

BlueMod+S/Central AT Command Reference

80507ST10754A Rev. 04 – 2016-11-16



APPLICABILITY TABLE

PRODUCT
BlueMod+S/Central



*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.

Copyright © Telit Communications S.p.A. 2016.



Contents

1. Introduction.....	7
1.1. Scope.....	7
1.2. Audience	7
1.3. Contact Information, Support	7
1.4. Document Organization	7
1.5. Text Conventions	8
1.6. Related Documents.....	8
2. Features.....	9
2.1. AT Command Mode	9
2.2. Escape Sequence	10
2.3. Multiplexing Mode (MUX)	10
2.4. Connection Establishment Procedure	11
2.5. Hangup.....	11
3. COMMAND SUMMARY	12
3.1. General	12
3.2. Bluetooth Low Energy	27
3.3. GATT	32
3.3.1. GATT Client Specific AT Mode Commands	32
3.3.2. GATT Client Specific MUX Commands.....	39
3.3.3. GATT Client Data Handling on MUX Channel.....	40
4. APPENDIX	41
4.1. Data Formats	41
4.1.1. Data Array	41
4.1.2. Bluetooth Address	41
4.1.3. UUID.....	41
4.1.4. Values.....	41
4.1.5. Bit Arrays	41
4.2. Linktype.....	42
4.3. AT Result Codes	42
4.4. Release/Error Codes.....	44
4.5. Events	45



4.6. MSCs	47
5. ACRONYMS AND ABBREVIATIONS	58
6. Document History.....	59



1. Introduction

1.1. Scope

This document specifies the command interface for the BlueMod+S/Central firmware.

1.2. Audience

Readers of this document should be familiar with the BlueMod+S 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+S Hardware User Guide, 1VV0301275
- [2] BlueMod+S Software User Guide, 1VV0301277
- [3] Bluetooth 4.0 Core Specification
- [4] BlueMod+S - BlueMod+SR Delta Reference



2. Features

The BlueMod+S/Central (“BlueMod+S”) 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. 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.

2.3. 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 data channel of a BLE GATT characteristic or the TIO data channel or the AT command interface (value FF).

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.

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



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

Result messages like RING, CONNECT and NO CARRIER may be sent in multiple frames by the BlueMod+S module. The host controller application needs to collect the data until the closing “<CR><LF>” is received.

2.4. Connection Establishment Procedure

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 spec..

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 spec. 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 data connections can be disconnected at once by setting the HANGUP pin (GPIO4) to high level.



3. COMMAND SUMMARY

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

- General 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 the following baud rates.



Value	Description
1200	1,200 bps
2400	2,400 bps
4800	4,800 bps
9600	9,600 bps
14400	14,400 bps
19200	19,200 bps
28800	28,800 bps
38400	38,400 bps
57600	57,600 bps
76800	76,800 bps
115200	115,200 bps
230400	230,400 bps
250000	250,000 bps
460800	460,800 bps
921600	921,600 bps
1000000	1,000,000 bps

To set a baud rate write the desired value in the command.

Examples:

AT%B9	Set baud rate to 230,400 bps
AT%B250000	Set baud rate to 250,000 bps



NOTE:

Information regarding the deviation of the real baud rate to the value set can be found in the *BlueMod+S 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 bandlist 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 is supported for compatibility reasons only. A Bluetooth low energy device has to answer every connection automatically. See also register S0 description.

Examples:

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

+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



+BMITM
SSP Man in the Middle Protection

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

NOTE:

This command is deprecated and only supported for backward compatibility reasons. Please use connection based (see ATD command) and service based (see LETIO parameter) security configuration instead.

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	parameter disabled, connection and service based configuration applies (see ATD command and LETIO parameter)
1	Man in the middle protection enabled, connection and service based configuration is ignored

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.

For possible combinations of I/O capabilities and the possibility of MITM protection/authentication level see command BIOCAP and the *BlueMod+S Software User Guide* [2].

+BMUX
Activate Multiplexing Mode

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

This command is used to activate the multiplexing mode protocol.

Value	Description
0	Normal AT mode
1	Non persistent multiplexing mode
2	Persistent multiplexing mode

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

The multiplexing mode 2 is stored persistent. To disable the multiplexing mode the command AT+BMUX=0 followed by AT&W must be entered. This mode is intended to be used with the AT+SYSTEMOFF command.

In multiplexing mode extended result codes are always active (see chapter 4.3).

+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. It is limited to 19 characters.

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

In BLE advertising the name is truncated to the first 10 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+S %4a	Display on remote end: BM+S 89AB
AT+BNAME=BM+S %4a	Display on remote end: BM+S 89AB
AT+BNAME=BM+S %:3a	Display on remote end: BM+S 9:AB
AT+BNAME=BM+S %3a	Display on remote end: BM+S 9AB
AT+BNAME=BM+S %:12a	Display on remote end: BM+S 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+S.

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




NOTE:

This command can only be executed while not connected.

+BNDLIST
Show Bonded Device List

AT syntax: **AT+BNDLIST**

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

Each entry in the **bonded-device list** contains the Bluetooth address, the linktype (see chapter 4.2) and the role of the remote device (“C” for client or “P” for peripheral).

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

Example:

AT+BNDLIST	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+S


NOTE:

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

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...4) the bonded-device list can hold.



The BlueMod+S can store up to 4 devices. The default size is 4.



NOTE:

Modification of this parameter will delete all devices in the bonded-device list.
Two entries of the same 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>**

This command controls the pairable mode of the BlueMod+S peripheral part.

When set to "0" the module is only connectable for clients stored in the local bondlist. New pairing requests will be rejected.

Mode	Description
0	No pairing allowed, BlueMod+S advertises TIO as "functional"
1	Pairing allowed, BlueMod+S 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 or 3.



+BSSPPIN	SSP Passkey Response
-----------------	-----------------------------

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

If an authentication is initiated, 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>,<address type> ?**

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

Example:

SSPPIN 00802507C08D,t2 ? AT+BSSPPIN 00802507C08D,t2,314546 OK RING CONNECT	Receive SSP passkey request Send SSP passkey response
--	--

D	Initiate Bluetooth Low Energy Link
----------	---

AT syntax: **ATD<brad>[,tx],GATT[,reusebond]**
 ATD<brad>[,tx],TIO[,reusebond]

**) Identifier TIO and parameter reusebond supported since firmware version 3.004*

This command addresses a Bluetooth device directly via its address.

Param.	Description
brad	Called Bluetooth remote device address (12 hex digits)
tx	x is the remote Bluetooth address type (see chapter Bluetooth Address) If not specified a public address is assumed.
reusebond	Optional parameter that automatically starts the encryption immediately after the GATT connection is established. This is possible only if bonding information is available for the remote device. If no such bonding information is available this parameter has no effect. Note: It is not possible to read out the information of an encrypted connection.

If the device is configured to “Central” role and initiates a connection to a peripheral device, it shall use the identifier GATT or TIO.

A GATT connection allows to use the GATT Client specific command (see chapter 3.3).

A TIO connection allows to transfer data transparently for the UART to the peer device.

If no identifier is given in the dial string, no connection attempt will be initiated. The command reports ERROR.

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

Dialing procedure ends after a timeout specified by S register S7.



Dialing procedures which cause a security procedure have additional timeouts depending on the requested security procedure (see parameter BIOCAP). The connection timeout specified in S7 is not valid after a security procedure is started.

Except for the <reusebond> parameter functionality described above all security procedures are expected to be requested by the remote device while connection setup or an active connection. In case the remote devices requests a security procedure the BlueMod+S automatically performs all necessary procedures to satisfy the security needs of the remote device only limited by the remote and local I/O capabilities (AT+BIOCAP).

+DFUMODE	Device Firmware Update Mode
-----------------	------------------------------------

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

This command sets the device firmware update mode.

To activate the mode it is necessary to store the settings and perform a reset or use the command AT+DFUSTART.

Value	Description
1	Device firmware update over serial interface
2	Device firmware update over the air (OTA)

+DFUNAME	Over The Air Update Name
-----------------	---------------------------------

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

This command sets the device name for the over the air firmware update mode.

The name is limited to 8 characters.

To activate the mode it is necessary to store the settings and perform a reset or use the command AT+DFUSTART.

Name	Description
BM+S_DFU	Device firmware update name is “BM+S_DFU”

+DFUSTART	Start Bootloader
------------------	-------------------------

AT syntax: **AT+DFUSTART**

This command sets the device into the configured firmware update mode. The command times out after 2 minutes.



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 (): **ATH <connHnd>**

This command disconnects the existing Bluetooth connection addressed by connHnd from the corresponding CONNECT event.

Examples:

ATH 0x10	Disconnects connection with connHnd 0x10
ATH 0x01	Disconnects connection with connHnd 0x01
ATH	Disconnects all existing connections

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+S %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 Telit"
7	Returns "OK"
8	Returns "ERROR"
9	Returns "OK"
77	Returns bootloader version
99	Returns the firmware creation date



+IOACFG	Config of Pin IOA
----------------	--------------------------

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

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>**

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 is active Output: High level if no lower layer connection is active

*) Supported since firmware version 2.106

+LOAD	Load Stored Parameter Setting
--------------	--------------------------------------

AT syntax: **AT+LOAD**

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

O	Return to Online State
----------	-------------------------------

AT syntax: **ATO**

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



+PNPPID	PnP Product ID
----------------	-----------------------

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

**) Supported since firmware version 3.004*

This command sets the product ID provided in the device information service (DIS). The format is a 16 bit hex value. The default value is 0xB010 (Telit product ID for BlueMod+S/Central firmware).

To activate a new value it is necessary to store the settings (**AT&W**) and perform a reset (**AT+RESET**).

