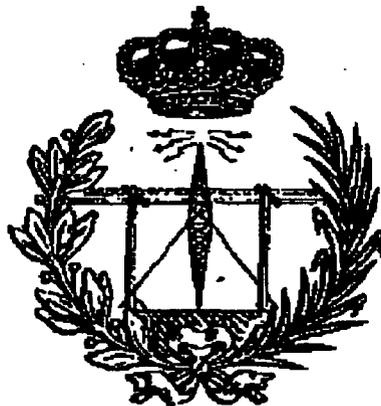


UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN**



PROYECTO FIN DE CARRERA

***SISTEMA INALÁMBRICO PARA MONITORIZACIÓN EN PC DE
SEÑALES ECG BASADO EN LA TECNOLOGÍA BLUETOOTH***

TUTOR: Dr. D. José Alberto Rabadán Borges

AUTOR: Germán Quintana Rodríguez

FECHA: Diciembre de 2005

AGRADECIMIENTOS

A todos aquellos que me han apoyado y animado durante estos años de carrera, y en especial a mi familia y mis amigos, gracias.

Índice General

MEMORIA	1
CAPÍTULO 1. Introducción	1
CAPÍTULO 2. Teoría del Electrocardiograma	7
2.1 Introducción.....	7
2.2 El corazón	8
2.2.1 Anatomía	8
2.2.2 Funcionamiento.....	9
2.3 El electrocardiograma.....	10
2.3.1 Definición.....	10
2.3.2 Modelo teórico.....	10
2.3.3 Significado de la señal ECG.....	12
2.3.4 Tipos de electrocardiograma	13
2.4 Características de la señal ECG	14
2.4.1 Características en amplitud y frecuencia.....	14
2.4.2 Componentes de ruido e interferencia.....	14
CAPÍTULO 3. Tecnología Bluetooth	15
3.1 Introducción.....	15
3.2 Conceptos preliminares.....	16
3.2.1 Descripción general.....	16
3.2.2 Elementos característicos de las redes Bluetooth.....	18
3.3 Radio.....	19
3.3.1 Introducción.....	19
3.3.2 Espectro ensanchado por salto de frecuencia.....	20
3.3.3 Modulación	21
3.3.4 Potencia emitida.....	22
3.4 Bandabase	23
3.4.1 Elementos característicos de Bandabase	23
3.4.1.1 Dirección de dispositivo Bluetooth.....	23
3.4.1.2 Multiplexación por división del tiempo	23

3.4.1.3 Sistema de tiempos	24
3.4.1.4 Tipos de enlaces	25
3.4.2 Estructura y tipos de paquetes Bluetooth	25
3.4.3 Controlador de enlace y estados de operación	27
3.5 Dirección de enlace	29
3.5.1 Mensajes LMP y canal lógico de dirección de enlace	29
3.5.2 Funciones desempeñadas por la dirección de enlace	30
3.5.2.1 Establecimiento del enlace	30
3.5.2.2 Intercambio de papeles	30
3.5.2.3 Control de potencia	31
3.5.2.4 Otras funciones	31
3.6 Interfaz controlador de host	32
3.6.1 Paquetes HCI	32
3.6.1.1 Comandos HCI	33
3.6.1.2 Paquetes de datos HCI	33
3.6.1.3 Eventos HCI	34
3.6.2 Capa de transporte HCI	35
3.6.3 Establecimiento de una conexión	35
3.7 Protocolo de control de enlace lógico y adaptación	37
3.7.1 Multiplexación mediante canales lógicos	37
3.7.2 Señalización L2CAP	38
3.7.2.1 Estructuras de señalización L2CAP	38
3.8 RFCOMM	41
3.8.1 Tipos de dispositivos RFCOMM	41
3.8.2 Tramas RFCOMM	42
3.8.2.1 Estructura de las tramas RFCOMM	43
3.8.2.2 Tramas de multiplexor	44
3.8.3 Establecimiento de una conexión	45
3.9 Protocolo de descubrimiento de servicios	46
3.9.1 La base de datos SDP	46
3.9.1.1 Atributos de servicio	46
3.9.1.2 Elementos de datos	47
3.9.1.3 Identificadores únicos universales (UUIDs)	48
3.9.2 Mensajes SDP	48

3.9.3 Establecimiento de una sesión SDP.....	49
3.10 Perfil Bluetooth puerto serie	50
CAPÍTULO 4. Descripción del Sistema.....	53
4.1 Introducción.....	53
4.2 Componentes del sistema	53
4.2.1 Unidad portátil	54
4.2.2 Unidad de visualización ECG	56
4.3 Funcionamiento del sistema.....	58
4.4 Conclusiones.....	59
CAPÍTULO 5. Unidad Portátil.....	61
5.1 Introducción.....	61
5.2 Diseño e implementación del hardware.....	62
5.2.1 Etapa de acondicionamiento de la señal ECG	63
5.2.1.1 Primera etapa de amplificación	64
5.2.1.2 Filtro paso alto.....	65
5.2.1.3 Segunda etapa de amplificación.....	66
5.2.1.4 Filtro paso bajo	67
5.2.1.5 Referencia de tensión.....	68
5.2.2 Microcontrolador.....	69
5.2.3 Módulo Bluetooth.....	70
5.2.4 Etapa de alimentación.....	71
5.2.5 Conectores y circuito de reset	72
5.2.6 Características de la PCB	73
5.3 Diseño e implementación del software.....	74
5.3.1 Implementación de capa HCI	75
5.3.1.1 Driver de bus físico.....	75
5.3.1.2 Capa HCI.....	76
5.3.2 Implementación de la capa L2CAP	77
5.3.3 Implementación de la capa SDP	79
5.3.4 Implementación de la capa RFCOMM.....	80
5.3.5 Implementación de la aplicación	82
5.4 Conclusiones.....	87

CAPÍTULO 6. Unidad de Visualización ECG	89
6.1 Introducción.....	89
6.2 Adaptador Bluetooth-USB	90
6.3 Aplicación de visualización.....	92
6.3.1 Entorno de programación	92
6.3.2 Interfaz de usuario.....	93
6.3.3 Procesado de la señal ECG	98
6.3.3.1 Cancelación de la interferencia de red de 50 Hz.....	99
6.3.3.2 Diezmado	101
6.3.3.3 Filtro de suavizado	101
6.3.3.4 Obtención de la frecuencia cardiaca	102
6.4 Conclusiones.....	104
CAPÍTULO 7. Medidas y Resultados	107
7.1 Introducción.....	107
7.2 Evaluación del enlace Bluetooth	108
7.3 Evaluación de la señal ECG visualizada	111
7.4 Conclusiones.....	116
CAPÍTULO 8. Conclusiones	117
BIBLIOGRAFÍA	121
PRESUPUESTO	123
PLIEGO DE CONDICIONES	131
PLANOS	135
ANEXOS	143
Anexo 1 Intercambio de paquetes Bluetooth.....	145
Anexo 2 Datasheets.....	165

Índice de Figuras

Figura 2.1 Dibujo esquemático del corazón humano	9
Figura 2.2 Derivaciones de Einthoven	11
Figura 2.3 Derivaciones de Wilson	12
Figura 2.4 Representación típica de una señal ECG	13
Figura 3.1 Arquitectura y pila de protocolos Bluetooth	17
Figura 3.2 Ejemplos de <i>piconet</i> y <i>scatternet</i>	18
Figura 3.3 Asignación de <i>slots</i> de tiempo a paquetes de distinto tamaño	24
Figura 3.4 Estructura de un paquete Bluetooth	25
Figura 3.5 Estados de operación del controlador de enlace	28
Figura 3.6 Intercambio de papeles maestro-esclavo	30
Figura 3.7 Posición de la capa HCI en la pila de protocolos Bluetooth	32
Figura 3.8 Estructura de un comando HCI	33
Figura 3.9 Estructura de paquetes de datos ACL y SCO	34
Figura 3.10 Estructura de un evento HCI	35
Figura 3.11 Establecimiento de una conexión a nivel HCI	36
Figura 3.12 Estructura de un paquete L2CAP	37
Figura 3.13 Estructura de un comando L2CAP	38
Figura 3.14 Establecimiento de una conexión a nivel L2CAP	40
Figura 3.15 Dispositivos RFCOMM tipo 1 y tipo 2	42
Figura 3.16 Estructura de una trama RFCOMM.....	44
Figura 3.17 Establecimiento de una conexión a nivel RFCOMM	45
Figura 3.18 Ejemplo de un elemento de datos	47
Figura 3.19 Estructura de una PDU SDP	48
Figura 3.20 Intercambio de mensajes durante una sesión SDP.....	50
Figura 4.1 Esquema completo del sistema diseñado.....	54
Figura 4.2 Diagrama de bloques simplificado de la unidad portátil	55
Figura 4.3 Diagrama de bloques de la unidad de visualización ECG.....	57
Figura 5.1 Diagrama de bloques de la unidad portátil.....	62
Figura 5.2 Canal de acondicionamiento de la señal ECG.....	63
Figura 5.3 Estructura de la primera etapa de amplificación	65
Figura 5.4 Estructura del filtro paso alto.....	65

Figura 5.5 Estructura de la segunda etapa de amplificación.....	67
Figura 5.6 Estructura del filtro paso bajo.....	67
Figura 5.7 Referencia de tensión de 1,2 voltios	68
Figura 5.8 Esquema de conexionado del PIC16F876.....	69
Figura 5.9 Placa del kit de desarrollo Bluetooth de Ericsson	71
Figura 5.10 Esquema de conexionado del regulador de tensión 78M05.....	72
Figura 5.11 Esquema del circuito de reset del sistema	72
Figura 5.12 Distribución de conectores de la unidad portátil.....	73
Figura 5.13 Diagrama temporal de transmisión de la señal ECG.....	82
Figura 5.14 Flujograma de la entidad de CONTROL	85
Figura 5.15 Flujogramas de la aplicación y rutinas de interrupción.....	86
Figura 5.16(a) Fotografía de la PCB implementada	88
Figura 5.16(b) Fotografía del módulo Bluetooth	88
Figura 6.1 Fotografía del adaptador Bluetooth-USB	90
Figura 6.2 Aplicación de control del adaptador BT-USB	91
Figura 6.3 Interfaz de usuario de la aplicación de visualización ECG.....	93
Figura 6.4 Ventana de diálogo de configuración de visualización.....	95
Figura 6.5(a) Aspecto de un canal habilitando las líneas de división	96
Figura 6.5(b) Ventana de captura de un ciclo de señal ECG	96
Figura 6.6 Ventana de diálogo de configuración de filtros	96
Figura 6.7(a) Selección de la zona de señal ECG a ampliar.....	98
Figura 6.7(b) Ampliación de la zona de señal ECG seleccionada.....	98
Figura 6.8 Procesado digital aplicado a la señal ECG	99
Figura 6.9 Diagrama de bloques clásico del filtrado adaptativo	99
Figura 6.10(a) Señal ECG original con interferencia de 50 Hz.....	100
Figura 6.10(b) Señal ECG filtrada mediante filtrado adaptativo	100
Figura 6.11(a) Señal ECG a la salida del filtro cancelador de la interferencia de red de 50 Hz	101
Figura 6.11(b) Señal ECG diezmada por dos.....	101
Figura 6.12(a) Señal ECG diezmada antes de ser filtrada	102
Figura 6.12(b) Señal ECG diezmada después de ser filtrada	102
Figura 6.13 Estructura del algoritmo de detección de complejos QRS	103
Figura 6.14 Señal a la salida del bloque promediador	104

Figura 7.1 Conexión del puerto serie del PC al interfaz RS-232 entre host y módulo Bluetooth.....	109
Figura 7.2 Aplicación de monitorización de intercambio de paquetes HCI....	110
Figura 7.3 Puntos de medición A y B en la etapa de acondicionamiento	111
Figura 7.4(a) Señal II en el punto A.....	112
Figura 7.4(b) Señal III en el punto A.....	112
Figura 7.5(a) Señal II en el punto B.....	112
Figura 7.5(b) Señal III en el punto B.....	112
Figura 7.6(a) Señal II visualizada en el PC	113
Figura 7.6(b) Señal III visualizada en el PC	113
Figura 7.6(c) Señal I visualizada en el PC	113
Figura 7.7(a) Señal II muestreada a 100 muestras/s y filtrada con el filtro de suavizado.....	114
Figura 7.7(b) Señal III muestreada a 100 muestras/s y filtrada con el filtro de suavizado.....	114
Figura 7.8(a) Unidad portátil junto al paciente electrónico y la fuente de alimentación.....	115
Figura 7.8(b) Unidad de visualización ECG	115
Figura 7.8(c) Unidad portátil.....	115
Figura 7.8(d) Adaptador Bluetooth-USB.....	115
Figura 7.8(e) Señal ECG visualizada en el osciloscopio digital.....	115
Figura 7.8(f) Paciente electrónico.....	115

Índice de Tablas

Tabla 3.1 Valores espectrales característicos de Bluetooth	20
Tabla 3.2 Clasificación de los dispositivos Bluetooth en función de la potencia emitida	22
Tabla P.1 Precios de materiales utilizados	125
Tabla P.2 Amortización del inmovilizado material	126
Tabla P.3 Amortización del inmovilizado inmaterial	127
Tabla P.4 Presupuesto aplicando impuestos	128
Tabla P.5 Resumen del presupuesto	129

MEMORIA



CAPÍTULO 1

INTRODUCCIÓN

En el proyecto que se presenta a continuación, se describe el diseño e implementación de un sistema de monitorización cardíaco basado en la tecnología inalámbrica Bluetooth (BT), el cual permite la adquisición, transmisión y visualización de señales ECG en aplicaciones de monitorización de pacientes.

Este proyecto se engloba dentro del ámbito interdisciplinar de la ingeniería biomédica. Éste es uno de los campos de aplicación donde las nuevas tecnologías han encontrado mayor grado de aceptación en los últimos años. En este sentido merecen ser destacadas las técnicas de monitorización de señales biomédicas, en concreto las relacionadas con el *electrocardiograma* (ECG). Sus aplicaciones en la medicina moderna van desde el diagnóstico de diversas patologías cardíacas a la monitorización de las constantes vitales de un paciente [3]. El desarrollo de las nuevas tecnologías ha permitido dotar de nuevas prestaciones a los equipos destinados a tal fin. Entre ellas destacan la aplicación de técnicas de procesado digital de señales, así como las diversas

tecnologías inalámbricas que permiten la sustitución de los siempre molestos cables que interconectan paciente y equipo de monitorización por enlaces inalámbricos.

En la actualidad existen dispositivos formados por módulos emisores y receptores adaptables a equipos de monitorización tradicionales. De esta forma se consigue sustituir los cables por una conexión inalámbrica. De hecho en esta escuela, dentro del Grupo de Tecnología Fotónica, se ha realizado un Proyecto Fin de Carrera basado en esta idea [2]. Para ello se ha recurrido a la aplicación de la tecnología Bluetooth, de forma que no sólo se consigue una alta fiabilidad y robustez en la conexión, sino que permite introducir este tipo de aplicaciones dentro del campo de la telemedicina, ya que se hace posible la interacción con otro tipo de tecnologías como OBEX o WAP, facilitando la conectividad con diversos tipos de redes (WAN, Internet, etc.). Por otro lado, dentro del ámbito de la Escuela Universitaria de Ingeniería Técnica de Telecomunicación también se ha llevado a cabo el diseño e implementación de un proyecto fin de carrera [6] similar al que se describe en esta memoria, pero implementando el enlace inalámbrico mediante un hardware específico, basado en la modulación digital FSK, desarrollado específicamente para dicho proyecto.

A partir de las posibles líneas de desarrollo establecidas por los trabajos mencionados en el párrafo anterior surge el proyecto fin de carrera aquí presentado. En éste se pretende aplicar la tecnología Bluetooth en el objetivo de sustituir los cables de interconexión entre paciente y equipo. Además se sustituirán los caros equipos de monitorización tradicionales por un simple ordenador personal (PC). Así es posible abaratar costes de equipamiento debido a lo comunes que son, hoy en día, este tipo de equipos informáticos en prácticamente cualquier entorno laboral o doméstico. Con el sistema desarrollado en este proyecto se amplían las posibles aplicaciones de los sistemas biomédicos, ya que permite el uso de tecnologías de bajo coste sin reducir la calidad de las medidas y proporcionando un entorno de trabajo familiar a todos los usuarios.

El sistema está formado por varios sensores biomédicos conectados a un módulo transmisor de bajo consumo, tamaño y peso, y un equipo de monitorización formado por un adaptador Bluetooth-USB que implementa la interfaz radio en la parte de recepción, conectado a un ordenador personal. El módulo transmisor debería

caracterizarse por sus reducidas dimensiones, de forma que al paciente su utilización le resulte lo más cómoda posible. Por tanto se tratará un sistema autónomo, de bajo peso y bajo consumo. Ya que el ECG implementado en este proyecto es el de monitorización solamente habrá cuatro sensores conectados al módulo emisor. En él se acondicionarán las señales procedentes de los sensores, se digitalizarán, multiplexarán y finalmente se enviarán mediante el enlace Bluetooth al ordenador personal. El módulo emisor o unidad portátil está formado básicamente por amplificadores y filtros destinados al acondicionamiento de las señales, un módulo Bluetooth de Ericsson, que implementa la interfaz radio de la unidad, y un microcontrolador PIC, encargado de realizar la conversión analógica-digital de las señales y enviar los datos al módulo Bluetooth para su posterior transmisión..

En cuanto al receptor, como ya se ha mencionado, está formado por un ordenador personal de medianas prestaciones, conectado a un dispositivo comercial Bluetooth. Este dispositivo captará la señal de radio, y enviará los datos recibidos al PC, donde una aplicación software obtendrá las señales anteriormente mencionadas y las representará en la pantalla del ordenador.

Los objetivos propuestos para la realización de este proyecto y que han de servir para evaluar los resultados del mismo son:

- Diseñar e implementar la circuitería de acondicionamiento de la señal ECG del módulo emisor.
- Diseñar e implementar el programa en ensamblador que se ejecutará en el microcontrolador del módulo emisor.
- Diseñar e implementar el módulo emisor, integrando para ello la circuitería de acondicionamiento, el microcontrolador PIC y el módulo Bluetooth.
- Diseñar e implementar la aplicación encargada de recibir los datos en el PC, para posteriormente poder realizar la visualización de los mismos en la pantalla del ordenador.

En cuanto a los medios materiales necesarios para la realización del diseño e implementación de este proyecto se hará uso de los siguientes recursos:

- Un ordenador personal.
- Herramientas informáticas:
 - MPLAB IDE 6.2
 - IC-PROG
 - Borland C++ Builder 5.0
 - Orcad Pspice 9.1
 - Protel
 - MATLAB

- Instrumentación electrónica:
 - Osciloscopio.
 - Generador de funciones.
 - Polímetro.
 - Fuente de tensión

- Materiales para la construcción del prototipo:
 - Módulo BLuetooth Rok101008 de Ericsson
 - Microcontrolador PIC16F876 de Microchip.
 - Circuitos integrados analógicos.
 - Elementos pasivos
 - Adaptador Bluetooth-USB BL-554 de Brain Boxes.
 - PC

A partir del segundo capítulo de esta memoria los diversos aspectos de diseño e implementación del proyecto se agrupan de la siguiente forma:

- Capítulo 2: una breve explicación de algunos conceptos básicos sobre el funcionamiento del corazón y la técnica del electrocardiograma necesarios para el desarrollo e implementación de la aplicación propuesta en este proyecto.

- **Capítulo 3:** introducción a la tecnología Bluetooth, profundizando en aquellos elementos que puedan tener mayor influencia sobre las prestaciones del sistema realizado, tales como procedimientos de conexión entre dispositivos, formato de las tramas de datos utilizadas o velocidades de transmisión, entre otros.
- **Capítulo 4:** descripción general de los elementos que componen el sistema y justificación de las decisiones de diseño tomadas para cada uno de los bloques que lo componen.
- **Capítulo 5:** descripción detallada del diseño e implementación del módulo emisor, tanto a nivel del software de comunicaciones Bluetooth desarrollado, como del hardware necesario para el acondicionamiento, digitalización y transmisión de las señales ECG.
- **Capítulo 6:** presentación de los componentes del módulo receptor y descripción de la aplicación software desarrollada y destinada a ejecutarse en el PC que se encargará de la tarea de análisis y presentación de la señal ECG recibida.
- **Capítulo 7:** explicación y evaluación de los resultados obtenidos.
- **Capítulo 8:** exposición de las conclusiones y de las futuras líneas de desarrollo.

Capítulo 1: Introducción

CAPÍTULO 2

TEORÍA DEL ELECTROCARDIOGRAMA

2.1 Introducción

En este capítulo se introducen algunos conceptos básicos relacionados con la aplicación de la técnica del electrocardiograma (ECG). Para ello se hace necesario introducir algunas nociones de anatomía y fisiología cardíaca, sin ser, por supuesto, exhaustivos. Con esto se pretende dar una visión general sobre la importancia del electrocardiograma en la detección y diagnóstico de las diversas patologías cardíacas, así como de la génesis y significado de las señales que aparecen durante la aplicación de esta técnica.

Posteriormente se detallarán algunos de los aspectos característicos de la señal ECG en cuanto a amplitud, ancho de banda y ruido que son necesarios conocer para poder realizar correctamente la adquisición, acondicionamiento y presentación de la misma al observador humano.

2.2 El corazón

2.2.1 Anatomía

Aproximadamente del tamaño del puño de una persona, el corazón es un órgano hueco, formado principalmente por tejido muscular, que se sitúa en la cavidad torácica entre la tercera y la sexta costilla ocupando una posición central, ligeramente inclinada a la izquierda. Su cometido es impulsar la sangre por todo el sistema circulatorio con el objetivo de suministrar oxígeno y nutrientes a todas las células del organismo, así como de eliminar el dióxido de carbono y otros deshechos propios del metabolismo celular. En un adulto en reposo, el corazón late entre 60 y 80 veces por minuto, bombeando en cada latido unos 70 ml de sangre [3].

El corazón se divide internamente en cuatro cavidades: dos **aurículas** en la parte superior, y dos **ventrículos** en la parte inferior. Existe una pared interna llamada **septum** que incomunica las aurículas entre sí. Los ventrículos también se encuentran separados por dicha estructura. La aurícula y el ventrículo derechos se comunican a través de la **válvula tricúspide**. Análogamente, la aurícula y el ventrículo izquierdos se comunican mediante la **válvula mitral**. A la salida de los ventrículos también existen unas válvulas. Son la **válvula pulmonar** para el ventrículo derecho, y la **válvula aórtica** para el ventrículo izquierdo. La principal función que cumplen las válvulas es controlar el paso de la sangre entre las distintas cavidades, y evitar que exista un reflujo hacia atrás de la sangre en su camino natural a través del corazón.

En el corazón existe otro elemento de vital importancia que merece ser destacado. Se trata del **sistema de conducción** de los impulsos eléctricos que controlan la contracción de los músculos cardiacos. Este sistema está formado por una compleja red de fibras y células especializadas, en la que destacan el **nodo seno-auricular (nodo SA)**, que se localiza en la parte alta de la aurícula derecha, el **nodo aurículo-ventricular (nodo AV)**, situado entre las aurículas y los ventrículos, y el **haz de Hiss**, también denominado **sistema de Hiss-Purkinje**, que se extiende hacia la parte inferior del corazón a lo largo del septum.

2.2.2 Funcionamiento

Se denomina latido a la contracción rítmica y periódica de los músculos del corazón, que tiene como objetivo impulsar la sangre por todo el sistema circulatorio. La sangre proveniente de todo el organismo, pobre en oxígeno y rica en dióxido de carbono, llega hasta la aurícula derecha. Ésta se contrae enviando la sangre al ventrículo derecho, desde donde, mediante una nueva contracción es impulsada hacia los pulmones. Aquí se produce el intercambio de dióxido de carbono por oxígeno, de forma que la sangre puede retornar de nuevo al corazón, donde es recibida por la aurícula izquierda, la cual a su vez la envía al ventrículo izquierdo para ser posteriormente bombeada al resto del organismo a través de la aorta.

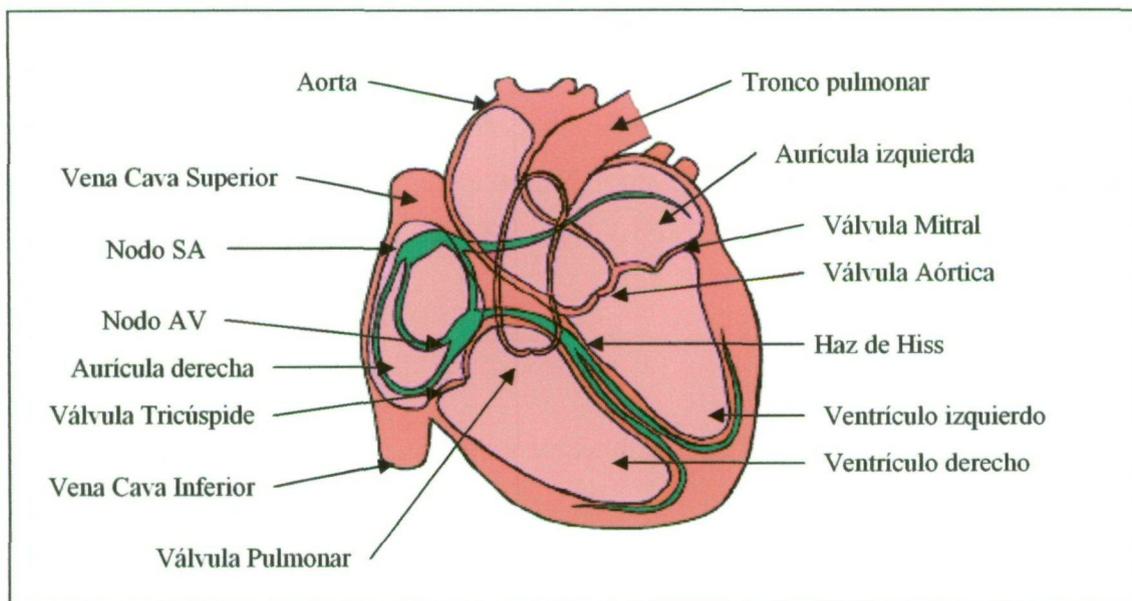


Figura 2.1 Dibujo esquemático del corazón humano.

Es evidente que se hace necesaria una buena coordinación en la contracción de los músculos cardiacos. De esto se encarga el **sistema de conducción**. El impulso eléctrico se genera inicialmente en el **nodo SA**. Debido a ello este nodo es considerado como el marcapasos natural del corazón. Desde aquí el impulso se extiende por toda la parte superior del corazón, causando la contracción de ambas aurículas. Esta fase del latido es denominada **diástole**. Seguidamente el impulso llega al **nodo AV**, donde sufre un retardo momentáneo, mientras se produce la contracción auricular. A continuación el impulso se transmite al **haz de Hiss**, el cual se ramifica por la parte inferior del corazón,

dando lugar a la contracción de los ventrículos. Esta fase se denomina **sístole**. Si por alguna razón la propagación de los impulsos eléctricos sufren algún tipo de interrupción, se retrasan o toman caminos equivocados, los latidos pueden volverse irregulares, o anormalmente rápidos o lentos, dando lugar a lo que se denomina una **arritmia cardiaca**.

2.3 El electrocardiograma

2.3.1 Definición

El **electrocardiograma o ECG** (también llamado **EKG**) es una prueba médica no invasiva e indolora que se utiliza para caracterizar la actividad eléctrica del corazón. En él se estudia la propagación de los impulsos eléctricos desde el nodo SA hasta los ventrículos. Observando las señales que se obtienen a partir de una serie de electrodos situados en distintas partes del cuerpo del paciente, es posible detectar defectos eléctricos o mecánicos del corazón, así como monitorizar la frecuencia cardiaca [5].

2.3.2 Modelo teórico

La base teórica sobre la que se fundamenta el electrocardiograma fue propuesta hace más de cien años por Einthoven. En ella se postula que la actividad eléctrica cardiaca puede ser modelada como un dipolo eléctrico caracterizado por un vector [5]. Debido a que el cuerpo humano es considerado, desde un punto de vista eléctrico, como puramente resistivo a las frecuencias características del ECG, es posible situar los electrodos en las extremidades del paciente, de forma que las diferencias de potencial registradas entre los distintos electrodos pueden representar fielmente a aquellas que se producen en el corazón [3].

Se define como **derivación** a la medida diferencial obtenida a partir de dos o más electrodos situados en diversos puntos del cuerpo. Einthoven estableció en su modelo [3] cuatro puntos en los que colocar los electrodos, una por cada extremidad. Estos puntos son: **LA** (Left Arm, brazo izquierdo), **RA** (Right Arm, brazo derecho), **LL** (Left Leg, pierna izquierda) y **RL** (pierna derecha). A partir de estos cuatro puntos se obtienen tres derivaciones denominadas **I**, **II** y **III** que se definen como

$$\begin{aligned} \text{I} &= V_{\text{LA}} - V_{\text{RA}} \\ \text{II} &= V_{\text{LL}} - V_{\text{RA}} \\ \text{III} &= V_{\text{LL}} - V_{\text{LA}} \end{aligned} \quad (2.1)$$

El electrodo de la pierna derecha (RL) se toma como masa de referencia del sistema. Según la ley de Kirchhoff para las tensiones, que dice que la suma de las tensiones dentro de una malla cerrada es igual a cero, se cumple que $\text{II} - \text{I} - \text{III} = 0$. En la figura 2.2 se muestra la relación entre las derivaciones propuestas por Einthoven, y el modelo de excitación cardíaca. El triángulo sobre el que se definen las diferencias de potencial asociadas a las derivaciones I, II y III es conocido como *triángulo equilátero de Einthoven*, y en él se aprecia como las proyecciones del vector de excitación cardíaca sobre los lados del triángulo son proporcionales a las diferencias de tensión medidas entre las extremidades.

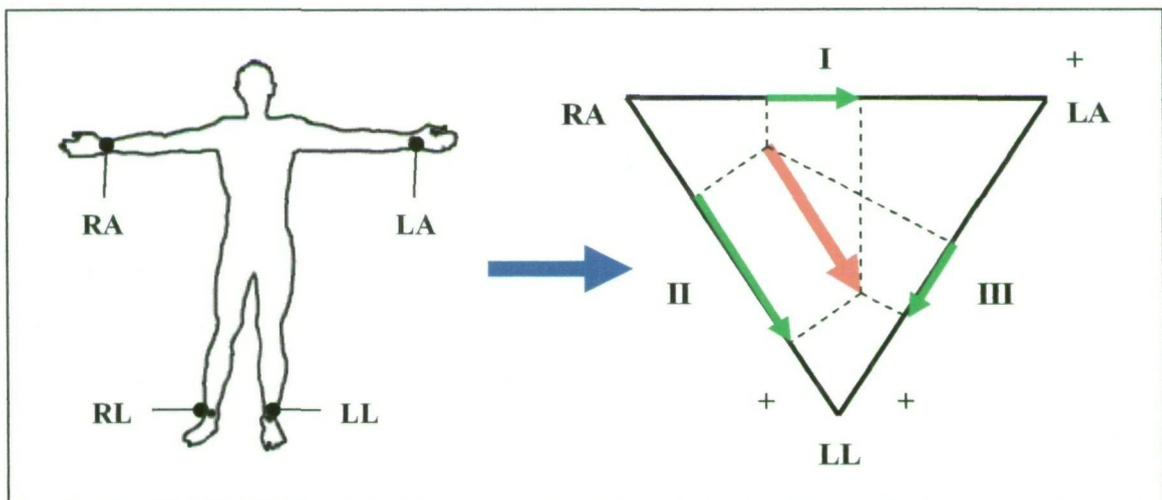


Figura 2.2 Relación entre el triángulo equilátero de Einthoven y la posición de los electrodos RA, LA, RL y LL. El vector en el interior del triángulo caracteriza la excitación cardíaca.

Posteriormente este modelo fue ampliado de forma que se añadieron nueve derivaciones más a las propuestas por Einthoven, sumando un total de doce derivaciones [3]. Wilson estableció las *derivaciones monopolares de las extremidades aumentadas* denominadas **aVL**, **aVR** y **aVF**, que se obtienen a partir de los mismos electrodos que las derivaciones bipolares de Einthoven. También definió las

derivaciones monopares precordiales, denominadas V_1 , V_2 , V_3 , V_4 , V_5 y V_6 , obtenidas a partir de seis electrodos situados en el pecho. Estas doce derivaciones son la base del ECG clínico estándar. En la figura 2.3 se muestra la obtención de las derivaciones propuestas por Wilson.

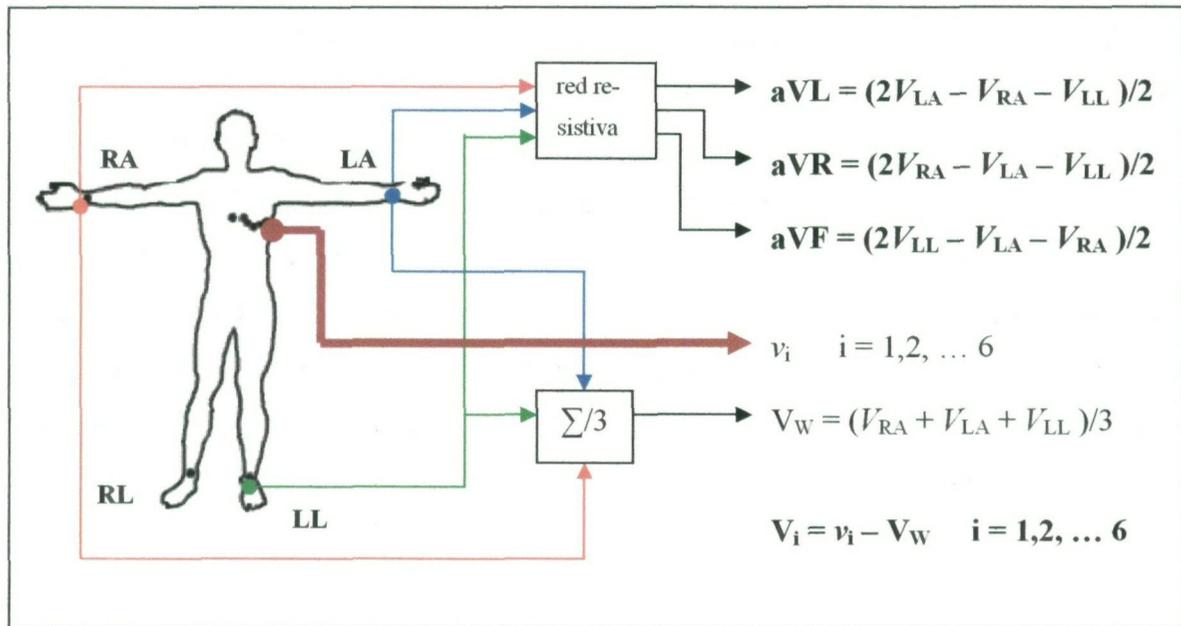


Figura 2.3 Obtención de las de las derivaciones propuestas por Wilson.

El bloque denominado red resistiva es un circuito compuesto por resistencias que realizan una ponderación de las señales de entrada para así obtener las derivaciones monopares aumentadas. La tensión V_W se obtiene promediando el valor de las tensiones en RA, LA y LL.

2.3.3 Significado de la señal ECG

En la figura 2.4 se muestra una representación típica de una señal ECG. En ella se observan la onda P, el complejo QRS y la onda T. Cada uno de estos elementos representa un determinado aspecto de la propagación de los impulsos eléctricos en el corazón. La onda P se corresponde con la activación de las aurículas. A continuación se produce un breve instante de inactividad hasta que la contracción ventricular provoca la aparición del complejo QRS, que se caracteriza por la gran amplitud de la onda R. Después tiene lugar otro periodo momentáneo de inactividad, hasta que se produce la

aparición de la onda T, debido a la vuelta de los ventrículos a su estado normal de reposo. En algunos individuos, puede aparecer un pequeño pico después de la onda T, denominado onda U. Su origen no está del todo claro, pero parece deberse a un potencial de repolarización [3].

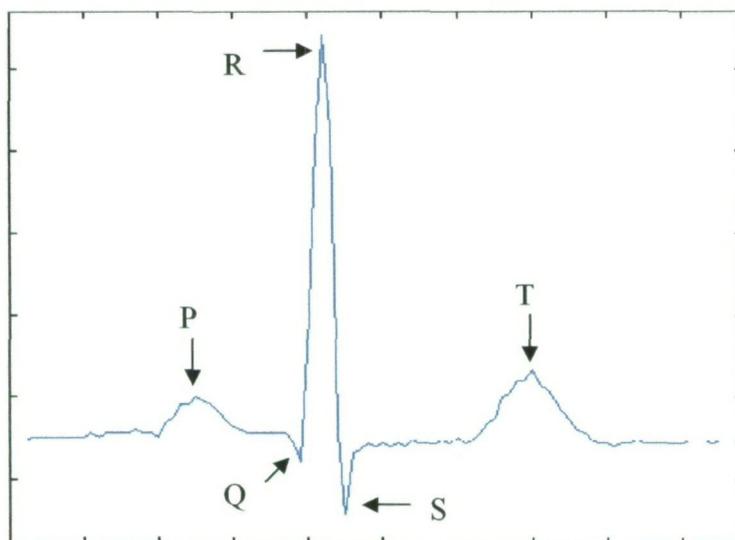


Figura 2.4 Representación de una señal ECG típica obtenida a partir de la derivación II. Se pueden observar las ondas P, Q, R, S y T.

2.3.4 Tipos de electrocardiograma

Existen varios tipos de ECG en función de la aplicación médica y del número de derivaciones utilizado [3]. Estos son:

- El **ECG clínico**: utiliza las 12 derivaciones estándar y suele trabajar con un ancho de banda de 0,05 a 150 Hz.
- El **ECG ambulatorio o Holter**: consiste en el registro del ECG durante un periodo de 24 horas. Normalmente sólo hace uso de 2 derivaciones.
- El **ECG de monitorización**: sólo utiliza 1 ó 2 derivaciones, ya que su objetivo principal es monitorizar el ritmo cardíaco del paciente. Para ello normalmente se utiliza la derivación II, por ser la de mayor amplitud. Se suele tomar la segunda derivación para el caso de que hubiera algún problema con la derivación

principal, como por ejemplo, una pérdida de contacto o desplazamiento del electrodo. El ancho de banda utilizado va de 0,5 a 50 Hz.

2.4 Características de la señal ECG

2.4.1 Características en amplitud y frecuencia

Como ya se ha mencionado, el espectro de la señal ECG se encuentra localizado principalmente entre 0 y 100-150 Hz, aunque dependiendo de la aplicación el ancho de banda de trabajo puede reducirse, como en el caso del ECG de monitorización, hasta los 50 Hz. Sin embargo también existen aplicaciones de alta resolución donde el ancho de banda puede llegar hasta los 500 Hz. En el caso particular del ECG de monitorización es de especial interés el complejo QRS para la obtención del ritmo cardiaco. Sus componentes espectrales se encuentran situadas en torno a los 17 Hz. La amplitud de pico de la señal ECG (dependiendo de la derivación) está en el rango de 1 a 2 mV [5]. Esto debe ser tenido en cuenta a la hora de determinar la ganancia de las etapas amplificadoras, sobre todo si posteriormente se va a realizar una conversión analógica-digital de la señal, de forma que se pueda aprovechar al máximo el rango de tensiones a la entrada del conversor.

2.4.2 Componentes de ruido e interferencia

Existen diferentes tipos de ruido e interferencias que pueden afectar a una correcta presentación del ECG, así como a la obtención de diversos parámetros a partir del mismo. En alta frecuencia existen componentes de ruido debido a la contracción muscular (ruido EMG) que deben ser filtradas. En baja frecuencia pueden aparecer componentes producidas por el movimiento de los electrodos. También la respiración produce una modulación en baja frecuencia por debajo de 0,5 Hz. En la banda de trabajo del ECG está presente la interferencia de 50 Hz producida por la red y los equipos situados en el entorno del paciente [5].

CAPÍTULO 3

TECNOLOGÍA BLUETOOTH

3.1 Introducción

Uno de los principales objetivos de este Proyecto Fin de Carrera es la aplicación de la tecnología Bluetooth en la implementación de un enlace inalámbrico entre un dispositivo portátil y un ordenador personal. Es por esta razón que el siguiente capítulo está dedicado a introducir conceptos elementales sobre el estándar de comunicaciones Bluetooth, necesarios para entender el trabajo realizado. Se detallarán los principales elementos en los que se basa esta tecnología y se explicará la arquitectura y los principales protocolos que la componen.

3.2 Conceptos preliminares

3.2.1 Descripción general

Bluetooth es un estándar de comunicaciones digitales inalámbricas, basado en la transmisión de ondas de radio de corto alcance y baja potencia que opera en la banda

ICM (industrial, científica y médica) de 2,4 GHz, por lo que no necesita de licencia administrativa [1,7]. Su principal aplicación es la sustitución de los cables de interconexión entre ordenadores personales y periféricos, así como entre otros dispositivos portátiles como teléfonos móviles, ordenadores portátiles y agendas electrónicas (PDA). Permite alcanzar velocidades de transmisión de 1 Mbps en su modalidad básica, y de hasta 3 Mbps en su versión de velocidad aumentada o **EDR** (*Enhanced Data Rate*) [7]. Su desarrollo ha traído consigo la aparición del concepto de **Redes de Área Personal (PAN)**, en las que las personas interactúan con las diversas tecnologías de la información de su entorno de un modo totalmente novedoso [1].

El estándar Bluetooth surgió a mediados de los noventa a partir de un estudio realizado por Ericsson con el objetivo de evaluar las posibilidades de interconectar sus teléfonos móviles con diversos accesorios mediante enlaces de radio. Posteriormente otras compañías como Intel, IBM, Toshiba, o Nokia se unieron a Ericsson para fundar una organización denominada **SIG** (*Special Interest Group*), con el objetivo de desarrollar la especificación del estándar Bluetooth y promover el uso y aplicación de esta tecnología [1]. Hoy en día este grupo engloba a más de 2000 empresas de todo el mundo.

La **especificación** del estándar Bluetooth es una especificación abierta y global que define completamente a cualquier sistema Bluetooth, desde el nivel físico hasta el nivel de aplicación. Para ello se define una pila de protocolos similar al modelo **OSI** [1]. En la figura 3.1 se muestra un posible esquema de la arquitectura de un sistema Bluetooth, donde se indican los protocolos y capas más importantes. Generalmente dentro de los sistemas Bluetooth estas capas se agrupan en **Módulo** y **Host**. El Módulo se implementa en hardware y firmware, mientras que el Host se realiza normalmente en software. El primero incluye las capas de **Radio**, **Bandabase**, Dirección de Enlace o **LM** (*Link Manager*) y una parte de la capa **HCI** (*Host Controller Interface*), que sirve de interfaz entre el Módulo y el Host. Este último abarca desde HCI hasta el nivel de aplicación. Entre estas capas se encuentran algunos de los protocolos más importantes definidos por la especificación Bluetooth, como son **L2CAP** (*Logical Link Controller and Adaptation Protocol*), **SDP** (*Service Discovery Protocol*) o **RFCOMM**, que proporciona los medios para emular en software un puerto serie [1,7]. Esta subdivisión

entre Host y Módulo no es un requisito que se ha de cumplir obligatoriamente, ya que existen dispositivos Bluetooth muy simples como auriculares inalámbricos que son implementados completamente en hardware [1].

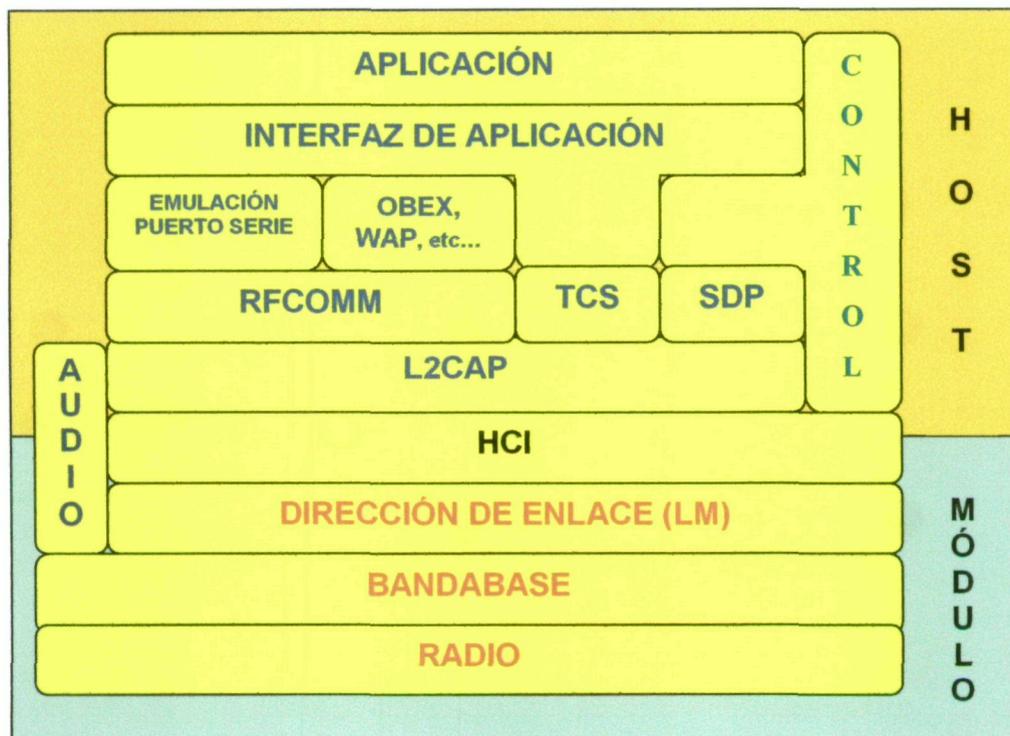


Figura 3.1 Arquitectura y pila de protocolos de la tecnología Bluetooth

En función del ámbito específico de aplicación de la tecnología Bluetooth, la especificación del estándar describe la forma de desarrollar e implementar esta tecnología para cada aplicación concreta. A esta descripción se la denomina **Perfil Bluetooth**, y define entre otras cosas los elementos de la pila que son necesarios implementar en cada caso. Existe una enorme variedad de perfiles Bluetooth predefinidos [7]. En el caso concreto de este proyecto se hará uso del perfil **puerto serie** (*Serial Port Profile*).

3.2.2 Elementos característicos de las redes Bluetooth

Las redes formadas por dispositivos Bluetooth poseen una serie de características que las diferencian enormemente de otro tipo de redes convencionales. En primer lugar hay que considerar el aspecto de red *ad-hoc*. Esto significa que se trata de redes que se crean y configuran dinámicamente cuando dos o más dispositivos

Bluetooth se aproximan entre sí, para realizar una un intercambio de información entre ellos [1]. Esta gran flexibilidad requiere de un estándar robusto y ciertamente más complejo que el de las redes tradicionales, pero de forma que dicha complejidad sea transparente al usuario, con el fin de que esta tecnología sea lo más accesible y cómoda posible.

