

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRONICA



TRABAJO DE GRADUACION

“IMPLEMENTACION Y PROGRAMACION DE UNA TARJETA INALAMBRICA BLUETOOTH”

INTEGRANTES: ALEXANDER ERNESTO PORTILLO CHACÓN
 SERGIO JOSE CASTRO FIGUEROA

DIRECTOR: ING. OSCAR DURAN VIZCARRA

ASESOR: ING. JUAN CARLOS CASTRO

SOYAPANGO 9 DE ENERO DEL 2006

INDICE

INTRODUCCIÓN	1
1.0 DESCRIPCIÓN GENERAL DE LA TECNOLOGÍA INALÁMBRICA BLUETOOTH	2
1.1 RADIO FRECUENCIA (RF)	4
1.1.1 Banda de frecuencias y arreglo de canales	4
1.1.2 Características de transmisión	4
1.1.3 FHSS: Salto en Frecuencia de Espectro Disperso	5
1.2 BANDA BASE	8
1.2.1 Descripción general	8
1.2.2 Canal físico	9
1.2.3 Enlace físico	9
1.2.4 Paquetes	10
1.2.4.1 Código de acceso	10
1.2.4.2 Cabecera de paquete	11
1.2.4.3 Carga útil	12
1.2.5 Corrección de errores	13
1.2.6 Transmisión/Recepción	13
1.2.7 Control de Canal	14
1.2.8 Seguridad en Bluetooth.	17
1.3 PROTOCOLO DE GESTIÓN DE ENLACE (LMP)	18
1.3.1 Establecimiento de conexión	19
1.4 INTERFAZ DEL CONTROLADOR DE HOST (HCI)	20
1.4.1 Capas mas bajas del “stack Bluetooth”	20
1.4.2 Posibles arquitecturas de bus físico	21
1.5 PROTOCOLO DE CONTROL Y ADAPTACIÓN DE ENLACE LÓGICO (L2CAP)	23
1.5.1 Canales	23
1.5.2 Operaciones entre Capas	23
1.5.3 Segmentación y Reensamblado	24
1.5.4 Eventos	25
1.5.5 Acciones	25
1.5.6 Formato del paquete de datos	25
1.5.7 Calidad de servicio (QoS)	26
1.6 PROTOCOLO DE DESCUBRIMIENTO DE SERVICIO (SDP)	27
1.6.1 Descripción General	27
1.6.2 Registros de servicio	27
1.6.3 El protocolo	27

1.7 RFCOMM	29
1.8 PERFILES BLUETOOTH	30
1.8.1 Perfil Genérico de Acceso (GAP)	31
1.8.2 Perfil de Puerto Serial	32
1.8.3 Perfil de Aplicación de Descubrimiento de Servicio (SDAP)	32
1.8.4 Perfil Genérico de Intercambio de Objetos (GOEP)	32
1.8.5 Perfil de Telefonía Inalámbrica	32
1.8.6 Perfil de Intercomunicador	33
1.8.7 Perfil de Manos Libres	33
1.8.8 Perfil “Dial-up Networking”	33
1.8.9 Perfil de Fax	33
1.8.10 Perfil de Acceso LAN	33
1.8.11 Perfil Object Push	34
1.8.12 Perfil de Transferencia de Archivos	34
1.8.13 Perfil de Sincronización	34
2.0 DESCRIPCIÓN DE HARDWARE Y PRODUCTOS BLUETOOTH	35
2.1 TARJETA BLUETOOTH	35
2.1.1 Descripción de la tarjeta Bluetooth	37
2.2 DESCRIPCIÓN DE COMPONENTES Y REQUERIMIENTOS EN LA TARJETA BLUETOOTH	44
2.2.1 Módulo Bluetooth EZURIO BISM2	44
2.2.2. Regulador de Voltaje “National LP2987”	48
2.2.3 MAXIM MAX3232	48
2.2.4 Adaptador Board to Board de Ezurio para el BISM2	49
2.3 PRODUCTOS BLUETOOTH	50
2.4 DISPOSITIVO BLUETOOTH USB	51
2.5 PROCEDIMIENTO DE CALIFICACION BLUETOOTH	52
3.0 SOFTWARE PARA EL HOST BLUETOOTH	53
3.1 ARQUITECTURAS DE INTEGRACIÓN DEL “STACK” DE PROTOCOLOS BLUETOOTH	56
3.2 PRINCIPALES SOFTWARE PARA EL HOST	58
3.2.1 Software Bluetooth con propiedad de licencia	58
3.2.2 Software Bluetooth con licencia pública (GPL)	61

4.0 CRITERIOS DE SELECCIÓN	63
4.1 DEFINIR CRITERIOS DE SELECCIÓN	63
4.2 SELECCIONAR ELEMENTOS A UTILIZAR	63
5.0 BLUETOOTH TESIS 2005 (BT2005)	64
5.1 INTRODUCCION	64
5.2 INSTALACION DEL SOFTWARE	64
5.3 DESCRIPCION GENERAL DE LA INTERFAZ GRAFICA DE USUARIO DEL SOFTWARE BLUETOOTH TESIS 2005	68
5.4 PASOS A REALIZAR PARA UNA CONEXIÓN FTP	70
5.5 PASOS A REALIZAR PARA TRANSFERIR UN ARCHIVO AL SERVIDOR	74
5.6 PASOS PARA TRANSFERIR UN ARCHIVO DESDE SERVIDOR HACIA LA PC	76
6.0 PRUEBAS DE DESEMPEÑO Y APLICACIÓN PARA LA TARJETA BLUETOOTH	78
6.1 PRUEBAS DE DESEMPEÑO EN TRANSFERENCIA DE ARCHIVO SIN OBSTACULO	78
6.2 PRUEBAS DE DESEMPEÑO EN TRANSFERENCIA DE ARCHIVO CON OBSTACULO	79
7.0 CONCLUSIONES	81
8.0 BIBLIOGRAFIA	82

INTRODUCCION

En este tiempo la modernización juega un papel importante en el mundo. Por esta razón surgen nuevas y mejores tecnologías enfocadas cada una a un campo de acción específico. La movilidad es uno de esos campos que incluye: teléfonos móviles (acceso a WAN) “Wide Area Network” (Red de Área Amplia), WLAN “Wireless Local Area Network” (Red de Área Local Inalámbrica) IEEE 802.11 (acceso a LAN) “Local Area Network” (Red de Área Local), etc.

Mediante este proyecto se enfoca una de las tecnologías inalámbricas más revolucionarias para este campo. Revolucionaria por su Accesibilidad, Movilidad y Tamaño, esta es la tecnología Bluetooth, la cual forma parte de una completa solución a los problemas de movilidad. En esta época de transición tecnológica, en la que El Salvador ha modernizado su infraestructura de comunicaciones y masificado poco a poco el acceso a la misma. Casos como el de la telefonía móvil de segunda y tercera generación (GSM “Global System for Mobile Communications” (Sistema Global para Comunicaciones Móviles)), implican más y mejores servicios (transmisión de audio y video con buena definición), que promueven e incentivan el uso de tecnologías como Bluetooth.

Esta Tesis, cimienta las bases para conocer y aprender la tecnología inalámbrica Bluetooth en la Universidad Don Bosco, implementar hardware Bluetooth y realizar nuevas aplicaciones dirigidas a desarrollar otra forma de comunicación inalámbrica.

1. DESCRIPCIÓN GENERAL DE LA TECNOLOGÍA INALÁMBRICA BLUETOOTH

En 1994, Ericsson Mobile Telecommunications comenzó un estudio para investigar la viabilidad de un interfaz de radio de baja potencia y bajo costo entre teléfonos móviles y sus accesorios. El estudio era parte de un proyecto más amplio que investigaba cómo diferentes dispositivos de comunicaciones se podrían conectar a la red celular a través del teléfono móvil. A medida que progresaba el proyecto se hizo evidente que las aplicaciones de un enlace de radio de corto alcance eran ilimitadas.

El trabajo de Ericsson en esta área atrajo la atención de compañías como IBM, Intel, Nokia y Toshiba que formaron el SIG (“Special Interest Group” Grupo de Interés Especial) Bluetooth en 1998 y desarrollaron el primer estándar de dicha tecnología. [1] Dicha norma recoge requerimientos hardware, software y de interoperabilidad, definiendo un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia.

Los principales objetivos que se pretende conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

La tecnología inalámbrica Bluetooth utiliza la banda de radio ISM (industrial, scientific, medical o industrial, científica, médica), mundialmente disponible, de 2,4 GHz.

Las bandas ISM (“Industrial Scientific Medical” Industrial, Científica y Médica) incluyen los rangos de frecuencia entre 902-928 MHz y 2,4-2.484 GHz que no requieren una licencia de operador otorgada por las autoridades reguladoras de telecomunicaciones. [2]

El uso de una banda común de frecuencias significa que se pueden emplear dispositivos que utilicen la especificación Bluetooth en cualquier parte del mundo, y serán capaces de enlazar con otros dispositivos similares, independientemente del país en que se encuentren.

La conexión entre dispositivos Bluetooth sigue un esquema maestro-esclavo, soportando hasta ocho dispositivos conectados simultáneamente en lo que es la estructura básica de comunicación, denominada piconet. También es posible que un mismo dispositivo participe de más de una piconet aunque no puede estar activo en más de una simultáneamente; se denomina red dispersa o scatternet.

Si un equipo se encuentra dentro del radio de cobertura de otro, éstos pueden establecer conexión entre ellos. En principio sólo son necesarias un par de unidades con las mismas características de hardware para establecer un enlace. Dos o más unidades Bluetooth que comparten un mismo canal forman una piconet. [3]

Como se puede observar en la Figura 1.1, la comunicación sobre Bluetooth se divide en varias capas, estas se tratarán en mayor detalle en capítulos posteriores.

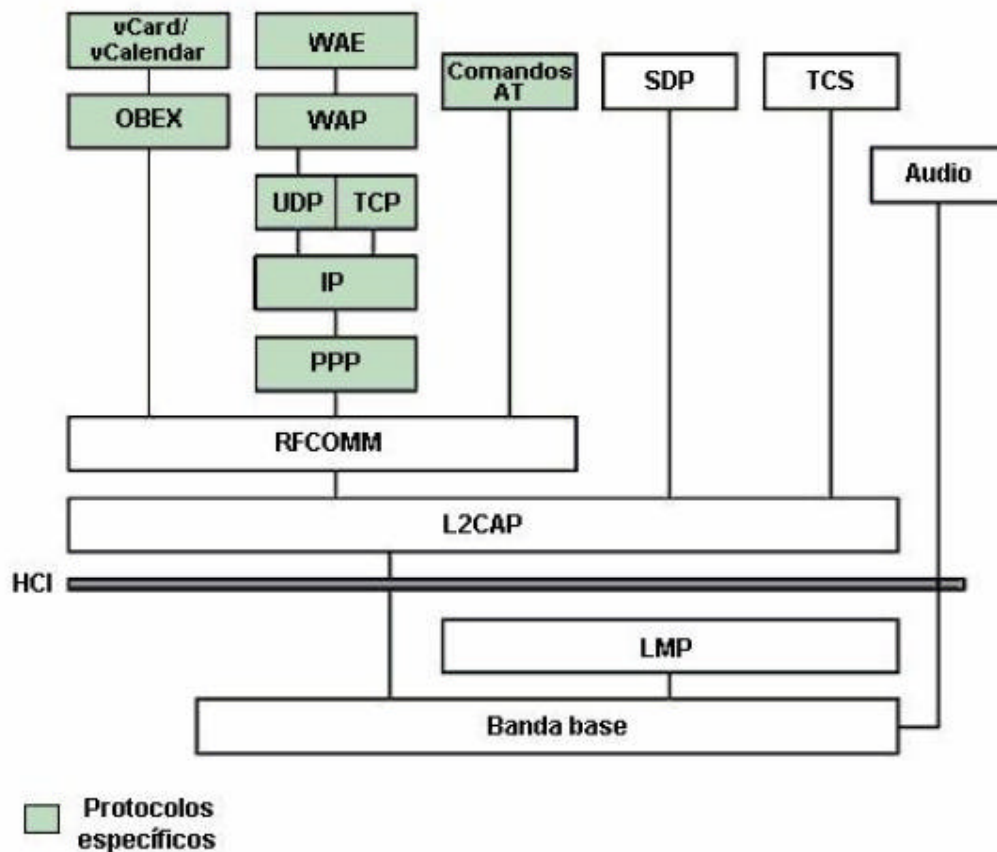


FIGURA 1.1
Stack de Protocolo Bluetooth

1.1 RADIO FRECUENCIA (RF)

1.1.1 Banda de frecuencias y arreglo de canales

Bluetooth opera entre la banda 2400- 2483.5 MHz y los canales RF están regidos según lo siguiente:

$$F=2402+k \text{ MHz, } k=0,1,\dots,78 \quad [4] \quad \text{Ec.1}$$

Estos canales son de 1MHz cada uno, además existen unos espacios de banda que son guardados o reservados. El inferior que es de 2MHz y el superior que es de 3.5MHz, con el fin de cumplir con el total de banda regulada.

Existen casos especiales del uso de esta banda en algunos países debido a las limitaciones de ancho de banda que estableen las leyes de cada país. Por ejemplo Francia en donde el rango de frecuencias es 2446.5-2483.5 y los canales correspondientes son:

$$F=2454+k \text{ MHz, } k=0,1,\dots,22 \quad [4] \quad \text{Ec.2}$$

1.1.2 Características de transmisión

En tanto a los requerimientos de los niveles de potencia en las antenas de los equipos Bluetooth, cuando el equipo no tiene este conector se asume una antena de 0dBi, en caso que la antena existente sea mayor a 0dBi deberá ser compensada.

Según los niveles de potencia los equipos son clasificados en tres niveles:

Potencia Clase	Máxima Salida de Potencia	Potencia Nominal de Salida	Potencia Mínima de Salida
1	100mW (20dBi)	N/A	1mW (0dBi)
2	2.5mW (4dBi)	1mW (0dBi)	0.25mW (0dBi)
3	1mW (0dBi)	N/A	N/A

Los niveles debajo de 0dBi son opcionales y se pueden usar para optimizar el consumo de energía y disminuir el nivel de interferencia.

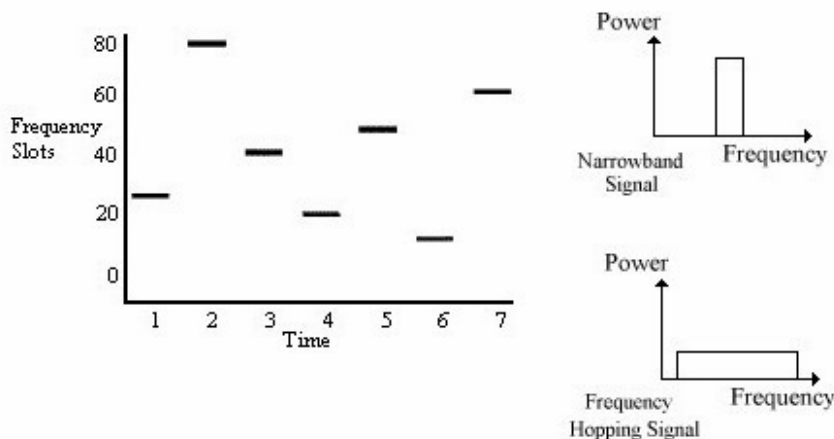
1.1.3 FHSS: Salto en Frecuencia de Espectro Disperso (“Frequency Hopping Spread Spectrum”)

Esta técnica se puede describir así:

- FHSS de banda estrecha consiste en que una trama de bits se envía ocupando ranuras específicas de tiempo (TS “Time Slots”) en diversos canales de radio-frecuencia.
- FHSS de banda ancha consiste en que durante el intervalo de 1 bit se conmutan diversos canales de radio-frecuencia.

Al igual que Ethernet los datos son divididos en paquetes de información, solo que estos paquetes son enviados a través de varias frecuencias, esto es conocido como "Hopping Pattern", la intención de enviar la información por varias frecuencias es cuestión de seguridad, ya que si la información fuera enviada por una sola frecuencia sería muy fácil interceptarla.

Además, para llevar acabo la transmisión de datos es necesario que tanto el aparato que envía como el que recibe información coordinen este denominado "Hopping Pattern" (fig. 1.2A). El estándar IEEE 802.11 utiliza FHSS, aunque hoy en día la tecnología que sobresale utilizando FHSS es Bluetooth [5], la figura 1.2B muestra un diagrama de bloques de una transmisión y recepción utilizando FHSS.



Fuente: http://www.wacal.com/Network/C_network3.html

FIGURA 1.2A
Patrón de saltos en espectro disperso

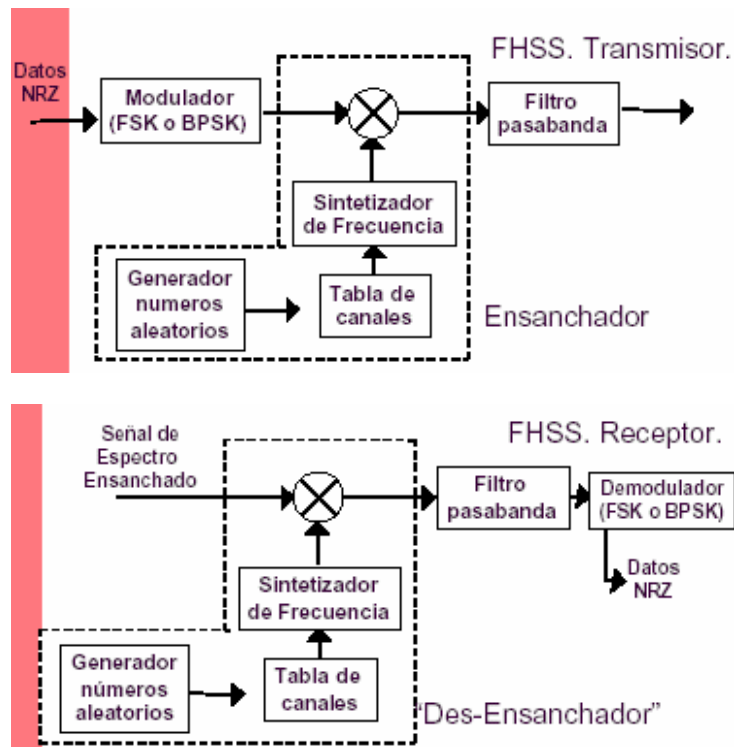


FIGURA 1.2B

Diagrama de Bloques de un Transmisor/ Receptor FHSS

Bluetooth hace los saltos de frecuencia utilizando el estándar FHSS, capaz de transmitir a velocidades de 1 Mbps y este estándar es apoyado por más de 2000 empresas de tecnología.

Bluetooth ha surgido últimamente como un posible sustituto a todo tipo de cable anexo a una computadora, debido a su costo y el apoyo de cientos de empresas. A su velocidad (1 Mbps) será capaz de sustituir las conexiones clásicas de cables paralelos y seriales, ya que es 3 y 6 veces más rápido (respectivamente) que estas conexiones en amplio uso en cualquier computadora.

Esto trae una cantidad interminable de posibilidades desde impresoras, monitores, conexiones de portátiles (“Laptops”), teclados, mouses .etc. Esta tecnología es capaz de

transmitir información efectivamente hasta una distancia de 10 metros entre aparatos que utilicen transmisores "Bluetooth", debido que se emplea FHSS el "Hopping Pattern" de Bluetooth es de 1600 veces por segundo, lo cual asegura que la transmisión de datos sea altamente segura.

En cuanto a su implementación Bluetooth utiliza el término piconet. Una piconet es un grupo de 2 u 8 aparatos que utilizan "Bluetooth", estos aparatos que forman parte del piconet comparten el mismo rango que es utilizado por un "Hopping Sequence", a su vez cada piconet contiene un aparato principal ("master") que es el encargado de coordinar el "Hopping Pattern" del piconet para que los demás aparatos ("slaves") sean capaces de recibir información.

1.2 BANDA BASE

1.2.1 Descripción general

Bluetooth soporta un canal de datos asíncrono y hasta tres canales de voz simultáneos (Síncronos) en los cuales los paquetes nunca son retransmitidos. El canal asíncrono soporta comunicación simétrica y asimétrica. En la comunicación asimétrica pueden ser enviados 723.3 kb/s desde el servidor y 57.6 kb/s hacia el servidor, mientras que en la comunicación simétrica pueden ser enviados 433 kb/s en ambas direcciones.

Bluetooth brinda conexión punto-a-punto o conexión punto-a-multipunto. Dos o más unidades compartiendo el mismo canal forman una piconet. Cada piconet debe tener un maestro y puede tener hasta siete esclavos activos, además pueden haber muchos más esclavos en estado “parked”. Estos esclavos no están activos en el canal sin embargo están sincronizados con el maestro con el fin de asegurar una rápida iniciación de comunicación. La interconexión de varias piconets forma una scatternet que es una red formada por varias piconet. En la Figura 1.3 se puede observar una piconet donde el PC actúa como maestro y los otros dispositivos son conectados como esclavos. [6]

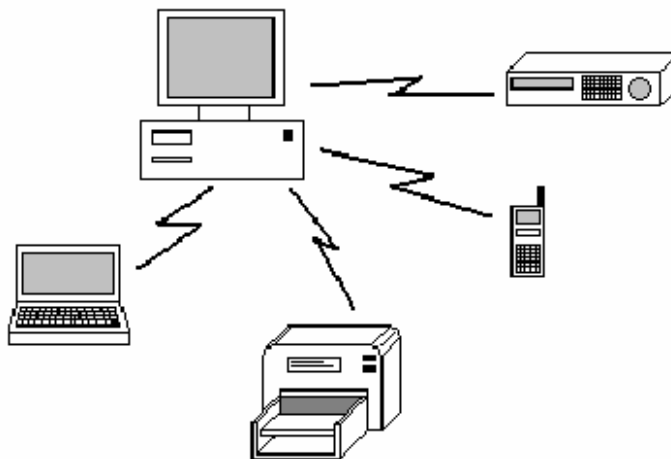


FIGURA 1.3
Diagrama de una piconet.

1.2.2 Canal físico

El canal físico contiene 79 frecuencias de radio diferentes, las cuales son accedidas de acuerdo a una secuencia de saltos aleatoria (FHSS). La razón de saltos estándar es de 1600 saltos/s. El canal está dividido en “Timeslots” (ranuras de tiempo), cada “slot” (ranura) corresponde a una frecuencia de salto y tiene una longitud de 625 μ s. Cada secuencia de salto en una piconet está determinada por la dirección del maestro de la piconet. Todos los dispositivos conectados a la piconet están sincronizados con el canal en salto y tiempo.

En una transmisión, cada paquete debe estar alineado con el inicio de un “slot” y puede tener una duración de hasta cinco “timeslots”. Durante la transmisión de un paquete la frecuencia es fija. Para evitar fallas en la transmisión, el maestro inicia enviando en los “timeslots” pares y los esclavos en los “timeslots” impares. En la Figura 1.4 se puede observar este esquema de transmisión. [7]

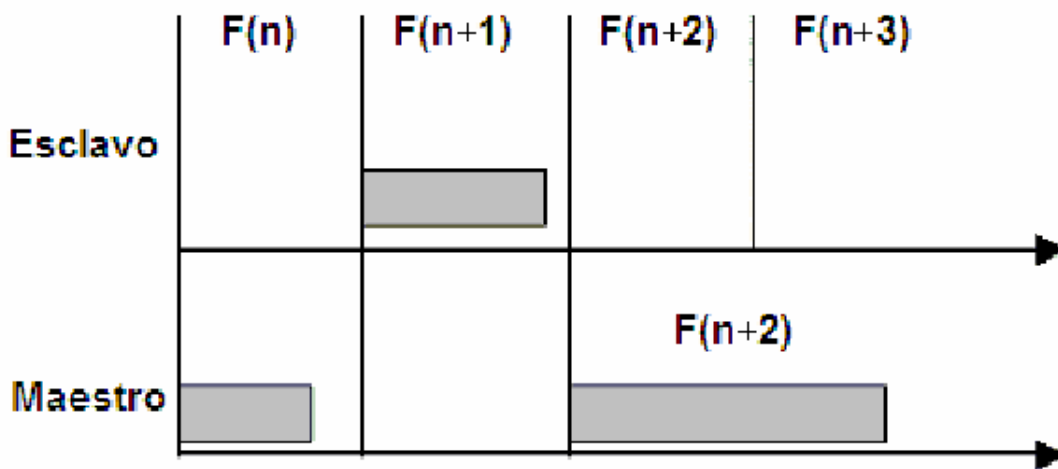


FIGURA 1.4
Transmisión en una *piconet*.

1.2.3 Enlace físico.

La comunicación sobre Bluetooth es perfecta para enlaces SCO “Synchronous Connection Oriented” Síncrono orientado a la conexión o enlaces ACL “Asynchronous Connectionless” Asíncrono no orientado a la conexión. El enlace SCO es una conexión simétrica punto-a-punto entre el maestro y un esclavo específico. Para lograr la

comunicación, el enlace SCO reserva “slots” en intervalos regulares en la iniciación, por esto el enlace puede ser considerado como una conexión de conmutación de circuitos. El enlace ACL es un enlace punto-a-multipunto entre el maestro y uno o más esclavos activos en la “piconet”. Este enlace de comunicación es un tipo de conexión de conmutación de paquetes. Todos los paquetes son retransmitidos para asegurar la integridad de los datos. El maestro puede enviar mensajes “broadcast” (de difusión) a todos los esclavos conectados dejando vacía la dirección del paquete, así todos los esclavos leerán el paquete.

