

eBMU

BlueMoon™ Universal Embedded

SPP-AT Application

eBMU (PMB 8753/2), Version 1.01

eBMU SPP-AT, Version 1.1

User's Manual

Software Description

Revision 2.0

Communication Solutions



Never stop thinking



Edition 2008-10-21

**Published by
Infineon Technologies AG
81726 Munich, Germany**

**© 2008 Infineon Technologies AG
All Rights Reserved.**

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (www.infineon.com).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

eBMU

Revision History: 2008-10-21, Revision 2.0

Previous Version: PMB8753-2_SPP_AT_specification_R2.pdf

Chapter	Subjects (major changes since last revision)
2	Pull up and tri-state acronyms added
3	Master slave role switch added
4.2	Default state for GPIO, info added
4.1.2.1.1	Changes in command status values
4.1.2.1.2	General error messages updated
4.2	<ul style="list-style-type: none"> • AT+JCAC GPIO parameter explanation linked to corresponding section • AT+JGPA reserved parameter info added, response info expanded • AT+JCCR change in the response info about ERR=0 • AT+JSDA Values range: 1-MTU added, MTU maximum value changed • AT+JDDS Change in the parameter values • AT+JPCR Change in the response, +RSLE is received now
4.3	<ul style="list-style-type: none"> • +RDII info on link lost while in stream mode added • +RRSI Master/Slave role switch indication added • +RPCI example
5	MTU value changed to 230 in examples

Trademarks of Infineon Technologies AG

ABM™, A-GOLD™, AOP™, BlueMoon™, CONVERGATE™, CONVERPATH™, COSIC™, C166™, DUALFALC™, DUSLIC™, E-GOLD™, ELASTec™, EPIC™, FALC™, FLEXISLIC™, GEMINAX™, GOLDMOS™, INCA™, IOM™, IPVDT™, ISAC™, IWE™, IWORX™, M-GOLD™, MUSLIC™, OCTALFALC™, OCTAT™, OMNITUNE™, OMNIVIA™, OPTIVERSE™, PASi™, PROSOC™, QUADFALC™, SCEPTRE™, SCOUT™, SEROCCO™, S-GOLD™, SICOFI™, SIEGET™, SLIC™, SMARTi™, SMARTiPM™, SMARTiPM+™, SMARTiUE™, SMARTi3G™, SMARTi3G+™, SMINT™, SOCRATES™, TRUENTRY™, VINAX™, VINETIC™, VIONTIC™, VOIPRO™, WDCT™, WildPass™, X-GOLD™, XMM™, X-PMU™, XWAY™.

Other Trademarks

Microsoft®, Visio®, Windows® of Microsoft Corporation. Linux® of Linus Torvalds. FrameMaker® of Adobe Systems Incorporated. APOXI®, COMNEON™ of Comneon GmbH & Co. OHG. PrimeCell®, RealView®, ARM®, ARM® Developer Suite™ (ADS), Multi-ICE™, ARM1176JZ-S™, CoreSight™, Embedded Trace Macrocell™ (ETM), Thumb®, ETM9™, AMBA™, ARM7™, ARM9™, ARM7TDMI-S™, ARM926EJ-S™ of ARM Limited. OakDSPCore®, TeakLite® DSP Core, OCEM® of ParthusCeva Inc. IndoorGPS™, GL-20000™, GL-LN-22™ of Global Locate. mipi™ of MIPI Alliance. CAT-iq™ of DECT Forum. MIPS™, MIPS II™, 24KEc™ of MIPS Technologies, Inc. Texas Instruments®, PowerPAD™, C62x™, C55x™, VLYNQ™, Telogy Software™, TMS320C62x™, Code Composer Studio™, SSI™ of Texas Instruments Incorporated. Bluetooth® of Bluetooth SIG, Inc. IrDA® of the Infrared Data Association. Java™, SunOS™, Solaris™ of Sun Microsystems, Inc. Philips®, I2C-Bus® of Koninklijke Philips Electronics N.V. Epson® of Seiko Epson Corporation. Seiko® of Kabushiki Kaisha Hattori Seiko Corporation. Panasonic® of Matsushita Electric Industrial Co., Ltd. Murata® of Murata Manufacturing Company. Taiyo Yuden™ of Taiyo Yuden Co., Ltd. TDK® of TDK Electronics Company, Ltd. Motorola® of Motorola, Inc. National Semiconductor®, MICROWIRE™ of National Semiconductor Corporation. IEEE® of The Institute of Electrical and Electronics Engineers, Inc. Samsung®, OneNAND®, UtRAM® of Samsung Corporation. Toshiba® of Toshiba Corporation. Dallas Semiconductor®, 1-Wire® of Dallas Semiconductor Corp. NOVeA™ of Virage Logic Corp. ISO® of the International Organization for Standardization. IEC™ of the International Engineering Consortium. EMV™ of EMVCo, LLC. Zetex® of Zetex Semiconductors. Rohm™ of Rohm Co., Ltd. Microtec® of Microtec Research, Inc. Verilog® of Cadence Design Systems, Inc. ANSI® of the American National Standards Institute, Inc. WindRiver® and VxWorks® of Wind River Systems, Inc. Nucleus™ of Mentor Graphics Corporation. OmniVision® of OmniVision Technologies, Inc. Sharp® of Sharp Corporation. Symbian OS® of Symbian Software Ltd. Openwave® of Openwave Systems, Inc. Maxim® of Maxim Integrated Products, Inc.

The information in this document is subject to change without notice.

