



**UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA**

**“COMUNICACIÓN INALÁMBRICA DE SISTEMAS IEEE 488  
MEDIANTE UNA PICORED BLUETOOTH”**

**TESIS**

**PARA OBTENER EL TÍTULO DE  
INGENIERO EN ELECTRÓNICA**

**PRESENTA**

**RAÚL FERNÁNDEZ ROJAS**

**DIRECTOR DE TESIS**

**ING. HERIBERTO ILDEFONSO HERNÁNDEZ MARTÍNEZ**

**HUAJUAPAN DE LEÓN, OAX.; DICIEMBRE DE 2005**



**Tesis presentada el 9 de diciembre de 2005  
ante los siguientes sinodales:**

**M.C. José Antonio Moreno Espinosa**

**M.C. Esteban Guerrero Ramírez**

**M.C. Enrique Guzmán Ramírez**

**Director de Tesis:**

**c.Dr. Heriberto I. Hernández Martínez**



## **Dedicatoria**

El presente trabajo está dedicado a mi familia; a mis padres que sin duda me han apoyado desde que inicié mis andaduras y que, a pesar de los tropiezos, siempre me han ayudado a seguir adelante.

Quiero dedicar principalmente a mi madre Virginia Rojas Ríos (Q.E.P.D.) que siempre estará en mi corazón y por ella seguiré luchando día tras día sin olvidar todo lo que me enseñó para llegar a ser la persona que soy.

A mi padre Raúl Fernández Chávez que ha sido un ejemplo a seguir, y que sin su comprensión y confianza no hubiera podido culminar mis estudios.

A mi abuelita Margarita Rojas Ríos que tanto a pedido por mí, y que tantas recomendaciones me ha hecho.

A mi hermana Berenice por la paciencia que ha tenido conmigo.

Son la fuente que me anima a seguir mi camino.

Raúl.



## Agradecimientos

Culminar este proyecto fue un largo proceso, que seguramente no hubiera podido realizar sin el apoyo de las personas que me rodean.

Quiero agradecer muy especialmente a mi director de tesis Heriberto I. Hernández Martínez por haberme dado la oportunidad de colaborar con él, y sobre todo por haberme permitido desarrollar y llevar a cabo esta tesis, porque siempre me alentó a seguir adelante y porque me mostró el camino a seguir.

A mis sinodales por la valiosa aportación a este trabajo, así también a todos mis maestros de la licenciatura.

A Virgilio y Jehú, por haber estado en las buenas y en las malas, por ser mis hermanos de mil batallas, por brindarme su apoyo y cariño incondicional y por que ellos han sido una parte importante en este trabajo (sigan así, no cambien por favor). A Francisco (Pancho) por que es mi alumno más destacado.

A todos mis compañeros y amigos (Ch.) quienes de una manera u otra me soportaron y alentaron durante el desarrollo de este trabajo, para ellos, mi cariño y gratitud.

Quiero agradecer al Ing. Josué Neftalí García Matías, por su ayuda técnica a lo largo del desarrollo del proyecto de tesis.

Quiero agradecer a las siguientes firmas, que hicieron posible que haya podido contar con las herramientas software y hardware que necesité para este proyecto: Microchip, por haber brindado los microcontroladores utilizados para la elaboración de este trabajo de tesis; Texas Instruments, por haber proporcionado los transceptores de control y de datos para el manejo del bus GPIB.

Por último, quiero agradecer a las siguientes personas: Steve Harris, Kay Hansen, Txus, Alberto Rionda, H.P. Lovecraft, Edgar A. Poe, John Milton, Isaac Asimov, Walter Giardino y Víctor García. UP THE IRONS!!!!

A todos, gracias por formar parte del *script* y *sound track* de la película de mi vida.

Raúl.





## Índice

Dedicatoria .....	v
Agradecimientos .....	vii
Índice .....	ix
Índice de tablas .....	xiii
Índice de figuras .....	xv
Resumen .....	xvii
Abstract .....	xix
1. Introducción .....	1
1.1. Metodología de desarrollo de sistemas empotrados .....	1
1.2. Planteamiento del problema .....	3
1.3. Objetivos de la tesis .....	4
1.4. Estructura de la tesis .....	5
2. GPIB .....	7
2.1. Evolución histórica .....	7
2.2. IEEE 488.1 .....	8
2.2.1. Especificaciones eléctricas .....	9
2.2.2. Especificaciones mecánicas .....	9
2.2.3. Especificaciones funcionales .....	9
2.2.3.1. Tipos de instrumentos GPIB .....	9
2.2.3.2. Funciones de la interfaz .....	10
2.2.3.3. Estructura del bus .....	10
2.2.3.3.1. Función de transferencia GPIB .....	10
2.2.3.3.2. Líneas de manejo de interfaz .....	11
2.2.3.4. Direcciones GPIB .....	12
2.2.3.5. Órdenes GPIB .....	12
2.2.3.5.1. Órdenes direccionadas .....	12
2.2.3.5.2. Órdenes universales multilínea .....	13
2.2.3.5.3. Órdenes de receptor y emisor .....	13

2.2.3.5.4. Órdenes secundarias .....	13
2.2.3.6. Sondeo .....	13
2.3. IEEE 488.2.....	14
2.3.1. Sintaxis y formato de datos.....	14
2.3.2. Codificación.....	14
2.3.3. Reporte de estado.....	14
2.3.4. Órdenes comunes.....	15
2.4. Órdenes estandarizadas para instrumentos programables .....	15
2.4.1. Formato de datos.....	15
3. Bluetooth.....	17
3.1. Soluciones inalámbricas .....	17
3.1.1. Infrarrojo.....	18
3.1.2. IEEE 802.11.....	18
3.1.3. HomeRF.....	19
3.1.4. Bluetooth.....	19
3.2. Grupo de interés especial Bluetooth .....	20
3.3. Historia de Bluetooth.....	21
3.4. Tecnología Bluetooth.....	21
3.4.1. RF .....	21
3.4.1.1. RF en el espectro de 2.4 GHz.....	22
3.4.1.2. Espectro expandido.....	22
3.5. Principios de operación Bluetooth.....	23
3.6. Topología Bluetooth .....	23
3.7. Arquitectura de protocolos Bluetooth.....	24
3.7.1. Capa de transporte .....	25
3.7.2. Protocolos inferiores de la capa de transporte .....	26
3.7.2.1. Subcapa de radio.....	26
3.7.2.2. Subcapa de banda base .....	27
3.7.2.2.1. Estados de un dispositivo Bluetooth.....	28
3.7.2.2.2. Operación de invitación.....	30
3.7.2.2.3. Operación de búsqueda.....	30
3.7.2.2.4. Código de acceso .....	31
3.7.2.2.5. Estado conectado .....	32
3.7.2.2.6. Tipos de BB_PDU de banda base.....	33
3.7.2.2.7. Tipos de enlaces de banda base .....	34
3.7.2.3. Subcapa de administrador de enlace.....	34
3.7.2.3.1. Manejo de seguridad.....	36
3.7.2.3.2. Autenticación de dispositivos.....	36
3.7.2.3.3. Encriptación del enlace.....	36
3.7.2.3.4. Manejo de potencia.....	36

---

3.7.2.3.5. Manejo de ancho de banda .....	37
3.7.2.3.6. Calidad de servicio para enlaces ACL.....	37
3.7.2.3.7. Manejo del controlador de enlace.....	38
3.7.2.3.8. Esquema de invitación.....	38
3.7.2.3.9. Rol del Maestro y del Esclavo.....	38
3.7.2.3.10. Reloj e información de los tiempos .....	38
3.7.3. Protocolos superiores del grupo de transporte.....	38
3.7.3.1. Subcapa L2CAP .....	39
3.7.3.1.1. Canales L2CAP .....	39
3.7.3.1.2. Tipos de L2CAP_PDU .....	40
3.7.3.1.3. Administración del canal L2CAP.....	41
3.7.3.1.4. Opciones de configuración.....	42
3.7.3.2. Subcapa HCI.....	43
3.7.3.2.1. Clases de HCI_PDUs .....	43
3.7.3.2.2. HCI_PDUs.....	46
3.7.3.2.3. HCI_PDUs de control de enlace.....	46
3.7.3.2.4. Políticas de enlace HCI_PDUs.....	46
3.7.3.2.5. Controlador de huésped y HCI_PDUs de banda base .....	46
3.7.3.2.6. Parámetros de información HCI_PDUs .....	47
3.7.3.2.7. Parámetros de estado HCI_PDUs.....	47
3.7.3.2.8. Examinación HCI_PDUs.....	47
3.7.4. Protocolos de capa intermedia.....	47
3.7.4.1. Subcapa RFCOMM .....	48
3.7.4.1.1. Protocolo RFCOMM .....	48
3.7.4.1.2. Proceso de desarrollo del protocolo RFCOMM.....	49
3.7.4.1.3. Análisis del protocolo RFCOMM .....	49
3.7.4.2. Subcapa SDP .....	50
3.7.4.2.1. Proceso de desarrollo de la subcapa SDP .....	50
3.7.4.2.2. Análisis de la subcapa SDP .....	50
3.7.4.3. Subcapa de protocolos IrDA.....	51
3.7.4.3.1. Protocolos de interoperabilidad IrDA .....	52
3.7.4.3.2. Proceso de desarrollo del protocolo de interoperabilidad IrDA.....	53
3.7.4.3.3. Protocolo OBEX.....	53
3.7.4.4. Subcapa de control de audio y telefonía.....	53
3.7.4.4.1. Operación de control de audio y telefonía.....	54
3.7.4.4.2. Protocolo TCS .....	54
3.7.4.4.3. Análisis del protocolo TCS .....	54
3.7.4.4.4. Desarrollo del audio Bluetooth.....	55
3.7.4.4.5. Análisis del audio en Bluetooth.....	55
3.7.4.5. Subcapa de red.....	55

3.7.5. Capa de aplicación .....	55
4. Desarrollo del CISBlue-488 .....	57
4.1. Especificación del CISBlue-488 .....	57
4.1.1. Segmentación del diseño .....	58
4.1.1.1. Componentes HW .....	58
4.1.1.1.1. Pasarela GPIB/RS232 .....	58
4.1.1.1.2. Módulo Bluetooth .....	58
4.1.1.2. Componentes SW .....	59
4.1.2. Iteración y refinamiento .....	59
4.1.3. Diseño de tareas HW y SW .....	60
4.1.3.1. Diseño HW .....	60
4.1.3.1.1. Comunicación con el GPIB .....	61
4.1.3.1.2. Comunicación entre la pasarela GPIB/RS232 y el módulo Bluetooth .....	61
4.1.3.2. Diseño SW .....	62
4.1.3.2.1. Configuración del CISBlue-488 .....	62
4.1.3.2.2. Lectura del GPIB .....	62
4.1.3.2.3. Comunicación serie .....	63
4.1.3.2.4. Escritura al GPIB .....	64
4.1.4. Integración HW y SW .....	64
4.1.5. Pruebas y liberación del producto .....	64
4.1.6. Mantenimiento y actualización .....	66
5. Resultados .....	67
5.1. Consumo de corriente del CISBlue-488 .....	67
5.2. Manejo de la función de transferencia del GPIB .....	67
5.3. Inicialización del CISBlue-488 .....	68
5.4. Pruebas de transmisión y alcance entre los módulos Bluetooth .....	69
5.5. Funcionamiento del CISBlue-488 en tiempo real .....	69
5.6. Velocidad de transferencia de datos .....	74
6. Conclusiones .....	77
Bibliografía .....	79
Sitios URL .....	80
Anexo A .....	A-1
Anexo B .....	B-1

## Índice de tablas

Tabla 2.1. Niveles de voltaje IEEE 488.1. ....	9
Tabla 2.2. Funciones básicas del estándar IEEE 488.1. ....	10
Tabla 2.3. Especificación del tipo de orden GPIB. ....	12
Tabla 2.4. Especificaciones básicas del estándar IEEE 488 [8]. ....	16
Tabla 3.1. Principales características de la tecnología de infrarrojos. ....	18
Tabla 3.2. Principales características de la tecnología IEEE 802.11. ....	18
Tabla 3.3. Principales características de la tecnología HomeRF. ....	19
Tabla 3.4. Principales características de la tecnología Bluetooth. ....	19
Tabla 3.5. Aplicaciones típicas de las tecnologías inalámbricas. ....	20
Tabla 3.6. Asignación de frecuencias en la banda 2.4 GHz. ....	22
Tabla 3.7. Parámetros de operación para la subcapa de radio. ....	27
Tabla 3.8. Paquete FHS. ....	31
Tabla 3.9. Entradas FSM y códigos de acceso utilizados durante los estados activos de un dispositivo. ....	32
Tabla 3.10. Campos de la cabecera BB_PDU. ....	33
Tabla 3.11. Formato de LMP_PDU. ....	35
Tabla 3.12. Formato de una L2CAP_PDU CL. ....	40
Tabla 3.13. El formato de una L2CAP_PDU CO. ....	40
Tabla 3.14. Orden de señalización L2CAP_Connection_Request. ....	42
Tabla 3.15. Orden de señalización L2CAP_Configuration_Request. ....	42
Tabla 3.16. Opciones de configuración. ....	42
Tabla 3.17. Orden HCI_PDU. ....	45
Tabla 3.18. Evento HCI_PDU. ....	45
Tabla 3.19. Datos HCI_PDU. ....	45
Tabla 3.20. Orden HCI_inquiry. ....	46
Tabla 3.21. Orden HCI_Hold_Mode. ....	46
Tabla 3.22. Orden HCI_Write_Page_Scan_Activity. ....	46
Tabla 4.1. Segmentación HW y SW del CISBlue-488. ....	58
Tabla 4.2. Asignación de funciones a los puertos del MCU. ....	60

Tabla 4.3. Dirección de memoria de las variables globales. ....	60
Tabla 4.4. Registros de configuración del MCU. ....	61
Tabla 4.5. Actualizaciones del CISBlue-488. ....	66
Tabla 5.1. Consumo de corriente de los componentes del CISBlue-488. ....	67
Tabla 5.2. Mensaje *IDN? enviado por el controlador del GPIB. ....	70
Tabla 5.3. Respuesta de identificación del instrumento receptor. ....	71

## Índice de figuras

Figura 1.1. Diagrama del ciclo de vida del diseño de sistemas empotrados. ....	3
Figura 1.2. Sistema inalámbrico de instrumentación electrónica programable.....	3
Figura 1.3. Estructura del trabajo de tesis. ....	5
Figura 2.1. Relación entre los estándares IEEE 488.1, IEEE 488.2 y SCPI [8].....	8
Figura 2.2. Configuración lineal y en estrella. ....	8
Figura 2.3. Conector GPIB y configuración de terminales. ....	9
Figura 2.4. Estructura del bus [8]. ....	10
Figura 2.5. Transmisión de datos usando la función de transferencia [URL8].....	11
Figura 3.1. Emblema de la tecnología Bluetooth. ....	21
Figura 3.2. Picored Bluetooth.....	23
Figura 3.3. Estados de una picored Bluetooth.....	24
Figura 3.4. Ejemplo de una red dispersa. ....	24
Figura 3.5. Arquitectura del protocolos Bluetooth. ....	25
Figura 3.6. Estructura general de un módulo Bluetooth.....	25
Figura 3.7. Capa de transporte.....	26
Figura 3.8. Operaciones de la subcapa de radio. ....	27
Figura 3.9. Principales funciones de la subcapa de banda base. ....	28
Figura 3.10. Máquina de estados de un dispositivo Bluetooth.....	28
Figura 3.11. Dirección de los dispositivos Bluetooth.....	29
Figura 3.12. Reloj Bluetooth. ....	29
Figura 3.13. Módulo de selección de frecuencia.....	30
Figura 3.14. Funciones del código de acceso. ....	32
Figura 3.15. Formato del BB_PDU.....	33
Figura 3.16. Funciones de la subcapa de administrador de enlace.....	35
Figura 3.17. (a) Transacción y negociación; (b) El Maestro ordena el ajuste del enlace.....	35
Figura 3.18. El grupo de protocolos de la capa de transporte. ....	39
Figura 3.19. Canales L2CAP.....	40
Figura 3.20. Transacciones de señalización L2CAP.....	41

Figura 3.21. Arquitectura de un sistema con interfaz de huésped.....	44
Figura 3.22. Clases de HCI_PDUs.....	44
Figura 3.23. Subcapas RFCOMM y SDP.....	47
Figura 3.24. Protocolos de la capa intermedia.....	48
Figura 3.25. Comunicaciones de enlaces seriales multiplexados sobre una conexión RFCOMM. ...	49
Figura 3.26. Estructura del registro de servicio SDP.....	51
Figura 3.27. Transacciones SDP.....	52
Figura 3.28. Protocolo OBEX.....	52
Figura 3.29. Subcapas de audio y TCS-BIN.....	53
Figura 3.30. Capa de aplicación.....	56
Figura 4.1. Módulo Bluetooth.....	59
Figura 4.2. Diagrama de flujo de la inicialización.....	62
Figura 4.3. Diagrama de flujo de la lectura del GPIB.....	63
Figura 4.4. Diagramas de flujo para la transmisión y recepción serie.....	63
Figura 4.5. Diagrama de flujo para la función escritura al GPIB.....	64
Figura 4.6. Lectura del bus GPIB.....	65
Figura 4.7. Pasarela GPIB/RS232.....	65
Figura 4.8. CISBlue-488 en funcionamiento.....	66
Figura 5.1. Función de transferencia realizada por el controlador y la pasarela GPIB/RS232.....	68
Figura 5.2. Inicialización del CISBlue-488.....	68
Figura 5.3. Diagrama de flujo para comprobar y evaluar el funcionamiento del CISBlue-488.....	72
Figura 5.4. Estado inicial del instrumento Agilent 33120A.....	72
Figura 5.5. Modifica la función a señal triangular.....	73
Figura 5.6. Modifica la frecuencia de operación a 63 KHz.....	73
Figura 5.7. Modifica el voltaje de operación a 1.5 Vpp.....	73
Figura 5.8. Lee los valores de frecuencia y de voltaje.....	74
Figura 5.9. Realiza prueba general al instrumento.....	74
Figura 5.10. Velocidades de transferencia del CISBlue-488.....	75



## Resumen

Los avances tecnológicos en el campo de la instrumentación electrónica han generado diversas soluciones respecto a los protocolos de comunicaciones, sin embargo el estándar IEEE 488, propuesto a mediados de los años 70s, sigue siendo el estándar *de facto* en el diseño y construcción de sistemas automatizados de medida (ATE, *Automated Test Equipment*). El estándar IEEE 488, conocido como GPIB (*General Purpose Instrumentation Bus*), presenta como principal desventaja el sistema de cableado, el cual es rígido y de corto alcance.

Actualmente, las comunicaciones inalámbricas en entornos personales han dado origen a un nuevo tipo de red conocida como PAN (*Personal Area Networks*), en donde el protocolo de comunicaciones Bluetooth es utilizado debido a su bajo costo y facilidad de operación.

Por otro lado, el diseño de un sistema empotrado (*embedded systems*) implica que tanto el software (SW) como el hardware (HW) se diseñen en paralelo mediante una metodología de desarrollo.

El presente trabajo de tesis plantea el diseño de un sistema inalámbrico de instrumentación electrónica programable mediante comunicación Bluetooth. Considerando una metodología de desarrollo de sistemas empotrados, se diseñó y construyó el dispositivo de Comunicaciones Inalámbricas de Sistemas IEEE 488 (CISBlue-488).

El CISBlue-488 realiza la conversión de protocolos IEEE 488/RS232/Bluetooth mediante una pasarela (*gateway*) para la comunicación entre el controlador GPIB (PC) y los diferentes instrumentos electrónicos programables existentes en el Laboratorio de Comunicaciones Digitales (LabCD) de la Universidad Tecnológica de la Mixteca.

Se define cada una de las fases de diseño y desarrollo del CISBlue-488, dando como resultado un producto robusto.



## **Abstract**

Advances in technology in the area of electronic instrumentation have generated a number of solutions which are related to the protocols of communication, however, the IEEE 488 standard, implemented in the 70s is still being used in the design and construction of Automated Test Equipment (ATE). Nevertheless, this standard better known as General Purpose Instrumentation Bus (GPIB), presents disadvantages i.e. its wiring system, which is rigid and of short reach.

Currently, the wireless communications with focus on personal environments have created a new type of networks better known as Personal Area Networks (PAN), Bluetooth, for example, is of easy operation and has proved to be an economic and reliable solution.

The design of an embedded system is usually carried out following a development methodology in which the software (SW) and the hardware (HW) are designed in parallel.

This thesis hence implements the design of a wireless system of programmable electronic instrumentation using Bluetooth communication. As a result, one Wireless IEEE 488 Communications System was designed and constructed (CISBlue-488) with basis on a methodology of development of embedded systems.

The CISBlue-488 converts the IEEE 488/RS232/Bluetooth protocols using a specific gateway. This gateway directly makes the communication between the GPIB (PC) controller and different programmable electronic instruments located in the Laboratory from Digital Communications (LabCD) of the Mixteca University of Technology.

Therefore, the aim of this thesis is to describe all the phases involved in the design and development of the CISBlue-488 through the design of embedded systems.



## 1. Introducción

Una computadora personal (PC, *Personal Computer*) se interconecta a sus periféricos mediante conexiones punto a punto, las cuales se han incrementado considerablemente. Lo anterior ha ocasionado el uso de un gran número de cables de conexión.

El campo de la instrumentación electrónica programable, en donde el sistema controlador generalmente es una PC, requiere de cables normalizados para su interconexión con instrumentos de medida y estimulación (multímetros, osciloscopios, sistemas de adquisición de datos, etc.). El bus de instrumentación de propósito general (GPIB, *General Purpose Instrumentation Bus*), cuyo estándar (IEEE 488) recomienda la utilización de cables rígidos para la comunicación en paralelo a distancias cortas, ha sido la solución más utilizada [3, 7, 8, 13].

Bluetooth es un protocolo de comunicaciones inalámbricas, cuya tecnología permite la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia (RF, *Radio Frequency*) en distancias cortas, tanto en ambientes de trabajo cerrados (hogar, oficinas, etc.) como en espacios públicos, además, es de bajo costo y presenta bajo consumo de potencia [URL3].

El Cuerpo Académico de Redes de Instrumentación (CA RdI) de la UTM cuenta entre sus líneas de investigación el estudio de los protocolos de comunicaciones industriales y de instrumentación electrónica. Resultado de ello son los trabajos realizados en el campo de la instrumentación electrónica programable y virtual [7, 8] y en el de las comunicaciones industriales [4].

### 1.1. Metodología de desarrollo de sistemas empotrados

El término sistemas empotrados (*embedded systems*) considera dispositivos encargados de realizar la etapa de control en en una gran variedad de aplicaciones, tales como: dispositivos electrónicos de consumo (videojuegos, reproductores de audio/vídeo, etc.); automoción (control de bolsas de aire, climatizador, etc.); industria (control de motores, robótica, etc.); y comunicaciones (teléfonos móviles, módems, etc.) [2].

El diseño de sistemas empotrados considera dos elementos conceptuales: en primer lugar está la arquitectura hardware (HW) construida en torno a un sistema basado en microprocesador ( $\mu$ P), microcontrolador (MCU, *Microcontroller*) o dispositivo de procesamiento digital de señales (DSP, *Digital Signal Processing*); y por otra parte está la arquitectura software (SW) que involucra a los sistemas operativos (OS, *Operating System*), sistemas operativos en tiempo real (RTOS, *Real-Time Operating System*), lenguajes de programación, compiladores, y herramientas de modelado, simulación y pruebas [1]. La integración de componentes HW y componentes SW se denomina *firmware*.

Las principales características de un sistema empotrado son [2]:

- Están dedicados a resolver una tarea específica.

- Cuentan con el soporte de un amplio conjunto de arquitecturas y procesadores.
- Presentan gran flexibilidad en cuanto a la asignación de entradas y salidas.
- Generalmente, forman parte de un sistema más grande.
- Presentan bajos costos de desarrollo.
- Se utilizan en aplicaciones críticas que requieren respuesta en tiempo real (*real-time*).
- Tienen un buen desempeño en ambientes bajo condiciones extremas.
- Integran varias opciones en cuanto a suministro y ahorro de energía.
- La mayoría permite almacenamiento de código objeto en una memoria tipo ROM (*Read Only Memory*).
- Requieren herramientas y métodos especializados para obtener sistemas robustos.

Generalmente el diseño de un sistema empujado implica que tanto el SW como el HW se desarrollen en paralelo. La Figura 1.1 muestra el ciclo de vida del diseño de sistemas empujados, cuyas fases se describen a continuación [2]:

- *Especificación del producto*: el diseño de cualquier sistema o producto inicia con la definición de los requerimientos que debe cumplir de forma clara y precisa. Esta fase debe incluir las consideraciones de diseño y las características técnicas y funcionales.
- *Segmentación del diseño*: debido a que la arquitectura de un sistema empujado contiene componentes HW y componentes SW, es necesario definir los problemas y requisitos que serán resueltos por HW y/o por SW.
- *Iteración y refinamiento HW y SW*: en esta fase se dividen los bloques especificados en la fase anterior en componentes más pequeños y se trabaja con herramientas específicas para cada componente, tanto HW como SW. Los diseñadores del HW pueden utilizar herramientas de simulación (simuladores de arquitecturas) para modelar el desempeño del MCU; y los diseñadores de SW pueden ejecutar programas de depuración para simular el desempeño del *firmware*.

Es necesario refinar los procesos o actividades que habrá de realizar cada fase con la finalidad de ahorrar tiempo y evitar incrementar los costos del producto final.

- *Diseño de tareas HW y SW*: en esta fase se diseñan todos los componentes HW y SW especificados en los requerimientos iniciales. Aunque en esta fase se trabaja en los componentes y funciones por separado, es necesario tener contacto con la contraparte del diseño para mantener coherencia en el producto final.
- *Integración HW y SW*: es necesario integrar los componentes desarrollados para verificar que interaccionan entre sí de manera correcta. Para lograr la integración total es necesario llegar a un acuerdo común en el que se establezcan mecanismos tales como diagramas de secuencia, combinaciones de entradas y salidas, etc.
- *Pruebas y liberación del producto*: en esta fase se realizan las pruebas del producto, y con base en los resultados obtenidos se realizan modificaciones y/o correcciones para adaptarlo a las necesidades del usuario. Es necesario considerar que dichas modificaciones y/o correcciones del HW o SW pueden incrementar el costo total del producto o sistema. Una vez realizadas las pruebas, se libera el producto.
- *Mantenimiento y actualización*: esta fase se encarga, si es necesario, de realizar modificaciones al producto una vez liberado. Se recomienda documentar detalladamente el diseño y desarrollo del producto para prever futuras modificaciones. Por otro lado, la actualización se refiere a liberar nuevas versiones del producto.

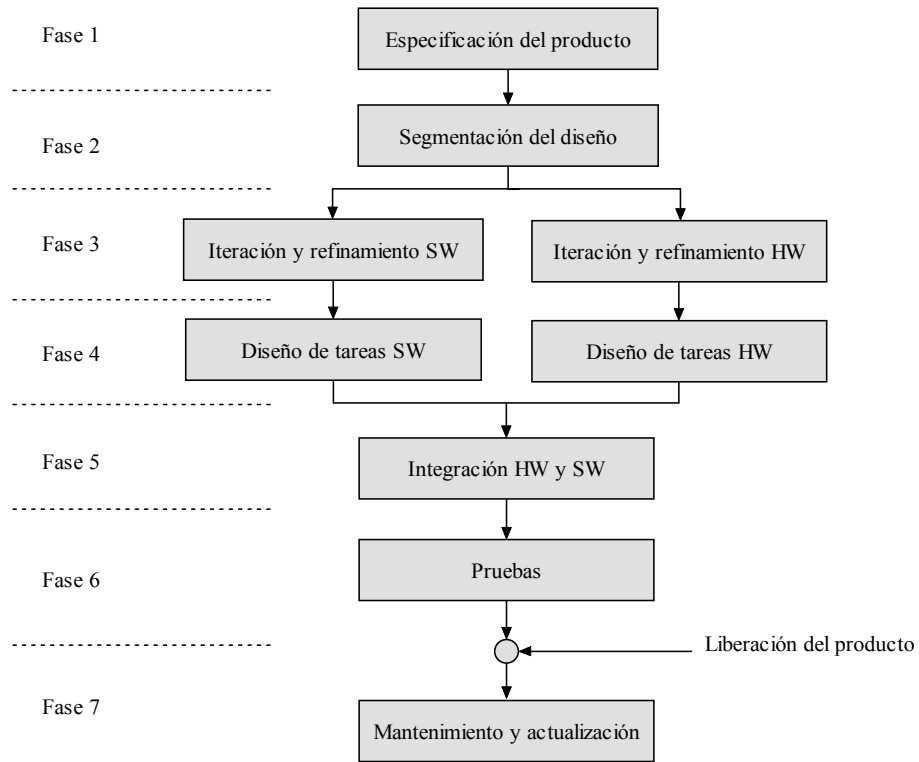


Figura 1.1. Diagrama del ciclo de vida del diseño de sistemas empujados.

## 1.2. Planteamiento del problema

El presente trabajo de tesis propone realizar la interconexión inalámbrica de instrumentos electrónicos con la PC mediante una pasarela (*gateway*) entre protocolos IEEE 488/Bluetooth, para ello se diseñará y construirá un dispositivo de Comunicaciones Inalámbricas de Sistemas IEEE 488 (CISBlue-488).

La Figura 1.2 muestra el diagrama a bloques de un sistema inalámbrico de instrumentación electrónica programable, el cual consta de los siguientes módulos:

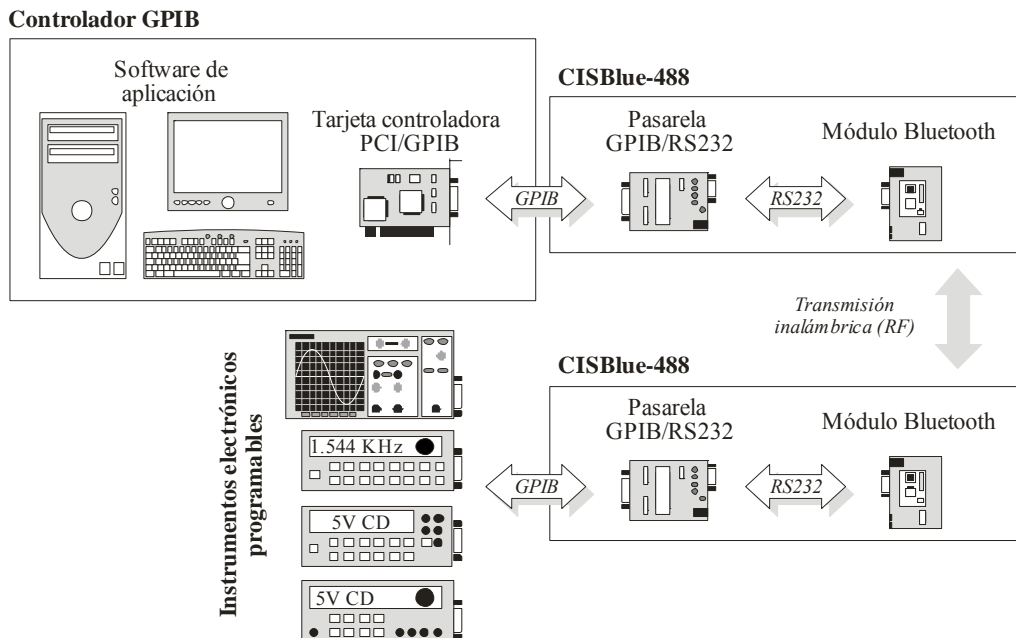


Figura 1.2. Sistema inalámbrico de instrumentación electrónica programable.

- *Controlador GPIB*: consta del SW de aplicación y una tarjeta controladora PCI/GPIB de la firma Agilent Technologies, integrados en una PC. Este módulo se encarga de enviar y recibir las órdenes (*commands*) y los datos a los instrumentos electrónicos programables mediante el protocolo de comunicaciones GPIB.
- *CISBlue-488*: es el sistema a desarrollar, cuya función principal es comunicar al controlador GPIB con los instrumentos electrónicos programables de forma inalámbrica. Consta de dos submódulos:
  - *Pasarela GPIB/RS232*: cuyo diseño se basa en el MCU PIC16F877 [10] de la firma Microchip [URL7] y su principal función es la conversión de los protocolos GPIB a RS232 y viceversa. Como interfaz física emplea los transceptores (*transceivers*) 75160 [14] para las líneas de datos y el 75161 [15] para las líneas de control del GPIB. Los datos capturados por los puertos de E/S se transmiten en forma serie al módulo Bluetooth mediante la USART (*Universal Synchronous Asynchronous Receiver Transmitter*) del PIC.
  - *Módulo Bluetooth*: se encarga de transmitir o recibir la información proporcionada por la pasarela GPIB/RS232 vía RF [5, 6]. Este submódulo fue fabricado por la firma sueca ConnectBlue [URL2] y proporciona comunicación RS232 mediante un conector tipo DB9.
- *Instrumentos electrónicos programables*: son los dispositivos electrónicos de medida y estimulación encargados de recibir las órdenes del controlador y responder con la información correspondiente.

La pasarela GPIB/RS232 se encarga de capturar las líneas de control y de datos del controlador GPIB, para convertirlas a tramas RS232 y trasmitirlas vía RF mediante el módulo Bluetooth. En el lado receptor se realiza el proceso inverso, con el módulo Bluetooth se captura la información y se pasa mediante RS232 a la pasarela GPIB/RS232, la cual convertirá la información a GPIB para que los instrumentos electrónicos programables puedan reconocer las órdenes del controlador GPIB. Se realiza el mismo proceso pero en sentido contrario cuando el instrumento actúa como transmisor.

El CISBlue-488 se desarrollará siguiendo la metodología de sistemas empotrados descrita con la finalidad de obtener un sistema robusto.

### 1.3. Objetivos de la tesis

El objetivo principal del presente trabajo de tesis es diseñar y construir un sistema que realice la interconexión inalámbrica de instrumentos electrónicos programables mediante el estándar IEEE 488.

Para cumplir con lo anterior, se plantean los siguientes objetivos secundarios:

- Estudio de los protocolos de comunicaciones IEEE 488 y Bluetooth, con base en una investigación bibliográfica.
- Diseño y construcción de una pasarela IEEE 488/Bluetooth, con base en la metodología de desarrollo de sistemas empotrados.
- Integración de los módulos Bluetooth al sistema para proporcionar la comunicación inalámbrica.
- Realizar las pruebas del sistema en el Laboratorio de Comunicaciones Digitales (LabCD) de la UTM, con la finalidad de integrar el CISBlue-488 a sus sistemas y dispositivos de instrumentación electrónica programable.



## 1.4. Estructura de la tesis

La estructura del documento de tesis se muestra en la Figura 1.3 y se detalla a continuación.

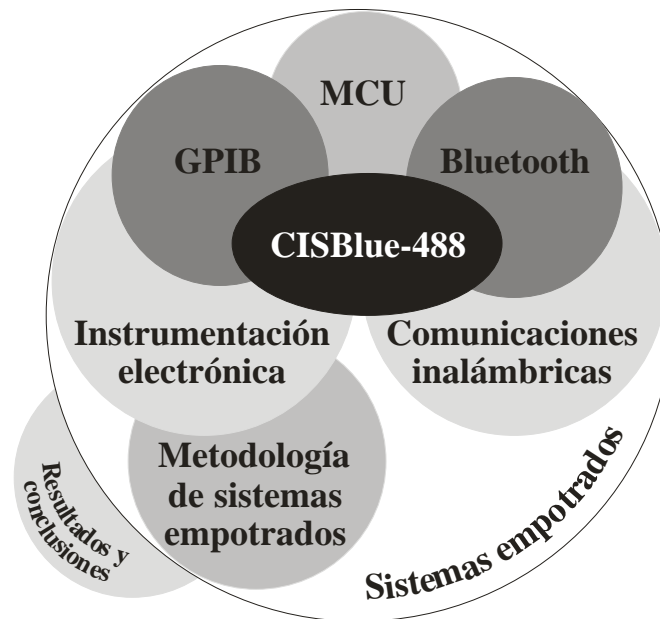
El capítulo 2 presenta el marco conceptual del GPIB, considerando las especificaciones eléctricas, mecánicas, funcionales y de procedimientos descritas por el estándar IEEE 488.

El capítulo 3 muestra comparativa entre las diferentes soluciones inalámbricas existentes en el mercado y describe la especificación de Bluetooth.

En el capítulo 4 se presenta el diseño y construcción del CISBlue-488 con base en la metodología de sistemas empotrados.

El capítulo 5 expone los resultados obtenidos en el presente trabajo de tesis.

Se presentan las referencias bibliográficas utilizadas en el desarrollo de esta tesis y finalmente, se presentan dos anexos, correspondiente a la instalación y configuración del módulo Bluetooth y las principales funciones de la interfaz GPIB, respectivamente.



**Figura 1.3.** Estructura del trabajo de tesis.



## 2. GPIB

A lo largo de la historia el hombre se ha enfrentado a la necesidad de medir ciertos parámetros en sus actividades cotidianas. Para ello ha desarrollado instrumentos que le ayuden a obtener dichas medidas y con ello surgió el concepto de instrumentación.

Mientras que el desarrollo industrial formalizó el campo de la instrumentación, los avances tecnológicos han permitido el desarrollo de complejos y avanzados instrumentos electrónicos de medida y estimulación. Cabe señalar que la idea de interconectar dichos instrumentos mediante un bus de comunicaciones y el surgimiento del concepto de instrumentación virtual han aportado ventajas considerables en cuanto a la automatización de procesos industriales [7]. Por lo anterior, la complejidad y capacidad de dichos instrumentos ha llegado a requerir de personal altamente capacitado en áreas de informática, comunicaciones y control.

### 2.1. Evolución histórica

En septiembre de 1965, la compañía Hewlett-Packard (HP), en la actualidad Agilent Technologies<sup>1</sup> [URL1], ideó la posibilidad de interconectar sus instrumentos mediante un sistema de comunicaciones en el cual un instrumento pudiera *hablar* a otro y viceversa. Como resultado surgió el bus de interfaz Hewlett-Packard (HPIB, *Hewlett Packard Interface Bus*).

