



ZOOM OUT BREW 2008 CONFERENCE

BREW[®], Bluetooth[®] and You

John Larkin, Staff Engineer
Qualcomm Incorporated

QUALCOMM[®]

Terms and Conditions of Usage

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by QUALCOMM Incorporated is under license. Other trademarks and trade names are those of their respective owners.



Introduction

- *Bluetooth* Wireless Technology
 - General information
 - Not tied to BREW
- IBT
 - BREW *Bluetooth* APIs
 - Should be tied to *Bluetooth* technology ☺
- Roadmap
 - Future of *Bluetooth* technology
 - Future of IBT APIs



Bluetooth Wireless Technology

- All about specifications
 - Core specification
 - Defines baseline functionality and features
 - 1.2 -> 2.0 -> 2.1 -> future
 - Comparable to Ethernet + ARP + IP + TCP/UDP
 - Profiles
 - Each profile has its own version
 - Defines application level details
 - Comparable to FTP, HTML, JavaScript, etc.



Bluetooth Technology Applications

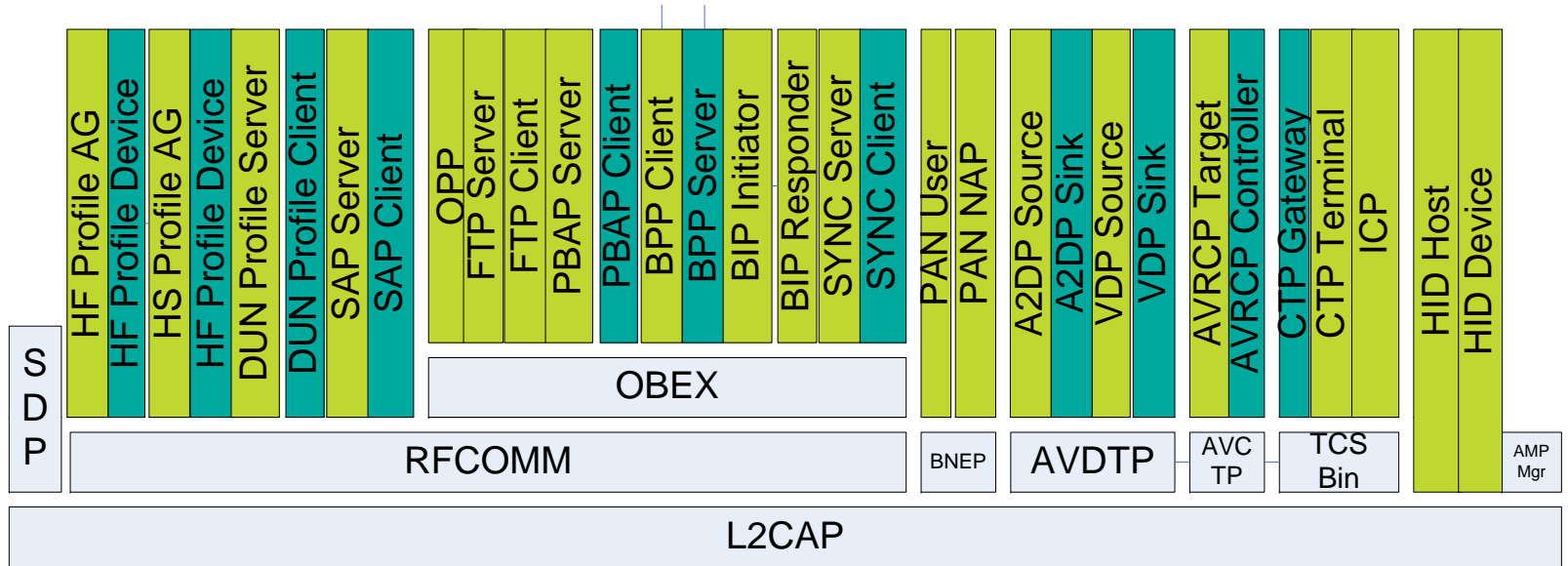
- Applications defined by profiles
 - A more complete profile list to follow
- Headsets & Carkits (HFP, A2DP, AVRCP)
 - Worn by the ~~Borg Cybermen~~ hip technology adopters
 - Mono (voice) + Stereo + Media Player control
- Object Transfer (OPP, FTP, PBAP, BIP, and more!)
 - Phone books; PDA synchronization; Contact exchange; File transfers
- Virtual Serial Port (SPP)
 - Anything serial ports can do (NMEA, ActiveSync, etc)



