	n/c

Fuente: Autor

#### 4.5 PRUEBA EN SUBESTACIÓN DE MEDIA TENSIÓN D.

Se realizaron tres pruebas en esta subestación de 12.4 kV (Ver anexo 20). En la primera se colocó el Maestro en la caseta de donde se encuentra ubicada la UTR, y Esclavo adentro de la celda 1 donde se encontraba la UPA (Ver Anexo 21), los resultados se muestran en la tabla 12. En la segunda prueba se colocó el esclavo fuera de la celda a una distancia de 1 m de ella (Ver figura 22), el maestro adentro de la caseta, en la tabla 13 se observan los resultados. En la tercera prueba, se colocó el esclavo al lado de la caja de control del transformador 1 (Ver anexo 22) obteniendo los resultados que se muestran en la tabla 14. Todas las pruebas fueron con un tiempo de duración de 30 minutos iniciando a las 11:30 AM.

La disposición de celdas y UTR se observa en la figura 22 (Vista superior):

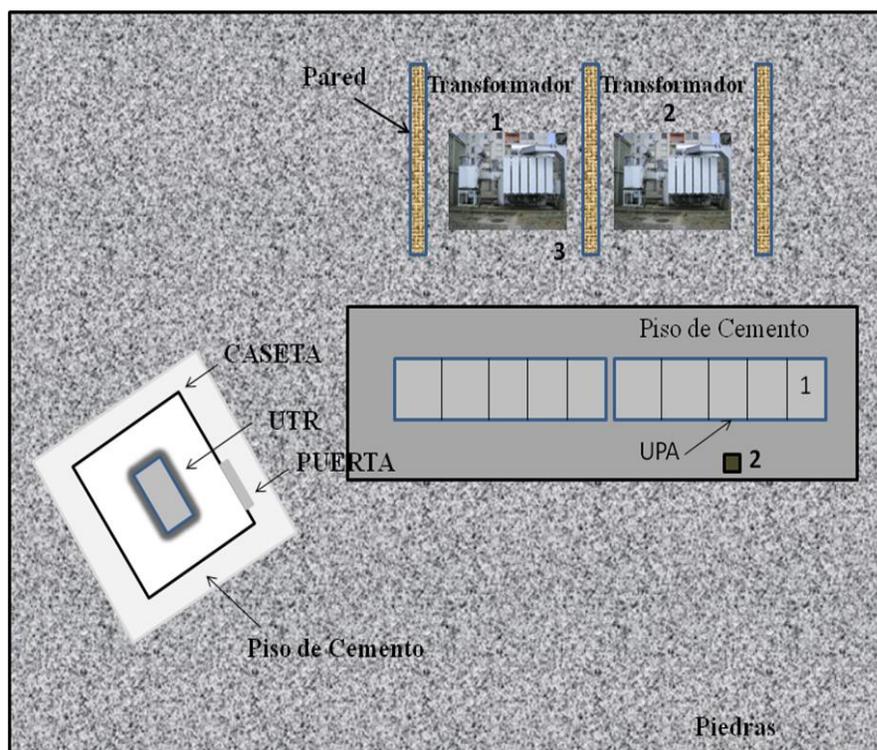


Figura 22. Estructura de Subestación de media tensión D. Fuente: Autor

Tabla 12. Resultados en subestación de media tensión D, Celda.

	Distancia	Caseta	Celda	BLUETOOTH	WI-Fi
PUERTA	20 m	Abierta	Abierta	n/c	n/c
PUERTA	20 m	cerrada	Cerrada	n/c	n/c
PUERTA	20 m	Cerrada	Abierta	n/c	n/c
PUERTA	20 m	Abierta	Cerrada	n/c	n/c

Fuente: Autor

Tabla 13. Resultados en subestación de media tensión D

	Distancia	Caseta	Afuera de la celda	BLUETOOTH	WI-Fi
PUERTA	20 m	Abierta	n/a	BER= 0	BER= 0
PUERTA	20 m	cerrada	n/a	BER= 0	BER= 0

Fuente: Autor

Tabla 14. Resultados en Transformador 1, Subestación de media tensión D

	Distancia	Caseta	Transformador	BLUETOOTH	WI-Fi
PUERTA	26 m	Abierta	n/a	BER= 0	BER= 0
PUERTA	26 m	cerrada	n/a	BER= 0	BER= 0
PUERTA	26 m	Cerrada	n/a	BER= 0	BER= 0
PUERTA	26 m	Abierta	n/a	BER= 0	BER= 0

Fuente: Autor

## ANÁLISIS DE RESULTADOS

Al realizar las pruebas en las instalaciones de Corporación 80-60, se esperaban los resultados mostrados en la tabla N° 6, ya que en este ambiente no hay interferencias o alteraciones que pudieran afectar la señal. Al hacer las pruebas por 12 horas permitió la comparación de la trama de 12500 bytes al menos 1440 veces. Sin obtener error. La tasa de error que se muestra en la tabla 7 concuerda con los cálculos teóricos realizados. Estos cálculos consistieron en cambiar la ubicación de los caracteres con respecto a la trama original y mediante el código ASCII en binario se contabilizaron los bits diferentes que habían entre las dos tramas.

Al observar las tablas 10, 11, 13 y 14, se aprecia en los resultados de las subestaciones, una tasa de error de 0 para los módulos Wi-Fi bluetooth, lo cual significó que los 12500 bytes enviados, llegaron correctamente al receptor, por lo que implica que la comunicación inalámbrica es completamente factible con los módulos Bluetooth y Wi-Fi, sin embargo, al observar los resultados de ZigBee (tabla 8 y 9), se observó que esta tecnología, si es afectada por este tipo de ambiente.

Para los dispositivos Bluetooth y Wi-Fi, como se observó en los resultados de las tablas 8, 9, 10, 11 y 12 de las subestaciones, al tener los módulos adentro de la celda con la puerta abierta, los dispositivos se comunicaban exitosamente. En el caso que se aprecia en la tabla 12, se debe a que en la subestación D, las celdas son más estrechas (Ver anexo 21) en comparación a las celdas de otras subestaciones (Ver anexo 19), y además de ser estrechas, también hay más equipos de material metálico instalado en dicha celda.

Estas pruebas permitieron descubrir, la mejor tecnología que se puede implementar en las subestaciones de media tensión (12.4kV, 4.8kV).

## CONCLUSIONES

Al realizar el estudio teórico de las tres tecnologías, la tecnología que mejor se adaptaba a la tasa de transferencia requerida por la empresa, era ZigBee, ya que la actual tasa entre la unidad terminal remota y la unidad de procesamiento de analógicos es de 4800 bps y la de ZigBee es de 250kbps. Además en cuanto al precio de los módulos, número de canales y dispositivos, requisitos de alimentación y rango, la mejor opción teóricamente era ZigBee, por el contrario, a efectos de los requerimientos técnicos de Corporación 80-60, esta tecnología tiene una desventaja con respecto a Bluetooth y Wi-Fi, la cual es el número de bytes por paquetes, dado que los paquetes que se envían entre la UTR y UPA tienen una longitud de 300 bytes y ZigBee 100 Bytes. Por esta razón la segunda opción beneficio-costos que cumple con los requerimientos técnicos de la empresa es Bluetooth.

Al hacer la comparación de los dispositivos a seleccionar para ser usados en este trabajo se comprobó el bajo costo de los módulos ZigBee, así como las diversas aplicaciones para las cuales fueron diseñadas cada tecnología.

Con las pruebas realizadas en campo, se demostró el uso efectivo de una comunicación inalámbrica en ambientes industriales, sin embargo, para los propósitos de Corporación 80-60 de comunicar la UTR con la UPA no es posible, debido a que al cerrar la celda donde se encuentra la UPA, no se puede realizar la conexión con el maestro, ubicado en la UTR. Una forma de lograr la comunicación de estos dispositivos es colocando la antena fuera de la celda, no obstante, como la NEMA especifica la robustez del equipo diseñado según su tipo. Las celdas que hay en la subestación son NEMA de tipo 4, esto quiere decir que están sellados contra agua y polvo, por esta razón y por también por el tipo de material con la que están construidas las celdas, se requeriría de permisos y del personal especializado para realizar la conexión de la antena hacia la Unidad de Procesamiento de Analógicos.

Las pruebas realizadas en distintas ubicaciones de las subestaciones y con distintos niveles de tensión, permitió demostrar que dichos niveles de tensión de los transformadores no afectan la señal en los módulos Bluetooth y Wi-Fi, por ello, estas tecnologías que pueden ser utilizadas en estos ambientes. La selección de Bluetooth o Wi-Fi dependerá de las características que requiere cada empresa.

Al observar los resultados de los módulos ZigBee se concluye que no son recomendados para este tipo de ambiente, sin embargo, no se menosprecia el trabajo que hace esta tecnología en gran variedad de aplicaciones.

Por lo tanto una comunicación inalámbrica en subestaciones eléctricas, si es factible siempre y cuando se solucione el problema de la celda.

## **RECOMENDACIONES**

Realizar pruebas en donde se midan niveles de campo electromagnéticos.

Realizar pruebas utilizando la señal de la UPA.

Como se apreció en las pruebas realizadas, la tecnología que mejor se ajustó al ambiente industrial fue Bluetooth, sin embargo, se recomienda hacer las pruebas con otros módulos Wi-Fi para poder afirmar con más certeza si es posible o no la comunicación.

Se recomienda hacer las pruebas por más tiempo: una semana, dos semanas y con condiciones climáticas diferentes, ya que todas las pruebas fueron realizadas con el día soleado con temperatura de 27°C.

Por último, si se adopta una tecnología para la comunicación inalámbrica en la subestación, realizar un circuito impreso en los entes especializados y buscar una solución para poder colocar los módulos afuera de la celda.

## REFERENCIAS BIBLIOGRÁFICAS

- [ 1] KiosKea. *Introducción a Wi-Fi (802.11 o Wi-Fi)* [en línea], <<http://es.kioskea.net/contents/wifi/wifiintro.php3Wi-Fi>> [Consulta: 2009]
- [2] Wi-Fi Alliance, *Discover and Learn* [en línea], <[http://www.wi-fi.org/discover\\_and\\_learn.php](http://www.wi-fi.org/discover_and_learn.php)> [Consulta 2009].
- [3] ISO/IEC 8802-11, *Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification*, ISO/IEC 8802-11:2005
- [4] Lindgren Magnus. *Physical Layer Simulations of the IEEE 802.11b Wireless LAN-standard*, (Master's Thesis), Suecia: Universidad de Lulea. 2001.
- [5] IEEE Computer Society, *Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11™-2006, (Revision of IEEE Std 802.11-1999).
- [6] Athanasopoulos Antonis, Topalis Evagelos, *Evaluation Analysis of the performance of IEEE 802.11b and IEEE 802.11g Standars*, IEEE communications Society, 2007
- [7] Houda Labiod, Hossan Afifi, Constantino De Santis. *Wi-Fi™, Bluetooth™, ZigBee™ and WiMax™?*, Springer, 2007.
- [8] IEEE Computer Society, *Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for wireless Personal Area Networks (WPANs)*, IEEE Std 802.15.1™-2005, (Revision of IEEE Std 802.15.1-2002).
- [9] IEEE Computer Society, *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std 802.15.4™-2005, (Revision of IEEE Std 802.15.4-2003).

