



BlueMod+P2x/HDP

Configuration Reference

Release r01

Trademarks

The Bluetooth[®] word mark and logos are owned by the Bluetooth SIG, Inc.. The NFC word mark and logos are owned by the NFC Forum. Any use of such marks by Stollmann E+V GmbH is under license. Other trademarks and trade names are those of their respective owners.

Table of contents

1	Introduction	6
2	Configuration Mode.....	7
2.1	How To Enter Configuration Mode	7
2.1.1	BlueMod+P2x/G2/HDP	7
2.1.2	BlueBSP+HDP	7
2.2	How To Use Configuration Mode	7
2.3	How To Do a Firmware Update in Configuration Mode	8
3	Configuration in Source Code	10
4	Configuration and Action Identifier	11
4.1	Device General Identifier.....	11
	BAUDRATE Baud Rate.....	11
	BTMODE Bluetooth Mode.....	12
	BCLASS Bluetooth Class of Device	12
	BLINKP Link Policy.....	15
	BNAME Local Device Name	15
	BOAD Bluetooth Own Device Address.....	16
	BPSM Scanning Capability	16
	BPSRI Page-Scan Repetition Interval.....	16
	BPSRM Page-Scan Repetition Mode	17
	BPSRW Page-Scan Repetition Window.....	18
	BROLE Device Role	18
	BSUPTIM Supervision Timeout	19
	DEFA Default Settings.....	19
	FLASH Load New Firmware	20
	PWRUNMODE Power Management Run Mode	20
	PWSTOPMODE Power Management Stop Mode	21
	QUIT, EXIT, GO Activate Parameter Changes	21
	RESET Reset Device	21
	RFMAXXPWR Maximum Output Power	22
	RSTMSG Startup Message.....	22

RSTTIM	Startup Timer	23
SAVE	Store Parameter Changes	23
SHOW	Show Parameter Settings	23
SHOWALL	Show All Parameter Settings.....	23
UICP	UICP Support.....	24
VER	Show Version String of Firmware	24
VERB	Show Version String of Bootloader.....	24
4.2	Device Security Identifier	25
BAUTHREQMTS	Authentication Requirements	25
BIOCAP	Device IO Capabilities.....	25
BND	Bond With a Bluetooth Device.....	26
BNDDEL	Delete Bonding Information.....	26
BNDLIST	Show Bonded Device List	27
BNDS	Storage Mode for Bonds	27
BNDSIZE	Bonded Devices List Size	27
BPAIRABLE	Pairable Mode.....	28
BPIN	Bluetooth Device PIN (Passkey)	28
4.3	HDP General Identifier	29
HDPCCINTO	HDP Control Incoming Timeout.....	29
HDPCCOUTTO	HDP Control Outgoing Timeout	29
HDPMCAPTO	HDP MCAP Timeout.....	30
HDPNAME	HDP Service Name	30
HDPRECONNECT	HDP/MCAP Reconnect	30
HDPRXAPDU	HDP Rx APDU Size	31
HDPSINT	HDP Sniff Interval	31
4.4	HDP Security Identifier.....	32
HDPAUTHORIN	HDP Authorization for Incoming	32
HDPAUTHOROUT	HDP Authorization for Outgoing.....	32
HDPMITMIN	HDP MITM Requirements Incoming.....	32
HDPMITMOUT	HDP MITM Requirements Outgoing	33
4.5	SPP General Identifier	34

SPPSINT	SPP Sniff Interval.....	34
4.6	SPP Security Identifier	35
SPPAUTHENOUT	SPP Authentication Outgoing	35
SPPAUTHOROUT	SPP Authorization Outgoing	35
SPPENCROUT	SPP Encryption Outgoing	36
SPPMITMOUT	SPP MITM Requirements Outgoing	36
4.7	DID General Identifier	37
IDSOURCE	Source of the Vendor ID.....	37
PRODUCTID	Product ID of the Device.....	37
PRODUCTVERS	Product Version of the Device	37
VENDORID	Vendor ID of the Device	38
5	History	39
6	Appendix A: Identifier List in Alphabetical Order	40

1 Introduction

This documentation describes the action and configuration identifier for the following Bluetooth devices:

- BlueMod+P24/G2 featuring HDP software version 1.550 or later
- BlueMod+P25/G2 featuring HDP software version 1.550 or later
- BlueBSP+HDP featuring HDP software version 3.000 or later

Three identifier types are defined:

- **Action:**
Identifies a configurator command that can be entered in configuration mode and initiates an action of the device (e.g. the command “bnd” initiates a bonding process with a remote Bluetooth device).
This configurator commands are only available on targets that do support the configurator command set.
- **Configuration:**
Identifies a configurator command that can be entered in configuration mode and can be used to configure or customize the device (e.g. parameter “bname” can be used to configure the Bluetooth device name of the device).
This configurator commands are only available on targets that do support the configurator command set.
- **Source:**
Identifies a global ANSI C variable that is part of the source code delivery and can be used to configure the device at compile time (e.g. ANSI C variable “bname” in a BSP source code package can be used to configure the Bluetooth device name of the device).
This configuration method is only available on targets that are delivered in source code and usually do not support the configurator command set.

2 Configuration Mode

2.1 How To Enter Configuration Mode

This chapter describes how to enter the configuration mode on different platforms.

2.1.1 BlueMod+P2x/G2/HDP

In online state the device uses LTP commands to communicate with the firmware of the device. To change the device parameters it is necessary to enter the configuration mode. To switch to the configuration mode use the following steps:

Reset the device. After receiving the “LTP Active Information” from the device, the “Exit LTP Request” has to be send. If this is successful, the firmware will response with a “#” character.