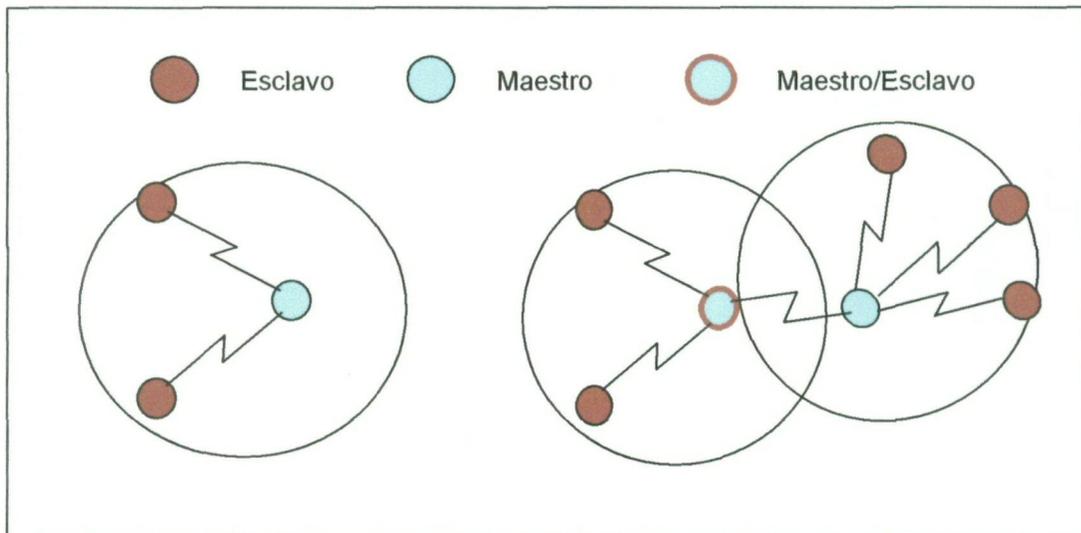


Figura 3.2 Piconet con dos esclavos y scatternet con maestro/esclavo

Dentro de una red formada por dos o más dispositivos Bluetooth, cada uno de estos adoptan un papel determinado: el de **Maestro** o **Esclavo**. El Maestro es el dispositivo que solicita el establecimiento de la conexión, y el Esclavo el que responde. Por tanto, dos Esclavos no pueden conectarse entre sí. El estándar Bluetooth define, además, dos formas de interconectar los dispositivos: **Punto a Punto** (*Point to Point*), cuando en la red existen solamente un Maestro y un Esclavo, y **Punto a Multipunto** (*Point to Multipoint*), cuando existen un Maestro y dos o más Esclavos [1,7].

Por otro lado, existen dos clases de topologías que pueden adoptar las redes Bluetooth en función del papel desempeñado por el Maestro [1,7]:

- **Piconet:** se da cuando el Maestro está conectado a al menos un Esclavo. Como máximo puede estarlo hasta con siete Esclavos, de forma que una piconet puede estar formada por un máximo de ocho dispositivos.
- **Scatternet:** una scatternet está formada por dos o más piconets. Esto sucede cuando el Maestro de una piconet se convierte en Esclavo de otra piconet distinta. Por tanto dicho dispositivo desempeña simultáneamente los roles de Maestro y Esclavo. También es posible que un Esclavo pertenezca a dos piconets distintas. Esto se consigue mediante un esquema de multiplexación temporal.

3.3 Radio

3.3.1 Introducción

Los dispositivos Bluetooth operan en la banda **ICM** (industrial, médica y científica) o **ISM** (*Industrial, Scientific and Medical*) de 2,4 GHz. Esta banda está disponible sin necesidad de licencia administrativa para su uso por parte de dispositivos que cumplan una serie de requerimientos en cuanto a potencia, características espectrales y de interferencia, especificados por el ETSI ETS 300-328 en Europa o el FCC CFR47 Part15 en Estados Unidos [1,7]. España, en concreto, armonizó su legislación con el resto de países europeos mediante la ORDEN CTE/630/2002 del CNAF (Cuadro Nacional de Atribución de Frecuencias), publicada en el BOE núm. 70 del 22 de marzo de 2002 [8].

La banda ICM es empleada por dispositivos tales como teléfonos y auriculares inalámbricos, alarmas de coches, o el propio estándar **WLAN** (*Wireless Local Area Networking*). Esto implica que esta banda de frecuencias sea proclive a experimentar la presencia de ruido e interferencias debido a la multitud de dispositivos que hacen uso de ella [1,7]. Bluetooth soluciona este problema mediante el uso de técnicas de espectro ensanchado por salto de frecuencias o **FHSS** (*Frequency Hopping Spread Spectrum*), control adaptativo de potencia y uso de paquetes de datos cortos [1].

En la tabla 3.1 se presentan los valores de los parámetros espectrales más característicos de la tecnología Bluetooth.

Banda ICM (GHz)	Banda de guarda inferior	Banda de guarda superior	Número de canales disponibles	Ancho de banda de canal
2,4000 – 2,4835	2 MHz	3,5 MHz	79	1 MHz

Tabla 3.1 Valores espectrales característicos de Bluetooth

Con estos valores se tiene que el primero de los 79 canales está centrado en 2,402 GHz y el último en 2,480 GHz.

3.3.2 Espectro ensanchado por salto de frecuencia

La técnica de espectro ensanchado por salto de frecuencia, más conocida por su terminología inglesa de frequency hopping o su acrónimo **FHSS** (*Frequency Hopping Spread Spectrum*) [2], tuvo su origen durante la II Guerra Mundial, y permite conseguir una comunicación segura y relativamente inmune a interferencias entre un transmisor y un receptor. Consiste, básicamente, en ir saltando periódicamente de un canal a otro siguiendo un patrón determinado por un **código pseudo-aleatorio**. Evidentemente tanto el transmisor como el receptor deben conocer este código de antemano para poder sincronizarse. Si en un momento dado el canal que se está usando sufre una interferencia producida por otro dispositivo, es poco probable que el siguiente canal al que se salte también sufra dicha interferencia. En concreto, Bluetooth utiliza un algoritmo de generación del código pseudo-aleatorio que maximiza la separación, dentro de la secuencia de salto, de canales adyacentes. Además se minimiza la probabilidad de interferencia entre piconets diferentes, ya que dos piconets próximas entre sí, cada una con su propia secuencia de salto característica, pueden colisionar al hacer uso del mismo canal simultáneamente [1].

Por otro lado la técnica de espectro ensanchado por salto de frecuencia se divide en dos tipos:

- **Salto de frecuencia rápido** (o *Fast Frequency Hopping*): el tiempo entre dos saltos de frecuencia consecutivos es **menor** que el tiempo de duración de bit. Es decir, cada bit de datos se envía por dos o más canales diferentes.
- **Salto de frecuencia lento** (o *Slow Frequency Hopping*): el tiempo entre dos saltos de frecuencia consecutivos es **mayor** que el tiempo de duración de bit. Por tanto por cada canal se transmiten dos o más bits de datos. Este es el tipo de salto de frecuencia que utiliza Bluetooth. De hecho, algunos tipos de paquetes de datos Bluetooth son transmitidos completamente por un solo canal.

3.3.3 Modulación

Bluetooth, en su especificación más reciente [7], define hasta tres tipos de modulaciones distintas en función de la velocidad de transmisión que se desee alcanzar. En su versión básica utiliza una modulación denominada **GFSK** (*Gaussian Frequency Shift Keying*). Consiste básicamente en una modulación 2-FSK que hace uso de un filtro Gaussiano para suavizar las transiciones bruscas de frecuencia. Debido a que los canales tienen un ancho de banda de un 1 MHz, con este esquema se alcanza una velocidad de transmisión a nivel de radio de 1 Mbps.

La modalidad de velocidad aumentada o EDR permite obtener velocidades de hasta 2 y 3 Mbps. Esto se consigue con esquemas de modulación basados en **PSK** (*Phase Shift Keying*), de forma que cada símbolo transmitido codifica varios bits de datos. Para alcanzar los 2 Mbps se recurre a una modulación $\pi/4$ -**DQPSK**, mientras que los 3 Mbps se consiguen con una modulación **8DPSK**. La modalidad de velocidad básica se mantiene principalmente para permitir compatibilidad hacia atrás con los dispositivos Bluetooth más antiguos [7].

3.3.4 Potencia emitida

En la especificación Bluetooth se definen tres clases de dispositivos diferentes en función de la potencia emitida: **clase 1**, **clase 2** y **clase 3**. En la tabla 3.2 se especifican sus características. Los dispositivos de clase 1 son los que mayor potencia emiten y pueden llegar a tener un alcance de hasta 100 metros, mientras que los de clase 3 suelen tener un alcance de unos 10 metros [1]. La mayoría de los dispositivos Bluetooth existentes pertenecen a esta última clase.

Clase de potencia	Máxima salida de potencia (Pmax)	Salida de potencia nominal	Mínima salida de potencia ¹	Control de potencia
1	100 mW (20 dBm)	---	1 mW (0 dBm)	Pmin < +4dBm a Pmax Opcional: Pmin ² a Pmax
2	2,5 mW (4 dBm)	1 mW (0 dBm)	0,25 mW (-6 dBm)	Opcional: Pmin ² a Pmax
3	1 mW (0 dBm)	---	---	Opcional: Pmin ² a Pmax

1. Potencia de salida mínima para configuración de máxima potencia.
2. Se aconseja un nivel de potencia mínima Pmin < -30 dBm, aunque no es obligatorio ya que puede elegirse en función de la aplicación.

Tabla 3.2 Clases de dispositivos en función de la potencia.

En cuanto a los dispositivos pertenecientes a la clase 1, éstos están obligados a implementar un sistema de control de potencia adaptativo denominado **RSSI** (*Received Signal Strength Indicator*). Este sistema se basa en la monitorización por parte del receptor de la potencia de señal recibida, y el posterior envío de paquetes de comandos al transmisor solicitando el aumento o disminución en el nivel de potencia transmitida. Este sistema no es obligatorio para los dispositivos de clase 2 y clase 3, aunque se recomienda que lo integren.

3.4 Bandabase

El nivel de bandabase juega un papel fundamental dentro de la arquitectura de cualquier sistema Bluetooth. Lleva a cabo la tarea de controlar los **saltos de frecuencia** y realiza la **codificación** o **decodificación** de los paquetes transmitidos o recibidos. También tiene la misión de mantener la **sincronización temporal** dentro del enlace físico que, como se verá más adelante, están basados en **multiplexación por división del tiempo** [1].

3.4.1 Elementos característicos de bandabase

A continuación se describirán algunos de los elementos más importantes que hay que tener en cuenta para poder entender el funcionamiento de la capa de bandabase.

3.4.1.1 Dirección de dispositivo Bluetooth

Antes de seguir adelante, es conveniente definir en este punto el concepto de **dirección de dispositivo Bluetooth (BD_ADDR)**. Se trata de una dirección **IEEE MAC** de 48 bits que es única para cada dispositivo. Se divide en **NAP** (*Non-significant Address Part*, BD_ADDR[47:32]), **UAP** (*Upper Address Part*, BD_ADDR[31:24]) y **LAP** (*Lower Address Part*, BD_ADDR[23:0]) [1].

En una **piconet** la secuencia de saltos sobre los 79 canales disponibles viene determinada por la dirección Bluetooth del **maestro**. Recuérdese que una piconet consistía en un conjunto de dispositivos Bluetooth (hasta 8 como máximo), donde uno de ellos actuaba como maestro, siendo éste el dispositivo encargado de iniciar el establecimiento de la conexión con el resto de los dispositivos, es decir los **esclavos**.

3.4.1.2 Multiplexación por división del tiempo

Bluetooth emplea para establecer enlaces bidireccionales entre maestro y esclavos (recuérdese que los esclavos no se comunican entre sí) la técnica de **multiplexación por división del tiempo (MDT)**. Para ello divide el tiempo en **slots** de 625 μ s de duración, coincidiendo cada salto de frecuencia con el comienzo de cada slot, de forma que se producen unos 1600 saltos por segundo. Los distintos paquetes

existentes a nivel de bandabase se transmiten completamente en 1, 3 ó 5 slots. Además cada paquete distinto se transmite **completamente** por un **solo** canal. Es decir que si se empieza a transmitir un paquete que necesita más de un slot de tiempo, no se producirá ningún salto de frecuencia aunque comience un nuevo slot. En la figura 3.3 se muestra este esquema de funcionamiento. El esquema MDT adoptado exige que cada slot o slots empleados en transmitir un paquete desde el maestro al esclavo estén seguidos de al menos un slot para que el esclavo confirme la correcta recepción, o no, de los datos transmitidos [1].

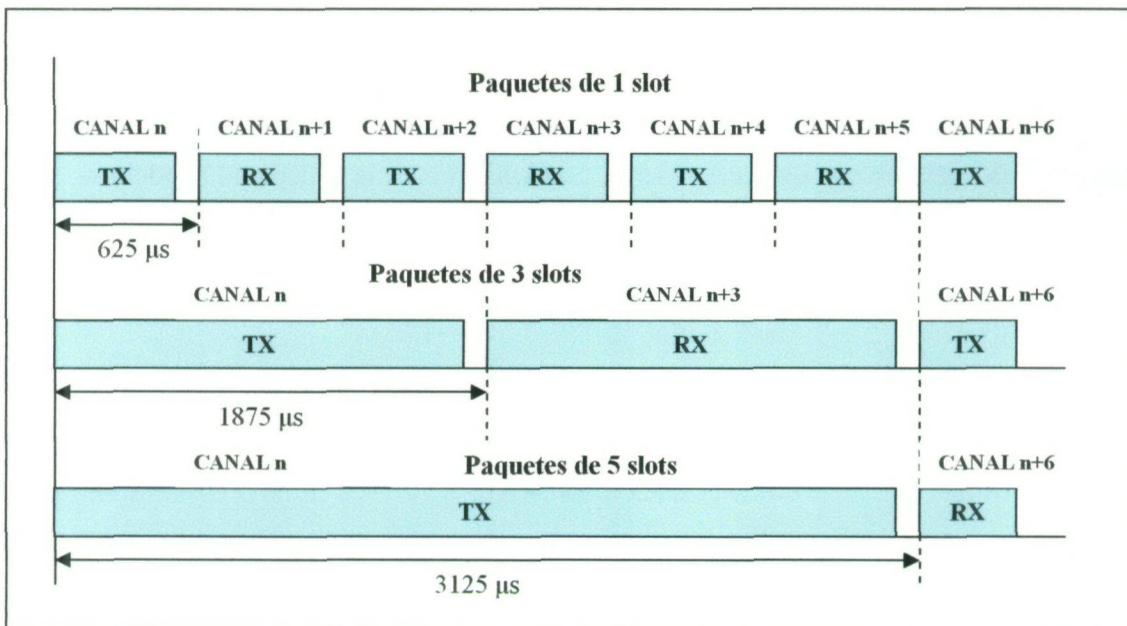


Figura 3.3 Asignación de slots a paquetes de distinto tamaño

3.4.1.3 Sistema de tiempos

Este se basa en un **contador** interno de 28 bits denominado **CLKN**. Este reloj se pone a cero cada vez que se enciende el dispositivo. El maestro de una piconet utiliza su propio CLKN para controlar el comienzo de cada slot y los saltos de frecuencia. Los esclavos, para sincronizarse con el reloj CLKN del maestro, realizan una estimación del mismo sumando un determinado valor a su propio CLKN. Esta estimación se denomina **CLK**. Por otro lado, cuando el maestro quiere conectarse con un esclavo que no esté sincronizado, también realiza una estimación del CLKN del esclavo sumando un valor arbitrario a su propio CLKN. Esta estimación se denomina **CLKE** [1].

3.4.1.4 Tipos de enlaces

Existen dos tipos de enlaces en función de los datos que pueden ser intercambiados por los dispositivos [1]:

- **ACL:** se trata de un enlace apropiado para la transmisión esporádica de datos. Permite comprobación de errores en los paquetes y su retransmisión si fuera necesario.
- **SCO:** este tipo de enlaces se usa para el intercambio regular y periódico de de datos. Para ello hace uso de slots reservados exclusivamente para SCO, de forma que tienen el ancho de banda garantizado. Son apropiados para la transmisión de datos en tiempo real como, por ejemplo, audio. Bluetooth permite un máximo de tres enlaces SCO dentro de una piconet, ya sea con uno o varios esclavos.

3.4.2 Estructura y tipos de paquetes Bluetooth

En la figura 3.4 se muestran los campos que conforman un paquete a nivel de bandabase.

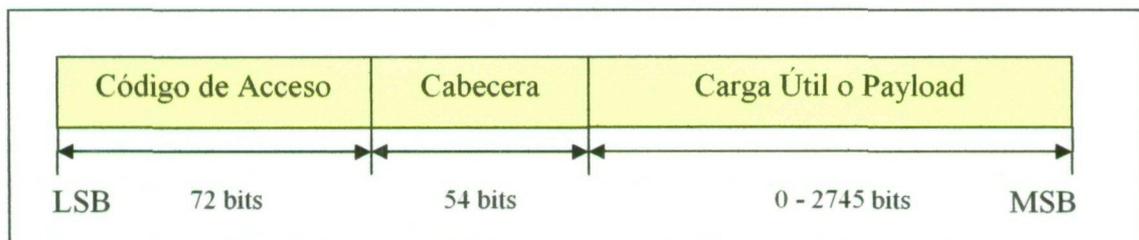


Figura 3.4 Estructura de un paquete Bluetooth.

En primer lugar está el **código de acceso**. Existen varios tipos de códigos de acceso. El **código de acceso de canal** (o *Channel Access Code*, **CAC**) se obtiene a partir de la dirección Bluetooth del maestro, y se utiliza para identificar los paquetes pertenecientes a una determinada piconet durante la operación normal del enlace. El **código de acceso de descubrimiento general** (o *General Inquiry Acces Code*, **GIAC**) es usado durante la fase de descubrimiento, y es un valor predefinido por el estándar.

Finalmente está el **código de acceso de dispositivo** (o *Device Access Code*, **DAC**), que se obtiene a partir de la dirección Bluetooth del esclavo y se emplea en el procedimiento de *paging* [1].

A continuación se encuentra la **cabecera**. Este campo contiene, entre otros parámetros, un identificador de esclavo (o *Active Member Address*, **AM_ADDR**). Éste permite identificar al esclavo al que se envía un paquete, o del cual procede el mismo. Además la cabecera incluye un campo de **CRC** (*Cyclic Redundant Check*) para detectar posibles errores [1].

La **carga útil** o **payload** puede transportar tanto datos ACL como SCO. Este campo también está protegido por un campo de CRC [1].

En cuanto a la clasificación de los paquetes Bluetooth, existen dos clases. En primer lugar están los paquetes de datos. Dentro de estos hay varios tipos en función de los datos que transportan (ACL, SCO o mixto) y del número de slots que abarcan. Son los paquetes DM1, DH1, DM3, DH3, DM5, DH5, AUX1, HV1, HV2 y HV3 [1,7].

Por otro lado, existen una serie de paquetes especiales usados para el mantenimiento y señalización del enlace. Éstos son:

- **ID**: contiene solamente el código de acceso.
- **NULL**: formado por el código de acceso y la cabecera. Se utiliza para confirmar la recepción de otros paquetes y para control de flujo. El propio paquete NULL no necesita confirmación.
- **POLL**: igual que NULL, pero requiere confirmación.
- **FHS**: se utiliza para sincronizar el salto de frecuencia. Para ello incluye información de la dirección Bluetooth y del reloj (CLKN) del dispositivo que lo envía.

En el siguiente apartado se indicará como se utilizan estos paquetes especiales durante las fases de descubrimiento y conexión entre dos dispositivos Bluetooth.

3.4.3 Controlador de enlace y estados de operación

Dentro de la capa de bandabase merece especial atención el **controlador de enlace** (o *Link Controller, LC*). Se encarga de gestionar el establecimiento y mantenimiento de los enlaces determinando los paquetes que deben ser transmitidos por bandabase en cada caso. El controlador de enlace se implementa mediante una **máquina de estados** [1]. Estos estados se relacionan con las distintas fases o etapas en las que puede encontrarse cualquier dispositivo Bluetooth, como se muestra en la figura 3.5. Los estados más importantes son:

- **Inactivo** (o *Standby*): el dispositivo está inactivo, no se transmiten datos y el subsistema de radio está apagado.
- **Descubrir** (o *Inquiry*): en este estado el dispositivo, que actúa como **maestro**, intenta descubrir otros dispositivos Bluetooth activos que estén cerca. Para ello transmite reiteradamente paquetes **ID**, que incluyen el código de acceso de descubrimiento general (**GIAC**), hasta que obtiene respuesta. La secuencia de salto de frecuencia se deriva del **GIAC**, pero se realiza a una velocidad el doble de la habitual.
- **Descubrirse** (o *Inquiry Scan*): un dispositivo en este estado asume el papel de **esclavo**, intentando ‘escuchar’ por si algún otro dispositivo en estado **descubrir** transmite paquetes **ID**. Si es así responde con un paquete **FHS**, donde incluye su **dirección Bluetooth** y el valor de su reloj **CLKN**. La secuencia de salto se obtiene también a partir del **GIAC**, pero se realiza a una velocidad mitad de la normal. De esta forma el maestro, cuya velocidad de salto es el doble, y el esclavo acaban coincidiendo en un mismo canal.
- **Conocer** (o *Page*): el maestro transmite un paquete **ID** con el **código de acceso de dispositivo (DAC)** del esclavo, obtenido a partir de su dirección Bluetooth. Para sincronizarse con la secuencia de salto del esclavo el maestro utiliza el **CLKE**, que es la estimación del **CLKN** del esclavo.
- **Darse a Conocer** (o *Page Scan*): el esclavo en este estado espera recibir un paquete **ID** con su **DAC**, y cuando lo recibe responde con un paquete **ID** con el mismo **DAC**. Entonces el maestro contesta con un paquete **FHS** donde informa sobre su **dirección Bluetooth** y su **CLKN**.

- Conexión (o Connection):** A partir de la **dirección Bluetooth** del maestro y de su **CLKN** el esclavo obtiene el **código de acceso de canal (CAC)** y la secuencia de salto que deben usar todos los dispositivos de la piconet. También realiza una estimación del **CLKN** del maestro, es decir **CLK**, para poder sincronizarse temporalmente con el maestro y el resto de dispositivos de la piconet. Además el esclavo recibe su **dirección de miembro activo (AM_ADDR)** o identificador de esclavo. A continuación, el maestro transmite un paquete **POLL** y el esclavo responde con un paquete **NULL**. Así se verifica que el enlace ha sido establecido satisfactoriamente. En adelante el esclavo ‘escucha’ todos los paquetes que transmite el maestro por si alguno va dirigido a él. Además en caso de que el esclavo no tenga nada que transmitir, éste debe seguir enviando paquetes **NULL** periódicamente, para mantener la sincronización con el maestro.

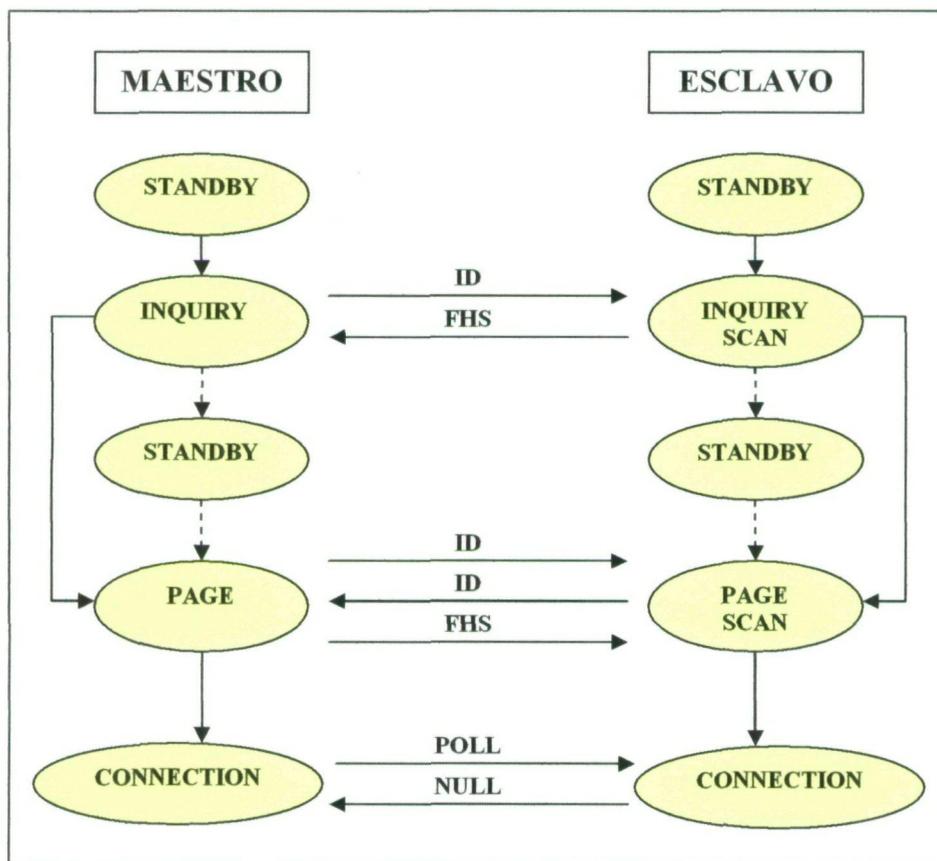


Figura 3.5 Estados de operación del controlador de enlace

3.5 Dirección de enlace

La capa de **dirección de enlace** (o *Link Manager*, **LM**) se encarga de gestionar la piconet **creando**, **configurando** y, llegado el caso, **destruyendo** los enlaces físicos con otros dispositivos. También controla las funciones de encriptación de datos implementadas en bandabase y lleva a cabo la **autenticación** de los dispositivos que se incorporan a la piconet [1].

3.5.1 Mensajes LMP y canal lógico de dirección de enlace

Para realizar las tareas que lleva a cabo **LM**, existen una serie de procedimientos establecidos que conforman el **protocolo de dirección de enlace** (o *Link Manager Protocol*, **LMP**), basado en el intercambio de **mensajes** o **PDU**s (*Protocol Data Units*) por parte de los **LM** de cada dispositivo. Aunque este intercambio de mensajes se produce exclusivamente a nivel de **LM**, éste puede estar desencadenado por una orden procedente de capas superiores.

Cada mensaje **LMP** comienza con un **identificador** de 1 bit en función de si es el maestro o el esclavo el que inicia el intercambio de mensajes. A continuación viene un **código de operación** que identifica el tipo de mensaje **LMP**. Seguidamente se encuentran los posibles **parámetros** que pueda ser necesario incluir.

Los mensajes **LMP** se transmiten mediante paquetes de un solo slot, utilizando el **canal lógico de dirección de enlace**. Este canal es distinto del utilizado para la transmisión de los paquetes que transportan datos de procedentes de capas superiores. Los mensajes **LMP** se distinguen a nivel de bandabase de los paquetes de datos **ACL** o **SCO** usando un identificador especial incluido dentro del campo de *payload* denominado **L_CH**.

3.5.2 Funciones desempeñadas por la dirección de enlace

3.5.2.1 Establecimiento del enlace

La capa de dirección de enlace se encarga de crear enlaces ACL mediante órdenes enviadas a la capa de bandabase. Una vez establecido el enlace ACL, el LM puede gestionar el establecimiento de un enlace SCO. Durante el establecimiento del enlace, a los esclavos que pasan a formar parte de la piconet, se les asigna la **dirección de miembro activo (AM_ADDR)**, como ya se ha comentado en el apartado que describe la capa de bandabase [1].

3.5.2.2 Intercambio de papeles

Bluetooth permite que el maestro de piconet se convierta en esclavo, y simultáneamente, que uno de los esclavos se convierta en maestro. El maestro original sigue siendo el maestro del resto de los esclavos, si estos existieran [1,7]. Este procedimiento se describe en la figura 3.6.

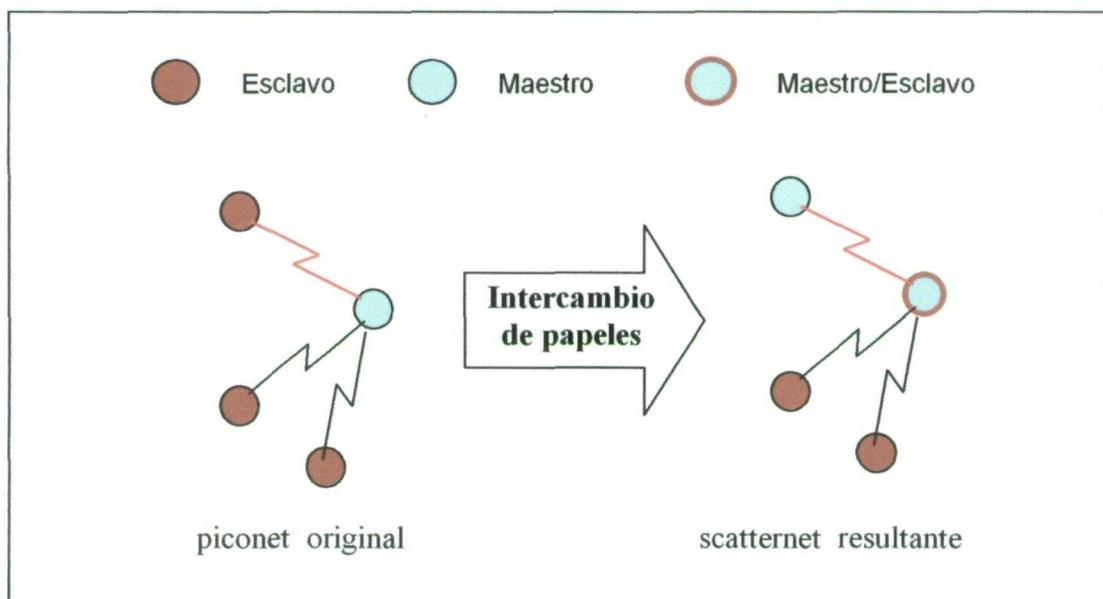


Figura 3.6 Intercambio de papeles maestro-esclavo.

3.5.2.3 Control de potencia

En los dispositivos Bluetooth se hace necesario mantener la potencia emitida por la capa de radio en los niveles más bajos posibles. De esta forma se consigue alargar la vida útil de las baterías, así como minimizar las interferencias con otras piconets cercanas. Bluetooth permite al dispositivo receptor solicitar cambios en los niveles de potencia emitida. Esta técnica se implementa mediante el intercambio de mensajes LMP [1].

3.5.2.4 Otras funciones

La capa de dirección de enlace tiene otras funciones aparte de las ya descritas como son:

- **Seguridad:** el LM se encarga de configurar y gestionar la encriptación de datos realizada en bandabase.
- **Calidad de servicio:** existen mensajes LMP destinados a configurar la calidad de servicio de una conexión. También es posible variar la velocidad de transferencia en función de la calidad del canal cambiando el tamaño de los paquetes utilizados.
- **Características soportadas:** puesto que algunas de las características definidas por el estándar Bluetooth son opcionales, existen mensajes LMP destinados a intercambiar información sobre dichas características.
- **Requerimiento del nombre:** todos los dispositivos Bluetooth tienen nombres “amigables” codificados con 248 bytes. Por esta razón existen mensajes LMP específicos para que un dispositivo Bluetooth “pregunte” a otro su nombre.

3.6 Interfaz controlador de host

El **interfaz controlador de host** (o *Host Controller Interface, HCI*) proporciona un método de comunicación estándar entre host y módulo. HCI permite **controlar** los recursos **hardware** del módulo, además de **transportar** los **datos** que son transmitidos o recibidos. En aquellos dispositivos en los que no se realice la abstracción de host y módulo, la capa HCI no es necesaria. Puesto que la capa HCI se divide entre el host y el módulo, se necesita algún tipo de elemento que los interconecte físicamente. Para ello se define el **bus físico** o **capa de transporte HCI**, que puede implementarse de varias formas. En la especificación Bluetooth se define el uso de tres tipos de capas de transporte: **USB, RS-232 y UART** [1,7].

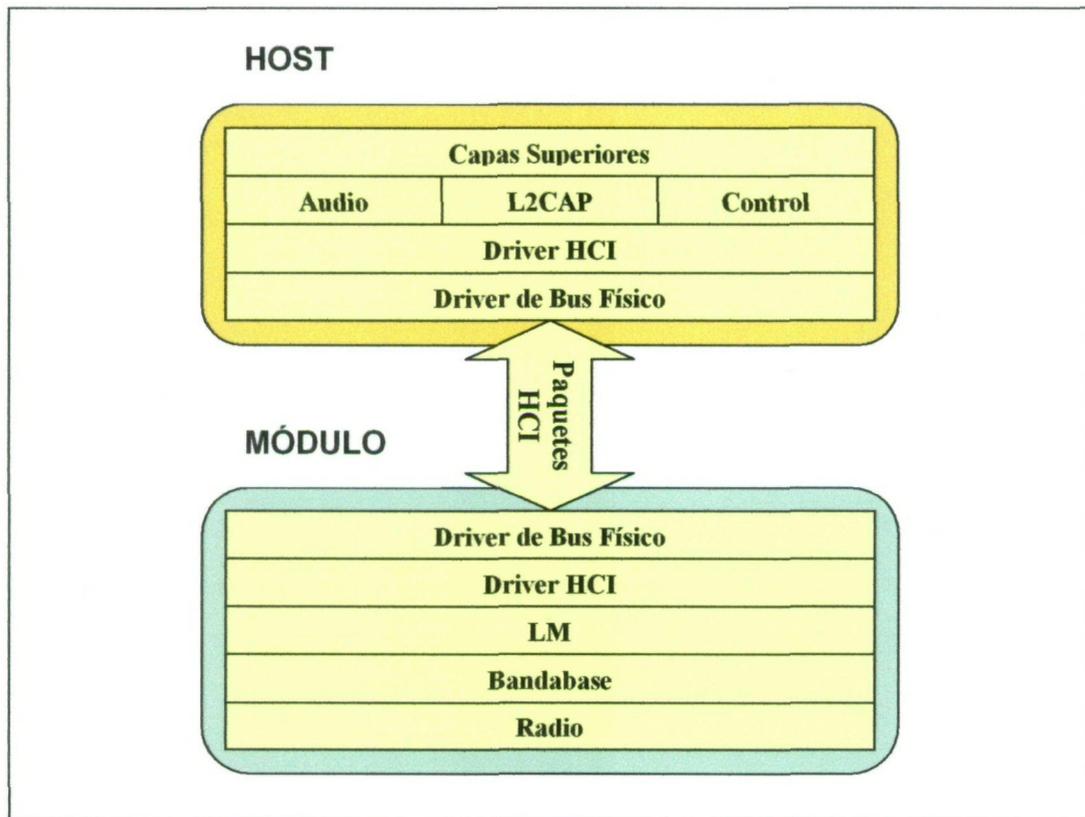


Figura 3.7 Posición del Interfaz Controlador de Host en la pila de protocolos Bluetooth

3.6.1 Paquetes HCI

El estándar Bluetooth define tres tipos de paquetes HCI: **comandos**, **eventos** y **paquetes de datos**.

3.6.1.1 Comandos HCI

Los paquetes HCI son paquetes especiales utilizados por el host para controlar el módulo Bluetooth y monitorizar su estado. En la figura 3.8 se muestra la estructura de un comando HCI.

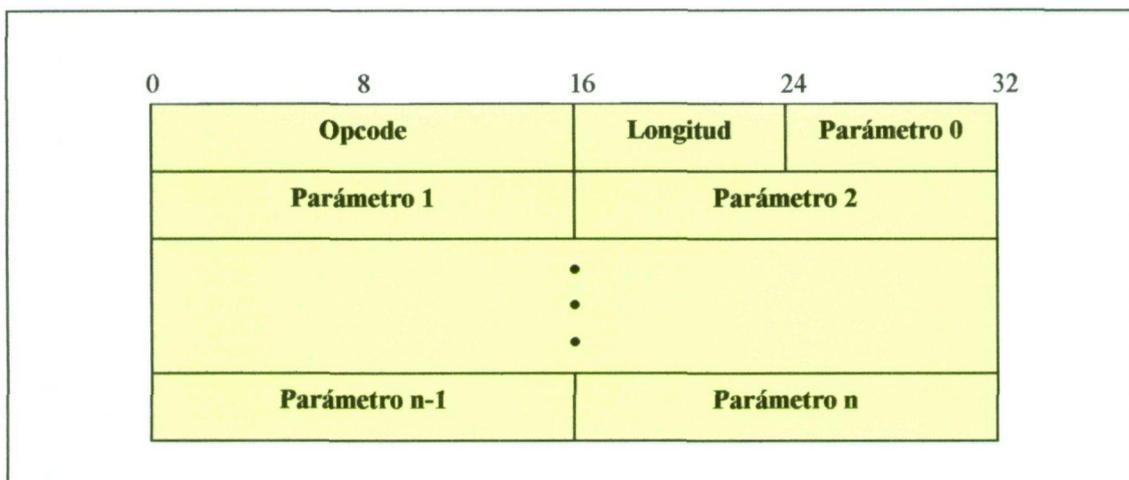


Figura 3.8 Estructura de un comando HCI

El primer campo de dicho paquete es el **código de operación** (u *Opcode*). El primer byte corresponde al *Opcode Group Field (OGF)*, que identifica el grupo al que pertenece el comando. Existen seis grupos distintos de comandos HCI en función de su finalidad. El segundo byte del *Opcode* contiene el *Opcode Command Field (OCF)*, que identifica a cada comando dentro de un determinado grupo. Al final del anexo 2 se explica la forma en que se codifica dicho campo. A continuación del *Opcode* viene un campo que indica la longitud en bytes de los parámetros que vienen a continuación.

3.6.1.2 Paquetes de datos HCI

Existen dos tipos de paquetes de datos HCI en función del tipo de información transportan. En la figura 3.9 se muestran sus formatos:

- **Paquetes de datos ACL:** el primer campo es un **identificador de conexión manejada** (o *Connection Handle*). A continuación viene el campo **PB** (*Packet Boundary*) que indica si el paquete HCI transporta en su campo de datos el

comienzo o la continuación de un paquete L2CAP. El campo **BC** (*Broadcast*) indica si el paquete corresponde a una conexión punto a punto o está dirigido a todos los esclavos. Siguen dos bytes del campo de **longitud** de datos (en bytes), y a continuación el campo de datos propiamente dicho. Los dispositivos Bluetooth deben tener buffers suficientemente grandes para almacenar paquetes de al menos 256 bytes. El tamaño máximo puede ser limitado empleando el comando **HCI_Host_Buffer_Size**.

- **Paquetes de datos SCO:** tienen una estructura similar a la de los paquetes ACL, salvo que no existen los campos PB o BC, estando su espacio reservado. Además el campo de longitud se limita a un byte, y por tanto el número máximo de bytes de datos es 256.

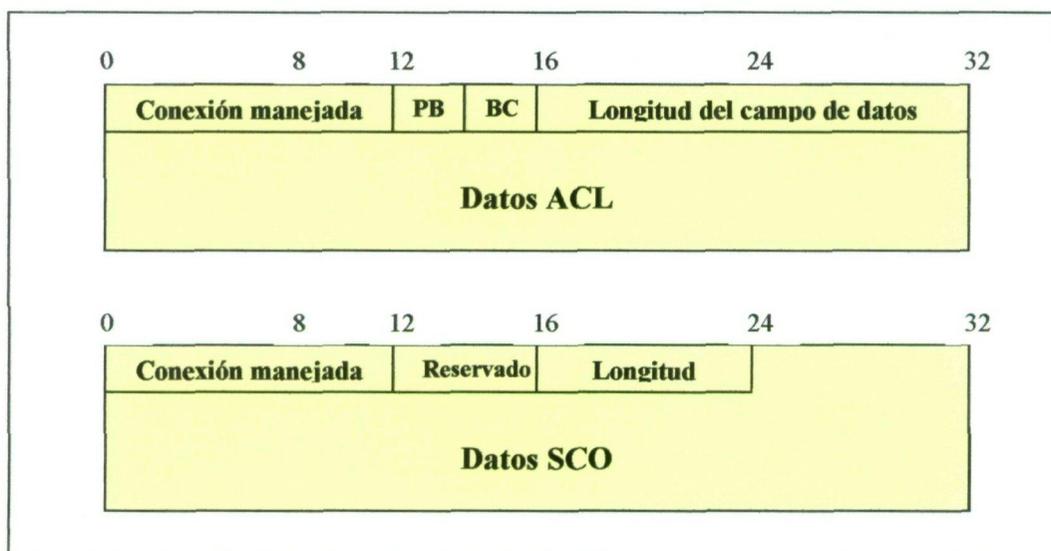


Figura 3.9 Estructura de los paquetes de datos ACL y SCO

3.6.1.3 Eventos HCI

El formato de estos paquetes se muestra en la figura 3.10. Se compone de un campo de **código de evento** (o *Event Code*) que identifica el tipo de evento. A continuación viene el campo de **longitud** que indica el número de bytes que ocupan los **parámetros** que siguen.

Los eventos HCI se producen bien como respuesta a los comandos HCI enviados al módulo, o bien como indicadores de las incidencias propias de la operación del enlace. De esta forma cuando el host envía un comando HCI al módulo, éste puede responder con un evento **HCI_Command_Complete** si el comando completa su ejecución inmediatamente, o un **HCI_Command_Status** si el comando tiene que esperar a ser “completado”.

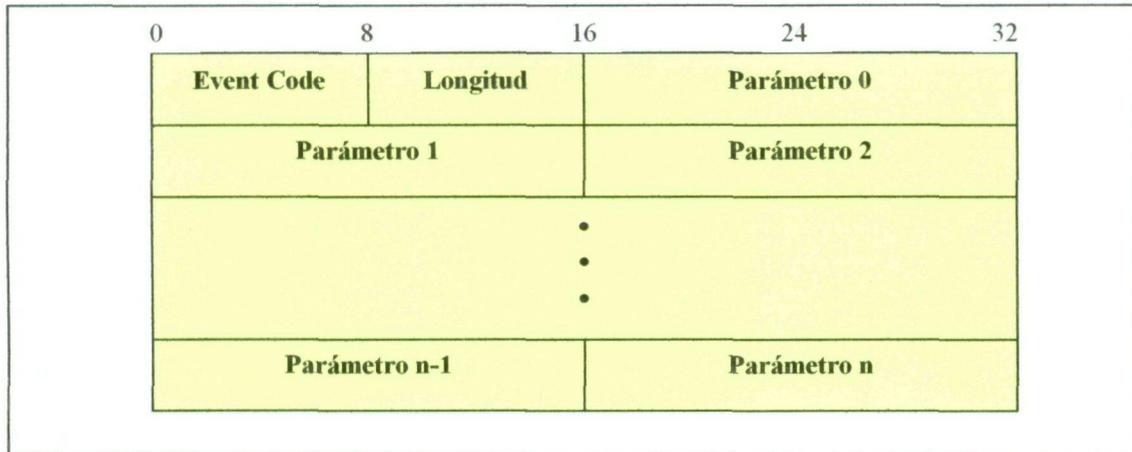


Figura 3.10 Estructura de un evento HCI

3.6.2 Capa de transporte HCI

Como ya se ha mencionado, HCI necesita una **capa de transporte** o **bus físico** para transmitir los paquetes HCI entre host y módulo. Esta capa de transporte puede ser **USB**, **RS-232** o una **UART**. En el caso de una **UART**, que es la capa de transporte utilizada en este proyecto, la especificación Bluetooth indica el uso de una etiqueta de un byte que se antepone al primer campo de cualquier paquete HCI. Para **comandos HCI** toma el valor **0x01**, para **eventos HCI** vale **0x02**, para paquetes de **datos SCO** vale **0x03** y para **datos ACL** vale **0x04**.

3.6.3 Establecimiento de una conexión

En la figura 3.11 se muestran los comandos y eventos involucrados en el establecimiento de un enlace Bluetooth durante las etapas de Descubrimiento (*Inquiring*) y Conocer (*Paging*).

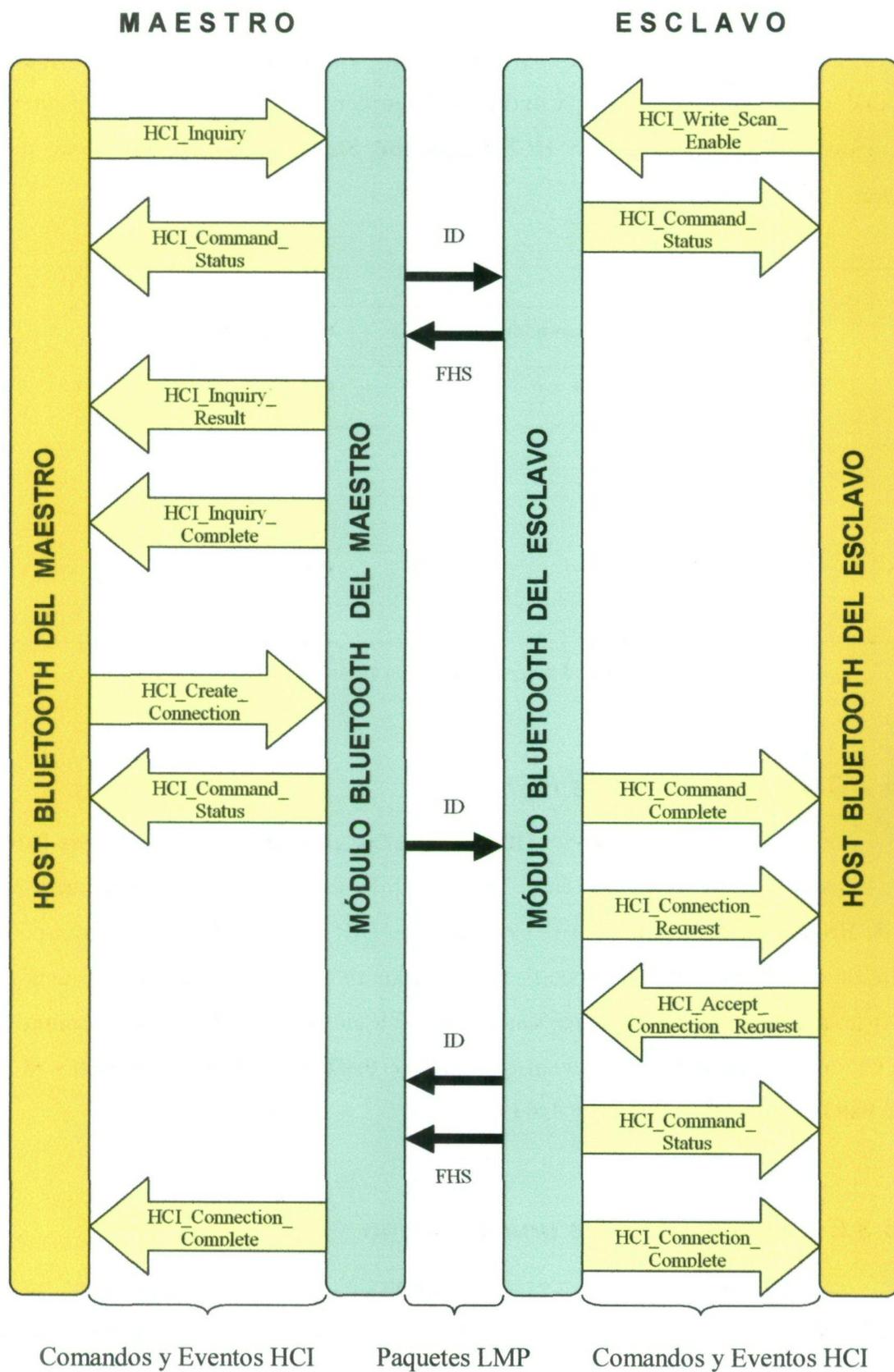


Figura 3.11 Secuencia de mensajes HCI durante el establecimiento de una conexión.

3.7 Protocolo de control de enlace lógico y adaptación

El **protocolo de de control de enlace lógico y adaptación** (o *Logical Link Control and Adaptation Protocol, L2CAP*) realiza la **multiplexación** de los diferentes protocolos de capas superiores y lleva a cabo tareas de **segmentación y reensamblaje** para permitir la transmisión de paquetes mayores de lo que pueden admitir las capas inferiores. L2CAP se basa en la utilización de conexiones ACL que deben ser establecidas previamente por una entidad de control ajena a la capa L2CAP (en la figura 3.1 se realiza esta abstracción). En cuanto a los enlaces SCO, éstos no son soportados por L2CAP [1].

3.7.1 Multiplexación mediante canales lógicos

La función de **multiplexación** que proporciona L2CAP permite que diferentes aplicaciones que usen distintos protocolos de capas superiores puedan utilizar el mismo enlace ACL [1].