[10] Bluetooth, *Architecture – Radio* [en línea]. <  
[http://www.bluetooth.com/Spanish/Technology/Works/Pages/Architecture\\_\\_Radio.aspx](http://www.bluetooth.com/Spanish/Technology/Works/Pages/Architecture__Radio.aspx)>, [Consulta 2009]

[11] Ergen Coleri Sinem, *ZigBee/IEEE 802.15.4 Summary*, paper, 2004

[12] IEEE 802.15 WPAN<sup>TM</sup> Task Group, *IEEE 802.15 TG4 Current Status*, IEEE 802.15, 2006.

[13] Tomasi, *Sistema de Comunicaciones Electronicas*, Pretice Hall, cuarta edición

[14] RF Café, *Bit Error Rate*, [en línea], <<http://www.rfcafe.com/>> [Consulta 2009].

## BIBLIOGRAFÍAS

Villela Tejeda Héctor, *Manual de C*, <http://www.fismat.umich.mx/mn1/manual/>, [Consulta 2009].

Urrutia Gorka, *El rincón de C*, <http://www.elrincondelc.com/cursoc/cursoc.html> [Consulta 2009]

Ridolfi Oscar, *Acerca del Puerto Serie*, <http://www.lu4ev.8m.com/noti8.html> [consulta 2009]

Microchip, *Section 17 UART*. Microchip Technology Inc. 2007.

Microchip, *dsPIC33FJ64GP202 Data Sheet*, Microchip Technology Inc. 2009.

Murcia Jose, *Puerto Serie RS-232*, ITIG, 2008

Bluetooth <http://www.bluetooth.com>, [Consulta 2009]

Artifox, *Introducción a los Microcontroladores*, Centro de Desarrollo e Investigación en Mecatronica, 2007

Wireless Wiki: <http://wireless.wikia.com/wiki/Contents> [consulta 2009]

Venezuelan Trasmision & Distribution, *Manual Técnico Unidad Terminal Remota V94-R*, Venezuela, Edición Inicial 1996.

Venezuelan Trasmision & Distribution, Manual de Usuario la Unidad de Procesamiento de Analógico modelo DAS-36/ODT4, Venezuela, Segunda edición 2000

Página de la empresa proveedora de equipos a la Corporación 80-60, <http://www.mouser.com/>,

Jennic, *Co-existence of IEEE 802.15.4 at 2.4 GHz*, Application Note, 2008

Kho Thomas, *Steganography in th 802.15.4 Physical Layer*, paper, 2007.

Willians Joseph, *The IEEE 802.11b Security Problem, Part 1*, IT PRO, 2001

Mahasukhon Puttipong, *Hempel Michael, BER Analysis of 802.11b Networks under Mobility*, IEEE Comminications Society, 2007

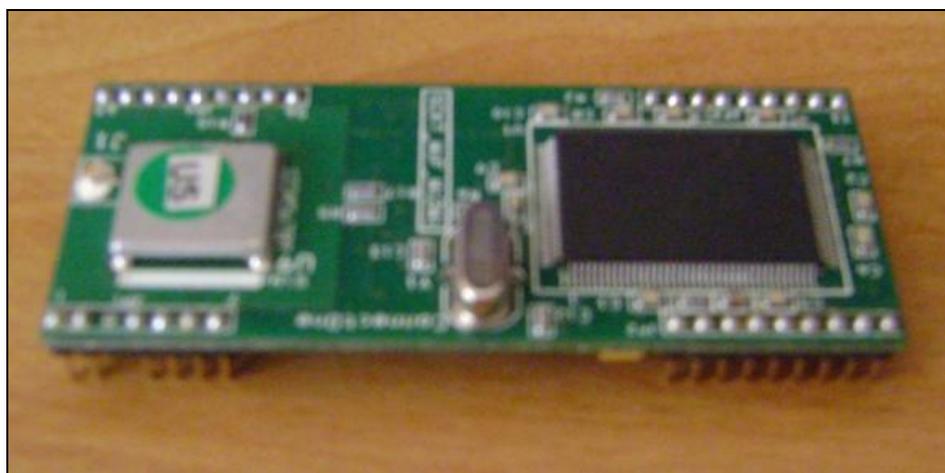
Park Seong Hong, Shin Young Soo, *Packet Error Rate Analysis of ZigBee Under WLAN and Bluetooth Interferences*, IEEE Transactions on Wireless Communications, Vol 6, 2007

Yusoff Rashid, *Bluetooth Performacnce Analysis in Personal Area Network (PAN)*, International RF and Microwave Conference Proceedings, 2006

Vásquez Belkys, *Molina Julio, Instructivo y normalización para la elaboración de trabajos especiales de grado*, Escuela de Ingeniería Eléctrica U.C.V, 2008

[ANEXO N° 1]

[Módulo Wi-Fi seleccionado]



Nombre: Secure Socket iWiFi™

Fabricante: Connect One

Alimentación: 3.3V +/- 10%.

Rango de temperatura de operación: -20° a 75 °C

Consumo de potencia: TX: 260mA; RX: 190mA

Conector de antena: U.FL

Interfaz: serial TTL.

Formato de dato serial: Carácter asincrónico; 8;N;1.

Se adquirieron dos módulos Secure Socket iWiFi™ para la realización de las pruebas.

[ANEXO N° 2]

[Módulo Bluetooth seleccionado]



Numero de parte: BISMS02BI-NA

Fabricante: Ezurio

Alimentación: 3.3V – 7V

Rango de temperatura de operación: -40° a 85 °C

Consumo de potencia: TX: +6dBm;

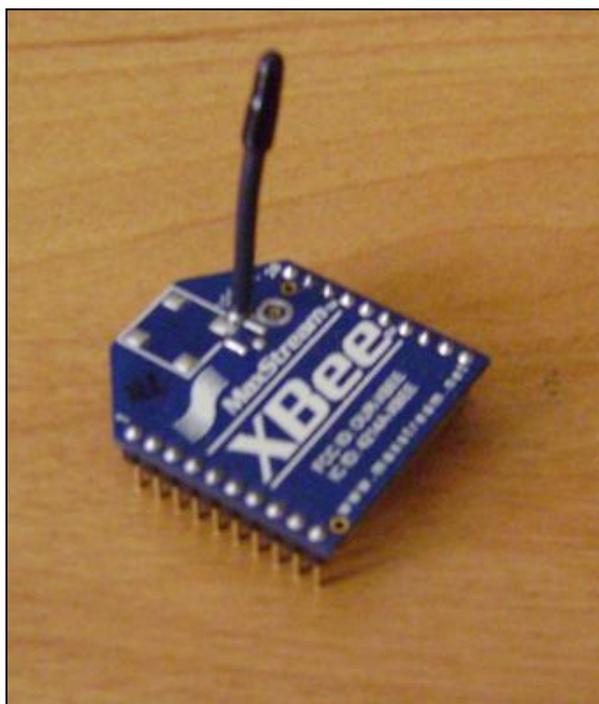
Conector de antena: U.FL

Interfaz: serial TTL.

Formato de dato serial: Carácter asincrónico; 8;N;1.

[ANEXO N° 3]

[Módulo Zigbee seleccionado]



Nombre: XBee 802.15.4 (Serie 1)

Fabricante: Digi

Alimentación: 2.8V - 3.4V

Rango de temperatura de operación: -40° a 85 °C

Consumo de potencia: TX: 45mA; RX: 50mA.

Antena: Wire Whip

Interfaz: 3.3V CMOS UART

Formato de dato serial: Carácter asincrónico; 8;N;1.

## [ANEXO N° 4]

[Hoja de datos del dsPIC33FJ64GP202]



## dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

### High-Performance, 16-Bit Digital Signal Controllers

#### Operating Range:

- Up to 40 MIPS operation (at 3.0-3.6V):
  - Industrial temperature range (-40°C to +85°C)
  - Extended temperature range (-40°C to +125°C)
- Up to 20 MIPS operation (at 3.0-3.6V):
  - High temperature range (-40°C to +140°C)

#### High-Performance DSC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set
- 16-bit wide data path
- 24-bit wide instructions
- Linear program memory addressing up to 4M instruction words
- Linear data memory addressing up to 64 Kbytes
- 83 base instructions: mostly 1 word/1 cycle
- Two 40-bit accumulators with rounding and saturation options
- Flexible and powerful addressing modes:
  - Indirect
  - Modulo
  - Bit-Reversed
- Software stack
- 16 x 16 fractional/integer multiply operations
- 32/16 and 16/16 divide operations
- Single-cycle multiply and accumulate:
  - Accumulator write back for DSP operations
  - Dual data fetch
- Up to  $\pm 16$ -bit shifts for up to 40-bit data

#### Direct Memory Access (DMA):

- 8-channel hardware DMA
- Up to 2 Kbytes dual ported DMA buffer area (DMA RAM) to store data transferred via DMA:
  - Allows data transfer between RAM and a peripheral while CPU is executing code (no cycle stealing)
- Most peripherals support DMA

#### Timers/Capture/Compare/PWM:

- Timer/Counters, up to five 16-bit timers:
  - Can pair up to make two 32-bit timers
  - One timer runs as a Real-Time Clock with an external 32.768 kHz oscillator
  - Programmable prescaler
- Input Capture (up to four channels):
  - Capture on up, down or both edges
  - 16-bit capture input functions
  - 4-deep FIFO on each capture
- Output Compare (up to four channels):
  - Single or Dual 16-bit Compare mode
  - 16-bit Glitchless PWM mode
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions

#### Interrupt Controller:

- 5-cycle latency
- Up to 49 available interrupt sources
- Up to three external interrupts
- Seven programmable priority levels
- Five processor exceptions

