BlueMod+SR
AT Command Reference

80507ST10752A Rev. 14 – 2016-12-07

APPLICABILITY TABLE

PRODUCT
BlueMod+SR

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

Notice

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

Copyrights

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

Computer Software Copyrights

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

Usage and Disclosure Restrictions

License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY “OTHER CODE”), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Contents

1. Introduction

1.1. Scope

This document specifies the command interface for the BlueMod+SR firmware.

1.2. Audience

Readers of this document should be familiar with the BlueMod+SR module and their ease of controlling by means of AT commands.

1.3. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit Technical Support Center (TTSC) at:

TS-SRD@telit.com

Alternatively, use:

http://www.telit.com/en/products/technical-support-center/contact.php

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

To register for product news and announcements or for product questions contact Telit Technical Support Center (TTSC).

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Document Organization

This document contains the following chapters (sample):

"Chapter 1: "Introduction" provides a scope for this document, target audience, contact and support information, and text conventions.

"Chapter 2: "Chapter two" gives an overview of important features of the product.

"Chapter 3: "Chapter three" describes in details the AT commands of the product.

1.5. Text Conventions

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

1.6. Related Documents

[1] BlueMod+SR Hardware User Guide, 1VV0301275

[2] BlueMod+SR Software User Guide, 1VV0301276

[3] Bluetooth 4.0 Core Specification

[4] BlueMod+S - BlueMod+SR Delta Reference

2. Features

The BlueMod+SR supports AT command mode and multiplexing mode.

2.1. AT Command Mode

Each command line consists of a prefix, a body and a terminator.

All command lines begin with the prefix **AT** (ASCII 065, 084) or **at** (ASCII 097, 116).

The body is a string of characters in the ASCII range 032-255. Control characters other than <CR> (carriage return; ASCII 013) and <BS> (back space; ASCII 008) in a command line are ignored.

Note: The control characters are configurable via S registers.

- *Carriage return character (CR)* *S3 register*
- *Line feed character (LF)* *S4 register*
- *Back space character (BS)* *S5 register*

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>. Multiple commands on the same command line are not allowed.

Commands have the following syntax:

Syntax	Description
AT<command>=<value><CR>	Write the value of the command

Responses are sent back to the host and can be any of the following:

Responses	Description
<CR><LF>value<CR><LF>	Read only value (e.g. AT+BOAD)
<CR><LF> list entry 1<CR><LF> list entry 2<CR><LF> … list entry n<CR><LF> <CR><LF>	List value (e.g. AT+BNDLIST)
<CR><LF>OK<CR><LF>	Successful final message
<CR><LF>ERROR<CR><LF>	Error message, command not supported

NOTE:

Exceptions of this syntax are marked separately.

2.2. Multiplexing Mode (MUX)

The multiplexing mode is used to handle incoming and outgoing data of different remote endpoints and command data.

Data has to be sent and are received in the following framing (all values in hexadecimal format):

Name	Description	Length	Value
Start	Start of frame	8 bit	CC
Channel ID	Channel identifier	8 bit	00 – FF
Length	Length of data	8 bit	-
Data	Max. 255 bytes data	Min. 0 byte Max. 255 bytes	-

The start byte is used to detect the start of a frame.

The channel ID determines the channel to send data to. This can be the link ID of an active link in case of classic Bluetooth, the data channel of a BLE endpoint or the AT command interface (value FF).

Channel ID	Description
00	Classic Bluetooth (SPP)
01	Terminal I/O
02 … FE	Customer defined GATT services
FF	AT command interface

The length field sets the length of the payload to send or received in bytes.

The data field consists of the payload data to send or receive.

Start of frame, channel ID, length and data are always transmitted in direct, binary form. AT commands have to be sent to the channel ID FF, simply prefixed with start of frame, FF, and length byte. Data received from the AT command interface are marked by channel ID FF. Line editing using backspace is not available in multiplexing mode.

Examples:

CC 01 0B 31 32 33 34 35 36 37 38 39 30 0D	Send data “1234567890<CR>” to channel 1 via MUX protocol
CC FF 06 41 54 49 39 39 0D	Send AT command “ATI99<CR>” via MUX protocol
CC FF 06 0D 0A 4F 4B 0D 0A	Receive response “<CR><LF>OK<CR><LF>” from AT command interface via MUX protocol

If the multiplexing mode is enabled the commands **ATA** and **ATH** and the result messages RING, CONNECT and NO CARRIER includes the additional parameter “channel ID”.

For Classic Bluetooth and Terminal I/O connections, the “channel ID” is assigned while connection setup (by messages RING for incoming connections and CONNECT for outgoing connections) and is valid until the connection is terminated (message NO CARRIER).

For customer defined GATT services, the “channel ID” is assigned while service definition for each defined characteristic separately and is valid until the BlueMod+SR is resetted.

2.3. Escape Sequence

To enter the command mode during an active data connection the following sequence (escape sequence) can be used:

<delay time [1]><+><+><+><delay time>

The time interval between each of the three plus signs must not exceed 1 second. The escape sequence remains transparent to the remote device.

NOTE:

The escape sequence character is configurable via S2 register.

2.4. Bluetooth Low Energy Connection Establishment Procedure

With Bluetooth low energy the parameters **AT+LECONINTMAX** and **AT+LESLAVELAT** are used for central connection establishment. The connection supervision timeout used for connection establishment is calculated due to conform to the Bluetooth core specification.

On a peripheral, 200 ms after an incoming connection establishment the peripheral will check if the used connection parameters are compatible with the parameters **AT+LECONINTMIN**, **AT+LECONINTMAX** and **AT+LESLAVELAT**. If the parameters are not compatible, the peripheral will update the connection parameters automatically to the configured parameters. The connection supervision timeout used by automatic parameter update is calculated due to conform to the Bluetooth core specification. If connection parameter update fails, the peripheral retries the connection parameter update. The maximum number of attempts is 3 and the time between the retries is 5 seconds.

2.5. Hangup

All active connections can be disconnected by a rising edge on the HANGUP pin (GPIO4).

[1] Delay time defined in the S12 register (see page 25)

3. COMMAND SUMMARY

The description of the commands is structured into the following parts:

- General commands
- Classic Bluetooth commands
- Bluetooth Low Energy (BLE) specific commands
- Generic Attribute Profile (GATT) specific commands

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

3.1. General

%B	Baud Rate

AT syntax: **AT%B<value>**

This command determines the baud rate for the UART.

The following standard baud rates are supported:

Value	Description
4	9,600 bps
5	19,200 bps
6	38,400 bps
7	57,600 bps
8	115,200 bps
9	230,400 bps
22	460,800 bps
23	921,600 bps

Additionally to the standard baud rates described above, it is possible to set custom baud rates to match a non standard baud rate of the application.

Custom baud rates can be set in a range from 9,600 – 921,600 bps. To set a custom baud rate write the desired value in the command.

Examples:

AT%B9	Set baud rate to 230,400 bps
AT%B330000	Set baud rate to 330,000 bps

NOTE:

Information regarding the deviation of the real baud rate to the value set can be found in the *BlueMod+SR Hardware User Guide [1]*.

&F Load Factory Defaults

AT syntax: **AT&F<value>**

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

Value	Description
0	Set all parameters except **AT+BNDLIST, AT+BNDSIZE** and **AT+UICP** to factory defaults
1	Set all parameters to factory defaults

NOTE:

Some restored values require an additional reset to get active (e.g. AT+LETIO, AT+UICP).

&W Store Active Configuration

AT syntax: **AT&W**

The active configuration is stored in non-volatile memory.

A Accept Incoming Call

AT syntax normal mode: **ATA**
AT syntax MUX mode: **ATA <channel ID>**

This command accepts an incoming call if automatic call acceptance is not set (register S0=0). An incoming call is always signaled with the unsolicited response “RING“ or code “2“, even if automatic call acceptance is selected.

The described behavior is only valid for Classic Bluetooth connections. A Bluetooth Low Energy connection will be accepted automatically (see also register S0 description).