Action	Response
Device Reset	«LTP Active Information message»
Send «LTP Exit Request» 0x8C 0x00 0x00 0x05 0x00	«LTP Exit Response message» «LTP Exit Information message» «#»

We suggest to use a terminal program (i.e. HyperTerminal) to enter the commands.

Note: The explicit description of “LTP Exit Request”, “LTP Exit Response”, “LTP Exit Information” and “LTP Active Information” messages can be found in the “LTP Interface Specification”.

2.1.2 BlueBSP+HDP

In a BlueBSP+HDP application that supports the configurator command-set the configurator is entered using function `sbsp_SysUtilityStart()` of the SBSP API. For further information look at the document “SBSP API Reference Specification” in chapter “Access to configuration module”.

2.2 How To Use Configuration Mode

Each command line sent from the DTE to the DCE consists of a body and a terminator.

The body is a string of characters in the ASCII range 032-127. Control characters other than <CR> (carriage return; ASCII 013) and <BS> (back space; ASCII 008) in a command line are ignored.

The terminator is <CR>.

There is no distinction between upper-case and lower-case characters. A command line can have a maximum length of 80 characters. It is automatically discarded if the input is longer. Corrections are made using <BS>.

Configuration commands have the following syntax:

Syntax	Description
<command>	Read the current value of the command
<command>?	Display a help text for the command
<command>=<value>	Write the value of the command

To get an overview of the commands supported by the Bluetooth device, a number of special commands are available, such as:

Command	Description
help	Show all available commands
show	Show commonly used parameters and their values
showall	Show all configurable parameters and their values

The factory-default values of the parameters marked using bold letter format.

2.3 How To Do a Firmware Update in Configuration Mode

The firmware can be updated using the **FLASH** action command.

Applications developed using the BlueBSP+HDP can be flashed into the hardware using the “Serial Module Updater”. For further information look at the “Serial Module Updater User Guide”.

To update the firmware in configuration mode, perform the steps below:

- Set the serial speed on the terminal emulation to the baud rate the device is configured to.
- On a target that runs on LTP Interface Reset the device wait for **the LTP Act Info** message and send the **Exit LTP Request** (refer to chapter 2) to enter configuration mode.
- Send the **FLASH** command string to start the firmware update procedure.
 - Alternative: Send the command string **FLASH=<baud rate>** and set the terminal emulation accordingly.
- After a few seconds, the following message will be displayed:
“Erasing flash EPROM now. Please wait ...”
- A few seconds later the following message will be displayed:
“Start your XMODEM transfer now (Ctrl-X aborts) ...”
- Start the upload transfer. Search for the firmware file to load into the device and upload the file using the 1k X-Modem protocol.

Note:

The character “C” received is the request character of the X-Modem protocol. It will stop being output after the upload procedure has started.

-
- After loading the complete firmware into flash memory, the device will automatically perform a reset.
 - After reset the device will start with the baud rate stored in NVRAM or with the firmware default. Be aware that the terminal emulation has to be set accordingly.
 - Make sure the new firmware file is present and enter **VER** to check the firmware version number.
 - Send the following commands to get firmware default values: **DEFA 1, SAVE**.

If the firmware update fails for some reason (file upload is interrupted, wrong file etc.) the device will run in bootloader mode. In this mode, the device responds to all AT commands besides **AT** and **AT**FLASH** with "BOOTLOADER ACTIVE". Enter the **AT**FLASH** command again to repeat the firmware update procedure.

3 Configuration in Source Code

A BlueBSP+HDP package that supports the configuration device configuration at compile time includes a ANSI C source file that can be changed by the BlueBSP+HDP customer and will become part of the firmware binary after a re-compile process is initiated by the customer.

For further descriptions (e.g. location of that ANSI C source file within the project sources) please refer to the BlueBSP+HDP documentation.

4 Configuration and Action Identifier

4.1 Device General Identifier

This chapter lists all general device related identifier.

BAUDRATE	Baud Rate
-----------------	------------------

Syntax: **baudrate | baudrate=<value>**

Identifier type: **configuration**

This command determines the baud rate for the UART.

Value	Description
1	1,200 bps
2	2,400 bps
3	4,800 bps
4	9,600 bps
5	19,200 bps
6	38,400 bps
7	57,600 bps
8	115,200 bps
9	230,400 bps

Note: This command is not available on the BlueBSP+HDP platform where the baud rate for the UART is configured with the SBSP API.

BBTMODE	Bluetooth Mode
----------------	----------------

Syntax: **bbtmode | bbtmode=<value>**

Identifier type: **configuration | source**

This parameter defines if the local device is in pairable mode. If this parameter is set to 0, all pairing attempts from remote devices will be automatically rejected.

Value	Description
0	Bluetooth 2.1 support disabled <i>Note: This mode is only for testing, it shall not be used for product level configuration</i>
1	Bluetooth 2.1 support enabled
2	Bluetooth 2.1 debug support enabled <i>Note: This mode is only for testing, it shall not be used for product level configuration</i>

BCLASS	Bluetooth Class of Device
---------------	---------------------------

Syntax: **bclass | bclass=<value>**

Identifier type: **configuration | source**

This command allows the manipulation of the Bluetooth class of device/service (CoD).

Note: Changing the class of device affects profile-specific requirements and may influence interoperability. Change this only if you are certain of all side effects.

