Design, specification and writing of **real-time embedded applications** for a **Bluetooth** product

By

Carolina Ramón Zárate

An internship report presented to the

Universidad Industrial de Santander (UIS) and

l'École Supérieure Chimie Physique Electronique de Lyon (CPE Lyon)

in fulfillment of the requirement for the degree of

Electronic Engineer

BROADCOM - WIDCOMM Inc

Sophia Antipolis, France

2003

ACKNOWLEDGEMENTS

Firstly, I would like to thank my Internship supervisor M. Mickael Jougit, for his kind help and support throughout the duration of this project.

I would also like to thank to whole staff of WIDCOMM group. They gave me the opportunity to realize this internship.

Then I also send my thanks, to the teachers of the third year and The Foreign Relations staff of CPE school.

I cannot end without thanking my family, on whose constant encouragement and love I have relied throughout my time at the Academy. Their courage and conviction will always inspire me. It is to them that I dedicate this work.

ABSTRACT

This work describes the architecture of the software development of a sink application, based on the Bluetooth 1.1 specifications, that handles both voice communication channel and data communication channel between a Bluetooth headset, a mobile phone and an audio player. It allows listening to music in the headset and automatically suspend the Audio/Video (AV) - stream as user answer a call and turns it right back up again when the call ends. The software design shall be robust and customizable to be compatible with almost all phones and audio sources supporting this wireless technology.

The application code is implemented in ANSI C and it supports the following profiles:

A2DP: Advanced Audio Distribution Profile (Requirements for Bluetooth Devices necessary for support of the high quality audio distribution).

AVRCP: Audio/Video Remote Control Profile (Requirements for Bluetooth Devices necessary for the support of the Audio / Video.)

HFP: Hands-free Profile (Defines the minimum set of functions such that a mobile phone can be used in conjunction with a hands-free device.)

HSP: Headset Profile (Requirements for Bluetooth Devices necessary to support the headset use case.)

Headset Application is provided on BROADCOM based platform (???) as a reference design. It can be ported to any Bluetooth platform (supporting included profiles as mandatory feature), shows NVRAM storage capability, supports voice dialing, dial last number and include an adapted user interface.

In addition to software design, this report also presents a test procedure and its results to validate the implemented design.

Table of Contents

List of Figures

List of Tables

Introduction

Bluetooth is a new wireless protocol that allows devices of any kind to discover themselves and communicate without need of user through a radio link. Two Bluetooth units just have to be less than 10 meters away to be able to exchange information.

This affords a wireless world, especially in:

- Offices, with Bluetooth in: keypads, mice, printers, notebooks, mobile phones, PDAs, faxes…
- Living rooms, with Bluetooth in: TVs, game stations, HiFi, MP3 readers, headsets…
- Cars, with Bluetooth in: keys, headsets, mobile phones, and navigation platforms…

Besides cable replacement (e.g. between an application running on a PC and a modem), Bluetooth also provides numerous services as auto-detection, service browsing (discovering of available services delivered by the devices) and so on. It supports numerous protocols, and allows multiplexing (i.e. numerous links at the same time).

Bluetooth devices are organized in mini-networks, where one device plays the role of master and all other ones the role of slave. Between devices either data or voice can be exchanged.

Since the early 90's, Bluetooth technology has undergone great expansion and with this, the user requirements every time become bigger.

The Audio applications and mobile phone applications are getting more and more popular and demanding in Bluetooth market. It is however not enough to make only the mobile applications or only audio application and it is precisely the goal of this project, to join both applications in a Headset device to support the 4 profiles simultaneously and make a compact and robust application leader in the market.

The report is organized as follows: Chapter 4 provides an overview Bluetooth, covering radio layers and profiles and the WIDCOMM stack solution. Chapter 5 presents the design, implementation and results of the application. And conclusions of this work are presented in Chapter 6.

Objectives

The purpose of this work is to develop a real time embedded application for handling an audio streaming and a voice channel aver a Bluetooth connection for a GSM phone headset.

The achievement of this main goal assumes:

- It is important to know in detail the Bluetooth Protocol, including the related profiles and specifications.

- Find an optimal software solution compatible with the most number of mobile phones and audio sources in the market.

- Analyze all different user scenarios to obtain a robust application.

- Implementation of a test procedure to validate the software solution.

Company Presentation

1.1.1 Activities

WIDCOMM, Inc. ("Wireless Internet and Data/Voice Communications") is a leading software provider of Bluetooth™ short-range wireless connectivity solutions. WIDCOMM, the first company to bring certified Bluetooth software to the world, is now the market leader in supplying Bluetooth software and systems to semiconductor, consumer electronic manufactures, and computing device OEMs.

The company's mission is to help these customers bring complete, interoperable, and seamless wireless connectivity solutions to market quickly and easily.

WIDCOMM is uniquely positioned to become the world's leading wireless connectivity platform company, an invaluable supplier and partner to chip and device OEMs seeking to bring wireless products to market quickly and cost-effectively.

WIDCOMM products include the following:

- Wireless connectivity software solutions integrated on-chip for Bluetooth
- Production-ready turnkey designs for products like headsets, phone adapters, printer adapters, access points, PDA, Microsoft window and PC peripherals
- Bluetooth for seamless connectivity in the home, office or car.

Widcomm is now one of the leading companies in Bluetooth software development by working with almost all semiconductor companies.

1.1.2 Histories

Wireless industry veterans Hiep Pham, President and CEO, and Rajiv Kumar, CTO, founded Widcomm in June 1998. Hiep was formerly founder, president and CEO of the Uniden San Diego Research and Development Center. At Uniden, he and Rajiv had assembled a top-tier team of engineers to create wireless communications devices.

In June 1998, Hiep and Rajiv recognized the trend towards convergence of wired PC services and wireless mobile cellular services. They foresaw the potential of Bluetooth™ wireless technology as a connectivity solution for an increasingly mobile society, rather than as merely a way to minimize PC cables. Together with a newly formed team of handpicked engineers, they formed WIDCOMM.

Market development has proved them to be right.

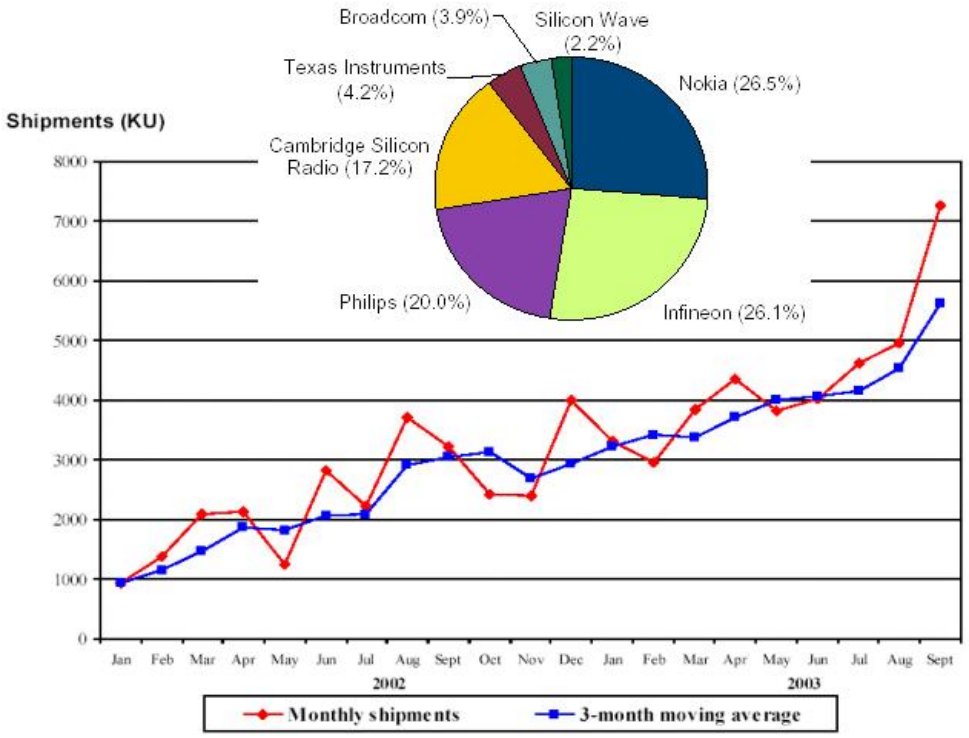


Figure 1: Widcomm Market

Widcomm is now one of the leading companies in Bluetooth software development by working with almost all semiconductor companies.

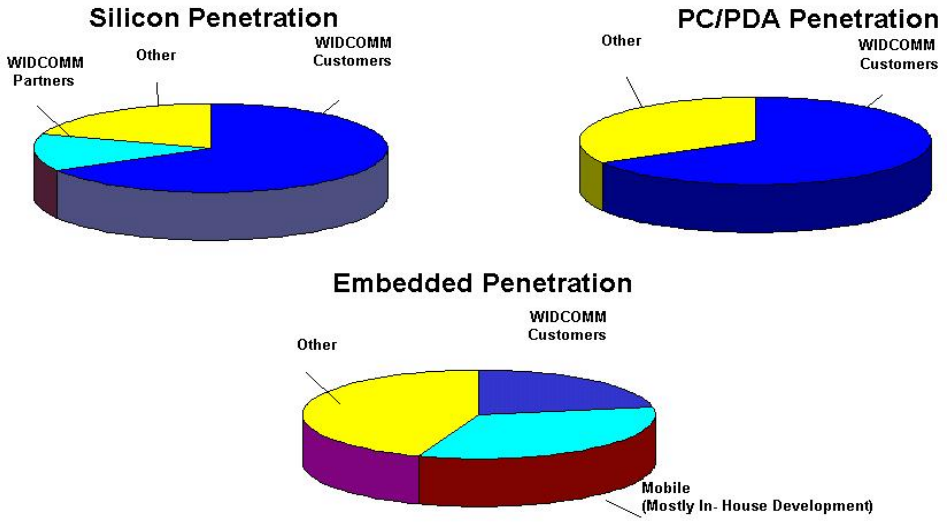


Figure 2: Widcomm Penetration

In April 19, 2004, Broadcom, a leading provider of highly integrated semiconductor solutions enabling broadband communications, has acquired Widcomm.

1.1.3 Partners

Widcomm works directly with selected strategic companies, in various domains:

- Embedded software solutions with TI, Atmel, CSR, Philips, Conexant, OKI,…
- Computing solutions (PC/Notebook computers, PDA/Handheld computing) with IBM, Compaq, Samsung, MSI, …
- Products (desktop accessories, audio devices, HID-USB devices) with Sony, Tecom, Nortel Networks, Plantronics,…

Figure 3: Widcomm Partners

1.1.4 Brief presentation of Broadcom

Broadcom is a semiconductor manufacturer founded in 1991 by Henry T. Nicholas III, Ph.D. and Henry Samueli, Ph.D. with the vision of enabling broadband communications by leveraging their combined 35 years of communications integrated circuit experience obtained at TRW in the military communications industry, at UCLA in the academic research community and at PairGain Technologies in the commercial telecommunications industry.

Broadcom Corporation is a leading provider of highly integrated semiconductor solutions that enable broadband communications and networking of voice, video and data services. We design, develop and supply complete system-on-a-chip (SoC) solutions incorporating digital, analog and radio frequency (RF) technologies, as well as related hardware and software system-level applications.

By applying a disciplined financial management, Broadcom has rapidly increased its revenue.

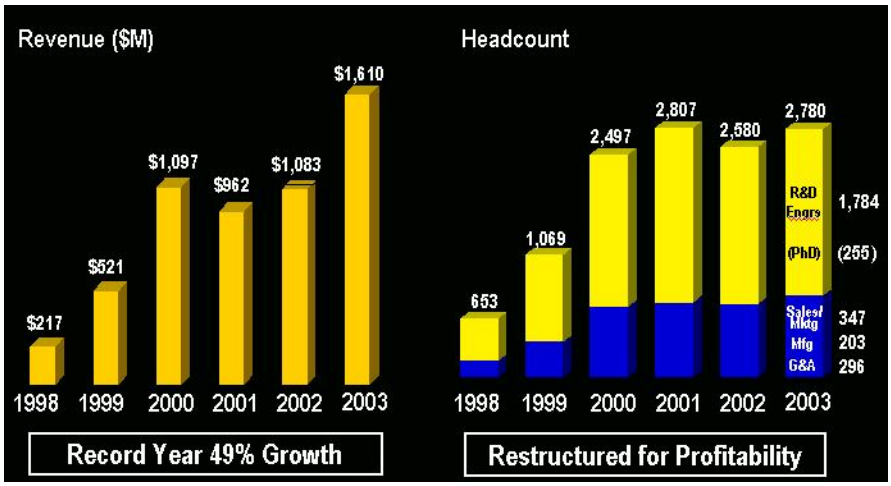


Figure 4: Broadcom Growth

Broadcom employed about 2800 employees all over the world.

Broadcom has engaged in more than 20 strategic acquisitions since 1999 to expand its product offerings and enable the convergence of technology into advanced system-on-a-chip solutions for the delivery of voice, video and data in residential broadband gateway; enterprise and storage networking; and mobile and wireless communications applications.

Features of Bluetooth

This chapter will describe an overview of the Bluetooth technology.

1.2 BLUETOOTH LAYERS

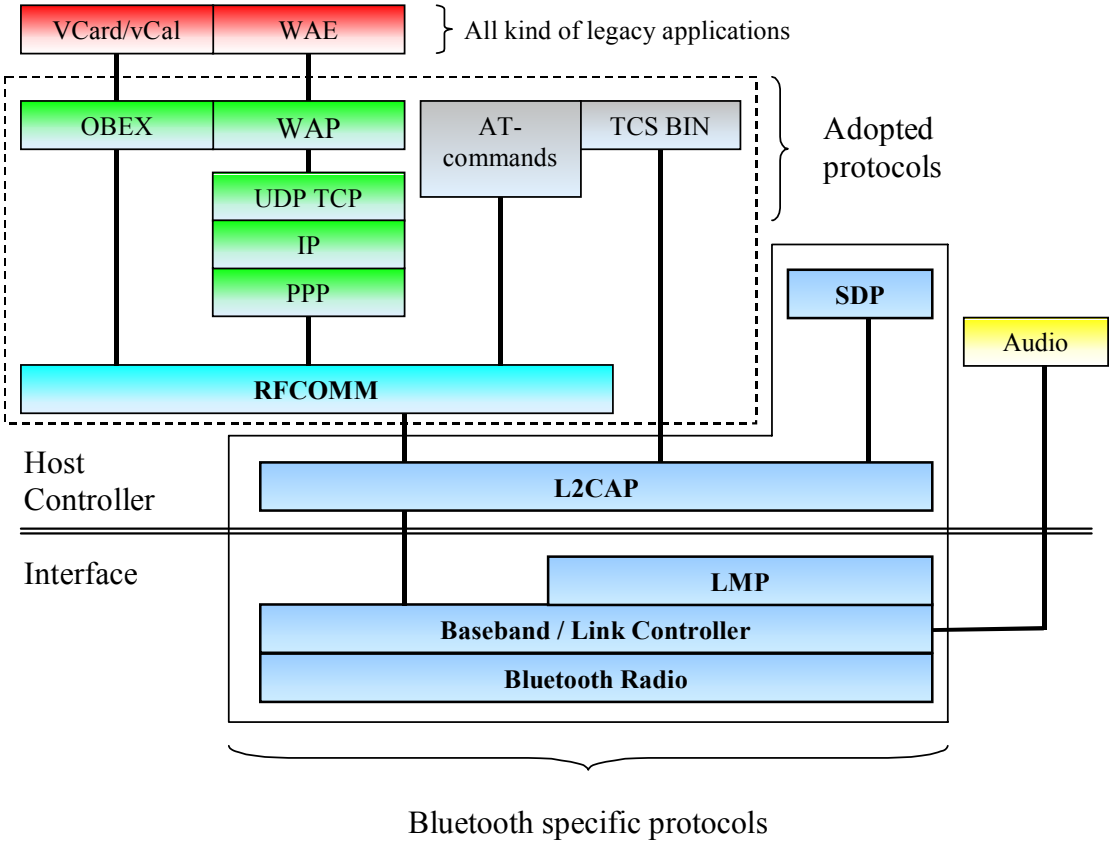


Figure 5: Bluetooth Layer

The Bluetooth radio is totally hardware, and the baseband is both software and hardware.