Last Trademarks Update 2008-05-27

Table of Contents

1	Introduction	6
2	Bluetooth Features	7
3	Serial Port Profile	8
3.1	Operation Modes.....	8
3.1.1	Command Mode.....	8
3.1.2	Stream Mode.....	8
3.2	PIN Assignments.....	8
4	Infineon SPP-AT Command and Response.....	9
4.1	AT Command and Response Format and Syntax	9
4.1.1	AT-commands	10
4.1.2	AT-responses	10
4.1.2.1	AT Response Parameter List for <status> and <ERROR>	10
4.1.2.1.1	Command Execution Status Values	10
4.1.2.1.2	General Error Messages	10
4.2	AT Command List Table	11
4.3	AT Responses List (not command triggered)	18
4.4	Crystal Auto Calibration (Frequency counter method).....	19
4.5	Low Power Mode Control.....	19
4.5.1	Host Initiates Low Power Mode Entry and Exit	20
4.5.2	Host Initiates Low Power Mode Entry, eBMU Initiates Exit	20
4.6	UART Baud Rate Change.....	21
4.7	Data Flow Control	21
4.8	Production Mode	21
4.9	SPP-AT Commands Operating with EEPROM.....	22
4.10	Security Mode	22
5	Example AT Commands and Responses	23
	References	26
	Terminology	27

1 Introduction

This document describes the AT commands applicable to Infineon's eBMU Bluetooth solution for embedded applications. The product utilizes a raw AT set of commands and responses over UART for Bluetooth communications and control on an embedded device.

2 Bluetooth Features

- Bluetooth v2.0 + EDR compliant.
- Device A and B support 1 ACL link with stream or command mode.
- Device A and B - Visible while connected.
- Device A and B - Visible/connectable when not connected.
- Device A and B - Device Discovery capable after receiving OK on data transfer.
- Sniff mode is supported with above capabilities.
- 5 trusted devices stored in EEPROM.
- Testing
- Enable DUT.
- Crystal calibration.
- H4 with UART HW flow control (RTS/CTS).
- Security modes
- Mode 1 and Mode 3.
- Master-Slave role switch

3 Serial Port Profile

Host communication sent over UART is always called command except while in stream mode, see below. All communication received by host application over UART is called response except while in stream mode.

3.1 Operation Modes

The specification defines two operation modes of the PMB 8753/2 also named eBMU throughout the document. For Bluetooth SPP there are two different roles specified; A-device (paging device) setting up the over the air connection and B-device which is connectable and accepts the connection (page scanning device).

3.1.1 Command Mode

In this mode the SPP application running on the eBMU will execute the AT commands sent from the host over the UART using the H4 UART protocol as specified within the Bluetooth SIG [1]. In this mode, the host application can send data to the eBMU, which are transmitted to the remote device which has a Bluetooth connection on SPP level with the eBMU. This mode is normally used when transmitting burst and packetized data.

Setting up/accepting Bluetooth SPP connections and/or searching for other Bluetooth devices are also operations done in this mode.

3.1.2 Stream Mode

In this mode, the host application will send un-packetized data to the eBMU, which are transmitted over the air to the remote device. This mode is normally used when transmitting small size of data in a random way and for serial cable replacement applications.

3.2 PIN Assignments

The table below shows the available GPIOs. GPIOs noted with “(Reserved)” can not be controlled by a host. P0.1 and P0.8 can be used as application GPIOs but not when they are used during the crystal calibration procedure. All application GPIOs are by default configured as tri-state.

PIN name	Default Configuration	Direction	GPIO	Description
UART_RX	I	I	P0.5 (Reserved)	UART
UART_TX	O PU	O	P0.4 (Reserved)	UART
UART_RTS	O PU	O	P0.6 (Reserved)	UART
UART_CTS	I	I	P0.7 (Reserved)	UART
SDA	O PU	I/O	P0.12 (Reserved)	I2C
SCL	I PU	I	P0.13 (Reserved)	I2C
PCMFR1	O PU	O	P0.0 (Reserved)	LPM
TX_Conf2	I	I	P0.14 (Reserved)	LPM
PCMCLK	Z	I/O	P0.1	APPL GPIO / XTAL
PCMIN	Z	I/O	P0.2	CAL
PCMOUT	Z	I/O	P0.3	APPL GPIO
PAON	Z	I/O	P0.8	APPL GPIO
PSEL0	Z	I/O	P0.9	APPL GPIO / XTAL
PSEL1	Z	I/O	P0.10	CAL
TX_Conf1	Z	I/O	P0.11	APPL GPIO
P015	Z	I/O	P0.15	APPL GPIO
				APPL GPIO

4 Infineon SPP-AT Command and Response

AT commands can only be sent while in command mode with the exception of stream connection cancel (^^^) which can be sent in stream mode. The expected response after sending an AT command is the "OK" response, see specification below. The host shall wait for a command to be terminated before sending a new one. A command is considered as terminated when "OK" and all subsequent related responses have been received.

There are also responses, which are not initiated by a sent command. They are in that case initiated by the remote Bluetooth device.

4.1 AT Command and Response Format and Syntax

All data exchanged between the host and eBMU is in ASCII format.

Parameters for commands and responses are given in decimal (DEC) base in ASCII format unless hexadecimal (HEX) base is specified.

MSB is always sent first. eBMU does not distinguish between upper and lower cases.

Example for values with decimal base in ASCII format

The number 255 in decimal corresponds to the following three characters '2', '5', '5'. E.g. AT+JSDA=010,1234567890 (the number 10 is given by three ASCII characters '0', '1', '0').

Example for values with hexadecimal base in ASCII format

The number 255 in decimal base corresponds to the number FF in hexadecimal base, thereby the number in ASCII format for hexadecimal base is represented by the following two characters 'F', 'F'.

