eUniStone SPP-AT Application SW 3.1

eUniStone, User's Manual Software Specification

Revision 0.2, 25-Jan-2013

Intel Public

Document Number: eUniStone_V1 00_UM_SD

The template (Word) of this document has been formally released by DOC department (IMC-DOC@intel.com).
Template data: T_TechDoc.dotm, Rev. 2.2, 06-Aug-2012

Revision History

Revision Number	Description	Revision Date
0.2	This SPP-AT specification update is based on the SW3.1 used on the eUniStone module PBA31309 at final verification. Updates to SPP-AT commands, SPP examples and BD-Data were done.	25-Jan-2013
0.1	Specification for the SPP-AT application with Bluetooth v2.1 features in the eUniStone chip PMB8754 which is used on the eUniStone module PBA31309. This specification is using AT commands similar to those of the predecessor chip PMB8753/2 on module PBA31308/2. This document is issued before final verification of the device as a preliminary version.	10-Oct-2012

Contents

Figures

No table of figures entries found.

Tables

1 Introduction

This document describes the AT commands applicable to Intel's Bluetooth chip PMB8754 (eBMU) which is used on the module PBA31309 (eUniStone). This product implements the Serial Port Profile according to Bluetooth Core Specification v2.1+EDR. Both initiating role (device A) and accepting role (device B) are supported. The device supports a single point-to-point connection. Up to 3 different services can be registered in order to be visible as 3 different types of Bluetooth accessory (e.g. for Notebooks, Android phones and Apple(TM) phones). The device can store the link authentication keys of up to 5 paired devices.

2 Bluetooth Features

- Bluetooth v2.1 + EDR compliant SPP implementation
  - Secure Simple Pairing, Security Mode 4
    - Association Models „Numeric Comparison", „Just Works" and „Passkey Entry" are supported
  - Encryption Pause Resume
  - Enhanced Power Control (BT3.0 feature of the BT Controller)
- Device A (initiating) and device B (accepting) role
- One point-to-point link for data transmission
  - octet by octet in stream mode
  - by packets in command mode (MTU size 500 bytes)
- Device is visible and connectable until the link has been set up Sniff and Sniff Sub Rating are supported on the link to save current
- Up to five trusted devices can be stored in EEPROM when 6th device is paired, the first device is deleted
- AT commands for development and manufacturing
  - Device Under test Mode for connection to a BT tester
  - Secure Simple Paring Debug mode to sniff and decrypt the air traffic
  - Crystal oscillator calibration
  - EEPROM configuration update
  - SW upgrade via UART and I2C
- UART with HW flow control (RTS/CTS)
  - Use of HW flow control is mandatory
  - UART baud rate may be changed in EEPROM configuration 9.6kbps to 3.25Mbps

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 PBA 31309 (eUniStone) throughout the document: Command Mode and Stream Mode.

For Bluetooth SPP, two different roles are specified:

- Device A (Dev A) initiating the over the air connection
- Device B (Dev B), which accepts the connection

3.1.1 Command Mode

In this mode the SPP application running on the eUniStone will execute the AT commands sent from the host over the UART. In this mode, the host application can send data packets to the eUniStone, which are transmitted to the remote device that has a Bluetooth connection on SPP level with the eUniStone. 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 eUniStone, which are transmitted over the air to the remote device. This mode is normally used when transmitting small amount of data in a random way and for serial cable replacement applications.

Command Mode typically yields higher throughput than stream mode, because the filling of air packets can be optimized.

3.2 Pin Assignments

The table below shows the available GPIOs. GPIOs noted with "(Reserved)" cannot 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.

Table 1. Pin Assignments.

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/OD	P0.12 (Reserved)	I2C
SCL	O PU	I/O/OD	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 CAL
PCMIN	Z	I/O	P0.2	APPL GPIO
PCMOUT	Z	I/O	P0.3	APPL GPIO
PAON	Z	I/O	P0.8	APPL GPIO / XTAL CAL
PSEL0	Z	I/O	P0.9	APPL GPIO
PSEL1	Z	I/O	P0.10	APPL GPIO
TX_Conf1	Z	I/O	P0.11	APPL GPIO
P015	Z	I/O	P0.15	APPL GPIO

For P0.12 and P0.13 it is possible to connect to host or external tool for download of .eep file to EEPROM. Any such use of I2C bus shall be done while onboard chip is in reset or not being active on the I2C bus. For module PBA31309 the SDA and SCL have an internal 2.5V pull-up.

4 Intel SPP-AT Command and Response

AT commands can only be sent while in command mode. The escape sequence, stream connection cancel (^^^), 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 eUniStone 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. eUniStone expects only upper case characters in command mode.

Example for values with decimal base in ASCII format

The number 255 in decimal corresponds to three characters '2', '5', '5'.

E.g. AT+JSDA=010,1234567890. Here the number 10 must be given by three ASCII characters '0', '1', '0', because the parameter requires a value consisting of three characters.

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 0x0010C64D67DC, 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

Three error messages may be received ERR=-1, ERR=-2 and ERR=-3.

- ERR = -1; Syntax Error
- ERR = -2; Command not allowed at present execution status
  - ERR=-2 is obtained when any of the following situations are met:
    - Create a new connection when already connected.
    - Device discovery when already connected.
    - Send data when not connected.
    - Accept connection request without a request.
    - Enter sniff or sniff sub rating mode if already ON or without being connected.
    - Enable SEC mode after another command has previously been sent.
    - Service discovery when connected.
    - Send data with length = 0.
    - Register more than three services
    - Disconnect when not connected
    - Reset during EEPROM write
    - Enable Device Under Test mode without being in Production mode
- ERR = -3; if internal unknown error occurs in protocol stack

4.2 AT Command and Response List Table

Table 2. AT Command and Response List Table.

