Bluetooth Sniffer v. 2.1 + EDR

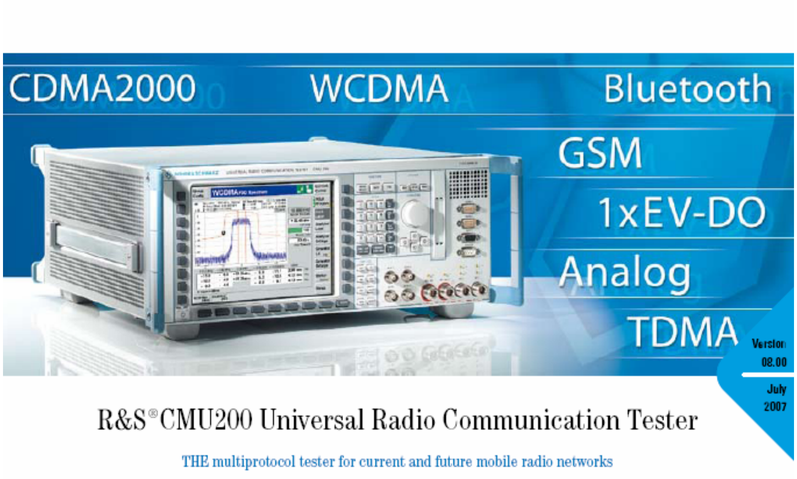
PC based Bluetooth Protocol Analyzer FTS4BT

- FTS for Bluetooth decodes all Bluetooth protocols and supports most Bluetooth profiles.
- Captures, decodes, displays, and filters in real time.
- Data is decoded at the frame, byte, and bit levels, which permits users to rapidly detect and isolate even the most minute and intermittent protocol-related problems.
- Continuous direct logging to disk with unlimited file size for data and speech (WAV file).
- Sniffs in the air and serial HCI simultaneously, with synchronized time stamping.
- Decodes and displays multiple protocol layers of multiple data frames simultaneously.
- Detects and displays protocol errors (in red) in real time.
- Supports data decryption and Simple Pairing with ECDH cryptography per BT version 2.1
- Channel chart with BER measurement in the header and the pay load.
- Packet Timeline chart and Throughput in Bits/sec over time.

ROHDE&SCHWARZ

frontline™

The Bluetooth Sniffer FTS4BT is complementary to the R&S Bluetooth testers:

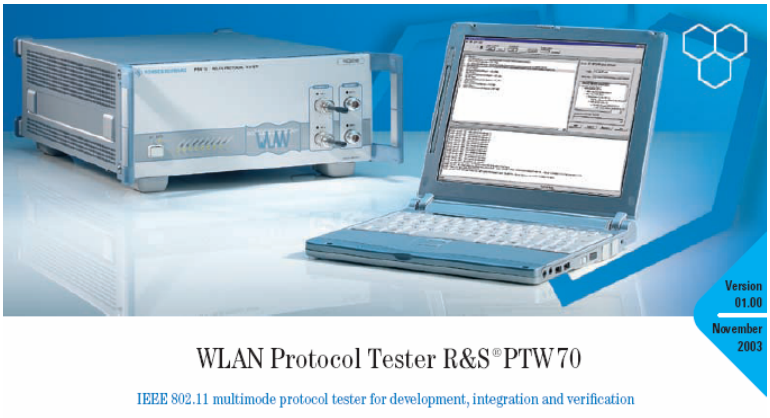


Version 02.00

July 2007

R&S®CMU200 Universal Radio Communication Tester

THE multiprotocol tester for current and future mobile radio networks

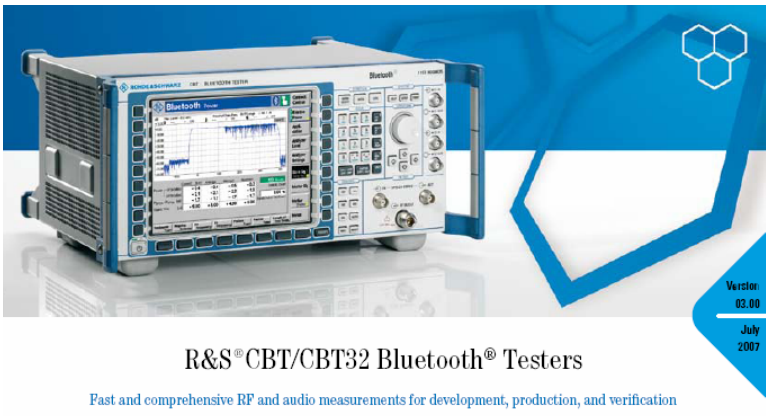


Version 01.00

November 2003

WLAN Protocol Tester R&S®PTW70

IEEE 802.11 multimode protocol tester for development, integration and verification



Version 03.00

July 2007

R&S®CBT/CBT32 Bluetooth® Testers

Fast and comprehensive RF and audio measurements for development, production, and verification

FTS4BT (Frontline Test System for Bluetooth)

The USB dongle was developed in conjunction with CSR (Cambridge Silicon Radio) and was the first protocol analyzer for Bluetooth® version 2.1, complementing the R&S Products like R&S PTW60/70 and R&S CMU200.