Examples:

ATA	Accepts connection in normal mode
ATA 0x00	Accepts connection in MUX mode at channel ID 0x00
ATA 0x01	Accepts connection in MUX mode at channel ID 0x01

+BARSSI Read Absolute RSSI Value

AT syntax normal mode: **AT+BARSSI**
AT syntax MUX mode: **AT+BARSSI=<link ID>**

Supported since firmware version 1.412

This read-only command contains the absolute receive signal strength value acquired on the last poll interval (see **AT+BSTPOLL** command). The value is returned as a signed byte in hexadecimal notation. If the update interval is 0, the returned value will also be 0.

The result of the **AT+BARSSI** command is a signed byte in hexadecimal notation. That means if the highest bit (bit 7) is 1, the number is negative, and you get the value by building the two's complement.

Value	Receive Signal
0x10	10 dBm
0x00	0 dBm
0xFF	-1 dBm
0xFE	-2 dBm
…	
0xEC	-20 dBm
0xD8	-40 dBm
0xC4	-60 dBm
0xB0	-80 dBm

The higher the value (hexadecimal value), the better the receive signal. Most common results are in the range of -20 dBm to -80 dBm.

When there is no Bluetooth connection the result isn't defined, amongst other things thats because there is no "neutral" value.

+BIOCAP SSP I/O Capabilities

AT syntax: **AT+BIOCAP=<value>**

This command sets the input and output capabilities of the device used for SSP.

Value	Description
0	Display only
1	Display Yes/No
2	Keyboard only
3	No input no output
4	Display and keyboard

NOTE:

By using AT+BIOCAP=4 with a Classic Bluetooth connection the BlueMod+SR signals I/O capability “Keyboard only” for an outgoing bond request and “Display only” for an incoming bond request to the remote device.

+BMITM SSP Man in the Middle Protection

AT syntax: **AT+BMITM=<value>**

This command controls the man in the middle (MITM) protection of the device during SSP. It has to be set in context with **AT+BIOCAP** command. In SSP there are scenarios where MITM protection is not possible.

Value	Description
0	Man in the middle protection disabled
1	Man in the middle protection enabled

In case the user choose a scenario where MITM protection is not possible but one of the communication devices is configured to **AT+BMITM=1** (MITM protection enabled), the pairing is refused.

Possible combinations of I/O capabilities and the possibility of MITM protection are described in the *BlueMod+SR Software User Guide [2]*.

NOTE:

This command influences the MITM protection for SPP and Terminal I/O, but not for GATT Server.
To activate the settings for Terminal I/O it is necessary to store the settings (AT&W) and perform a reset (AT+RESET).

+BMUX Activate Multiplexing Mode

AT syntax: **AT+BMUX=<value>**

Supported since firmware version 1.310

This command is used to activate the multiplexing mode protocol. It is only allowed if no lower layer Bluetooth connection exists.

In the default configuration the device is working in normal AT mode. Setting **AT+BMUX=1** enables the multiplexing mode.

NOTE:

After receiving "OK" in the response of the AT+BMUX=1 command all subsequent commands have to be entered in multiplexing frame format.

The multiplexing mode is not stored persistent. To disable the multiplexing mode the device must be reset.

In multiplexing mode extended result codes are always active (see chapter 4.2).

+BNAME Local Device Name

AT syntax: **AT+BNAME=<name>**

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

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

For classic Bluetooth the name length is limited to 30*) characters. In BLE mode the name length is limited to 25**) characters.

*) *Length extended from 19 to 30 characters since firmware version 1.531*

**) *Length extended from 19 to 25 characters since firmware version 1.544. In firmware versions prior 1.544 the name in BLE advertising is truncated to the first 7 characters*

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

"%"	Identifier start format element
<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 element

Examples: Device address = "0123456789AB"

AT+BNAME=BM+SR %4a	Display on remote end: BM+SR 89AB
AT+BNAME=BM+SR %4a	Display on remote end: BM+SR 89AB
AT+BNAME=BM+SR %:3a	Display on remote end: BM+SR 9:AB
AT+BNAME=BM+SR %3a	Display on remote end: BM+SR 9AB
AT+BNAME=BM+SR %:12a	Display on remote end: BM+SR 01:23:45:67:89:AB

+BNDDEL Delete Bonding Information

AT syntax: **AT+BNDDEL=<value>**

This command deletes the bonding information stored by the BlueMod+SR.

Value	Description
Bluetooth address	Delete the bond of the device with specified address from the bonded-device list
*	Delete all bonded devices from the bonded-device list

+BNDLIST Show Bonded Device List

AT syntax: **AT+BNDLIST**

This command shows information about the devices bonded with the BlueMod+SR.

Each Bluetooth Classic entry in the **bonded-device list** contains the Bluetooth address, the linktype (see chapter 4.1) and the Bluetooth friendly name of the bonded device.

Each Bluetooth LE entry in the **bonded-device list** contains the Bluetooth address, the linktype (see chapter 4.1) and the role of the bonded device*) ("C" for client or "P" for peripheral).

*) *Feature supported since firmware version 1.544*

There may be exist two entries for one device if it supports client and peripheral role both.

Example:

AT+BNDLIST	00802507C08D 0x00 BlueMod+B20 C0:8D 0080254800DA 0x00 BM+SR 00DA 0080254800DA 0x02 C 0080254800DA 0x02 P 9C04EB06ACA2 0x03 P OK

+BNDS Storage Mode for Bonds

AT syntax: **AT+BNDS=<mode>**

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 BlueMod+SR

NOTE:

By setting AT+BNDS to 0 the bonded-device list is deleted.

The bonding information is stored in the module flash. If your application does not need to store bonds switch this parameter to 0, to protect the module flash from unnecessary clear and write operations. Every flash has a limited number of clear cycles.

+BNDSIZE Bonded Devices List Size

AT syntax: **AT+BNDSIZE=<value>**

This command reduces the number of devices (1…8) the bonded-device list can hold.

The BlueMod+SR can store up to 8 devices. The default size is **8**.

NOTE:

Modification of this parameter will delete all devices in the bonded-device list.

Two entries of the same BLE device are counted as one device.

Example:

AT+BNDSIZE=1	Limit the number of bonded devices to 1

+BOAD Bluetooth Own Device Address

AT syntax: **AT+BOAD**

This command reads the Bluetooth devices' own device address.

NOTE:

This command is read only.

+BPAIRMODE Configure Pairable Mode

AT syntax: **AT+BPAIRMODE=<mode>**

Supported since firmware version 1.502

This command controls the pairable mode of the BlueMod+SR.

When set to "0" the module is only connectable for clients stored in the bonded device list (**AT+BNDLIST**). New pairing requests will be rejected.

Mode	Description
0	No LE/BR pairing allowed, BlueMod+SR advertises TIO as "functional"
1	LE/BR Pairing allowed, BlueMod+SR advertises TIO as "bondable and functional"

NOTE:

This command restricts the access only to security enabled characteristics.

For Terminal I/O this means AT+LETIO has to be set to 1.

For the GATT Server the permission parameters of the command AT+LEATTRIB have to be set to a value that requires authentication (e.g. "AT+LEATTRIB=charval,uuid=2A19,perm=0022,len=1" with "perm" for read with authentication permitted and write with authentication permitted).

+BRSSIOUT RSSI Output at I2C Interface

AT syntax normal mode: **AT+BRSSIOUT=<if>[,i2cadr=<value1>]**
AT syntax MUX mode: **AT+BRSSIOUT= linkid,<if>[,i2cadr=<value1>]**

Supported since firmware version 1.412

This command controls the RSSI output at additional interfaces. At the moment the RSSI output is available only at the I2C interface. By default the RSSI output at the I2C interface is disabled. The value is not stored non volatile.

If	Description
0	RSSI output deactivated
1	RSSI output at I2C interface

The BlueMod+SR works as an I2C master. The address of the slave has to be set by using the parameter i2cadr (hexadecimal format). Value range is all values in 8-bit area.

To address a slave device at address 0x80 the parameter ic2adr has to be set to 0x80.

The BlueMod+SR transmits the RSSI values (format like **AT+BARSSI** command) to the slave address defined in parameter i2cadr. The value is updated after the time defined in **AT+BSTPOLL** parameter.