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
AT+JAAC=<***auto_accept***>	***Auto Accept Connection requests*** Forces eUniStone to accept connection requests. ***auto_accept*** parameter (1 character): 0 Host will be notified on incoming connection indication (**+RCOI**) – (no auto accept). This is the default configuration. 1 eUniStone will automatically auto accept incoming connection request – (host will be notified but connection is accepted automatically)	**OK**	Dev B
AT+JACR=<***accept***>	***Accept Connection Request*** Shall be used as answer to a connect indication (*+RCOI*). ***accept*** parameter (1 character): 0 Not accepted 1 Accepted	**OK**	Dev B
AT+JCAC=<***trim_value***>, <***GPIO*** >	***Crystal Auto Calibrate*** ***trim_value*** parameter (4 characters / HEX base): The trim value is used to adjust the frequency on the GPIO chosen by the GPIO parameter. Range: 0x0000 – 0x03FF ***GPIO*** parameter (4 characters / HEX base): Two GPIOs can be used as output for the oscillator trim, either P01 or P08 (0002 and 0100). See section 4.4. **AT+JCAC** command can only be issued after production mode is enabled (**AT+JPRO=1**). See section 4.8.	**OK**	Dev A Dev B
AT+JCBD=<***bd_data***>	***Change BD_Data*** ***bd_data parameter*** (116 characters / HEX base / LSB first): The bd_data string	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	consists of following configuration data (with reference value): bdAddr (12 characters) = see Note 1 channelWordOffset (4 characters) = 0000 clkConf (2 characters) = 8A EEPROMSize (2 characters) = 80 inputFreq (8 characters) =018CBA80 lmpFeatures (16 characters) = 87591F987E0602BF lpmConf (2 characters) = 40 lpmDrift (2 characters) = FA lpmThreshold (2 characters) = 12 ulpmThreshold (2 characters) = 18 pmuConfig (4 characters) = 0080 rfPselD (8 characters) = 06050403 rfPselConf (2 characters) = 44 rssiMin (2 characters) = 0C rssiMax (2 characters) = 10 ddcTlConf (2 characters) = 02 uartBaudrate (2 characters) = 04 uartInvert (2 characters) = 00 uartPulls (2 characters) = 01 oscSettle (2 characters) = 10 bbConf (2 characters) = 24 rfConf (2 characters) = 04 txPowerRef0 (2 characters) = F2 txPowerRef1 (2 characters) = F8 txPowerRef2 (2 characters) = FE txPowerRef3 (2 characters) = 04 oscTrim (4 characters) = see Note 2 threeWireArqTimeout (2 characters) = 06 lmpVersion (2 characters) = 00 Three-Wire_LinkMsg_Time=00 Three-Wire_LPM_Time=00 reserved (12 characters) = 000000000000 Note: 1. The bdAddr is pre-programmed in PBA31309 module. It can be read through AT+JRBD. Don't overwrite the pre-programmed value accidentally. 2. The oscTrim value is pre-programmed in PBA31309 module. It can be read through AT+JRBD. Writing an incorrect value can change the RF functionality for the module.		

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	3. AT+JCBD command can only be issued after production mode is enabled (AT+JPRO=1). Changes in BD Data take place after production mode command is disabled (AT+JPRO=0) followed by a SW reset (AT+JRES). See sections 4.8 and 4.9 .		
AT+JCCR=<***bd_addr***>, <***service channel***>	***Create Connection Request*** Instructs eUniStone to connect to a remote Bluetooth device (prospective slave). ***bd_addr*** parameter (12 characters / HEX base): The Bluetooth address of the remote device ***service_channel*** parameter (2 characters / DEC base): Value range: 01-30 Which service channel to connect to can be received from a Service Discovery **AT+JSDS**.	**OK** (Followed by): **+RCCRCNF** = <***MTU_size***>, <***service***>, <***status***> ***MTU_size*** *parameter (3 characters/DEC base)* ***service*** *parameter, (4 or 32 characters/Dec base)* *Indicating which service the remote device is connected to.* The host of Dev A specifies which service it connects to in the **AT+JCCR** command and if needed it should be stored since *Dev A, service is set to 0000.* ***status*** *parameter (1 character):* If maximum number of allowed connections already exists: ERR=-2.	Dev A
AT+JDDS=<***Extended_Inquiry***>	***Device Discovery Start*** Causes eUniStone to start a Device Discovery (Inquiry and Remote Name Request) of the Bluetooth neighborhood. ***Extended Inquiry*** (1 characters /Dec base) 0 No remote name needed: Only information from EIR (Extended Inquiry Response) is used no RNR (Remote Name Request) is performed, all found addresses and the CoD are presented. 1 Shortened name requested: Shortened name is used if it is available. If it is not available RNR is performed. 2 Full name requested: RNR is performed for devices that do not respond with full name in EIR.	**OK** Then, if responses are returned: **+RDDSRES**=< ***bd_addr*** >, <***remote_name***>,<***COD***> For each response ***bd_addr*** parameter (12 characters / HEX base). ***remote_name*** parameter (variable length): Name of the remote device. If Extended Inquiry=0 then remote name will be empty if no name is available. If **Extended Inquiry**=1 or 2 **remote name** will be page timeout if the RNR does not find any devices answering. ***COD*** parameter (6	Dev A

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	Note 1: eUniStone will answer with the friendly name and the registered service(s) 2: The Responses are limited to maximum eight responses	characters / HEX base): Class of device (Completed by): **+RDDSCNF=<*status*>** ***status*** parameter (1 character):	
AT+JDIS=<***discoverable***>	***DIScoverable*** Forces eUniStone into Page Scan / Inquiry Scan states indefinitely (note: this makes the device discoverable). ***discoverable*** parameter (1 character): 0 No scans enabled. 1 Inquiry Scan enabled (visible). 2 Page Scan enabled (connectable). 3 Inquiry & Page Scan enabled (visible & connectable). Scan is automatically disabled when connected and at disconnection they are automatically enabled. Default configuration after a HW or SW reset is no scans enabled. Note: In a device with no service registered and discoverable, it will not accept any incoming SPP connection request.	**OK**	Dev B Dev A
AT+JDOI	***Download Application Image via UART*** After "OK" response a binary file with the EEPROM image may be sent. File open and send in binary format after "OK" is received. **AT+JDOI** command can only be issued after production mode is enabled (**AT+JPRO=1**). Changes in the EEPROM image are effective after production mode command is disabled (**AT+JPRO=0**) followed by a HW reset. See section 4.8.	**OK** after command is sent. **+RDOICNF** after .eep file has been written. eUniStone 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.	Dev A Dev B
AT+JEDT	***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: 1. Set Event Filter – allow all	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	connections. 2. Write Scan Enable – page and Inquiry. 3. Enable device under test. Device under test should be used without any security, because the RF tester cannot use authentication. **AT+JPRO=1** shall be the first command that is issued after startup. See section 4.8.		
AT+JGPA=<***reserved***>, <***read***>, <***set***>,<***clear***>	***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 P0.11 the bit field value is 0800. ***reserved*** parameter (4 characters / HEX base): shall be 0000 ***read*** parameter (4 characters / Hex base): Values for each bit: 0 No Action 1 Read ***set*** parameter (4 characters / HEX base): Values for each bit: 0 No Action 1 Set ***clear*** parameter (4 characters / HEX base): Values for each bit: 0 No Action 1 Clear	<***value***> ***value*** parameter (4 characters / HEX base): This value is the state of the GPIO PINs specified in read parameter. Values for each bit: 0 Low 1 High Note: When a pin is set as output the return value will be 0 for the specific pin. Set and clear may only be used on output pins. Reserved pins like e.g. UART pins will always return 0.	Dev A Dev B
AT+JGPC= <***direction***>, <***open_drain***>, <***pull_on/off***>, <***pull_up/down***>, <***tristate***>	***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**). ***direction*** parameter (4 characters / HEX base): Values for each bit: 0 OUT 1 IN ***open_drain*** parameter (4 characters / HEX base): Values for each bit: 1 OPEN ***pull_on/off*** parameter (4 characters	**OK**	D Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	/ HEX base): Values for each bit: 0 Pull OFF 1 Pull ON ***pull_up/down*** parameter (4 characters / HEX base): Value for each bit: 1 Pull UP 0 Pull DOWN ***tristate*** parameter (4 characters / HEX base): Value for each bit: 1 Tri-state See section 3.2 for available GPIO pins.		
AT+JPCR=<***length_PIN_code***>, <***PIN_code***>	***PIN Code Reply*** Sent to eUniStone in response to a PIN Code Request from a remote Bluetooth device (bd_addr). ***length_PIN_code*** parameter (2 characters / DEC base): Value range: 01-16 Length of PIN code ***PIN_code*** parameter (length= length_PIN_code): The PIN code to be sent to the eUniStone device, e.g. **AT+JPCR**=04,1234. The PIN code is an ACII string. Note: In the Bluetooth v2.1+EDR specification the wording is Passkey instead of PIN for SSP.	**OK** (Followed by): **+RSLE** if secure link is established	Dev A Dev B
AT+JPRO=<**mode**>	***PROduction mode*** If device under test shall be used without any security, **AT+JPRO** shall be the first command that is issued after startup. **AT+JPRO** is also used to return to normal mode after use of production mode commands. ***mode*** parameter (1 character): 0 Production mode OFF 1 Production mode ON See section 4.8 for commands requiring production mode.	**OK**	Dev A Dev B
AT+JRBD	***Read Bluetooth device Data*** Sent to eUniStone to retrieve the Bluetooth device Data	**+RRBDRES** bd_data parameter (116 characters / HEX base / LSB first): The bd_data	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
		string consists of following configuration data (with reference value): bdAddr (12 characters) = see Note 1 channelWordOffset (4 characters) = 0000 clkConf (2 characters) = 8A EEPROMSize (2 characters) = 80 inputFreq (8 characters) =018CBA80 lmpFeatures (16 characters) = 87591F987E0602BF lpmConf (2 characters) = 40 lpmDrift (2 characters) = FA lpmThreshold (2 characters) = 12 ulpmThreshold (2 characters) = 18 pmuConfig (4 characters) = 0080 rfPselD (8 characters) = 06050403 rfPselConf (2 characters) = 44 rssiMin (2 characters) = 0C rssiMax (2 characters) = 10 ddcTlConf (2 characters) = 02 uartBaudrate (2 characters) = 04 uartInvert (2 characters) = 00 uartPulls (2 characters) = 01 oscSettle (2 characters) = 10 bbConf (2 characters) = 24 rfConf (2 characters) = 04 txPowerRef0 (2 characters) = F2 txPowerRef1 (2 characters) = F8 txPowerRef2 (2 characters) = FE txPowerRef3 (2 characters)	