1.2.4 Paquetes

Los datos enviados sobre el canal de la “piconet” son convertidos en paquetes, éstos son enviados y el receptor los recibe iniciando por el “bit” menos significativo. Como se observa en la Figura 1.5, el formato de paquete general consta de tres campos: código de acceso, cabecera y carga útil. [8]

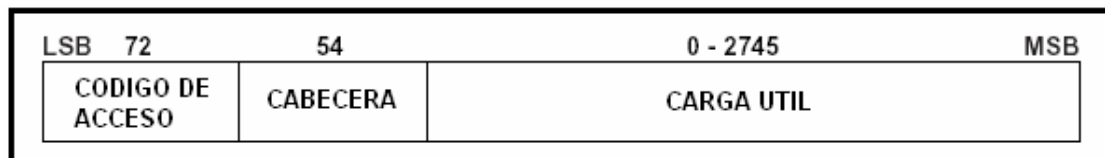


FIGURA 1.5
Formato de paquete general

1.2.4.1 Código de acceso

Es usado para sincronización e identificación. Todos los paquetes comunes que son enviados sobre el canal de la “piconet” están precedidos del mismo código de acceso al canal. Existen tres tipos diferentes de código de acceso:

- ✓ **Código de acceso al canal** - Para identificar los paquetes sobre el canal de la “piconet”.
- ✓ **Código de acceso de dispositivo** – Para procedimientos de señalización especiales, “paging” (servicio para transferencia de señalización o información en un sentido), entre otros.

- ✓ **Código de Acceso de Búsqueda (IAC)** – llamado IAC general cuando se quiere descubrir a otras unidades Bluetooth dentro del rango, o IAC dedicado cuando se desea descubrir unidades de un tipo específico.

El código de acceso consiste del *preámbulo*, *palabra de sincronización* y *“Trailer”* que es el que indica si lo siguiente es la cabecera del paquete o solo sincronización, la figura 1.6 muestra el formato del código de acceso.

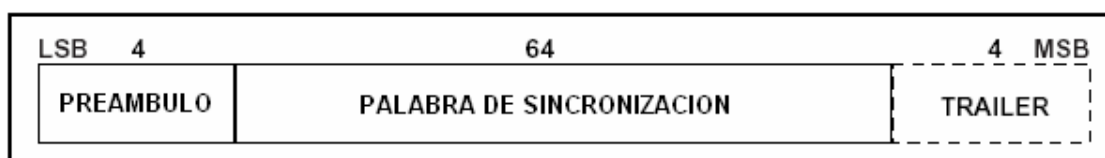


FIGURA 1.6
Formato del Código de Acceso

1.2.4.2 Cabecera de paquete

Como se observa en la Figura 1.7, la cabecera de paquete consta de seis campos:

- ✓ **Dirección** – Una dirección de dispositivo para distinguirlo de los demás dispositivos activos en la piconet.
- ✓ **Tipo** – Define qué tipo de paquete es enviado.
- ✓ **Flujo** – El “bit” de control de flujo es usado para notificar al emisor cuándo el “buffer” o almacenador temporal del receptor está lleno.
- ✓ **ARQN** – (“Acknowledge Receive Data”) reconocimiento de datos recibidos.
- ✓ **SEQN** – (“Sequential Numbering”) o numeración secuencial para ordenar los datos sobre el canal.
- ✓ **HEC** – (“Header Error Check”) Chequeo de redundancia cíclica de cabecera.

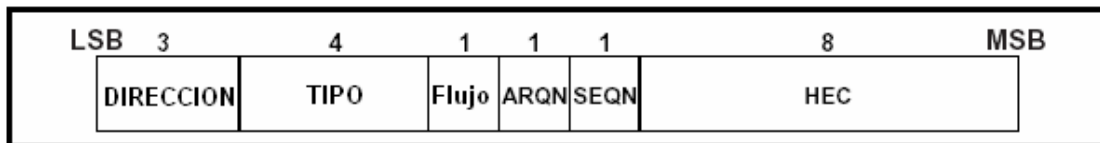


FIGURA 1.7
Formato de cabecera de paquete

1.2.4.3 Carga útil

La carga útil de un paquete puede ser dividida en dos tipos de campos:

- ✓ **Campo de Voz** – Consta de datos de voz de longitud fija y existe en paquetes de alta calidad de voz y paquetes combinados de datos-voz. No es necesaria ninguna cabecera de carga útil.
- ✓ **Campo de Datos** – Consta de tres partes, cabecera de carga útil, datos de carga útil, y código *CRC*.

En el caso del campo de datos a la carga útil se le agrega un encabezado que puede ser de 1 byte o de 2 byte como muestra la figura 1.8 y 1.9 respectivamente.

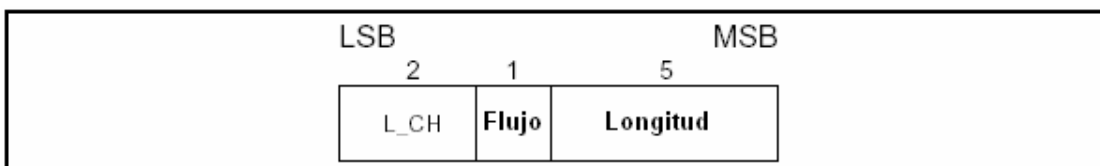


FIGURA 1.8
Formato de cabecera de carga útil para paquetes de un “slot”

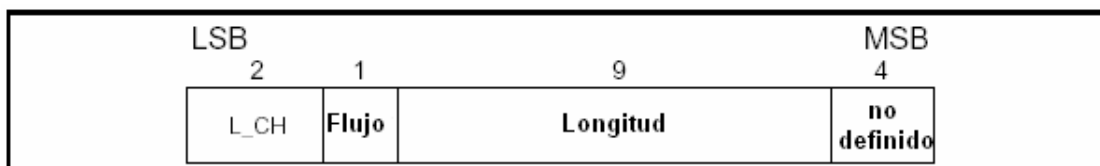


FIGURA 1.9
Formato de cabecera de carga útil para paquetes de más de un “slot”

1.2.5 Corrección de errores

En una comunicación *Bluetooth* existen varios esquemas diferentes de corrección de errores:

- ✓ En la cabecera, cada *bit* es repetido tres veces.
- ✓ En la carga útil se usa un esquema de “*código Hamming*”. Los *bits* de información son agrupados en secuencias de 10 *bits*, éstos son enviados como 15 *bits* y el algoritmo corrige todos los errores de un *bit* y detecta los errores de dos *bits*.
- ✓ Para garantizar una recepción correcta, todos los paquetes de datos son retransmitidos hasta que el emisor reciba una confirmación. La confirmación es enviada en la cabecera de los paquetes retornados.
- ✓ Los paquetes “**broadcast**” son paquetes transmitidos desde el maestro a todos los esclavos. No hay posibilidad de usar confirmación para esta comunicación, sin embargo, para incrementar la posibilidad de recibir correctamente un paquete, cada *bit* en el paquete es repetido un número fijo de veces.
- ✓ El chequeo de redundancia cíclica (*CRC*) se usa para detectar errores en la cabecera. La suma de comprobación *CRC* está contenida en el campo *HEC* de la cabecera de paquete. Los chequeos de redundancia cíclica también se aplican sobre la carga útil en la mayoría de los paquetes.
- ✓ Para asegurar que no desaparezcan paquetes completos, *Bluetooth* usa números de secuencia. Actualmente sólo se usa un número de secuencia de un *bit*. [8]

1.2.6 Transmisión/Recepción

Como se mencionó antes, el maestro de la “*piconet*” empieza enviando en “*timeslots*” pares y el esclavo en los impares.

Solamente el último esclavo direccionado está autorizado para enviar en el “*timeslot*” de los esclavos. Esto no causa problemas ya que el maestro siempre está inicializando todas

las conexiones y transmisiones nuevas. Cada esclavo espera las oportunidades de conexión dadas por el maestro. Los paquetes pueden ser más grandes que un "timeslot", debido a esto el maestro puede continuar enviando en los "timeslots" impares y viceversa. El sistema de reloj del maestro sincroniza a toda la "piconet". El maestro nunca ajusta su sistema de reloj durante la existencia de una "piconet", son los esclavos quienes adaptan sus relojes con un "offset" o compensación de tiempo con el fin de igualarse con el reloj del maestro. Este "offset" es actualizado cada vez que es recibido un paquete desde el maestro.

1.2.7 Control de Canal.

El control de canal describe cómo se establece el canal de una "piconet" y cómo las unidades pueden ser adicionadas o liberadas en la "piconet". La dirección del maestro determina la secuencia de saltos y el código de acceso al canal. La fase de la "piconet" está determinada por el sistema de reloj del maestro. Por definición, la unidad *Bluetooth* que inicia la conexión representa al maestro.

En *Bluetooth*, la capa de control de enlace se divide en dos estados principales: "standby" y *conexión*. Además existen siete sub-estados: "page", "page scan", "inquiry" (búsqueda), "inquiry scan", respuesta de maestro, respuesta de esclavo y respuesta a "inquiry". Los sub-estados son usados para agregar nuevos esclavos a una "piconet". Para moverse de un estado a otro se usan comandos de capas más altas o señales internas.

En *Bluetooth* se define un procedimiento de búsqueda que se usa en aplicaciones donde la dirección del dispositivo de destino es desconocida para la fuente. Esto puede ser usado para descubrir qué otras unidades *Bluetooth* están dentro del rango. Durante un sub-estado de "inquiry" o búsqueda, la unidad de descubrimiento recoge la dirección del dispositivo y el reloj de todas las unidades que respondan al mensaje de búsqueda, entonces la unidad puede iniciar una conexión con alguna de las unidades descubiertas. El mensaje de búsqueda difundido por la fuente no contiene información de ella, sin embargo, puede indicar qué clase de dispositivos deberían responder. Una unidad que permita ser descubierta, regularmente entra en un sub-estado de "inquiry scan" para responder a los mensajes de búsqueda.

Existen dos formas de detectar otras unidades. La primera, detecta todas las otras unidades en el rango de cobertura, y la segunda, detecta un tipo específico de unidades. Los

esclavos que se encuentran en el sub-estado de “page scan”, escuchan esperando su propio código de acceso de dispositivo. El maestro en el sub-estado “page” activa y conecta a un esclavo. El maestro trata de capturar al esclavo transmitiendo repetidamente el código de acceso de dispositivo en diferentes canales de salto. Debido a que los relojes del maestro y del esclavo no están sincronizados, el maestro no sabe exactamente cuándo y en qué frecuencia de salto se activará el esclavo.

Después de haber recibido su propio código de acceso de dispositivo, el esclavo transmite un mensaje de respuesta. Este mensaje de respuesta es simplemente el código de acceso de dispositivo del esclavo. Cuando el maestro ha recibido este paquete, envía un paquete de control con información acerca de su reloj, dirección, clase de dispositivo, etc. El esclavo responde con un nuevo mensaje donde envía su dirección. Si el maestro no obtiene esta respuesta en un determinado tiempo, él reenvía el paquete de control. Si el esclavo excede el tiempo de espera, entonces retorna al sub-estado de “page scan”. Si es el maestro quien lo excede, entonces retorna al sub-estado de “page” e informa a las capas superiores.

Cuando se establece la conexión, la comunicación inicia con un paquete de sondeo desde el maestro hacia el esclavo. Como respuesta se envía un nuevo paquete de sondeo y de esta forma se verifica que la secuencia de salto y la sincronización sean correctas. La Figura 1.10 muestra la inicialización de la comunicación sobre el nivel *banda base*.

Cada “transceiver” (receptor-transmisor) *Bluetooth* tiene una única dirección de dispositivo de 48 bits asignada, la cual está dividida en tres campos: campo *LAP*, campo *UAP* y campo *NAP*. Los campos *LAP* y *UAP* forman la parte significativa del código de acceso. En la Figura 1.11 se puede observar el formato de la dirección para un dispositivo *Bluetooth*. La dirección del dispositivo es conocida públicamente y puede ser obtenida a través de una rutina “inquiry”.

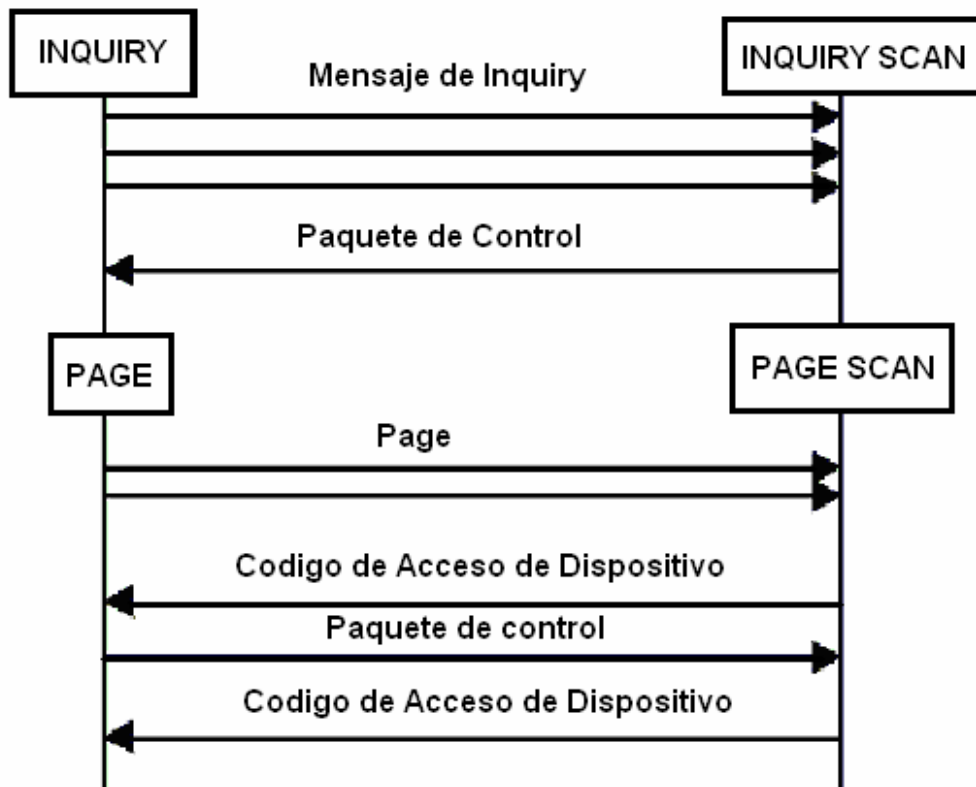


FIGURA 1.10

Iniciación de comunicación sobre el nivel banda base



FIGURA 1.11

Dirección de dispositivo Bluetooth

1.2.8 Seguridad en Bluetooth.

Con el fin de brindar protección y confidencialidad a la información, el sistema debe ofrecer medidas de seguridad en las dos capas, la de aplicación y de enlace. Todas las unidades *Bluetooth* tienen implementadas las mismas rutinas de autenticación y cifrado.

En la capa de enlace, estas rutinas constan de cuatro entidades diferentes: *una única dirección pública, dos llaves secretas y un número aleatorio* el cual es diferente para cada transacción.

Solamente es cifrada la carga útil. El *código de acceso* y la cabecera de paquete nunca son *cifrados*.

Cada tipo de unidad *Bluetooth* tiene una rutina de *autenticación* común. El maestro genera un número aleatorio y lo envía al esclavo, el esclavo usa este número y su propia identidad para calcular el número de autenticación. Luego, este número es enviado al maestro quien hace el mismo cálculo. Si los dos números generados son iguales entonces la autenticación es concedida.

El *cifrado*, frecuentemente está restringido por leyes de varios países. Para evitar estas limitaciones, en *Bluetooth* la llave de cifrado no es estática, ésta es deducida de la llave de autenticación cada vez que se activa la de cifrado. [8]

1.3 PROTOCOLO DE GESTIÓN DE ENLACE (LMP)

En el *protocolo de gestión de enlace*, *LMP* (“*Link Manager Protocol*”), se usan mensajes asociados con el establecimiento, seguridad y control. Los mensajes son enviados en la carga útil y no en los mensajes de datos de *L2CAP* (“*Logical Link Control and Adaptation Protocol*”). Los mensajes *LMP* son separados de los demás por medio de un valor reservado en uno de los campos de la cabecera de carga útil. Todos los mensajes *LMP* son extraídos e interpretados por la capa *LMP* del receptor, esto significa que ningún mensaje es enviado a capas superiores. [8]

Los mensajes *LMP* tienen mayor prioridad que los datos de usuario, esto significa que si la *gestión de enlace* necesita enviar un mensaje, éste no debe ser retrasado por otro tráfico. Solamente las retransmisiones de los paquetes del nivel de banda base pueden retrasar los mensajes *LMP*. Además, éstos no necesitan rutinas de reconocimiento ya que la capa banda base asegura un enlace confiable. El *protocolo de gestión de enlace* soporta mensajes para:

- ✓ Autenticación
- ✓ Paridad
- ✓ Cifrado
- ✓ Temporización y sincronización
- ✓ Versión y características
- ✓ “Switch” para desempeño como maestro o esclavo dependiendo de si el dispositivo es quien inicia (maestro) o no (esclavo) el enlace con otro dispositivo.
- ✓ Petición de nombre
- ✓ Desconexión
- ✓ Modo “hold”: el maestro ordena al esclavo entrar en este estado para ahorro de potencia.
- ✓ Modo “sniff”: para envío de mensajes en “timeslots” específicos.
- ✓ Modo “park”: para que el esclavo permanezca inactivo pero sincronizado en la “*piconet*”.
- ✓ Enlaces *SCO*
- ✓ Control de paquetes “multi-slot”
- ✓ Supervisión de enlace

1.3.1 Establecimiento de conexión

Después del procedimiento “paging”, el maestro debe encuestar al esclavo enviando paquetes de sondeo. El otro lado recibe este mensaje y lo acepta o rechaza, si es aceptado, la comunicación incluyendo las capas superiores están disponibles, ver Figura 1.12.

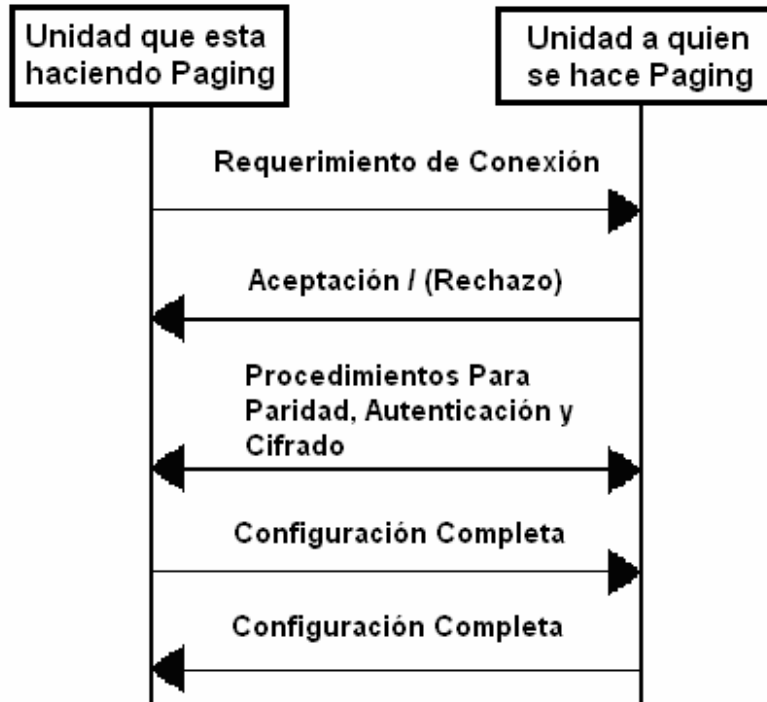


FIGURA 1.12
Establecimiento de la conexión

1.4 INTERFAZ DEL CONTROLADOR DE HOST (HCI)

La *HCI* (“*Host Controller Interface*”) proporciona una interfaz de comando al *controlador banda base* y a la *gestión de enlace*, además de acceso al hardware y a los registros de control. Esta interfaz brinda un método estándar para acceder a los recursos de banda base *Bluetooth*.

1.4.1 Capas mas bajas del “stack Bluetooth”.

A continuación se hace una breve descripción de las capas más bajas del “stack” de *software* y *hardware Bluetooth*. La Figura 1.13 da una idea de estas capas.

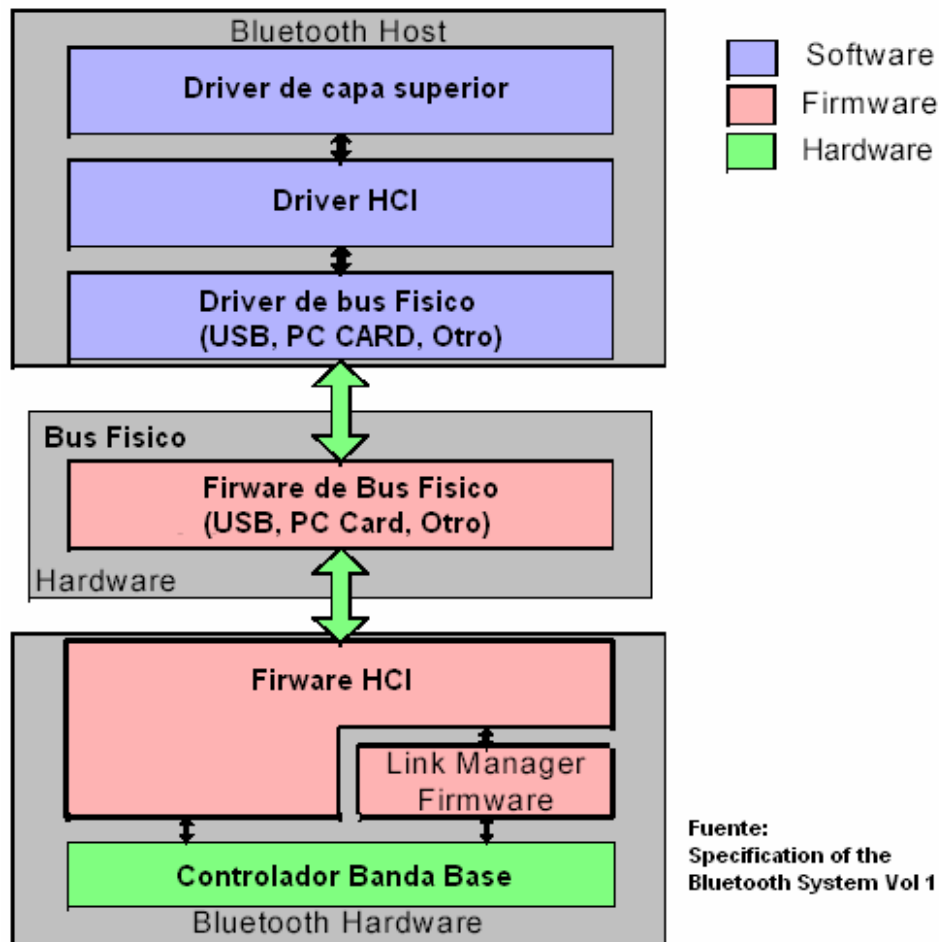


FIGURA 1.13

Diagrama general de las capas más bajas del “Stack Bluetooth”.

El “*firmware*” *HCI* implementa los comandos *HCI* para el *hardware* a través del acceso de comandos banda base, comandos de la *gestión de enlace*, hardware de registros de estado, registros de control y registros de eventos. La Figura 1.14 muestra el recorrido de un dato transferido de un dispositivo a otro. El Controlador o “driver” *HCI* (en el ‘host’) intercambia datos y comandos con el “*firmware*” *HCI* (en el hardware).

El “driver” de la capa de transporte, por ejemplo un bus físico, brinda a las dos capas *HCI* la posibilidad de intercambiar información.

1.4.2 Posibles arquitecturas de bus físico.

Los dispositivos *Bluetooth* tienen varias interfaces de bus físicas que pueden ser usadas para conectar el hardware.

Estos buses pueden tener diferentes arquitecturas y parámetros. El *controlador de “host”* soporta tres arquitecturas de bus físico, *USB*, *UART* y *PC Card*. Todas ellas pueden manejar varios canales lógicos sobre el mismo canal físico simple (a través de “**endpoints**”), por lo tanto los canales de control, datos y voz no requieren alguna interfaz física adicional. [8]

El objetivo principal de la capa de transporte del *controlador de host* es la transparencia entre el “driver” del *controlador de “host”* y el *controlador de host*.