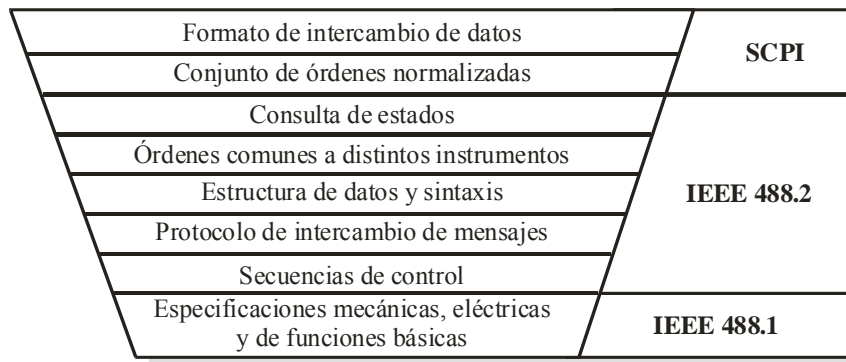
En abril de 1975, el HPIB logró la aceptación del IEEE (*Institute of Electrical and Electronics Engineers*) mediante la publicación del estándar IEEE 488/1975 “Interfaz Digital para Instrumentos Programables”. En noviembre de 1978, se revisó el estándar y como resultado se obtuvo el documento identificado como IEEE 488-1978, este último se conoce como el estándar del bus de instrumentación de propósito general (GPIB, *General Purpose Instrumentation Bus*).

En junio de 1987, el IEEE aprobó un nuevo estándar para dispositivos programables IEEE 488.2-1987 “Códigos, Formatos, Protocolos, y Órdenes comunes” y se renombró el documento original IEEE 488-1978, como IEEE 488.1.

Finalmente, el estándar IEEE 488 se forma del estándar IEEE 488.1, encargado de las especificaciones mecánicas, eléctricas y funcionales; y del estándar IEEE 488.2, el cual define la sintaxis, la estructura de los datos y la manera en que éstos se transmiten y representan por los dispositivos mediante código ASCII (*American Standard Code for Information Interchange*).

---

<sup>1</sup> En la actualidad Agilent Technologies es la división de HP dedicada a la fabricación y distribución de sistemas de medida y estimulación electrónica.



**Figura 2.1.** Relación entre los estándares IEEE 488.1, IEEE 488.2 y SCPI [8].

Hasta entonces, cada fabricante definía su propio conjunto de órdenes para cada instrumento [URL10], por ello en abril de 1990, un grupo de fabricantes de dispositivos de instrumentación electrónica anunciaron la especificación SCPI (*Standard Commands for Programmable Instruments*), la cual define un conjunto de órdenes comunes para instrumentos electrónicos programables.

En 2003, el estándar IEEE 488.1 adoptó el protocolo GPIB de alta velocidad, conocido como HS488 y propuesto por la firma National Instruments. HS488 alcanza velocidades de transferencia de datos de hasta 8 MBps y es compatible con el estándar IEEE 488.1 [URL8].

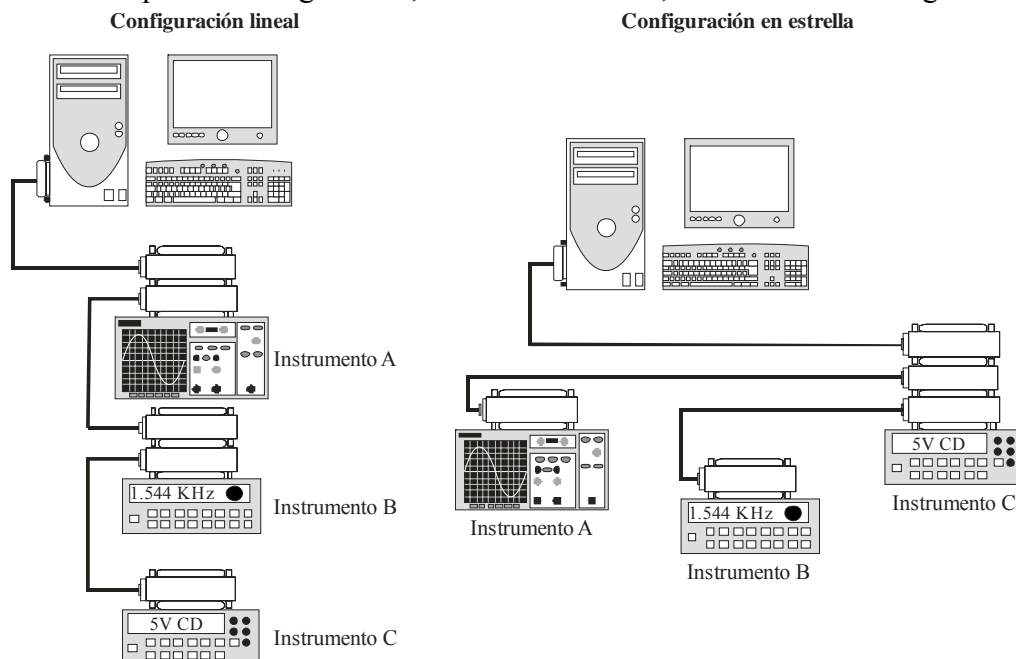
La Figura 2.1 muestra la relación entre los estándares IEEE 488 y SCPI.

## 2.2. IEEE 488.1

El estándar IEEE 488.1 permite la interconexión de hasta 15 dispositivos activos en el bus, incluyendo el controlador, define la longitud máxima del bus, la cual no debe ser mayor a 20 m, y recomienda la conexión de dispositivos al bus mediante cables GPIB de 2 m. de longitud.

La máxima velocidad de transferencia de datos es de 1 MBps (Megabyte por segundo) y se reduce a 250 o 500 KBps (Kilobyte por segundo) a distancias de bus mayores a los 20 m.

Todos los instrumentos en el bus se conectan en paralelo mediante un conector hermafrodita, lo cual permite dos tipos de configuración, lineal o en estrella, como muestra la Figura 2.2.



**Figura 2.2.** Configuración lineal y en estrella.

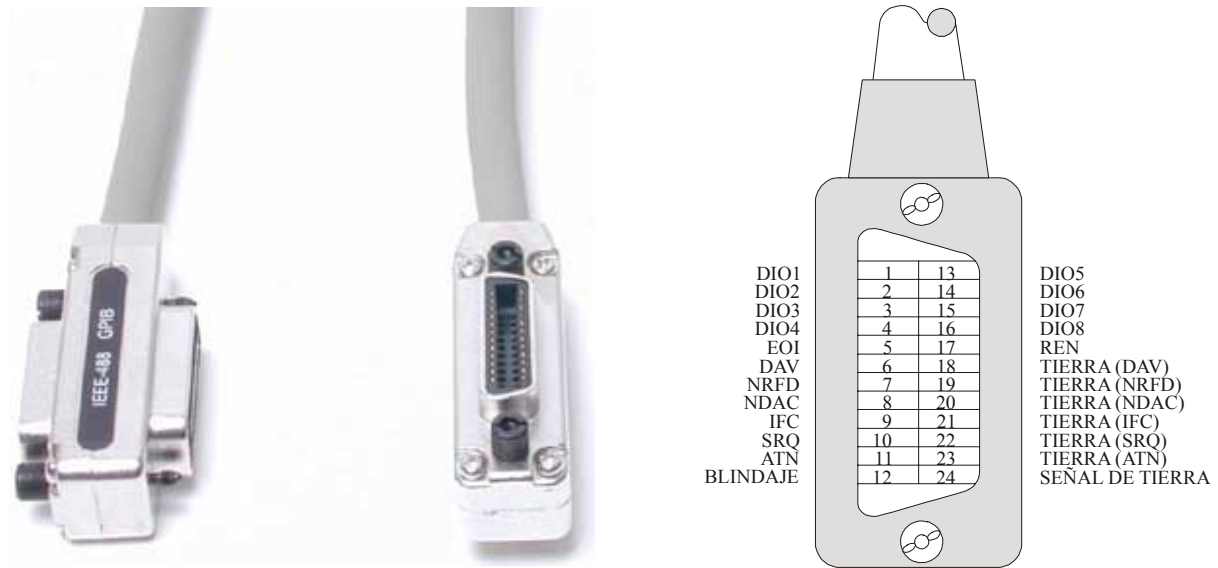


Figura 2.3. Conector GPIB y configuración de terminales.

### 2.2.1. Especificaciones eléctricas

El estándar IEEE 488.1 define un sistema basado en la lógica negativa, el cual utiliza un nivel lógico “0” para especificar un estado verdadero (*true*) y un nivel lógico “1” para especificar un estado falso (*not true*) (Tabla 2.1). Con ello se reduce la susceptibilidad de ruido en el estado verdadero y proporciona un estado falso sobre las líneas que no están conectas o en uso.

Tabla 2.1. Niveles de voltaje IEEE 488.1.

Nivel lógico	Nivel eléctrico
0	> + 2 Volts
1	< + 0.8 Volts

### 2.2.2. Especificaciones mecánicas

El estándar IEEE 488.1 especifica el tipo de conector utilizado para interconectar los dispositivos GPIB, el cual es un conector tipo americano o hermafrodita de 24 terminales (Figura 2.3).

### 2.2.3. Especificaciones funcionales

Los aspectos funcionales definen los tipos de instrumentos conectados al bus, las órdenes, datos, direcciones y funciones básicas; también considera los protocolos o mecanismos de transferencia de información y el mecanismo de reconocimiento.

#### 2.2.3.1. Tipos de instrumentos GPIB

El estándar IEEE 488.1 define tres tipos de instrumentos o dispositivos GPIB:

- *Emisor (Talker)*: dispositivo capaz de enviar información concerniente a su estado a través del bus.
- *Receptor (Listener)*: dispositivo que puede recibir órdenes y datos a través del bus cuando es direccionado.
- *Controlador (Controller)*: dispositivo encargado de controlar el sistema GPIB mediante la utilización de órdenes para el control del bus. El controlador GPIB puede ser tanto emisor como receptor, y generalmente es una PC.

Cualquier dispositivo puede ser un emisor o un receptor, pero no simultáneamente.

### 2.2.3.2. Funciones de la interfaz

El estándar IEEE 488.1 especifica diez funciones básicas que deben implementarse en todo dispositivo GPIB. Cada función cuenta con un nemónico de identificación, el cual consta de 1 a 3 letras para describir la interfaz (Tabla 2.2).

**Tabla 2.2.** Funciones básicas del estándar IEEE 488.1.

Nemónico	Capacidad	Instrumento
SH	Control de emisión	
AH	Control de recepción	Emisores
T/TE	Emisor/Emisor ampliado	y
L/LE	Receptor/Receptor ampliado	receptores
SR	Petición de servicio	
RL	Modo remoto o local	
PP	Sondeo paralelo	Controlador, emisores
DC	Inicialización	y
DT	Disparo	receptores
C	Controlador	

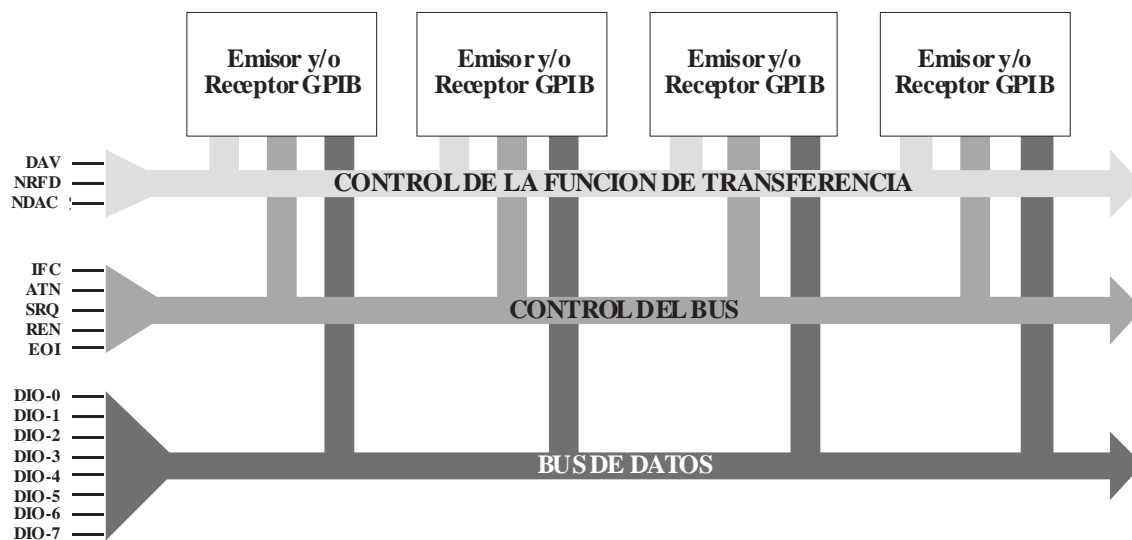
### 2.2.3.3. Estructura del bus

El GPIB es un canal de comunicación de dos vías y el flujo de los datos es en ambas direcciones (*full duplex*). El canal de comunicación que incluye cada dispositivo GPIB, carga los datos u órdenes en el bus utilizando 16 líneas (Figura 2.4):

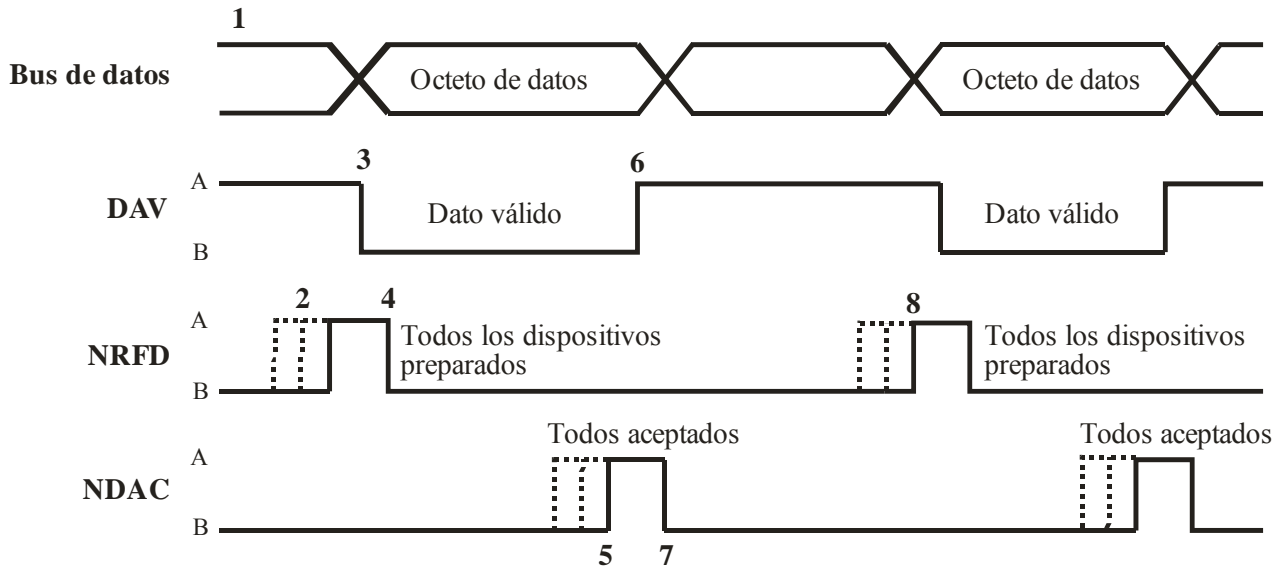
- La información de datos y direcciones se maneja mediante palabras de 8 bits (byte u octeto), mediante las líneas DIO a DI8.
- Las líneas DAV, NRFD y NDAC se utilizan para realizar la función de transferencia (*handshake*).
- Las cinco restantes proporcionan el manejo y el control del sistema GPIB.

#### 2.2.3.3.1. Función de transferencia GPIB

El propósito de la función de transferencia es asegurar que todos los datos y órdenes transmitidos sobre el GPIB sean recibidos adecuadamente por los receptores direccionados. Las líneas utilizadas para dicha función son las siguientes:



**Figura 2.4.** Estructura del bus [8].



**Figura 2.5.** Transmisión de datos usando la función de transferencia [URL8].

- *NRFD (Not Ready For Data)*: los dispositivos receptores utilizan esta línea para indicar que no pueden aceptar una transmisión de datos; para ello ponen la línea NRFD en bajo (*true*).
- *DAV (Data Valid)*: esta línea es utilizada por el emisor o controlador para validar las ocho líneas de datos en el GPIB, mediante su puesta a un nivel bajo.
- *NDAC (No Data Accepted)*: el receptor pone en bajo esta línea para indicar que aún no ha procesado los datos recibidos.

La función de transferencia se lleva a cabo de la siguiente manera (Figura 2.5):

1. El dispositivo emisor lee las líneas del bus y verifica la presencia de datos.
2. Todos los receptores están listos para recibir datos, y ponen su línea NRFD a un nivel alto (*false*).
3. El emisor pone la línea DAV a un nivel bajo para validar los datos en el bus.
4. Los receptores aceptan el primer dato y ponen la línea NRFD a un nivel bajo para procesar el dato e indicar que no están listos para recibir el siguiente octeto.
5. Cuando el receptor más lento recibe y acepta el dato, la línea NDAC es puesta en un nivel alto indicado que todos los receptores en el bus han aceptado el dato.
6. El emisor pone la línea DAV en un alto para indicar que los datos en el bus no son válidos.
7. Los receptores colocan la línea NDAC a un nivel bajo, preparándose para recibir el próximo octeto.
8. Los receptores liberan la línea NRFD, mediante un nivel alto, y se repite el proceso.

#### 2.2.3.3.2. Líneas de manejo de interfaz

Se definen cinco líneas del bus para el manejo de la interfaz, las cuales proporcionan un flujo ordenado de órdenes y datos:

- *ATN (Attention)*: puesta en nivel bajo indica que los datos cargados en el bus se refieren a una orden o a una dirección. En conjunto con la línea EOI, se utilizan para indicar un sondeo en paralelo.
- *EOI (End Or Identify)*: esta línea tiene dos funciones:

- Si proviene del controlador: la línea en bajo indica, junto con la línea ATN en bajo un sondeo en paralelo.
- Si proviene de un emisor: la línea en bajo indica junto con un octeto de datos que éste es el último octeto en transferirse.
- *IFC (Interface Clear)*: el controlador GPIB utiliza esta línea para inicializar<sup>2</sup> el bus.
- *REN (Remote Enable)*: es utilizada por el controlador GPIB para configurar los dispositivos en modo local o remoto.
- *SRQ (Service Request)*: cualquier dispositivo activo en el bus puede ocupar esta línea para realizar una petición de servicio al controlador GPIB.

#### 2.2.3.4. Direcciones GPIB

A cada dispositivo se asigna una dirección única, para ello el protocolo GPIB define un total de 31 direcciones llamadas direcciones primarias. Se asigna una dirección primaria a cada dispositivo mediante la configuración de interruptores (*switches* o *jumpers*) localizados en la parte posterior (*rear panel*) de los dispositivos. Actualmente, los dispositivos configuran su dirección mediante herramientas software o desde su panel frontal (*front panel*).

Las direcciones válidas pueden tomar un valor de 0 a 30, se reserva la dirección número 31 para propósitos de control y la dirección 21 para controlar la dirección del emisor/receptor.

Las direcciones primarias especifican dos códigos de direcciones sobre las líneas de datos, éstas diferencian entre las direcciones de emisor y las direcciones de receptor definidas en los bits 6 y 7 del octeto de datos.

#### 2.2.3.5. Órdenes GPIB

El estándar IEEE 488.1 define cuatro tipos de órdenes mediante los bits 7, 6 y 5 de la misma orden (Tabla 2.3).

**Tabla 2.3.** Especificación del tipo de orden GPIB.

Bit 7	Bit 6	Bit 5	Tipo de orden
0	0	0	Direccionadas
0	0	1	Universales multilinea
0	1	X	Receptor
1	0	X	Emisor
1	1	X	Secundarias

##### 2.2.3.5.1. Órdenes direccionadas

Los dispositivos en el bus no necesariamente deben responder a todas las órdenes del controlador GPIB, por ello se definen las órdenes direccionadas, las cuales se utilizan para operar en el modo de operación remota. Las órdenes direccionadas son las siguientes:

- *GET (Grup Execute Trigger)*: el dispositivo direccionado inicia una acción programada previamente.
- *SDC (Selected Device Clear)*: esta orden pone al dispositivo direccionado en un estado predeterminado (*reset*).

<sup>2</sup> Se refiere a configurar el bus en un estado conocido o predeterminado.



- *GTL (Go To Local)*: obliga a que el panel frontal del dispositivo direccionado cambie de modo de control remoto a modo de control manual.
- *PPC (Parallel Poll Configure)*: cuando el controlador GPIB realiza un sondeo en paralelo, direcciona al dispositivo y éste responde en la línea de datos correspondiente para indicar su estado actual.

#### 2.2.3.5.2. Órdenes universales multilínea

Existen cinco tipos de órdenes universales multilínea, las cuales afectan a todos los dispositivos conectados al bus y son transmitidas en la línea de datos:

- *DCL (Device Clear)*: tiene la función de regresar a cualquier dispositivo a un estado inicial.
- *LLO (Local Lockout)*: se utiliza para deshabilitar los controles del panel frontal del dispositivo direccionado.
- *SPE (Serial Poll Enable)*: mediante esta orden, todos los emisores en el bus asumen el modo de sondeo serie.
- *SPD (Serial Poll Disable)*: esta orden regresa a cada emisor en el bus a su estado normal después de haberse realizado un sondeo serie.
- *PPU (Parallel Poll Unconfigure)*: los dispositivos que reconocen esta orden se inicializan y pasan a un estado de paro, deshabilitándolos para responder a dicha orden.

#### 2.2.3.5.3. Órdenes de receptor y emisor

Este tipo de órdenes tienen la finalidad de identificar al receptor y al emisor durante una transmisión de datos. Para cambiar de modo emisor o receptor a un estado de reposo, el controlador GPIB envía las órdenes *Untalk* (1011111) y *Unlisten* (0111111) respectivamente, las cuales son técnicamente clasificadas como direcciones:

- *UNL (Unlisten)*: desactiva todos los receptores.
- *UNT (Untalk)*: desactiva todos los emisores.

#### 2.2.3.5.4. Órdenes secundarias

Las órdenes secundarias que define el estándar IEEE 488.1 son:

- *PPE (Parallel Poll Enable)*: configura los dispositivos para responder a un sondeo paralelo sobre una línea particular de E/S (DI/O).
- *PPD (Parallel Poll Disable)*: inhabilita el sondeo paralelo.

#### 2.2.3.6. Sondeo

El controlador GPIB cuenta con dos formas de conocer el estado actual de los dispositivos activos en el bus:

- *Sondeo serie (Serial Polling)*: se direcciona individualmente a cada dispositivo activo en el bus y éste regresa el octeto de estado, el cual indica su condición actual. La ventaja de este tipo de sondeo es que el controlador obtiene la identidad del dispositivo sondeado y su respectivo octeto de estado, y su principal desventaja es el tiempo que se requiere para realizar el sondeo de varios dispositivos.
- *Sondeo paralelo (Parallel Poll)*: esta forma de sondeo es más rápida que el sondeo serie, pero únicamente pueden ser sondeados hasta ocho dispositivos al mismo tiempo debido a que el controlador GPIB asigna a cada dispositivo un bit del octeto de estado [8].

## 2.3. IEEE 488.2

El estándar IEEE 488.2 fue desarrollado para eliminar los problemas y ambigüedades que presentaba el estándar original, por ejemplo:

- No definía un formato común para la representación de los datos.
- No definía un protocolo estándar de mensajes.
- No contaba con un conjunto de órdenes comunes a todos los dispositivos.
- El octeto de estado de cada dispositivo GPIB era diferente.

Para solucionar tales problemas, el estándar IEEE 488.2 proporciona las siguientes soluciones:

- Define formatos de datos que van desde números decimales a cadenas de caracteres.
- Describe un protocolo de intercambio de mensajes, que indican al dispositivo cómo reaccionar cuando ocurren condiciones no establecidas.
- Define un conjunto de órdenes comunes para una comunicación uniforme.
- Define un modelo de informe de estado para los instrumentos GPIB.

### 2.3.1. Sintaxis y formato de datos

El estándar IEEE 488.2 especifica el formato de los datos para cualquier tipo de mensaje que pueda ser enviado, incluyendo números y cadenas de caracteres, en el GPIB. Los tipos de números incluidos en el estándar son binario, octal y hexadecimal.

### 2.3.2. Codificación

El IEEE 488.2 define un conjunto de códigos para representar el tipo de datos (Anexo B):

- *Caracteres alfanuméricos*: ASCII 7 bits.
- *Caracteres numéricos enteros*: binario de 8 bits.
- *Números de punto flotante*: Recomendación IEEE 754.

### 2.3.3. Reporte de estado

El reporte de estado definido por el estándar IEEE 488.2 tiene la finalidad de extender la especificación del estándar original. El documento IEEE 488.1 no define los bits que forman el octeto de estado, dejando la implementación en mano de los diseñadores; tampoco define la forma en que se activan y/o desactivan los bits del octeto de estado.

El estándar IEEE 488.2 define los bits 4, 5 y 6 del octeto de estado, y las órdenes que permiten al usuario acceder al octeto de estado y a las estructuras de datos asociadas:

- *Bit 4 MAB (Message Available Bit)*: indica si la cola de salida (*output queue*) está vacía.
- *Bit 5 ESB (Event Status Bit)*: indica si se ha producido algún suceso en el registro de estado.
- *Bit 6 MSS (Master Summary Status Bit)*: indica si el dispositivo requiere solicitar una petición de servicio.

Para habilitar una petición de servicio, se utiliza el registro de habilitación de petición de servicio (SRER, *Service Request Enable Register*). El registro estándar de estado de evento (SESR, *Standard Event Status Register*) registra los cambios ocurridos en los distintos dispositivos, cada bit corresponde a una condición determinada; cuando se desea que una determinada situación sea detectada, se activa el bit correspondiente del registro estándar de habilitación de estado de evento (SE-SER, *Standard Event Status Enable Register*).

### 2.3.4. Órdenes comunes

Todos los dispositivos GPIB realizan un conjunto de funciones comunes para comunicarse a través del bus e informar de su estado actual. Antes de definir el estándar IEEE 488.2, cada dispositivo utilizaba su propio conjunto de órdenes para realizar sus funciones.

Las órdenes comunes no son específicas de los instrumentos y algunas son opcionales. El siguiente conjunto de órdenes asegura que todos los dispositivos conectados al bus se comuniquen adecuadamente:

- *Datos de sistema (System data)*: se utiliza para almacenar o recuperar información de los dispositivos conectados al sistema.
- *Operaciones internas (Internal operations)*: para realizar operaciones internas del dispositivo tales como reinicialización, autocalibración, prueba, etc.
- *Estado y evento (Status & Event)*: relacionados con el modelo de reporte de estado.
- *Sincronización (Synchronization)*: realiza la sincronización de las operaciones del dispositivo dentro del sistema GPIB.
- *Sondeo paralelo (Parallel poll)*: para realizar el sondeo paralelo.
- *Activación del dispositivo (Device trigger)*: activa el funcionamiento de los dispositivos.
- *Controlador (Controller)*: pasa la función de controlador a otro dispositivo del sistema.
- *Autoconfiguración (Autoconfigure)*: asigna direcciones a los dispositivos del sistema.
- *Macros*: defini nuevas órdenes con base en las definidas por el estándar IEEE 488.2.
- *Almacenar configuración (Store Settings)*: almacena el estado de un dispositivo.

## 2.4. Órdenes estandarizadas para instrumentos programables

La especificación SCPI unifica la información y las órdenes para dispositivos de distintos fabricantes, y puede interactuar de dos formas:

- *Compatibilidad vertical*: instrumentos de la misma clase pero de distintos fabricantes.
- *Compatibilidad horizontal*: mismas órdenes para controlar las mismas funciones de distintos instrumentos.

Se pueden definir nuevas órdenes para funciones no previstas siguiendo las reglas establecidas en la especificación SCPI. Cualquier compañía puede proponer nuevas órdenes, las cuales deben ser aprobadas por el consorcio SCPI, y una vez aceptadas se incluyen en la especificación [URL10].

### 2.4.1. Formato de datos

SCPI presenta problemas en el tratamiento de los datos debido a que:

- Los formatos numéricos (Binario, BCD, ASCII, etc.) y la estructura de los datos (agrupados, separados por comas, con cabecera, etc.) dependen de cada dispositivo.
- La información enviada, en muchos casos, no es autosuficiente.

Para solucionar lo anterior, se utiliza el formato *ADIF (Analog Data Interchange Format)*, el cual encapsula la información. Cada bloque contiene los datos y la información necesaria sobre la naturaleza de los mismos.

Los bloques del formato *ADIF* son los siguientes:

- *ADIF*: bloque general.
- *STD*: versión ADIF que se implementa.

- *IDENTIFY*: identifica los datos (fecha y hora).
- *ENCODE*: especifica formato numérico.
- *DIMENSION*: informa sobre los factores de conversión y unidad.
- *ORDER*: orden o secuencia de los datos.
- *TRACE*: agrupa lógicamente las variables.
- *VIEW*: agrupa lógicamente las trazas<sup>3</sup>.
- *DATA*: contiene los datos.

La Tabla 2.4 resume las especificaciones del estándar IEEE 488.

**Tabla 2.4.** Especificaciones básicas del estándar IEEE 488 [8].

Estándar	Especificaciones	Concepto	Descripción
IEEE 488.1	Mecánicas	Configuración del bus	En estrella ( <i>Star</i> ) o lineal ( <i>Lineal</i> ).
		Longitud del bus	2 m. entre instrumentos ( <i>Máxima longitud 20 m.</i> ).
		Conector	Instrumento ( <i>hembra</i> ).
			Cable ( <i>apilable: macho y hembra</i> ).
	Eléctricas	“1” lógico en la salida	0 ÷ 0,5 V.
		“1” lógico en la entrada	-0,6 ÷ 0,8 V.
		“0” lógico en la salida	2,4 ÷ 5 V.
		“0” lógico en la entrada	2 ÷ 5,5 V.
	Funcionales y de procedimiento	Tipo de instrumentos	Controlador ( <i>Controller</i> ), Emisor ( <i>Talker</i> ), Receptor ( <i>Listener</i> ).
		No. de instrumentos conectables	Máximo 15.
		No. de instrumentos activos	Más de la mitad de los instrumentos conectados deben estar activos.
		Velocidad	Máxima 1 MBps ( <i>distancias largas 200-300 KBps</i> ).
		Longitud de datos y órdenes	8 bits ( <i>Octeto</i> ) en código ASCII o binario.
		Tipos de órdenes	34 principales y 32 secundarias.
		Dirección de instrumento	Interruptores (6) o memoria pasiva.
		Direcciones del estándar	Emisores ( <i>Talkers</i> ): 31 principales y 32 secundarios. Receptores ( <i>Listeners</i> ): 31 principales y 32 secundarios.
		Funciones básicas de los instrumentos	10 (no presentes en todos los tipos de instrumentos).
		Mecanismos de sondeo	Serie ( <i>Serial polling</i> ) y Paralelo ( <i>Parallel polling</i> ).
		IEEE 488.2	De procedimiento

<sup>3</sup> Las trazas son utilizadas en dispositivos desplegados como osciloscopios o analizadores.

### 3. Bluetooth

Los avances tecnológicos en el campo de las computadoras personales (PC) ha incrementado el número de periféricos asociados, como son impresoras, bocinas, unidades de almacenamiento secundario, agendas electrónicas (PDAs, *Personal Digital Assistants*), etc. La mayoría de dichos periféricos requiere de una conexión punto a punto con la PC, lo cual incrementa el número de conexiones y con ello los problemas de cableado correspondientes. Además, cabe mencionar que los cables que conectan a dichos dispositivos utilizan una amplia variedad de conectores con diferentes tamaños, formas y número de terminales. Lo anterior ha dado lugar a definir un nuevo tipo de red denominada red de área personal (PAN, *Personal Area Network*).

Para resolver el problema anterior, se propuso la tecnología Bluetooth, la cual hace referencia a la especificación abierta de una tecnología que proporciona comunicación inalámbrica de corto rango para voz y datos. Dicha especificación incluye las siguientes características:

- *Especificación abierta*: el grupo de interés especial (SIG, *Special Interest Group*) de Bluetooth creó una especificación para comunicación inalámbrica disponible públicamente, y de manera gratuita, con la finalidad de alcanzar su estandarización internacional.
- *Rango corto*: mediante Bluetooth, los dispositivos pueden interconectarse mediante una interfaz aérea, utilizando ondas de radio para transmitir y recibir datos. La tecnología Bluetooth está diseñada para un rango corto de comunicación (10 m.) con bajo consumo de potencia.
- *Voz y datos*: algunas tecnologías de comunicaciones inalámbricas fueron diseñadas específicamente para transmisión de voz o datos, mientras que Bluetooth puede transmitir ambos.
- *Uso global*: la industria de las telecomunicaciones es regulada en muchas partes del mundo y generalmente el uso del espectro de radio frecuencias requiere de una licencia con estrictas obligaciones. Sin embargo, existen porciones de dicho espectro cuya utilización no requiere de una licencia, en una de ellas opera Bluetooth.

La tecnología Bluetooth es ideal para el reemplazo del cableado necesario para la interconexión de los dispositivos con una PC. La especificación de Bluetooth define una tecnología inalámbrica para reemplazar cables seriales, la cual también puede utilizarse para reemplazar otro tipo de cables como son el cable paralelo y los enlaces infrarrojos [URL4].

#### 3.1. Soluciones inalámbricas

Como la mayoría de especificaciones técnicas, la especificación de Bluetooth es una respuesta a los requerimientos del mercado. Originalmente, el grupo de trabajo de mercadotecnia del SIG ge-

neró un documento de requerimientos de mercado (MRD, *Marketing Requirements Document*), el cual incluye objetivos y modelos de uso. El propósito principal de dicha especificación es definir componentes para satisfacer los requerimientos del mercado y dentro de sus objetivos están: crear una especificación abierta, uso global sin licencia, bajo costo e interoperabilidad.

A continuación se describen las principales tecnologías inalámbricas utilizadas como soluciones empleando medios no guiados [12].

### 3.1.1. Infrarrojo

Algunos dispositivos pueden comunicarse a través de conexiones ópticas como es el infrarrojo, el cuál es un sistema de comunicaciones que utiliza luz infrarroja con una longitud de onda de 850 nm., entre dispositivos para transferir voz y datos. Se debe contar con una línea de vista que represente una trayectoria directa entre los dispositivos a comunicar, éstos a su vez deben estar posicionados cerca debido a que la conexión trabaja a distancias cortas de aproximadamente 5 m. de longitud. El infrarrojo está enfocado a enlaces punto a punto entre dos dispositivos para transferencia de datos y sincronización, la Tabla 3.1 muestra sus principales características.

**Tabla 3.1.** Principales características de la tecnología de infrarrojos.

Característica/Función	Descripción
Tipo de conexión	Infrarrojo, luz angosta (ángulo de 30 grados o menos)
Espectro	Óptico, 850 nm
Potencia de transmisión	100 mW
Velocidad de transferencia	10 Mbps y hasta 16 Mbps usando VFIR ( <i>Very Fast Infrared</i> )
Rango	Hasta 5 m.
Dispositivos soportados	2
Canales de voz	1
Seguridad de datos	El rango corto y ángulo angosto del haz infrarrojo proporciona seguridad
Direccionamiento	Cada dispositivo tiene un identificador físico de 32 bits, utilizado para establecer una conexión con otro dispositivo

### 3.1.2. IEEE 802.11

Otra opción de conectividad inalámbrica es la destinada a redes de área local (LAN, *Local Area Network*), descrita en el estándar IEEE 802.11. Las redes inalámbricas IEEE 802.11 y Bluetooth están enfocadas a aplicaciones completamente diferentes, ya que los dispositivos Bluetooth requieren baja potencia para transmitir pequeñas cargas de datos a 1 Mbps a distancias de hasta 10 m., mientras que los dispositivos IEEE 802.11 pueden transmitir datos hasta 11 Mbps en distancias menores a 100 m., para comunicación de datos en oficinas corporativas o campus donde no es viable la instalación mediante cableado. La Tabla 3.2 muestra las principales características de las LANs inalámbricas IEEE 802.11.

**Tabla 3.2.** Principales características de la tecnología IEEE 802.11.

Característica/Función	Descripción
Tipo de conexión	Espectro expandido (secuencia directa o salto de frecuencia)
Espectro	Banda ISM de 2.4 GHz
Potencia de transmisión	100 mW
Velocidad de transferencia	1 o 2 Mbps utilizando salto de frecuencia; 11 Mbps utilizando secuencia directa
Rango	Hasta 100 m. entre el punto de acceso ( <i>access point</i> ) y el Cliente
Dispositivos soportados	Múltiples dispositivos por punto de acceso; múltiples puntos de acceso por red
Canales de voz	Voz sobre IP (VoIP, <i>Voice over IP</i> )
Seguridad de datos	Autenticación: llamada-respuesta entre el punto de acceso y el Cliente vía WEP ( <i>Wired Equivalent Privacy</i> ) Encriptación: estándar 40 bits; 128 opcionales
Direccionamiento	Cada dispositivo tiene una dirección MAC de 48 bits para establecer una conexión con otro dispositivo

### 3.1.3. HomeRF

Otra tecnología inalámbrica es la especificación conocida como HomeRF (*Home Radio Frequency*), apoyada por más de 100 firmas que conforman el consorcio HomeRF. HomeRF proporciona la base para la comunicación de una amplia gama de dispositivos dentro del hogar mediante la especificación abierta de un protocolo de comunicaciones digitales inalámbricas.

HomeRF está basado en el SWAP (*Shared Wireless Access Protocol*), opera a 2 Mbps y eventualmente permite velocidades de transferencia de hasta 10 Mbps. Al igual que Bluetooth, HomeRF utiliza salto de frecuencia en espectro expandido (Inciso 3.4.1.2) para transmitir de manera confiable y segura voz y datos. La Tabla 3.3 resume las principales características de operación de HomeRF.

**Tabla 3.3.** Principales características de la tecnología HomeRF.

Característica/Función	Descripción
Tipo de conexión	Espectro expandido (salto de frecuencia)
Espectro	Banda ISM de 2.4 GHz
Potencia de transmisión	100 mW
Velocidad de transferencia	1 o 2 Mbps, dependiendo del esquema de modulación mediante salto de frecuencia
Rango	137 m.; cubre típicamente casa y jardín
Dispositivos soportados	Hasta 127 dispositivos por red
Canales de voz	6
Seguridad de datos	Algoritmo de encriptación <i>Blowfish</i> (sobre un trillón de códigos)
Direccionamiento	Cada dispositivo tiene una dirección MAC de 48 bits para establecer una conexión con otro dispositivo

### 3.1.4. Bluetooth

La especificación de Bluetooth es una tecnología global de bajo costo y potencia, tamaño pequeño, comunicación inalámbrica y comunicación con PCs, teléfonos móviles y otros dispositivos portátiles. Bluetooth elimina el cableado, las conexiones infrarrojas y no requiere de un adaptador inalámbrico especial con antena.