Bluetooth Management Modes

- Enough info to get you in trouble
 - Background scanning
 - Page Scanning (aka “Connectable”)
 - Impacts standby time
 - Inquiry Scanning (aka “Discoverable”)
 - Impacts standby time, & has security implications
 - User initiated actions
 - Paging (aka “Connecting”)
 - Needed for service connections
 - Currently needed to get the name & list of services
 - Inquiring (aka “Discovering”)
 - Returns list of *Bluetooth* addresses.

Bluetooth Spec Architecture Visual



Protocols

Profiles on Mobile Phones

Profiles on other platform



A Note on Interoperability

- Going back to the HTML example
 - How many TCP/IP stacks on the internet?
 - Anybody remember...
 - DOS & Windows 95?
 - thinnet vs. thicknet? And what's this star topology thing?
 - How many browsers are there?
 - Ever try browsing the web with BREW?
- Wide variety of *Bluetooth* solutions
 - Baseband
 - Host Protocol stack
 - Profile implementations
 - Applications



Bluetooth SIG

- Membership is easy
 - Access to working groups
 - Access to early versions of specifications
 - See <http://www.bluetooth.com> for more info
- Conferences
 - All Hands Meetings
 - UPF (UnPlug Fests)
 - An interoperability festival



Bluetooth Questions

- That's it for the general technology overview.



BREW IBT Design Goals

- Reduce complexity
 - Yet still allow it to be reasonably flexible
- Increase interoperability
 - Many interop problems will be addressed by the class implementations
 - But not all of them!
- Lower “specmanship” requirements
 - Reduce need for intimate knowledge of each specification



BREW IBT Interface Usage

- Users & use cases
 - IBT exposes a lot of *Bluetooth* functionality.
 - Availability for handset manufacturers to develop native UI using IBT.
 - Allows 3rd party applications to use some services
 - 3rd party applications can make a few assumptions
 - The native phone UI probably has the ability to:
 - Discover devices
 - Perform pairing
 - Enable HFP AG functionality
- Read class (.BID file) comments



BREW IBT Interface Usage (continued)

- IBTxxx_Enable()
 - Generally will enable a server, and allow for incoming connections
- IBTxxx_OnEventNotify()
 - Registers a signal which will be set when “something interesting” happens
 - “something interesting” == event
- IBTxxx_GetEvent()
 - Applications call this after an event happens
 - Interface keeps a list of events, and will report them in order to the application



BREW IBT Core Interfaces

- IBT
 - Object factory
 - Retrieve notification data
 - Get radio state
 - Use this to get access to all other classes
- IBT Connection
 - Get/Set status for existing connections
 - Not necessary for many IBT applications
- IBT DevDB
 - Shared device database.
 - Handles bonding/unbonding
 - You'll probably want to use this.