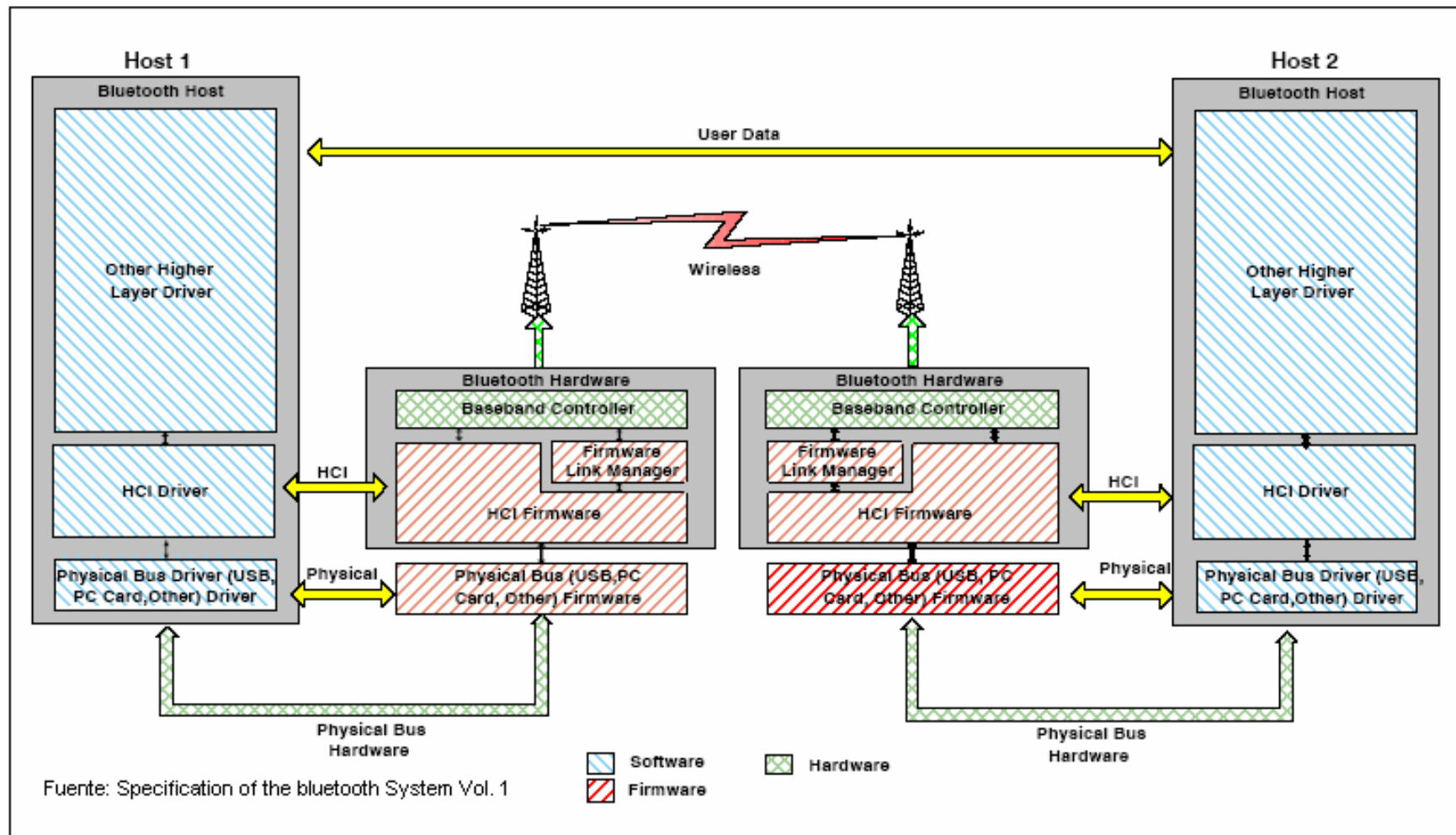


FIGURA 1.14

Diagrama general “end to end” de las capas de software más bajas

1.5 PROTOCOLO DE CONTROL Y ADAPTACIÓN DE ENLACE LÓGICO (L2CAP)

L2CAP se encuentra sobre el *protocolo de gestión de enlace (LMP)* y reside en la capa de enlace de datos. *L2CAP* permite a protocolos de niveles superiores y a aplicaciones la transmisión y recepción de paquetes de datos *L2CAP* de hasta 64 kilobytes, con capacidad de multiplexación de protocolo, operación de segmentación y reensamble, y abstracción de grupos. Para cumplir sus funciones, *L2CAP* espera que la *banda base* suministre paquetes de datos en “full duplex”, que realice el chequeo de integridad de los datos y que reenvíe los datos hasta que hayan sido reconocidos satisfactoriamente. Las capas superiores que se comunican con *L2CAP* son por ejemplo el *SDP protocolo de descubrimiento de servicio (“Service Discovery Protocol”)*, el *RFCOMM* y el *control de telefonía (TCS)*

1.5.1 Canales.

L2CAP está basado en el concepto de canales. Se asocia un identificador de canal, *CID (“Channel Identifiers”)*, a cada uno de los “**endpoints**” de un canal *L2CAP*. Los *CIDs* están divididos en dos grupos, uno con identificadores reservados para funciones *L2CAP* y otro con identificadores libres para implementaciones particulares. Los canales de datos orientados a la conexión representan una conexión entre dos dispositivos, donde un *CID* identifica cada “**endpoint**” del canal.

Los canales no orientados a la conexión limitan el flujo de datos a una sola dirección. La señalización de canal es un ejemplo de un canal reservado. Este canal es usado para crear y establecer canales de datos orientados a la conexión y para negociar cambios en las características de esos canales.

1.5.2 Operaciones entre Capas.

Las implementaciones *L2CAP* deben transferir datos entre protocolos de capas superiores e inferiores. Cada implementación debe soportar un grupo de comandos de señalización, además, debe ser capaz de aceptar ciertos tipos de eventos de capas

inferiores y generar eventos para capas superiores. En la Figura 1.15 se muestra esta arquitectura.

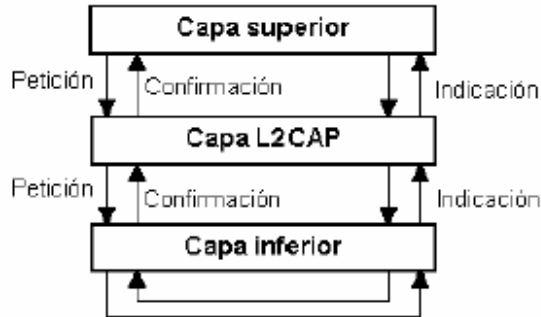


FIGURA 1.15
Arquitectura L2CAP

1.5.3 Segmentación y Reensamblado.

Los paquetes de datos definidos por el protocolo *banda base* están limitados en tamaño. Los paquetes *L2CAP* grandes deben ser segmentados en varios paquetes *banda base* más pequeños antes de transmitirse y luego deben ser enviados a la *gestión de enlace*. En el receptor los pequeños paquetes recibidos de la banda base son reensamblados en paquetes *L2CAP* más grandes. Varios paquetes *banda base* recibidos pueden ser reensamblados en un solo paquete *L2CAP* seguido de un simple chequeo de integridad. La segmentación y reensamblado *SAR* (“*Segmentation and Reassembly*”), funcionalmente es absolutamente necesaria para soportar protocolos usando paquetes más grandes que los soportados por la *banda base*. La Figura 1.16 muestra la segmentación *L2CAP* el primer segmento LM de un paquete *L2CAP* debe ser “10” seguido de los segmentos LM “01”.

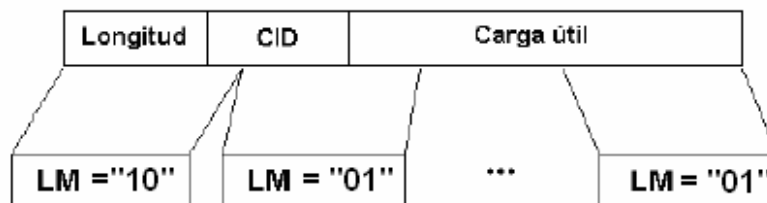


FIGURA 1.16
Segmentación L2CAP

1.5.4 Eventos.

Todos los mensajes y “timeouts” que entran en la capa *L2CAP*, son llamados eventos. Los eventos se encuentran divididos en cinco categorías: indicaciones y confirmaciones de capas inferiores, peticiones de señal y respuestas de capas *L2CAP*, datos de capas *L2CAP*, peticiones y respuestas de capas superiores, y eventos causados por expiraciones de tiempo.

1.5.5 Acciones.

Todos los mensajes y “timeouts” enviados desde la capa *L2CAP* son llamados acciones (en el lado del receptor estas acciones son llamadas eventos). Las acciones se encuentran divididas en cinco categorías: peticiones y respuestas a capas inferiores, peticiones y respuestas a capas *L2CAP*, datos a capas *L2CAP*, indicaciones a capas superiores, y configuración de “timers”.

1.5.6 Formato del paquete de datos.

L2CAP está basado en paquetes pero sigue un modelo de comunicación basado en canales. Un canal representa un flujo de datos entre entidades *L2CAP* en dispositivos remotos. Los canales pueden ser o no orientados a la conexión. Como se puede observar en la Figura 1.17, los paquetes de canal orientado a la conexión están divididos en tres campos: *longitud de la información*, *identificador de canal*, e *información*.



FIGURA 1.17
Paquete L2CAP

Los paquetes de canal de datos no orientados a la conexión son iguales a los paquetes orientados a la conexión pero adicionalmente incluyen un campo con información multiplexada de protocolo y servicio.

1.5.7 Calidad de servicio (QoS).

La capa *L2CAP* transporta la información de calidad de servicio a través de los canales y brinda control de admisión para evitar que canales adicionales violen contratos de calidad de servicio existentes.

Algunos esclavos pueden requerir un alto rendimiento o una respuesta rápida. Antes de que un esclavo con grandes peticiones sea conectado a una *piconet*, el esclavo trata de obtener una garantía a sus demandas. Puede solicitar una determinada razón de transmisión, tamaño del buffer de tráfico, ancho de banda, tiempo de recuperación de datos, etc. Por lo tanto, antes de que el maestro conecte a un nuevo esclavo o actualice la configuración de calidad, debe chequear si posee “timeslots” y otros recursos libres.

1.6 PROTOCOLO DE DESCUBRIMIENTO DE SERVICIO (SDP)

El *protocolo de descubrimiento de servicio*, *SDP*, brinda a las aplicaciones recursos para descubrir qué servicios están disponibles y determinar las características de dichos servicios.

1.6.1 Descripción General.

Un servicio es una entidad que puede brindar información, ejecutar una acción o controlar un recurso a nombre de otra entidad.

El *SDP* ofrece a los clientes la facilidad de averiguar sobre servicios que sean requeridos, basándose en la clase de servicio o propiedades específicas de estos servicios. Para hacer más fácil la búsqueda, el *SDP* la habilita sin un previo conocimiento de las características específicas de los servicios. Las unidades *Bluetooth* que usan el *SDP* pueden ser vistas como un servidor y un cliente. El servidor posee los servicios y el cliente es quien desea acceder a ellos. En el *SDP* esto es posible ya que el cliente envía una petición al servidor y el servidor responde con un mensaje. El *SDP* solamente soporta el descubrimiento del servicio, no la llamada del servicio.

1.6.2 Registros de servicio.

Los registros de servicio contienen propiedades que describen un servicio determinado. Cada propiedad de un registro de servicio consta de dos partes, un identificador de propiedad y un valor de propiedad. El identificador de propiedad es un número único de 16 bits que distingue cada propiedad de servicio de otro dentro de un registro. El valor de propiedad es un campo de longitud variable que contiene la información.

1.6.3 El protocolo.

El *protocolo de descubrimiento de servicio (SDP)* usa un modelo petición/respuesta. En la Figura 1.18 se muestra el procedimiento *SDP*.

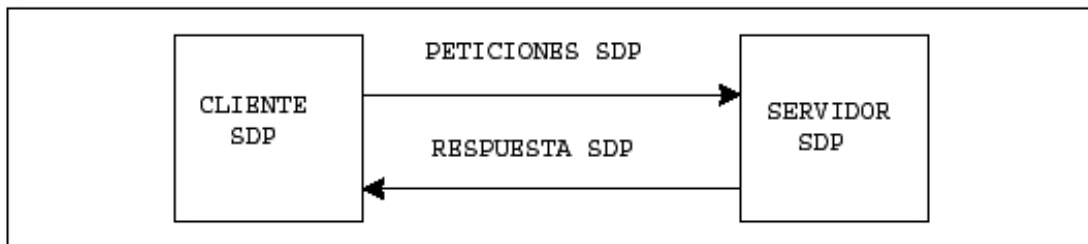


FIGURA 1.18

Procedimiento SDP

✓ **Petición de búsqueda de servicio:**

Se genera por el cliente para localizar registros de servicio que concuerden con un patrón de búsqueda dado como parámetro. Aquí el servidor examina los registros en su base de datos y responde con una *respuesta a búsqueda de servicio*.

✓ **Respuesta a búsqueda de servicio:**

Se genera por el servidor después de recibir una *petición de búsqueda de servicio* válida.

✓ **Petición de propiedad de servicio:**

Una vez el cliente ya ha recibido los servicios deseados, puede obtener mayor información de uno de ellos dando como parámetros el registro de servicio y una lista de propiedades deseadas.

✓ **Respuesta a propiedad de servicio:**

El *SDP* genera una respuesta a una *petición de propiedad de servicio*. Ésta contiene una lista de propiedades del registro requerido.

✓ **Petición de búsqueda y propiedad de servicio:**

Se suministran un patrón de servicio con servicios deseados y una lista de propiedades deseadas que concuerden con la búsqueda.

✓ **Respuesta de búsqueda y propiedad de servicio:**

Como resultado se puede obtener una lista de servicios que concuerden con un patrón dado y las propiedades deseadas de estos servicios.

1.7 RFCOMM

El protocolo *RFCOMM* brinda emulación de puertos seriales sobre el protocolo *L2CAP*. La capa *RFCOMM* es una simple capa de transporte provista adicionalmente de emulación de circuitos de puerto serial *RS-232*. El protocolo *RFCOMM* soporta hasta 60 puertos emulados simultáneamente. Dos unidades *Bluetooth* que usen *RFCOMM* en su comunicación pueden abrir varios puertos seriales emulados, los cuales son multiplexados entre sí. La Figura 1.19 muestra el esquema de emulación para varios puertos seriales.

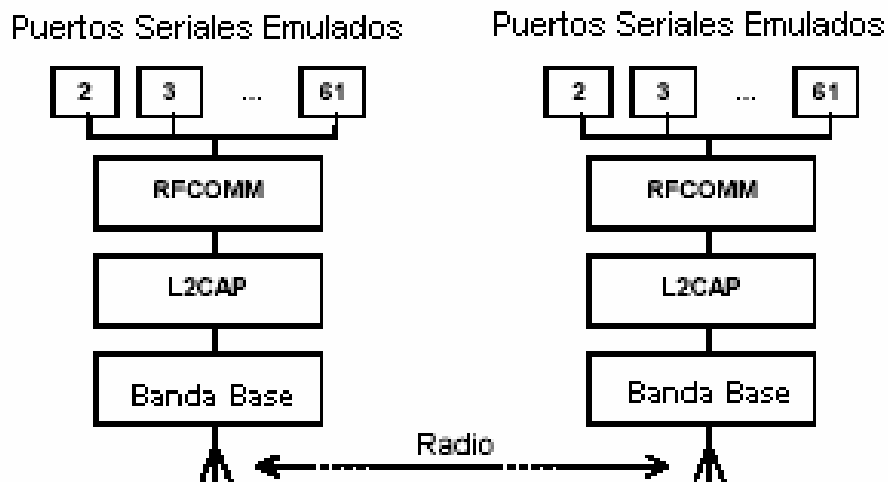


FIGURA 1.19

Varios puertos seriales emulados mediante RFCOMM

Muchas aplicaciones hacen uso de puertos seriales. El *RFCOMM* está orientado a hacer más flexibles estos dispositivos, soportando fácil adaptación de comunicación *Bluetooth*. Un ejemplo de una aplicación de comunicación serial es el protocolo *punto-a-punto (PPP)*. El *RFCOMM* tiene construido un esquema para emulación de “null modem” y usa a *L2CAP* para cumplir con el control de flujo requerido por alguna aplicación.

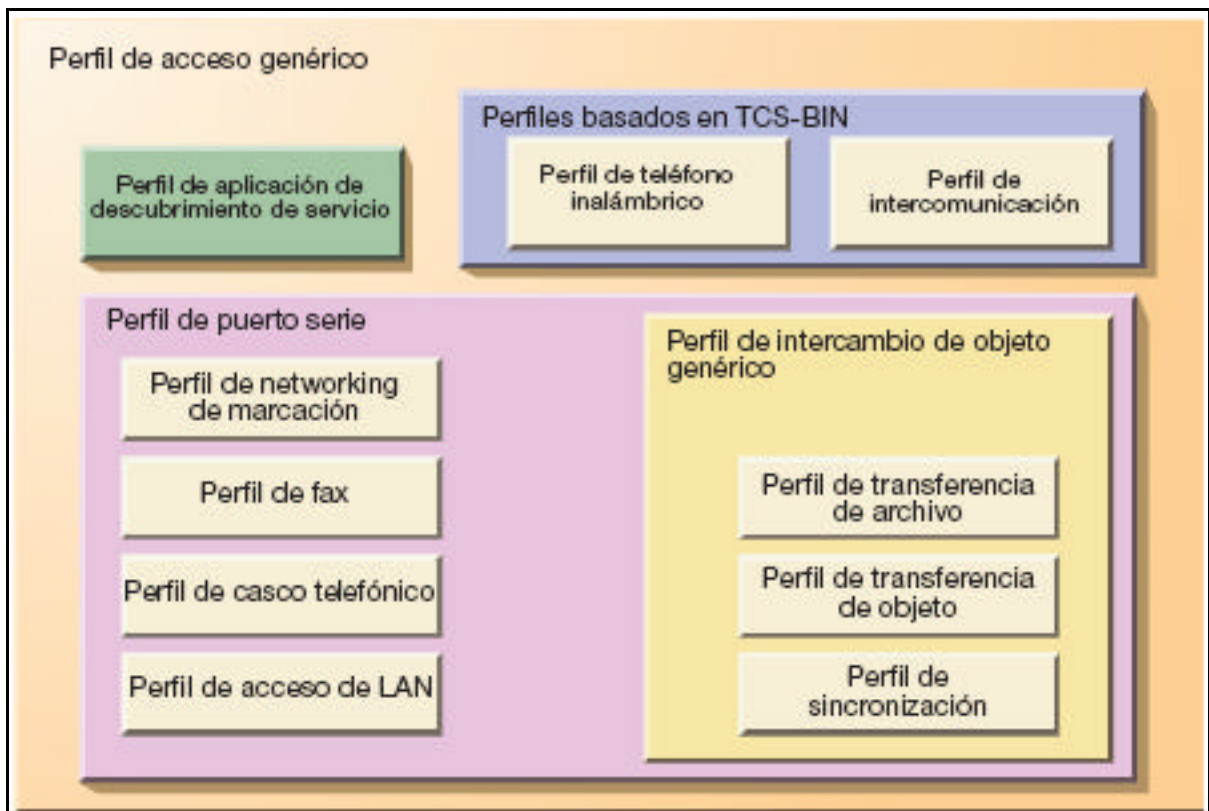
1.8 PERFILES BLUETOOTH

El estándar *Bluetooth* fue creado para ser usado por un gran número de fabricantes e implementado en áreas ilimitadas. Para asegurar que todos los dispositivos que usen *Bluetooth* sean compatibles entre sí son necesarios esquemas estándar de comunicación en las principales áreas. Para evitar diferentes interpretaciones del estándar *Bluetooth* acerca de cómo un tipo específico de aplicación debería ser implementado, el *Bluetooth* “Special Interest Group” (*SIG*), ha definido modelos de usuario y perfiles de protocolo. Un perfil define una selección de mensajes y procedimientos de las especificaciones *Bluetooth* y ofrece una descripción clara de la interfaz de aire para servicios específicos.

Un perfil puede ser descrito como una “rebanada” completa del “stack” de protocolo.

Existen cuatro perfiles generales definidos, en los cuales están basados directamente algunos de los modelos de usuario más importantes y sus perfiles. [28]

Estos cuatro modelos son *Perfil Genérico de Acceso (GAP)*, *Perfil de Puerto Serial*, *Perfil de Aplicación de Descubrimiento de Servicio (SDAP)* y *Perfil Genérico de Intercambio de Objetos (GOEP)*. A continuación se hace una breve descripción de estos y algunos otros perfiles *Bluetooth*. La Figura 1.20 muestra el esquema de los perfiles *Bluetooth*. En ella se puede observar la jerarquía de los perfiles, como por ejemplo que todos los perfiles están contenidos en el *Perfil Genérico de Acceso (GAP)*.



Fuente: Specification of the Bluetooth System Vol 1

FIGURA 1.20

Los Perfiles Bluetooth

1.8.1 Perfil Genérico de Acceso (GAP).

Este perfil define los procedimientos generales para el descubrimiento y establecimiento de conexión entre dispositivos *Bluetooth*. El *GAP* maneja el descubrimiento y establecimiento entre unidades que no están conectadas y asegura que cualquier par de unidades *Bluetooth*, sin importar su fabricante o aplicación, puedan intercambiar información a través de *Bluetooth* para descubrir qué tipo de aplicaciones soportan las unidades.

1.8.2 Perfil de Puerto Serial.

Este perfil define los requerimientos para dispositivos *Bluetooth*, necesarios para establecer una conexión de cable serial emulada usando RFCOMM entre dos dispositivos similares. Este perfil solamente requiere soporte para paquetes de un “slot”. Esto significa que pueden ser usadas tasas de transferencia de datos de hasta 128 kbps. El soporte para tasas más altas es opcional. RFCOMM es usado para transportar los datos de usuario, señales de control de *modem* y comandos de configuración. *El perfil de puerto serial es dependiente del GAP.*

1.8.3 Perfil de Aplicación de Descubrimiento de Servicio (SDAP).

Este perfil define los protocolos y procedimientos para una aplicación en un dispositivo *Bluetooth* donde se desea descubrir y recuperar información relacionada con servicios localizados en otros dispositivos. El *SDAP* es dependiente del *GAP*.

1.8.4 Perfil Genérico de Intercambio de Objetos (GOEP).

Este perfil define protocolos y procedimientos usados por aplicaciones para ofrecer características de intercambio de objetos. Los usos pueden ser, por ejemplo, sincronización, transferencia de archivos o modelo ‘Object Push’. Los dispositivos más comunes que usan este modelo son agendas electrónicas, *PDA*s, teléfonos celulares y teléfonos móviles. El *GOEP* es dependiente del *perfil de puerto serial*.

1.8.5 Perfil de Telefonía Inalámbrica.

Este perfil define cómo un teléfono móvil puede ser usado para acceder a un servicio de telefonía de red fija a través de una estación base. Es usado para telefonía inalámbrica de hogares u oficinas pequeñas. El perfil incluye llamadas a través de una estación base, haciendo llamadas de intercomunicación directa entre dos terminales y accediendo adicionalmente a redes externas. Es usado por dispositivos que implementan el llamado “teléfono 3-en-1”.

1.8.6 Perfil de Intercomunicador.

Este perfil define usos de teléfonos móviles los cuales establecen enlaces de conversación directa entre dos dispositivos. El enlace directo es establecido usando señalización de telefonía sobre *Bluetooth*. Los teléfonos móviles que usan enlaces directos funcionan como “walkie-talkies”.

1.8.7 Perfil de Manos Libres.

Este perfil define los requerimientos, para dispositivos *Bluetooth*, necesarios para soportar el uso de manos libres. En este caso el dispositivo puede ser usado como unidad de audio inalámbrico de entrada/salida. El perfil soporta comunicación segura y no segura.

1.8.8 Perfil “Dial-up Networking”.

Este perfil define los protocolos y procedimientos que deben ser usados por dispositivos que implementen el uso del modelo llamado *Puente Internet*. Este perfil es aplicado cuando un teléfono celular o *modem* es usado como un *modem* inalámbrico.

1.8.9 Perfil de Fax.

Este perfil define los protocolos y procedimientos que deben ser usados por dispositivos que implementen el uso de fax. En el perfil un teléfono celular puede ser usado como un fax inalámbrico.

1.8.10 Perfil de Acceso LAN.

Este perfil define el acceso a una *red de área local, LAN*, usando el protocolo punto-a-punto, *PPP*, sobre *RFCOMM*. *PPP* es ampliamente usado para lograr acceder a redes soportando varios protocolos de red. El perfil soporta acceso *LAN* para un dispositivo *Bluetooth* sencillo, acceso *LAN* para varios dispositivos *Bluetooth* y *PC-a-PC* (usando interconexión *PPP* con emulación de cable serial).

1.8.11 Perfil Object Push.

Este perfil define protocolos y procedimientos usados en el modelo “object push”. Este perfil usa el *GOEP*. En el modelo “object push” hay procedimientos para introducir en el “inbox”, sacar e intercambiar objetos con otro dispositivo *Bluetooth*.

1.8.12 Perfil de Transferencia de Archivos.

Este perfil define protocolos y procedimientos usados en el modelo de transferencia de archivos. El perfil usa el *GOEP*. En el modelo de transferencia de archivos hay procedimientos para chequear un grupo de objetos de otro dispositivo *Bluetooth*, transferir objetos entre dos dispositivos y manipular objetos de otro dispositivo. Los objetos podrían ser archivos o folders de un grupo de objetos tal como un sistema de archivos.

1.8.13 Perfil de Sincronización.

Este perfil define protocolos y procedimientos usados en el modelo de sincronización. Éste usa el *GOEP*. El modelo soporta intercambios de información, por ejemplo para sincronizar calendarios de diferentes dispositivos. [9]

2. DESCRIPCIÓN DE HARDWARE Y PRODUCTOS BLUETOOTH

Después de analizar y comprender la tecnología Bluetooth se procedió a hacer una búsqueda de productos disponibles en el mercado con el fin de conocer las diversas aplicaciones que se ofrecen y principalmente determinar qué producto adquirir o implementar para el desarrollo de esta tesis. En este capítulo se presenta una descripción detallada del hardware implementado y se hace una breve descripción de los diferentes tipos de productos Bluetooth disponibles así como el procedimiento mediante el cual se califica un producto como producto Bluetooth.

A continuación se detallan los elementos necesarios para realizar una red Bluetooth, ya sea implementando una tarjeta Bluetooth utilizando diferentes elementos disponibles en el mercado así como también utilizando un dispositivo Bluetooth USB.