#### Digital I/O:

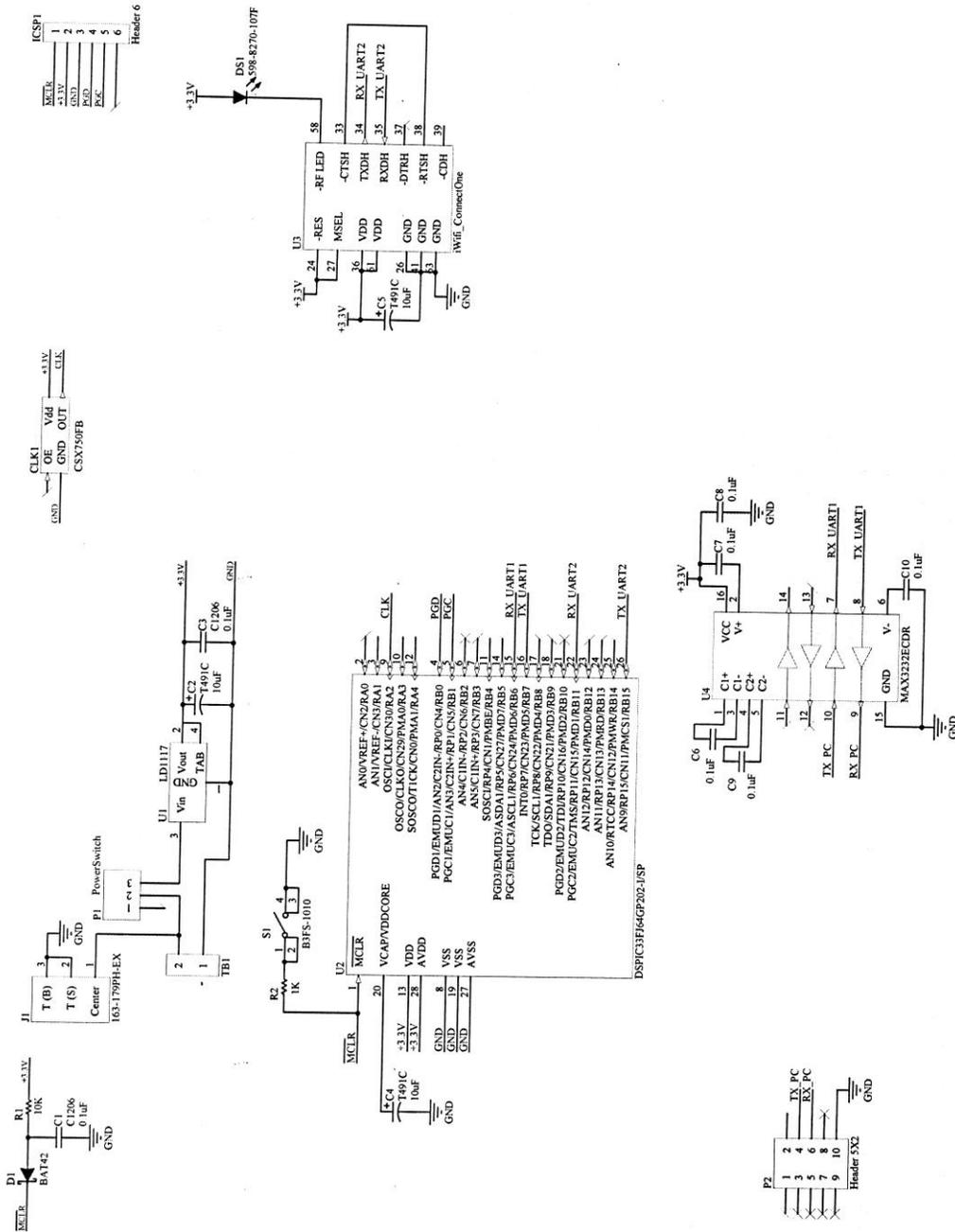
- Peripheral pin Select functionality
- Up to 35 programmable digital I/O pins
- Wake-up/Interrupt-on-Change for up to 31 pins
- Output pins can drive from 3.0V to 3.6V
- Up to 5V output with open drain configuration
- All digital input pins are 5V tolerant
- 4 mA sink on all I/O pins

#### On-Chip Flash and SRAM:

- Flash program memory (up to 128 Kbytes)
- Data SRAM (up to 16 Kbytes)
- Boot, Secure and General Security for program Flash

# [ANEXO N° 5]

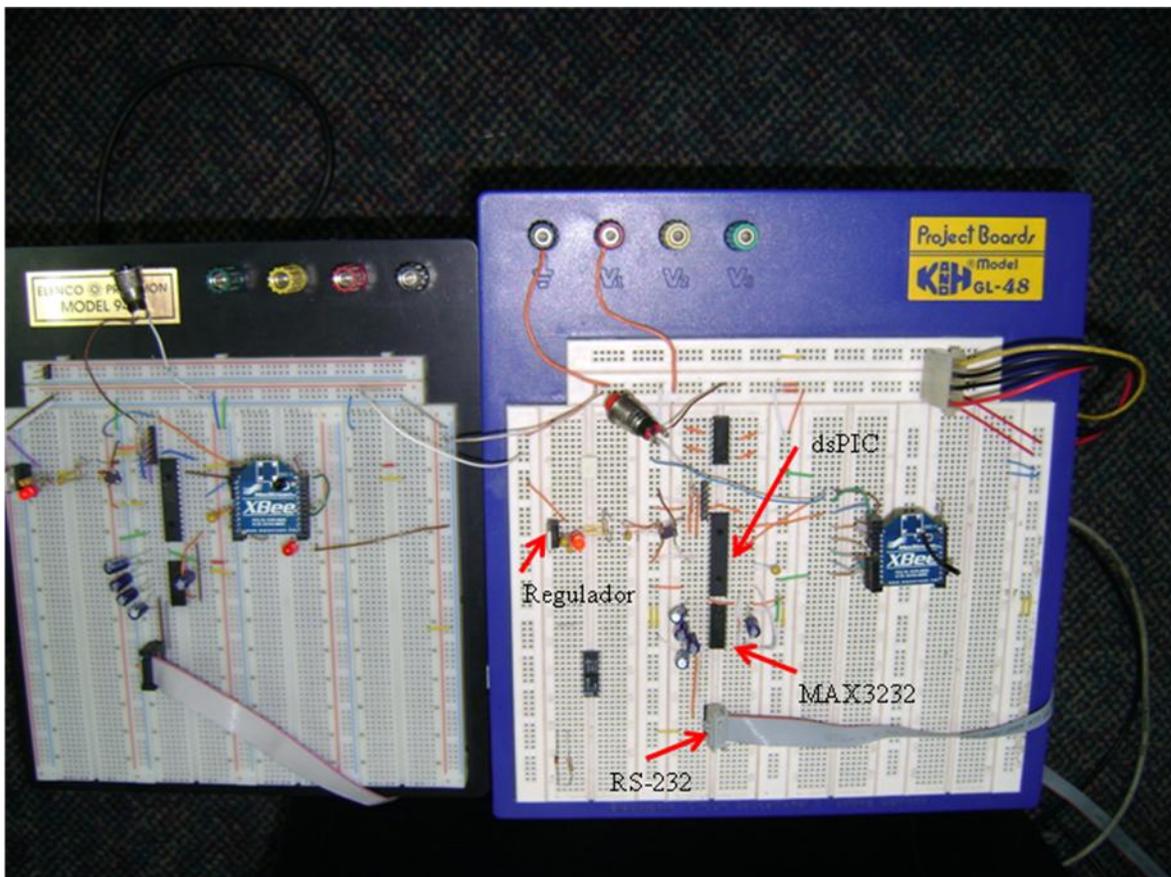
## [ESQUEMA DE CONEXIONES]



Esquemas de conexiones: regulador 3.3V, dsPIC, MAX3232, Módulo Wi-Fi

## [ANEXO N° 6]

[Montaje de módulos en Protoboard]



Conexión de módulos para la posterior verificación de comunicación entre la computadora y el módulo, además estos montajes, permitieron configurar la comunicación entre los módulos.

[ANEXO N° 7]  
[Código de Programación]

```

/*****
*****
PROGRAMA "CONFIGURACION DE DSPIC33FJ64GP202 PARA MODULO MAESTRO WIFI"
AUTOR: VÍCTOR FERNÁNDEZ
UCV
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA.

*****
*****/

#include <p33FJ64GP202.h>
#include <uart.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

// Activacion del oscilador interno
_FOSCSEL(FNOSC_FRC); // COMIENZA CON EL
OSCILADOR FRC

_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_EC); //HABILITA SOLO EL
CAMBIO DE RELOJ

//OSC2 ES RELOJ O/P //
RELOJ EXTERNO

_FWDT(FWDTEN_OFF); //WACTH
DOG DISABLE

void espera(long E); //DECLARACION DE LA
FUNCION ESPERA
void ini_timer2(void); //DECLARACION DE LA
FUNCION INICIALIZACION DEL TIMER 2
void ini_timer1(void); //DECLARACIÓN DE LA
FUNCION INICIALIZACION DEL TIMER 1

char
trama[134]={'A','b','C','d','E','f','G','h','I','j','K','l','M','n','O',
'p','Q','r','S','t','U','v',

'W','x','Y','z','a','b','c','d','e','f','g','h','i','j','k','l','m','n',
'o','p','q','r',

's','t','u','v','w','x','y','z','a','B','c','D','e','F','g','H','i','J',
'k','L','m','N',

```

```

'o','P','q','R','s','T','u','V','w','X','Y','Z','A','B','C','D','E','F
','G','H','I','J',

'K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','a','B
','c','D','e','F',

'g','H','i','J','k','L','m','N','o','P','q','R','s','T','u','V','w','X
','y','Z','a','b',
        'c','d'};
//134CARACTERES *8=1072 bits

char buf[4314]; // buffer que almacena
los caracteres a comparar 4288+26:4314
char buf1[50]; // buffer que almacena
los caracteres enviados desde la computadora.

//COMANDOS AT PARA LA CONFIGURACIÓN DE LOS MÓDULOS:
unsigned char comando0[11]
={ 'a','t','+', 'i','w','l','c','h','=', '5',13}; // Canal 5
unsigned char comando1[27]
={ 'a','t','+', 'i','w','l','s','i','=', '!', 'c','o','r','p','o','r','a',
'c','i',
        'o','n','8','0','-
','6','0',13}; //Establece a un SSID en una red Ac HOC

unsigned char comando2[9] ={'a','t','+', 'i','w','l','b','m',13};
//Modo b 11 Mbps
unsigned char comando3[21]
={ 'a','t','+', 'i','d','i','p','=', '1','9','2','.', '1','6','8','.', '2',
'0','.', '2',13}; //IP
unsigned char comando4[23]
={ 'a','t','+', 'i','s','n','e','t','=', '2','5','5','.', '2','5','5','.',
'2','5','5','.', '0',13}; //Mascara
unsigned char comando5[21]
={ 'a','t','+', 'i','i','p','g','=', '1','9','2','.', '1','6','8','.', '2',
'0','.', '1',13}; //Gateway

unsigned char comando6[22]
={ 'a','t','+', 'i','h','s','t','n','=', 'c','o','r','p','o','r','a','c',
'i','o','n','1',13}; //nombre del host
unsigned char comando7[13]
={ 'a','t','+', 'i','l','t','c','p',':', '9','.', '2',13}; //Crea socket
TCP
unsigned char comando8[13]
={ 'a','t','+', 'i','s','r','c','v',':', '0','0','0',13}; //chequea buffer
unsigned char comando9[20]
={ 'a','t','+', 'i','s','s','n','d','%', ':', '0','0','0','.', '1','2',
', '8','6','4', ':'};
//tamaño del buffer q
contendra 96 veces la trama