La tecnología Bluetooth fue diseñada para soportar transferencia de datos a 721 Kbps y múltiples canales de voz en un rango de 10 m. de longitud.

Bluetooth opera en el rango de 2.4 GHz y aunque la distancia que cubre es aproximadamente de 10 m., ésta se puede extender hasta 100 m. mediante amplificadores. Una de las principales ventajas de Bluetooth es que no requiere de un proceso de instalación ya que cumple con la norma *Plug&Play* de los sistemas operativos modernos. La Tabla 3.4 muestra las principales características de Bluetooth.

**Tabla 3.4.** Principales características de la tecnología Bluetooth.

Característica/Función	Descripción
Tipo de conexión	Espectro expandido (salto de frecuencia)
Espectro	Banda ISM de 2.4 GHz
Potencia de transmisión	100 mW
Velocidad de transferencia	1 Mbps utilizando salto de frecuencia
Rango	10 m.
Dispositivos soportados	Hasta 8 dispositivos por pired
Canales de voz	3
Seguridad de datos	Una clave de 128 bits para autenticación y una clave configurable entre 8 y 128 bits para encriptación
Direccionamiento	Cada dispositivo tiene una dirección MAC de 48 bits para establecer una conexión con otro dispositivo

La elección de la tecnología para una red inalámbrica depende de la aplicación, para ello la Tabla 3.5 lista las aplicaciones típicas de cada una de las tecnologías descritas.

**Tabla 3.5.** Aplicaciones típicas de las tecnologías inalámbricas.

Tecnología	Tipo de comunicación	Tipo de redes	Aplicaciones
Infrarrojo	Datos	Punto a punto	Transferencia de archivos a alta velocidad, control de dispositivos locales, etc.
IEEE 802.11	Voz y datos	LANs	Comunicación de datos en oficinas corporativas o campus.
HomeRF	Datos	Domésticas	Comunicación de datos entre computadoras y periféricos en el hogar o en oficinas pequeñas
Bluetooth	Voz y datos	PANs	Control de dispositivos remotos, comercio electrónico móvil, etc.

### 3.2. Grupo de interés especial Bluetooth

La especificación de Bluetooth es el resultado de la cooperación realizada por diversas firmas que conforman el SIG, dicho grupo fue creado con la finalidad de definir y promover la tecnología Bluetooth [URL4].

La tecnología inalámbrica Bluetooth fue desarrollada en Suecia por ingenieros de la firma de telecomunicaciones Telefonaktiebolaget LM Ericsson, actualmente Ericsson. En 1994, Ericsson inició un proyecto para estudiar la posibilidad de crear una interfaz de radio, de baja potencia y bajo costo, para eliminar el cableado entre los teléfonos móviles y sus accesorios.

Dentro de las áreas de la computación y de las telecomunicaciones, los nuevos desarrollos tecnológicos se basan en estándares internacionales, asegurando con ello un mercado de proveedores con bajos costos de adquisición. Ericsson reconoció que su proyecto podría ser ampliamente aceptado, si éste fuera adoptado y mejorado por un conjunto de firmas para obtener una especificación abierta y con el tiempo lograr su estandarización internacional.

En 1998, Ericsson atrajo la atención de las firmas líderes en la industria de la computación y de las telecomunicaciones para formar el SIG Bluetooth. Las firmas fundadoras (*Promoter Members*) del SIG son Ericsson, Intel, IBM, Nokia y Toshiba.

Conforme se fue desarrollando la especificación de Bluetooth, el SIG fue creciendo mediante la integración de nuevas firmas adoptadoras (*Adopter Members*)<sup>4</sup>. La versión 1.0 de la especificación fue publicada en julio de 1999, la cual consiste en dos documentos:

- *La fundación del núcleo (Bluetooth Specification Version 1.0 Foundation Core)*: el cual proporciona especificaciones de diseño utilizando la arquitectura de radio Bluetooth (*Bluetooth Radio Architecture*).
- *La fundación de perfiles (Bluetooth Specification Version 1.0 Foundation Profiles)*: que proporciona las recomendaciones para asegurar la interoperabilidad entre dispositivos desarrollados mediante tecnología Bluetooth.

El SIG está organizado por los siguientes grupos de trabajo, cada uno con un enfoque específico en tecnología y servicio de soporte:

- *Interfaz aérea*: enfocado en las capas de radio y de banda base.
- *Diseño y desarrollo de software*: encargado de la especificación de la arquitectura del protocolo.
- *Interoperabilidad*: enfocado en la descripción del sistema Bluetooth.

<sup>4</sup> Estos miembros tienen acceso a una licencia, libre de derechos de autor, con la finalidad de utilizar la especificación de Bluetooth para fabricar productos de comunicación inalámbrica Bluetooth, así también pueden aportar mejoras a dicha especificación.



- *Acatamientos*: define los procesos de prueba y certificación.
- *Legal*: maneja asuntos legales, tales como membresías.
- *Mercadotecnia*: promueve la tecnología Bluetooth.

En diciembre de 1999, las firmas 3Com, Lucent Technologies, Microsoft y Motorola, que inicialmente habían contribuido a la especificación original como compañías adoptadoras, se integraron al SIG.

### 3.3. Historia de Bluetooth

La mayoría de campos industriales se identifican mediante un nombre, el cual describe o asocia su tecnología de aplicación. Existen varias historias acerca de la elección del nombre Bluetooth, la más aceptada es la que se menciona a continuación.

Harald Blåtand fue rey de Dinamarca durante los años 940 y 985 d.C., a dicho rey se le atribuye la unificación de Dinamarca y Noruega, y la introducción del cristianismo a Escandinavia. La palabra Blåtand es de origen vikingo y significa “*Diente Azul*”, se considera que al rey Harald se le llamaba Blåtand debido a que continuamente tenía los dientes con un tono azul resultado de comer zarzamoras.

Debido a que la tecnología en cuestión fue desarrollada en Escandinavia, sus fundadores consideraron apropiado nombrarla como resultado de unificar compañías multinacionales de igual forma que el rey Blåtand unificó países. De esta manera surgió el nombre de Bluetooth, que actualmente a llegado a ser una marca comercial de gran aceptación. La Figura 3.1 muestra el emblema de la tecnología Bluetooth<sup>5</sup>, inspirado en las iniciales del rey Harald Blåtand.

### 3.4. Tecnología Bluetooth

La comunicación inalámbrica no es un concepto nuevo, la transmisión de radio, televisión, comunicación vía satélite, teléfonos celulares y controles remotos para televisión son ejemplos de aplicaciones reales. La mayoría de ellos utilizan ondas de radio y algunos infrarrojos, Bluetooth hace uso de RF de manera similar a la transmisión de radio o televisión [12].

#### 3.4.1. RF

La tecnología RF utiliza transmisores y receptores sintonizados para la transmisión de ondas de radio con un rango de frecuencias específico. La potencia del transmisor y la sensibilidad del receptor ayudan a determinar las distancias de comunicación. Una característica especial de las ondas de radio es la habilidad de penetrar objetos, característica que no comparten las ondas infrarrojas ya que los objetos reflejan sus ondas de luz. Por lo anterior, RF no requiere una línea de vista entre el transmisor y el receptor.

En RF se utiliza la modulación en frecuencia (FM, *Frequency Modulation*) para generar ondas de radio en un determinado espectro de frecuencia. La transmisión de radio en FM opera en el rango de 88 a 108 MHz, mientras que Bluetooth opera en los 2.4 GHz del espectro electromagnético.



**Figura 3.1.** Emblema de la tecnología Bluetooth.

<sup>5</sup> Bluetooth es una marca comercial propiedad de la firma sueca Telefonaktiebolaget LM Ericsson.

Debido a que el espectro útil de RF es finito, algunos países regulan su uso mediante la división en rangos de frecuencias y otorgan licencias para su utilización, sin embargo algunas frecuencias están reservadas y bajo ciertas condiciones no requieren licencia de uso.

El SIG junto con otras organizaciones, como el IEEE, han trabajado con autoridades regulatorias en algunos países con el propósito de adecuar las asignaciones de frecuencia para el uso sin licencia dentro del espectro de 2.4 GHz.

### 3.4.1.1. RF en el espectro de 2.4 GHz

El sistema Bluetooth opera en la banda industrial, científica y médica (ISM, *Industrial, Scientific and Medical*) de 2.4 GHz. Las bandas ISM son bandas libres que no requieren licencia de operación para uso en sistemas industriales, científicos y médicos. La Tabla 3.6 muestra las frecuencias ISM disponibles en la mayoría de países alrededor del mundo.

**Tabla 3.6.** Asignación de frecuencias en la banda 2.4 GHz.

Banda	Rango de frecuencias
UHF ISM	902 - 928 MHz
Banda S	2 - 4 GHz
Banda S ISM	2.4 - 2.5 GHz
Banda C	4 - 8 GHz
Banda C Satélite (Descendente)	3.7 - 4.2 GHz
Banda C Radar (Clima)	5.25 - 5.925 GHz
Banda C ISM	5.725 - 5.875 GHz
Banda C Satélite (Ascendente)	5.925 - 6.425 GHz
Banda X	8 - 12 GHz
Banda X Radar (Policía/Clima)	8.8 - 10.55 GHz
Banda Ku	12 - 18 GHz
Banda Ku Radar (Policía)	13.4 - 14 GHz

En todo el mundo, el espectro de 2.4 GHz no requiere licencia alguna para su utilización, pero es necesario cubrir algunas regulaciones y consideraciones, entre ellas:

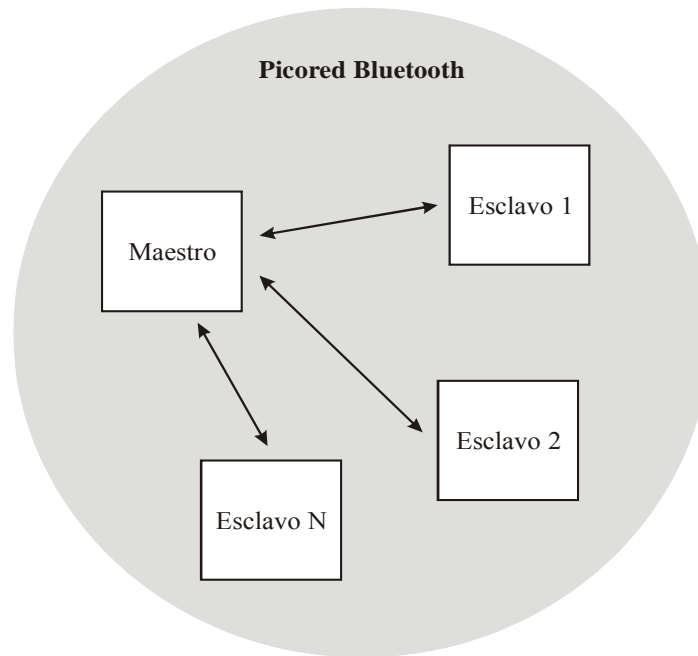
- El espectro está dividido en 79 canales.
- El ancho de banda es limitado a 1 MHz por canal.
- Debe utilizarse la técnica de espectro expandido en salto de frecuencia (FHSS, *Frequency-Hopping Spread-Spectrum*)

HomeRF e IEEE 802.11 hacen uso de este espectro, así como también los hornos de microondas utilizados en el hogar.

### 3.4.1.2. Espectro expandido

Dentro de las comunicaciones de RF, el espectro expandido se refiere a dividir el espectro disponible en cuanto a frecuencia, tiempo, código o algún otro método. Para enviar un mensaje, éste se divide en paquetes que posteriormente son transmitidos a través del espectro dividido. El espectro expandido por división de frecuencia o salto de frecuencia (FHSS), divide el espectro en diferentes rangos de frecuencias. Un paquete se transmite en un canal seleccionado, posteriormente se selecciona un nuevo canal<sup>6</sup> para transmitir el próximo paquete, y el proceso se repite. El receptor del mensaje debe conocer el patrón de saltos para sintonizar el canal correcto, recibir cada paquete y ensamblar el mensaje completo.

<sup>6</sup> Proceso llamado salto de frecuencia.



**Figura 3.2.** Piconet Bluetooth.

### 3.5. Principios de operación Bluetooth

Cuando dos dispositivos establecen un enlace Bluetooth, uno actúa como Maestro (*Master*) y el otro como Esclavo (*Slave*). La especificación permite asumir a los dispositivos cualquier rol, permitiendo que un dispositivo actúe como Maestro en un enlace y como Esclavo en otro.

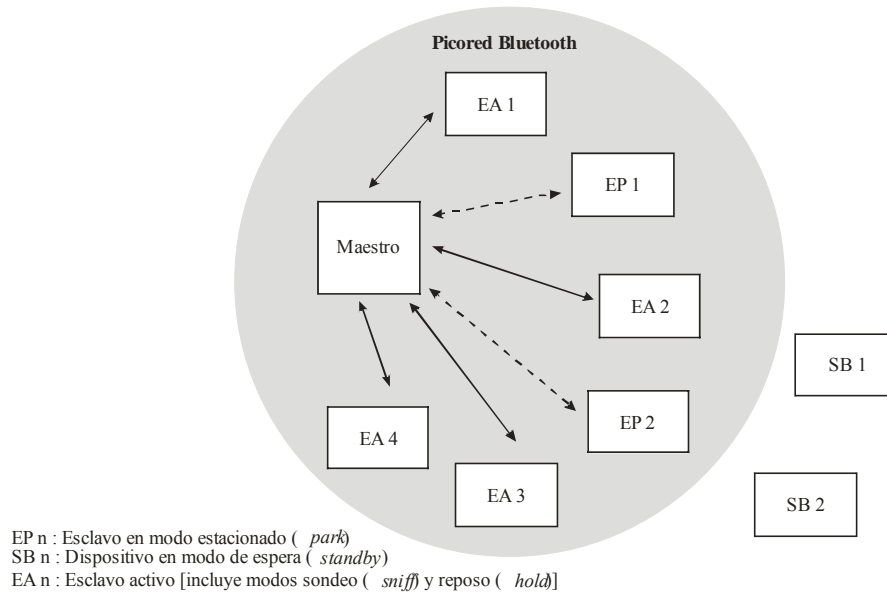
Un Maestro puede comunicarse con múltiples Esclavos, 7 activos y 255 en espera; todos los Esclavos deben comunicarse con un sólo Maestro, formando lo que la especificación llama una piconet (*piconet*). La Figura 3.2 muestra los roles de Maestro y Esclavo dentro de una piconet Bluetooth.

El rol de Maestro no implica privilegios especiales, únicamente dirige la sincronización de la comunicación entre los dispositivos. El dispositivo Maestro determina el patrón de salto de frecuencia y la fase para la secuencia de los saltos; generalmente el dispositivo que inicia la comunicación actúa como Maestro aunque algunas circunstancias pueden necesitar que el rol del Maestro y del Esclavo se intercambien; dicho intercambio comprende el proceso de migración de la secuencia de tiempo de transmisión en la piconet del anterior Maestro a la secuencia de tiempo de transmisión del nuevo Maestro [11].

### 3.6. Topología Bluetooth

El modelo de red utilizado en Bluetooth está basado en una red de comunicaciones P2P (*Peer to Peer*). Cuando un dispositivo se encuentra dentro del rango de otro dispositivo, automáticamente establecen un enlace de comunicación. Los dispositivos no siempre inician una comunicación en forma automática, éstos se pueden configurar para aceptar o rechazar el enlace con determinados dispositivos.

La Figura 3.3 ilustra la topología de una piconet Bluetooth, la cual consiste de un dispositivo Maestro y uno o varios Esclavos que se encuentren dentro de su cobertura (Figura 3.3). Puede existir el caso en que se encuentren dispositivos en la proximidad del Maestro, pero que no estén conectados a él y por lo tanto no forman parte de la piconet, incluyendo dispositivos en modo de reposo (*standby*) (Inciso 3.7.2.3.4).



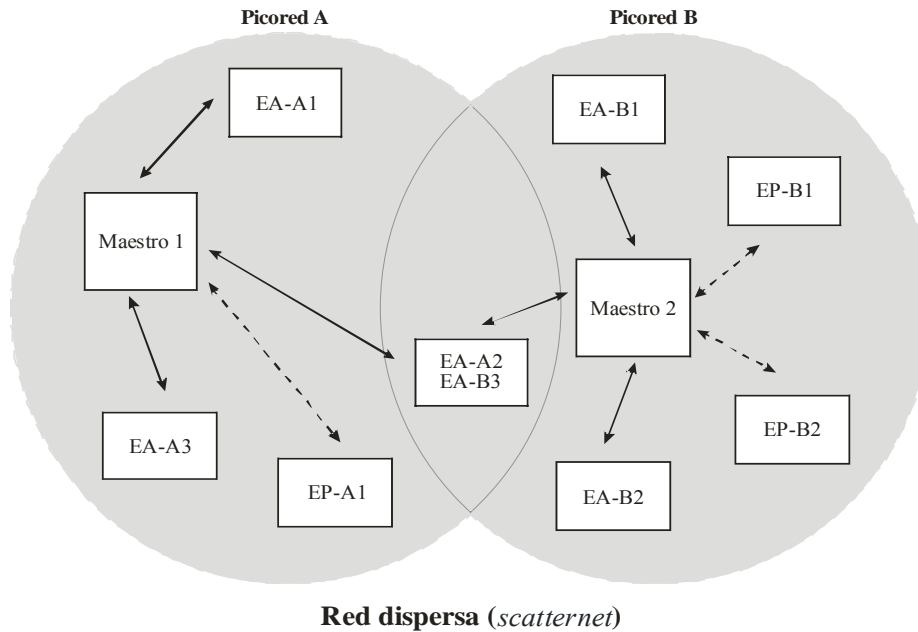
**Figura 3.3.** Estados de una picored Bluetooth.

Es posible que un dispositivo forme parte de más de una picored, y cuando dos o más picoreds están parcialmente sobrelapadas se forma una red dispersa (*scatternet*). Todos los principios de una picored se aplican a una red dispersa (Figura 3.4). Un dispositivo puede actuar como un Esclavo en una picored y puede tomar el rol de Maestro en otra picored.

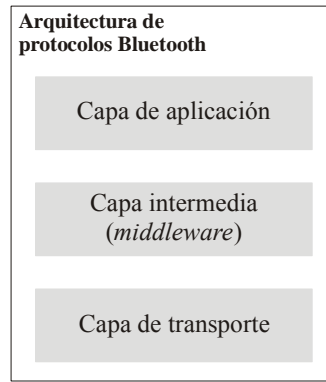
La topología de red dispersa proporciona un método flexible para que los dispositivos puedan mantener múltiples conexiones.

### 3.7. Arquitectura de protocolos Bluetooth

La parte importante de la especificación de Bluetooth describe la arquitectura de protocolos, la cual permite a los dispositivos localizar, conectar e intercambiar datos. La Figura 3.5 muestra los componentes que conforman la arquitectura de protocolos Bluetooth, la cual se divide en tres grupos [11]:



**Figura 3.4.** Ejemplo de una red dispersa.



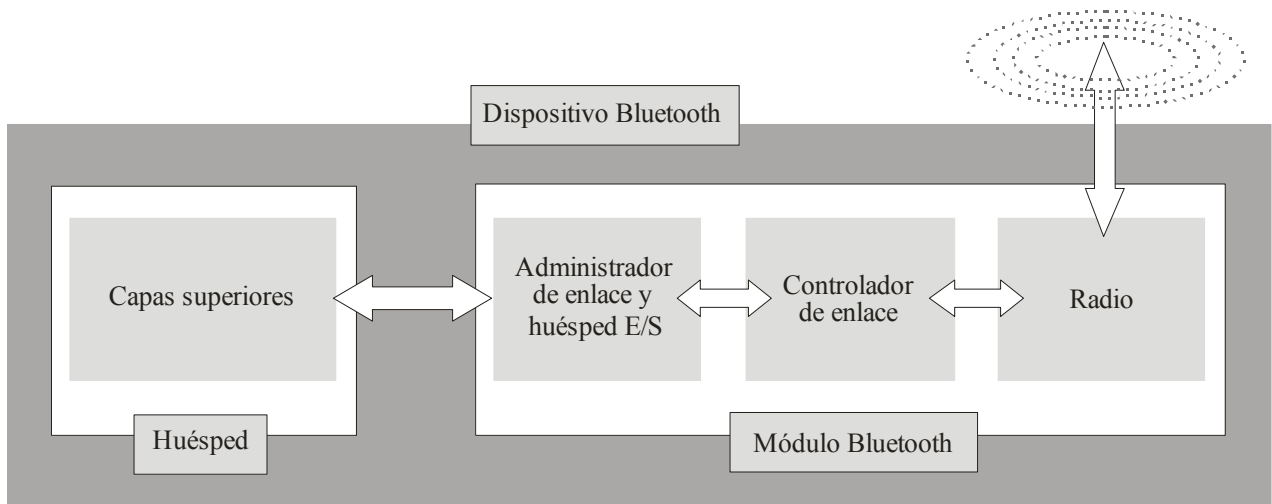
**Figura 3.5.** Arquitectura del protocolos Bluetooth.

- *Capa de transporte (Transport Layer)*: está formada por los protocolos diseñados para permitir a los dispositivos Bluetooth localizar a otros y crear, configurar y gestionar los enlaces físico y lógico. Los protocolos que conforman este grupo son: Radio (*Radio Protocol*), Banda base (*Baseband Layer*), Administrador de enlace (LMP, *Link Manager Protocol*), Control y adaptación de enlace lógico (L2CAP, *Logical Link Control and Adaptation Protocol*) e Interfaz del controlador del huésped (HCI, *Host Controller Interface*).
- *Capa intermedia (Middleware Layer)*: incluye protocolos estandarizados y protocolos desarrollados por el SIG para la comunicación inalámbrica, entre ellos se encuentran los relacionados con Internet como son: PPP (*Point-to-Point Protocol*), TCP/IP (*Transmission Control Protocol/Internet Protocol*), etc., protocolos de aplicación inalámbrica, protocolos adoptados de IrDA. También incluye al protocolo emulador de puerto serie llamado RFCOMM (*Radio Frequency Communication*), el cual es un protocolo de control de señales telefónicas y de localización de dispositivos.
- *Capa de aplicación (Application Layer)*: consiste de las aplicaciones que hacen uso de los enlaces Bluetooth.

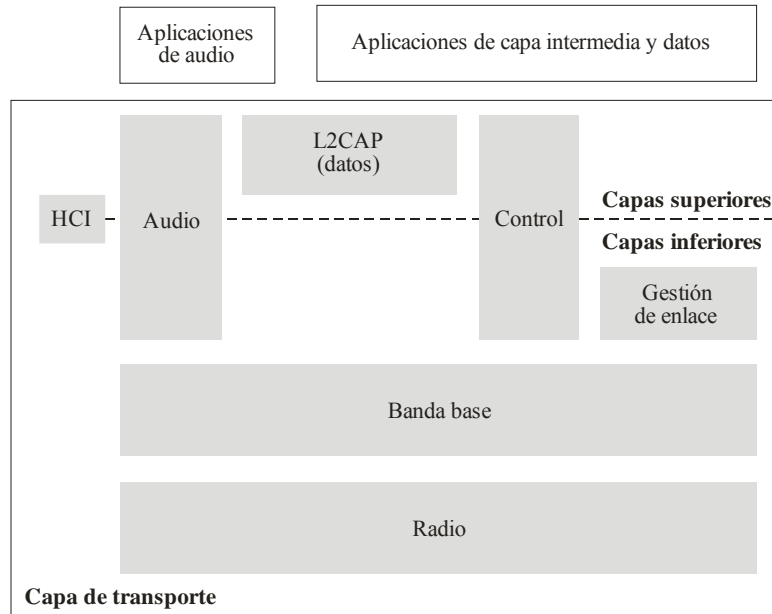
A continuación se detallan las capas y sus correspondientes protocolos.

### 3.7.1. Capa de transporte

Se presentan los protocolos y se describen las funciones de bajo nivel de un sistema Bluetooth, las cuales generalmente se realizan dentro de un módulo firmware Bluetooth como muestra la Figura 3.6.



**Figura 3.6.** Estructura general de un módulo Bluetooth.



**Figura 3.7.** Capa de transporte.

La Figura 3.7 muestra la organización de los protocolos de la capa de transporte que define la especificación, los cuales han sido desarrollados por el SIG para transportar datos y audio entre dispositivos Bluetooth.

Los protocolos de transporte soportan transmisión asíncrona de datos y transmisión síncrona de voz. Se brinda mayor prioridad a las aplicaciones de audio con la finalidad de mantener la calidad de servicio (QoS, *Quality of Service*), por ello el tráfico de audio evita todos los protocolos intermedios y es procesado directamente por la capa de banda base, la cual transmite el audio en paquetes pequeños mediante la interfaz aérea de Bluetooth. Todos los protocolos de este grupo son necesarios para soportar la comunicación entre dispositivos Bluetooth.

La capa de transporte divide los protocolos en inferiores y superiores, los cuales se describen a continuación.

### 3.7.2. Protocolos inferiores de la capa de transporte

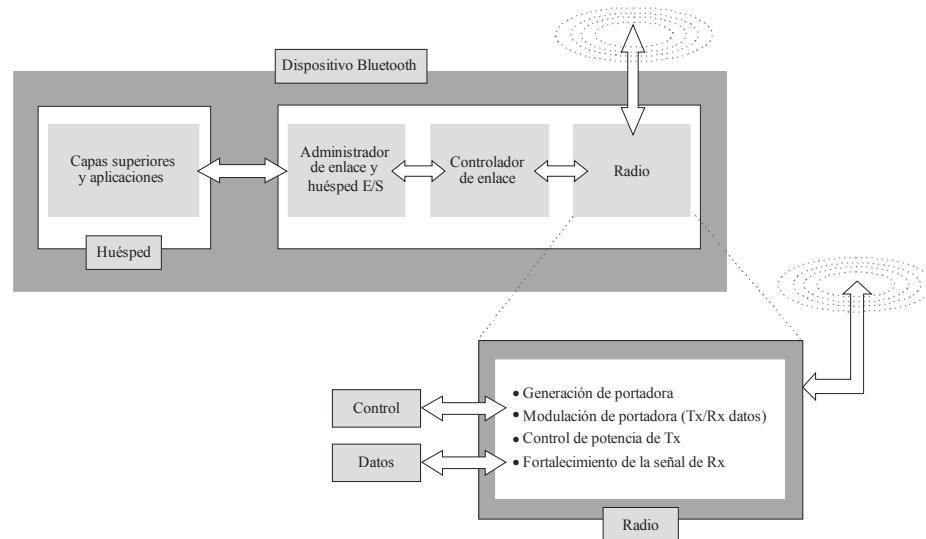
Los protocolos inferiores de la capa de transporte incluyen a las subcapas de radio, banda base y administrador de enlace.

#### 3.7.2.1. Subcapa de radio

La subcapa de radio consiste principalmente en una serie de especificaciones para el diseño de transceptores (*transceivers*) Bluetooth, incluyendo aspectos como emisiones espúreas dentro y fuera de la banda de operación, frecuencia, interferencia de canales adyacentes, características de intermodulación, etc.

El transceptor es un sistema de radio de FHSS operando sobre un número  $m$  de canales con 1 MHz de ancho de banda cada uno. En la mayoría de países  $m = 79$  y en algunos otros existen restricciones para la banda ISM de 2.4 GHz. De esta manera, Bluetooth puede operar con 79 o 23 canales con un ancho de banda de 1 MHz. Para el sistema de radio de FHSS operando en la banda ISM de 2.4 GHz, se restringe la potencia de transmisión a 1 W (30 dBm).

La Figura 3.8 ilustra las principales operaciones que realiza la subcapa de radio Bluetooth, también muestra un par de interfaces lógicas para transportar datos e información de control entre la subcapa de radio y el resto del sistema.



**Figura 3.8.** Operaciones de la subcapa de radio.

En el lado transmisor se incluye la frecuencia portadora, éste debe sintonizarla antes de transmitir un bit de información y también debe considerar el nivel de potencia para la transmisión. Respecto al lado receptor, éste debe conocer la frecuencia portadora y sintonizarla para recibir el bit de información. La Tabla 3.7 muestra los principales parámetros de operación de la subcapa de radio Bluetooth.

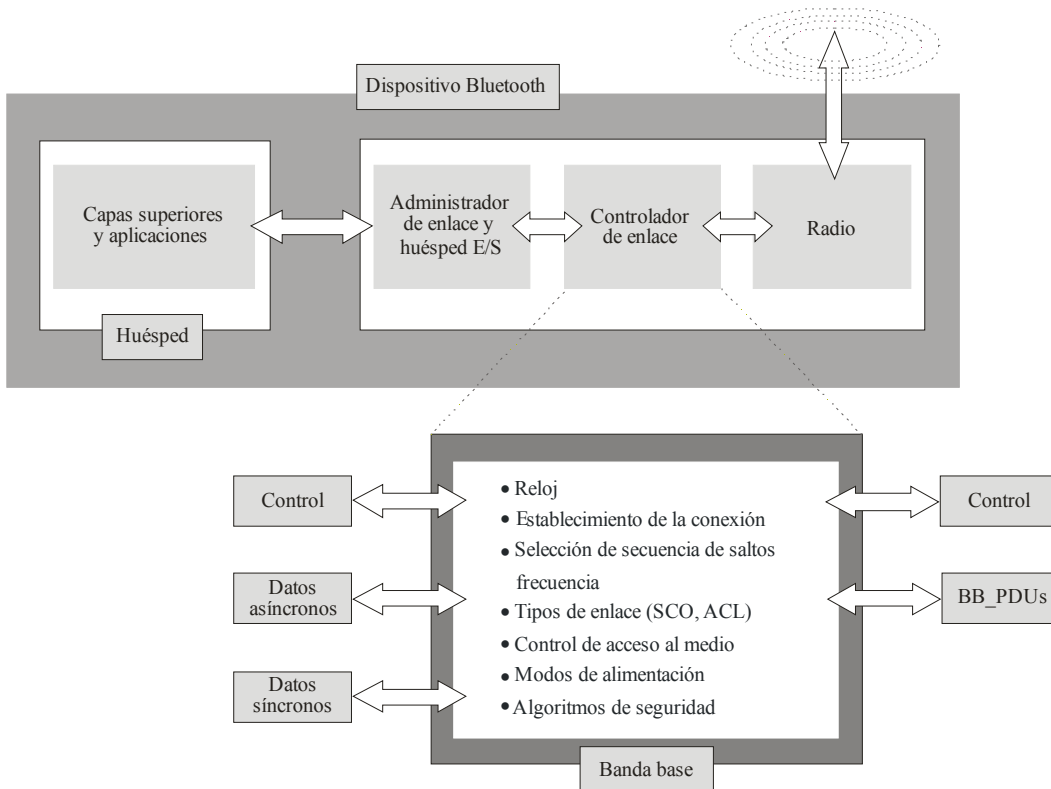
**Tabla 3.7.** Parámetros de operación para la subcapa de radio.

<b>Modulación</b>	GFSK ( <i>Gaussian Frequency-Shift Keying</i> )	Producto BT: 0.5 Índice de modulación: 0.28 - 0.35
<b>Velocidad de símbolo</b>	1 Msps (Msímbolo por segundo)	Usando GFSK binario, maneja una velocidad de enlace de 1 Mbps y un tiempo de transmisión de bit de 1 $\mu$ s
<b>Velocidad de saltos de frecuencia</b>	Generalmente 1,600 saltos por segundo 3,200 saltos por segundo para búsqueda e invitación	Tiempo de residencia: 625 $\mu$ s por salto Tiempo de residencia: 312.5 $\mu$ s por salto
<b>Potencia de transmisión</b>	Clase 3: 0 dBm (1 mW) Clase 2: 4 dBm (2.5 mW) Clase 1: 20 dBm (100 mW)	Un radio típico Bluetooth; control de potencia opcional menor a -30 dBm Control de potencia opcional por arriba de -30 dBm Control de potencia requerido menor de 4 dBm; control de potencia opcional por arriba de 4 dBm
<b>Sensibilidad del receptor</b>	Un receptor Bluetooth debe lograr un tasa de error de bit (BER, <i>Bit Error Rate</i> ) de 0.1 % con un nivel de señal de entrada de hasta -70 dBm	

### 3.7.2.2. Subcapa de banda base

La subcapa de banda base determina y configura la interfaz aérea Bluetooth; define el proceso mediante el cual los dispositivos buscan y se enlazan, los roles del Maestro y del Esclavo, y las secuencias de saltos de frecuencia utilizadas para la comunicación de los dispositivos. Lo anterior establece las reglas para compartir la interfaz aérea por diferentes dispositivos Bluetooth y están basadas en la técnica TDD (*Time Division Duplex*).

La Figura 3.9 muestra las principales funciones que realiza la subcapa de banda base, entre las que se incluye el control de las funciones de la piconet y de los propios dispositivos. Las funciones son: establecimiento de la conexión, selección de la secuencia de saltos de frecuencia y tiempos; modos de operación (control de potencia y operación de seguridad); y de acceso al medio, como sondeo (*polling*), tipo de paquetes, procesamiento de paquetes y tipos de enlaces. Asimismo define un conjunto de interfaces lógicas dedicadas a transportar datos e información de control entre la subcapa de banda base y el resto del sistema Bluetooth.



**Figura 3.9.** Principales funciones de la subcapa de banda base.

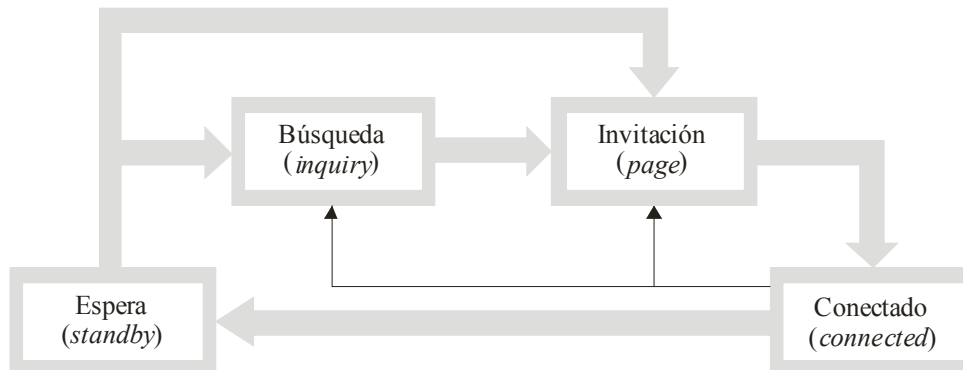
### 3.7.2.2.1. Estados de un dispositivo Bluetooth

Todos los dispositivos que utilizan tecnología inalámbrica Bluetooth necesitan formar parte de una picored para comunicarse con otros dispositivos.

Dentro de una picored, el rol principal de un Maestro es definir:

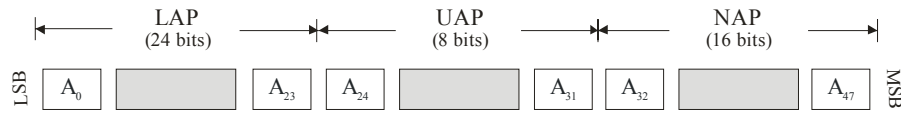
- La secuencia de salto de frecuencia que deben utilizar los miembros activos.
- La frecuencia de espera.
- La frecuencia de operación.
- Qué Esclavo debe transmitir y cuál será el siguiente en hacerlo.

La Figura 3.10 muestra la máquina de estados de los dispositivos Bluetooth. En estado conectado (*connected*), el dispositivo es miembro de una picored, mientras que cuando no está asociado con alguna picored se dice que se encuentra en espera (*standby*), siendo este último el estado por defecto (*default*) para todo dispositivo Bluetooth.



**Figura 3.10.** Máquina de estados de un dispositivo Bluetooth.





**Figura 3.11.** Dirección de los dispositivos Bluetooth.

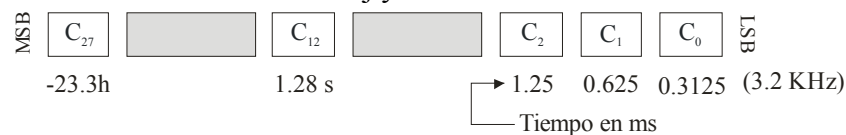
Para que un dispositivo se encuentre en estado conectado (*connected*) debe pasar del estado de búsqueda (*inquiry*) al estado de invitación (*page*). En estado de búsqueda, un dispositivo adquiere información referente a la identidad de otros dispositivos dentro de su vecindad, y éstos deben estar en un subestado de sondeo de búsqueda (*inquiry scan*) para escuchar y responder a las solicitudes del primero (*inquiries*). En estado de invitación, un dispositivo invita explícitamente<sup>7</sup> a otro dispositivo a formar una picored, el dispositivo deberá estar en subestado de sondeo de invitación (*page scan*) para escuchar y responder a las solicitudes (*pages*).

Para formar parte de una picored e intercambiar información dentro de la picored, el dispositivo necesita conocer la forma de crear, leer y escribir paquetes de información en la picored. Estas operaciones se asocian a los siguientes elementos:

- *La dirección del dispositivo Bluetooth:* la dirección Bluetooth (BD\_ADDR, *Bluetooth Device Address*) es una entidad estática y es única para cada dispositivo. Está compuesta por 48 bits, la cual se asigna por una autoridad regulatoria. La Figura 3.11 muestra los campos de dirección, la cual se divide en tres partes: dirección baja (LAP, *Lower Address Part*), dirección alta (UAP, *Upper Address Part*), y dirección no significativa (NAP, *Non-Significat Address Part*). Los 8 bits de la UAP y los 16 bits de la NAP constituyen el identificador único de la organización (OUI, *Organization Unique Identifier*), los cuales son asignados por la autoridad regulatoria y la LAP es asignada por el SIG.
- *El reloj del dispositivo Bluetooth:* cada dispositivo Bluetooth cuenta con un reloj de 28 bits, el cual se incrementa 3,200 veces por segundo o una vez cada 312.5  $\mu$ s., representado una frecuencia de operación de 3.2 KHz (Figura 3.12). El reloj Bluetooth juega un papel fundamental en decidir cuándo un dispositivo puede transmitir o escuchar una transmisión, en qué frecuencia de operación y qué tipo de paquetes de información transmite o escucha.
- *Secuencia de salto de frecuencia:* para que los dispositivos puedan comunicarse, deben transmitir y recibir en la misma frecuencia. El módulo de selección de frecuencia (FSM, *Frequency Selection Module*) contiene el proceso para seleccionar la próxima frecuencia a ser utilizada bajo ciertas condiciones de operación. En la Figura 3.13, la notación (x|y) significa que existen dos opciones permitidas, x o y, y solamente una puede entrar al módulo. La notación  $V[m:n]$ ,  $m \leq n$  denota la utilización de m bits a través de n bits del campo V; m es el bit menos significativo de los dos.