E.g. +RSNFCNF=3E80,2 ; The four ASCII characters '3', 'E', '8', '0' represent the number 3E80 in hexadecimal base which corresponds to 16000 in decimal base.

4.1.1 AT-commands

All AT-commands follow the format below;

AT+<command>=<parameter 1 (if required)>,<parameter 2 (if required)>,<parameter 3 (if required)> ,<...><carriage return><line feed>

E.g.: AT+JCCR=0010c64d67dc,01 (to connect to BD_ADDR "0010c64d67dc", service channel 1)

4.1.2 AT-responses

All AT-responses follow the format below with the exception of <OK> and <ERROR=>;

+<response>=<parameter 1 (if required)>,<parameter 2 (if required)>,<parameter 3 (if required)>,<...><carriage return><line feed>

E.g.: +RDAI =004,DATA (4 bytes ("DATA") received)

4.1.2.1 AT Response Parameter List for <status> and <ERROR>

4.1.2.1.1 Command Execution Status Values

Values for <status> general for all commands

- BT_OK 0
- BT_ERROR 1
- BT_TIMEOUT 4

4.1.2.1.2 General Error Messages

- ERR = -1; Syntax Error
- ERR = -2; Command not allowed at present execution status
 - ERR=-2 is obtained when any of the following situations is met:
 1. Create a new connection when already connected.
 2. Device discovery when already connected.
 3. Send data when not connected.
 4. Accept connection request without a request.
 5. Enter sniff mode without being connected.
 6. Enable SEC mode after another command has previously been sent.
 7. Service discovery when connected.
 8. Send data with length = 0.

4.2 AT Command List Table

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
AT+JAAC=<auto_accept>	<p>Auto Accept Connection requests Forces eBMU to accept connection requests.</p> <p>auto_accept parameter (1 octet): 0 – Host will be notified on incoming connection indication (+RCOI) – (no auto accept). Default value 1 – eBMU will automatically auto accept incoming connection request – (host will be notified but connection is accepted automatically)</p>	OK	Dev B
AT+JACR= <accept>	<p>Accept Connection Request Shall be used as answer to a connect indication (+RCOI).</p> <p>accept parameter (1 octet): 0 – Not accepted 1 – Accepted</p>	OK	Dev B
AT+JEDT	<p>Enable Device under Test This SPP-AT command enables the device under test. After this command has been sent it is possible for a remote tester to connect, this AT command corresponds to the three different HCI commands listed below:</p> <ol style="list-style-type: none"> 1. Set Event Filter – allow all connections. 2. Write Scan Enable – page and Inquiry. 3. Enable device under test. <p>Note: JEDT command can only be issued with security mode 1 and after production mode is enabled (JPRO=1). See section 4.8.</p>	OK	Dev A Dev B
AT+JCAC=<trim_value>,<GPIO >	<p>Crystal Auto Calibrate</p> <p>trim_value parameter (4 octets / HEX base): The trim value is used to adjust the frequency on the GPIO chosen by the GPIO parameter.</p> <p>GPIO parameter (4 octets / HEX base): Two GPIOs can be used as output for the oscillator trim, P01 and P08 (0002 and 0100). See section 4.4.</p> <p>Note: JCAC command can only be issued after production mode is enabled (JPRO=1). See section 4.8.</p>	<p>+RCACNF=<osc_trim> osc_trim parameter(4 octets / HEX base)</p> <p>Use AT+JCBD to store the osc_trim value to EEPROM.</p>	Dev A Dev B
AT+JCBD=<bd_data>	<p>Change BD_Data</p> <p>bd_data parameter (116 octets / HEX base): (reference values) msg.bdAddr (12 octets)= see note 1 msg.channelWordOffset (4 octets) =0000; msg.clkConf (2 octets)= 8A; msg.eepromSize (2 octets)=80;</p>	OK	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p>msg.inputFreq (8 octets)=018CBA80; msg.ImpFeatures (16 octets)=000019987E0602BF; msg.lpmConf (2 octets)=40; msg.lpmDrift (2 octets)=FA; msg.lpmThreshold (2 octets)=12; msg.ulpmThreshold (2 octets)=18; msg.pmuConfig (4 octets)=0080; msg.rfPselID (8 octets)=06050403; msg.rfPselConf (2 octets)=44; msg.rssiMin (2 octets)=0C; msg.rssiMax (2 octets)=10; msg.ddcTlConf (2 octets)=02; msg.uartBaudrate (2 octets)=04; msg.uartInvert (2 octets)=00; msg.uartPulls (2 octets)=01; msg.oscSettle (2 octets)=08; msg.bbConf (2 octets)=04; msg.rfConf (2 octets)=04; msg.txPowerRef0 (2 octets)=F2; msg.txPowerRef1 (2 octets)=F8; msg.txPowerRef2 (2 octets)=FE; msg.txPowerRef3 (2 octets)=04; msg.oscTrim (4 octets)=see note 2; msg.threeWireArqTimeout (2 octets)=06; msg.LMP_Version (2 octets)=00; msg.Reserved (16 octets)=0000000000000000;</p> <p>Note 1: This value shall be configured with the desired Bluetooth device Address (e.g. FFAA010203BB).</p> <p>Note 2: This value shall be configured with the value found during crystal calibration (see command AT+JCAC).</p> <p>Note 3: JCBD command can only be issued after production mode is enabled (JPRO=1). Changes in BD Data take place after production mode command is disabled (JPRO=0) followed by a SW reset (AT+JRES). See section 4.8.</p>		
AT+JCCR=<bd_addr>, <service channel>	<p>Create Connection Request Instructs eBMU to connect to a remote Bluetooth device (prospective slave).</p> <p>bd_addr parameter (12 octets / HEX base): The Bluetooth address of the remote device</p> <p>service_channel parameter (2 octets / DEC base): 0-30 Which service channel to connect to can be received from a Service Discovery (AT+JSDS)</p>	<p>OK (Followed by): +RCCRCNF=<MTU_size>,<status> MTU_size parameter (3 octets / DEC base) status parameter (1 octet)</p> <p>If maximum number of allowed connections already exists: ERR=-2.</p>	Dev A
AT+JDDS=<limit_inquiry_result>	<p>Device Discovery Start Causes eBMU to start a Device Discovery (Inquiry and Remote Name Request) of the Bluetooth neighborhood.</p>	<p>OK (Then, if responses are returned): +RDDSRES=< bd_addr ><remote_name>,<COD> (For each response)</p>	Dev A