```

```

unsigned char comando10[14]
={'a','t','+','i','s','f','s','h','%',':','0','0','0',13}; //borra
buffer de salida
unsigned char comando[13]
={'¿','E','S','T','A','S','_','L','I','S','T','O','?'}; //Pregunta
para el esclavo

//COMANDOS DEL ESCLAVO:
unsigned char coma0[12];
unsigned char coma1[22];
unsigned char coma2[24];
unsigned char coma3[22];
unsigned char coma4[23];
unsigned char coma5[25];
unsigned char coma6[25];
unsigned char coma7[14];
unsigned char coma8[15];
unsigned char coma9[15];
unsigned char coma10[22];
unsigned char coma11[12];
unsigned char coma12[13];
unsigned char coma13[16];

//VARIABLES GLOBALES:
unsigned int Indice;
unsigned int Numero_Paquete;
unsigned int m,x;
unsigned int indexRX=0;
unsigned int indexRX1=0;
int analiza_trama;
int count_byte;
int count_bytel;
int flagtrama=0;
int flagT2;
int flagT1;
int contador2500;
int B,B1;
float contador;
long delay;

#include "funciones.h"

/***** FUNCION PRINCIPAL *****/
*****/

int main (void)

{
    contador=0;
    contador2500=0;
    Numero_Paquete=0;
    flagT2=-1;
    flagT1=-1;

```

```

analiza_trama=0;
count_byte=0;

Ini_clock();
//Inicialización de reloj
ini_buf();
//Inicialización de buf
ini_buf1();
//Inicialización de buf1

RPOR3bits.RP7R=3;
//asignación de puerto RP7 UART1 TX
RPINR18bits.U1RXR = 6;
//Asignación de puerto RP6 UART1 RX

RPOR7bits.RP15R=5;
//Asignación de puerto RP15 UART2 TX
RPINR19bits.U2RXR=11;
//Asignación de puerto RP11 UART2 RX

TRISBbits.TRISB15 = 0;
//Asignación de puertos restantes como salida
TRISBbits.TRISB14 = 0;
TRISBbits.TRISB13 = 0;
TRISBbits.TRISB12 = 0;
TRISBbits.TRISB10 = 0;
TRISBbits.TRISB9= 0;
TRISBbits.TRISB8= 0;
TRISBbits.TRISB5= 0;
TRISBbits.TRISB4= 0;
TRISBbits.TRISB3= 0;
TRISBbits.TRISB2= 0;

LATBbits.LATB15=0;
//Asignación de puertos como bajos
LATBbits.LATB14=0;
LATBbits.LATB13=0;
LATBbits.LATB12=0;
LATBbits.LATB10=0;
LATBbits.LATB9=0;
LATBbits.LATB8=0;
LATBbits.LATB5=0;
LATBbits.LATB4=0;
LATBbits.LATB3=0;
LATBbits.LATB2=0;

PORTB = 0x0000;

Ini_uart1();
//Inicialización de UART1
Ini_uart2();
//Inicialización de UART2

```

```

        IFS1bits.U2TXIF = 0;           // Clear
the Transmit Interrupt Flag
        IEC1bits.U2TXIE = 1;         //
Transmit Interrupts disable

        IFS0bits.U1TXIF = 0;         // Clear
the Transmit Interrupt Flag
        IEC0bits.U1TXIE = 1;         //
Transmit Interrupts disable

        ini_timer2();                //
Inicialización del timer 1
        ini_timer1();                //
Inicialización del timer 2

        espera(2000);                //
Espera 2 seg antes de enviar primer comando
        Enviar_Comando0();

        while(1)                    // Bucle infinito
        {

            if(analiza_trama)
            {

                if(B1>2)              //Si B1
es mayor a 2 hay datos en el buf1
                {
                    Enviar_Comandos(); //Envia
los caracteres recibidos desde la computadora
                    B1=0;
                }

                if(buf[4]=='w')
                {
                    switch (buf[7])
                    {
                        case 'h': espera(2000);Enviar_Comando1();break;
                        case 'i': espera(2000);Enviar_Comando2();break;
                        case 'm': espera(2000);Enviar_Comando3();break;
                    }
                }

                if(buf[4]=='s')
                {
                    switch (buf[5])
                    {
                        case 'n': espera(2000);Enviar_Comando5();break;
                    }
                }

                if((buf[4]=='d') && (buf[7]=='='))

```

```

        {
        espera(2000);Enviar_Comando4();
        }
        if((buf[4]=='i') &&(buf[7]=='='))
        {
        espera(2000);Enviar_Comando6();
        }
        if((buf[4]=='h') &&(buf[7]=='n'))
        {
        espera(5000);Enviar_Comando7();
        }
        if((buf[4]=='l') &&(buf[7]=='p'))
        {
        espera(10000);Enviar_Comando8();
        }
        if((buf[2]=='E') &&(buf[9]=='2'))
        {
        espera(2000);Enviar_Comando8();
        }
        if((buf[16]=='E') &&(buf[23]=='2'))
        {
        espera(2000);Enviar_Comando8();
        }
        if(buf[18]=='C')
        {
        espera(2000);Enviar_Comando9();
        buf[18]=x;
        }
        if((buf[5]=='r') &&(buf[16]=='0'))
        {
        espera(3000); Enviar_Comando8();
        }
        if(buf[12]=='?')
        {
        espera(20000);Enviar_Comando8();           // Envía la
trama otra vez, se repite el ciclo.
        }
        if(buf[19]==':')
        {
        buf[19]=x;
        espera(3000);flagtrama=1;
        }

        analiza_trama=0;

    }

    if (flagtrama)
    {
        Enviar_Trama();
    }
}

```

```

}          // Cierre del main del programa

/*****FIN DEL
MAIN*****/

/*****          INTERRUPCION DE
RECEPCION DE UART2          *****/

void __attribute__((interrupt, no_auto_psv)) _U2RXInterrupt(void)
{
    IFS1bits.U2RXIF = 0;
        // Borra el flag de interrupcion
}
/*****FIN DE
INTERRUPTACION*****/

/*****          INTERRUPCION DEL
TRANSMISOR DEL UART2          *****/

void __attribute__((interrupt, no_auto_psv)) _U2TXInterrupt(void)
{
    IFS1bits.U2TXIF = 0;
}
/*****FIN DE
INTERRUPTACION*****/

/*****          INTERRUPCION DE
RECEPCION DE UART1          *****/

void __attribute__((interrupt, no_auto_psv)) _U1RXInterrupt(void)
{
    IFS0bits.U1RXIF = 0;
        // Borra el flag de interrupcion
}
/*****FIN DE
INTERRUPTACION*****/

/*****          INTERRUPCION DEL
TRANSMISOR DEL UART1          *****/

void __attribute__((interrupt, no_auto_psv)) _U1TXInterrupt(void)
{
    IFS0bits.U1TXIF = 0;
}

```

```

/*****FIN DE
INTERRUPCION*****/

/***** FUNCION:
INICIALIZACION DEL TIMER2 *****/

void ini_timer2(void)

{
T2CONbits.TON=0;           //disable timer
T2CONbits.TCS=0;          //select internal instruction
cycle clock
T2CONbits.TGATE=0;        //disable gated timer
mode
T2CONbits.TCKPS= 0b00;    //select1:1 prescaler
TMR2=0x00;                //clear timer register
PR2=88000;                //load the period value,
interrumpe cada 2.2mseg

IPC1bits.T2IP=7;          //set timer Interrupt priority
level
IFS0bits.T2IF=0;          //clear timer2 interrupt flag
IEC0bits.T2IE=1;         //enable timer2 interrupt

T2CONbits.TON=1;         //start timer
}

/***** INTERRUPCION DEL
TIMER2 *****/

void __attribute__((__interrupt__, auto_psv)) _T2Interrupt(void)
{
IFS0bits.T2IF = 0;        //Clear Timer1 interrupt flag

    if(delay)
    {
        delay--;
    }

    if (U2STAbits.OERR)    //verifica si hay overflow
        U2STAbits.OERR = 0; //borra la data recibida

    if (DataRdyUART2())   //Hay dato para leer en la
UART?
    {
        buf[indexRX]=ReadUART2(); //Almace el caracter en "buf"
        U1TXREG=buf[indexRX];     //envia el caracter a la
computadora
        indexRX++;                //Aumenta la posicion de
"buf"
        count_byte++;             //Cuenta los caracteres
recibidos
}

```

```

        flagT2=0;                //contador
    }
else                                //No hay dato en la UART
    {
        if(flagT2>=0)
            flagT2++;           //Aumenta contador
        if(flagT2>4)           //Luego de haber pasado 4
tiempos de caracter entra al ciclo
    {
        analiza_trama=1;       //Activa la bandera
        B=count_byte;         //almacena los caracteres
recibidos en "B"
        flagT2=-1;            //Coloca el contador en -
1
        count_byte=0;         //Inicializa "count_byte"
en 0
        indexRX=0;           //vuelve a la posición 0
de "buf"
    }
}

```

```

/***** FUNCION:
INICIALIZACION DEL TIMER1 *****/

```

```

void ini_timer1(void)

```

```

{
T1CONbits.TON=0;                //disable timer
T1CONbits.TCS=0;                //select internal instruction
cycle clock
T1CONbits.TGATE=0;              //disable gated timer
mode
T1CONbits.TCKPS= 0b00;          //select1:1 prescaler
TMR1=0x00;                       //clear timer registrer
PR1=45820;                       //load the period value,
interrumpe cada 1.1458mseg

IPC0bits.T1IP=7;                //set timer Interrupt priority
level
IFS0bits.T1IF=0;                //clear timer2 interrupt flag
IEC0bits.T1IE=1;                //enable timer2 interrupt

T1CONbits.TON=1;                //start timer
}

```

```

/***** INTERRUPCION DEL
TIMER1 *****/

```

```

void __attribute__((__interrupt__, auto_psv)) _T1Interrupt(void)
{

```

```

IFS0bits.T1IF = 0;                //Clear Timer1 interrupt flag

```

```

if (U1STAbits.OERR)           //verifica si hay overflow
    U1STAbits.OERR = 0;       //borra la data recibida

if (DataRdyUART1())
{
    buf1[indexRX1]=ReadUART1();
    indexRX1++;
    count_byte1++;
    flagT1=0;
}
else
{
    if(flagT1>=0)
        flagT1++;
    if(flagT1>4)
    {
        analiza_trama=1;
        B1=count_byte1;
        flagT1=-1;
        count_byte1=0;
        indexRX1=0;
    }
}
}

/*****
INICIALIZACION DEL TIMER1      *****/
FUNCION:

```

```

/*****
*****
PROGRAMA "CONFIGURACION DE DSPIC33FJ64GP202 PARA MODULO ESCLAVO WIFI"
AUTOR: VICTOR FERNANDEZ
UCV
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA.

*****
*****/

#include <p33FJ64GP202.h>
#include <uart.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

// Activacion del oscilador interno
_FOSCSEL(FNOSC_FRC); // COMIENZA CON EL
OSCILADOR FRC

_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_EC); //HABILITA SOLO EL
CAMBIO DE RELOJ

//OSC2 ES RELOJ O/P //
RELOJ EXTERNO

_FWDT(FWDTEN_OFF); //WACTH
DOG DISABLE

//FUNCIONES:
int contbit(unsigned char x); //DECLARACIÓN DE LA
FUNCIÓN QUE CUENTA LOS BITS ERRADOS.
void espera( long E); //DECLARACIÓN DE LA
FUNCIÓN ESPERA
void ini_timer2(void); //DECLARACION DE LA
FUNCION INICIALIZACION DEL TIMER 2
void ini_timer1(void); //DECLARACIÓN DE LA
FUNCION INICIALIZACION DEL TIMER 1
void verifica_trama(); //DECLARACIÓN DE LA
FUNCIÓN DE ALGORITMO REALIZADO
void muestra_bits(void); //MUESTRA LOS BITS
ERRADOS.
int calcula_bits(void); //DECLARACION
DE LA FUNCIÓN DE CALCULAR BER

char
trama[134]={'A','b','C','d','E','f','G','h','I','j','K','l','M','n','O
','p','Q','r','S','t','U','v',