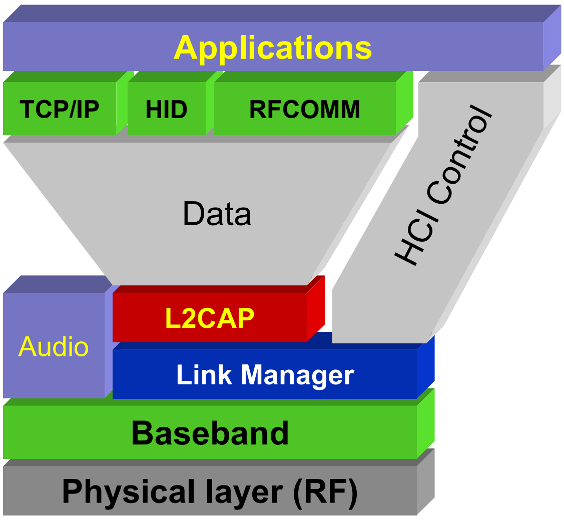
In addition to the measurements of the physical RF parameters by R&S CMU200 or R&S CBT and the programming capability of protocols by the R&S PTW, Frontline is analysing the application protocols and interoperability of Bluetooth® devices.

The Frontline Bluetooth® protocol analyzer, even though it is also decoding the Baseband, is orientated to the software development of the higher layers. Baseband is of minor importance there, all efforts go to the software application around this chip and that‘s the field of main application and strength of Frontline‘s Bluetooth® protocol analyzer.

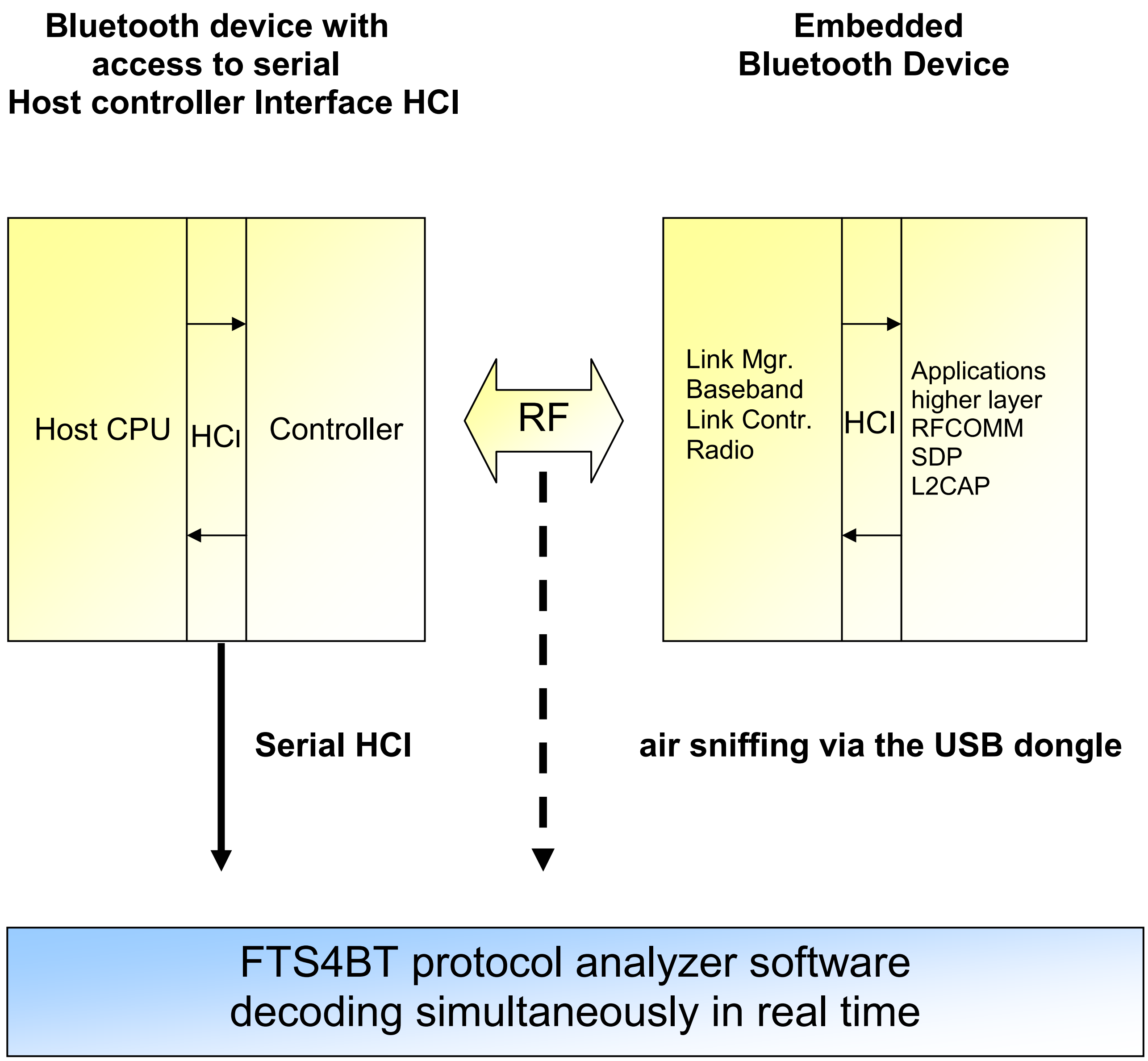
For application debugging, FTS4BT is used as a tool to figure out what is going on with the messaging in real time.

FTS4BT is being used also for verification. In difference to the R&S „Qualification and Compliance“ test, „verification“ is more a functional test of a Bluetooth® device.
Bluetooth® SIG (Special Interest Group) designates specialized organisations as a BQB (Bluetooth® Qualification Body). In order to qualify a Bluetooth® product (this could be a chip, mobile phone, PDA etc.) the BQB will verify the received documentation (they call this evidence) that shows that the product functions and conforms to the Bluetooth® spec.
FTS4BT is used to produce this documentation by capturing the traffic and then prepare the evidence for the BQB.

