

R&S® SMW-K60

Bluetooth® Enhanced Data Rate

User Manual



1175.6803.02 – 09

This document describes the following software options:

- R&S®SMW-K60
1413.4239.xx

This manual describes firmware version FW 3.20.390.xx and later of the R&S®SMW200A.

© 2015 Rohde & Schwarz GmbH & Co. KG

Mühlhofstr. 15, 81671 München, Germany

Phone: +49 89 41 29 - 0

Fax: +49 89 41 29 12 164

Email: info@rohde-schwarz.com

Internet: www.rohde-schwarz.com

Subject to change – Data without tolerance limits is not binding.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.

The *Bluetooth*® word mark and logos are owned by Bluetooth SIG, Inc. and any use of such marks by Rohde & Schwarz is under license.

Trade names are trademarks of the owners.

The following abbreviations are used throughout this manual: R&S®SMW200A is abbreviated as R&S SMW, R&S®WinIQSIM2™ is abbreviated as R&S WinIQSIM2; the license types 02/03/07/11/13/16/12 are abbreviated as xx.

Contents

1	Preface	5
1.1	Documentation Overview	5
1.2	Conventions Used in the Documentation	6
1.2.1	Typographical Conventions.....	6
1.2.2	Conventions for Procedure Descriptions.....	7
1.2.3	Notes on Screenshots.....	7
2	Welcome to the Bluetooth EDR/Low Energy Digital Standard	8
2.1	Accessing the Bluetooth Dialog	9
2.2	Scope	9
3	About this Digital Standard	10
3.1	Bluetooth Transport Modes	10
3.2	Bluetooth Packet Types	11
3.2.1	ACL packets.....	11
3.2.2	SCO and eSCO packets.....	12
3.2.3	Link control packets for ACL, SCO, eSCO transport modes.....	13
3.3	Packet Structure and Fields	13
3.3.1	Access code.....	14
3.3.2	Header.....	14
3.3.3	Payload format.....	15
3.4	Bluetooth Modulation Schemes	16
4	Bluetooth Configuration and Settings	18
4.1	General Settings	18
4.2	Trigger Settings	21
4.3	Marker Settings	26
4.4	Clock Settings	29
4.5	Local and Global Connector Settings	30
4.6	Bluetooth Channel Settings	31
4.6.1	Bluetooth Basic Rate + EDR.....	31
4.6.2	Packet Configuration - Bluetooth Basic Rate + EDR.....	32
4.6.3	Bluetooth Low Energy.....	37

4.6.4	Event / Frame Configuration - Bluetooth LE.....	41
4.6.4.1	Advertising Event / Frame Configuration Settings.....	43
4.6.4.2	Data Event Connection Settings.....	44
4.6.4.3	Channel Table Settings.....	46
4.6.5	Packet Configuration - Bluetooth LE.....	47
4.6.5.1	Packet Configuration Settings.....	48
4.6.5.2	Test Packet Configuration Settings.....	56
4.7	Dirty Transmitter Test.....	58
4.8	Filter/Clipping Settings.....	61
4.8.1	Filter Settings.....	61
4.8.2	Modulation Settings.....	63
4.8.3	Clipping Settings.....	64
4.9	Power Ramping Settings.....	66
5	Remote-Control Commands.....	68
5.1	General Commands.....	69
5.2	Filter/Clipping Settings.....	74
5.3	Trigger Settings.....	80
5.4	Marker Settings.....	87
5.5	Clock Settings.....	91
5.6	Power Ramping.....	92
5.7	Packet Configuration Setting.....	93
5.8	Dirty Transmitter Test.....	102
5.9	LE Commands.....	107
	List of Commands.....	130
	Index.....	134

1 Preface

1.1 Documentation Overview

The user documentation for the R&S SMW consists of the following parts:

- Getting started, printed manual
- Online help system on the instrument, incl. tutorials
- User manuals and online manual, see the product page
- Service manual, provided on the internet for registered users
- Instrument security procedures, see the product page
- General safety instructions, printed brochure
- Release notes, see the product page (download > firmware)
- Data sheet and brochures, see the product page (download > brochures and data sheets)
- Application notes, provided on the internet



You find the user documentation on the R&S SMW product page mainly at:

<http://www.rohde-schwarz.com/product/SMW200A.html> > "Downloads" > "Manuals"

Additional download paths are stated directly in the following abstracts of the documentation types.

Getting Started

Introduces the R&S SMW and describes how to set up and start working with the product. Includes basic operations, typical measurement examples, and general information, e.g. safety instructions, etc.

Online Help and Tutorials

The **online help** offers quick, context-sensitive access to the information needed for operation and programming. It contains the description for the base unit and the software options.

The **tutorials** offer guided examples and demonstrations on operating the R&S SMW.

User Manual and Online Manual

Separate manuals are provided for the base unit and the software options:

- **Base unit** manual
Contains the description of the graphical user interface, an introduction to remote control, the description of all SCPI remote control commands, programming examples, and information on maintenance, instrument interfaces and error messages. Includes the contents of the getting started manual.
- **Software option** manuals

Describe the specific functions of an option. Basic information on operating the R&S SMW is not included.

The **online manual** provides the contents of the user manual for immediate display on the internet.

Service Manual

Describes the performance test for checking the rated specifications, module replacement and repair, firmware update, troubleshooting and fault elimination, and contains mechanical drawings and spare part lists.

The service manual is available for registered users on the global Rohde & Schwarz information system (GLORIS).

Instrument Security Procedures

Deals with security issues when working with the R&S SMW in secure areas.

Data Sheets and Brochures

The data sheet contains the technical specifications of the R&S SMW. Brochures provide an overview of the instrument and deal with the specific characteristics, see <http://www.rohde-schwarz.com/product/SMW200A.html> > "Download" > "Brochures and Data Sheets".

General Safety Instructions

Contains basic safety instructions in English, Spanish, German and French.

Release Notes

Describes the firmware installation, new and modified features and fixed issues according to the current firmware version. You find the latest version at:

<http://www.rohde-schwarz.com/product/SMW200A.html> > "Downloads" > "Firmware"

Application Notes, Application Cards, White Papers, etc.

These documents deal with special applications or background information on particular topics, see <http://www.rohde-schwarz.com/appnotes>.

1.2 Conventions Used in the Documentation

1.2.1 Typographical Conventions

The following text markers are used throughout this documentation:

Convention	Description
"Graphical user interface elements"	All names of graphical user interface elements on the screen, such as dialog boxes, menus, options, buttons, and softkeys are enclosed by quotation marks.
KEYS	Key names are written in capital letters.
File names, commands, program code	File names, commands, coding samples and screen output are distinguished by their font.
<i>Input</i>	Input to be entered by the user is displayed in italics.
Links	Links that you can click are displayed in blue font.
"References"	References to other parts of the documentation are enclosed by quotation marks.

1.2.2 Conventions for Procedure Descriptions

When describing how to operate the instrument, several alternative methods may be available to perform the same task. In this case, the procedure using the touchscreen is described. Any elements that can be activated by touching can also be clicked using an additionally connected mouse. The alternative procedure using the keys on the instrument or the on-screen keyboard is only described if it deviates from the standard operating procedures.

The term "select" may refer to any of the described methods, i.e. using a finger on the touchscreen, a mouse pointer in the display, or a key on the instrument or on a keyboard.

1.2.3 Notes on Screenshots

When describing the functions of the product, we use sample screenshots. These screenshots are meant to illustrate as much as possible of the provided functions and possible interdependencies between parameters. The shown values may not represent realistic test situations.

The screenshots usually show a fully equipped product, that is: with all options installed. Thus, some functions shown in the screenshots may not be available in your particular product configuration.

2 Welcome to the Bluetooth EDR/Low Energy Digital Standard

The R&S SMW-K60 is a firmware application that adds functionality to generate signals in accordance with the Bluetooth Specification 2.1+EDR. The basic documents for this specification are regulations for Europe, Japan and North America. The frequency band defined for Bluetooth devices is the unlicensed 2.4 GHz ISM (Industrial Scientific medical) frequency band.

The latest Bluetooth Low Energy specification is supported as well.

The following list gives an overview of the R&S SMW-K60 features:

- Support for three transport modes, the ACL+EDR, SCO, eSCO+EDR transport modes
- Support of all packet types for both the Basic Rate and the Enhanced Data Rate modes
- Generation of signals with up to 5238 frames sequence length
- Configuration of the packet contents with a convenient packet editor or all data packets, both with optional data whitening
- Generation of signals in accordance to the Dirty Transmitter Test specification for both the basic and Enhanced Data Rates and with possibilities to change the start phase, the frequency drift rate and the frequency drift deviation
- Power Ramp Control with possibilities to choose ramp time, rise and fall offset
- Configuration of the clipping, filter and modulation settings

The following list gives an overview of the options provided by the R&S SMW for Bluetooth LE compliant signal generation:

- Support for two channel types, the Advertising and Data channel types.
- Support of all Bluetooth LE packet types.
- Sequence Length unit can be set to Event or Frame.
- Convenient packet editor for all supported packet types including optional data whitening.
- Dirty Transmitter Test compliant to RF test specification, with options to change start phase, frequency drift rate and frequency drift deviation.
- Power Ramp Control with configurable ramp time, rise and fall offsets.
- Clipping, filter and modulation settings supported.

This user manual contains a description of the functionality that the application provides, including remote control operation.

All functions not discussed in this manual are the same as in the base unit and are described in the R&S SMW user manual. The latest version is available at the R&S SMW [product page](#) >"Downloads" > "Manuals".

Installation

You can find detailed installation instructions in the delivery of the option or in the R&S SMW Service Manual.

2.1 Accessing the Bluetooth Dialog

To open the dialog with Bluetooth settings

- ▶ In the block diagram of the R&S SMW, select "Baseband > Bluetooth".

A dialog box opens that displays the provided general settings.

The signal generation is not started immediately. To start signal generation with the default settings, select "State > On".

2.2 Scope



Tasks (in manual or remote operation) that are also performed in the base unit in the same way are not described here.

In particular, this includes:

- Managing settings and data lists, like storing and loading settings, creating and accessing data lists, or accessing files in a particular directory.
- Information on regular trigger, marker and clock signals, and filter settings, if appropriate.
- General instrument configuration, such as checking the system configuration, configuring networks and remote operation
- Using the common status registers

For a description of such tasks, see the R&S SMW user manual.

3 About this Digital Standard

The R&S SMW provides you with ability to generate signals in accordance with Bluetooth Specification 2.1+EDR as well as the latest Bluetooth Low Energy specification.

Two modulation modes are used for Bluetooth: a mandatory mode, called the Basic Rate and an optional one, called the Enhanced Data Rate. The Basic Rate mode uses binary FM modulation and has data rate of 1 Mbps. The Enhanced Data Rate uses two types of PSK modulation, the $\pi/4$ -DQPSK or 8DPSK, and achieves data rates of 2 Mbps and 3 Mbps, respectively. All modulation schemes have the symbol rate equal to 1Ms/s.

A Time Division Duplex (TDD) scheme for full duplex transmission is defined for both modes.

The following sections provide some background information on basic terms and principles used in the Bluetooth EDR/Low Energy specifications.

- [Bluetooth Transport Modes](#)..... 10
- [Bluetooth Packet Types](#)..... 11
- [Packet Structure and Fields](#)..... 13
- [Bluetooth Modulation Schemes](#)..... 16

3.1 Bluetooth Transport Modes

There are three different transport modes defined in the Bluetooth specification, each of them with special applications:

- Synchronous Connection-Oriented (SCO)
The SCO transport mode is used for a symmetric point-to-point link establishment between a master and a specific slave in the piconet.
- Extended Synchronous Connection-Oriented (eSCO)
The eSCO transport mode is used for a symmetric or asymmetric, point-to-point link establishment between the master and a specific slave.
- Asynchronous Connection-Less (ACL)
The ACL transport mode is used for a point-to-multipoint link establishment between the master and all slaves participating on the piconet.

There are some common transmitted packets used by all transport modes and some specific packets defined for each transport mode.

3.2 Bluetooth Packet Types

3.2.1 ACL packets

The ACL packets are used for asymmetric links and they contain user data or control data. The table and the figures below give an overview of the ACL packets and their structure.

Table 3-1: ACL packet - Basic Rate

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Slot Number
DM1	1	0-17	2/3	Yes, 16-bit	1
DH1		0-27	no		
DM3	2	0-121	2/3		3
DH3		0-183	no		
DM5		0-224	2/3		
DH5		0-339	no		
AUX1	1	0-29		no	5

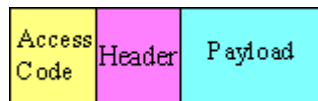


Figure 3-1: Packet Structure of ACL packets - Basic Rate

Table 3-2: ACL packets - Enhanced Rate

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Slot Number
2-DH1	2	0-54	no	Yes, 16-bit	1
2-DH3		0-367			3
2-DH5		0-679			5
3-DH1		0-83			1
3-DH3		0-552			3
2-DH5		0-1021			5



Figure 3-2: Packet Structure of ACL packets - Enhanced Rate

3.2.2 SCO and eSCO packets

The SCO and eSCO packets are used for symmetric links. The SCO packets are used for 64 kb/s speech transmission and for transparent synchronous data. The eSCO packets are also used for 64kb/s speech transmission and transparent data at 64 kb/s but also at other rates.

The tables and the figures below give an overview of the SCO and eSCO packets and their structure.

Table 3-3: SCO packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Slot Number
HV1	n.a.	10	1/3	no	n.a.
HV2		20	2/3		
HV3		30			
DV	1 (Data only)	10+(0-9)	2/3 (Data only)	Yes, 16-bit (Data only)	

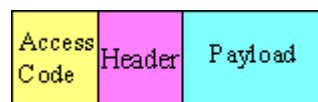


Figure 3-3: Packet Structure SCO packets

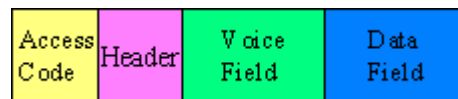


Figure 3-4: Packet Structure SCO packets (data only)

Table 3-4: eSCO packets - Basic Rate

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Slot Number
EV3	n.a.	1-30	no	Yes, 16-bit (Data only)	1
EV4		1-120	2/3		3
EV5		1-180	no		3

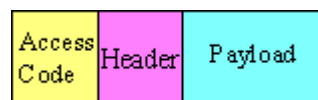


Figure 3-5: Packet Structure eSCO packets - Basic Rate

Table 3-5: eSCO packets - Basic Rate

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Slot Number
2-EV3	n.a.	1-60	no	Yes, 16-bit	1
2-EV5		1-360			3
3-EV3		1-90			1
3-EV5		1-540			3



Figure 3-6: Packet Structure eSCO packets - Enhanced Rate

3.2.3 Link control packets for ACL, SCO, eSCO transport modes

There are some common kinds of packet types. An overview of these packet types is given in the table below.

Table 3-6: Common link control packets

Transport modes	Type	Payload Header (bytes)	FEC	CRC	Application
SCO,eSCO,ACL	ID	n.a.	n.a.	n.a.	Paging, inquiry, response
SCO,eSCO,ACL	NULL				Carries Link information to the source, e.g. about successfully received signal (ARQN) or the state of the receiving buffer (FLOW)
SCO,eSCO,ACL	POLL				Similar to NULL packet, used by master to poll the slaves, must be confirmed
SCO,ACL	FHS	18	2/3	Yes	Page master response, inquiry response, in roll switch

Table 3-7: Common link control packets: packet structure

Packet Type ID	Packet Types NULL and POLL	Packet Types FHS
Access Code (DAK or IAC)	Access Code Header	Access Code Header Payload

3.3 Packet Structure and Fields

Allmost all Bluetooth transmitted packets have standard format and consist of the access code, the header and the payload with useful information. The exceptions are

the ID packet which consists of the access code only and NULL and POLL packets which carry only the access code and the header.

3.3.1 Access code

The access code is used for synchronisation, DC offset compensation and identification. The fields of the access code are shown in the figure below and their meaning is explained in the table below.

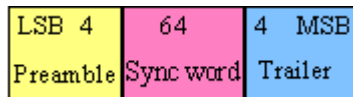


Table 3-8: The access code fields

Field	Description	Packets
Preamble	A fixed zero-one pattern of 4 symbols, used to facilitate DC compensation	All packets
Sync Word	A 64-bit code word derived from a 24 bit address, improves timing acquisition	All packets
Trailer	A fixed zero-one pattern of four symbols, extended DC compensation	All packets, except ID

3.3.2 Header

The Header contains link control information. The fields of the header are shown in the figure and their meaning is explained in the table below.

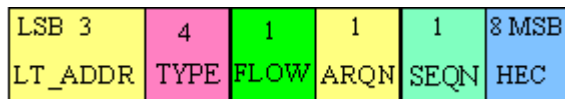


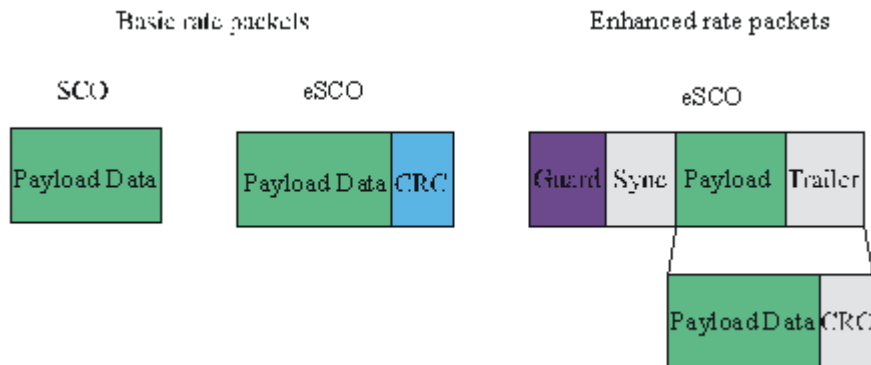
Table 3-9: The header fields

Field	Description	Packets
LT_ADDR	Logical transport address, indicates the destination slave for a packet in a master-to-slave transmission slot and the source slave for a slave-to-master transmission slot	
TYPE	Type code, specifies which packet type is used	
FLOW	Flow control, used for flow control of packets over the ACL logical transport. When the RX buffer in the recipient is full, a STOP indication shall be returned. When the RX buffer can accept data, a Go indication shall be returned.	All packets, except ID
ARQN	Automatic Repeat Request Number, acknowledgement indication, used to inform the source of a successful transfer of payload data with CRC can be positive acknowledged ACK or negative acknowledged NAK,	
SEQN	Sequential numbering scheme to order the data packet stream	
HEC	Header-error-check to check the header integrity	

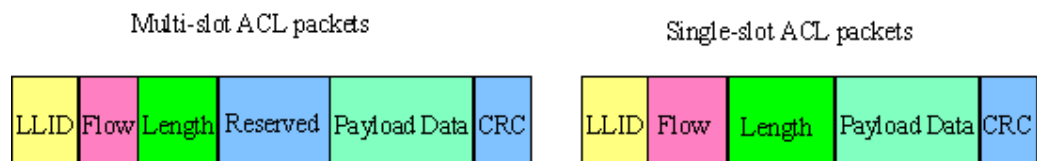
3.3.3 Payload format

The payload structure depends on the type of the data field and the data rate. Two fields are defined in the payload: the synchronous data field and the asynchronous data field. The ACL packets only have the asynchronous data field and the SCO and eSCO packets only have the synchronous data field. The exception is DV of SCO transport mode which has both data fields, synchronous and asynchronous.

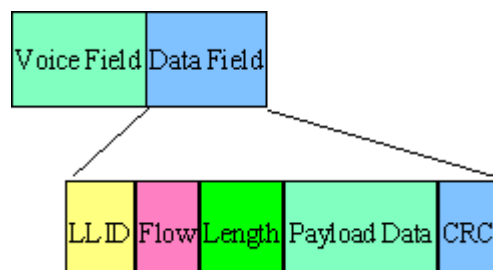
Synchronous data fields



Asynchronous data fields



Synchronous and Asynchronous data fields



The meaning of some payload fields is given in the table below.

Table 3-10: The payload fields

Field	Description
CRC	The cyclic redundancy error check
Guard, Sync	The guard time and synchronization sequence, used for physical layer change of modulation scheme
LLID	The logical link identifier, specifies the logical link
Flow	Field which controls the flow on the logical channels

The payload format and content of the FHS packet are different from other packets. The fields of the FHS packet are shown in the figure below and their meaning is explained in the table below.

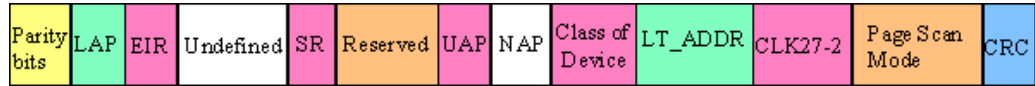


Table 3-11: The payload fields for the FHS packet

Field	Description
Parity bits	Form the first part of the sync word of the access code of the device that sends the FHS packet
LAP	Contains the lower address part of the device that sends the FHS packet
EIR	An extended inquiry response, provides miscellaneous information during the inquiry response procedure
Undefined	Reserved for future use and shall be set to zero
SR	The scan repetition field, indicates the interval between two consecutive page scan windows
Reserved	Shall be set to 10
UAP	Contains the upper address part of the device that sends the FHS packet
NAP	Contains the non-significant address part of the device that sends the FHS packet
Class of device	Contains the class of device of the device that sends the FHS packet. This field is defined in Bluetooth Assigned Numbers.
LT_ADDR	Contains the logical transport address
CLK27-2	Contains the value of the native clock of the device that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet
Page scan mode	Indicates which scan mode is used by default by the sender of the FHS packet

3.4 Bluetooth Modulation Schemes

The modulation used for the basic data rate packets is GFSK (Gaussian Frequency Shift Keying) with a bandwidth bit period product $BT=0.5$. The modulation index is between 0.28 and 0.35.

The modulation scheme used for enhanced data rate packets changes within the packet. The access code and packet header has GFSK modulation scheme and are transmitted with the Basic Rate 1Mbps, while the subsequent synchronisation sequence, payload and trailer sequence have a PSK type of modulation and are transmitted with a data rate of 2 Mbps or optionally 3 Mbps.

The PSK modulation, namely $\pi/4$ rotated differential encoded quaternary phase shift keying ($\pi/4$ -DQPSK) is defined for for the 2 Mbps transmission.

The PSK modulation, namely differential encoded 8-ary phase shift keying (8DPSK), is defined for the 3Mbps transmission.

The modulation types and corresponding packet types are given in the table below.

Table 3-12: The modulation types and corresponding packet types

Modulation type	Packet types
GFSK	ID, NULL, POLL, FHS, DM1, DH1, DM3, DH3, DM5, DH5, AUX1, HV1, HV2, HV3, DV, EV3, EV4, EV5
GFSK + $\pi/4$ -DQPSK	2-DH1, 2-DH3, 2-DH5, 2-EV3, 2-EV5
GFSK + 8DPSK	3-DH1, 3-DH3, 3-DH5, 3-EV3, 3-EV5

4 Bluetooth Configuration and Settings

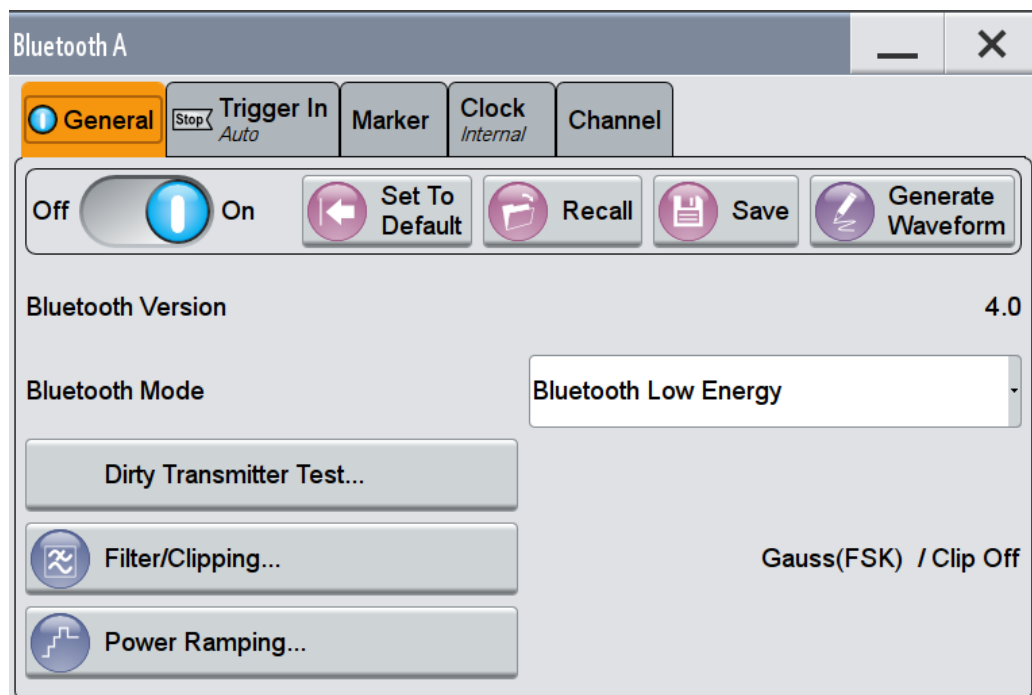
- ▶ To access the Bluetooth settings, select "Baseband > Bluetooth".

The remote commands required to define these settings are described in [Chapter 5, "Remote-Control Commands"](#), on page 68.

• General Settings	18
• Trigger Settings	21
• Marker Settings	26
• Clock Settings	29
• Local and Global Connector Settings	30
• Bluetooth Channel Settings	31
• Dirty Transmitter Test	58
• Filter/Clipping Settings	61
• Power Ramping Settings	66

4.1 General Settings

The tab provides access to the default and the "Save/Recall" settings. The selected bluetooth mode and transport mode determine the available parameters.



State	19
Set To Default	19
Save/Recall	19
Generate Waveform	20
Bluetooth Version	20

Bluetooth Mode.....	20
Transport Mode.....	20
Dirty Transmitter Test.....	21
Filter / Clipping.....	21
Power Ramping.....	21

State

Activates the standard and deactivates all the other digital standards and digital modulation modes in the same path.

Remote command:

[:SOURce<hw>] :BB:BT0oth:STATe on page 72

Set To Default

Calls the default settings. The values of the main parameters are listed in the following table.

Parameter	Value
State	Not affected by "Set to default"
Bluetooth Version	2.1 + EDR (2.1 = current version number)
Bluetooth Mode	Basic Rate + EDR
Transport mode	ACL (Asynchronous) + EDR
Packet type	DH1
Sequence length	1 Frames
Slot Timing	Tx Test Mode
Packet configuration	Packet Editor/ Whitening off
Dirty Transmitter Test	Not in Use
Filter	Gauss (FSK)
Clipping	Clipping off
Power Ramping	Cosine / 1 Symbols
Trigger	Auto
Marker	Restart
Clock	Internal

Remote command:

[:SOURce<hw>] :BB:BT0oth:PRESet on page 70

Save/Recall

Accesses the "Save/Recall" dialog, that is the standard instrument function for storing and recalling the complete dialog-related settings in a file. The provided navigation possibilities in the dialog are self-explanatory.

The file name and the directory, in which the settings are stored, are user-definable; the file extension is however predefined.

See also, chapter "File and Data Management" in the R&S SMW user manual.