```

```

'W','x','Y','z','a','b','c','d','e','f','g','h','i','j','k','l','m','n
','o','p','q','r',

's','t','u','v','w','x','y','z','a','B','c','D','e','F','g','H','i','J
','k','L','m','N',

'o','P','q','R','s','T','u','V','w','X','y','Z','A','B','C','D','E','F
','G','H','I','J',

'K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','a','B
','c','D','e','F',

'g','H','i','J','k','L','m','N','o','P','q','R','s','T','u','V','w','X
','y','Z','a','b',
        'c','d']; //134CARACTERES *8=1072 bits

char buf[4314];                // buffer que almacena los
caracteres a comparar 4288+26:4314
char buf1[50];                // buffer que almacena los
caracteres enviados desde la computadora.

//COMANDOS AT PARA LA CONFIGURACIÓN DE LOS MÓDULOS:
unsigned char coma0[11]
={'a','t','+','i','w','l','c','h','=','5',13};                // Canal 5

unsigned char coma1[27]
={'a','t','+','i','w','l','s','i','=','!','c','o','r','p','o','r','a',
'c','i','o','n',
        '8','0','-','6','0',13}; //

Se une a un SSID en una red Ac HOC
unsigned char coma2[9]
={'a','t','+','i','w','l','b','m',13}; //Modo b 11 Mbps
unsigned char coma3[21]
={'a','t','+','i','d','i','p','=','1','9','2','.','1','6','8','.','2',
'0','.','3',13}; //IP
unsigned char coma4[23]
={'a','t','+','i','s','n','e','t','=','2','5','5','.','2','5','5','.','
'2','5','5','.','0',13}; //Mascara
unsigned char coma5[21]
={'a','t','+','i','i','p','g','=','1','9','2','.','1','6','8','.','2',
'0','.','1',13}; //Gateway
unsigned char coma6[22]
={'a','t','+','i','h','s','t','n','=','c','o','r','p','o','r','a','c',
'i','o','n','2',13}; //nombre del host
unsigned char coma7[26]
={'a','t','+','i','s','t','c','p',':','1','9','2','.','1','6','8','.','
'2',
'0','.','2','.','9','.','8',13}; //Se une al socket creado por el
Maestro
unsigned char coma8[13]
={'a','t','+','i','s','d','m','p',':','0','0','0',13}; // Borra buffer

```

```

unsigned char coma9[14]
={'a','t','+','i','s','f','s','h','%',':','0','0','0',13};// Borra
socket
unsigned char coma10[18]
={'a','t','+','i','s','r','c','v','%',':','0','0','0',' ',' ','2','1','4','4',
13};//Chequea buffer de 2144B
unsigned char coma11[16]
={'a','t','+','i','s','s','n','d','%',':','0','0','0',' ',' ','8','%'}; //B
orra buffer de salida
unsigned char coma12[9]
={'C','O','R','R','E','C','T','O','X'}; //Respuesta al Maestro
unsigned char coma13[16]
={'T','R','A','M','A',' ','I','M','C','O','M','P','L','E','T','A'}; //
Indica si no llegan los bytes

//COMANDOS DEL MAESTRO:
unsigned char comando[13];
unsigned char comando0[12];
unsigned char comando1[30];
unsigned char comando2[22];

unsigned char comando3[24];
unsigned char comando4[22];
unsigned char comando5[23];

unsigned char comando6[14];
unsigned char comando7[14];
unsigned char comando8[15];
unsigned char comando9[15];
unsigned char comando10[22];
unsigned char comando11[12];
unsigned char comando12[19];
unsigned char comando13[14];

//VARIABLES GLOBALES
unsigned int Indice;
unsigned int m;
int bits_acumulados;
int bits_errados;
int bits_por_millon;
unsigned char MSB,msb;
unsigned char LSB,lsb;
float x,j;
int i,k,r,count;
int Numero_Paquete;
unsigned int indexRX=0;
unsigned int indexRX1=0;
int analiza_trama;
int enviar;
int count_byte;
int count_bytel;
int R,t;

```

```

int L,B,B1;
int flagtrama=0;
int flag_bits;
int MAX_bits;
int flagT2;
int flagT1;
int Bits_er;
int contador2500;
int resultado;
float contador;
char Q;
int ber;
long delay;

#include "funciones.h"

/***** FUNCION PRINCIPAL *****/

int main (void)

{

    Numero_Paquete=0;
    contador=0;
    enviar=0;
    contador2500=0;
    bits_errados=0;
    flagT2=-1;
    flagT1=-1;
    analiza_trama=0;
    count_byte=0;
    count_bytel=0;
    bits_acumulados = 0;
    Bits_er=0;
    MAX_bits=-1;
    bits_por_millon=0;
    L = 0;
    B=0;
    B1=0;
    k=0;

    Ini_clock();
    //Inicialización de reloj
    ini_buf();
    //Inicialización de buf
    ini_buf1();
    //Inicialización de buf1

    RPOR3bits.RP7R=3;
    //ASIGNACION DE PUERTO RP7  UART1 TX

```

```

RPINR18bits.U1RXR = 6;
//ASIGNACION DE PUERTO RP6  UART1 RX

RPOR7bits.RP15R=5;
//ASIGNACION DE PUERTO RP15  UART2 TX
RPINR19bits.U2RXR=11;
//ASIGNACION DE PUERTO RP11  UART2 RX

TRISBbits.TRISB15 = 0;
//Asignación de puertos restantes como salida
TRISBbits.TRISB14 = 0;
TRISBbits.TRISB13 = 0;
TRISBbits.TRISB12 = 0;
TRISBbits.TRISB10 = 0;
TRISBbits.TRISB9= 0;
TRISBbits.TRISB8= 0;
TRISBbits.TRISB5= 0;
TRISBbits.TRISB4= 0;
TRISBbits.TRISB3= 0;
TRISBbits.TRISB2= 0;

LATBbits.LATB15=0;
//Asignación de puertos como bajos
LATBbits.LATB14=0;
LATBbits.LATB13=0;
LATBbits.LATB12=0;
LATBbits.LATB10=0;
LATBbits.LATB9=0;
LATBbits.LATB8=0;
LATBbits.LATB5=0;
LATBbits.LATB4=0;
LATBbits.LATB3=0;
LATBbits.LATB2=0;
PORTB = 0x0000;

Ini_uart1();
//Inicialización de UART1
Ini_uart2();
//Inicialización de UART2
IFS1bits.U2TXIF = 0; //
Clear the Transmit Interrupt Flag
IEC1bits.U2TXIE = 1; //
Transmit Interrupts disable

IFS0bits.U1TXIF = 0; //
Clear the Transmit Interrupt Flag
IEC0bits.U1TXIE = 1; //
Transmit Interrupts disable

ini_timer1();
// Inicialización del timer 1
ini_timer2();
// Inicialización del timer 2

```

```

espera(2000); //espera 2 seg antes de enviar
primer comando
Enviar_Coma0();

while(1) // Bucle infinito
{

if(analiza_trama)
{

buf1          if(B1>2) //Si B1 es mayor a 2 hay datos en el
               {
               Enviar_Comandos();
               B1=0;
               }

if(buf[4]=='w')
{
    switch (buf[6])
    {
    case 'c': espera(2000);Enviar_Coma1();break;
    case 's': espera(2000);Enviar_Coma2();break;
    case 'b': espera(2000);Enviar_Coma3();break;
    }
}
if(buf[4]=='s')
{
    switch (buf[5])
    {
    case 'n': espera(2000);Enviar_Coma5();break;
    case 't': espera(2000);Enviar_Coma8();break;
    case 'd': espera(5000);Enviar_Coma9();break;
    case 'f': espera(5000);Enviar_Coma11();break;
    }
}
if((buf[4]=='d') && (buf[6]=='p'))
{
espera(2000);Enviar_Coma4();
}
if((buf[4]=='i') && (buf[6]=='g'))
{
espera(2000);Enviar_Coma6();
}
if((buf[4]=='h') && (buf[6]=='t'))
{
espera(5000);Enviar_Coma7();
}
if((buf[5]=='r') && (buf[16]=='0'))
{
espera(3000); Enviar_Comando8();
}

```

```

    }
    if ((buf[0]=='I') && (buf[3]=='0'))
    {
        espera(2000);Enviar_Coma8();
    }
    if (B==2170)
    {
        verifica_trama();
        Numero_Paquete++;
        if(Numero_Paquete==6)
        {
            Numero_Paquete=0;
            espera(10000);Enviar_Coma11();
        }
        else
        {
            espera(3000);Enviar_Coma10();
        }
    }
    if(buf[15]==':')
    {
        buf[15]=x;
        espera(2000);Enviar_Coma12();
    }
    if(buf[6]=='X')
    {
        espera(30000);Enviar_Coma10();
        buf[6]=x;
    }
    if ((buf[5]=='r') && (buf[21]=='0'))
    {
        espera(6000); Enviar_Coma11();
    }
    if ((buf[0]=='r') && (buf[1]=='e'))
    {
        espera(2000);muestra_bits();
    }
    analiza_trama=0;
}

}
// Cierre del main del
programa

/*****FIN DEL
MAIN*****/

/*****
RECEPCION DE UART2          INTERRUPCION DE
******/

void __attribute__((interrupt, no_auto_psv)) _U2RXInterrupt(void)

```

```

{
    IFS1bits.U2RXIF = 0;                //borra el flag de
interrupcion
}
/*****FIN DE
INTERRUPCION*****/

/*****                                INTERRUPTION DE
RECEPCION DE UART1      *****/

void __attribute__((interrupt, no_auto_psv)) _U1RXInterrupt(void)
{
    IFS0bits.U1RXIF = 0;                //borra el flag de
interrupcion
}
/*****FIN DE
INTERRUPCION*****/

/*****                                INTERRUPTION DEL
TRANSMISOR DEL UART2      *****/

void __attribute__((interrupt, no_auto_psv)) _U2TXInterrupt(void)
{
    IFS1bits.U2TXIF = 0;
}
/*****FIN DE
INTERRUPCION*****/

/*****                                INTERRUPTION DEL
TRANSMISOR DEL UART1      *****/

void __attribute__((interrupt, no_auto_psv)) _U1TXInterrupt(void)
{
    IFS0bits.U1TXIF = 0;
}
/*****FIN DE
INTERRUPCION*****/

/*****                                FUNCION: CUENTA LOS BITS 1 QUE
HAY EN EL BYTE      *****/

int contbit(unsigned char x)
{
    int count;
    for (count=0; x!=0; x>>=1)
        if ( x & 1)
            count++;
    return count;
}

/*****FIN DE CUENTA
BIT*****/