The RSSI output is in the format (3 bytes): <i2cadr> <channel ID><rssivalue>

In normal mode the channel ID is always 0x00. It is not intended that the slave sends requests to the master. The slave shall only read the data conform to the I2C specification and confirming it with an ACK.

Example:

AT+BRSSIOUT=1,i2cadr=0x80	Enable RSSI output at I2C interface, set I2C slave address to 0x80

NOTE:

The I2C address 0xAA is reserved for the NFC tag and cannot be used for RSSI output.

+BSSPCONF Secure Simple Pairing Confirmation

AT syntax: **AT+BSSPCONF <Bluetooth address>,<value>**

If a SSP is initiated and MITM is active (**AT+BMITM=1**), depending on the I/O capabilities (**AT+BIOCAP**) the AT interface generates an event SSPCONF and asks the user for confirmation.

Event: SSPCONF <Bluetooth address> <SSP passkey> ?

The user has to confirm the SSP passkey with the above command. If no confirmation is sent by the user within the bonding timeout or in case of active reject, the SSP is rejected with NO CARRIER message.

Value	Description
0	Reject SSP request
1	Accept SSP request

Example:

SSPCONF 00802507C08D 794851 ? AT+BSSPCONF 00802507C08D,1 OK RING CONNECT	Receive SSP pairing request Send SSP pairing confirmation

+BSSPPIN SSP Passkey Response

AT syntax: **AT+BSSPPIN <Bluetooth address>,<SSP passkey>**

If a SSP is initiated and MITM is active (**AT+BMITM=1**), depending on the I/O capabilities (**AT+BIOCAP**) the AT interface generates an event SSPPIN and asks the user for the SSP passkey.

Event: SSPPIN <Bluetooth address> ?

The user has to answer this request with the SSP passkey displayed on the remote device.

Example:

SSPPIN 00802507C08D ? AT+BSSPPIN 00802507C08D,314546 OK RING CONNECT	Receive SSP passkey request Send SSP passkey response

+BSSPDBG SSP Debug Mode

AT syntax: **AT+BSSPDBG=<value>**

This command allows to enable the SSP debug mode of the device. This mode is required to trace the SSP Bluetooth connection by using a sniffer.

Value	Description
0	SSP Debug mode off
1	SSP Debug mode on

NOTE:

SSP debug mode is for tracing purposes only and shall be deactivated for normal operation. Otherwise the connection may be unsecure.

+BSTPOLL Update Interval for Radio Statistics

AT syntax: **AT+BSTPOLL=<value>**

Supported since firmware version 1.412

This command configures the polling interval (in ms) for the **AT+BARSSI** command. A changed value becomes active for the next connection, so it needs to be set before establishing the connection. The polling interval applies to all connections (Classic and Low Energy).

Value	Description
0	No polling, statistics disabled
n=10…65535	Update statistics every *n* ms

Example:

AT+BSTPOLL=500	Update statistics every 500 ms

D Initiate Bluetooth Link

AT syntax: **ATD <brad> [,<ux> | ,TIO | ,SPP]**

This command addresses a Bluetooth device directly via its address. If a connection to a Bluetooth 2.0 device requiring the legacy security mode, the PIN must have been set up using **AT+BPIN** command.

Param.	Description
brad	Called Bluetooth remote device address (12 hex digits)
ux	With x=UUID of a service (4 hex digits)

If the device is configured to central role and initiates a Terminal I/O connection to a peripheral device, it shall use the identifier TIO.

If no identifier is given in the dial string, the device tries to establish a classic Bluetooth SPP connection.

Any character input while the BlueMod+SR is dialing will cancel the dialing procedure.

Spaces are ignored within a dial string.

NOTE:

The remote Bluetooth device has to be determined before issuing this link request. This can be done in the following ways:

- Get it manually by reading it from the sticker of the remote Bluetooth device.
- Request the address and service by using the **AT+BINQ** command.

Examples:

ATD 0080371443AB	Establish SPP connection to Bluetooth device 0080371443AB
ATD 0080371443AB,TIO	Establish Terminal I/O connection to Bluetooth device 0080371443AB

+D Start/Stop an Autodial Procedure

AT syntax: **AT+D=<value>**

Supported since firmware version 1.531

This command allows starting or stopping the autodial procedure manually. The configured parameters in AT+DSET and AT+DPARAMS will be used.

Additionally this command reports the status of the autodial procedure.

Value	Description
Start	Start an autodial procedure with configured parameters
Stop	Stop an active autodial procedure
?	Report the status of the autodial procedure (report format: status,number,count)

AT+D=Start reports ERROR if no dial string is configured in **AT+DSET**.

AT+D=? reports status, number, count followed by OK.

Report type	Description
Status	Off: Autodial procedure not running CALL: During call timeout (defined in AT+DPARAMS) DELAY: During pause (defined in AT+DPARAMS) between connection attempts
Number	Number of active dial string
Count	Number of active dial sequence (0 if endless autodial is configured in AT+DPARAMS)

Examples:

AT+D=Start OK	Start autodial procedure
AT+D=Stop OK	Stop autodial procedure
AT+D=? Off,0,0 OK	Autodial procedure not running
AT+D=? CALL,1,3 OK	Autodial procedure in call timeout, dial string no. 1, connection count 3
AT+D=? DELAY,2,4 OK	Autodial procedure in call delay, dial string no. 2, connection count 4

+DMODE Autodial Mode

AT syntax: **AT+DMODE=<value>**

Supported since firmware version 1.531

This command controls the autodial behavior that applies after reset. If set to 255 the autodial procedure starts automatically after resetting the module.

After changing the value an **AT&W** command is required to store the settings.

Value	Description
0	Autodial off
255	Dial all valid entries round robin

AT+DMODE=255 reports ERROR if no dial string is configured in **AT+DSET**.

+DPARAMS Autodial Parameters

AT syntax: **AT+DPARAMS=<timeout>,<pause>,<count>**

Supported since firmware version 1.531

This command allows to setup the call timeout, the delay timer and the connection attempt counter for the autodial procedure. The call timer is started with the dial attempt. If the dial attempt didn't succeed the delay timer is started. If the number of configured connection attempts is reached, the autodial procedure stops.

Parameter	Description
Timeout	0…255 (default **10**): call timeout in seconds
Pause	0…255 (default **20**): delay between connection attempts in seconds
Count	**0** (default): endless retry 1...255: number of connection attempts

NOTE:

To store the parameters over reset or power off use the AT&W command.

Examples:

AT+DPARAMS=10,20,0	Setup autodialer to dial endless with call timeout of 10 seconds and pause of 20 seconds
AT+DPARAMS=30,5,100	Setup autodialer to dial 100 times with call timeout of 30 seconds and pause of 5 seconds

+DSET Set an Autodial String

AT syntax: **AT+DSET=<number>,<dial string>**

Supported since firmware version 1.531

This command allows to setup up to 3 different dial strings for the autodialer. Each dial string can contain up to 20 characters.

Parameter	Description
Number	Number of dial string to be set (1, 2 or 3)
Dial string	Dial string as used for ATD command

Deleting a dial string is done by setting it empty.

NOTE:

To store the parameters over reset or power off use the AT&W command.

Examples:

AT+DSET=1,008025540108,TIO	Set dial string entry 1 with dial string 008025540108,TIO
AT+DSET=2,0080254800FC,SPP	Set dial string entry 2 with dial string 0080254800FC,SPP
AT+DSET=1,	Delete dial string entry 1

E Local Echo

AT syntax: **ATE<value>**

This command selects the local echo in command mode.

Value	Description
0 or empty	No local echo
1	Local echo on in command phase

H Disconnect

AT syntax normal mode: **ATH**
AT syntax MUX mode: **ATH <channel ID>**

This command disconnects the existing Bluetooth connection.

Examples:

ATH	Disconnects connection in normal mode
ATH 0x00	Disconnects connection in MUX mode at channel ID 0x00
ATH 0x01	Disconnects connection in MUX mode at channel ID 0x01

I Display Version Information

AT syntax: **ATI<value>**

Displays different information about version number and settings.