Remote command:

`[:SOURce<hw>] :BB:BT00th:SETTing:CATalog` on page 71

`[:SOURce<hw>] :BB:BT00th:SETTing:LOAD` on page 71

`[:SOURce<hw>] :BB:BT00th:SETTing:STORe` on page 72

`[:SOURce<hw>] :BB:BT00th:SETTing:DELeTe` on page 71

Generate Waveform

With enabled signal generation, triggers the instrument to store the current settings as an ARB signal in a waveform file. Waveform files can be further processed by the ARB and/or as a multi-carrier or a multi-segment signal.

The file name and the directory it is stored in are user-definable; the predefined file extension for waveform files is `*.wv`.

Remote command:

`[:SOURce<hw>] :BB:BT00th:WAVEform:CREate` on page 74

Bluetooth Version

Displays the current version of the standard.

The default settings and parameters provided are oriented towards the specifications of the version displayed.

Remote command:

`[:SOURce<hw>] :BB:BT00th:VERSion?` on page 73

Bluetooth Mode

Determines the digital Bluetooth standard. Basic Rate + EDR or Bluetooth Low Energy are available.

"Basic Rate +EDR"

Selects the Bluetooth mode Basic Rate + EDR.

Specific settings of the basic mode are described in [Chapter 4.6.1, "Bluetooth Basic Rate + EDR"](#), on page 31.

"Bluetooth Low Energy"

Selects the Bluetooth mode Low Energy. The settings concerning Bluetooth Low Energy mode are described in [Chapter 4.6.3, "Bluetooth Low Energy"](#), on page 37.

Remote command:

`[:SOURce<hw>] :BB:BT00th:BMODE` on page 109

Transport Mode

Only available for "Bluetooth Mode " set to "Basic Rate + EDR"

Selects the transport mode.

"ACL+EDR" The transport mode selected is used for a point-to-multipoint link establishment between the master and all the slaves participating on the piconet.

"SCO" The transport mode selected is used for a point-to-point link establishment between a master and a single slave in the piconet.

"eSCO+EDR" The transport mode selected is used for a symmetric or asymmetric point-to-point link establishment between a master and a specific slave.

Remote command:

[:SOURce<hw>] :BB:BT0oth:TMODe on page 73

Dirty Transmitter Test

Access to the "Dirty Transmitter Test" dialog, see [Chapter 4.7, "Dirty Transmitter Test"](#), on page 58.

Filter / Clipping

Access to the dialog for setting baseband filtering, the modulation settings and clipping, see [Chapter 4.8, "Filter/Clipping Settings"](#), on page 61.

Power Ramping

Access to the "Power Ramp Control" dialog, see [Chapter 4.9, "Power Ramping Settings"](#), on page 66.

4.2 Trigger Settings

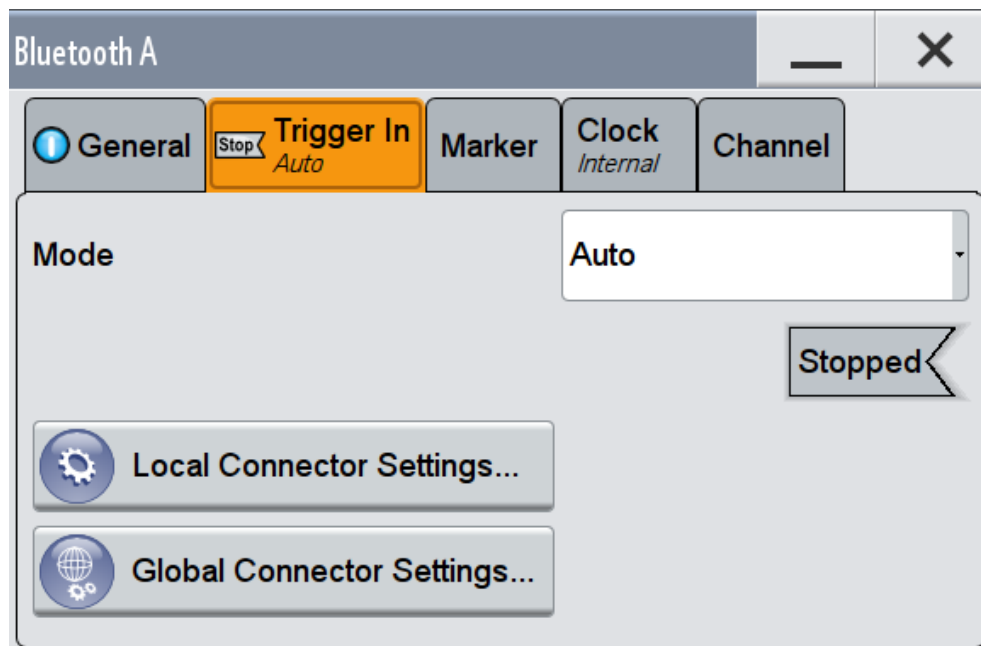
This tab provides an access to the settings necessary to select and configure the trigger, like trigger source, mode, trigger delay, trigger suppression, as well as to arm or trigger an internal trigger manually. The current signal generation status is displayed in the header of the tab together with information on the enabled trigger mode. As in the "Marker" and "Clock" tabs, this tab provides also an access to the settings of the related connectors.



This section focuses on the available settings.

For information on how these settings affect the signal, refer to section "Basics on ..." in the R&S SMW user manual.

- ▶ To access this dialog, select "Baseband > Bluetooth > Trigger In".



This dialog comprises the settings required for configuring the trigger.



Routing and Enabling a Trigger

The provided trigger signals are not dedicated to a particular connector but can be mapped to one or more globally shared USER or local T/M/(C) connectors.

Use the [Local and Global Connector Settings](#) to configure the signal mapping as well as the polarity, the trigger threshold and the input impedance of the input connectors.

To route and enable a trigger signal, perform the following *general steps*:

- Define the signal source and the effect of a trigger event, i.e. select the "Trigger In > Mode" and "Trigger In > Source"
- Define the connector, USER or T/M/(C), the selected signal is provided at, i.e. configure the [Local and Global Connector Settings](#).

Provided are the following settings:

Trigger Settings Common to All Basebands

To enable simultaneous signal generation in all basebands, the R&S SMW couples the trigger settings in the available basebands in any instrument's configuration involving signal routing with signal addition (e.g. MIMO configuration, routing and summing of basebands and/or streams).

The icon indicates that common trigger settings are applied.

You can access and configure the common trigger source and trigger mode settings in any of the basebands. An arm or a restart trigger event applies to all basebands, too. You can still apply different delay to each of the triggers individually.

Trigger Mode ← Trigger Settings Common to All Basebands

Selects trigger mode, i.e. determines the effect of a trigger event on the signal generation.

For more information, refer to chapter "Basics" in the R&S SMW user manual.

- "Auto"
The signal is generated continuously.
- "Retrigger"
The signal is generated continuously. A trigger event (internal or external) causes a restart.
- "Armed_Auto"
The signal is generated only when a trigger event occurs. Then the signal is generated continuously.
An "Arm" stops the signal generation. A subsequent trigger event (internal with or external) causes a restart.
- "Armed_Retrigger"
The signal is generated only when a trigger event occurs. Then the signal is generated continuously. Every subsequent trigger event causes a restart.
An "Arm" stops signal generation. A subsequent trigger event (internal with or external) causes a restart.
- "Single"
The signal is generated only when a trigger event occurs. Then the signal is generated once to the length specified at "Signal Duration".
Every subsequent trigger event (internal or external) causes a restart.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth [:TRIGger] :SEQuence` on page 86

Signal Duration Unit ← Trigger Settings Common to All Basebands

Defines the unit for describing the length of the signal sequence to be output in the "Single" trigger mode.

- | | |
|-------------------|--|
| "Sequence Length" | The selected unit for the entry of the length of the signal sequence at the output in the Single trigger mode is sequence length . |
| "Frames" | The selected unit for the entry of the length of the signal sequence at the output in the Single trigger mode is frame. |

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:TRIGger:SLUNit` on page 84

Trigger Signal Duration ← Trigger Settings Common to All Basebands

Enters the length of the signal sequence to be output in the "Single" trigger mode.

Use this parameter to deliberately output part of the signal, an exact sequence of the signal, or a defined number of repetitions of the signal.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:TRIGger:SLENgth` on page 83

Running/Stopped ← Trigger Settings Common to All Basebands

For enabled modulation, displays the status of signal generation for all trigger modes.

- "Running"

The signal is generated; a trigger was (internally or externally) initiated in triggered mode.

- "Stopped"

The signal is not generated and the instrument waits for a trigger event.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:RMODe` on page 83

Arm ← Trigger Settings Common to All Basebands

Stops the signal generation until subsequent trigger event occurs.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:ARM:EXECute` on page 81

Execute Trigger ← Trigger Settings Common to All Basebands

For internal trigger source, executes trigger manually.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:EXECute` on page 81

Trigger Source ← Trigger Settings Common to All Basebands

The following sources of the trigger signal are available:

- "Internal"
The trigger event is executed manually by the "Execute Trigger".
- "Internal (Baseband A/B)"
The trigger event is provided by the trigger signal from the other basebands. If common trigger settings are applied, this trigger source is disabled.
- "External Global Trigger 1 / 2"
The trigger event is the active edge of an external trigger signal provided and configured at the global USER connectors.
- "External Global Clock 1 / 2"
The trigger event is the active edge of an external global clock signal provided and configured at the global USER connectors.
- "External Local Trigger"
The trigger event is the active edge of an external trigger signal provided and configured at the local T/M/(C) connector. With coupled trigger settings, the signal has to be provided at the T/M/C 1/2/3 connectors.
- "External Local Clock"
The trigger event is the active edge of an external local clock signal provided and configured at the local T/M/C connector. With coupled trigger settings, the signal has to be provided at the T/M/C 1 connector.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:SOURce` on page 84

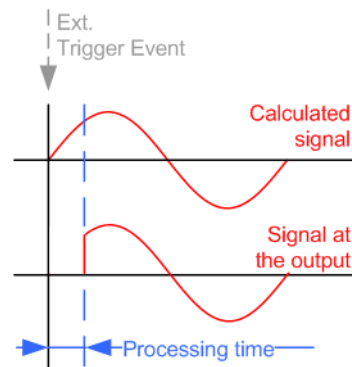
Sync. Output to External Trigger ← Trigger Settings Common to All Basebands

For an external trigger signal, enables/disables the output of a signal synchronous to the external trigger event.

"On"

Corresponds to the default state of this parameter.

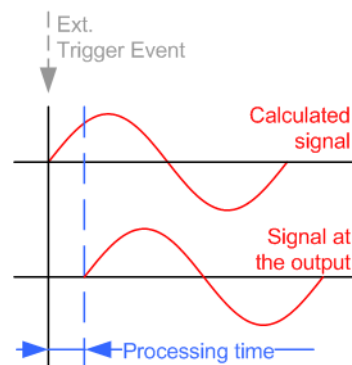
The signal calculation starts simultaneously with the external trigger event but because of the instrument's processing time the first samples are cut off and no signal is output. After elapsing of the internal processing time, the output signal is synchronous to the trigger event.



"Off"

The signal output begins after elapsing of the processing time and starts with sample 0, i.e. the complete signal is output.

This mode is recommended for triggering of short signal sequences with signal duration comparable with the processing time of the instrument.



Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger [:EXTernal] :SYNChronize:OUTPut`
on page 81

External Trigger Inhibit ← Trigger Settings Common to All Basebands

For external trigger signal or trigger signal from the other path, sets the duration with that any following trigger event is suppressed. In "Retrigger" mode for example, a new trigger event does not cause a restart of the signal generation until the specified inhibit duration does not expire.

For more information, see chapter "Basics" in the R&S SMW user manual.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger [:EXTernal] :INHibit` on page 86
`[:SOURce<hw>] :BB:BT0oth:TRIGger:OBASeband:INHibit` on page 82

Trigger Delay

Delays the trigger event of the signal from:

- The external trigger source
- The other path
- The other basebands (internal trigger), if common trigger settings are used.

Use this setting to:

- Synchronize the instrument with the device under test (DUT) or other external devices
- Postpone the signal generation start in the basebands compared to each other

For more information, see chapter "Basics on ..." in the R&S SMW user manual.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger [:EXTernal] :DELay` on page 85

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OBASeband:DELay` on page 82

4.3 Marker Settings

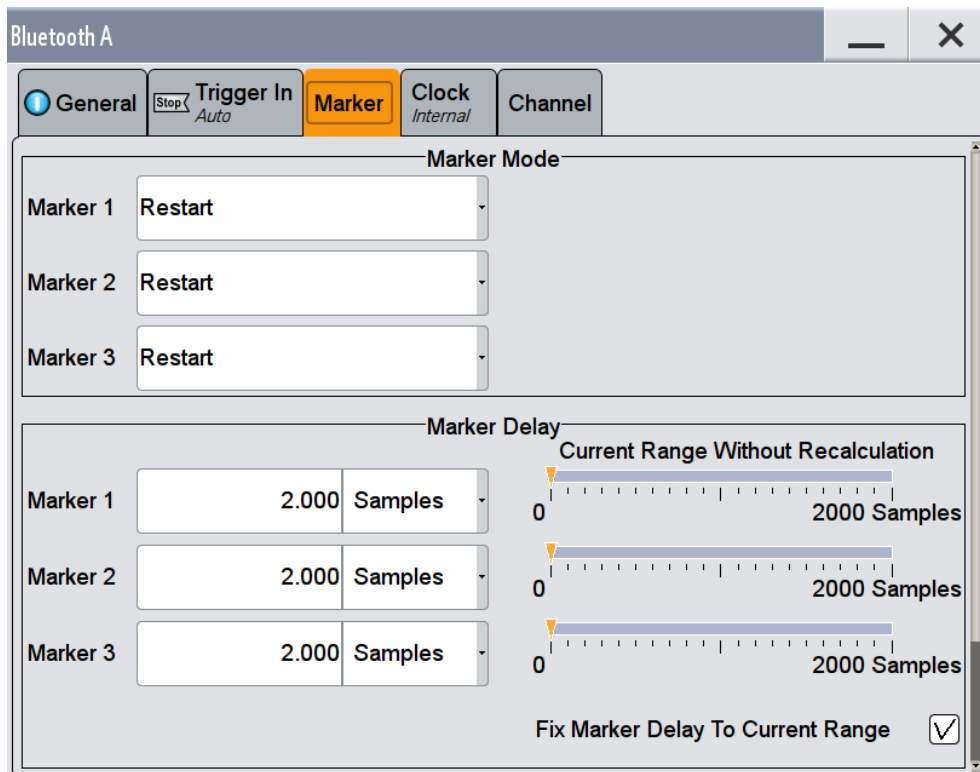
This tab provides an access to the settings necessary to select and configure the marker output signal, like the marker mode or marker delay settings.



This section focuses on the available settings.

For information on how these settings affect the signal, refer to section "Basics on ..." in the R&S SMW user manual.

- To access this dialog, select "Baseband > Bluetooth > Marker".



This dialog comprises the settings required for configuring the marker.



Routing and Enabling a Marker

The provided marker signals are not dedicated to a particular connector but can be mapped to one or more globally shared USER or local T/M/(C) connectors.

To route and enable a marker signal, perform the following *general steps*:

- Define the shape of the generated marker, i.e. select the "Marker > Mode"
- Define the connector, USER or T/M/(C), the selected signal is output at, i.e. configure the [Local and Global Connector Settings](#).

Provided are the following settings:

Marker Mode

Marker configuration for up to 3 marker channels. The settings are used to select the marker mode defining the shape and periodicity of the markers. The contents of the dialog change with the selected marker mode; the settings are self-explanatory.

- | | |
|---------------------|---|
| "Restart" | A marker signal is generated at the start of each signal sequence. |
| "Frame Start" | A marker signal is generated at the start of each frame. |
| "Frame Active Part" | The marker masks the active part of the frame. At the start of each burst, the marker signal changes to high. It changes back to low after the end of each burst. |

"Pulse" A regular marker signal is generated. The pulse frequency is defined by entering a divider. The frequency is derived by dividing the sample rate by the divider. The input box for the divider opens when "Pulse" is selected, and the resulting pulse frequency is displayed below it.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:PULSe:DIVider`

on page 90

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:PULSe:FREQuency?`

on page 90

"Pattern " A marker signal that is defined by a bit pattern is generated. The pattern has a maximum length of 32 bits and is defined in an input field which opens when pattern is selected.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:PATtern` on page 90

"ON/OFF Ratio" A regular marker signal that is defined by an ON/OFF ratio is generated. A period lasts one ON and OFF cycle. The "ON Time" and "OFF Time" are each expressed as a number of samples and are set in an input field which opens when ON/OFF ratio is selected.



Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:ONTime` on page 89

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:OFFTime` on page 89

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:MODE` on page 88

Marker x Delay

Defines the delay between the marker signal at the marker outputs relative to the signal generation start.

"Marker x" For the corresponding marker, sets the delay as a number of symbols.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:DELay` on page 88

"Current Range without Recalculation"

Displays the dynamic range within which the delay of the marker signals can be set without restarting the marker and the signal.

Move the setting mark to define the delay.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:DELay:MINimum?`

on page 88

`[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut<ch>:DELay:MAXimum?`

on page 88

"Fix marker delay to current range"

Restricts the marker delay setting range to the dynamic range.

Remote command:

[:SOURce<hw>] :BB:BT0oth:TRIGger:OUTPut:DElay:FIXed on page 87

4.4 Clock Settings

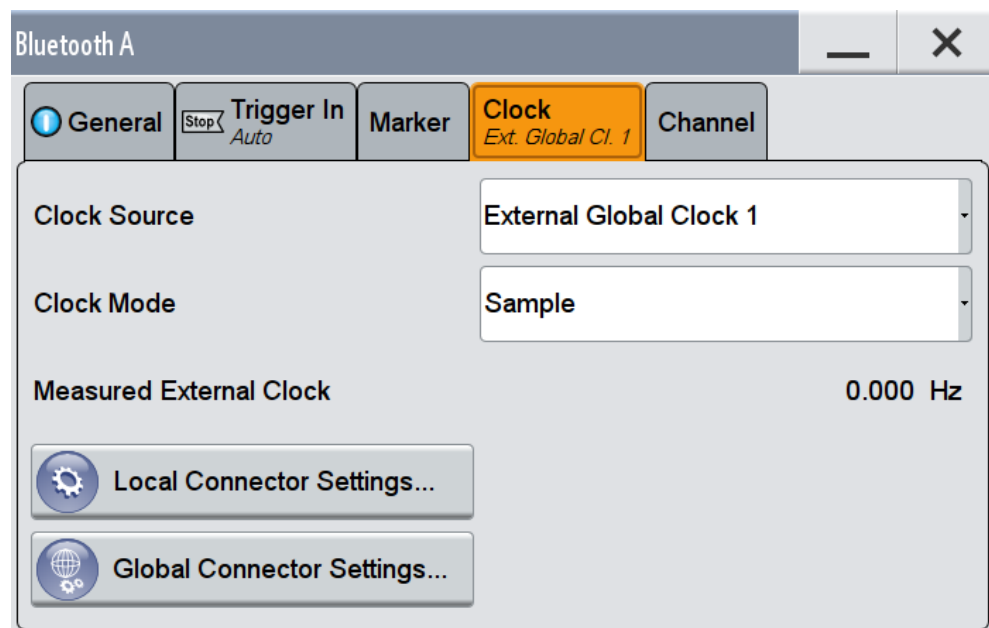
This tab provides an access to the settings necessary to select and configure the clock signal, like the clock source and clock mode.



This section focuses on the available settings.

For information on how these settings affect the signal, refer to section "Basics on ..." in the R&S SMW user manual.

- ▶ To access this dialog, select "Baseband > Bluetooth > Clock".



This dialog comprises the settings required for configuring the clock.



Defining the Clock

The provided clock signals are not dedicated to a particular connector but can be mapped to one or more globally shared USER and the two local T/M/C connectors.

Use the [Local and Global Connector Settings](#) to configure the signal mapping and the polarity, the trigger threshold, and the input impedance of the input connectors.

To route and enable a trigger signal, perform the following *general steps*:

- Define the signal source, that is select the "Clock > Source"
- Define the connector, USER or T/M/C, the selected signal is provided at, that is configure the [Local and Global Connector Settings](#).

Provided are the following settings.

Clock Source

Selects the clock source.

- "Internal"
The instrument uses its internal clock reference.
- "External Global Clock 1/2"
The instrument expects an external clock reference at the global USER connector, as configured in the "Global Connector Settings" dialog.
- "External Local Clock"
The instrument expects an external clock reference at the local T/M/C connector.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:CLOCK:SOURce` on page 91

Clock Mode

Enters the type of externally supplied clock.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:CLOCK:MODE` on page 91

Clock Multiplier

Enters the multiplication factor for clock type "Multiple".

Remote command:

`[:SOURce<hw>] :BB:BT0oth:CLOCK:MULTIplier` on page 91

Measured External Clock

Provided for permanent monitoring of the enabled and externally supplied clock signal.

Remote command:

`CLOCK:INPut:FREQuency?`

4.5 Local and Global Connector Settings

Each of the "Trigger In", "Marker" and "Clock" dialogs as well as the "Trigger Marker Clock" dialog provides a quick access to the related local and global connector settings.

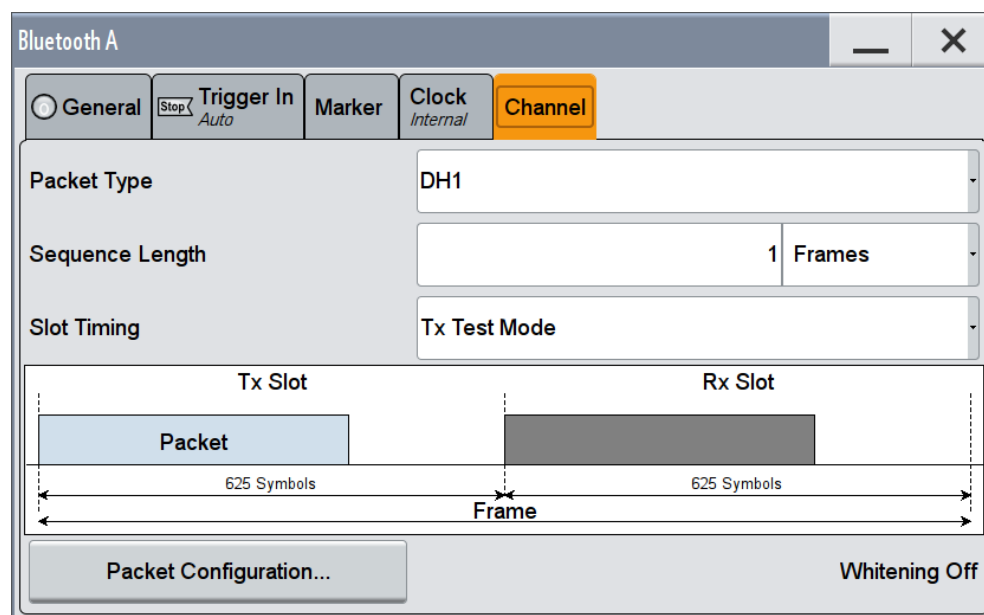
For more information, refer to the description R&S SMW User Manual, section "Local and Global Connectors".

4.6 Bluetooth Channel Settings

This dialog provides access to the "Bluetooth Basic Rate + EDR" or the "Bluetooth Low Energy" settings, depending on the selected bluetooth mode, see "[Bluetooth Mode](#)" on page 20.

4.6.1 Bluetooth Basic Rate + EDR

1. To access this dialog, select "Bluetooth > General > Bluetooth Mode > Basic Rate + EDR".
2. Select "Channel".



The dialog contains the parameters to define the packet type and provides access to the packet type configuration dialog. The graphic shows the frame structure of the selected packet type.

Packet Type

Selects the packet type.

The available packets depend on the selected [Transport Mode](#).

All packet types as defined in the Bluetooth specification are supported. For an overview, see [Chapter 3.2, "Bluetooth Packet Types"](#), on page 11.

Remote command:

[:SOURce<hw>] :BB:BT0oth:PTYPe on page 70

Sequence Length

Selects the sequence length in frames of the generated signal. The signal repeats after the specified number of frames.

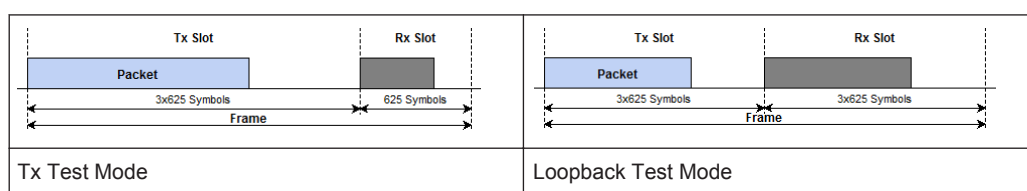
Remote command:

[:SOURce<hw>] :BB:BT0oth:SLENgth on page 72

Slot Timing

Selects the timing mode for the Rx slot.

The graphic below shows the frame structure of the selected [Packet Type](#) and slot timing.



A transmitted packet has a duration of $N \times 625 \mu\text{s}$ where N is an odd integer larger than 0. N depends on the type of the transmitted packet. In "Tx Test" mode, $N = 1$ for Rx slots.

"Tx Test Mode"

The transmitted Rx package takes 625 symbols, regardless of the selected packet type.

"Loopback Test Mode"

Extends the Rx slot time according to the selected packet type. For example, the Rx slot of [Packet Type > DH3](#) takes 3×625 symbols.

Remote command:

[:SOURce<hw>] :BB:BT0oth:STIMing on page 72

Packet Configuration

Access the "Packet Configuration" dialog, see [Chapter 4.6.2, "Packet Configuration - Bluetooth Basic Rate + EDR"](#), on page 32.

The current data source for packet and the data whitening state are displayed next to the button.

Remote command:

n.a.

4.6.2 Packet Configuration - Bluetooth Basic Rate + EDR

To access the packet configuration dialog, perform the following:

1. Select "Bluetooth > General > Bluetooth Mode > Basic Rate + EDR".

2. Select "Packet Configuration".

The dialog contains the parameters for configuring the packet type. The available parameters vary according to the selected [Packet Type](#) and data source.

Data Source for Packet

The data sent for each packet can be comfortably edited with the Packet Editor, or filled with a predefined ALL Data sequence.

"Packet Editor" Enables the edit mode to configure the packet fields individually.

"All Data" Fills the generated packets with the selected data source. This mode is useful if you need to load predefined data contents from a data list file or the data contents of the packet are not of interest.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:DSFPacket` on page 97

Data Whitening

Activates the Data Whitening.

Evenly distributed white noise is ideal for the transmission, and real data can be forced to look similar to white noise with different methods called Data Whitening.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:DWhitening` on page 98

Bluetooth Device Address (BD_ADDR)

Enters Bluetooth Device Address. Each Bluetooth device shall be allocated a unique 48-bit Bluetooth device address (BD_ADDR).

The BD_ADDR may take any values except the 64 reserved LAP values: 0x9E8B00 – 0x9E8B3F.

"NAP"	Selects non-significant address part. The length of NAP is 16 bits or 4 hexadecimal figures.
"UAP"	Selects upper address part. The length of UAP is 8 bits or 2 hexadecimal figures.
"LAP"	Selects lower address part. The length of LAP is 24 bits or 6 hexadecimal figures.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:BDANap](#) on page 95

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:BDUAp](#) on page 95

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:BDALap](#) on page 94

Logical Transport Address

(Available for all packet types except ID)

Enters the logical transport address for the header.

Each slave active in a piconet is assigned a primary logical transport address (LT_ADDR). The all-zero LT_ADDR is reserved for broadcast messages.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:LTAddress](#) on page 99

Flow Control (Header)

(Available for all packet types except ID)

Sets the FLOW bit in the header. This bit indicates start or stop of transmission of packets over the ACL logical transport.