2.1 TARJETA BLUETOOTH

Uno de los puntos esenciales para el desarrollo de esta tesis fue la utilización de hardware para la implementación. Después de conocer los diferentes tipos de productos Bluetooth disponibles, se hizo un análisis de las mejores opciones teniendo en cuenta ventajas, desventajas, costo, accesibilidad, etc. Para cumplir con el objetivo principal de este trabajo, es decir, la implementación de una red Bluetooth, se decidió construir sistemas semiembebidos Bluetooth utilizando módulos, antenas y demás dispositivos periféricos necesarios.

En la Figura 2.1 se muestra el diagrama de bloques de las tarjetas Bluetooth. En la gráfica se pueden observar las principales partes que las conforman.

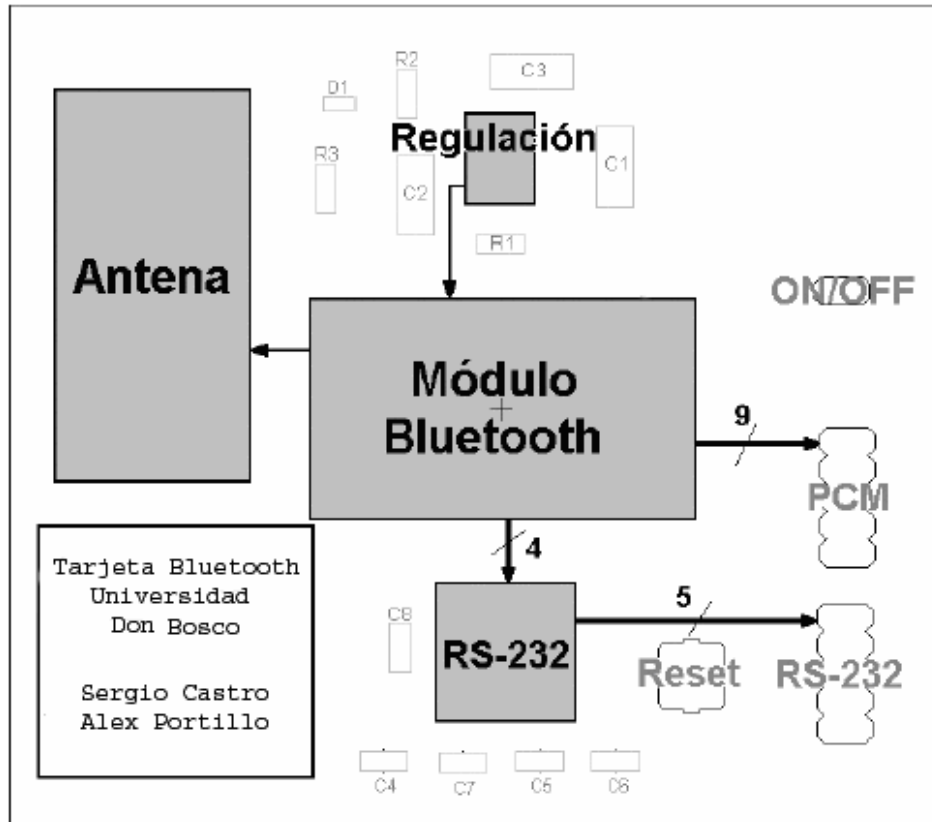


FIGURA 2.1
Diagrama de Bloques de las tarjetas Bluetooth

Los bloques funcionales de las tarjetas son:

✓ *Módulo Bluetooth:*

Es el “chip-multichip” que tiene implementadas las capas más bajas del “stack” y posee diferentes interfaces para lograr comunicación de voz y datos. Es el bloque más importante del sistema.

✓ *Antena:*

El módulo incluye antena, éste es un dispositivo externo necesario para lograr la comunicación *Bluetooth*.

✓ *Transceiver RS-232:*

Necesario para convertir las señales *RS-232* de una fuente de menor a mayor valor de voltaje.

✓ *Regulación de voltaje:*

Necesario para el suministro de potencia del módulo y el transceiver *RS-232*. El voltaje regulado es de 3.3V.

2.1.1 Descripción de la tarjeta Bluetooth.

A continuación se presenta una descripción general de la tarjeta, información más detallada se presenta en secciones posteriores. Entre las principales características de la tarjeta *Bluetooth* están:

- ✓ Módulo BISM2 de EZURIO de acuerdo a la *versión 2.0b* de *Bluetooth*
- ✓ Acceso a las señales del módulo
- ✓ *Pin de encendido externo*
- ✓ Salida RF de *potencia clase 2 (0dBm-rango de 100m)*
- ✓ *LED de monitoreo* para suministro de energía
- ✓ *Una interfaz* para diversas aplicaciones;
 UART para datos
- ✓ Tasa máxima de transmisión de datos sobre *UART* de 460,8 kbps
- ✓ *Reset* manual
- ✓ Capas inferiores del “*stack Bluetooth*” incluidas en el *hardware*, desde el *HCI* hasta la *capa de radio*.
- ✓ *Antena incluida* P8000CT-ND ANT 2.485 GHZ 802.11B BLUETOOTH
- ✓ Operación *Punto a Punto* (dependiendo del módulo utilizado)

La tarjeta *Bluetooth* únicamente soporta comunicación de datos, sin embargo posee una interfaz *PCM* disponible para comunicación de voz. La comunicación entre la tarjeta y el *controlador de host* se hace a través de la interfaz *UART/PCM*. El módulo

BISM2 usado soporta algunos de los perfiles definidos en la versión 2.0b de la especificación *Bluetooth*.

La tarjeta *Bluetooth* necesita una alimentación de 5VDC que pueden suministrarse a través de un terminal macho genérico. Posee un convertidor *DC/DC* que proporciona la alimentación al módulo *BISM2*.

El diagrama esquemático de la tarjeta se muestra en la Figura 2.2. Se puede observar los dispositivos utilizados para los diferentes bloques funcionales: módulo *Ezurio BISM2*, antena P8000CT-ND ANT 2.485 GHZ 802.11B BLUETOOTH, **transceiver** Maxim MAX3232 y regulador de voltaje National LP2987. Así también se presentan en las figuras 2.3 y 2.4 con las caras de los circuitos impresos utilizados para la elaboración de la tarjeta *Bluetooth*.

El voltaje regulado necesario para la alimentación del módulo y el “transceiver” *RS-232* es suministrado por el *LP2987*. Su circuito, tiene un pin de encendido externo (ON/OFF), que permite habilitar el suministro de potencia a la tarjeta. Además, para verificar el estado de la conexión, posee un *LED* de monitoreo (D1). El voltaje regulado es de 3.3V. El módulo posee diferentes interfaces de comunicación: *RS-232*, *PCM*.

Las señales *UART* que salen del módulo, antes de ir al conector macho *RS232* son convertidas a un voltaje mayor por medio del *MAX3232*, el cual transforma los niveles de voltaje al estándar universal *RS-232*. Al usar la interfaz *UART* el módulo funciona como un Equipo de Comunicación de Datos (DCE “Data Circuit-Terminating Equipment”), con una velocidad de 57,6 kbps por defecto.

La señal *RESET* esta conectada a un “Switch” pulsador normalmente abierto que conecta a +5 Voltios al ser este accionado.

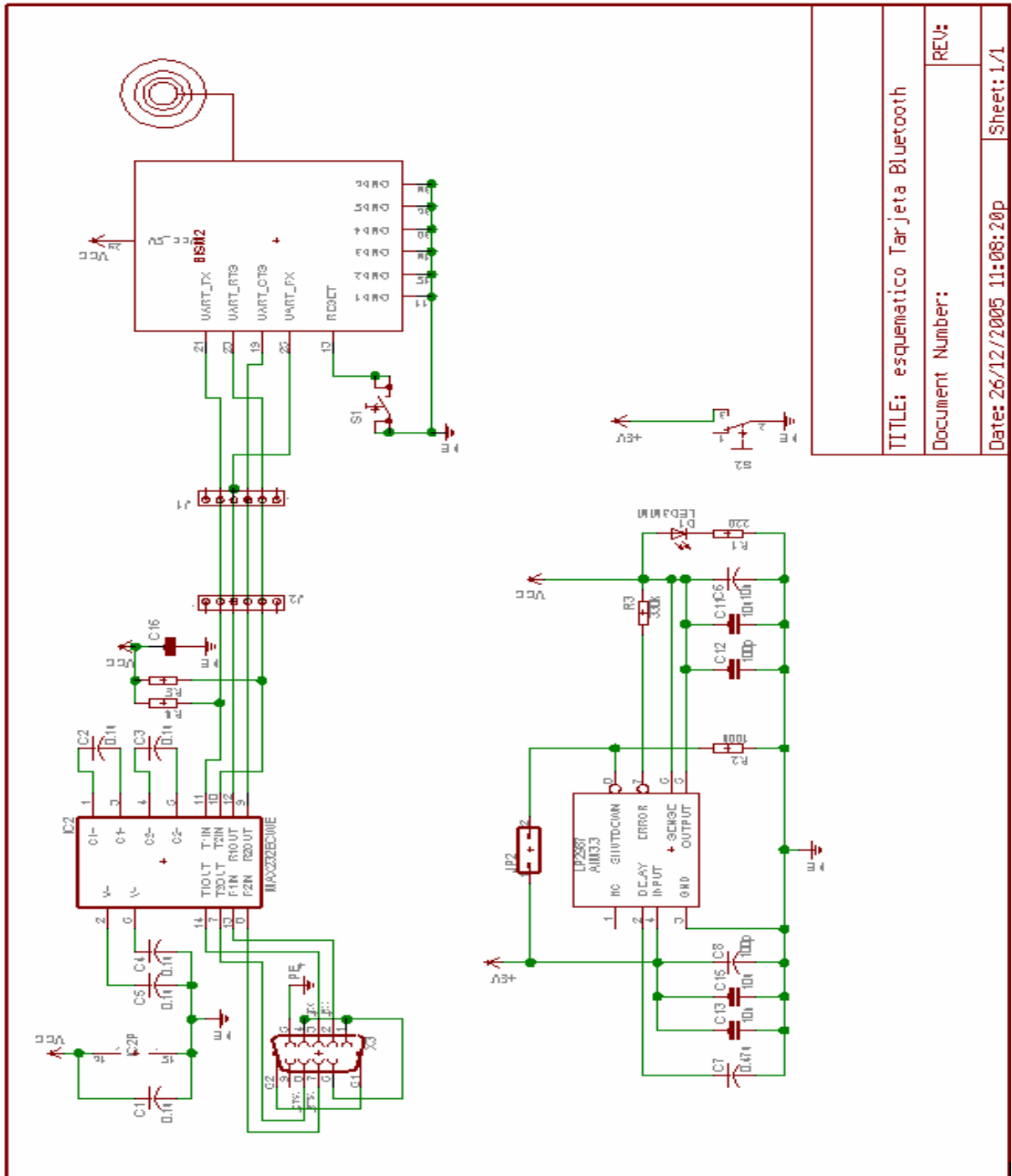


FIGURA 2.2
Diagrama esquemático de la tarjeta Bluetooth

TITLE: esquemático Tarjeta Bluetooth	
Document Number:	REV:
Date: 26/12/2005 11:08:20p	Sheet: 1/1

En la Tabla 2.1 se presenta la lista de componentes de una tarjeta *Bluetooth* detallando las referencias dentro de la tarjeta y de fabricante para cada componente.

No.	Tipo de Dispositivo	Referencia de Componente	Cant.
1	IC, Modulo Bluetooth	EZURIO BISM2	1
2	IC, Regulador de Voltaje	NATIONAL LP2987AIM-3.3 3.3 V voltaje	1
3	IC, Interfaz RS-232	MAXIM MAX3232ECWE	1
4	Antena	Antenna ANT-2.45-CHP-x 2.485 GHZ 802.11B BLUETOOTH	1
5	Capacitor 0.1uF	Tantalum 0.1 uF / 50V	5
6	Capacitor 0.47uF	Tantalum 0.47 uF / 50V	1
7	Capacitor 4.7uF	Tantalum 4.7 uF / 20V	2
8	Resistencia 100 K?	5% 1/8 Watt	1
9	Resistencia 330 K?	5% 1/8 Watt	1
10	Resistencia 220 ?	5% 1/4 Watt	1
11	Diodo Led	SOT-23	1
12	Conector RS-232		1
13	Switch		1

TABLA 2.1
Lista de componentes para la tarjeta Bluetooth

La Figura 2.5 muestra el diagrama del hardware de la tarjeta *Bluetooth*. La gráfica es una representación de cómo lucirá la tarjeta. Cada dispositivo utilizado tiene asociada una referencia, facilitando su localización física.

El Dispositivo Bluetooth final, lo podemos apreciar en las figuras 2.6 y 2.7 las cuales muestran al dispositivo con sus terminales, montado en una estructura de acrílico transparente, con la finalidad de proteger la tarjeta y sus conexiones internas y externas, además de permitir la visualización de los elementos que lo componen.

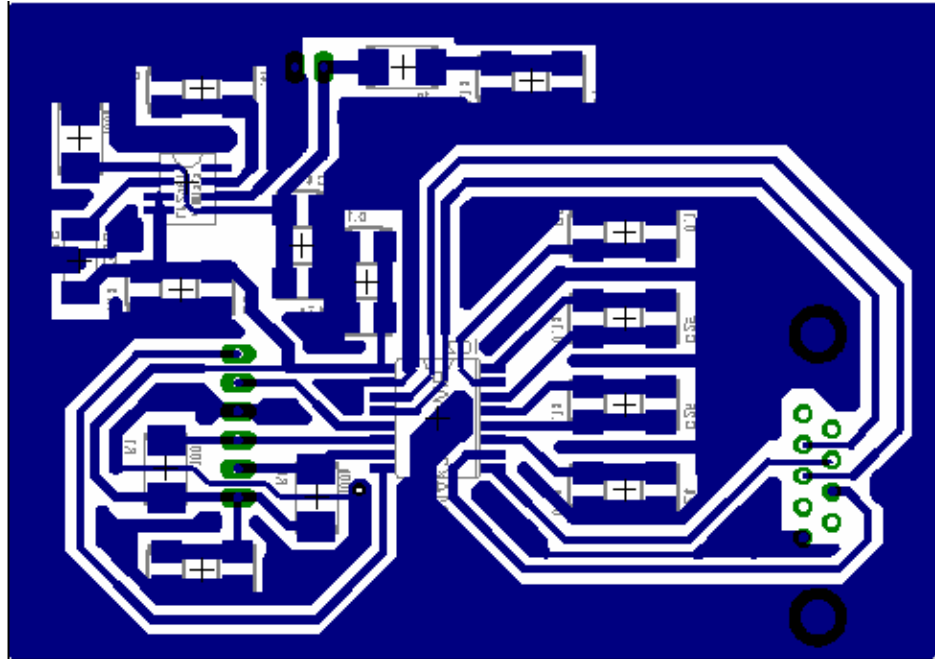


FIGURA 2.3

Cara superior de circuito impreso tarjeta bluetooth

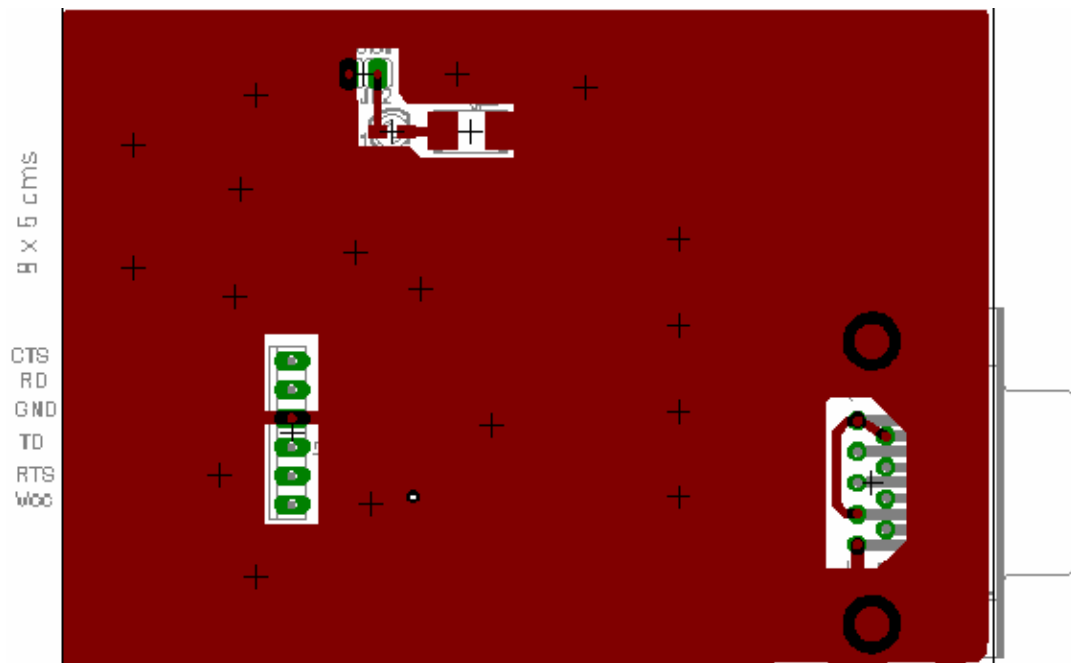


FIGURA 2.4

Cara inferior de circuito impreso tarjeta bluetooth

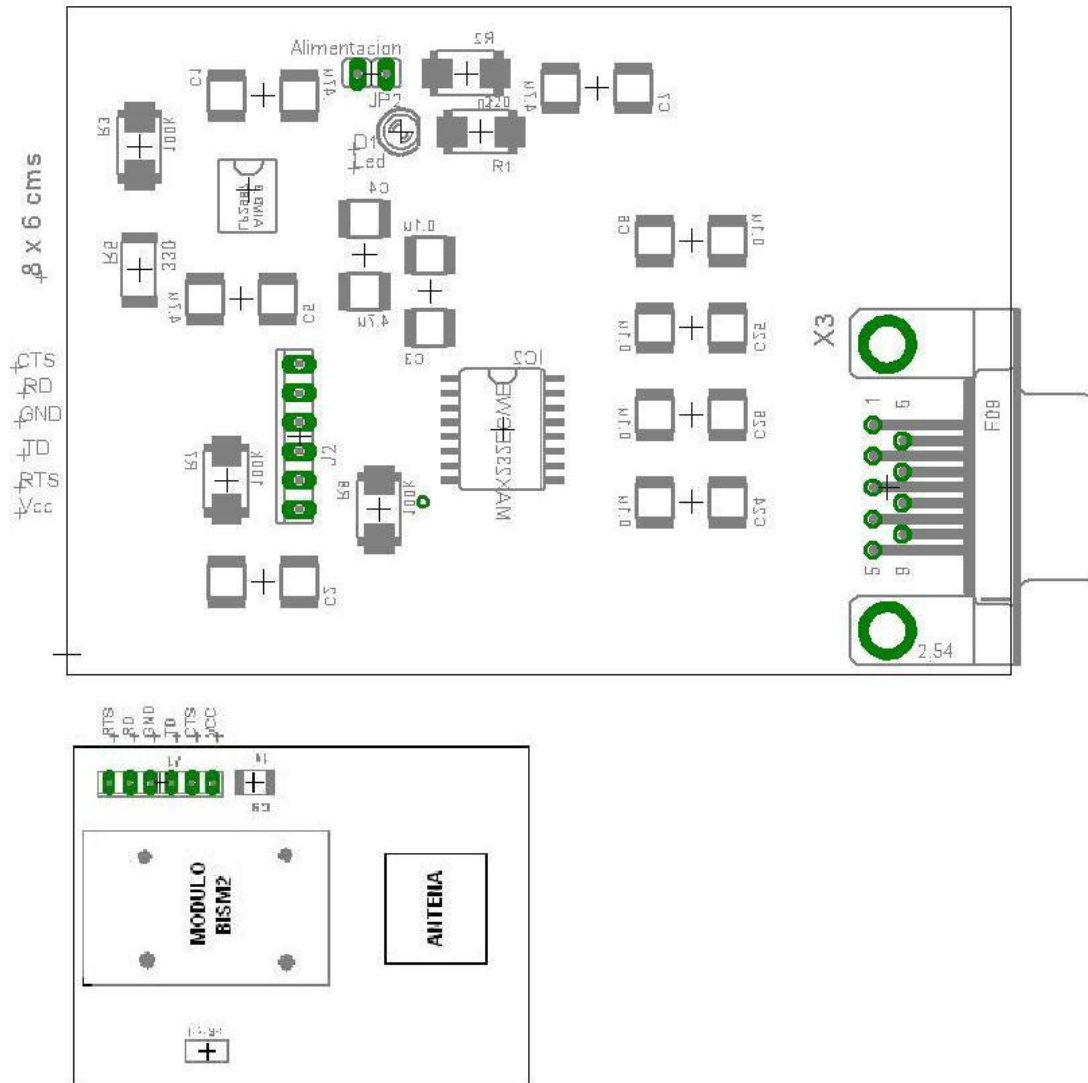


FIGURA 2.5
Diagrama del hardware de la tarjeta Bluetooth

Con el fin de conocer y entender mejor el funcionamiento de la tarjeta *Bluetooth*, a en el siguiente apartado se hace una descripción detallada de los principales dispositivos utilizados y los requerimientos que se deben tener en cuenta para la implementación de la tarjeta.

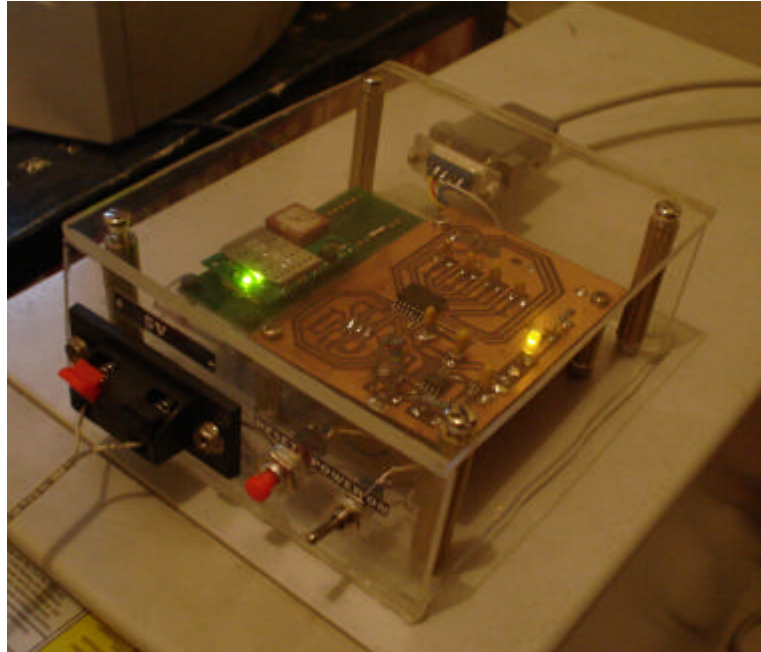


FIGURA 2.6
Hardware de la tarjeta Bluetooth



FIGURA 2.7
Tarjeta Bluetooth montada en estructura de acrílico

2.2 DESCRIPCIÓN DE COMPONENTES Y REQUERIMIENTOS PARA UNA TARJETA BLUETOOTH.

2.2.1 Módulo Bluetooth EZURIO BISM2.

Para la implementación de una tarjeta *Bluetooth* se puede utilizar el módulo de *Ezurio BISM2*. Su elección se hace teniendo en cuenta la funcionabilidad, la disponibilidad y la compatibilidad con otros dispositivos. A continuación se presenta una descripción general del módulo (ver figura 2.8). [11]

Las principales ventajas del módulo *Ezurio BISM2* son:

- ✓ Pre-calificado con la especificación *Bluetooth 2.0B*
- ✓ Potencia de salida *RF* clase 1
- ✓ Rango de cobertura de 100 metros.
- ✓ Aprobado por la **FCC** y la **ETSI**
- ✓ Tasa de transferencia máxima sobre *UART* de 300 Kbps
- ✓ Distintas interfaces: *UART* para datos, *PCM* para voz.
- ✓ Cristal oscilador interno
- ✓ “Firmware” *HCI* incluido
- ✓ Blindado

El BISM2 es un módulo de largo alcance para implementaciones *Bluetooth* en diferentes dispositivos electrónicos que soporta transmisiones de datos y voz. La comunicación entre el módulo y el controlador de “host” se lleva a cabo a través de una interfaz *UART*, la señal será amplificada por medio del “Transceiver” Max3232. El BISM2 es un módulo *Bluetooth Clase 2 (0dBm - +6 dBm)* y además soporta la mayoría de los perfiles *Bluetooth*.

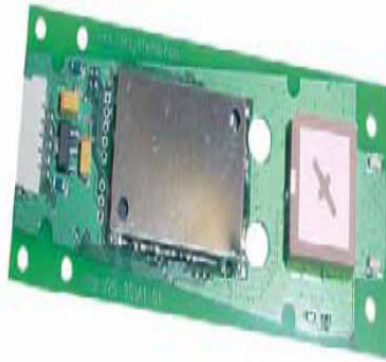


FIGURA 2.8
Chip Multichip Bluetooth BISM2

El BISM2 es un chip multi-chip de 40 pines de 22.8 x 33.8 x 7.6 mm, 8g aproximadamente, que por su dimensión facilita implementaciones en dispositivos portátiles y de tamaño reducido. El módulo consta de tres partes principales; Controlador Banda Base, Memoria Flash y Radio. Los bloques operacionales del BISM2 los conforman los tres anteriores, más un bloque de Manejo de Potencia y otro de Reloj.

El Controlador Banda Base es un ARM7 que controla la operación del radio “transceiver” a través de una interfaz UART, y adicionalmente tiene una interfaz de voz PCM y un convertidor análogo digital para entradas de señal para usos múltiples.