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p>limit_inquiry_result parameter (1 octet): Values range: 0-1. The maximum number of responses that can be obtained is 8.</p> <p>Note: values in the limit_inquiry_result parameter have no impact in the behavior of the command.</p>	<p>bd_addr parameter (12 octets / HEX base). remote_name parameter (variable length): Name of the remote device. Page Timeout is received as a name if the remote name is not found during device discovery COD parameter (6 octets / HEX base). Class of device</p> <p>(Completed by): +RDDSCNF=<status> status parameter (1 octet)</p>	
AT+JDIS=<discoverable> >	<p>DIScoverable Forces eBMU into Page Scan / Inquiry Scan states indefinitely (note: this makes the device discoverable).</p> <p>discoverable parameter (1 octet): 0- No scans enabled. 1- Inquiry Scan enabled. 2- Page Scan enabled. 3- Inquiry & Page Scan enabled.</p> <p>Note: Default value after a HW reset is no scans enabled</p>	OK	Dev B Dev A is discoverable but does not accept incoming connections.
AT+JDOI	<p>DOWNload Application Image via UART After "OK" response a binary file with the EEPROM image should be sent.</p> <p>File open and send in binary format after "OK" is received.</p> <p>Note: JDOI command can only be issued after production mode is enabled (JPRO=1). Changes in the EEPROM image are effective after production mode command is disabled (JPRO=0) followed by a HW reset. See section 4.8.</p>	<p>OK after command is sent.</p> <p>+RDOICNF after .eep file has been written.</p> <p>Note: eBMU calculates the size of the image that is being downloaded from the file header. The response +RDOICNF is generated when the calculated size is reached.</p>	Dev A Dev B
AT+JGPA=<reserved>,<read>,<set>,<clear>	<p>GPIO action All parameters of this command are bit fields of 16 bits corresponding to GPIOs P0.15 to P0.0. E.g. if pin P0.0 is the desired bit; the bit field value is 0001 and if the desired bit is P0.12 the bit field value is 0800.</p> <p>reserved parameter (4 octets) read parameter (4 octets / Hex base): Values for each bit: 0- No Action 1- Read set parameter (4 octets / HEX base): Values for each bit: 0- No Action 1- Set clear parameter (4 octets / HEX base): Values for each bit:</p>	<p>+GPOACNF=<value></p> <p>value parameter (4 octets / HEX base): This value is the state of the GPIO PINs specified in read parameter. Values for each bit: 0 – means low. 1 – means high.</p> <p>When the pins are set as outputs the return value will be 0 for the specific pin.</p> <p>Reserved pins will always return 0</p>	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	0- No Action 1- Clear		
AT+JGPC= <i><direction></i> , <i><open_drain></i> , <i><pull_on/off></i> , <i><pull_up/down></i> , <i><tristate></i>	GPIO Configuration All parameters of this command are bit fields of 16 bits corresponding to GPIOs P0.15 to P0.0 (See command AT+JGPA). <i>direction</i> parameter (4 octets / HEX base): Values for each bit: 1 - IN 0 - OUT <i>open_drain</i> parameter (4 octets / HEX base): Values for each bit: 1 - OPEN <i>pull_on/off</i> parameter (4 octets / HEX base): Values for each bit: 1 - Pull ON 0 - Pull OFF <i>pull_up/down</i> parameter (4 octets / HEX base): Value for each bit: 1 - Pull UP <i>tristate</i> parameter (4 octets / HEX base): Value for each bit: 1 - Tri-state Note: See section 3.2 for available GPIO pins.	OK	Dev A Dev B
AT+JPCR= <i><length_PIN_code></i> , <i><PIN_code></i>	PIN Code Reply Sent to eBMU in response to a PIN Code Request from a remote Bluetooth device (<i>bd_addr</i>). <i>length_PIN_code</i> parameter (2 octets / DEC base): Values= 1-16 Length of PIN code <i>PIN_code</i> parameter (length= <i>length_PIN_code</i>): The PIN Code to be sent to the remote Bluetooth device, e.g. AT+JPCR=04,1234 Note: the PIN code is an ASCII string.	OK (Followed by): +RSLE if secure link is established	Dev A Dev B
AT+JPRO= <i><mode></i>	PROduction mode <i>mode</i> parameter (1 octet): 1= Production mode ON 0= Production mode OFF Note: See section 4.8 for commands requiring production mode.	OK	Dev A Dev B
AT+JRBD	Read Bluetooth device Data Sent to eBMU to retrieve the Bluetooth Device Address and the Oscillator trimming value.	+RRBDRES= <i><bd_addr></i> , <i><oscTrim></i> <i>bd_addr</i> parameter (12 octets / HEX base): BD Address of remote device <i>oscTrim</i> parameter (4 octets)	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
		/ HEX base) Oscillator trim value	
AT+JRES	RESet SW reset of the system.	ROK as applications is restarted.	Dev A Dev B
AT+JRLS=<uuid>,<length_service_name ><service_name>,<service channel >,<CoD>	Register Local Service uuid parameter (4 octets / HEX base): uuid for supported profile e.g.1101 for Serial Port Profile length_service_name parameter (2 octets / DEC base): Values=1-16 Length of service name service_name parameter (length=length_service_name): Name for the service, no final delimiter is needed service channel parameter (2 octets / DEC base): Values range: 0-30 Which service channel to connect to; can be received from a Service Discovery with AT+JSDS CoD parameter (6 octets / HEX base) Class of device. The default CoD value is 000000	OK	Dev B
AT+JRRI	Read Revision Information	+RRRICNF=<revision> revision parameter (2 octets / HEX base)	Dev A Dev B
AT+JRTD=<bd_addr>	Remove Trusted Device – deletes the trusted device information for a registered device (bd_addr). bd_addr parameter (12 octets / HEX base): The Bluetooth address of the device that shall be removed from the list	OK	Dev A Dev B
AT+JSCR	Stream Connection Request Connects the SPP and UART streams, transparent communication will be enabled if both sides execute this command.	OK	Dev A Dev B
AT+JSDA=<length>,<data >	Send Data Request length parameter (3 octets / DEC base): number of bytes to be sent Values range: 1-MTU data parameter (see note for size): data to be sent Note: Maximum number of bytes for each packet is reported at connection confirmation (MTU_Size) MTU size for eBMU is 230 bytes but this parameter is negotiated under connection set up.	OK	Dev A Dev B
AT+JSDR	SPP Disconnect Request Forces an SPP disconnection.	OK	Dev A Dev B
AT+JSDS=<bd_addr>,<uuid>	Service Discovery Start Causes eBMU to start a service discovery of device with bd_addr and search for services defined by uuid.	OK (Then, if services are returned): +RSDSRES=<remote_servic	Dev A