+PNPPVER	PnP Product Version
-----------------	----------------------------

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

**) Supported since firmware version 3.004*

This command sets the product version provided in the device information service (DIS). The format is a 16 bit hex value. The default value is the version number of the particular Telit BlueMod+S/Central firmware, e.g. 0x3001 for firmware version 3.001.

To activate a new value it is necessary to store the settings (**AT&W**) and perform a reset (**AT+RESET**).



NOTE:

After setting the vendor ID (**AT+PNPVID**) to a different value than the default 0x008F the user has to set his own product version (otherwise the value 0x0200 will be used).

+PNPVID	PnP Vendor ID
----------------	----------------------

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

**) Supported since firmware version 3.004*

This command sets the vendor ID provided in the device information service (DIS). The format is a 16 bit hex value. The default value is 0x008F (Telit vendor ID).

To activate a new value it is necessary to store the settings (**AT&W**) and perform a reset (**AT+RESET**).



+PNPVSRC	PnP Vendor ID Source
-----------------	-----------------------------

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

**) Supported since firmware version 3.004*

This command sets the vendor ID source provided in the device information service (DIS).

Value	Description
1	Bluetooth SIG assigned company ID
2	USB assigned company ID

To activate a new value it is necessary to store the settings (**AT+W**) and perform a reset (**AT+RESET**).

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+S by a forced hardware reset (like power off/on).

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

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

This command sets the maximum output power of the Bluetooth radio of the device. A changed value becomes active immediately.



Value	Description
-128	Use factory default maximum output power of 4 dBm
4	4 dBm
0	0 dBm
-4	-4 dBm
-8	-8 dBm
-12	-12 dBm
-16	-16 dBm
-20	-20 dBm
-30	-30 dBm

All other values in the range of -127 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	1	Immediate call acceptance <i>Note: Setting of S0 only allow value 1 for BLE. In case of BLE connections always one RING is signalised and automatic call acceptance is set.</i>
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 connection (in s). 0 means no timeout.
S12	100	Delay time by using the escape sequence in 10 ms increments



+SYSTEMOFF	Enter System Off Mode
-------------------	------------------------------

AT syntax: **AT+SYSTEMOFF**

This command sets the module into low power mode during the time the module is not used. The module will restart on GPIO activity. The host can use the IOA pin to monitor the system status.

Possible use cases and an usage example are described in the *BlueMod+S Software User Guide* [2].

+UICP	Set UART Interface Control Protocol
--------------	--

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

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

To activate UICP, it is necessary to store the settings and perform a reset.

Mode	Description
0	UICP off
1	UICP on

The valid sequence to e.g. activate UICP is:

AT+UICP=1

AT&W

AT+RESET

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)

+LEFIXPIN	Fix PIN for Pairing Procedure
-----------	-------------------------------

Param.	Description
fixpin	6 digits pin value e.g. "000000" (digit 0..9 only) (default is 0 length pin "")

+LEADINTMAX	Maximum Advertising Interval
-------------	------------------------------

Value	Description
$n=20\dots10240$	Use maximum advertising interval of n ms
1280	Use maximum advertising interval of 1280 ms

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

+LEADINTMIN	Minimum Advertising Interval
-------------	------------------------------

+LECONINTMAX

Maximum Connection Interval

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

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

Value	Description
$n=6\dots3200$	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>**

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

Value	Description
$n=6\dots3200$	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

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.



Param.	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.

The connection establishment procedure is described in chapter 2.4.

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 signalled by event **LECONPARAM**.

The result is OK or ERROR.

+LECPEVENT	Enable LECONPARAM Event Signaling
-------------------	--

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

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.

+LEROLE	Bluetooth Low Energy Device Role
----------------	---

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

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”
2	Set device role to combined “Peripheral” and “Central”

When set to “Peripheral”, 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.



When set to the combined “Peripheral” and “Central” role (default) all above features are supported and have to be handled. If you only need one single role, please use parameter 0 or 1.

+LESLAVELAT	Slave Latency
--------------------	----------------------

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

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

Value	Description
$n=0\dots499$	Use a slave latency of n connection intervals
0	Use no slave latency

+LETIO	Enable Terminal I/O Service
---------------	------------------------------------

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

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 *)	Terminal I/O service enabled, authenticated pairing with encryption (MITM required)

*) Supported since firmware version 3.004

The valid sequence to change the setting is:

AT+LETIO=1

AT&W

AT+RESET

+LEADDATA	Setup Advertise Data for Customized Advertising
------------------	--

AT syntax: **AT+LEADDATA=<value₁> .. <value_n>**

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

$Value_k$ represents an octet in hexadecimal format, $k \leq 31$.



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:

41 54 2B 4C 45 41 44 44 41 54 41 3D 30 32 30 31 30 36 30 33 30 32 30 46 31 38 0D <i>(human readable: AT+LEADDATA=02010603020F18)</i>	Set flags + UUID of battery service
--	-------------------------------------

+LEADE	Enable Customized Advertising
---------------	--------------------------------------

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

This command controls the advertising behavior.

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

With **AT+LEADE=1** only the customized advertising value is advertised.

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 TIO 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 TIO advertising enabled
1	Customized advertising enabled, internal TIO advertising disabled
2	Reserved for future use
3	Advertising off, customized advertising disabled, internal TIO advertising disabled

+LEADPAR	Setup Parameters for Customized Advertising
-----------------	--

AT syntax: **AT+LEADPAR=par₁=<value₁>[, .. [,par_n=<value_n>]]**

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

par _n	value _n
advtype	Type of advertising: 0: undirected (default)
Optional	2: scannable
Coding: decimal.	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”

+LESCDATA

Setup Scan Response Data for Customized Advertising

AT syntax: **AT+LESCDATA=<value₁> .. <value_n>**

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

Value_k represents an octet in hexadecimal format, $k \leq 31$.

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:

41 54 2B 4C 45 53 43 44 41 54 41 3D 30 33 30 32 30
46 31 38 0D

Set UUID of battery service

(human readable:
AT+LESCDATA=03020F18)

3.3. GATT

3.3.1. GATT Client Specific AT Mode Commands

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

+LEBUUIDSET

Set 128 bit Base UUID

AT syntax: **AT+LEBUUIDSET=ux**

This command configures base UUIDs needed by the stack to identify 128 bit UUIDs correct.

128 bit UUIDs are module internally treated as 16 bit UUIDs with a defined base UUID. The Telit defined base UUID for Terminal IO V2 is 0000xxxx000010008000008025000000 with xxxx as the variable 16 bit UUID part.



To set a base UUID the 16 bit UUID part could have any legal 16 bit value. It is internally ignored for this command. If the internal base UUID table is full the command reports ERROR, otherwise it reports OK.

The UUID list shall be saved permanent with AT&W.

Param.	Description
ux	With x= 128 bit base UUID

+LEBUUIDDEL	Delete Base UUID
--------------------	-------------------------

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

This command deletes the base UUID information stored by the BlueMod+S in RAM.

To delete the UUID permanently from the flash, it is required to save the reduced list by AT&W, wait for OK and perform an AT+RESET command.

Value	Description
ux	With x= 128 bit base UUID
*	Delete all base UUIDs from the base UUID table.

+LEBUUIDLIST	Show Base UUID List
---------------------	----------------------------

AT syntax: **AT+LEBUUIDLIST**

This command shows information about the configured 128 bit base UUIDs within the BlueMod+S.

The variable 16 bit UUID part is marked with the character 'x'.

Example:

AT+LEBUUIDLIST	0000xxxx000010008000008025000000 0000xxxx111100002222008033330000 OK
----------------	--

+LESCAN	Search Bluetooth Low Energy Devices
----------------	--

AT syntax: **AT+LESCAN**
 AT+LESCAN=GATT[,<rssix>][,RAW]
 AT+LESCAN=<ux>[,<rssix>][,RAW]
 AT+LESCAN=<brad>[,<tx>][,<rssix>][,RAW]

With this command an automatic search for all discoverable Bluetooth Low Energy devices will be initiated. The discovery will last for a time defined by command **AT+LESCANDURATION**.



BlueMod+S/Central AT Command Reference

80507ST10754A Rev. 04 – 2016-11-16

Parameter	Description
rssix	Filter for devices with RSSI value stronger x
brad	Filter for Bluetooth remote device address (12 hex digits)
tx	x is the remote Bluetooth address type see chapter LinkType If not specified a public address is assumed
ux	With x=UUID of a service (4 or 32 hex digits)
GATT	Show all found devices
RAW	Hexdump of advertising and scan response data without duplicate filtering

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

The resulting list depends on the used command parameters.

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 following advertisement types are decoded:

TYPE	Description
CONN	Connectable undirected advertising
CONNDIR	Connectable directed advertising
SCAN	Scannable undirected advertising
NONCONN	Non-connectable undirected advertising
SCANRSP	The scan response will be generated only for devices with the advertisement types CONN and SCAN

Please note that more AD types could be decoded in future releases.

The output is filtered to show each unique advertising packet only once. If the internal filter table is full, all new advertising packets are shown unfiltered. The output in RAW mode is also unfiltered.

The optional parameter <rss> (8-bit signed value) can be used to perform a search only for devices with a RSSI value higher than the provided value. E.g. AT+LESCAN=rssi-50 will



show all devices with a RSSI value higher than -50dBm. This means -45dBm devices are shown, -55dBm devices are filtered.

The optional parameter <brad> (12 hex digits) can be used to perform a search for a device with the specified Bluetooth address. The optional parameter <tx> specifies the type of Bluetooth low energy address.

The optional parameter <ux> (16-bit or 128-bit UUID value) can be used to perform a search for devices which advertises a specific service.

To show devices supporting Terminal I/O only, the UUID FEFB shall be used (**AT+LESCAN=uFEFB**).

If the “RAW” parameter is given, the output will not contain decoded AD type data. Instead it will contain the Bluetooth address of the visible device in range, the RSSI, the advertisement type and the complete advertise or scan response data from the remote device. The data is displayed as an ASCII coded byte steam in hexadecimal values.