BREW IBT Core Interfaces

- IBT Mgr
 - Get/Set general radio configuration
 - Please don't use this unless you know what you're doing ☺
- IBT ServiceDiscovery
 - Discover nearby devices
 - Get remote device names
 - Get remote device services
 - Set discoverable mode
 - You'll want to use some of these APIs



A2DP

- Advanced Audio Distribution Profile
 - Distributes uni-directional high quality stereo audio
 - Functionally equivalent to the “wire” in a wired headphones.
 - Defines “SRC” and “SNK” roles.
- A2DP is NOT in any way (well, almost any way...)
 - Related to media players
 - Related to controlling audio content, generation or control



IBTA2DP

- Interface functionality
 - Connect / Disconnect
 - Get / Set some configuration parameters
 - Most functionality is contained within the implementation
 - I.e., audio routing, etc.
- Folks who should NOT generally use this API:
 - Media Players
 - Keep using the same CMX/QTV/whatever to play media
 - Sound-device selection apps
 - Anything not controlling low-level sound transport.



IBTA2DP_Notifier

- Handles notifications for A2DP events.
 - Receive events without access to control commands.
 - Connected / Disconnected
 - Start / Suspend
 - Open / Close
- Folks who should use this API
 - Media Players
 - Bluetooth status apps
 - Please follow guidelines in the *Bluetooth* SIG white paper on multi-profile use.
 - Pretty please?



AVRCP

- A/V Remote Control Profile
 - Defines “CT” (Controller) and “TG” (Target) roles
 - Defines (obviously) Remote Control abilities (AVRCP 1.0)
 - Play / Pause / Stop / Next Track / etc / etc...
 - Allows for vendor specific commands
 - Defines (not so obviously) extended Media Player control & status (AVRCP 1.3)
 - Allows CT to retrieve & register for changes in:
 - Media metadata
 - Media player state (playing/stopped, equalizer on/off, etc)
 - Eventually more stuff (AVRCP 1.4)



IBTAVRCPTarget

- Controls *Bluetooth* specific portions of AVRCP
 - Connect / Disconnect
- Currently only one instance allowed on the system
- Folks who should NOT generally use this API
 - Media Players



IRemoteControl_Target

- Works with BTAVRCP to deliver commands
 - Generic interface which may be used by non-*Bluetooth* transports
 - Currently the class implementation supports AVRCP 1.0
- Notable interface functions:
 - IRemoteControl_Target_RegCommandMask
 - IRemoteControl_Target_SetFocus
- Should be used by
 - Media players & other apps desiring remote control
 - Please follow guidelines in the *Bluetooth* SIG white paper on multi-profile use.
 - Pretty please?



HFP / HSP

- HeadSet Profile
 - Defines Hands Free (HF) and Audio Gateway (AG) roles
 - Simple profile
 - So simple, almost no one uses it
- Hands Free Profile
 - Defines Hands Free (HF) and Audio Gateway (AG) roles
 - Includes (too) fine grained detail about call status
 - Allows HF greater control over AG behavior



IBTAudioGateway

- Controls AG role
 - Incoming & outgoing connections
 - Configuration for AG behavior
 - Most functionality is contained within the implementation
 - I.e., audio routing, call handling, etc.
- Intended to be used by a single application
 - There can be only one