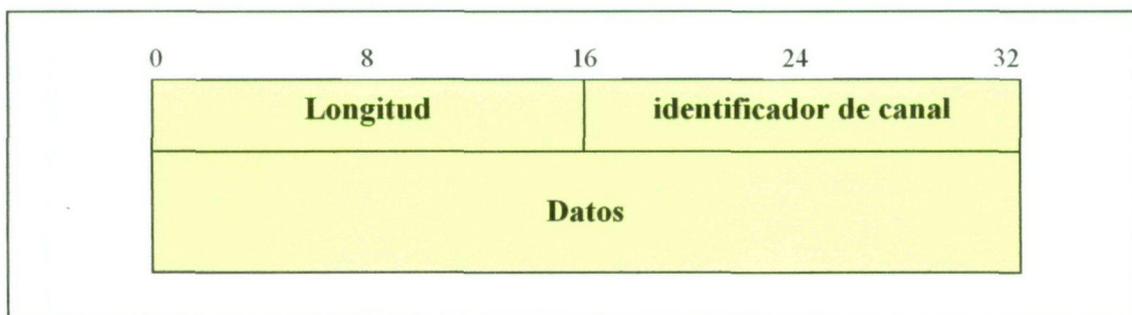


Figura 3.12 Estructura de un paquete L2CAP

L2CAP realiza la multiplexación mediante **identificadores de canal** (o *Channel Identifier, CID*) que identifican cada paquete transmitido. Así la capa L2CAP de receptor puede entregarlos correctamente a la aplicación indicada. Para cada **conexión o canal lógico** se define un **CID de origen** y un **CID de destino**, que no tienen por qué ser iguales. Durante el establecimiento de un canal lógico, el dispositivo que responde a la petición de conexión es informado del CID de origen. A continuación le asigna un CID de destino, e informa sobre él al dispositivo solicitante de la conexión. Estas parejas de identificadores son utilizados durante la configuración de la conexión y la

fase de desconexión. El **rango de valores** que pueden tomar tanto el CID de destino como el de origen está entre **0x040** y **0xFFFF** [1,7].

En la figura 3.12 se muestra el formato de un paquete L2CAP. Está formado por campo de **longitud**, el campo de **identificador de canal** y el campo de **datos**. El campo de longitud e identificador de canal se codifican en formato **Little Endian** (primero el byte menos significativo).

3.7.2 Señalización L2CAP

El **canal de señalización** L2CAP tiene asignado el identificador de canal **0x0001**, y permite realizar la **conexión**, **configuración** y **desconexión** de los enlaces a nivel L2CAP [1].

3.7.2.1 Estructuras de señalización L2CAP

Estas estructuras de señalización o comandos van alojadas en el campo de datos de los paquetes L2CAP como el de la figura 3.12. Adoptan un esquema basado peticiones y respuestas para llevar acabo las tareas encomendadas. En la figura 3.13 se muestra el formato de dichos mensajes. En primer lugar está el **código de operación** (u *Opcode*). Le sigue un identificador de 1 byte destinado a emparejar las peticiones con sus respuestas correspondientes. A continuación el campo de **longitud** indica la longitud en bytes del campo de datos, y finalmente el campo de **datos** transmitidos [1].

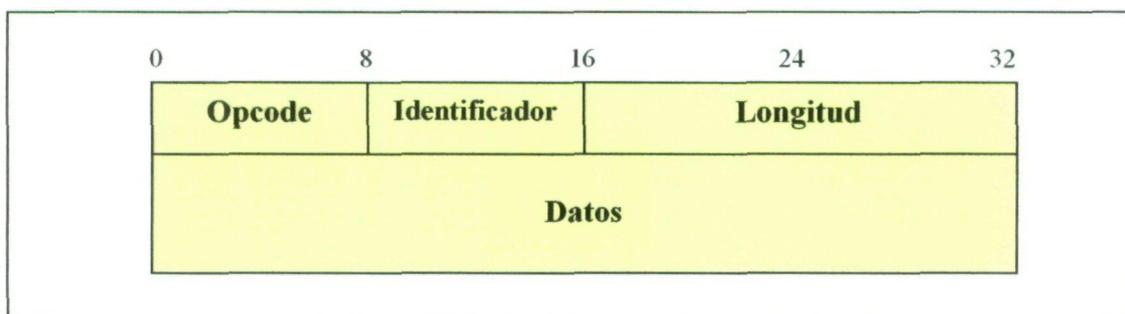


Figura 3.13 Estructura de un comando L2CAP

En total existen unas 11 estructuras de señalización distintas. Entre ellas las más importantes son [1]:

- **Command reject (*Opcode* = 0x01):** cuando el comando recibido es erróneo o resulta desconocido para el dispositivo que responde, se devuelve este comando. Incluye un parámetro denominado *Reason* que informa del motivo del rechazo del comando recibido.
- **Connection request (*Opcode* = 0x02):** sirve para solicitar la creación de un canal lógico con otro dispositivo. Tiene como parámetro el *Protocol Service Multiplexor* (PSM) que sirve para indicar cual es el protocolo de nivel superior que se va a usar sobre dicho canal L2CAP. También incluye el **CID de origen**.
- **Connection response (*Opcode* = 0x03):** es la respuesta a un comando *connection request*. Tiene dos campos, *Result* y *Status*, que informan de si la respuesta a la petición de conexión es satisfactoria o no, y además incluye el valor del CID asignado por el dispositivo que responde al canal lógico recién creado (**CID de destino**).
- **Configuration request (*Opcode* = 0x04):** una vez aceptada la petición de conexión, ésta debe ser configurada. Se negocian parámetros tales como el *Maximun Transmision Unit* (MTU) o tamaño máximo de los paquetes L2CAP a transmitir, además de otros parámetros relacionados con la calidad de servicio y tiempos máximos de espera.
- **Configuration response (*Opcode* = 0x05):** para confirmar los parámetros del *configuration request* se responde con una réplica idéntica de éste último. Si se desea negociar algún parámetro, se sustituye el valor original por el deseado. De esta forma se producen intercambios de *configuration request* y *configuration response* hasta que se llega a un acuerdo. Si no hay acuerdo se destruye el canal.
- **Disconnection request (*Opcode* = 0x06):** se utiliza para destruir un canal lógico.
- **Disconnection response (*Opcode* = 0x07):** confirmación de un comando *disconnection request*.

En la figura 3.14 se muestra un ejemplo del intercambio de comandos o estructuras de señalización llevado a cabo para establecer un canal lógico.

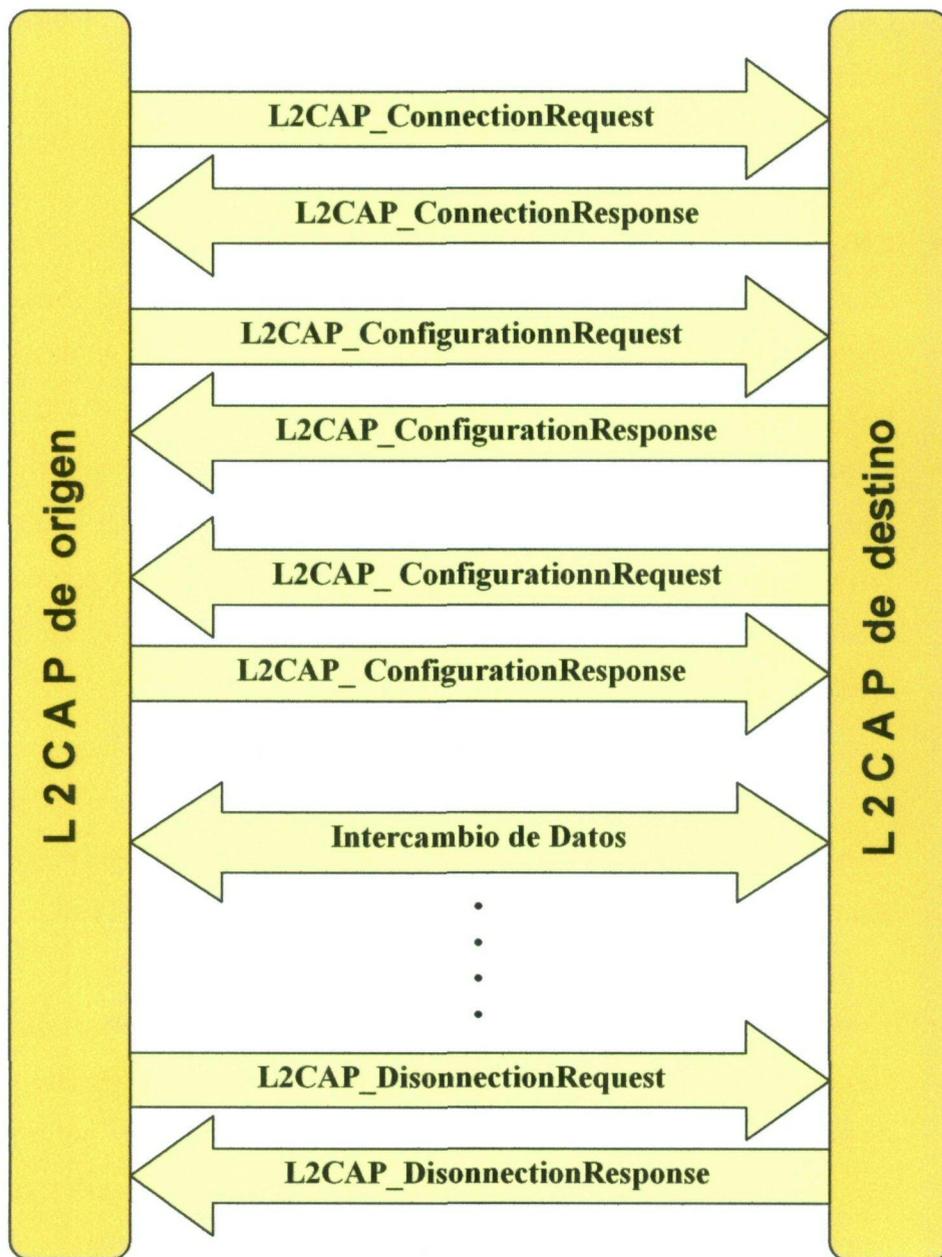


Figura 3.14 Secuencia de comandos L2CAP durante el establecimiento de una canal lógico.

3.8 RFCOMM

RFCOMM es un **protocolo de transporte fiable** que permite emular en software el estado y configuración de línea de un **puerto serie RS-232**. Además proporciona la posibilidad de establecer varios canales simultáneamente, utilizando la capacidad de L2CAP para multiplexar varias conexiones lógicas sobre el mismo enlace físico. RFCOMM no realiza tareas de corrección de errores. La responsabilidad de garantizar la integridad de los datos transmitidos recae sobre bandabase, además de llevar a cabo la gestión del control de flujo [1]. Lo que sí implementa RFCOMM son las siguientes características:

- Estado del módem: mediante las señales RTS/CTS, DSR/DTR, DCD, ring.
- Estado de la línea remota: break, overrun, parity.
- Configuración de puerto remoto: tasa de baudios, número de bits de datos, de paridad, etc.
- Negociación de parámetros: tamaño de las tramas
- Estructuración de los datos transmitidos en tramas.

En un ordenador, normalmente, la gestión de los puertos series la realiza una **UART (Universal Asynchronous Receiver Transmitter)**, para así descargar de trabajo al microprocesador. Gracias a que las UARTS son vistas por el microprocesador como posiciones de memoria, es posible emular un puerto serie en software configurando una zona de memoria de la misma forma que se haría con los registros de una UART. Además, y aunque RFCOMM permite emular el comportamiento de las líneas de un puerto serie, este protocolo trabaja realmente con datos en paralelo, de la misma forma que se produce entre un PC y una UART real [1].

3.8.1 Tipos de dispositivos RFCOMM

RFCOMM da soporte a los dos tipos de dispositivos que se muestran en la figura 3.15. Estos se definen como:

- **Tipo 1 (puerto serie emulado internamente):** entre la aplicación y RFCOMM se sitúa un módulo de emulación de puerto serie. Este módulo implementa un

interfaz de programación de aplicaciones (o *Application Programming Interface, API*) que utiliza la aplicación final para poder hacer uso de RFCOMM.

- **Tipo 2 (dispositivo intermedio con puerto serie físico):** el módulo **puerto proxy** se encarga de transmitir los datos recibidos del nivel RFCOMM a través de un puerto serie RS-232 real. En este caso el dispositivo Bluetooth es un simple adaptador Bluetooth-RS-232. Un ejemplo de esta aplicación sería la sustitución del cable RS-232 entre un PC y un módem, por dos dispositivos del tipo 2.

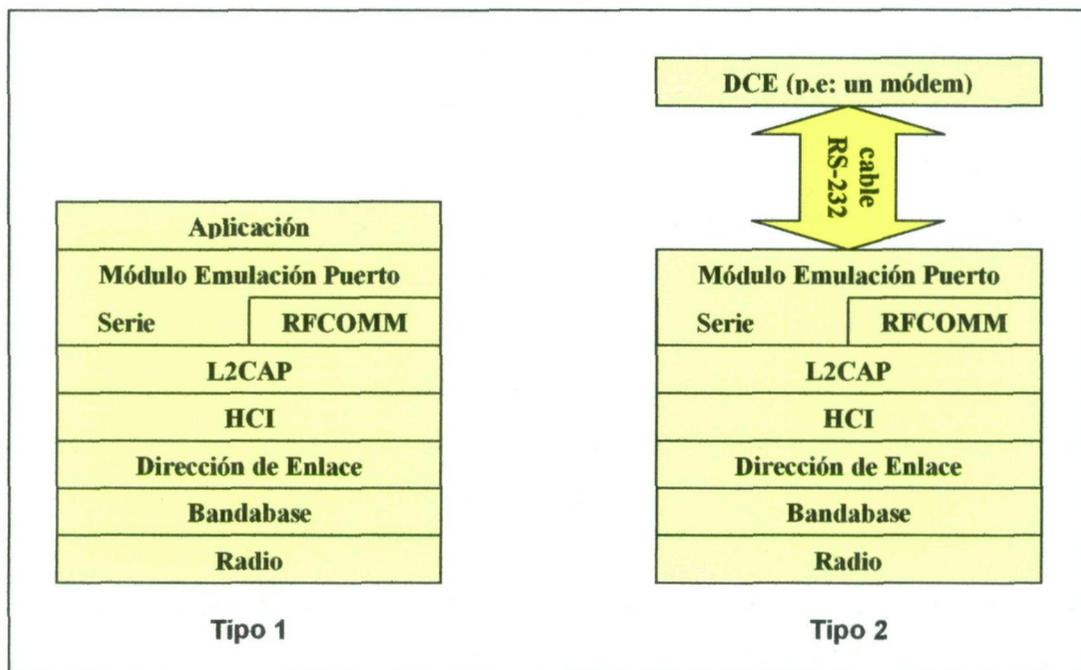


Figura 3.15 Dispositivos RFCOMM tipo 1 y tipo 2.

3.8.2 Tramas RFCOMM

RFCOMM está basado en la **opción básica de GSM TS 07.10** [1,7] que es un protocolo asimétrico usado por teléfonos móviles GSM para multiplexar varios flujos de datos sobre un mismo cable. Las tramas que utiliza RFCOMM se dividen en 5 tipos diferentes:

- **SABM (Start Asynchronous Balanced Mode):** es un **comando** utilizado para solicitar el establecimiento de una conexión.
- **UA (Unnumbered Acknowledgement):** es la **respuesta** afirmativa a una petición de conexión.
- **DISC (Disconnect):** **comando** de petición de desconexión.
- **DM (Disconnected Mode):** **respuesta** de rechazo a una petición de conexión.
- **UIH (Unnumbered Information with Header Check):** utilizada en el transporte de **datos de usuario**.

Las tramas SABM, UA, DM y DISC son **tramas de control** que se utilizan durante las fases de conexión y desconexión. Las tramas UIH normalmente llevan datos de usuario, pero también pueden llevar mensajes de control. Estos mensajes son los denominados **comandos de multiplexor** que serán vistos más adelante. Además RFCOMM permite crear varias canales simultáneamente, identificando cada conexión o canal mediante un identificador de canal (o *Data Link Connection Identifier*, **DLCI**) [1].

3.8.2.1 Estructura de las tramas RFCOMM

En la figura 3.16 se muestra la estructura de una trama RFCOMM. El primer campo es el campo de **dirección** (o *Address*). Permite identificar a cual de los diferentes canales que se multiplexan pertenece la trama. Este campo a su vez se divide en varios subcampos. El campo **EA** en RFCOMM siempre se pone a '1'. El campo **C/R** se pone a '1' en los comandos transmitidos por el dispositivo que solicita la creación de la conexión (o *iniciador*) y en las respuestas del dispositivo que responde a la solicitud de conexión. Cuando la dirección de los comandos y respuestas es la contraria, **C/R** vale '0'. El **bit de dirección (D)** siempre lo pone a '0' el *iniciador* y a '1' el que responde. El **número de canal de servidor** (o *Server Channel Number*) toma valores entre 0 y 31. El 0 se reserva para señalización y el resto permite multiplexar diferentes servicios o canales sobre RFCOMM. El bit de dirección y el número de canal de servidor conforman el **DLCI** [1,7].

El siguiente campo en la trama RFCOMM es el campo de **control**. Sirve para identificar el tipo de trama (SABM, UA, etc.). El quinto bit de este campo es el bit **P/F**. Un comando con este bit a '1' requiere ser respondido. La respuesta a dicho comando

debe también llevar el bit **P/F** a '1'. El campo de **longitud** comienza con el bit **EA**. Si vale '1', la longitud se codifica en los siguientes 7 bits. Si vale '0' se codifica en los siguientes 15 bits. El campo de **datos** sólo está presente en las tramas UIH, y debe estar formado por un número entero de bytes, de hasta 32767 como máximo. El campo **FCS** (*Frame Check Sequence*) permite detectar errores. Para las tramas SABM, DISC, UA y DM se calcula sobre los campos de dirección, control y longitud. Para la trama UIH sólo se calcula para los campos dirección y control.

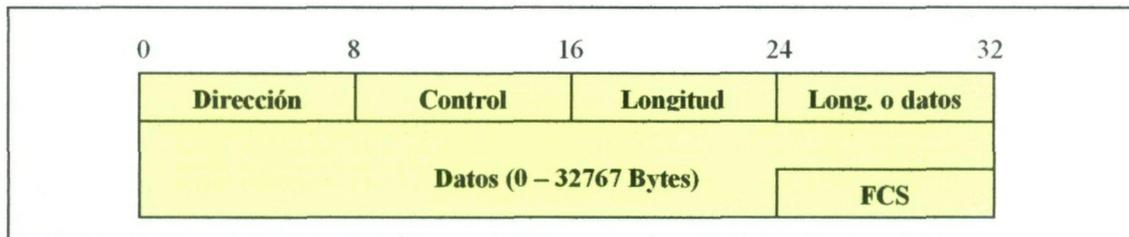


Figura 3.16 Estructura de una trama RFCOMM.

3.8.2.2 Tramas de multiplexor

Las tramas de multiplexor son tramas UIH que alojan en su campo de datos a los **comandos de multiplexor**. Estos comandos no son más que mensajes que se utilizan para controlar el enlace RFCOMM. Estas tramas utilizan un DLCI que toma el valor cero [1]. Hay 7 tipos diferentes de tramas de multiplexor:

- **PN (*DLC Parameter Negotiation*)**: permite negociar parámetros como la prioridad de un determinado canal, el tamaño máximo de las tramas de datos, etc. El procedimiento de negociación es opcional. Si no se produce se toman una serie de valores por defecto.
- **Test**: sirve para comprobar el estado de una conexión RFCOMM.
- **FCon/FCoff (*Flow Control on / Flow Control off*)**: sirve para realizar tareas de control de flujo. Se aplica simultáneamente a todos los canales.
- **MSC (*Modem Status Command*)**: permite realizar tareas de control de flujo sobre un único canal. Para ello simula las distintas señales de control de un puerto serie RS-232 mediante los parámetros incluidos en este comando.
- **RPN (*Remote Port Negotiation*)**: permite establecer parámetros de configuración propios de un puerto serie, como la velocidad de transmisión en baudios, el número de bits de datos, de paridad, etc.

- **RLS (Remote Line Status):** se utiliza para informar de errores en la comunicación al dispositivo que se encuentra en el otro extremo del enlace.
- **NSC (Non-Supported Command):** se envía cuando un dispositivo recibe un comando desconocido.

3.8.3 Establecimiento de una conexión

En la figura 3.17 se muestra el intercambio de mensajes que se da en el establecimiento de una conexión RFCOMM. Para establecer dicha conexión, el dispositivo iniciador debe haber solicitado anteriormente el **número de canal de servidor** de un determinado servicio mediante el protocolo **SDP**. Este protocolo se verá con más detalle en el punto 3.9. Una vez se dispone del canal de servidor, el iniciador debe establecer una conexión RFCOMM con $DLCI = 0$, que se utilizará para señalización mediante el intercambio de **tramas de multiplexor**. Después de esto es posible establecer otras conexiones RFCOMM (con números de canal de servidor entre 1 y 30) para transmitir datos de capas superiores.

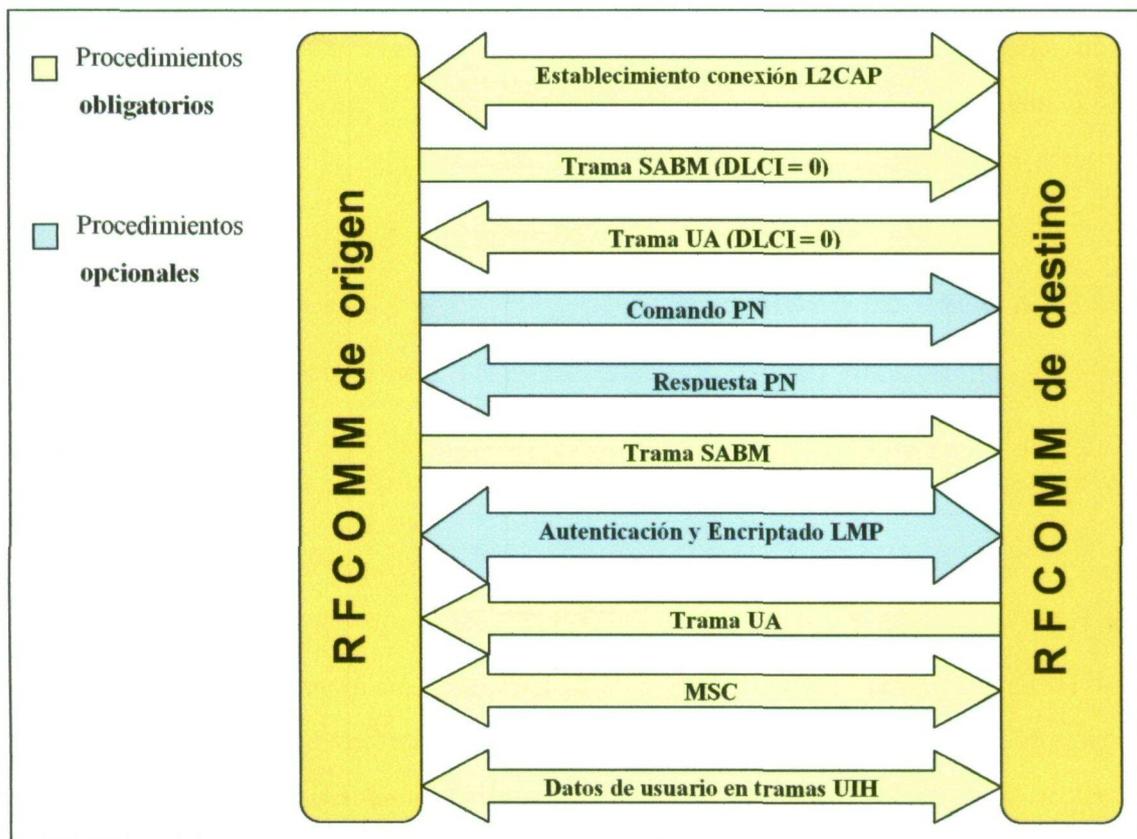


Figura 3.17 Establecimiento de una conexión RFCOMM.

3.9 Protocolo de descubrimiento de servicios

Como ya se ha mencionado, una de las características principales de las piconets Bluetooth es su dinamismo y configuración altamente cambiante. Un dispositivo puede entrar en una piconet, llevar a cabo un determinado intercambio de datos y salir en cuestión de minutos. Debido a esto se hace necesario algún tipo de método para informar de los **servicios** que proporcionan los dispositivos de una determinada piconet. El **protocolo de descubrimiento de servicios** (o *Service Discovery Protocol*, **SDP**) es el encargado de realizar esta tarea. SDP se sitúa, dentro de la pila de protocolos Bluetooth, sobre L2CAP, y al mismo nivel que RFCOMM. Por tanto es necesario establecer previamente un enlace L2CAP para poder iniciar una sesión SDP. Para ello se utiliza el **PSM** (*Protocol Service Multiplexor*) = **0x0001** [1].

El procedimiento para poder llevar a cabo el proceso de descubrimiento de servicios en Bluetooth se basa en un **modelo cliente-servidor**, de forma que un **servidor SDP** es cualquier dispositivo capaz de ofrecer una serie de servicios a otros dispositivos o **clientes SDP**. Además, cualquier dispositivo Bluetooth puede desempeñar tanto el papel de cliente como el de servidor. Por otro lado, Bluetooth no define ningún tipo de **interfaz de usuario** para llevar a cabo el descubrimiento de servicios, sino que es específico de cada implementación [1].

3.9.1 La base de datos SDP

La información de los diferentes servicios que ofrece cada dispositivo se almacena en una **base de datos**. Esta base de datos consiste básicamente en un **conjunto de registros** que almacena información referente a los servicios ofrecidos [1].

3.9.1.1 Atributos de servicio

Los **atributos de servicio** son los elementos que describen un determinado servicio SDP, y proporcionan la información que necesita un cliente para hacer uso de estos servicios. Cada atributo se compone de un **identificador** de 16 bits y un **valor** de longitud variable. Estos valores pueden ser enteros, booleanos, *strings*, etc. El registro de servicio se encarga de agrupar los valores de los distintos atributos necesarios para

describir un determinado servicio [1]. Existen 28 tipos diferentes de atributos, entre los que destacan los siguientes:

- **ServiceRecordHandle:** es un número de 32 bits que identifica a un determinado registro de servicio.
- **ProtocolDescriptorList:** lista de protocolos necesarios para usar el servicio.
- **BluetoothProfileDescriptor List:** lista de perfiles Bluetooth que pueden utilizar un determinado servicio.
- **ServiceName:** campo de caracteres que contiene el nombre del servicio, y que puede ser usado en el interfaz de usuario del cliente SDP.

3.9.1.2 Elementos de datos

Cada atributo posee un valor que puede ser de distintos tipos y tamaños. Para informar de los valores de los atributos a un cliente se utilizan los llamados **elementos de datos**, que no son más que estructuras de datos [1]. En la figura 3.18 se muestra su formato. El primer campo almacena el **descriptor del tipo** del elemento de datos, que indica el tipo de atributo contenido en el elemento de datos. El segundo campo es el **descriptor del tamaño** del elemento de datos. Si toma los valores 1, 2, 3 ó 4 este campo codifica la longitud del atributo que sigue. Si toma los valores 5, 6 ó 7, la longitud del atributo se codifica en los siguientes 1, 2 ó 4 bytes, respectivamente. A continuación viene el campo de **datos**, codificado en formato **big endian** (se transmite primero el byte más significativo), que contiene el valor del atributo.

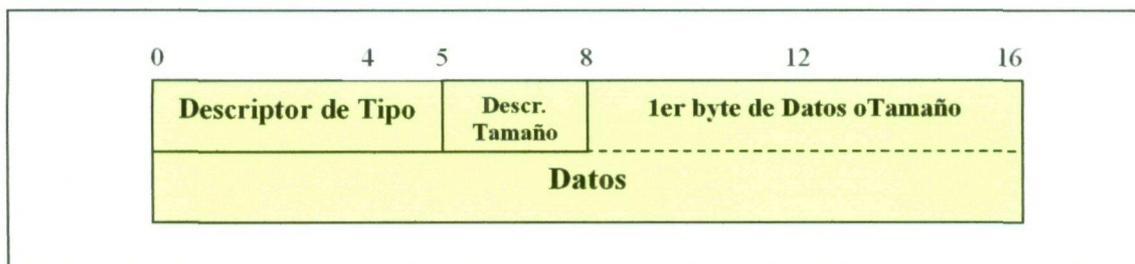


Figura 3.18 Ejemplo de un elemento de datos conteniendo un valor de atributo de 16 bits.

A veces es necesario enviar varios atributos simultáneamente. Para hacer esto se utiliza una **secuencia de elementos de datos**. En dicha estructura, el campo de datos

está compuesto no por el valor del atributo, sino por varios elementos de datos como el de la figura 3.18, conteniendo cada uno el valor de un atributo diferente.

3.9.1.3 Identificadores únicos universales (UUIDs)

Los **UUIDs** (*Universal Unique Identifiers*) son un **tipo especial** de números de 128 bits que pueden ser truncados a 32 ó 16 bits. Se utilizan para identificar a los registros de servicio, aunque también son uno de los tipos de valores a los que pueden pertenecer algunos de los atributos de servicio [1].

3.9.2 Mensajes SDP

Para que un cliente SDP pueda solicitar información sobre un determinado servicio, y que el servidor SDP se la envíe, es necesario que exista un mecanismo de intercambio de mensajes. Esto se realiza mediante **PDU**s (*Protocol Data Units*) **SDP**. En la figura 3.19 se muestra su estructura. Existen 7 tipos distintos de PDU's SDP. El primer campo consiste en un **identificador de PDU** (o **ID**), el segundo es un **identificador de transacción** que sirve para emparejar peticiones y respuestas. El siguiente es el campo de **longitud** de datos, y finalmente el campo de **datos** que contiene la información transmitida. Es posible dividir un mensaje en varias PDU's. Para ello existe un campo al final de la PDU, denominado parámetro de **estado de continuación** (*Continuation State*), cuyo valor indica si el mensaje continúa en otra PDU o termina en esa. Si el mensaje continúa, se añade un campo de información extra después del de estado de continuación, denominado campo de **información de continuación** (*Continuation Information*) [1,7].

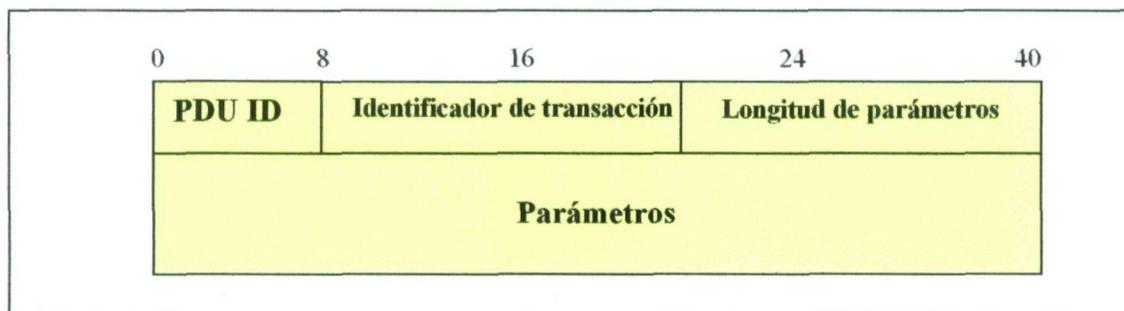


Figura 3.19 Estructura de una PDU SDP.

Existen 7 tipos de mensajes SDP diferentes:

- ***SDP_ErrorResponse* (ID = 0x01)**: cuando el servidor SDP no es capaz de contestar correctamente a un mensaje, responde con un *SDP_ErrorResponse*. Es el único mensaje que se transmite siempre utilizando exclusivamente una sola PDU.
- ***SDP_ServiceSearchRequest* (ID = 0x02)**: sirve para preguntar a un servidor SDP si dispone de un determinado servicio.
- ***SDP_ServiceSearchResponse* (ID = 0x03)**: es la respuesta habitual a un *SDP_ServiceSearchRequest*. Si la búsqueda es satisfactoria devuelve, entre otros parámetros, una lista de *ServiceRecordHandles* de los servicios disponibles que encajan con el patrón de búsqueda.
- ***SDP_ServiceAttributeRequest* (ID = 0x04)**: permite solicitar el valor de un determinado atributo perteneciente a un servicio.
- ***SDP_ServiceAttributeResponse* (ID = 0x05)**: devuelve el valor del atributo o atributos solicitados por un *SDP_ServiceAttributeRequest*.
- ***SDP_ServiceSearchAttributeRequest* (ID = 0x06)**: es una combinación de *SDP_ServiceSearchRequest* y *SDP_ServiceAttributeRequest*. Permite preguntar por la disponibilidad de un servicio, y por una serie de atributos, en caso de que el servicio esté disponible.
- ***SDP_ServiceSearchAttributeResponse* (ID = 0x07)**: es la respuesta par un *SDP_ServiceSearchAttributeRequest*.

3.9.3 Establecimiento de una sesión SDP

En la figura 3.20 se muestra un ejemplo de intercambio de mensajes durante el inicio y desarrollo de una sesión SDP. Como se observa es necesario establecer previamente un canal lógico L2CAP con PSM = 0x01.

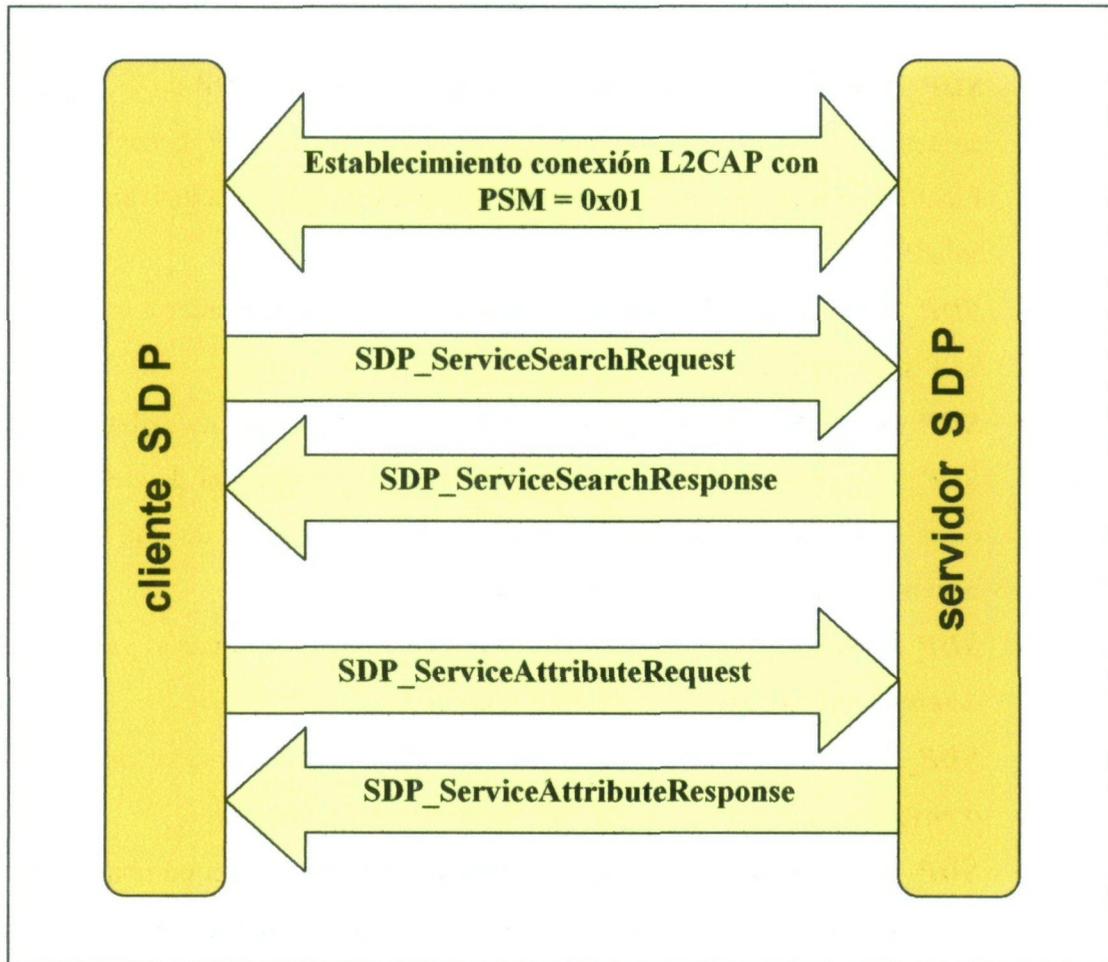


Figura 3.20 Intercambio de mensajes durante una sesión SDP

3.10 Perfil Bluetooth puerto serie

Como ya se ha comentado anteriormente, un perfil Bluetooth consiste en una descripción de cómo usar la especificación del estándar Bluetooth en la implementación de distintas aplicaciones finales. De esta forma se asegura la compatibilidad entre productos de distintos fabricantes. Para ello se definen que protocolos se implementan en cada perfil y la forma en que han de relacionarse entre sí [1].

El perfil **puerto serie** (o *serial port*) se basa en el uso de RFCOMM. También necesita del protocolo SDP para intercambiar información acerca de la disponibilidad del servicio que presta el perfil puerto serie. Por debajo se encontrarían L2CAP, HCI y las capas más bajas implementadas en el módulo Bluetooth [1]. Este perfil se aplica en

los dispositivos de tipo 1 y tipo 2 presentados en la figura 3.15. En concreto, en los dispositivos de **tipo 1**, el perfil puerto serie proporciona una forma de emular un puerto serie RS-232 en software mediante un **puerto serie virtual** [1]. De esta forma, aplicaciones destinadas a interactuar con puertos serie reales no tienen que ser modificadas. Es más, si lo que se desea es desarrollar una nueva aplicación, ésta simplemente debe utilizar las APIs ya existentes definidas para trabajar con puertos series reales.

Para establecer el puerto serie virtual es necesario llevar cabo una serie de acciones que deben ser gestionadas por una **entidad de control** (o *Device Manager*) [1]. En la figura 3.1 este elemento se presenta como el bloque denominado **CONTROL**. Los pasos a seguir se resumen brevemente en:

- Búsqueda del dispositivo con el que se establecerá la comunicación (Descubrimiento o *Inquiring*).
- Establecimiento de una conexión ACL, creando así un enlace físico entre los dispositivos (Conexión o *Paging*).
- Creación de un canal lógico L2CAP para que pueda ser usado por el protocolo SDP. Durante la sesión SDP el servidor informa al cliente del **número del canal de servidor** (o *Server Channel Number*) del servicio solicitado.
- Creación de un canal lógico L2CAP para que pueda ser usado por RFCOMM.
- Establecimiento de un canal de señalización a nivel RFCOMM, necesario para configurar y controlar la conexión RFCOMM (con DLCI = 0).
- Establecimiento de otro canal RFCOMM (con DLCI obtenido a partir número del canal de servidor) destinado al intercambio de datos de usuario.

Una vez hecho esto, las aplicaciones software que se encuentran en ambos extremos del enlace se conectan a los puertos serie virtuales creados e intercambian datos utilizando tramas RFCOMM.

CAPÍTULO 4

DESCRIPCIÓN DEL SISTEMA

4.1 Introducción

En este capítulo se describen los elementos que componen el sistema de monitorización ECG diseñado e implementado durante la realización de este Proyecto Fin de Carrera. Además se explicará brevemente la operación y funcionamiento de estos elementos y del sistema completo.

4.2 Componentes del sistema

En la figura 4.1 se muestra un dibujo esquemático del sistema desarrollado en el presente proyecto. Se compone de los siguientes elementos:

- **Unidad portátil**
- **Unidad de visualización ECG**

A continuación se pasará a explicar con más detalle cada uno de ellos.

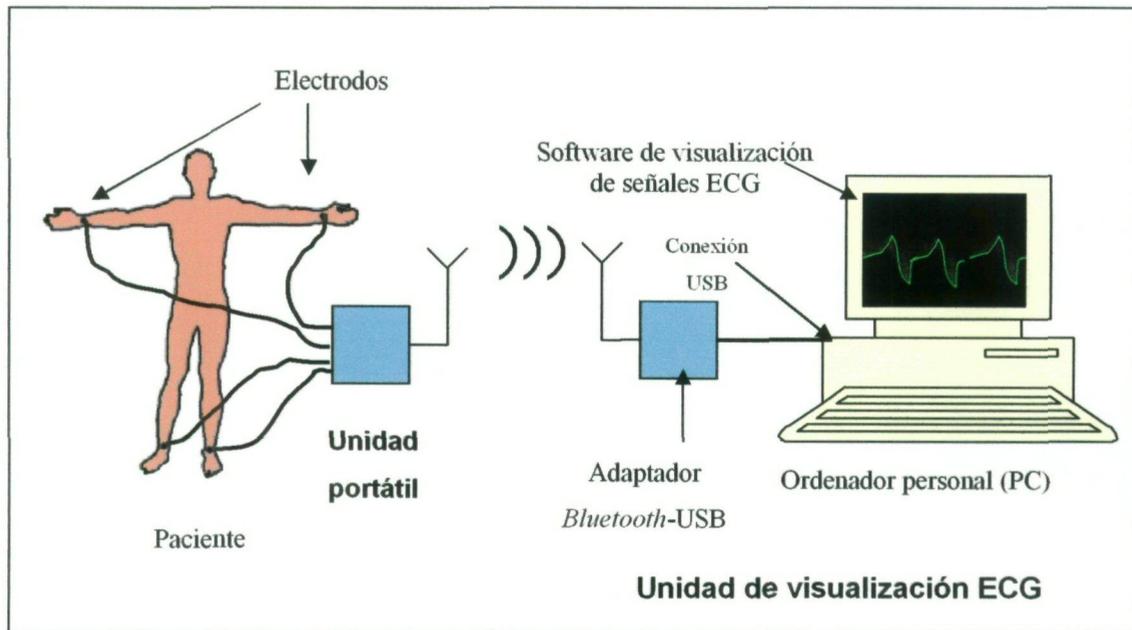


Figura 4.1 Esquema completo del sistema diseñado.

4.2.1 Unidad portátil

La **unidad portátil**, o módulo emisor como también es llamado en algún apartado de esta memoria, tiene como función realizar la **adquisición, acondicionamiento, digitalización y transmisión** de la señal ECG. En la figura 4.2 se muestra un diagrama de bloques donde se destacan las principales funciones que lleva a cabo cada una de las partes que componen la unidad portátil.

En primer lugar se encuentra el bloque de **adquisición de la señal ECG**. Éste se encarga de adquirir las **señales II y III** a partir de **4 electrodos** situados en las extremidades del paciente que constituyen las **derivaciones de Einthoven** (ver capítulo 2). La **señal I** se obtendrá posteriormente en la unidad de visualización ECG a partir de las señales II y III. En el desarrollo del prototipo del sistema se ha trabajado con un **paciente electrónico**. Este dispositivo no es más que un simulador de señales biológicas, capaz de generar, entre otras cosas, las señales características de un ECG clínico. Dicho dispositivo posee 4 terminales correspondientes a las derivaciones **RA, LA, RL y LL** (ver capítulo 2). Estos terminales se conectan al puerto o **conector de adquisición de señales** de la unidad portátil. A continuación se procede a **acondicionar**

la señal mediante una serie de etapas de **filtrado y amplificación**. Con ello se consigue obtener una señal apta para ser **digitalizada**.

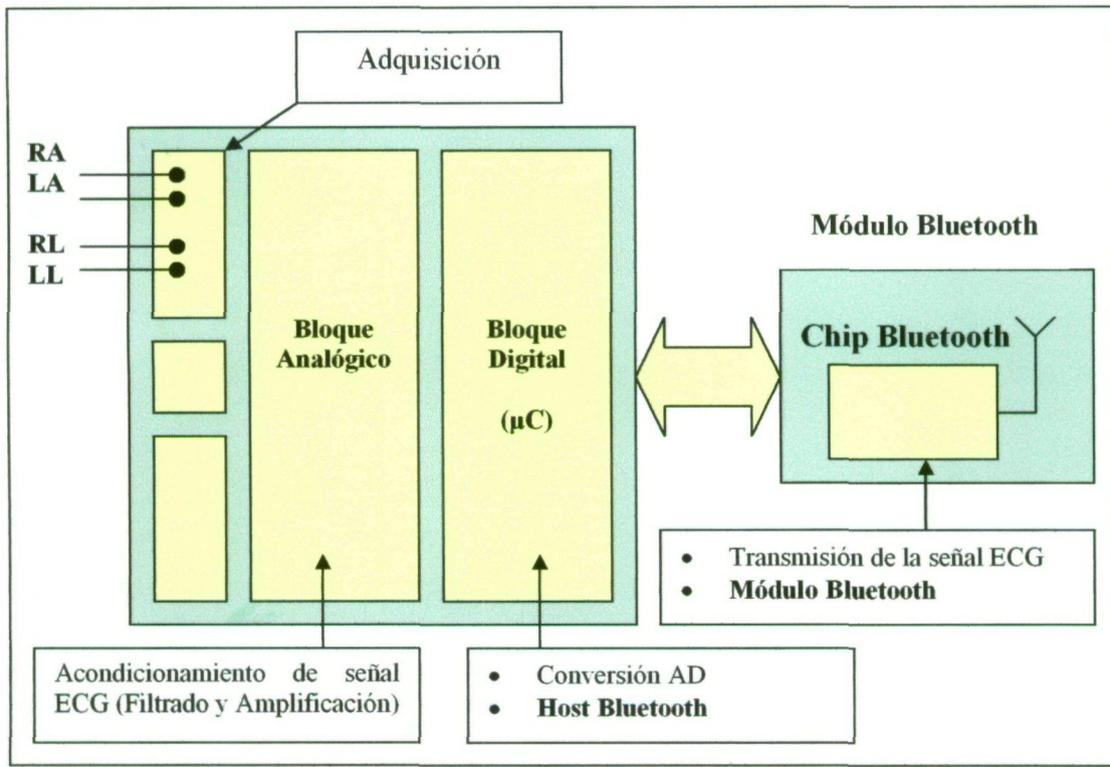


Figura 4.2 Diagrama de bloques simplificado de la unidad portátil

El corazón del bloque digital es un **microcontrolador PIC16F876**. Entre los recursos de los que dispone se encuentra un **convertor analógico-digital** de 5 canales y hasta 10 bits de resolución [4]. Este convertor es el encargado de realizar la digitalización de la señal ECG. En el microcontrolador se integra, además, el programa que implementa el **host Bluetooth**. Este host, junto con el **módulo Bluetooth**, permite transmitir los datos relativos a la señal ECG a otro dispositivo Bluetooth, en este caso la **unidad de visualización ECG**. Para ello se hace uso del perfil Bluetooth **puerto serie**, el cual requiere de la implementación de los protocolos **HCI, L2CAP, SDP y RFCOMM** [1]. Hay que decir que estos protocolos no se implementan en su totalidad, ya que el tiempo y los recursos necesarios para llevarlo a cabo serían excesivos, y por tanto estarían fuera del ámbito de desarrollo de un Proyecto Fin de Carrera. Además, las partes no implementadas supondrían un aumento injustificado en el hardware necesario y en el consumo de la unidad portátil, por lo que solamente se han desarrollado aquellas funciones de los protocolos imprescindibles para el correcto funcionamiento del enlace

Bluetooth entre la unidad portátil y el PC. En el capítulo 5 se especifican que elementos se implementan y cuales se dejan para futuras líneas de desarrollo. En cualquier caso la implementación en lenguaje ensamblador de este host ha representado sin duda el mayor reto que se ha surgido durante la realización del proyecto.

En cuanto al **módulo Bluetooth** que se emplea en la unidad portátil, se trata de un **chip Bluetooth comercial** de la compañía Ericsson modelo **ROK 101 008**. Este chip viene montado sobre una PCB que es independiente de la placa donde se sitúan el resto de elementos que conforman la unidad portátil. Este conjunto de PCB más chip Bluetooth se incluye dentro de los kits de desarrollo Bluetooth de Ericsson.

Esta configuración, en la que se tiene por un lado el módulo Bluetooth y por otro el resto de bloques de la unidad portátil, puede ser apropiado para un prototipo pero en un dispositivo final se debería integrar el chip Bluetooth en la misma placa que el resto de componentes. De hecho en el Laboratorio de Tecnología Fotónica se dispone de chips Bluetooth destinados a tal fin, aunque ante la complejidad del tipo de soldadura empleado (tipo BGA) se optó por desechar esta posibilidad por el momento.