There will be no duplicate filtering for advertising packets. All received packets (advertise or scan response) will be printed as soon they are received. If UUID filtering is on (AT+LESCAN=ux,RAW), the output will be printed after receiving the complete advertising data. In active scan mode these are the advertising and scan response packets.

Example:

AT+LESCAN	008025001241,t2 RSSI:-65 TYPE:CONN NAME:BM+S 1241 MNF:8F0009B0011000 UUID:FEFB OK
AT+LESCAN=GATT,RAW	008025001241,t2 RSSI:-65 TYPE:CONN DATA:02010608FF8F0009B0011000 008025001241,t2 RSSI:-65 TYPE:SCANRSP DATA:0A09424D2B5320313234310302FBFE OK

+LESCANDURATION

Duration for +LESCAN

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

This command configures how long the BlueMod+S is searching for discoverable Bluetooth Low Energy devices when the command AT+LESCAN is used.

Value	Description
0	Sets duration time to infinite.
1..300	Sets duration time between 1 seconds and 300 seconds (default=10)



+LESRVD

Service Discovery

AT syntax: **AT+LESRVD=connHnd[,ux]**

With this command an automatic search for services on the given connection handle will be initiated. A connection is required before using this command.

Param.	Description
connHnd	Connection handle from CONNECT event
ux	UUID of a service (4 or 32 hex digits)

The resulting list depends on the used command parameters.

AT+LESRVD=connHnd discovers all services.

Response: List of found services UUIDs.

AT+LESRVD=connHnd,ux discovers all characteristics for the given service UUID.

Response: Requested service UUID and a list of found characteristics with value handle, properties and UUIDs.

The first 4 bytes represent the hex coded **charHnd** value for this characteristic which has to be used with the other AT commands like **AT+LEREAD**.

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

PROP	Properties
0x02	Read
0x04	Write without response
0x08	Write
0x10	Notify
0x20	Indicate

A characteristic with properties **READ** and **WRITE** reports “PROP:0x0A”.

The result parameter **UUID** shows the 16 bit or 128 bit hexadecimal UUID value of the found characteristic.



Example: Discover all services

AT+LESRVD=connHnd	UUID:1800 UUID:1801 UUID:180A UUID:FEFB OK
-------------------	--

Example: Discover TIO service

AT+LESRVD=connHnd,uFEFB	UUID:FEFB 0x0015 PROP:0x04 UUID:00000001000010008000008025000000 0x0016 PROP:0x10 UUID:00000002000010008000008025000000 0x0018 PROP:0x08 UUID:00000003000010008000008025000000 0x0019 PROP:0x20 UUID:00000004000010008000008025000000 OK
-------------------------	---

+LEREAD	Read Characteristic
----------------	----------------------------

AT syntax: **AT+LEREAD=connHnd,charHnd**

With this command a read access to the characteristic defined by connHnd and charHnd is initiated.

Param.	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD

The read data is displayed as ASCII coded byte stream in hexadecimal values with a leading event identifier “LEREAD:”.

AT+LEREAD=connHnd,charHnd

LEREAD:connHnd,charHnd,<hexData> e.g. LEREAD:0x10,0x0016,017AFF for three byte value

OK

+LEWRITE	Write Characteristic
-----------------	-----------------------------

AT syntax: **AT+LEWRITE=connHnd,charHnd,<hexData>**

With this command a write with response access to the characteristic defined by connHnd and charHnd is initiated.



Param.	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD
hexData	ASCII coded byte stream as hexadecimal values e.g. 017aFF for a three byte value

After receiving the response from the GATT server, depending on the result code the result is OK or ERROR. Also unknown values for connHnd and charHnd will lead to an ERROR result.

The host application has to provide the correct number of data bytes for the addressed characteristic. This length information could be found in the profile/service specification for the addressed service.

+LEWRITECMD	Write without Response Characteristic
--------------------	--

AT syntax: **AT+LEWRITECMD=connHnd,charHnd,<hexData>**

With this command a write without response (write command) access to the characteristic defined by connHnd and charHnd is initiated.

Param.	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD
hexData	ASCII coded byte stream as hexadecimal values e.g. 017aFF for a three byte value

There is no result from the server side available. The result is OK if the write without response was sent to the server side. An ERROR is reported if sending was not possible. The result code does not contain information about the reception on the GATT server side.

+LECCCD	Enable/Disable Indications/Notifications
----------------	---

AT syntax: **AT+LECCCD=connHnd,charHnd,enable**

With this command the notifications and indications for the characteristic defined by connHnd and charHnd are enabled or disabled.

Param.	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD
enable	0: disable 1: enable notifications 2: enable indications

The result is OK or ERROR.



3.3.2. GATT Client Specific MUX Commands

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

+LEADDCHAN	Add MUX Channel to Connection
-------------------	--------------------------------------

AT syntax: **AT+LEADDCHAN=connHnd,charHnd,writeType**

This command adds a MUX channel for characteristic defined by charHnd to the connection defined by connHnd. All data transfer for this connection is now done using the MUX channel. A read is performed by issuing an AT+LEREADCHAN command on the AT channel. The data will be transferred using the corresponding MUX channel. All error conditions are signalled on the AT channel.

All data received within an indication or notification message is signalled on the corresponding MUX channel in a single MUX frame.

All data send to the MUX channel in a single MUX frame is send out as a write request. The type of request used by this channel is defined by the parameter “writeType”.

You can define only one channel per connection for a charHnd.

The commands AT+LEREAD, AT+LEWRITE and AT+LEWRITECMD respond ERROR when used with a charHnd with an active MUX channel.

Param.	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD
writeType	“RSP” : Write with response “CMD” : Write without response

In case of successful MUX channel creation a one byte hexadecimal coded MUX channel Id is signaled on a single line followed by the result code OK.

Format: <CR><LF><channel><CR><LF>
 <CR><LF>OK<CR><LF>

E.g. <CR><LF>0x17<CR><LF>
 <CR><LF>OK<CR><LF>

In case of an error condition the result code ERROR is signaled.

For more information on the MUX format see chapter 2.3.

+LEREADCHAN	Read MUX Channel
--------------------	-------------------------

AT syntax: **AT+LEREADCHAN=channel,[len]**

With this command a read access to the characteristic addressed by a MUX channel is initiated.



Param.	Description
channel	MUX channel Id from AT+LEADDCHAN
len	Length of characteristic data to be read (optional for characteristics <= 20 byte)

When using parameter len, the Host application has to provide the correct number of data bytes for the addressed characteristic. This length information could be found in the profile/service specification for the addressed service.

The read data is displayed in the corresponding MUX channel. The command responds OK or ERROR in the AT command channel. For more information on the MUX format see chapter 2.3.

AT+LEREADCHAN=23

OK -> in AT channel

Data will be received in MUX channel 23.

3.3.3. GATT Client Data Handling on MUX Channel

MUX channels are created during GATT server definition using the AT+LEREADCHAN command. MUX channels are valid during the connection.

All data for defined characteristics is sent in one MUX frame on the corresponding channel so a MUX frame has to have the defined data length for the characteristic addressed. If a characteristic has a length of 4, 4 bytes of data has to be sent. The only exception is a characteristic with a variable length. All data sizes between 1 and 20 are supported.

3.3.3.1. Error Handling

In MUX mode all recognized errors are signalled on the AT command channel using the event LEERROR.



4. APPENDIX

4.1. Data Formats

4.1.1. Data Array

Data arrays are encoded as a hexadecimal ASCII based byte stream.

E.g. a Byte array containing the four bytes 0x11, 0x22, 0x33 and 0x44 is encoded as:
11223344

4.1.2. Bluetooth Address

The BlueMod+S supports public and random Bluetooth addresses. The differentiation between the address types is done using the parameter “t2” for public addresses and “t3” for random addresses.

A Bluetooth address value itself is a special byte array variant. There are two valid representations.

The Bluetooth addresses “**008025540203**” and **00:80:25:54:02:03** are equivalent.

e.g. public address: 00:80:25:54:02:03,t2 or 008025540203,t2
random address: F1:B9:EB:41:D8:1E,t3 or F1B9EB41D81E,t3

4.1.3. UUID

UUIDs are special byte array variants.

16 Bit UUIDs are encoded with four hexadecimal digits. E.g. UUID 0xFEFB is encoded as FEFB.

128 Bit UUIDs are encoded with 32 hexadecimal digits. E.g. 00000002000010008000008025000000.

The format 00000002-0000-1000-8000-008025000000 known from the profile specifications is also supported for 128 bit UUIDs.

4.1.4. Values

A parameter value could be encoded hexadecimal or signed decimal. The value range depends on the command specification.

Hexadecimal values shall be encoded with a leading “0x” e.g. 0x01FF.

Positive decimal values shall be encoded without a leading character e.g. 512.

Negative decimal values shall be encoded with a leading “-” e.g. -69.

4.1.5. Bit Arrays

Bit arrays shall be coded as hexadecimal values with a leading “0x”.



For an example see parameter PROP in command AT+LESCAN.

4.2. Linktype

Linktype	Meaning
0x02	Bluetooth low energy using public address
0x03	Bluetooth low energy using random address

4.3. AT Result Codes

Result codes (numerical and verbose):

Numeric	Text	Meaning	Extended Result Codes
0	OK	Command completed	No
2	RING	Indicates an incoming call	Yes
3	NO CARRIER	Connection disconnected	Yes
4	ERROR	Illegal command or error	No
9	CONNECT GATT	GATT connection established	Yes
10	CONNECT TIO	TIO connection established	Yes

Extended result codes (numerical and verbose) are available after activation with **ATW1** command.

In multiplexing mode extended result codes are always active.

OK	Command Completed
----	-------------------

Syntax: OK

Command completed successfully.