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p>bd_addr parameter (12 octets / HEX base): BD Address of remote device</p> <p>uuid parameter (4 octets / HEX base): Service to search for e.g. 1101 for Serial Port Profile</p>	<p>e name <, <remote_service_channel></p> <p>remote_service_name parameter (variable length): Name of the remote service.</p> <p>remote_service_channel parameter (2 octets / DEC base)</p> <p>(For each service) (Completed by): +RSDSCNF=<status> status parameter (1 octet)</p>	
<p>AT+JSEC=<security_mode>, <Link_key_information>, <PIN_type>, <length_PIN_code>, <PIN_code></p>	<p>Enable SECURITY</p> <p>security_mode parameter (1 octet):</p> <ol style="list-style-type: none"> Security Mode 1 N/A Security Mode 3 (default) <p>link_key_information parameter (1 octet):</p> <ol style="list-style-type: none"> Inform about link key Don't inform about link key (default) <p>PIN_type parameter (1 octet)</p> <ol style="list-style-type: none"> Variable PIN Fixed PIN <p>length_PIN_code parameter (2 octets / DEC base): Length of PIN code. The maximum PIN length value is 16 (corresponding to a 16 octets long PIN code)</p> <p>PIN_code parameter (length=length_PIN_code): Normal user PIN, for example "0000" (default), This parameter is taken into account if PIN_TYPE is fixed.</p> <p>Note: If security mode shall be changed from default settings, AT+JSEC shall be the first command that is sent after a reset. Security mode 3 is the default security mode and the default PIN type is variable.</p>	OK	Dev A Dev B
<p>AT+JSLN=<length_friendly_name>, <friendly_name></p>	<p>Set Local device friendly Name Supports all ASCII characters.</p> <p>length_friendly_name parameter (2 octets / DEC base): Length of friendly name, the maximum value for length is 18.</p> <p>friendly_name parameter (length=length_friendly_name): No delimiter is required.</p>	OK	Dev A Dev B
<p>AT+JSNF=<sniff_Max>, <sniff_Min>, <sniff_attempt>, <sniff_tmo>, <on/off></p>	<p>Sniff Request Request a link to enter Sniff Mode. All command parameters are given in HEX base with the exception of on/off parameter. 1 ASCII character represents four bits in HEX base e.g. "A" is 1010. MSB given first.</p>	<p>+RSNFCNF= <sniff_Interval>, <mode></p> <p>sniff_Interval parameter (4 octets / HEX base): mode parameter (1 octet): One octet to indicate normal mode (mode=0) or</p>	Dev A Dev B

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p>sniff_Max parameter (4 octets / HEX base): Maximum allowed sniff interval Value to be written= N Time = N * 0.625 msec Range: 1.25 msec to 40.9 s</p> <p>sniff_Min parameter (4 octets / HEX base): Minimum allowed sniff interval Value to be written= N Time = N * 0.625 msec Range: 1.25 msec to 40.9 s</p> <p>sniff_attempt parameter (4 octets / HEX base): Number of sniff attempts Value to be written= N Length = N* 1.25 msec Time Range: 0.625msec - 40.9 s</p> <p>sniff_tmo parameter (4 octets / HEX base): The time out value for sniff attempts Value to be written= N Time = N * 0.625 msec Range: 0 msec to 40.9 s</p> <p>on/off parameter (1 octet): One octet to indicate Sniff ON (value=1) or Sniff OFF (value=0)</p>	sniff mode (mode=2)	
^ ^ ^	<p>Stream Connection Cancel End Streaming Mode</p> <p>Send -T0 - ^ -T1- ^ -T1- ^ -T0- with an interval of T1= 1 second between symbols and T0 > 1second.</p> <p>Note: This string is not terminated with CR LF.</p>	OK	Dev A Dev B