"Go" Allows the other devices to transmit new data.

"Stop" Stops the other devices from transmitting data temporarily.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:HFControl](#) on page 99

Acknowledgment

(Available for all packet types except ID)

Sets the ARQN bit of the packet header.

"NAK" Request to retransmit the previous payload.

"ACK" Previous payload has been received successfully.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:ACKnowledgement](#) on page 94

SEQN Start Value

(Available for all packet types except ID)

Sets the start value of the header SEQN bit.

The SEQN bit is present in the header to filter out retransmissions in the destination. The signal generator is altering this bit automatically on consecutive frames, if a sequence length of at least 2 frames is set.

"0" The SEQN bit starts with 0.

"1" The SEQN bit starts with 1.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:SNValue](#) on page 100

Data Source

(Available for all packet types except ID, POLL, NULL and FHS packets)

Selects the data source used for the payload.

The following standard data sources are available:

- "All 0, All 1"
An internally generated sequence containing 0 data or 1 data.
- "PNxx"
An internally generated pseudo-random noise sequence.
- "Pattern"
An internally generated sequence according to a bit pattern.
Use the "Pattern" box to define the bit pattern.
- "Data List/Select DList"
A binary data from a data list, internally or externally generated.
Select "Select DList" to access the standard "Select List" dialog.
 - Select the "Select Data List > navigate to the list file *.dm_iqd > Select" to select an existing data list.
 - Use the "New" and "Edit" functions to create internally new data list or to edit an existing one.
 - Use the standard "File Manager" function to transfer external data lists to the instrument.

See also:

- Section "Modulation Data" in the R&S SMW user manual.
- Section "File and Data Management" in the R&S SMW user manual.
- Section "Data List Editor" in the R&S SMW user manual

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:DATA](#) on page 96

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:DATA:DPATtern](#) on page 96

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:DATA:DSElection](#) on page 96

Data Length

(Available for all packet types except ID, POLL, NULL and FHS packets)

Enters the payload data length in bytes.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PCONfiguration:DLENgth](#) on page 97

Flow Control (Payload)

(Available for all packets types except ID, POLL, NULL, FHS, HV1, HV2, HV3, EV3, EV4, EV5, 2-EV3, 2-EV5, 3-EV3, 3-EV5 packets.)

Sets the FLOW bit in the payload (flow control per logical link)

"Go" Indicates start of transmission of ACL packets after a new connection has been established.

"Stop" Indicates stop of transmission of ACL packets before an additional amount of payload data is sent.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:PFControl` on page 99

Packet Length

(Available in All Data mode and for all packet types except ID packet)

Enters the packet length in symbols.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:PLENght` on page 100

EIR packet follows

(Available for FHS packets)

Indicates that an extended inquiry response packet may follow.

"Yes" Indicates that an EIR packet follows.

"No" Indicates that EIR does not follow.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:EIRPacketfollows` on page 98

Scan Repetition Mode

(Available for FHS packets)

The 2-bit scan repetition field indicates the interval between two consecutive page scan windows, determines the behavior of the paging device.

"R0" The scan interval is equal to the scan window T_w page scan (continuous scan) and maximal 1.28s.

"R1" The scan interval is maximal 1.28s.

"R2" The scan interval is maximal 2.56s.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:SRMode` on page 101

Class of Device

(Available for FHS packets)

A parameter received during the device discovery procedure, indicates the type of device and which types of service that are supported.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:PCONfiguration:CODevice` on page 95

Data Source (Voice Field)

(Available for DV packets)

Selects the Data Source for the voice field.

The following standard data sources are available:

- "All 0, All 1"
An internally generated sequence containing 0 data or 1 data.
- "PNxx"

- An internally generated pseudo-random noise sequence.
- "Pattern"
 - An internally generated sequence according to a bit pattern.
 - Use the "Pattern" box to define the bit pattern.
- "Data List/Select DList"
 - A binary data from a data list, internally or externally generated.
 - Select "Select DList" to access the standard "Select List" dialog.
 - Select the "Select Data List > navigate to the list file *.dm_iqd > Select" to select an existing data list.
 - Use the "New" and "Edit" functions to create internally new data list or to edit an existing one.
 - Use the standard "File Manager" function to transfer external data lists to the instrument.

See also:

- Section "Modulation Data" in the R&S SMW user manual.
- Section "File and Data Management" in the R&S SMW user manual.
- Section "Data List Editor" in the R&S SMW user manual

Remote command:

`[:SOURCE<hw>] :BB:BT00th:PCONfiguration:VDATA` on page 101

`[:SOURCE<hw>] :BB:BT00th:PCONfiguration:DATA:VDPattern` on page 96

`[:SOURCE<hw>] :BB:BT00th:PCONfiguration:DATA:VDSElection`
on page 97

4.6.3 Bluetooth Low Energy

The R&S SMW provides you with the ability to generate signals in accordance with Bluetooth Low Energy Specification (LE). Bluetooth Low Energy makes it possible to transfer data from low power devices running on the smallest of batteries to a larger device, such as a PC, a mobile phone, or a PDA. For the first time, a Bluetooth connection to a wristwatch, or a heart rate sensor, or a data transfer from a digital camera, is possible. The Bluetooth low energy chips will offer capabilities that do not replace or supersede the existing Bluetooth 2.x standards. Data rates are comparable to Bluetooth 1.1, and are data-only (no audio content).

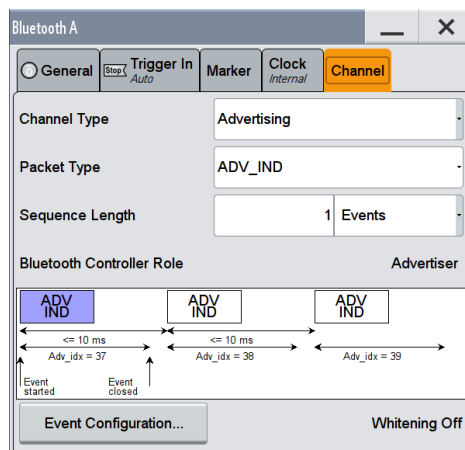
Bluetooth LE mode only uses the Basic Rate. The Basic Rate mode uses binary FM modulation and has a data rate of 1 MBps. The modulation scheme has the symbol rate equal to 1Ms/s.

For full duplex transmission, a Time Division Duplex (TDD) scheme is used.

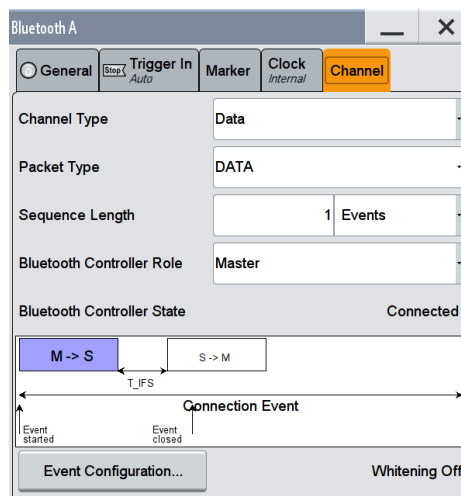
In the following description Bluetooth Low Energy is abbreviated as Bluetooth LE.

1. To access the *Bluetooth LE* settings, select "Bluetooth > General > Bluetooth Mode > Bluetooth Low Energy".
2. Select "Channel".

The "Channel" dialog varies depending on the selected "Channel Type"



Bluetooth LE "Channel Type Advertising"



Bluetooth LE "Channel Type Data"

The dialogs contain the parameters to define the packet type and provide access to the packet type configuration dialogs. The graphic shows the frame structure of the selected packet type.

Channel Type

Determines the channel type. Advertising and data are available. Refer to [Chapter 4.6.4, "Event / Frame Configuration - Bluetooth LE"](#), on page 41 for setting the respective parameters.

"Advertising" Selects channel type Advertising.

"Data" Selects channel type Data. Devices in a connected state transmit data channel packets in connection events with a start point and an interval.

Remote command:

[:SOURce<hw>] :BB:BT0oth:CTYPe on page 69

Packet Type

Selects the packet type.

The available packet types depend on the selected channel type, as shown in the table below ([Table 4-1](#)).

Table 4-1: Packet types of the respective channel types:

Packet Type	Advertising	Data
ADV_IND/	x	-
ADV_DIRECT_IND	x	-
ADV_NONCONN_IND	x	-
ADV_DISCOVER_IND	x	-
SCAN_REQ	x	-
SCAN_RSP	x	-

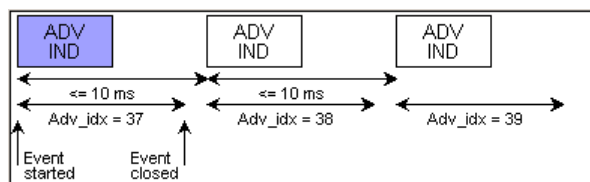
Packet Type	Advertising	Data
CONNECT_REQ	x	-
DATA	-	x
CONTROL_DATA ...	-	x
TEST PACKET	x	x

Depending on the [Bluetooth Controller Role](#) (master or slave), you can determine in detail the information of the "CONTROL_DATA", as shown in the following table [Table 4-2](#).

Table 4-2: Control information, available for master or slave.

CONTROL_DATA	Master	Slave
LL_CONNECTION_UPDATE_REQ	x	-
LL_CHANNEL_MAP_REQ	x	-
LL_TERMINATE_IND	x	-
LL_ENC_REQ	x	-
LL_RNC_RESP	-	x
LL_START_ENC_REQ	x	x
LL_START_ENC_RESP	x	x
LL_UNKNOWN_RESP	-	x
LL_FEATURE_REQ	x	-
LL_FEATURE_RESP	-	x
LL_PAUSE_ENC_REQ	x	-
LL_PAUSE_ENC_RESP	-	x
LL_VERSION_IND	x	x
LL_REJECT_IND	x	x

The graphic shows the frame structure of the selected packet type.



Remote command:

`[:SOURCE<hw>] :BB:BT0oth:UPTYPE` on page 128

Sequence Length

Selects the number of frames or events depending on the packet type. The signal repeats after the specified number of frames/events.

For SCAN_REQ and CONNECT_REQ packet, the sequence length is expressed in "Frames".

For TERMINATE_IND packets, a default value according to the specification is given:

- Master: 'SlaveLatency + 6'
- Slave: '6'

For all other packet types the sequence length is expressed in "Events".

Remote command:

`[:SOURCE<hw>] :BB:BT00th:USLength` on page 129

Bluetooth Controller Role

Determines the controller role.

Note: The available packet types for the selected channel types and the controller roles are described in "Bluetooth Controller Role" on page 40.

Depending on the channel type, the field either displays the appropriate role or you can select one:

- "Advertiser"
Displays the controller role corresponding to the packet type:
 - "Advertiser" for all ADV-xxx packet types and SCAN_RSP
 - "Scanner" for SCAN_REQ packet type
 - "Initiator" for CONNECT_REQ packet type
- "Data"
Assigns a role to the controller:
 - "Master"
 - "Slave"

Remote command:

`[:SOURCE<hw>] :BB:BT00th:BCRole` on page 108

Bluetooth Controller State

Shows the state of the bluetooth controller for channel type "Data".

Remote command:

`[:SOURCE<hw>] :BB:BT00th:BCText?` on page 69

Event / Frame Configuration

Provides access to the "Event Configuration" dialog, if the sequence length of the packet type is expressed in events, and accordingly, the "Frame Configuration" dialog, if it is expressed in frames, see [Chapter 4.6.4, "Event / Frame Configuration - Bluetooth LE"](#), on page 41.

The data whitening state is displayed next to the button.

Remote command:

n.a.

Test Packet Configuration

Provides access to the "Test Packet Configuration" dialog for packet type "TEST PACKET", see [Chapter 4.6.5.2, "Test Packet Configuration Settings"](#), on page 56.

4.6.4 Event / Frame Configuration - Bluetooth LE

1. To access the event/frame settings for *Bluetooth LE*, select "Bluetooth > General > Bluetooth Mode > Bluetooth Low Energy"
2. Select "Channel > Event / Frame Configuration".

The "Event" or "Frame" dialogs vary, depending on the used channel type:

Bluetooth A: Event Configuration

Advertising Settings

Advertising Event Interval: 20.0 ms Advertising Event Delay: 0.0 ms

Packet Configuration...

Center frequency(MHz)

2402 2404 2424 2426 2428 2478 2480

ADV Data Data ADV Data Data ADV

37 0 10 38 11 36 39

Channel Index

Channel Table

	Center Frequency /MHz	Channel Index	Channel Type	Channel State
Channel 0	2 402	37	Advertising	Used
Channel 1	2 426	38	Advertising	Unused
Channel 2	2 480	39	Advertising	Unused

Figure 4-1: Event Configuration Dialog of channel type Advertising

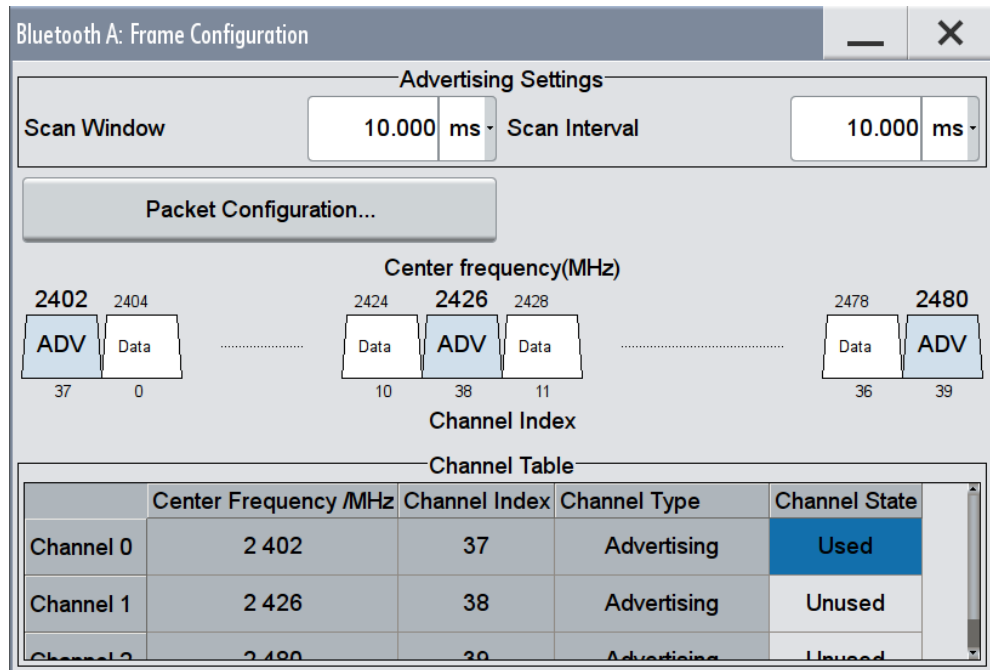


Figure 4-2: Frame Configuration Dialog of channel type Advertising

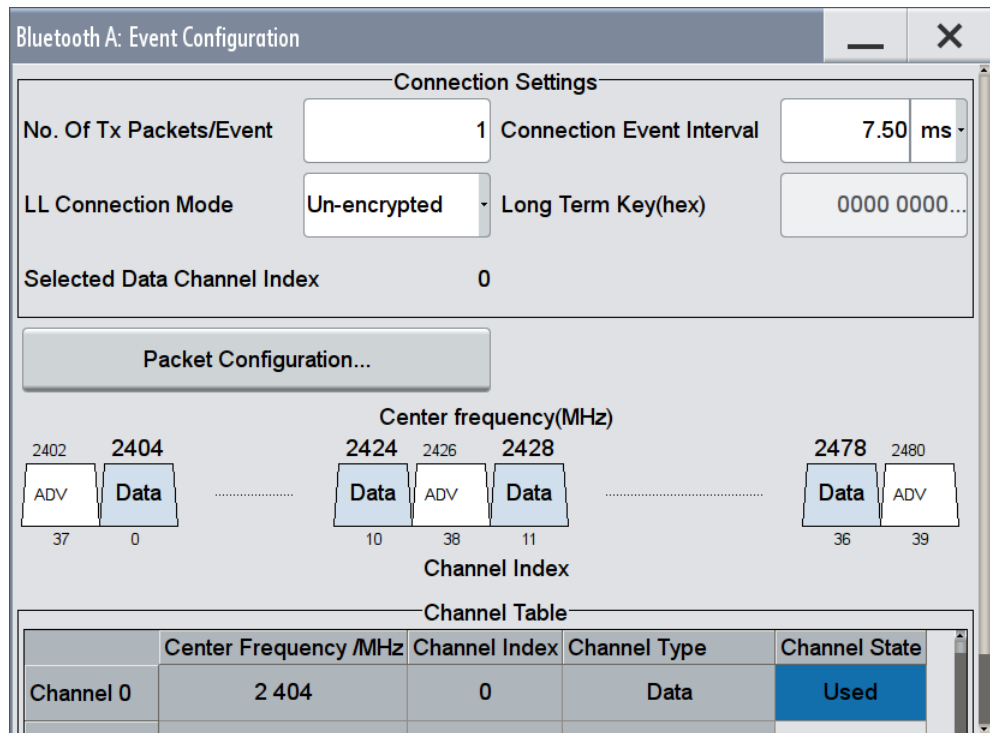


Figure 4-3: Event Configuration Dialog of channel type Data

The dialogs contain the parameters to configure the event or frame settings, and provide access to the packet configuration dialogs. The graphics show the distribution of the packets, the physical channel mapping and the channel indices. The channel table gives an overview of the used channels and their assignments.

4.6.4.1 Advertising Event / Frame Configuration Settings

Advertising Event Interval

Sets the time interval between two consecutive advertising events, with regard to the starting points.

Note: This parameter is relevant for advertising event configuration and for the packet types ADV-IND, ADV_DIRECT_IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:AEINterval` on page 112

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:ADINterval` on page 112

Advertising Event Delay

Sets a time delay between the start times of two consecutive advertising events. The value is added to the advertising event interval.

Note: This parameter is relevant for advertising event configuration and for the packet types ADV-IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:AEDelay` on page 111

Scan Window

Sets the length of the window during which the scanner is operating in the advertising channel.

Note that the scan window is less or equal to the value of the scan interval.

Note: This parameter is relevant for advertising frame configuration and for the packet type SCAN_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:SWINdow` on page 127

Scan Interval

Sets the time interval between the starting points of two consecutive windows during which the scanner is operating in an advertising channel.

Note: This parameter is relevant for advertising frame configuration and for the packet type SCAN_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:SINterval` on page 126

Advertising Packet Interval

Sets the time interval between packets starting points of two consecutive packets in the advertising channel.

Note: This parameter is relevant for advertising frame configuration and for the packet type SCAN_RSP.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:APInterval` on page 112

Transmit Window Offset

Displays the start point of the transmit window.

Note: This parameter is relevant for advertising frame configuration and for the packet type CONNECT_REQ.

This parameter is set in the Packet Configuration, see "[Transmit Window Offset](#)" on page 55.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:WOINfo?` on page 127

Transmit Window Size

Indicates the size of the transmit window, regarding to the start point.

Note that the scan window size is less or equal to the value of the connection interval.

Note: This parameter is relevant for advertising frame configuration and for the packet type CONNECT_REQ.

The parameter is set in the "Packet Configuration" dialog, see "[Transmit Window Size](#)" on page 54.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:WSINfo?` on page 127

Packet Configuration

Opens the dialog for setting the corresponding packet configuration.

This dialog is described in [Chapter 4.6.5.1, "Packet Configuration Settings"](#), on page 48.

Remote command:

n.a.

4.6.4.2 Data Event Connection Settings

No. of Tx Packets per Event

Sets the number of Tx packets per event. Each connection contains at least one data channel packet. The maximum number of packets per event is determined by the duration of the connection event interval.

Note: This parameter is relevant for data event connection settings.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PNUMber` on page 126

Connection Event Interval

Set the time interval between the start points of two consecutive connection events. Subsequent transmissions within an event are separated by this parameter in order to separate connecting event starting points in time.

Note: This parameter is relevant for data event connection settings and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:

[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CINterval
on page 115

[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NCINterval
on page 120

[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CINterval
on page 115

LL Connection Mode

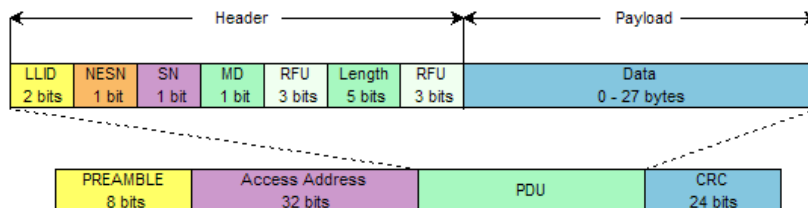
Select the link layer connection mode. In order to provide safe transmission of payload data, the data in the packet can be encrypted. If activated, the payload data follows MIC (Message authentication Code).

Note: This parameter is relevant for data event connection settings.

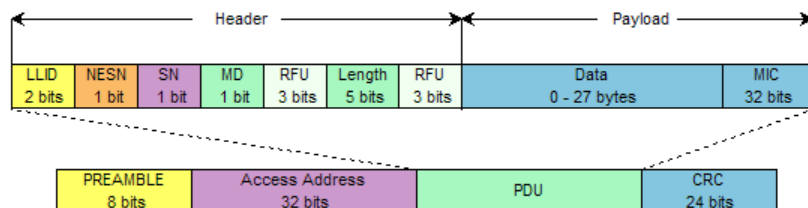
The following table shows which types of packets can be encrypted and / or unencrypted.

Packet Type	encrypted	unencrypted
DATA	X	X
CONNECTION_UPDATE_REQ	X	X
CHANNEL_MAP_REQ	X	X
LL_TERMINATE_IND	X	X
LL_ENC_REQ	-	X
LL_ENC_RSP	-	X
LL_START_ENC_REQ	-	X
LL_START_ENC_RSP	X	-
LL_FEATURE_REQ	X	X
LL_FEATURE_RSP	X	X
LL_PAUSE_ENC_REQ	-	X
LL_PAUSE_ENC_RSP	X	X
LL_VERSION_IND	X	X
LL_REJECT_IND	X	X
UNKNOWN_RSP	X	X

"Unencrypted" Payload data is transmitted without encoding.



"Encrypted" The link layer connection runs in encrypted mode.



Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:LCMode` on page 113

Long Term key (hex)

Indicates the time the controller needs to receive the long term key from the host. After this time, the controller is ready to enter into the last phase of encryption mode setup.

Note: This parameter is relevant for data event connection settings. In encrypted mode, the code can be edited.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:LTKey` on page 113

Selected Data Channel Index

Indicates the number of the first active data channel.

The data channel is selected for each connection event. The master and slave determine the used data channel by selecting from the list of used channels (see "[Channel Table](#)" on page 46).

Note: This parameter is relevant for data event connection settings.

Displays the data channel index currently selected.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:SDCI?` on page 123

4.6.4.3 Channel Table Settings

The channel table displays all parameters characterizing the channel and the current state.

Channel Table

The channel table displays all parameters characterizing the channel and the current state.

Every channel is represented with bit positioned as per the data channel index. LSB represents data channel index 0 and the bit in position 36 represents data channel index 36.

If the channel is used channel its bit is to be set to '1'. Bit value '0' indicates that the channel is unused.

The bits in positions 37, 38 and 39 shall be set to zero upon transmission and ignored upon receipt.

"Center Frequency"

Indicates the center frequency of a channel.

"Channel Index"

Indicates the channel index.

"Channel Type"

Indicates the channel type.

"Channel State"

Indicates used and unused data channels.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:ECOnfiguration:ACTable:CHANnel<ch0> : STATE on page 111`

`[:SOURce<hw>] :BB:BT0oth:ECOnfiguration:DCTable:CHANnel<ch0> : STATE on page 111`

`[:SOURce<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DCMTable: CHANnel<ch0> : STATE on page 111`

Remote command:

Entire data set:

`[:SOURce<hw>] :BB:BT0oth:ECOnfiguration:ACTable on page 110`

`[:SOURce<hw>] :BB:BT0oth:ECOnfiguration:DCTable on page 110`

`[:SOURce<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DCMTable on page 110`

4.6.5 Packet Configuration - Bluetooth LE

To access the *Bluetooth LE* packet configuration settings, perform the following:

1. Select "Bluetooth > General > Bluetooth Mode > Bluetooth LE".
2. Select "Channel > Event/Frame Configuration", according to the selected packet type.

3. In the corresponding "Event" or "Frame" dialog, select "Packet configuration".

The dialog contains the parameters for configuring the selected packet type. The available parameters vary according to the used channel and packet types.

4.6.5.1 Packet Configuration Settings

Data Whitening

Activates or deactivates the Data Whitening.

Evenly distributed white noise is ideal for the transmission and real data can be forced to look similar to white noise with different methods called Data Whitening. Applied to the PDU and CRC fields of all packet types, whitening is used to avoid long equal sequences in the data bit stream.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DWHitening`
on page 117

Access Address

Sets the access address of the link layer connection.

Bluetooth LE transmissions are based on an interface packet format, that consists of a preamble (8 bits), the access address (32 bits), the PDU and CRC (24 bits).

The access address' structure depends on the packet type:

- Data channel packets
The access address is a pseudo-random LL connection address, generated by the initiator of the LL connection. The address has to follow some specific rules, which are described in the "Bluetooth Low Energy Technology Specification".
- Advertising channel packets
The address is fixed to 0110101101111011001000101110001 with the left most bit sent first and being the LSB.

Note: This parameter is relevant for all available package types specified in events in the data channel, and frames in the advertiser channel, i.e. DATA, CONNECTION_UPDATE_REQ, CHANNEL_MAP_REQ, TERMINATE_IND, LLENC_REQ, LL_FEATURE_REQ, LL_PAUSE_ENC_REQ, LL_START_ENC_RSP, FEATURE_REQ, and CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:AADdress`
on page 113

NESN Start Value

Sets the start value of the next expected packet from the same device in the LL connection (**N**ext**E**xpected **S**equ**N**umber). This parameter can be set in the first event. From the second event this field is not indicated.

Note: This parameter is relevant for data event configuration and all data channel packet types except TEST_PACKET.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NSValue`
on page 121

SN Start Value

Sets the sequence number of the packet. This parameter can be set in the first event. From the second event this field is not indicated.

Note: This parameter is relevant for data event configuration and all data channel packet types except TEST_PACKET.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:SSValue`
on page 124

Device's Addr Type

Selects the address type of the controller device. Depending on the Bluetooth controller role either the Tx or Rx or both address types are assigned.

Subdivided into private and random, a Bluetooth LE device address consists of 48 bits. The format of the device address differs depending on the selected address type.

Note: This parameter is relevant for advertising event or frame configuration.

The bluetooth "Controller Role" and the packet type determine the available entries:

- **Tx** in conjunction with the packet types ADV_IND, ADV_DIRECT_IND, ADV_NONCONN_IND, ADV_DISCOVER_IND, SCAN_REQ, SCAN_RSP and CONNECT_REQ
- **Rx** for the packet types ADV_DIRECT_IND, SCAN_REQ and CONNECT_REQ

"Public" Allocates a unique 48 bit address to each bluetooth LE device. The public address is given from the registration authority IEEE.

"Random" Allocates a 48 bit address to each bluetooth LE device. A random address is optional.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:TAType`
on page 122

Data Source

Selects the data source used for the payload.