CONNECT	Connection Established
---------	------------------------

Syntax: CONNECT connType connHnd [<bdaddr linktype>]

With this result code the user is informed about the establishment of a connection. The connHnd has to be used for characteristic access for this device.

Param.	Description
connType	Type of connection GATT, TIO
connHnd	Connection handle or TIO MUX channel ID
Bdaddr	Remote Bluetooth address, only as extended result code
Linktype	Remote Bluetooth address type see chapter Linktype, only as extended result code

The parameter connType has different meanings depending on operation mode.



AT mode: connType represents the connection handle used for ATH and GATT client access commands like AT+LEREAD, etc..

MUX mode: Additionally to AT mode description, the value of connHnd represents the channel ID of the automatically established data MUX channel for connections with connType=TIO. Data MUX channels for connections with connType=GATT have to be established using the AT+LEADDCHAN command.

NO CARRIER

Connection Disconnected

Syntax: NO CARRIER [connHnd] [<error code>]

With this result code the user is informed about the disconnection of a connection.

Param.	Description
connHnd	Connection handle from CONNECT event
error code	Bluetooth release code



NOTE:

In general the NO CARRIER result code includes a connection handle. Only in the case a connection attempt using the ATD command is cancelled before the connection gets established a NO CARRIER result code without connection handle occurs.

RING

Link Request Received

Syntax: RING [<bdaddr linktype>]

With this result code the user is informed about an incoming connection request.

Param.	Description
Bdaddr	Remote Bluetooth address, only as extended result code
Linktype	Remote Bluetooth address type see chapter Linktype, only as extended result code

ERROR

Illegal Command or Error

Syntax: ERROR

With this result code the user is informed about an error condition. This could be e.g.:

- an illegal command syntax
- an illegal state for the command
- an error that cannot be indicated otherwise



4.4. Release/Error Codes

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 0x10 <0006>
	Verbose (ATV1)	NO CARRIER 0x10 <0006>
MUX mode	Numerical (ATV0)	3 0x10 <0006 0x01>
	Verbose (ATV1)	NO CARRIER 0x10 <0006 0x01>



4.5. Events

SSPPIN	SSP Passkey Request
---------------	----------------------------

Syntax: **SSPPIN Bdaddr,tx ?**

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

Parameter	Description
Bdaddr	Remote Bluetooth address
tx	x is the remote Bluetooth address type (see chapter Bluetooth Address)

SSPPIN	SSP Passkey Display
---------------	----------------------------

Syntax: **SSPPIN Bdaddr,tx Passkey**

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

Parameter	Description
Bdaddr	Remote Bluetooth address
tx	x is the remote Bluetooth address type (see chapter Bluetooth Address)
Passkey	PIN to be entered on remote side

LEIND	Indication Received
--------------	----------------------------

Syntax: **LEIND:connHnd,charHnd,<hexData>**

With this event data received over the air with an indication is displayed to the user. To receive these type of event please enable indications with command AT+LECCCD if allowed for the characteristic.

Parameter	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD
hexData	ASCII coded byte stream as hexadecimal values e.g. 017aFF for a three byte value

LENOTI	Notification Received
---------------	------------------------------

Syntax: **LENOTI:connHnd,charHnd,<hexData>**

With this event data received over the air with a notification is displayed to the user. To receive these type of event please enable notifications with command AT+LECCCD if allowed for the characteristic.



Parameter	Description
connHnd	Connection handle from CONNECT event
charHnd	Characteristic handle from AT+LESRVD
hexData	ASCII coded byte stream as hexadecimal values e.g. 017aFF for a three byte value

LEERROR	Error Condition Occurred
----------------	---------------------------------

Syntax: **LEERROR:type, [parameter]**

With this event the user is informed about error conditions. Depending on error type different parameters are provided.

Type	Parameter	Mode	Description
LEWRITE	connHnd, charHnd, cause	MUX	Write with response
LEREAD	connHnd, charHnd, cause	MUX	Read
LEMUX	connHnd, charHnd, cause	MUX	Generic error for several situations
LECHANN	channel, cause	MUX	Wrong channel number in command
LEAUTH	connHnd, charHnd, cause	MUX	Authentication error

LECONPARAM	Connection Parameters Updated
-------------------	--------------------------------------