FTS4BT‘s architecture enables quick release of new decoders as protocols are announced by the Bluetooth SIG. The user can rapidly develop and seamlessly integrate Vendor Extension decoders using the built-in FrameDecoderTM feature.



A standard Bluetooth® device consists of a Bluetooth® controller and a CPU with the Bluetooth® application software. These two elements are connected by the HCI (host controller interface). The controller with the LMP (layer3 link manager protocol) incorporates the Baseband and Physical layer (Bluetooth®-radio). Many Bluetooth® devices give physical access to the HCI (three wires) e. g. via a serial interface.

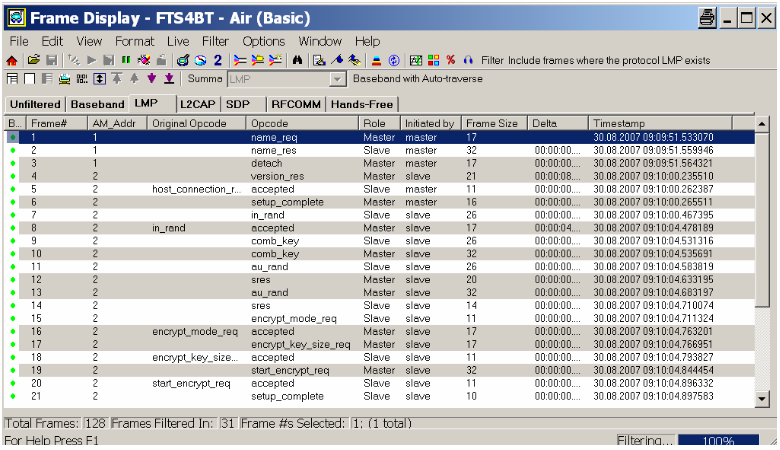


FTS4BT can be connected to the **HCI** of the device via a RS232 port, via USB or virtually to analyze commands and events between these Bluetooth® elements. At this point it can be checked if the application profiles and the serial emulation to the L2CAP (logical link control and adaption protocol) inside of one Bluetooth® device is correct.

In an active Piconet, where at least two Bluetooth® devices (one master, one or more slaves) interact with each other, the USB dongle is **air sniffing** the communication between those. This analysis is required to check interoperability of Bluetooth® devices from different vendors and to troubleshoot problems by detailed protocol decoding.

Pairing

Bluetooth devices on an encrypted link share a common link key in order to exchange encrypted data. This link key is derived from a shared PIN code, the master´s Bluetooth slot clock, the master´s BD_ADDR and a random number that is passed between the two devices. The pairing process is the sequence of events used to create this key, as shown in the Link Manager Protocol below.



Filter Include frames where the protocol LMP exists

Summa LMP

Baseband with Auto-traverse

B...	Frame#	AM_Addr	Original Opcode	Opcode	Role	Initiated by	Frame Size	Delta	Timestamp
●	1	1		name_req	Master	master	17		30.08.2007 09:09:51.533070
●	2	1		name_res	Slave	master	32	00:00:00....	30.08.2007 09:09:51.559946
●	3	1		detach	Master	master	17	00:00:00....	30.08.2007 09:09:51.564321
●	4	2		version_res	Master	slave	21	00:00:08....	30.08.2007 09:10:00.235510
●	5	2	host_connection_r...	accepted	Slave	master	11	00:00:00....	30.08.2007 09:10:00.262387
●	6	2		setup_complete	Master	master	16	00:00:00....	30.08.2007 09:10:00.265511
●	7	2		in_rand	Slave	slave	26	00:00:00....	30.08.2007 09:10:00.467395
●	8	2	in_rand	accepted	Master	slave	17	00:00:04....	30.08.2007 09:10:04.478189
●	9	2		comb_key	Slave	slave	26	00:00:00....	30.08.2007 09:10:04.531316
●	10	2		comb_key	Master	slave	32	00:00:00....	30.08.2007 09:10:04.535691
●	11	2		au_rand	Slave	slave	26	00:00:00....	30.08.2007 09:10:04.583819
●	12	2		sres	Master	slave	20	00:00:00....	30.08.2007 09:10:04.633195
●	13	2		au_rand	Master	slave	32	00:00:00....	30.08.2007 09:10:04.683197
●	14	2		sres	Slave	slave	14	00:00:00....	30.08.2007 09:10:04.710074
●	15	2		encrypt_mode_req	Slave	slave	11	00:00:00....	30.08.2007 09:10:04.711324
●	16	2	encrypt_mode_req	accepted	Master	slave	17	00:00:00....	30.08.2007 09:10:04.763201
●	17	2		encrypt_key_size_req	Master	slave	17	00:00:00....	30.08.2007 09:10:04.766951
●	18	2	encrypt_key_size...	accepted	Slave	slave	11	00:00:00....	30.08.2007 09:10:04.793827
●	19	2		start_encrypt_req	Master	slave	32	00:00:00....	30.08.2007 09:10:04.844454
●	20	2	start_encrypt_req	accepted	Slave	slave	11	00:00:00....	30.08.2007 09:10:04.896332
●	21	2		setup_complete	Slave	slave	10	00:00:00....	30.08.2007 09:10:04.897583

Total Frames: 128 Frames Filtered In: 31 Frame #s Selected: 1; (1 total)

For Help Press F1 Filtering... 100%

Frame 7 is the LMP_in_rand which is where a random number generated by the master is passed to the slave. The slave acknowledges that it has accepted the number in frame 8. In frames 9 and 10, the combination keys are passed between master and slave. In frames 11 an 13 the LMP_au_rand is being exchanged. This is the random number that has been encrypted by using the calculated link key. The response LMP_sres confirms that the same number was computed. When the encryption mode request is accepted, encrypted data can start to be sent.