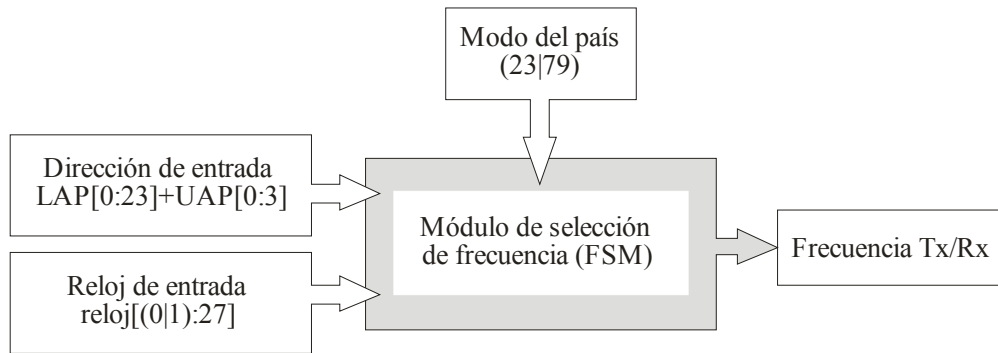
Dependiendo del país donde se utilice el dispositivo Bluetooth, los fabricantes configuran el FSM para operar en 23 o 79 canales de salto de frecuencia. Una vez obtenido el modo de salto de frecuencia, el reloj de entrada determina la frecuencia de secuencia de salto de frecuencia a utilizar.

En cada uno de los estados operacionales de la Figura 3.10, se aplica al FSM una combinación diferente de entradas de reloj y de direcciones.



**Figura 3.12.** Reloj Bluetooth.

<sup>7</sup> El dispositivo que realiza la invitación funge como Maestro.



**Figura 3.13.** Módulo de selección de frecuencia.

Durante la operación normal de la picored, se utiliza la secuencia de salto de canal y se selecciona una nueva frecuencia cada  $625 \mu\text{s}$ . Este intervalo es el periodo para el  $C_1$  del reloj Bluetooth (Figura 3.12) y el tiempo entre dos bits  $C_1$  sucesivos se conoce como ranura (*slot*).

#### 3.7.2.2.2. Operación de invitación

Un dispositivo invita (*paging*) a otro dispositivo (*paged*) a unirse en una picored mediante una invitación (*page*). El dispositivo que invita, selecciona una nueva frecuencia en la cual transmitirá la invitación cada  $312.5 \mu\text{s}$ . Durante los sondeos, cuando un dispositivo escucha para transmitir una invitación, selecciona una nueva frecuencia de sondeo cada  $1.28 \text{ s}$ ., el cual es el periodo del bit  $C_{12}$  (Figura 3.12).

Durante la operación de invitación, se utiliza una secuencia de salto de invitación (*page*), ésta es una secuencia periódica compuesta de 32 frecuencias uniformadamente distribuidas sobre los 79 canales de frecuencia permitidos en varios países dentro de la banda de  $2.4 \text{ GHz}$ .

El dispositivo que invita, utiliza la dirección Bluetooth y el reloj del dispositivo a comunicar para enviarle la invitación. El estado de invitación está compuesto de varios subestados, los cuales son ejecutados por posibles Maestros y Esclavos, dentro de dichos subestados el Maestro ejecuta la invitación (*page*) y la respuesta del Maestro (*master response*), mientras que el Esclavo ejecuta el sondeo de invitación (*page scan*) y la respuesta del Esclavo (*slave response*).

En el estado de invitación, un Maestro transmite unidades de datos del protocolo de banda base (*BB\_PDU*, *Baseband Protocol Data Unit*) de invitación, los cuales son recibidos por el Esclavo cuando éste se encuentra en el subestado de sondeo de invitación. La transmisión contiene únicamente el código de acceso al dispositivo (*DAC*, *Device Access Code*) del Esclavo, esta *BB\_PDU* se conoce como paquete ID del Esclavo (*Slave ID Packet*). El Esclavo también envía el paquete ID del Esclavo para notificar al Maestro que ha recibido la invitación. Finalmente, el Maestro entra en el subestado de respuesta del Maestro, durante el cual transmite sus elementos fundamentales y la dirección de miembro activo *AM\_ADDR* (*Active Member Address*) del Esclavo dentro de un paquete de sincronización de salto de frecuencia (*FHS*, *Frequency Hopping Synchronization*), el cual permite al Esclavo unirse y participar en la picored del Maestro. El Esclavo responde con un paquete ID, entra en el estado conectado y está listo para iniciar la comunicación en la picored.

#### 3.7.2.2.3. Operación de búsqueda

El propósito de los dispositivos en búsqueda es recolectar información sobre otros dispositivos Bluetooth en su proximidad.

Al igual que la operación de invitación, un dispositivo en búsqueda (*inquiry*) selecciona una nueva frecuencia en la cual realizará una búsqueda cada  $312 \mu\text{s}$ . Los dispositivos en búsqueda, ejecutan sondeos (*inquiry scan*), seleccionando una nueva frecuencia cada  $1.28 \text{ s}$ .

Durante la operación de búsqueda, se utiliza la secuencia de salto de búsqueda. Para generar dicha secuencia, el dispositivo que busca y el dispositivo en búsqueda utilizan los 28 bits menos significativos de la dirección de búsqueda (*inquiry address*), conocida como el código de acceso general de búsqueda (GIAC, *General Inquiry Access Code*) de la LAP. La secuencia de salto de búsqueda es una secuencia periódica compuesta de 32 frecuencias distribuidas sobre los 79 canales de frecuencia permitidos en la banda de 2.4 GHz.

El estado de búsqueda está compuesto de varios subestados por posibles Maestros y Esclavos, de éstos el Maestro ejecuta el subestado de búsqueda (*inquiry*) y los Esclavos ejecutan los subestados de sondeo de búsqueda (*inquiry scan*) y respuesta de búsqueda (*inquiry response*). En el subestado de búsqueda, un Maestro transmite paquetes de búsqueda, los cuales son recibidos por los Esclavos en el subestado de sondeo de búsqueda; una vez realizado lo anterior, los Esclavos entran en el subestado de respuesta de búsqueda y transmiten sus elementos fundamentales.

En respuesta a una búsqueda, un Esclavo transmite un paquete FHS conteniendo, junto con otra información, BD\_ADDR y el valor de su reloj. La Tabla 3.8 muestra el contenido del paquete FHS, donde los campos y los bits se transmiten en el orden del LSB al MSB.

**Tabla 3.8.** Paquete FHS.

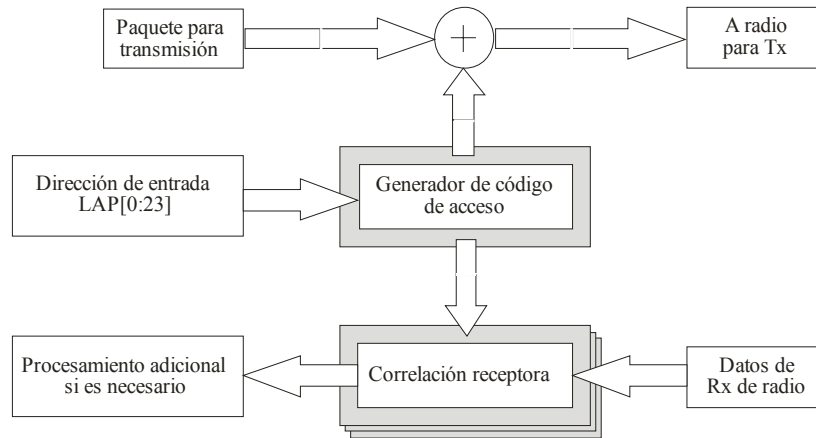
<b>Campo</b>	<b>Tamaño</b>	<b>Función</b>
Bits de paridad	34 bits	Utilizados para construir el DAC de invitación.
LAP	2 bits	LAP de la BD_ADDR.
Reservado	2 bits	Puesto a "00".
Parámetros de intervalos de invitación	4 bits	Dos subcampos de 2 bits definen el periodo y la duración de los sondeos de invitación sucesivos.
UAP	8 bits	UAP de la BD_ADDR.
NAP	16 bits	NAP de la BD_ADDR.
CoD	24 bits	Clase del dispositivo (CoD, <i>Class of Device</i> ).
AM_ADDR	3 bits	Para respuestas de búsqueda, este campo es puesto a "000"; para la respuesta de un Maestro es seguido por la repuesta de invitación de un Esclavo, y es puesto a la dirección AM_ADDR asignado al nuevo Esclavo activo.
CLOCK	26 bits	El tiempo actual del reloj Bluetooth del dispositivo
Modo de sondeo de invitación	3 bits	Es el modo de sondeo de invitación por defecto.

Para evitar colisiones ocasionadas por la respuesta simultánea de múltiples Esclavos a un mismo paquete de búsqueda ID (*inquiry ID*), se utiliza un mecanismo de retroalimentación. En tal mecanismo, el Esclavo entra en el subestado de respuesta de búsqueda; selecciona aleatoriamente un número RN (*Random Number*) menor o igual a 1,023 y suspende sus operaciones de respuesta de búsqueda durante al menos una ranura; el Esclavo puede permanecer en estado de espera, estado conectado o subestado de búsqueda si es necesario. Cuando el Esclavo regresa al estado de respuesta de búsqueda, responde con un paquete FHS correspondiente al primer paquete de búsqueda ID que recibe.

#### 3.7.2.2.4. Código de acceso

El código de acceso es un campo de 68 o 72 bits en cada BB-PDU que se actualiza antes de transmitir sobre el interfaz aérea de Bluetooth para identificar a la picored y sincronizar los bits entrantes. La Figura 3.14 muestra las funciones relacionadas con el código de acceso.

Para recibir, un dispositivo utiliza un módulo de correlación, el cual se puede sintonizar mediante un determinado código de acceso, el cual depende del tipo de operación:



**Figura 3.14.** Funciones del código de acceso.

- *En operación conectado:* durante la operación normal de la picored, antes de realizar la transmisión sobre una frecuencia dada de la secuencia de salto de canal (*channel-hopping*) se envía el código de acceso del canal generado (CAC, *Channel Access Code*) mediante la LAP de la dirección del Maestro de la picored.
- *En operación de invitación:* cada transmisión en una frecuencia dada por la secuencia de salto de invitación y cada respuesta a ésta son precedidas por el código de acceso del dispositivo generado (DAC, *Device Access Code*) mediante la LAP de la dirección del dispositivo invitado.
- *En operación de búsqueda:* cada frecuencia de secuencia de búsqueda de salto y su correspondiente respuesta son procesadas por un código de acceso de búsqueda (IAC, *Inquiry Access Code*). Existen 64 IACs: el GIAC, asociado a los 64 bits reservados de la LAP; y los restantes 63 IACs se conocen como IACs dedicados (DIACs, *Dedicated IACs*). La especificación de Bluetooth no define la utilización de los IACs, sólo recomienda que deben funcionar como un mecanismo de filtrado para identificar subconjuntos predefinidos de dispositivos que pueden realizar búsquedas.

La Tabla 3.9 resume los parámetros relacionados al proceso que debe realizar un dispositivo Bluetooth en cada uno de los estados operacionales.

**Tabla 3.9.** Entradas FSM y códigos de acceso utilizados durante los estados activos de un dispositivo.

Estado del dispositivo	Módulo de frecuencia de espera	Código de acceso
Conectado	Reloj: Maestro Dirección: Maestro	CAC, coinciden con el código de acceso del dispositivo Maestro.
Búsqueda	Reloj: propio Dirección: GIAC	GIAC o DIAC.
Invitación	Reloj: dispositivo invitado Dirección: dispositivo invitado	DAC del dispositivo invitado.

### 3.7.2.2.5. Estado conectado

Para mantener la sincronización de la picored, cada Esclavo añade la diferencia de dicho reloj respecto al del Maestro (*offset*) a su reloj nativo (*native clock*). De esta manera, el reloj del Maestro es el que regula el tiempo en la picored.

Con un reloj de referencia a todos los dispositivos en una picored, el tiempo de transmisión en una picored se divide en ranuras de transmisión de Maestro y Esclavo. Un Maestro inicia su transmisión en una ranura par, mientras que un Esclavo inicia su transmisión en una ranura impar.



**Figura 3.15.** Formato del BB\_PDU.

### 3.7.2.2.6. Tipos de BB\_PDU de banda base

La Figura 3.15 muestra el formato general de una BB\_PDU. La transmisión inicia con el LSB, cada BB\_PDU inicia con el código de acceso que identifica a la picored. El código de acceso tiene generalmente una longitud de 72 bits, el cual incluye un campo de rastreo (*trailer*) de 4 bits.

Si la BB\_PDU no tiene cabecera, y por lo tanto datos, no se realiza el rastreo y la BB\_PDU consiste solamente del código de acceso. Las BB\_PDUs de 68 bits son utilizados exclusivamente para transmitir búsquedas o invitaciones.

La cabecera de la BB\_PDU contiene datos del control de enlace, además de menos de 18 bits de información, los cuales son codificados con un código de corrección de errores (FEC, *Forward Error-Correcting*) a una razón de 1/3 de la velocidad más alta. Cada bit de la cabecera se transmite tres veces (uno de 54 bits). La Tabla 3.10 muestra los campos de la cabecera BB\_PDU.

**Tabla 3.10.** Campos de la cabecera BB\_PDU.

Campo	Tamaño	Función
<i>AM_ADDR</i>	3 bits	Activa la dirección del miembro asignada a un Esclavo activo cuando los dispositivos intercambian información de invitación.
<i>TYPE</i>	4 bits	Define 16 tipos de BB_PDU/datos.
<i>FLOW</i>	1 bit	Continúa y detiene el control de flujo de un dispositivo receptor en respuesta al transmisor.
<i>ARQN</i>	1 bit	Usado para el reconocimiento de la transmisión satisfactoria de los BB_PDUs; Si no se reciben reconocimientos, se retransmite la BB_PDU.
<i>SEQN</i>	1 bit	Secuencia simple (par/impar) para filtrar transmisiones duplicadas.
<i>HEC</i>	8 bits	La cabecera de comprobación de error (HEC, <i>Header Error Check</i> ) es generado por el polinomio $G_{HEC}(x)=x^8+x^7+x^5+x^2+x+1$ .

Durante el proceso de invitación, el Maestro asigna una *AM\_ADDR* de 3 bits al nuevo Esclavo. La *AM\_ADDR* toma el valor de 1 a 7 y es única para cada Esclavo activo en una picored. El valor 0 es reservado para transmisiones de difusión del Maestro a todos los Esclavos en la picored.

Los diferentes tipos de BB\_PDU se distinguen por:

- *Rol*: señalización o paquetes de datos.
- *Tipo de enlace*: datos asíncronos no orientados a conexión (ACL) o datos síncronos orientados a conexión (SCO).
- *Tamaño*: 1, 3 o 5 ranuras.
- *Codificación FEC*: sin FEC, codificación FEC a una razón de 2/3, codificación FEC a una razón de 1/3.

Los paquetes del control de enlace son:

- *ID*: utilizado en los procesos de búsqueda e invitación, y consiste solamente del código de acceso.
- *NULL*: utilizado principalmente para el reconocimiento de Esclavos y el control de flujo para la transmisión de información cuando el Esclavo no tiene nada que transmitir.
- *POLL*: enviado por un Maestro a un Esclavo para sondear alguna transmisión.
- *FHS*: utilizado en los procesos de búsqueda e invitación.

### 3.7.2.2.7. Tipos de enlaces de banda base

La especificación de Bluetooth define dos tipos de enlaces para soportar aplicaciones de voz y datos:

- *Enlace asíncrono no orientado a conexión (ACL, Asynchronous Connectionless)*: soporta transferencia de datos de alta prioridad, los cuales pueden ser datos de usuario o datos de control. ACL soporta enlaces simétricos o asimétricos, conmutación de paquetes, conexión punto a multipunto (*point-to-multipoint*), para transferir datos. Para conexiones simétricas, la máxima velocidad de transferencia es de 433.9 Kbps en ambas direcciones; y para conexiones asimétricas, la máxima velocidad de transferencia es de 723.2 Kbps en una dirección y 57.6 Kbps en la dirección contraria.
- *Enlace síncrono orientado a conexión (SCO, Synchronous Connection Oriented)*: soporta voz en tiempo real y tráfico multimedia mediante ancho de banda reservado. SCO proporciona enlaces simétricos, conmutación de circuito, conexión punto a punto (*point-to-point*), para transmisión de voz mediante tres canales síncronos de 64 Kbps cada uno.

Bluetooth puede soportar enlaces ACL y SCO en forma simultánea.

### 3.7.2.3. Subcapa de administrador de enlace

Las responsabilidades de la subcapa de control de enlace son definir las comunicaciones de banda base, conocer qué y cuándo debe esperar datos, la frecuencia portadora y la potencia de transmisión, parámetros que ejecutará la subcapa de banda base.

En cada dispositivo Bluetooth, la subcapa de control de enlace gestiona las propiedades de la interfaz aérea con otros dispositivos mediante el protocolo de administración del enlace (LMP, *Link Manager Protocol*). Estas propiedades incluyen la asignación del ancho de banda, para soportar un grado del servicio deseado para tráfico de datos (L2CAP); y reservación periódica del ancho de banda, para soportar tráfico de audio.

Esta subcapa utiliza un intercambio de información para autenticar los dispositivos y también se encarga de la encriptación de los datos transmitidos sobre la interfaz aérea. El LMP realiza el control respecto a la potencia necesaria para llevar a cabo la gestión de los modos de operación de banda base (Inciso 3.7.2.3.4).

Las subcapas de control y gestión de enlace, ambas conocidas como administrador de enlace, se encargan de que los dispositivos se comuniquen mediante el intercambio de mensajes Bluetooth. El protocolo de comunicaciones utilizado por estas subcapas se conoce como LMP, el cual define la unidad de datos del protocolo (PDU, *Protocol Data Unit*) conocida como LMP\_PDU. La Figura 3.16 muestra las principales funciones que realiza el administrador de enlace Bluetooth.

El LMP no carga datos de la aplicación, éste se basa en los datos de control que recibe de las capas superiores y del intercambio de LMP\_PDUs con los administradores de enlace de otros dispositivos; a su vez envía señales de control a las subcapas de banda base y de radio de su propio dispositivo. La comunicación de la subcapa de control de enlace no es en tiempo real, puede tomar cerca de 30 segundos en reaccionar a peticiones de otros dispositivos.

La Tabla 3.11 muestra el formato del paquete LMP\_PDU. Cuando la subcapa de control de enlace de un dispositivo inicia una transacción con el administrador de enlace de otro dispositivo, el último administrador de enlace responde con la LMP\_PDU correspondiente a la secuencia de transacción. Alternativamente, el administrador del enlace receptor responde con un LMP\_PDU de aceptación o negación como respuesta a la petición de transacción, si la respuesta es negativa se informa del motivo correspondiente.

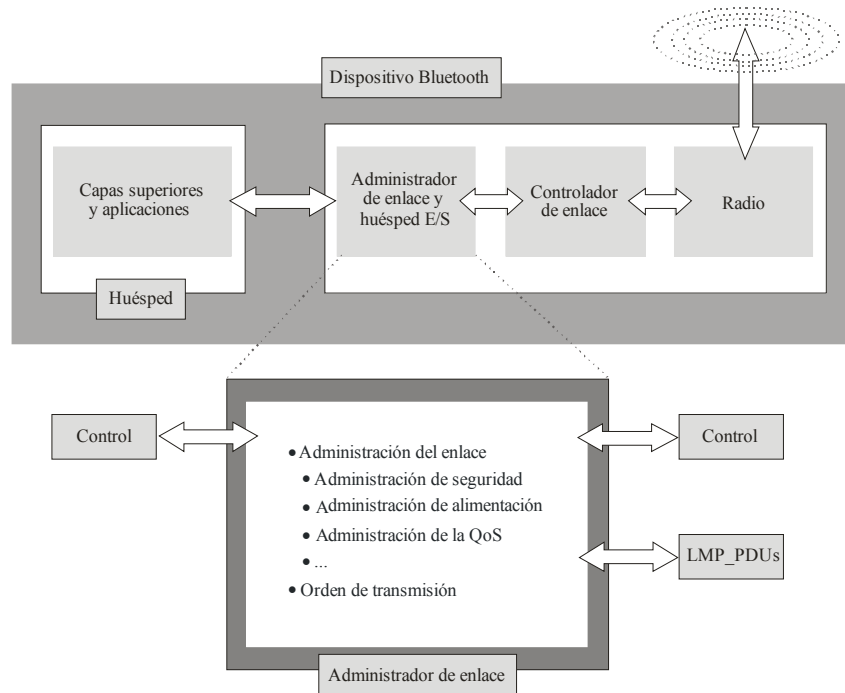


Figura 3.16. Funciones de la subcapa de administrador de enlace.

Tabla 3.11. Formato de LMP\_PDU.

Campo	Tamaño	Función
Transacción ID	1 bit	“0”: la transacción del administrador de enlace es iniciada por el Maestro “1”: la transacción del administrador de enlace es iniciada por un Esclavo
OpCode	7 bits	Identificador del LMP_PDU y el tipo de contenido de su carga
Carga ( <i>payload</i> )	0-17 octetos	El LMP_PDU se coloca dentro del DM1 BB_PDU; si la carga del LMP_PDU es menor que 9 octetos, se pueden utilizar DV BB_PDUs

La Figura 3.17 muestra dos tipos de LMP\_PDUs:

- Cualquiera de los administradores de enlace inician una transacción mediante una petición de transacción, el administrador del enlace receptor decide si acepta o rechaza la petición.
- El Maestro envía una orden para ser ejecutada por el Esclavo sin oportunidad de que el Esclavo rechace la orden o negocie sus parámetros.

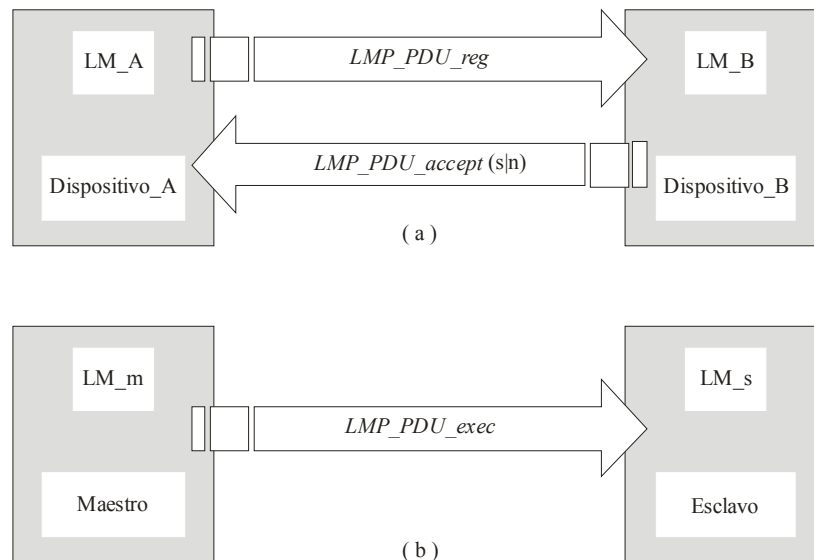


Figura 3.17. (a) Transacción y negociación; (b) El Maestro ordena el ajuste del enlace.

### 3.7.2.3.1. Manejo de seguridad

La especificación de Bluetooth emplea funciones de autenticación y encriptación. La autenticación previene el acceso indeseado a datos y funciones, también protege a la picored de intrusos (*hackers*) que intentan hacerse pasar por usuarios autorizados; la encriptación codifica los datos durante la transmisión para prevenir intercepciones y mantener la privacidad del enlace.

Bluetooth ofrece una arquitectura de seguridad flexible para gestión de acceso a los dispositivos y servicios autorizados.

### 3.7.2.3.2. Autenticación de dispositivos

La autenticación de dispositivos Bluetooth se basa en una transacción conocida como respuesta de descubrimiento (*challenge-response*). En esta transacción, se utiliza un verificador (*verifier*) para descubrir a un pretendiente (*claimant*) mediante el envío de un número aleatorio de 16 octetos en la PDU *LMP\_au\_rand*. El pretendiente opera sobre el número aleatorio y regresa el resultado de la operación al verificador mediante la PDU *LMP\_sres*. Si el resultado es el esperado por el verificador, éste considera al pretendiente como un dispositivo autenticado.

El proceso anterior ocurre cuando ambos dispositivos tienen una clave de enlace común (*link key*), la cual se utiliza para operar sobre un número aleatorio. Un dispositivo puede mantener claves de enlace idénticas o separadas para cada uno de los dispositivos que quiera autenticar. Si una clave de enlace no existe para un dispositivo, se envía una PDU *LMP\_au\_rand* y el pretendiente responde con una PDU *LMP\_not-accepted*, indicando que la clave está perdida y en este caso es necesario que el dispositivo esté emparejado (*paired*). El proceso de emparejamiento genera la clave de inicialización utilizada para autenticar los dispositivos y eventualmente crea una clave de enlace permanente.

### 3.7.2.3.3. Encriptación del enlace

Se encripta un enlace Bluetooth para proteger la privacidad del flujo de los datos. La encriptación está basada en una cifra de 1 bit y el tamaño de la clave de encriptación cambia con cada transmisión de BB\_PDUs.

La clave de encriptación se deriva de la clave de enlace utilizada para autenticar la comunicación de los dispositivos, esto implica que antes de utilizar la encriptación los dos dispositivos deben haber sido autenticados por lo menos una vez. El tamaño máximo para la clave de encriptación es de 128 bits, pero las autoridades regulatorias en varios países pueden limitar el tamaño de dicha clave.

Una petición de encriptación inicia con la PDU *LMP\_encryption\_mode\_req*, la cual incluye un parámetro que distingue la encriptación de un enlace entre dos dispositivos y la encriptación de paquetes de difusión. En el último caso, se crea una clave maestra para realizar la encriptación de múltiples dispositivos en la picored. Si se acepta la petición de encriptación, los dispositivos negocian el tamaño de la clave de encriptación intercambiando PDUs *LMP\_encryption\_key\_size\_req*. Si la negociación es satisfactoria, los dispositivos pueden iniciar la encriptación enviando la PDU *LMP\_start\_encryption*.

### 3.7.2.3.4. Manejo de potencia

Los dispositivos en un estado conectado pueden regular su asociación con la picored en la cual están conectados con la finalidad de ahorrar energía o atender otros procesos tales como la participación en una red dispersa, además de notificar el establecimiento de un enlace entre dispositivos, un dispositivo puede solicitar a otro el ajuste de la potencia de transmisión dependiendo de la calidad del enlace.



Estos modos son aplicados cuando la subcapa de banda base se encuentra en un estado de conexión; cuando no es así, la subcapa de banda base está en un estado de espera. Dentro del estado conectado existen cuatro modos:

- *Activo (active)*: un Esclavo activo debe escuchar todos los paquetes provenientes del Maestro. Este modo proporciona la respuesta más rápida pero también es la que consume más energía.
- *Sondeo (sniff)*: este modo se utiliza para reducir el consumo de energía, y en él, un Esclavo llega a estar periódicamente activo. El dispositivo Maestro se sincroniza durante algunos intervalos de tiempo para transmitir paquetes destinados a un Esclavo, éste escucha los paquetes del Maestro solamente durante el inicio de cada intervalo de tiempo.
- *Reposo (hold)*: en este modo un Esclavo puede dejar de escuchar paquetes del Maestro por un intervalo específico de tiempo (*hold time*), durante el cual no le es necesario escuchar paquetes y puede realizar otras tareas, como por ejemplo establecer enlaces con otros dispositivos.
- *Estacionado (parked)*: un Esclavo mantiene la sincronización con el Maestro pero éste no se encuentra activo. Debido a que la especificación sólo permite siete Esclavos activos en una piconet a la vez, el uso de este modo permite a los Maestros comunicarse con un número mayor de dispositivos.

En adición a los modos de banda base, otra característica de bajo consumo es la potencia de transmisión adaptativa (*adaptive transmission power*), característica que permite a los Esclavos informar al Maestro en caso de que su potencia de transmisión no sea la adecuada, con la finalidad de que el Maestro ajuste su potencia de transmisión. Además se utiliza un indicador de intensidad para la señal recibida (RSSI, *Received Signal Strength Indicator*); cuando el valor de RSSI se encuentra fuera de los límites permitidos, el Esclavo avisa al Maestro para que ajuste su potencia de transmisión, esto es especialmente útil cuando dos dispositivos están muy cerca y no se requiere de mucha potencia de transmisión.

#### 3.7.2.3.5. Manejo de ancho de banda

Los dispositivos Bluetooth tienen varias opciones disponibles para el manejo del ancho de banda asignado; pueden utilizar enlaces SCO, los cuales proporcionan transmisiones periódicas de alta prioridad para comunicaciones de voz con calidad telefónica. Las transmisiones sobre enlaces ACL también pueden emplearse para garantizar el ancho de banda a través de intervalos de sondeos restringidos.

Para incrementar la eficiencia de la transmisión y con ello incrementar el ancho de banda de un enlace ACL, los controladores de enlace pueden negociar el uso de BB\_PDUs de mayor tamaño.

#### 3.7.2.3.6. Calidad de servicio para enlaces ACL

Para controlar el ancho de banda mínimo asignado al tráfico ACL entre dos dispositivos o el máximo retardo de acceso de las BB\_PDUs ACL, se puede ajustar el intervalo de sondeo (*polling interval*) de un Esclavo. Un maestro puede imponer un nuevo intervalo de sondeo mediante una PDU *LMP\_quality\_of\_service*, y el Esclavo no puede rechazar el ajuste del intervalo de sondeo. Por otro lado, un Maestro o un Esclavo pueden solicitar la modificación del intervalo de sondeo mediante una PDU *LMP\_quality\_of\_service\_req*, la cual puede ser aceptada o rechazada por la otra parte.

### 3.7.2.3.7. Manejo del controlador de enlace

En esta sección se describen los parámetros relacionados con el controlador de enlace y los protocolos de banda base, tales como la negociación del esquema de invitación, exactitud de los tiempos, cambio en el rol Maestro-Esclavo.

### 3.7.2.3.8. Esquema de invitación

Cuando dos dispositivos se encuentran en comunicación, pueden reconectarse entre ellos de una manera más rápida debido a que evitan el proceso de búsqueda. Con el esquema de invitación, los dispositivos pueden anunciar o negociar dicho esquema para utilizarlo la próxima vez que requieran del mismo enlace, para ello utilizan las PDUs *LMP\_page\_mode\_req* y *LMP\_page\_scan\_mode\_req*.

### 3.7.2.3.9. Rol del Maestro y del Esclavo

Una picored soporta hasta tres enlaces SCO para comunicaciones de voz con calidad telefónica (64 Kbps). Los paquetes transmitidos sobre el enlace SCO son de prioridad alta y pueden sustituir cualquier otra actividad que el dispositivo pueda estar realizando. Un dispositivo pide el establecimiento de un enlace SCO mediante una PDU *LMP\_SCO\_link\_req*. Esta PDU contiene los parámetros de audio, el tipo de BB\_PDU SCO, y el modo aéreo (*air-mode*), el cual representa el tipo de codificación de voz.

Cuando el Maestro responde positivamente a un Esclavo o envía su propia *LMP\_SCO\_link\_req*, éste suministra el nivel  $D_{SCO}$  que identifica el tiempo de la primera transmisión para el nuevo enlace SCO, y un identificador único de enlace SCO.

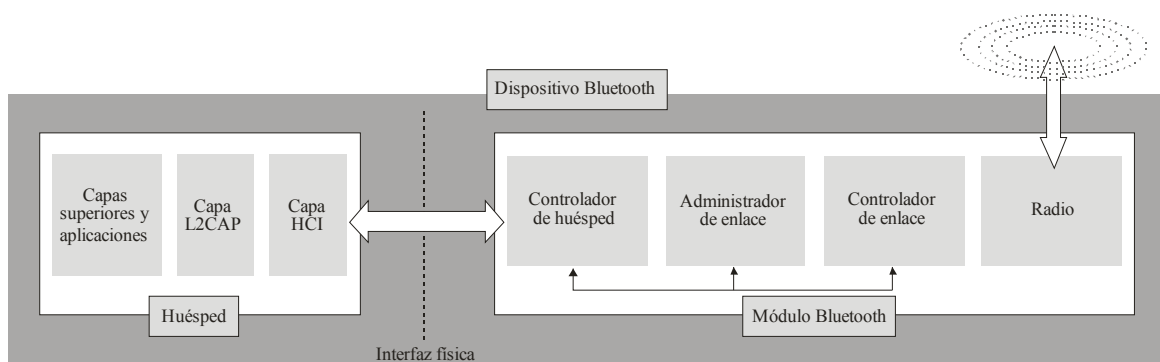
### 3.7.2.3.10. Reloj e información de los tiempos

Un dispositivo puede solicitar información de los tiempos actualizados de otro dispositivo con la finalidad de optimizar varias operaciones del controlador de enlace, lo anterior se realiza mediante los PDUs:

- *LMP\_clock\_offset\_req*: enviado por un Maestro a un Esclavo, el Esclavo regresa el nivel actual entre los relojes nativos del Esclavo y del Maestro.
- *LMP\_slot\_offset*: carga el nivel de ranura, en microsegundos, entre el inicio del tiempo de transmisión de una ranura de transmisión del Maestro y la correspondiente ranura de transmisión del Esclavo.
- *LMP\_timing\_accuracy\_req*: resulta en regresar el tiempo de inestabilidad (*jitter*), en microsegundos, y la desviación (*drift*), en partes por millón (ppm), del reloj del dispositivo receptor.
- *LMP\_supervision\_timeout*: enviado por un Maestro, incluye el valor de la supervisión del enlace usada por un controlador del enlace para detectar la pérdida de un enlace Bluetooth entre el Maestro y un Esclavo.

## 3.7.3. Protocolos superiores del grupo de transporte

Los protocolos inferiores del grupo de transporte ocultan los aspectos del medio de transmisión RF, los requerimientos de usuario de bajo costo y ahorro de energía, seguridad, problemas regulatorios, etc. Éstos se encargan de seleccionar el división de tiempo duplex (TDD, *Time Division Duplex*) del protocolo de acceso al medio basado en el sondeo en la banda base, y realizan las operaciones necesarias para llevar a cabo la comunicación entre sistemas Bluetooth en la banda ISM de 2.4 GHz, sin embargo el tamaño de una BB\_PDU es limitado y no es suficiente para comunicar los paquetes de tamaño grande que maneja Internet y aplicaciones similares.



**Figura 3.18.** El grupo de protocolos de la capa de transporte.

Por lo anterior, es necesaria una subcapa de adaptación para mover PDUs grandes a las capas superiores y regresar PDUs pequeños a las capas inferiores. Esta capa de adaptación fue originalmente llamada control de acceso al medio nivel 2 (MAC-2, *Level 2 Medium Access Control*). En el verano de 1998 se cambió el nombre de MAC-2 a L2CAP (Figura 3.18).

### 3.7.3.1. Subcapa L2CAP

En Bluetooth, el tráfico de datos está dirigido a las capas de control de enlace lógico y de aplicación. La capa L2CAP aísla a los protocolos de capas superiores de los detalles de los protocolos de capas inferiores. L2CAP soporta la multiplexación de protocolos, permitiendo a los protocolos superiores compartir la interfaz aérea; puede ejercer el control de admisión del tráfico entrante y coordinarse con las capas inferiores para mantener el nivel de servicio deseado.

La capa L2CAP trabaja únicamente con transmisiones de información asíncronas (ACL) mediante L2CAP\_PDUs. Las capas de transporte inferiores proporcionan una interfaz a L2CAP sobre la cual envía y recibe datos y mensajes de control.

La capa L2CAP soporta la multiplexación, para facilitar la segmentación y reensamblado de paquetes más grandes; el intercambio de información con QoS, para regular el control de los recursos de transmisión; y un grupo de abstracción a las capas superiores, lo cual permite direccionar a grupos de protocolos de capas superiores dentro de picoredes sin exponer el concepto de una picored a las capas superiores.

#### 3.7.3.1.1. Canales L2CAP

La comunicación entre subcapas L2CAP está basada en enlaces lógicos, llamados canales, a través de los cuales el tráfico L2CAP fluye entre los puntos finales (*endpoints*) dentro de cada dispositivo. Se asigna el punto final de un canal a un único identificador de canal (CID, *Channel Identifier*). Un CID es un identificador de 16 bits administrado localmente.

La Figura 3.19 muestra las subcapas L2CAP de tres dispositivos intercambiando información mediante canales L2CAP. Cada canal termina en un punto final dentro la capa L2CAP y cada punto final L2CAP es asociado únicamente con una entidad receptora de información, a la cual se dirige la información del L2CAP\_PDU para procesamientos adicionales. También identifica los diferentes tipos de canales L2CAP definidos en la especificación:

- *Canales persistentes orientados a conexión (CO, Connection Oriented)*: usados para comunicaciones bidireccionales.
- *Canales efímeros no orientados a conexión (CL, Connectionless)*: son unidireccionales y pueden ser utilizados para transmisiones de difusión a grupos de dispositivos.
- *Canales de señalización*: utilizados para intercambiar información de control, utilizada para establecer y configurar canales CO.

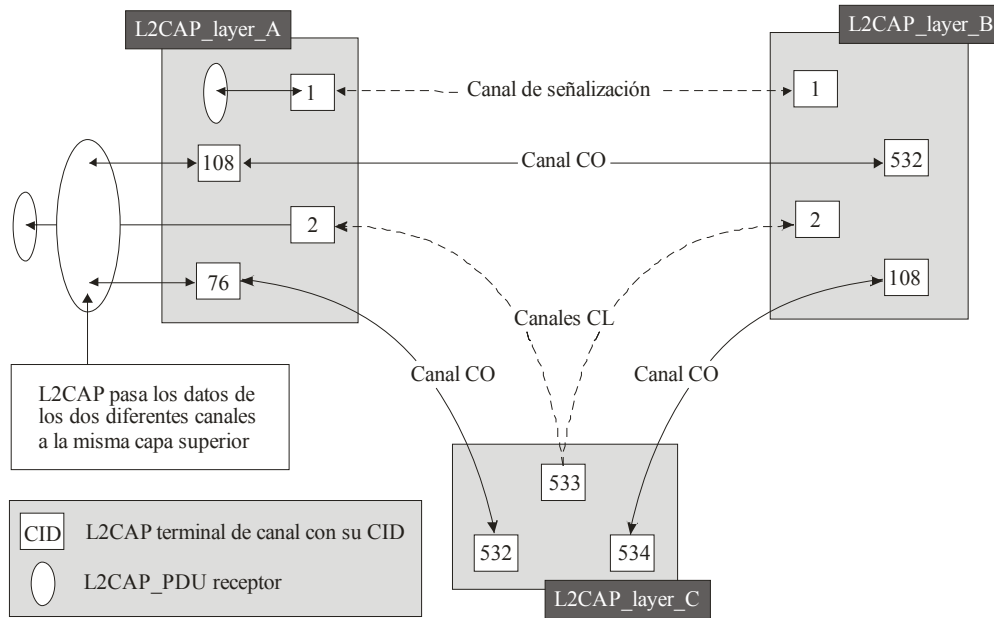


Figura 3.19. Canales L2CAP.