Note: This parameter is relevant for event configuration and packet types DATA, ADV_IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

The following standard data sources are available:

- "All 0, All 1"
An internally generated sequence containing 0 data or 1 data.
- "PNxx"
An internally generated pseudo-random noise sequence.
- "Pattern"
An internally generated sequence according to a bit pattern.
Use the "Pattern" box to define the bit pattern.
- "Data List/Select DList"
A binary data from a data list, internally or externally generated.
Select "Select DList" to access the standard "Select List" dialog.
 - Select the "Select Data List > navigate to the list file *.dm_iqd > Select" to select an existing data list.
 - Use the "New" and "Edit" functions to create internally new data list or to edit an existing one.
 - Use the standard "File Manager" function to transfer external data lists to the instrument.

See also:

- Section "Modulation Data" in the R&S SMW user manual.
- Section "File and Data Management" in the R&S SMW user manual.
- Section "Data List Editor" in the R&S SMW user manual

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DATA`
on page 116

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DATA:DPATtern` on page 116

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DATA:DSElection` on page 117

Data Length

Enters the payload data length in bytes.

Note: This parameter is relevant for event configuration with packet types ADV_IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:DLength`
on page 117

Connection Event Interval

Set the time interval between the start points of two consecutive connection events. Subsequent transmissions within an event are separated by this parameter in order to separate connecting event starting points in time.

Note: This parameter is relevant for data event connection settings and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CINterval`
on page 115

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NCINterval`
on page 120

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CINterval`
on page 115

Slave Latency

Sets a number of consecutive connection events the slave can ignore for asymmetric link layer connections.

Note: This parameter is relevant for data event and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:SLATency`
on page 124

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NSLatency`
on page 121

LL Connection Timeout

Defines the maximum time between two correctly received Bluetooth LE packets in the LL connection before the connection is considered lost.

Note: This parameter is relevant for data event and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:LCTimeouT`
on page 119

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NLCTimeouT`
on page 120

Connection Instant

Sets a connection instant for indicating the connection event at which the new connection parameters are taken in use.

Both the master and the slave have a 32-bit connection event counter per LL connection. It is reset to zero on the first connection event of the LL connection and incremented by one on every elapsed connection event interval of the LL connection.

Note: This parameter is relevant for data event configuration with the packet types CONNECTION_UPDATE_REQ and CHANNEL_MAP_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CINStant`
on page 115

Show / Hide Data Channel (Mapping) Table

In data event and advertising frame configuration with the packet types CHANNEL_MAP_REQ and CONNECT_REQ, calls / hides the channel map table that displays the used channels and their parameters.

The channel table is described in [Chapter 4.6.4.3, "Channel Table Settings"](#), on page 46.

Remote command:

n.a.

Hop Length

Sets the difference from the current channel to the next channel. The master and slave devices determine the data channel in use for every connection event from the channel map. Hop_length is set for the LL connection and communicated in the CONNECT_REQ packets.

Note: This parameter is relevant for data event and advertising frame configuration with the packet type CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:HLEnGth`
on page 119

Random Vector (hex)

Sets the random vector of the master for device identification.

The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

Note: This parameter is relevant for data event configuration with the packet type LLENC_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:RVEctor`
on page 123

Encrypted Diversifier (hex)

Sets the encrypted diversifier of the master for device identification. The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

Note: This parameter is relevant for data event configuration with the packet type LLENC_REQ.

Remote command:

```
[ :SOURCE<hw> ] :BB:BT0oth:ECOnfiguration:PCOnfiguration:
EDIVersifier on page 118
```

Session Key ID (hex)

Sets the master's or the slave's portion of the session key diversifier (SKDm/SKDs).

Note: This parameter is relevant for data event configuration with the packet types LL_ENC_REQ (Master) and LL_ENC_RSP (Slave).

Remote command:

```
[ :SOURCE<hw> ] :BB:BT0oth:ECOnfiguration:PCOnfiguration:MSKD
on page 120
```

Initialization Vector (hex)

Sets the master's or the slave's portion of the initialization vector (IVm/IVs).

Note: This parameter is relevant for data event configuration with the packet types LL_ENC_REQ (Master) and LL_ENC_RSP (Slave).

Remote command:

```
[ :SOURCE<hw> ] :BB:BT0oth:ECOnfiguration:PCOnfiguration:MIVector
on page 119
```

Feature Set Length

Enables that the feature set length is indicated.

FeatureSet indicates whether the Controller features are used or not. All the data in FeatureSet is RFU(zero).

Note: This parameter is relevant for data event configuration with the packet types FEATURE_REQ (Master) and FEATURE_RSP (Slave).

Remote command:

```
[ :SOURCE<hw> ] :BB:BT0oth:ECOnfiguration:PCOnfiguration:FSLength
on page 118
```

Unknown Type (hex)

Enables that an invalid control packet is indicated.

The CtrType field indicates the value of the LL control packet that caused the transmission of this packet.

This parameter is relevant for data event configuration with the packet type UNKNOWN_RSP (Slave).

Remote command:

```
[ :SOURCE<hw> ] :BB:BT0oth:ECOnfiguration:PCOnfiguration:UTYPE
on page 125
```

Controller's Device Addr

Sets the advertiser's device address.

In Bluetooth LE systems all the transmissions start with an 8 bit preamble followed by an access address. The access address is composed of the parts "Company_Id" (LSB) and the "Company_assigned" (MSB). Beside the address fields the notation is given.

For advertising channel packets the format of the device address differs, depending on the selected address type.

Note: This parameter is relevant for advertising event or frame configuration. Refer to [Tx/Rx Address Type - Bluetooth LE Device's Addr Type](#) for information about the available package types for the respective "Controller Roles".

- "Public Address Types"
 - The public address is given from the registration authority IEEE and is composed of:
 - LSB: 24 bits = company_assigned
 - MSB: 24 bits = company_id
- "Private Address Type"
 - A private address is optional and composed of:
 - LSB: 24 bits = hash
 - MSB: 24 bits = random

Remote command:

Company_Assigned and Company_Id in Advertiser's Device Address

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:ACID`

on page 114

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:ACASsigned`

on page 114

Company_Assigned and Company_Id in Scanner's Device Address

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:SCASsigned`

on page 114

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:SCID`

on page 114

Company_Assigned and Company_Id in Initiator's Device Address

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:ICASsigned`

on page 114

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:ICID`

on page 114

CRC Initial

Sets the initialization value for the CRC (Cyclic Redundary Check, 24 bits) calculation. A packet has been received correctly, when it has passed the CRC check.

Note: This parameter is relevant for advertising frame configuration and the packet type CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CIValue`

on page 116

Transmit Window Size

Sets the size of the transmit window, regarding to the start point.

Note that the scan window size is less or equal to the value of the connection interval, see "[Connection Event Interval](#)" on page 44.

Note: This parameter is relevant for advertising frame configuration and for the packet types CONNECT_REQ and CONNECTION_UPDATE_REQ.

This parameter is also indicated in the Frame Configuration Dialog.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:WSize`
on page 126

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NWSize`
on page 122

Transmit Window Offset

Sets the start point of the transmit window.

Note: This parameter is relevant for advertising frame configuration and for the packet types CONNECT_REQ and CONNECTION_UPDATE_REQ.

This parameter is also indicated in the Frame Configuration Dialog.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:WOffset`
on page 125

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:NWOffset`
on page 122

Sleep Clock Accuracy

Defines the master's clock accuracy with specified encoding. This parameter is used by the slave to determine required listening windows in the LL connection. It is a controller design parameter known by the Controller.

Note: This parameter is relevant for advertising frame configuration and the packet type CONNECT_REQ.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:SCAccuracy`
on page 123

Error Code

Sets the error code value to inform the remote device why the connection is about to be terminated in case of LL_TERMINATE_IND packet. On the other hand, this parameter for LL_REJECT_IND packet is used for the reason a request was rejected. A 8 bit value is set.

Note: This parameter is relevant for data frame configuration and the packet type LL_TERMINATE_IND and LL_REJECT_IND.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:ECODE`
on page 118

Company ID

Sets the company identifier of the manufacturer of the Bluetooth Controller. A 16 bit value is set.

Note: This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:CID`
on page 115

Version Number

Sets the version of the Bluetooth Controller specification (8 bits).

Note: This parameter is relevant for data frame configuration and the packet type LL_VERSION_IND

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:VNumber`
on page 125

Sub Version Number

Sets a unique value for each implementation or revision of an implementation of the Bluetooth Controller.

A 16 bit value is set.

Note: This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:ECOnfiguration:PCOnfiguration:SVNumber`
on page 124

Graph

The figure in the packet configuration dialog shows the packet structure of the currently selected packet type.

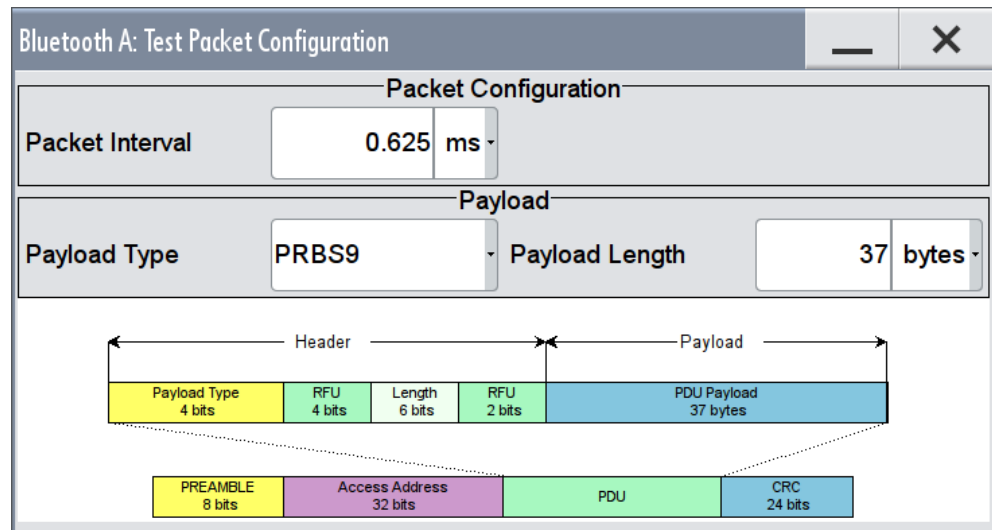
Remote command:

n.a.

4.6.5.2 Test Packet Configuration Settings

1. To access the *Bluetooth LE* test packet settings, perform the following:
2. Select "Bluetooth > General > Bluetooth Mode > Bluetooth Low Energy"
3. Select "Channel > Packet Type > TEST PACKET"

4. Select "Test Packet Configuration"



The dialog contains the parameters for configuring the test packet type. The graphic shows the distribution of the packets.



When you configure a "Dirty Transmitter Test", you also have direct access to the test packet dialog.

Packet Interval

Sets the time interval between two consecutive test packets, with regard to the starting points.

Test packet Interval

Note: This parameter is relevant for test packet types only.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:TPConfiguration:TPInterval`
on page 109

Payload Type

Selects the data source used for the payload test packets.

Note: This parameter is relevant for test packet types only.

"PRBS 9, 15" Select a PRBS-modulated data sequence (PRBS = pseudo random binary sequence) for testing.

"Pattern 1, 2, 3, 4, 5, 6"
Pattern is predefined.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:TPConfiguration:UPSource`
on page 110

Payload Length

Sets the payload length.

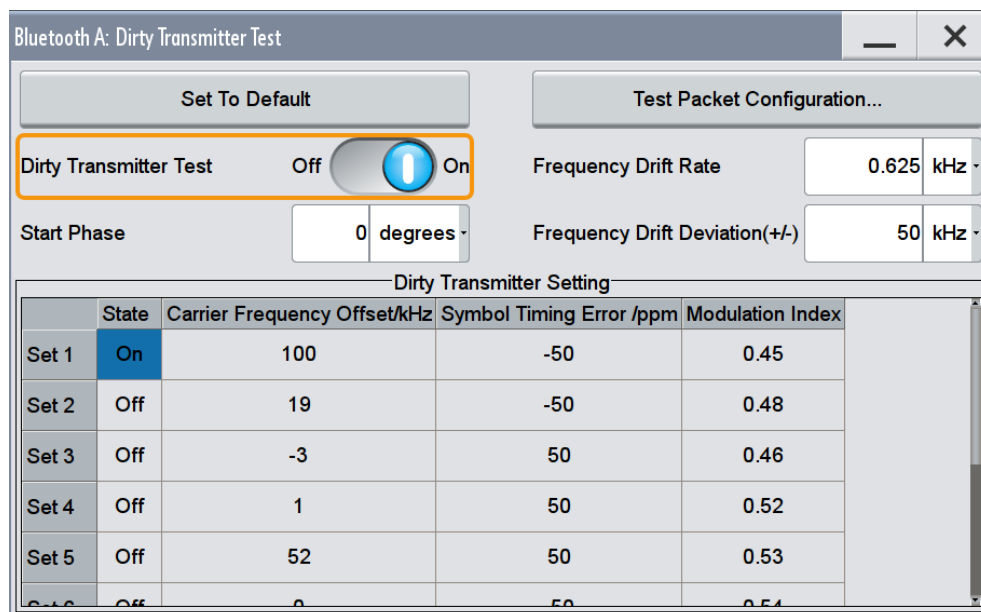
Note: This parameter is relevant for test packet types only.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:TPConfiguration:UPLength`
on page 110

4.7 Dirty Transmitter Test

- To access this dialog select "Bluetooth > General > Dirty Transmitter Test"



The dialog contains the parameters for configuring the "Dirty Transmitter Test" settings.

The Dirty Transmitter settings contain parameters which can be changed for the master signal in order to test the connection under 'dirty transmitter' conditions and define the influence on the receiver quality (bit error rate tests).

Dirty transmitter parameters according to the Bluetooth test specification (Basic Rate) are given in the table below.

Table 4-3: Dirty transmitter parameters according to the Bluetooth test specification (Basic Rate)

Set	Carrier Frequency Offset	Symbol Timing Error	Modulation Index
1	75	-20	0.28
2	14	-20	0.30
3	-2	+20	0.29
4	1	+20	0.32
5	39	+20	0.33

Set	Carrier Frequency Offset	Symbol Timing Error	Modulation Index
6	0	-20	0.34
7	-42	-20	0.29
8	74	-20	0.31
9	-19	-20	0.28
10	-75	+20	0.35

Dirty transmitter parameters according to the Bluetooth test specification (EDR) are given in the table below.

Table 4-4: Dirty transmitter parameters according to the Bluetooth test specification (EDR)

Set	Carrier Frequency Offset	Symbol Timing Error
1	0	0
2	+65	-20
3	-65	+20

Provided are the following settings:

Set to Default

Calls the default settings for the Dirty Transmitter Test.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:STDefault` on page 103

Test Packet Configuration

Access the "Test Packet Configuration" dialog for packet type "TEST PACKET", see [Chapter 4.6.5.2, "Test Packet Configuration Settings"](#), on page 56.

Dirty Transmitter Test

(Available only for packet types DH1, DH3, DH5, 2-DH1, 2-DH3, 2 - DH5, 3-DH1, 3-DH3, 3-DH5, 2-EV3, 2-EV5, 3-EV3, 3-EV5.)

Activates or deactivates the Dirty Transmitter Test.

For Basic Rate packets, each set of parameters in the "Dirty Transmitter Setting" table below is used for a duration of 20 ms. After 20 ms, the following set is used, continuing with the first set after the sequence is completed.

For EDR packets, the parameter sets apply for 20 packets each.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:DTTState` on page 102

Start Phase

Enters a start phase.

The start phase of the sine wave used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset is set here.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:DTTest:SPHase` on page 103

Frequency Drift Rate

Enters a frequency drift rate.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset with the set frequency drift rate.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:DTTest:FDRate` on page 103

Frequency Drift Deviation (+/-)

Enters a frequency drift deviation.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset. The maximum deviation reached during the drift equals the set frequency drift deviation.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:DTTest:FDDeviation` on page 102

Dirty Transmitter Setting

Indicates the dirty transmitter parameters according to the Bluetooth test specification (Basic Rate).

State ← Dirty Transmitter Setting

Activates or deactivates the corresponding parameter set.

If deactivated, the parameters are skipped in the sequence, and the next active set is used.

For Basic Rate packets, each set applies to 20ms of signal. For EDR packets, each set applies to 20 packets.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:DTTest:TABLE:LONG:SET<ch>:STATE`

on page 105

`[:SOURce<hw>] :BB:BT0oth:DTTest:TABLE:SHORT:SET<ch>:STATE`

on page 106

Carrier Frequency Offset kHz ← Dirty Transmitter Setting

Determines a carrier frequency offset.

The center frequency of the modulated RF carrier is offset by the specified value.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:DTTest:TABLE:LONG:SET<ch>:CFOffset`

on page 104

`[:SOURce<hw>] :BB:BT0oth:DTTest:TABLE:SHORT:SET<ch>:CFOffset`

on page 106

Symbol Timing Error ← Dirty Transmitter Setting

Sets the symbol timing error in ppm.

The symbol timing error modifies the symbol clock frequency by the set value

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:TABLE:LONG:SET<ch>:STERror`

on page 105

`[:SOURCE<hw>] :BB:BT0oth:DTTest:TABLE:SHORT:SET<ch>:STERror`

on page 107

Modulation Index ← Dirty Transmitter Setting

(Only for Basic Rate Packets)

Sets the modulation index.

The modulation index **h** specifies the frequency deviation, defined as:

$$h = \frac{2\Delta f}{f_{symbol}}$$

where f_{symbol} is the "symbol rate" and Δf is the "frequency deviation".

According to the Bluetooth standard, the modulation index may vary between 0.28 and 0.35.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:DTTest:TABLE:LONG:SET<ch>:MINdex`

on page 104

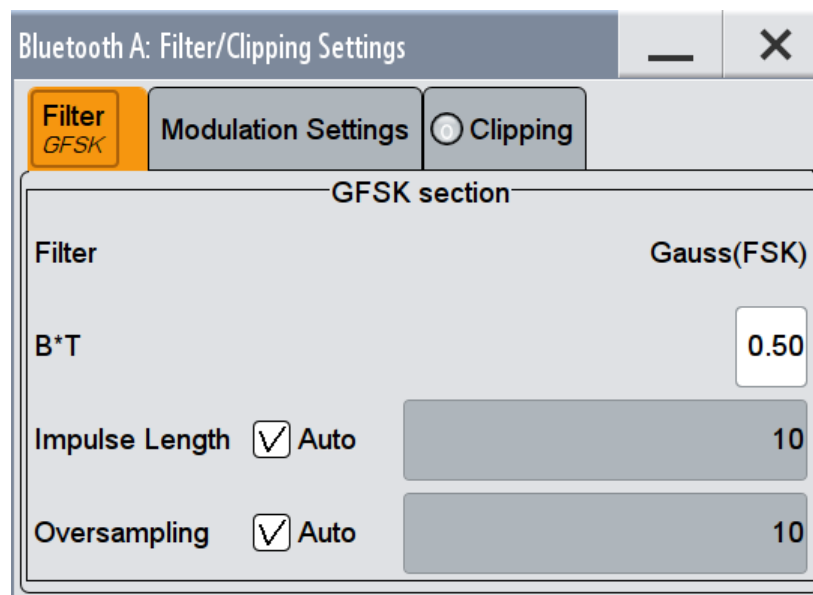
4.8 Filter/Clipping Settings

- ▶ To access filter / clipping settings, select "Baseband > Bluetooth > General > Filter/Clipping".

The dialog comprises the settings, necessary to configure the baseband filter, the modulation settings and to enable clipping.

4.8.1 Filter Settings

- ▶ To access the filter settings, select "Filter".



The dialog contains the parameters required for configuring the baseband filter.

Filter

Indicates the filter used for GFSK part.

Remote command:

n.a.

Roll Off Factor / B xT

Sets the filter parameter.

Sets the filter parameter.

The filter parameter offered ("Roll Off Factor" or "BxT") depends on the currently selected filter type. This parameter is preset to the default for each of the predefined filters.

Remote command:

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:APCO25](#) on page 78

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:COsine](#) on page 78

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:FGAuss](#) on page 78

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:GAUSSs](#) on page 78

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:PGAuss](#) on page 79

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:RCOSine](#) on page 79

[\[:SOURCE<hw>\]:BB:BT00th:FILTer:PARAmeter:SPHase](#) on page 80

Cut Off Frequency Factor

(available for filter parameter Lowpass only)

Sets the value for the cut off frequency factor. The cut off frequency of the filter can be adjusted to reach spectrum mask requirements.

Remote command:

[:SOURce<hw>] :BB:BT0oth:FILTer:PARAmeter:LPASs on page 79

Impulse Length

Displays the number of filter tabs. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

Remote command:

[:SOURce<hw>] :BB:BT0oth:FILTer:ILENgtH:AUTO [:STATe] on page 76

[:SOURce<hw>] :BB:BT0oth:FILTer:ILENgtH on page 76

Oversampling

Determines the upsampling factor. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

Remote command:

[:SOURce<hw>] :BB:BT0oth:FILTer:OSAMpLing:AUTO [:STATe] on page 77

[:SOURce<hw>] :BB:BT0oth:FILTer:OSAMpLing on page 76

Filter ($\pi/4$ DQPSK section)

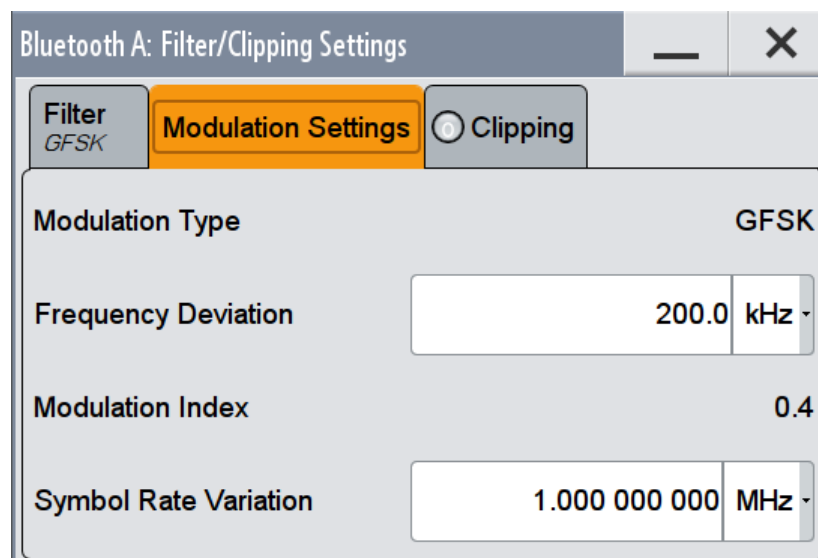
Selects the filter used for DQPSK/8DPSK sections with EDR packets.

Remote command:

[:SOURce<hw>] :BB:BT0oth:FILTer:TYPE on page 75

4.8.2 Modulation Settings

- To access this dialog select "Baseband > Bluetooth > General > Filter / Clipping > Modulation Settings".



The dialog comprises the settings, necessary to configure the modulation settings.

Modulation type

Displays the modulation type used for the current packet selection.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:FILTer:MTYPE` on page 77

Frequency deviation

Enter the frequency deviation of the frequency modulated part.

The frequency deviation can be varied in a range from 100.0 kHz to 200.0 kHz according to Bluetooth specification.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:MSEttings:FDEVIation` on page 77

Modulation index

Displays the modulation index resulting from the entered frequency deviation value.

Modulation index is calculated from the given frequency deviation and symbol rate values.

The modulation index **h** is defined as:

$$h = \frac{2\Delta f}{f_{symbol}}$$

where f_{symbol} is the "symbol rate" and Δf is the "frequency deviation".

According to the Bluetooth standard, the modulation index is allowed to vary between 0.28 and 0.35.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:FILTer:MINDEX` on page 77

Symbol Rate Variation

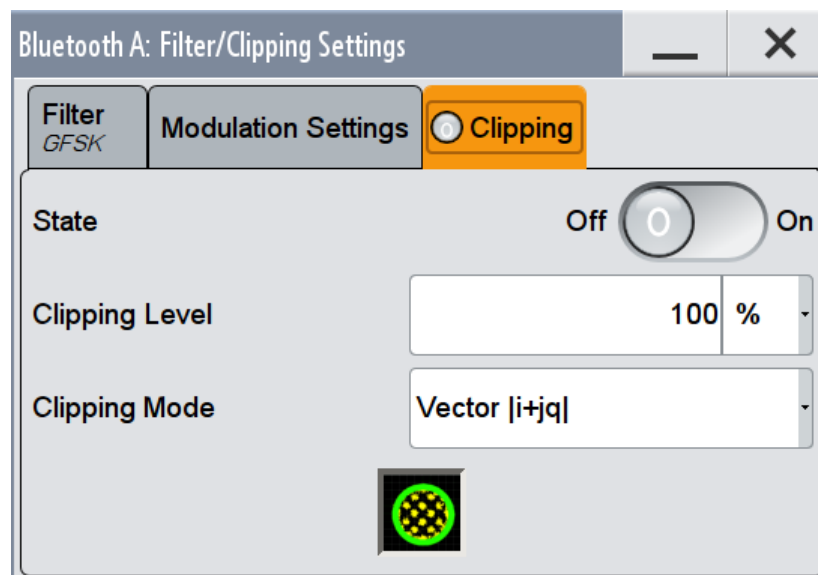
Enter the symbol rate.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:SRATE:VARIation` on page 80

4.8.3 Clipping Settings

- ▶ To access the clipping settings, select "Baseband > Bluetooth > General > Filter / Clipping > Clipping".



The dialog comprises the settings, necessary to configure the clipping.

Clipping State

Switches baseband clipping on and off.

Baseband clipping is a very simple and effective way of reducing the crest factor of the signal. Since clipping is done prior to filtering, the procedure does not influence the spectrum. The EVM however increases.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:CLIPping:STATe` on page 75

Clipping Level

Sets the limit for clipping.

This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

Remote command:

`[:SOURCE<hw>] :BB:BT0oth:CLIPping:LEVel` on page 74

Clipping Mode

Selects the clipping method. A graphic illustration of the way in which these two methods work is given in the dialog.

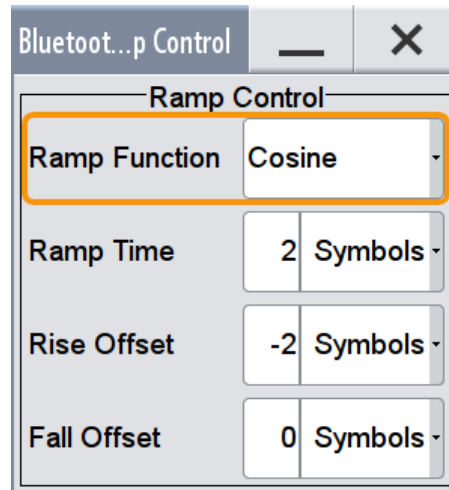
- "Vector $|i + jq|$ "
The limit is related to the amplitude $|i + jq|$. The I and Q components are mapped together, the angle is retained.
- "Scalar $|i|, |q|$ "
The limit is related to the absolute maximum of all the I and Q values $|i| + |q|$. The I and Q components are mapped separately, the angle changes.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:CLIPping:MODE](#) on page 75

4.9 Power Ramping Settings

- ▶ To access this dialog select "Bluetooth > General > Power Ramping"



The dialog comprises the settings, necessary to configure power ramping.

Ramp Function

Selects the form of the transmitted power, i.e. the shape of the rising and falling edges during power ramp control.

"Linear" The transmitted power rises and falls with linear fashion.