SPP

- Serial Port Profile

- Acts like a serial port (as you might expect...)
- Uses RFCOMM protocol

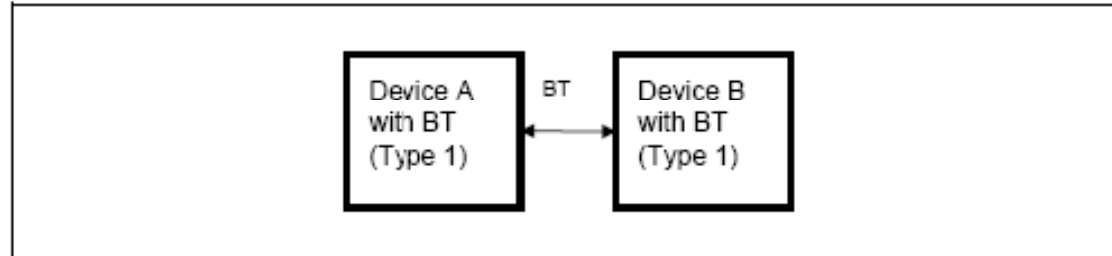


Figure 1.2: RFCOMM Direct Connect

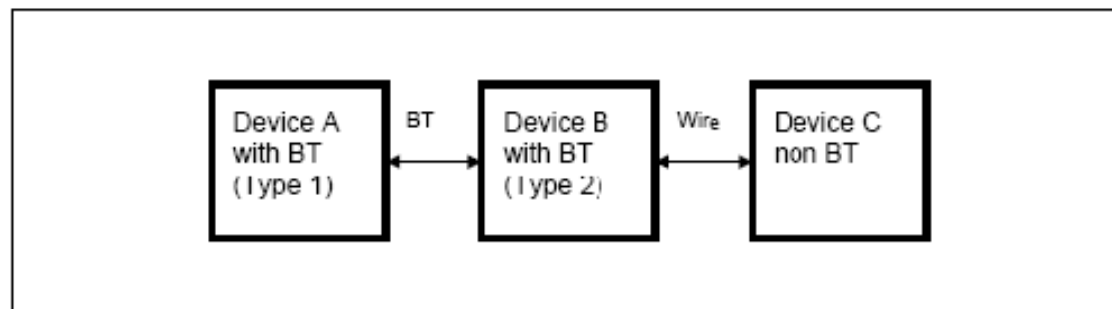


Figure 1.3: RFCOMM used with legacy COM device



IBTSPP

- Soon to be deprecated
 - Uses IPort and AEECallbacks
 - Should use IPort1 and ISignal
- Inherits from IPort
 - Extends IPort to include full serial capability
 - Extends IPort to include Bluetooth specific functionality
- Base interface used by
 - IBTSPPClient
 - IBTSPPServer



IBTSPPClient

- Includes functionality to act as a SPP client
 - Accepts incoming connections from remote servers
- Soon to be deprecated
- Inherits & extends IBTSPP
 - Published before IBTConnection / IBTMgr / etc
 - Includes duplicated functionality



IBTSPPServer

- Includes functionality to act as a SPP server
 - Creates outgoing connections to remote clients
- Soon to be deprecated
- Inherits & extends IBTSPP
 - Published before IBTConnection / IBTMgr / etc
 - Includes duplicated functionality



Questions?

- That's it for the information for *Bluetooth* interface support in BREW 3.1.5 SP01.



Bluetooth SIG Roadmap

- Future core specifications:
 - Alternate MAC/PHY (AMP)
 - High speed physical layers
 - UWB / 802.11
 - Ultra Low Power physical layers
- Future profile specifications:
 - Enhancements to AVRCP (version 1.4)
 - MAP (Message Access Profile)
 - Health Device Profile
 - Lots more you could find out about as a SIG member...
 - Other profiles for the automotive environment



BREW 5.0 Roadmap

- Additional core specification support:
 - *Bluetooth* Core Specification 2.1 + EDR
 - Secure Simple Pairing (SSP)
 - Extended Inquiry Response (EIR)



BREW Mobile Platform (5.0) Roadmap

- Additional profile support:
 - AVRCP 1.3
 - Support for Media Player status and media metadata
 - BPP
 - Basic Printing Profile
 - FTP
 - File Transfer Profile
 - OPP
 - Object Exchange Profile
 - PBAP
 - Phone Book Access Profile
 - BIP
 - Basic Imaging Profile



Devices supporting IBT

- Bluetooth support in BREW
 - First major support was in BREW 3.1.5 SP01
 - Phone OEMs and Carriers have final say on which profiles will be supported
 - A phone supporting a particular profile may not support an IBT interface to that profile



References & Resources

- Bluetooth SIG web site: <http://www.bluetooth.com>
 - All Bluetooth specifications
 - White papers
 - [Simultaneous use of HFP, A2DP, and AVRCP Profiles Whitepaper](#)
- BREW website: <http://brew.qualcomm.com>
 - BREW 3.1.5 SP01
 - Future support for BREW Mobile Platform