Each part of the whole stack has a specific role to play. **The link controller and the link manager protocol (LMP)** are up to manage the physical connection between two Bluetooth devices in itself, following the specific protocol over the air Bluetooth protocol.

The **logical link controller and adaptation protocol (L2CAP)** has to transform data coming from higher layers into packets that can be handled by the baseband. It also manages logical connections between two Bluetooth devices, allotting logical links between two remote applications.

The **service discovery protocol (SDP)** has to find out which services are provided by other devices in range.

The **RFCOMM** module is mainly a multiplexer module, using virtual serial ports. It assigns a logical channel to each application using the Bluetooth module.

Higher software layers are used depending on the application that runs over the host controller.

To resume, exchanges between two devices rely on:

- A physical link, provided by the Bluetooth radio, and managed by the **link controller** and the **link manager**;
- A logical link, between two Bluetooth devices, managed by the **L2CAP** module;
- A local logical channel, assigned by the **RFCOMM** module.
-

In range Bluetooth devices are organized in mini-networks :

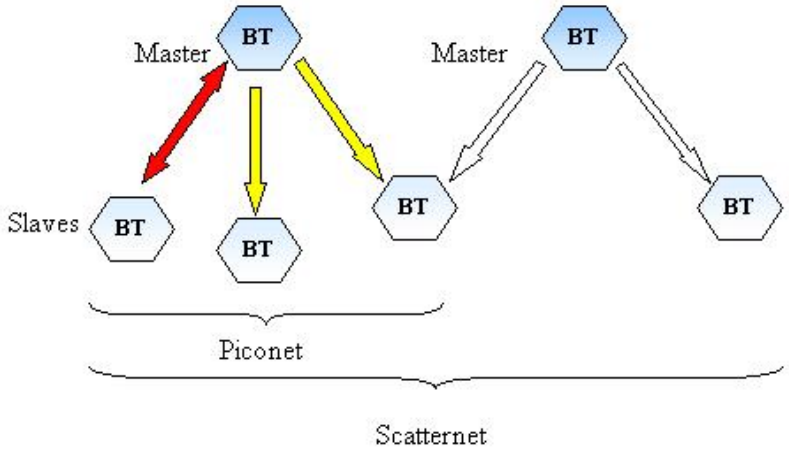


Figure 6: Network structure Bluetooth

- A piconet is a temporary network between 1 master and up to 7 slaves.

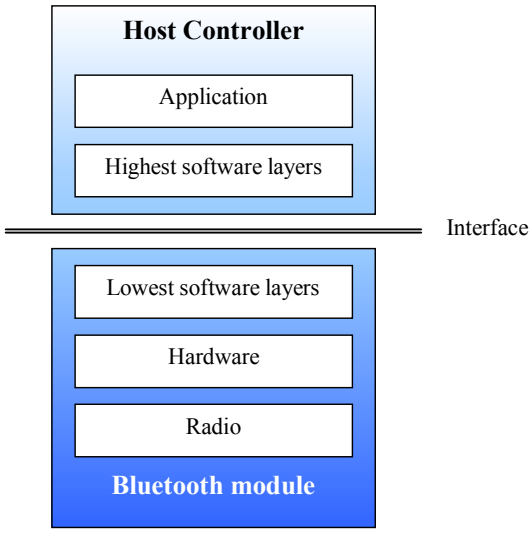
- A piconet is totally defined by its master. The master sets all the “rules” during the existence of this network.

- A Bluetooth device can be at the same time master of a piconet and slave in other ones.

The application topology will be a piconet with two sclaves, where headset will be the master and the mobile phone and the audio source will be the sclaves.

Types of Bluetooth devices

At present, two types of Bluetooth devices exist: “hosted” devices and “non-hosted” devices. Hosted devices are actually made of two parts. The first one is the Bluetooth module itself, integrating the Bluetooth radio, hardware and the lowest part of the software. The second part is hosted by the host controller (e.g. a PC). It consists in the highest part of the software, and application (typically legacy application) using it.

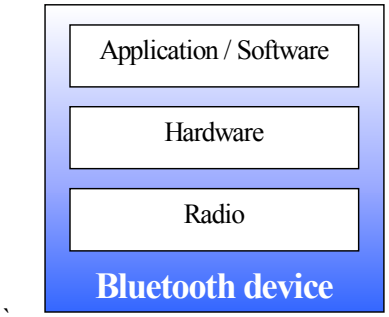


"Hosted" Bluetooth device

The two parts making up the Bluetooth device (i.e. the module + the host controller) are bounded with a physical link, typically USB or RS232. To ensure portability and compatibility, a logical Host Controller Interface is defined, which lists all the commands that a Bluetooth module has to understand, so that any kind of application running on any kind of host controller is able to drive a bluetooth module.

Furthermore, the whole software stack functions are precisely defined, with its specific or adapted layers, once again to ensure compatibility between different vendors' solutions.

The second type of device is "non-hosted" device. In fact, it means that the whole software and the application (typically specific) is embedded in the Bluetooth module.



"Non-hosted" Bluetooth device

A Bluetooth headset is the perfect type of device that doesn't need anyway a host controller. A really basic application runs directly over the hardware and the radio. So it doesn't need generic interface (neither physical nor logical). Note that finally, there is no difference from the user point of view between these two types of implementation.

1.2.1 BLUETOOTH RADIO

This module creates the physical link between two BT devices.

The RF signal is transmitted in the ISM band (Industrial – Scientific – Medical unregulated band). Its uses 79 sub carriers with 1MHz interband between 2.402GHz and 2.480GHz. To avoid disturbance induced by equipment already operating in the same band of frequency, mainly microwave ovens, a fast frequency hopping sequence is used.

Within each piconet, a single pseudo-random hopping sequence synchronizes all the slave devices with the master. With a fast frequency of 1600hops/s the fading problem is solved, avoiding the necessity of a diversity antenna system.

The technical features are resumed in the next table:

Radio frequency range	2.4 GHz band (ISM band)
Frequency spacing	1 MHz
Number of radio channels	79 channels
Output power	Class 1: +20dBm (100mW); Class 2: 0dBm (1mW)
Duplex method	TDD: Time Division Duplex
Spread spectrum	FH-SS: Frequency Hopping Spread Spectrum
Frequency hopping bit rate	1600 hops per second
Modulation method	GFSK m ≤ 0.35
Modulation bit rate	1 Mbps
Voice channel	64k PCM(A-law, μ-law), CVSD (Continuous Variable Slope Delta Modulation)
Maximum data channel rate	Symmetric communication: 432.6kbps, Asymmetric communication: 721kbps/56kbps

Table 1: Bluetooth specification table

1.2.2 BASEBAND / LINK CONTROL

The Baseband module role is to transmit to the radio module the data to be sent. It encapsulates data from higher layer into specific packets according to the Bluetooth protocol.

Packets are sent via physical channels (i.e. RF transmission) on logical channels divided into 625μs time slots. Frequency hops are done (basically) each time slot. To ensure full duplex communication, baseband module uses time duplex division.

To perform such multiplexing, Bluetooth protocol uses a combination of circuit and packet switching. Circuit switching uses dedicated channels to send messages, and packet switching uses packets with a delivery address to transmit data.

Different kind of physical links:

-SCO (Synchronous Connection Oriented) link: a point-to-point link (i.e. master to slave link);

-ACL (Asynchronous Connectionless) link: point-to-multipoint link.

A master Bluetooth device supports up to three SCO links in a symmetric link from a master to a specific slave. A slave supports either up to three SCO links from the same master or two SCO links from different masters. SCO packets are exchanged during reserved time slots. ACL packets are exchanged during remaining slots.

The headset application will support only one SCO link and two ACL links.

- **SCO packets**

Type	User payload (bytes)	FEC	CRC	Rate (Kb/s)
HV1	10	1/3	No	64
HV2	20	2/3	No	64
HV3	30	No	No	64

Table 2: SCO packets

HV: High Voice quality.

In addition to HV types, you can find the DV (Data-Voice) packet, which consists in 1 payload header byte; 10 voice bytes plus 0 to 9 data bytes user payload. Data is coded with a 2/3 FEC and has CRC. Rate is 64kb/s for voice plus 57.6kb/s for data (symmetric).

As result of some problems about the receipt of the remote control commands, the SCO packet type was limited to HV2 and HV3 in the application, in order to optimize the bandwidth and allow the data flux.

- **ACL packets**

Type	Payload header (bytes)	User payload (bytes)	FEC	CRC	Symmetric rate	Asymmetric rate (in Kbytes/s)	
						Forward	Reverse
DM1	1	0-17	2/3	Yes	108.8	108.8	108.8
DH1	1	0-27	No	Yes	172.8	172.8	172.8
DM3	2	0-121	2/3	Yes	258.1	387.2	54.4
DH3	2	0-183	No	Yes	390.4	585.6	86.4
DM5	2	0-224	2/3	Yes	286.7	477.8	36.3
DH5	2	0-339	No	Yes	433.9	723.2	57.6
AUX1	1	0-29	No	No	185.6	185.6	185.6

Table 3: ACL packets

DM: Data Medium rate packets. They are used when interferences may change the data.

DH: Data High rate packets. They are used in “clean” places: they have no FEC.

1.2.2.1 Channel control

A piconet is entirely defined by the master (Initially, the master is the device who first initiated the piconet, but then swaps between the master and a slave occur really often). The master’s BD_ADDR generates the FH hopping sequence and the channel access code; and the master’s clock is the root of the whole timing.

There are two main states for Bluetooth devices: standby and connection, plus a lot of sub-states between those two states:

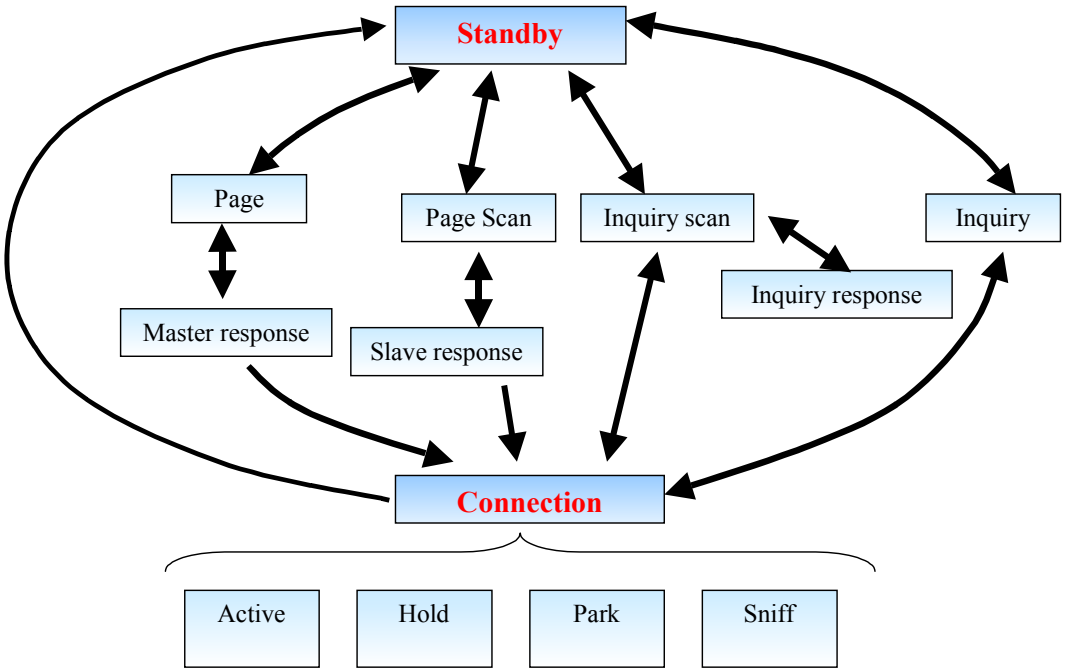
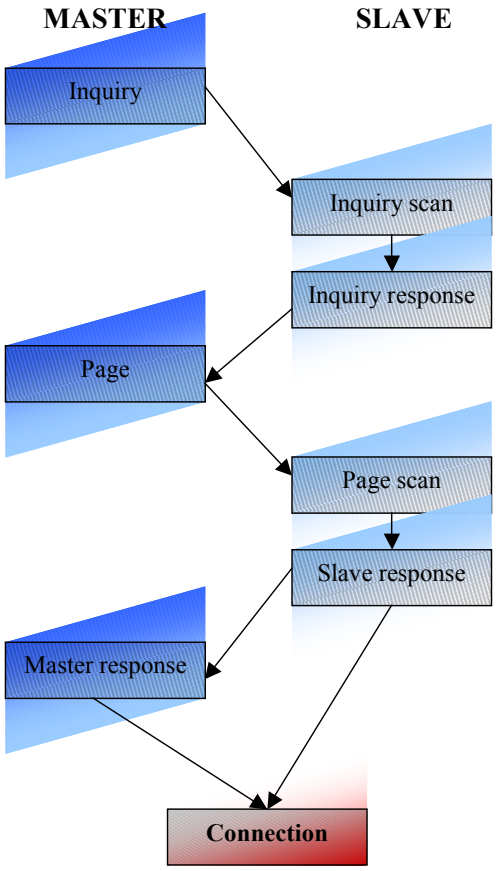


Figure 7: Main States for Bluetooth Devices

Headset must be in sniff mode when there isn't a call active.

The different steps to connect two Bluetooth devices are described below



Connection step	Description
Inquiry	A discovering unit collects BD_ADDR and clocks of all BT devices in range which answer its inquiry message
Inquiry scan	BT devices are listening for scan messages
Inquiry response	Only slave responds, with a FHS packet
Page	Master sends page message to a slave (with slave's Device Access Code) to activate and connect it.
Page scan	Slave listens for its own DAC
Slave response	Slaves respond to master's page message and enter connection state (after receiving FHS packet).
Master response	After slave response, master send a FHS message and wait for a replies to enter connection state

Figure 8: Steps to connect two Bluetooth devices

1.2.3 Audio Stream

Audio transmission on the Bluetooth air-interface is made directly between two basebands on SCO links (which don't pass through L2CAP). Codec use either a 64kb/s log PCM format (A-law or µ-law) or a 64Kb/s CVSD (Continuous Variable Slope Delta Modulation).

1.2.4 HCI: Host Controller Interface

The HCI provides a uniform interface to access the Bluetooth hardware capabilities. For "hosted" devices, the software stack is divided into three parts:

- Legacy application over numerous layers (BT specific or not);
- Specific Bluetooth protocols embedded in the Bluetooth device: link manager and link controller;
- A logical and physical interface (USB, RS232…) to link the host and the BT device.

HCI provides generic (in compliance with Bluetooth specification) commands, events signalization, and data and voice transmission understandable by the HCI firmware, and so ensure portability.

Each HCI firmware is specific of its device: it translates former commands into hardware compatible commands to access baseband commands, registers…

Three host controller transport layers have been defined in Bluetooth specification: HCI USB, RS232 and UART transport layers.

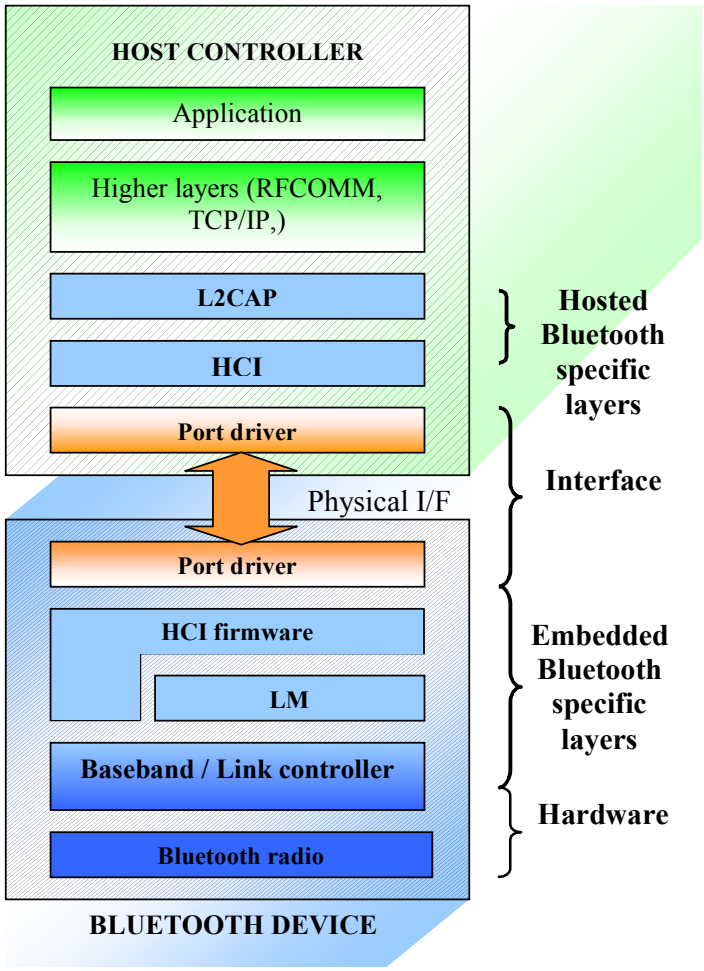


Figure 9: Host Control Interface

1.2.5 LMP: Link Manager Protocol

LMP module is used for:
- Link set-up
- Security
- Control

A link manager essentially talks to the remote one to exchange information and control through the link controller. Exchanges with higher layers are possible but a little bit hazy (they could be of interest to inform the LM about all kind of settings such as security level, QoS, and so on…).