Simple Pairing in BT version 2.1

The goal of simple pairing is to simplify the pairing process for the user and at the same time ensure the security in Bluetooth wireless technology. The maximum security in version 2.0 + EDR or earlier is achieved when applying the 16 character alphanumeric PIN. The difficulty for an attacker to calculate the secret link key is the same for simple pairing but easier for the user as the protection with simple pairing is independent of the length of the PIN code.

Simple pairing uses Elliptic Curve Diffie-Hellman (ECDH) public key cryptography. ECDH provides a very high degree of strength against passive eavesdropping attacks. Active eavesdropping, also known as man-in-the-middle (MITM) attack, is protected with a 1 in 1 000 000 chance with simple pairing.

Decryption process

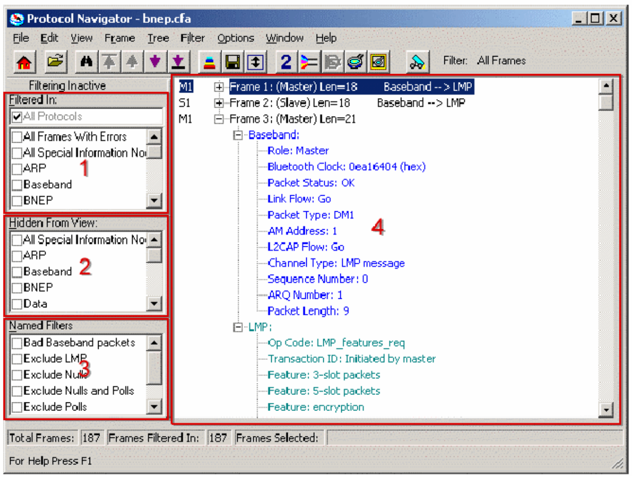
<u>Version 2.0</u>

FTS4BT must compute the same link key being used by the devices being sniffed. Since the link key is never sent over the air, FTS4BT must know the PIN code and capture the LMP_in_rand, both LMP_comb_keys and both LMP_au_rand/LMP_sres pairs. After FTS4BT has calculated the link key, it is capable decrypt data successfully.

<u>Version 2.1</u>

Secure simple pairing messages are captured but FTS4BT can not (yet) compute the ECDH link key from the data in the messages. This means that any encrypted data will not be automatically decrypted.
FTS4BT now can decrypt messages if the link key is provided by the user during the capture process.

Multiple Protocol Layers within multiple data frames



Filtered In:
☑ All Protocols
☐ All Frames With Errors
☐ All Special Information No
☐ ARP
☐ Baseband
☐ BNEP

Hidden From View:
☐ All Special Information No
☐ ARP
☐ Baseband
☐ BNEP
☐ Data

Named Filters
☐ Bad Baseband packets
☐ Exclude LMP
☐ Exclude Nulls
☐ Exclude Nulls and Polls
☐ Exclude Polls

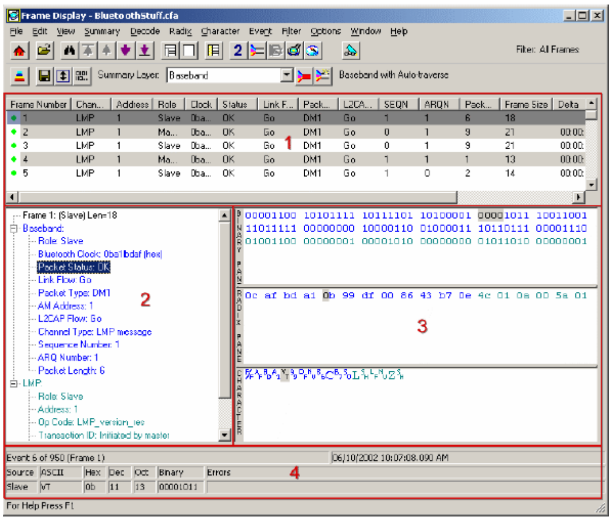
```
M1   ⊞ Frame 1: (Master) Len=18      Baseband --> LMP
S1   ⊞ Frame 2: (Slave) Len=18       Baseband --> LMP
M1   ⊟ Frame 3: (Master) Len=21
        ⊟ Baseband:
             Role: Master
             Bluetooth Clock: 0ea16404 (hex)
             Packet Status: OK
             Link Flow: Go
             Packet Type: DM1
             AM Address: 1
             L2CAP Flow: Go
             Channel Type: LMP message
             Sequence Number: 0
             ARQ Number: 1
             Packet Length: 9
        ⊟ LMP:
             Op Code: LMP_features_req
             Transaction ID: Initiated by master
             Feature: 3-slot packets
             Feature: 5-slot packets
             Feature: encryption
```

Total Frames: 187 Frames Filtered In: 187 Frames Selected:

Protocol Navigator Window: presents decodes of multiple protocol layers within multiple data frames. Simplifies the process of understanding the complex relationships between multiple data frames and the protocol layers that comprise the frames.

1. Protocol Selection Pane: enables you to view only data frames that contain selected Bluetooth protocols.
2. Protocol Suppression Pane: enables you to suppress selected protocols within the Bluetooth data frames.
3. Frame Filter Pane: enables you to select pre-defined filter conditions that include or exclude specific data frame types.
4. Protocol Decode Pane: displays one or more decoded data frames, with text-based explanations of the various protocol elements.