The CoD consists of 3 octets (24 bits). Bits 23 through 13 define the service class, bits 12 through 8 define the major device class, and bits 7 through 2 define the minor device class. Bits 1 and 0 are reserved and must always be set to 0.

The service class field is a bit field; no bit, one bit or several bits can be set, depending on the profile requirements.

Service classes:

Bit	Description
13	Limited discoverable mode
14	Reserved
15	Reserved
16	Positioning (location identification)
17	Networking (LAN, ad-hoc, ...)
18	Rendering (printing, speaker, ...)
19	Capturing (scanner, microphone, ...)
20	Object transfer (v-inbox, v-folder, ...)
21	Audio (speaker, microphone, headset service, ...)
22	Telephony (cordless telephony, modem, headset service, ...)
23	Information (Web server, WAP server, ...)

The major device class field represents the highest level of granularity for defining a Bluetooth device. The main function of a device is used to determine the major device class setting (bits 12 through 8 in the CoD).

Major device class:

Bit	12	11	10	9	8	Description
	0	0	0	0	1	Computer (desktop, notebook, PDA, organizer, ...)
	0	0	0	1	0	Phone (cellular, cordless, payphone, modem, ...)
	0	0	0	1	1	LAN/network access point
	0	0	1	0	0	Audio/video (headset, speaker, stereo, video display, VCR, ...)
	0	0	1	0	1	Peripheral (mouse, joystick, keyboards, ...)
	0	0	1	1	0	Imaging (printing, scanner, camera, display, ...)
	0	0	1	1	1	Wearable
	0	1	0	0	0	Toy
	0	1	0	0	1	Health
	1	1	1	1	1	Uncategorized, specific device code not specified

The minor device class field (bits 7 through 2 in the CoD) can be interpreted only in the context of the major device class (but independently of the service class field). The meaning of the bits may therefore change depending on the major device class.

Minor device class values for the "Computer" major device class:

Bit	7	6	5	4	3	2	Description
	0	0	0	0	0	0	Uncategorized, specific device code not assigned
	0	0	0	0	0	1	Desktop workstation
	0	0	0	0	1	0	Server-class computer
	0	0	0	0	1	1	Laptop
	0	0	0	1	0	0	Handheld PC/PDA (clam shell)
	0	0	0	1	0	1	Palm-sized PC/PDA
	0	0	0	1	1	0	Wearable computer (watch-sized)

Minor device class values for the “Phone” major device class:

Bit	7	6	5	4	3	2	Description
	0	0	0	0	0	0	Uncategorized, specific device code not assigned
	0	0	0	0	0	1	Cellular
	0	0	0	0	1	0	Cordless
	0	0	0	0	1	1	Smart phone
	0	0	0	1	0	0	Wired modem or voice gateway
	0	0	0	1	0	1	Common ISDN access

Minor device class values for the “Health” major device class:

Bit	7	6	5	4	3	2	Description
	0	0	0	0	0	0	Undefined
	0	0	0	0	0	1	Blood pressure monitor
	0	0	0	0	1	0	Thermometer
	0	0	0	0	1	1	Weighing scale
	0	0	0	1	0	0	Glucose meter
	0	0	0	1	0	1	Pulse oximeter
	0	0	0	1	1	0	Heart/pulse rate monitor
	0	0	0	1	1	1	Health data display

For the description of other minor device classes, refer to the Bluetooth specification.

Note: Values can be entered in hexadecimal notation (0x...). Leading zeros can be omitted.

Examples:

bclass=0x0900	Health, undefined
bclass=0x0210	Wired modem
bclass=0x090C	Weighing scale

BLINKP	Link Policy
---------------	--------------------

Syntax: **blinkp | blinkp=n**

Identifier type: **configuration | source**

This command defines the link policy (bit mask) for device requests for a new Bluetooth link.

Bit	2	1	0	Function
	0	0	1	Enable role switch support
	1	0	0	Enable sniff support

Examples:

blinkp=1	Device supports role switch feature
blinkp=4	Device supports sniff feature
blinkp=5	Device supports role switch feature and sniff feature

BNAME	Local Device Name
--------------	--------------------------

Syntax: **bname | bname=<name>**

Identifier type: **configuration | source**

This command allows the modification of the local device name. The device name is shown on a remote Bluetooth device during device/service discovery. It is limited to 50 characters.

The device name can contain a format string to include the device's own address or parts of it in the name.

Format: "%[<s>][<d>a"

“%”	Identifier start format string
<s>	Character separator on byte order (optional)
<d>	Number (1-12) of digits included in device name (optional, default is 4)
“a”	Identifier end format string

Examples: Device address = "0123456789AB"

bname=BlueMod+P2x/G2/HDP %:4a	Display on remote end: BlueMod+P2x/G2/HDP 89:AB
bname=BlueMod+P2x/G2/HDP %4a	Display on remote end: BlueMod+P2x/G2/HDP 89AB
bname=BlueMod+P2x/G2/HDP %:3a	Display on remote end: BlueMod+P2x/G2/HDP 9:AB
bname=BlueMod+P2x/G2/HDP %3a	Display on remote end: BlueMod+P2x/G2/HDP 9AB
bname=BlueMod+P2x/G2/HDP %:12a	Display on remote end: BlueMod+P2x/G2/HDP 01:23:45:67:89:AB

BOAD	Bluetooth Own Device Address
-------------	-------------------------------------

Syntax: **boad**

Identifier type: **action**

This command reads the Bluetooth devices' own device address.

Note: This command is read only.