- La *Memoria Flash* es usada con el *Controlador Banda Base*.
- El Radio es implementado del tipo BC 04.
- En el *Manejo de Potencia* se regula y filtra el voltaje de alimentación V_{cc} que típicamente es de 3.3V.
- El Reloj interno trabaja a 13MHz con una precisión en el oscilador de cristal de 20ppm.

La figura 2.9 muestra las partes antes mencionadas.

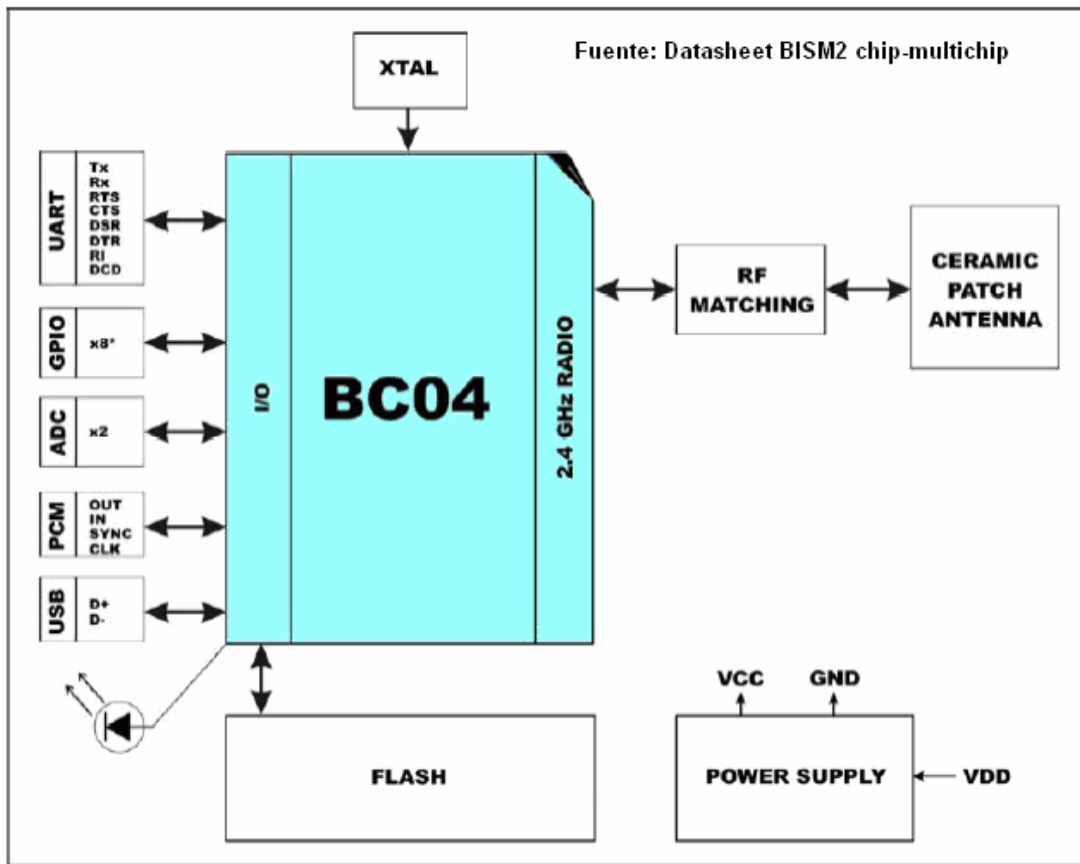


FIGURA 2.9
Diagrama de Bloques del BISM2 Chip- Multichip

El módulo *BISM2* incluye partes *HW/FW* del “stack” de protocolo *Bluetooth*. Las partes de *HW* (“*Hardware*”) incluidas son el *Radio* y la *Banda Base*, y las de *FW* (“*Firmware*”), las cuales residen en la *Flash*, son la *Interfaz del Controlador de Host HCI*, el *Manejador de Enlace LM*. El *BISM2* incluye un emulador de comandos AT, los cuales son ampliamente usados por programadores de comunicaciones de MODEM. [12] En la Figura 2.10 se pueden observar las partes de *HW/FW* incluidas en el módulo.

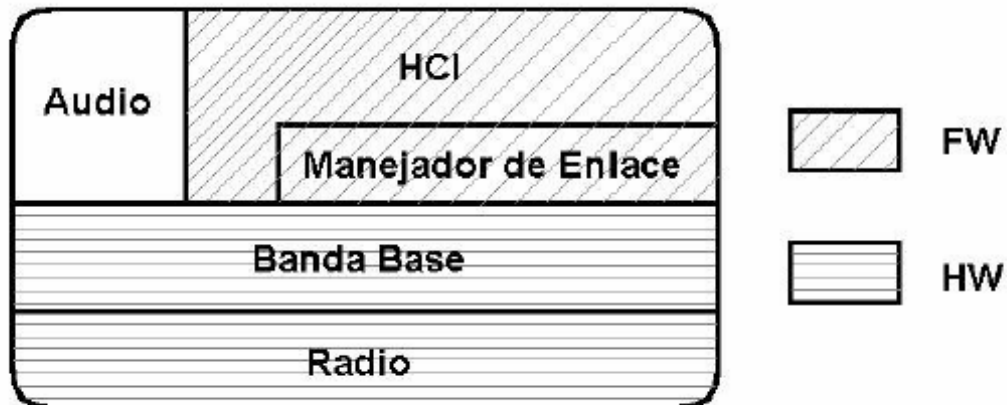


FIGURA 2.10
HW/FW del módulo Ezurio BISM2

El módulo Bluetooth BISM2 tiene las siguientes interfaces:

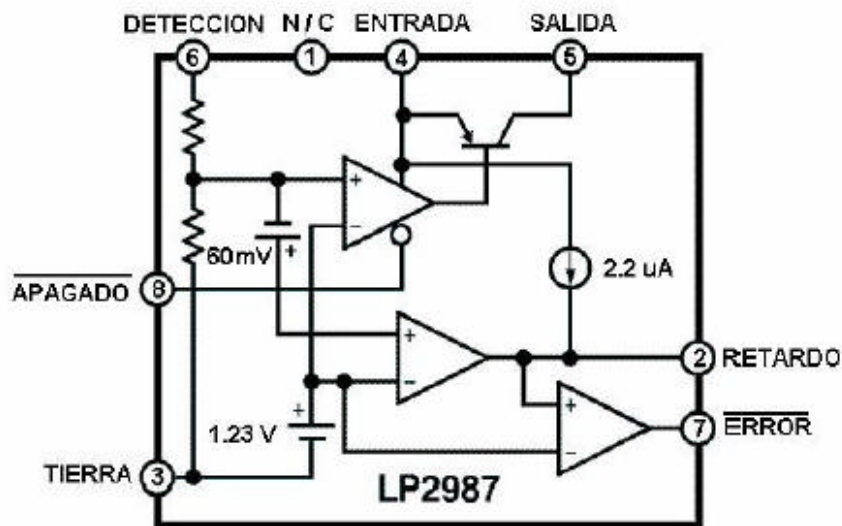
- ✓ **Interfaz de Radio:** el módulo es un dispositivo *clase 1* con una salida máxima de potencia de *6 dBm*. El rango nominal del módulo con una antena típica es de *hasta 200m (a + 2 dBm)*.
- ✓ **Banda Base:** Se usa una estructura de red “ad-hoc” con un número máximo de siete unidades activas en una sola “piconet”.
- ✓ **Interfaces HW del Módulo.**
 - **Interfaz *UART*** (“Universal Asynchronous receiver Transmitter”): esta interfaz es acorde al estándar industrial 16C450 y soporta las siguientes *tasas de transferencia de bits*: 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800 y 961200 bits/s. Existen “FIFOs” (“First IN First OUT”) de 128 bytes asociadas con la *UART*. Para esta interfaz se tienen cuatro señales, TxD & RxD se usan para flujo de datos, y RTS & CTS se usan para control de flujo.
 - **Interfaz de voz PCM:** la interfaz estándar PCM tiene una razón de muestreo de 8kHz (PCM_SYNC). El reloj PCM es variable entre 200kHz y 2MHz. Los

datos PCM pueden ser: PCM 13-16 bits, μ -Law 8 bits ó A-Law 8 bits. En este caso se tienen cuatro líneas de conexión PCM_IN, PCM_OUT, PCM_SYNC y PCM_CLK.

2.2.2. Regulador de Voltaje “National LP2987”.

Para la regulación de voltaje de la tarjeta *Bluetooth* se usa el regulador de voltaje *National LP2987AIM 3.3*. Éste es un regulador de voltaje de 200mA de salida fija con un reset de retardo de encendido implementado con un capacitor externo. El voltaje regulado necesario en la tarjeta *Bluetooth* es de 3.3V, con el cual se alimentan el módulo *Bluetooth* y el *MAX3232*. El regulador tiene una salida de precisión de 0.5%. [13]

La Figura 2.11 muestra el diagrama de bloques para el *LP2987*.



Fuente: *National Semiconductors*

FIGURA 2.11
Diagrama de bloques del regulador de voltaje LP2987

2.2.3 MAXIM MAX3232.

El *MAXIM MAX3232* es una interfaz de comunicación con requerimientos de baja potencia. Este “transceiver” habilita el cambio en *RS-232* de una alimentación de 3.0V a una de 5.5V. Tiene 2 receptores y 2 “drivers”. El *MAX3232* solamente requiere cuatro pequeños capacitores externos de 0.1 μ F y es garantizado para correr a una tasa de transferencia de datos

de 120kbps (MAX3232) o 250kbps (MAX3232E) manteniendo los niveles de salida RS-232.
[14]

2.2.4 Adaptador Board to Board de Ezurio para el BISM2.

Con el fin de hacer de la tarjeta *Bluetooth* un sistema más flexible, se utiliza una base soporte para el módulo de *EZURIO BISM2*. El soporte es del tipo “Hirose DF12C”. Este soporte permite mayor accesibilidad a los pines del modulo. En la Figura 2.12 se puede observar el soporte a implementar en el diseño de la tarjeta. Éste posee 40 pines de conexión al igual que el módulo de *Ezurio* a utilizar.

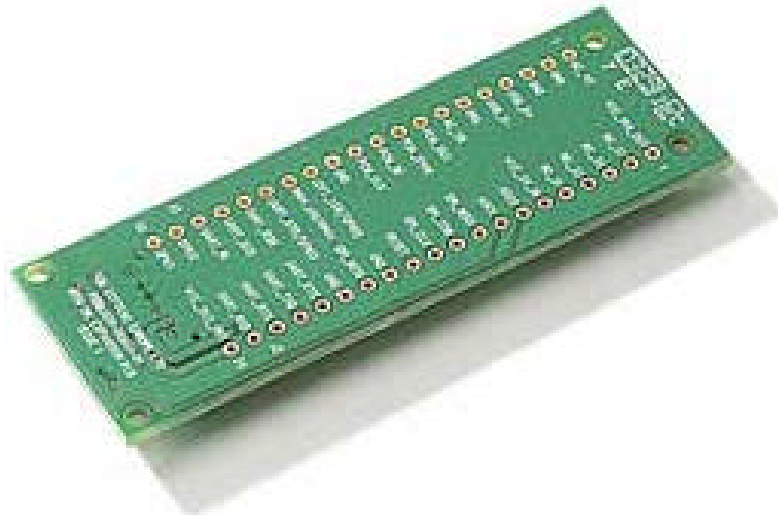


FIGURA 2.12

Ezurio Board to Board Connector (Hirose DF12C - 40 DS – 0.5V)

2.3 PRODUCTOS BLUETOOTH

La oferta de productos *Bluetooth* en el mercado es muy amplia y variada. Existen productos de muchos tipos y de diversos fabricantes. Entre los tipos de productos más destacados están: partes para telefonía celular, tales como auriculares Manos Libres, Adaptadores USB para conexión del teléfono con la computadora, etc.; reproductores de audio Mp3, adaptadores USB, tarjetas PC, tarjetas PCMCIA y productos específicos muy variados. En la Tabla 2.2 se nombran algunos de ellos.

PRODUCTO	DESCRIPCION	PRECIO
Sony Ericsson Bluetooth Headset [15]	Auricular para teléfonos celulares	\$80.00
Microsoft Bluetooth 2.0 Elite keyboard + Mouse USB [16]	Tclado y Mouse	\$68.00
Dell Axim x30 PDA, 624 mHz, 64 MB, WiFi, Bluetooth [17]	Agenda Electrónica con Bluetooth	\$225.00
DIVAGEM XM-320 BLUETOOTH MP3 PLAYER, HF, FM [18]	Mp3 Player con Bluetooth	\$150.00
V600 Motorola Bluetooth Camera Phone [19]	Teléfono Celular con Bluetooth	\$250.00

TABLA 2.2

Algunos productos Bluetooth disponibles en el mercado

2.4 DISPOSITIVO BLUETOOTH USB

Una de las maneras mas comunes de lograr una eficaz comunicación Bluetooth en una PC es la utilización de un Dispositivo Bluetooth USB disponible al publico en la mayoría de lugares de ventas de accesorios para computadoras, la cual es la opción mas común para cualquier usuario en general que desee este tipo de comunicación sin tener que ser experto en comunicaciones inalámbricas e implementación de tarjetas utilizando elementos diversos.

Un ejemplo de estos dispositivos se muestra a continuación en la siguiente figura:



FIGURA 2.13
Dispositivo Bluetooth USB

En este dispositivo ya se encuentran integrados los elementos detallados anteriormente en la implementación de la tarjeta, con excepción de que este dispositivo se comunica únicamente por medio USB.

También incluye las capas inferiores del stack Bluetooth (Firmware, Banda Base, RfComm, etc.) El software es incluido en un disco con los drivers para su utilización en la PC.

Al estar Bluetooth pensado de forma tal que, cualquier fabricante lo implemente, el grupo SIG formado por todos estos fabricantes se encarga de velar que todos los productos Bluetooth tengan las mismas consideraciones en su desarrollo siendo así que todos se puedan comunicar independientemente del fabricante que lo distribuya.

2.5 PROCEDIMIENTO DE CALIFICACIÓN BLUETOOTH

El propósito de este sub-capítulo es brindar una descripción del procedimiento que se debe seguir para calificar un producto como Producto Bluetooth. Con el propósito de minimizar o eliminar problemas de interoperatividad el “Bluetooth Qualification Program” (BQP) ha definido un grupo específico de procedimientos. Este es un requerimiento necesario antes de que un producto pueda ser calificado de acuerdo con Bluetooth.

La calificación Bluetooth es el proceso mediante el cual un fabricante demuestra conformidad con la especificación Bluetooth y el BQP es la base que establece las reglas y procedimientos de calificación. El BQP fue creado independientemente de algún tipo de aprobación administrativa obligatoria de reglamentaciones nacionales o internacionales. Un fabricante de productos con la tecnología inalámbrica Bluetooth debe suscribirse al contrato de membresía del SIG Bluetooth. Cuando el producto cumple con la especificación Bluetooth, es adicionado a la lista de productos Bluetooth. Esto otorga a la Compañía Miembro una licencia que cubre los derechos de patente de la propiedad intelectual de derechos de autor y le permite usar la marca Bluetooth en productos y actividades de mercadeo de acuerdo con el libro de la marca Bluetooth. [20]

3. SOFTWARE PARA EL HOST BLUETOOTH

El software para el “host” Bluetooth corresponde a las capas del “stack” de protocolos y utilidades Bluetooth implementadas en software e instaladas en el “host”. Un “host” puede ser cualquier sistema microprocesado programable (PCs, teléfonos celulares, “Mouse”, impresoras, teclados, sensores inalámbricos, etc.), capaz de ejecutar las líneas de código correspondientes al “stack” de protocolos para el “host”. Es necesario tener una interfaz UART o USB para comunicar el “host” y el dispositivo Bluetooth.

Son muchos los “stacks” de protocolos Bluetooth disponibles para el “host”, implementados en diversos lenguajes de programación y sobre distintas plataformas, siendo también muchas las empresas interesadas en su desarrollo. Independientemente de la plataforma o el lenguaje de programación, estos se basan en el CORE y los PERFILES de la especificación del sistema Bluetooth.

La Figura 3.1 muestra un modelo de implementación del “stack” de protocolos Bluetooth, discriminando las capas que están implementadas en el “host” y las que están en el Hardware Bluetooth (módulos, tarjetas, sistemas de desarrollo Bluetooth), como es el caso a desarrollar en esta tesis.

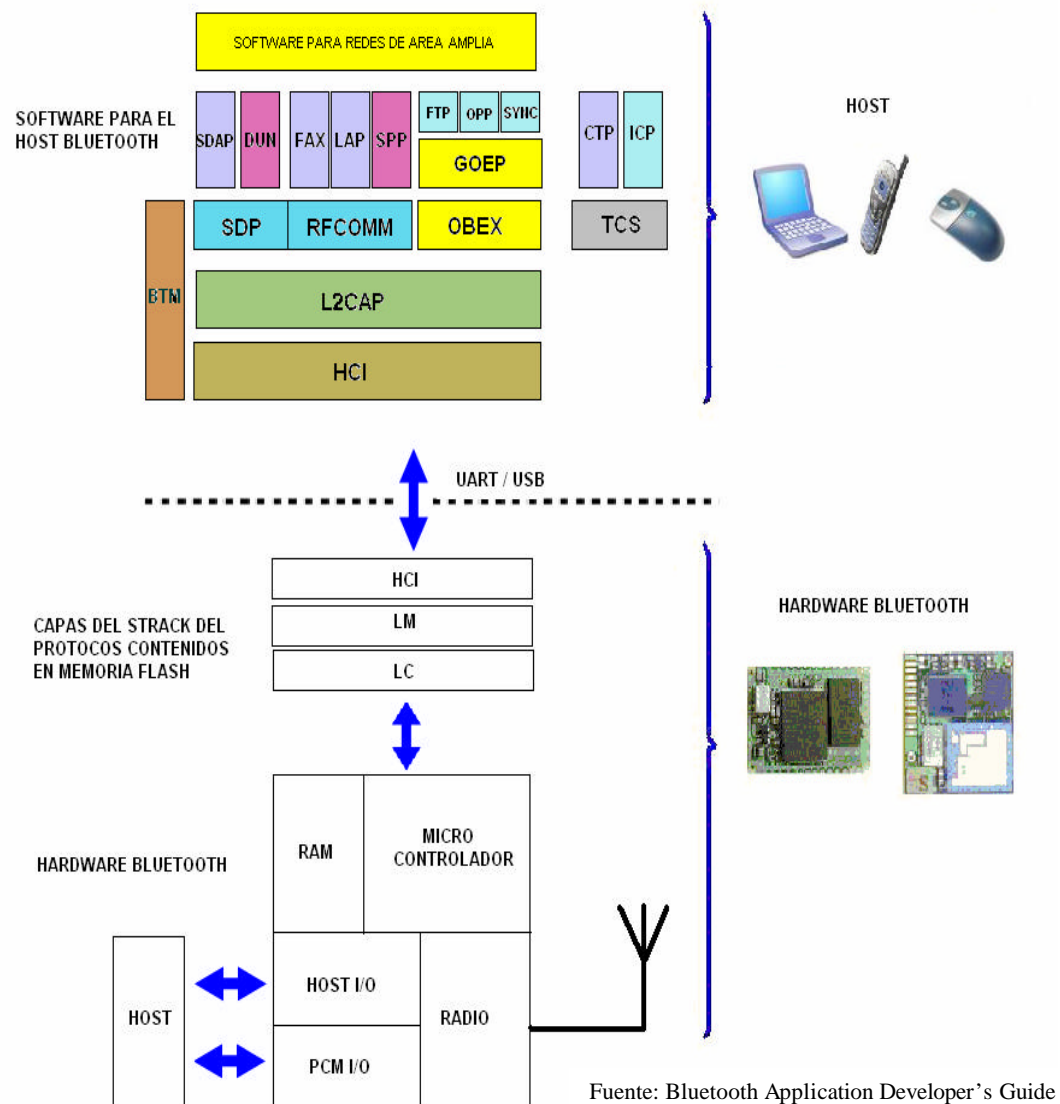


FIGURA 3.1

Modelo de implementación del protocolo Bluetooth, Host – Hardware

La Figura 3.2 describe esquemáticamente la implementación del “stack” de protocolos Bluetooth al nivel de software, firmware y hardware, especificando su ubicación ya sea en el host o en el hardware Bluetooth.

El host y el hardware Bluetooth se comunican a través del HCI (Host Controller Interface) o Interfaz Controladora de host. El firmware HCI implementa los Comandos

HCI para el hardware Bluetooth teniendo acceso a los comandos de banda base, manejador de enlace, registros de estatus del hardware, registros de control y registros de eventos.

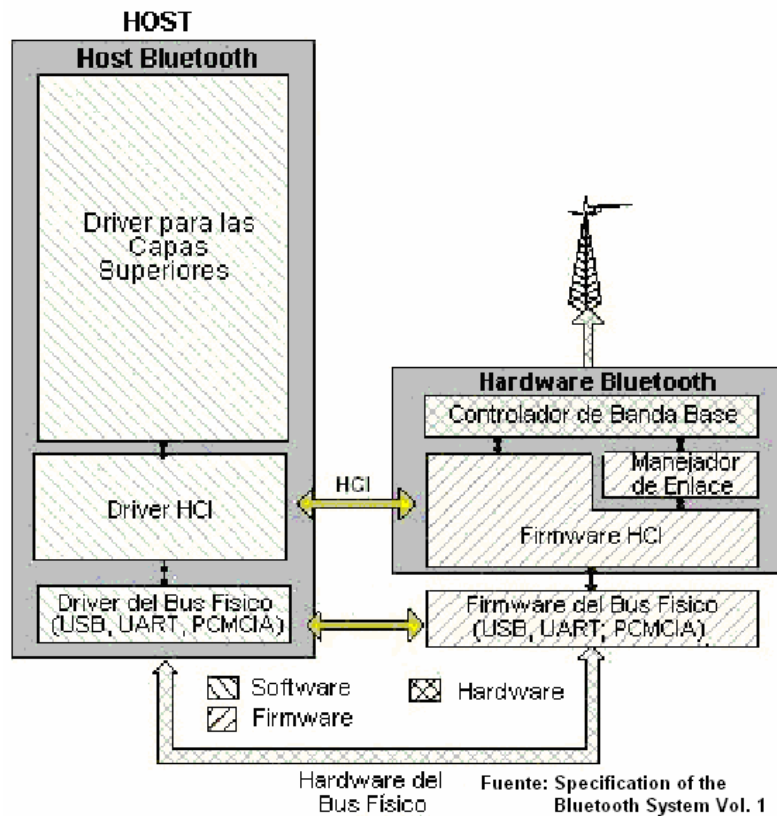


FIGURA 3.2

Modelo de Implementación del protocolo Bluetooth. Software – Firmware – Hardware

Muchas capas pueden existir entre el driver HCI ubicado en el host y el firmware HCI ubicado en el hardware Bluetooth. Estas capas intermedias, se encargan del control y transporte de datos a través de un medio físico (un bus físico ya sea USB, PC Card, RS232, u otro), para que se establezca una comunicación transparente entre el driver HCI y el firmware HCI, permitiendo el intercambio de datos y comandos entre estos dos.

El host recibirá notificaciones asíncronas de los eventos HCI independientemente de la capa de control de transporte del host que se esté usando. Los eventos HCI son usados para notificar al host cuando algo ocurre.

3.1 ARQUITECTURAS DE INTEGRACIÓN DEL “STACK” DE PROTOCOLOS BLUETOOTH

La mayor o menor integración del “stack” de protocolos Bluetooth en un sistema, depende del tipo de producto que se está desarrollando; de esta manera y como se puede observar en la Figura 3.3, podrían implementarse tres diferentes arquitecturas. Cabe resaltar que no son las únicas.

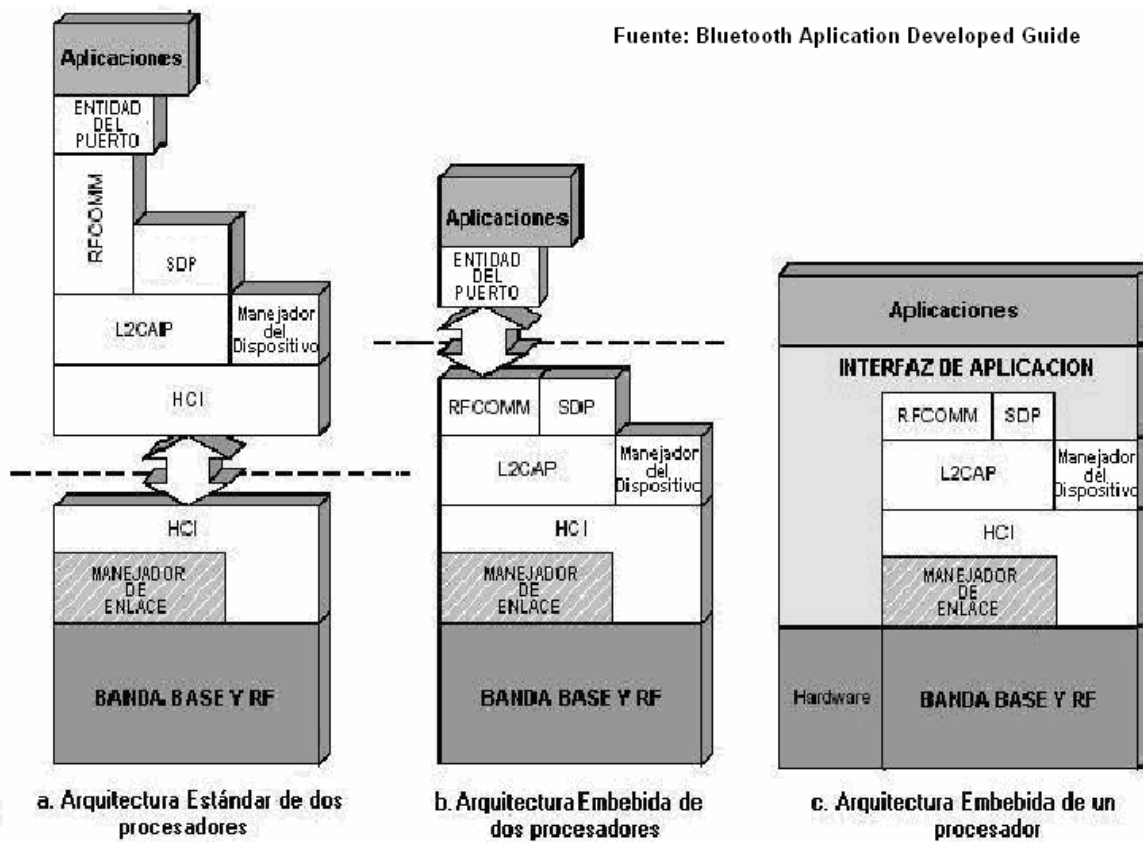


FIGURA 3.3

Tres modelos de Integración del stack de protocolos Bluetooth

La Figura 3.3a describe una solución estándar de dos procesadores, en donde la división entre las capas altas de protocolo y las bajas toma lugar en el HCI, esto asume que el “host” es en general algún tipo de plataforma computacional cuyas capacidades de comunicación son “ilimitadas”. Este es el caso de las computadoras personales.