User Data, from the frame level to the bit level


Frame Display - BluetoothStuff.cfa

File Edit View Summary Decode Radix Character Event Filter Options Window Help
Filter: All Frames
Summary Layer: Baseband
Baseband with Auto traverse
1
2
3
4

Frame Number	Chan...	Address	Role	Clock	Status	Link F...	Pack...	L2CA...	SEQN	ARQN	Pack...	Frame Size	Delta
1	LMP	1	Slave	0ba...	OK	Go	DM1	Go	1	1	6	18	
2	LMP	1	Ma...	0ba...	OK	Go	DM1	Go	0	1	9	21	00:00:
3	LMP	1	Slave	0ba...	OK	Go	DM1	Go	0	1	9	21	00:00:
4	LMP	1	Ma...	0ba...	OK	Go	DM1	Go	1	1	1	13	00:00:
5	LMP	1	Slave	0ba...	OK	Go	DM1	Go	1	0	2	14	00:00:

```
Frame 1: (Slave) Len=18
Baseband:
    Role: Slave
    Bluetooth Clock: 0ba1bdaf (hex)
    Packet Status: OK
    Link Flow: Go
    Packet Type: DM1
    AM Address: 1
    L2CAP Flow: Go
    Channel Type: LMP message
    Sequence Number: 1
    ARQ Number: 1
    Packet Length: 6
LMP:
    Role: Slave
    Address: 1
    Op Code: LMP_version_res
    Transaction ID: Initiated by master
```

```
BINARY PANE
00001100 10101111 10111101 10100001 00001011 10011001
11011111 00000000 10000110 01000011 10110111 00001110
01001100 00000001 00001010 00000000 01011010 00000001

RADIX PANE
0c af bd a1 0b 99 df 00 86 43 b7 0e 4c 01 0a 00 5a 01
```

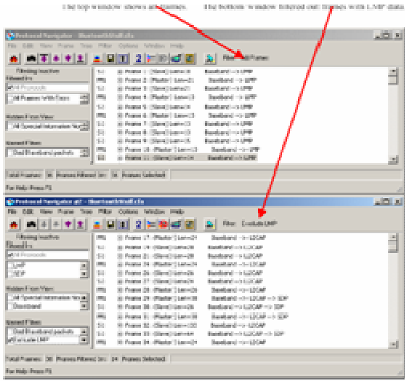
Event 6 of 950 (Frame 1)				06/10/2002 10:07:08.090 AM	
Source	ASCII	Hex	Dec	Oct	Binary	Errors
Slave	VT	0b	11	13	00001011	

For Help Press F1

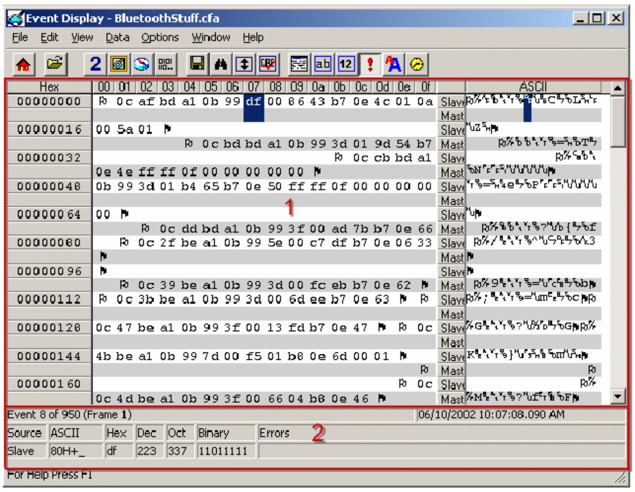
Frame Display Window: presents various views of user data, from the frame level to the bit level, including a detailed decode of a user-selected data frame using a layer-by-layer tree structure. The protocol decode presents text-based descriptions of the protocol-related information resident in each data frame.

1. Summery Pane: displays a high level overview of each Bluetooth data frame, including all significant fields associated with the selected protocol layer.
2. Frame Display Decode Pane: displays a single decoded data frame at levels of detail ranging from the frame level to the bit level.
3. Data Panes: display Bluetooth data in various formats, including binary, hexadecimal, and characters.
4. Selected Data Detail Section: provides protocol-related information pertaining to user-selected portions of the data frame.

Window Duplication: the Duplicate Button creates a second window that is identical to the first. Duplication enables you to view two segments of the data stream simultaneously, or to compare identical data streams with different filter conditions applied.



Data at the byte level


Event Display - BluetoothStuff.cfa

```
Hex       00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f        ASCII
00000000  ⚐ 0c af bd a1 0b 99 df 00 86 43 b7 0e 4c 01 0a  Slave
                                                           Mast
00000016  00 5a 01 ⚑                                        Slave
                   ⚐ 0c bd bd a1 0b 99 3d 01 9d 54 b7      Mast
00000032                                  ⚐ 0c cb bd a1    Slave
          0e 4e ff ff 0f 00 00 00 00 00 ⚑                  Mast
00000048  0b 99 3d 01 b4 65 b7 0e 50 ff ff 0f 00 00 00 00  Slave
                                                           Mast
00000064  00 ⚑                                             Slave
               ⚐ 0c dd bd a1 0b 99 3f 00 ad 7b b7 0e 66    Mast
00000080     ⚐ 0c 2f be a1 0b 99 5e 00 c7 df b7 0e 06 33   Slave
          ⚑                                                Mast
00000096  ⚑                                                Slave
             ⚐ 0c 39 be a1 0b 99 3d 00 fc eb b7 0e 62 ⚑    Mast
00000112  ⚐ 0c 3b be a1 0b 99 3d 00 6d ee b7 0e 63 ⚑ ⚐     Slave
                                                           Mast
00000128  0c 47 be a1 0b 99 3f 00 13 fd b7 0e 47 ⚑ ⚐ 0c    Slave
                                                           Mast
00000144  4b be a1 0b 99 7d 00 f5 01 b8 0e 6d 00 01 ⚑      Slave
                                                  ⚐        Mast
00000160                                       ⚐ 0c        Slave
          0c 4d be a1 0b 99 3f 00 66 04 b8 0e 46 ⚑         Mast
```