BPSM	Scanning Capability
-------------	----------------------------

Syntax: **bpsm | bpsm=<mode>**

Identifier type: **configuration | source**

This parameter controls the visibility of the device and its ability to accept connections and its reaction to paging and/or inquiry requests. If set to "0" all paging/inquiry requests from other Bluetooth devices will be ignored, and the RF receive part of the Bluetooth device is disabled.

Mode	Description
0	Page scan and inquiry scan are disabled; the Bluetooth device is not connectable and not discoverable
1	Inquiry scan is enabled; the Bluetooth device is discoverable, but not connectable
2	Page scan is enabled; the Bluetooth device is connectable, but not discoverable
3	Page scan and inquiry scan are enabled; the Bluetooth device is connectable and discoverable

BPSRI	Page-Scan Repetition Interval
--------------	--------------------------------------

Syntax: **bpsri | bpsri=<value>**

Identifier type: **configuration**

This parameter modifies the page-scan repetition interval (in milliseconds) of the Bluetooth device.

Value	Description
$n=11\dots2560$	Use page-scan repetition interval of n ms
1280	Use page-scan repetition interval of 1280 ms

Note: Due to internal conversion it can happen that the value is not set to the exact entered value. In this case it will be set to the next lower value.

BPSRM	Page-Scan Repetition Mode
--------------	----------------------------------

Syntax: **bpsrm | bpsrm=<mode>**

Identifier type: **configuration | source**

This parameter modifies the page-scan repetition mode of the Bluetooth device. Possible values according to the Bluetooth base-band specification are:

Mode	Description
0	R0 – continuous scan
1	R1 – 1.28 s interval, 11.25 ms window
2	R2 – 2.56 s interval, 11.25 ms window
255	Values of BPSRI (interval) and BPSRW (windows) parameter applies to page-scan repetition mode. This setting is only allowed for identifier type “configuration”

Continuous scanning decreases the connect time down to about 200 ms:

- R0: about 200 ms
- R1: about 2.2 s
- R2: about 3.5 s

BPSRM should be set to the same value for the scanning and the paging device.

Note: Setting the page-scan repetition mode to continuous scan significantly increases the power consumption.

BPSRW	Page-Scan Repetition Window
--------------	------------------------------------

Syntax: **bpsrw | bpsrw=<value>**

Identifier type: **configuration**

This parameter modifies the page-scan repetition window (in milliseconds) of the Bluetooth device.

Value	Description
<i>n</i> =11...2560	Use page-scan repetition window of <i>n</i> ms
11	Use page-scan repetition window of 11 ms

Note: Due to internal conversion it can happen that the value is not set to the exact entered value. In this case it will be set to the next lower value.

BROLE	Device Role
--------------	--------------------

Syntax: **brole | brole=<mode>**

Identifier type: **configuration | source**

This command controls the role the device requests for a new Bluetooth link. If no specific action in this respect is taken, the device that initiates the connection will be the master. The terminating device (the one accepting a connection) will be the slave.

Mode	Description
0	Slave required
1	Slave preferred
2	Don't care
3	Master preferred
4	Master required

If the role is set to "required", the device enforces a switch. If this fails, the link is disconnected.

"Preferred" means that the device tries to perform a switch but does not care about the result.

BSUPTIM	Supervision Timeout
----------------	---------------------

Syntax: **bsuptim | bsuptim=<value>**

Identifier type: **configuration | source**

This command defines the supervision timeout the device requests for a new Bluetooth link.

Value	Description
$n=1\dots65535$	Supervision timeout after $n*0.625$ ms (n is decimal value)
32000	Supervision timeout after 20 s

Note: The supervision timeout can only be set by the Piconet master.

Example:

bsuptim=32000	Supervision timeout of 20 s
----------------------	-----------------------------

DEFA	Default Settings
-------------	------------------

Syntax: **defa=<value>**

Identifier type: **action**

The factory-default values will be loaded. For storing values in non-volatile memory, use the **SAVE** command.

Value	Description
0	Set general parameters to factory defaults
1	Set general and proprietary parameters to factory defaults

FLASH	Load New Firmware
--------------	--------------------------

Syntax: **flash[=<baud rate>]**

Identifier type: **action**

This commands loads new firmware into the Bluetooth device. The current firmware will be overwritten. The firmware will be stored in the previously used part of flash memory.

For the optional parameter <baud rate> the following values are allowed:

Value	Description
115200	use 115,200 bps for flashing
230400	use 230,400 bps for flashing
460800	use 460,800 bps for flashing
921600	use 921,600 bps for flashing

While uploading, the following checks will be performed:

- File transfer protocol is XMODEM 1K
- An overall firmware checksum is used
- The firmware type written in the module header of the firmware must be compatible with the hardware type and permitted firmware types (stored inside the bootloader).

For details about the firmware update procedure, see page 8 “Firmware Update”.

PWRUNMODE	Power Management Run Mode
------------------	----------------------------------

Syntax: **pwruntime | pwruntime=<mode>**

Identifier type: **configuration**

This command defines the power manager run mode policy. Please notice this is only a run policy, the power manager will select a system policy that will be equal or better in terms of system performance to the **PWRUNMODE** parameter.

Mode	Description
0	RunDoNotCare
0x10	RunOverdrive
0x20	RunOperational
0x30	RunReducedSpeed
0x40	RunLowSpeed

PWSTOPMODE	Power Management Stop Mode
-------------------	-----------------------------------

Syntax: **pwstopmode | pwstopmode=<mode>**

Identifier type: **configuration**

This command defines the power manager stop mode policy. Please notice this is only a stop policy, the power manager will select a system policy that will be equal or better in terms of system performance to the **PWSTOPMODE** parameter.