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
		= 04 oscTrim (4 characters) = see Note 2 threeWireArqTimeout (2 characters) = 06 ImpVersion (2 characters) = 00 Three-Wire_LinkMsg_Time=00 Three-Wire_LPM_Time=00 reserved (12 characters) = 000000000000 Notes: 1. The bdAddr is pre-programmed in PBA31309 module. It can be read through **AT+JRBD**. Don't overwrite the pre-programmed value accidentally. 2. The oscTrim value is pre-programmed on PBA31309 module. It can be read through **AT+JRBD**. Writing an incorrect value can make the module inoperable. 3. **AT+JCBD** command can only be issued after production mode is enabled (**AT+JPRO**=1). Changes in BD Data take place after production mode command is disabled (**AT+JPRO**=0) followed by a SW reset (**AT+JRES**),	
AT+JRES	***RESet*** SW reset of the system.	**ROK** eUniStone with EEPROM software is restarted.	Dev A Dev B
AT+JRLS=<***length_uuid***>, <***length_service_name*** >, <***uuid***>, <***service_name***>, <***service_channel*** >, <***CoD***>	***Register Local Service*** Up to three services may be registered. ***length_uuid***: (2 characters / DEC base) Length for the uuid 4 or 32 ***length_service_name*** parameter (2 characters / DEC base): Value range: 01-16 Length of service name ***uuid*** parameter:	**OK**	Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	uuid for supported profile 4 or 32 characters / HEX base e.g.1101 for Serial Port Profile e.g. 00011010000100080000080 5F9B34FB for Serial Port Profile ***service_name*** parameter (length= length_service_name): Name for the service, no final delimiter is needed ***Service_channel*** parameter (2 characters /DEC base): Value range: 01-30. This is the service channel number seen by remote side. ***CoD*** parameter (6 characters / HEX base): Class of device. The default CoD value is 000000 Notes: 1. A Dev B may register up to three services. 2. Once the command is issued the service is registered even though an identical service already has been registered. 3. Services need to be re-registered after a SW or HW reset.		
AT+JRRI	***Read Revision Information***	<***revision***> ***revision*** parameter (2 characters / HEX base) *Note:* *SPP-AT SW revision 3x*	Dev A Dev B
AT+JRSD	***Read Stored Devices***	**OK** (Followed by): **+RRSDCNF**=<***no***>, <***bd_addr_1***> , to <***bd_addr_no***> for all stored device with a maximum of five devices. ***no*** parameter (1 character/Dec base) The number of stored devices, ***bd_addr_1*** is the oldest. In case no=0 then no devices are stored. ***bd_addr*** parameter (12 character/Hex base) BD Address of the stored device	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
AT+JRTD=<***bd_addr***>	***Remove Trusted Device*** Delete the trusted device information for a registered device (***bd_addr***). ***bd_addr*** parameter (12 characters / HEX base): The Bluetooth address of the device that shall be removed from the list. <***bd_addr***>=000000000000: Delete all trusted devices.	**OK**	Dev A Dev B
AT+JSBR=<***baud_rate***>	***Set Baud Rate*** Set the UART baud rate temporarily (until next HW or SW reset). The host shall wait for the OK response before changing its baud rate. *<**baud_rate**> parameter (7 characters/DEC base) is the wanted Baud rate.* E.g. AT+JSBR=0009600 is 9.6kbps.The flow stop is high during the change and when flow stop is released the new UART speed shall be used. Typically 50ms to change baud rate. Note All baud rates from Table 6. Available UART baud rates.are supported. For other baudrates please ask your technical support.	**OK**	Dev A Dev B
AT+JSCR	***Stream Connection Request*** May only be used when a SPP connection is established. 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 characters / DEC base): number of bytes to be sent Value range: 001 to max MTU_Size ***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***). The value for this parameter is negotiated by the two devices during connection set	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	up.		
AT+JSDR	***SPP Disconnect Request*** Forces a SPP disconnection.	**OK**	Dev A Dev B
AT+JSDS= <***bd_addr***>, <***length_uuid***>, <***uuid***>	***Service Discovery Start*** Causes eUniStone to start a service discovery of device with bd_addr and search for services defined by uuid. ***bd_addr*** parameter (12 characters / HEX base): BD Address of remote device ***length_uuid*** (2 characters /Dec base) Length of uuid, 04 or 32 ***uuid*** parameter (04/32 characters / HEX base): Service to search for e.g. 1101 or 00001101000010008000000805F9B34FB for Serial Port Profile Short uuid is 4 characters while long uuid is 32 characters.	**OK** (Then, if services are returned): ***+RSDSRES=*** <***remote_service_name***>, <***remote_service_channel***>, ***remote_service_name*** parameter (variable length): Name of the remote service. ***remote_service_channel*** parameter (2 characters/ DEC base) For each service available. (Completed by): **+RSDSCNF**= <***status***> status parameter (1 character)	Dev A
AT+JSEC= <***security_mode***>, <***PIN_type***>, <***length_PIN_code***>, <***PIN_code***>, <***Input_capability***>, <***Output_ capability***>	***Enable Security*** ***security_mode*** parameter (1 character): 1 N/A 2 N/A 3 N/A 4 Security Mode 4 ***PIN_type*** parameter (1 character) 1 Variable PIN (default) 2 Fixed PIN ***length_PIN_code*** parameter (2 characters / DEC base): Length of PIN code. The maximum PIN length value is 16 (corresponding to a 16 characters long PIN code) ***PIN_code*** parameter (length=length_PIN_code): Normal user PIN, for example "0000"., This parameter is taken into account if **PIN_type** is fixed.	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	Input_capability parameter (1 character/Dec) 0 Input None 1 Input Yes/No 2 Input Keyboard ***Output_capability*** parameter (1 charachter/Dec) 0 Output None 1 Output Yes/No **AT+JSEC** shall be the first command that is sent after a SW reset except if Production Mode commands like „Enable Device Under Test" are required. In that case the **AT+JPRO** shall be the first command after startup and no **AT+JSEC** shall be sent.. Note: Pairing with remote legacy devices: • Fixed PIN is only used if remote devices have security mode 3 enabled • Fixed PIN needs to be enabled with the **AT+JSEC** command (default is variable PIN) • If both devices have enabled fixed PIN and pairing is initiated (if remote device has security mode 3 enabled) the paring will fail (as described in Bluetooth Core specification v2.1+EDR [1]. • The PIN code that is used for auto accept connection is the one given in the **AT+JSEC** command but only if fixed PIN is configured. • To provide a PIN with the **AT+JSEC** command has no effect if variable PIN is used. • To auto accept a connection when Security mode 3 is used by remote device, fixed PIN needs to be enabled. • There is no default fixed PIN (variable PIN is default), the used PIN code is the one given in the **AT+JSEC** command if fixed PIN is used. „Out Of Band (OOB) Association Model" for SSP is not supported. The Link keys are stored in EEPROM. The security level information, if		