En la Figura 3.3b se puede observar una arquitectura embebida de dos procesadores, la cual permite que los productos puedan diseñarse para incorporar Bluetooth donde el “host” es un recurso limitado y no es capaz de soportar la adición de una funcionalidad Bluetooth. Un ejemplo de este caso es un teléfono móvil; no todos los teléfonos móviles tienen los suficientes recursos para implementar pilas de protocolo adicionales.

La Figura 3.3c muestra una arquitectura embebida de un procesador; este es el caso de un producto que implementa el “stack” de protocolos Bluetooth completo en un solo chip. Un ejemplo es un “manos libres” inalámbrico, sin embargo, el modelo aplica a cualquier dispositivo inalámbrico pequeño.

3.2 PRINCIPALES SOFTWARE PARA EL HOST

Son muchas las compañías que trabajan en el desarrollo de “stacks” de protocolos Bluetooth para el “host”, en especial, los fabricantes de módulos Bluetooth, quienes ofrecen productos de prueba y “kits” de desarrollo que incluyen software para el “host”, cuya finalidad es facilitar y agilizar el desarrollo de aplicaciones. Este software no es usualmente de libre distribución, es decir que su licencia tiene un costo, el cual, en la mayoría de los casos, está incluido en el costo del “kit” de desarrollo.

Sin embargo, los fabricantes de módulos Bluetooth no son los únicos que ofrecen software para el “host”, hay muchas otras empresas y universidades interesadas en su desarrollo, razón por la cual es posible encontrar software demostrativo y de libre distribución con licencia pública GPL (GNU “Public License”).

En esta tesis se trata únicamente el software para sistemas cuyo “host” corresponde a un PC, es decir, no se cubre software para sistemas embebidos. Los requerimientos tales como sistema operativo, capacidad de memoria, entre otros, son propios de cada software.

3.2.1 Software Bluetooth con propiedad de licencia.

Debido a que muchas de las características de estos diferentes tipos de software son muy parecidas, se da una breve descripción de los más representativos. [21]

- ✓ “Bluetooth Host Stack” Software para el “host” desarrollado por Ericsson, el cual implementa las siguientes capas del “stack” de protocolos: “driver” HCI, L2CAP, RFCOMM, SDP y opcionalmente OBEX y TCS. Este “stack” está escrito en ANSI C, y es independiente del ambiente de desarrollo (compilador, “debugger”, “linker”, etc.); esto sumado al concepto de Sistema Operativo Virtual, hace que este “stack”, sea adaptable tanto a sistemas operativos para sistemas embebidos como a los sistemas operativos estándar tales como Linux y Windows. Las siguientes son

herramientas de software disponibles para Windows, basadas en el Bluetooth Host “Stack” de Ericsson: “Bluetooth PC reference Stack”, “Bluetooth Development Kit” y “Bluetooth Application Tool Kit”.

- ✓ “Bluetooth Software Suite” Desarrollado por “Digianswer” A/S, este software escrito para Microsoft Windows además de brindar un API (Interfaz de Programa de Aplicación) de fácil uso, soporta los siguientes perfiles: acceso general, servicio de descubrimiento de aplicaciones, puerto serial, red de acceso conmutado, fax, acceso a LAN, OBEX, Object Push, transferencia de archivos y Ethernet.
- ✓ “BlueStack” Software desarrollado por MEZOE, una división de Cambridge Consultants Ltd. Escrito en lenguaje C, implementa al igual que los dos anteriores, un driver HCI y las capas superiores del protocolo Bluetooth. Son ofrecidos varios productos basados en el BlueStack para Microsoft Windows: este es el caso de StackPrimer, Proto Developer e Interface Express.
- ✓ “BISM2 EZURIO” El software esta basado en comandos AT que son los comandos utilizados en la comunicación con MODEM , estos, permiten con una simple serie de comandos, realizar funciones complejas, para el manejo de las comunicaciones Bluetooth de una manera sencilla, combinando el uso de comandos y registros, con el fin de administrar Perfiles.

La Tabla 3.1A y 3.1B muestran una recopilación de algunos productos con propiedad de licencia que implementan el “stack” de protocolos Bluetooth para el “host”; se indica además la empresa que lo desarrolla, herramientas de desarrollo para cada “stack” y el sistema operativo sobre el cual se ejecuta.

NOMBRE DEL STACK	FABRICANTE	HERRAMIENTAS DE DESARROLLO	SISTEMA OPERATIVO SOPORTADO POR LA HERRAMIENTA
Bluetooth Host Stack	Ericsson	<ul style="list-style-type: none"> ✓ Bluetooth PC reference Stack ✓ Bluetooth Development Kit ✓ Bluetooth Application ✓ Tool Kit 	Microsoft Windows
BISM2	EZURIO	<ul style="list-style-type: none"> ✓ Bluetooth Serial Module ✓ Referente Manual (Comandos AT) ✓ Blue2i 	Microsoft Windows
Bluetooth Software Suite	Digianswer A/S	<ul style="list-style-type: none"> ✓ Digianswer Bluetooth Software Suite ✓ Digianswer Bluetooth HCI ✓ Digianswer Bluetooth Statistics ✓ Digianswer Bluetooth HCI Router ✓ Digianswer HCI Test Application ✓ Digianswer API 	Microsoft Windows
BlueStack	Mezoe	<ul style="list-style-type: none"> ✓ Proto Developer Tool 	Microsoft Windows
Stack Partner Program	Nacional Semiconductor	Stack Partner Program	<ul style="list-style-type: none"> ✓ Microsoft Win98, ME, 2000 y XP ✓ Microsoft WinCE ✓ Linux

TABLA 3.1 A
Software Bluetooth para el Host con propiedad de licencia

NOMBRE DEL STACK	FABRICANTE	HERRAMIENTAS DE DESARROLLO	SISTEMA OPERATIVO SOPORTADO POR LA HERRAMIENTA
Rappore Technologies' Software Developers Kit	Rappore Technologies	Software Developers Kit	No especificado Código fuente desarrollado en lenguaje C, C++ y Java
Socket Bluetooth Software	Socket	Bluetooth Connection Kit	✓ Windows Powered Pocket PC
BlueStack Software Development Kit	Cambridge Silicon Radio	BlueStack Software Development Kit	✓ Windows CE Windows NT
Bluefrog Demonstration FrogStack	Reselec ag Bluefrog	Bluefrog Demonstration FrogStack	✓ Net OS 3.0 ✓ Linux

TABLA 3.1B

Software Bluetooth para el Host con propiedad de licencia

3.2.2 Software Bluetooth con licencia pública (GPL)

En esta sección se tratan las principales características de los tipos de productos de libre distribución con licencia GPL desarrollados para Linux y disponibles en Internet. Los principales son: "OpenBT", "BlueDrekar", "Bluez", "Affix".

- ✓ "OpenBT" Axis Communications Inc. desarrolló, en primera instancia, un "driver" Bluetooth para Linux llamado OpenBT, el cual puede ser usado tanto en ambientes eLinux (Linux Embebido) como en Linux para PC. Este fue el primer "stack" de protocolos disponible, siendo ahora un proyecto de código fuente abierto, desarrollado en un "kernel" de Linux 2.0.33, de tal manera que no debe presentar problemas de funcionamiento en versiones posteriores del "kernel".

Implementa el perfil de LAN mediante enlaces PPP sobre RFCOMM y además una forma simple del perfil de descubrimiento de servicios (SDPP).

El código fuente en sus versiones para PC y eLinux, se encuentra disponible en el portal de Internet de “Sourceforge”. [22]

- ✓ “BlueDrekar” El “driver” Bluetooth de IBM, disponible bajo licencia GPL, implementa la capa de transporte UART de la Interfaz Controladora del “host” (HCI). Este código sirve de referencia para escribir “drivers” para otras capas de transporte HCI, como USB. El “BlueDrekar middleware” es una implementación de referencia de las capas altas del protocolo Bluetooth, disponible bajo licencia alpha Works.

Según el fabricante, este código ha sido desarrollado y probado en PCs con procesadores de arquitectura i486 o superior, corriendo un “kernel” de Linux 2.2.12-20 (Red Hat 6.1), 2.2.14-5.0 (Red Hat 6.2), 2.2.16-22 (Red Hat 7). Han sido utilizados para realizar las pruebas de desempeño los módulos Bluetooth de Ericsson con la capa de transporte UART y firmware versión P7C (v1.0A) y P3E (v1.0B). [23]

- ✓ “Affix” Este “stack” de protocolo para Linux fue desarrollado por el Centro de Investigación de Nokia en Helsinki, Finlandia. El software Bluetooth de “Affix” trabaja sobre plataformas Intel, ARM (iPaq, ARM9 y otras) y PowerPC (iMac). Las últimas versiones del software están disponibles en el sitio web de “Affix” [24]

- ✓ “Bluez” Este es el stack de protocolos Bluetooth oficial de Linux, el cual es parte del kernel 2.4 y posteriores. “Bluez” brinda a sus usuarios la capacidad de comunicación con dispositivos Bluetooth, entre los que se tienen adaptadores USB, teléfonos móviles y puntos de acceso entre otros, además de conexión inalámbrica entre dos o más computadoras. Bluez consta del “core” HCI, “drivers” HCI para UART, USB y emulador de HCI, módulo del protocolo L2CAP y utilidades de configuración y prueba. [25].

3.3 CRITERIOS DE SELECCION

3.3.1 DEFINIR CRITERIOS DE SELECCION.

Se toma en cuenta que se siguió el modelo básico para la implementación de una tarjeta inalámbrica Bluetooth. Los elementos principales como antena, dispositivo Bluetooth, “Transceiver”, etc. no se encuentran disponibles en nuestro país, por lo que hubo que buscar un lugar en el extranjero en el que se podían comprar.

- Por su precio: No solamente hay que fijarse en el precio del dispositivo, si no que también en el valor de el envío a nuestro país.
- Por su accesibilidad: Que contenga material de carácter informativo para la implementación de la tarjeta, por ejemplo “datasheets”.
- Por su diseño / tamaño: Que sea de tamaño reducido pero no exageradamente pequeño ya que esto dificultaría su manejo. Por tanto el diseño que sea accesible en cuanto a conexiones.
- Por marca: Que sea preferiblemente de Fabricantes conocidos y de larga trayectoria en el campo de la electrónica.

3.3.2 SELECCIONAR ELEMENTOS A UTILIZAR.

Elemento	Elemento Utilizado	Elemento Similar en el Mercado	Criterio de Selección
Modulo Bluetooth	EZURIO BISM2	National LMX9820A [30] ERICSSON ROK 101008	Por Accesibilidad, Precio y Disponibilidad.
Transceiver RS232	MAXIM MAX3232	ADM101E - Craft Port ™ [32]	Por Precio de envío y Funcionabilidad y venta unitaria
Regulador de voltaje	National LP2987	Philips TDA3663 [33]	Por Fabricante, por Precio de envío, se compro junto al Transceiver.

4.0 BLUETOOTH TESIS 2005 (BT 2005)

4.1 INTRODUCCION

En este capítulo se explica el funcionamiento del software creado para la demostración de una comunicación Bluetooth.

El Software Bluetooth es una aplicación que permite al usuario trabajar en un ambiente gráfico (GUI Interfaz Gráfica de Usuario) para el intercambio de archivos por medio del uso de el Perfil de Transferencia de Archivo (ver sección 1.8.12) que se encuentra dentro del Perfil de Puerto Serial (ver sección 1.8.2) del Stack de Protocolos Bluetooth.

Este software fue diseñado en Microsoft Visual Basic [27], ya que es un ambiente familiarizado con Windows y nos permite elaborar nuestra aplicación con la mejor funcionalidad y estética posible.

4.2 INSTALACION DEL SOFTWARE

A continuación se detallan los pasos para la instalación del Software Bluetooth.

Si la opción de instalación no empieza automáticamente al introducir el disco del software en la PC, se seguirá el siguiente paso:

- Ejecutar el archivo Bluetooth.msi como el de la figura 4.1



FIGURA 4.1

Archivo ejecutable de instalación de Bluetooth tesis 2005

- Con el cual aparecerá el asistente de instalación de Windows mostrado en la figura 4.2

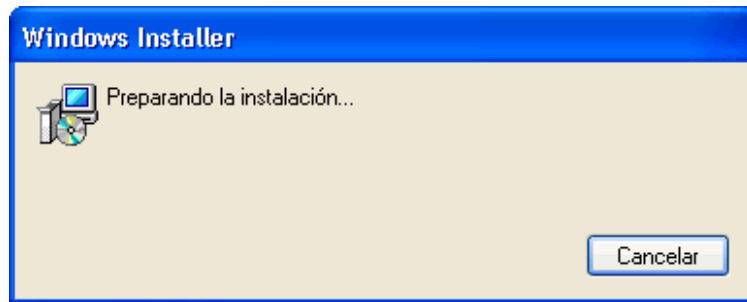


FIGURA 4.2

Indicación de Instalación en curso.

- Presionar el botón NEXT de la configuración inicial (ver figura 4.3).

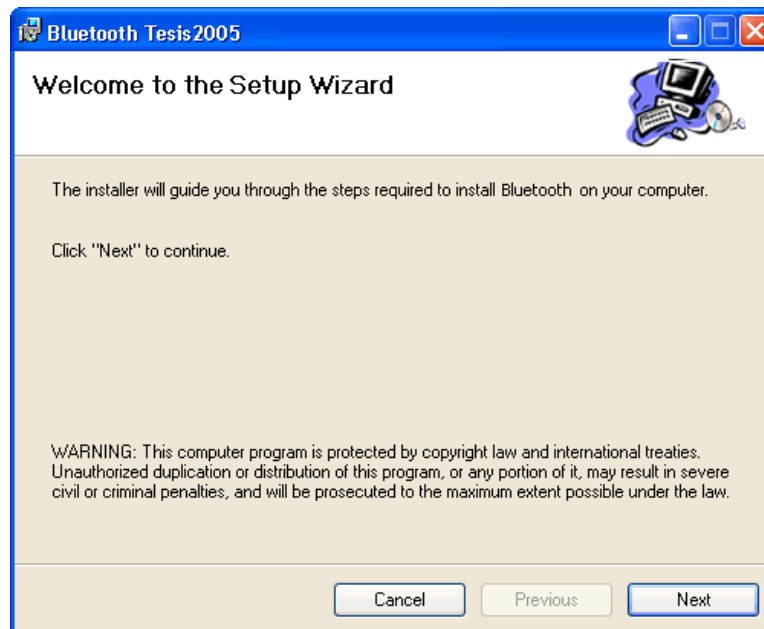


FIGURA 4.3

Configuración inicial de Instalación Bluetooth tesis 2005.

- Seleccionar la ubicación donde se instalará el software bluetooth y luego presionar NEXT. (ver figura 4.4)

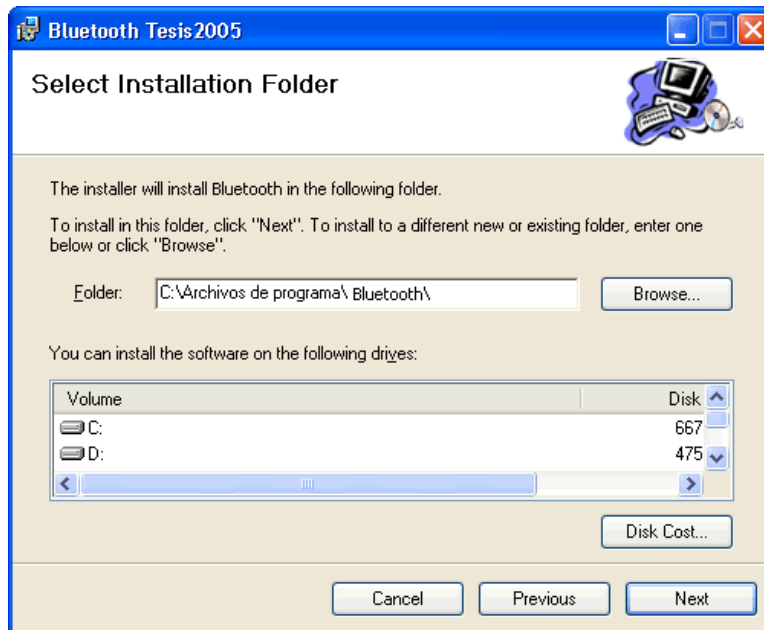


FIGURA 4.4

Segunda parte de Instalación de Bluetooth Tesis 2005

- Presionar el botón Next para iniciar el proceso de instalación. (ver figura 4.5 y 4.6)

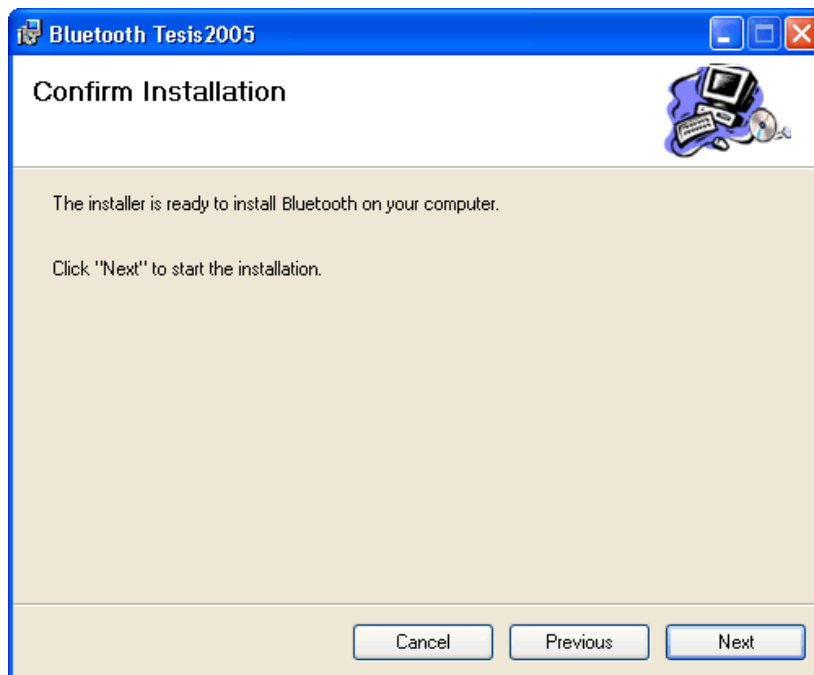


FIGURA 4.5

Confirmación de instalación

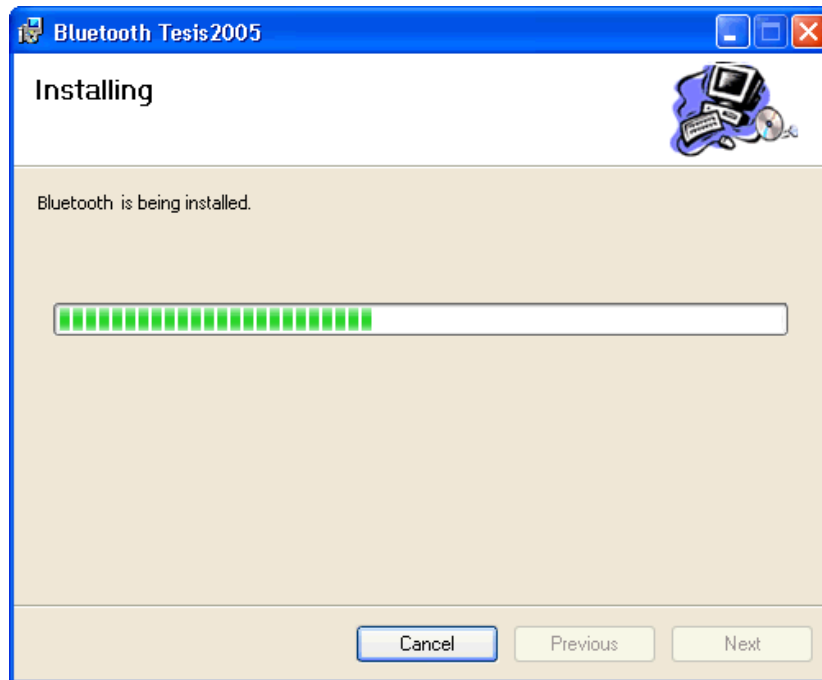


FIGURA 4.6

Progreso de Instalación

- La instalación ha sido completada presionar el botón Close (ver figura 4.7)

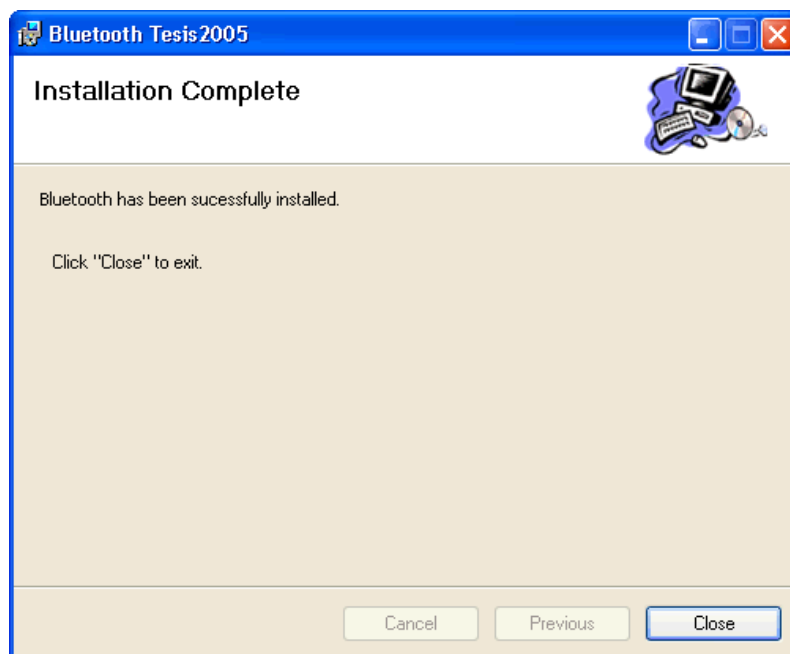


FIGURA 4.7

Instalación Finalizada

4.3 DESCRIPCION GENERAL DE LA INTERFAZ GRAFICA DE USUARIO DEL SOFTWARE BLUETOOTH TESIS 2005

El Software Bluetooth, llamado en adelante Bluetooth Tesis 2005 como se muestra en la figura 4.8, brinda el entorno grafico de usuario (GUI), el cual permite la búsqueda (“Inquiry”) de un dispositivo bluetooth (1) que se encuentre dentro del alcance de cobertura en la piconet, además almacena información de los dispositivos encontrados, así como también, establece a estos dispositivos como de confianza, lo que permite un acceso directo al ellos, sin necesidad de realizar una vinculación o “Pairing”.