4.2.2 Unidad de visualización ECG

En la figura 4.3 se muestra en esquema de la unidad de visualización ECG que consiste, básicamente, en una **aplicación software de visualización** que se ejecuta en un **ordenador personal (PC)**: Además, este PC dispone de un **adaptador Bluetooth-USB** (en adelante **adaptador BT-USB**) que le permite conectarse a otros dispositivos Bluetooth.

El adaptador BT-USB utilizado es un dispositivo comercial de la marca BrainBoxes modelo **BL-554**. Este dispositivo requiere la instalación previa de unos **drivers** en donde se implementa el host Bluetooth del dispositivo, y de una **aplicación** que permite el control y configuración de la conexión Bluetooth por parte del usuario. El adaptador BT-USB junto con el software asociado dispone del servicio prestado por

el perfil puerto serie, y se encarga de crear el **puerto serie virtual** una vez establecida la conexión con la unidad portátil.

La **aplicación de visualización** es otro de los elementos desarrollados específicamente para este proyecto. Esta aplicación se encarga de abrir el puerto serie virtual y leer los datos que recibe a través de él. Para ello hace uso del conjunto de rutinas de la **API (Application Programming Interface) Microsoft Win32** que se utilizan cuando se trabaja con los puertos serie reales así como los puertos paralelos del ordenador (COM1, COM2, LPT1, etc.) en un **sistema operativo Windows** [9]. De esta manera y gracias a la utilización de esta API, la aplicación desarrollada puede ser instalada en cualquier sistema informático con sistema operativo Windows que cuente con un terminal Bluetooth que implemente el perfil puerto serie (PDA, portátiles, Tablet-PC,...).

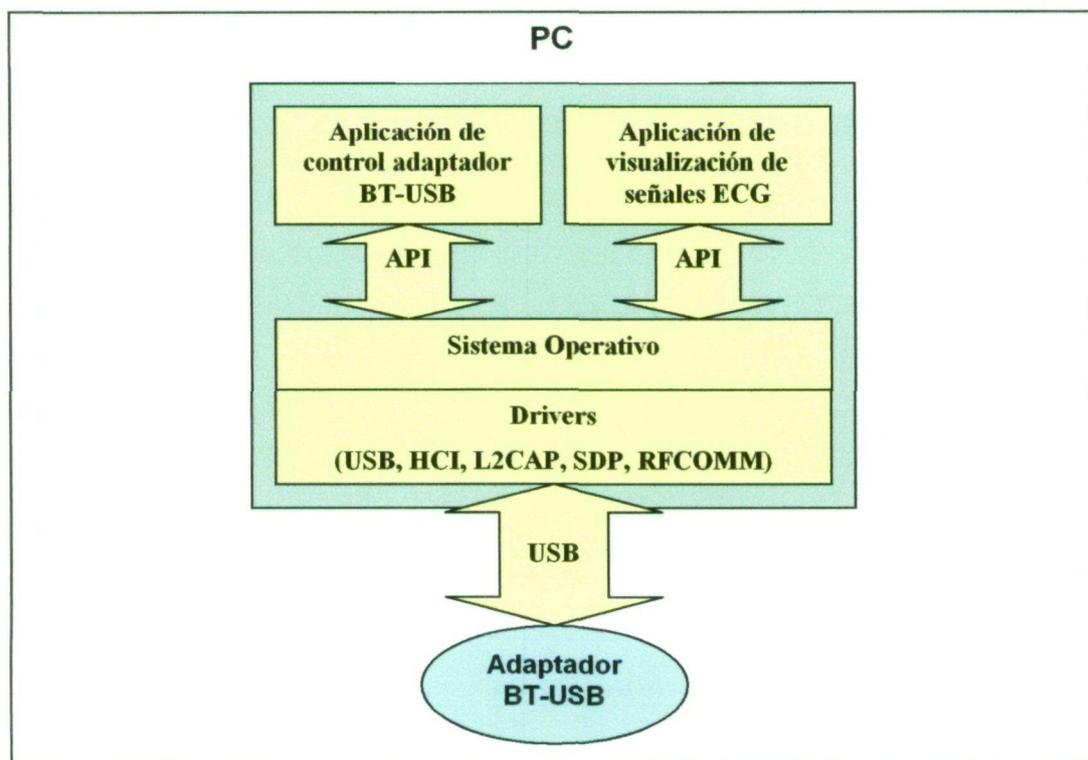


Figura 4.3 Diagrama de bloques de la unidad de visualización ECG.

La aplicación de visualización se encarga no solamente de representar gráficamente las señales ECG a partir de los datos recibidos, sino que lleva a cabo

diversas tareas de procesamiento digital para mejorar la calidad de la visualización y obtener información de adicional a partir de los datos recibidos.

En primer lugar se encarga de eliminar la interferencia de red de 50 Hz aplicando la técnica de **filtrado adaptativo** [10]. También realiza, a partir de la señal ECG, una estimación de la **frecuencia cardiaca** del paciente [5]. Este dato, importantísimo en las aplicaciones de monitorización cardiaca, es representado también gráficamente en la interfaz de usuario de la aplicación.

Otra de las características de la aplicación de visualización es que permite capturar en un archivo de mapa de bits (extensión .bmp) las representaciones de las señales I, II y III, así como almacenar en un fichero ASCII (extensión .txt) los valores de las muestras recibidas para su posterior estudio con otro tipo de herramientas informáticas, como por ejemplo MATLAB. Esta característica puede ser interesante para su aplicación en técnicas ECG de tipo ambulatorio o Holter (ver capítulo 2).

4.3 Funcionamiento del sistema

Para que el sistema funcione correctamente es necesario seguir una serie de pasos de forma que pueda establecerse una conexión entre la unidad portátil y la unidad de visualización ECG. Debido a que la unidad portátil carece de un interfaz de usuario algunos parámetros de la conexión deben ser configurados antes de activarla. Este dispositivo va a desempeñar el papel de **maestro** y la unidad de visualización ECG será el **esclavo**. Por tanto en primer lugar hay que ejecutar la aplicación de control del adaptador BT-USB para, a continuación, configurarlo en modo *Inquiry Scan* (ver capítulo 3). En este punto también hay que establecer el nivel de seguridad requerido. En el caso que nos ocupa se utilizará el nivel más bajo, de forma que no se llevará a cabo ningún procedimiento de **autenticación** entre ambos dispositivo mediante un PIN de acceso.

El siguiente paso consiste en activar la unidad portátil, ya sea conectando la alimentación o reiniciando el dispositivo. Entonces la unidad portátil iniciará automáticamente el procedimiento de *Inquiry*. Una vez descubierto el adaptador BT-USB se procede a establecer una **conexión ACL** con el mismo. Los pasos anteriores son

notificados por la aplicación de control del adaptador BT-USB. Seguidamente se produce la creación del **puerto serie virtual** en el PC. Tan pronto esto ocurre, la unidad portátil empieza a transmitir datos. Por tanto el siguiente paso es ejecutar la aplicación de visualización en el PC y configurarla con el nombre del puerto serie virtual creado para que ésta pueda empezar a leer los datos recibidos por dicho puerto serie, de forma que puedan ser representados gráficamente en la pantalla del ordenador.

4.4 Conclusiones

En este capítulo se ha presentado de forma general la estructura y funcionamiento del sistema realizado en el presente Proyecto Fin de Carrera. En los siguientes capítulos se desarrollarán con más detalle cada una de las partes descritas aquí.

CAPÍTULO 5

UNIDAD PORTÁTIL

5.1 Introducción

En este capítulo se aborda con mayor profundidad la descripción de cada una de las partes fundamentales que componen la unidad portátil. Se explicarán, además, las decisiones de diseño adoptadas, así como las técnicas o procedimientos empleados durante su implementación.

Como ya se ha mencionado, la unidad portátil se compone de un bloque analógico encargado de acondicionar las señales II y III obtenidas con la ayuda de los electrodos. También existe un bloque digital, formado principalmente por un **microcontrolador PIC16F876** y un **módulo Bluetooth comercial**. Estos elementos poseen los recursos suficientes para poder implementar un dispositivo Bluetooth. El microcontrolador además tiene la misión de llevar a cabo la conversión analógica-digital de las señales previamente acondicionadas en la parte analógica de la unidad portátil.

5.2 Diseño e implementación del hardware

El **hardware** de la unidad portátil se compone de seis partes fundamentales: **etapa de acondicionamiento de señal ECG, microcontrolador, módulo Bluetooth, etapa de alimentación, conectores y circuito de reset** del sistema. Todos estos elementos, excepto el módulo Bluetooth, se encuentran agrupados en una única placa de circuito impreso (PCB). Esta PCB ha sido **diseñada e implementada** completamente durante la realización de este proyecto. El *layout* de la PCB se ha generado con la ayuda de la herramienta informática PROTEL. Para el diseño de la etapa de acondicionamiento de señal ECG se ha recurrido al uso de PSPICE. Aunque la complejidad de la circuitería analógica no entrañaba grandes dificultades, el uso de PSPICE ha permitido garantizar el correcto funcionamiento del diseño antes de ser implementado.

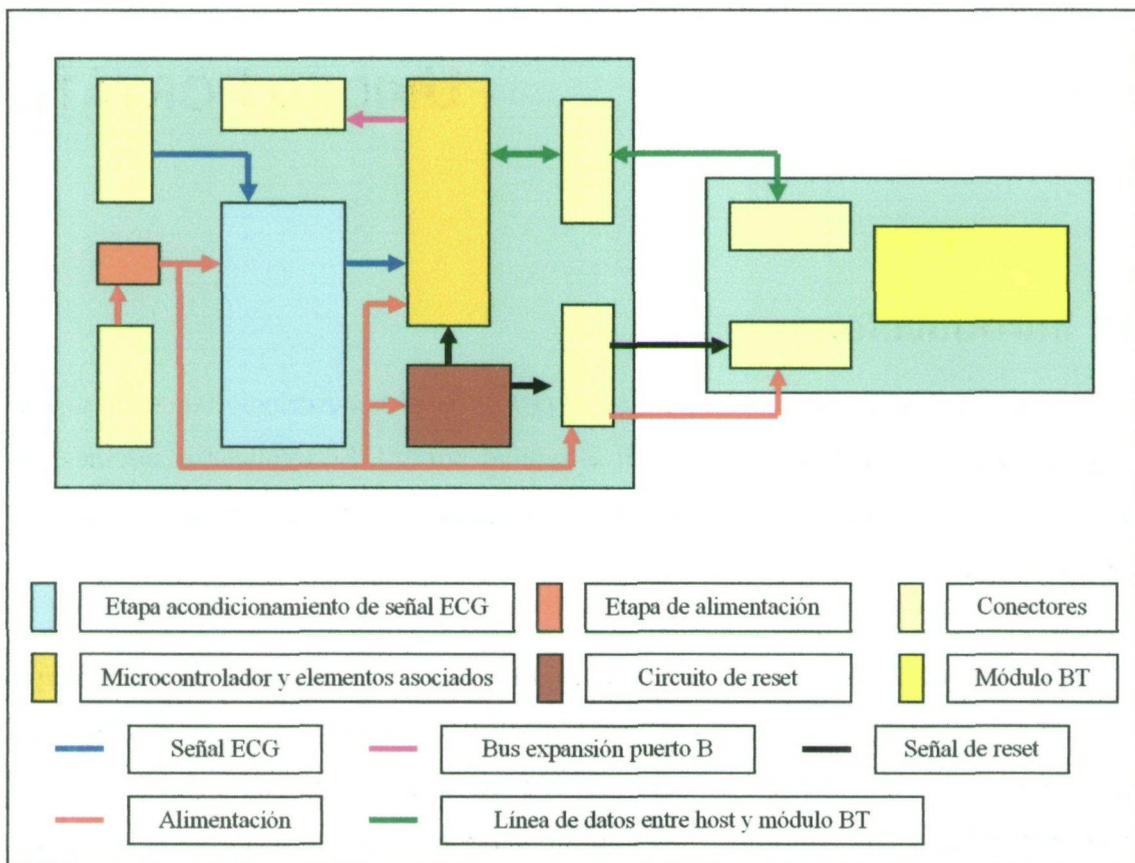


Figura 5.1 Diagrama de bloques de la unidad portátil.

Los **componentes pasivos** utilizados en la fase de implementación de la unidad portátil, es decir resistencias y condensadores, son todos de tipo SMD (*Surface*

Mounting Device) con el objetivo de simplificar la tarea de soldadura y obtener un tamaño del diseño más reducido. Los **circuítos integrados** (IC) son también de tipo SMD, excepto los amplificadores de instrumentación utilizados en la primera etapa de amplificación, ya que no estaban disponibles en su versión SMD. El microcontrolador PIC utilizado también se emplea en su versión de encapsulado *Dual in Line* (DIP), ya que es la única que permite la extracción del mismo de la placa para ser reprogramado durante el proceso de desarrollo del software de la unidad portátil.

5.2.1 Etapa de acondicionamiento de la señal ECG

Constituye la parte analógica de la unidad portátil. Se compone de **dos canales idénticos** cuya misión es **amplificar** y **filtrar** las señales II y III. En la figura 5.2 se muestra un diagrama de bloques simplificado de uno de estos canales. Cada canal se compone de dos etapas de amplificación y dos etapas de filtrado. Las dos primeras tienen la misión de, a partir de una señal que originalmente posee una amplitud de como máximo un par de milivoltios, obtener otra que abarque los niveles de tensión de entrada con los que trabaja el conversor AD, que en este caso se encuentra entre 0 y 5 voltios.

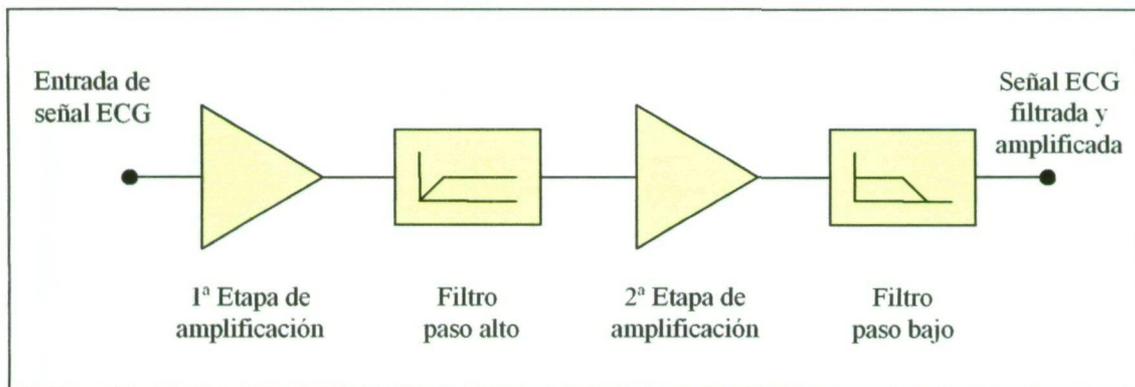


Figura 5.2 Diagrama de bloques de uno de los canales de acondicionamiento de señal ECG.

Las etapas de filtrado se encargan de limitar el ancho de banda de la señal antes de ser muestreada. Así se evita el *aliasing* y se eliminan ruidos de alta frecuencia como por ejemplo el que provoca la contracción de los músculos (ruido EMG) [3]. También se elimina el nivel de continua que incorpora la señal original. En este proyecto el **ancho de banda de trabajo** se encuentra entre **0,5 y 100 Hz**.

5.2.1.1 Primera etapa de amplificación

Esta etapa está formada por un **amplificador de instrumentación AD623AN** de Analog Devices en su versión de **encapsulado DIP**. Se trata de un dispositivo que se caracteriza por un **bajo consumo** (575 μA de corriente máxima de alimentación) y un ancho de banda de 800 KHz para una ganancia $G = 1$. Con estos valores cumple sobradamente los requerimientos de este proyecto. Además mantiene una **CMRR** (*Common Mode Rejection Ratio*) constante de 90 dB casi hasta los 200 Hz. Otro de los motivos por los que se escogió este amplificador es que puede ser alimentado con tensiones de entre 3 y 12 voltios para el terminal positivo, con el terminal negativo conectado a masa (en inglés esta configuración es denominada *Single Supply*). En el caso de este proyecto, y como se verá más adelante, la tensión de alimentación de los elementos activos tanto analógicos como digitales es de 5 voltios.

En cuanto a la **ganancia**, ésta puede ser fijada mediante una resistencia externa (R_g). En este caso se escoge una resistencia de **330 Ω** , con lo que se obtiene la ganancia dada por la siguiente expresión:

$$G = \left(1 + \frac{100\text{K}\Omega}{R_g} \right) = \left(1 + \frac{100\text{K}\Omega}{330} \right) = 303,03 \quad (5.1)$$

En teoría este amplificador es capaz de proporcionar ganancias de hasta 1000, aunque incluso en este caso se haría necesaria una segunda etapa de amplificación debido a la pequeña amplitud de la señal original. En la figura 5.3 se muestra una representación esquemática de la 1ª etapa de amplificación de la señal II, donde se muestra la **configuración diferencial** de las señales procedentes de los electrodos. La señal II se define como la diferencia entre las tensiones RA y LL tomando como masa del sistema la tensión RL. Para la señal III sería igual, pero sustituyendo LA por RA.

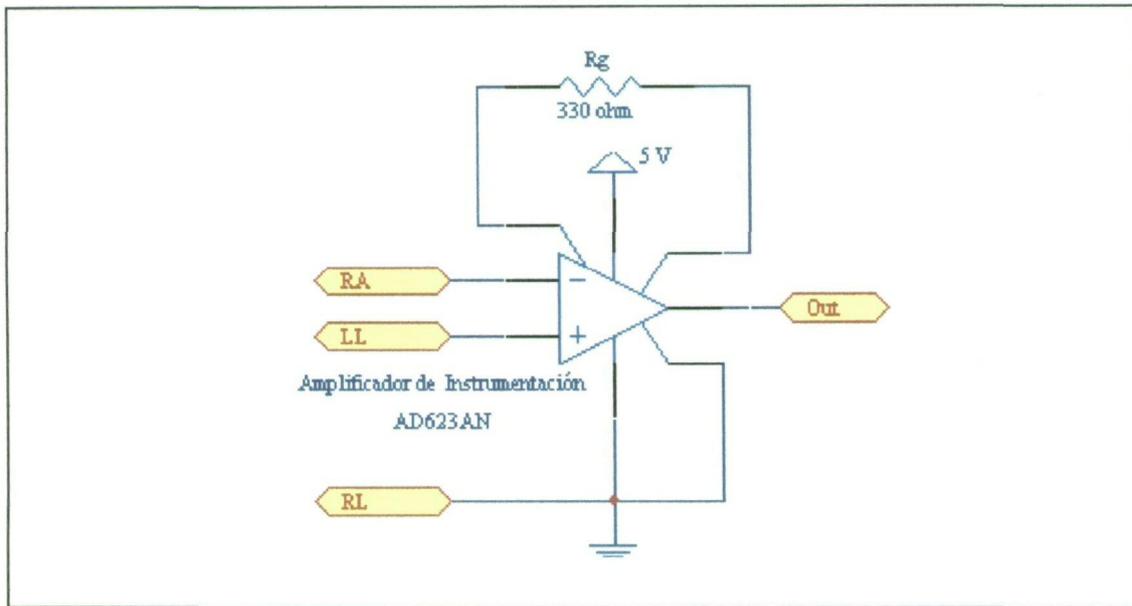


Figura 5.3 Estructura de la primera etapa de amplificación.

5.2.1.2 Filtro paso alto

En la figura 5.4 se muestra la estructura de este filtro RC, cuya misión es principalmente la de eliminar el nivel de continua y el ruido provocado por la respiración del paciente existente a la salida del amplificador de instrumentación.

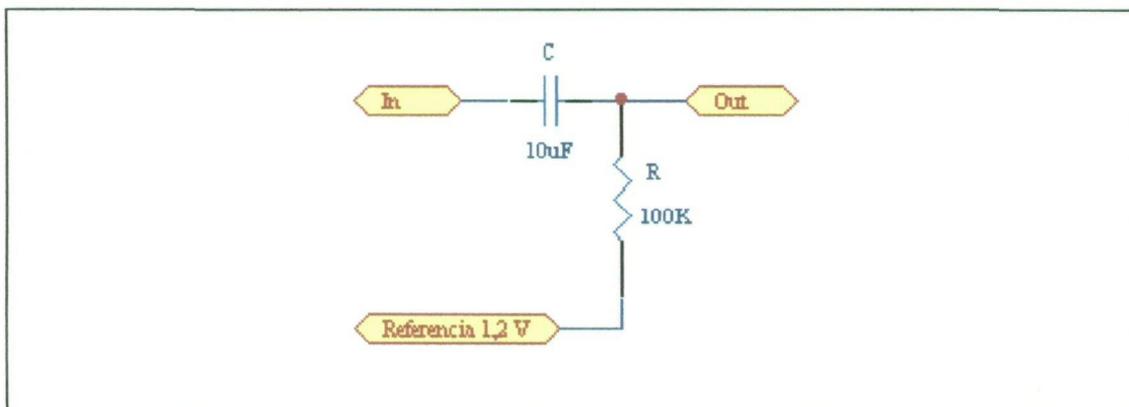


Figura 5.4 Estructura del filtro paso alto.

Otra de las misiones de este filtro paso alto es añadir un **nivel de continua de 1,2 voltios** a la señal que se introduce en la siguiente etapa de amplificación. Este nivel de continua se mantiene hasta la salida del filtro paso bajo que constituye la última

etapa del canal de acondicionamiento, y es necesario para centrar la señal ECG amplificada dentro del margen de tensiones de trabajo del conversor AD del microcontrolador. Para ello se conecta la resistencia del filtro paso alto a la referencia de tensión de 1,2 voltios.

Los valores de R y C elegidos son $30 \text{ K}\Omega$ y $10 \text{ }\mu\text{F}$ respectivamente, con lo que se establece una **frecuencia de corte**, según la expresión 5.2, de **0,53 Hz**.

$$f_c = \frac{1}{2\pi RC} \quad (5.2)$$

5.2.1.3 Segunda etapa de amplificación

La segunda etapa de amplificación está formada por un **amplificador operacional** en configuración **no inversora**. Para implementar esta etapa se ha elegido el IC **AD8542AR** de Analog Devices. Este circuito integrado se presenta en un encapsulado SMD e integra dos amplificadores operacionales, de forma que se emplea uno para cada canal de acondicionamiento. El montaje utilizado con cada uno de estos amplificadores operacionales es el que se muestra en la figura 5.5. Los valores escogidos para las resistencias R_1 y R_2 son $1 \text{ K}\Omega$ y $4,72 \text{ K}\Omega$ respectivamente. Estos valores determinan una **ganancia** de **5,72**, según la expresión 5.3.

$$G = 1 + \frac{R_2}{R_1} \quad (5.3)$$

Otra particularidad de esta etapa es que la rama realimentada también está conectada a la referencia de 1,2 voltios. Debido a esto, y a que la señal ECG de entrada ya tiene un nivel de continua idéntico, la señal amplificada por esta etapa posee también a la salida un nivel de continua de 1,2 voltios. De esta forma, y como ya se ha mencionado anteriormente, se consigue centrar la señal dentro del margen de tensiones de trabajo del conversor analógico-digital.

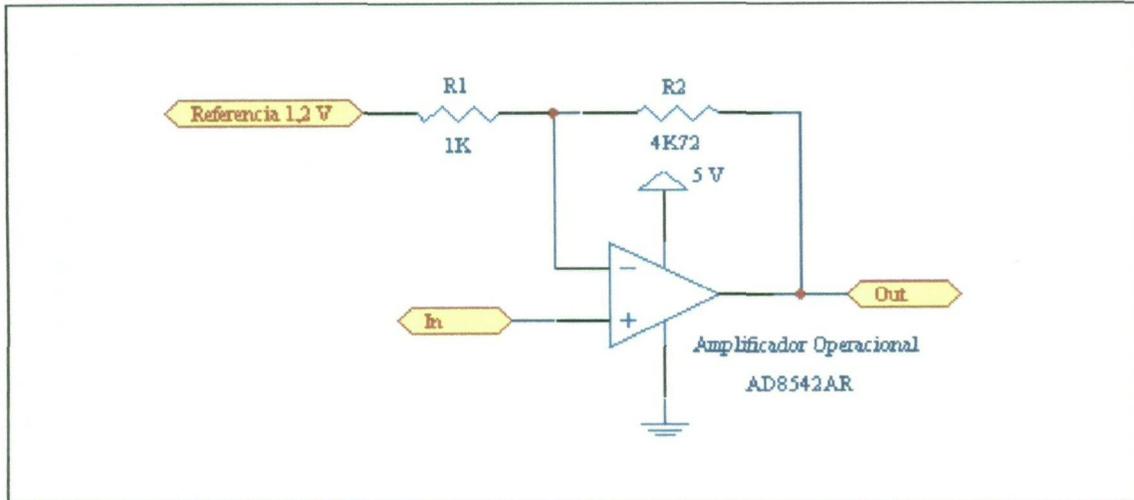


Figura 5.5 Estructura de la segunda etapa de amplificación.

5.2.1.4 Filtro paso bajo

Esta etapa filtra la señal ECG antes de que sea muestreada por el conversor analógico-digital. Así se elimina ruido alta frecuencia y se evita el *aliasing*. Se trata de un **filtro RC** cuya **frecuencia de corte** se sitúa en **88,4 Hz**. Teóricamente la frecuencia de corte debería ser de 100 Hz, pero debido a la baja selectividad del filtro RC empleado se ha decidido reducir el valor de este parámetro. Con la ayuda de la expresión 5.2, que también es válida para filtros paso bajo, se calculan los valores del condensador y la resistencia, que en este caso son de **1 μF** y **1,8 $\text{K}\Omega$** respectivamente, como se muestra en la figura 5.6.

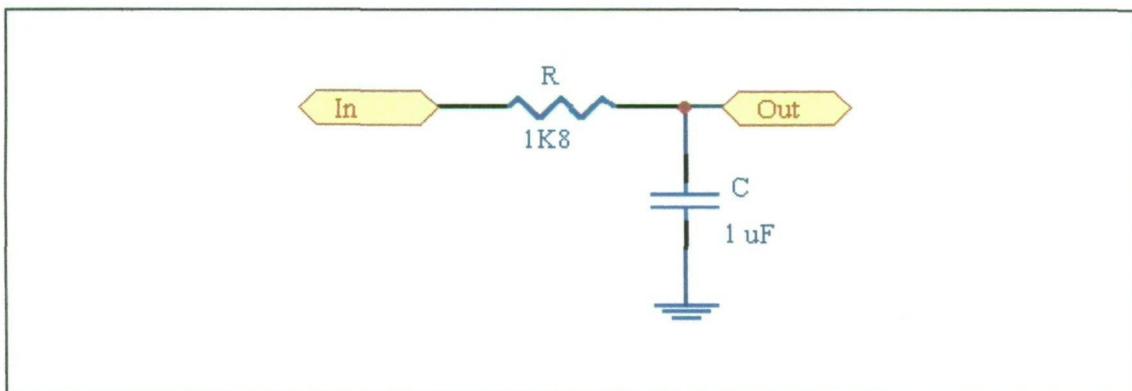


Figura 5.6 Estructura del filtro paso bajo.

En el diseño de este filtro también hay que tener en cuenta a la hora de seleccionar los valores de R y C que la impedancia vista por el conversor, es decir la **impedancia de salida** del canal de acondicionamiento, debe ser inferior según el fabricante del microcontrolador a **10 K Ω** [4]. En este diseño se cumple esta condición, ya que la resistencia del filtro limita el valor máximo de la impedancia equivalente de salida. Esta resistencia no se conecta a tierra, sino a la salida del amplificador operacional de la segunda etapa de amplificación, pero como la impedancia de salida de un amplificador operacional suele ser pequeña (**50 Ω** en el caso del amplificador AD8542AR), se puede hacer la suposición de que la resistencia del filtro paso bajo está conectada a tierra.

5.2.1.5 Referencia de tensión

La referencia de tensión de 1,2 voltios utilizada por el filtro paso alto y la segunda etapa de amplificación está formada por un circuito integrado **LM4140ACM** de National Semiconductor y uno de los amplificadores operacionales de un **AD8542AR** configurado como **buffer de tensión**, tal como se muestra en la figura 5.7. Aunque el LM4140ACM puede suministrar hasta 8 mA de corriente máxima, se decidió añadir el buffer de tensión por precaución, ya que solamente existe un único bloque de este tipo para ambos canales. De esta forma es posible ofrecer hasta un **máximo de 30 mA** al conjunto de los elementos conectados a este circuito.

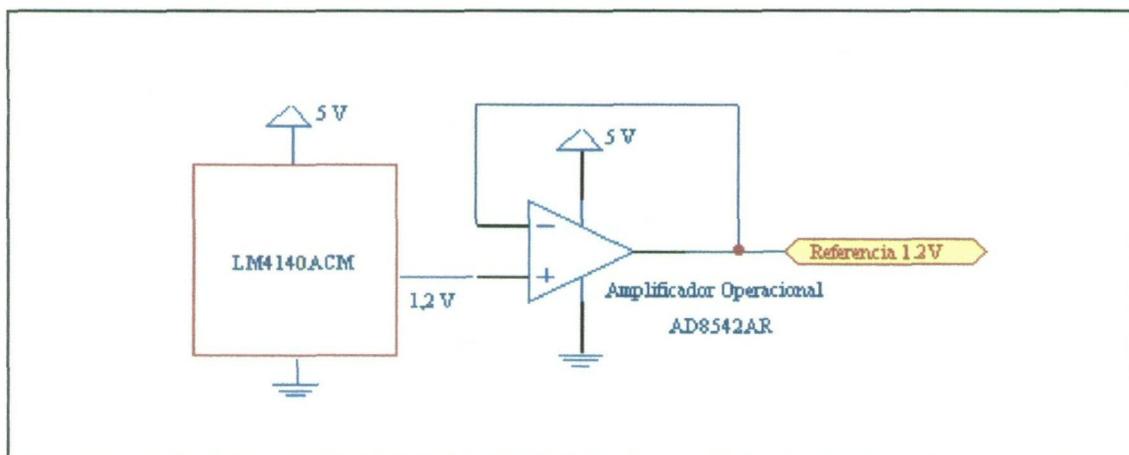


Figura 5.7 Referencia de tensión de 1,2 voltios.

5.2.2 Microcontrolador

El corazón de la unidad portátil lo constituye un microcontrolador **PIC16F876** de la compañía Microchip. Como ya se ha dicho anteriormente, se encarga de realizar el proceso de **conversión analógica-digital** de la señal ECG y en él se implementa el **host Bluetooth** encargado de controlar y gestionar el enlace inalámbrico con la unidad de visualización ECG.

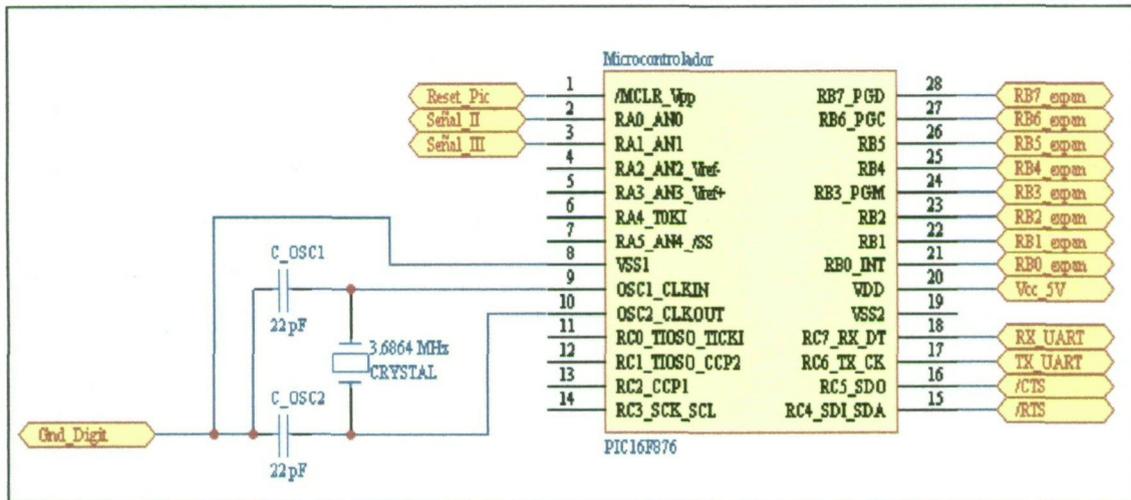


Figura 5.8 Esquema de conexionado del PIC16F876

Este microcontrolador dispone entre otros recursos de un **convertor analógico-digital** de 5 canales y hasta 10 bits de resolución, una **UART** que se utiliza en la comunicación con el módulo Bluetooth, y una **memoria de programa (FLASH)** de **8 Kpalabras**. Otras características de este dispositivo que lo hacen muy interesante son un juego reducido de 35 instrucciones, 3 **puertos de entrada/salida**, 3 **timers** y una **velocidad de CPU hasta 20 MHz**, además de un precio muy competitivo. En cuanto a **memoria de datos (RAM)** dispone solamente de **368 bytes**, lo que puede ser un factor limitante a la hora de implementar los buffers de transmisión y recepción del host Bluetooth. En la figura 5.8 se muestra como se interconecta el microcontrolador a otros elementos de la PCB. Además se observa el circuito de reloj compuesto por dos condensadores y un cristal de cuarzo de **3,6864 MHz**. La elección de este valor para la frecuencia de operación se encuentra estrechamente ligada a la velocidad de transmisión con que se configura la UART del microcontrolador, que es de **57600 bps**. La frecuencia de reloj seleccionada es la única que permite establecer esta velocidad de transmisión sin error [4]. Además el **ciclo de instrucción** se obtiene como la inversa del

valor de la frecuencia de oscilación del circuito de reloj dividido por cuatro. En este caso es de **1 μ s**. Salvo algunas instrucciones de salto que requieren de dos ciclos, la mayoría de las instrucciones se ejecutan en un solo ciclo de instrucción.

La placa del kit de desarrollo Bluetooth dispone de un transceptor RS-232 **ADM3202** conectado a la UART del chip Bluetooth. Por tanto el microcontrolador debe hacer uso de un dispositivo similar para poder conectarse al módulo Bluetooth. En este caso se trata de un IC **MAX-232ECSE** (encapsulado SMD). Este integrado necesita de la utilización de 4 condensadores externos de 100 nF para su correcto funcionamiento, y su principal función es convertir niveles de tensión CMOS/TTL a RS-232 y viceversa. Esto permite conectar las UARTs del microcontrolador y módulo Bluetooth en configuración de **módem nulo** haciendo uso de las señales **TXD**, **RXD**, **RTS** y **CTS**. El microcontrolador dispone solamente de las líneas TXD y RXD para recepción y transmisión de datos. Las líneas CTS y RTS se implementan mediante las líneas 4 y 5 del puerto C.

5.2.3 Módulo Bluetooth

Como ya se comentado anteriormente, el módulo Bluetooth utilizado consiste en un **chip Bluetooth** de Ericsson modelo **ROK101008** que se monta sobre una PCB de doble capa. Esta placa forma parte del **kit de desarrollo Bluetooth** comercializado por Ericsson. En ella se incluye, además, una antena, un regulador de tensión, un transceptor RS-232, un pulsador de reset, un conector RS-232 y otro USB, tal como se muestra en la figura 5.9. El chip Bluetooth cumple con la especificación 1.1 del estándar Bluetooth y en cuanto a potencia pertenece a los dispositivos de **clase 2** (0 dBm). La única diferencia de este chip con otros destinados específicamente a la fabricación de dispositivos Bluetooth es que está habilitado exclusivamente para conexiones punto a punto. Por tanto con este módulo no es posible establecer piconets con más de un esclavo.

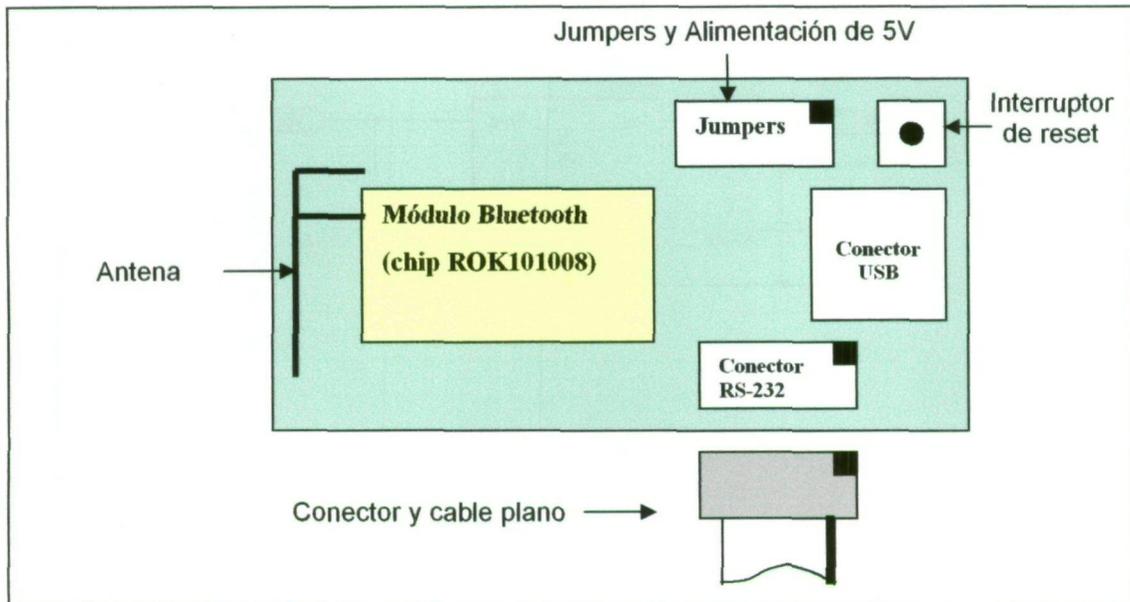


Figura 5.9 Placa del kit de desarrollo Bluetooth de Ericsson empleada en la unidad portátil.

5.2.4 Etapa de alimentación

La unidad portátil funciona con una tensión continua de **5 voltios**. Ésta se obtiene a partir de un **regulador de tensión L-78M05CDT** (encapsulado SMD) de National Semiconductors capaz de suministrar a la salida hasta **500 mA** de corriente. Este dispositivo requiere del uso de dos condensadores externos de **100** y **330 nF**, y puede ser alimentado con tensiones de entre **6** y **35 voltios**. Esto permite alimentar el sistema con cualquier tipo de pilas o baterías recargables que trabajen dentro de este rango de tensiones. Un valor muy estandarizado es el de 9 voltios. Se escoge la tensión de funcionamiento de 5 voltios porque es la tensión de alimentación que requiere la placa del kit de desarrollo Bluetooth, aunque en realidad el chip Bluetooth propiamente dicho funciona con 3,3 voltios. El resto de circuitos integrados utilizados en el diseño permiten también trabajar con tensiones menores de 5 voltios. Por tanto en futuros desarrollos en los que se integre el chip Bluetooth en la misma placa que el resto de elementos, se puede considerar la posibilidad de bajar la tensión de alimentación a 3,3 voltios, con la consiguiente reducción en el consumo de potencia. En la figura 5.10 se muestra un esquema de conexionado del regulador de tensión.

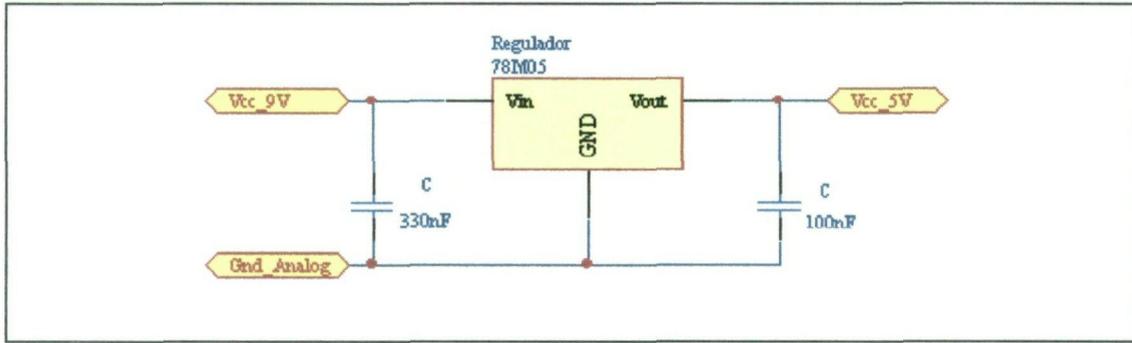


Figura 5.10 Esquema de conexionado del regulador de tensión 78M05.

5.2.5 Conectores y circuito de reset

El circuito de reset se encarga de reiniciar el microcontrolador y el módulo Bluetooth simultáneamente. Esta formado por dos resistencias de *pull-up* de **2,2 K Ω** y dos condensadores de **100 pF** configurados en dos redes RC paralelas e independientes que se activan a la vez mediante un pulsador doble como el de la figura 5.11.

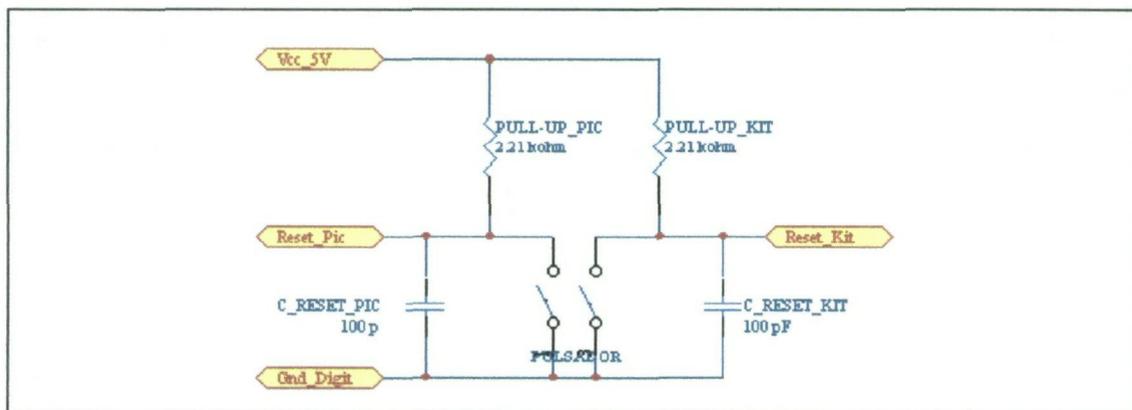


Figura 5.11 Esquema del circuito de reset del sistema

Los condensadores están conectados a la patilla **/MCLR** del microcontrolador y al conector de alimentación y *jumpers* de la placa en la que se integra el módulo Bluetooth, en el cual se encuentra la línea de activación del reset.

La PCB sobre la que se monta el microcontrolador dispone de 5 conectores de 10 líneas cada uno. En la figura 5.12 se muestra la posición de cada uno de ellos sobre la placa. El **conector de adquisición de señal ECG** tiene habilitadas solamente 4 líneas, a las cuales se conectan los electrodos **RA**, **LA**, **LL** y **RL**. Recuérdese que este último terminal se conecta a la masa analógica del sistema. Del **conector de alimentación** se utilizan exclusivamente 2 líneas, correspondientes a los terminales positivo y negativo de la pila o batería empleada. El **conector de expansión del puerto B** permite conectar las 8 líneas del puerto B del microcontrolador a una placa de LEDs u otro tipo de testigo o indicador. Esto resulta de utilidad en la tarea desarrollo y depuración del programa que se vaya a ejecutar en el microcontrolador. Los conectores que restan son el **conector de alimentación y reset del módulo Bluetooth**, mediante el cual el módulo puede ser alimentado y eventualmente reiniciado, y el **conector del interfaz RS-232** por el cual se produce el intercambio de datos entre el host y el módulo Bluetooth.

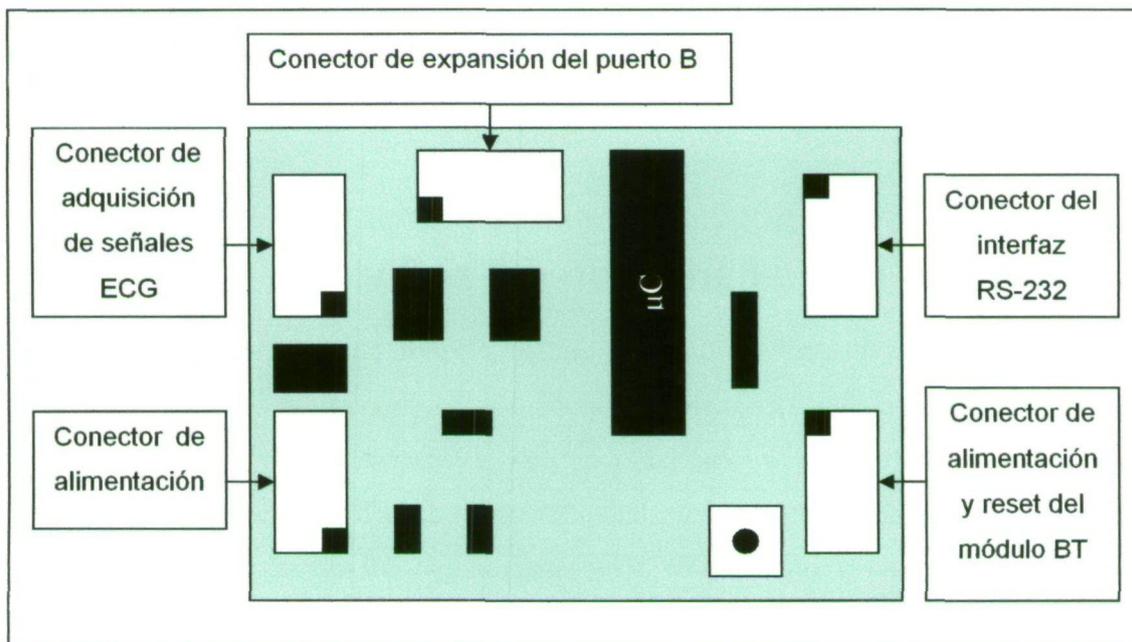


Figura 5.12 Distribución de los conectores sobre la placa de la unidad portátil.

En negro se marca el pin 1 de cada conector.

5.2.6 Características de la PCB

La placa diseñada es una PCB de doble capa, aunque los componentes se colocan solamente por una cara. Tiene unas dimensiones de 6 x 8 cm, pero en un diseño

final se podrían reducir este tamaño utilizando exclusivamente componentes SMD y colocándolos por ambas caras.

Desde un punto de vista eléctrico la PCB se caracteriza por poseer dos **planos de masa** distintos. Uno **analógico** y otro **digital**. El primero sirve como referencia para los elementos que componen los canales de acondicionamiento. El segundo es utilizado por el microcontrolador y el módulo Bluetooth, así como por el transceptor RS-232. Esto evita que la señal de alta frecuencia generada por el circuito de reloj del microcontrolador provoque ruidos o parásitos indeseados en los elementos analógicos. Las dos masas se interconectan en un punto cercano al conector de alimentación de la placa. Aunque es aconsejable realizar la interconexión de ambos planos de masa mediante una bobina que atenúe las componentes de alta frecuencia provenientes de la masa digital, la componente resistiva de la bobina puede provocar una **caída de tensión** en la misma debido a la **corriente de retorno** que puede hacer esta técnica inviable. En el caso de este proyecto, para evitar caídas de tensión, constatadas para los diversos valores de inductancia de las bobinas probadas, y puesto que no se observaron componentes de ruido de alta frecuencia apreciables, se optó por no introducir la bobina de desacoplo.

5.3 Diseño e implementación del software

El software desarrollado para la unidad portátil consiste básicamente en un programa realizado en lenguaje ensamblador destinado a ejecutarse en el microcontrolador PIC16F876. Por tanto ha sido necesario el aprendizaje previo del juego de instrucciones del microcontrolador, así como de la configuración interna de sus recursos de memoria y registros de programación [4]. El entorno de programación utilizado es el MPLAB IDE 6.2 que ofrece gratuitamente la compañía Microchip. Este software proporciona las herramientas necesarias para editar código en lenguaje ensamblador, depurarlo y compilarlo. La programación del microcontrolador se ha realizado con la ayuda del software IC-PROG, también de distribución gratuita, y un programador tipo JDM que se conecta al PC a través del puerto serie.