```

```

/*****
*FUNCION: CALCULA BIT*****/

int calcula_bits(void)
{
    int J=0;
    int BERR;

    for (r=26;r<=2143;r++)
    {
        if (trama[J]!= buf[r])
        {
            t=trama[J]^buf[r];           // aplico XOR entre
las dos tramas
            k=contbit(t);
            Bits_er=Bits_er+k;           // bits errados
            k=0;
            J++;
        }
        else
            J++;
        if (J==134)
            J=0;
    }

    BERR=Bits_er;
    Bits_er=0;

return BERR;

}
/*****FIN DE CALCULA
BER*****/

/*****          FUNCION: INICIALIZACION
DEL TIMER2          *****/

void ini_timer2(void)

{
T2CONbits.TON=0;           //disable timer
T2CONbits.TCS=0;           //select internal instruction
cycle clock
T2CONbits.TGATE=0;           //disable gated timer
mode
T2CONbits.TCKPS= 0b00;       //select1:1 prescaler
TMR2=0x00;                   //clear timer registrer
PR2=88000;                   //interrumpe cada 2.2mseg

IPC1bits.T2IP=7;           //set timer Interrupt priority
level

```

```

IFS0bits.T2IF=0;           //clear timer2 interrupt flag
IEC0bits.T2IE=1;         //enable timer2 interrupt

T2CONbits.TON=1;         //start timer
}

/*****FIN DE INICIALIZACION
DEL TIMER*****/

/*****          FUNCION: INICIALIZACION
DEL TIMER1          *****/

void ini_timer1(void)

{
T1CONbits.TON=0;         //disable timer
T1CONbits.TCS=0;         //select internal instruction
cycle clock
T1CONbits.TGATE=0;       //disable gated timer
mode
T1CONbits.TCKPS= 0b00;   //select1:1 prescaler
TMR1=0x00;               //clear timer registrer
PR1=45820;               //load the period value,
interrumpe cada 1.1458mseg

IPC0bits.T1IP=7;         //set timer Interrupt priority
level
IFS0bits.T1IF=0;         //clear timer2 interrupt flag
IEC0bits.T1IE=1;         //enable timer2 interrupt

T1CONbits.TON=1;         //start timer
}

/*****FIN DE INICIALIZACION
DEL TIMER*****/

/*****          INTERRUPCION DEL
TIMER2          *****/

void __attribute__((__interrupt__, auto_psv)) _T2Interrupt(void)
{
IFS0bits.T2IF = 0;       //Clear Timer1 interrupt flag

    if(delay)
    {
        delay--;
    }

    if (U2STAbits.OERR)   //verifica si hay overflow
        U2STAbits.OERR = 0; //borra la data recibida

    if (DataRdyUART2())

```

```

        {
        buf[indexRX]=ReadUART2 ();
        U1TXREG=buf[indexRX];
        indexRX++;
        count_byte++;
        flagT2=0;
        }
else
    {
    if(flagT2>=0)
        flagT2++;
    if(flagT2>4)
        {
        analiza_trama=1;
        B=count_byte;
        flagT2=-1;
        count_byte=0;
        indexRX=0;
        }
    }
}

/***** INTERRUPTION DEL
TIMER1 *****/

void __attribute__((__interrupt__, auto_psv)) _T1Interrupt(void)
{
IFS0bits.T1IF = 0; //Clear Timer1 interrupt flag

    if (U1STAbits.OERR) //verifica si hay overflow
        U1STAbits.OERR = 0; //borra la data recibida

    if (DataRdyUART1())
    {
    buf1[indexRX1]=ReadUART1 ();
    indexRX1++;
    count_bytel++;
    flagT1=0;
    }
else
    {
    if(flagT1>=0)
        flagT1++;
    if(flagT1>4)
        {
        analiza_trama=1;
        B1=count_bytel;
        flagT1=-1;
        count_bytel=0;
        indexRX1=0;
        }
    }
}

```

```

}

/***** FUNCION VERIFICA
TRAMA *****/

void verifica_trama(void)
{
    bits_errados = calcula_bits();
    bits_acumulados = bits_acumulados + bits_errados;
    indexRX = 0;
    count_byte=0;
    contador2500++;

if(contador2500 == 6)
    {
        contador2500=0;
        bits_por_millon = bits_acumulados;
        bits_acumulados=0;

        if(bits_por_millon >MAX_bits)
        {
            MAX_bits = bits_por_millon;
            bits_por_millon = 0;
            ber= MAX_bits;
            MSB=ber;
            LSB=ber>>8;

        }

    }

}

/***** FIN DE VERIFICA
TRAMA *****/

/*****MUESTRA BITS
ERRADOS*****/

void muestra_bits(void)
{
    U1TXREG='#';
    while (BusyUART1());
    U1TXREG=LSB;
    while (BusyUART1());
    U1TXREG=MSB;
    while (BusyUART1());

}

```

```

// funciones victor
#include <p33FJ64GP202.h>
#include <uart.h>
#include "funciones.h"

#define FCY 40000000 //40Mhz
#define BAUDRATE 4800
#define BRGVAL ((FCY/BAUDRATE)/16)-1

extern float contador;
extern int flagtrama;
extern char buf[4314];
extern char buf1[50];
extern int Indice;
extern int m;
extern char trama[134];
extern int B1,B;
extern int enviar;
extern long delay;

extern unsigned char comando0[11];
extern unsigned char comando1[27];
extern unsigned char comando2[9];

extern unsigned char comando3[21];
extern unsigned char comando4[23];
extern unsigned char comando5[21];

extern unsigned char comando6[22];
extern unsigned char comando7[13];
extern unsigned char comando8[13];
extern unsigned char comando9[20];
extern unsigned char comando10[14];
extern unsigned char comando[13];

extern unsigned char coma0[11];
extern unsigned char coma1[27];
extern unsigned char coma2[9];
extern unsigned char coma3[21];
extern unsigned char coma4[23];
extern unsigned char coma5[22];
extern unsigned char coma6[23];
extern unsigned char coma7[26];
extern unsigned char coma8[13];
extern unsigned char coma9[14];
extern unsigned char coma10[18];
extern unsigned char coma11[16];
extern unsigned char coma12[9];
extern unsigned char coma13[16];

```

```

/*****FUNCION DE CONFIGURACION DEL
RELOJ EXTERNO*****/

void Ini_clock (void)

{
    // Fosc= FIN*(M/(N1*N2)) =
    FIN*((PLLDIV+2)/((PLLPRE+2)*2*(PLLPOST+1)))
    // FIN = frecuencia de entrada al pin del cristal externo 10Mhz
    // Fosc= frecuencia de salida del PLL

    PLLFBD=30; // M=32
M=PLLDIV + 2
    CLKDIVbits.PLLPOST=0; // N1=2 N1= PLLPRE
+ 2
    CLKDIVbits.PLLPRE=0; // N2=2 N2=
2*(PLLPOST+1)

    __builtin_write_OSCCONH(0x03); // Initiate Clock Switch
to Primary // Oscillator

    with PLL (NOSC=0b011)
        __builtin_write_OSCCONL(0x01); // Start clock switching

    while (OSCCONbits.COSC != 0x03); // espera q ocurra el
cambio de reloj
    while (OSCCONbits.LOCK != 1){}; // espera por el
enganche del PLL
}

/*****FUNCION DE INICIALIZACION
DE UART2:*****/

void Ini_uart2(void)

{

    U2MODEbits.UARTEN= 0; //UART2
DESHABILITADO
    U2MODEbits.USIDL= 0; //CONTINUA
OPEREANDO EN MODO idle (STAND BY)
    U2MODEbits.IREN= 0; //IrDA
encoder and decoder disabled
    U2MODEbits.RTSMD= 0; //Seleccion de MODO
SIMPLE para U2RTS
    U2MODEbits.UEN=0; // TX Y RX
HABILITADO, CTS Y RTC DESHABILITADO

```

```

        U2MODEbits.WAKE=0;                // permitir
despertar al recibir un caracter
        U2MODEbits.LPBACK=0;            //MODO LOOPBACK
DESHABILITADO
        U2MODEbits.ABAUD=0;            //BAUD RATE
DESABILITADO
        U2MODEbits.URXINV=0;          //UART IdleState =
1
        U2MODEbits.BRGH= 0;            //generates
16 clocks per bit period
        U2MODEbits.PDSEL=0;            //8-bit data,
no parity *
        U2MODEbits.STSEL=0;            // One Stop
bit

        // Load a value into Baud Rate Generator.
        U2BRG = BRGVAL;                // 40Mhz osc, 9600 Baud

        // UxSTA: UARTx STATUS AND CONTROL REGISTER

        U2STAbits.UTXISEL1=0;          //Interrupt when a
character is transferred to the Transmit Shift Register.
        U2STAbits.UTXISEL0 = 0;
        U2STAbits.UTXINV=0;            //UxTX Idle
state is '1'
        U2STAbits.UTXBRK= 0;           //Sync Break
transmission disabled or completed
        U2STAbits.UTXEN=0;             //Transmit
disabled, any pending transmission is aborted and buffer is reset.
        U2STAbits.UTXBF= 0;           //Transmit
buffer is not full, at least one more character can be written
        U2STAbits.TRMT=0;             //Transmit
Shift Register is not empty, a transmission is in progress or queued
in the transmit buffer
        U2STAbits.URXISEL=0;          //Interrupt is set
when any character is received and transferred from the UxRSR to the
receive
        U2STAbits.ADDEN=0;            //Address
Detect mode disabled
        U2STAbits.RIDLE=0;            //Data is
being received
        U2STAbits.PERR= 0;           //Parity
error has not been detected
        U2STAbits.FERR=0;            //Framing
error has not been detected
        U2STAbits.OERR=0;            //Receive
buffer has not overflowed. Clearing a previously set OERR bit (1 ? 0
transition) resets
        U2STAbits.URXDA=0;           //Receive
buffer is empty

```

```

IPC7 = 0x4400; // Int.
Priority level: 4 for both TX & RX

IFS1bits.U2TXIF = 0; // Clear the
Transmit Interrupt Flag
IEC1bits.U2TXIE = 1; // Transmit
Interrupts disable, Interrupt request not enabled
IFS1bits.U2RXIF = 0; // Clear the
Recieve Interrupt Flag
IEC1bits.U2RXIE = 0; // Enable Recieve
Interrupts

U2MODEbits.UARTEN = 1; // Turn ON UART2

U2STAbits.UTXEN = 1; // Transmission
Enabled

}

/*****FIN
UART*****/

/*****FUNCION DE INICIALIZACION
DE UART2:*****/

void Ini_uart1(void)

{

    U1MODEbits.UARTEN= 0; //UART2
    DESHABILITADO
    U1MODEbits.USIDL= 0; //CONTINUA
    OPEREANDO EN MODO idle (STAND BY)
    U1MODEbits.IREN= 0; //IrDA
    encoder and decoder disabled
    U1MODEbits.RTSMD= 0; //Seleccion de MODO
    SIMPLE para U2RTS
    U1MODEbits.UEN=0; // TX Y RX
    HABILITADO, CTS Y RTC DESHABILITADO
    U1MODEbits.WAKE=0; // permitir
    despertar al recibir un caracter
    U1MODEbits.LPBACK=0; //MODO LOOPBACK
    DESHABILITADO
    U1MODEbits.ABAUD=0; //BAUD RATE
    DESABILITADO
    U1MODEbits.URXINV=0; //UART IdleState =
    1
    U1MODEbits.BRGH= 0; //generates
    16 clocks per bit period (16x baud clock, Standard mode)
    U1MODEbits.PDSEL=0; //8-bit data,
    no parity *