3.7.3.1.2. Tipos de L2CAP\_PDU

Existen dos tipos de L2CAP\_PDU:

- L2CAP\_PDU CL: utiliza canales CL y cuenta con los campos que muestra la Tabla 3.12; el campo PSM (*Protocol and Service Multiplexor*) identifica al receptor de la L2CAP\_PDU CL y el mínimo valor soportado por la unidad de transmisión máxima (MTU, *Maximum Transmission Unit*) de las L2CAP\_PDU CL es de 670 octetos.
- L2CAP\_PDU CO: utiliza canales CO, y cuenta con los campos que muestra la Tabla 3.13. El campo PSM en una L2CAP\_PDU CO es tratado como parte de la carga.

Tabla 3.12. Formato de una L2CAP\_PDU CL.

Campo	Tamaño	Función
<b>L2CAP_PDU_CL_Header</b>		
Longitud	2 octetos	Longitud total en octetos de la L2CAP_PDU excluyendo los campos de longitud y CID.
Destino_CID	2 octetos	Indica el CID del punto final destino del canal L2CAP usado para esta transmisión.
PSM	≥2 octetos	Multiplexor de protocolo y servicio.
<b>L2CAP_PDU_CL_Payload</b>		
Carga ( <i>payload</i> )	(Longitud - PSM) octetos	Carga datos L2CAP_PDU CL; el máximo tamaño posible es 65,535 octetos menos el tamaño de campo PSM, el cual típicamente es de 2 octetos.

Tabla 3.13. El formato de una L2CAP\_PDU CO.

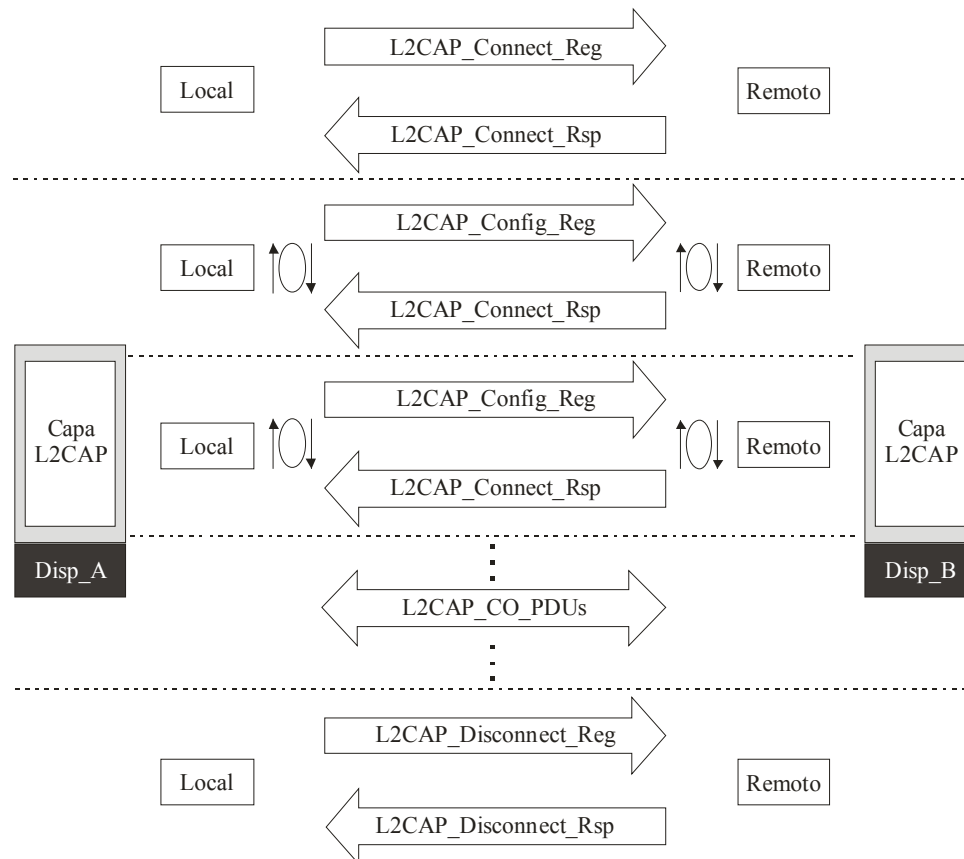
Campo	Tamaño	Función
<b>L2CAP_PDU_CO_Header</b>		
Longitud	2 octetos	Longitud total en octetos de este L2CAP_PDU excluyendo la L2CAP header.
Destino_CID	2 octetos	Indica el CID del punto final destino del canal L2CAP usado para esta transmisión.
<b>L2CAP_PDU_CO_Payload</b>		
Carga ( <i>payload</i> )	Longitud octetos	Carga datos L2CAP_PDU CO; el máximo tamaño posible es 65,535 octetos.

### 3.7.3.1.3. Administración del canal L2CAP

Los canales OC L2CAP utilizan una técnica de señalización para enlazarse, configurarse y finalizar el enlace. La señalización L2CAP consiste en PDUs de peticiones y repuestas enviadas a través del canal de señalización, con valor reservado CID “0x0001”. Un dispositivo nombrado como local, envía un PDU de petición L2CAP de señalización a un dispositivo remoto, y el dispositivo remoto responde con un PDU de respuesta L2CAP de señalización. Cada transacción es identificada por un identificador de transacción (*transactionID*) que realiza una petición de un dispositivo local con la subsecuente respuesta del dispositivo remoto.

La Figura 3.20 ilustra una secuencia de transacciones de señalización L2CAP entre un dispositivo local y un dispositivo remoto. Dicha secuencia inicia cuando la capa L2CAP del dispositivo local transmite una petición para establecer un canal entre los dispositivos; el dispositivo remoto responde aceptando o rechazando la petición; si se acepta la petición, el dispositivo local inicia la configuración del canal para la comunicación en una dirección; la subcapa L2CAP negocian los parámetros de configuración; el dispositivo remoto responde positivamente o negativamente a los parámetros de configuración del dispositivo local; una respuesta negativa causa una fase de negociación de la configuración entre los dispositivos hasta que lleguen a un acuerdo; una vez realizada la configuración, los dispositivos intercambian los roles de dispositivo local y remoto, y el proceso de configuración continua en la dirección contraria; al terminar la fase de configuración, el canal está listo para recibir y transportar PDUs de capas superiores sobre el nuevo canal. El canal L2CAP termina cuando cualquiera de los dispositivos inicia la fase de terminación.

Cuando un dispositivo local establece un canal L2CAP CO con un dispositivo remoto, envía una orden de señalización *L2CAP\_Connection\_Request* (Código “0x02”) al dispositivo remoto. Los campos de esta orden se muestran en la Tabla 3.14.



**Figura 3.20.** Transacciones de señalización L2CAP.

**Tabla 3.14.** Orden de señalización L2CAP\_Connection\_Request.

Campo	Tamaño	Función
<i>L2CAP_Connection_Request_Command_Payload</i>		
PSM	≥ 2 octetos	Identifica el destino de la carga transmitida al dispositivo remoto.
Source_CID	2 octetos	El CID para el punto final destino del canal L2CAP en el dispositivo local.

El campo PSM es un campo de tamaño variable, con un tamaño mínimo de dos octetos, utilizado en la multiplexación de varios protocolos sobre la subcapa L2CAP. En particular, el dispositivo local utiliza el campo PSM para informar al dispositivo remoto hacia donde dirigir la carga de la L2CAP\_PDU sobre el canal.

El dispositivo remoto regresa una orden *L2CAP\_Connection\_Response* (Código “0x03”) para informar al dispositivo local si acepta o no la petición de conexión. El dispositivo remoto puede regresar una petición de conexión pendiente (*connection pending*) para informar al dispositivo local que la petición de conexión ha sido recibida pero aún no se ha tomado la decisión. Si una conexión es rechazada, el dispositivo remoto proporciona la razón, la cual podría incluir razones de seguridad o valores PSM no soportados.

La configuración de un canal puede realizarse poco después de haberse iniciado la comunicación normal del canal. La Tabla 3.15 muestra los campos claves de la orden *L2CAP\_Configuration\_Request* (Código “0x04”). La duración de las negociaciones de configuración depende de la implementación, siendo el tiempo máximo de 120 segundos.

**Tabla 3.15.** Orden de señalización L2CAP\_Configuration\_Request.

Campo	Tamaño	Función
<i>L2CAP_Configuration_Request_Command_Payload</i>		
Destination_CID	2 octetos	Para el canal a ser configurado, identifica el CID del punto final del canal en el dispositivo que recibe esta orden.
Flags	2 octetos	En la versión 1.0 solamente el LSB es usado para dar a conocer que opciones de configuración adicionales son seguidas en una orden de configuración subsecuente.
Config_Options	Variable	Opciones de configuración para el canal a ser configurado.

#### 3.7.3.1.4. Opciones de configuración

La Tabla 3.16 muestra los parámetros de comunicación que pueden ajustarse mediante el proceso. La interpretación de estos parámetros depende del dispositivo que origina la orden *L2CAP\_Configuration\_Request*.

La opción QoS incluye un parámetro *Service\_Type* que determina si el tráfico será enviado en la dirección especificada por el dispositivo local. Si es afirmativo, el *Service\_Type* determina si el tráfico será tratado por el dispositivo local con el mejor esfuerzo (*best effort*) o garantizado (*guaranteed*), el primer caso es el tipo de servicio por defecto soportado por cualquier capa de implementación L2CAP, mientras que en tráfico garantizado, el dispositivo local proporciona los parámetros QoS asociados con su flujo de tráfico.

**Tabla 3.16.** Opciones de configuración.

Parámetro	Función
<i>MTU_CO</i>	Identifica la carga máxima de la L2CAP_PDU, en octetos, que el dispositivo local puede aceptar sobre este canal; como mínimo $MTU_{CO} = 48$ octetos y por defecto $MTU_{CO} = 672$ octetos.
<i>Flush_Timeout</i>	Identifica el monto del tiempo, en múltiplos de 1.25 ms, durante el cual el administrador del enlace del dispositivo local continuará intentando transmitir los segmentos de banda base de un L2CAP_PDU.
<i>QoS</i>	Identifica la especificación para el flujo del tráfico del dispositivo local sobre este canal.

La señalización comúnmente utilizada son las órdenes de señalización *L2CAP\_Connection\_Request* y *L2CAP\_Configuration\_Request*, junto con sus correspondientes órdenes de respuesta. Una petición y respuesta de terminación de canal utiliza la orden de señalización *L2CAP\_Disconnection\_Request\_Response*.

La señalización L2CAP proporciona dos órdenes para examinar y recolectar información. Las órdenes *L2CAP\_Echo\_Request* y *L2CAP\_Echo\_Response* se utilizan para examinar la conexión de un dispositivo remoto. La orden *echo* contiene un porción de carga opcional, la cual se puede utilizar para pasar información entre dispositivos de forma apropiada, pero no estandarizada.

Las órdenes *L2CAP\_Information\_Request* y *L2CAP\_Information\_Response* se utilizan para solicitar información de la especificación o implementación.

### 3.7.3.2. Subcapa HCI

Las subcapas de radio, banda base y administrador de enlace pueden incorporarse dentro de un módulo Bluetooth, el cual se enlaza a un dispositivo huésped (*host*) para habilitarlo de comunicación inalámbrica Bluetooth. El módulo se une al huésped mediante una interfaz física llamada transporte de huésped (*host transport*), la cual puede ser mediante USB (*Universal Serial Bus*), RS-232 o UART.

La interfaz del controlador de huésped (HCI, *Host Controller Interface*) permite a las capas superiores, incluyendo a las aplicaciones, acceder a las capas de banda base, control de enlace y otros registros hardware.

El controlador de huésped (*Host Controller*) es el responsable de interpretar la información recibida del huésped y dirigirla a los componentes apropiados del módulo, como por ejemplo al controlador del enlace. De igual forma el controlador de huésped reúne datos y estados del firmware del módulo y los transmite dentro del huésped como sea necesario. El módulo Bluetooth contiene: la interfaz de radio, el controlador del enlace, el administrador del enlace y el huésped para ligar el módulo a un huésped.

Para permitir la interoperabilidad de módulos no integrados de diferentes fabricantes, el SIG ha definido: una interfaz estándar, para comunicarse con el controlador de huésped del módulo en una manera que sea independiente de la interfaz física; y un mecanismo de transporte, usado entre el huésped y el controlador de huésped.

El SIG también ha definido un protocolo de comunicación para cargar información entre el huésped y el controlador de huésped, el cual se conoce como HCI. HCI define el conjunto de funciones de un módulo Bluetooth que pueden acceder al huésped y sus aplicaciones; es el guardián de los servicios que este módulo puede proporcionar a sus usuarios.

La Figura 3.21 muestra la arquitectura de protocolos con un HCI. La parte del HCI contiene un conjunto de interfaces a las capas superiores definidas por la especificación mediante distintas HCI\_PDUs.

#### 3.7.3.2.1. Clases de HCI\_PDUs

Existen tres clases de HCI\_PDUs para intercambiar información (Figura 3.22):

- *Órdenes HCI\_PDUs*: envían información de control y de manejo; y son enviadas de la capa HCI hacia el controlador de huésped.
- *Eventos HCI\_PDUs*: envían información de control y de manejo del controlador de huésped a la capa HCI.
- *Datos HCI\_PDUs*: cargan fragmentos de datos L2CAP\_PDUs y SCO. La especificación los divide en dos categorías, uno para datos L2CAP asíncronos y uno para datos síncronos.

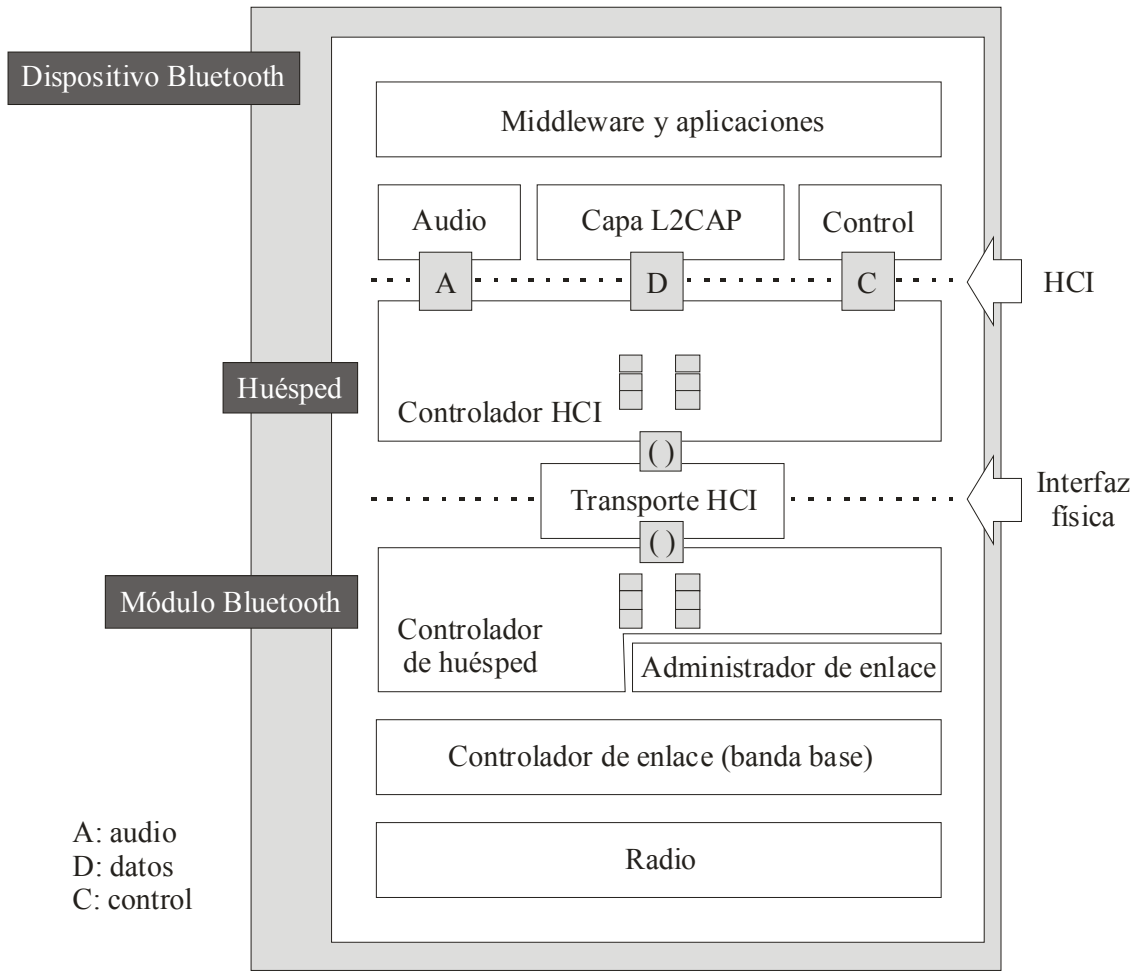


Figura 3.21. Arquitectura de un sistema con interfaz de huésped.

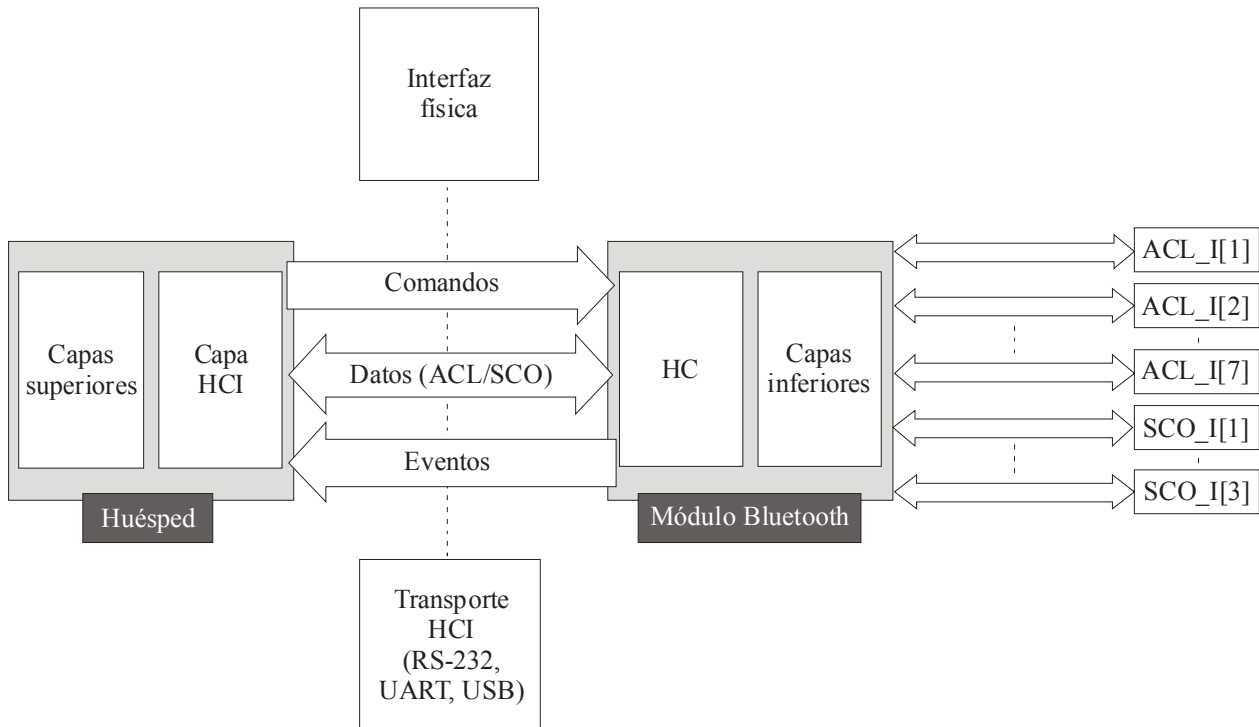


Figura 3.22. Clases de HCI\_PDUs.



La Tabla 3.17 muestra la estructura de una orden HCI\_PDU, la cual incluye el *OpCode* para identificar el tipo de orden.

**Tabla 3.17.** Orden HCI\_PDU.

Campo	Tamaño	Función
<b>HCI_Command_Header</b>		
<i>OpCode</i>	2 octetos	El subcampo de grupo OpCode (OGP, <i>OpCode Group Subfield</i> ) consta de 6 bits para identificar el grupo al que pertenece el <i>OpCode</i> : “111110” identifica un OGF reservado para la examinación del logotipo Bluetooth; “111111” identifica un OGF reservado para órdenes utilizadas durante la fabricación del módulo; Otros valores indican otros grupos tales como control de enlace, políticas de enlace y otras.
<i>Payload_Length</i>	1 octeto	El subcampo de la orden OpCode (OCF, <i>OpCode command subfield</i> ) consta de 10 bits para identificar una orden HCI específica con el particular OGF. La longitud de la carga de la orden HCI_PDU en octetos.
<b>HCI_Command_Payload</b>		
Carga ( <i>payload</i> )	1 octeto	La carga de una orden HCI_PDU es estructurada como una secuencia de campos de tamaño variable para varios parámetros relacionados a esta orden.

La Tabla 3.18 muestra la estructura de un evento HCI\_PDU. Similar a una orden HCI\_PDU, éste consiste de un campo *Event\_Code* para identificar el evento y una sección de carga con un número de parámetros dependiente del evento HCI\_PDU.

**Tabla 3.18.** Evento HCI\_PDU.

Campo	Tamaño	Función
<b>HCI_Event_Header</b>		
<i>Event_Code</i>	1 octeto	Identifica el evento: “0xFE” es reservado para eventos de especificación del logotipo Bluetooth; “0xFF” es reservado para eventos utilizados durante la fabricación del módulo.
<i>Payload_Length</i>	1 octeto	La longitud de la carga del evento HCI_PDU en octetos.
<b>HCI_Event_Payload</b>		
Carga ( <i>payload</i> )	1 octeto	La carga de un evento HCI_PDU es estructurada como una secuencia de campos de tamaño variable para varios parámetros relacionados a este evento.

La Tabla 3.19 muestra la estructura de un dato HCI\_PDU. La transmisión de datos HCI\_PDU a través de la interfaz física es regulada por el tamaño de la memoria intermedia (*buffer*) disponible en el lado receptor. Lo anterior implica que una L2CAP\_PDU grande puede ser fragmentado dentro de la capa HCI antes de enviarse hacia el controlador del huésped.

**Tabla 3.19.** Datos HCI\_PDU.

Campo	Tamaño	Función
<b>HCI_Datat_Header</b>		
<i>Connection_Handle</i>	12 bits	Identifica el enlace de banda base sobre el cual los datos son transmitidos o recibidos.
<i>Flags</i>	4 bits	Transmisiones ACL: compuestas de dos subcampos: <i>Packet_Boundary_Flag</i> identifica el inicio o continuación de un PDU de capas superiores; y <i>Broadcat_Flag</i> identifica el factor de expansión para la transmisión ACL. Transmisiones SCO, campo reservado.
<i>Payload_Length</i>	2 octetos	Longitud de la carga de los datos HCI_PDU en octetos.
<b>HCI_Datat_Payload</b>		
Carga ( <i>payload</i> )	<i>Payload_Length</i> octetos	Datos a ser cargados sobre el enlace de banda base ACL o SCO identificado por el contenido del campo <i>Connection_Handle</i> .

### 3.7.3.2.2. HCI\_PDUs

Existen diversas órdenes HCI\_PDU organizadas en grupos identificados por el subcampo OGF en la cabecera de la orden HCI\_PDU. Para la mayoría de órdenes, la información relacionada a su estado y ejecución, se carga mediante dos eventos especiales:

- *Command\_Status\_Event*: enviado inmediatamente después de haber recibido una orden del controlador del huésped para indicar el estado de la orden.
- *Command\_Complete\_Event*: para indicar que ha finalizado la ejecución de una orden y para regresar parámetros relacionados.

### 3.7.3.2.3. HCI\_PDUs de control de enlace

Las órdenes en este grupo, son identificados mediante el subcampo OGF con el valor “000001”. La orden *HCI\_Inquiry* (Tabla 3.20) informa al módulo para cambiar al modo búsqueda, usando un código de acceso dado.

**Tabla 3.20.** Orden HCI\_inquiry.

Orden		HCI_Inquiry	
OCF		“0000000001”	
Parámetros	LAP	3 octetos	Parte de direcciones inferiores usadas para generar el código de acceso de búsqueda.
	Inquiry_Length	1 octeto	Indica la máxima duración para la búsqueda: 1.28–61.44 s.
	Num_Responses	1 octeto	Indica el número máximo de respuestas a ser reunidas.

### 3.7.3.2.4. Políticas de enlace HCI\_PDUs

Este tipo de órdenes se identifican mediante el valor “000010” en el campo OGF. Este grupo contiene órdenes que permiten a un dispositivo establecer políticas de manejo de potencia a través de los modos reposo, sondeo y estacionado. La Tabla 3.21 muestra las órdenes HCI\_PDU que utiliza el controlador del huésped para indicar, a las subcapas controlador de enlace y banda base para cambiar al modo reposo con los parámetros proporcionados.

**Tabla 3.21.** Orden HCI\_Hold\_Mode.

Orden		HCI_Hold_Mode	
OCF		“0000000001”	
Parámetros	Connection_Handle	2 octetos	Identifica la conexión para ser fijada en modo <i>hold</i> ; solamente se utilizan los 12 LSBs.
	Hold_Mode_Max_Interval	2 octetos	Indica el intervalo máximo <i>hold</i> negociable. (0.625 ms–40.9 s).
	Hold_Mode_Min_Interval	2 octetos	Indica el intervalo mínimo <i>hold</i> negociable (0.625 ms–40.9 s).

### 3.7.3.2.5. Controlador de huésped y HCI\_PDUs de banda base

Este grupo de órdenes se identifican con el valor “000011” en el subcampo OGF. Este grupo contiene órdenes que permiten al huésped acceder y configurar varios registros que almacenan parámetros operacionales. La Tabla 3.22 muestra el formato de la orden HCI\_PDU para configurar los parámetros de sondeo de búsqueda.

**Tabla 3.22.** Orden HCI\_Write\_Page\_Scan\_Activity.

Orden		HCI_Write_inquiry_Scan_Activity	
OCF		“0000011100”	
Parámetros	Inquiry_Scan_Interval	2 octetos	Determina el intervalo entre inicios satisfactorios de sondeos de búsqueda (11.25 ms–2.56 s)
	Inquiry_Scan_Window	2 octetos	Determina la duración de una operación continua (11.25 ms–2.56 s)

### 3.7.3.2.6. Parámetros de información HCI\_PDUs

Este grupo de órdenes se identifica con el valor “000100” mediante el subcampo OGF. Éste incluye órdenes que solicitan información estática del *hardware* y *firmware* electrónicamente grabada en el dispositivo durante su fabricación:

- Versión de diferentes protocolos que el módulo puede soportar.
- Características soportadas por el controlador de enlace.
- País en que operará del módulo.
- BD\_ADDR del módulo.
- Información de la memoria temporal para paquetes ACL y SCO.

La información solicitada es devuelta en una respuesta *Command\_Complete\_Event*.

### 3.7.3.2.7. Parámetros de estado HCI\_PDUs

Este grupo de órdenes se identifican con el valor “000101” mediante el subcampo OGF “000101”. Éste incluye órdenes que solicitan información que se actualiza dinámicamente, por ejemplo existe una orden HCI\_PDU para recuperar información relacionada con la calidad del enlace y el indicador de la fuerza de la señal recibida (RSSI, *Received Signal Strength Indicador*).

### 3.7.3.2.8. Examinación HCI\_PDUs

Este grupo de órdenes se identifican con el valor “000110” mediante el subcampo OGF, para examinar el módulo Bluetooth.

## 3.7.4. Protocolos de capa intermedia

A continuación se describe la subcapa el RFCOMM, basada en el protocolo de emulación del puerto serie Bluetooth, y el protocolo de servicio de descubrimiento (SDP, *Service Discovery Protocol*). RFCOMM y SDP son subcapas superiores a la sub capa L2CAP y utilizan conexiones L2CAP para completar sus respectivas funciones. Estos protocolos se desarrollaron específicamente para su uso con la comunicación inalámbrica Bluetooth. La Figura 3.23 muestra ambas subcapas dentro de la arquitectura de protocolos Bluetooth.

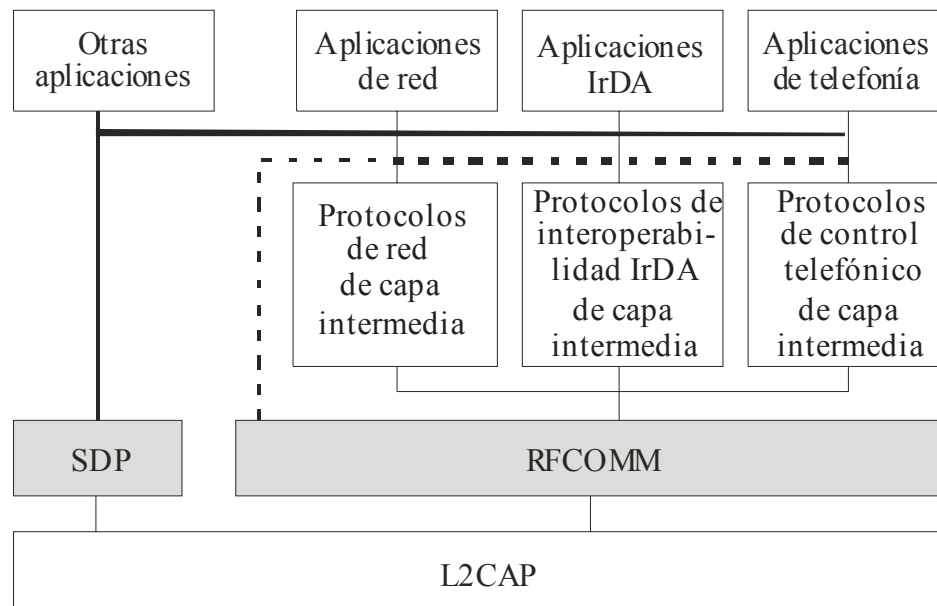
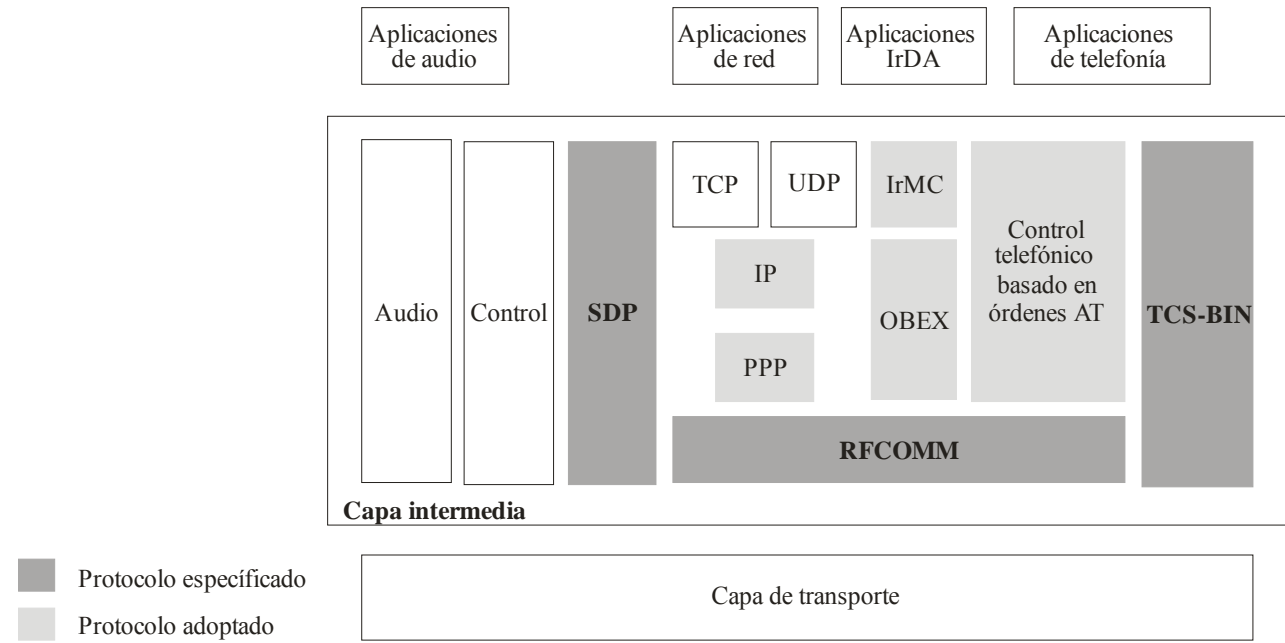


Figura 3.23. Subcapas RFCOMM y SDP.



**Figura 3.24.** Protocolos de la capa intermedia.

La Figura 3.24 muestra el grupo de protocolos de capa intermedia, los cuales hacen uso de los protocolos de transporte y presentan interfaces estandarizadas a las capas de aplicación para comunicarse a través de las capas de transporte. Dichos protocolos son:

- *RFCOMM*: presenta una abstracción de un puerto serie.
- *SDP*: para describir servicios disponibles y para localizar servicios requeridos.
- *Protocolos de interoperabilidad IrDA (Infrared Data Association)*: adoptados del estándar IrDA.
- *Protocolo de control telefónico (TCP, Telephony Control Protocol)*: usado para control de llamadas telefónicas, las cuales se utilizan para transmisión de audio o de datos.

### 3.7.4.1. Subcapa RFCOMM

Debido a que el puerto serie es una de las interfaces de comunicación más utilizada entre dispositivos de cómputo, la especificación de Bluetooth define un puerto serie llamado RFCOMM, el cual presenta virtualmente un puerto serie para facilitar la migración de aplicaciones modeladas para ambientes seriales cableados a comunicaciones seriales inalámbricas.

#### 3.7.4.1.1. Protocolo RFCOMM

Las interfaces seriales son fundamentales en dispositivos de cómputo y comunicaciones, típicamente estos dispositivos cuentan con un puerto serie, medio por el cual se comunica con otros dispositivos. Debido a que la tecnología Bluetooth está proyectada para remplazar los cables, la arquitectura de protocolos necesita soportar comunicación serie en el mismo modo que se hace con cables.

El SIG ha definido una capa en la pila del protocolo parecida a una interfaz serie, esta se conoce como RFCOMM. En el mundo de las computadoras personales, las interfaces seriales son llamadas puertos COM. El nombre de RFCOMM connota un modelo inalámbrico (RF) de un puerto COM virtual.

### 3.7.4.1.2. Proceso de desarrollo del protocolo RFCOMM

La motivación para desarrollar la subcapa RFCOMM fue soportar las aplicaciones iniciales de Bluetooth, cuyos principales requerimientos de comunicación serie fueron:

- *Comunicaciones seriales multiplexadas*: soporte simultáneo a diferentes Clientes, de manera que el puerto serie necesita estar compartido a través de conexiones multiplexadas.
- *Compatibilidad RS-232*: la emulación de estas señales permite a RFCOMM presentar a los Clientes una apariencia de un puerto serie virtual.
- *Estatus y configuración remota*.
- *Puerto serie interno y externo*: RFCOMM necesita emular ambos tipos de puertos seriales.
- *Establecimiento y terminación de la conexión*.
- *Aplicabilidad*.
- *Control de flujo*.

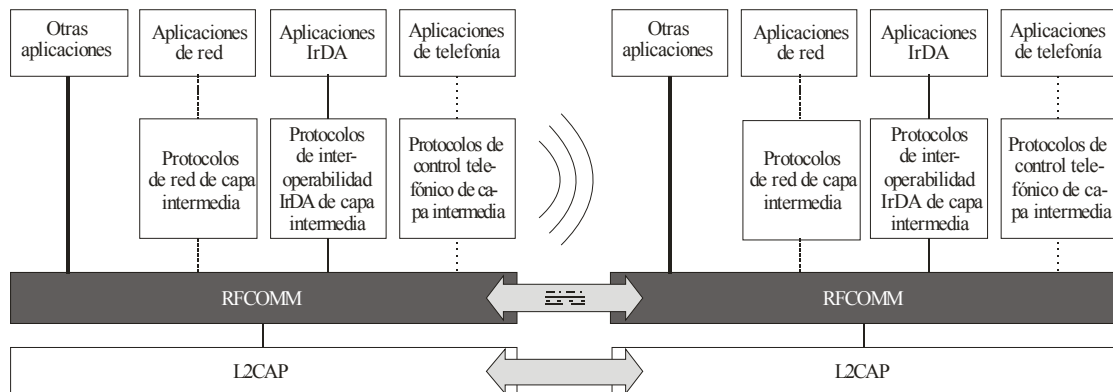
### 3.7.4.1.3. Análisis del protocolo RFCOMM

El principal propósito de este protocolo es habilitar el soporte para aplicaciones en escenarios cableados. RFCOMM utiliza una conexión L2CAP para iniciar un enlace lógico serie entre dos dispositivos. Solamente se permite una conexión RFCOMM entre dos dispositivos en un tiempo dado, pero la conexión puede estar multiplexada para dar soporte a múltiples conexiones seriales entre los dispositivos. El primer Cliente RFCOMM establece la conexión RFCOMM sobre L2CAP; y los demás usuarios de la conexión existente pueden hacer uso de las capacidades de multiplexación de RFCOMM para establecer nuevos canales sobre el enlace existente; y el último usuario en dejar el enlace serie RFCOMM deberá terminar la conexión RFCOMM.

Cada enlace multiplexado es identificado por un número llamado identificador de conexión de enlace de datos (DLCI, *Data Link Connection Identifier*). La Figura 3.25 ejemplifica las comunicaciones seriales multiplexadas usando RFCOMM sobre L2CAP.

La especificación permite hasta 60 enlaces lógicos seriales multiplexados sobre una conexión RFCOMM. RFCOMM soporta dos tipos de dispositivos: punto final de comunicación (*communication endpoint*), por ejemplo computadora y periféricos; y punto intermedio de comunicación (*communication midpoint*), por ejemplo los modems.