Mode	Description
0	StopDoNotCare
0x80	NoStop
0x90	StopNormal
0xA0	StopReduced
0xB0	StopSleep
0xC0	StopStandby

QUIT, EXIT, GO	Activate Parameter Changes
-----------------------	-----------------------------------

Syntax: **quit | exit | go**

Identifier type: **action**

These commands activate the current parameter settings and exits the BlueRS+ Configurator (without storing the parameters in non-volatile memory).

RESET	Reset Device
--------------	---------------------

Syntax: **reset**

Identifier type: **action**

This command resets the whole functionality of the Bluetooth device by a forced hardware reset (like power off/on).

*Note: See also the **RSTTIM** parameter.*

RFMAXTXPWR	Maximum Output Power
-------------------	-----------------------------

Syntax: **rfmaxtxpwr | rfmaxtxpwr=<value>**

Identifier type: **configuration | source**

This command sets the maximum output power of the Bluetooth radio of the device. The firmware ensures the device never speaks more loudly than this value. The chip uses this for page, inquiry and their scan responses. This is also the power used for new connections.

Value	Description
-128	Use factory default maximum output power value
-20...16	Maximum output power in dBm

All other values in the range of -128 to 127 could be set with this command as well, but the equal or next lower value from the power table will be set internally. Furthermore the value will be set to a value amongst minimum and maximum output power value of the device.

Example:

rfmaxtxpwr=0	The maximum output power will be set to 0 dBm
--------------	---

RSTMSG	Startup Message
---------------	------------------------

Syntax: **rstmsg | rstmsg=<mode>**

Identifier type: **configuration**

After power-on, the following startup message can be displayed with a fixed speed of 9,600 bps, 8 data bits, no parity, 1 stop bit:

+++ Press <CR>,<CR>,<ESC>,<ESC> to enter BlueRS+ configurator +++

Mode	Description
0	Inactive, no startup message will be send after power-on
1	Active, startup message will be send after power-on

RSTTIM	Startup Timer
---------------	----------------------

Syntax: **rsttim | rsttim=<value>**

Identifier type: **configuration**

This command defines the startup delay after a reset. Within this period, the internal configuration mode can be entered after a reset of the device (see **RSTMSG**).

Value	Description
<i>n</i> =3...255	Startup delay <i>n</i> *50 ms
3	Startup delay 150 ms

SAVE	Store Parameter Changes
-------------	--------------------------------

Syntax: **save**

Identifier type: **action**

This command stores the current set of parameters in non-volatile memory.

SHOW	Show Parameter Settings
-------------	--------------------------------

Syntax: **show**

Identifier type: **action**

This command displays the current set of parameters.

SHOWALL	Show All Parameter Settings
----------------	------------------------------------

Syntax: **showall**

Identifier type: **action**

This command displays all accessible parameters.

UICP	UICP Support
-------------	---------------------

Syntax: **uicp | uicp=<value>**

Identifier type: **configuration**

This parameter devices if the UICP protocol is enabled for the communication UART.

Note that UICP requires specific functionality of its peer. Due to that, the UICP shall only be enabled if the peer supports this protocol. If UICP is enabled, but the peer does not support the according functionality, malfunction of the overall system might be the result.

Value	Description
0	UICP is disabled
1	UICP is enabled

Note: This parameter is only available on platforms that support UICP.

VER	Show Version String of Firmware
------------	--

Syntax: **ver**

Identifier type: **action**

This command displays detailed information about the firmware version and Bluetooth device type.

Example:

ver	PG2a010 V1.210 Jul 3 2009 09:51:01
-----	------------------------------------

VERB	Show Version String of Bootloader
-------------	--

Syntax: **verb**

Identifier type: **action**

This command displays detailed information about the bootloader version and Bluetooth device type.

Example:

verb	BOOT LOADER V2.07 090513 0722-00-01
------	-------------------------------------

4.2 Device Security Identifier

This chapter lists all device security related identifier.

BAUTHREQMTS	Authentication Requirements
--------------------	------------------------------------

Syntax: **bauthreqmts | bauthreqmts=<value>**

Identifier type: **configuration | source**

This parameter defines the security requirements for a new pairing attempt. If the local device is in pairable mode (see **BTMODE** parameter) and a remote device initiates a pairing procedure, this requirements apply.

Value	Description
0	No MITM, no store
1	With MITM, no store
2	No MITM, dedicated bond
3	With MITM, dedicated bond
4	No MITM, general bond
5	With MITM, general bond

BIOCAP	Device IO Capabilities
---------------	-------------------------------

Syntax: **biocap | biocap=<mode>**

Identifier type: **configuration | source**

This command defines the IO capabilities of the local device that can be used for SSP (Secure Simple Pairing) security procedures.

Mode	Description
0	Display only
1	Display Yes/No
2	Keyboard only
3	No IO capabilities

Examples:

biocap=1	Device provides a numerical display and Yes/No buttons that can be used for SSP procedures
----------	--

BND	Bond With a Bluetooth Device
------------	-------------------------------------

Syntax: **bnd=<device>[,PIN]**

Identifier type: **action**

This command initiates a bonding process with a remote Bluetooth device.

The optional <PIN> parameter is the passkey used to generate the bond. If <PIN> is omitted, the passkey configured with the **BPIN** command is used.

If the bonding succeeds the Bluetooth device returns “SUCCESS”, otherwise “FAILED”.