```

```

    U1MODEbits.STSEL=0; // One Stop
bit

    // Load a value into Baud Rate Generator.
    U1BRG = BRGVAL; // 40Mhz osc, 9600 Baud

    // UxSTA: UARTx STATUS AND CONTROL REGISTER

    U1STAbits.UTXISEL1=0; //Interrupt when a
character is transferred to the Transmit Shift Register.
    U1STAbits.UTXISEL0 = 0;
    U1STAbits.UTXINV=0; //UxTX Idle
state is '1'
    U1STAbits.UTXBRK= 0; //Sync Break
transmission disabled or completed
    U1STAbits.UTXEN=0; //Transmit
disabled, any pending transmission is aborted and buffer is reset.
    U1STAbits.UTXBF= 0; //Transmit
buffer is not full, at least one more character can be written
    U1STAbits.TRMT=0; //Transmit
Shift Register is not empty, a transmission is in progress or queued
in the transmit buffer
    U1STAbits.URXISEL=0; //Interrupt is set
when any character is received and transferred from the UxRSR to the
receive
    U1STAbits.ADDEN=0; //Address
Detect mode disabled
    U1STAbits.RIDLE=0; //Data is
being received
    U1STAbits.PERR= 0; //Parity
error has not been detected
    U1STAbits.FERR=0; //Framing
error has not been detected
    U1STAbits.OERR=0; //Receive
buffer has not overflowed. Clearing a previously set OERR bit (1 ? 0
transition) resets
    U1STAbits.URXDA=0; //Receive
buffer is empty

    IPC7 = 0x4400; // Int.
Priority level: 4 for both TX & RX

    IFS0bits.U1TXIF = 0; // Clear the
Transmit Interrupt Flag
    IEC0bits.U1TXIE = 1; // Transmit
Interrupts disable, Interrupt request not enabled
    IFS0bits.U1RXIF = 0; // Clear the
Recieve Interrupt Flag
    IEC0bits.U1RXIE = 0; // Enable Recieve
Interrupts

    U1MODEbits.UARTEN = 1; // Turn ON UART1

```

```

        U1STAbits.UTXEN = 1;                                // Transmission
Enabled
    }

/*****FIN
UART*****/

/*****INICIALIZACION DEL
BUFER*****/

void ini_buf(void)
{
    int i;
    for (i=0;i<=4314;i++)
    {
        buf[i]=0x0000;
    }
}

/*****INICIALIZACION DEL
BUFER1*****/

void ini_buf1(void)
{
    int i;
    for (i=0;i<=50;i++)
    {
        buf1[i]=0x0000;
    }
}

/*****FIN DEL LA
INICIALIZACION*****/

void Enviar_Comando0 (void)
{
    for (Indice=0;Indice<11;Indice++)
    {
        U2TXREG=comando0[Indice];
        while(BusyUART2());
    }
}

void Enviar_Comando1 (void)
{
    for (Indice=0;Indice<27;Indice++)
    {
        U2TXREG=comando1[Indice];

```

```

        while(BusyUART2());
    }
}
void Enviar_Comando2 (void)
{
    for (Indice=0;Indice<9;Indice++)
    {
        U2TXREG=comando2[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando3 (void)
{
    for (Indice=0;Indice<21;Indice++)
    {
        U2TXREG=comando3[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando4 (void)
{
    for (Indice=0;Indice<23;Indice++)
    {
        U2TXREG=comando4[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando5 (void)
{
    for (Indice=0;Indice<21;Indice++)
    {
        U2TXREG=comando5[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando6 (void)
{
    for (Indice=0;Indice<22;Indice++)
    {
        U2TXREG=comando6[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando7 (void)
{
    for (Indice=0;Indice<13;Indice++)
    {
        U2TXREG=comando7[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando8 (void)
{

```

```

        for (Indice=0;Indice<13;Indice++)
        {
            U2TXREG=comando8[Indice];
            while(BusyUART2());
        }
    }
void Enviar_Comando9 (void)
{
    for (Indice=0;Indice<20;Indice++)
    {
        U2TXREG=comando9[Indice];
        while(BusyUART2());
    }
}
void Enviar_Comando10 (void)
{
    for (Indice=0;Indice<14;Indice++)
    {
        U2TXREG=comando10[Indice];
        while(BusyUART2());
    }
}
void Enviar_Trama (void)
{
    for (Indice=0;Indice<134;Indice++)
    {
        U2TXREG=trama[Indice];
        while(BusyUART2());
    }

    contador++;

    if (contador==96)
    {
        espera (1000);Enviar_Comando10();

        for (Indice=0;Indice<13;Indice++)
        {
            U2TXREG=comando[Indice];
            while(BusyUART2());
        }

        flagtrama=0;
        contador=0;
    }
}
void Enviar_Coma0 (void)
{
    for (Indice=0;Indice<11;Indice++)
    {
        U2TXREG=coma0[Indice];
        while(BusyUART2());
    }
}

```

```

    }
}
void Enviar_Coma1 (void)
{
    for (Indice=0;Indice<27;Indice++)
    {
        U2TXREG=coma1[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma2 (void)
{
    for (Indice=0;Indice<9;Indice++)
    {
        U2TXREG=coma2[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma3 (void)
{
    for (Indice=0;Indice<21;Indice++)
    {
        U2TXREG=coma3[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma4 (void)
{
    for (Indice=0;Indice<23;Indice++)
    {
        U2TXREG=coma4[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma5 (void)
{
    for (Indice=0;Indice<21;Indice++)
    {
        U2TXREG=coma5[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma6 (void)
{
    for (Indice=0;Indice<22;Indice++)
    {
        U2TXREG=coma6[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma7 (void)
{
    for (Indice=0;Indice<26;Indice++)

```

```

        {
            U2TXREG=coma7[Indice];
            while(BusyUART2());
        }
    }
void Enviar_Coma8 (void)
{
    for (Indice=0;Indice<13;Indice++)
    {
        U2TXREG=coma8[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma9 (void)
{
    for (Indice=0;Indice<14;Indice++)
    {
        U2TXREG=coma9[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma10 (void)
{
    for (Indice=0;Indice<18;Indice++)
    {
        U2TXREG=coma10[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma11 (void)
{
    for (Indice=0;Indice<16;Indice++)
    {
        U2TXREG=coma11[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma12 (void)
{
    for (Indice=0;Indice<9;Indice++)
    {
        U2TXREG=coma12[Indice];
        while(BusyUART2());
    }
}
void Enviar_Coma13 (void)
{
    for (Indice=0;Indice<16;Indice++)
    {
        U2TXREG=coma13[Indice];
        while(BusyUART2());
    }
}

```

```

void Enviar_Comandos (void)
{
    for (m=0;m<B1;m++)
    {
        U2TXREG=buf1[m];
        while(BusyUART2());
    }
}
void Enviar_Comandos2 (void)
{
    int r;
    for (r=0;r<B;r++)
    {
        U1TXREG=buf[r];
        while(BusyUART2());
    }
}

/***** funcion de
espera*****/
void espera( long E)
{
    delay=E;
    while(delay);
}

```

```

#include <p33FJ64GP202.h>

void Ini_clock(void); /*DECLARACION DE LA
FUNCION DEL RELOJ EXTERNO*/
void Ini_uart2(void); /*DECLARACION DE LA
FUNCION INICIALIZACION DEL UART2*/
void Ini_uart1(void); /*DECLARACION DE LA
FUNCION INICIALIZACION DEL UART1*/
void ini_buf(void);
void ini_buf1(void);
void espera(long E);

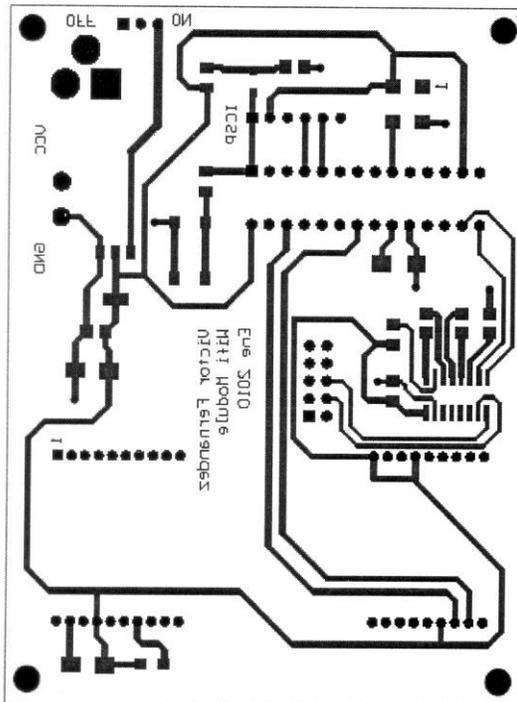
void Enviar_Coma0 (void);
void Enviar_Coma1 (void);
void Enviar_Coma2 (void);
void Enviar_Coma3 (void);
void Enviar_Coma4 (void);
void Enviar_Coma5 (void);
void Enviar_Coma6 (void);
void Enviar_Coma7 (void);
void Enviar_Coma8 (void);
void Enviar_Coma9 (void);
void Enviar_Coma10 (void);
void Enviar_Coma11 (void);
void Enviar_Coma12 (void);
void Enviar_Coma13 (void);

void Enviar_Comando0 (void);
void Enviar_Comando1 (void);
void Enviar_Comando2 (void);
void Enviar_Comando3 (void);
void Enviar_Comando4 (void);
void Enviar_Comando5 (void);
void Enviar_Comando6 (void);
void Enviar_Comando7 (void);
void Enviar_Comando8 (void);
void Enviar_Comando9 (void);
void Enviar_Comando10 (void);
void Enviar_Comando (void);
void Enviar_Trama (void);
void Enviar_Comandos (void);
void Enviar_Comandos2 (void);

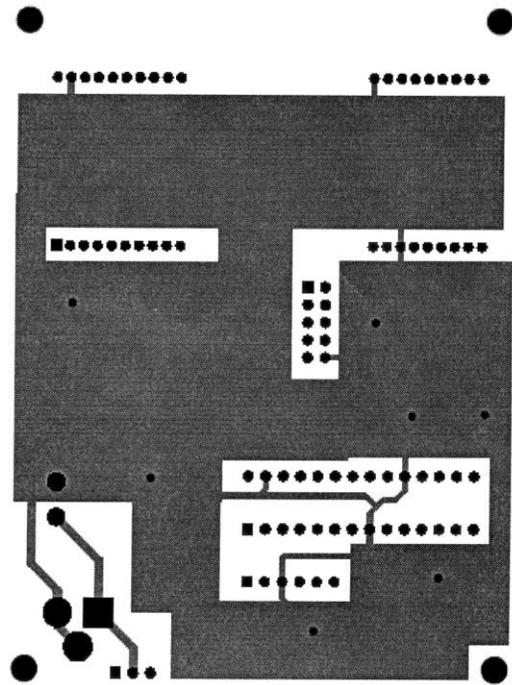
```