The link manager also manages: authentication and encryption, mode and SCO connections:

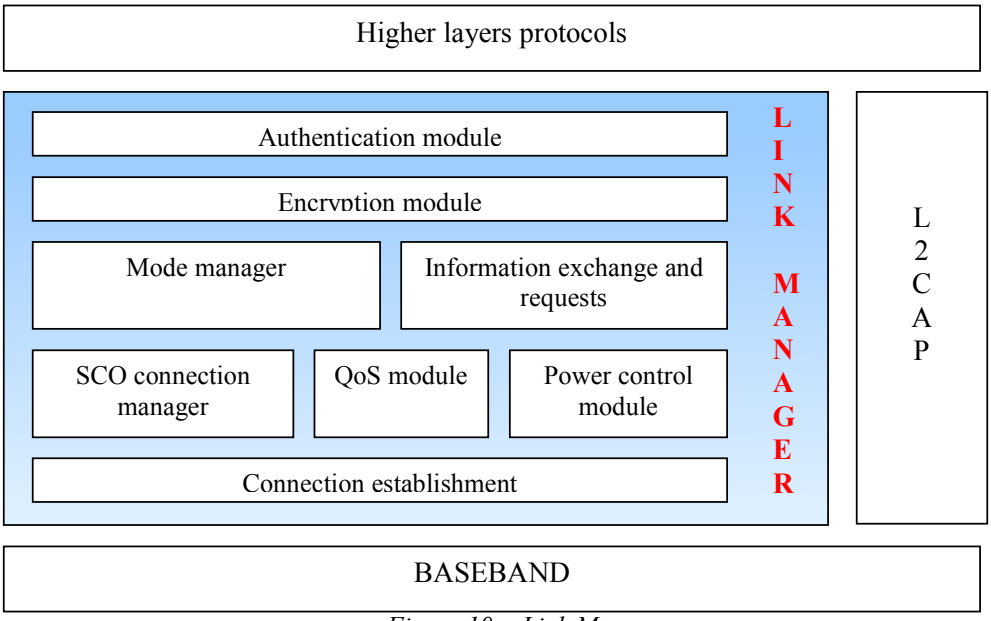


Figure 10: Link Manager

1.2.5.1 Bluetooth security

Security in Bluetooth is a great issue as any device can access another one via the over-the-air interface. There are two main built-in security techniques to prevent eavesdropping (spying) and falsifying: authentication and encryption. Authentication is used to prevent unwanted access to data and to prevent the falsification of the message originator. Encryption is used to prevent eavesdropping or spying.

These features are based on a secret link key (shared by the paired devices) generated during pairing procedure. The Generic Access Profile defines three security modes for a device:

- Security mode 1 (non-secure): a device will initiate no security procedure;
- Security mode 2 (service-level enforced security): no security procedure is initiated until L2CAP channel establishment;
- Security mode 3 (link-level enforced security): security procedures are initiated before the link set-up at the LMP level is completed.

To provide such services, four entities are defined:

Entity	size
BD_ADDR	48 bits
Private user key, for authentication	128 bits
Private user key for encryption	8-128 bits
RAND (random number)	128 bits

1.2.6 L2CAP: Logical Link Control & Adaptation Protocol

L2CAP provides:
- Protocol multiplexing
- SAR: Segmentation And Reassembly
- QoS: Quality of Service
- Group abstractions

1.2.6.1 General operations

- L2CAP only supports **ACL links**
- Thanks to SAR and protocol multiplexing, L2CAP transforms any kind of data from higher layers into packets (up to 64 Kbytes) understandable by the baseband
- L2CAP uses the concept of channels to establish pathways between different applications on BT devices. Channel endpoints are given local ID: CID.
- L2CAP provides two types of channels: connectionless and connection-oriented (different from SCO link) channels. Connection-oriented data channels represent a connection between two devices (so CID identifies each endpoint of this channel), and on the other hand, connectionless channels restrict data flow to one direction, from a master to a "group"of BT slaves. In this case CID represents one or more remote devices.

Each channel is bound to one and only one protocol. Each packet received on a channel is directed toward corresponding higher-level protocol.

1.2.7 SDP: Service Discovery Protocol

Service Discovery module provides some kind of virtual plug'n'play over the air protocol: it allows discovery of all Bluetooth devices in range and lists their available services characteristics. The set of available services is dynamically updated regarding to RF proximity of the devices.

SDP Client-Server functioning:

SDP protocol involves communication between an SDP server and an SDP client. (A single BT device may act both as a server or a client, depending on the overhead application.)

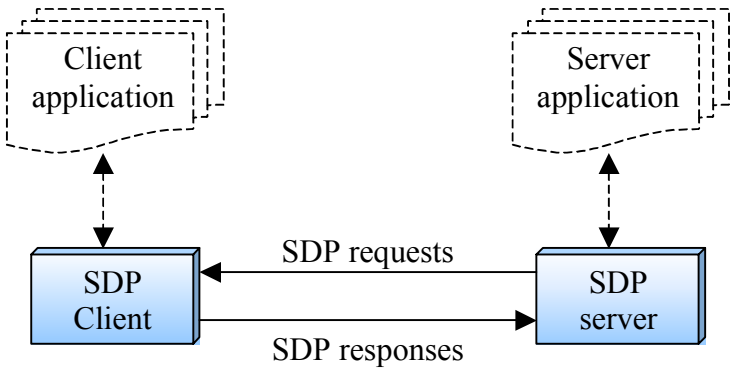


Figure 11: SDP

A service is actually any kind (HW or SW or both) of entity, which can provide information, perform an action or control a resource on behalf of another entity. For each service discovered, the SDP server maintain a service record in which all the service description is described and assign an handler ID to it.

The application will not use the SDP services because it will not search the devices in range.

1.2.8 RFCOMM (Based over ETSI TS GSM 07.10)

The RFCOMM module provides emulation and multiplexing of serial (RS232C) ports over L2CAP. It supports up to sixty simultaneous connections between two Bluetooth devices (i.e. up to 60 applications can use the same Bluetooth module, for example numerous applications running on the same PC can use a single BT transmitter: one wants to access a printer, another one (or more) a remote keypad…). RFCOMM is intended to cover applications that make use of the serial ports of the devices in which they reside.

It emulates all RS232 signals: signal common, TD, RD, RTS, CTS, DSR, DTR, DCD and RI.

It also supports multiple emulated serial ports. Each ongoing connection is identified by a DLCI: Data Link Connection Identifier.

Following frame types from TS 07.10 are supported:

SABM (Set Asynchronous Balanced Mode) command

UA (Unnumbered Acknowledgement) response

DM (Disconnected Mode) response

DISC (Disconnect) command

UIH (Unnumbered Information with Header check) command and response.

At any time, there must be at most one RFCOMM session between any pair of device.

If a device has to create a new DLC, it must check if an RFCOMM session is active between him and the other target device. If such link exists the device will establish a new DLC through it.

Start-up procedure:

- Establish L2CAP channel
- Start RFCOMM multiplexer with SABM (on DLC 0) and wait for UA response.

Flow control:

L2CAP flow control: provided by the LM.

Wired serial ports flow control: software control with XON/XOFF or circuits RTS/CTS.

RFCOMM: FCON and FCOFF, modem status command

Interactions between two devices:

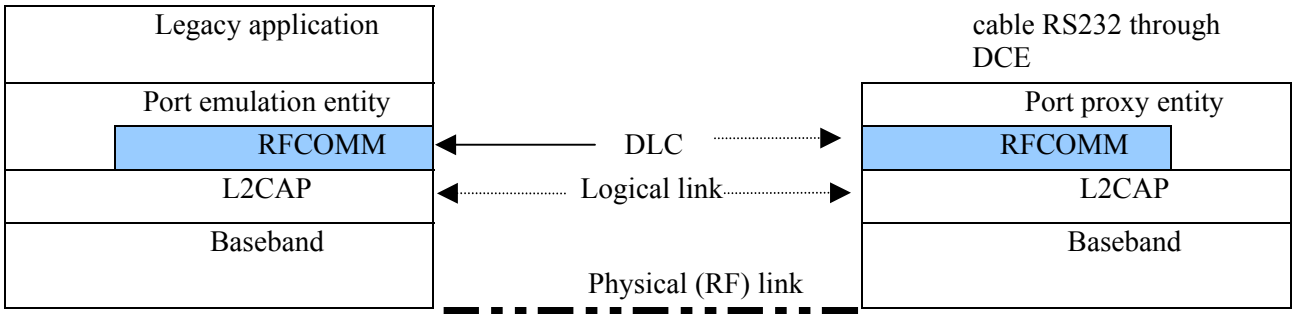


Figure 12: Interactions between two devices

Connexion steps:

- Create a physical link over the air (RF) between both basebands;
- Open a logical channel between L2CAP layers;
- Attribute a DLC in the RFCOMM multiplexer.

1.2.9 Profiles

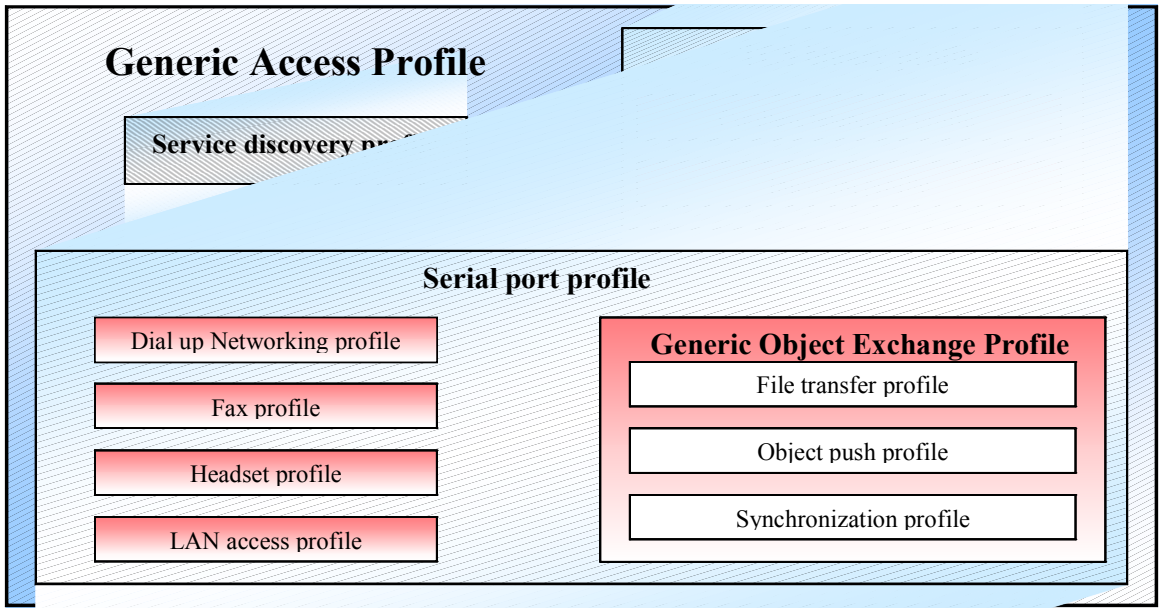


Figure 13: Profiles

The profiles specify how a set of protocols can be used for implementing a particular usage model. Profiles are important to have a common understanding of the protocol stack in order to promote interoperability of usage model implementations.

To ensure compliance, all the applications of Bluetooth devices are officially depicted in the profiles. Profiles are organized in layers, so that profiles using the same low-level protocol can be grouped.

1.2.9.1 GAP – Generic Access Profile

This profile defines how two BT devices discover each other and establish a connection. GAP ensures that any two BT units, even from different manufacturers, can exchange information in order to discover what type of applications both units support.

A Bluetooth device has four main parameters:

- Its address: BD_ADDR
- Its name
- Its passkey (Bluetooth PIN)
- Its class of device

The GAP profile specifies the use of these elements through all the steps to authenticate, begin a communication, set security and so on, so that links and channels are established.

All the others profiles rely on this first one: it is the basis of any application.

1.2.9.2 Service Discovery Profile (SDP)

This profile defines precisely how SDP interacts with the other layers to discover and list all available services. Scenarios covered by this profile are the following:

- Search for services by service class
- Search for services by service attributes
- Service browsing

Of course, the Service Discovery Profile relies on the Service Discovery Protocol.

1.2.9.3 CTP – Cordless Telephony Profile

The Cordless Telephony Profile defines the protocol and procedures that shall be used by devices implementing the so called '3-in-1-phone'. CTP includes support for a gateway, that interfaces a PSTN to Bluetooth, and one or more handset terminals that can communicate through the gateway.

1.2.9.4 SPP – Serial Port Profile

The serial port profile defines the protocols and procedures that shall be used by devices using Bluetooth for RS232 serial cable emulation. So the scenario covered by this profile deals with legacy applications using Bluetooth as a cable substitute, through a virtual serial port.

1.2.9.5 GOEP – Generic Object Exchange Profile

This profile defines the set of protocols and procedures to be used by applications handling object exchanges. Typical Bluetooth units using this profile are notebook PCs, PDAs, mobile phones and smart phones. The GOEP describes the procedures for pushing (or pulling) data from one Bluetooth unit to another.

1.2.9.6 SYNC – Synchronization Profile

The SYNC profile provides a set of procedures for the synchronization of Bluetooth enabled PDA or mobile phones with a Bluetooth enabled computer.

1.2.9.7 HID – Human Interface Device Profile

The HID Profile provides a set of procedures for human interface devices such as Bluetooth enabled keyboards, mice, etc.

1.2.9.8 Headset Profile (HSP)

The HSP provides a set of procedures used by wireless headset or audio gateway Bluetooth devices. Two roles are defined for Bluetooth devices in this profile, Audio Gateway **(AG)** and Headset **(HS)**.

1.2.9.9 PAP – Phone Access Profile

The PAP provides additional procedures – above those provided by HFP – to allow Bluetooth Enabled devices to gain access to services provided by Bluetooth enabled cell phones. It can be used to access information in the phone such as the phonebook, SMS messages, and calendar info. The Phone Access Profile was designed for automotive use, however, the access to cell phone call control and data makes it a potential candidate for a variety of applications.

1.2.9.10 HFP – Hands Free Profile

Hands Free Profile provides a set of procedures to implement an automotive handsfree device and its audio gateway. It is common that on both the handsfree device, and the audio gateway, that both HFP and HSP would be supported simultaneously. Therefore, this architecture also benefits from being more efficient in this configuration.

1.2.9.11 A2DP– Advances Audio Distribution Profile

The A2DP defines the protocols and procedures that realize distribution of audio content of high quality in mono or stereo on ACL channels. A typical usage case is the streaming of music content from a stereo music player to headphones or speakers. The audio data is compressed in a proper format for efficient use of the limited bandwidth

1.2.9.12 AVRCP– Audio Video Remote Control Profile

Features and procedures required in order to ensure interoperability between Bluetooth devices with audio/video control functions in the Audio/Video distribution scenarios. This allows bluetooth devices to control AV equipment. Devices that support this profile may support audio/video streaming by also implementing the Advanced Audio Distribution Profile and/or Video Distribution Profile.

Design Application

This chapter describes the main features on the application that was developed in this Internship.

1.1 THE APPLICATION CONTEXT

The user scenario of the Headset application is shown in Figure 14.

Figure 14: Application Context

Headset can receive an AV-stream since the audio source but it also can handle calls. The ACL connection with the audio source will use 3 L2CAP channels, one to transmit the audio stream and two for data control. Between the mobile phone and the headset, the ACL connection will use 2 L2CAP channels and also, a SCO connection will be used to the voice transmition.