Una vez el cliente esté vinculado con el servidor, el sistema muestra el nombre del servidor (2) y los archivos contenidos en la carpeta compartida del directorio raíz de intercambio Bluetooth del servidor (3). Por otra parte, se muestra un emulador del explorador de Windows de mi PC, que permite navegar en los archivos del cliente (4). Ya en esta etapa se puede realizar la transferencia de los archivos de un host (cliente) hacia el otro host (Server) arrastrando el archivo desde el cliente hacia el servidor y viceversa o también utilizando el menú contextual

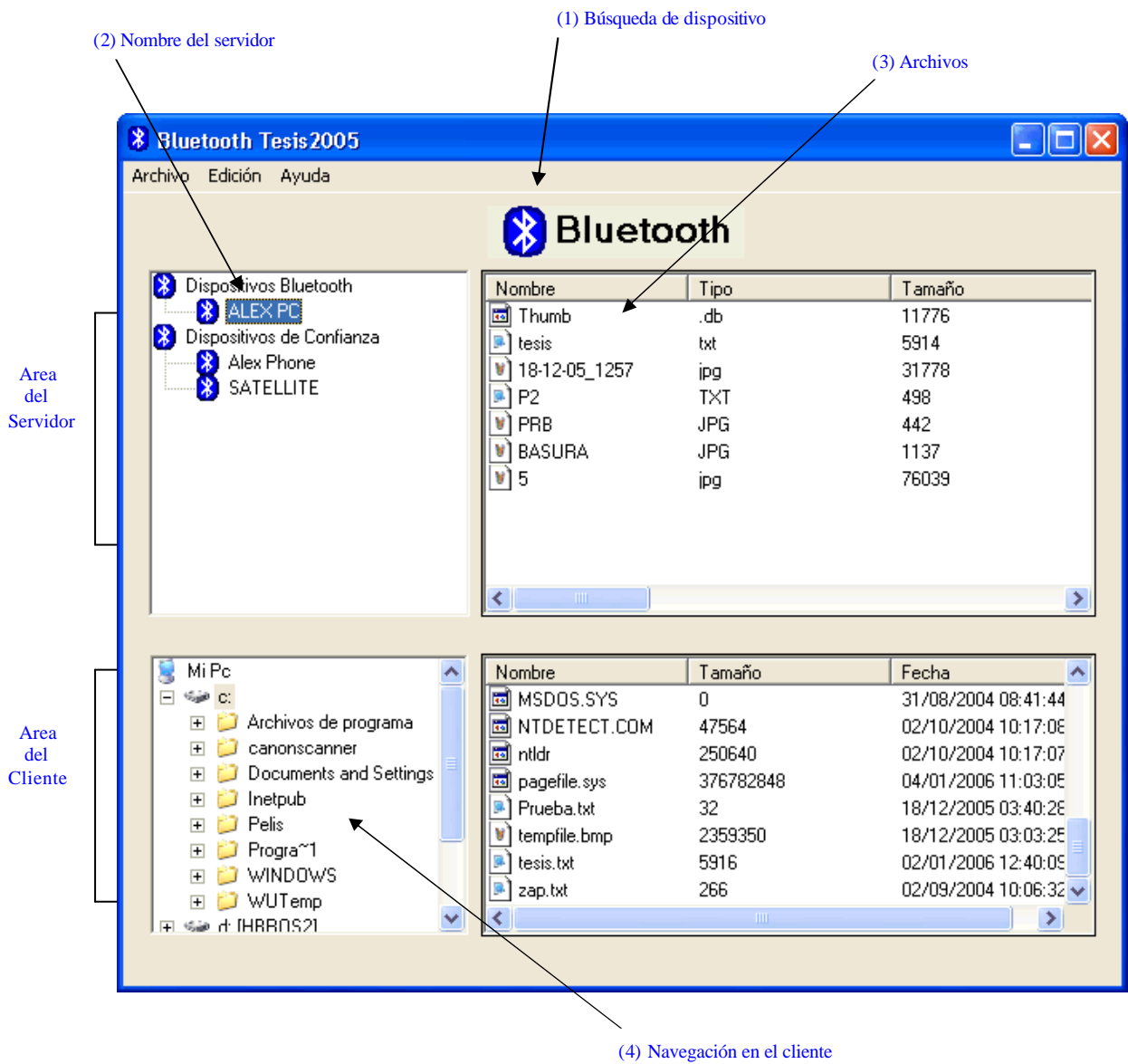


FIGURA 4.8
Objetos de Bluetooth Tesis 2005

4.4 PASOS A REALIZAR PARA UNA CONEXIÓN FTP

Para realizar la comunicación con un dispositivo, el cliente y el servidor utilizan el perfil “GAP” (Perfil de Acceso General), el cual brinda la comunicación entre dispositivos, proporcionando los estados de búsqueda (Inquiry), con el fin de vincular los dispositivos, es decir, sincronizar los relojes (maestro-esclavo), compartir información acerca de cual es la dirección física, el nombre del dispositivo Bluetooth y la clase de dispositivo entre otros.

- Presionar en la ventana principal el símbolo Bluetooth que muestra la figura 4.9

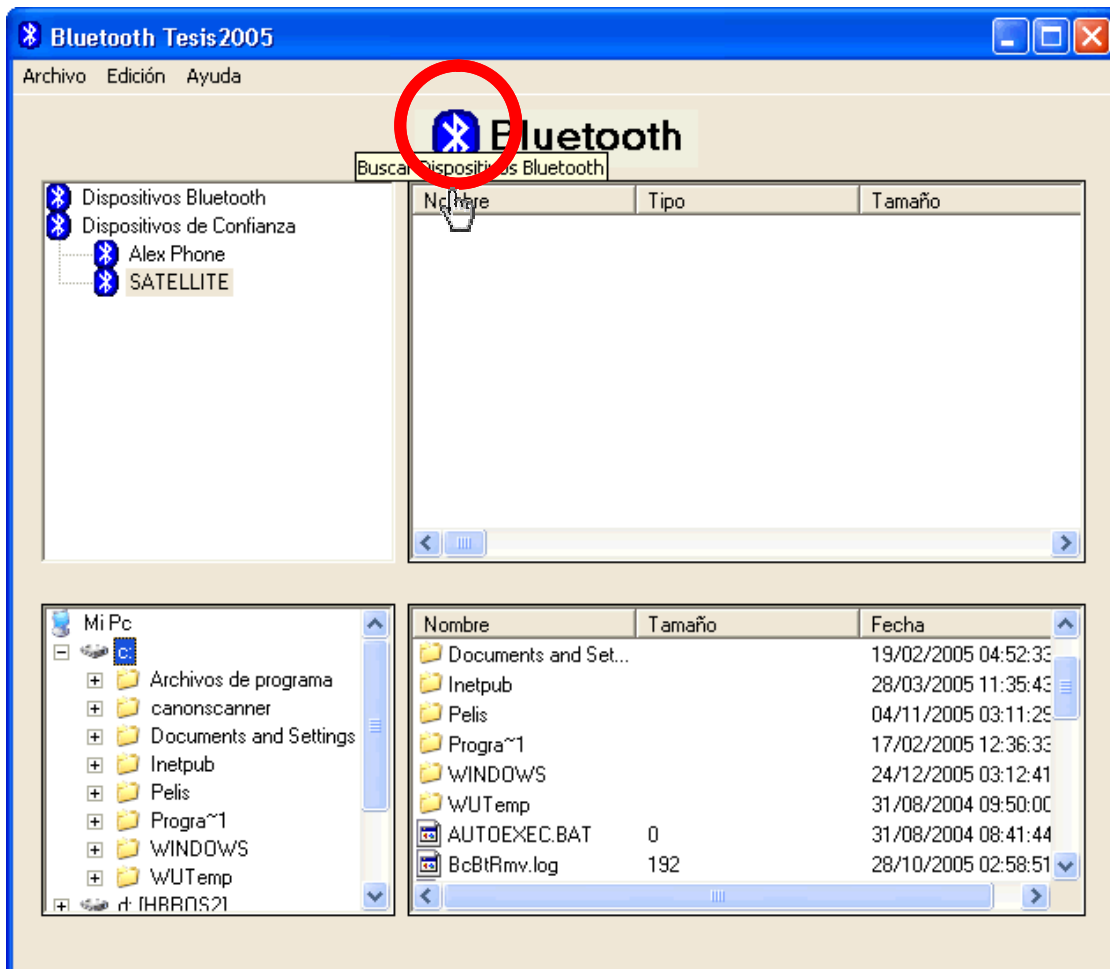


FIGURA 4.9

Búsqueda de dispositivos Bluetooth

- Se desplegarán los dispositivos encontrados, tanto los antiguos como los nuevos, en el caso que sean nuevos, se ubicaran dentro del grupo “dispositivos Bluetooth” y en el caso que ya hayan sido escogidos como dispositivos de confianza [26] dentro del grupo “Dispositivos de confianza” los que se encuentren dentro de este último ya no necesitaran pedirse la llave de seguridad (PIN) cada vez que realicen una transferencia de archivos, los archivos que se encuentren dentro de la carpeta de documentos compartidos Bluetooth del dispositivo seleccionado se desplegarán en la ventana paralela, para agregar un dispositivo nuevo como dispositivo de confianza, se hace “clic” derecho sobre el dispositivo que se quiera agregar y se selecciona “Agregar a dispositivos de confianza” y listo, como se observa en la figura.4.10.

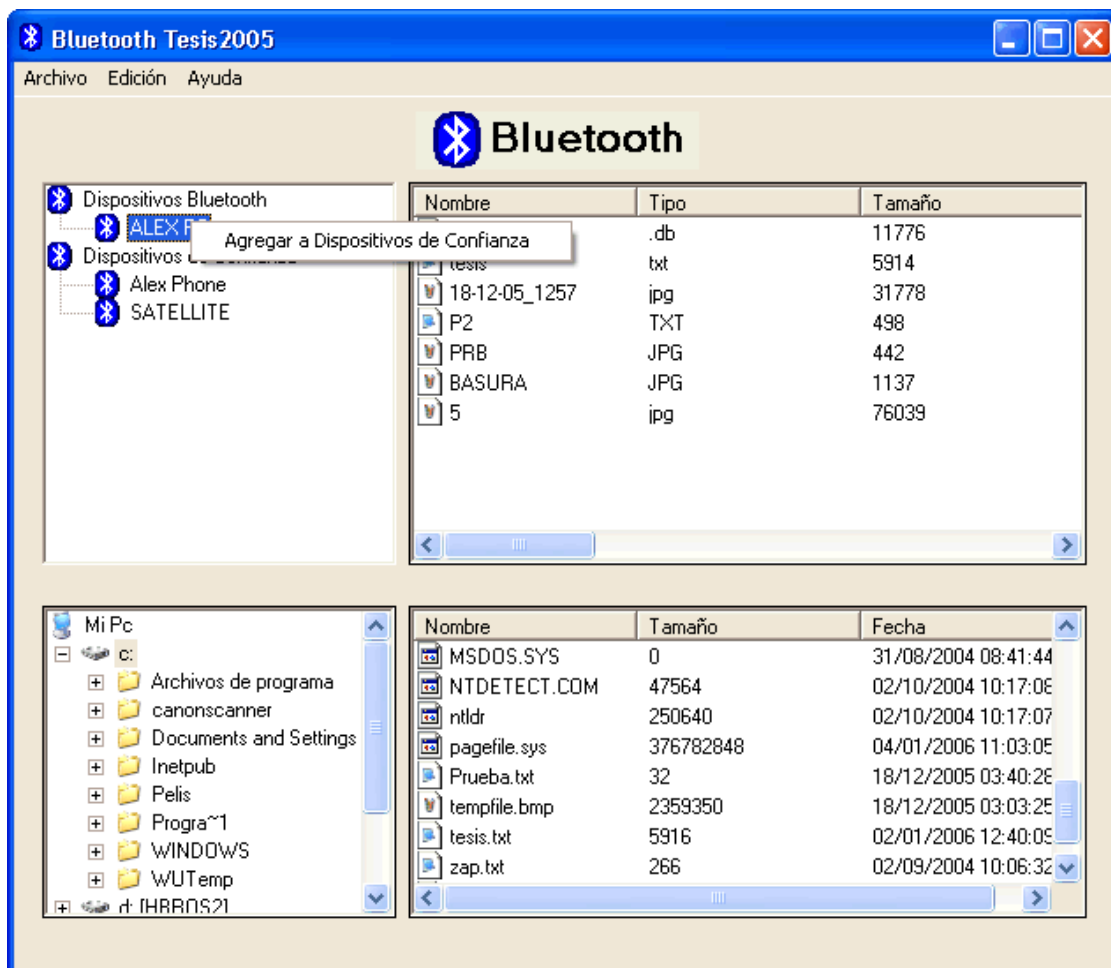


FIGURA 4.10

Agregar a dispositivo de confianza

- Para vincular un dispositivo nuevo basta con hacer “clic” en el dispositivo que se quiera vincular y preguntará a este si quiere vincular, al aceptar aparecerá una ventana en los 2 dispositivos pidiendo un PIN (Personal Identification Number) que tiene que ser igual para ambos dispositivos, después de esto muestra un mensaje de PIN correcto y despliega los archivos del directorio raíz del servidor FTP. Para quitar un contacto de los dispositivos de confianza basta con hacer “clic” derecho sobre el dispositivo a quitar y seleccionar eliminar dispositivos de confianza, ver la figura 4.11

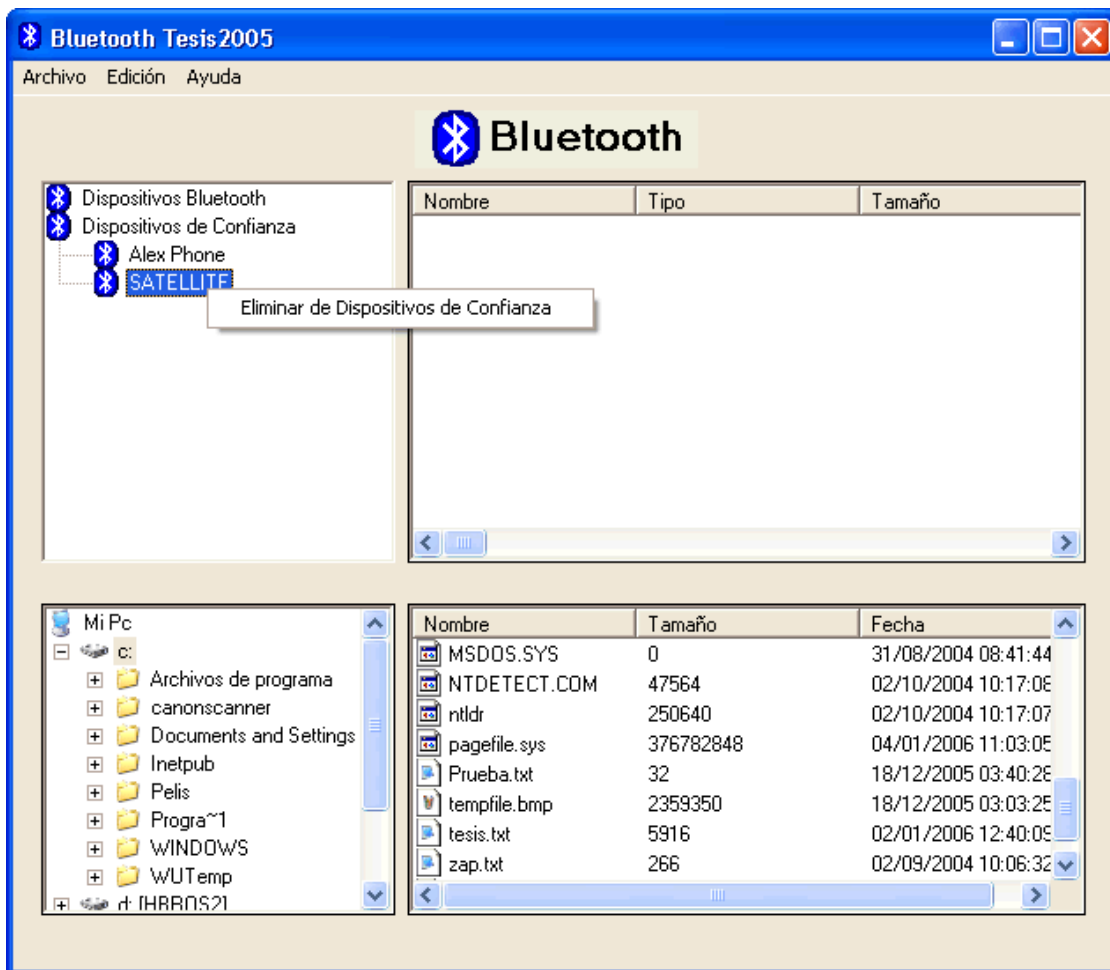


FIGURA 4.11

Eliminar dispositivo de confianza

En la figura 4.12 se muestra el flujograma para realizar la conexión con un dispositivo

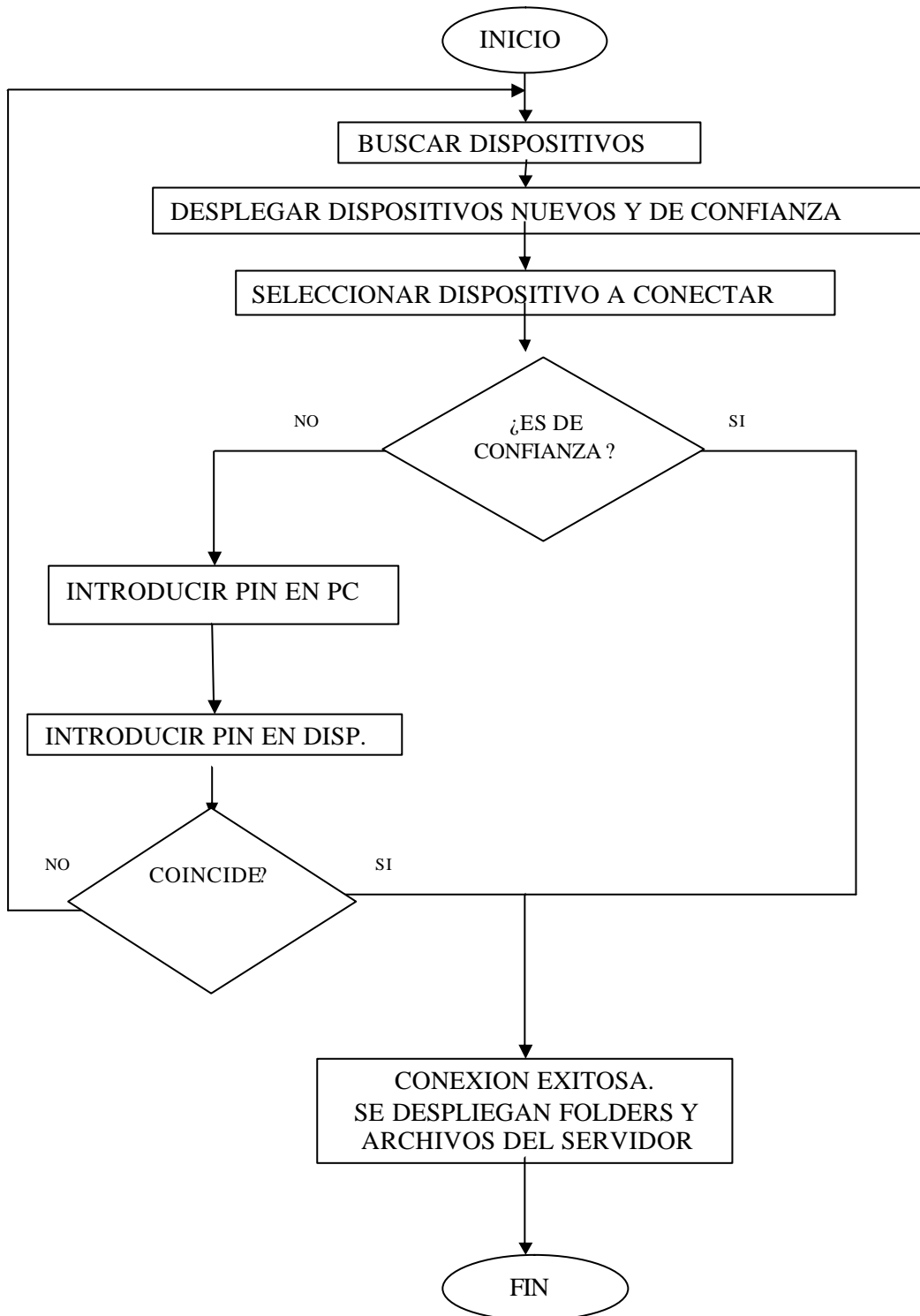


FIGURA 4.12

Diagrama de flujo del proceso de conexión

4.5 PASOS A REALIZAR PARA TRANSFERIR UN ARCHIVO AL SERVIDOR

- Haber realizado el proceso de vinculación.
- Realizar la conexión con el dispositivo servidor.
- Seleccionar navegando con el explorador, el archivo a transferir desde el cliente (la PC), luego arrastrar hacia la ventana del servidor o seleccionar “Enviar archivo del menú principal”.
- Se realiza la transferencia completando la barra de progreso ubicada en la parte inferior de la ventana como se muestra en la figura 4.13, y listo.

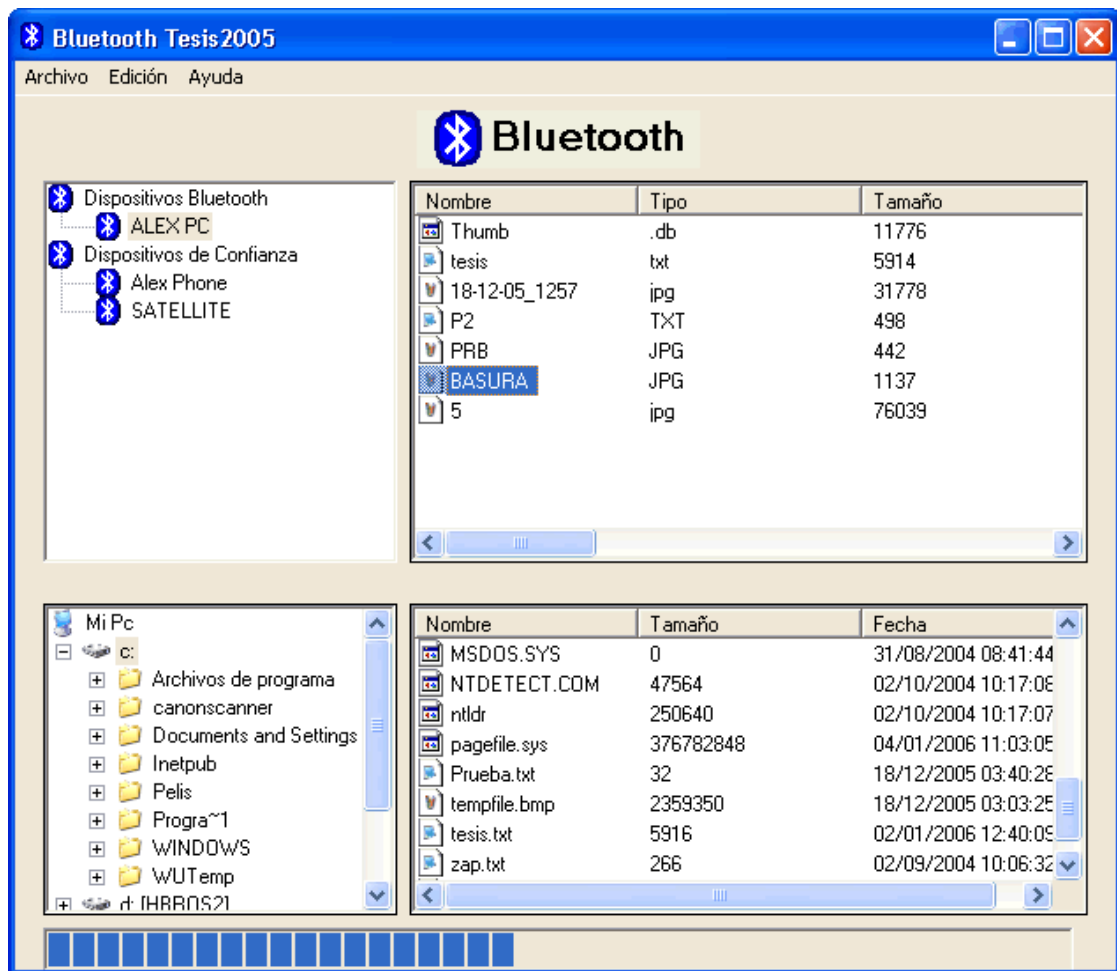


FIGURA 4.13

Transferencia de un archivo.