4.3 AT Responses List (not command triggered)

AT Response	Usage	Command to acknowledge the response	Dev A / Dev B
ROK	Start up response		Dev A Dev B
+RPCI=<bd_addr>	PIN Code Indication bd_addr parameter (12 octets / HEX base): MSB is sent first.	AT+JPCR=<length_PIN_code>, <PIN_code> length_PIN_code parameter (2 octets / DEC base): Length of PIN code PIN_code parameter (length=length_PIN_code):	Dev A Dev B
+RCOI=<bd_addr>	Connect Indication bd_addr parameter (12 octets / HEX base): See previous response for information on the format.	AT+JACR= <accept> accept parameter (1 octet): 0 – Not accepted 1 – Accepted	Dev B Note: B should disable connectable mode
+RDAI=<length>,<data>	Data Indication length parameter (3 octets / DEC base): number of bytes to be sent data parameter (length=length from previous parameter) Received data	Not available during stream mode.	Dev A Dev B
+RDII	Disconnect Indication	Received on the side that has not initiated the disconnection. During stream mode, if the link is lost, the response +RDDI is received after the link supervision time out which is 20 seconds long.	Dev A Dev B
+RSLE	Secure Link Established		Dev A Dev B
+RSNFCNF=<sniff_interval>,<mode>	Sniff mode confirmation sniff_interval parameter (4 octets / HEX base). mode parameter (1 octet): One octet to indicate Normal mode (mode=0) or sniff mode (mode=2)	Received on the side that has not issued the sniff command (AT+JSNF)	Dev A Dev B
+RRSI	Role Switch Indication	Received when the remote device performs a Master/Slave role switch	Dev A Dev B

4.4 Crystal Auto Calibration (Frequency counter method)

The frequency of the eBMU's crystal oscillator can be calibrated with a built in capacitance array that is configured by the BD_DATA value Osc_trim.

Instrument required is a frequency counter, for example Agilent 53131A universal counter. This method also requires that one of the GPIO pins P01 or P08 is available.

The calibration will be done on a 32 MHz clock generated from the 26 MHz internal reference clock. This 32 MHz clock should be adjusted to be within ± 2 ppm (± 64 Hz) accuracy.

The command requires that the device is in production mode, please follow the sequence below:

1. Connect the frequency counter to the appropriate test point
2. Enter Production Mode AT+JPRO=1
3. Use The AT+JCAC command to define the test point and the Osc_trim value
4. Measure the frequency of the 32 MHz signal with the counter
5. Iterate steps 3 and 4 until ± 2 ppm are reached. Then write the corresponding trim value to the parameter Osc_trim in the BD-data with AT+JCBD
6. Leave production mode AT+JPRO=0
7. Perform a SW reset

4.5 Low Power Mode Control

The low power mode protocol for eBMU is based on hardware signaling only. No SPP commands or responses are required. The existing flow control signals for the UART are used together with two GPIOs to tell the other device (host or controller) when it may enter low power mode, when it should wake up and when it cannot transmit because the first device is in low power mode.

Low power mode can be used when the device is in:

- Page-/inquiry-scan mode
- Connected with link in sniff
- Disconnected

In all other states low power mode shall not be used, a few examples of other states are listed below:

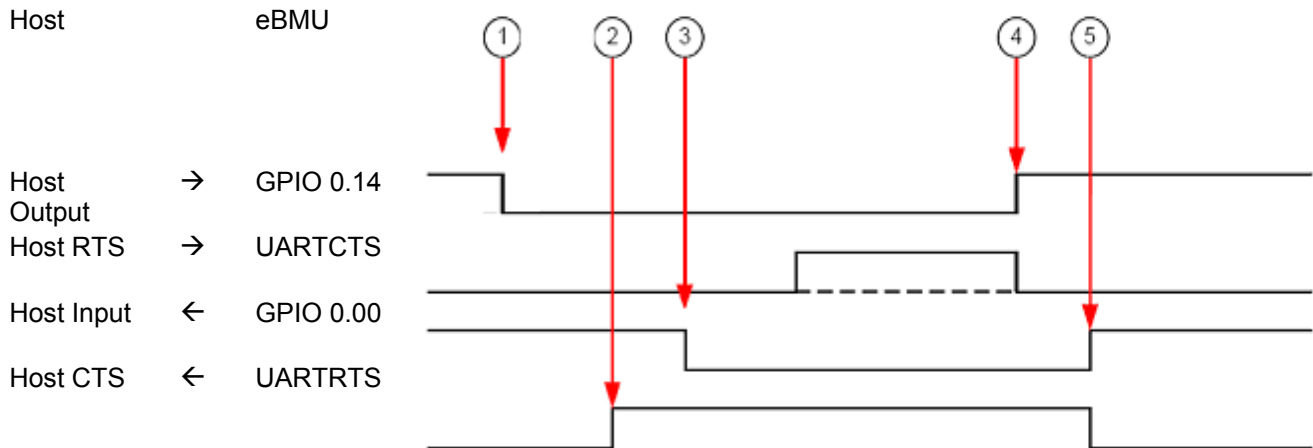
- During connection set-up
- During device discovery
- ACL link without sniff

To allow the eBMU to enter low power mode, the host sets PIN P0.14 low. When eBMU is ready, it will also allow the host to enter LPM by setting P0.0 low. Before entering LPM, the host shall set UART CTS of eBMU high. Before entering LPM, eBMU will set its own UART RTS high.