1.2 DESIGN CONSIDERATION

The following criteria has influenced the software design:

- The software design shall be robust in state machine architecture, compatible with the Bluetooth 1.1 specification and customizable to support different phone devices, AV-devices and user interface specifications.
- Headset must be Master of all connection to allow the AV-stream throughput while the RFC is open.
- The State machine shall synchronize the AV, HF and HS services. The HS and HF services have the priority on the AV service.
- The led activity must be assigned to each headset state to allow the user know the results of Bluetooth Operations.
- Headset must be limited discoverable and can connected with paired devices only.
- The packet type of SCO connection must be HV2and HV3 to optimize the bandwidth.
- All paired devices are stored in NVMM in mode FIFO.

1.2.1 System Context

The system context of the Headset Application is shown in Figure 155.

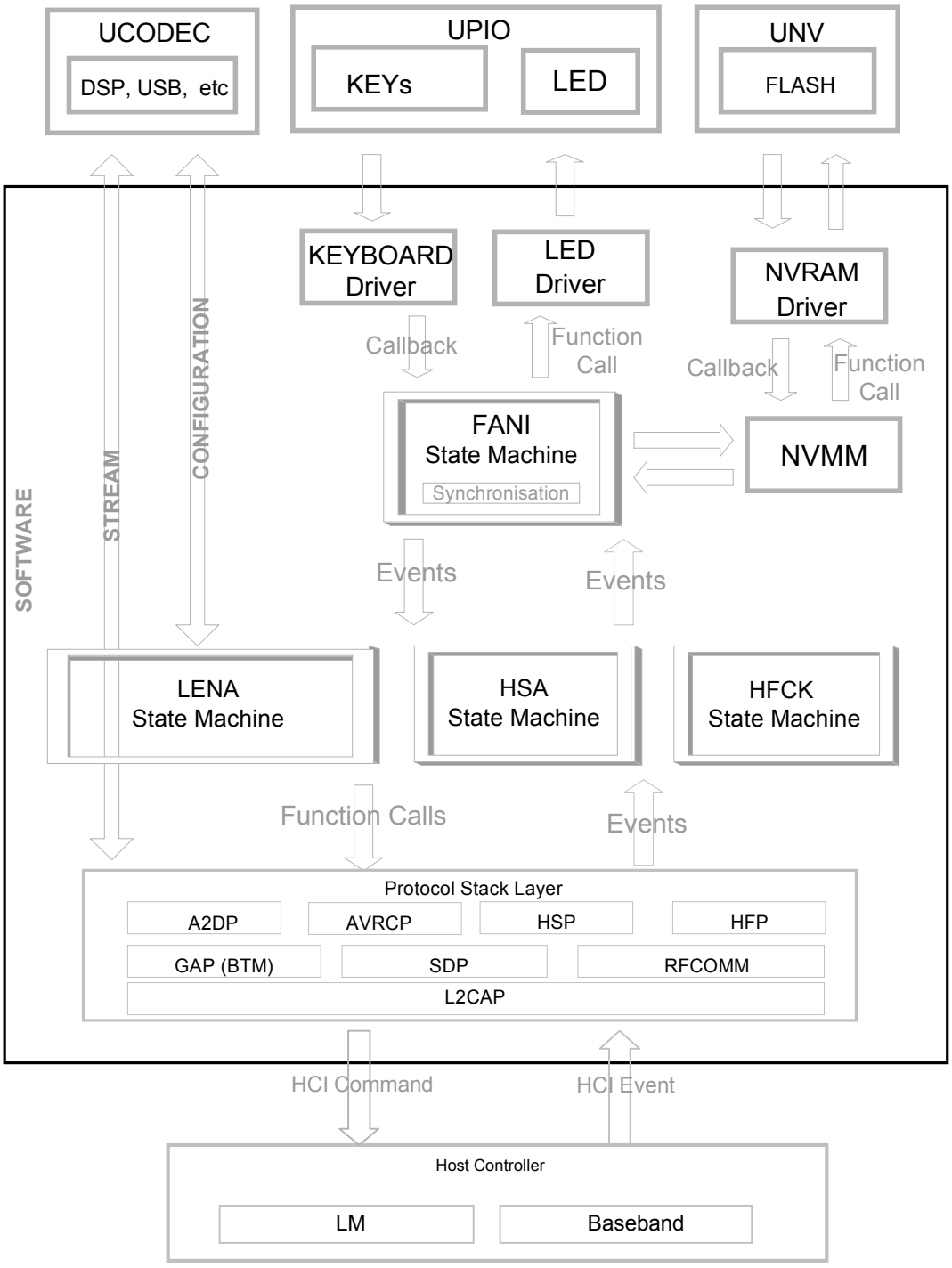


Figure 15: System Context

Headset Application is a state machine that is processed in the context of the application task. It synchronizes all modules around (Applications, Non volatile data storage, User interface) It is responsible of:

- Processing the command and feedback to and from the applications.
- Initializing NVMM, keyboard, and led drivers
- Processing the user input.
- Processing the user output.
- Trig user feedback.
- Synchronizer the discovery and connection process.
- Request data write, read and removal to and from NVMM.
- Managing audio connections.

LENA is a state machine responsible of:

- Processing the command and feedback to and from the WIDCOMM stack.
- Notify FANI state machine
- Registering the A2DP and AVRCP profiles.
- Establishing and releasing the data path between the Ucodec driver and the profile channels.
- Decoding AV-stream.

HSA and HFCK are state machines responsible of:

- Processing the command and feedback to and from the WIDCOMM stack.
- Notify FANI state machine.
- Registering the HSP and HFP profiles.
- Managing AT commands and responses.

NVMM is the Non Volatile Memory Manager. It handles all the flash access or similar non-volatile memory. It runs in the context of the application task.

LENA, HSA and HFCK uses WIDCOMM stack that is implemented according to the 1.1 specifications.

1.3 DETAILED DESIGN

1.3.1 Event Handling

The FANI state machine is processed within a GKI process task. It uses SME (State Machine Engine) services to run its state machine. It processes a transition of a state machine instance.

The Generic Kernel Interface (GKI) provides a small, lightweight kernel interface between the WIDCOMM Stack and the target hardware. GKI features a real-time multitasking mechanism for embedded applications to run the WIDCOMM Bluetooth stack alone, or within another real-time OS.

1.3.2 FANI State Machine Description

This module describes the state machine. It has one instance per device. The application is point to multi point. The diagram is shown in Figure 16.

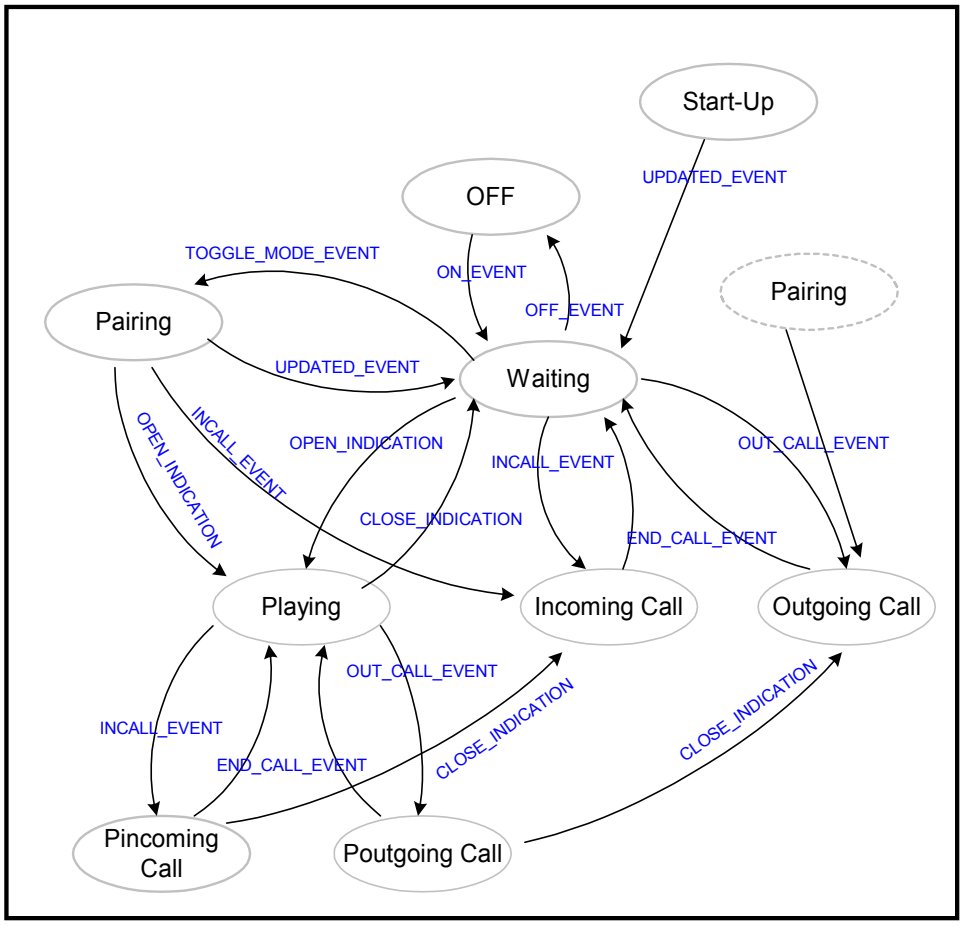


Figure 16: State diagram for FANI

Start-Up is the first state at power up. In this state, the NVRAM is read and the control block is updated with the paired devices information. (UPDATED _EVENT)

After Start-Up, the next state is **waiting**, in this state, the application is waiting for a connection request from any paired device (OPEN_INDICATION) or for a key press from the user (TOGGLE_MODE_EVENT or OFF EVENT) or for a call procedure if a phone is connected (INCALL_EVENT or OUT_CALL_EVENT). The Headset is connectable but it isn't discoverable.

When the headset is turned off, state machine goes to **Off state**, in this state, page scan and inquiry scan are stopped, the system is in low power mode and it only waits for "turn on" input. (ON_EVENT).

Pairing State is the only state where the headset is discoverable and its ready to be pair with any remote device. After pairing process, the NVRAM is updated with the new-paired device information (UPDATED _EVENT).

When an incoming call or an outgoing call, the state machine is in Incoming call or outgoing **call state** to handle the call procedure.

In **playing state**, an AV-source is connected and Headset is receiving an audio streaming. If a call procedure is started, the state machine goes to Pincoming Call State or Poutgoing Call state to suspend the AV-streaming when call is up and restarted it again when the call ends (END_CALL_EVENT).

1.3.3 LENA State Machine Description

This module describes the state machine. It has one instance per device. The application is point to point. The diagram is shown in Figure 17.

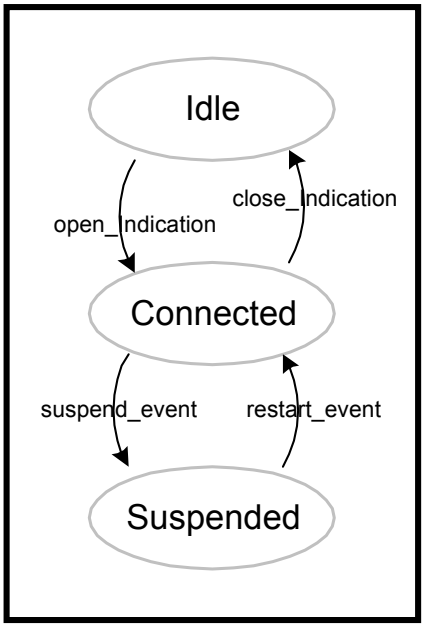


Figure 17: State diagram for LENA

Start-Up is the first state at power up. In this state the connection is basically off. A2DP and AVRCP profile are registered and it is created an audio stream. This state is not showed in the diagram.

After Start-Up, the next state is **Idle**, in this state, no source is connected and the application is waiting for a connection request from any source. (OPEN_INDICATION). The codec is configured to handle a particular stream

When a source is connected, the state machine goes to **Connected state**, in this state, page scan and inquiry scan are stopped and remote and local volume control is handled. The AV-stream is received.

If a SUSPEND_EVENT is received, the state machine goes to **suspended state**. In this state, the AV-stream is suspended. If a RESTART_EVENT is received, the AV-stream is restarted and the state machine goes to **connected state**.

1.3.4 HFCK and HSA State Machine Description

This module describes the state machine. Each application, HSA and HFCK, has the same design of state machine. It has one instance per device. The application is point to point. The diagram is shown in Figure 18.

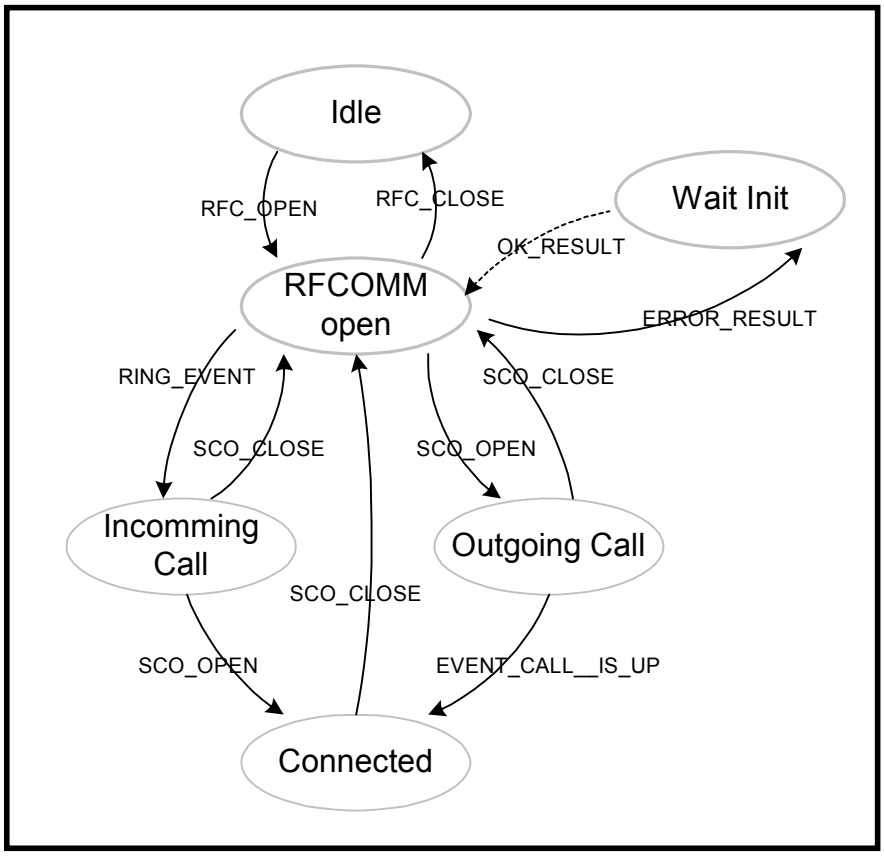


Figure 18: State diagram for HFCK and HSA

Start-Up is the first state at power up. In this state, the Hands-free service or Headset service is registered.

After Start-Up, the next state is **Idle**, in this state, no phone is connected and the application is waiting for a connection request from any cell phone.

The HFCK_EVENT_ERROR_RESULT is sent to the application, according with the 0.96 specification, when the command AT+BRSF will not be sent in the initialization. This event will start the "manual" initialization and state machine will go to **Wait Init state**.

When the RFC is open, state machine goes to **RFCOMM open state.** In this state**,** a cell phone is connected. Headset it's waiting for an incoming call or an outgoing call. When a call is in process, the state machine goes to **Incoming call** or **Outgoing call state**, as correspond.

When the SCO is open and the call is up, the state machine is in **connected state**.

1.3.5 NVRAM Manager (NVMM)

NVMM is a module that has the responsibility of accessing securely the Non Volatile Memory (NVM). In a preemptive multi-task environment, serializing the NVRAM access is mandatory to avoid any collision accessing NVRAM driver resources. Depending on platform the NVMM maybe implemented in flash, EEPROM, chip, etc.

As a consequence NVMM provides a collection of message passing based API calls. All the NVRAM access is processed in the context of the application task to avoid any reentrance accessing the NVRAM driver services.

NVMM offers generic services that should meet the needs of the FANI application.

The NVMM offer the ability to store information within blocks. Nevertheless the information embedded in each block is detailed in FANI NVRAM Information.