"Cosine" The transmitted power rises and falls with a cosine-shaped edge. This gives rise to a more favorable spectrum than the Linear setting.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PRAMPing:RFUNction](#) on page 92

Ramp Time

Sets the power ramping rise time and fall time for a burst.

Remote command:

[\[:SOURCE<hw>\]:BB:BT0oth:PRAMPing:RTIME](#) on page 93

Rise Offset

Sets the offset in the rising edge of the envelope at the start of a burst. A positive value moves the ramp into the beginning of a transmitted packet and a negative value introduces an additional guard period before the start of the packet.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:PRAMping:ROFFset` on page 93

Fall Offset

Sets the offset of the falling edge of the envelope at the end of a burst. A positive value introduces a guard period after the end of the packet and a negative value moves the ramp into the end part of the transmitted packet.

Remote command:

`[:SOURce<hw>] :BB:BT0oth:PRAMping:FOFFset` on page 92

5 Remote-Control Commands

The following commands are required to perform signal generation with the Bluetooth options in a remote environment. We assume that the R&S SMW has already been set up for remote operation in a network as described in the R&S SMW documentation. A knowledge about the remote control operation and the SCPI command syntax are assumed.



Conventions used in SCPI command descriptions

For a description of the conventions used in the remote command descriptions, see section "Remote Control Commands" in the R&S SMW user manual.

The commands in the `SOURCE:BB:BT00th` subsystem are described in three sections, separated into general remote commands, commands for Packet Configuration settings and commands for Dirty Transmitter Test settings.

This subsystem contains commands for the primary and general settings of the Bluetooth standard. These settings concern activation and deactivation of the standard, setting filter, clock, trigger and clipping settings, defining the symbol rate variation and the sequence length, as well as the preset and power adjust setting.

Common Suffixes

The following common suffixes are used in remote commands:

Suffix	Value range	Description
ENTity<ch>	1 .. 4	entity in a multiple entity configuration with separate baseband sources ENTity3 4 require option R&S SMW-K76
SOURCE<hw>	[1] 4	available baseband signals only SOURCE1 possible, if the keyword ENTity is used
OUTPut<ch>	1 .. 3	available markers



Using SCPI command aliases for advanced mode with multiple entities

You can address multiple entities configurations by using the SCPI commands starting with the keyword `SOURCE` or the alias commands starting with the keyword `ENTity`.

Note that the meaning of the keyword `SOURCE<hw>` changes in the second case.

For details, see section "SCPI Command Aliases for Advanced Mode with Multiple Entities" in the R&S SMW user manual.

The following commands specific to the Bluetooth are described here:

5.1 General Commands

<code>[:SOURce<hw>]:BB:BT0oth:BCText?</code>	69
<code>[:SOURce<hw>]:BB:BT0oth:CTYPE</code>	69
<code>[:SOURce<hw>]:BB:BT0oth:PRESet</code>	70
<code>[:SOURce<hw>]:BB:BT0oth:PTYPE</code>	70
<code>[:SOURce<hw>]:BB:BT0oth:SETTing:CATalog</code>	71
<code>[:SOURce<hw>]:BB:BT0oth:SETTing:DELeTe</code>	71
<code>[:SOURce<hw>]:BB:BT0oth:SETTing:LOAD</code>	71
<code>[:SOURce<hw>]:BB:BT0oth:SETTing:STORE</code>	72
<code>[:SOURce<hw>]:BB:BT0oth:SLENgth</code>	72
<code>[:SOURce<hw>]:BB:BT0oth:STATe</code>	72
<code>[:SOURce<hw>]:BB:BT0oth:STIMing</code>	72
<code>[:SOURce<hw>]:BB:BT0oth:TMODe</code>	73
<code>[:SOURce<hw>]:BB:BT0oth:VERSion?</code>	73
<code>[:SOURce<hw>]:BB:BT0oth:WAVeform:CREate</code>	74

`[:SOURce<hw>]:BB:BT0oth:BCText?`

Queries the state/roll of the controller.

Return values:

<code><BcText></code>	string
	Connected (for data channel type) Shows that the state is Connected.
	Advertiser (for advertising channel type) the current bluetooth Controller Role is Advertiser for all ADV-xxx packet types and SCAN_RSP.
	Scanner (for advertising channel type) the current bluetooth Controller Role is Scanner for SCAN_REQ packet type
	Initiator (for advertising channel type) the current bluetooth Controller Role is Initiator for CONNECT_REQ packet type

Example: `SOUR:BB:BT0:BC?`
Queries the state/roll of the controller.

Usage: Query only

Manual operation: See "[Bluetooth Controller State](#)" on page 40

`[:SOURce<hw>]:BB:BT0oth:CTYPE <CType>`

Determines the channel type. Advertising and data are available.

Parameters:

<CType> ADVertising | DATA

ADVertising

Selects channel type Advertising.

DATA

Selects channel type Data. Devices in a connected state transmit data channel packets in connection events with a start point and an interval.

*RST: ADVertising

Example:

```
SOUR:BB:BTO:CTYP ADV
channel type Advertising.
SOUR:BB:BTO:CTYP DATA
channel type Data.
```

Manual operation: See "[Channel Type](#)" on page 38**[:SOURce<hw>]:BB:BT0oth:PRESet**

Sets the parameters of the digital standard to their default values (*RST values specified for the commands).

Not affected is the state set with the command `SOURce<hw>:BB:BT0oth:STATE``[:SOURce<hw>]:BB:BT0oth:STATE`**Example:**

```
BB:BTO:PRES
resets all the Bluetooth settings to default values.
```

Usage:

Event

Manual operation: See "[Set To Default](#)" on page 19**[:SOURce<hw>]:BB:BT0oth:PTYPe <PType>**

The available packets depend on the selected transport mode. All packet types as defined in the Bluetooth specifications are supported.

Parameters:

<PType> ID | NULL | POLL | FHS | DM1 | DH1 | DM3 | DH3 | DM5 | DH5 | AUX1 | ADH1 | ADH3 | ADH5 | AEDH1 | AEDH3 | AEDH5 | HV1 | HV2 | HV3 | DV | EV3 | EV4 | EV5 | EEV3 | EEV5 | EEEV3 | EEEV5

*RST: DH1

Example:

```
BB:BTO:PTYP NULL
sets the packet type.
```

Manual operation: See "[Packet Type](#)" on page 31

[[:SOURce<hw>]:BB:BTOoth:SETTing:CATalog <Catalog>

This command reads out the files with Bluetooth settings in the default directory. The default directory is set using command `M MEM:CDIRectory`. Only files with the file extension `*.bto` will be listed.

Parameters:

<Catalog> string

Example:

```
M MEM:CDIR '/var/user/temp/bluetooth'
sets the default directory to /var/user/temp/bluetooth.
BB:BTO:SETT:CAT?
reads out all the files with Bluetooth settings in the default direc-
tory.
Response: 'Bluetooth_EDR', 'Bluetooth_SCO'
the files Bluetooth_EDR and Bluetooth_SCO are available.
```

Manual operation: See ["Save/Recall"](#) on page 19

[[:SOURce<hw>]:BB:BTOoth:SETTing:DELete <Filename>

This command deletes the selected file with Bluetooth settings. The directory is set using command `M MEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.bto` will be deleted.

Parameters:

<Filename> string

Example:

```
BB:BTO:SETT:DEL '/var/user/temp/bluetooth'
deletes the specified file with Bluetooth settings.
```

Manual operation: See ["Save/Recall"](#) on page 19

[[:SOURce<hw>]:BB:BTOoth:SETTing:LOAD <Filename>

This command loads the selected file with Bluetooth settings. The directory is set using command `M MEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.bto` will be loaded.

Parameters:

<Filename> string

Example:

```
BB:BTO:SETT:LOAD 'bluetooth_1'
loads file bluetooth_1.
```

Manual operation: See ["Save/Recall"](#) on page 19

[[:SOURce<hw>]:BB:BT0oth:SETTING:STORE <Filename>

This command stores the current Bluetooth settings into the selected file. The directory is set using command `MMEM:CDIRECTORY`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. Bluetooth settings are stored as files with the specific file extensions `*.bto`.

Setting parameters:

<Filename> string

Example: `BB:BT0:SETT:STOR 'bluetooth_1'`
stores the current Bluetooth settings into file `bluetooth_1`.

Usage: Setting only

Manual operation: See "[Save/Recall](#)" on page 19

[[:SOURce<hw>]:BB:BT0oth:SLENGth <SLength>

Sets the sequence length of the Bluetooth signal in number of frames. This signal is calculated in advance and output in the arbitrary waveform generator.

Parameters:

<SLength> integer

Range: depending on the number of states in dirty transmitter test to dynamics

*RST: 1

Example: `BB:BT0:SLEN 10`
sets the sequence length to 10 frames.

Manual operation: See "[Sequence Length](#)" on page 32

[[:SOURce<hw>]:BB:BT0oth:STATe <State>

Activates the standard and deactivates all the other digital standards and digital modulation modes in the same path.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `SOURce1:BB:BT0oth:STATe`

Manual operation: See "[State](#)" on page 19

[[:SOURce<hw>]:BB:BT0oth:STIMing <SlotTiming>

Selects the Rx slot timing mode.

Parameters:

<SlotTiming> TX | LOOPback

*RST: TX

Example: BB:BTO:PTYP DH3
sets the packet type.
BB:BTO:STIM LOOP
selects loopback test mode.

Manual operation: See "[Slot Timing](#)" on page 32

[:SOURce<hw>]:BB:BT0oth:TMODe <TMode>

The command selects the transport mode.

Parameters:

<TMode> ACL | SCO | ESCO

ACL

The selected transport mode is used for a point-to-point multi-point link establishment between the master and all the slaves participating on the piconet.

SCO

The selected transport mode is used for a point-to-point link establishment between a master and a single slave in the piconet.

ESCO

The selected transport mode is used for a symmetric or asymmetric point-to-point link establishment between a master and a specific slave.

*RST: ACL

Example: BB:BTO:TMOD ACL
selects transport mode ACL.

Manual operation: See "[Transport Mode](#)" on page 20

[:SOURce<hw>]:BB:BT0oth:VERSion?

Queries the version of the Bluetooth standard underlying the definitions.

Return values:

<Version> string

Example: BB:BTO:VERS?
queries the Bluetooth version.
Response: Version 2.1+EDR

Usage: Query only

Manual operation: See "[Bluetooth Version](#)" on page 20

[[:SOURce<hw>]:BB:BT0oth:WAVeform:CREate <Filename>

This command creates a waveform using the current settings of the "Bluetooth" menu. The file name is entered with the command. The file is stored with the predefined file extension * .wv. The file name and the directory it is stored in are user-definable.

Setting parameters:

<Filename> string

Example:

```
MMEM:CDIR '/var/user/temp/waveform'
sets the default directory to /var/user/temp/waveform.
BB:BT0:WAV:CRE 'bluetooth_1'
creates the waveform file bluetooth.wv in the default directory.
```

Usage: Setting only

Manual operation: See "Generate Waveform" on page 20

5.2 Filter/Clipping Settings

[[:SOURce<hw>]:BB:BT0oth:CLIPping:LEVel.....	74
[[:SOURce<hw>]:BB:BT0oth:CLIPping:MODE.....	75
[[:SOURce<hw>]:BB:BT0oth:CLIPping:STATe.....	75
[[:SOURce<hw>]:BB:BT0oth:FILTer:TYPE.....	75
[[:SOURce<hw>]:BB:BT0oth:FILTer:ILENght.....	76
[[:SOURce<hw>]:BB:BT0oth:FILTer:ILENght:AUTO[:STATe].....	76
[[:SOURce<hw>]:BB:BT0oth:FILTer:OSAMpling.....	76
[[:SOURce<hw>]:BB:BT0oth:FILTer:OSAMpling:AUTO[:STATe].....	77
[[:SOURce<hw>]:BB:BT0oth:FILTer:MINDEX.....	77
[[:SOURce<hw>]:BB:BT0oth:FILTer:MTYPE.....	77
[[:SOURce<hw>]:BB:BT0oth:MSETtings:FDEVIation.....	77
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:APCO25.....	78
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:COSine.....	78
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:FGAuss.....	78
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:GAUSS.....	78
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:LPASS.....	79
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:PGAuss.....	79
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:RCOSine.....	79
[[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:SPHase.....	80
[[:SOURce<hw>]:BB:BT0oth:SRATe:VARiation.....	80

[[:SOURce<hw>]:BB:BT0oth:CLIPping:LEVel <Level>

Sets the limit for level clipping (Clipping). This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

Parameters:

<Level> integer
 Range: 1 to 100
 Increment: 1
 *RST: 100
 Default unit: PCT

Example:

```
BB:BTO:CLIP:LEV 80
sets the limit for level clipping to 80% of the maximum level.
BB:BTO:CLIP:STAT ON
activates level clipping.
```

Manual operation: See "[Clipping Level](#)" on page 65

[[:SOURce<hw>]:BB:BTOoth:CLIPping:MODE <Mode>

The command sets the method for level clipping (Clipping).

Parameters:

<Mode> VECTor | SCALar
VECTor
 The reference level is the amplitude | i+jq |.
SCALar
 The reference level is the absolute maximum of the I and Q values.
 *RST: VECTor

Example:

```
BB:BTO:CLIP:MODE VECT
sets the amplitude as reference level.
```

Manual operation: See "[Clipping Mode](#)" on page 65

[[:SOURce<hw>]:BB:BTOoth:CLIPping:STATe <State>

The command activates level clipping (Clipping). The value is defined with the command `BB:BTO:CLIPping:LEVel`, the mode of calculation with the command `BB:BTO:CLIPping:MODE`.

Parameters:

<State> 0 | 1 | OFF | ON
 *RST: OFF

Example:

```
BB:BTO:CLIP:STAT ON
activates level clipping.
```

Manual operation: See "[Clipping State](#)" on page 65

[[:SOURce<hw>]:BB:BTOoth:FILTer:TYPE <Type>

Selects the filters used for $\pi/4$ DQPSK and 8DPSK modulations. This opens a selection window containing all the filters available to the instrument.

Parameters:

<Type> RCOSine | COSine | GAUSs | LGAuss | CONE | COF705 |
 COEqualizer | COFequalizer | C2K3x | APCO25 | SPHase |
 RECTangle | PGAuss | LPASs | DIRac | ENPShape |
 EWPSshape
 *RST: RCOSine

Example:

BB:BTO:FILT:TYPE RCOS
 sets the filter type RCOSine.

Manual operation: See "[Filter \(\$\pi/4\$ DQPSK section\)](#)" on page 63

[:SOURce<hw>]:BB:BTOoth:FILT:ILENth <ILength>

Sets the impulse length (the number of filter taps).

Parameters:

<ILength> integer
 Range: 1 to depends on oversampling
 *RST: 10

Example:

BB:BTO:FILT:ILEN 10
 sets the number of filter tabs to 10.

Manual operation: See "[Impulse Length](#)" on page 63

[:SOURce<hw>]:BB:BTOoth:FILT:ILENth:AUTO[:STATE] <State>

The command activates/deactivates the impulse length state. If activated, the most sensible parameter values are selected. The value depends on the coherence check.

Parameters:

<State> 0 | 1 | OFF | ON
 *RST: ON

Example:

BB:BTO:FILT:ILEN:AUTO ON
 the most sensible parameters are selected automatically.

Manual operation: See "[Impulse Length](#)" on page 63

[:SOURce<hw>]:BB:BTOoth:FILT:OSAMpling <OSampling>

The command sets the upsampling factor.

Parameters:

<OSampling> integer
 Range: 1 to 32
 *RST: 10

Example:

BB:BTO:FILT:OSAM 10
 sets the upsampling factor to 10.

Manual operation: See "[Oversampling](#)" on page 63

Manual operation: See ["Frequency deviation"](#) on page 64

[[:SOURce<hw>]:BB:BTOoth:FILTer:PARAmeter:APCO25 <Apco25>

Sets the roll-off factor for filter type APCO25.

Parameters:

<Apco25> float
 Range: 0.05 to 0.99
 Increment: 0.01
 *RST: 0.2

Example: BB:BTO:FILT:PAR:APCO25 0.2
 sets the roll-off factor to 0.2 for filter type APCO25.

Manual operation: See ["Roll Off Factor / B xT"](#) on page 62

[[:SOURce<hw>]:BB:BTOoth:FILTer:PARAmeter:COSSine <Cosine>

Sets the roll-off factor for the Cosine filter type.

Parameters:

<Cosine> float
 Range: 0 to 1.0
 Increment: 0.01
 *RST: 0.1

Example: BB:BTO:FILT:PAR:COSS 0.35
 sets the roll-off factor to 0.35 for filter type Cosine.

Manual operation: See ["Roll Off Factor / B xT"](#) on page 62

[[:SOURce<hw>]:BB:BTOoth:FILTer:PARAmeter:FGAUSS <FGauss>

Sets the B x T for the Gauss filter type.

Parameters:

<FGauss> float
 Range: 0.15 to 2.5
 Increment: 0.01
 *RST: 0.5

Example: BB:BTO:FILT:PAR:FGA 0.5
 sets B x T to 0.5 for the Gauss filter type for the GFSK section of the packet.

Manual operation: See ["Roll Off Factor / B xT"](#) on page 62

[[:SOURce<hw>]:BB:BTOoth:FILTer:PARAmeter:GAUSS <Gauss>

Sets the B x T for the Gauss filter type.

Parameters:

<Gauss> float
 Range: 0.15 to 2.5
 Increment: 0.01
 *RST: 0.5

Example:

BB:BTO:FILT:PAR:GAUS 0.5
 sets B x T to 0.5 for the Gauss filter type for $\pi/4$ DQPSK or 8DPSK sections.

Manual operation: See "[Roll Off Factor / B x T](#)" on page 62

[[:SOURce<hw>]:BB:BTOoth:FILT:PAR:LPASSs <LPass>

Sets the cut off frequency factor for a lowpass filter (ACP Opt.).

Parameters:

<LPass> float
 Range: 0.05 to 2
 Increment: 0.01
 *RST: 0.5

Example:

BB:BTO:FILT:PAR:LPAS 1
 sets the cut off frequency factor for a lowpass filter

Manual operation: See "[Cut Off Frequency Factor](#)" on page 62

[[:SOURce<hw>]:BB:BTOoth:FILT:PAR:PGAuss <PGauss>

Sets the B x T for the Pure Gauss filter type.

Parameters:

<PGauss> float
 Range: 0.15 to 2.5
 Increment: 0.01
 *RST: 0.5

Example:

BB:BTO:FILT:PAR:PGA 0.5
 sets B x T to 0.5 for the Pure Gauss filter type.

Manual operation: See "[Roll Off Factor / B x T](#)" on page 62

[[:SOURce<hw>]:BB:BTOoth:FILT:PAR:RCOSine <RCosine>

Sets the roll-off factor for the Root Cosine filter type.

Parameters:

<RCosine> float
 Range: 0 to 1.0
 Increment: 0.01
 *RST: 0.4

Example: `BB:BTO:FILT:PAR:RCOS 0.22`
sets the roll-off factor to 0.22 for filter type Root Cosine.

Manual operation: See ["Roll Off Factor / B x T"](#) on page 62

[[:SOURce<hw>]:BB:BTOoth:FILT:PAR:SPHase <SPhase>

Sets the B x T for the Split Phase filter type.

Parameters:

<SPhase> float
Range: 0.15 to 2.5
Increment: 0.01
*RST: 2

Example: `BB:BTO:FILT:PAR:SPH 0.5`
sets B x T to 0.5 for the Split Phase filter type.

Manual operation: See ["Roll Off Factor / B x T"](#) on page 62

[[:SOURce<hw>]:BB:BTOoth:SRATE:VARIation <Variation>

Sets the symbol rate.

Parameters:

<Variation> float
Range: 4E2 to 15E6
Increment: 1E-3
*RST: 1E6

Example: `BB:BTO:SRAT:VAR 1`
sets the symbol rate variation to 1 MHz.

Manual operation: See ["Symbol Rate Variation"](#) on page 64

5.3 Trigger Settings

This section lists the remote control commands, necessary to configure the trigger.

[:SOURce<hw>]:BB:BTOoth:TRIGger:ARM:EXECute.....	81
[:SOURce<hw>]:BB:BTOoth:TRIGger:EXECute.....	81
[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal]:SYNChronize:OUTPut.....	81
[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASeband:DELay.....	82
[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASeband:INHibit.....	82
[:SOURce<hw>]:BB:BTOoth:TRIGger:RMODE.....	83
[:SOURce<hw>]:BB:BTOoth:TRIGger:SLENGth.....	83
[:SOURce<hw>]:BB:BTOoth:TRIGger:SLUNit.....	84
[:SOURce<hw>]:BB:BTOoth:TRIGger:SOURce.....	84

<code>[:SOURce<hw>]:BB:BT0oth:TRIGger[:EXTErnal]:DELay</code>	85
<code>[:SOURce<hw>]:BB:BT0oth:TRIGger[:EXTErnal]:INHibit</code>	86
<code>[:SOURce<hw>]:BB:BT0oth:TRIGger[:SEQuence</code>	86

`[:SOURce<hw>]:BB:BT0oth:TRIGger:ARM:EXECute`

The command stops signal generation for trigger modes "Armed Auto" and "Armed Retrigger". A subsequent internal or external trigger event restart signal generation.

Example: `BB:BT0:TRIG:ARM:EXEC`
stops signal generation for trigger modes "Armed Auto "and "Armed Retrigger".

Usage: Event

Manual operation: See "Arm" on page 24

`[:SOURce<hw>]:BB:BT0oth:TRIGger:EXECute`

The command executes a trigger. The internal trigger source must be selected using the command `SOUR:BB:BT0:TRIG:SOUR INT` and a trigger mode other than "AUTO" must be selected using the command `SOUR:BB:BT0:TRIG:SEQ`.

Example: `BB:BT0:TRIG:SOUR INT`
sets internal triggering.
`BB:BT0:SEQ RETR`
sets Retrigger mode, i.e. every trigger event causes signal generation to restart.
`BB:BT0:TRIG:EXEC`
executes a trigger.

Usage: Event

Manual operation: See "Execute Trigger" on page 24

`[:SOURce<hw>]:BB:BT0oth:TRIGger[:EXTErnal]:SYNChronize:OUTPut <Output>`

(enabled for "Trigger Source" External)

Enables/disables output of the signal synchronous to the external trigger event.

Parameters:

<Output> 0 | 1 | OFF | ON

ON

The signal calculation starts simultaneously with the external trigger event but because of the instrument's processing time the first samples are cut off and no signal is outputted. After elapsing of the internal processing time, the output signal is synchronous to the trigger event.

OFF

The signal output begins after elapsing of the processing time and starts with sample 0, i.e. the complete signal is outputted.

*RST: ON

Example:

```
BB:BTO:TRIG:SOUR EXT
```

sets external triggering.

```
BB:BTO:TRIG:EXT:SYNC:OUTP ON
```

enables synchronous output to external trigger

Manual operation: See "[Sync. Output to External Trigger](#)" on page 24

[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASband:DELay <Delay>

Specifies the trigger delay (expressed as a number of samples) for triggering by the trigger signal from the second path.

Parameters:

<Delay> float

Range: 0 to 65535

Increment: 0.01

*RST: 0

Example:

```
BB:BTO:TRIG:SOUR OBAS
```

sets for path A the internal trigger executed by the trigger signal from the second path (path B).

```
BB:BTO:TRIG:OBAS:DEL 50
```

sets a delay of 50 symbols for the trigger.

Manual operation: See "[Trigger Delay](#)" on page 26

[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASband:INHibit <Inhibit>

The command specifies the number of samples by which a restart is to be inhibited following a trigger event. This command applies only for triggering by the second path.

Parameters:

<Inhibit> integer

Range: 0 samples to $(1 \ll 26) - 1$ u samples

Increment: 1 sample

*RST: 0 samples

Example: `BB:BTO:TRIG:SOUR OBAS`
 sets for path A the internal trigger executed by the trigger signal from the second path (path B).
`BB:BTO:TRIG:OBAS:INH 200`
 sets a restart inhibit for 200 samples following a trigger event.

Manual operation: See "[External Trigger Inhibit](#)" on page 25

[:SOURce<hw>]:BB:BTOoth:TRIGger:RMODe <RMode>

Queries the current status of signal generation for all trigger modes with Bluetooth modulation on.

Parameters:

<RMode> RUN | STOP

RUN

the signal is generated. A trigger event occurred in the triggered mode.

STOP

the signal is not generated. A trigger event did not occur in the triggered modes, or signal generation was stopped by the command `:BB:BTO:TRIG:ARM:EXECute` (armed trigger modes only).

*RST: STOP

Example: `BB:BTO:SEQ ARET`
 selects the Armed_Retrigger mode.
`BB:BTO:TRIG:RMOD?`
 queries the current status of signal generation.
 Response: RUN
 the signal is generated, an external trigger was executed.

Manual operation: See "[Running/Stopped](#)" on page 23

[:SOURce<hw>]:BB:BTOoth:TRIGger:SLENgth <SLength>

Defines the length of the signal sequence to be output in the Single trigger mode (`SOUR:BB:BTO:SEQ SING`). The unit is defined with command `SOUR:BB:BTO:TRIG:SLUNit`.

Parameters:

<SLength> integer

Range: 1 to 7000

*RST: 1

Example: `BB:BTO:SEQ SING`
 sets trigger mode Single.
`BB:BTO:TRIG:SLUN FRAM`
 sets unit frames for the entry of sequence length.
`BB:BTO:TRIG:SLEN 200`
 sets a sequence length of 200 frames.

Manual operation: See ["Trigger Signal Duration"](#) on page 23

[[:SOURce<hw>]:BB:BTOoth:TRIGger:SLUNit <SIUnit>

Defines the unit for the entry of the length of the signal sequence (SOUR:BB:BTO:TRIG:SLEN) to be output in the "Single" trigger mode (SOUR:BB:BTO:SEQ SING).

Parameters:

<SIUnit> FRAME | SEQUENCE | EVENT
 *RST: SEQUENCE

Example:

BB:BTO:SEQ SING
 sets trigger mode Single.
 BB:BTO:TRIG:SLUN FRAM
 sets unit frames for the entry of sequence length.
 BB:BTO:TRIG:SLEN 2
 sets a sequence length of 2 frames. The current frame will be output twice after the next trigger event.

Manual operation: See ["Signal Duration Unit"](#) on page 23

[[:SOURce<hw>]:BB:BTOoth:TRIGger:SOURce <Source>

Selects the trigger signal source and determines the way the triggering is executed. Provided are internal triggering by means of a command, external trigger signal via one of the provided local or global connectors and triggering by a signal from the other paths.

Parameters:

<Source> INTB | INTernal | OBASeband | EGT1 | EGT2 | EGC1 | EGC2 | ELTRigger | INTA | ELClock | BEXternal | EXternal

INTernal

Internal

INTA | INTB

Internal trigger from the other baseband

EGT1 | EGT2

External global trigger

EGC1 | EGC2

External global clock

ELTRigger

External local trigger

ELCLock

External local clock

OBASeband|BEXternal|EXternal

Provided only for backward compatibility with other R&S signal generators.

The R&S SMW accepts these values und maps them automatically as follow:

EXternal = EGT1, BEXternal = EGT2, OBASeband = INTA or INTB (depending on the current baseband)

*RST: INTernal

Example:

```
BB:BTO:TRIG:SOUR INT
selects an internal trigger source
```

Manual operation: See "[Trigger Source](#)" on page 24

```
[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXternal]:DELay <Delay>
```

Sets the trigger delay.