Bonded devices are stored in the **bndlist**. The number of bonded devices which can be stored in the bonded device list is defined in the **BNDSize** parameter. If all entries are used, a new bond will overwrite the least recently used one.

Example:

bnd=010203040506,1234	Bond with device with address 010203040506 and use 1234 as PIN
-----------------------	--

BNDDEL	Delete Bonding Information
---------------	-----------------------------------

Syntax: **bnddel=<mode>**

Identifier type: **action**

This command deletes the bonding information stored by the Bluetooth device.

Mode	Description
b<x>	Delete the bond of the device with device selector b<x> from bndlist ; the remaining bond list entries get new index numbers
all	Delete all bonded devices

BNDLIST Show Bonded Device List

Syntax: **bndlist**

Identifier type: **action**

This command shows information about the devices bonded with the Bluetooth device.

Example:

bndlist	b01: 000461811C0F THA-W2K b02: 00802500211A Loopback RS+E OK
---------	--

BNDS Storage Mode for Bonds

Syntax: **bnds | bnds=<mode>**

Identifier type: **configuration | source**

This command controls the storage mode for bonding information (link keys).

Mode	Description
0	Bonds persists for the duration of the authenticated connection
1	Bonds are permanently stored in the NVRAM of the Bluetooth device
2	Bonds are stored in RAM of the Bluetooth device and will be lost after power cycle
3	Bonds are expected to be stored externally so bonding information will be requested via "AuthResultRequestInd" BlueHDP or LTP command

BNDSIZE Bonded Devices List Size

Syntax: **bndsize | bndsize=<value>**

Identifier type: **configuration | source**

This command reduces the number of entries (1...4) the bonded-device list can hold. The default size is **4**.

Note: Modification of this parameter will delete all entries in the bonded-device list.

Example:

bndsize=1	Limit the number of bonded devices to 1
-----------	---

BPAIRABLE	Pairable Mode
------------------	---------------

Syntax: **bpairable | bpairable=<mode>**

Identifier type: **configuration | source**

This parameters controls the pairable mode of the device. New pairings can only be generated when pairable mode is enabled.

Mode	Description
0	Pairable mode is disabled
1	Pairable mode is enabled

BPIN	Bluetooth Device PIN (Passkey)
-------------	--------------------------------

Syntax: **bpin | bpin=<old_pin>,<new_pin>**

Identifier type: **configuration**

This command sets the PIN for establishing a connection in restricted mode. The PIN has a maximum length of 16 alphanumeric characters; the factory-default is "0000". This PIN is always checked if:

- the Bluetooth device is set to restricted mode and a connection is established
- the Bluetooth device is not set to restricted mode but the remote Bluetooth device is set to restricted mode.

After a pairing has taken place (PINs successful exchanged), Bluetooth links can be established between these paired devices independent of the settings for restricted mode and the PIN.

To reset the PIN to the factory-default, use the **DEFA 1** command (all parameters will be set to the factory-default).

Note: Prefer PINs composed of only decimal digits. Do not assume that a remote device with limited user-interface capabilities supports alphabetic characters.

Example:

bpin=0000,1234	Set PIN to 1234 (old PIN was 0000)
----------------	------------------------------------

4.3 HDP General Identifier

This chapter lists all general Health Device Profile (HDP) related identifier.

HDPCCINTO	HDP Control Incoming Timeout
------------------	-------------------------------------

Syntax: **hdppccinto | hdppccinto=<value>**

Identifier type: **configuration | source**

This parameter defines a timeout interval in milliseconds for idle MCAP control channel connections if no data channels are connected and the connection was initiated from the remote device.

If this parameter is set to a value other than zero, the MCAP control channel will be disconnected automatically if no MCAP commands are exchanged for the defined interval.

Value	Description
0	No timeout defined
1 – 65535	Timeout in milliseconds

Example:

hdppccinto=10000	Idle timeout of 10s
------------------	---------------------

HDPCCOUTTO	HDP Control Outgoing Timeout
-------------------	-------------------------------------

Syntax: **hdppccoutto | hdppccoutto=<value>**

Identifier type: **configuration | source**

This parameter defines a timeout interval in milliseconds for idle MCAP control channel connections if no data channels are connected and the connection was initiated from the local device.

If this parameter is set to a value other than zero, the MCAP control channel will be disconnected automatically if no MCAP commands are exchanged for the defined interval.

Value	Description
0	No timeout defined
1 – 65535	Timeout in milliseconds

Example:

hdppccoutto=10000	Idle timeout of 10s
-------------------	---------------------

HDPMCAPTO	HDP MCAP Timeout
------------------	-------------------------

Syntax: **hdpmcapto | hdpmcapto=<value>**

Identifier type: **configuration | source**

This parameter defines a timeout for MCAP responses from the remote device. If this parameter is set to a value other than zero and a MCAP command is initiated from the local device but not responded by the remote device within the timeout interval the MCL (including all data channels that might be established) will be disconnected.

Value	Description
0	No timeout defined
1 – 65535	Timeout in milliseconds

Example:

hdpmcapto=30000	MCAP timeout is 30s
------------------------	---------------------

HDPNAME	HDP Service Name
----------------	-------------------------

Syntax: **hdpname | hdpname=<string>**

Identifier type: **configuration | source**

This command sets the service name of the Health Device Profile SDP record.

Example:

hdpname=Stollmann HDP	Sets the name of the HDP SDP record to “Stollmann HDP”
------------------------------	--

HDPRECONNECT	HDP/MCAP Reconnect
---------------------	---------------------------

Syntax: **hdpreconnect | hdpreconnect=<value>**

Identifier type: **configuration | source**

This parameter allows to enable/disable the support for the HDP/MCAP reconnect feature.