1.3.5.1 NVRAM Information

The AV-Headset application needs to store in NVMM :

- Bluetooth protocol information such as BD addresses and link keys
- Remote Service information such as the supported service by each remote.

Both kind of information are linked together and should not change dynamically. One block is allocated per set of devices (correspondents to FANI_DEVICE_NUM = 5 par default) and contains all these data.

1.3.5.2 Behavior

FANI application will read the NVRAM using *NVMM_Get* at startup, if the memory is empty, FANI will store the control block structure using *NVMM_Store*. FANI will modify the NVRAM using *NVMM_UpdateBlock*. The NVRAM storage is implemented FIFO type.

1.3.5.3 Audio streaming:

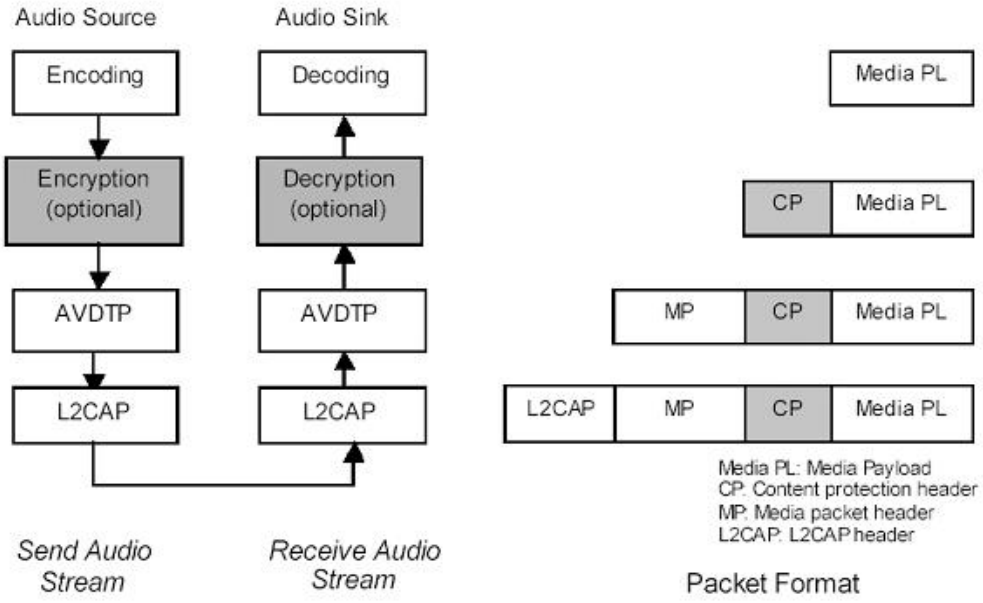


Figure 19: Audio Streaming

The Audio streaming arrives directly to the decoder from the stack.

1.3.5.4 Audio codecs

Many audio codecs can be used within the A2DP profile :

- Low complexity subband codec (SBC), which is the only mandated codec to ensure the interoperability
- MPEG 1,2 Audio (of which the well known mp3), optional
- MPEG 2,4 AAC
- ATRAC family (proprietary codec owned by Sony Corporation)
- All non A2DP codecs

In this case, the SBC was implemented.

The low complexity subband codec is an audio coding system specially designed for Bluetooth Audio/Video (AV) applications to obtain high quality audio at medium bit rates, and having a low computational complexity. SBC uses 4 or 8 subbands, an adaptive bit allocation algorithm, and simple adaptive block PCM quantizer

1.3.6 User Interface – Keyboard and Led Driver

1.3.6.1 Keyboard driver

The keyboard driver allows detecting short and long key press. The user inputs are sent to the application via call back function. The keyboard driver uses the general purpose IO universal driver so that it can be easily ported.

1.3.6.1.1 Behavior

FANI application will register a call back using *KBDV_RegisterCB* and the call back will be called according to the IO status. Each callback function handles the correspondent actions that must be realized when a key is pressed.

1.3.6.2 Led Driver

LED driver provides generic API that the application can call at any time to put the LED in different state (blink, solid, flashing, hold etc…).

The application defines blinking, flashing and hold period. Period between each IO light on or off in 0.1 second unit for flashing and blinking state. It is the hold timeout for the hold state. The following values correspond to the Fani led periods definition. They are redefined to any value.

1.3.6.2.1 Behavior

FANI application will set a LED period using LED_Set.

1.3.6.3 Hardware overview

1.3.6.3.1 Keyboard

The table below provides an overview of the AV – Headset application buttons.

Button Actions	Physical Interface	Key Action
TURN ON **TURN DEVICE IN PARING MODE**	Button 1	Short Key Press
TURN OFF	Button 1	Long Key Press
ANSWER CALL **DIAL LAST NUMBER**	Button 2	Short Key Press
REJECT CALL **ACTIVE VOICE RECOGNITION** **END CALL**	Button 2	Long Key Press
VOLUME UP	Button 3	Long Key Press Short Key Press
VOLUME DOWN	Button 4	Long Key Press Short Key Press

Table 4: Button Functionality

1.3.6.3.2 Led

The table below provides an overview of the AV-Headset application LEDs.

Led Number	Comment
LED1	Flashing indicates following operations in process: 1. Playing Mode or 2. Calling Mode Fast Flashing indicates following operation in process: 1. Pairing Mode
LED2	Flashing indicates following operation in process: 1. Volume changes

Table 5: LED Functionality

1.4 WIDCOMM BTE

WIDCOMM's popular Bluetooth™ Communications Software for Embedded Systems (BTE) is a complete Bluetooth protocol stack, profile and development solution. BTE enables semiconductor companies, OEM product manufacturers, and embedded application developers to quickly and easily port Bluetooth wireless technology onto their platform of choice.

1.4.1 BTE STACK Architecture

¡Error! No se encuentra el origen de la referencia. shows the architecture implemented of WIDCOMM's BTE Stack. BTE is divided into three major components:

1. Lower Stack Layer
2. Core Stack Layer
3. Bluetooth Profiles

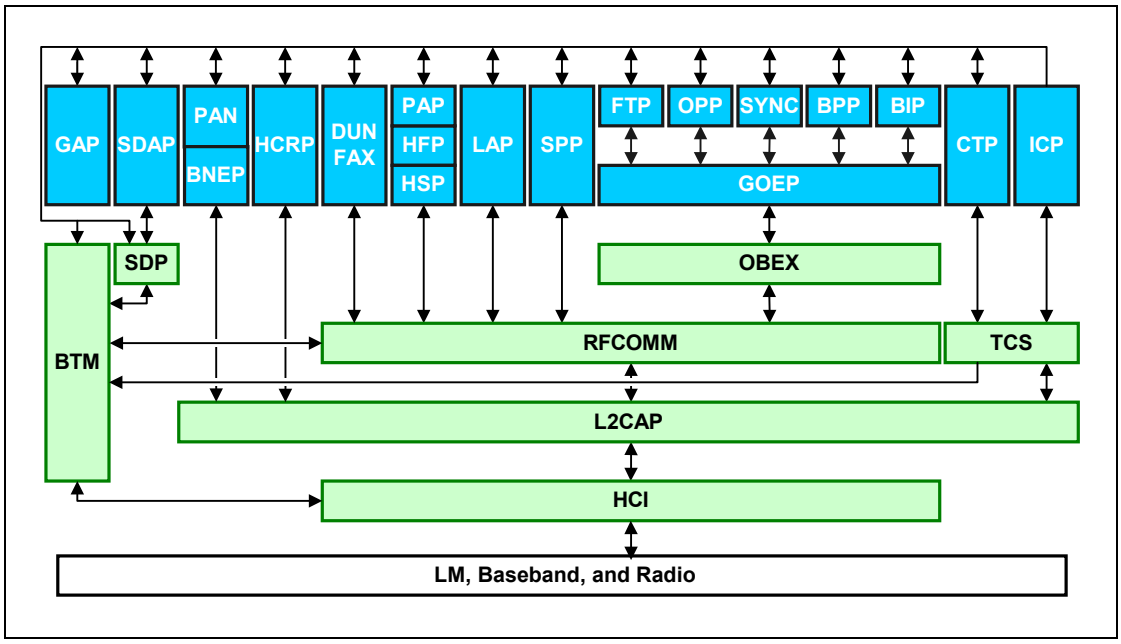


Figure 20: BTE Architecture

The Lower Layer consists of the Bluetooth radio and hardware used for control of the radio. The Core Stack contains protocols common to most Bluetooth Applications. The Bluetooth profiles interface with the core stack, and implement procedures required for interoperation between Bluetooth devices.

1.4.2 BTE Insight – application development and scripting tool

BTE Insight is a PC-based Bluetooth development environment for embedded applications. It provides Bluetooth emulation that allows execution of embedded applications without the need for target hardware. BTE Insight can utilize USB, H4 UART, BCSP/UART connected or simulated radios. The scripting engine provides an extensive test environment and test scripts for standard HCI commands, upper

layer stack and many Bluetooth profiles. The easy to use graphical interface greatly simplifies application development and validation. It provides

The application was totally developed based in the platform BTE WIDCOMM. The graphical interface is show as follows.

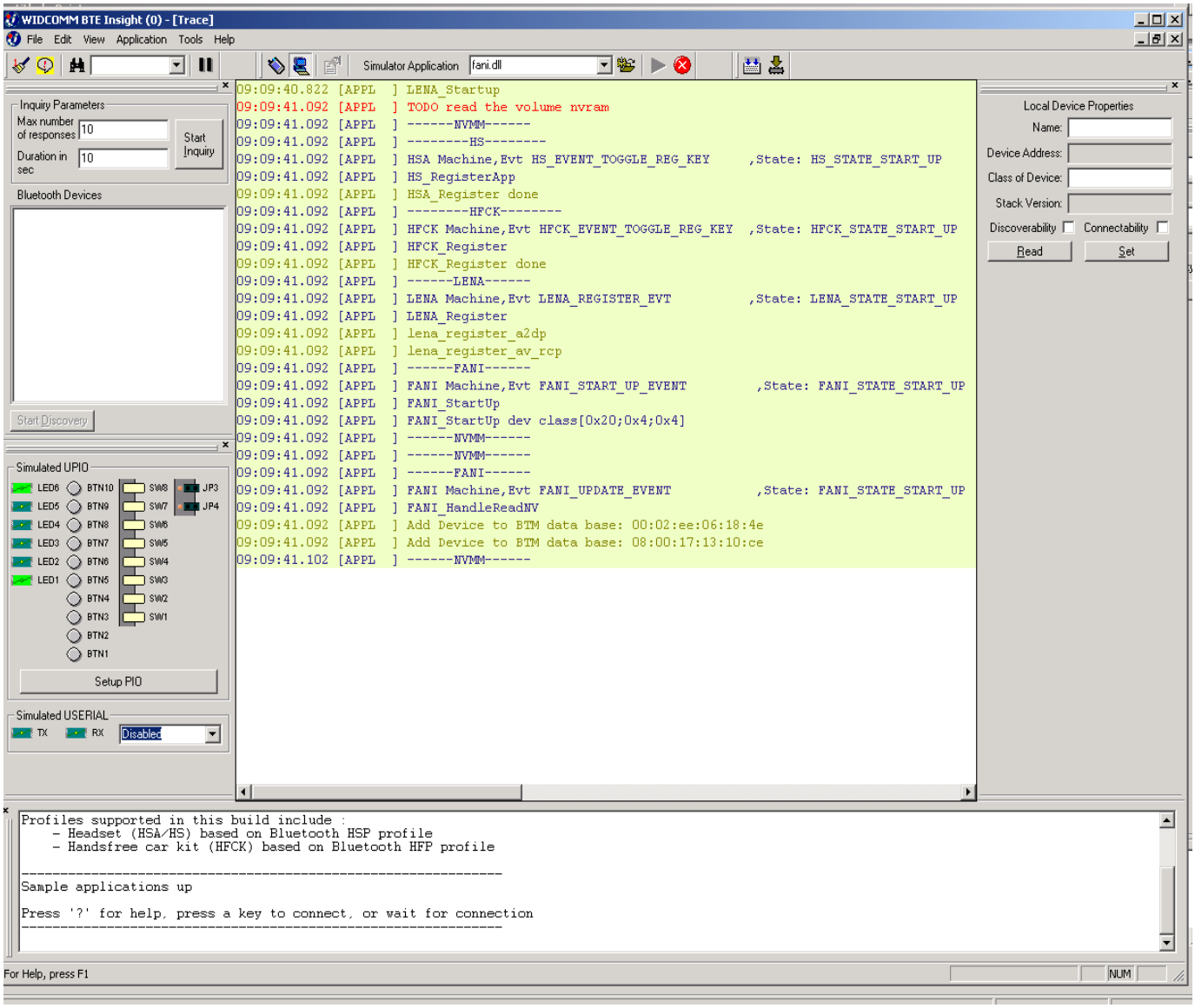


Figure 21: BTE Insight

1.5 CALL FLOWS

The following graphs correspond to the trace visualized in BTE Insight en compliance with the behavior that must have the application.

1.5.1 Start up sequence

Figure 22 shows: the BTM API and NVMM API functions calls made by Headset application, and the events returned to it via the application callback, in order to perform the initialization sequence.

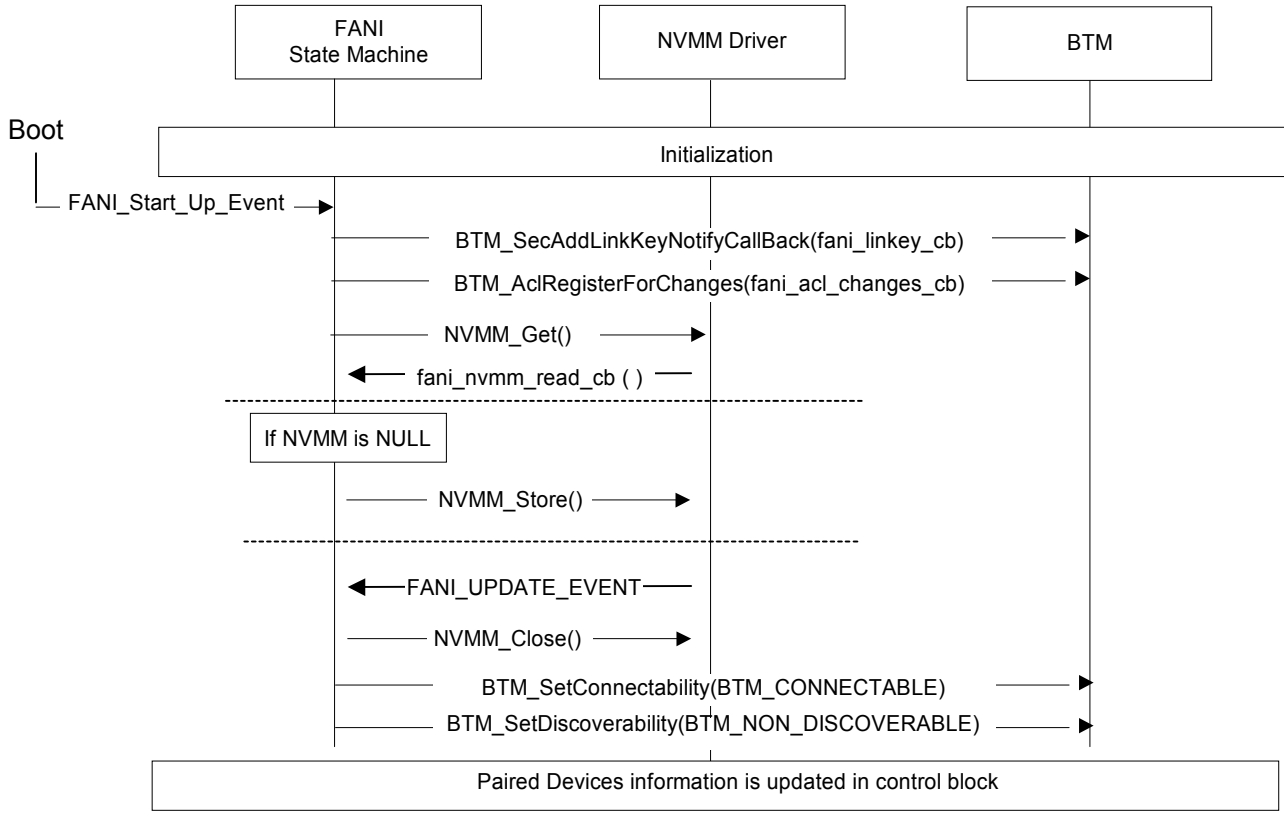


Figure 22: FANI Initialization

FANI_START_UP_EVENT is sent to fani state machine at start up to start it up. It registers the link key and ACL changes call back. In addition, it reads the NVRAM by calling NVMM_Get ( ). If it's Null, FANI will store a new structure by calling NVMM_Store ( ) else local control block will be updated and headset will turn connectable and not discoverable.