Event 8 of 950 (Frame 1)				06/10/2002 10:07:08.090 AM	
Source	ASCII	Hex	Dec	Oct	Binary	Errors
Slave	80H+_	df	223	337	11011111	

For Help Press F1

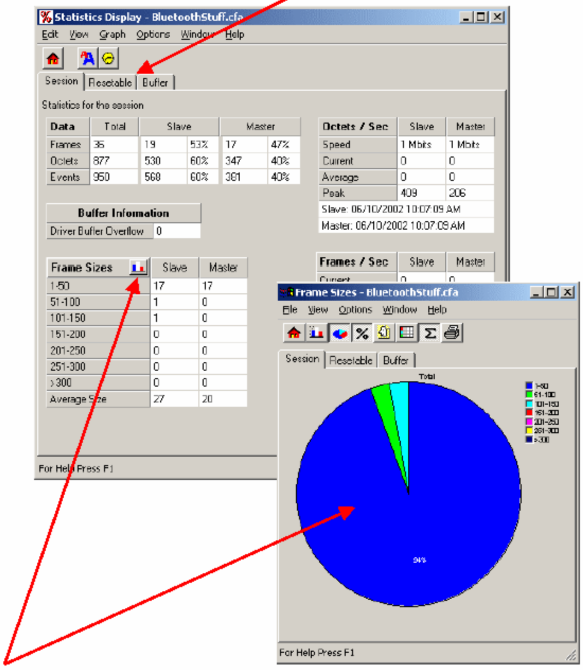
Event Display Window: used to monitor and analyze data at the byte level, and to conduct searches for data patterns.

1. Byte Display Panes: display data flows among Bluetooth devices, in various formats, at the byte level.

2. Selected Data Detail Section: provides protocol-related information pertaining to user-selected, multi-frame portions of the data stream.

Statistical data flow parameters

Analyze traffic statistics in three different modes: for the entire session, from a user-specified point in the session, and for data in a capture buffer.



Statistics Display - BluetoothStuff.cfa

Edit View Graph Options Window Help
Session | Resetable | Buffer

Statistics for the session

Data	Total	Slave		Master	
Frames	35	19	53%	17	47%
Octets	877	530	60%	347	40%
Events	950	568	60%	381	40%

Octets / Sec	Slave	Master
Speed	1 Mbits	1 Mbits
Current	0	0
Average	0	0
Peak	409	206

Slave: 06/10/2002 10:07:09 AM

Master: 06/10/2002 10:07:09 AM

Buffer Information	
Driver Buffer Overflow	0

Frame Sizes	Slave	Master
1-50	17	17
51-100	1	0
101-150	1	0
151-200	0	0
201-250	0	0
251-300	0	0
>300	0	0
Average Size	27	20

Frames / Sec	Slave	Master
Current	0	0

For Help Press F1

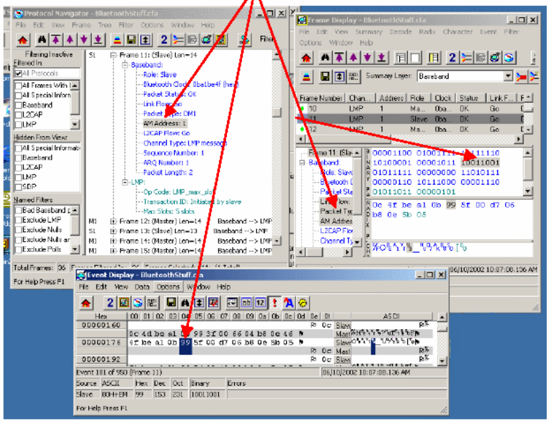
Frame Sizes - BluetoothStuff.cfa

For Help Press F1

Click on the graphics icon to obtain a graphical view of the data.

Statistics Window: provides statistical information on Bluetooth data flow parameters such as number of characters, frames and errors.

Information highlighted on one of the three windows is highlighted on all three.