En un ambiente multiplexado, se prefiere que cada canal serie sea visto como una entidad independiente, con su propio conjunto de señales de datos y de control. RFCOMM cumple con lo anterior con la finalidad de definir un canal de control específico a través del cual la información se transmita como datos. La velocidad de transmisión de la comunicación inalámbrica Bluetooth se determina por los tipos de paquetes y estructuras enviadas sobre la interfaz aérea.



**Figura 3.25.** Comunicaciones de enlaces seriales multiplexados sobre una conexión RFCOMM.

### 3.7.4.2. Subcapa SDP

El SDP es un proceso mediante el cual los dispositivos y servicios en la red pueden localizar, reunir información y hacer uso de otros servicios en la red. En redes tradicionales como LANs, estos servicios pueden ser configurados y manejados por el administrador de la red. En estos ambientes, el administrador o usuario final realiza la configuración necesaria para que alguien participe en la red y pueda utilizar los servicios de algún otro miembro de la red.

El servicio de redes administradas es conveniente para algunas redes fijas, pero no para redes móviles temporales, como es el caso de las redes Bluetooth, las cuales requieren de soluciones dinámicas, flexibles y adaptativas. El SIG observó el creciente número de este tipo de redes y la resultante demanda de sistemas autoconfigurables, es decir que estos dispositivos y servicios simplemente deben descubrir a otros, negociando lo que quieren realizar. Los protocolos de servicio de descubrimiento ayudan a habilitar esta autoconfiguración, y cabe destacar que los servicios de descubrimiento no es un concepto nuevo o único de la tecnología inalámbrica Bluetooth.

#### 3.7.4.2.1. Proceso de desarrollo de la subcapa SDP

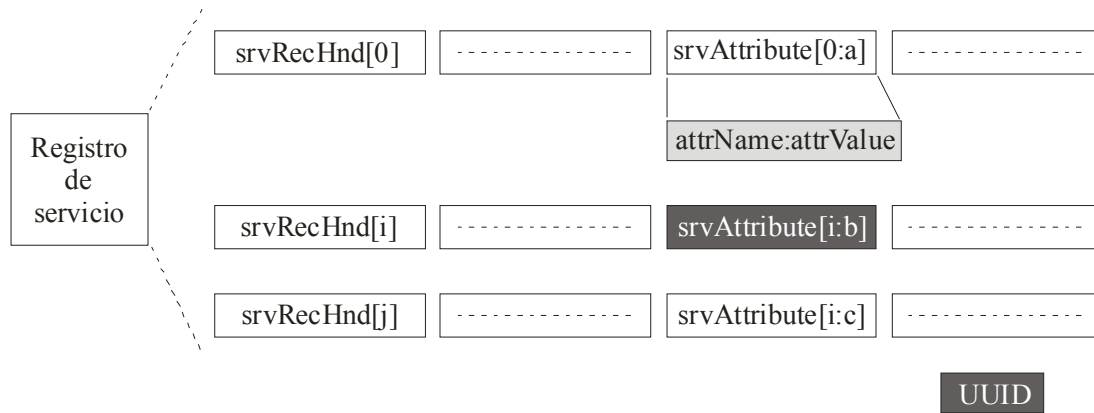
La necesidad de un componente de descubrimiento de servicios en la arquitectura de protocolos fue reconocida al inicio del proceso de desarrollo de la especificación de Bluetooth. En el verano de 1998, algunos trabajos preliminares iniciados por Intel y Ericsson dieron inicio al SDP. Los objetivos para la creación de SDP fueron:

- *Simplicidad*: debido a que el descubrimiento de servicios es una parte de casi todos los casos de uso de Bluetooth, es deseable que el proceso de descubrimiento de servicios sea tan simple como sea posible para su ejecución.
- *Compacto*: el descubrimiento de servicios es una operación típica a realizarse tan pronto como el enlace sea establecido, el tráfico de SDP debe de ser mínimo para ser factible y que el descubrimiento de servicios no prolongue el proceso de inicialización de la comunicación.
- *Versatilidad*: con el constante crecimiento de herramientas y servicios, que no pueden preverse o predecirse, es importante para SDP ser fácilmente extensible y versátil para soportar nuevos servicios que podrían desarrollarse posteriormente.
- *Localización de servicios por clase y por atributo*: en redes *ad-hoc* es importante habilitar Clientes y usuarios para localizar rápidamente un servicio específico cuando sea necesario. Esto resulta más fácil en servicios de una clase en general (por ejemplo “impresora”), pero si se busca algo más específico (por ejemplo “impresora IBM de color”) es necesario contar con atributos específicos asociados con estos servicios.
- *Sondeo de servicios*: además de la localización de servicios por clase o atributo, es útil realizar un simple sondeo a los servicios disponibles para determinar si existe alguno de interés.

#### 3.7.4.2.2. Análisis de la subcapa SDP

SDP tiene la intención de permitir a los dispositivos localizar servicios disponibles. Dentro de las picoredes Bluetooth, los dispositivos y servicios varían frecuentemente, por ello SDP fue desarrollado para satisfacer los requerimientos de tales ambientes.

SDP define una topología Cliente/Servidor. Cualquier dispositivo puede asumir cualquier rol en un tiempo dado, actuando a veces como un Cliente del servicio y a veces como un proveedor de servicio (Servidor).



**Figura 3.26.** Estructura del registro de servicio SDP.

El proveedor de servicio necesita mantener una lista que describa los servicios que proporciona; esta lista es el registro de servicio (*service registry*), el cual consiste de una colección de atributos de servicio, los cuales contienen información sobre la clase del servicio, información sobre las capas de la arquitectura del protocolo necesarias para interactuar con el servicio. En la Figura 3.26 se muestra la estructura general de un registro de servicios.

El registro de servicios contiene dos tipos de atributos:

- *Atributos de servicios universales (Universal service attributes)*: pueden aplicarse a todos los tipos de servicio y no son obligatorios, a excepción de: atributo de clase de servicio, el cual define la clase o el tipo de servicio; y manejo de registro de servicio (*service record handle*), el cual sirve de puntero, o referencia al registro de servicios y es usado por el Cliente para acceder al registro de servicios del servidor.

Cada atributo en un registro de servicios consiste de un identificador de atributo (*attribute ID*, un entero de 16 bits) y un valor asociado con este ID de atributo.

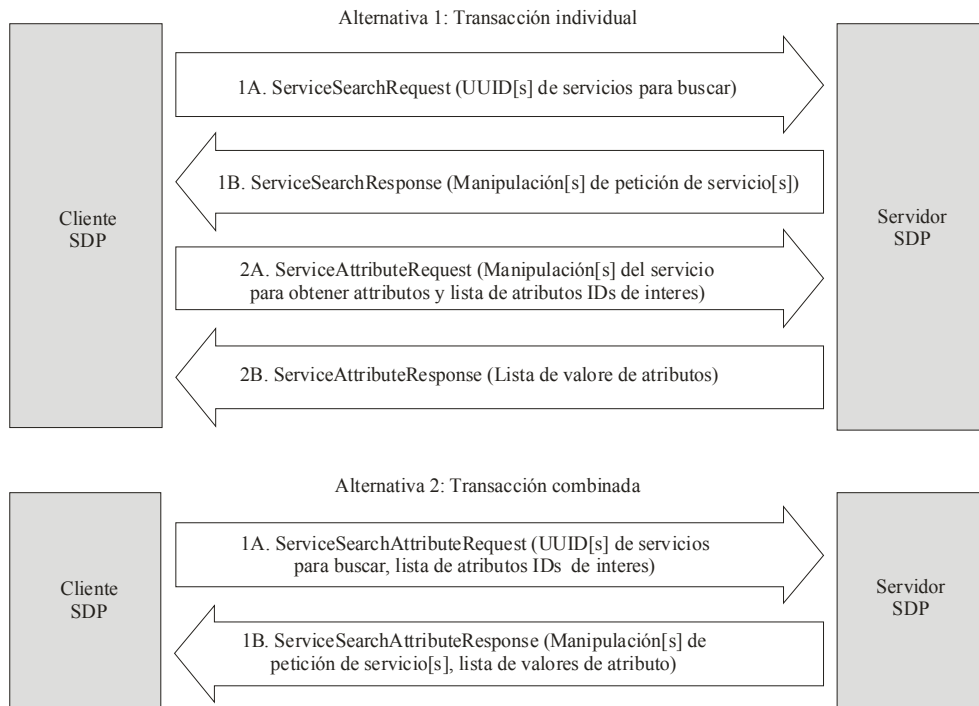
- *Atributos de servicios específicos (Specific service attributes)*: son relevantes únicamente para clases específicas de un servicio.

El descubrimiento de un servicio en Bluetooth es una operación simple: el Cliente especifica el servicio de interés y el Servidor responde, indicando algún servicio disponible que coincida con el especificado por el Cliente. Para SDP, esto consiste en el envío de una petición del Cliente en la forma de una unidad de datos del protocolo SDP (SDP\_PDU) indicando qué servicio está buscado, y el Servidor envía una respuesta, también en la forma de una SDP\_PDU.

Para completar el proceso, el Cliente y Servidor necesitan un formato estándar para representar el servicio de interés. Para este propósito, SDP define los identificadores universalmente únicos (UUIDs, *Universally Unique Identifiers*). Un UUID está compuesto de 128 bits que pueden ser creados algorítmicamente y contar con la garantía virtual de ser únicos. La Figura 3.27 ilustra las transacciones SDP.

### 3.7.4.3. Subcapa de protocolos IrDA

La subcapa intermedia de operabilidad IrDA (*IrDA interoperability middleware*) incluye protocolos adoptados de la Asociación de Datos Infrarrojos (IrDA, *Infrared Data Association*), conocida como IrOBEX (u OBEX, *Object Exchange*) y de las Comunicaciones Móviles Infrarrojas (IrMC, *Infrared Mobile Communications*). La interoperabilidad de aplicación IrDA es un principio fundamental en el diseño de la comunicación inalámbrica Bluetooth, y de esta manera la subcapa intermedia de interoperabilidad IrDA es un elemento clave en la arquitectura de protocolos Bluetooth.



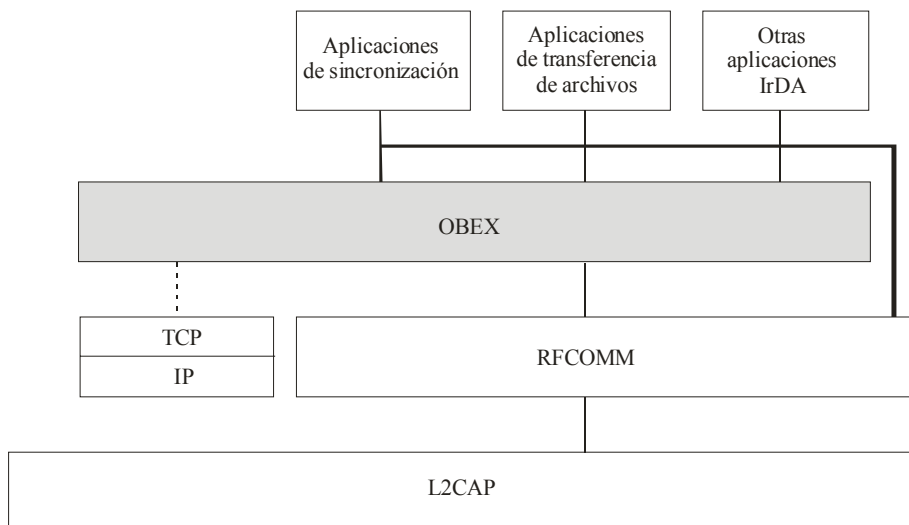
**Figura 3.27.** Transacciones SDP.

El protocolo de sesión OBEX utiliza una estructura PDU, permitiendo a las capas superiores trabajar con elementos de datos lógicos en un alto nivel de abstracción, que el formato de paquetes utilizado por los protocolos de transporte o incluso por RFCOMM. En la Figura 3.28 se muestra OBEX dentro de la arquitectura de protocolos Bluetooth.

#### 3.7.4.3.1. Protocolos de interoperabilidad IrDA

Dentro de las aplicaciones más comunes de IrDA está el intercambio de archivos y otros objetos. Los dispositivos comúnmente utilizados en estos escenarios son el mismo conjunto de dispositivos que emplea la tecnología Bluetooth.

Bluetooth emplea la interoperabilidad de IrDA para operar sobre ambos enlaces, IrDA y Bluetooth, y la manera más simple de realizarlo es mediante el uso del mismo protocolo de sesión en ambos ambientes.



**Figura 3.28.** Protocolo OBEX.



### 3.7.4.3.2. Proceso de desarrollo del protocolo de interoperabilidad IrDA

Dado el objetivo de interoperabilidad entre aplicaciones de IrDA y Bluetooth, una meta inicial del SIG fue producir una especificación que permitiera una aplicación sencilla para operar sobre ambas tecnologías inalámbricas. Lo anterior permitió considerar a OBEX como el punto dentro de la arquitectura de protocolos IrDA para considerarse dentro de la arquitectura Bluetooth y permitir aplicaciones con el mismo protocolo en ambos ambientes.

### 3.7.4.3.3. Protocolo OBEX

RFCOMM es el protocolo Bluetooth sobre el cual OBEX está diseñado. Las consideraciones para OBEX sobre RFCOMM son:

- *Funciones Cliente y Servidor:* la especificación indica que ambas funciones deben ser soportadas por dispositivos implementando el protocolo de interoperabilidad OBEX IrDA.
- *Multiplexación RFCOMM:* todas las transacciones OBEX deben utilizar un canal servidor RFCOMM con la finalidad de que diferentes Clientes de RFCOMM puedan utilizar la característica de multiplexación de protocolo.
- *Soporte de SDP:* las aplicaciones de OBEX en ambientes Bluetooth necesitan ser capaces de hacer uso de SDP. Los Clientes OBEX necesitan obtener la información relevante sobre el servicio OBEX de su registro de servicios en el servidor OBEX.

OBEX proporciona un protocolo de sesión para transacciones entre dos dispositivos. IrDA define sesiones orientadas a conexión y no orientados a conexión. Con OBEX, el rol del Cliente normalmente es asumido por el dispositivo que inicia la transacción, mientras el dispositivo que responde llega a ser el Servidor. Las PDUs OBEX consisten de una cabecera, un indicador de tamaño y los argumentos y parámetros asociados con la transacción particular.

### 3.7.4.4. Subcapa de control de audio y telefonía

El protocolo de control telefónico es realizado por la capa TCS-BIN (*Telephony Control Specification Binary*), mientras que el audio puede ser cargado sobre la banda base. TCS-BIN es una codificación binaria para control telefónico basado en paquetes, aunque las aplicaciones que utilizan órdenes AT (*Attention*) sobre el puerto serie RFCOMM pueden ser una forma de control telefónico.

Por sí mismo el audio no es una capa de protocolos, únicamente es un paquete que puede transmitirse directamente sobre la capa de banda base. La Figura 3.29 muestra el audio y TCS-BIN dentro de la arquitectura de protocolos Bluetooth.

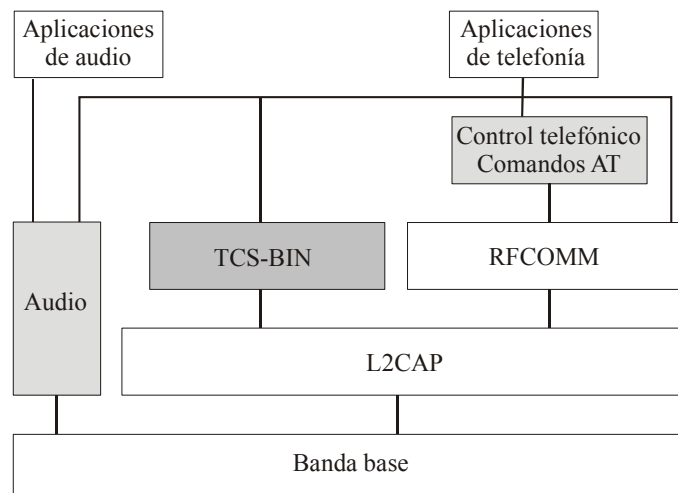


Figura 3.29. Subcapas de audio y TCS-BIN.

#### 3.7.4.4.1. Operación de control de audio y telefonía

TCS-BIN es usada para aspectos de control de llamadas de telefonía, incluyendo el establecimiento y terminación de las llamadas junto con muchas otras funciones que son aplicables a las llamadas de teléfono. TCS-BIN puede ser usado para controlar ambas llamadas de voz y datos. Cuando una llamada de voz es realizada, el elemento de audio de la pila es usada para cargar su contenido; en el caso de llamadas de datos el dato puede ser cargado sobre las capas de transporte de la pila.

TCS-BIN también define un método para que los dispositivos intercambien información de señalización de llamadas sin tener una conexión de llamada establecida entre ellos; éste es llamado TCS-BIN sin conexión. Asimismo TCS-BIN es el único protocolo de control telefónico definido como una entidad separada en la especificación, y es el protocolo en el cual se basan varias propiedades de telefonía.

#### 3.7.4.4.2. Protocolo TCS

El control telefónico está relacionado con las funciones de audio, y en realidad fue el audio el que condujo la necesidad para el control telefónico. Al principio los requerimientos de audio apuntaron hacia la necesidad de algunas funciones de control, lo cual inicialmente fue presentado como funciones de control de audio. Debido a que las aplicaciones en telefonía fueron incrementándose, llegó a ser evidente la necesidad de contar con funciones de control telefónico y un grupo de trabajo fue formado para definir estas funciones para la pila del protocolo.

De esta manera fue como nació la especificación TCS-AT. Esta especificación tenía la intención de describir como el estándar AT podría ser direccionado sobre la pila de protocolo Bluetooth y definir algún nuevo comando AT requerido por Bluetooth. TCS-AT fue diseñado para soportar aplicaciones que enviaran y recibieran comandos AT sobre un puerto serie. Mientras tanto, un protocolo binario de control telefónico basado en paquetes empezó a ser definido dentro de la pila del protocolo Bluetooth. Llamado TCS-Binary o TCS-BIN, éste fue adaptado de una especificación existente ITU\_T (*International Telecommunications Union Telecommunication*).

A principio de 1999 se llegó a la conclusión que el futuro de las aplicaciones de control telefónico estaba orientado al estilo TCS-BIN, asimismo se observó que TCS-BIN proporcionaba todas las funciones necesarias para todos los perfiles basados en la telefonía. Finalmente, también se determinó que la especificación TCS-AT no proporcionaba nuevas funciones específicas para ambientes Bluetooth. Así fue como TCS-BIN se sumó como un protocolo separado en la pila. EL SIG decidió remover TCS-AT como una especificación separada, aunque las funciones no fueron removidas, únicamente el nombre.

#### 3.7.4.4.3. Análisis del protocolo TCS

En adición a la especificación TCS se definieron tres principales áreas funcionales:

- Control de llamada. TCS actúa como una máquina de estado, realizando las operaciones necesarias para avanzar una llamada de un estado al siguiente, y buscando el estado resultante. En ambientes multipunto, TCS utiliza señalización multipunto para alertar a todos los teléfonos de una llamada entrante, esto puede entonces establecer un canal con el teléfono que responda la llamada.
- Manejo de grupo. Las funciones de manejo de grupo usan el concepto de un grupo de usuarios inalámbricos (WUG, *Wireless User Group*). Un grupo puede utilizar las funciones de manejo de grupo TCS para tomar ventaja en algunas funciones que proporciona TCS. Estas funciones incluyen un método para que un dispositivo haga uso de los servicios telefónicos de otro dispositivo en el grupo. Un dispositivo en un WUG puede pedir los

servicios de otro dispositivo en el WUG; TCS llama a esto una petición de los derechos de acceso (*access rights request*).

- TCS sin conexión. TCS-BIN también proporciona una forma en la cual los dispositivos intercambien información de señalización de llamadas sin dar lugar a una llamada o tener una conexión TCS establecida. Esto es llamado TCS sin conexión, el cual proporciona un tipo de banda lateral en la cual los dispositivos dentro de un WUG puedan enviar mensajes a otros sin tener una conexión TCS establecida entre ellos. La especificación define solamente un formato de mensaje para TCS sin conexión llamado *CL Info*. Estos mensajes pueden contener solamente dos tipos de información: control de audio (ganancia del micrófono, volumen de la bocina) e información extra (información no especificada en TCS).

#### 3.7.4.4.4. Desarrollo del audio Bluetooth

El audio ha sido una parte inherente de Bluetooth desde su inicio y de esta manera siempre ha sido integrado dentro del diseño fundamental de la pila del protocolo. El audio es cargado sobre enlaces SCO en la capa de banda base.

#### 3.7.4.4.5. Análisis del audio en Bluetooth

Dos tipos de esquemas de codificación son especificados para audio Bluetooth. El primero es la modulación por codificación de pulsos (PCM, *Pulse Code Modulation*) con cualquiera de los dos tipos de compresión logarítmica.

El segundo esquema de codificación es la codificación delta (CVSD, *Continuous Variable Slope Delta*) mas inmune a interferencia y de esta manera mejor para la comunicación de voz sobre el enlace inalámbrico.

En Bluetooth, el audio, especialmente voz, se trata por separado debido a que el tráfico de audio es síncrono y presenta requerimientos especiales en cuanto a tiempos; por otro lado el tráfico de voz es dirigido directamente a la capa de banda base.

La comunicación de audio Bluetooth se realiza a 64 Kbps mediante los esquemas de codificación de datos: modulación de código de pulsos (PCM) y/o modulación delta (CVSD). Los canales de audio Bluetooth pueden soportar otras formas de audio como son música o trozos de audio.

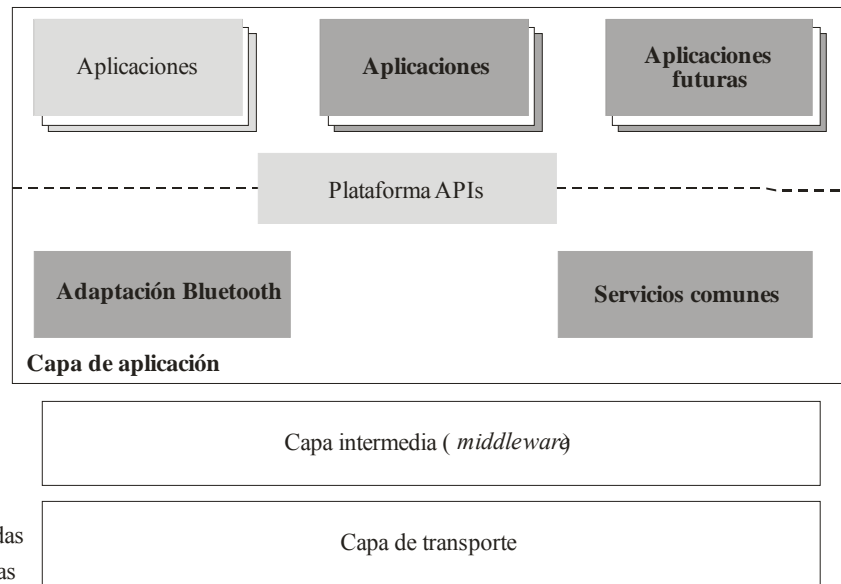
#### 3.7.4.5. Subcapa de red

Bluetooth utiliza topología de red P2P de forma similar a la que se utiliza en redes LAN, sin embargo, la tecnología hace uso de permisos de redes, particularmente en conexiones de redes más grandes a través de una conexión de marcación o mediante una red de punto de acceso.

### 3.7.5. Capa de aplicación

Este grupo se refiere al SW de aplicación que representa la parte alta de la arquitectura de protocolos y es proporcionado por fabricantes de dispositivos Bluetooth. La Figura 3.30 ilustra las diferentes posibilidades de organización para el SW de aplicación Bluetooth.

El SIG define únicamente los protocolos de las capas intermedias y de transporte, y para la capa de aplicación define las interfaces de programación de aplicación (APIs, *Application Programming Interfaces*), sin embargo, las aplicaciones necesitan definir escenarios para hacer uso de la comunicación inalámbrica Bluetooth.



**Figura 3.30.** Capa de aplicación.

## 4. Desarrollo del CISBlue-488

En este capítulo se describe el diseño y construcción del CISBlue-488 con base en la metodología de desarrollo de sistemas empotrados (Subcapítulo 1.1).

### 4.1. Especificación del CISBlue-488

El sistema inalámbrico de instrumentación electrónica programable emplea tres protocolos de comunicaciones:

- *GPIB*: para comunicar al controlador GPIB con los instrumentos electrónicos programables.
- *RS232*: para lograr la interfaz entre el GPIB y el módulo Bluetooth.
- *Bluetooth*: para controlar el GPIB con los instrumentos electrónicos programables mediante un enlace de RF en distancias cortas.

Considerando las especificaciones de dichos protocolos, el CISBlue-488 debe cumplir con lo siguiente:

- Utilizar un MCU que integre una UART (*Universal Asynchronous Receiver/Transmitter*) para comunicación RS232.
- Manipular las 16 líneas de transmisión del protocolo GPIB mediante un MCU.
- Controlar la función de transferencia del protocolo GPIB (Párrafo 2.2.3.3.1).
- Manejar la información en paquetes de 8 bits de forma serie.
- La pasarela GPIB/RS232 debe manipular la información del IEEE 488 (Capítulo 2).
- Manejar los estados del GPIB, mediante el estándar IEEE 488.1 (Subcapítulo 2.2).
- Utilizar niveles TTL.

Además de las especificaciones de cada protocolo, se definen los siguientes requerimientos:

- Manejar bajo consumo de potencia.
- Construir un producto de bajo coste.
- Compatibilidad con el estándar IEEE 488.2.

La pasarela GPIB/RS232 es el controlador del CISBlue-488, ya que ésta se encarga de: manipular las señales del GPIB; proporcionar los datos al módulo Bluetooth para su transmisión en el lado transmisor; recibir los datos del módulo Bluetooth; y controlar las líneas GPIB del instrumento correspondiente.

### 4.1.1. Segmentación del diseño

Como se mencionó con anterioridad, un sistema empotrado consta de un componente SW y un componente HW, por ello es necesario especificar las tareas correspondientes. La Tabla 4.1 muestra la segmentación correspondiente al diseño del CISBlue-488.

**Tabla 4.1.** Segmentación HW y SW del CISBlue-488.

Tareas HW	Tareas SW
Acoplamiento de señales TTL/RS232	Conversión de GPIB a tramas RS232 y viceversa
Acoplamiento de las señales de control y datos del GPIB	Validación GPIB y control de la función de transferencia
Transmisión y recepción de datos de forma inalámbrica vía RF	Sincronización de la transmisión y recepción inalámbrica

#### 4.1.1.1. Componentes HW

El HW del CISBlue-488 consta de dos partes principales, la pasarela GPIB/RS232 y el módulo Bluetooth, los cuales se describen a continuación.

##### 4.1.1.1.1. Pasarela GPIB/RS232

Para realizar la implementación de la pasarela GPIB/RS232 se eligió el MCU PIC16F877 de la firma Microchip debido a las siguientes características:

- Integra una USART (*Universal Synchronous Asynchronous Receiver Transmitter*).
- Frecuencia de reloj de hasta 20 MHz.
- Cuenta con cinco puertos de E/S, con lo cual permite manipular todas las líneas del GPIB.
- Se cuenta con el entorno de desarrollo integrado MPLAB IDE y el programador PICS-TART Plus, necesarios para programar el MCU PIC16F877, ambos de la firma Microchip.
- Dispone de arquitectura Harvard.
- Opera en el rango de 2.0 a 5.5 VCD.

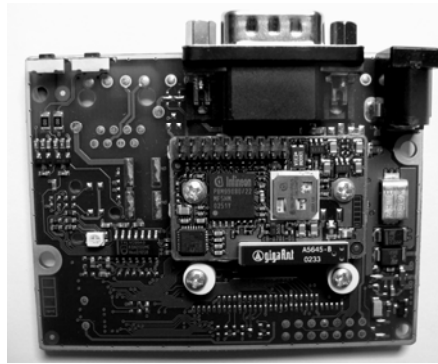
Además del MCU PIC16F877, la pasarela GPIB/RS232 integra los siguientes componentes:

- Convertidor de niveles TTL/RS232, MAX232 [6].
- Transceptor SN75160 de la firma Texas Instruments para acoplar el octeto de datos GPIB.
- Transceptor SN75161 de la firma Texas Instruments para acoplar el octeto de control GPIB.
- Conector tipo americano de 24 terminales.
- Conector tipo DB9.

##### 4.1.1.1.2. Módulo Bluetooth

Se utilizó el módulo OEM Module Adapter II de la firma connectBlue (Figura 4.1), considerando las siguientes características:

- No requiere de la manipulación humana.
- Cuenta con enlace de comunicación automático.
- Soporta conexión RS232 con otros dispositivos OEM (*Original Equipment Manufacturer*) mediante un conector tipo DB9.
- Dimensiones pequeñas (58 x 76 mm).
- Antena integrada.



**Figura 4.1.** Módulo Bluetooth.

- Opera a 5 VCD.
- Potencia de transmisión de 0 dBm (1mW).

El módulo Bluetooth consta de los siguientes componentes:

- Conector tipo DB9.
- Adaptador de puerto serie 33i de la firma ConnectBlue [URL2].
- Antena interna de la firma gigaAnt [URL5].

El Anexo A describe las características eléctricas y mecánicas del módulo Bluetooth.

#### **4.1.1.2. Componentes SW**

El MCU se programó con base en los requisitos que presentan los protocolos GPIB y Bluetooth. El programa se realizó en el entorno de desarrollo MPLAB IDE versión 6.60 de la firma Microchip, el cual se distribuye en forma gratuita (*freeware*). El programa principal realiza las siguientes funciones:

- Configuración del CISBlue-488 (Inicialización E/S).
- Envío y recepción de datos mediante RS232.
- Manejo de la función de transferencia del GPIB (Párrafo 2.2.3.3.1).
- Lectura y escritura al GPIB.
- Manipulación de las líneas de datos y de control del GPIB (Inciso 2.2.3.3).
- Validación de señales GPIB.
- Control de transmisión/recepción de los paquetes serie.
- Conversión de señales GPIB a paquetes RS232 y viceversa.

#### **4.1.2. Iteración y refinamiento**

En esta fase se refinan las particiones, enfatizando en las ambigüedades existentes en fases anteriores, de las cuales se mencionan las siguientes:

- Manejar una velocidad de transmisión serie de 115,200 Baudios.
- Realizar una herramienta para visualizar los datos en el GPIB.
- Realizar la interfaz de usuario.

Para el diseño del HW se utilizaron las siguientes herramientas:

- *Tarjeta de programación PICSTART Plus*: para programar el MCU PIC16F877.
- *Tarjeta controladora PCI/GPIB de la firma Agilent Technologies*: para realizar las funciones referentes al GPIB [URL1].

- *Unidad de adquisición de datos HYDRA de la firma FLUKE*: como instrumento electrónico programable mediante GPIB [URL6].

En el diseño de los componentes SW se emplearon las siguientes herramientas:

- *Entorno de desarrollo MPLAB IDE de la firma Microchip*: para establecer los bits de configuración, compilar, borrar y descargar los programas en formato ensamblador al PIC.
- *SW MPLAB SIM*: para modelar y simular los programas realizados.
- *SW TERM90 de la firma Symantec*: se utilizó como emulador de una terminal RS232 [URL11], con la finalidad de comprobar los datos.
- *SW VISA Assistant de la firma Agilent Technologies*: como interfaz de usuario para controlar el GPIB.

#### 4.1.3. Diseño de tareas HW y SW

A continuación se describe el diseño y desarrollo de las tareas y funciones descritas anteriormente.

##### 4.1.3.1. Diseño HW

Considerando los componentes y herramientas seleccionadas, se configuraron los siguientes parámetros:

- Puertos de E/S del MCU.
- Asignación de direcciones de memoria para las variables globales.
- Velocidad de transmisión serie a 115,200 Baudios.
- Registros del MCU.

A continuación se describen los parámetros que se definieron para el diseño del HW del CIS-Blue-488.

La Tabla 4.2 muestra las funciones asignadas a cada puerto del MCU.

**Tabla 4.2.** Asignación de funciones a los puertos del MCU.

Puerto	Función
A	Controla la función de transferencia del GPIB, las terminales TE ( <i>Talk Enable</i> ) del transceptor de control y del transceptor de datos.
B	Captura y transmite el octeto de datos al GPIB.
C	Mediante las terminales RC6 y RC7 se reciben y transmiten los datos en forma serie (USART).
D	Captura y transmite las líneas de control del GPIB (EOI, REN, IFC, ATN), excluyendo a SRQ.
E	Controla las terminales DC ( <i>Direction Control</i> ) del transceptor de control y SRQ del mismo transceptor.

Se emplearon cuatro variables globales, la Tabla 4.3 muestra la función y la dirección asignada a cada una de ellas.

**Tabla 4.3.** Dirección de memoria de las variables globales.

Variable	Función	Dirección
Conta	Conocer cuántos datos se han recibido o enviado	0x20
Datos	Almacena el octeto de datos GPIB que recibe vía serie	0x21
Control	Almacena el octeto de control GPIB que recibe vía serie	0x22
Conta2	Valida el fin de transmisión de datos del instrumento GPIB	0x23



La Tabla 4.4 muestra los registros del MCU utilizados.

**Tabla 4.4.** Registros de configuración del MCU.

Tipo	Nombre	Función	Dirección	Banco
Configuración	STATUS	Estado de la ALU, RESET y memoria de datos	03h, 83h, 103h, 183h	1, 2, 3 y 4
	INTCON	Habilitación de interrupciones y banderas	0Bh, 8Bh, 10Bh, 18Bh	1, 2, 3 y 4
E/S	PORTA	Datos del puerto A	05h	1
	TRISA	Configuración del puerto A	85h	2
	PORTB	Datos del puerto B	06h	1
	TRISB	Configuración del puerto B	86h	2
	PORTC	Datos del puerto C	07h	1
	TRISC	Configuración del puerto C	87h	2
	PORTD	Datos del puerto D	08h	1
	TRISD	Configuración del puerto D	88h	2
	PORTE	Datos del puerto E	09h	1
	TRISE	Configuración del puerto E	89h	2
USART	TXSTA	Control y estado de transmisión	98h	2
	RCSTA	Control y estado de recepción	18h	1
	SPBRG	Generador de la frecuencia de transmisión	99h	2
	TXREG	Transmisión de datos a puerto serie	19h	1
	RCREG	Recepción de datos de puerto serie	1Ah	2

#### 4.1.3.1.1. Comunicación con el GPIB

Para acoplar las líneas del GPIB a la pasarela GPIB/RS232 se interfazaron los transceptores SN75160 y SN75161 al PIC16F877.

La salida del transceptor de datos SN75160 se conectó al puerto B del MCU para capturar y enviar datos al GPIB; la terminal 1 (TE, *Talk Enable*) del transceptor se conectó a la terminal RA5 del puerto A del MCU; y la terminal PE (*Pullup Enable*) se conectó a la señal de voltaje VCC con la finalidad de controlar la transmisión y recepción del transceptor.

La salida del transceptor de control SN75161 se conectó al puerto D (ATN, EOI, IFC, REN) del MCU; las señales correspondientes a la función de transferencia GPIB (DAV, NRFD y NDAC) se conectaron al puerto A (RA0, RA1 y RA2, respectivamente); y la salida de la señal SRQ se conectó a la terminal RE1 del puerto E para monitorear las peticiones de servicio que requieran los instrumentos. El control de la transmisión/recepción se realizó mediante las terminales 1 (TE, *Talk Enable*) y 11 (DC, *Direction Control*), las cuales se conectaron a las terminales RA3 del puerto A y RE0 del puerto E.

#### 4.1.3.1.2. Comunicación entre la pasarela GPIB/RS232 y el módulo Bluetooth

Con base en la especificación de los protocolos GPIB y Bluetooth, se capturan los octetos de datos y de control GPIB; se convierten a formato serie mediante la USART del MCU (terminal RC6 del puerto C), la cual dirige los datos serie al convertidor de niveles TTL MAX232 [9] para acoplar los niveles de voltaje; finalmente, el circuito MAX232 pasa los datos al módulo Bluetooth, el cual se

encarga de recibirlos mediante su interfaz de puerto serie, convertirlos al formato de las tramas Bluetooth y transmitirlos vía RF.

#### 4.1.3.2. Diseño SW

De acuerdo al diseño HW, se consideraron las funciones SW definidas en las especificaciones iniciales, las cuales se describen a continuación.

##### 4.1.3.2.1. Configuración del CISBlue-488

En esta función se inicializan los puertos de E/S, la USART y el vector de interrupciones del MCU, así como también se configuran las interrupciones (Figura 4.2).

##### 4.1.3.2.2. Lectura del GPIB

Para leer el contenido del GPIB, es necesario utilizar su función de transferencia (Párrafo 2.2.3.3.1), y para transmitir las líneas de datos y de control vía serie es necesario disminuir la velocidad del GPIB (1 Mbps) a una velocidad adecuada al protocolo RS232. En este caso se eligió la velocidad de 115,200 Baudios, debido a que es la velocidad máxima que soporta el módulo Bluetooth. En la Figura 4.3 se muestra el diagrama de flujo que realiza dicha función.