En el programa en ensamblador desarrollado se implementan las capas HCI, L2CAP, SDP y RFCOMM, además de la aplicación encargada de digitalizar la señal ECG y transmitirla, tal como se especifica para el perfil puerto serie [1]. Hay que decir que el host implementado no integra todos los recursos y funcionalidades definidos por el estándar Bluetooth para las distintas capas que lo componen. Esto se debe no sólo a limitaciones temporales en la realización del proyecto, sino también a limitaciones de memoria del microcontrolador utilizado. Debido a las características propias de la programación en lenguaje ensamblador y a la estructuración en capas o niveles del host Bluetooth en función de los distintos protocolos usados, la estructura del programa implementado es sumamente jerárquica. En el nivel más alto se encuentran la parte del código relacionado con las tareas que desempeña la entidad de **control** y aquellas propias de la **aplicación de usuario**. Esta parte del código realiza llamadas a otras rutinas de nivel inferior, relacionadas a menudo con las distintas capas de la pila de protocolos. Por ejemplo, tomemos el caso del establecimiento de una conexión con otro dispositivo. Para ello la entidad de control realiza una llamada a una rutina perteneciente a nivel L2CAP, que a su vez realiza una llamada otra subrutina que se encarga de llamar a las rutinas pertenecientes a la capa HCI. Como se observa el anidamiento de rutinas es muy importante. En este capítulo no se presenta el código desarrollado, sino que se describen brevemente las rutinas más importantes que conforman el programa. El código se incluye en el CD adjunto a esta memoria.

5.3.1 Implementación de la capa HCI

5.3.1.1 Driver de bus físico

La **capa de transporte o bus físico** utilizado por HCI (ver capítulo 3) es una UART. Por tanto existen una serie de rutinas en el programa que se encargan de habilitar y gestionar este periférico. Estas rutinas son utilizadas por otras rutinas de nivel superior que llevan a cabo la transmisión y recepción de paquetes a HCI. Las rutinas para el control de la UART son:

- **RX_TX:** sirve para configurar la UART para una velocidad de transmisión de 57600 Kbps, 1 bit de stop, ninguno de paridad y 8 bits de datos.
- **EN_TX:** habilita la UART para transmitir datos.
- **DIS_TX:** deshabilita UART para transmitir datos.

- **EN_RX:** habilita la UART para recibir datos.
- **DIS_RX :** deshabilita la UART para recibir datos.
- **RX:** cada vez que la UART recibe un dato, se produce una interrupción que es atendida mediante la ejecución de esta rutina que se encarga de transferir el **byte** recibido a memoria de datos (RAM). La posición de memoria en la que coloca el dato es un parámetro establecido por rutinas de nivel superior.
- **TRANSM:** se encarga de colocar en el registro de transmisión de la UART el **byte** a transmitir. Esta rutina también es utilizada por rutinas de nivel superior.
- **ESPERA:** cada vez que se encarga a la UART la transmisión de un **byte** se llama a esta rutina que aguarda a que se produzca la interrupción por fin de transmisión.

5.3.1.2 Capa HCI

Las rutinas que se explican a continuación se encuentran por encima de las rutinas de control de la UART. Por tanto para llevar a cabo las operaciones de transmisión y recepción de datos hacen uso de dichas rutinas. Y a su vez las rutinas pertenecientes a L2CAP utilizan las rutinas pertenecientes a HCI. En este nivel se encuentran las siguientes rutinas:

- **TX_HCI:** esta rutina se encarga de transmitir tanto comandos HCI como paquetes de datos de capas superiores. Para ello hace uso de las rutinas **TRANSM** y **ESPERA**. Para ello hay que colocar previamente en memoria de datos el paquete de datos a transmitir. Los parámetros de entrada necesarios para que esta rutina funcione correctamente son la **dirección de comienzo** del paquete y la **longitud** del mismo. El **buffer de datos** comienza en la dirección **0xA0**, aunque el paquete de datos a transmitir puede cargarse a partir de ésta u otra más alta.
- **RX_PACK:** esta rutina se utiliza para recibir eventos HCI y paquetes de datos de capas superiores. La rutina encargada de colocar los datos recibidos en memoria cada vez que la UART eleva una interrupción por la recepción de un byte es **RX**. La rutina **RX_PACK** se encarga simplemente de analizar la cabecera del paquete recibido para determinar si es un **evento** o un **paquete de**

datos, así como la **longitud** del mismo para saber cuando se ha terminado la recepción. Los datos recibidos se cargan también a partir de la dirección **0xA0**.

- **DATA_HCI**: se utiliza en la transmisión de paquetes de datos HCI. Para ello hace uso de la rutina **TX_HCI**. Necesita como parámetros de entrada el identificador de conexión manejada o *handle*, el valor de los campos **PB** y **BC** (ver capítulo 3) y la **longitud** del campo de datos (ver capítulo 3). Esta rutina es utilizada por las rutinas de nivel superior de la capa L2CAP para transmitir comandos y paquetes de datos.

En este proyecto se implementan únicamente los **comandos HCI** imprescindibles para que la unidad portátil pueda establecer una conexión con un dispositivo remoto asumiendo el papel de maestro. Para ello hacen uso de la rutina **TX_HCI**. Los comandos HCI implementados son:

- **RESET**: realiza un reset del chip Bluetooth.
- **INQUIRY**: sirve para descubrir otros dispositivos Bluetooth cercanos.
- **CREATE_CONNECTION**: se encarga de, una vez descubierto algún dispositivo Bluetooth, intentar establecer una conexión.

Para establecer una conexión no se llama directamente a las rutinas **INQUIRY** y **CREATE_CONNECTION**, sino que para ello existe una rutina denominada **HCI_ConnectReq** que se encarga de ello. Esta rutina es utilizada por la rutina encargada de establecer la conexión a nivel L2CAP en caso de que no exista una conexión física previamente establecida. Después de llamar a las rutinas **INQUIRY** y **CREATE_CONNECTION** es necesario llamar a la rutina **RX_PACK** para que se encargue de procesar la respuesta por parte del dispositivo remoto.

5.3.2 Implementación de la capa L2CAP

Las rutinas que conforman el nivel L2CAP se dividen en 3 categorías: rutinas que sirven de **interfaz con las capas superiores**, rutinas de **procesado de paquetes L2CAP** y rutinas **auxiliares**.

Las **rutinas de interfaz con las capas superiores (L2CA_XXX)** son utilizadas por las capas SDP y RFCOMM, así como por la entidad de control. Ésta última se encarga de crear y configurar las conexiones lógicas, mientras que SDP y RFCOMM solamente hacen uso de la rutina de las rutinas de transmisión y recepción de paquetes de datos L2CAP. Las rutinas definidas dentro este grupo son las siguientes:

- **L2CA_ConnectReq:** petición de establecimiento de una conexión o canal lógico.
- **L2CA_ConnectRsp:** respuesta a una petición de conexión.
- **L2CA_ConfigReq:** petición de configuración de una conexión.
- **L2CA_ConfigRsp:** respuesta a una petición de configuración.
- **L2CA_DisconnectReq:** petición de desconexión de un canal lógico.
- **L2CA_DisconnectRsp:** respuesta a una petición de desconexión de un canal lógico.
- **L2CA_DataWriteReq:** petición para el envío de datos de capas superiores.
- **L2CA_DataReadReq:** petición para leer datos de un canal lógico.

Las **rutinas de procesamiento de paquetes L2CAP (L2CAP_XXX)** son utilizadas por las rutinas de interfaz con las capas superiores. Las rutinas L2CAP_XXX son utilizadas por las rutinas L2CA_XXX. Se encargan de crear en memoria los mensajes o paquetes de datos L2CAP y llamar a la rutina **DATA_HCI** para llevar a cabo su transmisión. Las rutinas implementadas son las siguientes:

- **L2CAP_ConnectReq:** genera y transmite un comando *connection request*.
- **L2CAP_ConnectRsp:** genera y transmite un comando *connection response*.
- **L2CAP_ConfigReq:** genera y transmite un comando *configuration request*.
- **L2CAP_ConfigRsp:** genera y transmite un comando *configuration response*.
- **L2CAP_DisconnectReq:** genera y transmite un comando *disconnection request*.
- **L2CAP_DisconnectRsp:** genera y transmite un comando *disconnection response*.
- **L2CAP_Data:** se encarga de transmitir datos de capas superiores. Es utilizada por la rutina **L2CA_DataWriteReq**.

- **L2CAP_espera_llegada:** esta rutina hace uso de **RX_PACK** y es utilizada por la rutina **L2CA_DataReadReq** en la recepción de datos, aunque también es utilizada por el resto de rutinas **L2CA_XXX** para procesar la respuesta que la capa **L2CAP** remota da a un determinado comando. También se encarga de recibir los eventos **HCI** que se producen durante la fase de establecimiento de un canal físico (*paging*) cuando se llama a la rutina **L2CA_ConnectReq**.

Finalmente se encuentran las **rutinas auxiliares** que son utilizadas tanto por las rutinas de interfaz con las capas superiores como por las de procesado de paquetes **L2CAP**. Entre ellas destacan las siguientes:

- **HCI_ConnectReq:** es utilizada por la rutina **L2CA_ConnectReq** para crear un canal físico en caso de que este no exista, para así poder establecer un canal lógico sobre él. Se encarga de llamar a las rutinas **HCI** encargadas de esta tarea.
- **GESTION_DE_CANALES:** esta rutina es utilizada tanto por rutinas **L2CA_XXX** como por las **L2CAP_XXX**, y se encarga de crear y destruir canales lógicos. En memoria de datos existen dos variables que funcionan como registros de control y configuración, y determinan el funcionamiento de esta rutina. Son las variables **MEM_CANALES** y **ESTADO_CANALES**.
- **BUSCA_Y_DIRECCIONA_Ch:** es utilizada por las rutinas **L2CA_XXX** de configuración y desconexión de canal. Se encarga de buscar el **CID** remoto correspondiente a un determinado **CID** local. Para ello utiliza **GESTION_DE_CANALES**, que es quien se encarga en último término de devolver el **CID** remoto como parámetro de salida.

5.3.3 Implementación de la capa SDP

Las rutinas pertenecientes al nivel **SDP** se encargan básicamente de cargar en memoria de datos los mensajes **SDP**, enviarlos y procesar la respuesta del servidor **SDP** del dispositivo Bluetooth remoto. En este proyecto se implementan solamente dos rutinas relacionadas con el protocolo **SDP**. Estas son:

- **SDP_ServiceSearchRequest:** esta rutina crea en memoria de datos un mensaje **SDP_ServiceSearchRequest** a partir de la posición de memoria **0xA9** y lo

transmite haciendo uso de la rutina del nivel L2CAP **L2CA_DataWriteReq**. A continuación aguarda una respuesta y la recibe empleando la rutina **L2CA_DataRead**. Una vez recibida la respuesta la analiza y obtiene el valor del *ServiceRecordHandle* del servicio solicitado, en este caso el servicio **puerto serie**.

- **SDP_ServiceSearchAttributeRequest**: una vez el *ServiceRecordHandle* del servicio puerto serie es conocido, esta rutina se encarga de cargar a partir de la posición **0xA9** de memoria un mensaje *SDP_ServiceSearchAttributeRequest* y transmitirlo. Para ello también hace uso de rutina **L2CA_DataWriteReq**. La respuesta es recibida mediante la rutina **L2CA_DataRead** y analizada para obtener el valor del *ServerChannel* que tiene asignado el servicio puerto serie en el dispositivo remoto.

Evidentemente con las rutinas implementadas la unidad portátil no puede desempeñar la función de **servidor SDP**, solamente la de **cliente SDP**. Las rutinas de la capa SDP son utilizadas por la **entidad de control** del host Bluetooth implementado. Para ello debe previamente haber creado un canal lógico haciendo uso de las rutinas de la capa L2CAP.

5.3.4 Implementación de la capa RFCOMM

En la implementación de las rutinas pertenecientes a la capa RFCOMM tampoco se da soporte a todos los posibles mensajes o tramas definidas para este protocolo. Solamente se tienen en cuenta aquellas que son imprescindibles para el establecimiento de la conexión RFCOMM y la transmisión de datos provenientes de la capa de aplicación. Las rutinas creadas son:

- **RFCOMM_ConnectionRequest**: esta rutina se encarga de cargar una trama SABM a partir de la posición de memoria dada por un parámetro de entrada y transmitirla. Para ello utiliza la rutina **L2CA_DataWriteReq**. Esta rutina es llamada por la entidad de control para solicitar el establecimiento de una canal RFCOMM.

- **RFCOMM_MSC:** esta rutina crea en memoria, a partir de una dirección también dada por un parámetro de entrada, una trama UIH conteniendo un comando MSC y la transmite, haciendo uso de la rutina **L2CA_DataWriteReq**. Esta rutina también es llamada por la entidad de control del host.
- **RFCOMM_Data:** esta rutina es utilizada por la capa de aplicación y se encarga de crear la cabecera de las tramas UIH que contienen los datos de la señal ECG enviados. Para la transmisión de esta trama se hace uso de la rutina **L2CA_DataWriteReq**. La posición de comienzo en memoria del paquete también es configurable.

Existen, además de las anteriores, dos rutinas auxiliares que se utilizan en la creación de alguno de los campos de las tramas RFCOMM. Son las siguientes:

- **Addr_RFCOMM:** esta rutina es utilizada por la rutina **RFCOMM_Data** para crear el campo de **dirección**. Para ello emplea el valor del número de **canal de servidor** o *Server Channel*. Las otras rutinas que se encargan de transmitir las tramas SABM y MSC no tienen necesidad de utilizar esta rutina porque el campo de dirección de estas tramas viene determinado por el valor del **canal de señalización**, que toma siempre el mismo valor (**0x0001**).
- **FCS_RFCOMM:** esta rutina calcula el campo FCS de las tramas UIH y de las SABM que no pertenezcan al canal de señalización. Es utilizada por la rutina **RFCOMM_Data** y por la **entidad de control** para la transmisión las tramas SABM y MSC pertenecientes al canal de datos, ya que en este último el campo de dirección viene determinado por el valor del *Server Channel* suministrado por el servidor SDP del dispositivo remoto. En la implementación de esta rutina no se ha seguido el algoritmo que se presenta en el documento de especificación del estándar Bluetooth, sino que se ha utilizado un algoritmo alternativo basado en una serie de operaciones sobre una tabla de 256 valores que requiere de cálculos menos complejos.

5.3.5 Implementación de la aplicación

La aplicación se encarga de llevar a cabo la conversión analógica-digital de las señales II y III provenientes de los canales de acondicionamiento, para que puedan ser posteriormente transmitidas a la unidad de visualización ECG.

En primer lugar hay que señalar que el microcontrolador PIC16F876 dispone de un solo conversor para 5 canales de entrada distintos. En este caso sólo se utilizan 2 de estos canales de entrada. Por tanto las señales II y III no se pueden muestrear simultáneamente, sino de forma secuencial. En la tarea de muestrear las señales interviene el **Timer0** del microcontrolador. Este periférico se configura de forma que provoque una interrupción cada **2,5 ms**. La rutina del servicio de interrupción del Timer0 habilita el conversor analógico-digital y se produce el muestreo de uno de los dos canales utilizados, en este caso **AN0** ó **AN1** [4]. Puesto que los canales se muestrean de forma alterna, se produce un retardo de **5 ms** entre la adquisición de dos muestras consecutivas de cada una de las señales. Este valor se corresponde con la inversa de la **frecuencia de muestreo de la señal ECG**, que en este proyecto se establece en **200 muestras/s**.

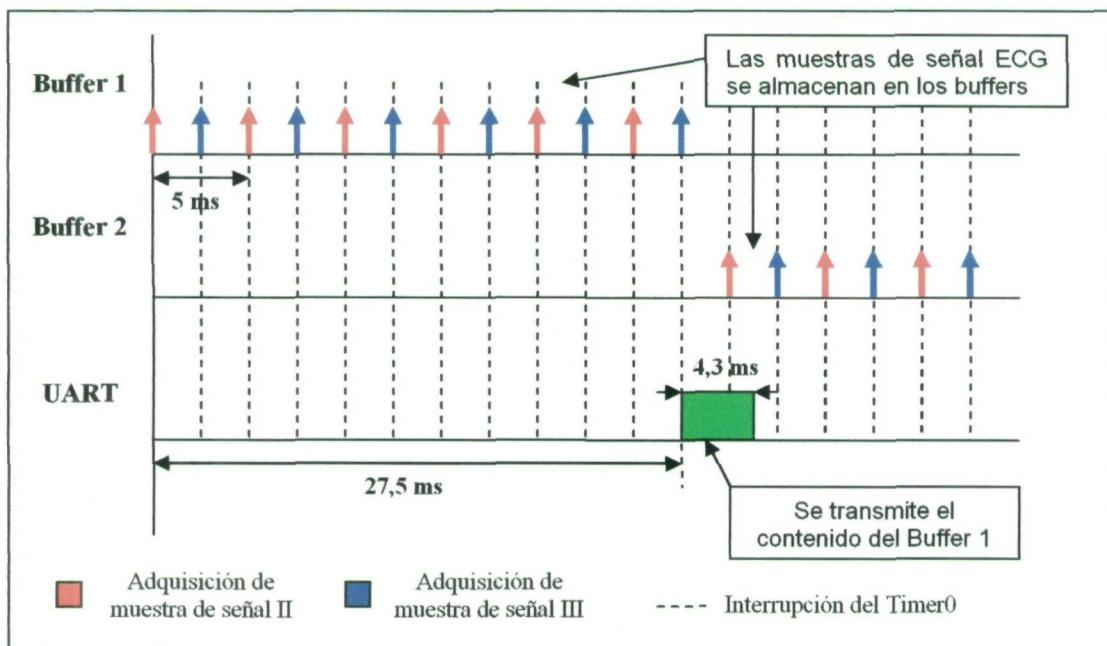


Figura 5.13 Diagrama temporal de la transmisión de las muestras de señal ECG.

Cada una de las **dos muestras** obtenidas se codifica utilizando **8 bits** de resolución, y es transferida automáticamente a la memoria de datos. En memoria de datos existen **dos buffers** de datos, cada uno con capacidad para **12 muestras**, seis por cada canal. El objetivo de utilizar dos buffers distintos es el de poder transmitir el contenido de uno de ellos mientras el otro se llena. El tiempo que tarda cada uno de estos buffers en llenarse representa la mayor parte del **retardo** que sufre la transmisión de la señal ECG. Cada buffer tarda poco más de **27,5 ms** en llenarse debido a que debe esperar a que se produzcan **doce** conversiones (el tiempo empleado en la conversión de la última muestra es despreciable). En la figura 5.13 se muestra el funcionamiento de este sistema.

En la transmisión de los paquetes de datos por parte de la UART se produce otro retardo significativo, aunque mucho menor que el relacionado con el llenado de los buffers. La velocidad de transmisión con que se configura la UART es de **57600 bps**. A cada byte de datos a transmitir se le anexa un bit de paridad y otro de stop, con lo que se transmiten 10 bits en total. Por tanto se tiene una velocidad de transmisión de **5760 bytes/s**. El tiempo que tarda en transmitirse un byte es la inversa de dicho valor, es decir **173,61 µs**. Si se tienen 13 bytes de cabecera en el paquete de datos a nivel HCI más 12 de datos, resulta que en la transmisión de los 25 bytes resultantes se emplean **4,3 ms**. Con estos valores se obtiene, aproximadamente un **retardo máximo total en la transmisión de la señal ECG de 31,8 ms**, que puede considerarse como un valor bastante aceptable.

El hecho de transmitir **12 muestras de datos** (6 por cada señal) en cada paquete está relacionado con el número de muestras que necesitan los algoritmos de la unidad de visualización para poder generar una muestra de señal válida sin tener que esperar a la recepción de otro paquete. Este punto se verá con más detalle en el capítulo 6. Por otro lado, y teniendo en cuenta que la cabecera a nivel HCI está formada por 13 bytes, enviar un número menor de muestras haría la transmisión ineficiente.

El ancho de banda utilizado habitualmente en el ECG de monitorización llega hasta los 50 Hz [3], por lo que bastaría con muestrear a 100 muestras/s, pero en este

proyecto se elige una frecuencia de muestreo de 200 muestras/s debido a que, como se verá en el próximo capítulo, una de las características de la aplicación de visualización ECG es la capacidad de almacenar en un fichero en formato ASCII el valor de las muestras de señal ECG para su posterior estudio. Por tanto es interesante tener una frecuencia de muestreo alta para conseguir una mayor resolución de la señal almacenada.

En cuanto a las rutinas que se implementan en el nivel de aplicación, la más importante es **Generador_de_Buffer**. Una vez que el enlace Bluetooth se ha establecido satisfactoriamente entre la unidad portátil y el PC, esta rutina empieza a ejecutarse de forma continua. Se encarga del llenado alternativo de los dos buffers de datos existentes en memoria RAM, y de la transmisión de su contenido en cuanto uno de ellos esté lleno, realizando una llamada a la rutina de nivel inferior **RFCOMM_Data**.

La rutina **Generador_de_Buffer** se encarga también de controlar la inicialización del Timer0 mediante una llamada a la rutina **Establece_Timer0**, de forma que éste genere una interrupción cada 2,5 ms. Además esta subrutina se encarga de habilitar el conversor AD cada vez que es llamada, de forma que se pueda realizar la adquisición de una nueva muestra de señal. La rutina de servicio asociada a la interrupción por fin de conversión, denominada **AD_INT**, se encarga de determinar en cada instante el siguiente canal que se va a muestrear (AN0 ó AN1).

En la figuras 5.14 y 5.15 se muestran los diagramas de flujo correspondientes a las rutinas de CONTROL y de la aplicación encargadas del establecimiento de la conexión y de la gestión de los buffers, así como de las rutinas de interrupción del Timer0 y del conversor AD asociadas a la ejecución de la rutina de aplicación.

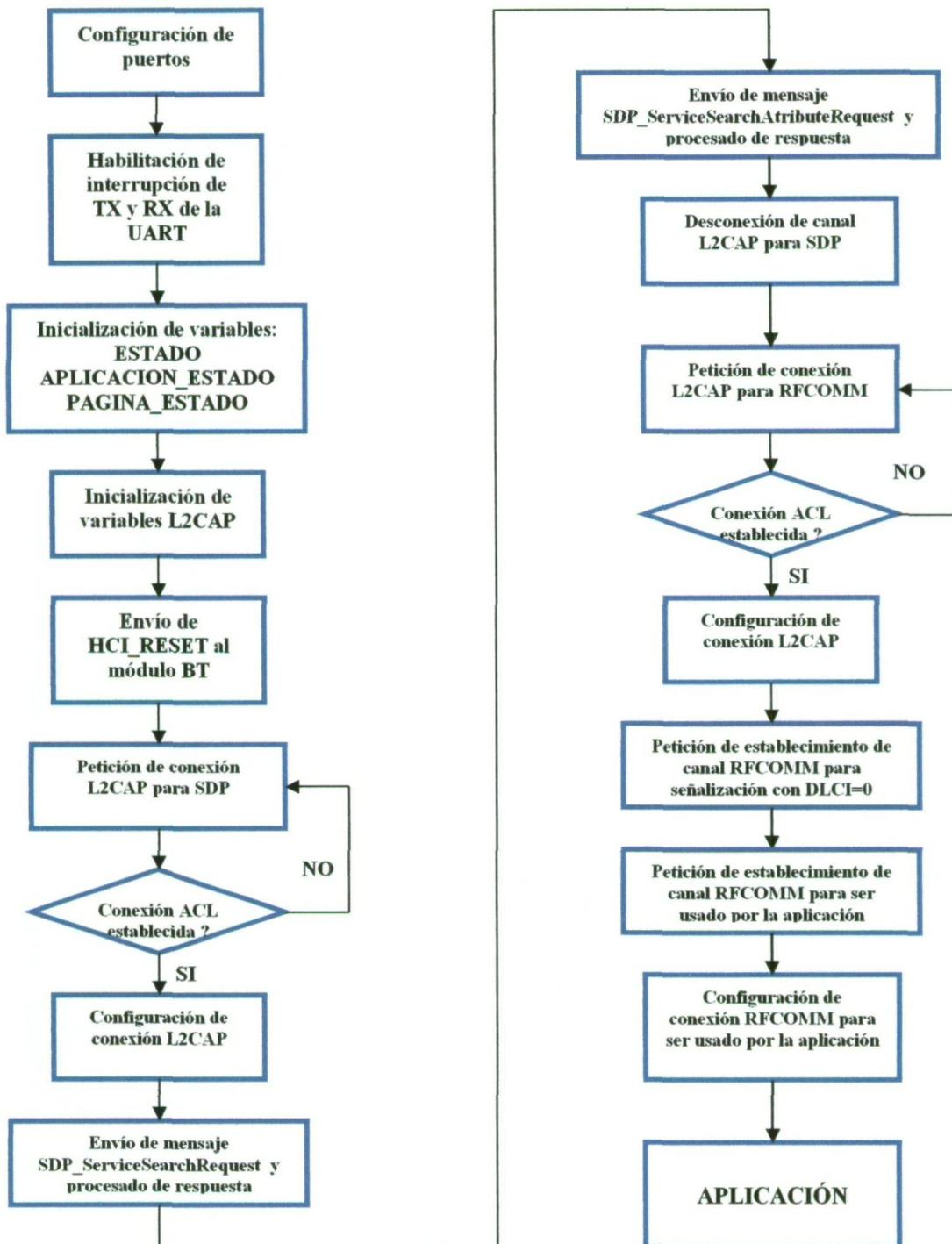


Figura 5.14 Flujograma de la fase de establecimiento de conexión por parte de la entidad de CONTROL.

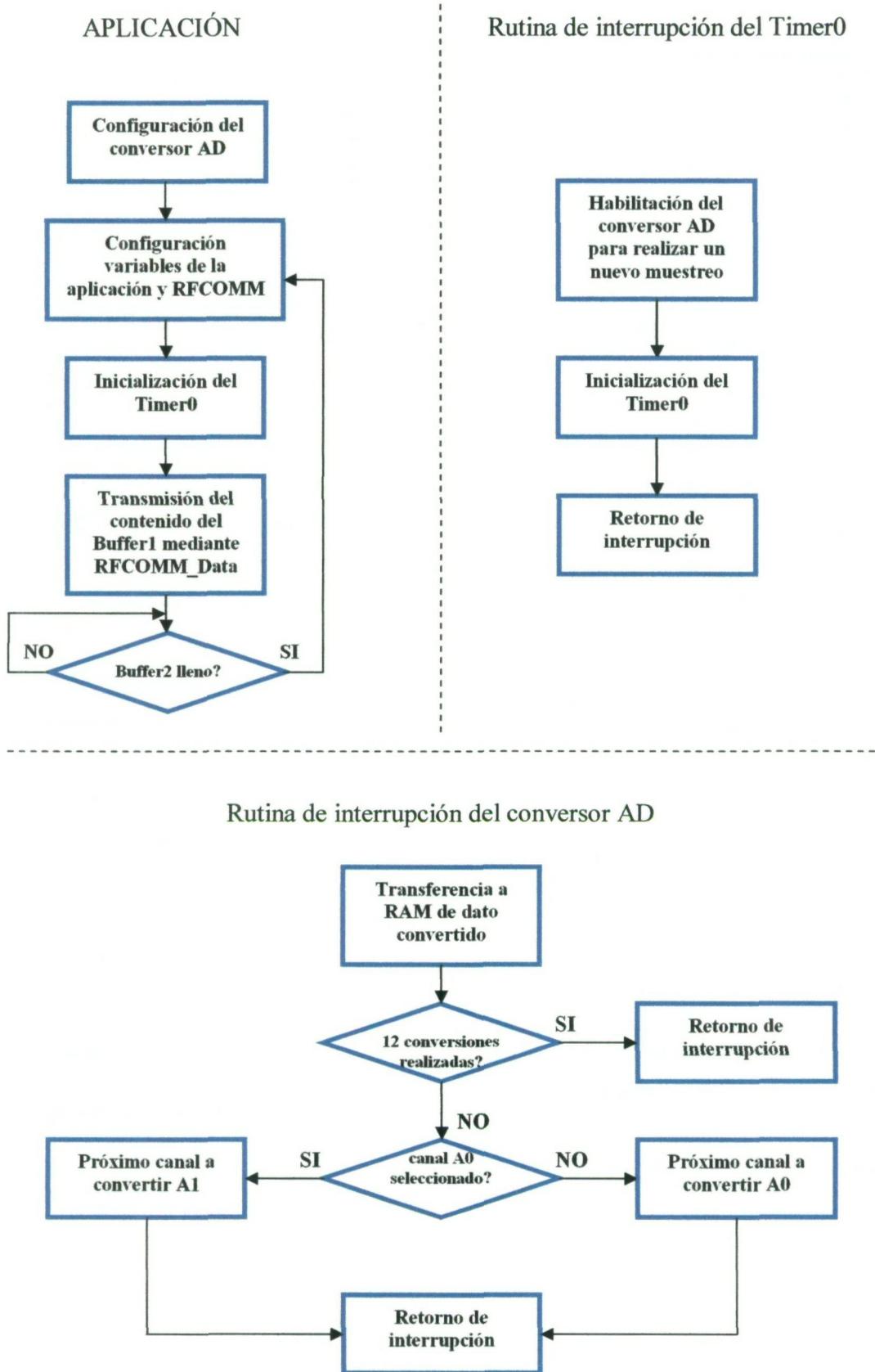


Figura 5.15 Flujogramas de la rutina de aplicación y de las rutinas de interrupción asociadas.

5.4 Conclusiones

El diseño e implementación de la unidad portátil ha sido uno de las tareas más importantes desarrolladas en este proyecto. Este elemento está compuesto de diferentes bloques, cada uno de ellos con una misión específica. En primer lugar se encuentra la **etapa de acondicionamiento de señal ECG**, formada por filtros y amplificadores. En este bloque se hace uso de amplificadores de instrumentación **AD623AN** y amplificadores operacionales **AD8542AR** de Analog Devices. Los filtros empleados son simplemente filtros de tipo RC. El **microcontrolador PIC16F876** se encarga digitalizar las señales ECG y gestionar el enlace Bluetooth. El **módulo Bluetooth** en el que se monta el chip **ROK 101 008** de Ericsson desempeña tareas de control del enlace de bajo nivel e incluye la interfaz radio de la unidad portátil. La **etapa de alimentación** utiliza un regulador de tensión **L-78M05CDT** que se encarga de generar la tensión de 5 voltios necesaria para alimentar al resto de elementos de la unidad. El **circuito de reset** sirve para reiniciar tanto el módulo BT como el microcontrolador. Finalmente están los **conectores** que permiten alimentar la unidad, conectar los electrodos de adquisición de señal ECG e interconectar el módulo BT con el resto de elementos de la unidad portátil.

En los anexos de esta memoria se incluyen los esquemas de conexionado completos y detallados de los diferentes elementos que componen la PCB diseñada e implementada, así como el *layout* de la misma.

La implementación del host Bluetooth en el microcontrolador PIC16F876 resultó ser más complicado de lo esperado, debido a la dificultad de depurar el código escrito en lenguaje ensamblador. Para esta tarea no se ha utilizado ningún tipo de placa de desarrollo, sino que se ha empleado la PCB donde se montaría finalmente el microcontrolador. El host implementado dispone de los recursos básicos necesarios para poder establecer un enlace Bluetooth con el adaptador BT-USB y transmitir los datos relativos a la señal ECG. Por encima del host Bluetooth se sitúa la aplicación encargada de realizar la conversión analógica-digital y multiplexar los datos a transmitir.

En la figura 5.16 se muestran dos fotografías, una de la PCB implementada y otra de la placa del kit de desarrollo de Ericsson utilizada en este proyecto. Ambos elementos componen la unidad portátil desarrollada.

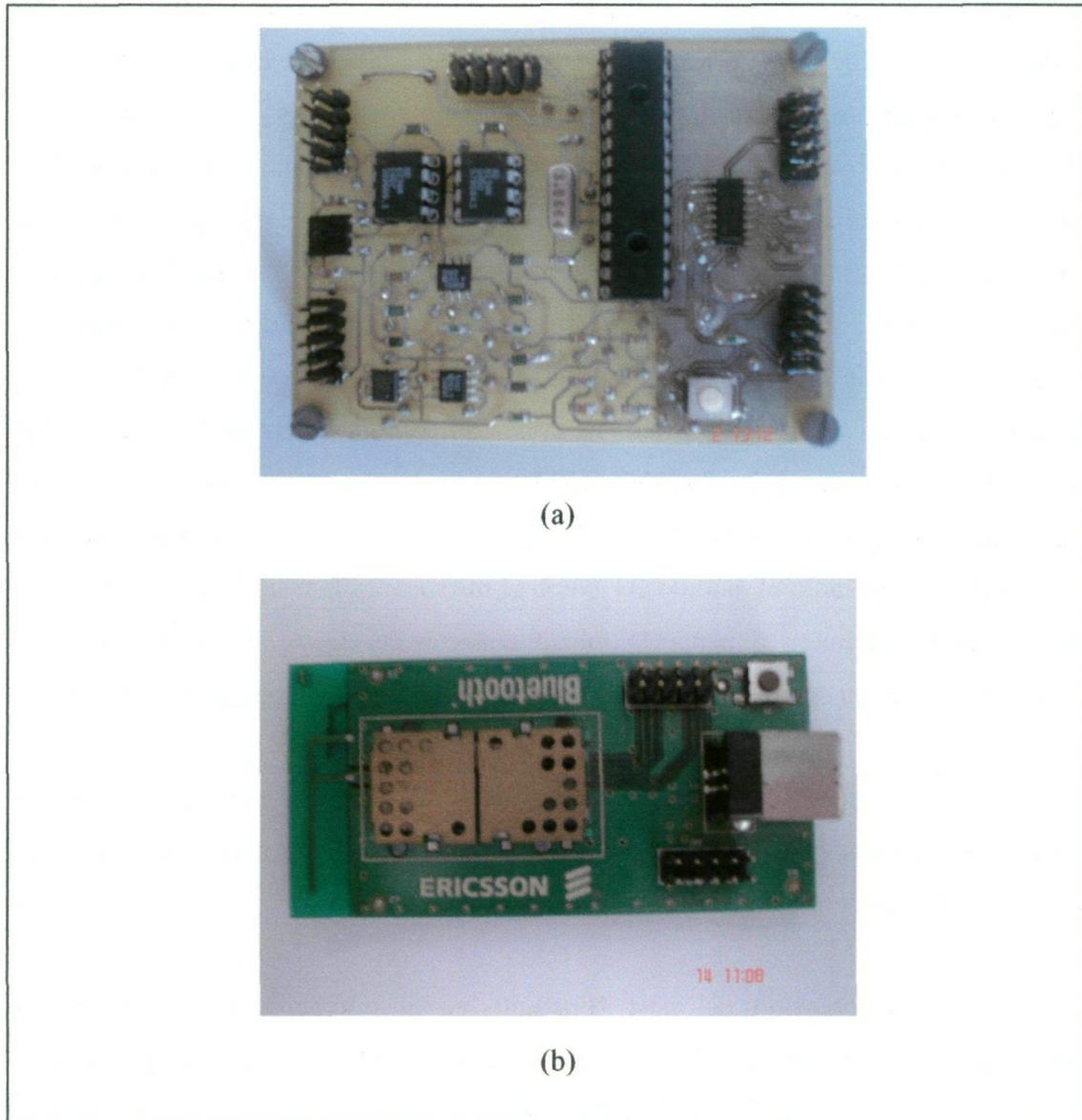


Figura 5.16 Fotografías de la PCB realizada (a) y del módulo BT (b) que conforman la unidad portátil

CAPÍTULO 6

UNIDAD DE VISUALIZACIÓN ECG

6.1 Introducción

En este capítulo se describen algunos de los elementos que conforman la **unidad de visualización ECG**. Básicamente se trata del **adaptador BT-USB**, de su software asociado y de **la aplicación de visualización**. El ordenador personal al que se conecta el adaptador BT-USB y en el que se ejecuta la aplicación de visualización no es objeto de una explicación detallada, dejando la descripción más específica de sus características técnicas para el correspondiente apartado dentro del Pliego de Condiciones que se adjunta a esta memoria explicativa. Una gran parte de este capítulo se dedica a la descripción del interfaz de usuario y del funcionamiento de la aplicación desarrollada para poder visualizar la señal ECG, así como de los algoritmos de procesado empleados. También se describe brevemente el entorno de desarrollo empleado, en este caso se trata del **C++ Builder 5** de Borland.

6.2 Adaptador Bluetooth-USB

El **adaptador Bluetooth-USB** empleado es un **dispositivo comercial** de la compañía BrainBoxes, en concreto el modelo BL-554. Este periférico cumple con la versión 1.1 del estándar Bluetooth por lo que carece de la característica EDR [7]. Por tanto la velocidad de transmisión máxima en un único sentido a nivel de bandabase es de solo 721 Kbps. En cuanto a potencia pertenece a los dispositivos de clase 2, lo que le permite establecer conexiones de hasta 50 metros de alcance. En la figura 6.1 se muestra una fotografía de este dispositivo.

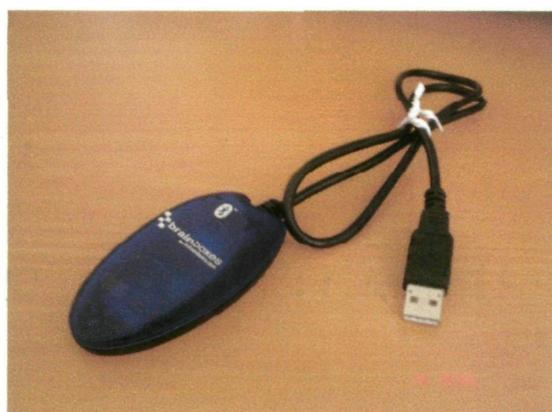


Figura 6.1 Adaptador Bluetooth-USB de BrainBoxes.

En el software que acompaña a este dispositivo se implementa el host del adaptador BT-USB. Las distintas capas de protocolos Bluetooth, desde la capa de transporte USB hasta RFCOMM, así como los distintos perfiles Bluetooth integrados, se implementan como **drivers**, que son utilizados por el sistema operativo del ordenador a la hora de llevar a cabo cualquier operación con el hardware en el que se implementa la parte de módulo Bluetooth del adaptador BT-USB.

La aplicación en la que se integra el interfaz de usuario del adaptador BT-USB permite el control y configuración de este dispositivo. En la figura 6.2 se muestra el interfaz gráfico de dicha aplicación. Dicha aplicación se encarga de gestionar las distintas funciones que permiten establecer una conexión con otro dispositivo Bluetooth. Entre las más importantes se encuentran la función *Discovery* que se utiliza para descubrir los dispositivos Bluetooth próximos. La función *Connect* permite establecer una conexión con un determinado servicio disponible en un dispositivo

remoto. Cada uno de estos servicios se relaciona con alguno de los perfiles Bluetooth definidos por el estándar. En el caso concreto de este dispositivo se incluyen entre otros el servicio *Serial Port* (asociado al perfil puerto serie) que es el utilizado en este proyecto para establecer la comunicación con la unidad portátil. Otros servicios disponibles, pero no los únicos, son el *LAN Access Point* que permite conectar el dispositivo a una red de área local; el servicio *OBEX File Transfer*, que permite transferir ficheros entre dispositivo mediante el uso del protocolo OBEX; o el servicio *Dial Up Networking*, que permite conectarse a un módem vía Bluetooth [1].

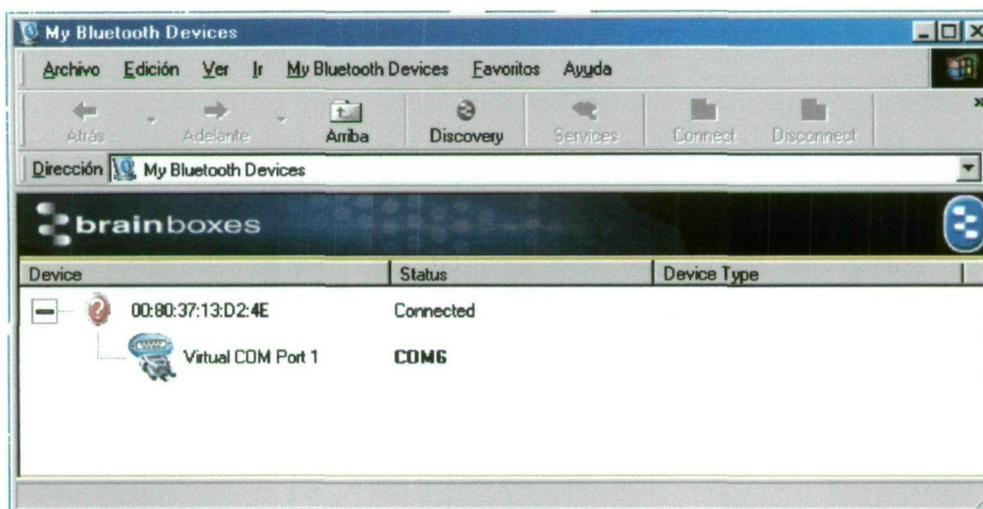


Figura 6.2 Aplicación de control del adaptador BT-USB.

En este proyecto el adaptador BT-USB tendrá asignado el papel de esclavo, ya que será la unidad portátil quien tome la iniciativa durante el establecimiento de la conexión, y por tanto actúe como maestro. Para que pueda ser establecida dicha conexión el adaptador BT-USB debe ser configurado previamente de forma específica. En el menú principal se debe seleccionar *My Bluetooth Devices*, *My Properties*. En la ventana de diálogo asociada se ha de seleccionar las opción de seguridad *Low (Mode 1)* y deshabilitar la opción de encriptación de datos. También hay que habilitar las características *Discoverable*, *Connectable* y *Auto-accept Connections*. En la figura 6.2 también se puede apreciar como cuando se establece una conexión mediante el servicio *Serial Port* con otro dispositivo, en este caso con la unidad portátil, en la interfaz gráfica de la aplicación se indica la dirección Bluetooth del dispositivo remoto y el puerto serie virtual creado (COM6) con el que debe interactuar la aplicación de visualización.

6.3 Aplicación de visualización

La aplicación de visualización permite al observador humano visualizar las señales ECG transmitidas, así como otro tipo de información derivado de las mismas. El desarrollo de esta aplicación ha sido una de las tareas más importantes llevadas a cabo en la realización de este Proyecto Fin de Carrera. El **sistema operativo** sobre el que funcionará la aplicación de visualización es **Microsoft Windows 98 SE**.

6.3.1 Entorno de programación

El entorno de programación utilizado es el **C++ Builder 5** de **Borland**. La versión utilizada puede resultar algo obsoleta, pero para los requerimientos de este proyecto sus prestaciones han sido suficientes. Esta aplicación también podría haberse desarrollado con la ayuda de Visual C# o Visual basic, pero se optó por Builder debido al conocimiento previo que tenía el proyectando de este entorno de programación.

C++ Builder 5 no es más que un **entorno de desarrollo visual** de aplicaciones GUI (*Graphic User Interface*). Permite desarrollar aplicaciones para Windows y es la propia herramienta la que se encarga de la gestión visual de los elementos incluidos en el interfaz gráfico desarrollado, como botones, menús, etc. Otra de las características más importantes de este tipo de entornos es que su ejecución está basada en **eventos**. Es decir, cuando se ejecuta la aplicación, ésta, en un primer momento, no realiza ninguna acción hasta que el usuario no provoca un evento como por ejemplo la activación de algún control o la introducción de un dato. Es entonces cuando se produce la ejecución del código asociado a dicho evento [12].

Otra de las características de este entorno de trabajo es que está basado en **objetos**. Todos los elementos gráficos disponibles (*componentes* en terminología de Builder) con los que se trabajan son objetos además de aquellos que puedan ser creados por el propio programador. Por tanto es necesario tener algunas nociones de **programación orientada a objetos** para poder utilizar esta herramienta [12].

6.3.2 Interfaz de usuario

El interfaz de usuario de la aplicación de visualización se compone de una serie de **controles de activación y configuración** de las distintas funciones de la aplicación y de un **área de visualización** compuesta por tres canales: **canal 1**, **canal 2** y **canal 3** donde se visualizan las señales I, II y III. En la figura 6.3 se muestra la interfaz diseñada.



Figura 6.3 Interfaz de usuario de la aplicación de visualización ECG.

En la parte superior izquierda de la interfaz se encuentra el control que permiten seleccionar el **puerto serie virtual** del que se deben leer los datos biomédicos correspondientes a la señal ECG. Justo debajo están los controles que permiten configurar las **escalas de amplitud y tiempo**. Los valores que pueden ser seleccionados son **1 ó 0,5 mV/div** para la amplitud y **0,5 ó 0,25 sg/div** para el eje temporal. Estos valores están referidos a las líneas de división que pueden aparecer en cada uno de los canales cuando éstas son habilitadas. Esta característica se verá más adelante con más detalle. En el área de visualización, que ocupa la mayor parte del interfaz de la

aplicación, se representan las señales I, II y III (ver capítulo 2). En la parte superior derecha bajo el anagrama BPM (*Beats Per Minute*) se indica el valor de la **frecuencia cardiaca** obtenido mediante el procesado de la señal ECG recibida.

En la parte izquierda del interfaz se encuentran los botones que controlan las acciones que puede llevar a cabo la aplicación. Son los siguientes:

- **Visualizar ECG:** al pulsar este botón la aplicación realiza la representación continua de las señales ECG en tiempo real, así como la presentación visual del parámetro de frecuencia cardiaca (BPM).
- **Pausar Visualización:** este botón sirve para detener la representación continua de la señal ECG, de forma que se mantiene en cada canal la representación visual de las señales justo en el instante de activar este control.
- **Zoom In:** permite realizar un *zoom* de una determinada zona de cada canal previamente seleccionada con el ratón. Para ello debe haberse pausado previamente la visualización de las señales.
- **Zoom Out:** permite volver a visualizar en cada canal la representación original de cada señal después de haber realizado un *zoom in*.
- **Activar/Desactivar Sonido:** este control habilita o deshabilita la indicación sonora asociada a cada latido. También existe un sonido de alarma asociado a una posible parada cardiaca.
- **Salir:** sirve para cerrar la aplicación.

Además de los elementos de control y configuración ya descritos existe un **menú principal** en la parte superior del interfaz. Este menú está compuesto por cuatro elementos. Los tres primeros, de izquierda a derecha, es decir *Canal 1*, *Canal 2* y *Canal 3* tienen asociado cada uno de ellos un **menú vertical desplegable** que incluye dos opciones: la primera es *Guardar en Fichero*, que se utiliza para almacenar en un fichero ASCII con extensión *txt* el valor de las muestras de la señal ECG visualizada en el canal asociado; la segunda es *Capturar Imagen*, que permite capturar una imagen en formato *bmp* de la señal visualizada en cada uno de los respectivos canales.

El cuarto elemento del menú principal es **Configuración**. El menú desplegable asociado incluye otros dos elementos. El primero es **Visualización**, que se utiliza para configurar diversas características referentes a la visualización de las señales ECG. En la figura 6.4 se muestra la **ventana de diálogo** asociada.

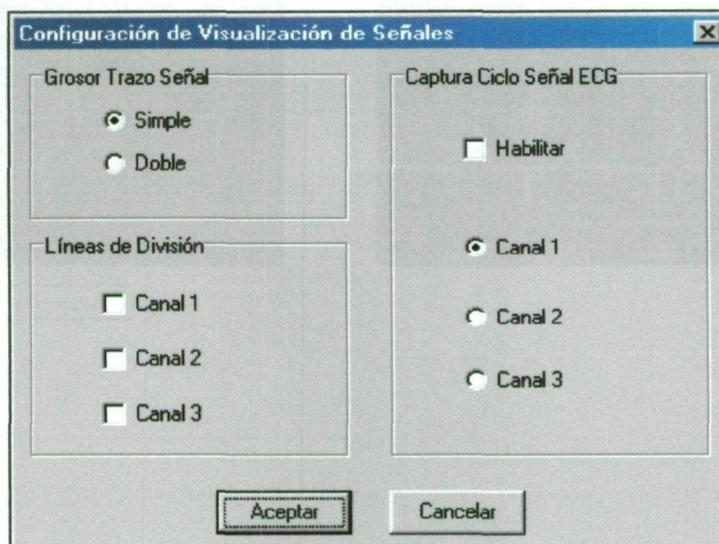


Figura 6.4 Ventana de diálogo de configuración las características de visualización.