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	needed, may be stored by the host.		
AT+JSPD= <***debug_mode***>	***Secure Simple Pairing Debug Mode:*** Activates SSP debug Mode ***debug_mode*** (1 character/DEC base) 0 Disabled 1 Enabled Note: This AT command will allow sniffing of encrypted links with a Bluetooth sniffer, if the Secure Simple Pairing procedure is used by both devices.	**OK**	Dev A Dev B
AT+JSKN=<***Notification_Type***>	***Send Keypress Notification*** ***Notification_Type*** (1 character/Dec base) 0 Passkey entry started 1 Passkey digit entered 2 Passkey digit erased 3 Passkey cleared 4 Passkey entry completed	**OK**	Dev A Dev B
AT+JSLN= <***length_friendly_name***>, <***friendly_name***>	***Set Local device friendly Name*** Supports all ASCII characters. ***length_friendly_name*** parameter (2 characters / DEC base): Length of friendly name, the maximum value for length is 18. ***friendly_name*** parameter (length=length_friendly_name): No delimiter is required. The friendly name will be used for EIR and RNR, the friendly name should be written at start-up or reset. The default friendly name after reset is, ”eUniStone SPP with BT2.1 features”.	**OK**	Dev A Dev B
AT+JSNF=<***sniff_Max***>, <***sniff_Min***>, <***sniff_attempt***>, <***sniff_tmo***>,<***on/off***>	***Sniff Request*** Request a link to enter Sniff Mode. All command parameters are given in HEX base with the exception of on/off parameter. ***sniff_Max*** parameter (4 characters / HEX base): Maximum allowed sniff interval Value to be written= N Time = N * 0.625 ms Range: 1.25 ms to 40.9 s	**+RSNFCNF**= <***sniff_Interval***>,<***mode***> ***sniff_Interval*** parameter (4 characters / HEX base): ***mode*** parameter (1 character): Normal mode (mode=0) or sniff mode (mode=2) If no connection exists: ERR=-2.	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	sniff_Min parameter (4 characters / HEX base): Minimum allowed sniff interval Value to be written= N Time = N * 0.625 ms Range: 1.25 ms to 40.9 s ***sniff_attempt*** parameter (4 characters / HEX base): Number of sniff attempts Value to be written= N Length = N* 1.25 ms Time Range: 0.625 ms - 40.9 s ***sniff_tmo*** parameter (4 characters / HEX base): The time out value for sniff attempts Value to be written= N Time = N * 0.625 ms Range: 0 msec to 40.9 s ***on//off*** parameter (1 character): Sniff ON (value=1) or Sniff OFF (value=0)		
AT+JSNS=<***Maximum_Latency***>, <***Maximum_Remote_timeout***>, <***Minimum_Local_Timeout***>	***Sniff Sub rating*** ***Maximum_Latency*** (4 characters / HEX base): The maximum allowed sniff sub rate of the remote device. ***Minimum_Remote_timeout*** (4 characters / HEX base): Minimum base sniff sub rate timeout that the remote device may use. ***Minimum_Local_Timeout*** (4 characters / HEX base): Minimum base sniff sub rate timeout that the local device may use. Note : 1. If the Host does not write the sniff sub rating parameters prior to sniff sub rating being initiated by the Link Manager the default values shall be used. 2. Setting all sub rate values to zero is equivalent to sniff mode without sub rating enabled.	**OK** If sniff is issued also +ESNS shall be received	Dev A Dev B
AT+JUCR=<***Status***>	***User Confirmation Reply Command***	**OK**	Dev A Dev B

AT Command	Usage	Resulting response(s) from eUniStone	Dev A / Dev B
	Status parameter (1 character DEC base): 0 Not accepted 1 Accepted		
^ ^ ^	***Stream Connection Cancel*** Exit Streaming Mode: Send 3 escape characters ^ ^ ^ with an initial wait time of T0 > 100 ms after the last transmitted byte and an interval of 100 ms < T1 < 1100 ms between the symbols. T0 - ^ -T1- ^ -T1- ^ This string is not terminated with <CR>< LF>.	**OK**	Dev A Dev B

4.3 AT Responses List (not command triggered)

Some „responses" are not actually responses to a command, but they are triggered by events on the Bluetooth link, like „Sniff Mode Confirmation" and „Disconnect Indication" or „Role Switch Indication". These events can also occur in stream mode. When that happens, the characters for the response will be transmitted to the host within the received data stream.