Value	Description
0 or empty	Returns the device name (e.g. “BM+SR %4a“)
1	Returns “0”
2	Returns “OK”
3	Returns the version string: “V1.xyz”
4	Returns the manufacturers name: “Stollmann E+V GmbH”
5	Returns “ERROR”
6	Returns the copyright string: “(c) Copyright Stollmann E+V GmbH”
7	Returns “OK”
8	Returns “ERROR”
9	Returns “OK”
99	Returns the firmware creation date

+IOACFG Config of Pin IOA

AT syntax: **AT+IOACFG=<value>**

Supported since firmware version 1.103

This command configures the function of the IOA pin (GPIO8). GPIO8 can be configured as output signal “Device Ready". The signal is low active, a low level shows the device ready to process commands and establish Bluetooth connections after startup.

Value	Description
0	Disconnected (no function, lowest power consumption)
1	Output: Signal “Device Ready”

+IOBCFG Config of Pin IOB

AT syntax: **AT+IOBCFG=<value>**

Supported since firmware version 1.502

This command configures the function of the IOB pin (GPIO3). GPIO3 can be configured as output signal "Data Carrier Detect" (DCD). The signal is low active. A low level signals that the device is connected on Terminal I/O level to a remote device.

Value	Description
0	Disconnected (no function, lowest power consumption)
1	Output: Signal “DCD”
2 *)	Output: Low level if a lower layer connection (classic or BLE) is active Output: High level if no lower layer connection (classic or BLE) is active

*) *Supported since firmware version 1.540*

NOTE:

In MUX mode two connections (classic and BLE) can be established in parallel. In this scenario the IOB pin is signaling the first established connection and the last released connection.

+IOCCFG Config of Pin IOC

AT syntax: **AT+IOCCFG=<value>**

Supported since firmware version 1.531

This command configures the function of the IOC pin (GPIO2). GPIO2 can be configured as output signal set to high or low level.

Value	Description
0	Input: Analog (no function, lowest power consumption)
10	Output: Set GPIO2 to low level
11	Output: Set GPIO2 to high level

+LOAD Load Stored Parameter Setting

AT syntax: **AT+LOAD**

This command loads all parameters stored in non-volatile RAM.

+NFCMODE Set NFC Mode

AT syntax: **AT+NFCMODE=<value>**

Supported since firmware version 1.502

This command sets the operation mode of the NFC interface.

The BlueMod+SR provides the possibility to connect the NFC tag NXP: NT3H1101 at the I2C interface at address 0xAA.

Value	Description
0	NFC interface off
1	Automatic mode

NOTE:

If automatic mode is activated make sure that pairable mode is enabled (AT+BPAIRMODE=1).

O Return to Online State

AT syntax: **ATO**

If the BlueMod+SR is in command mode after issuing an escape sequence while a connection is active, ATO returns the BlueMod+SR to data mode.

Q Suppress Results

AT syntax: **ATQ<value>**

This command allows/suppresses result codes and messages.

Value	Description
0 or empty	Enable result messages after command input
1	Suppress result messages after command input

+RESET Reset Device

AT syntax: **AT+RESET**

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

NOTE:

The AT+RESET command has no response message.

+RFMAXTXPWR Maximum Output Power

AT syntax: **AT+RFMAXTXPWR=<value>**

Supported since firmware version 1.412

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 value for Classic Bluetooth (page, inquiry and scan responses) and BLE (advertising, scan response). This is also the power used for new connections.

Value	Description
-128	Use factory default maximum output power
-128…127	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:

AT+RFMAXTXPWR=0	The maximum output power will be set to 0 dBm

S AT S Register

AT syntax: **ATSx=<value>**

This command configures the S register settings.

Mode	Description
Sx?	Show the current setting of register Sx
Sx=1	Set register Sx to 1

AT command S register set (all values in decimal format):

Register	Value	Description
S0	0 **1** 2…n	No automatic call acceptance; acceptance of an incoming call is controlled by the data terminal (**ATA** command after RING) Immediate call acceptance Call acceptance after n RING messages *Note: Setting of S0 register applies for classic Bluetooth connections only. In case of BLE connections always one RING is signalized and automatic call acceptance is set.*
S2	**43**	Character for escape sequence
S3	**13**	Carriage-return character
S4	**10**	Line-feed character
S5	**08**	Backspace character
S7	**30**	Wait time for carrier (in s)
S12	**100**	Delay time by using the escape sequence in 10 ms increments
S30	0 **1**	Escape characters will not be transmitted to the remote device Escape characters will be transmitted to the remote device

+UICP Set UART Interface Control Protocol

AT syntax: **AT+UICP=<mode>**

This command sets the mode of the UART Interface Control Protocol (UICP).

Mode	Description
0	UICP off
1	UICP on

V Result Message Format

AT syntax: **ATV<value>**

This command determines the format of the result messages.

Value	Description
0 or empty	Result message is presented numerically (followed by <CR>)
1	Result message is presented as text

W Extended Result Codes

AT syntax: **ATW<value>**

This command enables/disables extended result codes.

Value	Description
0 or empty	Result message is presented without extended result codes
1	Result message is presented with extended result codes (include error causes)

3.2. Classic Bluetooth

+BCLASS	Bluetooth Class of Device

AT syntax: **AT+BCLASS=<value>**

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:

AT+BCLASS=**0x1F00**	Uncategorized, specific device code not specified
AT+BCLASS=0x0210	Wired modem
AT+BCLASS=0x090C	Weighing scale

+BCRYPT Encryption

AT syntax: **AT+BCRYPT=<mode>**

Enable or disable the encryption of the information transferred via Bluetooth.

Mode	Description
0	Encryption disabled
1	Encryption enabled

+BFCON Fast Connection Mode

AT syntax: **AT+BFCON=<mode>**

Supported since firmware version 1.521

This command controls the fast connection mode for incoming connections. If enabled the page scan (default: 1.28 sec) and inquiry scan (default: 2.56 sec) intervals are set to 100 ms.

Mode	Description
0	Fast connection mode disabled
1	Fast connection mode enabled

NOTE:

- Using fast connection mode will increase the power consumption.
- To optimize the fast connection mode any LE services (e.g. AT+LETIO=0) should be switched off.

+BINQ Search Bluetooth Devices

AT syntax: **AT+BINQ**

With this command an automatic search for all discoverable Bluetooth devices will be initiated. The discovery will last for approx. 10 seconds.

As a result, a list will be output containing the Bluetooth address of the visible devices in range, the class of device, the RSSI and (in case of EIR) the Bluetooth friendly name.

Example:

AT+BINQ	008025081528 0x00001F00 -056 BlueMod+B20 15:28 008025149CD9 0x00000900 -062 BlueMod+P2x/G2/HDP 008025084FCC 0x00001F98 -058 0080251292DA 0x00001F00 -080 00802507C099 0x00001F00 -060 BlueMod+B20 C0:99 00802500211A 0x00001F00 -055 008025480003 0x00001F00 -061 BM+SR 0003 008025480001 0x00001F00 -045 BM+SR 0001 00802512D637 0x00001F00 -066 00043E622288 0x00001F00 -043 00802510D0DE 0x00001F00 -053 OK

NOTE:

Any character input while the BlueMod+SR is searching will abort the search procedure.

+BINQSERV Discover Services of Device

AT syntax: **AT+BINQSERV=<brad> [,ux]**

This command performs a service discovery on a single device.

Param.	Description
brad	Bluetooth remote device address (12 digits)
ux	Specific UUID to search for

The command needs a Bluetooth device address as parameter.

If the remote device has no visible services, or no service of the device match the UUID specified in the command string, the answer contains Bluetooth address and device name only.