1.5.2 Pairing Process

Figure 23 shows scenario for the case where a remote device is paired with the headset.

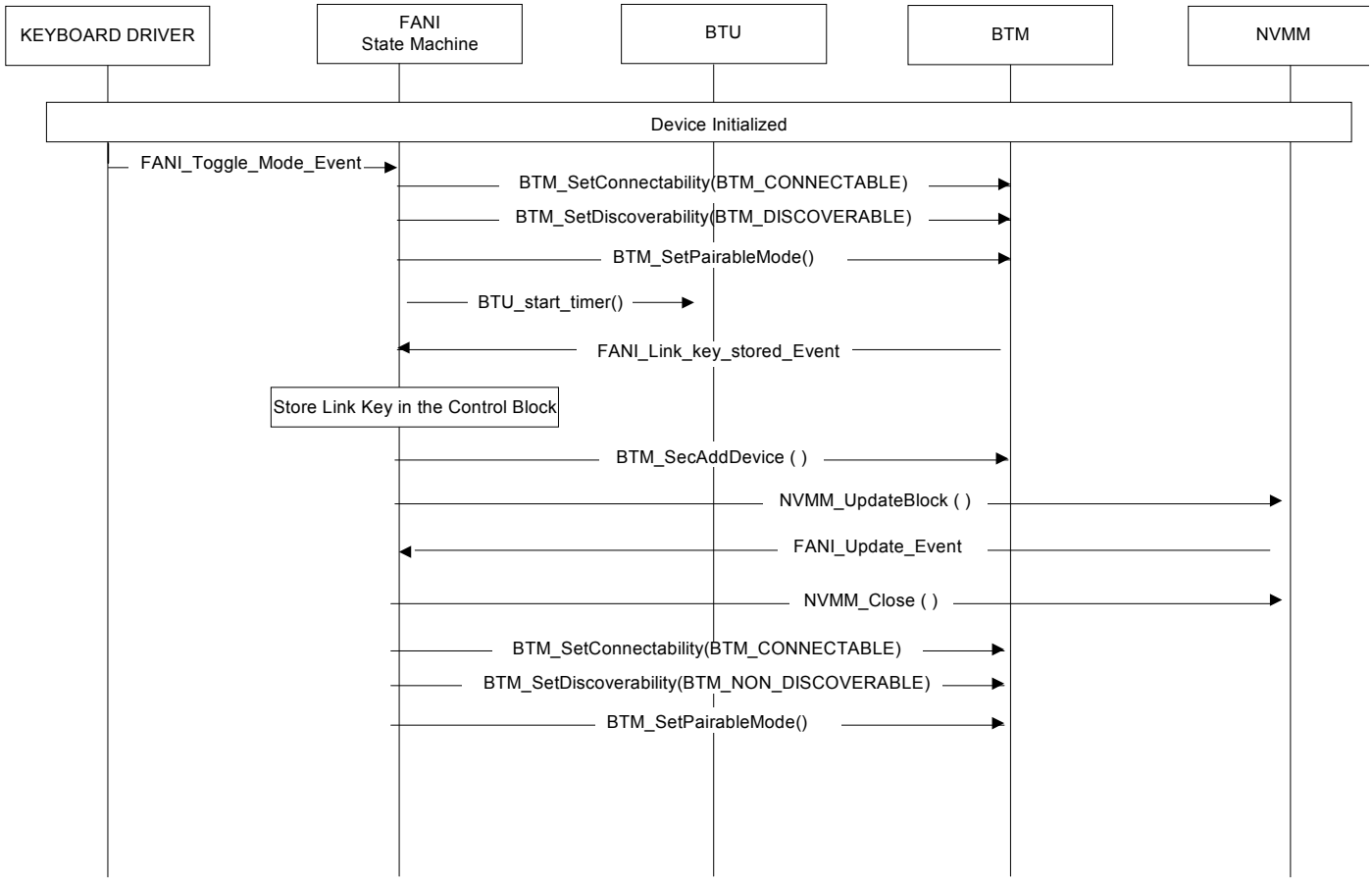


Figure 23: Pairing Process

FANI_TOGGLE_MODE_EVENT is sent to the application and Headset is turned discoverable and connectable.

When a device is paired with the headset, the link key is received (FANI_LINKKEY_STORED_EVENT) and stored in local control block.

After that, the new device is added in BTM database by calling BTM_SetParaibleMode ( ) in order to avoid link key request each time connection.

Finally, the NVRAM is updated with new paired device information by calling NVMM_UpdateBlock ( ).

FANI_UPDATE_EVENT is sent to application to notify the NVRAM successful storage and with this, NVRAM entry is liberated, discoverability is stopped and headset is went out the pairing mode.

1.5.3 Phone Connection Request

Figure 24 shows scenario for a case when service level connection establishment is initiated from mobile phone.

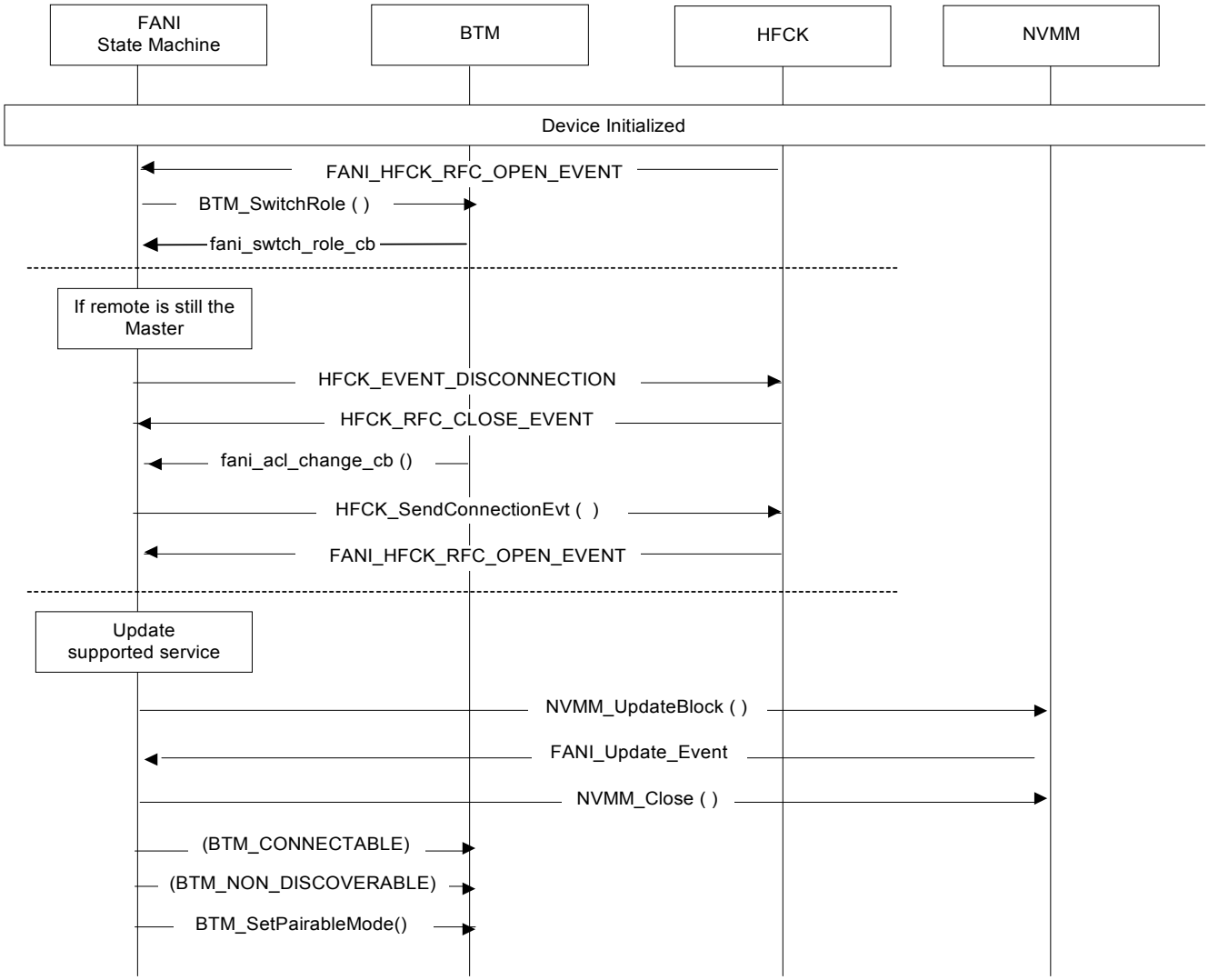


Figure 24: Phone connection request

FANI_HFCK_OPEN_EVENT is sent to the application when RFC is open. At this time, Headset will switch the role by calling BTM_SwitchRole ( ). If remote doesn't support this feature, Headset will send a disconnection request. When the ACL connection has been deleted from the ACL database, a new connection request is sent to remote with the intention of reestablish the connection and obtain the headset be the Master of connection.

1.5.4 Turning On the headset.

Figure 25 shows scenario for a case when headset is turned on. Phone 1 and Phone 2 have previously connected with the headset.

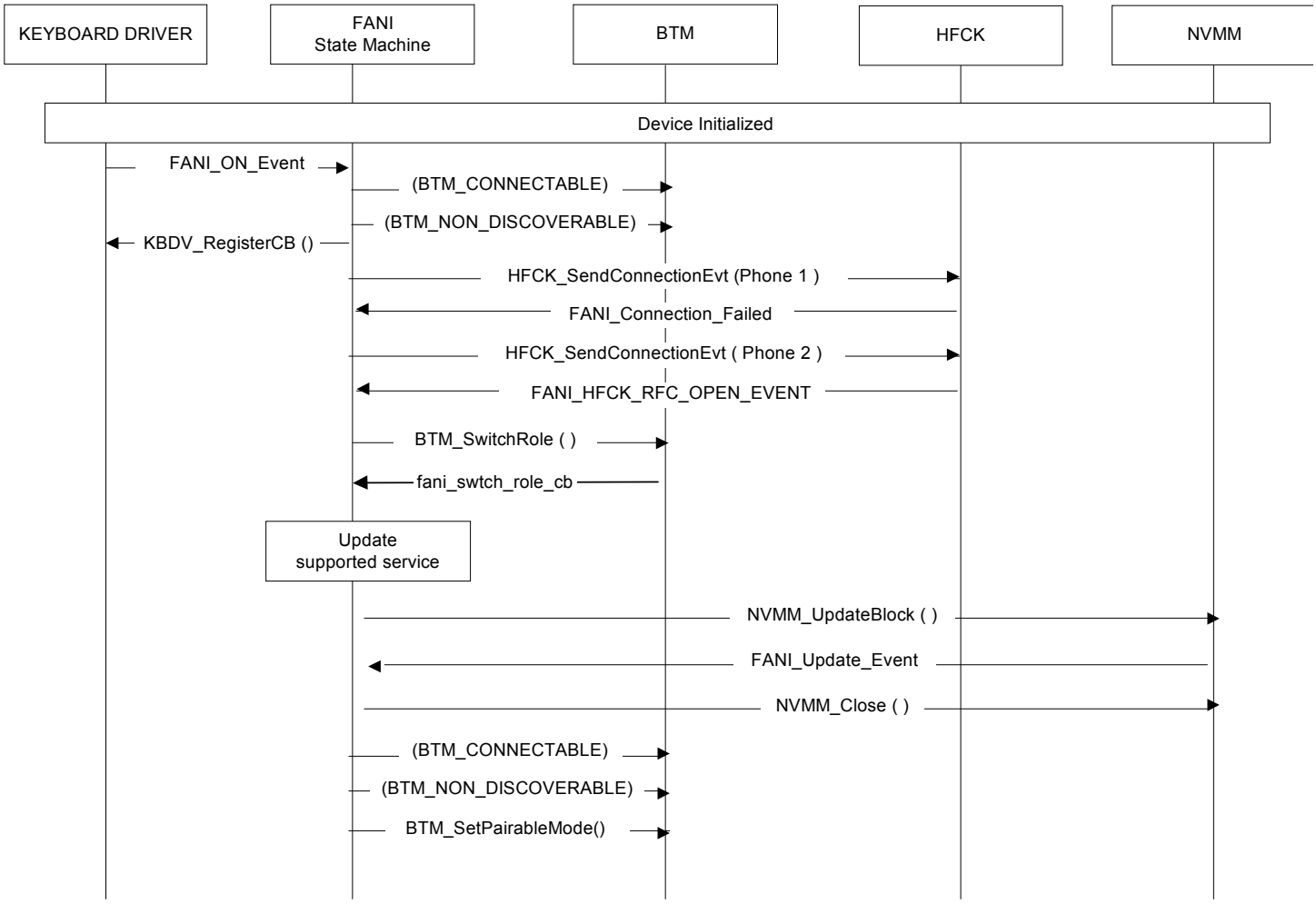


Figure 25: Turning On

FANI_ON_EVENT is sent t the application when headset is turned on. At this time, headset turns connectable and not discoverable and a keyboard callback is registered.

FANI sends a connection event to try to connect with the latest paired phone but due to the phone isn't available, FANI_CONNECTION_FAILED is sent to the application to notify the connection request isn't successful and to send a new connection request with the next device. FANI_HFCK_RFC_OPEN_EVENT is received so RFC was opened.

After that, headset updates the remote supported service and stores it in NVRAM.

1.5.5 Incoming call

Figure 26 shows scenario for the case where AV-device is connected while HFCK successfully receives an incoming call from AG

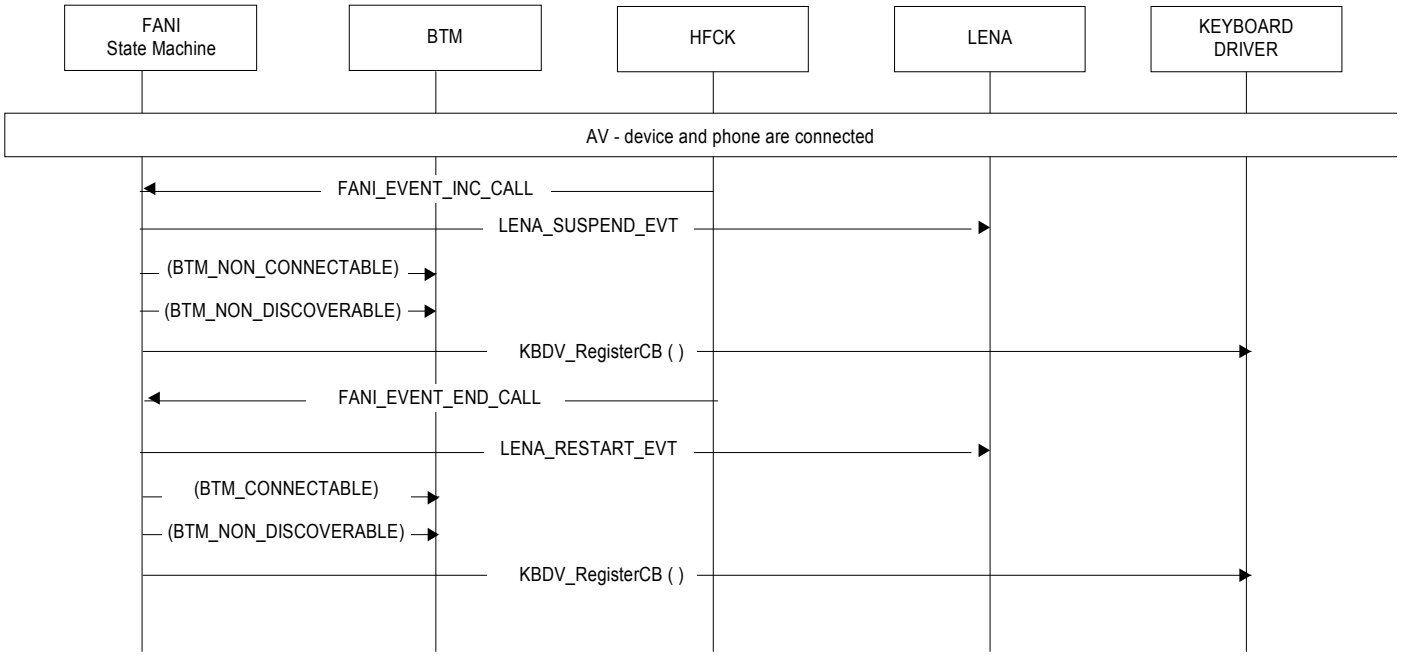


Figure 26: Incoming call process

When an incoming call is in process, FANI_EVENT_INC_CALL is received from HFCK in order to send an event to LENA to suspend the AV-streaming, stop the discoverability and connectability and register a keyboard call back.