Example: data / command / data: 89 AB CD *AT+RSNFCNF=0000,0* EF 12 34 56

To avoid this, the following measures are needed:

- Start stream mode only after authentication is completed. Usually Role Switch doesn't occur at a later stage any more.
- Avoid using sniff mode with stream mode, or, if the peer application is known, make sure the remote device will not exit sniff mode during data transmission.
- Make use of the feature described in section 4.11. The host in stream mode can monitor P0.1 to detect the end of the stream data before link disconnection.

Table 3. AT Responses List (not command triggered).

AT Response	Usage	Command to acknowledge the response/indication	Dev A / Dev B
ROK	***Start up response***		Dev A Dev B
+RCCRCNF= <***MTU_size***>, <***service***>, <***status***>	***Connect Confirm:*** ***MTU_size*** parameter (3 characters / DEC base) ***service*** parameter,(4 or 32 characters/Dec base) shows which service the remote device (Dev A) is connected to. ***status*** parameter (1 character):		Dev A Dev B

AT Response	Usage	Command to acknowledge the response/indication	Dev A / Dev B
	0 no connection 1 connection Note: Dev A only receives a dummy Service (0000) and host should store information needed about devices and services.		
+RPCI=<***bd_addr***>, <***simple_pairing***>	***PIN Code Indication*** ***bd_addr*** parameter (12characters / HEX base): MSB is sent first. ***simple_pairing*** (1 characters / HEX base): 0 Legacy Pairing 1 Secure Simple Pairing	***AT+JPCR=<length_PIN_code>***, <***PIN_code***> ***length_PIN_code*** parameter (2 characters / DEC base): Length of PIN code ***PIN_code*** parameter: Up to 16 characters	Dev A Dev B
+RPNE= <***Numerical_Value***>	***Passkey Notification Event*** ***Numerical_Value*** parameter (6 characters / Dec base): Numeric value to be displayed. Valid values are; 000000 – 999999.		Dev A Dev B
+RUCE= <***Numerical_Value***>	***User Confirmation Event*** ***Numerical_Value*** parameter (6 characters / Dec base): Numeric value to be displayed. Valid values are; 000000 – 999999.		Dev A Dev B
+RKNI= <***Notification_Type***>	***Key press Notification Indication Event*** ***Notification Type*** parameter (1 character/Dec base) 0 Passkey entry started 1 Passkey digit entered 2 Passkey digit erased 3 Passkey cleared 4 Passkey entry completed		Dev A Dev B
+RCOI=<***bd_addr***>	***Connect Indication*** ***bd_addr*** parameter (12 characters / HEX base): See previous response for information on the format.	**AT+JACR**= <***accept***> ***accept*** parameter (1 character): 0 Not accepted 1 Accepted	Dev B
+RDAI=<***length***>, <***data***>	***Data Indication*** ***length*** parameter (3 characters / DEC base): Number of bytes to be sent		Dev A Dev B

AT Response	Usage	Command to acknowledge the response/indication	Dev A / Dev B
	data parameter (length=length from previous parameter): Received data Note: Not available during stream mode.		
+RDII	***Disconnect Indication*** Received on both sides. During stream mode, if the link is lost, the response **+RDII** is received after the link supervision time out of 20 seconds.		Dev A Dev B
+RSLE	***Secure Link Established***		Dev A Dev B
+RSNFCNF=<***sniff_Interval***>,<***mode***>	***Sniff mode confirmation*** ***sniff_Interval*** parameter (4 characters / HEX base). ***mode*** parameter (1 character): 0 Normal mode 1 sniff mode Note : The response is received in both devices.		Dev A Dev B
+ESNS <***Maximum_Transmit_Latency***>,<***Maximum_Receive_Latency***>,<***Maximum_Remote_timeout***>, <***Minimum_Local_Timeout***>, <***Status***>	***Sniff Sub rating event:*** ***Maximum_Transmit_Latency:*** Maximum latency for data being transmitted from the local device to the remote device. ***Maximum_Receive_Latency*** Maximum latency for data received by the local device from the remote device. ***Maximum_Remote_timeout*** The base sniff sub rate timeout in baseband slots that the remote device shall use. ***Minimum_Local_Timeout*** The base sniff sub rate timeout in baseband slots that the local device will use. ***Status***		Dev A Dev B

AT Response	Usage	Command to acknowledge the response/indication	Dev A / Dev B
+RRSW	***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 eUniStone module PBA31309 includes a 26MHz crystal which provides the reference clock for the eBMU chip PMB8754. The crystal oscillator is tuned during production of the module.

During development, it can happen that the oscillator trim value „oscTrim" in BD Data is lost, for example by an invalid „Change BD Data" command. For this reason, the crystal tuning procedure is described here.

The method requires a frequency counter[1] to be connected to either P0.1 or P0.8. After issuing a test command with the GPIO port and a trim_value, the chip will output a 32 MHz clock on the selected pin. The tester shall adjust the trim_value until it is within ±2ppm (±64 Hz) accuracy.

Finally, the trim_value must be programmed to the EEPROM with command „Change BD Data". The module specific BD Address must be known to use this command.

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. Power up the device
3. Enter Production Mode AT+JPRO=1
4. Read the module specific BD Address with AT+JRBD command
5. Use The AT+JCAC command to define the test point and the trim_value
6. Measure the frequency of the 32 MHz signal with the counter
7. Iterate steps 5 and 6 until ±2 ppm are reached.
8. Then write the corresponding trim_value to the parameter oscTrim in the BD-data with AT+JCBD. Use the BD Address that was read in step 4.
9. Leave production mode AT+JPRO=0
10. Perform a SW reset

[1] for example Agilent 53220A universal counter

4.5 Low Power Mode Control

The low power mode (LPM) protocol for eUniStone 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. The eUniStone informs when the host may enter low power mode, when the host should wake up and when the module cannot receive anything on the UART because it is in low power mode. The signaling is the same for the host to the controller.

The eUniStone will enter low power mode in the following modes:

- Disconnected and idle
- visible and / or connectable: during scan intervals
- connected with link in sniff and/or sniff sub rating mode: during sniff intervals

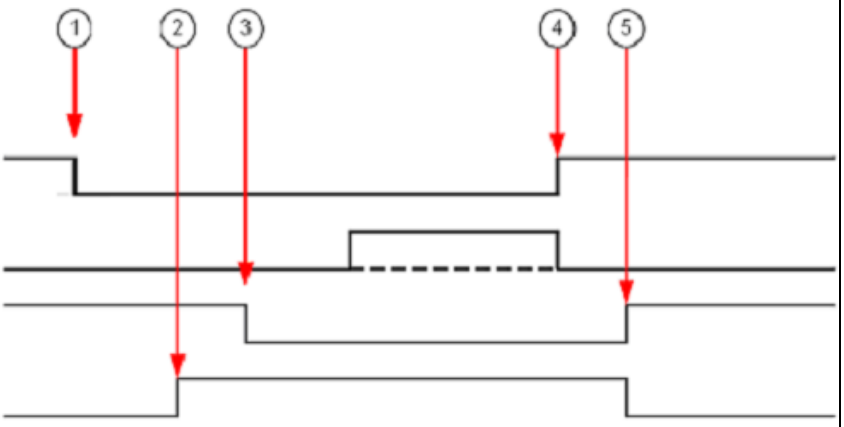
To allow the eUniStone to enter low power mode (LPM), the host sets PIN P0.14 low. When eUniStone is ready, it will also allow the host to enter LPM by setting P0.0 low. Before entering LPM, the host shall set UART CTS of eUniStone high. Before entering LPM, eUniStone will set its own UART RTS high.