Examples:

Get service information for device with Bluetooth address 008025081541	
AT+BINQSERV=008025081541	008025081541 BlueMod+B20 15:41 uuid: 1101 sChan: 0001 sName: SPP uuid: 1106 sChan: 0002 sName: OBEX FTP OK
Get service information for device with Bluetooth address 008025081541 for service 0x1101	
AT+BINQSERV=008025081541,u1101	008025081541 BlueMod+B20 15:41 uuid: 1101 sChan: 0001 sName: SPP OK
Get service information for device with Bluetooth address 008025081541 for service 0x1102 (device does not support service 0x1102)	
AT+BINQSERV=008025081541,u1102	008025081541 BlueMod+B20 15:41 OK

+BND Bond With a Bluetooth Device

AT syntax: **AT+BND=<bdaddr>**

This command initiates a bonding process with a remote Bluetooth device. <bdaddr> is the device address of the remote device.

If the bonding succeeds BlueMod+SR returns “SUCCESS”, otherwise “FAILED”.

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

Example:

AT+BND=010203040506	Bond with device with address 010203040506

+BOSRV Own Service Profile (UUID)

AT syntax: **AT+BOSRV=<uuid>**

This parameter defines the service the BlueMod+SR reports to a remote device.

UUID	Description
0x1101	Serial port

+BPIN Bluetooth Device PIN (Passkey)

AT syntax: **AT+BPIN=<PIN>**

This command sets the PIN for establishing a connection to a Bluetooth 2.0 device (legacy pairing). The PIN has a maximum length of 16 alphanumeric characters; the factory-default is "0000".

This PIN is checked if:

- the BlueMod+SR establishes a connection to a Bluetooth 2.0 device
- the BlueMod+SR is connected from a Bluetooth 2.0 device if either device requires security

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

To reset the PIN to the factory-default, use the **AT&F1** 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:

AT+BPIN=1234	Set PIN to 1234

+BPSM Scanning Capability

AT syntax: **AT+BPSM=<mode>**

This parameter controls the visibility of the device, that is 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 BlueMod+SR is disabled.

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

+BSIZE Block Size

AT syntax: **AT+BSIZE=<value>**

This command defines the maximum length of a data block transmitted via Bluetooth.

Value	Description
n=16…330	Block size of *n* bytes is used
330	Block size of 330 bytes is used

+BSMODE Sniff Mode

AT syntax: **AT+BSMODE=<value>**

Supported since firmware version 1.310

This command sets the supported sniff mode. The default value is 1.

Value	Description
0	No sniff support
1	Passive sniff support
2	Active sniff using 250*) ms sniff interval
3	Active sniff using 500*) ms sniff interval

*) *The Bluetooth stack creates two values related to the mentioned value, a minimum and a maximum value. The minimum value is 12.5% lower and the maximum value is 12.5% higher than the mentioned value.*

Sniffing is used to reduce power consumption during an active classic Bluetooth connection. When a Bluetooth link is in sniff mode both connected devices can enter regularly a power-saving state for a short time interval. The length of this interval is called “sniff interval” and is negotiated between the connected devices.

Note that in order to enter sniff mode, both devices need to support sniff mode.

If active sniff support is enabled, the BlueMod+SR tries to setup a sniff interval with the value mentioned in the table above.

NOTE:

The sniff interval value has an impact on the propagation delay and maximum throughput on the Bluetooth link.

+BSNAME Local Service Name

AT syntax: **AT+BSNAME=<name>**

This command defines the service name of the BlueMod+SR serial port. The service name is shown on a remote Bluetooth device during a service discovery sequence. It is limited to 23 characters.

Examples:

AT+BSNAME=SPP	Own service name of the serial port is “SPP”
AT+BSNAME=RS+SrvName	Own service name of the serial port is “RS+SrvName”

3.3. Bluetooth Low Energy

+LEADDATA	Setup Advertise Data for Customized Advertising

AT syntax: **AT+LEADDATA=<value$_1$> .. <value$_n$>**

Supported since firmware version 1.310

This command is used to setup the advertise data for a customized advertising.

This parameter value cannot be stored using **AT&W**.

Value$_k$ represents an octet in hexadecimal format, k <= 31.

The coding of the data is according to the *Bluetooth 4.0 Core Specification / Vol. 3, Part C, Chapter 11 and 18 (Length/Type/Value coding) [3]*.

Example:

CC FF 1B 41 54 2B 4C 45 41 44 44 41 54 41 3D 30 32 30 31 30 32 30 33 30 32 30 46 31 38 0D *(human readable: AT+LEADDATA=02010203020F18)*	Set flags + UUID of battery service

+LEADE	Enable Customized Advertising

AT syntax: **AT+LEADE=<value>**

Supported since firmware version 1.310

This command controls the advertising behavior.

This parameter value cannot be stored using **AT&W**.

With **AT+LEADE=0** only the built-in Terminal I/O service is advertised.

With **AT+LEADE=1** only the customized advertising value is advertised which needs to be configured with **AT+LEADDATA** (and optional **AT+LESCDATA**) first.

With **AT+LEADE=3** the module stops all advertising. With disabled advertising the client (e.g. iPhone) is not able to discover the device or to connect to the device. This should only be done when the service is not in use to save battery power.

With disabled internal Terminal I/O due to **AT+LETIO=0**, the values 0 and 3 show the same behavior. There will be no advertising and no connection.

Value	Description
0	Customized advertising disabled, internal Terminal I/O advertising enabled
1	Customized advertising enabled, internal Terminal I/O advertising disabled
2	Reserved for future use
3	Advertising off, customized advertising disabled, internal Terminal I/O advertising disabled

+LEADINTMAX Maximum Advertising Interval

AT syntax: **AT+LEADINTMAX=<value>**

Supported since firmware version 1.103

This command configures the maximum advertising interval (in milliseconds) for a Bluetooth Low Energy Peripheral.

Value	Description
n=20…10240	Use maximum advertising interval of *n* ms
1280	Use maximum advertising interval of 1280 ms

NOTE:

Make sure that the value of AT+LEADINTMAX is higher or equal the value of AT+LEADINTMIN.

+LEADINTMIN Minimum Advertising Interval

AT syntax: **AT+LEADINTMIN=<value>**

Supported since firmware version 1.103

This command configures the minimum advertising interval (in milliseconds) for a Bluetooth Low Energy Peripheral.

Value	Description
n=20…10240	Use minimum advertising interval of *n* ms
1280	Use minimum advertising interval of 1280 ms

NOTE:

Make sure that the value of AT+LEADINTMAX is higher or equal the value of AT+LEADINTMIN.

+LEADPAR Setup Parameters for Customized Advertising

AT syntax: **AT+LEADPAR=par$_1$=<value$_1$>[, .. [,par$_n$=<value$_n$>]]**

Supported since firmware version 1.310

This command is used to setup parameters for a customized advertising.

This parameter value cannot be stored using AT&W.

par$_n$	**value$_n$**
advtype Optional Coding: decimal.	Type of advertising: **0: undirected (default)** 2: scannable 3: non-connectable

This command is optional, if not submitted these defaults apply:

- advtype = 0 - advertising type “undirected”

NOTE:

The values for the minimum and maximum of the advertising interval may be set with the AT+LEADINTMIN and AT+LEADINTMAX commands.

Example:

AT+LEADPAR=ADVTYPE=0	Set type of advertising “undirected”

+LECONINTMAX Maximum Connection Interval

AT syntax: **AT+LECONINTMAX=<value>**

Supported since firmware version 1.103

This command configures the maximum connection interval for a Bluetooth Low Energy connection. The unit is in 1.25 milliseconds timeslots.

Value	**Description**
n=6…3200	Use maximum connection interval of *n* * 1.25 ms
32	Use maximum connection interval of 40 ms

NOTE:

Make sure that the value of AT+LECONINTMAX is higher or equal the value of AT+LECONINTMIN.

+LECONINTMIN Minimum Connection Interval

AT syntax: **AT+LECONINTMIN=<value>**

Supported since firmware version 1.103

This command configures the minimum connection interval for a Bluetooth Low Energy connection. The unit is in 1.25 milliseconds timeslots.

Value	Description
n=6…3200	Use minimum connection interval of *n* * 1.25 ms
16	Use minimum connection interval of 20 ms

NOTE:

Make sure that the value of AT+LECONINTMAX is higher or equal the value of AT+LECONINTMIN.