When the call finished, FANI_EVENT_END_CALL is sent to the application in order to sent LENA_RESTART_EVT to restart the AV-streaming, turn headset connectable and register the correspond keyboard call back.

1.5.6 Voice Recognition Request

Figure 27 shows scenario for a voice recognition request when RFC is closed.

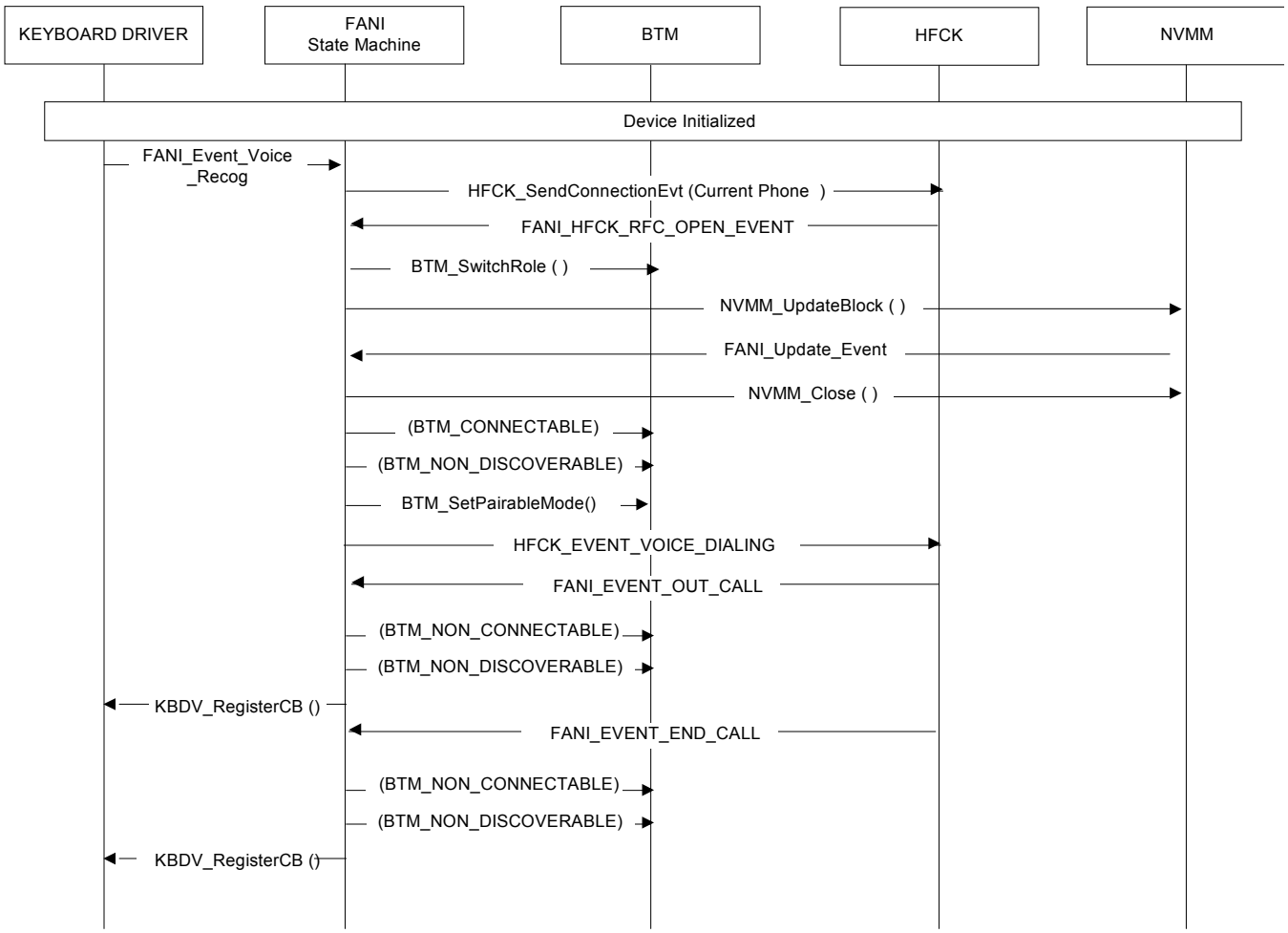


Figure 27: Voice Recognition Request

Voice recognition activation can be requested while RFC is closed.

In this case, FANI will send a connection request before sending the voice dialing activation request..

When FANI_HFCK_RFC_OPEN_EVENT is received, FANI will send the voice dialing activation request and HFCK state machine will handle the sending of the appropriately command when the Service level connection is up and the sequence of initialization AT commands finishes.

The next procedure is the same as an outgoing call.

CONCLUSIONS

The following contributions have resulted from this project:

I have developed an application for transmitting data and audio streaming between three Bluetooth devices, based in the BTE Widcomm Stack and BTE Widcomm Insight. The application demonstrates the capabilities of all used profiles. It allows easy porting to other platform using generics interfaces such as GKI, Universal driver etc…

BTE WIDCOMM has demonstrated robustness and it was sufficient for the requirements of the application development.

The Bluetooth specification is being widely adopted by industry leaders. The possibility for new applications is very exciting with this technology. In the future, the bluetooth Bandwidth must be increased to expand the new applications possibilities. In fact, in this application, there were some obstacles to handle the phone connection and the AV device connection at the same time, due to the bandwidth limitation.

Another limitation encountered in this project was related to the fact that all mobile phones on the market use different sequences of connection and the old phones has many bugs so it was necessary to find the compatibility with each one.

References

Books

[1] Bray, Jennifer and Sturman, Charles F , *Bluetooth – Connect Without Cables* , Prentice Hall PTR , 2001

Specification documents

[2] Bluetooth SIG, *Specification of the Bluetooth System*, Core, version 1.1, 2001
[3] Bluetooth SIG, *Specification of the Bluetooth System*, *Profiles, version 1.1*, 2001

URLs

[4] Derfler Frank J. Jr., *Crossed Signals: 802.11b, Bluetooth, and HomeRF*, *http://www.zdnet.com/products/stories/reviews/0,4161,2470132,00.html*.
[5] Bluetooth SIG, *Bluetooth Wireless Infosite*, *http://www.bluetooth.com/*.
[6] The official Bluetooth membership site, *http://www.bluetooth.org*
[7] The WIDCOMM site, *http://www.widcomm.com*
[8] The BROADCOM site, *http://www.broadcom.com*

ANEXE A Glossary

ACL Asynchronous Connectionless Link. An Asynchronous (packet-switched) connection between two devices created on the LMP level. This type of link is used primarily to transmit ACL packet data.

AG - Audio Gateway This is the device that is the gateway of the audio, both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer.

API (function calls) Application Programming Interface. (This applies to any software interface, not just software components deemed application.)

Application Layer The group of protocols at the user level. The application layer in the Bluetooth protocol layers will contain those protocols involved with the user interface (UI).

AT Command Handler A module that handles the AT commands which control a phone or modem (between a DTE and a DCE).

ATM Asynchronous Transfer Mode.

Authentication The process of verifying 'who' is at the other end of the link. Authentication is performed for devices. In Bluetooth, this is achieved by the authentication procedure based on the stored link key or by pairing (entering a PIN).

Authorization The process of deciding if device X is allowed to have access to service Y. This is where the concept of trusted exists. Trusted devices (the device is authenticated and indicated as "trusted"), are allowed access to services. Untrusted or unknown devices may require authorization based on user interaction before it is allowed access to the services. This does not principally exclude that the authorization might be given by an application automatically. Authorization always includes authentication.

Baseband The baseband describes the specifications of the digital signal processing part of the hardware: the Bluetooth link controller, which carries out the baseband protocols and other low-level link routines.

BD_ADDR Bluetooth Device Address.

Bluetooth (BT) An open specification for wireless communication of data and voice. It is based on a low-cost short-range radio link facilitating protected ad hoc connections for stationary and mobile communication environments. Bluetooth clock the master timing mechanism defined by the master of the piconet. Bluetooth device A device that contains hardware and software allowing it to communicate with another Bluetooth device.

Bluetooth device class A parameter that indicates the type of device and which types of services that are supported. The class is received during the discovery procedure. The parameter contains the major and minor device class fields. The term "Bluetooth device class" is used on the UI level.

Bluetooth device name The name of the device. (248 bytes maximum)

Bluetooth device type The term "Bluetooth device type" is used on the UI level. This term overrides the terms "Bluetooth device class" and "Bluetooth service type" when there is a mix of information containing both Bluetooth Device Class and Bluetooth Service Types.

Bluetooth passkey The name of the PIN. The term "Bluetooth passkey" is used in the UI. See PIN.

Bluetooth service type One or more services a device can provide to other devices. The service information is defined in the service class field of the Bluetooth device class parameter.

Bluetooth Session The activity and participation of a device on a piconet.

Bond A link key that is exchanged between two devices. The key is used for future authentication between the devices. See also bonding.

Bonding Bonding is the creation of a relationship between two devices. The bond is a link key .The relationship is created when the link key is exchanged between two devices. The devices are known to each other prior to the bonding procedure. A user initiates the bonding procedure and enters a passkey with the explicit purpose of creating a bond between two devices. This differs from the authenticate using a passkey procedure where the user is requested to enter a passkey during the establishment of the link.

Browser An application that allows interaction with Internet web pages.

Channel A logical connection on L2CAP level between two devices serving a single application or higher layer protocol.

Class of device See Bluetooth device class. Also abbreviated as CoD.

CO Connection-oriented.

CODEC Coder/Decoder. A device that converts analog to digital, and digital to analog for transmission over a digital communications system.

Component An architecture element denoting an identifiable set of software that performs a well-defined purpose.

Connect to service The establishment of a connection to a service. If not already done, this includes establishment of a physical link, link and channel as well.

Connectable devices Any device within range that will respond to paging from an initiator device.

Connectable mode A device that responds to paging (an attempt to establish a communication link) is said to be in connectable mode. The opposite of connectable mode is non-connectable mode.

Connected device A device that is currently connected to the LocDev.

Connection A connection between two peer applications or higher layer protocols mapped onto a channel.

Connecting A phase in the communication between devices when a connection between them is being established. (Connecting phase follows after the link establishment phase is completed.)

Connectivity A domain of interconnected components that adhere to a defined set of connection rules. The set of rules is termed Connectivity Architecture.

CRC Cyclic Redundancy Check.

CTP Cordless Telephone Profile.

CVSD Continuous Variable Slope Delta Modulation.

DAC Device Access Code.

DCE Data Circuit-Terminating Equipment. In serial communications, DCE refers to a device between the communication endpoints whose sole task is to facilitate the communications process; typically a modem.

Device Discovery The mechanism to request and receive the Bluetooth address, clock, and class of device, used page scan mode, and names of devices.

Device Layer The group of protocols that handles the hardware in a Bluetooth device. The device layer handles components such as the display, keypad, and RF communications.

Device security level Access to a device can be denied based on the required device security level. There are two levels of device security: trusted device and untrusted device. See also service security level.

DH Data-High Rate. Data packet type for high rate data.

Discoverable device A Bluetooth device in range that will respond to an inquiry (normally in addition to responding to page.

Discoverable mode A device that can respond to an inquiry is said to be in a discoverable mode. There are two types of discoverable modes: limited discoverable mode and general discoverable mode. The opposite of discoverable mode is non-discoverable mode. See also silent device.

DLCI Data Link Connection Identifier.

DM Data - Medium Rate. Data packet type for medium rate data.

DSR Data Set Ready. A device sets an RS-232 DSR signal when it is ready to accept data.

DTE Data Terminal Equipment. In serial communications, DTE refers to a device at the endpoint of the communications path; typically a computer or terminal.

DV Data Voice. Data packet type for data and voice.

ETSI European Telecommunications Standards Institute.

FEC Forward Error Correction.

FH Frequency Hopping.

FHS Frequency Hopping Synchronization.

FIFO First In First Out.

FSK Frequency Shift Keying. A type of modulation.

GIAC General Inquire Access Code. See also general discoverable mode.

General discoverable mode A device that can be discovered continuously or for no specific condition is said to be in general discoverable mode. See also discoverable mode.

GM Group Management.

GSM Global System for Mobile communications. GSM is a digital cellular communications technology that is available in Europe and the US. GSM offers multiple services for the subscriber such as short message service.

GW Gateway. A Bluetooth enabled base station, which is connected to external network.

HA Host Application. A software program that uses Bluetooth.

HCI Host Controller Interface.

Headset A microphone and earpiece used to conduct conversations. Headsets can be connected directly to a cellular device or remotely using Bluetooth communications technology.

HEC Header-Error-Check.

Host A software and hardware platform in which the Bluetooth package runs.

Headset (HS) – This is the device acting as the Audio Gateway's remote audio input and output mechanism.

HFCK WIDCOMM, Inc.'s implementation of the HFP.

HAS WIDCOMM, Inc.'s implementation of the HSP.

HV High quality Voice. (e.g., HV1 packet.)

HW Hardware.

IAC Inquiry Access Code.

Idle mode A device is in idle mode when it has no established links to other devices. In this mode, the device may discover other devices. In general, a device sends inquiry codes (GIAC, LIAC) to other devices. Any device that allows inquiries will respond with information. If the devices decide to form a link, then (bonding will occur.

IEEE Institute of Electronic and Electrical Engineering.

IETF Internet Engineering Task Force.

Initiator The Bluetooth device initiating an action to another Bluetooth device. The device receiving the action is called the acceptor. The initiator is typically part of an established link.

Inquiry Procedure The inquiry procedure enables a device to discover which devices are in range, and determine the addresses and clocks for the devices. After the inquiry procedure has completed, a connection can be established using the paging procedure.

Inquiry State A mode that a LocDev enters when searching for services.

Inquiry Scan State A mode that a RemDev enters when advertising that a service is available.

IP Internet Protocol.

IPX Internetwork Packet eXchange. Novell's protocol used by Netware. A router with IPX routing can interconnect LANs so that Novell Netware clients and servers can communicate.

IrDA Infrared Data Association. A method for communication between electronic devices, using 880-nm infrared light.

ISDN Integrated Services Digital Networks.

ISM Industrial, Scientific, Medical.

ITU International Telecommunication Union

Key Management The handling and control of encryption keys.

Known device A device for which at least the BD_ADDR is stored.

L2CA

Logical Link Controller and Adaptation.

L2CAP

Logical Link Controller and Adaptation Protocol.

L_CH Logical Channel.

LAN Local Area Network.

LAP LAN Access Point.

LAP Lower Address Part.

LENA WIDCOMM, Inc.'s implementation of the A2DP.

Link Shorthand for an ACL link.

LC Link Controller. The Link Controller manages the link to the other Bluetooth devices. It is the low-level baseband protocol handler.

LCP Link Controller Protocol.

LFSR Linear Feedback Shift Register.

LIAC Limited Inquiry Access Code. See also limited discoverable mode.

Limited discoverable mode A device that responds to an inquiry for limited purposes. For example, a device may respond for a limited period of time, during temporary conditions, or for a specific event. Typically, the device is responding to a limited inquiry based on an inquiry using the LIAC. See also discoverable mode.

Link key The authentication key used to establish a link between devices. See also bonding.

LM Link Manager. The Link Manager software entity carries out link setup, authentication, link configuration, and other protocols.

LMP Link Manager Protocol. The LMP is used for peer-to-peer communication.

LMP-authentication An LMP level procedure for verifying the identity of a remote device. The procedure is based on a challenge-response mechanism using a random number, a secret key and the BD_ADDR of the non-initiating device. The secret key used can be a previously exchanged link key or an initialization key created based on a PIN (as used when pairing).

LMP-pairing A LMP procedure that authenticates two devices based on a PIN and subsequently creates a common link key that can be used as a basis for a trusted relationship or a (single) secure

connection. The procedure consists of the steps: creation of an initialization key (based on a random number and a PIN), lmp-authentication based on the initialization key and creation of a common link key.