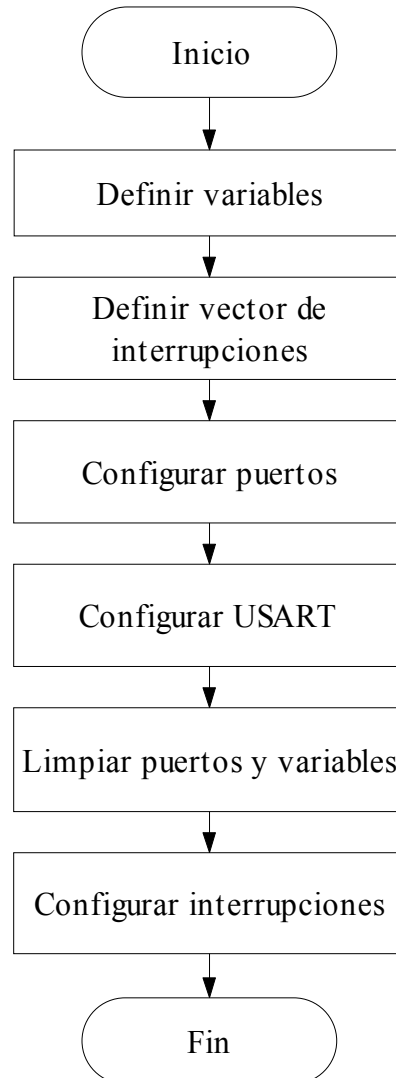
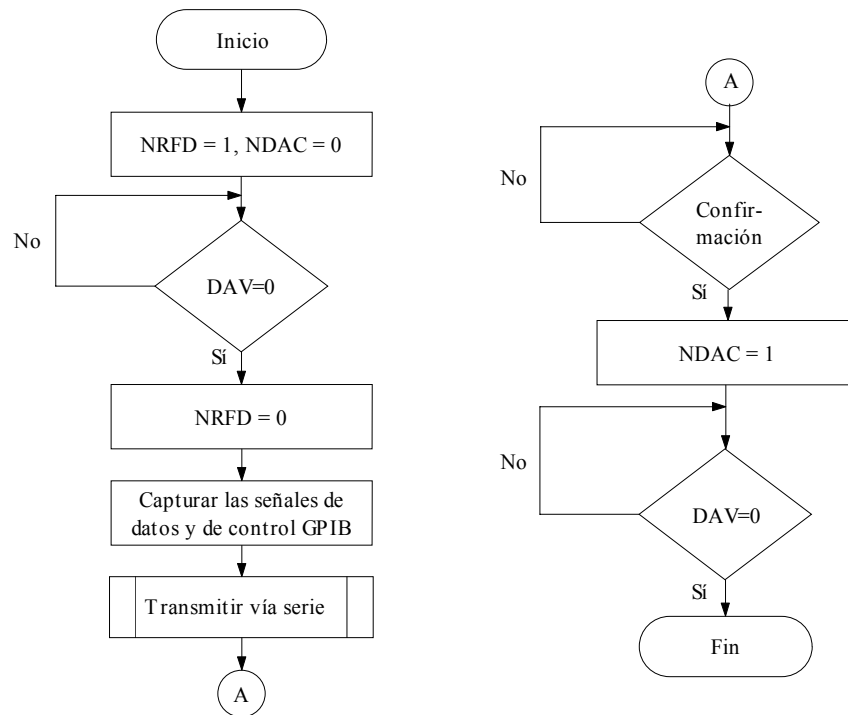


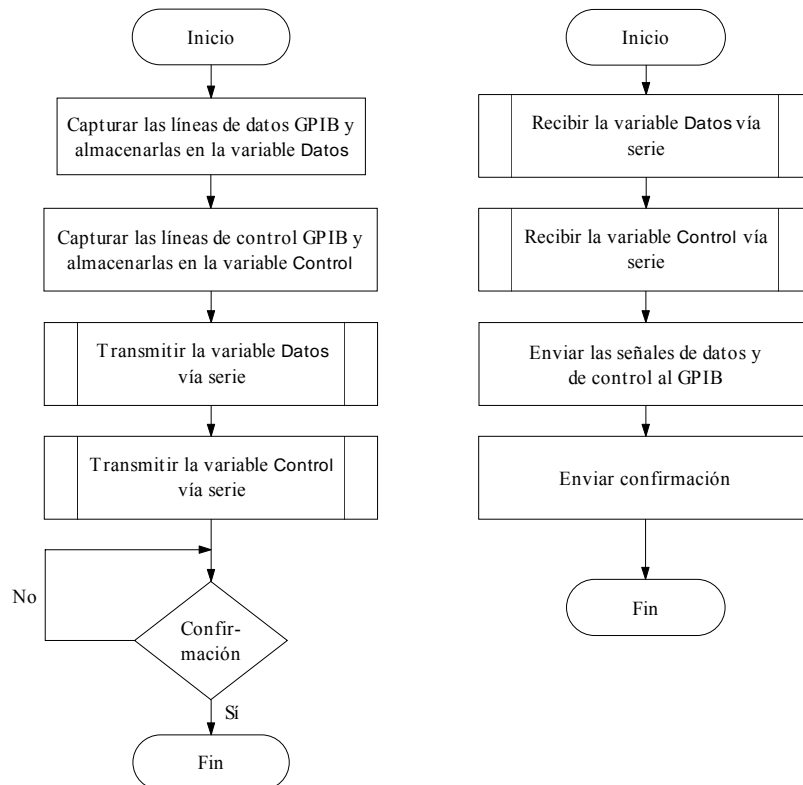
Figura 4.2. Diagrama de flujo de la inicialización.



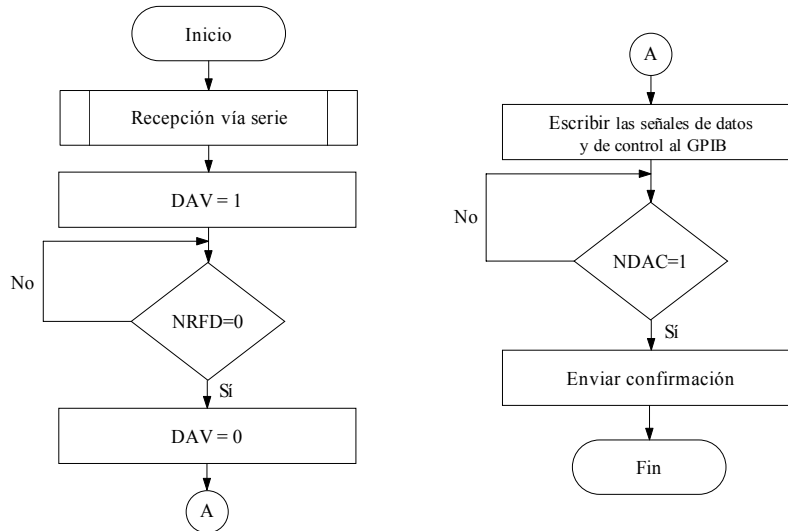
**Figura 4.3.** Diagrama de flujo de la lectura del GPIB.

4.1.3.2.3. *Comunicación serie*

Esta función envía y/o recibe los datos capturados del GPIB y los convierte a tramas RS232/GPIB mediante la USART del MCU, también se encarga de sincronizar el envío de los datos al GPIB. Esta función está basada en los diagramas de flujo de la Figura 4.4.



**Figura 4.4.** Diagramas de flujo para la transmisión y recepción serie.



**Figura 4.5.** Diagrama de flujo para la función escritura al GPIB.

#### 4.1.3.2.4. Escritura al GPIB

De igual forma que la función de lectura del GPIB, es necesario utilizar la función de transferencia. La Figura 4.5 muestra el diagrama de flujo para la función de escritura.

#### 4.1.4. Integración HW y SW

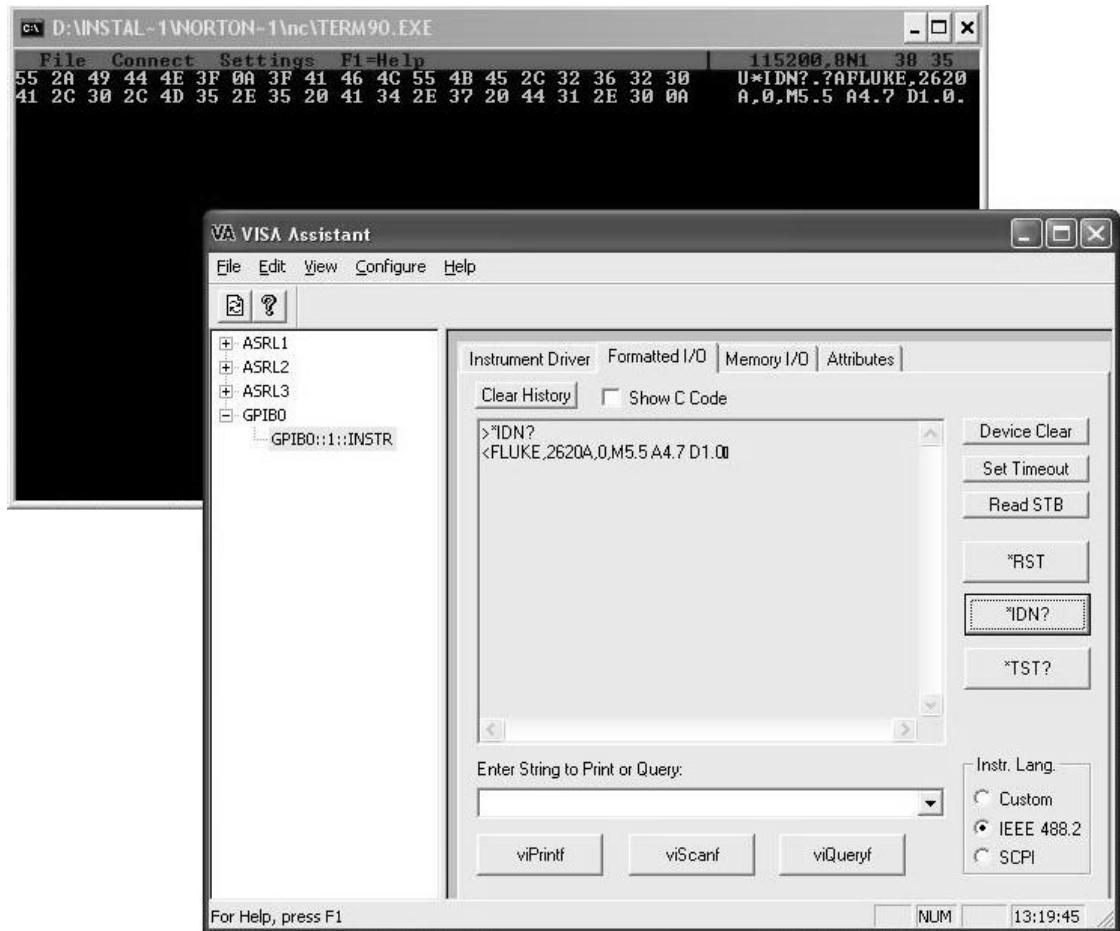
En esta fase se integraron los componentes HW y SW desarrollados en las fases anteriores; las tareas principales fueron:

- Verificar la correspondencia de los puertos de E/S tanto en HW como en SW.
- Comprobar el funcionamiento de los módulos Bluetooth con la pasarela GPIB/RS232 a velocidad de transferencia de datos de 115,200 Baudios.
- Construir un circuito de reinicio para la pasarela GPIB/RS232.
- Incluir un cristal de 20 MHz como reloj de operación.
- Verificar el correcto funcionamiento de los circuitos integrados que integran el CISBlue-488.

#### 4.1.5. Pruebas y liberación del producto

A continuación se enlistan las pruebas realizadas para verificar el correcto funcionamiento del CISBlue-488:

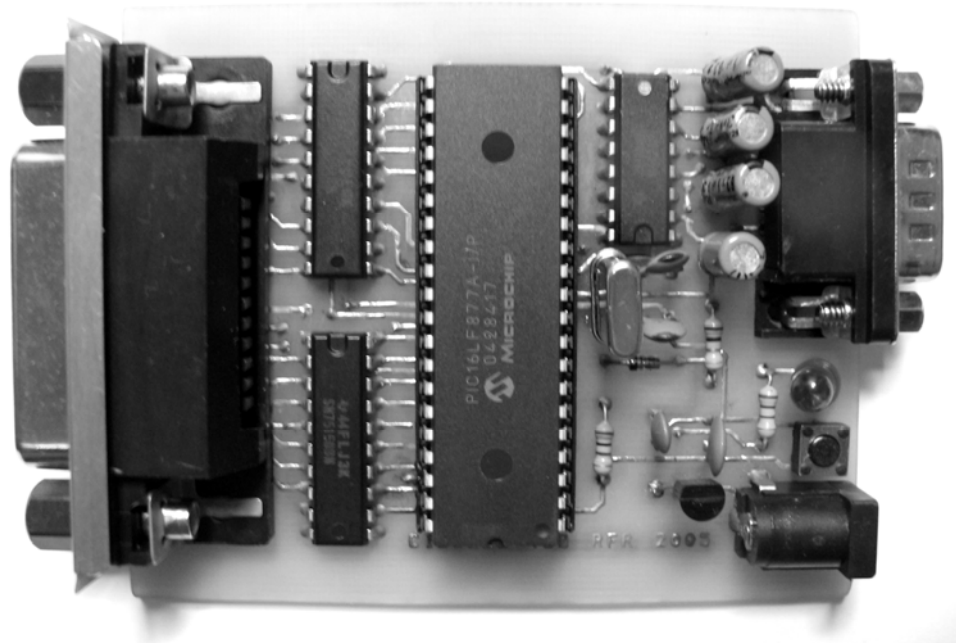
- La primera prueba realizada fue capturar las líneas de datos y de control del GPIB (transmitidas con ayuda de la herramienta VISA Assistant) mediante el manejo de la función de transferencia, dichas líneas se visualizaron en la herramienta TERM90 (Figura 4.6).
- Se comprobó el funcionamiento del sistema a diferentes velocidades de transferencia serie (4,800, 9,600, 19,200, 38,400, 57,600 y 115,200 Baudios).
- Identificación correcta de los datos y órdenes GPIB del instrumento receptor.
- Validación de los datos y órdenes del GPIB.
- Control de la transmisión y/o recepción serie (sincronización).
- Transmisión y recepción inalámbrica completa (controlador-receptor-controlador).



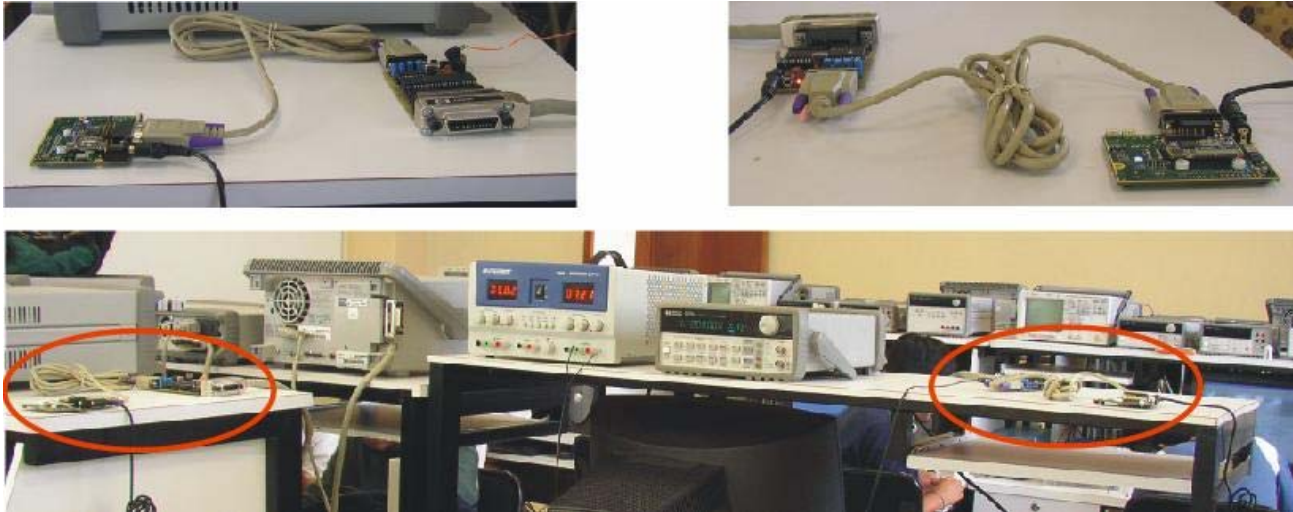
**Figura 4.6.** Lectura del bus GPIB.

Una vez realizadas las pruebas, se diseñó y elaboró el circuito impreso de la pasarela GPIB/RS 232 en el programa Orcad versión 9.0 [URL9]. La Figura 4.7 muestra el producto final desarrollado.

La Figura 4.8 muestra el CISBlue-488 funcionando.



**Figura 4.7.** Pasarela GPIB/RS232.



**Figura 4.8.** CISBlue-488 en funcionamiento.

#### 4.1.6. Mantenimiento y actualización

La Tabla 4.5 muestra las actualizaciones realizadas durante el ciclo de vida del CISBlue-488.

**Tabla 4.5.** Actualizaciones del CISBlue-488.

Versión	Actividad
0.1	Se leía el contenido del GPIB
0.2	Integración de los transceptores
0.3	Implementación de la función de transferencia GPIB
0.4	Se utilizó la velocidad de transferencia de 115,200 Baudios
0.5	Envío de paquetes vía serie mediante un cable serie
0.6	El instrumento receptor reconoce los datos recibidos
0.7	Captura de datos del instrumento
0.8	Sincronización de los datos
0.9	El controlador reconoce los datos del instrumento
1.0	Integración de los módulos Bluetooth
1.1	Validación de las señales de datos y de control GPIB
1.2	Función para el control de la señal SRQ
1.3	Elaboración del circuito impreso de la pasarela GPIB/RS232
1.4	Verificación final del CISBlue-488

## 5. Resultados

En este capítulo se describen los resultados obtenidos respecto al consumo de corriente del CISBlue-488, el manejo de la función de transferencia del GPIB, la inicialización del CISBlue-488, las pruebas de transmisión y alcance entre el sistema transmisor y el sistema receptor, así como el funcionamiento del CISBlue-488 en tiempo real.

### 5.1. Consumo de corriente del CISBlue-488

El CISBlue-488 está formado por cuatro componentes (**¡Error! No se encuentra el origen de la referencia.**) considerando que tanto en la parte transmisora como en la receptora se utilizan los mismos componentes, pero con diferente configuración.

La pasarela GPIB/RS232 se alimenta con 5.7 VCD y el módulo Bluetooth con 5 VCD. La Tabla 5.1 muestra el consumo aproximado de cada componente en modo reposo (*idle*) y en el momento en que interactúan con el GPIB, cabe destacar que dicho consumo incrementa notablemente.

**Tabla 5.1.** Consumo de corriente de los componentes del CISBlue-488.

Componente	Consumo de corriente	
	En modo reposo	Interactuando con el GPIB
Pasarela GPIB/RS232	TX	90 mA
	RX	125 mA
Módulo Bluetooth	TX	70 mA
	RX	70 mA

### 5.2. Manejo de la función de transferencia del GPIB

Como se analizó en el Párrafo 2.2.3.3.1, la función de transferencia del GPIB es necesaria para llevar a cabo la comunicación de manera adecuada y garantizar que los octetos enviados y recibidos estén libres de errores.

El instrumento receptor, en este caso la pasarela GPIB/RS232, espera a que el controlador del bus, la PC, active la línea DAV (1 lógico) para indicar que los datos existentes en el bus son válidos; posteriormente el instrumento receptor responde activando la línea NRFD (1 lógico) indicando que se encuentra listo para recibir datos; al recibir los datos el instrumento receptor activa la línea NDAC (0 lógico) para indicar que ha aceptado la información previamente recibida.

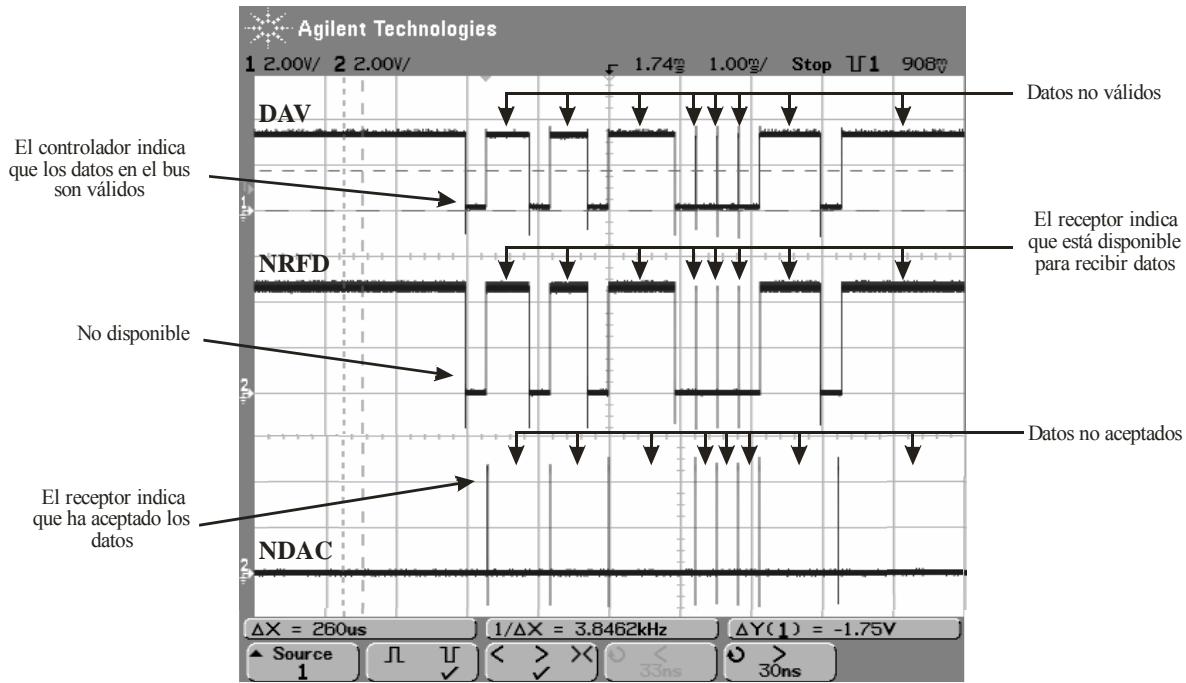


Figura 5.1. Función de transferencia realizada por el controlador y la pasarela GPIB/RS232.

Lo anterior se puede observar en la Figura 5.1, la cual muestra el proceso de envío y recepción de octetos entre el GPIB y la pasarela GPIB/RS232.

### 5.3. Inicialización del CISBlue-488

Para que el CISBlue-488 opere correctamente, es necesario seguir el diagrama de flujo que muestra la Figura 5.2, como se observa en dicha figura los tres primeros pasos se refieren a la conexión de los cables GPIB y seriales al sistema, así como del suministro de energía al sistema; en el siguiente paso es necesario reinicializar la pasarela que actuará como transmisor, lo cual es importante, de lo contrario el GPIB se bloqueará debido a que el sistema no responderá al controlador; el último paso es ejecutar el SW de aplicación (Agilent VEE, VISA Assistant, etc.).

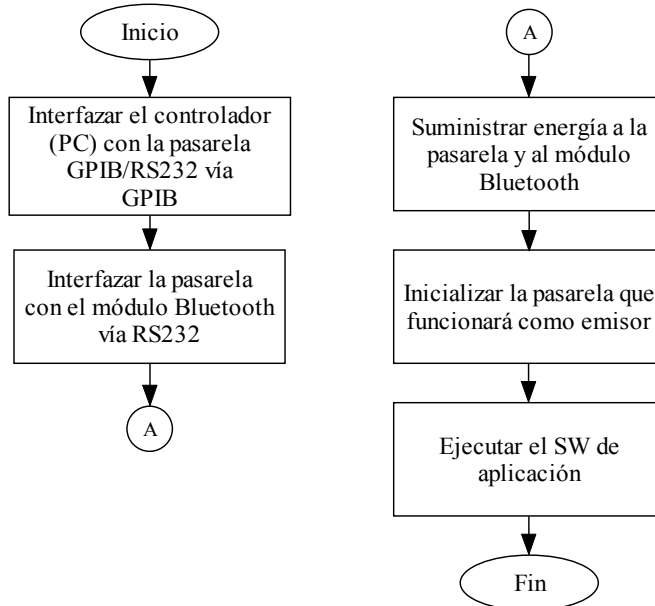


Figura 5.2. Inicialización del CISBlue-488.



## 5.4. Pruebas de transmisión y alcance entre los módulos Bluetooth

Las pruebas de transmisión se realizaron entre los módulos Bluetooth para determinar la distancia máxima de cobertura del sistema, así como conocer los errores de transmisión que pudieran tener los módulos al exponerlos a diferentes entornos de trabajo.

Antes de iniciar la transmisión, los dispositivos deben enlazarse. Para ello los módulos realizan sondeos con la finalidad de conocer qué dispositivos Bluetooth se encuentran en su esfera de proximidad. Al encontrar un dispositivo, los dispositivos inician con un proceso de autenticación y por último se establece la comunicación.

Durante las pruebas para cuantificar el número de errores de transmisión, se observó que siempre que el enlace entre los dispositivos Bluetooth esté establecido no se produce ningún error de comunicación. Sin embargo, una vez que se sobrepasa la distancia límite, la conexión se pierde. En estos casos no se reciben datos hasta que el enlace Bluetooth se reestablece nuevamente, lo cual ocurre cuando los dispositivos vuelven a realizar el proceso de sondeo. Una vez recuperada la transmisión, ésta queda libre de errores.

De esta forma se puede resumir que, en entornos donde no exista ruido electromagnético (EM) excesivo, la conexión está libre de errores siempre y cuando no se sobrepase la distancia máxima para la cual los dispositivos están enlazados.

En un entorno normal, con paredes u otros objetos entre los dispositivos enlazados, el alcance máximo es aproximadamente de 6 m; si la conexión se pierde, es necesario aproximar los dos módulos Bluetooth a una distancia mínima de 3 m. para asegurar de que éstos vuelvan a recuperar el enlace. A distancias mayores no está garantizado que se recupere el enlace, resultando en una transmisión espúrea que sólo transmite valores de vez en cuando. En un entorno libre de obstáculos la longitud máxima del enlace Bluetooth alcanza una distancia aproximada de 8 m.

## 5.5. Funcionamiento del CISBlue-488 en tiempo real

Para verificar el correcto funcionamiento del CISBlue-488, se llevaron a cabo las pruebas que se detallan a continuación.

En primer lugar se instaló y configuró el CISBlue-488 siguiendo el diagrama de flujo de la Figura 5.2.

La primera prueba consistió en el manejo de las líneas del GPIB, la cual se realizó al comparar la orden enviada por el controlador (PC) y las tramas serie que envía la pasarela GPIB/RS232 mediante la aplicación VISA Assistant y la herramienta Term90 (Figura 4.6).

La Tabla 5.2 muestra los datos enviados por el controlador del bus (PC), en este caso envía una orden \*IDN? al instrumento receptor; el controlador cambia a modo receptor mediante el envío del carácter LF (Anexo B).

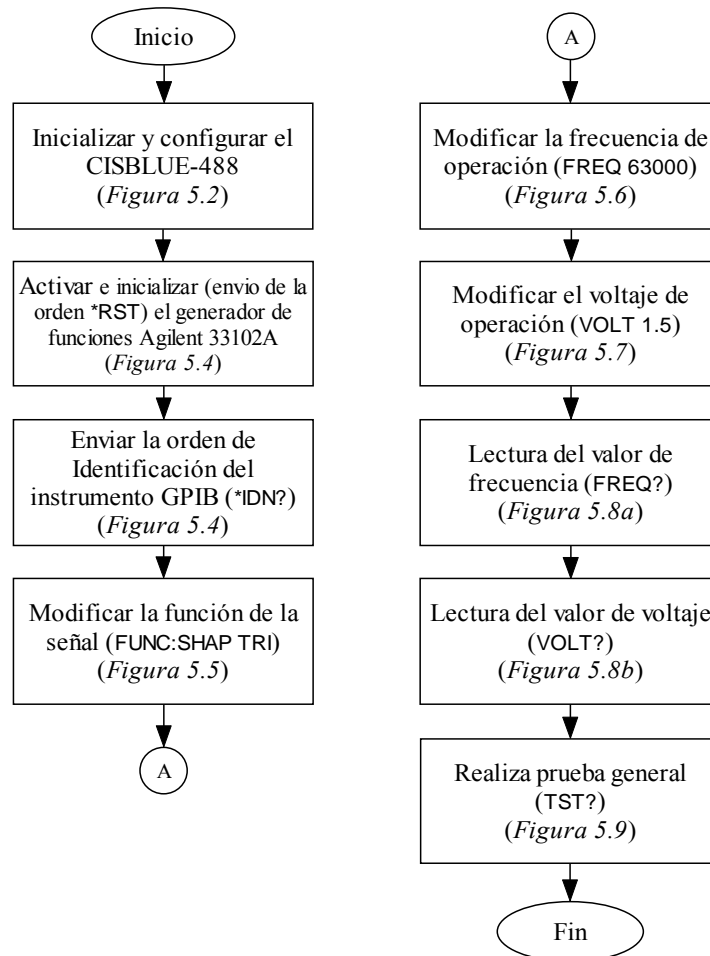
La Tabla 5.3 muestra las líneas de datos y de control que regresa el instrumento receptor; una vez que recibe correctamente la orden \*IDN?, cambia su función a modo transmisor como respuesta de haber recibido el carácter LF. El instrumento también envía el carácter LF al controlador indicándole que es el último dato que enviará, y ambos, controlador y receptor, vuelven a su estado normal.

Tabla 5.2. Mensaje \*IDN? enviado por el controlador del GPIB.

Líneas de control								Líneas de datos								Dato			
REN	IFC	NDAC	NRFD	DAV	EOI	ATN	SRQ	HEX	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	HEX	ASCII	Mensaje
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	0	0	44	0	1	0	1	0	1	0	1	55	U	MTA21
0	1	1	0	0	1	0	0	64											
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	0	0	44	0	0	1	1	1	1	1	1	3F	?	UNL
0	1	1	0	0	1	0	0	64											
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	0	0	44	0	0	1	0	0	0	0	1	21	!	MLA1
0	1	1	0	0	1	0	0	64											
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	1	0	46	0	0	1	0	1	0	1	0	2A	*	MLA10
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	0	1	0	0	1	49	I	MTA9
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	0	0	1	0	0	44	D	MTA7
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	0	1	1	1	0	4E	N	MTA14
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	0	1	1	1	1	1	1	3F	?	UNL
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	0	1	0	59											
0	1	0	0	0	0	1	0	82	0	0	0	0	1	0	1	0	0A	LF	-

**Tabla 5.3.** Respuesta de identificación del instrumento receptor.

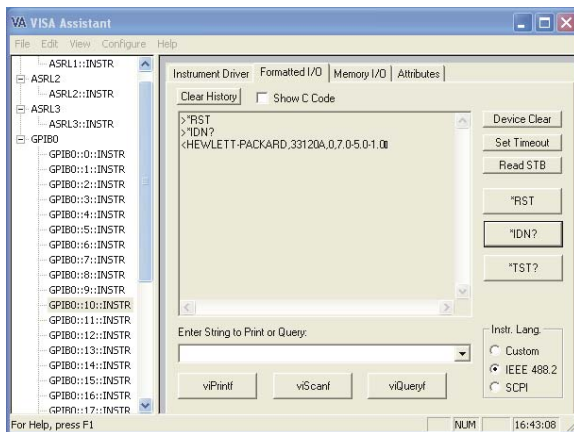
Líneas de control								Líneas de datos								Dato			
REN	IFC	NDAC	NRFD	DAV	EOI	ATN	SRQ	HEX	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	HEX	ASCII	Mensaje
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	0	0	44	0	0	1	1	1	1	1	1	3F	?	UNL
0	1	1	0	0	1	0	0	64											
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	0	0	44	0	0	1	1	0	1	0	1	35	5	PPC
0	1	1	0	0	1	0	0	64											
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	0	0	44	0	1	0	0	0	0	0	1	41	A	MTA1
0	1	1	0	0	1	0	0	64											
0	1	0	1	1	1	0	0	5C											
0	1	0	0	0	1	1	0	46	0	1	0	0	0	1	1	0	46	F	MTA6
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	0	1	1	0	0	4C	L	MTA12
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	1	0	1	0	1	55	U	MTA21
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	0	1	0	1	1	4B	K	MTA11
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	1	1	0	5E											
0	1	0	0	0	1	1	0	46	0	1	0	0	0	1	0	1	45	E	MTA5
0	1	1	0	0	1	1	0	66											
0	1	0	1	1	0	1	0	5A											
0	1	0	0	0	0	1	0	42	0	0	0	0	1	0	1	0	0A	LF	-



**Figura 5.3.** Diagrama de flujo para comprobar y evaluar el funcionamiento del CISBlue-488.

La segunda prueba consistió en comprobar y evaluar el correcto funcionamiento del CISBlue-488 mediante la aplicación VISA Assistant. Para llevar a cabo lo anterior se utilizó el generador de funciones Agilent 33120A y con ayuda de la aplicación VISA Assistant se enviaron órdenes a dicho instrumento siguiendo el diagrama de flujo de la Figura 5.3.

Las Figuras 5.4-5.9 muestran las modificaciones que sufre el instrumento al recibir las órdenes del controlador GPIB.



**Figura 5.4.** Estado inicial del instrumento Agilent 33120A.

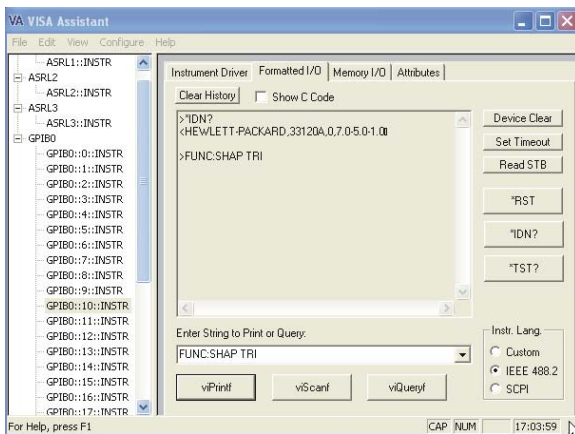


Figura 5.5. Modifica la función a señal triangular.

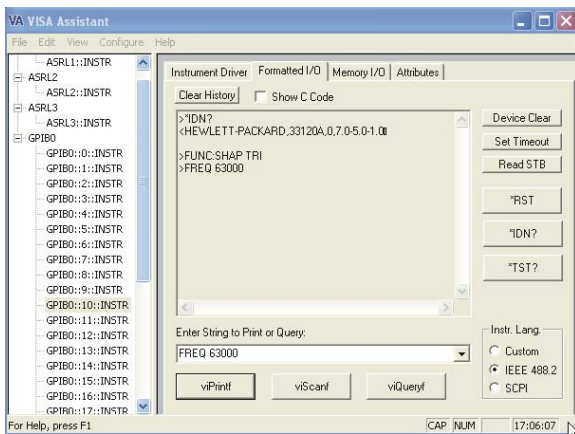
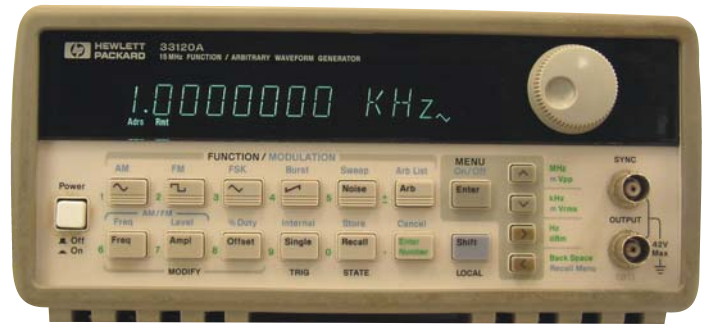


Figura 5.6. Modifica la frecuencia de operación a 63 KHz.

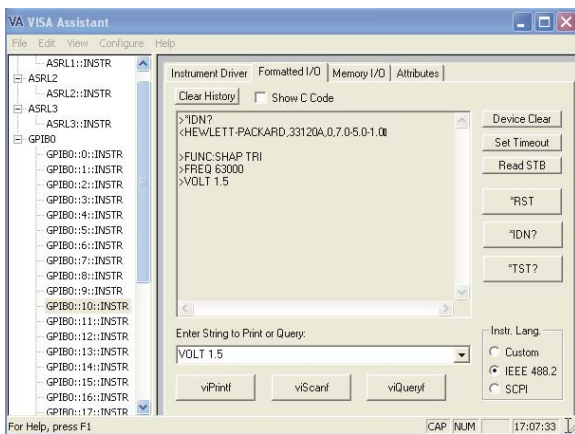


Figura 5.7. Modifica el voltaje de operación a 1.5 Vpp.



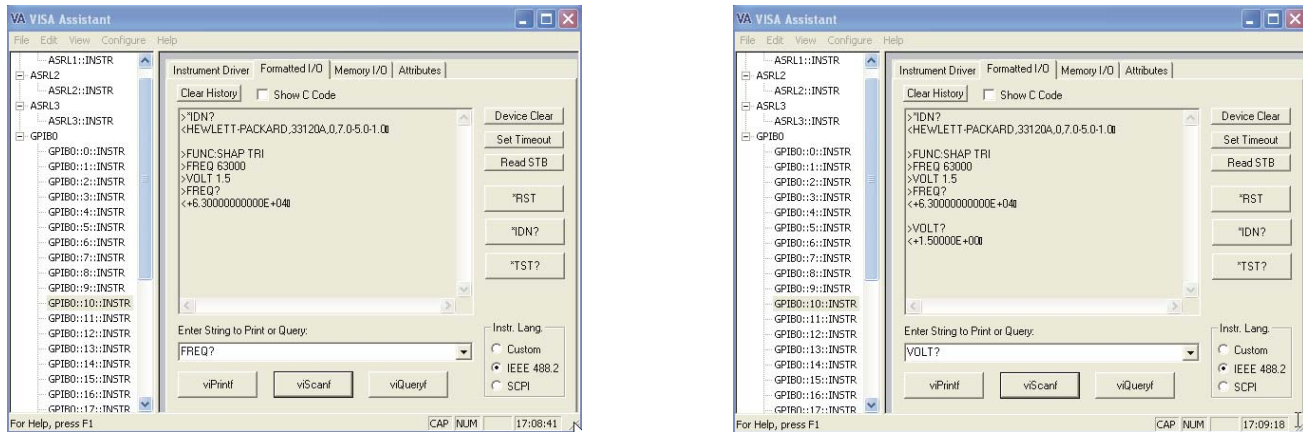


Figura 5.8. Lee los valores de frecuencia y de voltaje.

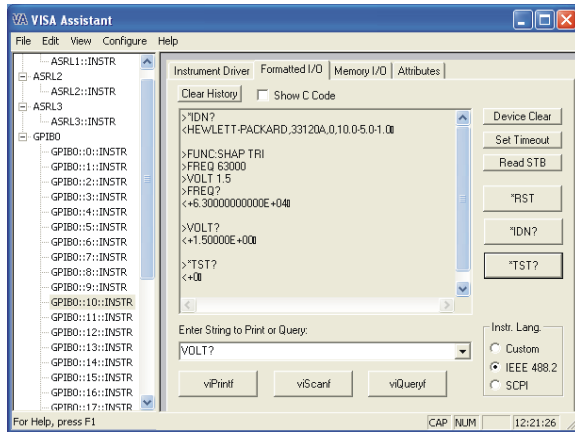


Figura 5.9. Realiza prueba general al instrumento.

## 5.6. Velocidad de transferencia de datos

El CISBlue-488 está compuesto principalmente de dos sistemas, la pasarela GPIB/RS232 y el módulo Bluetooth. La pasarela convierte datos GPIB a tramas RS232. Las tramas RS232 son enviadas al módulo Bluetooth a través de su puerto serie a una velocidad de 115,200 Baudios. El protocolo GPIB maneja una velocidad de transferencia de 1 MBps, la pasarela GPIB/RS232 emplea una velocidad de 115,200 Baudios, y el módulo Bluetooth tiene una velocidad de transferencia de 1 Mbps (Figura 5.10).