+LECONPARAM Connection Parameter Handling

AT syntax: **AT+LECONPARAM=connHnd,[connIntMin],connIntMax,slaveLat[,connTimeout]**
AT+LECONPARAM?connHnd

Supported since firmware version 1.540

The command **AT+LECONPARAM** used with "=" requests new connection parameters to be used for the connection defined by connHnd.

The command **AT+LECONPARAM** used with "?" shows the active connection parameters used by the connection defined by connHnd. A LECONPARAM event is generated containing the active connection parameters.

Parameter	Description
connHnd	Connection handle from CONNECT event
connIntMin	6…3200 minimum connection interval in steps of 1.25 ms (mandatory for peripheral)
connIntMax	6…3200 maximum connection interval in steps of 1.25 ms
slaveLat	0…499 connection intervals
connTimeout	Optional connection supervision timeout in steps of 10 ms. Will be calculated internally if not specified. Has to be calculated according to Bluetooth core spec.

NOTE:

If the device does not support a connection handle in the CONNECT message use 0x00 as fix value for connHnd.

The connection establishment procedure is described in chapter 2.3.

In central role the optional parameter connIntMin is used for negotiation procedure with the peripheral. If the central does not support this feature, it will report an error. In peripheral role the parameter connIntMin is mandatory.

The new connection parameters are signaled by event LECONPARAM.

The result is OK or ERROR.

+LECPEVENT Enable LECONPARAM Event Signaling

AT syntax: **AT+LECPEVENT =<enable>**

Supported since firmware version 1.540

This command enables/disables the automatic signaling of LECONPARAM events.

enable	Description
0	Disable automatic LECONPARAM event signaling
1	Enable automatic LECONPARAM event signaling

The result is OK or ERROR.

+LENAME Request Remote Device Name over BLE

AT syntax: **AT+LENAME=<bdaddr>**

Supported since firmware version 1.310

This command is used to request the remote device name using Bluetooth Low Energy.

In case the remote device is a single mode BLE device, this command is the only possibility to retrieve the remote device name.

NOTE:

To request the remote device name over BLE the parameter AT+LEROLE must be set to 1 (Central).

+LEROLE Bluetooth Low Energy Device Role

AT syntax: **AT+LEROLE=<value>**

Supported since firmware version 1.103

This command configures the Bluetooth Low Energy role of the device.

Value	Description
0	Set device role to “Peripheral”
1	Set device role to “Central”

When set to “Peripheral” (default), the device advertises and accepts incoming BLE connections. Searching for Bluetooth Low Energy devices with **AT+LESCAN** command is not possible.

When set to “Central”, the device is invisible and does not accept incoming BLE connections. The device can search for peripherals using **AT+LESCAN** command and initiate outgoing connections using the **ATD** command.

+LESCAN Search Bluetooth Low Energy Devices

AT syntax: **AT+LESCAN[=uuid]**

Supported since firmware version 1.103

With this command an automatic search for all discoverable Bluetooth Low Energy devices will be initiated. The discovery will last for approx. 10 seconds.

NOTES:

To perform searching for Bluetooth Low Energy devices the parameter AT+LEROLE must be set to 1 (Central).

Any character input while the BlueMod+SR is searching will abort the search procedure.

As a result, a list will be output containing the Bluetooth addresses of the visible devices in range, the advertisement type, the RSSI, the Bluetooth friendly name, the TX level, manufacturer specific data and all UUIDs contained in the advertising and scan response data, if available, of the remote device.

Bluetooth address (including address type), RSSI and advertising type are always provided. All other values like Bluetooth friendly name, TX level, manufacturer specific data and UUID are optional and depends of the advertising data of the discovered device. UUIDs can be 16-bit or 128-bit values.

Value	Description
001122334455,tx	Bluetooth address, address type
RSSI:	RSSI value
TYPE:	Advertisement type
NAME:	Bluetooth friendly name (optional)
TX:	TX level (optional)
MNF:	Manufacturer specific data (optional)
UUID:	UUID(s) (optional)

The optional parameter <uuid> (16-bit value) can be used to perform a search for devices which supports a specific service. If the advertising data of the discovered device matches with the specified UUID, all other UUIDs supported by the device will be output too.

To show devices supporting Terminal I/O V2 only, the UUID 0xFEFB shall be used.

Examples:

AT+LESCAN	008025489E2E,t2 RSSI:-058 TYPE:CONN NAME:BM+SR 9 MNF:8F0009B0011000 UUID:0x53544D544552494F5345525631303030 UUID:0xFEFB 008025540193,t2 RSSI:-057 TYPE:CONN NAME:BM+S 0193 MNF:8F0009B0011000 UUID:0xFEFB 0080255455B9,t2 RSSI:-064 TYPE:NONCONN NAME:BATT 55B9 MNF:8F0030323833372C6D563D32373531 FBD8BC2E03F4,t3 RSSI:-073 TYPE:CONN NAME:BM+S/ADC MNF:8F0001900102000000 UUID:0x1815 OK
AT+LESCAN=0xFEFB	008025489E2E,t2 RSSI:-058 TYPE:CONN NAME:BM+SR 9 MNF:8F0009B0011000 UUID:0x53544D544552494F5345525631303030 UUID:0xFEFB 008025540193,t2 RSSI:-057 TYPE:CONN NAME:BM+S 0193 MNF:8F0009B0011000 UUID:0xFEFB OK

+LESCDATA Setup Scan Response Data for Customized Advertising

AT syntax: **AT+LESCDATA=<value$_1$> .. <value$_n$>**

Supported since firmware version 1.310

This command is used to setup the scan response data for a customized advertising.

Value$_k$ represents an octet in hexadecimal format, k <= 31.

This parameter value cannot be stored using **AT&W**.

The coding of the data is according to the *Bluetooth 4.0 Core Specification / Vol. 3, Part C, Chapter 11 and 18 (Length/Type/Value coding) [3]*.

Example:

41 54 2B 4C 45 53 43 44 41 54 41 3D 30 33 30 32 30 46 31 38 0D *(human readable: AT+LESCDATA=03020F18)*	Set UUID of battery service

+LESLAVELAT Slave Latency

AT syntax: **AT+LESLAVELAT=<value>**

Supported since firmware version 1.103

This command configures the slave latency (in connection intervals) for a Bluetooth Low Energy connection.

Value	Description
n=0…499	Use a slave latency of *n* connection intervals
0	Use no slave latency

+LETIO Enable Terminal I/O Service

AT syntax: **AT+LETIO=<value>**

Supported since firmware version 1.310

This command controls the Terminal I/O service. If set to 0 the Terminal I/O service is disabled. To activate the change, it is necessary to store the settings (**AT&W**) and perform a reset (**AT+RESET**).

Value	Description
0	Terminal I/O service disabled (no advertising, no characteristics)
1	Terminal I/O service enabled, security is required with encryption (no MITM)
2	Terminal I/O service enabled, no security (authentication or encryption) required

3.4. GATT

All commands described in this chapter can only be used in AT mode or in the AT channel in multiplexing mode (**AT+BMUX=1**).

3.4.1. GATT Server

The BlueMod+SR in a GATT server configuration does not support connection orientated result messages RING, CONNECT and NO CARRIER.

+LEATTRIB	Define Attributes for a Service

AT syntax: **AT+LEATTRIB=<type>[,par$_1$=<value$_1$>[, .. ,par$_n$=<value$_n$>]..]]**

Supported since firmware version 1.310

This command is used to define attributes for one or more services in the GATT server. The maximum number of attributes supported is limited to 61 which assures that at least one service with 20 characteristics (without permissions and user description) can be defined. This restriction results from the limited size of the internal attribute definition array. Note that depending on the parameters internally more than one attribute per command may be created, see parameter description below for details.

The GAP and GATT services that each GATT server must expose are built-in services in the BlueMod+SR and thus shall not be defined by the application!