The host can wake up eUniStone by setting UART CTS of eUniStone low again and setting P0.14 high again. eUniStone 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 leaving the Low Power Mode.

Table 4. Host Initiates Low Power Mode Entry and Exit.

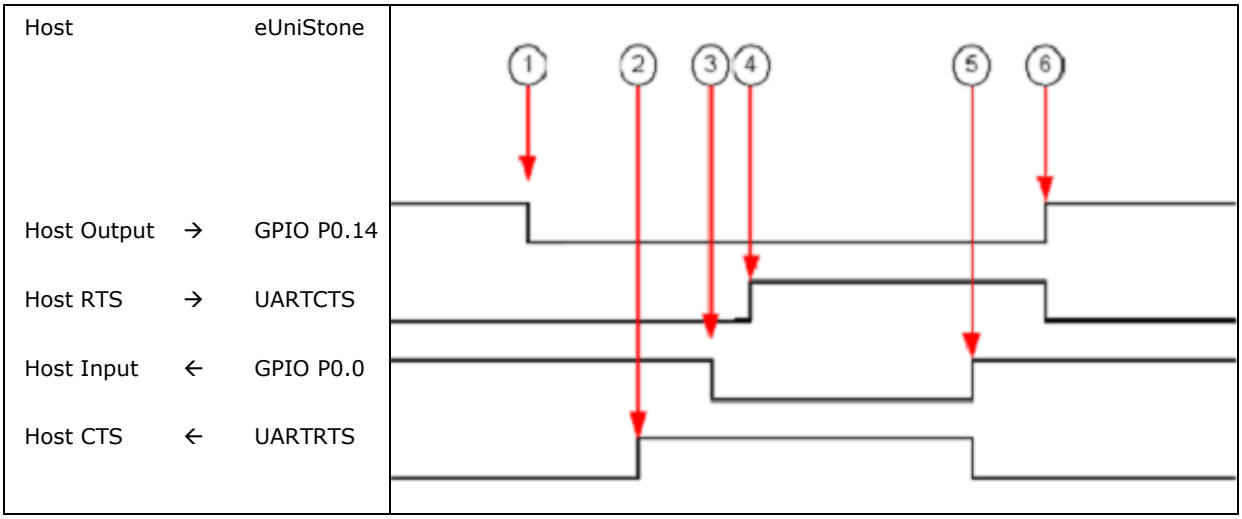
Host		eUniStone	
Host Output	→	GPIO P0.14	
Host RTS	→	UARTCTS	
Host Input	←	GPIO P0.0	
Host CTS	←	UARTRTS	

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

4.5.2 Host Initiates Low Power Mode Entry, eUniStone Initiates Exit

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

Table 5. Host Initiates Low Power Mode Entry, eUniStone Initiates Exit.



1. The host allows eUniStone to enter low power mode
2. eUniStone enters low power mode
3. eUniStone allows the host to enter low power mode
4. The host enters low power mode
5. eUniStone 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 permanently by writing into the UART_Baudrate parameter of the BD_DATA.

The available UART baud rates are:

Table 6. Available UART baud rates.

UART Baud Rate (bauds/s)	Module baud Rate (bauds/s)	Configuration value for BD_DATA
9600	9615	00
19200	19230	01
38400	38461	02
57600	57522	03
115200	115044	04
230400	230088	05

UART Baud Rate (bauds/s)	Module baud Rate (bauds/s)	Configuration value for BD_DATA
460800	464285	06
921600	928571	07
1843200	1857142	08
3250000	3250000	09

Remark: The two highest baud rates are not supported using the USB/UART bridge chip on the USB dongles that are provided for evaluation of the module.

Procedure to change UART Baud Rate:

1. Enter production mode.
   --> AT+JPRO=1
   <-- OK
2. Read the module specific BD_ADDR and OSC_Trim values and store the values in the host or on the tester.
   --> AT+JRBD
   <-- +RRBDRES= E4 2B 05 19 03 00 00 00 8A 00 80 BA 8C 01 BF 02 06 7E 98 1F 59 87 40 FA 12 18 80 00 03 04 05 06 44 0C 10 02 07 00 01 10 24 04 F2 F8 FE 04 D6 01 06 00 00 00 00 00 00 00 00 00
   e.g. bdAddr=0x000319052BE4 ... oscTrim=0x01D6
3. Change UART baudrate using the "Change BD_Data" command by specifying the corresponding configuration value uartBaudrate in BD_DATA. The original bdAddr and oscTrim values on the module must also be given in the BD_DATA parameter.
   e.g. Change baudrate to 921600 (index uartBaudrate=07), on default BD_DATA.
   -->AT+JCBD = XX XX XX XX XX XX 00 00 8A 00 80 BA 8C 01 BF 02 06 7E 98 1F 59 87 40 FA 12 18 80 00 03 04 05 06 44 0C 10 02 07 00 01 10 24 04 F2 F8 FE 04 ZZ ZZ 06 00 00 00 00 00 00 00 00 00

   (XXXXXXXXXXXX = BD address, ZZZZ = Osc_Trim)
   <-- OK

Note: The BD_DATA parameter of the AT+JCBD command is composed in the "LSB first" format.

4. Exit production mode.
   --> AT+JPRO=0
   <-- OK
5. SW Reset
   --> AT+JRES
   <-- ROK (sent with new baud rate, typically 70ms after AT+JRES)

The configuration will be written to EEPROM after the "AT+JCBD" command. The EEPROM access will be terminated before the "OK" response. **Power off during that time may corrupt the EEPROM data.**

After this configuration, the module will always use the new baud rate.

4.7 Data Flow Control

eUniStone 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 Accessing EEPROM

Power dropouts or HW reset during an EEPROM access can cause inoperability of the module.

The EEPROM contents are read at startup, i.e.

- after power up
- when leaving the HW reset state
- after a SW reset

The EEPROM access is terminated when the ROK is received by the host.

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

1. AT+JDOI (Download Image): Enables writing new image content to the EEPROM when <CMD><CR><LF> is sent. OK is received as a confirmation for the command, send whole image byte wise as data and wait for +RDOICNF to continue. After a SW reset all parameters are updated in RAM.
2. AT+JCBD (change BD_DATA): It writes the BD_DATA section of the EEPROM when <CMD><CR><LF> is sent, when receiving OK after AT+JRES command all parameters are updated in RAM.
3. AT+JRTD (Remove Trusted Device). It has erased the associated BD_ADDR and link-key from the EEPROM, when OK is received.
4. AT+JCCR (Create Connection Request): The resulting link key will be written into EEPROM (both in Dev A and Dev B), security level is not stored. Wait for +RCCRCNF to continue.
5. AT+JRES (RESet): The content of EEPROM is read and loaded into RAM. The EEPROM access is finished when the ROK is received.