Parameters:

<Delay> float
 Range: 0 to 65535
 Increment: 0.01
 *RST: 0

Example:

```
BB:BTO:TRIG:SOUR EXT
sets an external trigger.
BB:BTO:TRIG:EXT:DEL 50
sets a delay of 50 symbols for the trigger.
```

Manual operation: See "[Trigger Delay](#)" on page 26

[[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal]:INHibit <Inhibit>

Specifies the number of samples by which a restart is to be inhibited following an external trigger event.

Parameters:

<Inhibit> integer
 Range: 0 to 21.47*symbRate
 *RST: 0

Example:

BB:BTO:TRIG:SOUR EXT
 selects an external trigger.
 BB:BTO:TRIG:EXT:INH 200
 sets a restart inhibit for 200 samples following a trigger event.

Manual operation: See "[External Trigger Inhibit](#)" on page 25

[[:SOURce<hw>]:BB:BTOoth[:TRIGger]:SEQUence <Sequence>

The command selects the trigger mode.

Parameters:

<Sequence> AUTO | RETRigger | AAUTO | ARETrigger | SINGLE

AUTO

The modulation signal is generated continuously.

RETRigger

The modulation signal is generated continuously. A trigger event (internal or external) causes a restart.

AAUTO

The modulation signal is generated only when a trigger event occurs. After the trigger event the signal is generated continuously, signal generation is stopped with command SOUR:BB:BTO:TRIG:ARM:EXEC and started again when a trigger event occurs.

ARETrigger

The modulation signal is generated only when a trigger event occurs. The device automatically toggles to RETRIG mode.

Every subsequent trigger event causes a restart.

Signal generation is stopped with command

SOUR:BB:BTO:TRIG:ARM:EXEC and started again when a trigger event occurs.

SINGLE

The modulation signal is generated only when a trigger) event occurs. After the trigger event, the signal is generated once to the set sequence length (SOUR:BB:BTO:TRIG:SLEN). Every subsequent trigger event causes a restart.

*RST: AUTO

Example: `BB:BTO:SEQ AAUT`
sets the "Armed_auto" trigger mode; the device waits for the first trigger (e.g. with *TRG) and then generates the signal continuously.

Manual operation: See "Trigger Mode" on page 23

5.4 Marker Settings

This section lists the remote control commands, necessary to configure the markers.

OUTPut<ch>

The numeric suffix to **OUTPut** distinguishes between the available markers.

<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut:DELay:FIXed</code>	87
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:DELay</code>	88
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:DELay:MINimum?</code>	88
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:DELay:MAXimum?</code>	88
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:MODE</code>	88
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:ONTime</code>	89
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:OFFTime</code>	89
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:PATTern</code>	90
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:PULSe:DIVider</code>	90
<code>[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:PULSe:FREquency?</code>	90

[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut:DELay:FIXed <Fixed>

The command restricts the marker delay setting range to the current range. In this range the delay can be set without restarting the marker and signal. If a delay is entered in setting ON but is outside this range, the maximum possible delay is set and an error message is generated.

The numeric suffix in **OUTPut** has no significance for this command, since the setting always affects every marker.

Parameters:

<Fixed> 0 | 1 | OFF | ON
*RST: OFF

Example: `BB:BTO:TRIG:OUTP:DEL:FIX ON`
restricts the marker signal delay setting range to the current range.

Manual operation: See "Marker x Delay" on page 28

[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay <Delay>

Defines the delay between the signal on the marker outputs and the start of the signal, expressed in terms of samples. Command

`BB:BTO:TRIGger:OUTPut:DELay:FIXed` can be used to restrict the range of values to the dynamic range, i.e. the range within which a delay of the marker signals can be set without restarting the marker and signal.

Parameters:

<Delay>	float
	Range: 0 to 16777215
	Increment: 1E-3
	*RST: 0

Example: `BB:BTO:TRIG:OUTP2:DEL 1600`
sets a delay of 1600 samples for the corresponding marker signal.

Manual operation: See "[Marker x Delay](#)" on page 28

[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay:MINimum?
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay:MAXimum?

Queries the maximum marker delay for setting `:BB:BTO:TRIG:OUTP:DEL:FIX ON`.

Return values:

<Maximum>	float
	Range: 0 to 16777215
	Increment: 1E-3
	*RST: 2000

Example: `BB:BTO:TRIG:OUTP:DEL:FIX ON`
restricts the marker signal delay setting range to the dynamic range.

`BB:BTO:TRIG:OUTP:DEL:MAX?`

queries the maximum of the dynamic range.

Response: 2000

the maximum for the marker delay setting is 2000 samples.

Usage: Query only

Manual operation: See "[Marker x Delay](#)" on page 28

[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:MODE <Mode>

The command defines the signal for the selected marker output.

Parameters:

<Mode>

REStart | StARt | ACTive | PULSe | PATTern | RATio

REStart

A marker signal is generated at the start of each signal sequence.

StARt

A marker signal is generated at the start of each frame

ACTive

The marker masks the active part of the frame. At the start of each burst, the marker signal changes to high. It changes back to low after the end of each burst.

PULSe

A regular marker signal is generated. The clock frequency is defined by entering a divider. The frequency is derived by dividing the symbol rate by the divider. The input box for divider opens when Pulse is selected, and the resulting pulse frequency is displayed below.

PATTern

A marker signal that is defined by a bit pattern is generated. The pattern has a maximum length of 32 bits and is defined in an input field which opens when pattern is selected.

RATio

A regular marker signal corresponding to the Time Off / Time On specifications in the commands

`SOURce:BB:BTO:TRIGger:OUTPut:OFFTime` and

`SOURce:BB:BTO:TRIGger:OUTPut:ONTime` is generated.

*RST: REStart

Example:

`BB:BTO:TRIG:OUTP:MODE REST`

selects the marker generation at the start of each signal sequence

Manual operation: See "[Marker Mode](#)" on page 27

`[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:ONTime <OnTime>`

`[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:OFFTime <OffTime>`

Sets the number of samples in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:BTO:TRIGger:OUTPut:MODE RATio` on the marker outputs is OFF.

Parameters:

<OffTime>

integer

Range: 1 to 16777215

*RST: 1

Example:

`BB:BTO:TRIG:OUTP:OFFT 2000`

sets an OFF time of 2000 samples for the corresponding marker signal.

Manual operation: See "[Marker Mode](#)" on page 27

[:SOURce<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:PATTern <Pattern>

Selects the data for a pattern.

Parameters:

<Pattern> 64 bits
*RST: #H2,2

Example:

BB:BT0:TRIG:OUTP:MODE PATT
sets the marker mode for the corresponding marker signal
BB:BT0:TRIG:OUTP:PATT #B010101,6
sets the pattern

Manual operation: See "[Marker Mode](#)" on page 27

[:SOURce<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

The clock frequency is defined by entering a divider. The frequency is derived by dividing the symbol rate by the divider. The input box for divider opens when Pulse is selected, and the resulting pulse frequency is displayed below it.

Parameters:

<Divider> integer
Range: 2 to 1024
*RST: 2

Example:

BB:BT0:TRIG:OUTP:MODE:PULS
selects marker mode Pulse
BB:BT0:TRIG:OUTP:PULS:DIV 2
sets a divider for the clock frequency.

Manual operation: See "[Marker Mode](#)" on page 27

[:SOURce<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:PULSe:FREQuency?

Queries the marker pulse frequency.

Return values:

<Frequency> float
Range: 2 to 1024
Increment: 1E-3
*RST: 2

Example:

BB:BT0:TRIG:OUTP:MODE:PULS
selects marker mode Pulse
BB:BT0:TRIG:OUTP:PULS:DIV 2
sets a divider for the clock frequency
BB:BT0:TRIG:OUTP:PULS:FREQ
displays resulting pulse frequency

Usage: Query only
Manual operation: See "Marker Mode" on page 27

5.5 Clock Settings

This section lists the remote control commands, necessary to configure the clock.

<code>[:SOURce<hw>]:BB:BT0oth:CLOCK:MODE</code>	91
<code>[:SOURce<hw>]:BB:BT0oth:CLOCK:MULTiplier</code>	91
<code>[:SOURce<hw>]:BB:BT0oth:CLOCK:SOURce</code>	91

`[:SOURce<hw>]:BB:BT0oth:CLOCK:MODE <Mode>`

Enters the type of an externally supplied clock. When `MSAMple` is used, a multiple of the clock is supplied and the clock is derived internally from this. The multiplier is entered with the command `[:SOURce<hw>]:BB:BT0oth:CLOCK:MULTiplier`.

Parameters:

<Mode> SAMPLE | MSAMple
 *RST: SAMPLE

Example: `SOURce1:BB:BTO:CLOCK:MODE MSAMple`
 sets the type of externally supplied clock.

Manual operation: See "Clock Mode" on page 30

`[:SOURce<hw>]:BB:BT0oth:CLOCK:MULTiplier <Multiplier>`

Specifies the multiplication factor for clock type multiple (`:BB:BT0:CLOCK:MODE MSAMple`) in the case of an external clock source.

Parameters:

<Multiplier> integer
 Range: 1 to 64
 Increment: 1
 *RST: 4

Example: `SOURce1:BB:BTO:CLOCK:SOURce EGC1`
 selects the external clock source.
`SOURce1:BB:BTO:CLOCK:MODE MSAMple`
 sets the clock type
`SOURce1:BB:BTO:CLOC:MULTiplier 12`
 the multiplier for the external clock rate is 12.

Manual operation: See "Clock Multiplier" on page 30

`[:SOURce<hw>]:BB:BT0oth:CLOCK:SOURce <Source>`

Selects the clock source.

Parameters:

<Source> INTernal | EGC1 | EGC2 | ELClock | EXTernal

INTernal
The instrument uses its internal clock reference

EGC1|EGC2
External global clock

ELCLock
External local clock

EXTernal
EXTernal = EGC1
Setting only; provided for backward compatibility with other R&S signal generators.

*RST: INTernal

Example:

BB:BTO:CLOC:SOUR: INT
selects an internal clock reference.

Manual operation: See "Clock Source" on page 30

5.6 Power Ramping

[:SOURce<hw>]:BB:BTOoth:PRAMping:FOFFset	92
[:SOURce<hw>]:BB:BTOoth:PRAMping:RFUNction	92
[:SOURce<hw>]:BB:BTOoth:PRAMping:ROFFset	93
[:SOURce<hw>]:BB:BTOoth:PRAMping:RTIME	93

[:SOURce<hw>]:BB:BTOoth:PRAMping:FOFFset <FOffset>

The command sets the offset of the falling edge of the envelope at the end of a burst. A positive value introduces a guard period after the end of the packet and negative value moves the ramp into the end part of the transmitted packet.

Parameters:

<FOffset> integer

Range: -32 to 32

Increment: 1

*RST: 0

Default unit: symbols

Example:

BB:BTO:PRAM:FOFF 8.0
sets the offset in the falling edge of the envelope to 8.0 symbols.

Manual operation: See "Fall Offset" on page 67

[:SOURce<hw>]:BB:BTOoth:PRAMping:RFUNction <RFunction>

The command selects the form of the transmitted power, i.e. the shape of the rising and falling edges during power ramp control.

Parameters:

<RFunction> LINear | COSine
 *RST: COSine

Example:

BB:BTO:PRAM:RFUN LIN
 sets linear shape for the rising and falling edges during power ramp control.

Manual operation: See "[Ramp Function](#)" on page 66

[:SOURce<hw>]:BB:BT0oth:PRAMPing:ROFFset <ROffset>

The command sets the offset in the rising edge of the envelope at the start of a burst. A positive value moves the ramp into beginning of a transmitted packet and a negative value introduce an additional guard period after the end of the packet.

Parameters:

<ROffset> integer
 Range: -32 symbols to 32 symbols
 Increment: 1 symbol
 *RST: 0 symbols

Example:

BB:BTO:PRAM:ROFF 8.0
 sets the offset in the rising edge of the envelope to 8.0 symbols.

Manual operation: See "[Rise Offset](#)" on page 66

[:SOURce<hw>]:BB:BT0oth:PRAMPing:RTIME <RTIME>

The command sets the power ramping rise time and fall time for a burst.

Parameters:

<RTIME> integer
 Range: 1 symbol to 32 symbols
 Increment: 1 symbol
 *RST: 1 symbol

Example:

BB:BTO:PRAM:TIME 2.0
 sets the power ramping rise time and fall time for a burst to 2 samples.

Manual operation: See "[Ramp Time](#)" on page 66

5.7 Packet Configuration Setting

[:SOURce<hw>]:BB:BT0oth:PCONfiguration:ACKnowledgement	94
[:SOURce<hw>]:BB:BT0oth:PCONfiguration:BDALap	94
[:SOURce<hw>]:BB:BT0oth:PCONfiguration:BDANap	95
[:SOURce<hw>]:BB:BT0oth:PCONfiguration:BDUAp	95
[:SOURce<hw>]:BB:BT0oth:PCONfiguration:CODEvice	95

<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DATA</code>	96
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DATA:DPATtern</code>	96
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DATA:DSELection</code>	96
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DATA:VDPATtern</code>	96
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DATA:VDSElection</code>	97
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DLENgth</code>	97
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DSFPacket</code>	97
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:DWHitening</code>	98
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:EIRPacketfollows</code>	98
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:HFControl</code>	99
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:LTADdress</code>	99
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:PFControl</code>	99
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:PLENgth</code>	100
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:SNSValue</code>	100
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:SRMode</code>	101
<code>[:SOURce<hw>]:BB:BT0oth:PCONfiguration:VDATA</code>	101

`[:SOURce<hw>]:BB:BT0oth:PCONfiguration:ACKNowledgement` <Acknowledgement>

This command sets the ARQN bit of the packet header..

Parameters:

<Acknowledgement> NAK | ACK

NAK

Request to retransmit the previous payload.

ACK

Previous payload has been received successfully.

*RST: ACK

Example:

```
BB:BTO:PTYP DH1
selects the packet type DH1.
BB:BTO:PCON:DSFP PED
enable packet editor under data source for packet
BB:BTO:PCON:ACKN ACK
sets positive acknowledgement
```

Manual operation: See "[Acknowledgment](#)" on page 34

`[:SOURce<hw>]:BB:BT0oth:PCONfiguration:BDALap` <BdaLap>

Sets the lower address part of Bluetooth Device Address. The length of LAP is 24 bits or 6 hexadecimal figures.

Parameters:

<BdaLap> 24 bits
Range: #H000000 to #HFFFFFFF
*RST: 80

Example:

```
BB:BTO:PCON:BDAL #H000000, 24
sets the lower address part
```

Manual operation: See "[Bluetooth Device Address \(BD_ADDR\)](#)" on page 33

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:BDANap <BdaNap>

Enters the non-significant address part of Bluetooth Device Address. The length of NAP is 16 bits or 4 hexadecimal figures.

Parameters:

<BdaNap> 16 bits
 Range: #H0000 to #HFFFF
 *RST: ABCD

Example: BB:BTO:PCON:BDAN #H0000, 16
 sets the non-significant address part

Manual operation: See "[Bluetooth Device Address \(BD_ADDR\)](#)" on page 33

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:BDAUap <BdaUap>

Enters the upper address part of Bluetooth Device Address. The length of UAP is 8 bits or 2 hexadecimal figures.

Parameters:

<BdaUap> 8 bits
 Range: #H00 to #HFF
 Increment: 1
 *RST: 48

Example: BB:BTO:PCON:BDAN #H00, 8
 sets the non-significant address part

Manual operation: See "[Bluetooth Device Address \(BD_ADDR\)](#)" on page 33

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:CODevice <CoDevice>

A parameter received during the device discovery procedure, indicates the type of device and which types of service that are supported.

Parameters:

<CoDevice> 24 bits
 Range: #H000000 to #HFFFFFFF

Example: BB:BTO:PTYP FHS
 sets the packet type
 BB:BTO:PCON:DSFP PED
 enable packet editor under data source for packet
 BB:BTO:PCON:COD '020104'
 sets the class of device

Manual operation: See "[Class of Device](#)" on page 36

[[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA <Data>

Selects the data source used for the payload.

Parameters:

<Data> ALL0 | ALL1 | PATTern | PN09 | PN11 | PN15 | PN16 | PN20 |
PN21 | PN23 | DLISt
*RST: PN09

Example:

BB:BTO:PTYP FHS
sets the packet type
BB:BTO:PCON:DSFP PED
enable packet editor under data source for packet
BB:BTO:PCON:DATA ALL1
sets the data type.

Manual operation: See ["Data Source"](#) on page 35

[[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DPATtern <DPattern>

Selects the data for a pattern.

Parameters:

<DPattern> 64 bits
*RST: #H0,1

Example:

BB:BTO:PCON:DATA PATT
sets the data type.
BB:BTO:PCON:DATA:DPAT #B010101, 6
selects the data for a pattern

Manual operation: See ["Data Source"](#) on page 35

[[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DSELection <DSelection>

The command selects data list file.

Parameters:

<DSelection> string
Increment: 1

Example:

BB:BTO:PCON:DATA DLIS
selects the data type.
BB:BTO:PCON:DSEL bluetooth_1
selects the file for the data.

Manual operation: See ["Data Source"](#) on page 35

[[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:VDPAttern <VdPattern>

Sets the bit pattern for the voice data.

Parameters:

<VdPattern> 64 bits
 Increment: 1
 *RST: #H0,1

Example:

BB:BTO:PCON:DATA:PATT
 selects the data type.
 BB:BTO:PCON:DATA:VDPA #B010101, 6
 selects the bit pattern for the voice data.

Manual operation: See "[Data Source \(Voice Field\)](#)" on page 36

[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:VDSElection <VdSelection>

The command selects the data list for voice data.

Parameters:

<VdSelection> string
 Increment: 1

Example:

BB:BTO:PCON:VDAT DLIS
 selects the data type.
 BB:BTO:PCON:VDSE bluetooth_1
 selects the file for the data.

Manual operation: See "[Data Source \(Voice Field\)](#)" on page 36

[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DLENgth <DLength>

Sets the payload data length in bytes.

Parameters:

<DLength> integer
 Range: 0 to depends on packet type
 Increment: 1
 *RST: depends on packet type

Example:

BB:BTO:PTYP DH1
 sets the packet type.
 BB:BTO:PCON:DSFP PED
 enable packet editor under data source for packet
 BB:BTO:PCON:DLEN 25
 sets the data length.

Manual operation: See "[Data Length](#)" on page 35

[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DSFPacket <DsfPacket>

The command selects the data source for the selected packet type.

Parameters:**<DsfPacket>** PEDit | ADATa**PED**

Enables Packet Editor. All packet fields can be configured individually.

ADAT

Fills the generated packets with the selected data source. Useful if predefined data contents are loaded with a data list file or the data contents of the packet are not of interest.

***RST:** PEDit**Example:**

BB:BTO:PCON:DSFP PED

enables packet editor under data source for packet.

Manual operation: See ["Data Source for Packet"](#) on page 33**[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:DWhitening <DWhitening>**

The command activates or deactivates the Data Whitening.

Parameters:**<DWhitening>** 0 | 1 | OFF | ON***RST:** OFF**Example:**

BB:BTO:PCON:DWH ON

activates data whitening.

Manual operation: See ["Data Whitening"](#) on page 33**[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:EIRPacketfollows <EirPacketFollow>**

The command indicates that an extended inquiry response packet may follow.

Parameters:**<EirPacketFollow>** YES | NO**YES**

Indicates that EIR packet follows.

NO

Indicates that EIR packet does not follow.

***RST:** NO**Example:**

BB:BTO:PCON:PTYP FHS

sets the packet type.

BB:BTO:PCON:DSFP PED

enable packet editor under data source for packet

BB:BTO:PCON:EIRP YES

the EIR packet follows.

Manual operation: See ["EIR packet follows"](#) on page 36

[:SOURce<hw>]:BB:BTOoth:PCONfiguration:HFControl <HfControl>

The command sets the FLOW bit in the header. This bit indicates start or stop of transmission of packets over the ACL logical transport.

Parameters:

<HfControl> GO | STOP

GO

Allows the other devices to transmit new data.

STOP

Stops the other devices from transmitting data temporarily.

*RST: GO

Example:

BB:BTO:PCON:PTYP DH1

sets the packet type.

BB:BTO:PCON:DSFP PED

enable packet editor under data source for packet.

BB:BTO:PCON:HFC GO

allows the other devices to transmit new data.

Manual operation: See "[Flow Control \(Header\)](#)" on page 34

[:SOURce<hw>]:BB:BTOoth:PCONfiguration:LTAddress <LtAddress>

The command enters the logical transport address for the header. Each slave active in a piconet is assigned a primary logical transport address (LT_ADDR). The all-zero LT_ADDR is reserved for broadcast messages.

Parameters:

<LtAddress> integer

Range: 0 to 7

Increment: 1

*RST: 0

Example:

BB:BTO:PCON:PTYP DH1

sets the packet type.

BB:BTO:PCON:DSFP PED

enable packet editor under data source for packet

BB:BTO:PCON:LTAD 0

sets the logical transport address equal zero.

Manual operation: See "[Logical Transport Address](#)" on page 34

[:SOURce<hw>]:BB:BTOoth:PCONfiguration:PFControl <PfControl>

The command sets the FLOW bit in the payload (flow control per logical link).

Parameters:

<PfControl> GO | STOP

GO

Indicates the start of transmission of ACL packets after a new connection has been established.

STOP

Indicates the stop of transmission of ACL packets before an additional amount of payload data is sent.

*RST: GO

Example:

BB:BTO:PCON:PTYP DH1

sets the packet type.

BB:BTO:PCON:DSFP PED

enable packet editor under data source for packet

BB:BTO:PCON:PFC GO

allows the flow per logical link.

Manual operation: See "[Flow Control \(Payload\)](#)" on page 35

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:PLENgt <PLength>

Sets the packet length in symbols.

Parameters:

<PLength> integer

Range: 1 to depends on packet type

Increment: 1

*RST: depends on packet type

Example:

BB:BTO:PCON:DSFP ADAT

fills the all data under data source for packet.

BB:BTO:PCON:PLEN 1

sets the packet length.

Manual operation: See "[Packet Length](#)" on page 36

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:SNValue <SnSvalue>

The command sets the start value of the header SEQN bit. The SEQN bit is present in the header to filter out retransmissions in the destination. The signal generator is altering this bit automatically on consecutive frames, if a sequence length of at least 2 frames is set.

Parameters:

<SnSvalue> integer

Range: 0 to 1

Increment: 1

*RST: 1

Example: BB:BTO:PCON:PTYP DH1
sets the packet type.
BB:BTO:PCON:DSFP PED
enables packet editor under data source for packet.
BB:BTO:PCON:SNSV ONE
sets the SEQN bit of the first CRC data packet at the start of a connection.

Manual operation: See "[SEQN Start Value](#)" on page 34

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:SRMode <SrMode>

The command indicates the interval between two consecutive page scan windows, determines the behavior of the paging device.

Parameters:

<SrMode> R0 | R1 | R2

R0
The scan interval is equal to the scan window T w page scan (continuous nscan) and maximal 1.28s.

R1
The scan interval is maximal 1.28s.

R2
The scan interval is maximal 2.56s.

*RST: R0

Example: BB:BTO:PCON:PTYP FHS
sets the packet type.
BB:BTO:PCON:DSFP PED
enables packet editor under data source for packet.
BB:BTO:PCON:SRM R0
sets the scan repetition mode.

Manual operation: See "[Scan Repetition Mode](#)" on page 36

[:SOURCE<hw>]:BB:BTOoth:PCONfiguration:VDATa <VData>

The command selects the data source for the voice field.

Parameters:

<VData> ALL0 | ALL1 | PATTErn | PN09 | PN11 | PN15 | PN16 | PN20 |
PN21 | PN23 | DLISt

*RST: PN09

Example: BB:BTO:PCON:VDAT ALL1
sets the voice data type.

Manual operation: See "[Data Source \(Voice Field\)](#)" on page 36

5.8 Dirty Transmitter Test

<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:DTTState</code>	102
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:FDDeviation</code>	102
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:FDRate</code>	103
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:SPHase</code>	103
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:STDefault</code>	103
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE</code>	103
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:LONG:SET<ch>:CFOffset</code>	104
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:LONG:SET<ch>:MINDex</code>	104
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:LONG:SET<ch>:STATE</code>	105
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:LONG:SET<ch>:STERror</code>	105
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:SHORT:SET<ch>:CFOffset</code>	106
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:SHORT:SET<ch>:STATE</code>	106
<code>[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:SHORT:SET<ch>:STERror</code>	107

`[:SOURce<hw>]:BB:BTOoth:DTTTest:DTTState <DttState>`

The command activates/deactivates the Dirty Transmitter Test. For Basic Rate packets, each set of parameters in the Dirty Transmitter Setting table below is used for a duration of 20 ms. After 20 ms, the following set is used, continuing with the first set after the sequence is completed.

For EDR packets, the parameter sets apply for 20 packets each.

Parameters:

`<DttState>` 0 | 1 | OFF | ON
 *RST: OFF

Example:

`BB:BTO:DTT:DTTS ON`
 activates the Dirty Transmitter Test.

Manual operation: See "[Dirty Transmitter Test](#)" on page 59

`[:SOURce<hw>]:BB:BTOoth:DTTTest:FDDeviation <FdDeviation>`

Sets a frequency drift rate.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset. The maximum deviation reached during the drift equals the set frequency drift deviation.

Parameters:

`<FdDeviation>` integer
 Range: -100 to 100
 *RST: 25

Example:

`BB:BTO:DTT:FDD 25`
 enters frequency drift deviation.

Manual operation: See "[Frequency Drift Deviation \(+/-\)](#)" on page 60

[[:SOURce<hw>]:BB:BT0oth:DTTTest:FDRate <FdRate>

Sets a frequency drift rate.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset with the set frequency drift rate.

Parameters:

<FdRate> 0.3 KHz | 0.5 KHz | 1.6 KHz | 10 KHz
 Range: depends on packet type to depends on packet type
 Increment: 0.001
 *RST: depends on packet type

Example: BB:BT0:DTT:FDR 1.6
 enters frequency drift rate.

Manual operation: See "[Frequency Drift Rate](#)" on page 60

[[:SOURce<hw>]:BB:BT0oth:DTTTest:SPHase <SPHase>

The command enters a start phase.

The start phase of the sine wave used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset is set here.

Parameters:

<SPHase> integer
 Range: 0 to 359
 Increment: 1
 *RST: 0
 Default unit: degree

Example: BB:BT0:DTT:SPH 0
 enters a start phase.

Manual operation: See "[Start Phase](#)" on page 59

[[:SOURce<hw>]:BB:BT0oth:DTTTest:STDefault

The command calls the default settings for the Dirty Transmitter Test.

Example: BB:BT0:DTT:STD
 calls the default settings.

Usage: Event

Manual operation: See "[Set to Default](#)" on page 59

[[:SOURce<hw>]:BB:BT0oth:DTTTest:TABLE <Table>

Opens the table settings.

Parameters:

<Table> NOTable | SHORT | LONG

Example:

```
BB:BTO:PTYP DH1
calls the default settings.
BB:BTO:DTT:TABL LONG
calls the default settings.
```

[:SOURce<hw>] :BB:BT0oth:DTT:TABLE:LONG:SET<ch>:CFOffset <CfOffset>

Sets a carrier frequency offset.

The carrier frequency offset shows the deviation of the transmitted initial center frequency from carrier frequency.