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

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



4.6. MSCs

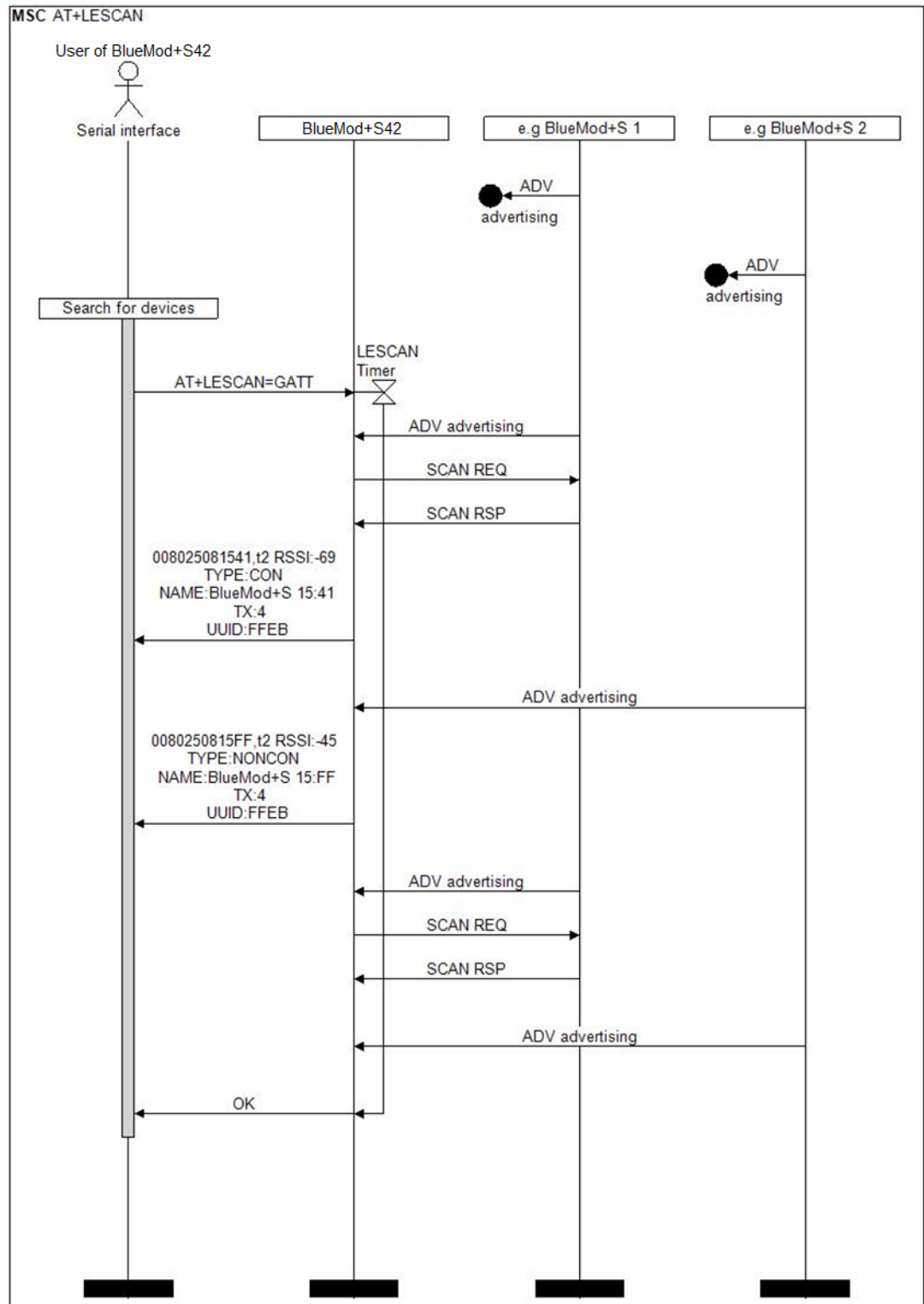


Figure 1: Searching for devices



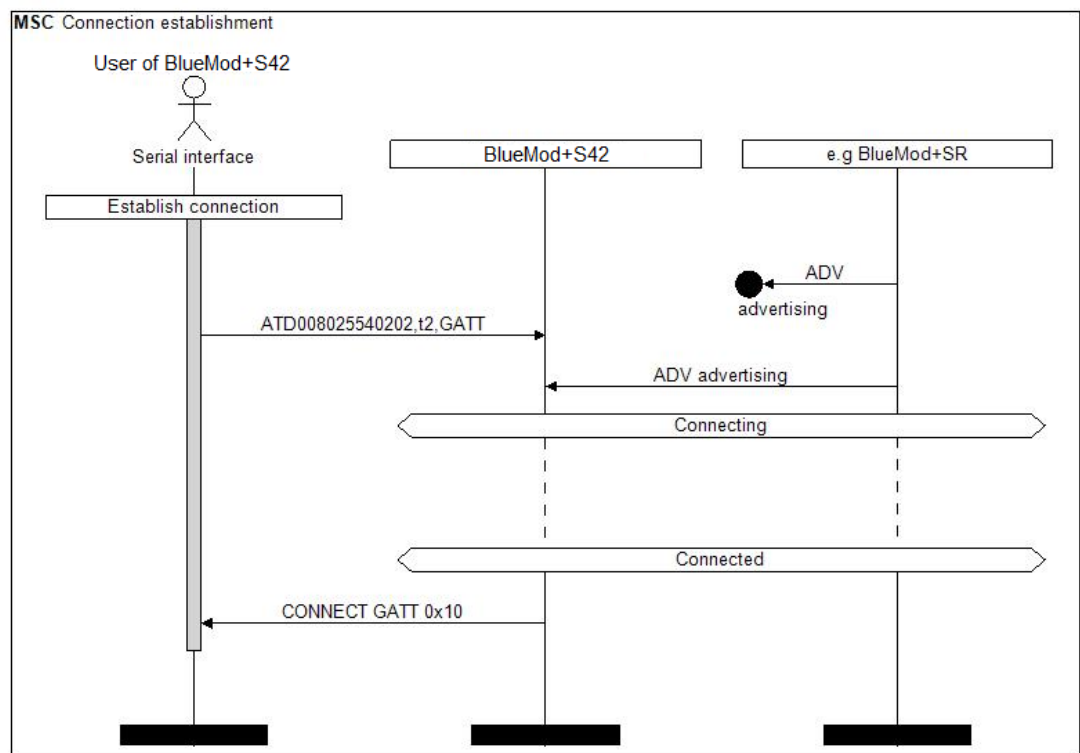


Figure 2: Connection establishment with public type Bluetooth address

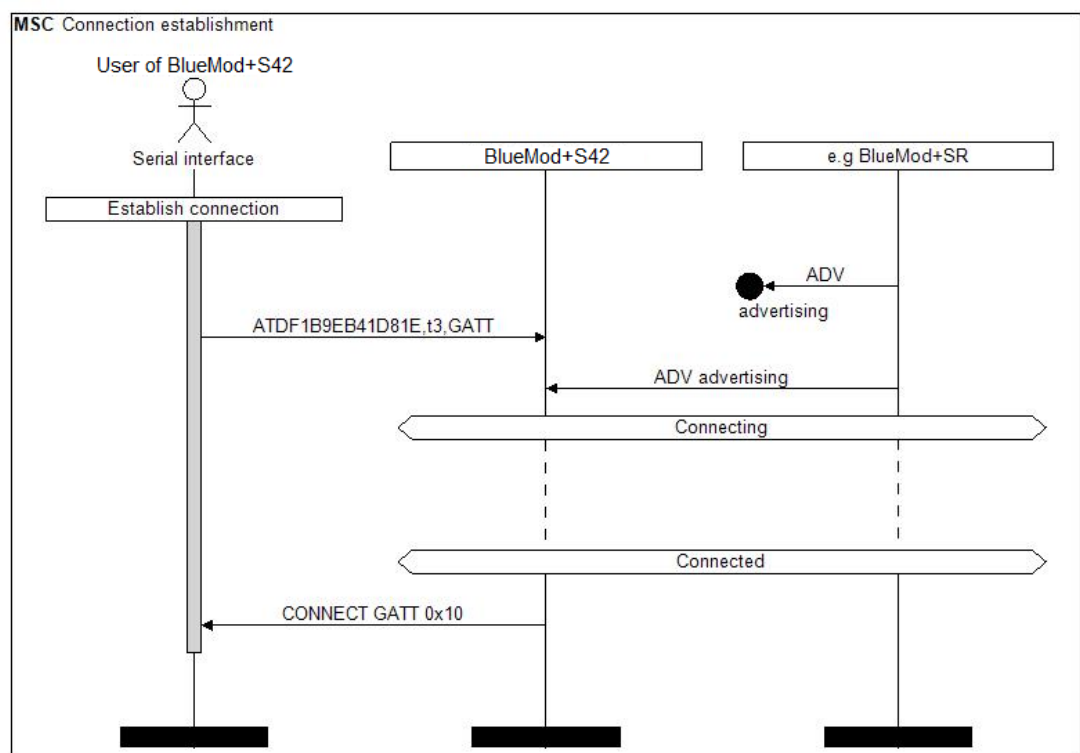


Figure 3: Connection establishment with random type Bluetooth address



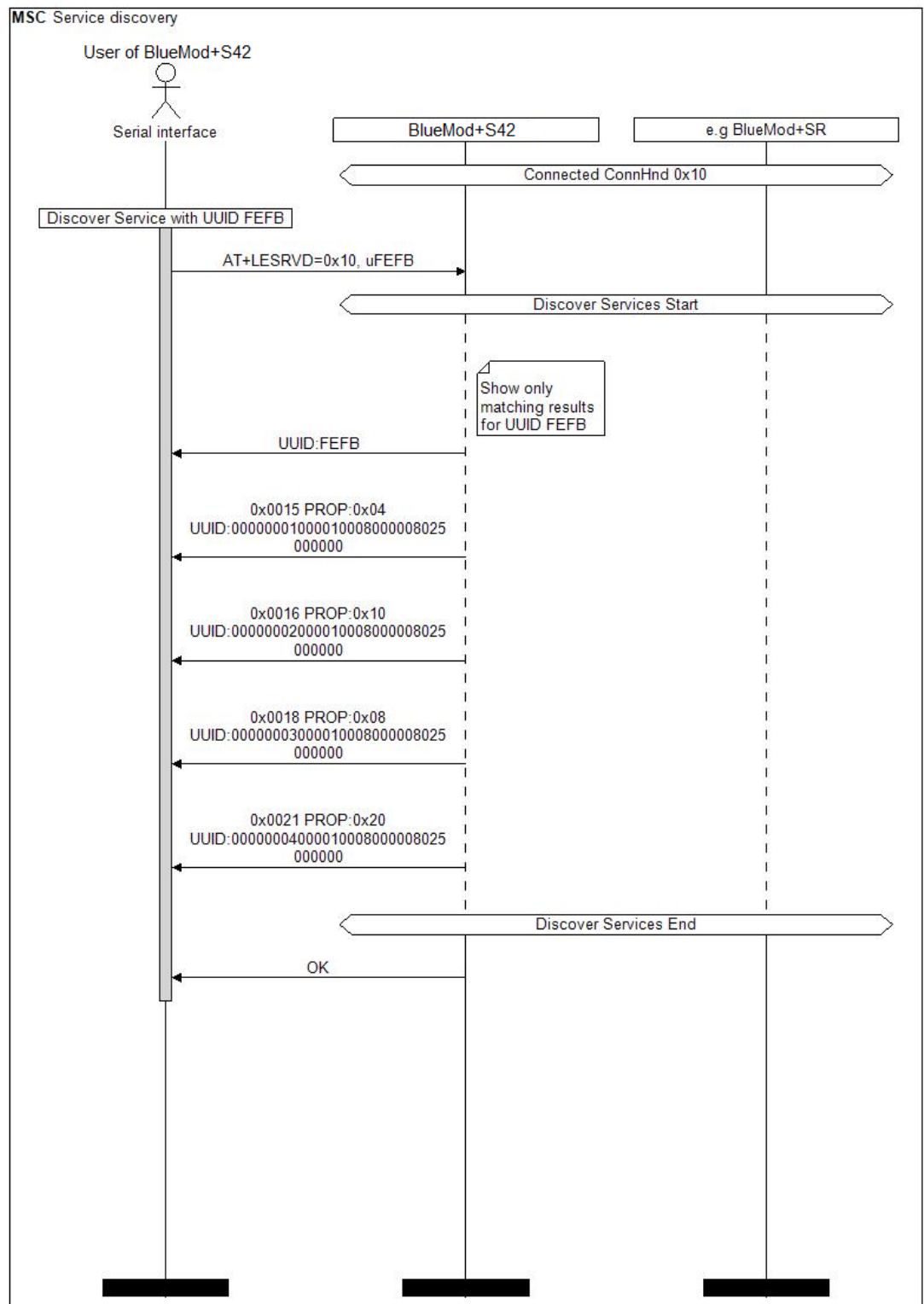


Figure 4: Service discovery

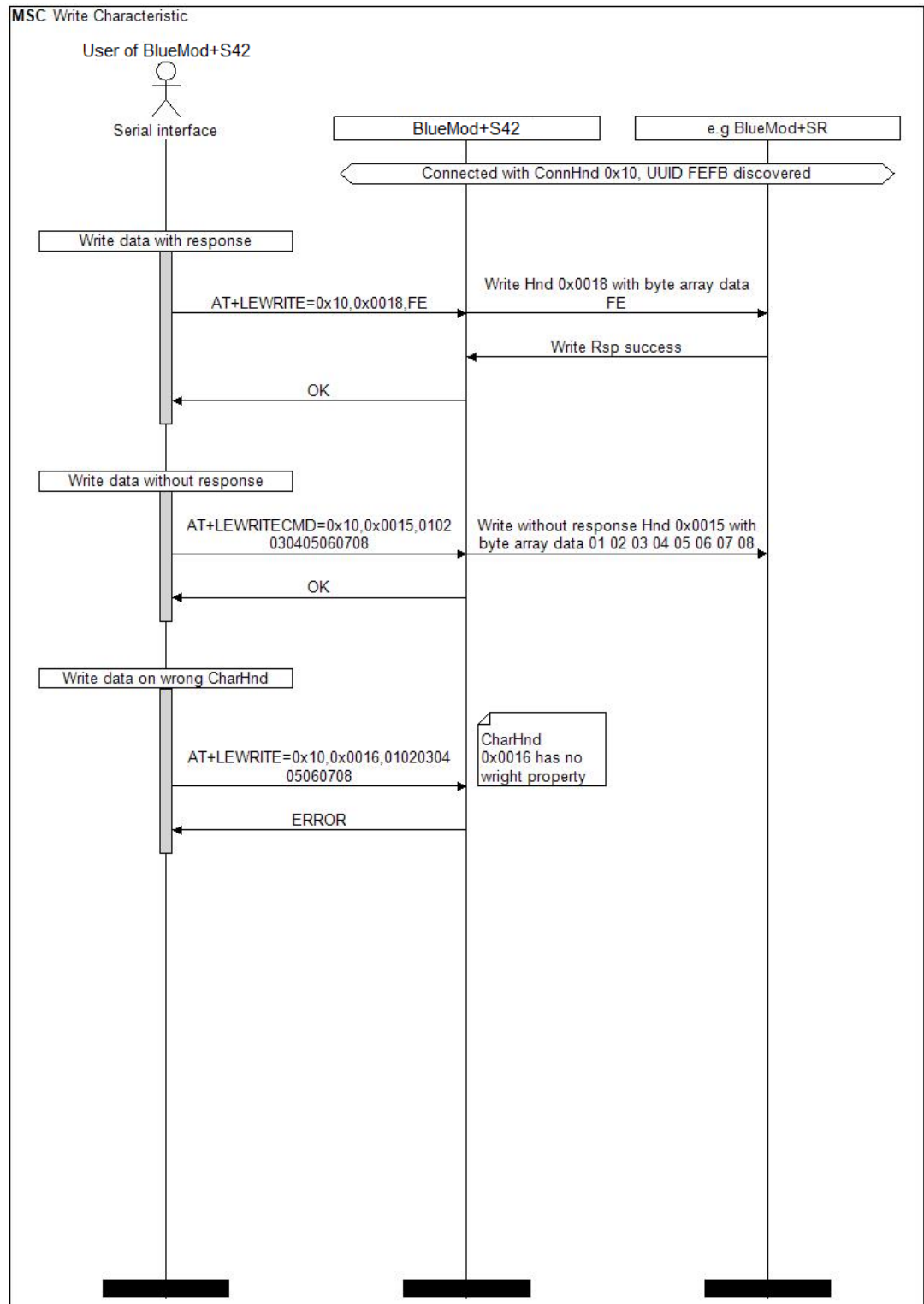