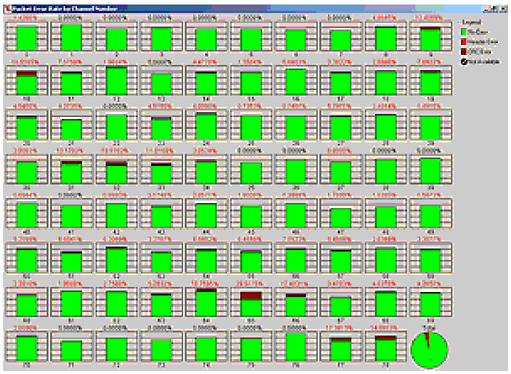


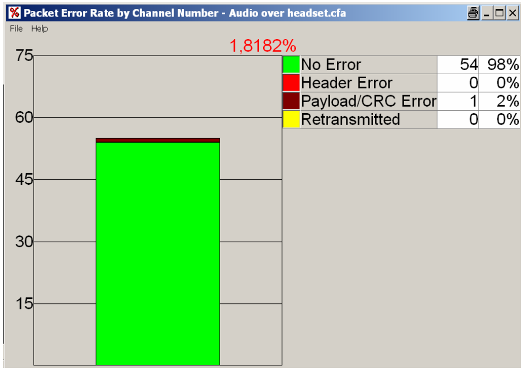
Protocol Navigator - BluetoothStuff.cfa
Frame Display - BluetoothStuff.cfa
Event Display - BluetoothStuff.cfa

Window Synchronization: the Protocol Navigator, Frame display, and Event Display Windows are synchronized. Selecting information in any of these windows highlights the same information in the other two windows

Packet Error Rate Analysis

Graphically viewing the packet error rate on all 79 channels at the same time or on extended slots of interest only, provides immediate insight into baseband performance.



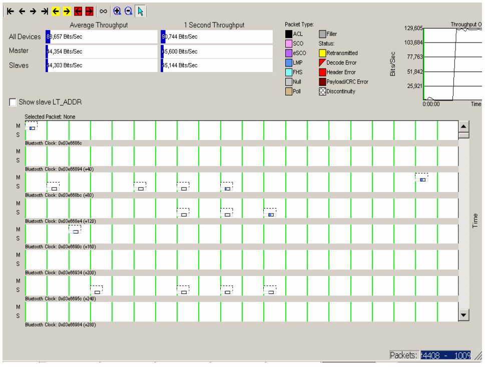


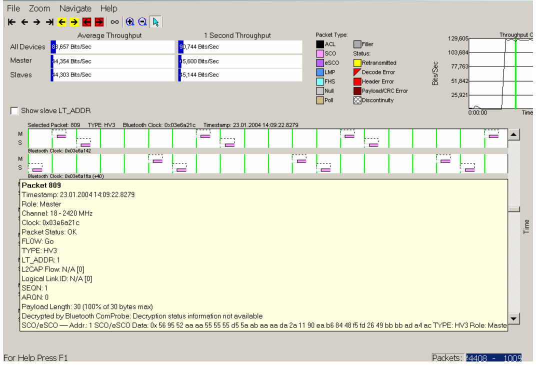
Packet Error Rate by Channel Number - Audio over headset.cfa

🟩	No Error	54	98%
🟥	Header Error	0	0%
🟥	Payload/CRC Error	1	2%
🟨	Retransmitted	0	0%

Packet Timeline chart

View Throughput over time and drill down to packet-level detail, to isolate problems.





	Average Throughput	1 Second Throughput
All Devices	88,657 Bits/Sec	90,744 Bits/Sec
Master	44,354 Bits/Sec	45,600 Bits/Sec
Slaves	44,303 Bits/Sec	45,144 Bits/Sec

☐ Show slave LT_ADDR

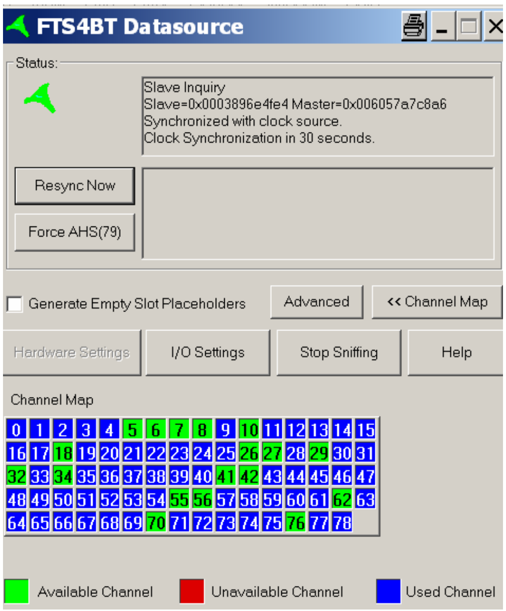
Selected Packet: 809 TYPE: HV3 Bluetooth Clock: 0x03e6a21c Timestamp: 23.01.2004 14:09:22.8279

Bluetooth Clock: 0x03e6a142

Bluetooth Clock: 0x03e6a16a (+40)

Packet 809
Timestamp: 23.01.2004 14:09:22.8279
Role: Master
Channel: 18 - 2420 MHz
Clock: 0x03e6a21c
Packet Status: OK
FLOW: Go
TYPE: HV3
LT_ADDR: 1
L2CAP Flow: N/A [0]
Logical Link ID: N/A [0]
SEQN: 1
ARQN: 0
Payload Length: 30 (100% of 30 bytes max)
Decrypted by Bluetooth ComProbe: Decryption status information not available
SCO/eSCO — Addr.: 1 SCO/eSCO Data: 0x 56 95 52 aa aa 55 55 55 d5 5a ab aa aa da 2a 11 90 ea b6 84 48 f5 fd 26 49 bb bb ad a4 ac TYPE: HV3 Role: Maste

Packets: 4408 - 1009



Status:

Slave Inquiry
Slave=0x0003896e4fe4 Master=0x006057a7c8a6
Synchronized with clock source.
Clock Synchronization in 30 seconds.

☐ Generate Empty Slot Placeholders

Channel Map

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	

🟩 Available Channel 🟥 Unavailable Channel 🟦 Used Channel

Channel Map

From the FTS4BT Datasource window the view of the 79 Bluetooth channels can be selected.
The used channels for the actual Bluetooth Connection are shown in blue; the other available channels are green. In a disturbed environment, caused e.g. by a microwave oven in the neighbourhood or WLAN occupation of certain channels, Bluetooth will avoid those frequencies in the hopping scheme.
Those channels are unavailable and will be indicated in red.