[ANEXO N° 8]

[Realización de diagrama de Circuito impreso]



Parte de arriba

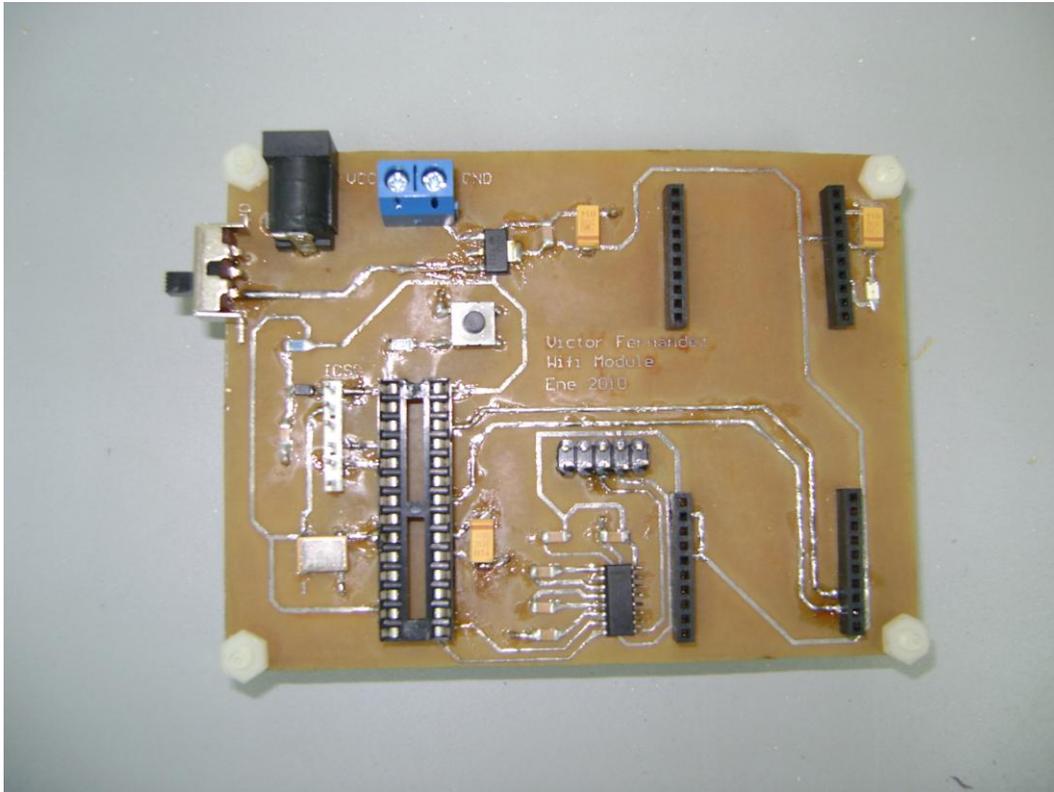


Parte de abajo

Para la realización del diagrama del circuito impreso se utilizó el programa Altium Designer. Luego de realizar el circuito se procedió a su transferencia a la baquelita.

[ANEXO N° 9]

[Conexión de componentes en baquelita]



Una vez transferido el diagrama del circuito impreso a la baquelita, se soldaron superficialmente los componentes electrónicos.

[ANEXO N° 10]

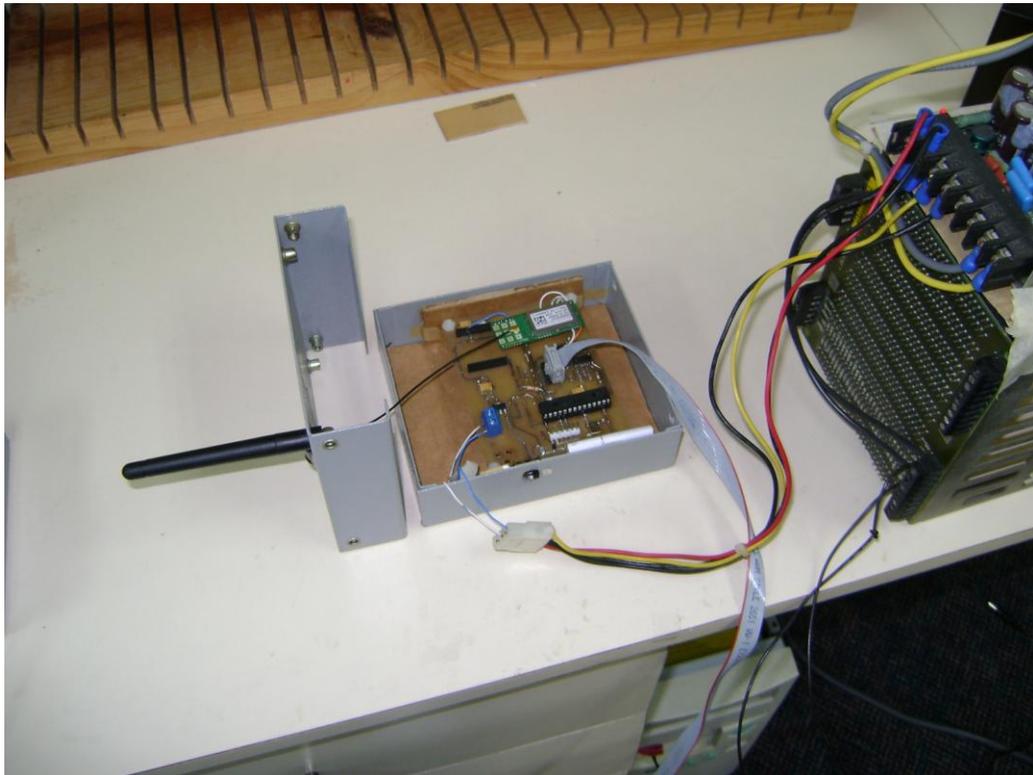
[Montaje de módulo en baquelita]



Montaje del módulo Wi-Fi, luego de haber soldado los componentes a la baquelita y conexión de la antena con el módulo.

[ANEXO N° 11]

[Colocación del circuito en caja]



Se adecuó una caja metálica provista por la empresa, para la colocación del circuito, a fin de facilitar su manejo.

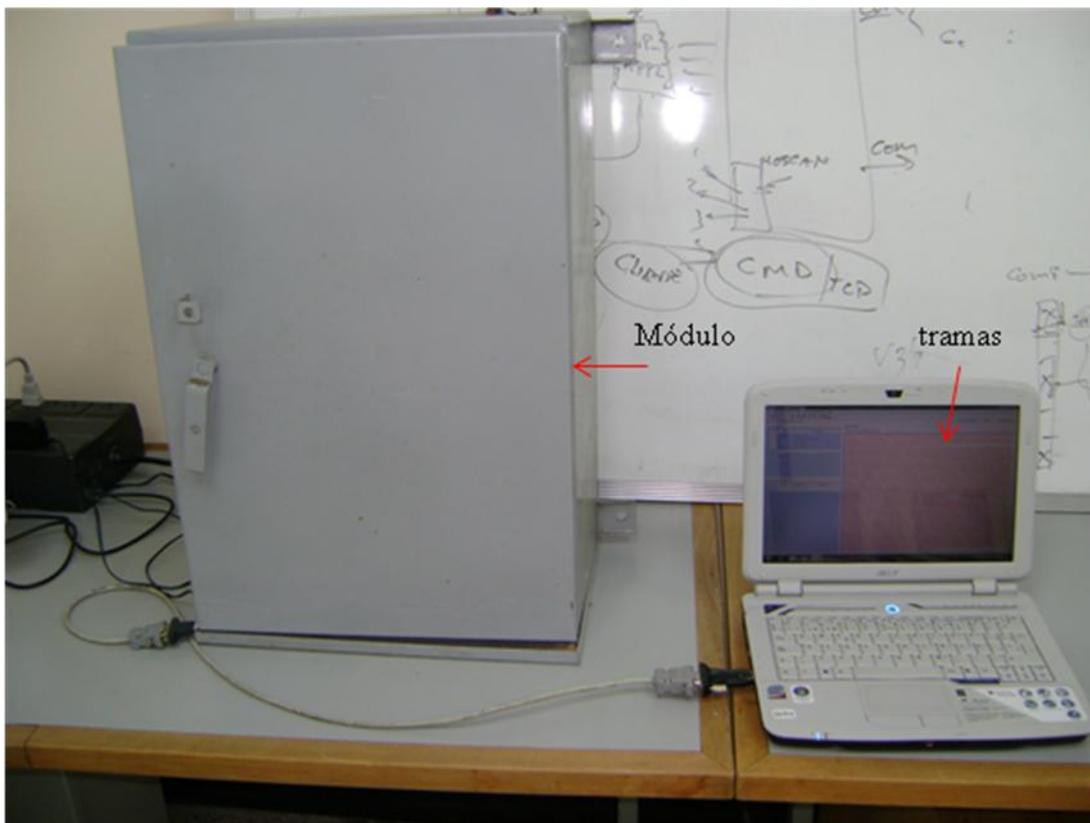
[ANEXO N° 12]

[Pruebas en sede de Corporación 80-60,1]



[ANEXO N° 13]

[Pruebas en sede de Corporación 80-60,2]



En la laptop se puede observar las tramas recibidas por el módulo esclavo, el cual, está ubicado adentro del cajetín metalito, sin embargo este cajetín no es totalmente cerrado y poseía un orificio circular de 5 cm en la parte superior.

[ANEXO N° 14]

[Subestación de media tensión A]



Ubicación de equipos en la subestación.

[ANEXO 15]

[Prueba en Subestación de media tensión A]



Ubicación de módulos en caja de control de Transformador.

[ANEXO N° 16]

[Subestación de media tensión, B]



En esta subestación las conexiones hacia los transformadores de corriente y tensión son aéreas.

[ANEXO N° 17]

[Prueba en Subestación de media tensión B]



Colocación de Esclavo y Maestro en la ubicación de UPA Y UTR respectivamente.

[ANEXO N° 18]

[Subestación de Media Tensión C]



Ubicación de celdas, UPA y transformadores.

[ANEXO N° 19]

[Prueba en Subestación de Media Tensión C]



Se colocaron los módulos Maestro y esclavo, donde estan ubicados UPA y UTR respectivamente, se observó que al tener una abertura la celda, se comunican los módulos.

[ANEXO N° 20]

[Subestación de media Tensión D]



Ubicación de equipos en la subestación.

[ANEXO 21]

[Prueba en Subestación de media tensión D, (1)]



[ANEXO N° 22]

[Pruebas en Transformador 1 de la subestación D]



Colocación del módulo esclavo al lado de la caja de control del transformador para verificar posibles alteraciones en la transmisión generada por el campo magnético del transformador.