Dentro de la ventana de configuración de visualización de señales se observan tres apartados. El apartado de **Grosor Trazo Señal** se utiliza para configurar el grosor del trazo empleado en la representación visual de las señales ECG. El apartado **Líneas de División** sirve para habilitar la visualización de una retícula calibrada numéricamente en función del valor seleccionado en los controles de configuración de amplitud y tiempos de la ventana principal. En cuanto al apartado de **Captura Ciclo Señal ECG**, éste permite visualizar en una ventana distinta de la ventana principal un solo ciclo de una de las señales ECG. Esta visualización permite observar con más detalle las características de cada una de estas señales, y se actualiza automáticamente cada vez que se completa la recepción de un nuevo ciclo de señal ECG. También incluye un botón que permite capturar en formato *bmp* la imagen visualizada en esta ventana. En la figura 6.5 se muestra el aspecto uno de los canales con las líneas de división habilitadas y la ventana de visualización del ciclo de señal ECG.

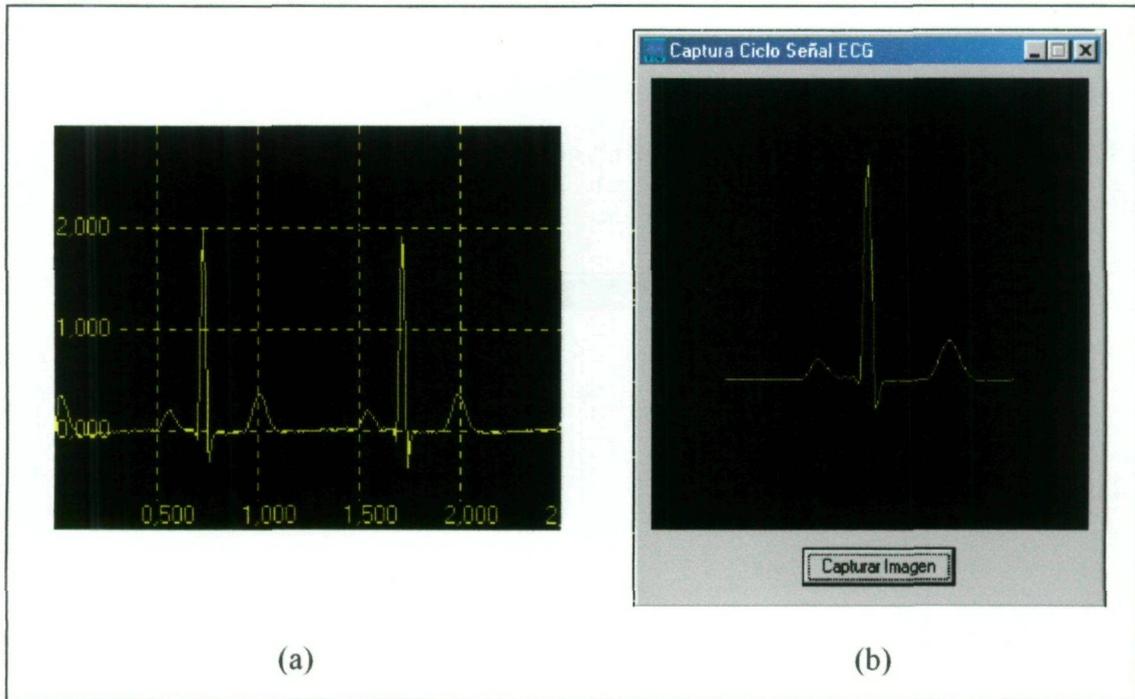


Figura 6.5 (a) Aspecto del canal 1 habilitando la visualización de las líneas de división. (b) Ventana de captura del ciclo de señal ECG.

La segunda opción dentro del menú desplegable asociado al elemento *Configuración* del menú principal es *Filtros*. En la figura 6.6 se muestra el aspecto de esta ventana.

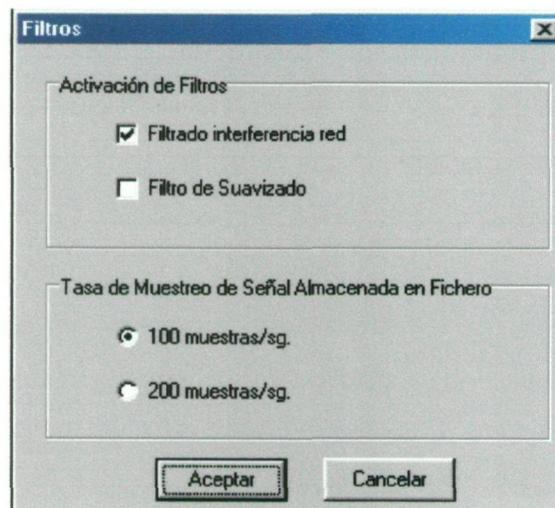


Figura 6.6 Ventana de diálogo de configuración de filtros y tasa de muestreo

En el apartado de *Activación de Filtros* existen dos casillas que permiten la habilitación del filtro de eliminación de la interferencia de red de 50 Hz y de un filtro de

suavizado que puede ser de utilidad en situaciones en que la señal ECG sea muy ruidosa. En el próximo apartado se explicarán estos filtros más detalladamente.

El apartado de la ventana de diálogo denominado *Tasa de Muestreo de Señal Almacenada en Fichero* se refiere a la frecuencia de muestreo de la señal cuyas muestras se almacenan en un fichero cuando esta función es habilitada. La señal es muestreada originalmente en la unidad portátil a una frecuencia de **200 muestras/sg.**, pero para visualizar la señal en cada uno de los canales del área de visualización, esta señal es **diezmada por dos**, resultando una tasa de muestreo efectiva de **100 muestras/sg.** De esta forma es posible representar en cada canal el doble de ciclos de señal ECG, de forma que se facilita el estudio visual de la misma. Por tanto existe la posibilidad de almacenar en un fichero las muestras obtenidas a la frecuencia de muestreo original, o una réplica de la señal visualizada por la aplicación, muestreada a la mitad de frecuencia. Esta señal muestreada a sólo 100 muestras/sg. permite la detección de muchos problemas cardiacos, como por ejemplo las arritmias, en un estudio o procesado posterior de los ficheros obtenidos (ver ECG Holter en capítulo 2), pero con la ventaja de que los ficheros generados tienen la mitad de tamaño.

Como ya se ha mencionado, la aplicación permite realizar un *zoom* de cada una de las señales visualizadas en los canales, con la ayuda de los controles *Zoom In* y *Zoom Out*. Merece ser destacada la complejidad de la implementación de esta función, ya que tuvo que ser programada desde cero, puesto que no existía ningún método propio de los componentes visuales incluidos en Builder que realizase esta operación de forma automática. En la figura 6.7 se muestran dos imágenes correspondientes a la señal ECG visualizada en el canal 1. La figura 6.7(a) corresponde a la versión original de la misma, representada en función de la escala de amplitud y tiempos seleccionada, donde se indica la zona de la señal a ampliar, y la figura 6.7(b) es la versión ampliada de la misma después de realizar el *zoom*. Obsérvese como la calibración numérica de la retícula es modificada en función de la profundidad del *zoom*.

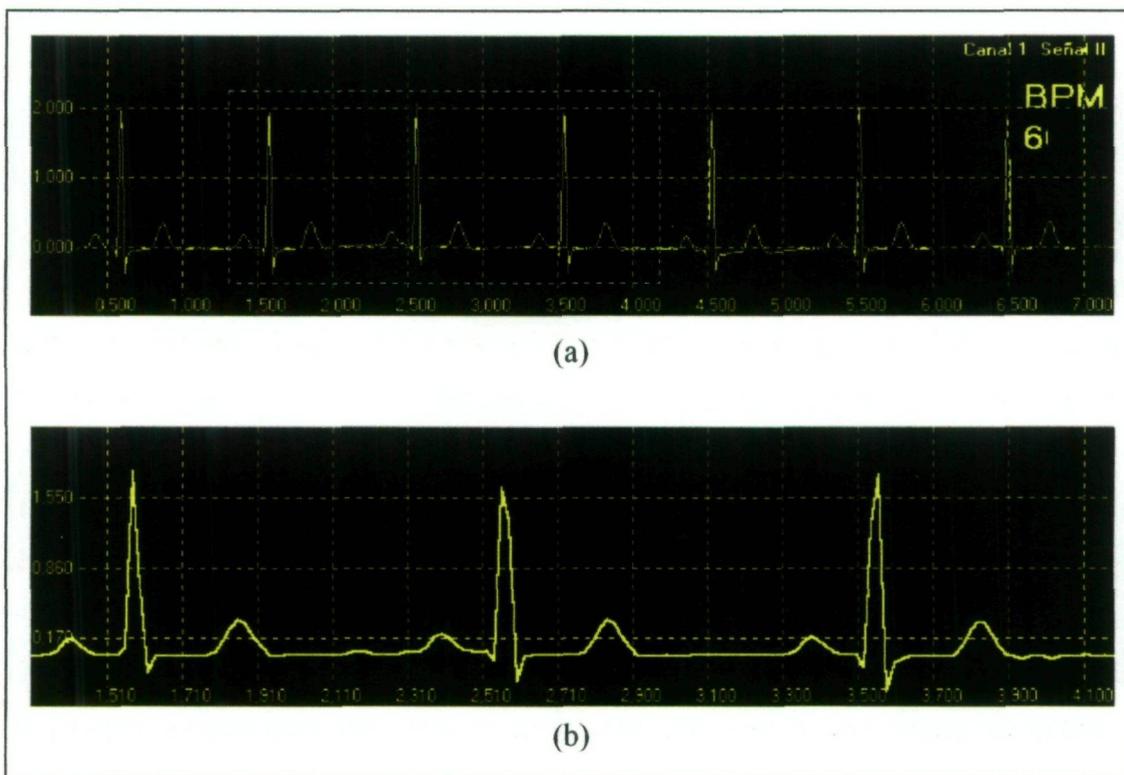


Figura 6.7 (a) Selección de la zona de la señal ECG que se desea ampliar. (b) Ampliación de la zona de la señal ECG seleccionada.

6.3.3 Procesado de la señal ECG

La correcta visualización de la señal ECG requiere de un procesado previo de los datos transmitidos por la unidad portátil. En primer lugar hay que obtener la **señal I** a partir de la **señal II** y de la **señal III**, según la expresión $I = II - III$ (ver capítulo 2). El problema es que las señales II y III están muestreadas en instantes de tiempo distintos, separados por un intervalo de **2,5 milisegundos** (ver capítulo 5). La solución es realizar una interpolación de la señal III a partir de las muestras disponibles para obtener una estimación del valor de dicha señal en el instante correspondiente al muestreo de la señal II. En este caso se utiliza una simple **interpolación lineal** a partir de las dos muestras adyacentes. Una vez llevada a cabo esta operación es posible realizar el procesado de las señales I, II y III. En la figura 6.8 se muestra el diagrama de bloques correspondiente al procesado digital aplicado a las mismas.

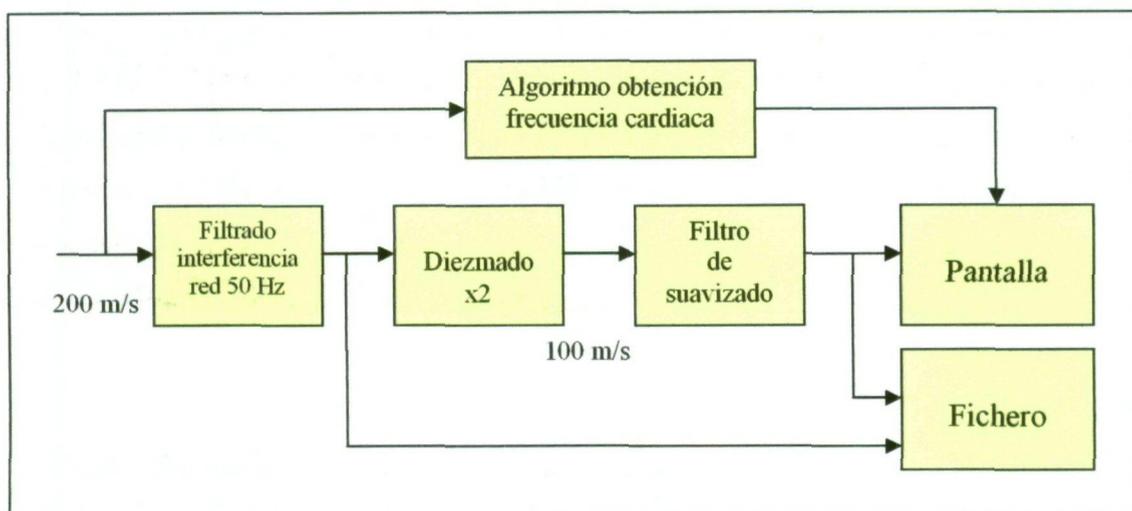


Figura 6.8 Procesado digital aplicado a la señal ECG.

En el apartado dedicado a la explicación del interfaz de usuario ya se ha mencionado que tanto el **filtrado de interferencia de red** como el **filtro de suavizado** son **opcionales**. Es decir pueden ser desactivados por el usuario cuando éste así lo desee. Además la señal que puede ser almacenada en fichero es, o bien la original muestreada a 200 muestras/sg., o la diezmada, pero no ambas a la vez.

6.3.3.1 Cancelación de la interferencia de red de 50 Hz

La cancelación de la interferencia de 50 Hz provocada por la red eléctrica se consigue mediante la aplicación de un algoritmo de **filtrado adaptativo**. En este caso se utiliza el **algoritmo de paso descendente LMS (Least Mean Square)** [10]. El filtrado adaptativo se basa en el diagrama de bloques de la figura 6.9.

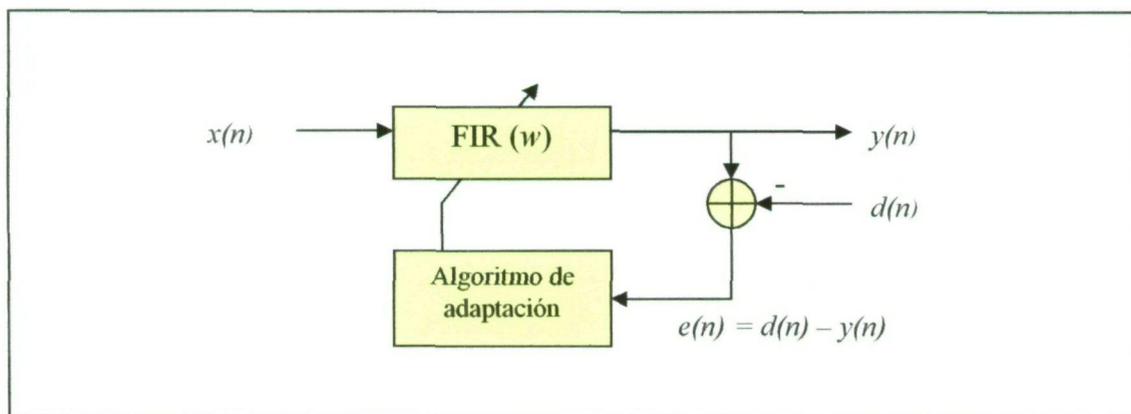


Figura 6.9 Diagrama de bloques clásico del filtrado adaptativo.

Para aplicar el esquema clásico del filtrado adaptativo al caso concreto que nos ocupa, se asigna la señal ECG con la interferencia añadida a la señal $d(n)$ y la señal $x(n)$ se iguala a un tono de referencia de 50 Hz generado internamente por la aplicación. En cuanto a la señal $e(n)$, ésta se corresponde con la señal ECG filtrada. El funcionamiento del algoritmo de adaptación se basa en la minimización de la potencia de la señal $e(n)$. Para ello es necesario que el tono de 50 Hz de referencia tenga la misma amplitud y fase que la señal interferente introducida por la red. De ello se encarga el filtro FIR, cuyos coeficientes son modificados por el algoritmo de adaptación en función de la potencia de la señal $e(n)$, que alcanza un mínimo cuando se consigue eliminar completamente la señal interferente.

En el algoritmo implementado en la aplicación se utiliza un **filtro FIR** de solamente **dos coeficientes**, y se aplica un parámetro μ igual a **0,2**. Estos valores fueron establecidos a partir de simulaciones realizadas previamente en MATLAB. En la figura 6.10 se muestran los resultados de una simulación en MATLAB de la aplicación del filtrado adaptativo. La señal ECG original se obtuvo con ayuda de la aplicación de visualización a partir de los datos transmitidos por la unidad portátil.

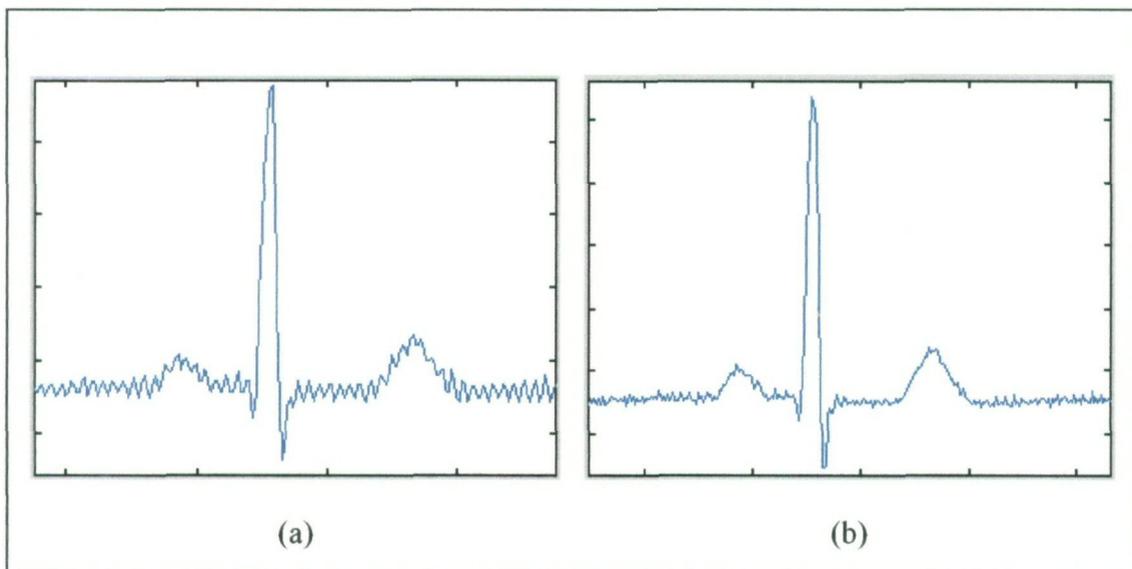


Figura 6.10 (a) Señal ECG original con interferencia de 50 Hz. (b) Señal ECG filtrada mediante la técnica de filtrado adaptativo.

6.3.3.2 Diezmado

La señal ECG muestreada a 200 muestras/sg. es **diezmada por dos** para facilitar su representación visual en pantalla. Además así se ajusta el ancho de banda de la señal al utilizado habitualmente en el ECG de monitorización (ver capítulo 2) con lo que se elimina algo de ruido. Para ello se filtra la señal con un **filtro de Hanning** cuya **frecuencia de corte** se encuentra en **50 Hz** y posteriormente se realiza el **muestreo**. El filtro de *Hanning* empleado no es más que un **filtro FIR paso bajo** cuya ecuación en diferencias es:

$$y(n) = (1/4) \cdot [x(n) + 2x(n-1) + x(n-2)] \quad (6.1)$$

En la figura 6.11 se muestra la señal ECG en la que se ha eliminado la interferencia de red de 50 Hz (correspondiente a la que aparece en la figura 6.10(b)) antes y después de ser diezmada.

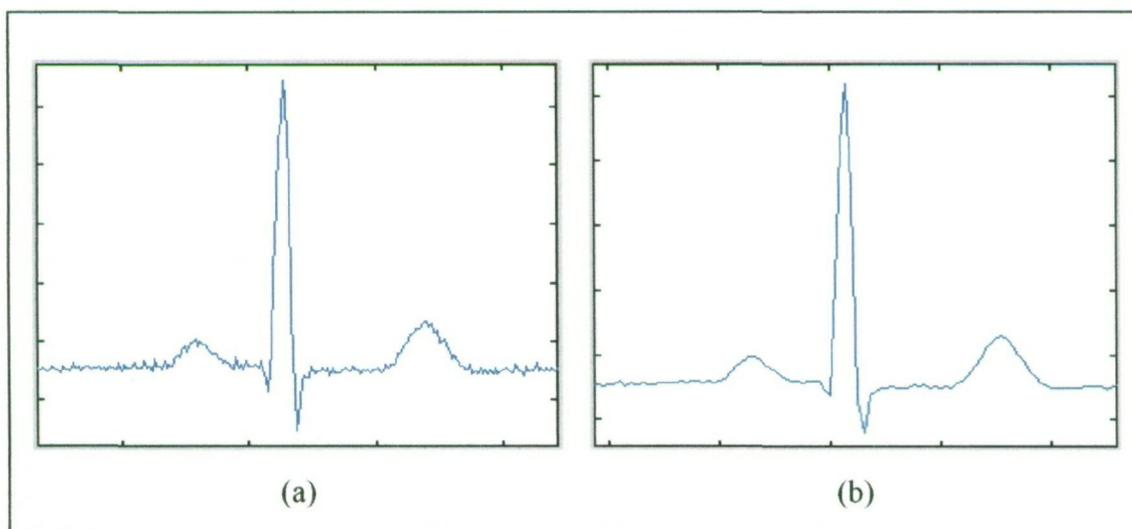


Figura 6.11 (a) Señal ECG a la salida del filtro cancelador de la interferencia de red. (b) Señal ECG diezmada por dos.

6.3.3.3 Filtro de suavizado

El filtro de suavizado empleado es un **filtro FIR pasobajo** cuya **frecuencia de corte** se encuentra en **19 Hz**. Puede resultar de utilidad en entornos muy ruidosos. Al igual que el filtro cancelador de la interferencia de red, puede ser desactivado por el

usuario en los casos en que no se necesite. Su comportamiento viene dado por la siguiente ecuación en diferencias:

$$y(n) = (1/21) \cdot [-2x(n) + 3x(n-1) + 6x(n-2) + 7x(n-3) + 6x(n-4) + 3x(n-5) - 2x(n-6)] \quad (6.2)$$

En la figura 6.12 se muestra el resultado de filtrar la señal ECG diezmada. Para poder apreciar mejor el comportamiento de este filtro se procedió a añadir **ruido blanco gaussiano** a la señal ECG original (correspondiente a la mostrada en la figura 6.10(a)). El efecto de este ruido se puede observar comparando la señal de la figura 6.11(b) y la de la figura 6.12(a). El **valor máximo de potencia de ruido** para el que la señal ECG filtrada (figura 6.12(b)) mantiene una calidad aceptable es de alrededor de **1,6 mW**.

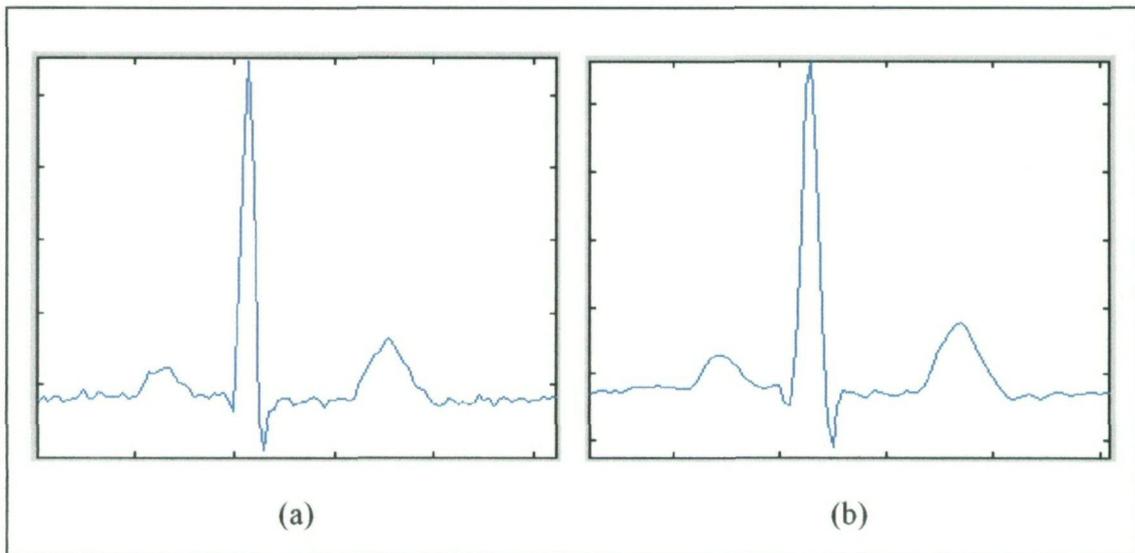


Figura 6.12 (a) Señal ECG diezmada antes de ser filtrada. (b) Señal ECG diezmada después de ser filtrada.

6.3.3.4 Obtención de la frecuencia cardíaca

El algoritmo de obtención de la frecuencia cardíaca, implementado en la aplicación de visualización, se basa en un algoritmo de **detección del complejo QRS** en señales ECG desarrollado por Pan y Tompkins [11]. En la figura 6.13 se muestran los elementos que componen este algoritmo.

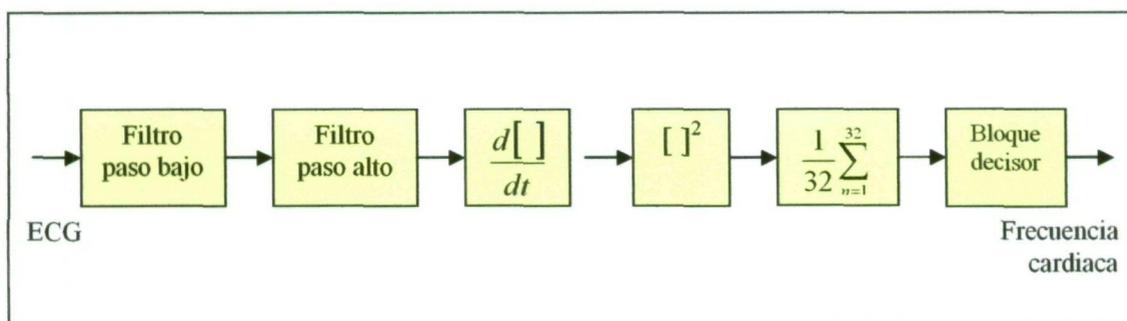


Figura 6.13 Estructura del algoritmo de detección de complejos QRS.

Las dos primeras etapas del filtro conforman un filtro paso banda cuya misión es dejar pasar exclusivamente las componentes espectrales características del complejo QRS. La frecuencia de corte del **filtro paso bajo** es de **11 Hz** y su ecuación en diferencias viene dada por la expresión 6.3.

$$y(n) = (1/32) \cdot [2y(n-1) - y(n-2) + x(n) - 2x(n-6) + x(n-12)] \quad (6.3)$$

El **filtro paso alto** posee una frecuencia de corte de **5 Hz** y su ecuación en diferencias es la descrita por la expresión 6.4.

$$y(n) = y(n-1) - (1/32) \cdot x(n) + x(n-16) - x(n-17) + (1/32) \cdot x(n-32) \quad (6.4)$$

El siguiente bloque se encarga de realizar el cálculo de la **primera derivada** de la señal filtrada. El complejo QRS se distingue de las ondas P y T por presentar pendientes muy elevadas que se manifiestan en forma de máximos o mínimos relativos en la señal diferenciada. La fórmula de derivación aplicada es la siguiente.

$$y(n) = (1/8) \cdot [2x(n) + x(n-1) - x(n-3) - 2x(n-4)] \quad (6.5)$$

El siguiente paso consiste en calcular el **valor cuadrático** de la señal diferenciada. De esta forma los picos negativos de la derivada (básicamente debidos a la pendiente del tramo descendente de la onda R) aparecen con valores positivos.

A continuación se procesa la señal con un bloque **promediador**. Este elemento se encarga de calcular el valor medio de la señal sobre una ventana de 32 muestras. Esta

ventana se desplaza sobre la señal de entrada al bloque promediador, de forma que se consigue unificar en uno solo los posibles picos de la señal previamente elevada al cuadrado. En la figura 6.14 se muestra el aspecto de la señal promediada. Cada uno de los picos de amplitud de esta señal se relaciona con un complejo QRS.

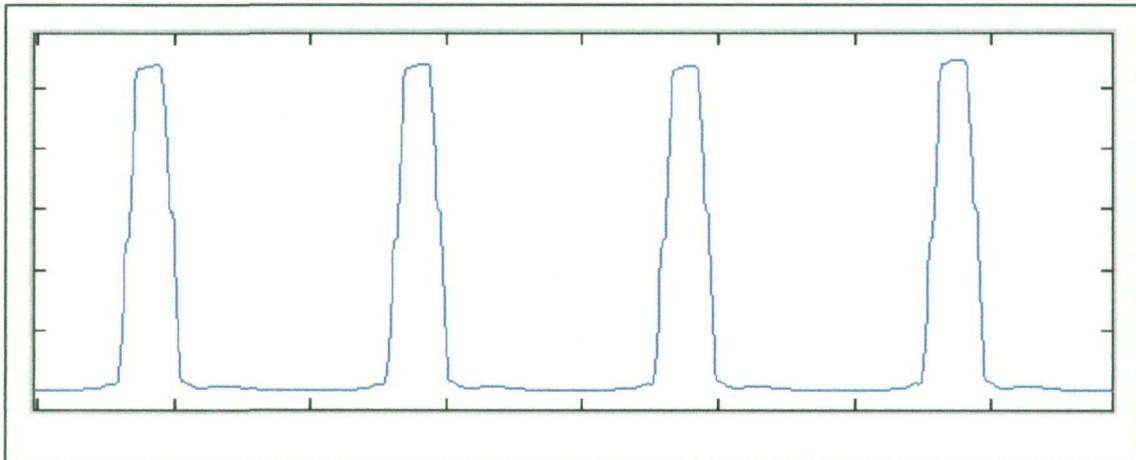


Figura 6.14 Señal a la salida del bloque promediador.

El **bloque decisor** se encarga de determinar si un pico de señal aparecido en la señal promediada se corresponde con la aparición de un complejo QRS, y por tanto con un latido. Este bloque se basa en un **umbral de decisión variable** que se adapta a posibles variaciones de amplitud de la señal promediada.

6.4 Conclusiones

La **unidad de visualización ECG** permite monitorizar la señal ECG y el ritmo cardiaco del paciente bajo observación. Esta unidad está formada por un adaptador Bluetooth-USB conectado a un ordenador personal. Durante la realización de este proyecto se ha desarrollado una aplicación windows que se encarga de la representación visual de la señal ECG en la pantalla del ordenador. Esta permite realizar las siguientes operaciones:

- Visualizar las señales I, II y III de un electrocardiograma.
- Obtener y representar visualmente el valor de la frecuencia cardiaca.
- Asociar a cada latido una indicación sonora.

- Generar una alarma sonora en caso de parada cardiaca.
- Modificar las escalas de tiempo y amplitud de los canales donde se representan las señales ECG.
- Realizar un *zoom* de las señales visualizadas en cualquiera de los canales.
- Capturar una imagen en formato *bmp* de la visualización de cualquiera de los canales.
- Almacenar en un fichero ASCII con extensión *txt* las muestras de cualquiera de las señales visualizadas, con dos posibles valores para la frecuencia de muestreo: 200muestras/s y 100 muestras/s.
- Visualización en una ventana independiente un único ciclo de cualquiera de las señales ECG visualizadas.
- Eliminación de la interferencia de red de 50 Hz de la señal ECG.
- Filtrado de suavizado de la señal ECG.

Con las características arriba expuestas esta aplicación permite llevar a cabo no sólo la monitorización cardiaca de un paciente, sino también realizar un estudio de las señales ECG visualizadas en busca de posibles anomalías.

CAPÍTULO 7

MEDIDAS Y RESULTADOS

7.1 Introducción

En este capítulo se presentan los resultados obtenidos durante las pruebas efectuadas en el laboratorio relativas al funcionamiento del sistema desarrollado. Con ello se pretende proporcionar un conjunto de datos y observaciones que posibiliten la evaluación objetiva de las prestaciones del sistema, así como del trabajo realizado en las tareas de diseño e implementación del mismo.

Los resultados presentados se dividen básicamente en dos categorías. La primera se refiere al establecimiento y control del enlace Bluetooth. La segunda trata de la calidad de la representación visual de la señal ECG en la unidad de visualización. Este apartado resulta más complicado de abordar debido que esta calidad es evaluada en gran parte en base a los criterios subjetivos de un observador humano. Por esta razón se ha intentado establecer criterios más objetivos relacionados con la presencia de ruido o interferencias no deseadas en la señal ECG visualizada.

7.2 Evaluación del enlace Bluetooth

Uno de los principales objetivos planteados inicialmente en este Proyecto Fin de Carrera era la implementación de los recursos básicos propios de un host Bluetooth, de forma que pudiera establecerse una conexión con un dispositivo Bluetooth remoto. Es evidente que el cumplimiento de este objetivo era un requisito imprescindible en la realización de este proyecto, y que **dicho objetivo ha sido completado de forma totalmente satisfactoria.**

En este punto habría que destacar la dificultad y complejidad que ha entrañado la correcta programación de los protocolos y procedimientos de intercambio de mensajes entre la unidad portátil y el adaptador BT-USB. Una de las principales causas de ello es la dificultad de **depurar** el código desarrollado en lenguaje ensamblador.

Cuando se produce un error en el establecimiento del enlace Bluetooth, éste puede deberse bien a un error en el mensaje enviado (que puede ser un comando HCI, un paquete L2CAP, un mensaje SDP, etc.), o bien a un error en el procesamiento de la respuesta enviada por el dispositivo remoto. Estos errores sólo pueden ser detectados en tiempo de ejecución, debido a que se hace necesaria la interacción con el adaptador BT-USB.

Inicialmente la estrategia adoptada para afrontar el problema de la depuración del código fue la de habilitar uno de los puertos de E/S del microcontrolador conectándolo a una batería de LEDs, de forma que pudiese mostrarse algún tipo de indicación luminosa en función de los eventos producidos durante la ejecución del programa. Esta estrategia pronto demostró ser lenta e ineficiente.

Al final el método empleado en la monitorización del intercambio de mensajes fue el desarrollo de una aplicación con la ayuda de *C++ Builder*, que a través de uno de los puertos serie físicos de un PC es capaz de mostrar el flujo de datos entre el host y el módulo BT de la unidad portátil (recuérdese que el elemento de unión entre host y módulo no es más que un interfaz RS-232, el mismo que usan los puertos serie del ordenador). Para ello es necesario conectar algunas de las líneas del cable de unión entre host y módulo a uno de los puertos serie de un PC.

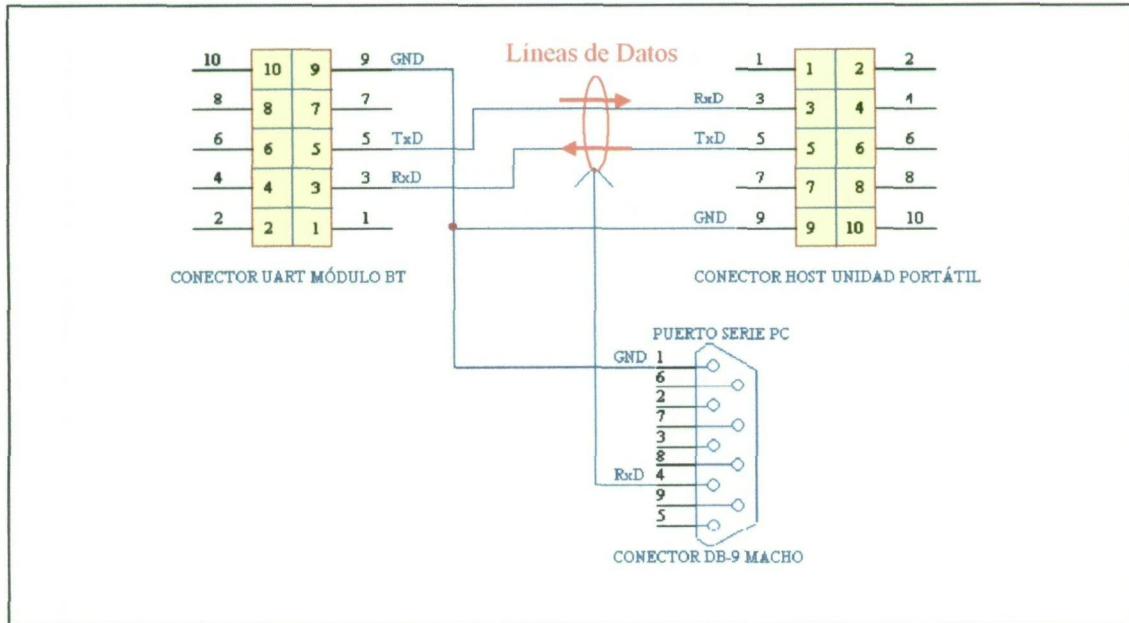


Figura 7.1 Diagrama de conexión del puerto serie del PC al interfaz RS-232 entre host y módulo BT.

En la figura 7.1 se muestra como se debe realizar la conexión del interfaz entre host y módulo BT de la unidad portátil con el puerto serie del PC. La numeración de los pines de los conectores de la placa que monta el módulo BT y la PCB que integra el host se corresponde con la expuesta en la descripción del hardware de la unidad portátil (ver capítulo 5). La línea **RxD** del puerto serie del ordenador se debe conectar a una de las dos líneas de datos que interconectan el módulo y el host Bluetooth, en función del flujo de datos que se desee monitorizar.

En la figura 7.2 se muestra la interfaz de la aplicación desarrollada, en la que se observan los valores hexadecimales con los que se codifican los mensajes intercambiados entre host y módulo. Evidentemente es necesario tener un conocimiento previo de la codificación de los diferentes tipos de paquetes Bluetooth para poder interpretar los datos visualizados.

En uno de los anexos de esta memoria se incluye la descripción de todos paquetes intercambiados entre host y módulo (a nivel HCI) obtenidos con la ayuda de esta aplicación.

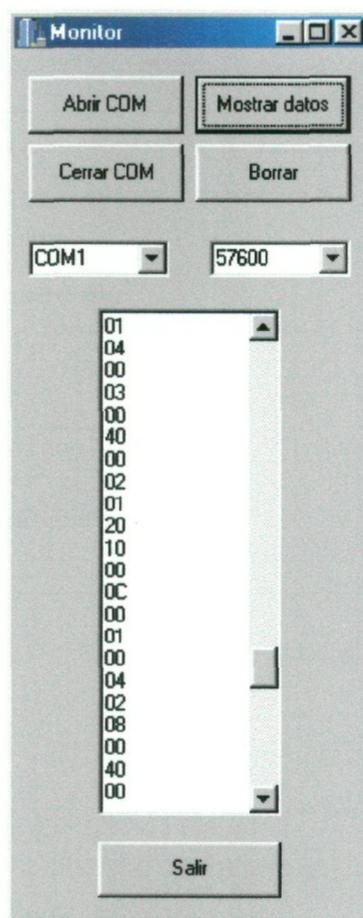


Figura 7.2 Interfaz de la aplicación de monitorización del intercambio de paquetes HCI.

Durante las pruebas en el laboratorio la separación entre la unidad portátil y la unidad de visualización era habitualmente de aproximadamente **1 metro**. También se hicieron pruebas aumentando la distancia de separación hasta **más de 6 metros**, sin que se observaran problemas para poder establecer la conexión. Hay que recordar que tanto el módulo Bluetooth de Ericsson utilizado en la unidad portátil, como el adaptador BT-USB pertenecen a la **clase 2** y permiten establecer enlaces con visión directa de hasta 50 metros. En cualquier caso normalmente la separación entre el paciente y el equipo de monitorización suele ser de pocos metros. Durante todo el proceso de desarrollo del sistema no se observaron desvanecimientos del enlace Bluetooth ni la aparición de errores en la transmisión de los datos biomédicos. Esto se debe en gran parte a la gran robustez de los sistemas de espectro ensanchado empleado por Bluetooth y a los esquemas de detección de errores implementados en la capa de bandabase.

Ocasionalmente se observó la imposibilidad de establecer la conexión entre la unidad portátil y el adaptador BT-USB debido a problemas del software de gestión del adaptador o al propio sistema operativo del PC que se bloqueaba, requiriendo del reinicio del mismo.

7.3 Evaluación de la señal ECG visualizada

Para poder evaluar la calidad de la señal ECG visualizada se procedió a obtener muestras de la señal ECG en dos puntos de los **canales de acondicionamiento** de la unidad portátil (ver capítulo 5), según se muestra en la figura 7.3, para posteriormente compararlas con la señal ECG visualizada en el PC.

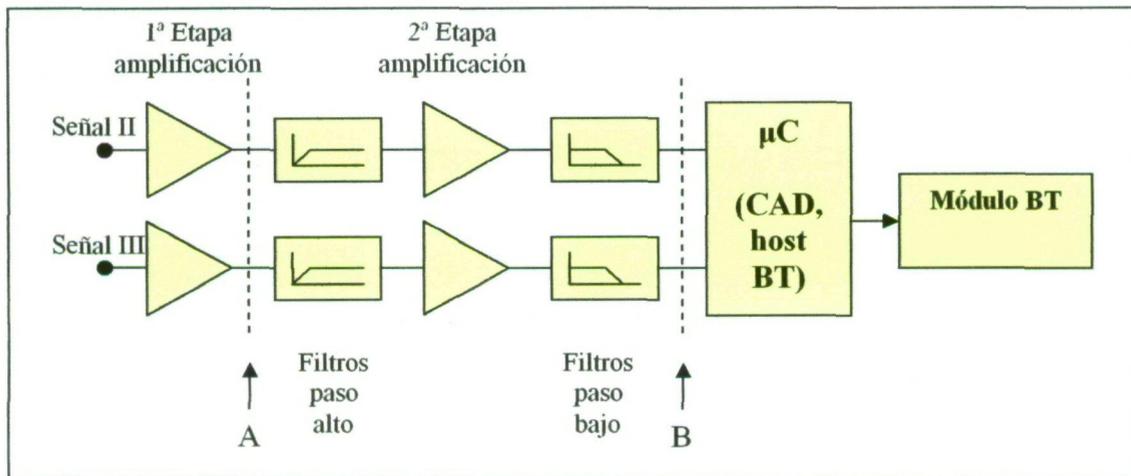


Figura 7.3 Puntos A y B donde se toman las muestras de la señal ECG de los canales de acondicionamiento de la unidad portátil.

Primeramente se obtuvo una muestra de la señal ECG a la salida de los **amplificadores de instrumentación (punto A)**. Este es el punto donde se puede obtener una imagen más fiel de la señal ECG original, ya que la única alteración sufrida por la señal hasta ese momento es la amplificación diferencial, y por tanto conserva gran parte del ruido y de las interferencias que acompañan a la señal ECG. El otro punto donde se obtuvo una segunda muestra de la señal ECG es la salida de los **filtros paso bajo (punto B)**, que constituyen la última etapa de los canales de acondicionamiento, y se sitúan justamente antes de la entrada al **convertor AD**.

En la figura 7.4 se muestran las **señales II y III** obtenidas con la ayuda del osciloscopio digital a la salida de los amplificadores de instrumentación.

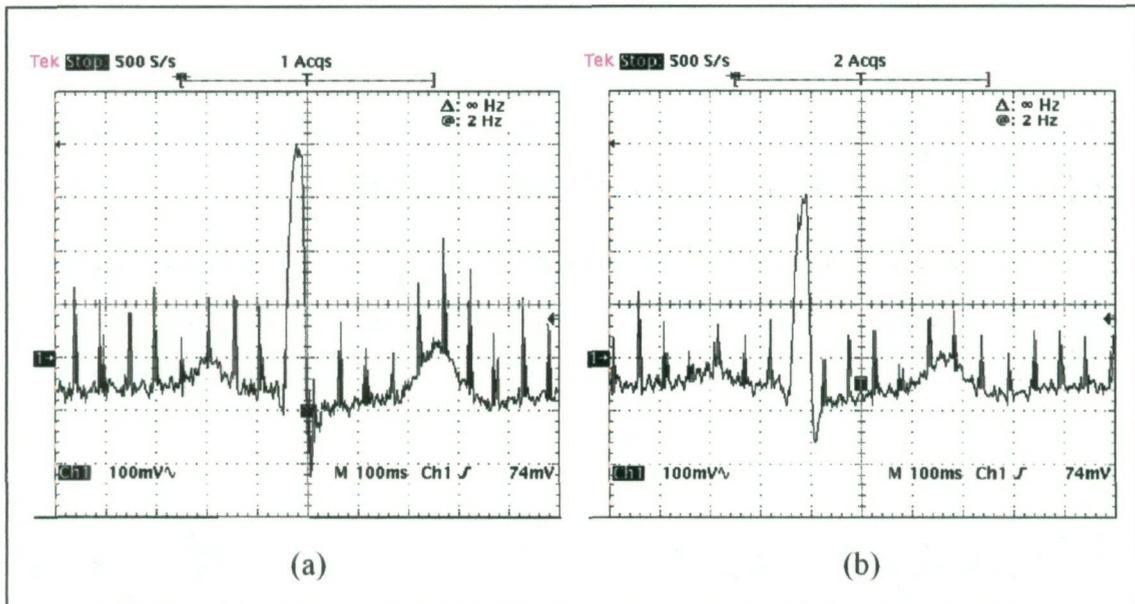


Figura 7.4 (a) Señal II en el punto A. (b) Señal III en el mismo punto.

En las señales de la figura 7.4 se observa la gran cantidad de ruido e interferencias que suelen acompañar a la señal ECG. En la figura 7.3 se muestra el aspecto de las señales II y III a la salida de los filtros paso bajo. Se puede observar la reducción del ruido con respecto a las señales de la figura 7.2, así como la amplificación efectuada por la segunda etapa de amplificación (ver capítulo 5).

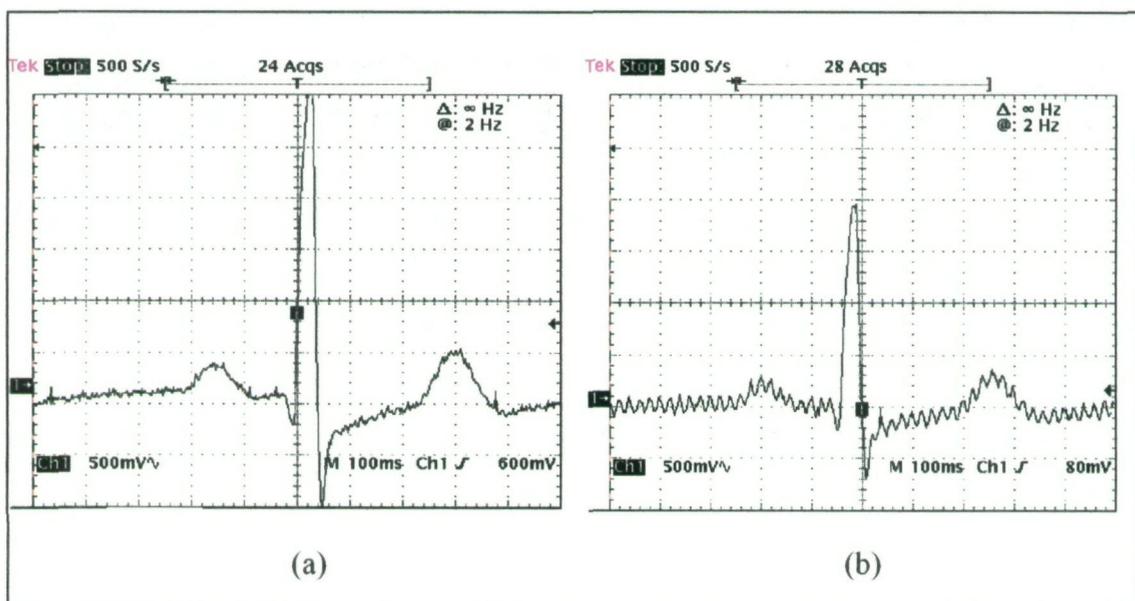


Figura 7.5 (a) Señal II en el punto B. (b) Señal III en el mismo punto.