The host can wake up eBMU by setting UART CTS of eBMU low again and setting P0.14 high again. eBMU can wake up the host by setting its own UART RTS low again and setting P0.0 high again.

4.5.1 Host Initiates Low Power Mode Entry and Exit

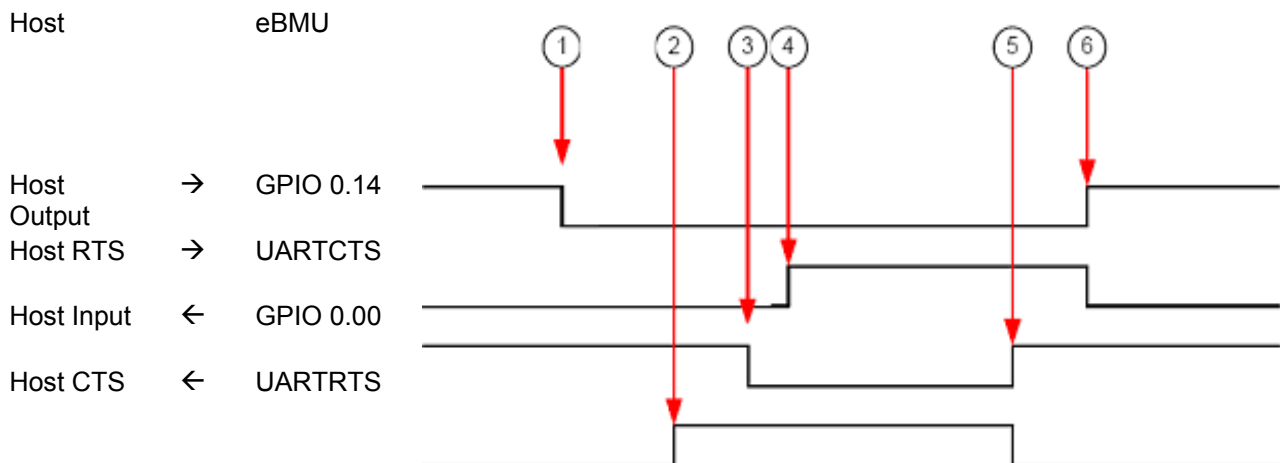
The picture below describes when the host initiates Low Power Mode and host initiates the Low Power Mode.



1. The host allows eBMU to enter low power mode
2. eBMU enters low power mode
3. eBMU allows the host to enter low power mode, the host may, if it can, enter low power mode
4. The host requests the eBMU to wake up
5. eBMU wakes up

4.5.2 Host Initiates Low Power Mode Entry, eBMU Initiates Exit

The picture below describes when the host initiates Low Power Mode and host initiates the Low Power Mode.



1. The host allows eBMU to enter low power mode
2. eBMU enters low power mode
3. eBMU allows the host to enter low power mode
4. The host enters low power mode
5. eBMU requests the host to wake up
6. The host wakes up

4.6 UART Baud Rate Change

The baud rate of the UART can be changed by writing into the UART_Baudrate parameter of the BD_DATA.

The available UART Baud rates are:

UART Baud Rate (bauds/s)	Configuration value for BD_DATA
9600	0
19200	1
38400	2
57600	3
115200	4
230400	5
460800	6
921600	7
1843200	8
3250000	9

Procedure to change UART Baud Rate

1. Enter into production mode:
AT+JPRO=1
2. Received confirmation from eBMU:
OK
3. Write BD_Data command and specify the UART Baud Rate in the designated parameter according to the values provided above.
4. Exit production mode:
AT+JPRO=0
5. Received confirmation from eBMU:
OK
6. SW Reset

Note: Original BD_Address and Osc_trim values need to be preserved. Read both with AT+JRBD command before changing the UART baud rate.

4.7 Data Flow Control

eBMU acknowledges a data packet sent by send data command (AT+JSDA) with an "OK" response when it has been transmitted. The host shall wait for the acknowledgement before sending a new packet.

4.8 Production Mode

Production mode is used for configuration and test purposes, the production mode shall be entered in order to execute the following commands:

AT+JDOI: DOWnload EEPROM Image

AT+JEDT: Enable Device under Test

AT+JCAC: Crystal Auto Calibration

AT+JCBD: Change BD_Data

General procedure to use production mode related commands:

1. HW reset
2. Enter production mode (AT+JPRO=1)
3. Execute command (e.g. "AT+JDOI")
4. Exit production mode (AT+JPRO=0)
5. HW reset

4.9 SPP-AT Commands Operating with EEPROM

The following commands are related to operations (read/write) in the EEPROM:

1. AT+JDOI (Download Image): It writes the whole content of the EEPROM.
2. AT+JCBD (change BD_DATA): It writes the BD_DATA section of the EEPROM.
3. AT+JRTD (Remove Trusted Device). It erases the associated BD_ADDRESS and link-key from the EEPROM.
4. AT+JRLS (Register Local Service): It writes the CoD value into EEPROM.
5. AT+JCCR (Create connection with security mode 3): The resulting link key will be written into EEPROM (both in DEVA and DEVB).
6. AT+JRES (RESet): The content of EEPROM is read and loaded into RAM.
7. AT+JSLN (Set Local Name): The local name is written into EEPROM.

Note: AT+JRBD does not read from EEPROM, but from the configuration that has been read into the device after startup.

4.10 Security Mode

Pairing is done automatically when security mode 3 is used.