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

4.10 Security Mode

Security mode 4 is used when remote device also supports Secure Simple Pairing (SSP). The three pairing procedures in Security mode 4 are;

- Just works
- Numerical comparison
- Passkey entry

The procedure used depends on the input/output capabilities of the connecting devices. The second and third procedure below requires user interaction.

- Just works; requires no user interaction
- Numerical Comparison; a 6-digit number displayed and confirmed on both devices
- Passkey entry; a 6-digit number used to authenticate connection when one of the devices displays a passkey and the other device enters a passkey.

Security mode 2 is used when a legacy device (not able to use SSP) is connecting to PBA31309. The pairing procedures for security mode 2 are;

- No pin (connect without pairing)
- Pin entry; pin entry if variable pin is used
- Just works; automatically if fixed pin is used

The procedure used depends on the security settings in the two devices.

The parameter *simple pairing* in the +RPCI event indicates to the host if it is a legacy paring (security mode 2) or a secure simple paring in progress. The AT+JPCR command is used for both legacy and SSP (pin and passkey) as response to the +RPCI event.

Different types of devices has different supported input/output capabilities, the table below gives examples of a typical type of devices for each combination of input/output capabilities.

Table 7. Input and Output Capabilities.

Capability		
Input	**Output**	
0	0	Sensor
0	1	Display
1	0	Headset
1	1	Headset with display
2	0	Keyboard
2	1	Cellular phone

As already mentioned the input/output capabilities of the local and the remote device are used to determine the type of SSP that should be used. PBA31309 has fixed Security Level setting 3. The table below lists the different combinations of

input/output capabilities and the expected outcome of the SSP procedure between two SPP devices.

Table 8. Expected SSP procedure depending on input output capability.

Dev A			Dev B			
Input	**Output**		**Input**	**Output**		**SSP procedure**
0	0		0	0		Just works
0	0		0	1		Just works
0	0		1	0		Just works
0	0		1	1		Just works
0	0		2	0		Just works
0	0		2	1		Just works
0	1		0	0		Just works
0	1		0	1		Just works
0	1		1	0		Just works
0	1		1	1		Numerical comparison
0	1		2	0		Passkey entry
0	1		2	1		Numerical comparison
1	0		0	0		Just works
1	0		0	1		Just works
1	0		1	0		Just works
1	0		1	1		Just works
1	0		2	0		Just works
1	0		2	1		Just works
1	1		0	0		Just works
1	1		0	1		Numerical comparison
1	1		1	0		Just works
1	1		1	1		Numerical comparison
1	1		2	0		Passkey entry
1	1		2	1		Numerical comparison
2	0		0	0		Just works
2	0		0	1		Passkey entry
2	0		1	0		Just works
2	0		1	1		Passkey entry
2	0		2	0		Passkey entry
2	0		2	1		Passkey entry
2	1		0	0		Just works
2	1		0	1		Passkey entry
2	1		1	0		Just works
2	1		1	1		Numerical comparison

Dev A			Dev B			
Input	**Output**		**Input**	**Output**		**SSP procedure**
2	1		2	0		Passkey entry
2	1		2	1		Numerical comparison

See examples in section 5 for MSC session descriptions on SPP-AT level.

4.11 GPIO Indication of Connection Status

The GPIO pin P0.1 (pin E5) is used to indicate the connection status. P0.1 is HIGH when eUniStone device is connected and LOW when there is no connection. The transition from HIGH to LOW happens prior to sending the "+RDII" indication via UART. Hosts that cannot monitor the incoming data stream for the "+RDII" indication in stream mode can monitor P0.1 in eUniStone.

P0.1 is configured as input pin by default. To use this feature the host must send the AT command "*AT+JGPC=FFFD,0000,0000,0000,FFF*D" which configures P0.1 as an output pin.

5 Example AT Commands and Responses

In all Message Sequence Charts (MSC) below are Dev A and Dev B PBA31309 except in the legacy device example when a PBA31308/2 is used as legacy device. When connecting to a device, e.g. a mobile phone, the GUI will prompt the user to input e.g. PIN or push OK depending on the pairing procedures supported by the mobile phone.

In the below example tables the Message Sequence Charts marks messages from Host to eUniStone with a "C<" for commands. An "R>" for response marks messages from eUniStone to Host

Table 9. Example of AT Commands and Responses.

Example 1: Device Discovery, Extended Inquiry = 0 (only friendly name from EIR devices)					
Dev A			**Dev B**		
Seq No.	**Direction R>/C<**	**Command /response**	**Seq No.**	**Direction R>/C<**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JDIS**=3
			4.	R>	**OK**
			5.	C<	**AT+JRLS**=04,11,1101,Serial port,01,000000
			6.	R>	**OK**
			7.	C<	**AT+JSLN**=04,devB
			8.	R>	**OK**
9.	C<	**AT+JDDS**=0			
10.	R>	**OK**			
11.	R>	**+RDDSRES**=0003199E8B25,devB,000000			
12.	R>	**+RDDSRES**=90C11566186C,Xperia arc S Erik,58020C			
13.	R>	**+RDDSRES**=549B1282F47C,,5A020C			
14.	R>	**+RDDSCNF**=0			

Example 2: Device Discovery, Extended Inquiry = 2 (Device name from all found devices)					
Dev A			**Dev B**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JDIS**=3
			4.	R>	**OK**
			5.	C<	**AT+JRLS**=04,11,1101,Serial port,01,000000
			6.	R>	**OK**
			7.	C<	**AT+JSLN**=04,devB
			8.	R>	**OK**
9.	C<	**AT+JDDS**=2			
10.	R>	**OK**			
11.	R>	**+RDDSRES**=0003199E8B25,devB,000000			
12.	R>	**+RDDSRES**=90C11566186C,Xperia arc S Erik,58020C			
13.	R>	**+RDDSRES**=549B1282F47C,GT-I9100,5A020C			
14.	R>	**+RDDSCNF**=0			

Example 3: Service Discovery (Dev B registers 16 bits uuid, Dev A searches for 16 bits uuid)

Dev A			**Dev B**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JSEC**=4,1,04,1111,0,0
			4.	R>	**OK**
			5.	C<	**AT+JDIS**=3
			6.	R>	**OK**
			7.	C<	**AT+JRLS**=04,11,1101,Serial port,01,000000
			8.	R>	**OK**
			9.		**AT+JSLN**=04,devB
			10.		**OK**
11.	C<	**AT+JSEC**=4,1,04,1111,0,0			
12.	R>	**OK**			
13.	C<	**AT+JSDS**=0003199E8B25,04,1101			
14.	R>	**OK**			
15.	R>	**+RSDSRES**=Serial port,01			
16.	R>	**+RSDSCNF**=0			