Parameters:

<CfOffset> integer
 Range: -150 to 150
 *RST: 1
 Default unit: kHz

Example:

```
BB:BTO:PTYP DH1
sets the packet type.
BB:BTO:DTT:TABL LONG
enters the table type
BB:BTO:DTT:TABL:LONG:SET2:CFOF 14
enters a carrier frequency offset.
```

Manual operation: See "[Carrier Frequency Offset kHz](#)" on page 60

[:SOURce<hw>] :BB:BT0oth:DTT:TABLE:LONG:SET<ch>:MINDex <MIndex>

Sets the modulation index.

The modulation index specifies the frequency deviation.

The modulation index h is defined as

$$h = \frac{2\Delta f}{f_{symbol}}$$

where

f_{symbol} = "symbol rate", set with the command `[:SOURce<hw>] :BB:BT0oth:SRATE:VARiation`

Δf = "frequency deviation", set with the command `[:SOURce<hw>] :BB:BT0oth:MSETtings:FDEVIation`

According to the Bluetooth standard, the modulation index is allowed to vary between 0.28 and 0.35.

Parameters:

<MIndex> float
 Range: 0.28 to 0.55
 Increment: 0.01
 *RST: 0.28

Example:

BB:BTO:PTYP DH1
 sets the packet type.
 BB:BTO:DTT:TABL LONG
 enters the table type
 BB:BTO:DTT:TABL:LONG:SET2:MIND 0.3
 enters a modulation index.

Manual operation: See "[Modulation Index](#)" on page 61

[[:SOURce<hw>]:BB:BTOoth:DTTtest:TABLE:LONG:SET<ch>:STATE <State>

Activates or deactivates the corresponding parameter set for the long table. If a set deactivated, its parameters are skipped in the sequence. Instead, the next active set is used.

For Basic Rate packets, each set applies to 20ms of signal.

Parameters:

<State> 0 | 1 | OFF | ON
 *RST: 1

Example:

BB:BTO:PTYP DH1
 sets the packet type.
 BB:BTO:DTT:TABL LONG
 enters the table type
 BB:BTO:DTT:TABL:LONG:SET2:STAT ON
 activates the set 2 in the long table.

Manual operation: See "[State](#)" on page 60

[[:SOURce<hw>]:BB:BTOoth:DTTtest:TABLE:LONG:SET<ch>:STERror <StError>

Sets a symbol timing error in ppm.

The Symbol Timing Error modifies the symbol clock frequency by the set amount.

Parameters:

<StError> integer
 Range: -150 to 150
 Increment: 1
 *RST: 1
 Default unit: ppm

Example: BB:BTO:PTYP DH1
sets the packet type.
BB:BTO:DTT:TABL LONG
enters the table type
BB:BTO:DTT:TABL:LONG:SET2:STER -20
enters a symbol timing error.

Manual operation: See "[Symbol Timing Error](#)" on page 60

[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:SHORT:SET<ch>:CFOffset
<CfOffset>

Sets a carrier frequency offset.

The carrier frequency offset shows the deviation of the transmitted initial center frequency from carrier frequency.

Parameters:

<CfOffset> integer
Range: -150 to 150
*RST: 1
Default unit: kHz

Example: BB:BTO:PTYP DH1
sets the packet type.
BB:BTO:DTT:TABL SHOR
enters the table type
BB:BTO:DTT:TABL:SHOR:SET2:CFOF 65
enters a carrier frequency offset.

Manual operation: See "[Carrier Frequency Offset kHz](#)" on page 60

[:SOURce<hw>]:BB:BTOoth:DTTTest:TABLE:SHORT:SET<ch>:STATE <State>

Activates or deactivates the corresponding parameter set in the short table. If a set deactivated, its parameters are skipped in the sequence. Instead, the next active set is used.

For EDR packets, each set applies to 20 packets.

Parameters:

<State> 0 | 1 | OFF | ON
*RST: 1

Example: BB:BTO:PTYP DH1
sets the packet type.
BB:BTO:DTT:TABL SHOR
enters the table type
BB:BTO:DTT:TABL:SHOR:SET2:STAT ON
activates the set 2 in the short table.

Manual operation: See "[State](#)" on page 60

[[:SOURce<hw>]:BB:BT0oth:DTTTest:TABLE:SHORT:SET<ch>:STERror <StError>

Sets a symbol timing error in ppm.

The Symbol Timing Error modifies the symbol clock frequency by the set amount.

Parameters:

<StError> integer
 Range: -150 to 150
 Increment: 1
 *RST: 1

Example:

BB:BT0:PTYP DH1
 sets the packet type.
 BB:BT0:DTT:TABLE SHOR
 enters the table type
 BB:BT0:DTT:TABLE:SHORT:SET2:STER 20
 enters a symbol timing error.

Manual operation: See "Symbol Timing Error" on page 60

5.9 LE Commands

[[:SOURce<hw>]:BB:BT0oth:BCRole.....	108
[[:SOURce<hw>]:BB:BT0oth:BMODE.....	109
[[:SOURce<hw>]:BB:BT0oth:DTTTest:TPConfiguration:TPInterval.....	109
[[:SOURce<hw>]:BB:BT0oth:DTTTest:TPConfiguration:UPLength.....	110
[[:SOURce<hw>]:BB:BT0oth:DTTTest:TPConfiguration:UPSource.....	110
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:ACTable.....	110
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:DCTable.....	110
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DCMTable.....	110
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:ACTable:CHANnel<ch0>:STATE.....	111
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:DCTable:CHANnel<ch0>:STATE.....	111
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DCMTable: CHANnel<ch0>:STATE.....	111
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:AEDelay.....	111
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:ADInterval.....	112
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:AEInterval.....	112
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:APIInterval.....	112
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:LCMode.....	113
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:LTKey.....	113
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:AADdress.....	113
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ACID.....	114
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ACASsigned.....	114
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCASsigned.....	114
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCID.....	114
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ICASsigned.....	114
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ICID.....	114
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:CID.....	115
[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:CINstant.....	115

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:CINTerval.....	115
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:CIValue.....	116
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DATA.....	116
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DATA:DPATtern.....	116
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DATA:DSELectioN.....	117
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DLENGth.....	117
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DWHitening.....	117
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ECODE.....	118
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:EDIVersifier.....	118
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:FSLength.....	118
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:HLENGth.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:LCTimeout.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:MIVector.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SIVector.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:MSKD.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SSKD.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NCINterval.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NLCTimeout.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NSLatency.....	121
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NSValue.....	121
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NWOFset.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NWSize.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:TAType.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:RAType.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:RVEctor.....	123
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCACcuracy.....	123
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SDCI?.....	123
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SLATency.....	124
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SSValue.....	124
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SVNumber.....	124
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:UTYPE.....	125
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:VNUMber.....	125
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:WOFset.....	125
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:WSIZE.....	126
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PNUMber.....	126
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SINTerval.....	126
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SWINdow.....	127
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:WOINfo?.....	127
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:WSINfo?.....	127
[:SOURce<hw>]:BB:BT0oth:UPTYPE.....	128
[:SOURce<hw>]:BB:BT0oth:USLength.....	129

[:SOURce<hw>]:BB:BT0oth:BCRole <BcRole>

Determines the controller role. Depending on the selected channel type different roles are assigned to the controller. For channel type "Data", master or slave can be assigned. If channel type "Advertising" is selected, the parameter is read only and displayed directly above the graph.

Parameters:

<BcRole> MASTer | SLAVe | ADVertiser | SCANner | INITiator

MASTER

Assigns master role to the controller.

SLAVE

Selects slave as controller role.

ADVertiser|SCANner|INITiator

Assigned roles depending on the selected packet type of the respective channel type.

*RST: MASTer

Example:

SOUR:BB:BTO:BCR MAST

master as controller role.

SOUR:BB:BTO:BCR SLAV

slave as controller role.

Manual operation: See "[Bluetooth Controller Role](#)" on page 40

[:SOURce<hw>]:BB:BT0oth:BMODE <BMode>

Determines the digital Bluetooth standard. Basic Rate + EDR or Bluetooth Low Energy are available.

Parameters:

<BMode> BASic | BLENergy

BASic

Selects the Bluetooth mode Basic Rate + EDR.

BLENergy

Selects the Bluetooth mode Low Energy.

*RST: BASic

Example:

SOUR:BB:BTO:BMOD BLEN

Bluetooth mode Low Energy.

Manual operation: See "[Bluetooth Mode](#)" on page 20

[:SOURce<hw>]:BB:BT0oth:DTTTest:TPConfiguration:TPInterval <TpInterval>

Sets the time interval between two consecutive test packets, with regard to the starting points.

Parameters:

<TpInterval> float

Range: 0.625E-3 to 12.5E-3

Increment: 0.625E-3

*RST: 0.625E-3

Example:

SOUR:BB:BTO:DTT:TPC:TPIN 1.0

sets a time interval of 1.0.

Manual operation: See "[Packet Interval](#)" on page 57

[[:SOURce<hw>]:BB:BT0oth:DTTTest:TPCconfiguration:UPLength <UpLength>

Sets the payload length.

Parameters:

<UpLength> integer
 Range: 1 to 37
 *RST: 1

Example: SOUR:BB:BT0:DTT:TPC:UPL 37
 sets a payload length of 37.

Manual operation: See "[Payload Length](#)" on page 57

[[:SOURce<hw>]:BB:BT0oth:DTTTest:TPCconfiguration:UPSource <UpSource>

Selects the data source used for the payload test packets.

Parameters:

<UpSource> PN09 | PAT1 | PAT2 | PN15 | PAT3 | PAT4 | PAT5 | PAT6
PN9 / PN15
 Select a PRBS-modulated data sequence
 (PRBS = pseudo random binary sequence) for testing.
PAT1 ... PAT6
 Pattern is predefined.
 *RST: PN09

Example: SOUR:BB:BT0:DTT:TPC:UPS PN09
 PRBS-modulated data sequence for testing.

Manual operation: See "[Payload Type](#)" on page 57

[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:ACTable <AcTable>

[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:DCTable <DcTable>

**[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DCMTable
 <DcmTable>**

The data channel map table indicates the entire data set of the channels in a table.

Every channel is represented with bit positioned as per the data channel index. LSB represents data channel index 0 and the bit in position 36 represents data channel index 36.

If the channel is used channel its bit is to be set to '1'. Bit value '0' indicates that the channel is unused.

The bits in positions 37, 38 and 39 shall be set to zero upon transmission and ignored upon receipt.

Note: This parameter is relevant for data event and advertising frame configuration with the packet types:

- CHANNEL_MAP_REQ

- CONNECT_REQ

Parameters:

<DcmTable> string

Example:

```
SOUR:BB:BTO:ECON:ACTable NOT
Entire data set for Advertising Channel Table
SOUR:BB:BTO:ECON:DCT NOT
Entire data set for Data Channel Table
SOUR:BB:BTO:ECON:PCON:DCMT NOT
Entire data set for Channel Map Table
```

Manual operation: See "[Channel Table](#)" on page 46

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:ACTable:CHANnel<ch0>:STATe
<State>
```

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:DCTable:CHANnel<ch0>:STATe
<State>
```

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DCMTTable:
CHANnel<ch0>:STATe <State>
```

Indicates used and unused data channels.

Note: The previously used syntax `...:SET<ch>:STATe` has been replaced by `...:CHANnel<ch>:STATe`. Compatibility to the previous commands is given.

This parameter is relevant for data event and advertising frame configuration with the packet types:

- CHANNEL_MAP_REQ
- CONNECT_REQ

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example:

```
SOUR:BB:BTO:ECON:ACT:CHAN:STAT ON
State in Advertising Channel Table
SOURce:BB:BTO:ECON:DCT:CHAN:STAT ON
State in Data Channel Table
SOUR:BB:BTO:ECON:PCON:DCMT:CHAN:STAT ON
State in Data Channel Map Table
```

Manual operation: See "[Channel Table](#)" on page 46

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:AEDelay <AeDelay>
```

Sets a time delay between the start times of two consecutive advertising events. The value is added to the advertising event interval.

Parameters:

<AeDelay> float
 Range: 0 to 10E-3
 Increment: 0.1E-3
 *RST: 0

Example:

SOUR:BB:BTO:ECON:AED 10.0
 sets a time delay of 10.0 s.

Manual operation: See "[Advertising Event Delay](#)" on page 43

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:ADInterval <AdInterval>

Sets the time interval between two consecutive advertising events for packet type "ADV_DIRECT_IND".

Parameters:

<AdInterval> float
 Range: 1.05E-3 to 3.75E-3
 Increment: 0.01E-3
 *RST: 3.75E-3

Example:

SOUR:BB:BTO:ECON:ADIN 1.3
 sets a time interval of 1.3.

Manual operation: See "[Advertising Event Interval](#)" on page 43

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:AEInterval <AeInterval>

Sets the time interval between two consecutive advertising events, with regard to the starting points.

Parameters:

<AeInterval> float
 Range: 5E-3 to depends on oversampling
 Increment: 0.1E-3
 *RST: 20E-3

Example:

SOUR:BB:BTO:ECON:AEIN 20.0
 sets a time interval of 20.0.

Manual operation: See "[Advertising Event Interval](#)" on page 43

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:APInterval <ApInterval>

Sets the time interval between packets starting points of two consecutive packets in the advertising channel.

Parameters:

<ApInterval> float
 Range: 1.3E-3 to 10E-3
 Increment: 0.1E-3
 *RST: 10E-3

Example:

SOUR:BB:BTO:ECON:APIN 1.3
 sets a time interval of 1.3.

Manual operation: See "[Advertising Packet Interval](#)" on page 43

[[:SOURce<hw>]:BB:BTOoth:ECONfiguration:LCMode <LcMode>

Selects the link layer connection mode. In order to provide safe transmission of payload data, the data in the packet can be encrypted. If activated, the payload data follows MIC (Message authentication Code).

Parameters:

<LcMode> UENC | ENC
UENC
 Payload data is transmitted without encoding.
ENC
 The link layer connection runs in encrypted mode.
 *RST: UENC

Example:

SOUR:BB:BTO:ECON:LCM UENC
 without encoding.
 SOUR:BB:BTO:ECON:LCM ENC
 in encrypted mode.

Manual operation: See "[LL Connection Mode](#)" on page 45

[[:SOURce<hw>]:BB:BTOoth:ECONfiguration:LTKKey <LtKey>

Indicates the time the controller needs to receive the long term key from the host. After this time, the controller is ready to enter into the last phase of encryption mode setup.

Parameters:

<LtKey> 128 bits

Example:

SOUR:BB:BTO:ECON:LCM ENC
 SOUR:BB:BTO:ECON:LTK
 #H00000000000000000000000000000000,128
 In encrypted mode, the code can be edited.

Manual operation: See "[Long Term key \(hex\)](#)" on page 46

[[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:AAddress <AAddress>

Sets the access address of the link layer connection (32-bit string).

Parameters:**<AAddress>** 32 bits**Example:**

SOUR:BB:BTO:ECON:PCON:AADD #H00000000, 32
sets an access address.

Manual operation: See "[Access Address](#)" on page 48

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ACID <Acid>

**[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ACASsigned
<AcAssigned>**

**[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCASsigned
<ScAssigned>**

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCID <Scid>

**[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ICASsigned
<IcAssigned>**

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ICID <Icid>

Sets the advertiser's device address. In bluetooth low energy systems all the transmissions start with an 8 bit preamble followed by an access address. The access address is composed of a the parts "Company_Id" (LSB) and the "Company_assigned" (MSB). Beside the address fields the notation is given. For advertising channel packets the format of the device address differs, depending on the selected address type.

- "Public Address Types"
The public address is given from the registration authority IEEE and is composed of:
 - LSB: 24 bits = company_assigned
 - MSB: 24 bits = company_id
- "Private Address Type"
A private address is optional and composed of:
 - LSB: 24 bits = hash
 - MSB: 24 bits = random

Parameters:**<Icid>** 24 bits**Example:**

SOUR:BB:BTO:ECON:PCON:ACID #H000000, 24
SOUR:BB:BTO:ECON:PCON:ACAS #H000000, 24
Company_Assigned and Company_Id in Advertiser's Device
Address)
SOUR:BB:BTO:ECON:PCON:SCAS #H000000, 24
SOUR:BB:BTO:ECON:PCON:SCID #H000000, 24
Company_Assigned and Company_Id in Scanners Device
Address)
SOUR:BB:BTO:ECON:PCON:ICAS #H000000, 24
SOUR:BB:BTO:ECON:PCON:ICID #H000000, 24
Company_Assigned and Company_Id in Initiators Device
Address)

Manual operation: See "[Controller's Device Addr](#)" on page 53

[[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCONfiguration:CID <Cid>

Sets the company identifier of the manufacturer of the Bluetooth Controller. A 16 bit value is set.

Note: This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

Parameters:

<Cid> 16 bits
*RST: 0

Example: SOUR:BB:BT0:ECON:PCON:CID #H0000,16
sets the company ID.

Manual operation: See "[Company ID](#)" on page 55

[[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCONfiguration:CINstant <CInstant>

Sets a connection instant for indicating the connection event at which the new connection parameters are taken in use.

Parameters:

<CInstant> integer
Range: 1 to depends on sequence length
*RST: 1

Example: SOUR:BB:BT0:ECON:PCON:CINS 2
sets a connection instant.

Manual operation: See "[Connection Instant](#)" on page 51

[[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCONfiguration:CINterval <CInterval>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Set the time interval between the start points of two consecutive connection events.

Parameters:

<CInterval> float
Range: 7.5E-3 to depends on oversampling
Increment: 1.25E-3
*RST: 7.5E-3

Example: SOUR:BB:BT0:UPTY CREQ
sets packet type CONNECT_REQ
SOUR:BB:BT0:ECON:PCON:CINT 7.5
sets a time interval.

Manual operation: See "[Connection Event Interval](#)" on page 44

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:CIValue <CiValue>

Sets the initialization value for the CRC (Cyclic Redundary Check, 24 bits) calculation. A packet has been received correctly, when it has passed the CRC check.

Parameters:

<CiValue> 24 bits

Example:

SOUR:BB:BT0:ECON:PCON:CIV #H000000,24
sets the initialization value for the CRC.

Manual operation: See "[CRC Initial](#)" on page 54

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DATA <Data>

Selects the data source used for the payload.

Parameters:

<Data> ALL0 | ALL1 | PATtern | PN09 | PN11 | PN15 | PN16 | PN20 |
PN21 | PN23 | DLISt

All 0 / All 1

0 data and 1 data is generated internally.

Pattern

Pattern is user definable.

PN xx

Pseudo-random noise sequence. XX can be equal to 9, 11, 15, 16, 20, 21, 23.

Data List

Internal data from a programmable data list is used. The data list can be generated by the Data List Editor or generated externally.

*RST: PN09

Example:

SOUR:BB:BT0:ECON:PCON:DATA ALL0 | ALL1
SOUR:BB:BT0:ECON:PCON:DATA PATT
SOUR:BB:BT0:ECON:PCON:DATA:DPAT #H3F,8
SOUR:BB:BT0:ECON:PCON:DATA PN09
SOUR:BB:BT0:ECON:PCON:DATA DLIS
SOUR:BB:BT0:ECON:PCON:DATA:DSEL 'bluetooth-le'

Manual operation: See "[Data Source](#)" on page 50

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DATA:DPATtern <DPattern>

Selects the data for a pattern.

Parameters:

<DPattern> 64 bits

*RST: #H0,1

Example: BB:BTO:ECON:PCON:DATA PATT
sets the data type.
BB:BTO:ECON:PCON:DATA:DPAT #B010101, 6
selects the data for a pattern

Manual operation: See ["Data Source"](#) on page 50

[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA:DSElection
<DSelection>

Selects data list file.

Parameters:
<DSelection> string

Example: BB:BTO:ECON:PCON:DATA DLIS
selects the data type.
BB:BTO:ECON:PCON:DSEL Bluetooth-le
selects the file for the data.

Manual operation: See ["Data Source"](#) on page 50

[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DLENGTH
<DLength>

Sets the payload data length in bytes.

Parameters:
<DLength> integer
Range: 0 to depends on packet type
*RST: 31

Example: SOUR:BB:BTO:ECON:PCON:DLEN 31
payload data length is 31 bytes.

Manual operation: See ["Data Length"](#) on page 50

[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DWHitening
<DWhitening>

Activates or deactivates the Data Whitening. Evenly distributed white noise is ideal for the transmission and real data can be forced to look similar to white noise with different methods called Data Whitening. Applied to the PDU and CRC fields of all packet types, whitening is used to avoid long equal sequences in the data bit stream.

Parameters:
<DWhitening> 0 | 1 | OFF | ON
*RST: 0

Example: BB:BTO:ECON:PCON:DWH ON
activates data whitening.

Manual operation: See ["Data Whitening"](#) on page 48

[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ECODE <ECode>

Sets the error code value to inform the remote device why the connection is about to be terminated in case of LL_TERMINATE_IND packet. On the other hand, this parameter for LL_REJECT_IND packet is used for the reason a request was rejected. A 8 bit value is set.

Note: This parameter is relevant for data frame configuration and the packet type:

- LL_TERMINATE_IND
- LL_REJECT_IND

Parameters:

<ECode> 8 bits

Example: SOUR:BB:BT0:ECON:PCON:ECOD #H00,8
sets the error code

Manual operation: See "[Error Code](#)" on page 55

[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:EDIVersifier <EDiversifier>

Sets the encrypted diversifier of the master for device identification. The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

Parameters:

<EDiversifier> 16 bits

Example: SOUR:BB:BT0:ECON:PCON:EDIV #H0000, 16
sets the encrypted diversifier of the master.

Manual operation: See "[Encrypted DIVersifier \(hex\)](#)" on page 52

[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:FSLength <FsLength>

Enables that the feature set length is indicated. FeatureSet indicates whether the Controller features are used or not. All the data in FeatureSet is RFU(zero).

Parameters:

<FsLength> integer
Range: 1 to 26
*RST: 26

Example: SOUR:BB:BT0:ECON:PCON:FSL 12
feature set length is 12.

Manual operation: See "[Feature Set Length](#)" on page 53

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:HLEnGth
<HLength>
```

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets the difference from the current channel to the next channel. The master and slave devices determine the data channel in use for every connection event from the channel map. Hop_length is set for the LL connection and communicated in the CONNECT_REQ and CHANNEL_MAP_REQ packets.

Parameters:

```
<HLength>          integer
                    Range:    5 to 16
                    *RST:    5
```

Example:

```
SOUR:BB:BT0:UPTY CREQ
sets packet type CONNECT_REQ
SOUR:BB:BT0:ECON:PCON:HLEN 10
hop length is 10.
```

Manual operation: See "[Hop Length](#)" on page 52

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:LCTimeout
<LcTimeout>
```

(for data event and advertising frame configuration with the packet type: CONNECT_REQ)

Defines the maximum time between two correctly received Bluetooth Low Energy packets in the LL connection before the connection is considered lost.

Parameters:

```
<LcTimeout>        float
                    Range:    100E-3 to 32000E-3
                    Increment: 10E-3
                    *RST:    100E-3
```

Example:

```
SOUR:BB:BT0:UPTY CREQ
sets packet type CONNECT_REQ
SOUR:BB:BT0:ECON:PCON:LCT 150
LL Connection Timeout is 150.
```

Manual operation: See "[LL Connection Timeout](#)" on page 51

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:MIVector
<MiVector>
```

```
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SIVector
<SiVector>
```

Sets the master's or the slave's portion of the initialization vector(IVm/IVs).

Parameters:**<SiVector>** 32 bits**Example:**

```
SOUR:BB:BTO:ECON:PCON:MIV #H0000000000000000, 32
(Master).
```

```
SOUR:BB:BTO:ECON:PCON:SIV #H0000000000000000, 32
(Slave).
```

```
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:MSKD <Mskd>
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SSKD <Sskd>
```

Sets the master's or the slave's portion of the session key diversifier (SKDm/SKDs).

Parameters:**<Sskd>** 64 bits**Example:**

```
SOUR:BB:BTO:ECON:PCON:MSKD
#H0000000000000000, 64
(Master).
```

```
SOUR:BB:BTO:ECON:PCON:SSKD
#H0000000000000000, 64
(Slave).
```

```
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NCInterval
<NcInterval>
```

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets the time interval between the start points of two consecutive connection events.

Parameters:**<NcInterval>** float

Range: 7.5E-3 to depends on oversampling

Increment: 1.25E-3

*RST: 7.5E-3

Example:

```
SOUR:BB:BTO:UPTY CUR
sets packet type CONNECTION_UPDATE_REQ
SOUR:BB:BTO:ECON:PCON:NCIN 7.5
sets a time interval.
```

Manual operation: See "[Connection Event Interval](#)" on page 44