Value	Description
0	Reconnect support disabled
1	Reconnect support enabled

HDP RX APDU SIZE	HDP Rx APDU Size
-------------------------	-------------------------

Syntax: **hdprxapdu | hdprxapdu=<value>**

Identifier type: **configuration | source**

This parameter defines the maximum IEEE APDU size in bytes the local device can receive and re-assemble. Valid values are 48 to 65535 bytes.

Value	Description
48 – 65535	Maximum receive IEEE APDU size

Example:

hdprxapdu=1000	A maximum APDU size of 1000 bytes is accepted
-----------------------	---

HDP SNIFF INTERVAL	HDP Sniff Interval
---------------------------	---------------------------

Syntax: **hdpsint | hdpsint=<value>**

Identifier type: **configuration | source**

This parameter defines the sniff interval (in 0.625 millisecond slots) for HDP connections. If this parameter is set to a value other than zero, the local device will request to set the Bluetooth connection to sniff mode with the defined value.

Please be aware that this parameter does not grant that the addressed Bluetooth connection will go to sniff mode with the value requested.

Legal values are in the range of 6 to 1104 slots.

Value	Description
0	No sniff requested
6-1104	Sniff requested

Example:

hdpsint=100	A sniff interval of 62.5 milliseconds will be requested
--------------------	---

Note: This parameter is only available on targets that support Bluetooth sniff.

4.4 HDP Security Identifier

This chapter lists all Health Device Profile (HDP) security related identifier.

HDPAUTHORIN HDP Authorization for Incoming

Syntax: **hdppauthorin | hdppauthorin=<value>**

Identifier type: **configuration | source**

This parameter defines if authorization for incoming HDP connections is required. If this parameter is set to 1, an user level authorization is initiated for each incoming HDP connection.

Value	Description
0	No authorization required
1	Authorization is required

HDPAUTHOROUT HDP Authorization for Outgoing

Syntax: **hdppauthorout | hdppauthorout=<value>**

Identifier type: **configuration | source**

This parameter defines if authorization for outgoing HDP connections is required. If this parameter is set to 1, an user level authorization is initiated for each outgoing HDP connection.

Value	Description
0	No authorization required
1	Authorization is required

HDPMITMIN HDP MITM Requirements Incoming

Syntax: **hdppmitmin | hdppmitmin=<value>**

Identifier type: **configuration | source**

This parameter defines if MITM (man-in-the-middle) protection is required for an incoming HDP connection. If this parameter is set to 1, only connections that provide MITM protection according to SSP definitions are accepted.

Value	Description
0	No MITM protection required
1	MITM protection required

HDPMITMOUT	HDP MITM Requirements Outgoing
-------------------	---------------------------------------

Syntax: **hdpmitmout | hdpmitmout=<value>**

Identifier type: **configuration | source**

This parameter defines if MITM (man-in-the-middle) protection is required for an outgoing HDP connection. If this parameter is set to 1, only connections that provide MITM protection according to SSP definitions are allowed.

Value	Description
0	No MITM protection required
1	MITM protection required

4.5 SPP General Identifier

This chapter lists all general Serial Port Profile (SPP) related identifier.

SPPSINT	SPP Sniff Interval
----------------	---------------------------

Syntax: **sppsint | sppsint=<value>**

Identifier type: **configuration | source**

This parameter defines the sniff interval (in 0.625 millisecond slots) for SPP connections. If this parameter is set to a value other than zero, the local device will request to set the Bluetooth connection to sniff mode with the defined value.

Please be aware that this parameter does not grant that the addressed Bluetooth connection will go to sniff mode with the value requested.

Legal values are in the range of 6 to 1104 slots.

Value	Description
0	No sniff requested
6-1104	Sniff requested

Example:

sppsint=100	A sniff interval of 62.5 milliseconds will be requested
-------------	---

Note: This parameter is only available on targets that supports SPP and Bluetooth sniff.

4.6 SPP Security Identifier

This chapter lists all Serial Port Profile (SPP) security related identifier.

SPPAUTHENOUT	SPP Authentication Outgoing
---------------------	------------------------------------

Syntax: **sppauthenout | sppauthenout=<value>**

Identifier type: **configuration | source**

This parameter defines if authentication for outgoing SPP connections is required. If this parameter is set to 1, an authentication is initiated for each outgoing SPP connection. This parameter applies only for connections to devices that do not support SSP. Connections to remote devices that support SSP are always authenticated.

Value	Description
0	No authentication required
1	Authentication is required

Note: This parameter is only available on platforms that support SPP.

SPPAUTHOROUT	SPP Authorization Outgoing
---------------------	-----------------------------------

Syntax: **sppauthorout | sppauthorout=<value>**

Identifier type: **configuration | source**

This parameter defines if authorization for outgoing SPP connections is required. If this parameter is set to 1, an user level authorization is initiated for each outgoing SPP connection.

Value	Description
0	No authorization required
1	Authorization is required

Note: This parameter is only available on platforms that support SPP.

SPPENCROUT	SPP Encryption Outgoing
------------	-------------------------

Syntax: **sppencrout | sppencrout=<value>**

Identifier type: **configuration | source**

This parameter defines if encryption for outgoing SPP connections is required. If this parameter is set to 1, encryption is requested for each outgoing SPP connection.

Please note that encryption of a Bluetooth link requires a successful authentication.