El flujograma para realizar la transferencia de un archivo al servidor se muestra en la figura 4.14

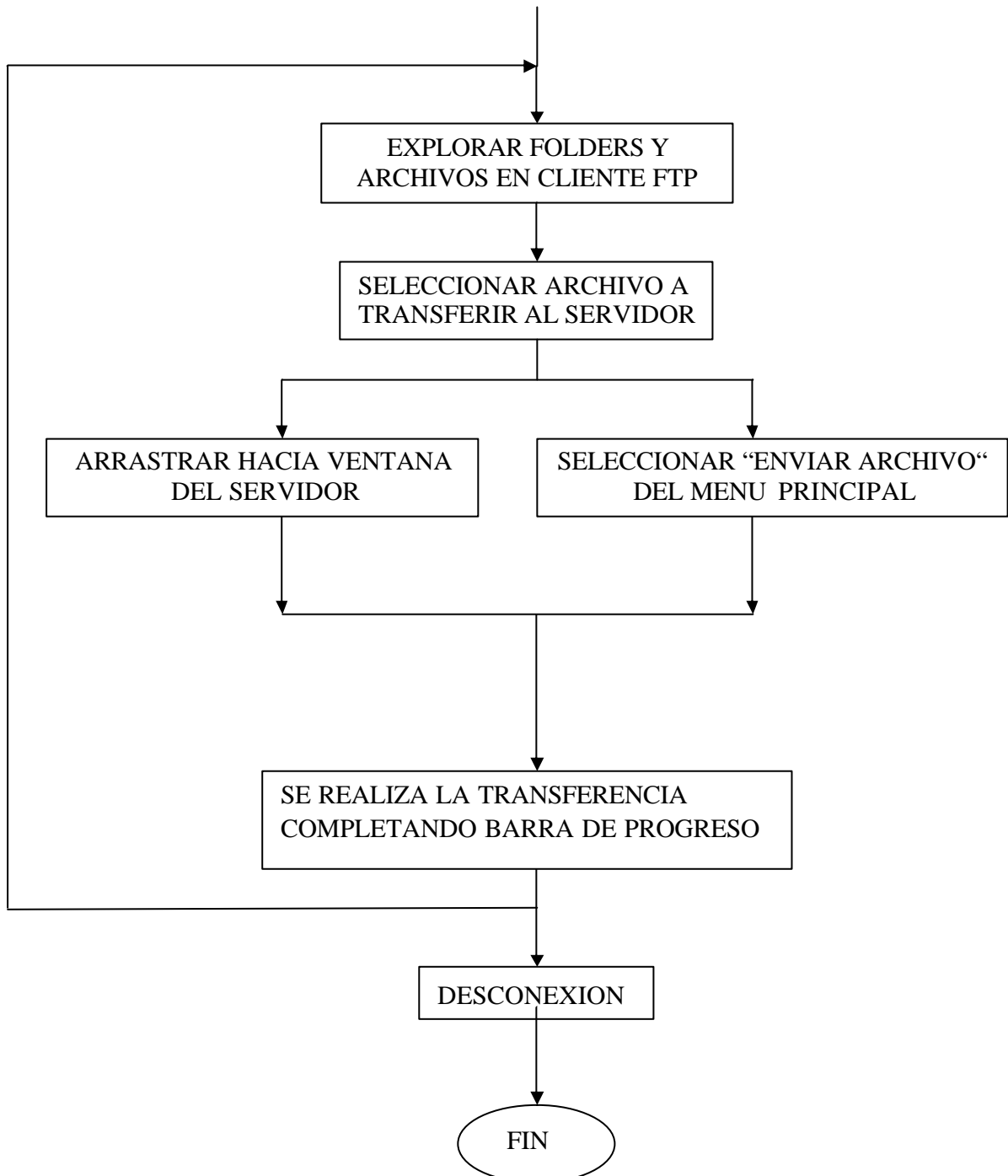


FIGURA 4.14

Diagrama de flujo de la Transferencia hacia el Servidor

4.6 PASOS PARA TRANSFERIR UN ARCHIVO DESDE SERVIDOR HACIA LA PC

- Haber realizado el proceso de vinculación.
- Realizar la conexión con el dispositivo servidor
- Explorar Fólderes y archivos en servidor FTP
- Seleccionar el archivo a recibir del servidor.
- Seleccionar la carpeta destino del Cliente.
- Arrastrar hacia la ventana del cliente o seleccionar “copiar archivo” del menú principal como lo muestra la figura 4.15.
- ❖ Si lo que se desea es eliminar el archivo, seleccionar “Eliminar archivo en dispositivo Bluetooth” después de seleccionar el archivo

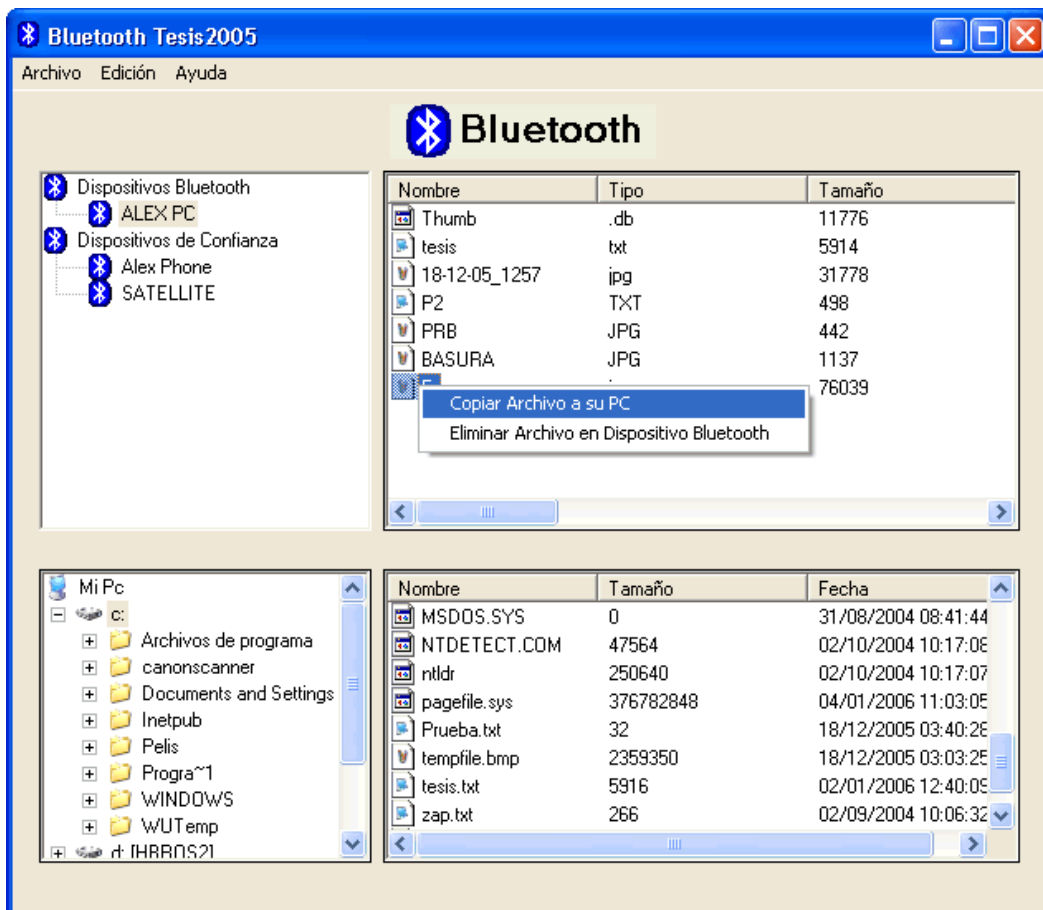


FIGURA 4.15

Copia de un archivo desde el servidor

La figura 4.16 denota el proceso para realizar una transferencia desde el servidor.

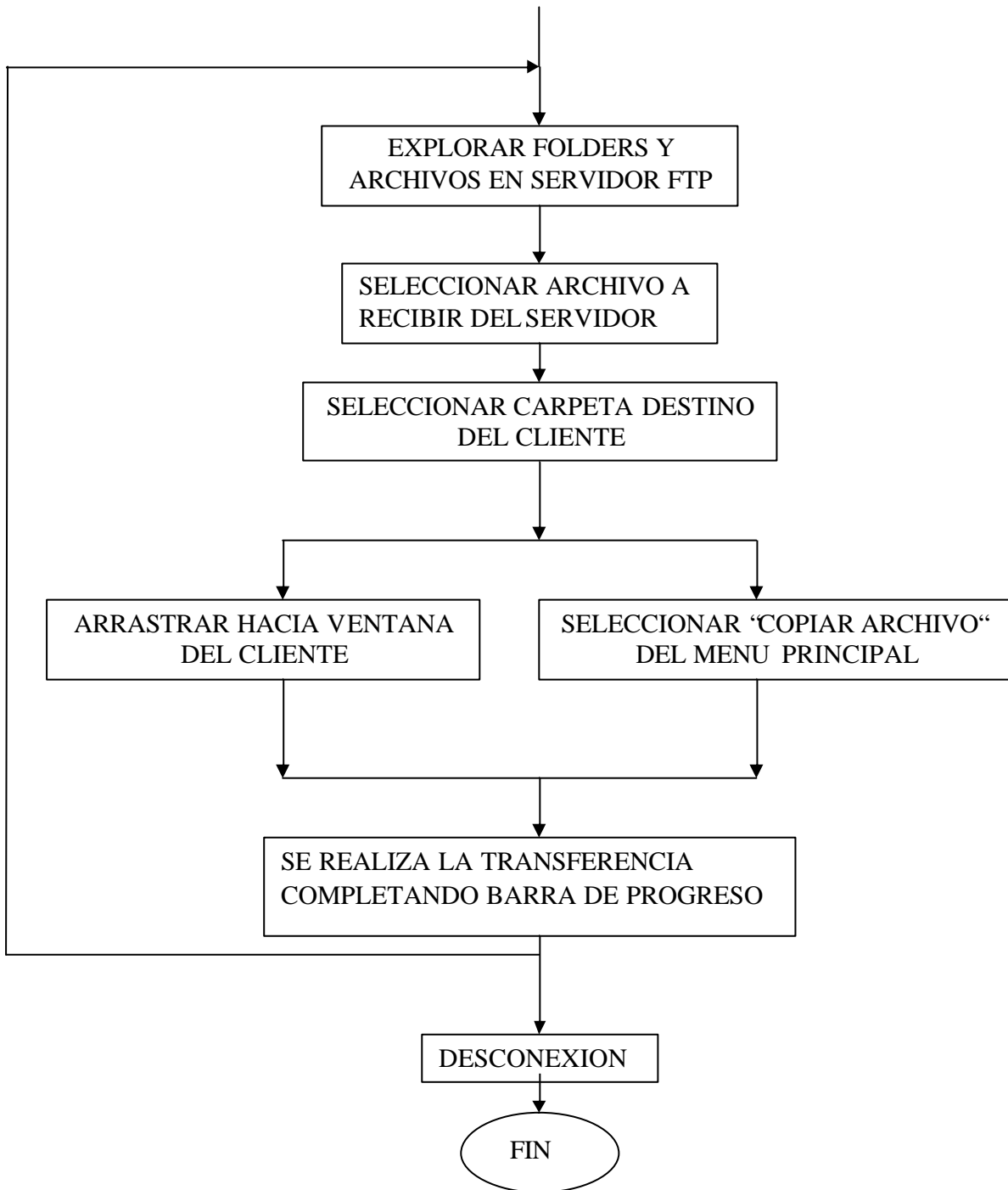


FIGURA 4.16

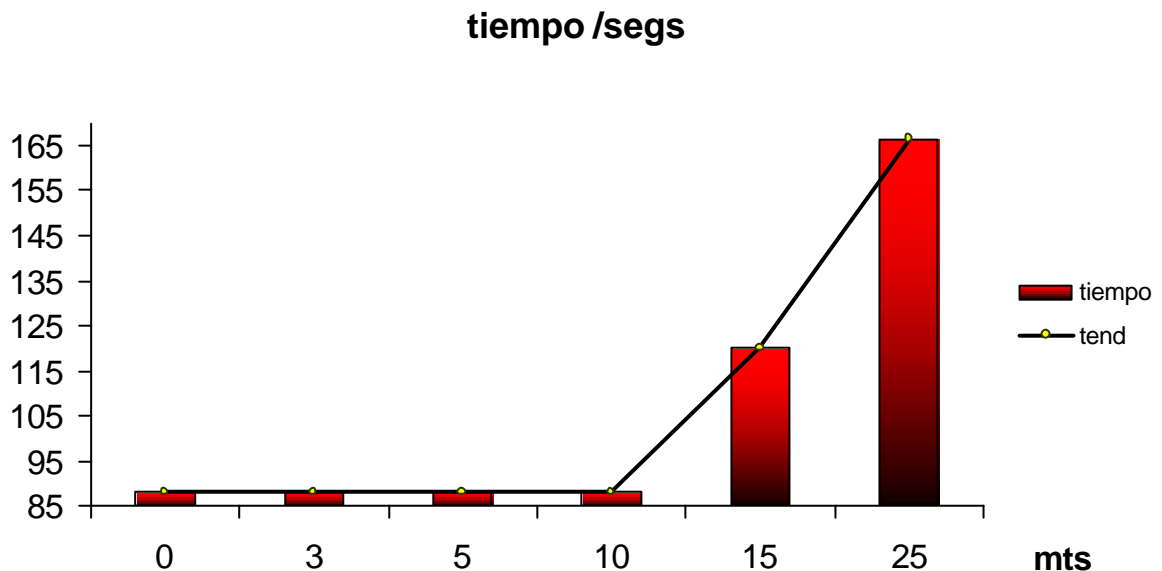
Diagrama de flujo de la Transferencia desde el Servidor

5.0 PRUEBAS DE DESEMPEÑO Y APLICACIÓN PARA LA TARJETA BLUETOOTH

5.1 Pruebas de desempeño en transferencia de archivo sin obstáculos.

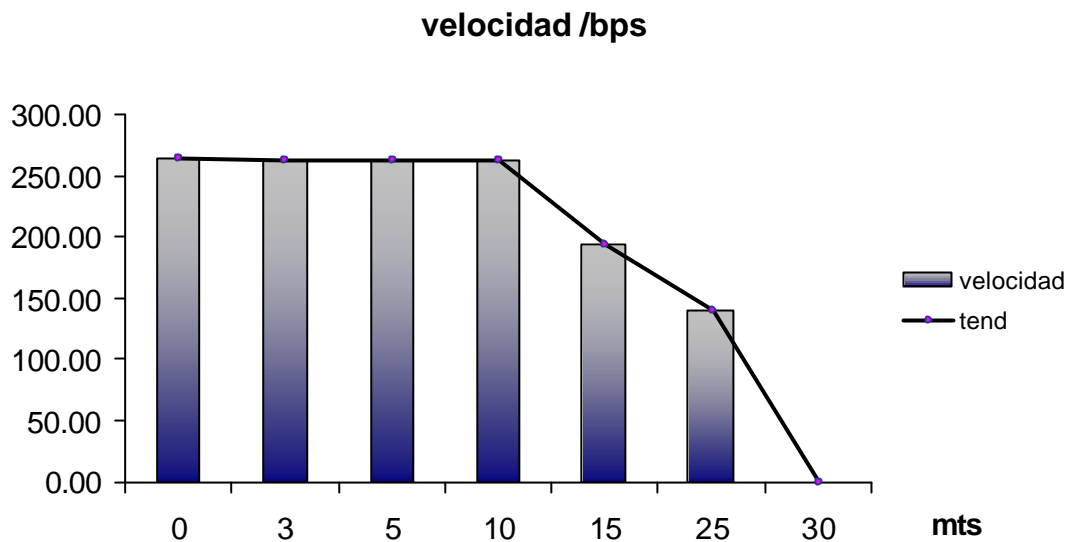
Se realizaron diversas pruebas con la tarjeta BLUETOOTH en conexión con un dispositivo BLUETOOTH USB como el mostrado en la figura 2.13.

Se hizo la prueba transfiriendo el mismo archivo (foto) de tamaño fijo de 23.2 kb a diferentes distancias y sin obstáculos de por medio, midiendo así el tiempo en que se completa la transferencia y la velocidad en que es transferido dicho archivo.



GRAFICA 5.1

Grafica de tiempo de duración de transferencia vrs. distancia



GRAFICA 5.2

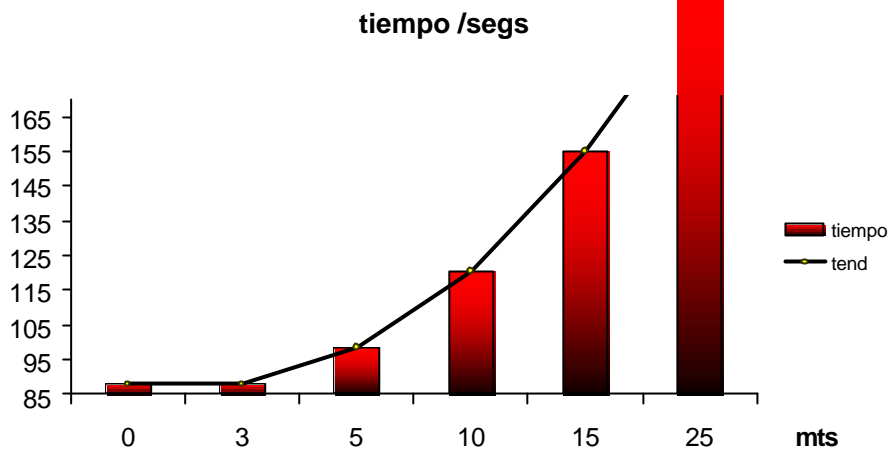
Grafica de velocidad de transmisión vrs. distancia

Se puede observar claramente en las graficas 5.1 y 5.2 que aunque el módulo BLUISM2 sea clase 2, con un rango de cobertura de 100 metros, la conexión empieza a decaer al sobrepasar los 10 metros, ya que el dispositivo BLUETOOTH USB es clase 1 con solamente 10 metros de rango de cobertura.

Adentro del rango de los 10 metros, la transferencia de un archivo de cualquier tipo fue lograda satisfactoriamente.

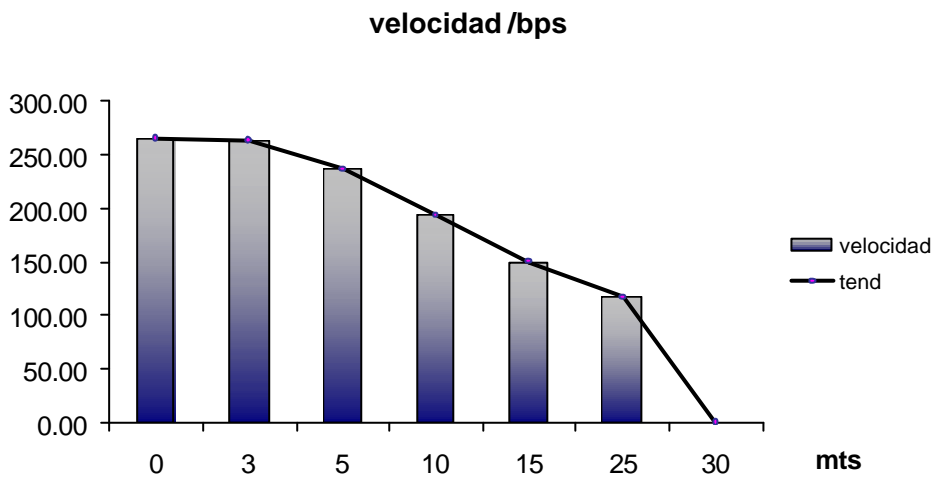
5.2 Pruebas de desempeño en transferencia de archivo con obstáculos.

Se realizaron las mismas pruebas con el mismo archivo pero esta vez con una pared de por medio. Dicha pared tiene 50cm de grosor y esta hecha de cemento. Los resultados fueron los que muestran a continuación en las graficas 5.3 y 5.4:



GRAFICA 5.3

Grafica de tiempo de duración de transferencia vrs. distancia con obstáculos.



GRAFICA 5.4

Grafica de velocidad de transmisión vrs. distancia con obstáculos.

Se puede ver que la comunicación empieza a decaer a los 5 metros, la transferencia tarda más tiempo en completarse y la velocidad de la transmisión disminuye a menor distancia.

6.0 CONCLUSIONES

- Se ha estudiado una tecnología nueva para El Salvador como son las conexiones inalámbricas utilizando Bluetooth y se ha creado un material didáctico como aporte a la Universidad Don Bosco, el cual enseña las bases para implementar una red usando esta moderna tecnología.
- Se puede confirmar que Bluetooth es una manera accesible de intercambiar información en un corto rango de distancia. Definitivamente reemplaza cables y enlaces infrarrojos.
- Se implementó el uso de los perfiles Bluetooth para la transferencia de datos, se logró el intercambio de archivos entre dos dispositivos Bluetooth en una red de 10 metros. Con el uso de un software funcional y amigable para el usuario, que incluye todas las funciones básicas para la transferencia de archivos y búsqueda de dispositivos Bluetooth.
- El software fue diseñado enteramente por nosotros por lo que no se recurrió en ningún gasto extra. La realización de una tarjeta Bluetooth no es económica, ni tampoco accesible, muchos de los elementos para su construcción se han comprado en otros países. Y sus costos son muy elevados ya que hay que pagar el importe a nuestro país y sus respectivos impuestos.
- Teniendo como base de comparación el dispositivo Bluetooth USB se demuestra que el tamaño de la tarjeta Tesis Bluetooth 2005 es más grande, por lo que no se puede asegurar que construir una tarjeta Bluetooth es de reducido tamaño y en cuanto al desempeño pudimos constatar que la velocidad de transferencia de un archivo entre dos dispositivos Bluetooth disminuye al tener un obstáculo de por medio y el tiempo que se tarda en completar la transferencia aumenta al no tener línea vista entre ambos dispositivos Bluetooth. Estas pruebas de desempeño simulan un ambiente ideal y un ambiente real el cual sería una oficina, plantas de producción, etc.

- Al trabajar con altas frecuencias, es muy importante el diseño de las pistas del circuito impreso como también el material de la tarjeta con la cual se va a trabajar.
- El equipo de medición capaz de alcanzar frecuencias de 2.4 GHz no está disponible para el uso público en nuestro país.
- El software Bluetooth para Windows es propio de cada dispositivo y no permite modificación alguna. Los emuladores y programas de control de esta tecnología se adquieren comprando sus respectivas licencias.
- La ayuda para resolver las dudas que aparecieron durante la elaboración de este proyecto fue recibida por personas afuera de El Salvador, que trabajan en el diseño de nuevos dispositivos que utilizan tecnología Bluetooth. Entre estos países se encuentran: USA, Alemania e Inglaterra, siendo este último el país origen de el modulo utilizado en esta tarjeta.

7.0 BIBLIOGRAFIA

- [1] SIG, Bluetooth. Acerca del Grupo de Interés Especial (SIG). 28- Ago – 2005.
<<http://www.bluetooth.com/about/>>
- [2] SIGET. Cuadro Nacional de Atribución de Frecuencias (SIGET). 28- Ago – 2005.
<<http://www.siget.gob.sv/BusquedaPublica.aspx?sector=2&tipo=4&titulo=t7&ordenado=0&dir=DESC>>
- [3] WIRELESS DEVELOPER NETWORK. Introducción al Bluetooth. 30-Ago-2005.
<<http://www.wirelessdevnet.com/channels/bluetooth/features/bluetooth.html>>
- [4] BLUETOOTH.COM. Especificación del Sistema Bluetooth Vol.1. 2001.
- [5] VELASCO, ALEJANDRO. Bluetooth Frequency Hopping Spread Spectrum.
30-Ago-2005 <<http://www.monografias.com/trabajos14/modulac-frecuencia/modulac-frecuencia.shtml#salto>>
- [6] WIRELESS DEVELOPER NETWORK. Introducción al Bluetooth. 30-Ago-2005.
<[http://www.wirelessdevnet.com/channels/bluetooth/features/blue tooth.html](http://www.wirelessdevnet.com/channels/bluetooth/features/blue%20tooth.html)>
- [7] ARCHUNDIA, PAPACETZI. Introducción al Bluetooth. 1-Sep-2005
<http://140.148.3.250/udl_a/servlet/mx.udlap.ict.tales.html.Block?Thesis=865&Type=C&Chapter=3>
- [8] BLUETOOTH.COM. Especificación del Sistema Bluetooth Vol.1. 2001.
- [9] STONE STREET ONE. Introduction to Bluetooth. 1-Sep-2005.
<<http://www.stonestreetone.com/PDF/IntroductiontoBluetooth.pdf>>
- [10] BLUETOOTH.COM. Especificación del Sistema Bluetooth Vol.1. 2001.

- [11] EZURIO LTD. Ezurio Bluetooth Intelligent Serial Modules. 19-Dic-2005.
<<http://www.ezurio.com/products/bism/>>
- [12] TDK SYSTEMS, AT Command Set. Europe. 2004
- [13] NATIONAL SEMICONDUCTOR. National Semiconductors LP2987 Datasheet.
USA. Marzo, 2005.
- [14] MAXIM INTEGRATED PRODUCTS. Maxim MAX3232 Transceiver Datasheet.
Diciembre, 2003.
- [15] SONY ERICSSON MOBILE COMMUNICATIONS AB. Accesorios SonyEricsson.
3-Sept-2005.
<http://www.sonyericsson.com/spg.jsp?cc=cam&lc=es&ver=4000&template=pp4_1&zone=pp&lm=pp4&cid=112>
- [16] MICROSOFT CORPORATION. Teclado y Mouse de Microsoft. 3-Sept-2005
<<http://www.microsoft.com/products/info/product.aspx?view=22&type=ovr&pcid=76e91501-ff1c-44cf-947f-003f3fe25a8c>>
- [17] DELL INC. Agendas Electrónicas Dell Axxim 3-Sept-2005.
<http://configure.us.dell.com/dellstore/config.aspx?c=us&cs=19&l=en&oc=X50624_HD1&s=dhs>
- [18] EBAY.COM. Mp3 Player con Bluetooth 3-Sept-2005.
<http://cgi.ebay.com/Bluetooth-MP3-Player-Phone-Diva-Gem-128MB-Glamour-NIB_W0QQitemZ5803704982QQcategoryZ73839QQssPageNameZWVWQOrdZ1QQcmdZViewItem>

[19] MOTOROLA. Teléfono Celular Motorola V600. 3-Sept-2005.

<<http://www.hellomoto.co.cr/v600/flash/default.shtml>>

[20] BLUETOOTH. Programa de Calificación Bluetooth. 3-Sept-2005.

<<http://qualweb.bluetooth.org/>>

[21] WIRELESS TELECOMMUNICATIONS. Bluetooth Protocol Snack. 3-Sept-2005.

<<http://www.thewirelessdirectory.com/Bluetooth-Software/Bluetooth-Protocol-Stack.htm>>

[22] SOURCEFORGE.NET. Sourceforge Open BT Stack 3-Sept-2005.

<<http://sourceforge.net/projects/openbt>>

[23] IBM.COM. Alpha Works Bluetooth Stack. 3-Sept-2005

<<http://www.alphaworks.ibm.com/tech/bluedrekar>>

[24] DMITRY KASATKIN, CARLOS CHINEA. Affix Bluetooth Protocol Stack.

3-Sept-2005. <<http://affix.sourceforge.net/>>

[25] BLUEZ PROJECT. Bluez Linux Protocol Stack. 3-Sept-2005.

<<http://www.bluez.org/>>

[26] LUDOVIC ROUSSEAU, CHRISTOPHE ARNOUX. Añadir un dispositivo de confianza en una Piconet Bluetooth. 3-Sept-2005.

<<http://www.gemplus.com/smart/rd/publications/pdf/RAC01too.pdf#search='bluetooth%20trusted%20device'>>

[27] MICROSOFT CORPORATION. Microsoft Visual Basic. 3-Sept-2005.

<<http://msdn.microsoft.com/vbasic/>>