```
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NLCTimeout
<NlcTimeout>
```

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Defines the maximum time between two correctly received Bluetooth Low Energy packets in the LL connection before the connection is considered lost.

Parameters:

<NlcTimeout> float
 Range: 100E-3 to 32000E-3
 Increment: 10E-3
 *RST: 100E-3

Example:

SOUR:BB:BTO:UPTY CUR
 sets packet type CONNECTION_UPDATE_REQ
 SOUR:BB:BTO:ECON:PCON:NLCT 150
 LL Connection Timeout is 150.

Manual operation: See "[LL Connection Timeout](#)" on page 51

[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NSLatency
 <Nslatency>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets a number of consecutive connection events the slave can ignore for asymmetric link layer connections.

Parameters:

<Nslatency> integer
 Range: 0 to depends on LL connection timeout and connection event interval
 *RST: 0

Example:

SOUR:BB:BTO:UPTY CUR
 sets packet type CONNECTION_UPDATE_REQ
 SOUR:BB:BTO:ECON:PCON:NSL 10
 sets the number of consecutive connection events.

Manual operation: See "[Slave Latency](#)" on page 51

[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NSValue
 <Nsvalue>

Sets the start value of the next expected packet from the same device in the LL connection ("N"ext"E"xpected "S"equence"N"umber). This parameter can be set in the first event. From the second event this field is not indicated.

Parameters:

<Nsvalue> integer
 Range: 0 to 1
 *RST: 1

Example:

SOUR:BB:BTO:ECON:PCON:NSV 1
 Start Value is 1.

Manual operation: See "[NESN Start Value](#)" on page 49

**[[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCOnfiguration:NWOffset
<NwOffset>**

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets the start point of the transmit window.

Parameters:

<NwOffset> float
 Range: 0 to depends on connection event interval
 Increment: 1.25E-3
 *RST: 0

Example:

SOUR:BB:BT0:UPTY CUR
 sets packet type CONNECTION_UPDATE_REQ
 SOUR:BB:BT0:ECOn:PCOn:NWOf 800.0
 sets the start point of the transmit window.

Manual operation: See "[Transmit Window Offset](#)" on page 55

[[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCOnfiguration:NWSize <NwSize>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets the size of the transmit window, regarding to the start point.

Parameters:

<NwSize> float
 Range: 1.25E-3 to depends on connection event interval
 Increment: 1.25E-3
 *RST: 1.25E-3

Example:

SOUR:BB:BT0:UPTY CUR
 sets packet type CONNECTION_UPDATE_REQ
 SOUR:BB:BT0:ECOn:PCOn:NWS 8.0
 sets the size of the transmit window.

Manual operation: See "[Transmit Window Size](#)" on page 54

**[[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCOnfiguration:TAType <TaType>
[:SOURce<hw>]:BB:BT0oth:ECOnfiguration:PCOnfiguration:RAType <RaType>**

Selects the address type of the controller device. Depending on the Bluetooth controller role either Tx or Rx or both address types are assigned. Subdivided into private and random, a Bluetooth LE device address consists of 48 bits. The format of the device address differs depending on the selected address type.

Parameters:**<RaType>** PUBLIC | RANDom**PUBLIC**

Allocates a unique 48 bit address to each bluetooth LE device. The public address is given from the registration authority IEEE.

RANDom

Allocates a 48 bit address to each bluetooth LE device. A random address is optional.

***RST:** PUBLIC**Example:**

SOUR:BB:BTO:ECON:PCON:TATY PUBL

SOUR:BB:BTO:ECON:PCON:RATY RAND

[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:RVECtor <RVector>

Sets the random vector of the master for device identification. The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

Parameters:**<RVector>** 64 bits**Example:**

SOUR:BB:BTO:ECON:PCON:RVEC

#H0000000000000000,64

sets the random vector of the master.

Manual operation: See "[Random Vector \(hex\)](#)" on page 52**[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCACcuracy <ScAccuracy>**

Defines the master's clock accuracy with specified encoding. This parameter is used by the slave to determine required listening windows in the LL connection. It is a controller design parameter known by the Controller.

Parameters:**<ScAccuracy>** SCA0 | SCA1 | SCA2 | SCA3 | SCA4 | SCA5 | SCA6 | SCA7***RST:** SCA0**Example:**

SOUR:BB:BTO:ECON:PCON:SCAC SCA1

sets the encoding value

Manual operation: See "[Sleep Clock Accuracy](#)" on page 55**[[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SDCI?**

Queries the number of the first active data channel.

Return values:**<SelectedChannel>** integer**Range:** 0 to 36***RST:** 0

- Example:** `SOURce1:BB:BT0oth:ECONfiguration:SDCI?`
- Usage:** Query only
- Manual operation:** See "[Selected Data Channel Index](#)" on page 46

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SLATency
 <SLatency>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets a number of consecutive connection events the slave can ignore for asymmetric link layer connections.

Parameters:

<SLatency> integer
 Range: 0 to depends on LL connection timeout and connection event interval
 *RST: depends on LL connection timeout and connection event interval

Example: `SOUR:BB:BT0:UPTY CREQ`
 sets packet type CONNECT_REQ
`SOUR:BB:BT0:ECON:PCON:SLAT 10`
 sets the number of consecutive connection events.

Manual operation: See "[Slave Latency](#)" on page 51

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SSValue <SsValue>

Sets the sequence number of the packet. This parameter can be set in the first event. From the second event this field is not indicated.

Parameters:

<SsValue> integer
 Range: 0 to 1
 *RST: 0

Example: `SOUR:BB:BT0:ECON:PCON:SSV 1`
 sets the sequence number of the packet.

Manual operation: See "[SN Start Value](#)" on page 49

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SVNumber
 <SvNumber>

Sets a unique value for each implementation or revision of an implementation of the Bluetooth Controller. A 16 bit value is set.

Note: This parameter is relevant for data frame configuration and for the packet type: LL_VERSION_IND.

Parameters:

<SvNumber> 16 bits
 *RST: 0

Example:

SOUR:BB:BTO:ECON:PCON:SVN #H0000,16
 sets the sub version number

Manual operation: See "[Sub Version Number](#)" on page 56

[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:UTYPE <UType>

Enables that an invalid control packet is indicated. The CtrType field indicates the value of the LL control packet that caused the transmission of this packet.

Parameters:

<UType> 8 bits

Example:

SOUR:BB:BTO:ECON:PCON:UTYP #H8, 00
 enables that an invalid control packet is indicated.

Manual operation: See "[Unknown Type \(hex\)](#)" on page 53

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:VNUMBER
 <VNumber>**

Sets the company identifier of the manufacturer of the Bluetooth Controller. A 8 bit value is set.

Note: This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

Parameters:

<VNumber> 8 bits
 *RST: 0

Example:

SOUR:BB:BTO:ECON:PCON:VNUM #H00,8
 sets the version number

Manual operation: See "[Version Number](#)" on page 56

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:WOFFSET
 <WOffset>**

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets the start point of the transmit window.

Parameters:

<WOffset> float
 Range: 0 to depends on connection event interval
 Increment: 1.25E-3
 *RST: 0

Example: SOUR:BB:BTO:UPTY CREQ
 sets packet type CONNECT_REQ
 SOUR:BB:BTO:ECON:PCON:WOFF 800.0
 sets the start point of the transmit window.

Manual operation: See "[Transmit Window Offset](#)" on page 55

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:WSize <WSize>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets the size of the transmit window, regarding to the start point.

Parameters:

<WSize> float
 Range: 1.25E-3 to depends on connection event interval
 Increment: 1.25E-3
 *RST: 1.25E-3

Example: SOUR:BB:BTO:UPTY CREQ
 sets packet type CONNECT_REQ
 SOUR:BB:BTO:ECON:PCON:WSIZ 8.0
 sets the size of the transmit window.

Manual operation: See "[Transmit Window Size](#)" on page 54

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PNUMBER <PNumber>

Sets the number of Tx packets per event. Each connection contains at least one data channel packet. The maximum number of packets per event is determined by the duration of the connection event interval.

Parameters:

<PNumber> integer
 Range: 1 to depends on connection event interval
 *RST: 1

Example: SOUR:BB:BTO:ECON:PNUM 2580
 sets the number of Tx packets per event.

Manual operation: See "[No. of Tx Packets per Event](#)" on page 44

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SINTERval <SInterval>

Sets the time interval between the starting points of two consecutive windows during which the scanner is operating in an advertising channel.

Parameters:

<SInterval> float
 Range: 10E-3 to depends on oversampling and the number of advertising channel table states
 Increment: 0.625E-3
 *RST: 10E-3

Example:

SOUR:BB:BTO:ECON:SINT 3.5
 sets the time interval.

Manual operation: See "[Scan Interval](#)" on page 43

[[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SWINdow <SWindow>

Sets the length of the window during which the scanner is operating in the advertising channel. Note that the scan window is less or equal to the value of the scan interval.

Parameters:

<SWindow> float
 Range: 10E-3 to 10240E-3
 Increment: 0.625E-3
 *RST: 10E-3

Example:

SOUR:BB:BTO:ECON:SWIN 2.5
 sets the length of the window.

Manual operation: See "[Scan Window](#)" on page 43

[[:SOURce<hw>]:BB:BTOoth:ECONfiguration:WOINfo?

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Queries the start point of the transmit window.

Return values:

<WoInfo> string

Example:

SOUR:BB:BTO:UPTY CREQ
 sets packet type CONNECT_REQ
 SOUR:BB:BTO:ECON:PCON:WOIN?
 queries the start point of the transmit window.

Usage:

Query only

Manual operation: See "[Transmit Window Offset](#)" on page 44

[[:SOURce<hw>]:BB:BTOoth:ECONfiguration:WSINfo?

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Queries the size of the transmit window, regarding to the start point.

Return values:**<WsInfo>** string**Example:**

```
SOUR:BB:BTO:UPTY CREQ
sets packet type CONNECT_REQ
SOUR:BB:BTO:ECON:PCON:WSIN?
queries the size of the transmit window.
```

Usage:

Query only

Manual operation: See "[Transmit Window Size](#)" on page 44**[[:SOURce<hw>]:BB:BTOoth:UPTYe <UpType>**

Selects the packet type. The available packets depend on the selected channel type.

Parameters:

<UpType> AIND | ADINd | ANINd | SREQ | SRSP | CREQ | ADCind | DATA | CURReq | CMReq | TIND | EREQ | ERSP | SEReq | SERSp | URSP | FREQ | FRSP | TPACket | PEReq | PERSp | VIND | RIND

***RST:** AIND**Example:**

```
SOUR:BB:BTO:UPTY AIND
SOUR:BB:BTO:UPTY ADIN
SOUR:BB:BTO:UPTY ANIN
SOUR:BB:BTO:UPTY ADC
SOUR:BB:BTO:UPTY SREQ
SOUR:BB:BTO:UPTY SRSP
SOUR:BB:BTO:UPTY CREQ
SOUR:BB:BTO:UPTY TPAC
```

Channel type "Advertising"

```
SOUR:BB:BTO:UPTY DATA
SOUR:BB:BTO:UPTY CUR
SOUR:BB:BTO:UPTY CMR
SOUR:BB:BTO:UPTY TIND
SOUR:BB:BTO:UPTY LER
SOUR:BB:BTO:UPTY SERS
SOUR:BB:BTO:UPTY FREQ
SOUR:BB:BTO:UPTY TPAC
```

Channel type "Data" and Controller role "Master".

```
SOUR:BB:BTO:UPTY DATA
SOUR:BB:BTO:UPTY TIND
SOUR:BB:BTO:UPTY LERS
SOUR:BB:BTO:UPTY SER
SOUR:BB:BTO:UPTY SERS
SOUR:BB:BTO:UPTY URSP
SOUR:BB:BTO:UPTY FRSP
SOUR:BB:BTO:UPTY TPAC
```

Channel type "Data" and Controller role "Slave".**Manual operation:** See "[Packet Type](#)" on page 38

[[:SOURce<hw>]:BB:BT0oth:USLength <UsLength>

Selects the number of frames or events depending on the packet type. The signal repeats after the specified number of frames/events.

For SCAN_REQ and CONNECT_REQ packet, the sequence length is expressed in "Frames".

For TERMINATE_IND packets, a default value according to the specification is given:

Master: 'SlaveLatency + 6'

Slave: '6'

For all other packet types the sequence length is expressed in "Events".

Parameters:

<UsLength> integer

Range: depending on the number of states in dirty transmitter test to dynamics

*RST: 1

Example:

SOUR:BB:BT0:USL 1000

selects the number of frames or events.

Manual operation: See "[Sequence Length](#)" on page 39

List of Commands

[:SOURce<hw>]:BB:BT00th:BCRole.....	108
[:SOURce<hw>]:BB:BT00th:BCText?.....	69
[:SOURce<hw>]:BB:BT00th:BMODE.....	109
[:SOURce<hw>]:BB:BT00th:CLIPping:LEVel.....	74
[:SOURce<hw>]:BB:BT00th:CLIPping:MODE.....	75
[:SOURce<hw>]:BB:BT00th:CLIPping:STATe.....	75
[:SOURce<hw>]:BB:BT00th:CLOCK:MODE.....	91
[:SOURce<hw>]:BB:BT00th:CLOCK:MULTiplier.....	91
[:SOURce<hw>]:BB:BT00th:CLOCK:SOURce.....	91
[:SOURce<hw>]:BB:BT00th:CTYPe.....	69
[:SOURce<hw>]:BB:BT00th:DTTTest:DTTState.....	102
[:SOURce<hw>]:BB:BT00th:DTTTest:FDDeviation.....	102
[:SOURce<hw>]:BB:BT00th:DTTTest:FDRate.....	103
[:SOURce<hw>]:BB:BT00th:DTTTest:SPHase.....	103
[:SOURce<hw>]:BB:BT00th:DTTTest:STDefault.....	103
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE.....	103
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:LONG:SET<ch>:CFOFset.....	104
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:LONG:SET<ch>:MINDex.....	104
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:LONG:SET<ch>:STATe.....	105
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:LONG:SET<ch>:STERror.....	105
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:SHORT:SET<ch>:CFOFset.....	106
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:SHORT:SET<ch>:STATe.....	106
[:SOURce<hw>]:BB:BT00th:DTTTest:TABLE:SHORT:SET<ch>:STERror.....	107
[:SOURce<hw>]:BB:BT00th:DTTTest:TPConfiguration:TPINterval.....	109
[:SOURce<hw>]:BB:BT00th:DTTTest:TPConfiguration:UPLength.....	110
[:SOURce<hw>]:BB:BT00th:DTTTest:TPConfiguration:UPSourcE.....	110
[:SOURce<hw>]:BB:BT00th:ECONfiguration:ACTable.....	110
[:SOURce<hw>]:BB:BT00th:ECONfiguration:ACTable:CHANnel<ch0>:STATe.....	111
[:SOURce<hw>]:BB:BT00th:ECONfiguration:ADINterval.....	112
[:SOURce<hw>]:BB:BT00th:ECONfiguration:AEDelay.....	111
[:SOURce<hw>]:BB:BT00th:ECONfiguration:AEINterval.....	112
[:SOURce<hw>]:BB:BT00th:ECONfiguration:APINterval.....	112
[:SOURce<hw>]:BB:BT00th:ECONfiguration:DCTable.....	110
[:SOURce<hw>]:BB:BT00th:ECONfiguration:DCTable:CHANnel<ch0>:STATe.....	111
[:SOURce<hw>]:BB:BT00th:ECONfiguration:LCMode.....	113
[:SOURce<hw>]:BB:BT00th:ECONfiguration:LTKey.....	113
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:AADDRESS.....	113
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:ACASSigned.....	114
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:ACID.....	114
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:CID.....	115
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:CINStant.....	115
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:CINterval.....	115
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:CIValue.....	116
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:DATA.....	116
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:DATA:DPATtern.....	116
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:DATA:DSEllection.....	117
[:SOURce<hw>]:BB:BT00th:ECONfiguration:PCONfiguration:DCMTable.....	110

[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DCMTable:CHANnel<ch0>:STATe.....	111
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DLENgth.....	117
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:DWHitening.....	117
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ECODE.....	118
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:EDIVersifier.....	118
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:FSLength.....	118
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:HLENgth.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ICASSigned.....	114
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:ICID.....	114
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:LCTimeout.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:MIVector.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:MSKD.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NCINterval.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NLCTimeout.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NSLatency.....	121
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NSValue.....	121
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NWOFfset.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:NWSize.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:RAType.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:RVECTor.....	123
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCACcuracy.....	123
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCASSigned.....	114
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SCID.....	114
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SIVector.....	119
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SLATency.....	124
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SSKD.....	120
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SSValue.....	124
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:SVNumber.....	124
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:TAType.....	122
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:UTYPE.....	125
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:VNUMBER.....	125
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:WOFfset.....	125
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PCONfiguration:WSIZE.....	126
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:PNUMBER.....	126
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SDCI?.....	123
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SINTerval.....	126
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:SWINdow.....	127
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:WOINfo?.....	127
[:SOURce<hw>]:BB:BT0oth:ECONfiguration:WSINfo?.....	127
[:SOURce<hw>]:BB:BT0oth:FILTer:ILENgth.....	76
[:SOURce<hw>]:BB:BT0oth:FILTer:ILENgth:AUTO[:STATe].....	76
[:SOURce<hw>]:BB:BT0oth:FILTer:MINdex.....	77
[:SOURce<hw>]:BB:BT0oth:FILTer:MTYPE.....	77
[:SOURce<hw>]:BB:BT0oth:FILTer:OSAMpling.....	76
[:SOURce<hw>]:BB:BT0oth:FILTer:OSAMpling:AUTO[:STATe].....	77
[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:APCO25.....	78
[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:COSSine.....	78
[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:FGAuss.....	78
[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:GAUSS.....	78
[:SOURce<hw>]:BB:BT0oth:FILTer:PARAmeter:LPASS.....	79

[:SOURce<hw>]:BB:BT00th:FILTer:PARAmeter:PGAuss.....	79
[:SOURce<hw>]:BB:BT00th:FILTer:PARAmeter:RCOSine.....	79
[:SOURce<hw>]:BB:BT00th:FILTer:PARAmeter:SPHase.....	80
[:SOURce<hw>]:BB:BT00th:FILTer:TYPE.....	75
[:SOURce<hw>]:BB:BT00th:MSETtings:FDEVIation.....	77
[:SOURce<hw>]:BB:BT00th:PCONfiguration:ACKNowledgement.....	94
[:SOURce<hw>]:BB:BT00th:PCONfiguration:BDALap.....	94
[:SOURce<hw>]:BB:BT00th:PCONfiguration:BDANap.....	95
[:SOURce<hw>]:BB:BT00th:PCONfiguration:BDUAJap.....	95
[:SOURce<hw>]:BB:BT00th:PCONfiguration:CODevice.....	95
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DATA.....	96
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DATA:DPATtern.....	96
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DATA:DSELEction.....	96
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DATA:VDPATtern.....	96
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DATA:VDSELEction.....	97
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DLENgth.....	97
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DSFPacket.....	97
[:SOURce<hw>]:BB:BT00th:PCONfiguration:DWHitening.....	98
[:SOURce<hw>]:BB:BT00th:PCONfiguration:EIRPacketfollows.....	98
[:SOURce<hw>]:BB:BT00th:PCONfiguration:HFControl.....	99
[:SOURce<hw>]:BB:BT00th:PCONfiguration:LTAddress.....	99
[:SOURce<hw>]:BB:BT00th:PCONfiguration:PFControl.....	99
[:SOURce<hw>]:BB:BT00th:PCONfiguration:PLENgth.....	100
[:SOURce<hw>]:BB:BT00th:PCONfiguration:SNSValue.....	100
[:SOURce<hw>]:BB:BT00th:PCONfiguration:SRMode.....	101
[:SOURce<hw>]:BB:BT00th:PCONfiguration:VDATa.....	101
[:SOURce<hw>]:BB:BT00th:PRAMPing:FOFFset.....	92
[:SOURce<hw>]:BB:BT00th:PRAMPing:RFUNction.....	92
[:SOURce<hw>]:BB:BT00th:PRAMPing:ROFFset.....	93
[:SOURce<hw>]:BB:BT00th:PRAMPing:RTIME.....	93
[:SOURce<hw>]:BB:BT00th:PRESet.....	70
[:SOURce<hw>]:BB:BT00th:PTYPE.....	70
[:SOURce<hw>]:BB:BT00th:SETTing:CATalog.....	71
[:SOURce<hw>]:BB:BT00th:SETTing:DELeTe.....	71
[:SOURce<hw>]:BB:BT00th:SETTing:LOAD.....	71
[:SOURce<hw>]:BB:BT00th:SETTing:STORe.....	72
[:SOURce<hw>]:BB:BT00th:SLENgth.....	72
[:SOURce<hw>]:BB:BT00th:SRATe:VARiation.....	80
[:SOURce<hw>]:BB:BT00th:STATe.....	72
[:SOURce<hw>]:BB:BT00th:STIMing.....	72
[:SOURce<hw>]:BB:BT00th:TMODe.....	73
[:SOURce<hw>]:BB:BT00th:TRIGger:ARM:EXECute.....	81
[:SOURce<hw>]:BB:BT00th:TRIGger:EXECute.....	81
[:SOURce<hw>]:BB:BT00th:TRIGger:OBASeband:DELay.....	82
[:SOURce<hw>]:BB:BT00th:TRIGger:OBASeband:INHibit.....	82
[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut:DELay:FIXed.....	87
[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:DELay.....	88
[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:DELay:MAXimum?.....	88
[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:DELay:MINimum?.....	88
[:SOURce<hw>]:BB:BT00th:TRIGger:OUTPut<ch>:MODE.....	88

[:SOURCE<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:OFFTime.....	89
[:SOURCE<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:ONTime.....	89
[:SOURCE<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:PATtern.....	90
[:SOURCE<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:PULSe:DIVider.....	90
[:SOURCE<hw>]:BB:BT0oth:TRIGger:OUTPut<ch>:PULSe:FREQuency?.....	90
[:SOURCE<hw>]:BB:BT0oth:TRIGger:RMODe.....	83
[:SOURCE<hw>]:BB:BT0oth:TRIGger:SLENgth.....	83
[:SOURCE<hw>]:BB:BT0oth:TRIGger:SLUNit.....	84
[:SOURCE<hw>]:BB:BT0oth:TRIGger:SOURce.....	84
[:SOURCE<hw>]:BB:BT0oth:TRIGger[:EXTernal]:DELay.....	85
[:SOURCE<hw>]:BB:BT0oth:TRIGger[:EXTernal]:INHibit.....	86
[:SOURCE<hw>]:BB:BT0oth:TRIGger[:EXTernal]:SYNChronize:OUTPut.....	81
[:SOURCE<hw>]:BB:BT0oth:UPTYpe.....	128
[:SOURCE<hw>]:BB:BT0oth:USLength.....	129
[:SOURCE<hw>]:BB:BT0oth:VERSion?.....	73
[:SOURCE<hw>]:BB:BT0oth:WAVeform:CREate.....	74
[:SOURCE<hw>]:BB:BT0oth[:TRIGger]:SEQuence.....	86

Index

A

Access address	48
ACK	34
Acknowledgment	34
ACL+EDR	20
Activate Dirty Transmitter Test	59
Advertiser	40
Advertiser's address type	49
Advertiser's device address	53
Advertising channel table	46
Advertising channel type	38
Advertising event delay	43
Advertising event Interval	43
Advertising packet interval	43
Advertising settings	
Advertising event delay	43
Advertising event interval	43
Advertising packet interval	43
Scan Interval	43
Scan window	43
Test packet interval	57
Test packet payload length	57
Transmit window offset	44, 55
Transmit window size	44, 54
All data	33
Application cards	6
Application notes	6
Arm	24
Arm Trigger	81
Armed_Auto	23, 86
Armed_Retrigger	23, 86
Auto	23, 86

B

Baseband clipping	65
Baseband filter	62, 63
Basic rate+EDR	
Bluetooth	20
Bluetooth	
Basic rate+EDR	20
Default settings	19
Dirty transmitter test	21
Filter/Clipping	21
Low energy	20
Mode	20
Power ramping	21
Save/Recall	19
Set to default	19
State	19
Version	20
Brochures	6
BxT	62

C

Carrier frequency offset (DTT)	60
Center frequency	46
Channel index	46
Channel state	46
Channel type	38, 40, 46
Channel type advertising	38

Channel type data	38
Class of Device	36
Clipping Level	65
Clipping Mode	65, 75
Clipping state	65
Clock	
Mode	30
Multiplier	30
Source	30
Common trigger settings	22
Company ID	55
Connection event interval	44, 51
Connection settings	
Connection event interval	44, 51
LL connection mode	45
Long term key (hex)	46
No. of Tx Packets per Event	44
Selected data channel	46
Controler role master	40
Controler role slave	40
Controller role advertiser	40
Controller role initiator	40
Controller role scanner	40
Controller state	40
Controller's device address	53
Conventions	
SCPI commands	68
Coupled trigger settings	22
CRC initial	54
Crest factor	65
Current range without recalculation	28
Cut off frequency factor	62

D

Data channel table	46
Data channel type	38
Data length	
Packet configuration	35, 50
Data sheets	6
Data source	
Packet configuration	35, 50
Voice Field	36
Data whitening	33, 48
Data Whitening	117
Deactivate Dirty Transmitter Test	59
Default settings	
Bluetooth	19
Delay	
Marker	28
Trigger	26
Device address	33
Device's address type	49
Dirty transmitter test	
Bluetooth	21
Dirty Transmitter Test	
Carrier frequency offset	60
Frequency drift deviation	60
Frequency drift rate	60
Modulation index	61
Start phase	59
state	59
State (long table)	60

State (short table)	60	LL connection mode	45
Symbol timing error	60	LL connection timeout	51
Documentation overview	5	Logical transport address	34
DQPSK/8DPSK filter	63	Long term key (hex)	46
E		Low energy	
EIR Packet Follows	36	Bluetooth	20
Encrypted DIVERSifier(hex)	52	M	
Encrypted mode	45	Manual Trigger	81
Error code	55	Marker	
eSCO+EDR	20	Frame active part	27
Event configuration	40	Marker delay	28
Channel table	46	Marker mode	27
Execute Trigger	81	Master	40
External trigger		Measured external clock	30
Delay	26	MIC (message authentication Code)	45
F		Mode	
Fall offset	67	Bluetooth	20
Feature set length	53	Modulation index (DTT)	61
Filter	62, 63	Modulation index	64
Filter parameter		Modulation type	64
Roll off	62	N	
Filter/Clipping		NAK	34
Bluetooth	21	NAP	33
Filtering, Clipping	61	NESN start value	49
Fix marker delay to current range	28	No. of Tx Packets per Event	44
Fix marker to dynamic range	87	O	
Flow control (header)	34	Online help	5
Flow control (Payload)	35	Oversampling	63
Frame configuration	40	Oversampling Auto State	77
Channel table	46	Oversamplingt	76
Frames		P	
Sequence length	39	Packet configuration	
Frequency deviation	64	Access address	48
Frequency drift deviation	60	Acknowledgment	34
Frequency drift rate	60	Advertiser's address type	49
G		Center frequency	46
Gauss FSK filter	62	Channel index	46
Generate		Channel state	46
Waveform file (Bluetooth)	20	Channel type	46
Getting started	5	Class of Device	36
H		Company ID	55
Hide data channel mapping table	52	Connection event interval	44, 51
Hide data channel table	52	CRC initial	54
Hop length	52	Data whitening	33, 48
I		Device address	33
Impulse length	63	EIR Packet Follows	36
Impulse Length Auto State	76	Encrypted DIVERSifier(hex)	52
Initialization vector (hex)	53	Error code	55
Initiator	40	Feature set length	53
Initiator's address type	49	Flow control (header)	34
Initiator's device address	53	Flow control (Payload)	35
Installation	9	Hide data channel mapping table	52
Instrument security	6	Hide data channel table	52
L		Hop length	52
LAP	33	Initialization vector (hex)	53
LL connection event count	51	Initiator's address type	49
		Initiator's device address	53
		LAP	33
		LL connection event count	51

LL connection timeout	51
Logical transport address	34
NAP	33
NESN start value	49
Packet Length	36
Random vector (hex)	52
Scan Repetition Mode	36
Scanner's device address	53
SEQN start value	34
Session Key ID (hex)	53
Show data channel mapping table	52
Show data channel table	52
Simulation mode	33
Slave latency	51
Sleep clock accuracy	55
SN start value	49
Sub version number	56
UAP	33
Unknown type (hex)	53
Packet Configuration	32, 44
Version number	56
Packet Configuration:Data Whitening	117
Packet editor	33
Packet Length	36
Packet structure	56
Packet type	38
Packet Type	31
Payload type test packet	57
Power ramping	
Bluetooth	21, 66
Private address type	49, 53
Public address type	49, 53
R	
Ramp function	66
Ramp Function	92
Ramp time	66
Ramp Time	93
Random vector (hex)	52
Release notes	6
Retrigger	23, 86
Rise offset	66
Roll off factor	62
S	
Safety instructions	6
Save/Recall	
Bluetooth	19
Scan Interval	43
Scan Repetition Mode	36
Scan window	43
Scanner	40
Scanner's device address	53
SCO	20
Select data channel index	46
Selected data channel	46
SEQN start value	34
Sequence length	39
Sequence Length	32
Service manual	6
Session key ID (hex)	53
Set to default	
Bluetooth	19
Set to default (Dirty Transmitter Test)	59
Show data channel mapping table	52
Show data channel table	52
Signal duration unit	23
Signal generation status	23
Simulation mode	33
Single	23
Slave	40
Slave latency	51
Sleep clock accuracy	55
Slot Timing	32
SN start value	49
Standard settings	
Bluetooth	19
Start phase	59
State	
Bluetooth	19
State (long table)	60
State (short table)	60
Sub version number	56
Symbol rate variation	64
Symbol timing error (DTT)	60
Sync. output to external trigger	24
T	
Test packet configuration	59
Test packet interval	57
Test packet payload length	57
Test packet payload type	57
Transmit window offset	44, 55
Transmit window size	44, 54
Transport Mode	20
Transport Mode - ACL+EDR	20
Transport Mode - eSCO+EDR	20
Transport Mode - SCO	20
Trigger delay	26
Trigger Inhibit Other Baseband	82
Trigger mode	23
Trigger Mode	86
Trigger signal duration	23
Trigger source	24
Tutorials	5
U	
UAP	33
Unencrypted	45
Unknown type (hex)	53
User manual	5
V	
Version	
Bluetooth	20
Version number	56
W	
Waveform file	
Create (Bluetooth)	20
White papers	6