LSB Least Significant Bit.

Master device A device that initiates an action or requests a service on a piconet. See also LocDev.

MS Multiplexing sub layer.

MSB Most Significant Bit.

MSC Modem Status Command.

MTU Maximum Transmission Unit.

N/A Not applicable.

Name Discovery The mechanism to request and receive a device name.

NDIS Network Driver Interface Specification.

New device See unknown device.

Non-connectable mode A device that does not respond to paging (an attempt to establish a communication link) is said to be in non-connectable mode. The opposite of non-connectable mode is connectable mode.

Non-discoverable mode A device that cannot respond to an inquiry is said to be in non-discoverable mode. The device will not enter the INQUIRY_RESPONSE state in this mode. See also discoverable mode.

Non-pairable mode A device that does not accepts pairing is said to be in non-pairable mode. The opposite of non-pairing mode is pairable mode.

OS Operating System.

Page A baseband state where a device transmits page trains and processes any eventual responses to the page trains.

Page Scan State A mode where a device listens for page trains containing its own device access code (DAC). A mode that a RemDev enters when advertising that a service is available.

Page State A mode that a LocDev enters when searching for services. The LocDev sends out a page to notify other devices that it wants to know about the other devices and/or their services.

Page train A series of paging messages sent over the baseband.

Paged device A paged device is typically contacted by a paging device to establish a communication link. See acceptor.

Paging The act of attempting to establish a communication link.

Paging device A paging device is typically attempting to establish a communication link with other devices. See initiator.

Paging Procedure With the paging procedure, an actual connection can be established. The paging procedure typically follows the inquiry procedure. Only the Bluetooth device address is required to set up a connection. Knowledge about the clock will accelerate the setup procedure. A unit that establishes a connection will carry out a page procedure and will automatically be the master of the connection.

Pairable mode A device that accepts pairing is said to be in pairable mode. The opposite of pairing mode is non-pairable mode. paired device A device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase). See also pre-paired device and un-paired device.

Pairing The creation and exchange of a link key between two devices. The devices (LocDev and RemDev) use the link key for future authentication when exchanging information. Pairing is also called an association between a LocDev and a RemDev based on a common link key. The link key is

also referred to as a bond. Pairing can also establish a link by the user entering a PIN, which is authenticated by the device providing the service.

Parked Unit(s) Devices in a piconet, which are synchronized but do not have a MAC addresses.

PC Personal Computer.

PC Card A hardware device that is attached to or installed in a PC to enable the PC to communicate with other Bluetooth devices.

PCM Pulse Coded Modulation.

PCMCIA Personal Computer Memory Card International Association.

PDA Personal Digital Assistant.

PDU Protocol Data Unit. (i.e., a message.)

Phone Services Database The portion of the BT implementation that stores information about device services, both local services and remote services.

Physical channel A synchronized Bluetooth baseband-compliant RF hoping sequence.

Physical link A Baseband level association between two devices established using paging. A physical link comprises a sequence of transmission slots on a physical channel alternating between master and slave transmission slots.

Piconet A collection of devices connected via Bluetooth technology in an ad hoc fashion. A piconet starts with two connected devices, such as a portable PC and cellular phone, and may grow to eight connected devices. All Bluetooth devices are peer units and have identical implementations. However, when establishing a piconet, one unit will act as a master and the other(s) as slave(s) for the duration of the piconet connection. All devices hare the same physical channel defined by the master device parameters (clock and BD_ADDR).

PIN Personal Identification Number. The Bluetooth PIN is used to authenticate two devices that have not previously exchanged link key. By exchanging a PIN, the devices create a trusted relationship. The PIN is used in the pairing procedure to generate the initial link that is used for further identification.

PIN (BB) The PIN used on the baseband level. The PIN (BB) is used by the baseband mechanism for calculating the initialization key during the paring procedure. (128 bits)

PIN (UI) The PIN used on the user interface level. The PIN (UI) is the character representation of the PIN that is entered on the UI level.

PnP Plug and Play.

PPP Point-to-Point Protocol.

Profile A description of the operation of a device or application.

PSM Protocole/Service Multiplexer.

PSTN Public switched telephone network.

QoS Quality of Service.

RAND Random number.

RF Radio Frequency.

RFCOMM Serial Cable Emulation Protocol based on ETSI TS 07.10.

RS-232 A serial communications interface. Serial communication standards are defined by the Electronic Industries Association (EIA).

RTOS Real Time Operating System.

RX Receiver.

SAR Segmentation and Reassembly.

Scatternet Multiple independent and non-synchronized piconets form a scatternet.

SCO Synchronous Connection Oriented link. A synchronous (circuit-switched) connection for reserved bandwidth communications, e.g. voice, between two devices created on the LMP level by reserving slots periodically on a physical channel. This type of link is used primarily to transport SCO packets (voice data). Supports time-bounded information like voice. (Master to single slave.) SCO links can be established only after an ACL link has first been established.

SD Service Discovery.

SDA Service Discovery Application. Also sometimes called the Service Discovery User Application.

SDAP Service Discovery Application Profile.

SDP Service Discovery Protocol.

SDP client The SDP in a Local Device (LocDev). The SDP client requests service information from SDP servers.

SDP server The SDP in a Remote Device (RemDev). The SDP server responds to requests made by SDP clients.

SDP Session The exchange of information between an SDP client and an SDP server. The exchange of information is referred to as an SDP transaction.

SDP Transaction The exchange of an SDP request from an SDP client to an SDP server, and the corresponding SDP response from an SDP server back to the SDP client.

Security Manager The module in a Bluetooth device that controls security aspects of communications to other Bluetooth devices.

Security Mode 1 A device will not initiate any security. A non-secure mode.

Security Mode 2 A device does not initiate security procedures before channel establishment on L2CAP level This mode allows different and flexible access policies for applications, especially running applications with different security requirements in parallel. A service level enforced security mode.

Security Mode 3 A device initiates security procedures before the link setup on LMP level is completed. A link level enforced security mode.

SEQN Sequential Numbering scheme.

SerDscApp Service Discovery Application.

Serial Interface An interface to provide serial communications. service This term refers to a service that one device provides for others. Examples are printers, PIM. Synchronization servers, modems (or modem emulators).

Service Discovery See SDP.

Service Layer The group of protocols that provides services to the application layer and the driver layer in a Bluetooth device.

Service Record Database A database that contains the service discovery-related information.

Service security level Access to services can be denied based on the required service security level. There are three levels of service security: authorization and authentication; authentication only, and no security (open to all). Encryption can be another security requirement for service use in addition to the requirements listed above. Encryption is typically applied at the physical level (baseband). See also device security level.

SIG Special Interest Group.

Silent device A device that is in discoverable mode but cannot respond due to other baseband activity is said to be a silent device. The device could also be in non-discoverable mode and would also not respond to an inquiry.

SIM Subscriber Identity Module. The SIM is a non-volatile storage device that contains information about your phone. This allows the SIM to be used in any GSM phone.

SNK Sink

Slave Unit All devices in a piconet that are not the master.

Smart peripheral See intelligent peripheral.

SME State Machine Engine

SP Service Provider.

SrvDscApp Service Discovery Application.

SRC Source

SSI Signal Strength Indication.

SW Software.

TBD To Be Defined.

TCP Transport Control Protocol.

TCP/IP Transport Control Protocol/Internet Protocol.

TCS Telephone Control protocol Specification.

TCS-AT A set of AT-commands by which a mobile phone and modem can be controlled in the multiple usage models. In BT, AT-commands are based on ITU-T recommendation v.250 and ETS 300 916(GSM 07.07). In addition, the commands used for fax services are specified by the implementation. TCS-AT will also be used for dial-up networking and headset profiles.

TCS Binary Bluetooth Telephony Control protocol Specification using bit-Oriented protocol. It is also referred to as the TCS-BIN system. TCS-BIN will be used for cordless telephony profiles.

TDD Time Division Duplex

TL Terminal.

TLO Terminal Originating a Call.

TLT Terminal terminating a call.

TTP Tiny Transport Protocol between OBEX and UDP [TBD].

TX Transmit.

UA User Asynchronous. Asynchronous user data.

UAP Upper Address Part.

UART Universal Asynchronous Receiver Transmitter. A device which converts parallel data into serial data for transmission, or it converts serial data into parallel data for receiving data.

UDP User Datagram Protocol.

UDP/IP User Datagram Protocol/Internet Protocol.

UI User Interface. The area on a device that contains interface mechanisms such as displays, dialog boxes, manuals, packaging, advertising, etc., where the user is likely to encounter Bluetooth terminology and parameters.

UIAC Unlimited Inquiry Access Code.

Unknown device A device that is currently not connected with the (LocDev and the LocDev has not paired with it in the past. Also called a new device. No information about the device is stored (e.g., BD_ADDR, link key, or other information).

WIDCOMM Bluetooth Stack WIDCOMM, Inc.'s implementation of the Bluetooth protocol and its related profiles to interconnect devices

ANEXE B Test Procedure Results

The goal of the Test Procedure Results is to cover all aspects of the general Headset testing that is required for acceptance by the wireless product vendors, service providers, and Bluetooth certification.

During the application testing, the PC with BTE Insight works as a Headset Device and the Wireless Devices (Cellular Phones, PDA's, Players, etc.) work as Audio Gateways Devices.

The following table shows the obtained results:

System Test Cases						
Title	**Sony Ericsson Z600**	**Nokia 6310i**	**Motorola V625**	**Ericsson T39m**	**IPAQ H5450**	**Comment Pass / Fail**
Playing Music					X	**Pass**
AV- device connection request and Playing Music while the phone is connected (Phone has initiated the connection).	X				X	**Pass**
		X			X	**Pass**
			X		X	**Pass**
Playing Music just after the AV – device has been paired with the headset.					X	**Pass**
AV- device connection request and Playing Music while the phone is connected (HS has initiated the connection)	X				X	**Pass**
		X			X	**Pass**
			X		X	**Pass**
Pairing AV- device and Playing Music while the phone is connected (Phone has initiated the connection).	X				X	**Pass**
		X			X	**Pass**
			X		X	**Pass**
Pairing AV- device and Playing Music while the phone is connected (Headset has initiated the connection).	X				X	**Pass**
		X			X	**Pass**
			X		X	**Pass**
Pairing the phone with the headset while the AV-device is connected.	X				X	**Pass**
		X			X	**Pass**
			X		X	**Pass**
				X	X	**Pass**
Pairing the phone with the headset while the AV-stream is going on.	X				X	**Pass**
		X			X	**Pass** Some under run during pairing process.
			X		X	**Pass**
				X	X	**Pass**
Outgoing calls using dial pad just after the phone has connected for the first time.	X					**Pass**
		X				**Pass**
			X			**Pass**

System Test Cases						
				X		**Pass**
Outgoing calls using dial pad (phone has initiated the connection)	X					**Pass**
		X				**Pass**
			X			**Pass**
Outgoing calls using dial pad (phone isn't connected)				X		**Pass**
Outgoing calls using dial pad (headset has initiated the connection)	X					**Pass**
		X				**Pass**
			X			**Pass**
Outgoing calls using voice dialing (phone has initiated the connection)	X					**Pass**
		X				**Pass**
			X			**Pass**
Outgoing calls using dial pad while the AV- stream is going on. (Phone isn't connected)				X	X	**Pass**
Outgoing calls using dial pad while the AV- stream is going on. (Phone has initiated the connection)	X				X	**Pass** AV-Dev loses connection if the call lasts long time
		X			X	**Pass**
			X		X	**Pass**
Outgoing calls using dial pad while the AV- stream is going on. (Headset has initiated the connection)	X				X	**Pass** AV-Dev loses connection if call lasts long time
		X			X	**Pass**
			X		X	**Pass**
Outgoing calls using voice dialing (Headset has initiated the connection).	X					**Pass**
		X				**Pass**
			X			**Pass**
Outgoing calls using voice dialing when phone isn't connected with the headset	X					**Pass**
		X				**Pass**
			X			**Pass**
				X		**Pass**
Outgoing calls using voice dialing while AV-stream is going on. (Phone has initiated the connection)	X				X	**Pass** AV-Dev loses connection if call lasts long time
		X			X	**Pass**
			X		X	**Pass**
Outgoing calls using voice dialing while AV-stream is going on. (Headset has initiated the connection)	X				X	**Pass** AV-Dev lost connection if call is hold long time.
		X			X	**Pass**
			X		X	**Pass**
Outgoing calls using voice dialing while AV-stream is going on and phone isn't connected with the headset.	X				X	**Pass** AV-Dev loses connection if call lasts long time
		X			X	**Pass**

headset.			X		X	**Pass** SCO is opened two times.
				X	X	**Pass** Under run while phone connection
Outgoing calls while the AV-device is connected (Phone has initiated the connection)	X				X	**Pass** AV-Dev loses connection if call lasts long time
		X			X	**Pass**
			X		X	**Pass**
Outgoing calls while the AV-device is connected (Headset has initiated the connection)	X				X	**Pass** AV-Dev loses connection if call lasts long time.
		X			X	**Pass**
			X		X	**Pass**
Accepting Incoming Calls by the phone. (Phone has initiated the connection)	X					**Pass** SCO isn't open
		X				**Pass**
			X			**Pass**
				X		**Pass** SCO isn't open
Accepting Incoming Calls by the headset. (Phone has initiated the connection)	X					**Pass**
		X				**Pass**
			X		X	**Pass**
				X	X	**Pass**
Accepting Incoming Calls by the phone. (Headset has initiated the connection)	X					**Pass** SCO isn't open
		X				**Pass**
			X			**Pass**
Accepting Incoming Calls by the headset. (Headset has initiated the connection)	X					**Pass**
		X				**Pass**
			X			**Pass**
Accepting Incoming Calls by the phone when the AV- stream is going on (Phone has initiated the connection).	X				X	**Pass** SCO isn't open
		X			X	**Pass**
			X		X	**Pass**
				X	X	**Pass** Call isn't recovered by headset
Accepting Incoming Calls by the headset when the AV- stream is going on (Phone has initiated the connection).	X				X	**Pass** AV-Dev loses connection
		X			X	**Pass**
			X		X	**Pass**
				X	X	**Pass**
Accepting Incoming Calls by the phone when the AV- stream is going on (Headset has initiated the connection)	X				X	**Pass** SCO isn't open
		X			X	**Pass**
			X		X	**Pass**
Accepting Incoming Calls by the headset when the AV- stream is going on (Headset has initiated the connection)	X				X	**Pass** AV-Dev loses connection
		X			X	**Pass**
			X		X	**Pass**
Rejecting Incoming Calls. (Phone	X					**Pass**

System Test Cases						
has initiated the connection)		X				**Pass**
			X			**Pass**
Rejecting Incoming Calls. (Headset has initiated the connection)	X					**Pass**
		X				**Pass**
			X			**Pass**
Rejecting Incoming calls when AV- stream is going on. (Phone has initiated the connection)	X					**Pass**
		X			X	**Pass**
			X		X	**Pass**
Rejecting Incoming calls when AV- stream is going on. (Headset has initiated the connection)	X					**Pass**
		X			X	**Pass**
			X		X	**Pass**
Dialing last number. (Phone has initiated the connection)	X					**Pass**
		X				**Pass**
			X			**Pass**
Dialing last number. (Headset has initiated the connection)	X					**Pass**
		X			X	**Pass**
			X		X	**Pass**
Dialing last number while AV-stream is going on. (Headset has initiated the connection)	X				X	**Pass** AV-Dev loses connection
		X			X	**Pass**
			X		X	**Pass**
Dialing last number while AV-stream is going on and Phone isn't connected.	X				X	**Pass** AV-Dev loses connection
		X				**Pass**
			X		X	**Pass** Some under run during connection
Phone connection request while the AV- stream is going on	X				X	**Pass** Some under run during connection
		X			X	**Pass** Some under run during connection.
			X		X	**Pass** Some under run during connection.
Phone connection request while AV-device is connected	X				X	**Pass**
		X				**Pass**
			X		X	**Pass**
Phone connection request	X					**Pass**
		X				**Pass**
			X			**Pass**
		X			X	**Pass**
			X		X	**Pass**
		X				**Pass**
			X			**Pass**
Turning Off when phone is connected and AV-stream is going on	X				X	**Pass**
		X			X	**Pass**
			X		X	**Pass**