Figure 5: Write characteristic

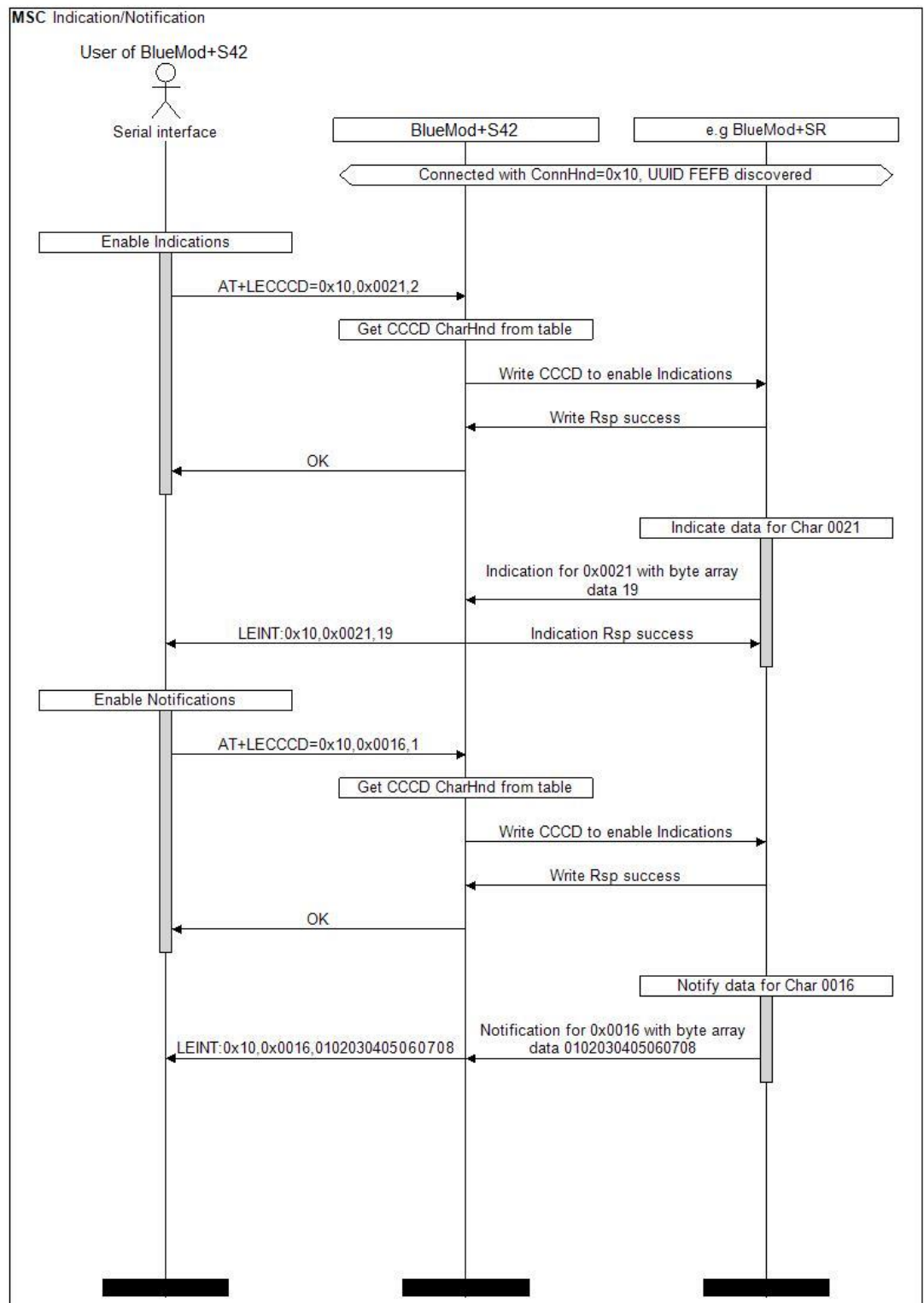


Figure 6: Indications and notification handling

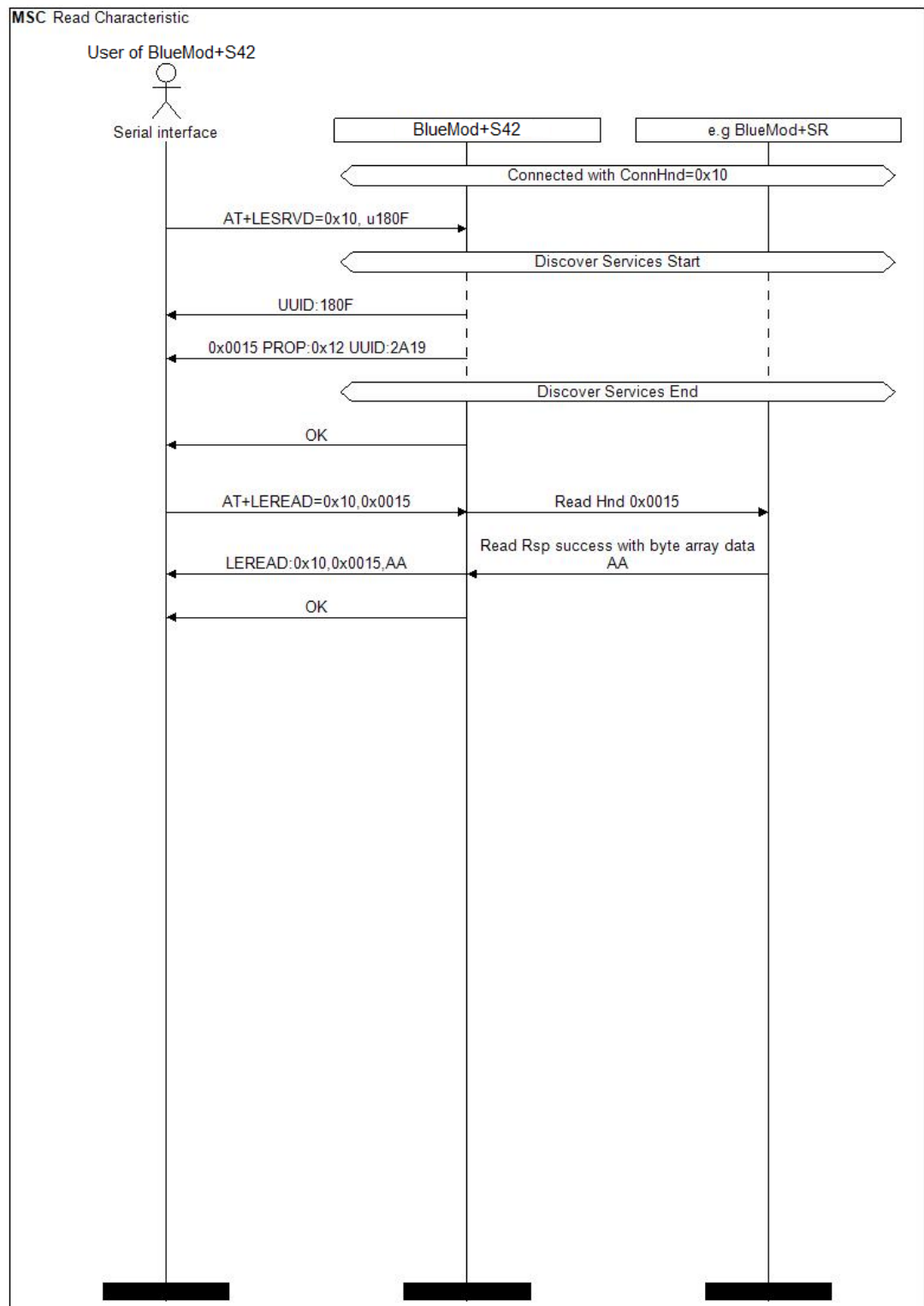


Figure 7: Read characteristic



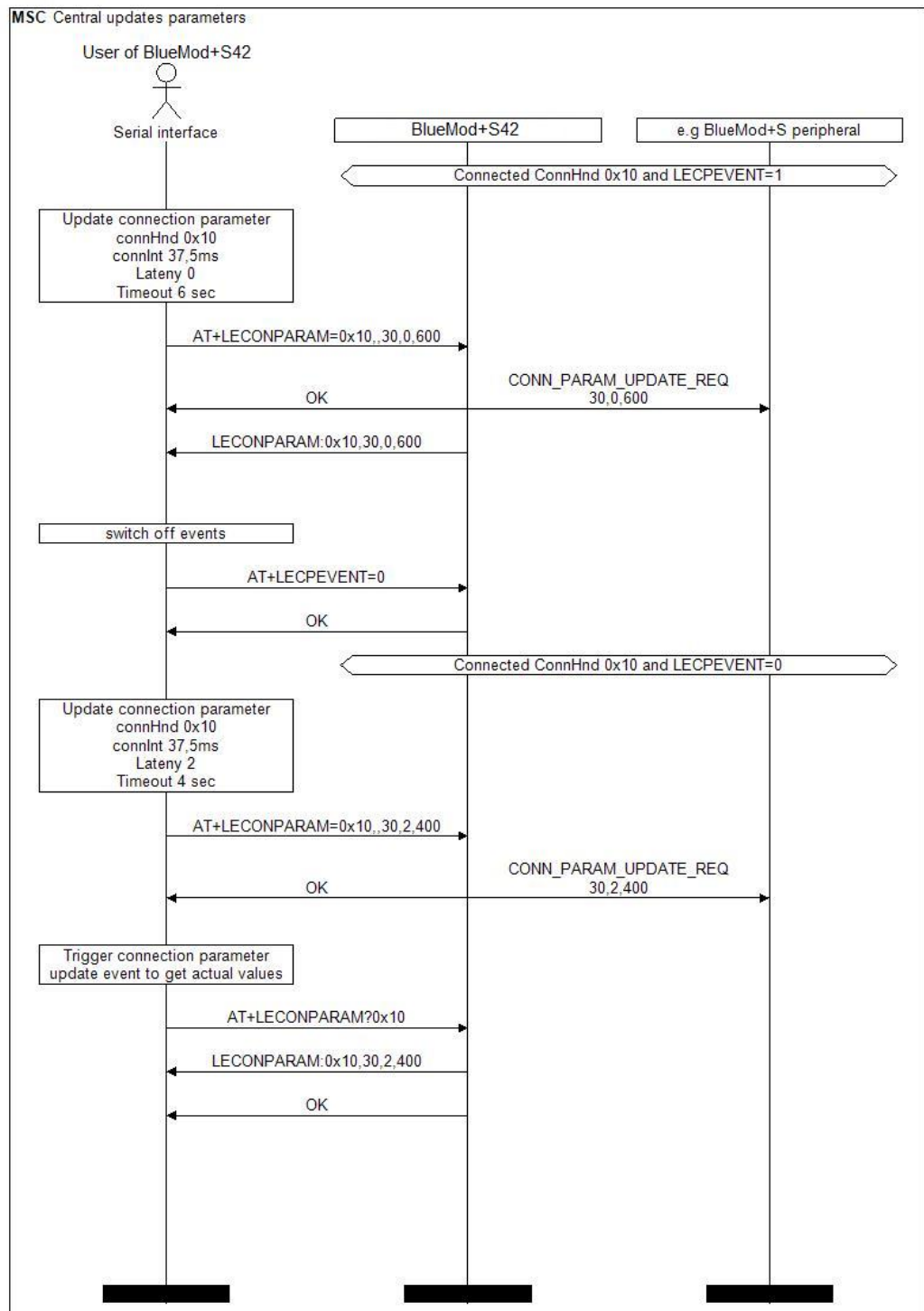


Figure 8: Central connection parameter update

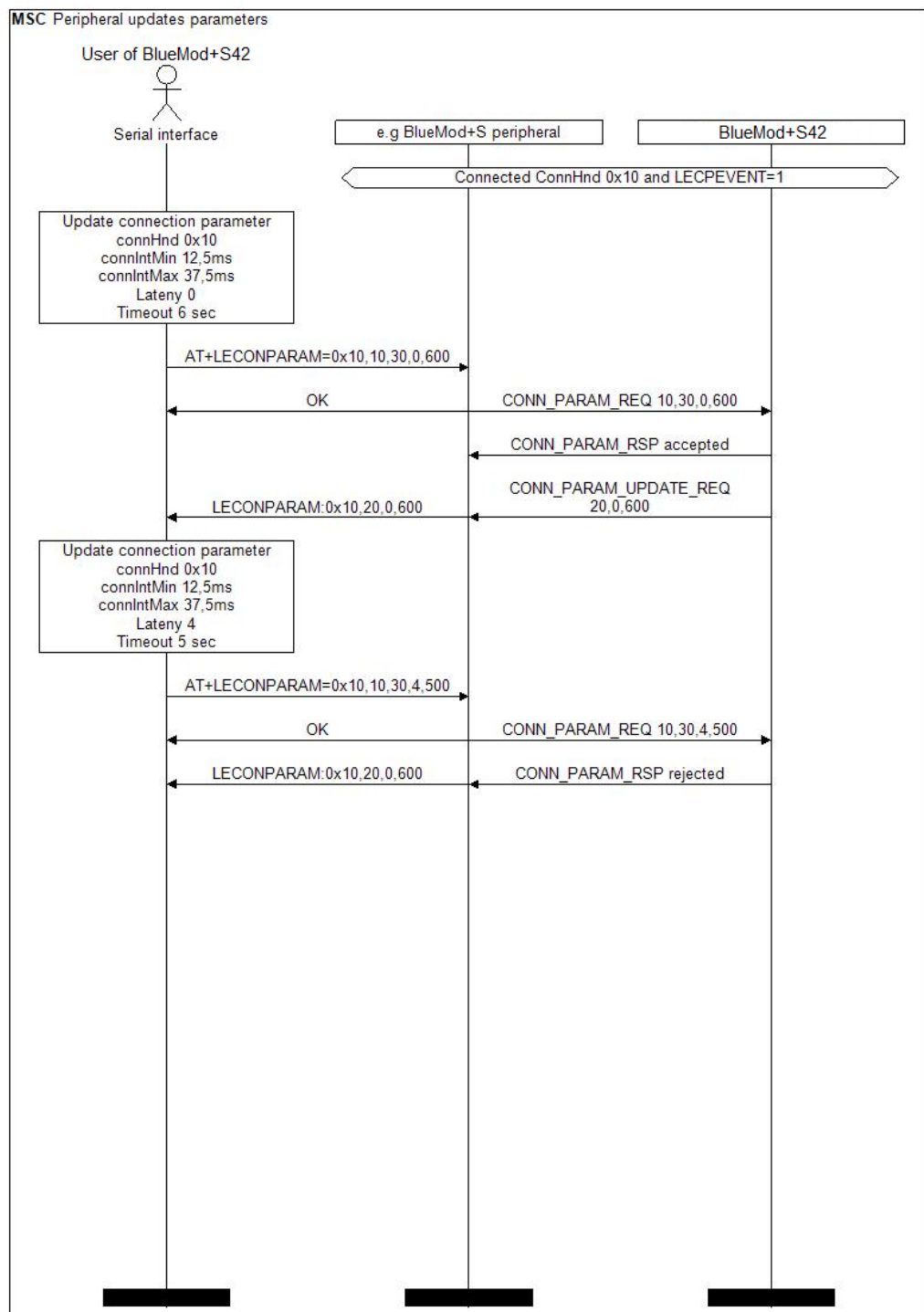


Figure 9: Peripheral connection parameter update

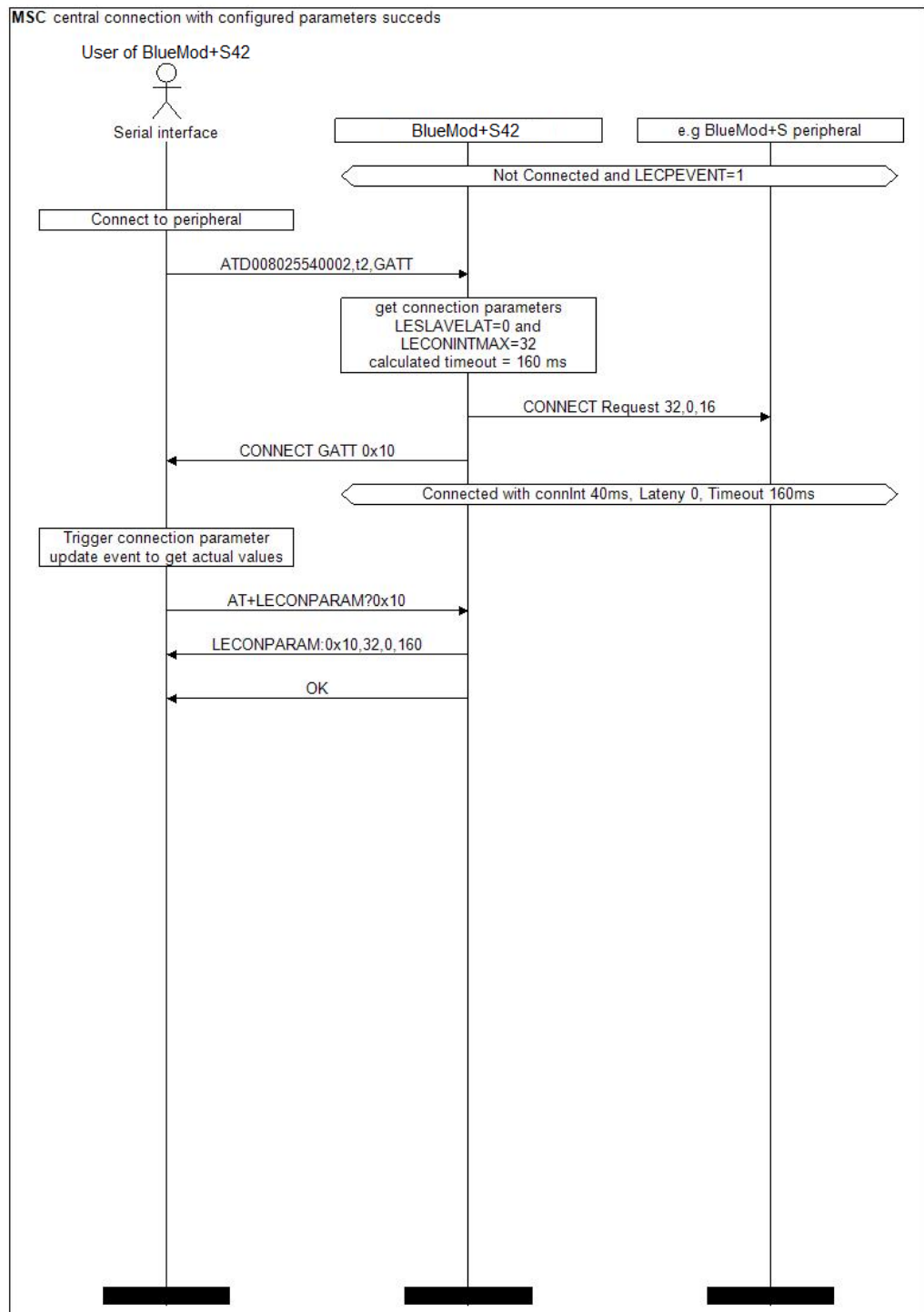


Figure 10: Central connection parameters during connection establishment