Example 4: Service Discovery (Dev B registers 128 bits uuid, Dev A searches for 16 bits uuid and 128 bits uuid)					
Dev A			**Dev B**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JSEC=**4,1,04,1111,0,0
			4.	R>	**OK**
			5.	C<	**AT+JDIS**=3
			6.	R>	**OK**
			7.	C<	**AT+JRLS**=32,11,000011010000 1000800000805f9b34fb,Serial port,01,000000
			8.	R>	**OK**
			9.	C<	**AT+JSLN**=04,devB
			10.	R>	**OK**
11.	C<	**AT+JSEC**=4,1,04,1111,0,0			
12.	R>	**OK**			
13.	C<	**AT+JSDS**=0003199E8B2532,0 0001101000010008000005f 9b34fb			
14.	R>	**OK**			
15.	R>	**+RSDSRES**=Serial port,01			
16.	R>	**+RSDSCNF**=0			
17.	C<	**AT+JSDS**=0003199E8B25,04, 1101			
18.	R>	**OK**			
19.	R>	**+RSDSRES**=Serial port,01			
20.	R>	**+RSDSCNF**=0			
Example 5: Secure Simple Pairing, Just works (Dev A connects to Dev B, no input/output capabilities for any of the devices)					
Dev A Security mode 4			**Dev B Security mode 4**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JSEC=**4,1,04,1111,0,0
			4.	R>	**OK**
			5.	C<	**AT+JDIS**=3

			6.	R>	**OK**
			7.	C<	**AT+JRLS**=04,11,1101,Serial port,01,000000
			8.	R>	**OK**
			9.	C<	**AT+JAAC=**1
			10.	R>	**OK**
			11.	C<	**AT+JSLN**=04,devB
			12.	R>	**OK**
13.	C<	**AT+JSEC=**4,1,04,1111,0,0			
14.	R>	**OK**			
15.	C<	**AT+JCCR**=0003199E8B25,01			
16.	R>	**OK**			
			17.	R>	**+RSLE**
18.	R>	**+RSLE**			
			19.	R>	**+RCOI=0003199E8B00**
20.	R>	**+RCCRCNF**=500,0000,0			
			21.	R>	**+RCCRCNF=**500,1101,0

Example 6: Secure Simple Pairing, Passkey entry (Dev A connects to Dev B and DevA has input capability keyboard but no output capability, Dev B has input and output capability)

Dev A Security mode 4			**Dev B Security mode 4**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JSEC=**4,1,04,1111,2,1
			4.	R>	**OK**
			5.	C<	**AT+JDIS**=3
			6.	R>	**OK**
			7.	C<	**AT+JRLS=04,11**,1101,Serial port,01,000000
			8.	R>	**OK**
			9.	C<	**AT+JAAC**=1
			10.	R>	**OK**
			11.	C<	**AT+JSLN**=04,devB
			12.	R>	**OK**
13.	C<	**AT+JSEC=**4,1,04,1111,2,0			
14.	R>	**OK**			
15.	C<	**AT+JCCR**=0003199E8B25,01			
16.	R>	**OK**			

17.	R>	**+RPCI=**0003199E8B00,1			
			18.	R>	**+RPNE=**716986
19.	C<	**AT+JPCR=06,716986**			
20.	R>	**OK**			
21.	R>	**+RSLE**			
			22.	R>	**+RSLE**
			23.	R>	**+RCOI=0003199E8B00**
			24.	R>	**+RCCRCNF=500,1101,0**
25.	R>	**+RCCRCNF=500,0000,0**			
Example 7: Secure Simple Pairing, Numerical comparison (Dev A connects to Dev B and both has input and output capabilities)					
Dev A Security mode 4			**Dev B Security mode 4**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JSEC=**4,1,04,1111,2,1
			4.	R>	**OK**
			5.	C<	**AT+JDIS**=3
			6.	R>	**OK**
			7.	C<	**AT+JRLS=04,11,**1101,Serial port,01,000000
			8.	R>	**OK**
			9.	C<	**AT+JAAC**=1
			10.	R>	**OK**
			11.	C<	**AT+JSLN**=04,devB
			12.	R>	**OK**
13.	C<	**AT+JSEC=**4,1,04,1111,2,1			
14.	R>	**OK**			
15.	C<	**AT+JCCR**=0003199E8B25,01			
16.	R>	**OK**			
17.	R>	**+RUCE=**576351			
			18.	R>	**+RUCE=**576351
19.	C<	**AT+JUCR=**1			
20.	R>	**OK**			
			21.	C<	**AT+JUCR=**1
			22.	R>	**OK**
23.	R>	**+RSLE**			

			24.	R>	**+RSLE**
			25.	R>	**+RCOI=**0003199E8B00
26.	R>	**+RCCRCNF=**500,0000,0			
			27.	R>	**+RCCRCNF=**500,1101,0
Example 8: Legacy pairing (Dev A has in- and output capabilities)					
Dev A Security mode 4			**Dev B Security mode 3 (Legacy device)**		
Seq No.	**Direction**	**Command /response**	**Seq No.**	**Direction**	**Command /response**
1.	R>	**ROK**			
			2.	R>	**ROK**
			3.	C<	**AT+JSEC=**3,1,1,04,1111
			4.	R>	**OK**
			5.	C<	**AT+JDIS**=3
			6.	R>	**OK**
			7.	C<	**AT+JRLS**=1101,11,Serial port,01,000000
			8.	R>	**OK**
			9.	C<	**AT+JAAC**=1
			10.	R>	**OK**
			11.	C<	**AT+JSLN**=04,devB
			12.	R>	**OK**
13.	C<	**AT+JSEC=**4,1,04,1111,2,1			
14.	R>	**OK**			
15.	C<	**AT+JCCR**=0003199E8B24,01			
16.	R>	**OK**			
			17.	R>	**RPCI=**0003199E8B00
			18.	C<	**AT+JPCR=04,1111**
			19.	R>	**OK**
20.	R>	**RPCI**=0003199E8B24,0			
21.	C<	**AT+JPCR=04,1111**			
22.	R>	**OK**			
23.	R>	**+RSLE**			
			24.	R>	**+RSLE**
			25.	R>	**+RCOI=**0003199E8B00
26.	R>	**+RCCRCNF=**350,0000,0			
			27.	R>	**+RCCRCNF=**350,0

6 References

No./Name	Title	Source
1	Bluetooth specification v2.1+EDR	www.bluetooth.org

7 Terminology

A	
APPL	Application
AT	Attention (from Hayes command set)
B	
BT	Bluetooth
C	
CR	Carriage Return
CTS	Clear To Send
D	
DEC	DECimal
E	
eBMU	BlueMoon(TM) Universal with Embedded SPP application (v2.1 means that it supports v2.1+EDR features)
eUniStone	Bluetooth Module using the eBMU chip for implementation of the Serial Port Profile
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

Terminology

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