The presence of parameters *par*$_1$,..,*par*$_n$ depends on the value of <*type*>:

type	**par$_1$**	**par$_2$**	**par$_3$**
pserv Mandatory	uuid=<*16/128bit UUID*> Mandatory Coding: hexadecimal.		
char Mandatory	prop=<*properties*> Mandatory Coding: hexadecimal. *properties* may have the bitmask values *Read*, *Write Without Response*, *Write*, *Notify* and *Indicate* set. Note: internally the controller generates an additional Client Characteristic Configuration Descriptor (CCCD) attribute with permissions „readable and writable without authentication or authorization“ if bits *Notify* or *Indicate* are set. Note: type=*char* requires at least two free entries in the internal attribute definition array (one more is needed for subsequent command with type=*charval*)!		
charval Mandatory	uuid=<*16/128bit UUID*> Mandatory Coding: hexadecimal.	perm=<*permissions*> Mandatory Coding: hexadecimal. 16-bit value that decodes the access permissions and authentication requirements.	len=<*length*> Mandatory Coding: decimal. Maximum supported length is 20 bytes.

type	par1	par2	par3
chardcccd Optional	perm=<*permissions*> Mandatory Coding: hexadecimal. This command is needed only if a CCCD shall be generated with permissions other than „readable and writable without authentication or authorization“. See note in *type*=char description.		
chardusrd Optional	usrd=<*user description* > Mandatory Coding: UTF-8 string. User description string. Note: - Since *Extended Properties* (see *type*=char, parameter *properties* description) are not supported the remote peer may not write to this characteristic descriptor. - The number of characters is limited to 13. Internally the character sequence is terminated by a zero byte resulting in an ASCII-Z string.	perm=<*permissions*> Optional Coding: hexadecimal. 16-bit value that decodes the access permissions and authentication requirements. If this parameter is omitted the default *readable without authentication or authorization* applies.	
complete Mandatory	No parameter. Used to signal that all attribute definitions have been sent to the controller.		

The characteristic properties are coded as a hexadecimal bitmask as defined in *Bluetooth Core Spec 4.0 Volume3 Part G Chapter 3.3.1.1 [3]*.

Value	Properties
02	Read
04	Write without response
08	Write
10	Notify
20	Indicate

The values can be combined, for example read & notify result in 12.

The attribute permissions (parameter perm=<*permissions*>) are bit coded in a 16 bit hexadecimal value.

Bit	Value	Function
0,1	0 1 2 3	Read not permitted Read permitted Read with authentication permitted Read with authentication and MITM protection permitted
2..3	-	Reserved
4,5	0 1 2 3	Write not permitted Write permitted Write with authentication permitted Write with authentication and MITM protection permitted
6..11	-	Reserved
12..15	-	Requested encryption keysize - 1

Example: value for read only is 0001.

The complete service/s is/are defined through repeated submissions of the **AT+LEATTRIB** command (see example below).

Due to the GATT definitions in [3] the **AT+LEATTRIB** commands must be submitted in a specific order:

Definition of first service:

AT+LEATTRIB=pserv, …

Definition of first characteristic of first service:

AT+LEATTRIB=char, …

AT+LEATTRIB=charval, …

… optional characteristic descriptors (AT+LEATTRIB=chardxxx) …

Definition of second characteristic of first service:

AT+LEATTRIB=char, …

AT+LEATTRIB=charval, …

… optional characteristic descriptors (AT+LEATTRIB=chardxxx) …

…

Completion of service and characteristics definition:

AT+LEATTRIB=complete

Upon successful execution of the **AT+LEATTRIB** command with *type=charval* the device returns the data channel ID which is associated to the characteristic value.

Format: 0x<*channel ID*>

Data belonging to the characteristic value is exchanged over multiplexer data *channel ID*. The channel ID value is coded in decimal digits.

Example:

The example below shows the Battery Service. Battery Service is a simple service which exposes the battery charging level as single characteristic value.

Command	Response	Description
CC FF 1C 41 54 2B 4C 45 41 54 54 52 49 42 3D 70 73 65 72 76 2C 75 75 69 64 3D 31 38 30 46 0D *(human readable:* *AT+LEATTRIB=pserv,uuid=180F)* CC FF 19 41 54 2B 4C 45 41 54 54 52 49 42 3D 63 68 61 72 2C 70 72 6F 70 3D 31 32 0D *(human readable:* *AT+LEATTRIB=char,prop=12)*	CC FF 06 0D 0A 4F 4B 0D 0A *(OK)* CC FF 06 0D 0A 4F 4B 0D 0A *(OK)*	Declares the properties of the battery level value
CC FF 2E 41 54 2B 4C 45 41 54 54 52 49 42 3D 63 68 61 72 76 61 6C 2C 75 75 69 64 3D 32 41 31 39 2C 70 65 72 6D 3D 30 30 30 31 2C 6C 65 6E 3D 31 0D *(human readable:* *AT+LEATTRIB=charval,uuid=2A19,perm=0001, len=1)*	CC FF 08 0D 0A 30 78 30 32 0D 0A CC FF 06 0D 0A 4F 4B 0D 0A *(0x02 OK)*	Declares the battery level value (one byte in the range 0,..,100). Battery level values are exchanged over multiplexer channel 2
CC FF 15 41 54 2B 4C 45 41 54 54 52 49 42 3D 63 6F 6D 70 6C 65 74 65 0D *(human readable:* *AT+LEATTRIB=complete)*	CC FF 06 0D 0A 4F 4B 0D 0A *(OK)*	Completes the service definition sequence

If a connection exists values written to data multiplexer channel 2 will be sent to the remote device. If no connection exists, the last value written to a multiplexer channel will be stored in RAM. In case a connection from or to a remote device is established, the remote device will receive the last value that the application wrote to channel 2. If the application did not write a value to the data channel yet, the remote device will get an error response with error code >= 0x80 on their read request.

4. APPENDIX

4.1. Linktype

The linktype shows if a link is a classic Bluetooth or a Bluetooth Low Energy link.

Linktype	Meaning
0x00	Classic BR/EDR
0x02	Bluetooth low energy using public address
0x03	Bluetooth low energy using random address
0x0A	Bluetooth low energy using resolved address

4.2. AT Result Codes

Result codes (numerical and verbose):

Numeric	Text	Meaning
0	OK	Command completed
1	CONNECT	Connection established
2	RING	Indicates an incoming call (link request received)
3	NO CARRIER	Connection terminated by normal disconnect or connection setup does not succeeded because the remote Bluetooth device was not found (e.g. wrong address, out of range)
4	ERROR	Illegal command or error that cannot be indicated otherwise
6	NO DIALTONE	Local connection setup error

The connection orientated result messages RING, CONNECT and NO CARRIER are signaled for SPP and Terminal I/O connections only.

4.3. Extended Result Codes

In AT command mode extended result codes (numerical and verbose) are available after activation with ATW1 command.

In multiplexing mode extended result codes are always active.

4.3.1. CONNECT

For the AT result code CONNECT the following extended result codes are available:

Syntax normal mode: CONNECT <<Bdaddr> <linktype>>
Syntax MUX mode: CONNECT <<Bdaddr> <linktype> <channel ID>>

Examples:

Normal mode	Numerical (ATV0)	1 <00802501D11A 0x00>
	Verbose (ATV1)	CONNECT <00802501D11A 0x00>
MUX mode	Numerical (ATV0)	1 <00802501D11A 0x00 0x00>
	Verbose (ATV1)	CONNECT <00802501D11A 0x00 0x00>

4.3.2. RING

For the AT result code RING the following extended result codes are available:

Syntax normal mode: RING <<Bdaddr> <linktype>>
Syntax MUX mode: RING <<Bdaddr> <linktype> <channel ID>>

Examples:

Normal mode	Numerical (ATV0)	2 <00802501D11A 0x00>
	Verbose (ATV1)	RING <00802501D11A 0x00>
MUX mode	Numerical (ATV0)	2 <00802501D11A 0x00 0x00>
	Verbose (ATV1)	RING <00802501D11A 0x00 0x00>

4.3.3. NO CARRIER

For the AT result code NO CARRIER the following extended result codes are available:

Syntax normal mode: NO CARRIER <<Bluetooth release/error code>>
Syntax MUX mode; NO CARRIER <<Bluetooth release/error code> <channel ID>>

The following table shows the release/error codes:

Error code	Meaning
0x0000	Success
0x0001	Accept
0x0002	Reject
0x0003	Resource error
0x0004	Invalid parameter
0x0005	Invalid state
0x0006	Connection disconnect
0x0007	Connection paused
0x0008	Connection lost
0x0009	Authentication failed
0x000A	Flow control violation
0x000B	Init timeout
0x000C	Init out of sync
0x000D	Init hardware failure
0x000E	Lower layer error
0x00FD	Unspecified
0x00FE	Not supported

Examples:

Normal mode	Numerical (ATV0)	3 <0006>
	Verbose (ATV1)	NO CARRIER <0006>
MUX mode	Numerical (ATV0)	3 <0006 0x00>
	Verbose (ATV1)	NO CARRIER <0006 0x00>

4.4. Events

LECONPARAM	Connection Parameters Updated

Syntax: **LECONPARAM:connHnd,connInt,slaveLat,connTimeout**

Supported since firmware version 1.540

With this event the user is informed about a connection parameter update.