Security mode 3 is the default security mode. If security mode 1 (no authentication and no encryption) is the desired mode, then AT+JSEC should be the first command issued specifying security mode 1 after a HW reset.

The pair procedure involves a PIN code indication +RPCI and a secure link established response +RSLE.

If a trusted device shall be removed, then the remove trusted device command (AT+JRTD) shall be used.

See example in section 5 for a detailed session description.

5 Example AT Commands and Responses

Example: Device Discovery

Dev A			DevB		
Seq No.	Direction	Command /response	Seq No.	Direction	Command /response
1.	Host<-eBMU	ROK			
			2.	Host<-eBMU	ROK
			3.	Host->eBMU	AT+JDIS=3
			4.	Host<-eBMU	OK
			5.	Host->eBMU	AT+JRLS=1101,11,Serial port,01,000000
			6.	Host<-eBMU	OK
7.	Host->eBMU	AT+JDDS=0			
8.	Host<-eBMU	OK			
9.	Host<-eBMU	+RDDSRES=0003199E8B25,devB,000000			
10.	Host<-eBMU	+RDDSRES=001842E869D5,Javier N95,5A020C			
10.	Host<-eBMU	+RDDSCNF=0			

Example: Service Discovery

Dev A			DevB		
Seq No.	Direction	Command /response	Seq No.	Direction	Command /response
1.	Host<-eBMU	ROK			
			2.	Host<-eBMU	ROK
			3.	Host->eBMU	AT+JSEC=1,1,1,04,1111
			4.	Host<-eBMU	OK
			5.	Host->eBMU	AT+JDIS=3
			6.	Host<-eBMU	OK
			7.	Host->eBMU	AT+JRLS=1101,11,Serial port,01,000000
			8.	Host<-eBMU	OK
9.	Host->eBMU	AT+JSEC=1,1,1,04,1111			
10.	Host<-eBMU	OK			
11.	Host->eBMU	AT+JSDS=0003199E8B25,1101			

Example AT Commands and Responses

12.	Host<-eBMU	OK			
13.	Host<-eBMU	+RSDSRES=Serial port,01			
14.	Host<-eBMU	+RSDSCNF=0			

Example: Connect with security mode 1 (No PIN request)

DevA			DevB		
Security mode 1			Security mode 1		
1.	Host<-eBMU	ROK			
			2.	Host<-eBMU	ROK
			3.	Host->eBMU	AT+JSEC=1,1,1,04,1111
			4.	Host<-eBMU	OK
			5..	Host->eBMU	AT+JDIS=3
			6..	Host<-eBMU	OK
			7..	Host->eBMU	AT+JRLS=1101, 11, Serial Port, 01, 000000
			8..	Host<-eBMU	OK
			9.	Host->eBMU	AT+JAAC=1
			10.	Host<-eBMU	OK
11.	Host->eBMU	AT+JSEC=1,1,1,04,1111			
12.	Host<-eBMU	OK			
13.	Host->eBMU	AT+JCCR=0003199E8B25,0 1			
14.	Host<-eBMU	OK			
15.	Host<-eBMU	+RCCRCNF=230,0			
			16.	Host<-eBMU	+RCOI=0003199E8B35
			17.	Host<-eBMU	+RCCRCNF=230,0

Example: Connect with security mode 3 (With PIN request)

DevA			DevB		
Security mode 3 (default)			Security mode 3 (default)		
1.	Host<-eBMU	ROK			
			2.	Host<-eBMU	ROK
			3.	Host->eBMU	AT+JDIS=3
			4.	Host<-eBMU	OK
			5.	Host->eBMU	AT+JRLS=1101,11,Serial port,01, 000000
			6.	Host<-eBMU	OK

Example AT Commands and Responses

			7.	Host->eBMU	AT+JAAC=1
			8.	Host<-eBMU	OK
9.	Host->eBMU	AT+JCCR= 0003199E8B25,01			
10.	Host<-eBMU	OK			
11.	Host<-eBMU	+RPCI=0003199E8B25			
12.	Host->eBMU	AT+JPCR=04,0000			
13.	Host<-eBMU	OK			
			14.	Host<-eBMU	+RPCI=0003199E8B35
			15.	Host->eBMU	AT+JPCR=04,0000
			16.	Host<-eBMU	OK
			17.	Host<-eBMU	+RSLE
			18.	Host<-eBMU	+RCOI=0003199E8B35
			19.	Host<-eBMU	+RCCRCNF=230,0
20.	Host<-eBMU	+RSLE			
21.	Host<-eBMU	+RCCRCNF=230,0			

References

- [1] www.bluetooth.org

Terminology

A

APPL	Application
AT	Attention (from Hayes command set)

B

BT	Bluetooth
----	-----------

C

CR	Carry Return
CTS	Clear To Send

D

DEC	DECimal
-----	---------

E

eBMU	embedded BMU
------	--------------

G

GPIO	General Purpose Input Output
------	------------------------------

H

HEX	HEXadecimal
HW	Hardware

I

I/O	Input/Output
I2C	Inter-Integrated Circuit

L

LF	Line Feed
----	-----------

M

MSB	Most Significant Bit
MTU	Maximum Transmission Unit

O

OSC	OSCillator
-----	------------

P

PU	Pull Up
----	---------

R

RFCOMM	Radio Frequency Communication
RTS	Request To Send
RX	Reception

S

SCL	Serial CLock
-----	--------------

SDA	Serial DAta
SPP	Serial Port Profile
SW	Software
T	
TX	Transmission
U	
UART	Universal Asynchronous Receiver Transmitter
Z	
Z	Tri-state

<http://www.infineon.com>