Value	Description
0	No encryption required
1	Encryption is required

Note: This parameter is only available on platforms that support SPP.

SPPMITMOUT	SPP MITM Requirements Outgoing
------------	--------------------------------

Syntax: **sppmitmout | sppmitmout=<value>**

Identifier type: **configuration | source**

This parameter defines if MITM (man-in-the-middle) protection is required for an outgoing SPP connection. If this parameter is set to 1, only connections that provide MITM protection according to SSP definitions are allowed.

Value	Description
0	No MITM protection required
1	MITM protection required

Note: This parameter is only available on platforms that support SPP.

4.7 DID General Identifier

This chapter lists all Device Information Profile (DID) related identifier.

IDSOURCE	Source of the Vendor ID
-----------------	--------------------------------

Syntax: **idsource | idsource=<value>**

Identifier type: **configuration | source**

This command sets the source for the vendor ID of the device. Possible values are:

Value	Description
1	BT assigned ID
2	USB assigned ID

Example:

idsource=2	Sets vendor ID source to "USB assigned ID"
------------	--

PRODUCTID	Product ID of the Device
------------------	---------------------------------

Syntax: **productid | productid=<value>**

Identifier type: **configuration | source**

The HDP mandates that each device provides an DIP (Device IP Profile) service record. This command sets the product ID of the device.

Example:

productid=0x0000	Sets the product ID of the device to 0x0000
------------------	---

PRODUCTVERS	Product Version of the Device
--------------------	--------------------------------------

Syntax: **productvers | productvers=<value>**

Identifier type: **configuration | source**

This command sets the product version of the device.

Example:

productvers=0x0000	Sets the product version of the device to 0.0.0
productvers=0x0213	Sets the product version of the device to 2.1.3

VENDORID	Vendor ID of the Device
----------	-------------------------

Syntax: **vendorid | vendorid=<value>**

Identifier type: **configuration | source**

The HDP mandates that each device provides an DIP (Device IP Profile) service record. This command sets the vendor ID of the device.

Example:

vendorid= 0x0742	Sets the vendor ID of the device to 0x0742
-------------------------	--

5 History

Version	Release Date	By	Change description
r01d01	27.06.2011	ka	Initial version, stripped down from BlueMod+HDP Command Reference
r01d02	28.06.2011	Ta	Formal corrections
r01	28.06.2011	bg	released

6 Appendix A: Identifier List in Alphabetical Order

This section contains a list of all action and configuration commands and parameters in alphabetical order of their name. Additionally three columns defining the type for each identifier.

Three identifier types are defined:

- **Action:**
Identifies a configurator command that initiates an action of the device (e.g. the command “bnd” initiates a bonding process with a remote Bluetooth device). This configurator commands are only available on targets that do support the configurator command set.
- **Configuration:**
Identifies a configurator command that can be used to configure or customize the device (e.g. parameter “bname” can be used to configure the Bluetooth device name of the device). This configurator commands are only available on targets that do support the configurator command set.
- **Source:**
Identifies a global ANSI C variable that is part of the source code delivery and can be used to configure the device at compile time (e.g. ANSI C variable “bname” in a BSP source code package can be used to configure the Bluetooth device name of the device). This configurator commands are only available on targets that do **NOT** support the configurator command set.

Name	Action	Configuration	Source	Page
BAUDRATE		x		11
BAUTHREQMTS		x	x	25
BBTMODE		x	x	12
BCLASS		x	x	12
BIOCAP		x	x	25
BLINKP		x	x	15
BNAME		x	x	15
BND	x			26
BNDDEL	x			26
BNDLIST	x			27
BNDS		x	x	27
BNDSIZE		x	x	27
BOAD	x			16
BPAIRABLE		x	x	28
BPIN		x		28
BPSM		x	x	16
BPSRI		x		16
BPSRM		x	x	17
BPSRW		x		18
BROLE		x	x	18
BSUPTIM		x	x	19
DEFA	x			19
EXIT (QUIT, EXIT, GO)	x			21
FLASH	x			20
GO (QUIT, EXIT, GO)	x			21
HDPAUTHORIN		x	x	32
HDPAUTHOROUT		x	x	32
HDPCCINTO		x	x	29
HDPCCOUTTO		x	x	29
HDPMCAPTO		x	x	30
HDPMITMIN		x	x	32
HDPMITMOUT		x	x	33
HDPNAME		x	x	30
HDPRECONNECT		x	x	30

Name	Action	Configuration	Source	Page
HDPRXAPDU		x	x	31
HDPSINT		x	x	31
IDSOURCE		x	x	37
PRODUCTID		x	x	37
PRODUCTVERS		x	x	37
PWRUNMODE		x		20
PWSTOPMODE		x		21
QUIT (QUIT, EXIT, GO)	x			21
RESET	x			21
RFMAXTXPWR		x	x	22
RSTMSG		x		22
RSTTIM		x		23
SAVE	x			23
SHOW	x			23
SHOWALL	x			23
SPPAUTHENOUT		x	x	35
SPPAUTHOROUT		x	x	35
SPPENCROUT		x	x	36
SPPMITMOUT		x	x	36
SPPSINT		x	x	34
UICP		x		24
VENDORID		x	x	38
VER	x			24
VERB	x			24

Stollmann Entwicklungs- und Vertriebs-GmbH
Mendelssohnstraße 15 D
22761 Hamburg
Germany

Phone: +49 (0)40 890 88-0
Fax: +49 (0)40 890 88-444
E-mail: info@stollmann.de
www.stollmann.de