Parameter	Description
connHnd	Connection handle from CONNECT event
connInt	Actual connection interval in steps of 1.25 ms
slaveLat	Actual slave latency in connection intervals
connTimeout	Actual connection supervision timeout in steps of 10 ms

NOTE:

If the device does not support a connection handle in the CONNECT message the fix value 0x00 is used for connHnd.

SSPCONF SSP Passkey Confirmation

Syntax: **SSPCONF Bdaddr Passkey ?**

With these event the module requests the confirmation of the passkey displayed on both devices.

Parameter	Description
Bdaddr	Remote Bluetooth address
Passkey	Passkey to be confirmed on both devices

SSPPIN SSP Passkey Display

Syntax: **SSPPIN Bdaddr Passkey**

With this event the module shows the PIN to be entered on the remote device.

Parameter	Description
Bdaddr	Remote Bluetooth address
Passkey	PIN to be entered on remote side

SSPPIN SSP Passkey Request

Syntax: **SSPPIN Bdaddr ?**

With this event the module requests the entry of the PIN displayed on the remote device.

Parameter	Description
Bdaddr	Remote Bluetooth address

5. ACRONYMS AND ABBREVIATIONS

AT	Attention Command
GATT	Generic Attribute Profile
MUX	Multiplexing
SSP	Secure Simple Pairing
UUID	Universal Unique Identifier

6. Document History

Revision	Date	Changes
r01d01	2012-12-04	Initial draft version
r01	2013-01-20	Removed command &v Removed unsupported baudrates of command %B Revised Title of document Changed syntax of +BPIN Added note in +BNDS Added default value of +BNDSIZE and +BSNAME Added examples to +BINQSERV Changed headline of +BSSPCONF and +BSSPPIN Removed not supported commands CMDS, CAPA, BRAD, DBITS, PRTY, SBITS, RSTMSG, RSTTIM, BACCL Revised description of D Removed ATI77 from description of I Revised command syntax of +BND, +BNDDEL, +BNDS, +BNDSIZE, +BOSRV, +BPSM using the "=" to set values Added chapter Escape Sequence Added description of command syntax Added chapter Result Codes Added maximum length of BSNAME Revised maximum value of BNDSIZE Added chapter Hangup
r02d01	2013-03-04	Harmonized syntax of the examples Revised syntax of examples in BINQSERV, BNDSIZE command Revised description of BND command
r02d02	2013-05-02	Added new commands: +IOACFG, +LEADINTMAX, +LEADINTMIN, +LECONINTMAX, +LECONINTMIN, +LEROLE, +LESCAN, +LESLAVELAT Added example in +BINQSERV command S registers Removed unsupported S registers S9 and S91 Corrected default value of S30 register Added new value “4 - Display and keyboard” in BIOCAP command Modified default value and maximum length of +BNAME command Added identifier TIO in D command Added example in +BINQ command Revised syntax of Extended Result Codes Added error codes to Extended Result Codes Revised description of E, I, Q, V and W command

Revision	Date	Changes
r02d03	2013-05-21	Add chapters General, Classic Bluetooth and Bluetooth Low Energy Commands
r02	2013-05-24	Version r02 released
r03	2013-06-20	Revised description of +BNDDEL command
r04	2013-10-18	Added new commands: +BSMODE, +LENAME, +BMUX Added chapter 1.2 Multiplexing Mode (MUX) Added GATT commands: +LEADDATA, +LEADE, +LEADPAR, +LEATTRIB, +LESCDATA, +LETIO Corrected syntax to show current setting of S registers Corrected maximum value of +LEADINTMAX and +LEADINTMIN commands Corrected default value of S30 command Updated description and example in +BNDLIST command Added linktype 0x0A for Bluetooth low energy resolved address Added channel ID in Extended Result Codes for multiplexing mode Corrected description of D command Corrected values of +BSMODE command Added note to +LENAME command Added an overview about events in appendix Corrected values of +LESLAVELAT command Added note to S0 register Added more precisely description of &F command Removed values 1 and 4 of advtype of +LEADPAR
r05d01	2014-01-28	Added new commands: +BARSSI, +BSTPOLL, +BRSSIOUT, +RFMAXTXPWR Added modified behavior of +BMUX command Added new value and new default “Terminal I/O without security” in +LETIO command Added maximum number of attributes in +LEATTRIB command
r05d02	2014-02-24	Added new value “no sniff support” in +BSMODE command Corrected example in +LESCDATA command Clarified limitations of the AT+LEATTRIB command (max. number of attributes, user description ..)
r05d03	2014-05-20	Added “channel ID” when using MUX mode in ATA and ATH command and in CONNECT, RING, NO CARRIER result codes Improved description of ATA command Improved description of NO CARRIER result code Removed NO ANSWER result code Added specific order of +LEATTRIB commands Added description of channel ID assignment Modified syntax of returned data channel ID in +LEATTRIB command type charval

Revision	Date	Changes
r05	2014-05-28	Version r05 released
r06	2014-09-17	Correct characteristic default value description Remove note from +LEATTRIB table Revised description of +LEADPAR +BNAME in LE scan response is truncated to 7 bytes +RESET command has no response message +BND command only applicable for Classic Bluetooth Revised description of “channel ID” in MUX mode Added note for using BIOCAP=4 with Classic Bluetooth
r07	2014-12-23	Corrected page reference in +BNDLIST command Added new commands +BPAIRMODE, +NFCMODE, +IOBCFG Added result code NO DIALTONE Revised description of D and +BSTPOLL command Added I2C address reserved for the NFC tag
r08	2015-02-11	Added firmware dependency for new implemented commands Revised description of +BRSSIOUT command
r09	2015-04-10	Added new command +BFCON Commands +LEADDATA, +LEADE, +LEADPAR and +LESCDATA moved from “GATT Server” chapter to “Low Energy” chapter Extended behavior of +LEADE command Revised description of &F command More precised description of AT Result Codes Increased maximum number of attributes in +LEATTRIB command
r10	2015-06-17	Improve description of BNDSIZE regarding maximum number of bonded devices The leading "+" is missing in the headline of most commands Improve description about the command line in introduction
r11	2015-11-25	Added new commands: +D, +DMODE, +DPARAMS, +DSET, +IOCCFG Corrected extended result codes syntax New default value of ATI/ATI0 command Moved +BARSSI command to chapter 2.1 General Extended the length of Bluetooth classic device name (+BNAME) to 30 characters Improved description of +BMITM command Improved description of HANGUP feature
r12	2016-03-02	Added new commands +LECONPARAM, +LECPEVENT Added new event LECONPARAM Added new chapter “Bluetooth Low Energy Connection Establishment Procedure” Added new value “2” of +IOBCFG command Added responses of list commands
r13	2016-05-26	Telit cover page added

Revision	Date	Changes
r14	2016-12-07	Revised description of +LESCAN command Extended the length of local device name (+BNAME) in BLE mode to 25 characters Removed the truncation of local device name (+BNAME) in BLE advertising Added the role of the remote device in +BNDLIST command Added note to +BINQ and +LESCAN command that any character input aborts the active search process