En el capítulo 6 se mostraron algunas imágenes de las señales obtenidas con ayuda de la aplicación de visualización. En la figura 7.6 se muestran las señales I, II y III, tras serles aplicado el **filtro de suavizado**.

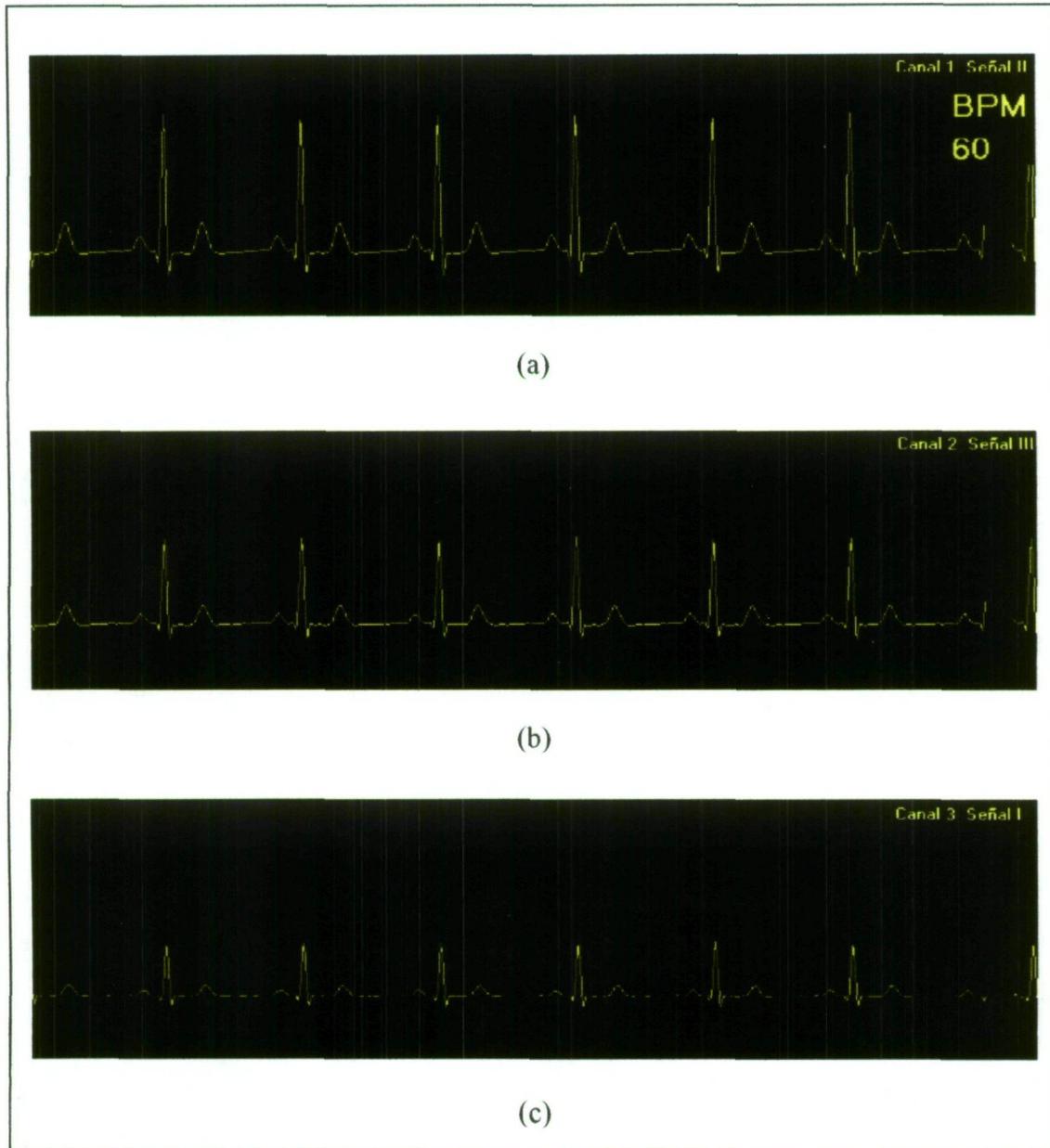


Figura 7.6 Señal II (a), III (b) y I (c) visualizadas y en el PC.

Se puede observar la diferencia de estas señales con respecto a las obtenidas con el osciloscopio digital a la salida de los canales de acondicionamiento (figura 7.5), sobre todo en los niveles de ruido. Las señales de la figura 7.6 mantienen en gran medida, a pesar del filtrado, su morfología original. En la figura 7.7 se muestran las mismas

señales correspondientes a 7.6(a) y 7.6(b), pero visualizadas con la ayuda de MATLAB a partir de los ficheros generados por la aplicación de visualización.

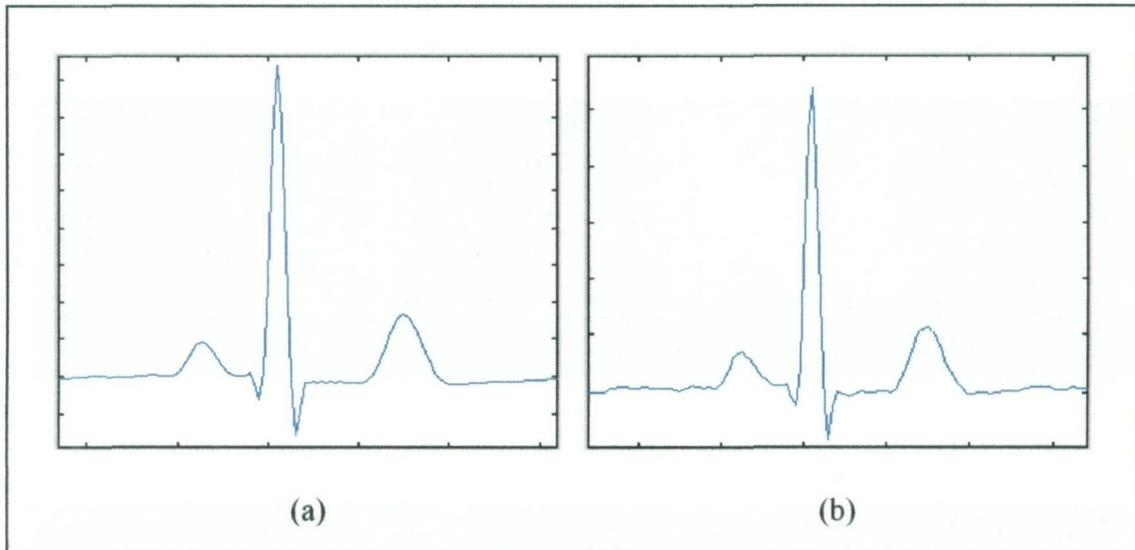


Figura 7.7 (a) Señal II y (b) señal III muestreadas a 100 muestras/sg y filtradas con el filtro de suavizado.

Una forma de evaluar la calidad de las señales de la figura 7.7 es fijarse no sólo en la ausencia de ruido, sino también en que se mantenga la relación entre las amplitudes de las ondas P, R y T, ya que el filtro de suavizado podría atenuar la amplitud de la onda R. Otro aspecto a evaluar es que no se produzca distorsión del segmento ST (tramo de señal comprendido entre la onda S y la T). La variación en la forma del segmento ST puede, en algunos casos, ser indicativo de problemas en el músculo cardíaco [3].

En cualquier caso en el ECG de monitorización, que es el objetivo de este proyecto, el aspecto fundamental es la extracción de la frecuencia cardíaca y la observación de posibles arritmias, así como la indicación de una posible parada cardíaca. En este sentido el **algoritmo de extracción de la frecuencia cardíaca** demostró su **correcto funcionamiento** para los distintos valores de este parámetro seleccionados en el paciente electrónico.

En la figura 7.8 se muestran algunas imágenes del sistema desarrollado en funcionamiento.

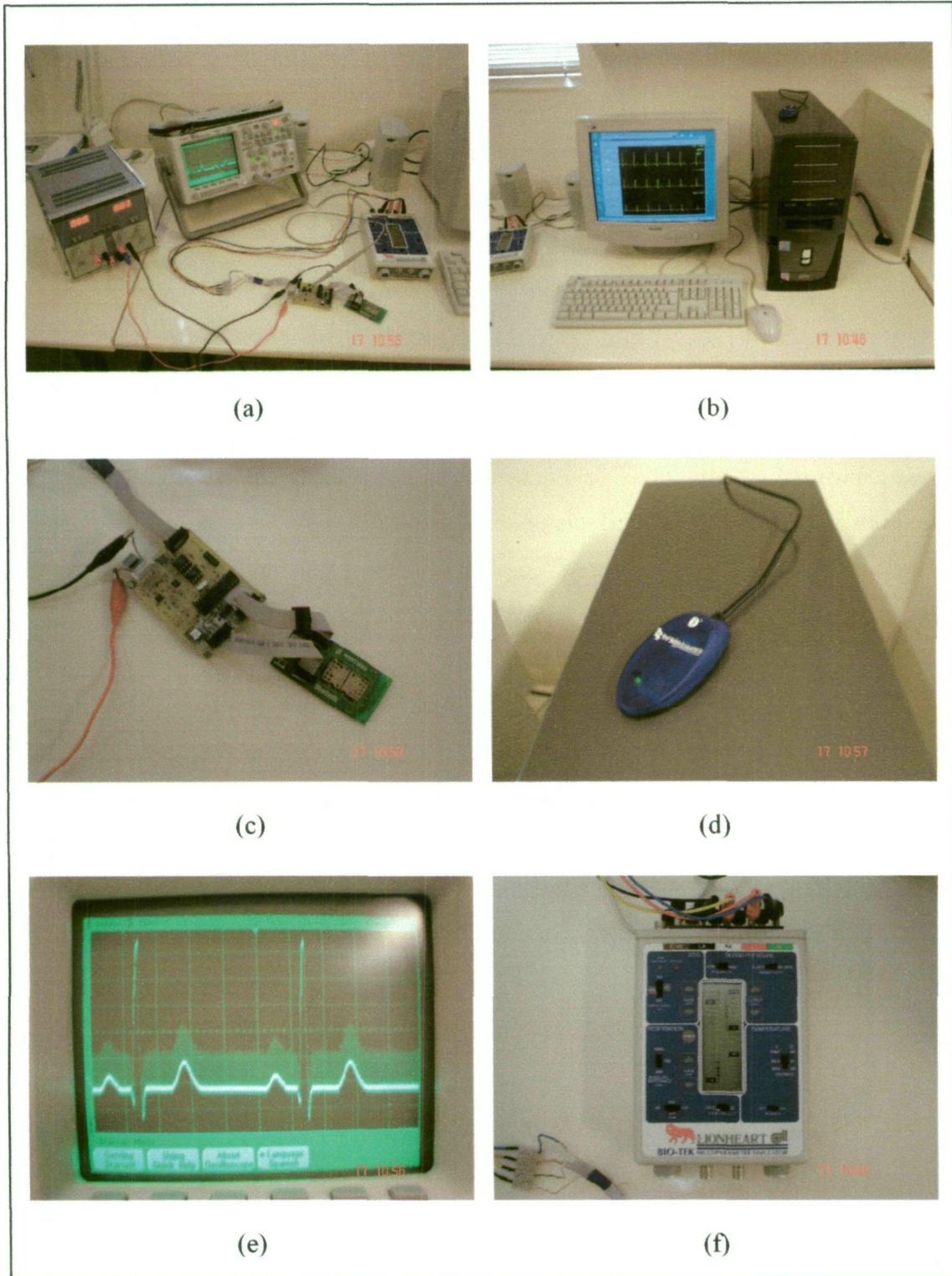


Figura 7.8 (a) Unidad portátil junto al paciente electrónico y la fuente de alimentación. (b) Unidad de visualización ECG. (c) Unidad portátil. (d) Adaptador BT-USB. (e) Señal ECG visualizada en el osciloscopio digital. (f) Paciente electrónico.

En la figura 7.8(a) se muestra la unidad portátil conectada al paciente electrónico. Junto a ella se encuentran la fuente de alimentación y el osciloscopio digital con el que se obtuvo la imagen de la figura 7.8(e). La figura 7.8(b) corresponde la unidad de visualización ECG, formada por el PC y el adaptador BT-USB (sobre la torre del ordenador). En la pantalla del ordenador se observa la aplicación de visualización en ejecución. En la figura 7.8(c) se muestra la unidad portátil y en la 7.8(d) se observa el adaptador BT-USB con más detalle. En la figura 7.8(f) se muestra el paciente electrónico utilizado.

7.4 Conclusiones

En este capítulo se han presentado los resultados obtenidos durante la realización de este proyecto. Entre ellos se encuentra el correcto establecimiento del enlace Bluetooth entre la unidad portátil y la unidad de visualización. Se comentan brevemente las dificultades presentadas durante el desarrollo del host Bluetooth de la unidad portátil y el procedimiento empleado para abordar estos problemas.

Por otro lado se ha hecho un breve análisis de la calidad de la señal ECG visualizada en el PC, comparándola con muestras de la misma señal en dos puntos concretos de la etapa de acondicionamiento de señal ECG de la unidad portátil. La calidad de la señal visualizada es lo suficientemente buena no sólo para aplicaciones de monitorización cardiaca, sino también en aplicaciones de estudio clínico de las señales visualizadas para detectar posibles anomalías.

CAPÍTULO 8

CONCLUSIONES

En este Proyecto Fin de Carrera se ha diseñado e implementado el prototipo de un sistema de monitorización ECG que permite la adquisición, procesado y visualización de las señales I, II y III de un electrocardiograma en un ordenador personal. Además, la aplicación software empleada en la presentación visual de las señales posibilita también la obtención del valor instantáneo de la frecuencia cardiaca, y permite generar una alarma sonora en el caso de que el paciente entre en parada cardiaca. Por otro lado, la posibilidad de almacenar en un fichero las muestras de la señal ECG permite su análisis y estudio posterior para poder detectar diversos tipos de anomalías cardiacas, como por ejemplo una posible arritmia.

En la realización de este proyecto se introducen dos elementos diferenciadores con respecto a los equipos de monitorización tradicionales. Por un lado, la utilización de la tecnología *Bluetooth* permite prescindir del cableado de interconexión entre sensores y equipo de monitorización. Además abre la posibilidad de interacción del sistema con redes de transmisión de datos ya existentes, lo que permite el poder realizar

la monitorización ECG sin necesidad de que exista una cercanía física entre el paciente y los equipos de visualización de las señales. Para ello el presente proyecto se basa en trabajos previamente realizados en esta universidad [2] y relacionados con la tecnología *Bluetooth*.

El otro elemento destacable en este proyecto es la novedad que supone el sustituir los costosos equipos de monitorización tradicionales por un simple PC. Como ya se ha mencionado, para ello se ha desarrollado un software específico para llevar a cabo el procesado y visualización de las señales ECG.

La mayor dificultad que se ha presentado durante la realización de este proyecto ha sido sin duda la implementación del *host* Bluetooth de la unidad portátil. Esto se debe a que el establecimiento de la conexión dependía en gran medida del comportamiento del adaptador BT-USB. Aunque la especificación del estándar Bluetooth establece claramente las normas generales de cómo debe ser este procedimiento, también es cierto que algunos aspectos concretos no se detallan con claridad. Además la implementación del *host* en un microcontrolador sin la ayuda de una placa de desarrollo que facilite la tarea de depuración del código ha sido complicada. Para afrontar este problema ha sido necesario emplear estrategias alternativas, como el desarrollo de una aplicación capaz de monitorizar el intercambio de paquetes entre *host* y módulo BT de la unidad portátil.

Aunque el *host* implementado dista mucho de estar completo, éste incorpora los recursos básicos necesarios para establecer una conexión Bluetooth haciendo uso del perfil *Serial Port*, y que puede tomarse como base para el desarrollo de otros perfiles Bluetooth destinados a otras aplicaciones.

En cuanto a futuras líneas de desarrollo, las prestaciones de este sistema pueden ser ampliadas, mediante mínimas modificaciones, de forma que en un solo PC se puedan monitorizar varios pacientes simultáneamente. Además se contempla la posibilidad de realizar un tratamiento de las señales para su posterior transmisión por algún tipo de red de transporte de datos, permitiendo su integración en aplicaciones en el ámbito de la telemedicina.

Otras posibles mejoras son el perfeccionamiento del *host* Bluetooth de la unidad portátil para hacer posible el uso de cualquier tipo de dispositivo adaptador BT-USB, o la ampliación del sistema para adaptarlo a aplicaciones de electrocardiograma clínico, que trabaja con 12 señales. También se podría estudiar la posible aplicación del sistema aquí presentado a la transmisión y monitorización de otro tipo de señales biomédicas, como son las de un encefalograma (EEG), estudio del sueño, etc. Estas aplicaciones podrían ser abordadas mediante algunos circuitos adicionales y la modificación de los programas de aplicación, manteniendo el núcleo de circuitos y software desarrollados en este proyecto.

BIBLIOGRAFÍA

[1] “BLUETOOTH Connect Without Cables”; Jennifer Bray, Charles F Sturman. Ed. Prentice Hall, 2001.

[2] “Diseño de un Sistema de Interconexión de Sensores Biomédicos mediante Enlaces basados en la Tecnología Bluetooth”, Proyecto Fin Carrera, Luis Alberto Rodríguez García, ULPGC 2002.

[3] “The biomedical engineering handbook”; Joseph Bronzino. IEEE press, 1995.

[4] “Microcontroladores ‘PIC’. La clave del diseño ”; Eugenio Martín Cuenca, José M^a. Angulo Usategui, Ignacio Angulo Martínez. Ed. Thomson, 2003.

[5] “Biomedical digital signal processing: C-language examples and laboratory experiments for the IBM PC”; Willis J. Tomkins. Prentice Hall, 1993.

[6] “Diseño de un sistema de transmisión de la señal ECG vía radio”, Proyecto Fin de Carrera, José David Valido Moreno, ULPGC 2003.

[7] “Specification of the BLUETOOTH SYSTEM”; Bluetooth 2.0 + EDR.
www.bluetooth.org , 2004, visitada en agosto de 2005.

Bibliografía

- [8] “ORDEN CTE/630/2002, del 14 de marzo, por la que se aprueba el Cuadro Nacional de Atribución de Frecuencias (CNAF)”; Boletín Oficial del Estado (BOE) número 70, 5727. Viernes 22 de marzo de 2002.
- [9] “Microsoft Developer Network”. <http://msdn.microsoft.com/> , visitada en abril de 2005.
- [10] “Adaptive filter theory”. Simon Haykin. Prentice-Hall, 1986.
- [11] “A real time QRS detection algorithm”. Jiapu Pan, Willis J. Tompkins. *IEEE Trans. Biomed. Eng.* BME -32:230-36, 1985.
- [12] “Programación con C++ Builder 5”. Francisco Charte Ojeda. Ed. Anaya Multimedia, 2000.

PRESUPUESTO



PRESUPUESTO

Para realizar el cálculo del presupuesto se considerará en primer lugar el baremo establecido por el “*Colegio Oficial de Ingenieros de Telecomunicación*” (COIT) para el 2005. Posteriormente se aplicarán impuestos.

1.1 Presupuesto previo

Según lo establecido por el COIT en lo referente a “*Baremos de Honorarios Orientativos para Trabajos Profesionales en 2005*”, los conceptos que se han de tener en cuenta a la hora de elaborar un presupuesto son los siguientes:

- Trabajo tarifado
- Coste de los materiales y amortizaciones
- Redacción del proyecto
- Derechos de visado del COIT
- Gastos de tramitación y envíos

1.1.1 Trabajo tarifado por tiempo empleado

El importe de las horas de trabajo empleadas para la realización del proyecto se calcula según las recomendaciones del COIT con la siguiente fórmula:

$$H = C_t \cdot 72 \cdot H_n + C_t \cdot 93 \cdot H_e \quad (\text{euros})$$

Donde:

- H son los honorarios totales por el tiempo dedicado
- H_n son las horas normales trabajadas (dentro de la Jornada Laboral)
- H_e son las horas especiales
- C_t es un factor de corrección en función del número de horas trabajadas

Para la realización de este proyecto se han empleado 1440 horas (8 horas/día · 5 días/semana · 4 semanas/mes · 9 meses), todas ellas dentro del horario normal. En este cómputo se incluye la fase de documentación previa.

Según el COIT, el coeficiente C_t tiene un valor de 0,4 para trabajos de más de 1080 horas. Por tanto se obtiene que:

$$H = 0,4 \cdot 72 \cdot 1440 + 0,4 \cdot 93 \cdot 0 = 41.472,00 \quad (\text{euros})$$

Los honorarios totales por tiempo empleado asciende a **cuarenta y un mil cuatrocientos setenta y dos euros (41.472,00 €)**.

1.1.2 Coste de los materiales y amortizaciones

Este apartado se divide en los siguientes grupos:

- Coste de los materiales empleados
- Amortización del inmovilizado material
- Amortización del inmovilizado inmaterial

La vida útil o periodo de amortización para la instrumentación y equipos del laboratorio (inmovilizado material) se establece en un de **5 años**, excepto para los ordenadores que se amortizan en **3 años**. Para las aplicaciones y herramientas informáticas utilizadas (inmovilizado inmaterial) se establece también un periodo de amortización de **3 años**. Para ello se utilizará un sistema de amortización lineal o constante, en el que se supone que el inmovilizado material e inmaterial se deprecian de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula utilizando la siguiente fórmula:

$$\text{Cuota anual de amortización} = \frac{\text{Valor de adquisición} - \text{Valor residual}}{\text{número de años de vida útil}}$$

El *valor residual* es el valor teórico que se supone que tendrá el elemento tras la finalización del periodo de vida útil.

1.1.2.1 Coste de los materiales empleados

Para la realización de este proyecto han sido necesarios los siguientes elementos y componentes electrónicos:

Componentes	Cantidad	Precio (€)	Importe (€)
Kit de desarrollo Bluetooth Ericsson	1	500	500
Adaptador Bluetooth-USB BrainBoxes BL-554	1	120	120
PIC16F876-20/SP	1	11,51	11,51
MAX-232ECSE	1	4,07	4,07
L-78M05CDT	1	0,60	0,60
AD623AN	2	4,95	9,90
AD8542AR	2	11,22	22,44
LM-4140ACM-1.2	1	9,93	9,93
Placa de circuito impreso	1	5,35	5,35
Componentes pasivos y otros elementos	Varios	--	27,20
Importe total			711,00 €

Tabla P.1 Precios de materiales utilizados.

Por lo tanto el coste total del material empleado en el proyecto asciende a la cantidad de **setecientos once euros (711,00 €)**.

1.1.2.2 Amortización del inmovilizado material

Dentro del epígrafe de inmovilizado material se incluyen los ordenadores y la instrumentación de laboratorio. Teniendo en cuenta que la duración del proyecto ha sido inferior a un año y que el periodo de amortización se establece en 3 años para los ordenadores y en 5 años para el resto de los equipos de laboratorio, los costes de amortización se calcularán para el primer año. En la siguiente tabla se presentan los costes de amortización del inmovilizado material:

Concepto	Coste total de adquisición (€)	Valor residual (€)	Coste de amortización (€)
PC Pentium IV 1500 MHz	1200	300	300
Osciloscopio digital Tektronix	24040,48	6010	3606
Fuente alimentación	330,5	82	49,70
Simulador señales biomédicas	900	200	140
Polímetro digital	70	20	10
Importe total			4105,70 €

Tabla P.2 Amortización del inmovilizado material.

El coste total de las amortizaciones del inmovilizado material asciende a **cuatro mil ciento cinco euros con setenta céntimos (4.105,70 €)**.

1.1.2.3 Amortización del inmovilizado inmaterial

El inmovilizado inmaterial incluye las aplicaciones y herramientas informáticas utilizados durante la realización del proyecto. El periodo de amortización estipulado para estos elementos es de 3 años. Al igual que el inmovilizado material, los costes de amortización del inmovilizado inmaterial se calcularán para el primer año. En la siguiente tabla se presentan dichos costes:

Concepto	Coste total de adquisición (€)	Valor residual (€)	Coste de amortización (€)
MS Windows 98 SE	150	60	30
MS Office XP	339	110	76,33
C++ Builder 5	2500	300	733,3
Matlab 5.3	3000	890	630
Protel 99 SE	1900	940	320
Orcad Pspice 9.1	2000	900	366,67
MPLAB IDE 6.2	0	0	0
IC-PROG	0	0	0
Importe total			2156,30 €

Tabla P.3 Amortización del inmovilizado inmaterial.

El coste total de las amortizaciones del inmovilizado material asciende a **dos mil ciento cincuenta y seis euros con treinta céntimos (2.156,30 €)**.

1.1.3 Redacción del proyecto

El importe de la redacción del proyecto se calcula haciendo uso de la siguiente fórmula:

$$R = 0,07 \cdot P \cdot C_n \quad (\text{euros})$$

Donde:

- P es el valor acumulado del presupuesto del proyecto
- C_n es el coeficiente de ponderación en función del presupuesto

El presupuesto acumulado hasta este punto (P) asciende a **cuarenta y ocho mil cuatrocientos cuarenta y cinco euros (48.445,00 €)**. Como el coeficiente de ponderación (C_n) para presupuestos de esta cuantía viene definido por el COIT en $0,45$, el coste derivado de la redacción del proyecto es de **mil quinientos veintiséis euros con dos céntimos (1.526,02 €)**.

1.1.4 Derechos de visado del COIT

Los gastos de visado del COIT se tarifican mediante la siguiente expresión:

$$V = 0,006 \cdot P \cdot C_v \quad (\text{euros})$$

Donde:

- P es el valor acumulado del presupuesto del proyecto
- C_v es el coeficiente de ponderación en función del presupuesto

El presupuesto acumulado hasta este punto (P) asciende a **cuarenta y nueve mil novecientos setenta y un euros con dos céntimos (49.971,02 €)**. Como el coeficiente de ponderación (C_v) para presupuestos de esta cuantía viene definido por el COIT en $0,9$, el coste derivado de los derechos de visado del proyecto es de **doscientos sesenta y nueve euros con ochenta y cuatro céntimos (269,84 €)**.

1.1.5 Gastos de tramitación y envíos

Los gastos de tramitación y envíos son fijos y se estipulan en **seis euros con un céntimo (6,01 €)**.

1.2 Aplicación de impuestos

El coste total de del proyecto antes de aplicarle los correspondientes impuestos asciende a **cincuenta mil doscientos cuarenta y seis euros con ochenta y siete céntimos (50.256,87 €)**, a lo que hay que sumar el 5% de IGIC. Por lo tanto, el coste definitivo del proyecto es:

Presupuesto total antes de impuestos	50.246,87 €
IGIC (5 %)	2512,34 €
Total	52.759,21 €

Tabla P.4 Presupuesto aplicando impuestos.

1.3 Presupuesto final

El presupuesto final pormenorizado es el siguiente:

Concepto	Importe (€)
Trabajo tarifado por tiempo	41.472,00
Costes del material	711,00
Amortización del inmovilizado material	4.105,70
Amortización del inmovilizado inmaterial	2.156,30
Redacción del proyecto	1.526,02
Derechos de visado del COIT	269,84
Gastos de tramitación y envíos	6,01
Impuestos (IGIC 5 %)	2.512,34
Total	52.759,21

Tabla P.5 Resumen del presupuesto.

El presupuesto total, impuestos incluidos, asciende a la cantidad de: **CINCUENTA Y DOS MIL SETECIENTOS CINCUENTA Y NUEVE EUROS CON VEINTIÚN CÉNTIMOS (52.759,21 €).**

Las Palmas de Gran Canaria a 4 de diciembre de 2005

Germán Quintana Rodríguez

PLIEGO DE CONDICIONES



PLIEGO DE CONDICIONES

En este apartado se recogen las condiciones y requisitos técnicos de funcionamiento del sistema desarrollado. Se especifican además los pasos a seguir para la correcta instalación del software de visualización en un ordenador personal y el procedimiento de puesta en funcionamiento del sistema.

1. Requisitos técnicos de funcionamiento

1.a) Unidad portátil

- **Alimentación:** pilas o baterías recargables de entre 7 y 24 voltios, capaces de suministrar hasta 70 mA de corriente. La conexión de la alimentación se especifica en los esquemáticos incluidos en los anexos.
- **Electrodos:** 4 electrodos ECG, cables y gel conductor. La conexión de los electrodos a la unidad portátil debe efectuarse de la forma que se especifica en los esquemáticos incluidos en los anexos.

1.b) Unidad de visualización

- **Requisitos HW:** el PC en el que se alojará la aplicación software debe tener al menos las siguientes características:
 - Pentium IV 1500 MHz
 - 128 MB de RAM
 - Un puerto USB libre
 - Unidad de CD ROM
 - El monitor configurado para 1024 x 768 píxeles

- **Requisitos SW:** sistema operativo Microsoft Windows 98 SE.

1.c) Dispositivo de comunicaciones BT

- El alcance de la conexión inalámbrica vendrá dado por las características de los dispositivos Bluetooth utilizados. Como mínimo el sistema debe contar con dispositivos de clase 3 que permiten alcances nominales de hasta 10 metros. Para el sistema propuesto en este proyecto se utilizará un adaptador BT-USB de Brain Boxes, model BL-554.

2. Instalación del software en el ordenador personal

2.a) Instalación del software del control del adaptador BT-USB de Brain Boxes

- Introducir el CD de instalación del software de Brain Boxes en unidad de CD ROM.
- Conectar el adaptador Bluetooth-USB a un puerto USB libre.
- Cuando aparezca la ventana de ‘Nuevo hardware encontrado’, seleccionar el CD de Brain Boxes (‘Buscar driver más apropiado...’ y luego ‘Especificar ubicación’) y seguir las instrucciones de instalación del software.
- Este programa instalará en el escritorio el programa de control del adaptador BT-USB denominado ‘My Bluetooth Devices’.

2.b) Instalación de la aplicación de visualización 'Monitor ECG'

- Copiar del CD adjunto a la memoria el contenido de la subcarpeta 'Ejecutable' dentro de la carpeta 'Aplicación de visualización' a una carpeta vacía en el disco duro del PC.
- Crear un enlace directo del archivo 'Monitor_ECG.exe' en el escritorio.

3. Procedimiento de puesta en funcionamiento del sistema

Para poner en funcionamiento el sistema de monitorización ECG desarrollado se deben seguir los siguientes pasos:

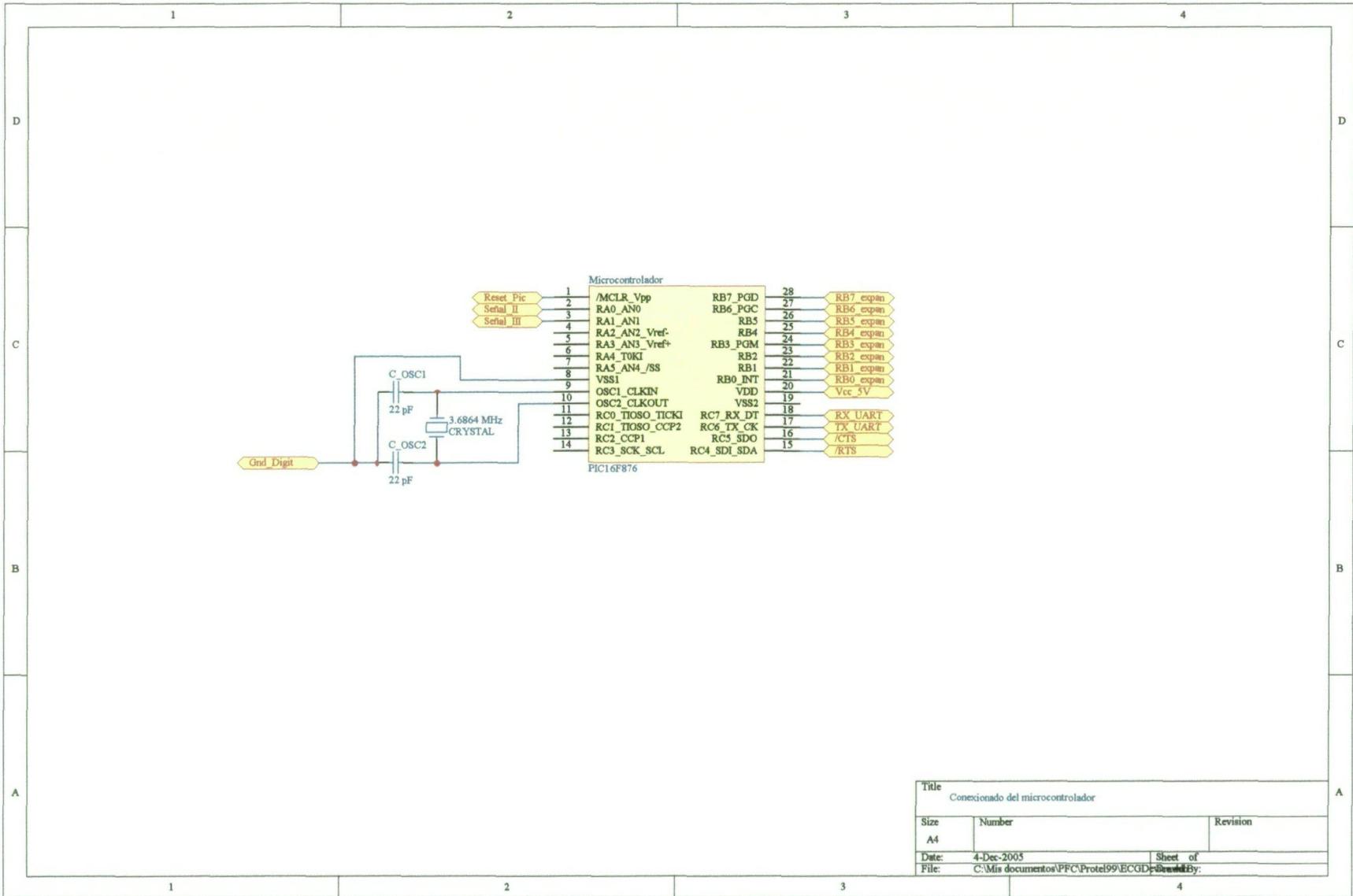
- Ejecutar el programa de control del adaptador BT-USB 'My Bluetooth Devices' que se ha creado en el escritorio.
- En el menú principal de 'My Bluetooth Devices' se debe seleccionar *My Bluetooth Devices, My Properties*. En la ventana de diálogo asociada se ha de seleccionar las opción de seguridad *Low(Mode 1)* y deshabilitar la opción de encriptación de datos. También hay que habilitar las características *Discoverable, Connectable* y *Auto-accept Connections*.
- Ejecutar la aplicación 'Monitor ECG'.
- Conectar la alimentación de la unidad portátil y pulsar el botón de reset de la misma.
- Esperar unos segundos a que se establezca el enlace Bluetooth. Cuando esto se produzca, la aplicación 'My Bluetooth Devices' indicará el puerto serie virtual creado para la conexión, junto a la dirección Bluetooth del módulo BT de la unidad portátil.
- En la interfaz gráfica de la aplicación 'Monitor ECG' seleccionar el puerto serie creado.
- Pulsar el botón *Visualizar ECG*. En el capítulo 6 se explica el funcionamiento del resto de los controles de la aplicación.
- Si se desea cambiar la configuración de los filtros o la visualización de la señal ECG se debe primeramente pausar la visualización con el botón *Pausar Visualización*. Esto también se aplica a captura de una imagen en formato *bmp* o

de las muestras de señal en un fichero de texto. Una vez modificada la configuración de la aplicación se debe pulsar de nuevo el botón *Visualizar ECG*.

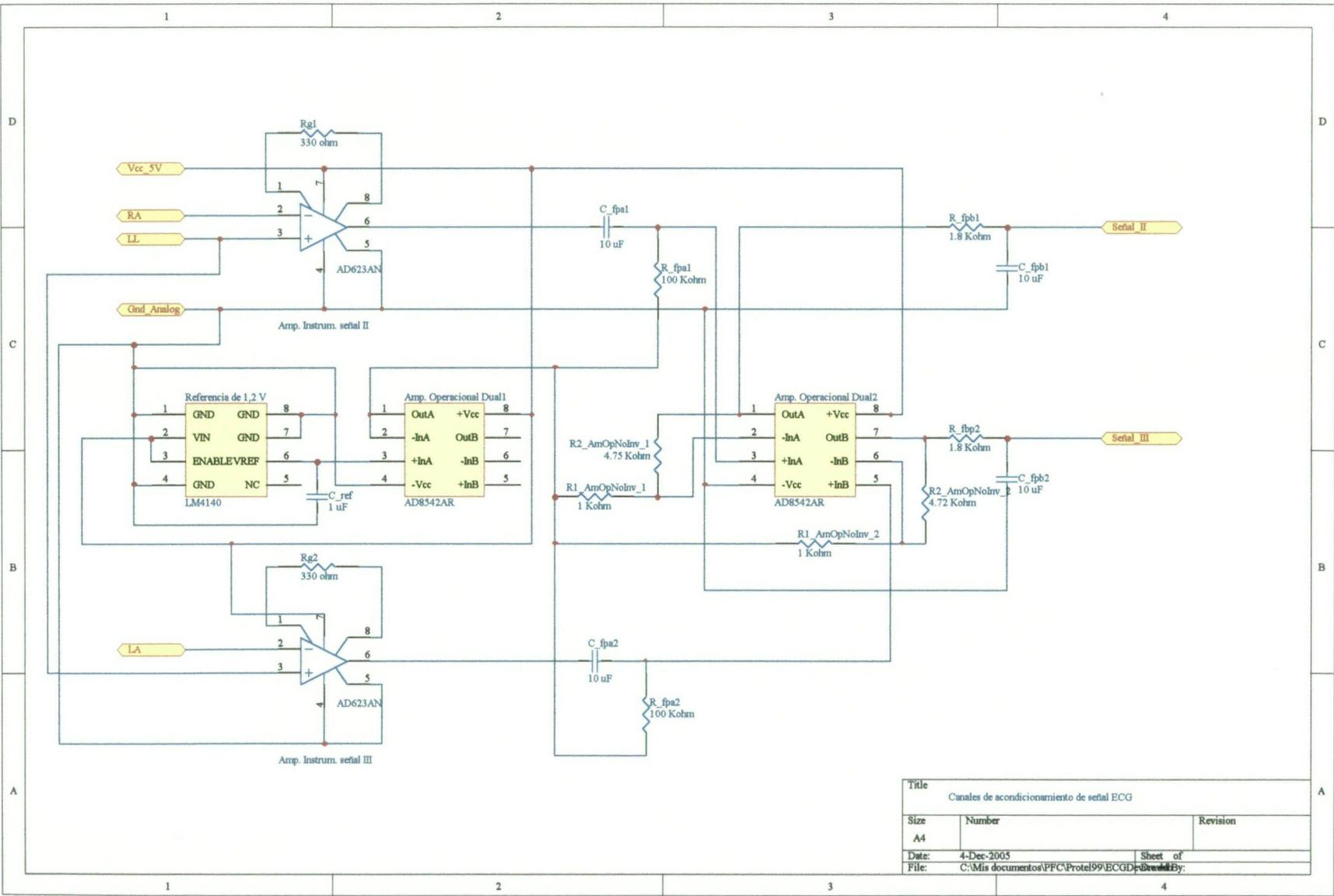
PLANOS



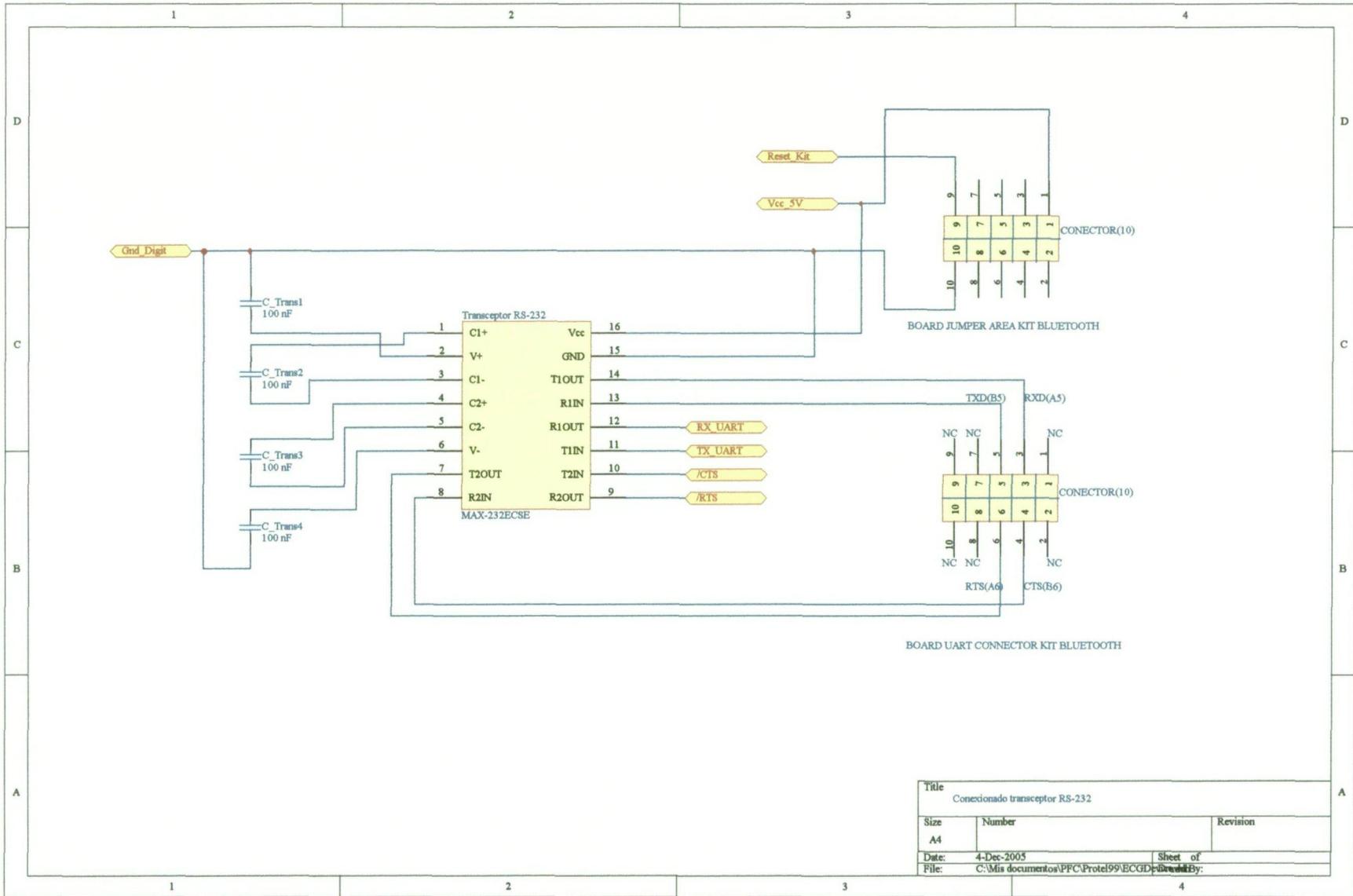
ESQUEMÁTICOS Y LAYOUTS DE LA UNIDAD PORTÁTIL



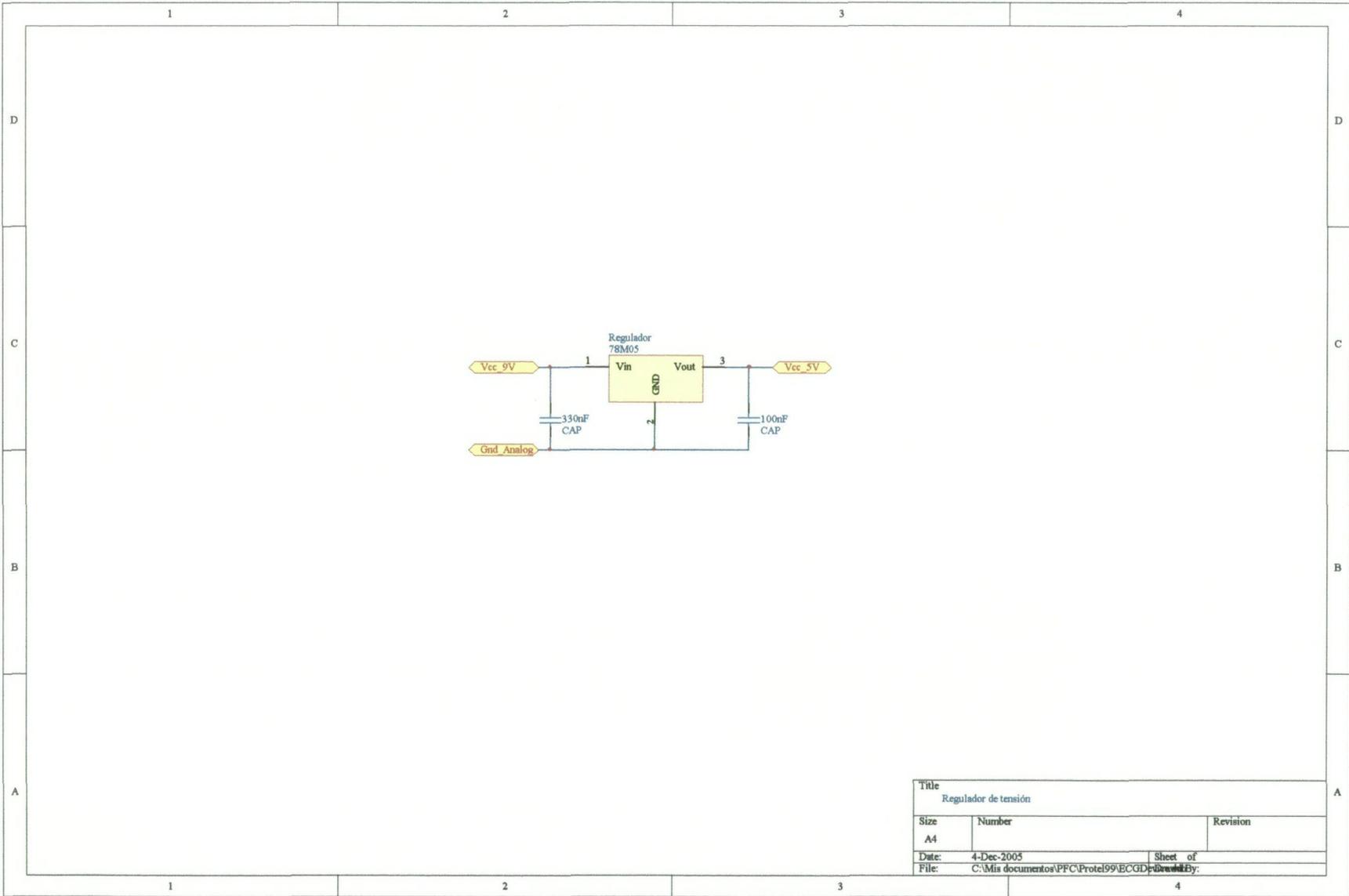
Title		
Conexión del microcontrolador		
Size	Number	Revision
A4		
Date:	4-Dec-2005	Sheet of
File:	C:\Mis documentos\FFC\Protel99\ECGD	Drawn by:



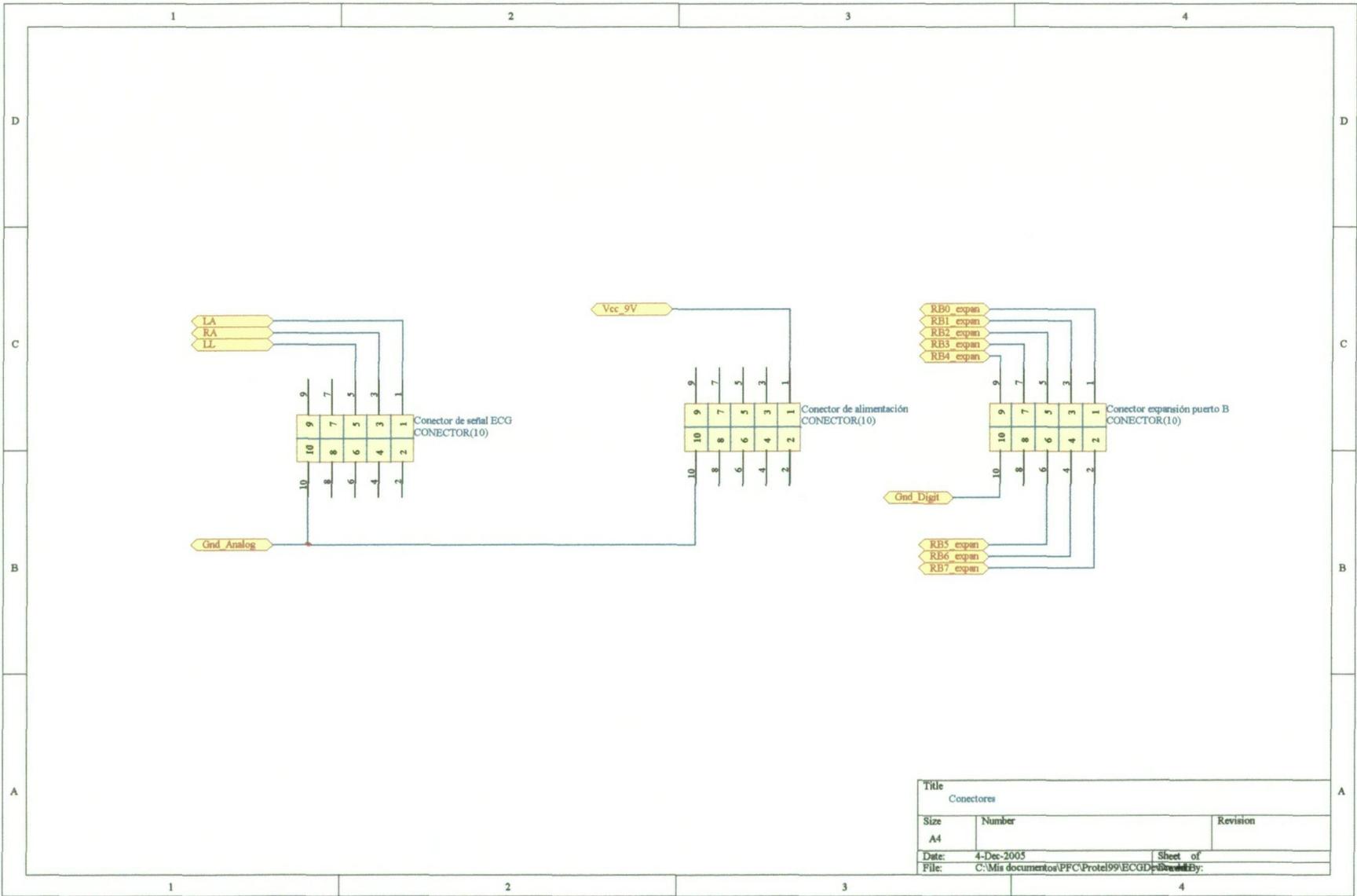
Title		
Canales de acondicionamiento de señal ECG		
Size	Number	Revision
A4		
Date:	4-Dec-2005	Sheet of
File:	C:\Mis documentos\PFC\Protel99\ECGD	Drawn By:



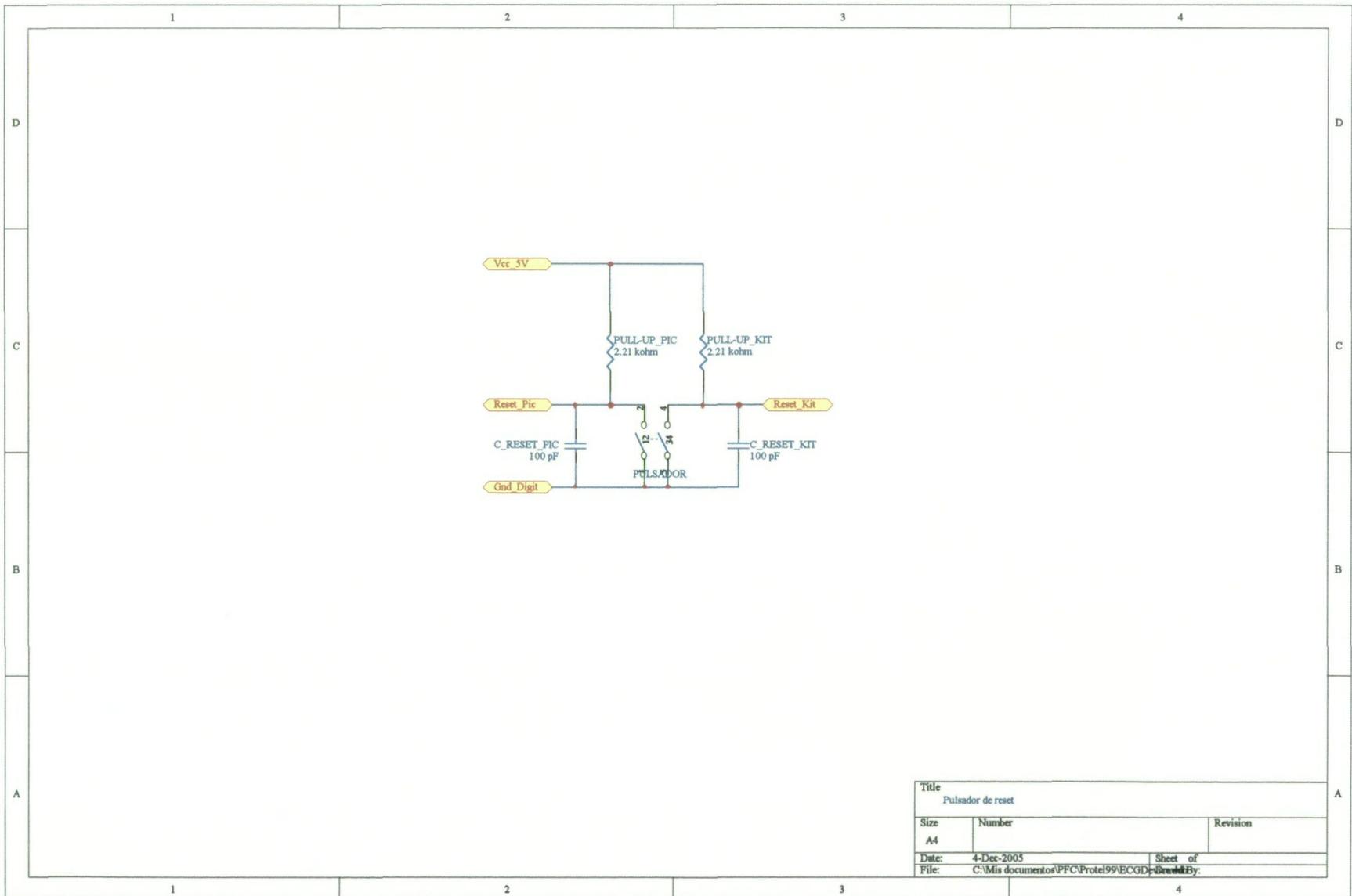
Title		
Conexión de transceptor RS-232		
Size	Number	Revision
A4		
Date:	4-Dec-2005	Sheet of
File:	C:\Mis documentos\PFC.Protele99\ECGD	Drawn By:



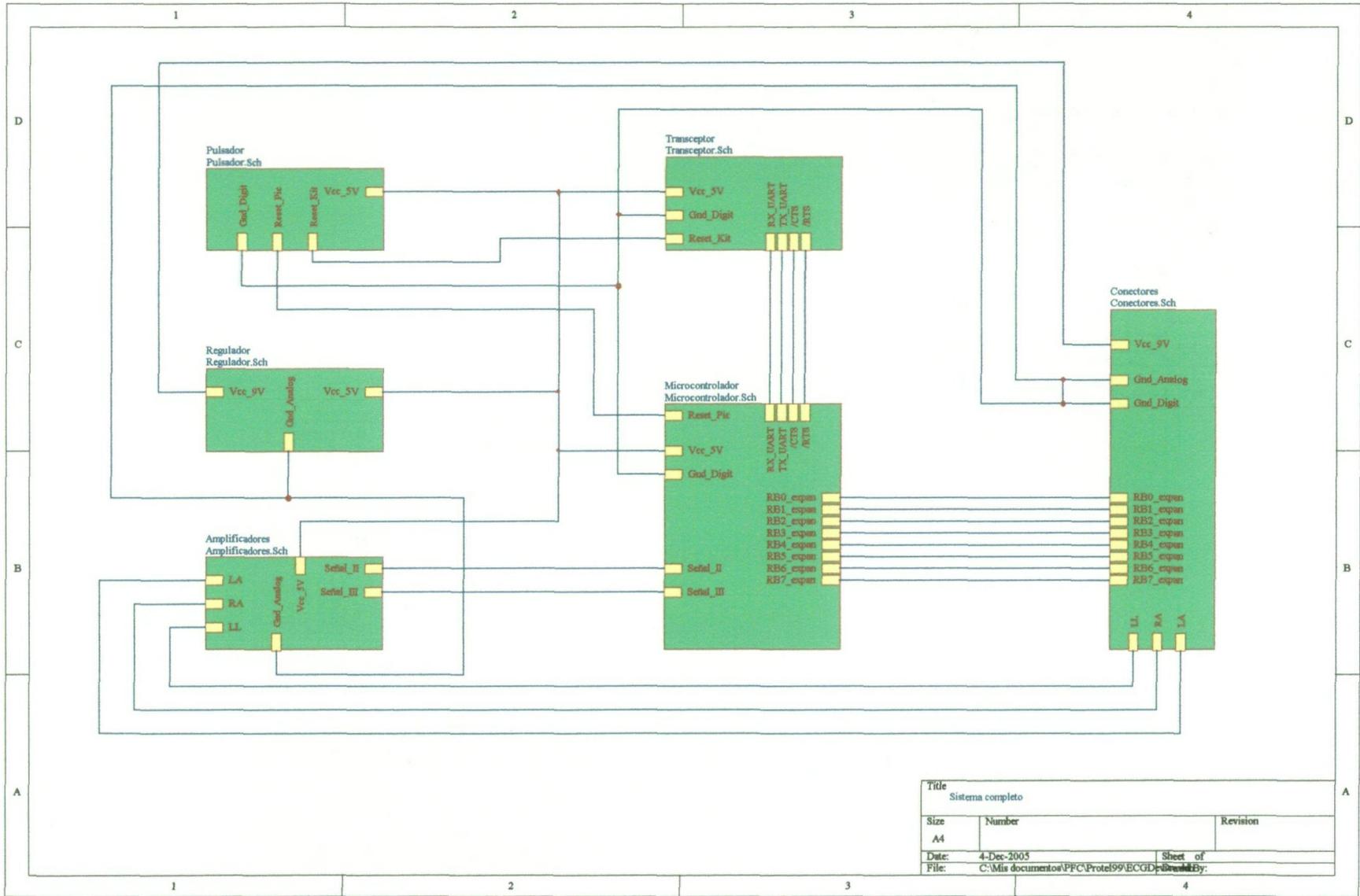
Title		
Regulador de tensión		
Size	Number	Revision
A4		
Date:	4-Dec-2005	Sheet of
File:	C:\Mis documentos\PFC\Protel99\ECGD	Drawn By:



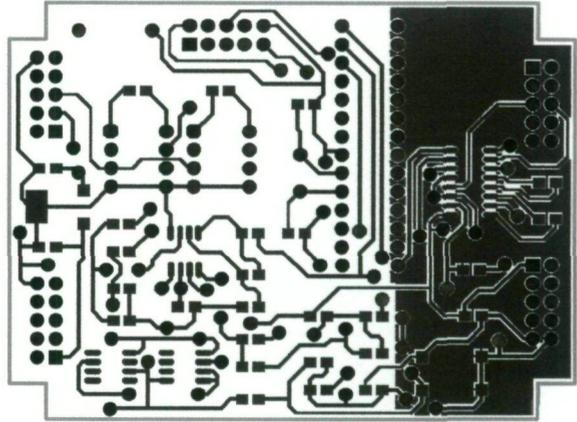
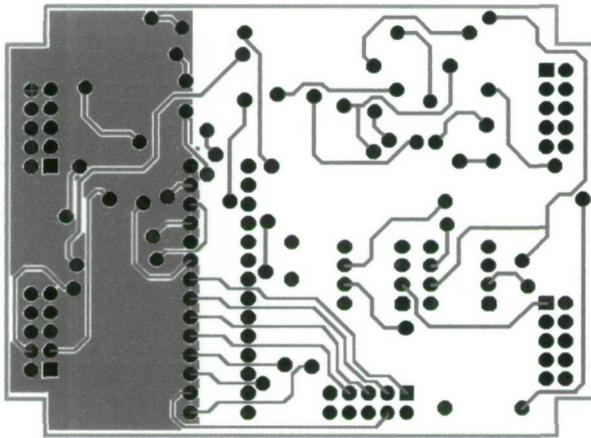
Title Conectores		
Size A4	Number	Revision
Date: 4-Dec-2005	Sheet of	
File: C:\Mis documentos\PPC\Protel99\ECGD	Drawn by:	



Title		
Pulsador de reset		
Size	Number	Revision
A4		
Date:	4-Dec-2005	Sheet of
File:	C:\Mis documentos\PFC\Protel99\ECGD	Drawn By:



Title		
Sistema completo		
Size	Number	Revision
A4		
Date:	4-Dec-2005	Sheet of
File:	C:\Mis documentos\VFC\Protel99\ECGD	Drawn By:



ANEXOS



ANEXO 1

INTERCAMBIO DE PAQUETES BLUETOOTH

ANEXO 1: Intercambio de paquetes Bluetooth

En este anexo se describe el intercambio de paquetes (en formato hexadecimal) entre Host Bluetooth y Módulo Bluetooth de la unidad portátil. En la columna de la izquierda (HOST BLUETOOTH) se detallan los paquetes enviados desde el Host hacia el Módulo, mientras que en la columna de la derecha (MÓDULO BLUETOOTH) se sitúan los paquetes que se envían desde el Módulo hacia el Host.

Los paquetes que envía el Host pueden ser tanto COMANDOS HCI para la correcta configuración y control del Módulo Bluetooth, como paquetes de capas superiores (L2CAP, RFCOMM, etc.) que se transmiten al Dispositivo Bluetooth Remoto. Análogamente los paquetes que envía el Módulo al Host, pueden ser tanto EVENTOS HCI provenientes del Módulo Bluetooth local, como paquetes de capas superiores enviados por el Dispositivo Bluetooth Remoto.

En cada paquete se nombran algunos de los campos o parámetros que se consideran más importantes. En cualquier caso aquellos que no se mencionan pueden ser consultados en el documento de especificación del estándar Bluetooth.

NOTA 1 : La capa de transporte utilizada para interconectar HOST y MÓDULO es una UART. Por tanto al comienzo de cada paquete HCI (todos los que describen en este documento) se añade un campo que indica el tipo de datos transmitidos. Los valores que toma este campo son los siguientes:

- Comando HCI : 0x01
- Datos ACL : 0x02
- Datos SCO : 0x03
- Evento HCI : 0x04

NOTA 2 : El sentido del flujo de datos entre HOST y MÓDULO se indica con los siguientes signos:

- datos enviados desde el HOST al MÓDULO
- ← datos enviados desde el MÓDULO al HOST

HOST BLUETOOTH

MÓDULO BLUETOOTH

RESET DEL MÓDULO BLUETOOTH

01 COMANDO HCI →
03 Opcode Cmd. RESET
0C
00

← 04 EVENTO HCI
0E Cmd. Complete Event
04 Long.
01
03 Opcode Cmd. causante
0C del evento (RESET)
00

DESCUBRIMIENTO DE DISPOSITIVOS BLUETOOTH CERCANOS (INQUIRY)

* El módulo Bluetooth realiza una búsqueda de dispositivos Bluetooth cercanos

01 COMANDO HCI →
01 Opcode Cmd. HCI_Inquiry
04
05 Long.
33 LAP(para derivar el inquiry access code)
8B LAP
9E LAP
20 N; Máx. Inquiry time=(N*1.28 sg)
01 Máx. núm. de respuestas

← 04 EVENTO HCI
 0F Cmd. Status Event
 04 Long.
 00
 01
 01 Opcode Cmd.
 04 HCI_Inquiry

← 04 EVENTO HCI
 02 Inquiry Result Event
 0F Long.
 01 n° de respuestas
 FD BD_ADDR
 07 "
 00 "
 4F "
 0A "
 00 "
 01 Page Scan Repetition Mode
 00 Page Scan Period Mode
 00 Page Scan Mode
 04 Class of Device
 01 "
 92 "
 37 Clock Offset
 26 "

← 04 EVENTO HCI
 01 Inquiry Complete Event
 01 Long.
 00 Status
 No se transmite Num_Responses.
 Tal vez pq el máx es 1.

CONEXIÓN CON DISPOSITIVO BLUETOOTH DESCUBIERTO (CONNECTION)

01 COMANDO HCI →
05 Opcode Cmd. HCI_Create_Connection
04
0D Long.
FD BD_ADDR
07 "
00 "
4F "
0A "
00 "
08 Packet_Type
00 "
01 Page_Scan_Repetition_Mode
00 Page_Sacn_Mode
00 Clock_Offset
00 "
00 Allow_Role_Switch

← 04 EVENTO HCI
0F Cmd. Status Event
04 Long.
00 Status
01 Num_HCI_Command_Packets
05 Cmd. Opcode (causante del
04 Command Status Event)

← 04 EVENTO HCI
03 Connection Complete Event
0B Long.
00 Status
01 Connection_Handle
00 "
FD BD_ADDR
07 "
00 "
4F "
0A "

00 "
 01 Link_Type
 00 Encryption_Mode

← 04 EVENTO HCI
 1B Max Slots Change
 03 Long.
 01 Connection Handle
 00 "
 05 LMP_Max_Slots

ESTABLECIMIENTO DE CONEXIÓN A NIVEL DE L2CAP PARA INTERCAMBIO DE PAQUETES SDP

* Establecimiento de un canal lógico a nivel de L2CAP para ser utilizado por SDP

L2CAP Connection Request Packet →

02 DATOS ACL
 01 Connection_Handle, PB y BC
 20 "
 0C Long.
 00 "
 08 Long. L2CAP Signalling Packet
 00 "
 01 Connection ID (Signalling)
 00 "
 02 L2CAP Connection Request Packet Code
 01 Identifier
 04 Length
 00 "
 01 PSM (SDP)
 00 "
 40 Source CID
 00 "

← L2CAP Connection Response Packet

02 DATOS ACL
01
20
10
00
0C
00
01
00
03 L2CAP Connection Resp. Code
01
08
00
40 Destination CID
00
40 Source CID
00
00
00
00
00

L2CAP Configuration Request →

02 DATOS ACL
01
20
10
00
0C
00
01
00
04 L2CAP Configuration Request Code
02
08
00
40 Destination CID
00 "
00 Flags
00 " (C=0)
01 Configuration Options
02 "
30 " (MTU)
00 " (MTU)

← L2CAP Configure Response packet

02 DATOS ACL

01

20

12

00

0E

00

01

00

05 L2CAP Configure Response Code

02

0A

00

40

00

00

00

00

00

01

02

30

00

← L2CAP Configuration Request packet

02 DATOS ACL

01

20

0C

00

08

00

01

00

04 L2CAP Configuration Request Code

01

04

00

40

00

00

00

L2CAP Configure Response packet →

02 DATOS ACL
01
20
0E
00
0A
00
01
00
05 L2CAP Configure Response packet Code
01
06
00
40 Source CID
00 "
00
00
00
00

INTERCAMBIO DE PAQUETES SDP

* El propósito de este intercambio de paquetes es obtener el *Server Channel Number* utilizado por el servicio *Serial Port*, y necesario para establecer una conexión con dicho servicio.

SDP Service Search Request →

02 DATOS ACL (L2CAP Data Packet)
01
20
00
0D
00
40 Destination CID
00

02 SDP Service Search Request PDU ID
 00
 01
 00
 08
 35 Tipo de atributo: secuencia de datos. Longitud de atributo en siguiente byte.
 03 Longitud de atributo = 3 bytes
 19 Tipo de atributo: UUID. Longitud = 2 bytes.
 11 UUID Serial Port Class (búsqueda del servicio 'Serial Port')
 01 "
 00 MaximunServiceRecordCount
 0A "
 00 ContinuationState

← SDP Service Search Response

02 DATOS ACL (L2CAP Data Packet)
 01
 20
 1A
 00
 16
 00
 40 Destination CID
 00
 03 SDP Service Search Response PDU ID
 00
 01
 00
 00
 03
 00
 03
 00
 01 ServiceRecordHandle 'Serial Port 1'
 00
 00
 00
 01 ServiceRecordHandle 'Serial Port 2'
 00
 01
 00
 01 ServiceRecordHandle 'Serial Port 3'
 00
 02
 00
 04

05
01
01
00
05
00

SDP Service Attribute Request →

02 **DATOS ACL (L2CAP Data Packet)**

01
20
15
00
00
40
00

04 **SDP Service Attribute Request PDU ID**

00
02
00
0C

00 **Service Record Handle**

01 "
00 "
00 "
00 "

80 **MaximiumAttributeByteCount**

35
03
09

00 **AttributeID = ProtocolDescriptorList**

04 "
00

← **SDP Service Attribute Response**

02 **DATOS ACL (L2CAP Data Packet)**

01
20
22
00
1E
00
40

```

00
05 SDP Service Attribute Response PDU ID
00
02
00
19
00
16
35
14
09
00 UUID Attribute ID: ProtocolDescriptorList
04 "
35
0F
35
06
19
01 UUID L2CAP Protocol
00 "
09
00
03
35
05
19
00 UUID RFCOMM Protocol
03 "
08
01 Server Channel Number
00
    
```

DESCONEXIÓN A NIVEL L2CAP DEL CANAL UTILIZADO POR SDP

* Una vez ya no es necesario el canal creado para intercambiar paquetes SDP éste se destruye.

L2CAP Disconnect Request packet →

```

02 DATOS ACL
01
20
    
```

Anexo 1

0C
00
08
00
01
00
06 L2CAP Disconnect Request packet code
02
04
00
40 Destination CID
00 "
40 Source CID
00 "

← L2CAP Disconnect Response packet

02
01
20
0C
00
08
00
01
00
07 L2CAP Disconnect Response code
02
04
00
40 Destination CID
00 "
40 Source CID
00 "

ESTABLECIMIENTO DE CONEXIÓN A NIVEL DE L2CAP PARA INTERCAMBIO DE PAQUETES RFCOMM

* Establecimiento de un canal lógico a nivel de L2CAP para ser utilizado por RFCOMM

L2CAP Connection Request Packet →

02 **DATOS ACL**
 01 **Connection_Handle, PB y BC**
 20 "
 0C **Long.**
 00 "
 08 **Long. L2CAP Signalling Packet**
 00 "
 01 **Connection ID (Signalling)**
 00 "
 02 **L2CAP Connection Request Packet Code**
 03 **Identifier**
 04 **Length**
 00 "
 03 **PSM (RFCOMM)**
 00 "
 41 **Source CID**
 00 "

← L2CAP Connection Response Packet

02 **DATOS ACL**
 01
 20
 10
 00
 0C
 00
 01
 00
 03 **L2CAP Connection Response Code**
 03
 08
 00
 40 **Destination CID**
 00
 41 **Source CID**

00
00
00
00
00

L2CAP Configuration Request →

02 DATOS ACL
01
20
10
00
0C
00
01
00
04 L2CAP Configuration Request Code
04
08
00
40 Destination CID
00 "
00 Flags
00 " (C=0)
01 Configuration Options
02 "
30 " (MTU)
00 " (MTU)

← L2CAP Configure Response packet

02 DATOS ACL
01
20
12
00
0E
00
01
00
05 L2CAP Configure Response Code
04
0A
00
41

00
00
00
00
00
01
02
30
00

← L2CAP Configuration Request packet

02 DATOS ACL

01
20
0C
00
08
00
01
00

04 L2CAP Configuration Request Code

02
04
00
41
00
00
00

L2CAP Configure Response packet →

02 DATOS ACL

01
20
0E
00
0A
00
01
00

05 L2CAP Configure Response Code

Anexo 1

02
06
00
40
00
00
00
00
00

ESTABLECIMIENTO DE UNA CONEXIÓN A NIVEL RFCOMM

* Una vez quede establecida la conexión RFCOMM es posible enviar datos haciendo uso del perfil Serial Port, o paquetes de capas superiores como OBEX, para la implementación de otro tipo de perfiles.

Petición de conexión con DLCI = 0 para 'signalling'. →

02 DATOS ACL
01
20
08
00
04
00
40
00
03 SABM Frame (DLCI=0)
3F
01
1C

← Confirmación de conexión con DLCI = 0

02
01
20
08
00
04

00
41
00
03
73 UA Frame (DLCI=0)
01
D7

Petición de conexión con DLCI = 1 para datos. →

02 DATOS ACL
01
20
08
00
04
00
40
00
0B SABM Frame (DLCI=000001)
3F
01
59

← Confirmación de conexión con DLCI = 1

02
01
20
08
00
04
00
41
00
0B
73 UA Frame (DLCI=1)
01
92

← MSC Command

02
01
20
0C
00
08
00
40
00
01 UIH Frame (MSC Command)
EF
09
E3
05
0B
0D
AA

MSC Response →

02
01
20
0C
00
08
00
40
00
01 UIH Frame (MSC Response)
EF
09
E1
05
0B
0D
AA

Transmisión de datos biomédicos →

02

01

20

00

0D

00

40

00

0B UIH Frame (Datos)

EF

13

XX Datos

XX "

9A

"

"

"

"

"

"

ANEXO 2

DATASHEETS

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

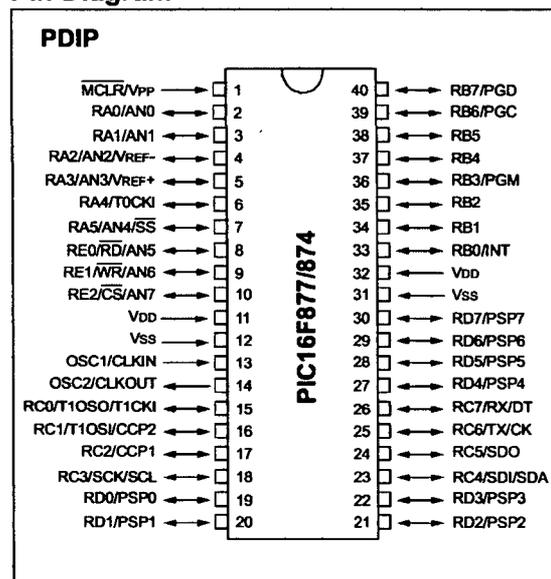
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)



Single Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier

AD623

FEATURES

- Easy to Use
- Higher Performance than Discrete Design
- Single and Dual Supply Operation
- Rail-to-Rail Output Swing
- Input Voltage Range Extends 150 mV Below Ground (Single Supply)
- Low Power, 575 μ A Max Supply Current
- Gain Set with One External Resistor
- Gain Range 1 (No Resistor) to 1,000

HIGH ACCURACY DC PERFORMANCE

- 0.1% Gain Accuracy ($G = 1$)
- 0.35% Gain Accuracy ($G > 1$)
- 25 ppm Gain Drift ($G = 1$)
- 200 μ V Max Input Offset Voltage (AD623A)
- 2 μ V/ $^{\circ}$ C Max Input Offset Drift (AD623A)
- 100 μ V Max Input Offset Voltage (AD623B)
- 1 μ V/ $^{\circ}$ C Max Input Offset Drift (AD623B)
- 25 nA Max Input Bias Current

NOISE

- 35 nV/ $\sqrt{\text{Hz}}$ RTI Noise @ 1 kHz ($G = 1$)

EXCELLENT AC SPECIFICATIONS

- 90 dB Min CMRR ($G = 10$); 84 dB Min CMRR ($G = 5$) (@ 60 Hz, 1K Source Imbalance)
- 800 kHz Bandwidth ($G = 1$)
- 20 μ s Settling Time to 0.01% ($G = 10$)

APPLICATIONS

- Low Power Medical Instrumentation
- Transducer Interface
- Thermocouple Amplifier
- Industrial Process Controls
- Difference Amplifier
- Low Power Data Acquisition

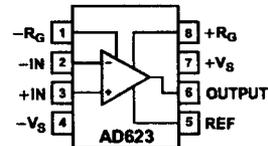
PRODUCT DESCRIPTION

The AD623 is an integrated single supply instrumentation amplifier that delivers rail-to-rail output swing on a single supply (+3 V to +12 V supplies). The AD623 offers superior user flexibility by allowing single gain set resistor programming, and conforming to the 8-lead industry standard pinout configuration. With no external resistor, the AD623 is configured for unity gain ($G = 1$) and with an external resistor, the AD623 can be programmed for gains up to 1,000.

The AD623 holds errors to a minimum by providing superior AC CMRR that increases with increasing gain. Line noise, as well as line harmonics, will be rejected since the CMRR remains constant up to 200 Hz. The AD623 has a wide input

CONNECTION DIAGRAM

8-Lead Plastic DIP (N),
SOIC (R) and μ SOIC (RM) Packages



common-mode range and can amplify signals that have a common-mode voltage 150 mV below ground. Although the design of the AD623 has been optimized to operate from a single supply, the AD623 still provides superior performance when operated from a dual voltage supply (± 2.5 V to ± 6.0 V).

Low power consumption (1.5 mW at 3 V), wide supply voltage range, and rail-to-rail output swing make the AD623 ideal for battery powered applications. The rail-to-rail output stage maximizes the dynamic range when operating from low supply voltages. The AD623 replaces discrete instrumentation amplifier designs and offers superior linearity, temperature stability and reliability in a minimum of space. Until the AD623, this level of instrumentation amplifier performance has not been achieved.

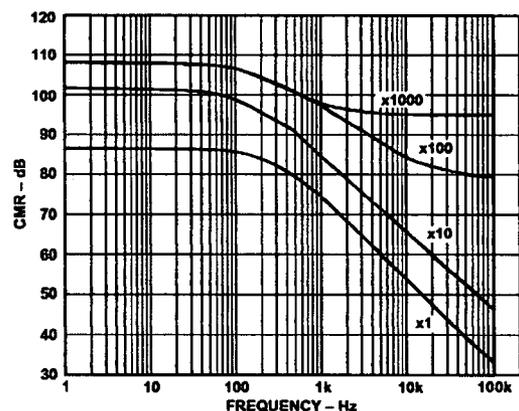


Figure 1. CMR vs. Frequency, +5 V_S , 0 V_S

REV. C

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>
Fax: 781/326-8703 © Analog Devices, Inc., 1999

AD8541/AD8542/AD8544

FEATURES

Single-Supply Operation: 2.7 V to 5.5 V
Low Supply Current: 45 μ A/Amplifier
Wide Bandwidth: 1 MHz
No Phase Reversal
Low Input Currents: 4 pA
Unity Gain Stable
Rail-to-Rail Input and Output

APPLICATIONS

ASIC Input or Output Amplifier
Sensor Interface
Piezo Electric Transducer Amplifier
Medical Instrumentation
Mobile Communication
Audio Output
Portable Systems

GENERAL DESCRIPTION

The AD8541/AD8542/AD8544 are single, dual, and quad rail-to-rail input and output single-supply amplifiers featuring very low supply current and 1 MHz bandwidth. All are guaranteed to operate from a 2.7 V single supply as well as a 5 V supply. These parts provide 1 MHz bandwidth at a low current consumption of 45 μ A per amplifier.

Very low input bias currents enable the AD8541/AD8542/AD8544 to be used for integrators, photodiode amplifiers, piezo electric sensors, and other applications with high source impedance. Supply current is only 45 μ A per amplifier, ideal for battery operation.

Rail-to-rail inputs and outputs are useful to designers buffering ASICs in single-supply systems. The AD8541/AD8542/AD8544 are optimized to maintain high gains at lower supply voltages, making them useful for active filters and gain stages.

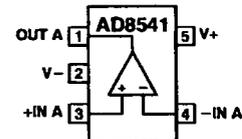
The AD8541/AD8542/AD8544 are specified over the extended industrial temperature range (-40°C to $+125^{\circ}\text{C}$). The AD8541 is available in 8-lead SOIC, 5-lead SC70, and 5-lead SOT-23 packages. The AD8542 is available in 8-lead SOIC, 8-lead MSOP, and 8-lead TSSOP surface-mount packages. The AD8544 is available in 14-lead narrow SOIC and 14-lead TSSOP surface-mount packages. All MSOP, SC70, and SOT versions are available in tape and reel only.

REV. D

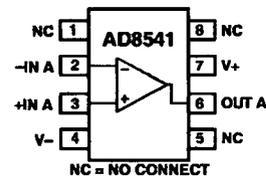
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

PIN CONFIGURATIONS

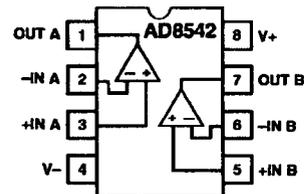
5-Lead SC70 and SOT-23
(KS and RT Suffixes)



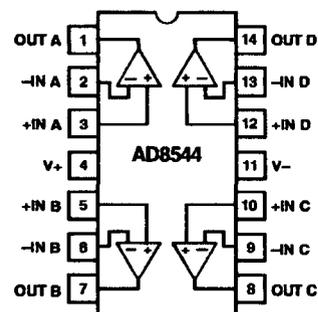
8-Lead SOIC
(R Suffix)



8-Lead SOIC, MSOP, and TSSOP
(R, RM, and RU Suffixes)



14-Lead SOIC and TSSOP
(R and RU Suffixes)



ROK 101 008 Bluetooth PnP Module

Key Features

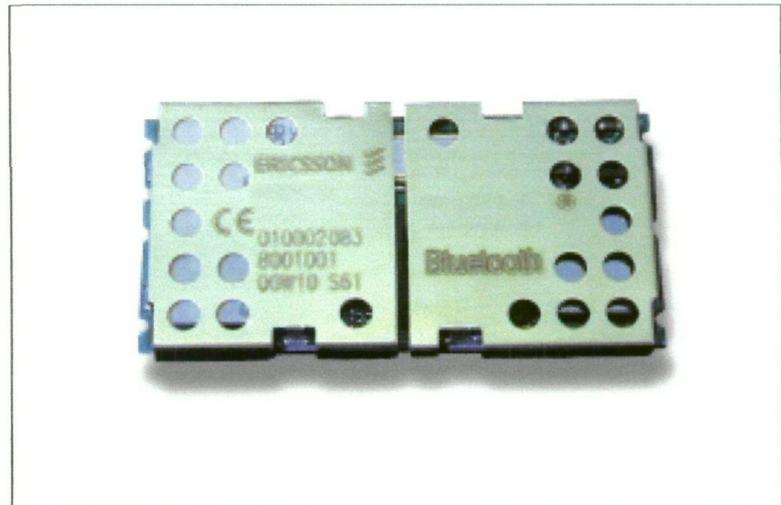
- Qualified to Bluetooth 1.1
- RF output power class 2
- FCC and ETSI approved
- 460 kb/s max data rate over UART
- UART and PCM interface
- I²C interface
- Internal crystal oscillator
- HCI firmware included
- Point to Point (PnP) connection
- Built-in shielding

Supported Bluetooth Profiles

- Generic Access Profile
- Service Discovery Application Profile
- Serial Port Profiles
 - Dial-up networking
 - Fax
 - Headset
- Generic Object Exchange Profiles
 - File transfer
 - Object Push
 - Synchronisation

Suggested Applications

- Computers and peripherals
- Handheld devices and accessories
- Access points



Description

ROK 101 008 is a short-range module for implementing Bluetooth functionality into various electronic devices. The module consists of three major parts; a baseband controller, a flash memory, and a radio that operates in the globally available 2.4–2.5 GHz free ISM band.

Both data and voice transmission is supported by the module. Communication between the module and the host controller is carried out via UART and PCM interface.

ROK 101 008, which is compliant with Bluetooth version 1.1, is a Class 2 Bluetooth Module (0 dBm) and is type-approved.

LM4140

High Precision Low Noise Low Dropout Voltage Reference

General Description

The LM4140 series of precision references are designed to combine high accuracy, low drift and noise with low power dissipation in a small package.

The LM4140 is the industry's first reference with output voltage options lower than the bandgap voltage.

The key to the advance performance of the LM4140 is the use of EEPROM registers and CMOS DACs for temperature coefficient curvature correction and trimming of the output voltage accuracy of the device during the final production testing.

The major advantage of this method is the much higher resolution available with DACs than is available economically with most methods utilized by other bandgap references.

The low input and dropout voltage, low supply current and output drive capability of the LM4140 makes this product an ideal choice for battery powered and portable applications.

The LM4140 is available in three grades (A, B, C) with 0.1% initial accuracy and 3, 6 and 10 ppm/°C temperature coefficients. For even lower Tempco, contact National Semiconductor.

The device performance is specified over the temperature range (0°C to +70°C) and is available in compact 8-pin SO package.

For other output voltage options from 0.5V to 4.5V, contact National Semiconductor.

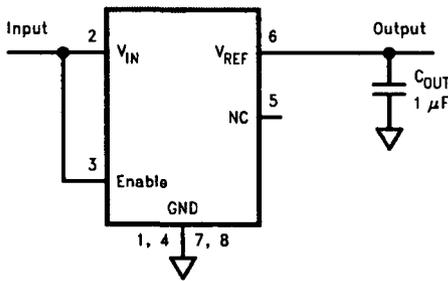
Features

- High initial accuracy: 0.1%
- Ultra low noise
- Low Temperature Coefficient: 3 ppm/°C (A grade)
- Low voltage operation: 1.8V
- SO-8 package
- Low dropout voltage: 20 mV (typ) @ 1mA
- Supply Current: 230 µA (typ), ≤ 1 µA disable mode
- Enable pin
- Output voltage options: 1.024V, 1.250V, 2.048V, 2.500V, and 4.096V
- Custom voltages from 0.5V to 4.5V
- Temperature range (0°C to 70°C)

Applications Summary

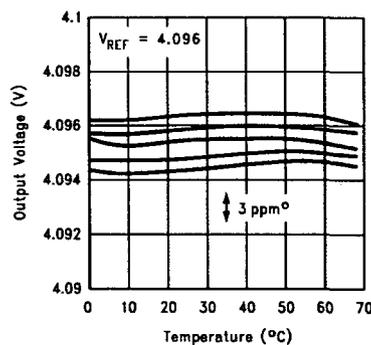
- Portable, battery powered equipment
- Instrumentation and test equipment
- Automotive
- Industrial process control
- Data acquisition systems
- Medical equipment
- Precision scales
- Servo systems
- Battery charging

Typical Application



10107901
C_{OUT}, Output bypass capacitor. See text for selection detail.

Typical Temperature Coefficient
(Sample of 5 Parts)



10107923
Refer to the Ordering Information Table in this Data Sheet for Specific Part Number

4. Circuit board description

The hardware consists of a two-layer printed circuit board equipped with a UART buffer, a voltage regulator, a few passive components and the Bluetooth Module of Ericsson. This Bluetooth Module (ROK101 008) includes the Ericsson Baseband device, a Flash Memory and the Ericsson Radio Module device.

For technical details of the Bluetooth Module please look in the datasheet rok101008.pdf included on the Tool Kit CD.

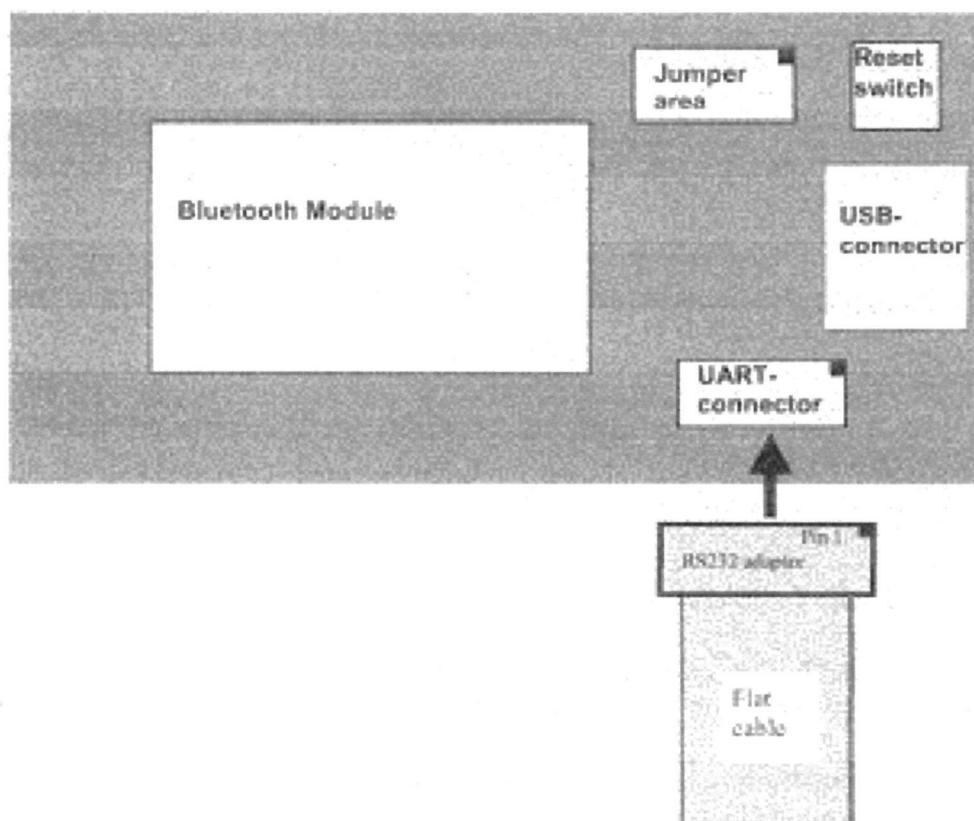


Figure 2: The Tool Kit circuit board

To make easy access possible to certain signals, a jumper area is included on the Tool Kit. The signals in this jumper area and in the other two board connectors are listed in the tables below. Designators within brackets refer to the pin description for the Bluetooth Module in its data sheet.

PCM enables voice use together with an external CODEC-system, but there is no support for this implementation in the Tool Kit.

Note that pin 1 of the connectors is marked with a square dot in the previous layout picture.

4.1 Board Jumper area

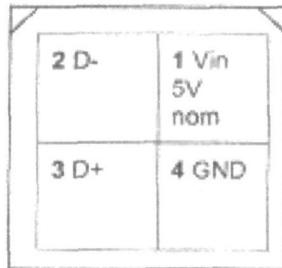
9 RESET (R3)	7 PCM_IN (A1)	5 PCM_OUT (A2)	3 WAKE_UP (B4)	1 Vin 5V nom
10 GND	8 DETACH (C1)	6 VCC 3.3V * (C2, C4 and C6)	4 PCM_SYNK (A3)	2 PCM_CLK (A4)

*) Pin 6 is an output with a load capacity of a few milliamps only.

4.2 Board Uart connector

9 GND	7 not used	5 TXD (B5)	3 RXD (A5)	1 not used
10 not used	8 not used	6 RTS (A6)	4 CTS (B6)	2 not used

4.3 Board USB connector



Note: The USB functionality of the Bluetooth Module (ROK 101 008) is present, but not included in the data sheet (rok101008.pdf).

4.4 Connecting to another power source than USB

If an alternative to the power supply through USB is preferred, the requirements are listed below:

- Supply voltage: min +4.4 V, max +5.25 V connected to Jumper area pin 1 (relative GND pin 10)
- Minimum supply current: 100 mA

APPENDIX C

Bluetooth Information

This part will be updated and new information will be added continuously.

Radio

Fast Frequency hopping (1600channel hop/s) with 79(23) channels available (2.402 to 2.480 GHz) and a symbol rate of 1Ms/s over the air exploits the maximum channel bandwidth in the unlicensed ISM band.

To sustain a high transfer rate in busy radio environment, the frequency hopping together with advanced coding techniques maximizes the throughput.

During Page and Inquiry the hopping frequency is risen to 3200 hops/s to enhance the time needed for connection set-up.

Modulation technique is a binary Gaussian Frequency Shift Keying GFSK, with a BT product of 0.5. The channel bandwidth is 1 MHz and the frequency deviation from the carrier frequency of the RF channel is between +/-140 to +/-175 kHz for representing a '1'/'0'.

A rapid process is ongoing to harmonise Spanish, French and Japanese frequency ranges with the rest of the world.

Country	Frequency range	RF channels	
Europe & USA	2400-2483.5 MHz	f = 2402 + k MHz	k = 0...78
Japan	2471-2497 MHz	f = 2473 + k MHz	k = 0...22
Spain	2445-2475 MHz	f = 2449 + k MHz	k = 0...22
France	2446.5-2483.5 MHz	f = 2454 + k MHz	k = 0...22

Table C1. Frequency ranges used.

Data and parameter formats

There are exceptions in the Bluetooth system for data and parameter formats – general rules below.

- All values are in Binary and Hexadecimal little Endian formats
- Negative values must use 2's complement format
- Array parameter notation is parameterA[i], parameterB[i],...
- All parameter values are sent/received in little Endian format. The least significant byte is sent first – unless noted otherwise.

HCI Opcodes

The Opcodes have been changed during the ongoing standardisation work. Below is the description on how to send opcodes to the module.

Below is the general HCI command packet format depicted as well as a byte order description.

UART speed setting

The baud rate is changed with an Ericsson specific HCI command. HCI_Ericsson_Set_Uart_Baud_Rate

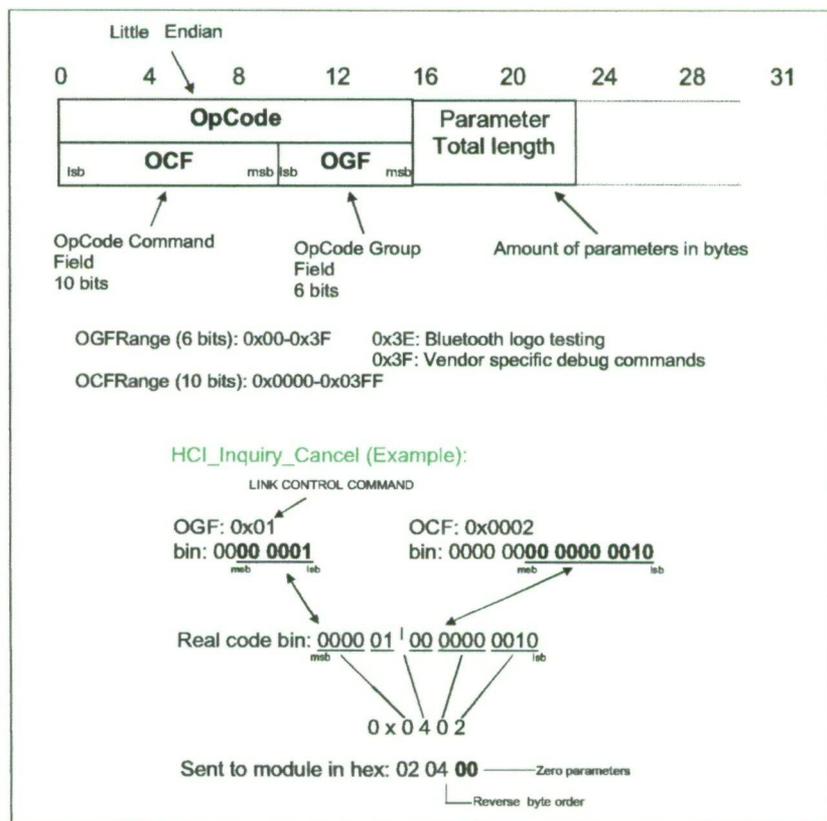


Figure C1. Byte order sent to module