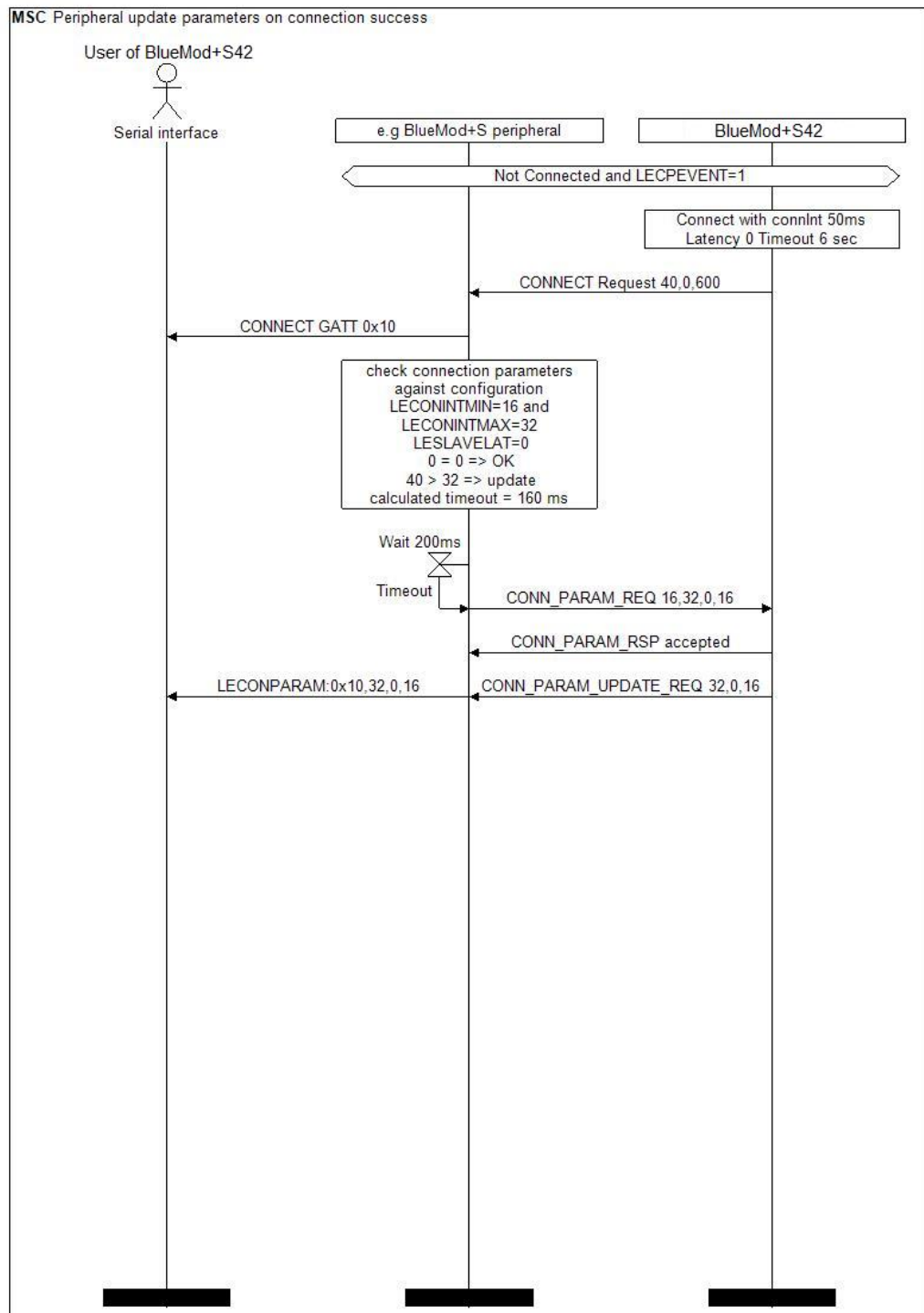


Figure 11: Peripheral connection establishment parameter update success

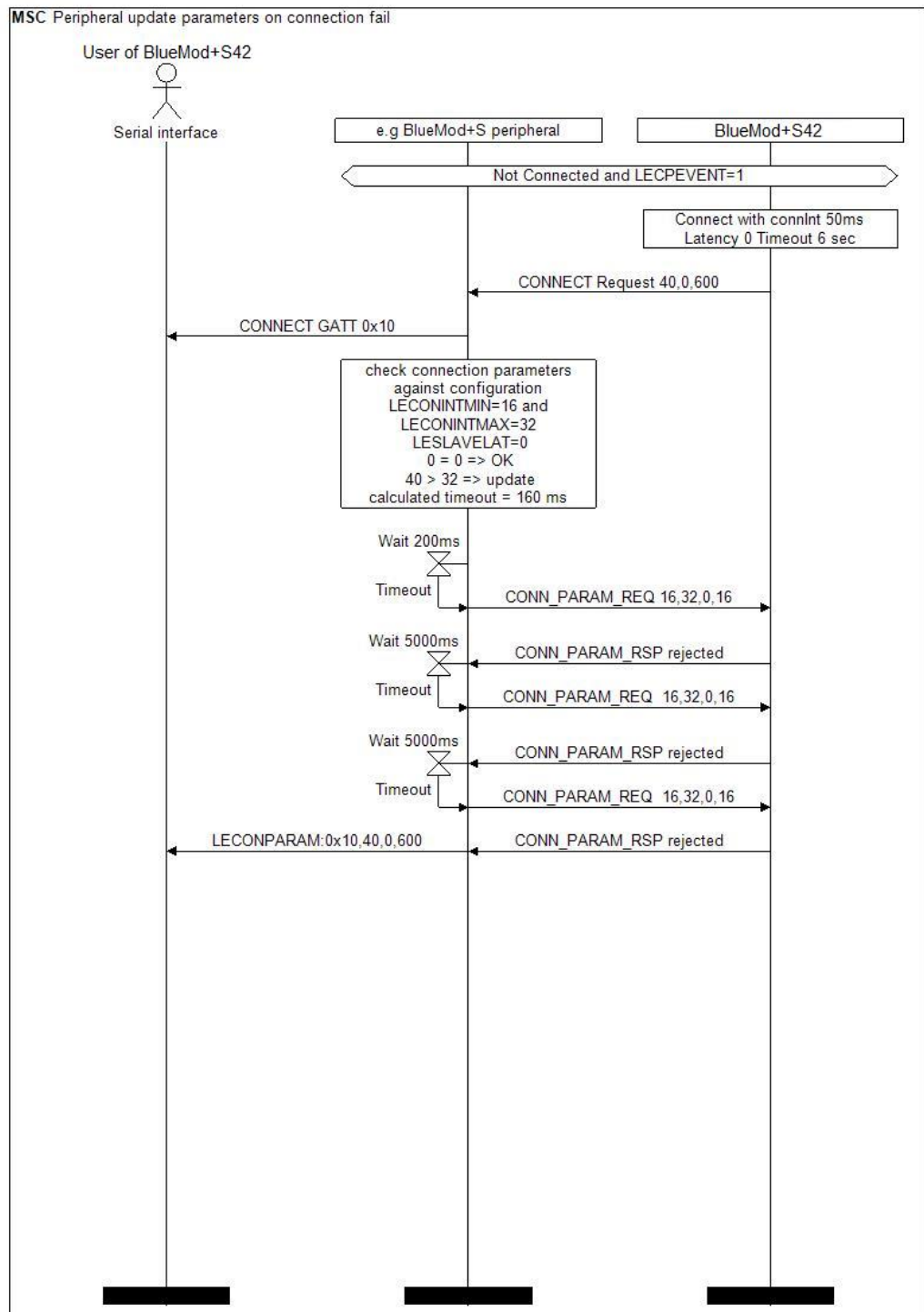


Figure 12: Peripheral connection establishment parameter update fail

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	2015-11-05	Initial version
r01d03	2015-11-19	+LECPUPDATE removed and replaced with new command +LECONPARAM new command +LECPEVENT new event LECONPARAM added new chapter “Connection Establishment Procedure” added connection parameter MSCs
r01	2016-01-18	First release
r02	2016-05-13	Added new value “2” of +IOBCFG command Added responses of list commands Added “CONNECT GATT” and “CONNECT TIO” result codes Added a list of supported advertisement types Corrected some typos
r03	2016-05-25	Telit cover page added
r04	2016-11-16	Document converted to Telit template, Added new identifier TIO and optional parameter “reusebond” to ATD command, Added +PNPVSRC, +PNPVID, +PNPPID, +PNPPVER commands, Added address type in +BSSPPIN command, Added new value “3” of +LETIO command