Por lo anterior, la velocidad de transferencia de datos del CISBlue-488 es de 115.2 Kbps debido a que es la velocidad a la cual trabaja la pasarela GPIB/RS232, siendo 115,200 Baudios la velocidad más alta que acepta el módulo Bluetooth en su puerto serie.

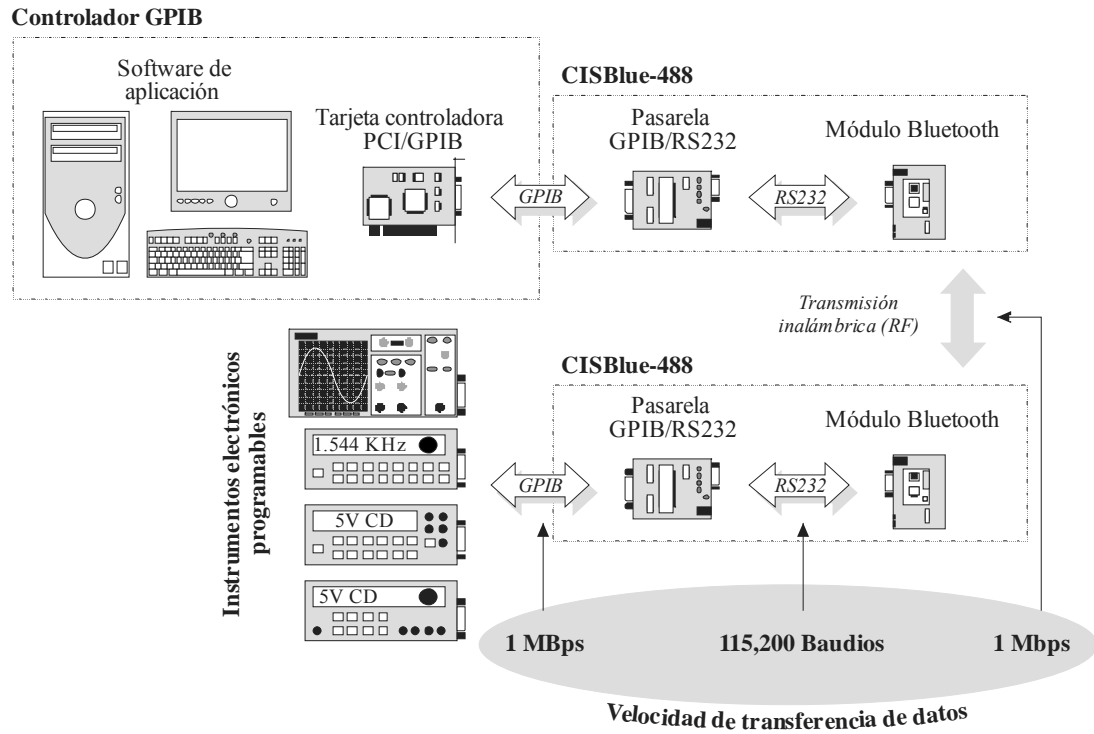


Figura 5.10. Velocidades de transferencia del CISBlue-488.





## 6. Conclusiones

Durante el desarrollo del presente trabajo de tesis se llevaron a cabo las siguientes actividades:

- Estudio de los protocolos de comunicaciones IEEE 488 y Bluetooth, con base en una investigación bibliográfica.
- Considerando la metodología de desarrollo de sistemas empotrados (Subcapítulo 1.1), se implementó el CISBlue-488 basado en la arquitectura del MCU PIC16F877.
- Las pruebas del sistema se llevaron a cabo en el LabCD de la UTM, considerando que una de las finalidades del sistema es ampliar el rango de utilización de los sistemas y dispositivos de instrumentación electrónica con los que cuenta dicho laboratorio.

Se pueden listar las siguientes aportaciones del trabajo realizado:

- Como complemento al estudio del protocolo GPIB, se ha desarrollado un programa en lenguaje ensamblador que permite capturar las tramas en el bus en tiempo real con ayuda de la herramienta TERM90.
- Se ha implementado satisfactoriamente una picored inalámbrica Bluetooth entre los instrumentos electrónicos programables y el controlador (PC) mediante el CISBlue-488. Dicho sistema ayudará a eliminar la gran cantidad de cables GPIB utilizados y por consiguiente se reducirá considerablemente el consumo de corriente.
- El CISBlue-488 puede ser utilizado en el LabCD por alumnos y profesores que necesiten emplearlo para sus labores académicas y de investigación.
- Se elaboró un tutorial (Anexo A) para utilizar los módulos Bluetooth, el cual describe los procesos de instalación, configuración y uso. Estos módulos pueden ser utilizados para interconectar dispositivos de manera inalámbrica que cumplan con los requisitos correspondientes.

En la presente tesis se conjugan protocolos diferentes pero relacionados entre sí. Por un lado se encuentra el GPIB (IEEE 488) que permite la interconexión de instrumentos electrónicos de medida y estimulación. Por otro lado se encuentra el protocolo de comunicaciones inalámbricas Bluetooth, cuya tecnología permite la transmisión de voz y datos entre diferentes equipos mediante un enlace de RF en distancias cortas a bajo consumo de potencia. Por último se encuentra el protocolo de comunicaciones serie RS232, el cual se utiliza como enlace entre el protocolo GPIB y la especificación de Bluetooth. Asimismo, se realizó un estudio de la metodología de desarrollo de sistemas empotrados, el cual fue fundamental durante el ciclo de vida del CISBlue-488.

Con base en lo anterior se puede concluir que:

- La tecnología Bluetooth ha comprobado su eficacia al conectar dispositivos en forma inalámbrica, a cortas distancias, que exijan una confiabilidad en el enlace, bajo consumo de potencia e integridad de los datos.
- Los módulos Bluetooth utilizados tienen un alcance limitado a 10 m, pero la especificación de Bluetooth define un alcance máximo de 100 m, lo cual es posible mediante la utilización de módulos con mayor potencia de transmisión. Sin embargo, puede presentar inconvenientes como mayor emisión de radiación y por consiguiente mayor consumo de corriente.
- El empleo de la metodología de desarrollo de sistemas empotrados durante el diseño e implementación del CISBlue-488 ayudó a la obtención de un sistema robusto que cumple con las especificaciones iniciales del producto.
- La velocidad de transferencia máxima del sistema CISBlue-488 es de 115.2 Kbps, debido a la velocidad de operación de los módulos Bluetooth, los cuales tienen un rango de velocidad de transmisión de 300 a 115,200 Baudios.

Actualmente, existe una clara tendencia al uso de redes inalámbricas para la interconexión de una gran variedad de sistemas, principalmente por dos motivos:

- Al bajo costo que representa la infraestructura de las redes inalámbricas.
- La posibilidad y comodidad de usar los sistemas desde cualquier localización, eliminando las restricciones que representa el cableado.

Las herramientas utilizadas en este trabajo se conjugaron a favor de la aplicación, cumpliendo satisfactoriamente con los objetivos propuestos e introduciendo una tecnología que puede emplearse para futuros proyectos de las carreras de Ingeniería en Electrónica, Ingeniería en Computación, Ingeniería Industrial e Ingeniería en Mecatrónica.

Con base en la experiencia obtenida, se proponen los siguientes trabajos futuros:

- Desarrollo de un sistema de monitoreo de variables eléctricas y mecánicas.
- Diseño de subsistemas de comunicación inalámbrica para los dispositivos de instrumentación electrónica programable existentes en el LabCD.
- Conexión de diversos RTUs (*Real Time Unit*) de forma inalámbrica, por ejemplo los PLCs (*Programmable Logic Controller*) de la firma Siemens con que cuenta la UTM.

## Bibliografía

- [1] Ball S.: Embedded Microprocessor Systems, Real World Design, Newnes, 1998.
- [2] Berger A.: Embedded Systems Design. An Introduction to Processes, Tools, and Techniques, CMP Books, 2002.
- [3] Caristy, A.: IEEE 488 General Purpose Instrumentation Bus Manual, Academic Press Inc., EUA, 1989.
- [4] Chamú, C.: Desarrollo de un Sistema Educativo para la Enseñanza del Protocolo de Comunicaciones CAN, Tesis de Licenciatura, UTM, Abril 2005.
- [5] ConnectBlue: OEM Serial Port Adapter TM 2<sup>nd</sup> Generation. Electrical & Mechanical Data-sheet, Data Sheet cBProduct-0212-01, connectBlue AB, 2003.
- [6] ConnectBlue: Serial Port Adapter TM 2<sup>nd</sup> Generation. User Manual 2.11, connectBlue AB, 2003.
- [7] García, J.: Administración de los Recursos de un Laboratorio de Comunicaciones Digitales mediante su Instalación, Configuración y Puesta en Funcionamiento, Tesis de Licenciatura, UTM, Junio 2004.
- [8] González, A.: Estudio del protocolo IEEE 488 mediante el desarrollo de una herramienta de simulación, Tesis de Licenciatura, UTM, Abril 2003.
- [9] Maxim: Max220-Max249. +5-Powered, Multichannel RS-232 Drivers/Receivers, Data Sheet 19-4323, Rev 7tx; 11/97, Maxim Integrated Products, 1997.
- [10] Microchip: PIC16F877. 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, Data Sheet DS30292C, Microcip Technology Inc., 2001.
- [11] Miller B. & Bisdikian C.: Bluetooth Revealed. The Insider's Guide to an Open Specification for Global Wireless Communications, Prentice Hall, 2001.
- [12] Muller N.: Bluetooth Demystified, McGraw-Hill, 2001.
- [13] National Instruments: GPIB Tutorial, Instrumentation Newsletter, National Instrument Corporation, 2002.
- [14] Texas Instruments: SN75160B. Octal General-Purpose Interface Bus Transceiver, Data Sheet SLLS004B, Texas Instruments Incorporated, 1995.
- [15] Texas Instruments: SN75161B, SN75162B. Octal General-Purpose Interface Bus Transceivers, Data Sheet SLLS005B, Texas Instruments Incorporated, 1995.

## Sitios URL

- [URL1] <http://www.agilent.com> “Página electrónica de la empresa Agilent Technologies Incorporated”, 2004.
- [URL2] <http://www.connectblue.se> “Página electrónica de la empresa connectBlue AB”, 2004.
- [URL3] <http://www.bluetooth.com> “Página electrónica de la tecnología inalámbrica Bluetooth”, 2004.
- [URL4] <http://www.bluetooth.org> “Página electrónica del grupo de interés especial (SIG) Bluetooth”, 2004.
- [URL5] <http://www.gigaant.com> “Página electrónica de la empresa gigaAnt”, 2004.
- [URL6] <http://www.fluke.com> “Página electrónica de la empresa Fluke Corporation”, 2004.
- [URL7] <http://www.microchip.com> “Página electrónica de la empresa Microchip Technology Incorporated”, 2004.
- [URL8] <http://www.ni.com> “Página electrónica de la empresa National Instrument Corporation”, 2004.
- [URL9] <http://www.orcad.com> “Página electrónica de la empresa Cadence Design Systems, Inc.”, 2004.
- [URL10] <http://www.scpiconsortium.org> “Página electrónica del consorcio SCPI”, 2004.
- [URL11] <http://www.symantec.com> “Página electrónica de la empresa Symantec Corporation”, 2004.
- [URL12] <http://www.ti.com> “Página electrónica de la empresa Texas Instruments Incorporated”, 2004.

## Anexo A

En este Anexo se describen las características eléctricas y mecánicas del módulo adaptador II OEM y del adaptador de puerto serie OEM 13i (Figura A.1), los cuales conforman el módulo Bluetooth que se utilizó para transmitir y recibir datos de manera inalámbrica en el presente trabajo de tesis.

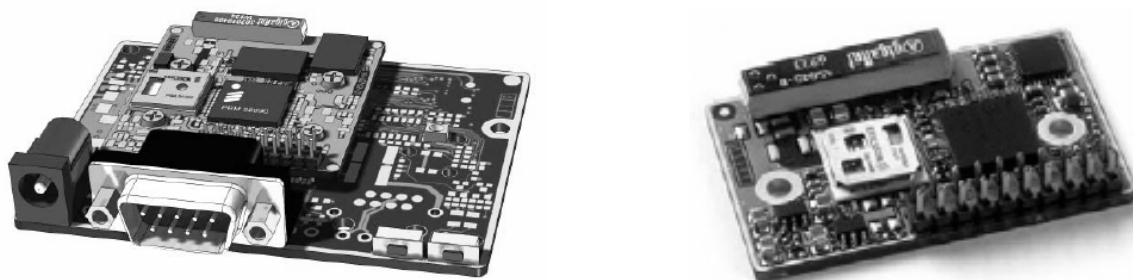


Figura A.1. Módulo adaptador II OEM y adaptador de puerto serie OEM.

### A.1. Adaptador de puerto serie OEM 13i

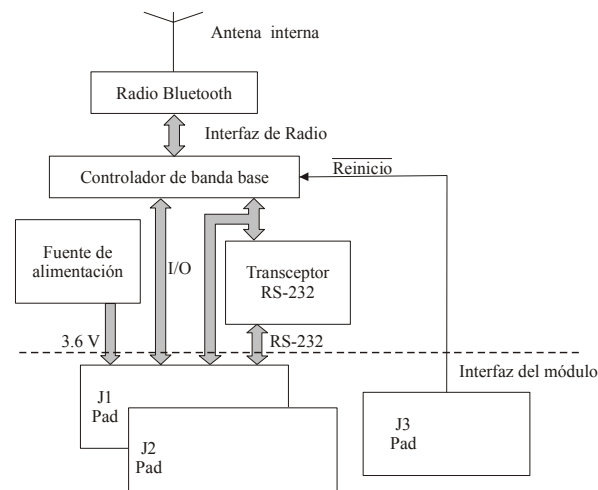
El adaptador de puerto serie OEM 13i se puede utilizar para comunicar productos Bluetooth tales como computadoras portátiles (*Laptop*), PDA, teléfonos móviles, dispositivos industriales, dispositivos en automóviles, etc. mediante puerto serie. La Tabla A.1 lista sus principales características.

Tabla A.1. Principales características del adaptador puerto serie.

Adaptador de puerto serie OEM 13i con antena integrada	
Modelo	OEMSPA13i
Módulo ID OEM	cB-0701-01
Tipo	Clase 2 / 0 dBm
Alimentación	3.0 – 6 VDC
Nivel lógico	UART y RS232

La Figura A.2 muestra el diagrama a bloques del módulo ID cB-0701-01, el cual está basado en el controlador de banda base PM 990 80 de la firma Infineon<sup>8</sup> y en el radio PBA 313 05 (0 dBm). El controlador PBM 990 80 cuenta con memoria interna SRAM y Flash.

<sup>8</sup> anteriormente Ericsson Microelectronics.



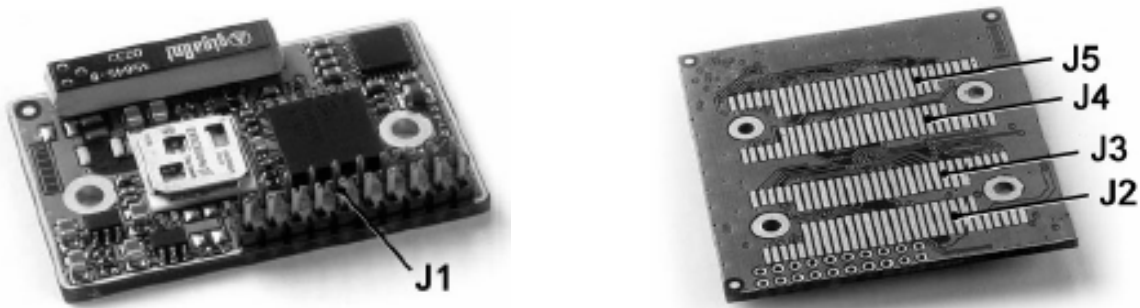
**Figura A.2.** Diagrama a bloques del módulo cB-0701-01.

### A.1.1. Interfaz eléctrica y conectores

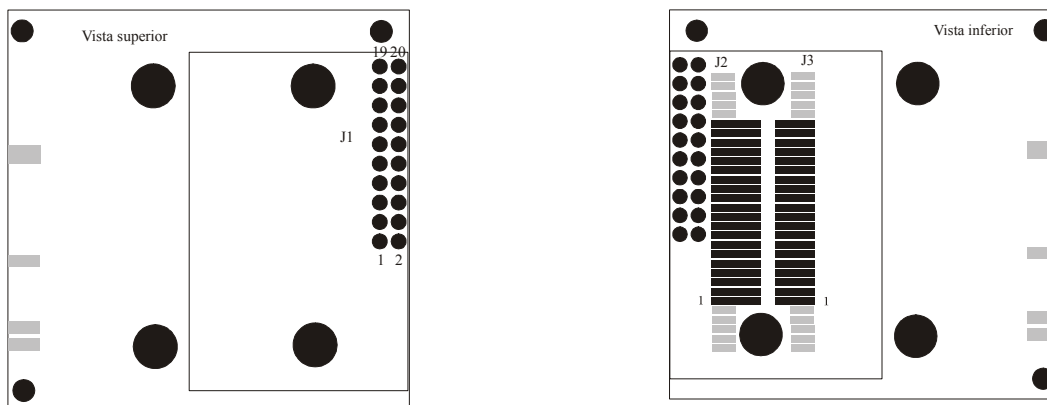
El módulo cB-0701-01 presenta dos formas de conexión (Figura A.3)<sup>9</sup>:

- Mediante el conector J1.
- Mediante las terminales de montaje superficial J2 y J3.

La Figura A.4 muestra la asignación de los terminales para dichas conexiones. Las Tabla A.2 y A.3. muestran la asignación de las terminales de los conectores J1, J2 y J3 del adaptador de puerto serie (la terminal de reinicio es la única terminal disponible en el conector J3).



**Figura A.3.** Conectores J1, J2 y J3 del adaptador.



**Figura A.4.** Asignación de terminales para el conector J1, J2 y J3.

<sup>9</sup> El módulo Bluetooth utilizado no cuenta con los conectores J4 y J5, a los que hace referencia la Figura A.3.

**Tabla A.2.** Asignación de terminales para los conectores J1 y J2.

Terminal	Nombre	Señal	Nivel de señal	Tipo	Descripción
1,2	VSS	Tierra física ( <i>Ground</i> )	0 V		Terminales conectados internamente
3,4	VCC_5V	Señal de suministro de energía ( <i>Power</i> )	3.3V-6V		Terminales conectados internamente
5	RS232-CTS	Listo para enviar ( <i>Clear To Send</i> )	RS232	Entrada	Control de flujo
6	RS232-TxD	Transmisión de datos ( <i>Transmit Data</i> )	RS232	Salida	
7	RS232-RTS	Petición de envío ( <i>Request To Send</i> )	RS232	Salida	Control de flujo
8	RS232-RxD	Recepción de datos ( <i>Receive Data</i> )	RS232	Entrada	
9	RS232-DTR	Terminal de datos lista ( <i>Data Terminal Ready</i> )	RS232	Salida	
10	RS232-DSR	Conjunto de datos listo ( <i>Data Set Ready</i> )	RS232	Entrada	Listo para recibir o transmitir datos
11	RED/Mode	LED en color rojo (salida) y entrada en modo interfaz serie	CMOS	E/S	Señal multiplexada: <i>RED</i> : no válida hasta 500 ms después de encendido; <i>Mode</i> : durante el encendido selecciona un nivel lógico RS232 (alto) o UART (bajo).
12	Switch-0	Interruptor de función	CMOS	Entrada	Usado por la función “Conexión de señal externa” [5]
13	GREEN/Switch-1	LED en color verde (salida) e interruptor de realmacenamiento ( <i>Restore</i> )	CMOS	E/S	Señal multiplexada: <i>GREEN</i> : no válida hasta 500 ms después de encendido; <i>Switch-1</i> : si el nivel es bajo, las unidades regresan a un estado serie preestablecido
14	BLUE	LED en color azul (salida)	CMOS	Salida	Ver Tabla A.7
15	UART1-CTS	Listo para enviar ( <i>Clear To Send</i> )	CMOS	Entrada	Control de flujo
16	UART1-TxD	Transmisión de datos ( <i>Transmit Data</i> )	CMOS	Salida	
17	UART1-RTS	Petición de envío ( <i>Request To Send</i> )	CMOS	Salida	Control de flujo
18	UART1-RxD	Recepción de datos ( <i>Receive Data</i> )	CMOS	Entrada	
19	UART1-DTR	Terminal de datos lista ( <i>Data Terminal Ready</i> )	CMOS	Salida	
20	UART1-DSR	Conjunto de datos listo ( <i>Data Set Ready</i> )	CMOS	Entrada	Listo para recibir o transmitir datos

**Tabla A.3.** Asignación de terminales para el conector J3.

Pin	Nombre	Señal	Nivel de señal	Tipo	Descripción
1-19, 20					Reservados (no conectar)
19	RESET	Reinicialización del sistema	CMOS	Entrada	Se activa en bajo

El módulo ID cB-0701-01 tiene una fuente de alimentación lineal, la cual hace que la corriente sea constante si el voltaje de alimentación varía. El consumo de corriente depende del estado de la operación, la velocidad de transferencia de datos y la carga en las terminales de la interfaz RS232 (Tabla A.4). La Tabla A.5 muestra las señales de entradas y salidas, y la Tabla A.6 las características térmicas del módulo ID cB-0701-01, respectivamente.

**Tabla A.4.** Consumo de corriente del módulo ID cB-0701-01.

Símbolo	Modo		Valor	Unidades		
Vel. de transferencia <sup>1</sup>			≤57.6 >57.6	Kbps		
$I_{cc}$ <sup>2</sup>	No conectada	Promedio	25	60	mA	
		Pico	85	120	mA	
	No conectada, habilitada en modo paro	No conectada	Promedio	0.6	0.7	mA
		Conectada	Promedio	1.1	1.5	mA
	Conectada	Detenido ( <i>Idle</i> )	Promedio	35	70	mA
			Pico	100	150	mA
		Transmitiendo	Promedio	50	85	mA
			Pico	100	150	mA
	$I_{RS232}$ Salida <sup>3</sup>	RS232	Promedio	7	7	mA
	$R_{RS232}$ Carga <sup>4</sup>		Típico	20	20	kΩ

<sup>1</sup> La frecuencia del reloj interno se incrementa cuando la velocidad de transferencia de datos es mayor a 57.6 Kbps.

<sup>2</sup> VCC\_5V = 5 VCD.

<sup>3</sup> Las señales de salida RS232 TxD, RTS y DTR con una carga de 3 KΩ, no incluido en  $I_{cc}$ , dependiente de la carga.

<sup>4</sup> Carga interna en cada terminal de entrada RS232.

**Tabla A.5.** Señales de entradas y salidas del módulo ID cB-0701-01.

Símbolo	Parámetro		Valor	Unidades
$V_{entrada}$ (bajo)	Nivel lógico bajo del voltaje de entrada	Min	-0.3	V
		Max	0.8	V
$V_{entrada}$ (alto)	Nivel lógico alto del voltaje de entrada	Min	2.05	V
		Max	3.2	V
$V_{salida}$ (bajo)	Nivel lógico bajo del voltaje de salida	Max	0.1	V
$V_{salida}$ (alto)	Nivel lógico alto del voltaje de salida	Min	2.8	V
$I_{GPIO}$	Corriente continua	Max	4	mA
$C_{GPIO}$ <sup>1</sup>	Capacitancia de entrada	Típica	30	pF
$C_{reinicio}$	Capacitancia de entrada de reinicio	Típica	20	nF
$V_{T\_OUT\_RS232}$	Voltaje de salida del transmisor RS232	Típica	± 6	V
$V_{R\_IN\_RS232}$	Voltaje de entrada del receptor RS232, nivel alto (bajo lógico)	Min	3	V
		Típico	6	V
		Max	25	V
$V_{R\_IN\_RS232}$	Voltaje de entrada del receptor RS232, nivel bajo (alto lógico)	Max	-3	V
		Típico	-6	V
		Min	-25	V

<sup>1</sup> Entrada/Salida de propósito general (GPIO, *General Purpose Input/Output*)



**Tabla A.6.** Características térmicas del módulo ID cB-0701-01.

Parámetro	Valor	Unidades
Temperatura de almacenamiento ( <i>Storage</i> )	Min	-40 °C
	Max	+85 °C
Temperatura de operación recomendada	Min	-20 °C
	Max	+75 °C
Temperatura de operación máxima	Min	-30 °C
	Max	+75 °C

### A.1.2. Modos de operación

El módulo ID cB-0701-01 cuenta con diferentes modos de operación (Tabla A.7), los cuales se indican mediante el color del LED indicador. Las señales del LED se activan en nivel bajo.

**Tabla A.7.** Modos de operación del módulo ID cB-0701-01.

Modo	Estado	LED RGB color	GPIO verde	GPIO Azul	GPIO Rojo
Datos	Detenido	Verde	Bajo	Alto	Alto
AT, ECI	Detenido	Naranja	Bajo	Alto	Bajo
Datos, AT, ECI	Conectando	Violeta	Alto	Bajo	Bajo
Datos, AT, ECI	Conectado	Azul	Alto	Bajo	Alto

El módulo ID cB-0701-01 cuenta con una antena interna de montaje superficial (SMD, *Internal Surface Mounted*) cuyas principales características se muestran en la Tabla A.8.

**Tabla A.8.** Características de la antena interna del módulo ID cB-0701-01.

Parámetro	Información
Nombre	Mica 2.4 GHz
Fabricante	gigaAnt [URL5]
Tipo	Antena parche ( <i>patch</i> )
Polarización	Lineal
Ganancia	+ 2.7 dBi
Dimensiones (LxAnxAI)	20.5 x 3.6 x 3.35 mm
Conector	Montaje superficial

Por último, la Tabla A.9 muestra las propiedades de Bluetooth que proporciona el módulo ID cB-0701-01.

**Tabla A.9.** Información de Bluetooth.

Parámetro	Información
Radio Bluetooth	Infineon/Ericsson PBD 313 05 (RAN)
Controlador de banda base Bluetooth	Infineon/Ericsson PBM 990 80 (FLINK)
Potencia de salida RF	Clase 2, 1.5dBm (Típica)
Nivel de sensibilidad de recepción	-80dBm (0.1% BER)
Nivel de entrada en la recepción (max)	+14 dBm
Frecuencia de salida	2.402 – 2.480 GHz, Banda ISM
Operación punto a multipunto	Sí
Pila Bluetooth	Pila huésped empotrada
Versión de la especificación de Bluetooth	1.1

## A.2. Módulo adaptador II OEM

El módulo adaptador II OEM es una herramienta de desarrollo diseñada para interfazar el adaptador de puerto serie OEM de la firma connectBlue.

La interfaz eléctrica de los productos OEM consiste de un área de 60 pistas (*pads*) (2x30 de 1 mm) y de un área de terminales de 2x10 de 2 mm. para facilitar una rápida instalación y uso.

El módulo adaptador consiste de:

- Un conector macho DB9 como interfaz RS232.
- Un conector de alimentación de 2.1 mm. de 5 VDC.
- Un interruptor para reestablecer la configuración serie.
- Un interruptor para conexión Bluetooth
- Un LED RGB para indicar el estado del módulo.

La Tabla A.10 muestra la asignación de terminales del conector DB9 del módulo adaptador II OEM.

**Tabla A.10.** Asignación de terminales del conector DB9.

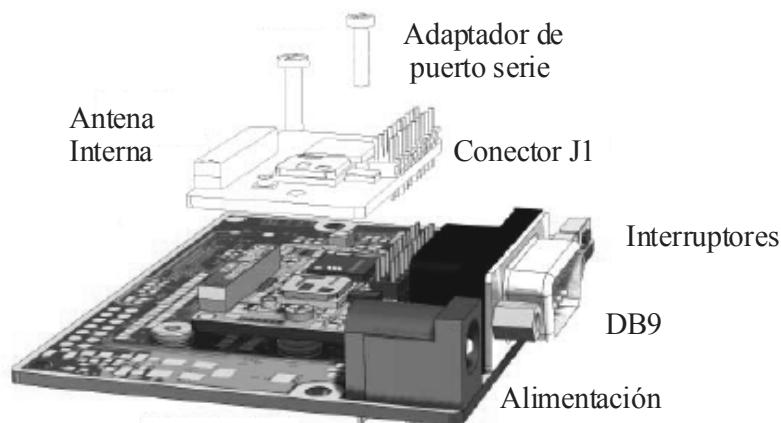
Terminal	Nombre	Señal
1	NC	No conectado ( <i>Not Conennected</i> )
2	RD	Recepción de datos ( <i>Receive Data</i> )
3	TD	Transmisión de datos ( <i>Transmit Data</i> )
4	DTR	Terminal de datos lista ( <i>Data Terminal Ready</i> )
5	GND	Tierra física ( <i>Ground</i> )
6	DSR	Conjunto de datos listo ( <i>Data Set Ready</i> )
7	RTS	Petición de envío ( <i>Request To Send</i> )
8	CTS	Listo para enviar ( <i>Clear To Send</i> )
9	NC	No conectado ( <i>Not Conennected</i> )

### A.2.1. Ensamble del módulo Bluetooth

Para utilizar el módulo Bluetooth es necesario ensamblar el módulo adaptador de puerto serie OEM al módulo adaptador II OEM mediante los conectores J2 y J3 (Figura A.5).

### A.3. Instalación y funcionamiento del módulo Bluetooth

El módulo Bluetooth se comunica con su sistema huésped (*host*) mediante RS232 o nivel lógico tipo UART. Una vez conectado a su huésped, se configura y se habilita la comunicación Bluetooth.



**Figura A.5.** Ensamble de los módulos.

Cuando el módulo Bluetooth inicia una conexión a otro dispositivo, actúa como Maestro, mientras que si acepta una conexión de otro dispositivo, actúa como Esclavo. En la mayoría de los casos se comunica con un solo dispositivo Bluetooth, pero es posible que tenga comunicación con varios dispositivos Bluetooth simultáneamente (Subcapítulo 3.6).

A continuación se describen los procesos de instalación y configuración del módulo Bluetooth.

### A.3.1. Instalación

El primer paso que se realiza es conectar el módulo Bluetooth al sistema huésped mediante un cable serie; posteriormente se conecta a la red eléctrica para suministrarle energía (Figura A.6).

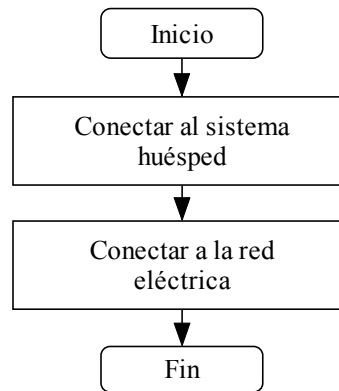


Figura A.6. Instalación del módulo Bluetooth.

### A.3.2. Configuración

Antes de usar el módulo es necesario configurarlo, para ello existen tres métodos:

- Mediante el asistente de configuración basado en una PC.
- Mediante órdenes AT.
- Mediante el protocolo ECI y el sistema de desarrollador Bluetooth.

En este caso se utiliza el asistente de configuración basado en una PC, los otros métodos se pueden consultar en la página WEB del fabricante [URL2].

#### A.3.2.1. Configuración mediante el asistente de configuración

El primer paso que se debe realizar es instalar el SW contenido en el CD nombrado “*Serial Port Adapter*”; una vez instalado, se ejecuta el SW para configurar el puerto serie COM correspondiente (Figura A.7).



Figura A.7. Configuración del puerto serie COM.

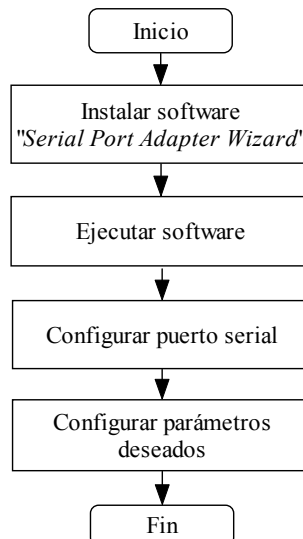
Una vez configurado el puerto serie, se configuran las siguientes características (Figura A.8):

- *Modo de seguridad Bluetooth*: proporciona dos opciones, sin seguridad o alta seguridad. Al seleccionar alta seguridad, se debe realizar un enlace (*pairing*) con el dispositivo remoto para poder establecer la conexión.
- *Nombre del dispositivo*: cada dispositivo Bluetooth tiene un nombre único dentro de una red.
- *Modo de operación*: permite seleccionar si el dispositivo es: no conectable (*non-connectable*), conectable (*connectable*) o conectable y descubrible (*connectable and discoverable*). Si un dispositivo está en modo descubrible, puede ser encontrado cuando otros dispositivos realicen búsquedas.
- *Desconexión automática*: posibilita al módulo Bluetooth desconectarse automáticamente después de un tiempo específico sin ninguna actividad.
- *Configuración vía Bluetooth*: permite la configuración del módulo Bluetooth mediante un enlace Bluetooth.

En la Figura A.9 se muestra el diagrama de flujo de la configuración de los módulos Bluetooth con base al asistente de configuración.



**Figura A.8.** Configuración de las características del módulo Bluetooth.



**Figura A.9.** Configuración del módulo Bluetooth.

## Anexo B

Este Anexo enlista los mensajes multilínea de la interfaz y describe los nemónicos y mensajes que corresponden a las funciones de la interfaz GPIB.

Los mensajes de la interfaz multilínea (Tablas B.1 y B.2) son órdenes definidas en el estándar IEEE 488. Los mensajes son enviados y recibidos con señales ATN positivas. Las funciones de la interfaz incluyen la inicialización del bus, el direccionamiento y desdireccionamiento de los dispositivos, y la configuración de los dispositivos en modos de programación local o remoto. Para más información acerca de estos mensajes, consultar el estándar ANSI/IEEE 488.1-1987 “*IEEE Standard Digital Interface for Programmable Instrumentation*” [8].

**Tabla B.1.** Mensajes de la interfaz multilínea.

HEX	DEC	ASCII	Mensaje	HEX	DEC	ASCII	Mensaje
00	0	NUL	-	20	32	SP	MLA0
01	1	SOH	GTL	21	33	!	MLA1
02	2	STX	-	22	34	“	MLA2
03	3	ETX	-	23	35	#	MLA3
04	4	EOT	SDC	24	36	\$	MLA4
05	5	ENQ	PPC	25	37	%	MLA5
06	6	ACK	-	26	38	&	MLA6
07	7	BEL	-	27	39	‘	MLA7
08	8	BS	GET	28	40	(	MLA8
09	9	HT	TCT	29	41	)	MLA9
0A	10	LF	-	2A	42	*	MLA10
0B	11	VT	-	2B	43	+	MLA11
0C	12	FF	-	2C	44	,	MLA12
0D	13	CR	-	2D	45	-	MLA13
0E	14	SO	-	2E	46	.	MLA14
0F	15	SI	-	2F	47	/	MLA15
10	16	DEL	-	30	48	0	MLA16
11	17	DC1	LLO	31	49	1	MLA17
12	18	DC2	-	32	50	2	MLA18
13	19	DC3	-	33	51	3	MLA19
14	20	DC4	DCL	34	52	4	MLA20
15	21	NAK	PPU	35	53	5	MLA21
16	22	SYN	-	36	54	6	MLA22
17	23	ETB	-	37	55	7	MLA23
18	24	CAN	SPE	38	56	8	MLA24
19	25	EM	SPD	39	57	9	MLA25
1A	26	SUB	-	3A	58	:	MLA26
1B	27	ESC	-	3B	59	;	MLA27
1C	28	FS	-	3C	60	<	MLA28
1D	29	GS	-	3D	61	=	MLA29
1E	30	RS	-	3E	62	>	MLA30
1F	31	US	CFE	3F	63	?	UNL

**Tabla B.1.** Mensajes de la interfaz multilinea (Continuación).

HEX	DEC	ASCII	Mensaje	HEX	DEC	ASCII	Mensaje
40	64	@	MTA0	60	96	`	MSA0,PPE
41	65	A	MTA1	61	97	a	MSA1,PPE,CFG1
42	66	B	MTA2	62	98	b	MSA2,PPE,CFG2
43	67	C	MTA3	63	99	c	MSA3,PPE,CFG3
44	68	D	MTA4	64	100	d	MSA4,PPE,CFG4
45	69	E	MTA5	65	101	e	MSA5,PPE,CFG5
46	70	F	MTA6	66	102	f	MSA6,PPE,CFG6
47	71	G	MTA7	67	103	g	MSA7,PPE,CFG7
48	72	H	MTA8	68	104	h	MSA8,PPE,CFG8
49	73	I	MTA9	69	105	i	MSA9,PPE,CFG9
4A	74	J	MTA10	6A	106	j	MSA10,PPE,CFG10
4B	75	K	MTA11	6B	107	k	MSA11,PPE,CFG11
4C	76	L	MTA12	6C	108	l	MSA12,PPE,CFG12
4D	77	M	MTA13	6D	109	m	MSA13,PPE,CFG13
4E	78	N	MTA14	6E	110	n	MSA14,PPE,CFG14
4F	79	O	MTA15	6F	111	o	MSA15,PPE,CFG15
50	80	P	MTA16	70	112	p	MSA16,PPD
51	81	Q	MTA17	71	113	q	MSA17,PPD
52	82	R	MTA18	72	114	r	MSA18,PPD
53	83	S	MTA19	73	115	s	MSA19,PPD
54	84	T	MTA20	74	116	t	MSA20,PPD
55	85	U	MTA21	75	117	u	MSA21,PPD
56	86	V	MTA22	76	118	v	MSA22,PPD
57	87	W	MTA23	77	119	w	MSA23,PPD
58	88	X	MTA24	78	120	x	MSA24,PPD
59	89	Y	MTA25	79	121	y	MSA25,PPD
5A	90	Z	MTA26	7A	122	z	MSA26,PPD
5B	91	[	MTA27	7B	123	{	MSA27,PPD
5C	92	\	MTA28	7C	124		MSA28,PPD
5D	93	]	MTA29	7D	125	}	MSA29,PPD
5E	94	^	MTA30	7E	126	~	MSA30,PPD
5F	95		UNT	7F	127	DEL	-

**Tabla B.2.** Definiciones de los mensajes de la interfaz multilinea.

Mensaje	Descripción
CFE*	Configuration Enable
CFG*	Configure
DCL	Device Clear
GET	Group Execute Trigger
GTL	Go To Local
LLO	Local Lockout
MLA	My Listen Ardes
MSA	My Secondary Adress
MTA	My Talk Address
PPC	Parallel Poll Configue
PPD	Parallel Poll Disable
PPE	Parallel Poll Enable
PPU	Parallel Poll Unconfigure
SDC	Selected Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
TCT	Take Control
UNL	Unlisten
UNT	Untalk

\* Este mensaje de la interfaz multilinea es una extensión propuesta a la especificación IEEE 488 para soportar el protocolo HS488.