Technical Description

Data Capture Modes

Sniffs data in three modes simultaneously, with synchronized timestamping:

- Through the air via an attached device called the Bluetooth ComProbe, which connects to the USB port on the host PC.
- From the serial HCI interface between a Bluetooth Host CPU and a Bluetooth Host Controller (Bluetooth Device).
- By "virtual sniffing" via the product's Live Import feature, which permits any application to feed data into FTS for Bluetooth via Microsoft's COM interface for interprocess communication.

Piconet Synchronization Methods

- Master Inquiry.
- Slave Inquiry.
- Passive Slave Page.

Data Capture

- Captures, decodes, displays, and filters in realtime.
- Supports full piconet data capture.
- Captures, decrypts, and analyzes encrypted data.
- Supports Pairing, Master/Slave Switching, Park, Hold, and Sniff switches, and Null and Poll capture.
- Captured data includes data bytes and error conditions.
- Data can be captured to RAM or directly to disk.
- Capture buffer sizes are limited only by available memory or disk space.

Data Displays

- Data can be analyzed in realtime using the Event Display, Frame Display, Signal Display, Protocol Navigator, and Statistics Windows.
- Decodes and displays multiple protocol layers of multiple data frames simultaneously.
- Data can be analyzed simultaneously in each of the product's three operating modes: air, serial HCI, and Live Import.
- Multiple synchronized windows can be viewed simultaneously.
- Byte level information is displayed in either a split-line format, or in a mixed format.
- Characters can be displayed in ASCII, EBCDIC, or Baudot.
- Nonprintable characters are displayed using hex and mnemonics.

Error Detection

- Detects and displays protocol errors (in red) in realtime.
- Packet Timeline chart indicating packet type, retransmitted packets, error type and throughput over time.

Search criteria include:

- Byte Level: timestamp, errors, and patterns (patterns include wildcard characters at the bit level, nibble level, and byte level);
- Frame Level: frame decodes can be searched on the text of the decode, with or without wildcards.

Display Filters

- Display Filter parameters include protocol, data pattern, and an advanced mode for complex filters.

Statistics Display

- Provides session totals for characters, frames, events, errors, characters per second, and percentage utilization.

Data Transmission (serial HCI only)

- Data strings and files can be transmitted one time, multiple times, or continuously.
- Data transmission delays can be inserted in millisecond, second, or minute increments.

Timestamping

- Sniffs data in three modes simultaneously, with synchronized timestamping.
- Provides both absolute and relative displays of event timing.
- Calculates time intervals between events (delta time) and data rate.
- Microsecond resolution.

Configuration Management

- I/O configurations, protocol stacks, and filters can be saved for future use.
- Supports user-definable protocol stacks.

Help

- Comprehensive online help provides complete operating instructions.
- Quick Start Guide enables rapid product set-up and operation.

FrameDecoder

- Quickly create custom decoders for proprietary protocols and Vendor Extensions.

Air Sniffing

- Air interface is a small lightweight USB dongle.
- Any Windows PC with a USB port and 64 MB RAM. (Air sniffing is not supported under Windows 95 and NT. Captured data may be viewed under 95 and NT.)

Virtual Sniffing

- Any Windows PC with 64 MB RAM.

Serial HCI Sniffing

- Supports HCI UART (H4) and BCSP.
- Up to 115.2 Kbps using any Windows PC with two serial ports and 64 MB RAM.
- FTS monitoring cables included.
- Up to 921 Kbps with optional high-speed serial card(s).
- Supports board-level (TTL) voltages with optional TTL to RS-232 converters.

Protocols Supported:	**Bluetooth® stack** Baseband, LMP, L2CAP, HCI, UART (H4), HCI, OBEX, TCS Binary, CMTP, HDLC, SDP, RFCOMM, TCS, AVDTP, BNEP, HCRP, HID, AVCTP, AVDTP, OBEX **TCP/IP stack** Decodes all key Dial-Up Networking and TCP/IP protocols IPv4, HTTP, NBNS, UDP, IPv6 DHCPDNS, NBSS, TCP, ICMP, SMB, ARP, NBDS, SMTP **Common LAN Protocols** 802.2, SNAP, IPX, NCP, IBM_NETBIOS, AT commands **Common WAN Protocols** AsyncPPP, PPP
Profiles Supported:	**Audio visual:** A2DP, AVRCP, GAVDP, VCP, VDP **Car:** HFP, PBAP, SAP **Printing:** BIP, BPP, HCRP **Other:** CIP, CTP, DUN, FAX, FTP, GOEP, HID, HSP, ICP, LPP, OPP, PAN, SDAP, SPP, SYNCH, UDI
Bit Order:	LSB (normal) or MSB (reversed) first
Character Sets:	ASCII, 7-bit ASCII, EBCDIC, Baudot
Parity:	None, Even, Odd, Mark, Space
Word Length:	5, 6, 7 or 8 bits
Stop Bits:	1, 1.5 or 2 bits
Timestamping:	Absolute and relative timestamping: microsecond resolution. Timestamps are synchronized over all three operation modes.

Order number:

FTS for Bluetooth with two years warranty and software care:	FTS4BT
One year additional software care:	FTS4BT-PS1
Three years of additional software care:	FTS4BT-PS3
Additional USB-dongle:	ComProbe

Version 2.1 , September 2007